**Statistical Machine Learning (F20ML/F21ML) Coursework**

This assessment is worth 40% of your course mark

**Due: 3:30pm, Thursday 16th November 2023 (Week 10)**

**Overview**

This assessment aims to assess the following:

* Your understanding of several commonly used machine learning models.
* Your ability to apply these models for solving real-world problems
* Your ability to perform systematic evaluations on these models when applied to problems and gain insights and critical thinking.

Please read through the following important points before you begin:

**This is assessed coursework**. You are allowed to discuss this assignment with other students, but you should not copy their work, and you should not share your own work with other students. Plagiarism is not acceptable. Heriot-Watt’s plagiarism policy is here

<https://www.hw.ac.uk/students/studies/examinations/plagiarism.htm> We will be carrying out automated plagiarism checks on both code and text submissions.

**Special note for re-using existing code or large language models.** If you are re-using code that you have not yourself written, then this must clearly be indicated, making clear which parts were not written by you and clearly stating where it was taken from. If your code is found elsewhere by the person marking your work, and you have not mentioned this, you may find yourself having to go before a disciplinary committee and face grave consequences.

**Late submission and extensions.** Late submissions will be marked according to the university's late submissions policy, i.e. a 30% deduction if submitted within 5 working days of the deadline, and a mark of 0% after that. The deadline for this work is not negotiable. If you are unable to complete the assignment by the deadline due to circumstances beyond your control (e.g. illness or family bereavement), you should complete and submit a mitigating circumstances application: <https://www.hw.ac.uk/students/studies/examinations/mitigating-circumstances.htm>

**Detailed Description**

**1. Quick Summary**

In this assessment, you will build different machine learning algorithms for classifying a given dataset. You will write one report and submit a Jupyter Notebook file with necessary Python files to present your findings and results. You will complete this project **individually** **(no teamwork)**.

The structure of your report and Jupyter Notebook file should follow the order of the questions with appropriate subsection titles. For any clarifications on this assessment, please ask **Dr Wei Pang** ([w.pang@hw.ac.uk](mailto:w.pang@hw.ac.uk), Edinburgh Campus), **Dr William Yoo** ([w.yoo@hw.ac.uk](mailto:w.yoo@hw.ac.uk), Malaysia Campus), or **Dr Hadj Batatia** ([h.batatia@hw.ac.uk](mailto:h.batatia@hw.ac.uk), Dubai Campus)

**2. Programming Language**

You will need to use Python to complete the coursework, and you will use Jupyter Notebook to run your experiments, save and present your experimental results.

**3. Detailed Tasks- Detecting Phishing Website**

**Data Source**: [https://archive.ics.uci.edu/ml/datasets/phishing+websites#](https://archive.ics.uci.edu/ml/datasets/phishing+websites)

**What you need to do**:

You should use different machine learning algorithms to classify the dataset, present your findings and results in the report along with the Python source code based on your work, and you must cover the following contents in the report:

1. A brief description of the above data set, including the format of data, and how you load and process the data with Python (no more than 200 words). [2 marks]
2. Using Exploratory Data Analysis (<https://en.wikipedia.org/wiki/Exploratory_data_analysis>) to investigate the dataset and report anything noteworthy. Please use figures or tables to summarise the dataset statistically and concisely. (not more than 300 words). [3 marks]
3. Write your own Python program to implement **a Perceptron** for classification from scratch. The classification accuracy should be at least 65% on the dataset. [4 marks]
4. Finetune the hyperparameters of your **Perceptron** algorithm to further improve the learning performance and make performance comparisons under different settings. You should use training/validation/test dataset split. [3 marks]
5. Instead of training/validation/test split, finetune your **Perceptron** with 10-fold cross-validation to perform your evaluation. The following evaluation metrics need to be reported: Confusion Matrix, Accuracy, F-score, ROC curve. [6 marks]
6. Use the following algorithms instead of **Perceptron** to perform the same experiments, hyperparameter tuning, and evaluation as in Question 5: Decision Tree, Neural Network, Naïve Bayes, and KNN.
   * For F20ML students, **KNN** needs to be implemented from scratch, and you cannot use any machine learning libraries to implement KNN. You can use any machine learning libraries or existing software to implement other algorithms in this question, and you can use any variants of the above algorithms you deem fit.
   * For F21ML students, **both KNN and** **Neural Network** need to be implemented from scratch, and you are not allowed to use any machine learning libraries to implement KNN or Neural Network, but you can use machine learning libraries to implement other algorithms in this question.
7. Provide details of how you use the libraries and software and how you further configure the algorithms to improve their learning performance. For F20ML, you should detail how you implement and improve your KNN. For F21ML, you should also detail how you implement and improve your KNN and Neural Network. [10 marks]
8. Make systematic and fair comparisons on the performance of the algorithms. You should use tables and figures to present your results. [8 marks]
9. Analyse the results and draw conclusions based on the experimental results that you have produced. [4 marks]

**Notice and Tips:**

* In summary, F20ML students need to implement Perceptron and KNN from scratch. F21ML students need to implement Perceptron, KNN, and Neural Networks from Scratch.
* Please do not use any existing machine learning libraries (e.g. scikit-learn, AutoKeras) or off-the-shelf software when implementing and improving the Perceptron algorithm in Questions 3 and 4 and the KNN and neural network (for F21ML students) in Question 6. They have to be implemented by your own programs. **Otherwise, zero marks may be awarded** for Questions 3 and 4, and lower marks will be awarded for Question 6.
* You can use non-machine learning libraries such as NumPy to assist your implementation.
* You can use libraries and software for performing train/val/validation, cross-validation, and evaluation in Questions 4, 5 and 6.
* Some machine learning algorithms may have random elements, which means they may get different results in each run. Whenever necessary, you should use statistical analysis on the results of multiple trials of the same algorithm.
* Consider when comparing the performance of different algorithms on the same dataset, how can you make the comparison as fair as possible?
* Please save experimental results (including the figures generated from the results) in the Jupyter Notebook file as evidence of your work.

**4. Submission**

You should submit the following items to CANVAS:

1. **Your report** in PDF format. Your report should
   * Include “**The declaration of authorship” as the first page.** Note: No marks will be awarded without the declaration of authorship.
   * Be no more than 8 pages in length (max of 3000 words, not including the declaration of authorship and references). You should take this into account when planning your experiments. If you have more results than you have space for, then pick the results that you think are most insightful and briefly mention other experiments you carried out.
   * Be written in Arial, or a similar font, with a minimum font size of 12.
   * Include useful references to the wider literature. For instance, you might use references to books or papers to justify particular implementation or hyperparameter choices, or you could compare your findings to those reported elsewhere. Use standard referencing styles for this.
2. **Python source code files**.
   * All source code files should be compressed as a single zip file and submitted to CANVAS. The zip file should be named as “F2nML\_H0XXXXXXX\_FirstName\_FamilyName.zip”, where F2nML stands for F20ML or F21ML, H0XXXXXX stands for your student ID number (H number). FirstName and FamilyName stand for your first name and family name, respectively.
   * The source code files should include a Jupyter Notebook file named as “F2nML\_H0XXXXXXX\_YourFirstName\_YourFamilyName.ipynb”.
   * The Perceptron implementation should be written in a Python file named as “perceptron.py”, and your Jupyter Notebook should be able to import this file and call the Perceptron model there.
   * The KNN and neural network (for F21ML students) implementation should be written in Python files, named as “knn.py” and “ann.py”, respectively. Similarly, they should be callable from the Jupyter Notebook file.
3. **Marking Criteria**

* We will not solely look at the learning performance of your models, but a classification accuracy of less than 65% is not acceptable.
* Instead, we will look at how you implement or configure your algorithms to make improvement on the learning performance.
* We will also look at clarity and reproducibility: how easy a third person can understand and reproduce your results based on your report and code. Specifically,
  + The quality of your report, including organisation, quality of writing, brevity.
  + The quality of your code, including quality of documentation, organisation, readability, efficiency, and ease of maintenance.
* We will look at the design and report of your experiments (including the appropriate use of tables and figures) and the depth of your analysis, including critical thinking.
* We will award going extra miles, for example, an advanced hyperparameter tuning algorithm was used to finetune the hyperparameters so that the machine learning model can achieve better performance.
* In specific circumstances (e.g., if we suspect a submitted coursework was not done by the student), we may call that student for a technical interview to assess their comprehension of the submitted work. If a student fails to demonstrate a good understanding of their submitted work, their mark may be reduced; in case of severe violations, we may report the case to school for disciplinary investigation.