UNIVERSITY OF LONDON

MSc Computer Science

Module: CSM010 – Applied Machine Learning (AML)

Coursework: April to June 2023 study session

Submission Deadline: Monday 10th July 2023 at 13.00 (GMT/BST)

- Please Note: You are permitted to upload your Coursework in the final submission area as many times as you like before the deadline. You will receive a similarity/originality score which represents what the Turnitin system identifies as work similar to another source. The originality score can take over 24 hours to generate, especially at busy times e.g. submission deadline.

- If you upload the wrong version of your Coursework, you are able to upload the correct version of your Coursework via the same submission area. You simply need to click on the 'submit paper' button again and submit your new version before the deadline.

  In doing so, this will delete the previous version which you submitted and your new updated version will replace it. Therefore your Turnitin similarity score should not be affected. If there is a change in your Turnitin similarity score, it will be due to any changes you may have made to your Coursework.

- Please Note, when the due date is reached, the version you have submitted last, will be considered as your final submission and it will be the version that is marked.

- **Once the due date has passed, it will not be possible for you to upload a different version of your assessment.**

Therefore, you must ensure you have submitted the correct version of your assessment which you wish to be marked, by the due date.

Coursework Description

The aim of this coursework is to provide a hands-on, practical, assessment of your machine learning skills; to work creatively on a dataset of a real-world application; to define a learning problem, discuss data attributes, evaluate suitable learning algorithm(s)—analytically or through your implementation, and to present your findings and conclusions.

Background

This is your chance to investigate a real-world machine learning classification problem using an appropriate data set and techniques.

You will obtain some data, analyse it, select appropriate features using some of the techniques we have covered, and then train and evaluate a number of candidate machine learning algorithms suited to classification tasks, before choosing the one that performs the best on your data.

You can find a range of recommended datasets for machine learning classification problems in the UCI Machine Learning Repository and Kaggle. Many are categorised according to the type of machine learning task they are best suited.

Your approach to implementing a machine learning solution (or a predictive model) and your written analysis that describes your approach and design choices will form the basis of the assessment for this coursework.

Software Requirements:

First a note on requirements terminology. We will use the convention employed in ISO standards. Specifically, we use **shall** to refer to functionality that is mandatory to implement, while **should** refers to guidance or recommendation i.e. the implementation of the specified feature is not mandatory.

- Your source code **shall** be submitted in a Jupyter Notebook, with clear headings (see the written report software requirements) highlighting each component of your solution from data collection and pre-processing to final model construction, prediction, and evaluation of the results. The Jupyter notebook for your project can be created on Codio if you do not have Jupyter notebook installed on our local machine.

- Your source code **should** be well-structured using appropriate indentation and spacing to support visual readability.

- You **shall** place all source code and any other files or directories required for the software to run in the Master branch of your allocated github repository. You **should** use this github repository throughout development so that your commit history is visible to the marker.

- Your implementation **shall** consider all the software requirements detailed below, where appropriate to the final solution.

Written report and software requirements:

You **shall** present your code and written report using the following headings to clearly separate each step of the analysis. You **shall** consider all the points listed under each section as part of your implementation and written report so that you can best demonstrate your methodology and critical analysis skills.

1. **The data:**

- Describe your chosen dataset, why is it interesting, what real-world problem could be solved or how would it be useful to industry e.g. marketing, business intelligence, medicine, or molecular biology?
- Reference the source of the data, where did you find it?

- Discuss any issues with the data; is there any sampling bias or skew in the representation of the different attributes; missing values?
- Discuss the pre-processing steps taken to get the data in the form that can be used for machine learning; did you normalise, standardise, or scale any attributes in the data, if so, why was this necessary.

2. **Constructing and Selecting Features:**

- Separate the features from the target variable.
- Apply feature selection techniques (e.g. filter, or wrapper) and report on the selected features.
- Consider whether feature importance provides any additional benefit in narrowing down the best candidate features.

2. **Building ML algorithms:**

- You **shall** apply **at least three** candidate machine learning algorithms suited to classification tasks and identify their machine learning category, e.g. supervised, unsupervised, semi-supervised.
- Discuss the selection strategies for the candidate algorithms (what motivated your choice?).
- Find the best configuration for each model, consider the model design components e.g. number of features, size of the test and train set, max depth for decision trees etc.
- Perform model-specific optimisations and iteratively debug model as complexity is added. Consider using an appropriate

resampling technique when the dataset is small when evaluating the candidate algorithms e.g. cross validation.

- Discuss the selection strategies for searching for the best configuration (e.g. trial and error, grid search, random search etc.)

3. **Evaluating models and analysing the results:**

- Evaluate the classification performance (e.g. accuracy, F1, Precision, Recall, and confusion matrix) of the candidate ML models on the test data and interpret the results based on the information provided by the evaluation metrics. Is the precision and recall higher for some classes, did some models perform best in terms of recall or precision for particular classes.
- Discuss general model trade-offs (accuracy versus interpretability) of the chosen models and propose two models (e.g. features selected and ML classifiers) that performed the best overall with respect to the evaluation metrics and justify your choice.

Format of the report:

You must produce a report of 2,000 words (±10%). The cover page must show your name, student number, the word count of your report (excluding appendices), module title, and a suitable title.

You are encouraged to produce tables and figures that help convey maximum information within the word count; and help provide the relevant detail, e.g. summary of the data, features selected, and evaluation metrics. These can be presented in the appendix, and referenced in the main text of your report.

The report must be in digital format (not hand-written) and presented using either Arial 10 point or Times New Roman 11 point font for the main body of the text and 1.5 line spacing. Pages must have a minimum of 2.54 (1 inch) margins, i.e. MS Word 'normal' margins. IEEE referencing must be used when including references. The page number should be in the footer.

You are strongly recommended to upload it in PDF format where possible, especially if including tables or figures.

How to submit:

Work on your coursework using the repository allocated to you in the AML module github classroom. To claim your repository, please follow the submission link from the assessment section of the VLE or follow this link:

https://classroom.github.com/a/CkoQBcWI

Make sure you use the repository provided in the classroom throughout development, so that the examiners can track your progress and process cf. Development Style section on the marking rubric below.

Note:

Make sure that in addition to your source code you also include any other files or directories that are needed for your program to run.

To submit your work for marking, clone the version of the github repository that you wish to be considered by cloning it to Codio.

Go to the coursework assignment in Codio.

Open the terminal (using **Tools-> Terminal**) and type

```
>> git clone YOUR_GITHUBCLASSROOM_REPO_LINK
```

where YOUR_GITHUBCLASSROOM_REPO_LINK is the URL of your assigned GitHub project repository.

Enter the credentials of your GitHub account. Remember that you need to use a personal access token as your password.

Note

You must provide both:

- the GitHub repo with the final version of your coursework (and history)
- the Jupyter notebook for your project on Codio (for marking).

Both entries contribute to your mark for the implementation.

Marking Rubric.

Your mark will be determined according to the rubric available in the README.md file in Codio.

Assessment Criteria:

Please refer to Appendix C of the Programme Regulations for detailed Assessment Criteria.

Plagiarism:

This is cheating. Do not be tempted and certainly do not succumb to temptation. Plagiarised copies are invariably rooted out and severe penalties apply. All assignment submissions are electronically tested for plagiarism. More information may be accessed via: https://learn.london.ac.uk/mod/page/view.php?id=3214