MLDG - Machine Learning Dataset Generator (Pilot Findings)

For Final Year Open-Book Machine Learning Summative Assessment

Keith Quille
School of Enterprise Computing and Digital Transformation, TU Dublin
Tallaght, Dublin, Ireland
Keith.Quille@TUDublin.ie

Keith Nolan
School of Enterprise Computing and Digital Transformation, TU Dublin
Tallaght, Dublin, Ireland
Keith.Nolan@TUDublin.ie

Lidia Vidal-Meliá
Dept. of Business Administration and Marketing, Universitat Jaume I
Castelló, Spain
lvidal@uji.es

Brett A. Becker
School of Computer Science, University College Dublin
Dublin , Ireland
Brett.Becker@ucd.ie

ABSTRACT

Following the COVID-19 pandemic, many institutions are considering the continuation of open-book assessments. Students cite many reasons for preferences for open-book assessments, both held in proctored environments and assessments that can be held remotely. Despite the numerous benefits, open-book assessments impose a number of challenges for educators, which is compounded if the assessment has the option to be taken in class (proctored) and remotely simultaneously. The challenges not only exist around academic integrity but also exist from a student-centred viewpoint of fairness and the equality of the assessment.

This paper first presents the development of a tool for generating student-centred datasets (using the MLDG - Machine Learning Dataset Generator) which are unique but fair to allow for both in-class and remote summative assessment for a final-year Machine Learning course, simultaneously. Second, a study was conducted over a two-year period to trail the tool and assessment environment, where it reported positive findings from a student's viewpoint, academic integrity viewpoint and the workload from course instructors (as the tool also generates an information file to reduce grading time). The results are encouraging and provide evidence that MLDG and the assessment approach used to promote a fair assessment with academic integrity for Machine Learning students. This tool and assessment approach could also be applied in other subjects such as databases thus having value outside of Machine Learning courses for the Computing Education Research community.

KEYWORDS

Online exam; Open-book; Artificial Intelligence; Machine Learning

1 INTRODUCTION

Proctored closed-book assessments (CBA) have been traditionally adopted as a method to evaluate summative assessment as part of final-year undergraduate courses. These assessments expect students to prepare what questions might appear on the assessment and study accordingly. Arguments in favour of CBA include maintaining academic integrity, alignment with learning objectives, gauging students' comprehension, and testing their ability to reproduce the information in various forms within a given time frame. Whilst this practice could arguably be necessary for fields such as medicine in which quick decision-making is necessary, using CBE in other disciplines could inadvertently encourage rote, surface learning (lower-order questioning), disengagement with the material, and most importantly increase anxiety, particularly amongst neurodivergent students [3–7]. Generally, within the IT field, there is no need to memorise by heart information that could be searched for online which is also recognised practice by many in the industry, but rather the education should focus on encouraging analytical and logical skills, nurturing creativity, and motivating students to devise elegant solutions and testing those abilities accordingly. This paper describes the application of a tool that generates main datasets as part of an open-book exam in an Applied Machine Learning course, and students' attitudes and performance as a result of this assessment. This approach may also have benefits for assessment at second level [2, 11].

2 MLDG DEVELOPMENT

The development of MLDG consisted of three main components, the inputs, the system (where the processing occurs) and the final outputs, where these components are represented in . This section will discuss each component. In addition, the Python scripts which conduct the processing and generate the outputs (as well as sample inputs) can be found at .

2.1 Inputs

MLDG takes two elements as inputs. The first input is the list of all students taking the assessment. For our local context, the student IDs are used as an initial starting numeric value which forms as a salt key which is combined with the index of the student in the student list. This is then used as the seed value for the random function which allows for the replication of outputs. Any other assigned numeric values could also work for generating the seed values. An example of the CSV file format can be seen in .

A single dataset is also loaded for this component. The dataset used for the examples in this paper is the IBM HR Employee Attrition rate dataset[1]. The administrator of the system needs to input the attributes (column) names (this could also be automatically completed if the column names are in the CSV file via Pandas), as well as several optional values: - A missing value (a default missing value for later in the process in case no missing values and/or outliers exist); - The number of attributes that each student will be assigned; - The percentage of the dataset that each student will be randomly assigned

[1] https://www.kaggle.com/datasets/pavansubhasht/ibm-hr-analytics-attrition-dataset

2.2 Processing (The MLDG System)

It should be first noted that the following processing is done per student in the student list. The generated random number seed value (from the student ID and the salt function), is used to randomly select a subset of the original dataset. The size of the subset is based on the value specified by the system user. For our recent assessments students were assigned a subset 60% the size of the original dataset.

Next, each student sub-dataset is now assessed for errors and outliers, specifically for the assigned attributes. For our recent assessments, students were assigned four attributes. For students in this course as discussed in Section 3.2, three standard deviations were the identifier for outliers and missing values were any value not possible for that attribute range (in the majority of cases it was negative values).

If the students' four attributes contained at least one attribute with missing data or outliers, the dataset did not undergo any further processing and as was then saved as a unique student dataset, with the students ID as the csv file name. If the four attributes contained no errors or outliers, missing values and outliers are added to two randomly selected attributes. The size of the allocation of the missing data and outliers generated by MLDG varied in size randomly (see Table 1 for an example output), just as they would in real-world datasets.

2.3 Output

MLDG produces two outputs, the individual student datasets and the information file for assessment. MLDG produces a unique dataset per student, which is saved in a folder in the same directory as the tool. The file names are the student's ID Numbers.

Table 1 presents a sample Information file for assessment output. The first two rows are the high-level information, such as the original number of instances, the missing value that is in the dataset if any, the percentage of the dataset used per student file and the corresponding number of instances assigned to each student.

The following rows are the metadata from each student. These include (irrespective if missing data and/or outliers were present in the original subset or generated) the *Missing value* assigned to each student; *Attribute 1* assigned to each student and with that the number of missing values for that attribute, the number of outliers at two and three standard deviations for that attribute. As MLDG for this example assessment used four attributes, three of these were omitted from Table 1, due to space constraints, thus just the first randomly assigned attribute is listed. *Start Index* and *End Index* are the randomly assigned start and end indexes used for the dataset selection for each student, *Total missing* is the total number of missing values for each student, *Total Outliers* is the total number of outlier values for each student and *Total Attributes with Issues* is the number of attributes that have issues from a total of four.

3 MLDG IN PRACTICE

This study implemented MLDG over two years in a final year undergraduate course, Applied Machine Learning with four student cohorts. The rationale for selecting this course was that it included a final-year high-stakes open-book assessment. Other courses were considered (such as databases and data analytics, CS1 [8, 9] where the MLDG could be used), however, the final year Applied Machine Learning course was selected as it was the only course that at present employed open-book assessments. We acknowledge that any of these courses could have been selected for this study if these assessments were migrated to open book. MLDG could also have been used for closed-book proctored assessments however we aimed to not only obtain feedback for its use as a tool to generate datasets but also in the context of open-book assessment. The students sitting open-book assessments within this course consisted of both part-time (PT) and full-time (FT) students from 2021 and 2022. In total 52 students took the open book assessment (2021 n = 25 and 2022 n = 27). In addition, both cohorts had the option to sit the open-book assessment remotely and on campus. All student cohorts were in their final year of study at Hons Bachelor Degree level. Full-time students in this study were majoring in Computing with Software Development or Computing with Machine Learning and AI. Details on this course and the assessment components (where MLDG was used) are discussed in the following subsections.

3.1 Applied Machine Learning Course Overview

The Applied Machine Learning (AML) course descriptor [12] is based on the following learning outcomes: Apply data pre-processing and data exploration techniques in the context of the Machine Learning process; Demonstrate knowledge of Machine Learning techniques, their methods and application; Determine the Machine Learning techniques and methods for particular scenarios; Evaluate the models produced, using relevant performance metrics.

The programming language used is Python, with Jupyter Notebooks as the IDE, or Google CoLab which is a cloud-based Jupyter notebook [1]. Assessment is broken down into two in-class open-book assessments, which require students to explore a dataset and carry out appropriate data pre-processing techniques for the machine learning process (thus, perfectly suited for the use of MLDG (weighted at 25% of the course grade). The two assessments have a combined weighting of 50% with the end-of-term exam worth the remaining 50%. The course is delivered at NFQ level 8 (Honours Bachelor's Degree Level). The exam was also open-book where additional information can be found in the literature [10].

3.2 Assessment

The assessment since COVID was open book where prior to MLDG the instructor manually generated the individual datasets for the open-book assessment. This section describes the assessment where MLDG was applied, (the first student assessment weighted at 25%) which was administered approximately one-third of the way through the Applied Machine Learning course. The assessment which aimed to assess Learning Outcome One as described in Section 3.1, consisted of each student receiving a unique dataset (of similar context) with some pre-processing components automatically added, such as missing data and or outliers. The assessment itself can be found at [12].

Each student is randomly assigned a set of four attributes (as done in the MLDG); in addition, each student has a unique data-set based on the same attributes and attribute ranges. Students had to open the dataset, visualise each of the attributes, the student had to work with four attributes (from the total set of attributes which

Table 1: Information File for Assessment (using the IBM HR Employee Attrition rate dataset)

Instances	Attributes (inc class)	Missing Values	Number of Attributes	Percentage of Dataset Used	Number of Rows per Student
1470	25	-5	4	60%	882

ID Number	Surname	Missing Value	Attribute 1	Missing N	Two STD N	Three STD N	Start Index	End Index	Total Missing N	Total Outliers N	Total Attributes with Issues
123456	A	-2	DailyRate	114	0	0	430	1312	117	4	2
123457	B	-4	Distance From Home	220	47	0	582	1464	220	4	2
123458	C	-5	Distance From Home	0	51	0	93	975	0	8	1

vary from assessment to assessment depending on the base dataset used). For the specific four attributes, students have to identify and discuss (giving reason based on graphical and statistical data) any missing data or outliers (if any) for each of the four attributes. They need to document what they plan to do with the missing values and/or outliers. After that, they have to mark and clean the data for the four attributes (if applicable). From here students had to conduct attribute selection using appropriate techniques, to normalise or standardise the dataset, and finally save the final datasets (with the highest performing attributes).

The assessment has a time limit of two hours (with an additional 30 minutes for upload) following the same model as the open book examination that the research group developed [10]) and can be taken in class or remotely. The final Jupyter Notebook is uploaded to the University's virtual learning environment (VLE) both as a .ipynb file and as an HTML (or PDF) file where the latter is run through the University's plagiarism detection system (Urkund a Turnitin product[2]). For those students who are sitting the CA remotely, the instructor conducts a viva with 20% of them randomly selected. The instructor releases the student names after the CA, where each student will receive a 10-minute slot. The aim of the viva is an additional form of academic integrity to identify if the answer was indeed written by the student.

3.3 Student Experiences

All assessments took place during a scheduled lab time with the lecturer available in person in the lab. Students had the opportunity to take the assessment either in person in the lab or virtually with access to the lecturer through a Microsoft Teams meeting. Following the completion of the continuous assessments, all students were given the opportunity to give feedback in relation to the continuous assessment. Feedback was collected over a two-year period with 12 students responding in 2021 and 14 students responding in 2022 (an overall response rate of 43%). Students were asked questions in relation to if they had sat open-book assessments before, their motivation for choosing the modality of sitting the open-book assessment, the difficulty of the assessment compared to a traditional assessment, questions relating to the instructions, clarity, time pressures, etc., and finally students had the option to give both positive and negative feedback.

[2]https://www.ouriginal.com/

Of the 26 respondents when asked if they had previously sat an open-book assessment, 22(84%) had said "No", 1(5%) had said "Maybe" and 3(11%) had said "Yes". Students were asked what modality they would prefer if given the choice. The options were "Open-book In lab", "Open-Book Online" and "Either". Interestingly, 18 of the 26 respondents said they would prefer Open-book Online. Drilling down on this, students were asked how they actually sat the assessment, online or in-lab and there was an even split between those 18 students with 9 choosing either method. Those that sat the CA in the lab cited reasons such as "more focused atmosphere", "easier access to teacher" and was a "more official test" along with the fact they were on campus already. In relation to students' perception of the difficulty of the open-book CA compared to a traditional closed-book CA, 10 out of the 27 students indicated that the CA, was in a way, easier with some eluding that it was because they could look up small code fragments which they may have forgotten. Most of the students responded suggesting that it was the same or similar. 3 students indicated it was harder. Table 2 shows questions students were asked around clarity of instructions, process, question-wording, policy and procedures, and the lecturer's availability. As can be seen, most found the process either good or excellent, however, the time limit for the assessment was not as strong.

Table 2: Responses to academic fairness of the assessments: Poor, Fair, Average, Good, Excellent

	P	F	A	G	E
Clarity of instruction	0	1	1	4	21
Online process	0	1	1	7	18
CA Questions (ignoring difficulty)	0	1	2	9	15
CA time limit	2	1	3	8	13
Upload Procedure	0	1	3	7	16
Urkund Plagiarism	0	0	8	8	11
Viva for plagiarism purposes	0	1	5	9	12
Lecturer availability pre CA	0	0	0	4	23
Lecturer availability during the CA	0	0	0	3	24

Students were asked about the fairness of the CA. Four questions were asked with responses from Very unfair to Very fair. The option

of students to take the CA in class or virtually; The unique datasets used in the CA with unique errors and unique attributes; The use of random viva's for academic integrity; The time limit used in CA1.

One student responded to all of the above questions as "Very unfair" and so their responses were received with caution. One student found the use of random vivas was very unfair while 1 found it somewhat unfair. Three found the time limit somewhat unfair with one finding the use of unique errors and attributes somewhat unfair. Interestingly, 26 of the 27 students said the open-book assessment method should continue. The students were asked two final questions, any positive or negative feedback. In the positive feedback, students suggested that it was a fair assessment, reduced anxiety and allowed students to focus on the tasks at hand rather than trying to debug errors. Examining the disadvantages, students cited concerns such as lack of time be it for the CA or for time to upload the CA. Others criticised the validity of the fairness of the dataset with reasons that the datasets might not be evenly balanced. This comment was interesting as the instructor would have explained the process prior to the CA attempting to address such concerns.

3.4 Student Results

This section aims to compare student results prior to COVID (2018 and 2019 cohorts with traditional proctored assessment) with the 2021 and 2022 who have been exposed to MLDG . 2020 was not considered (as discussed in Section 3) as this was the year of COVID and MLDG was not deployed, rather the instructor manually developed the individual datasets. The results, number of students and standard deviation are presented in Table 3.

Table 3: Student grades

	2018	**2019**	**2021**	**2022**
Number of Students	53	41	25	27
Mean Grade	57.2%	55.5%	58.1%	73.0%
Standard Deviation	17.3	22.3	28.2	24.5
Skewness	-0.60	-0.34	0.00	-1.40

As the data was non-parametric (skew values in Table 3), a Kruskal-Wallis H test was conducted and indicated that there is a significant difference between the different groups, $X^2(3) = 13.58$, $p = .0035$, with a mean rank score of 67.3 for 2018, 65.44 for 2019, 70.9 for 2021, 100.31 for 2022 (Confidence Interval = 0.95). The Post-Hoc Dunn's test was then conducted using a Bonferroni corrected p values. Interestingly the only year that had a statistically significantly difference was 2022 (with a higher average performance) – 2018-2019, 2019 and 2021 did not have statistically significant differences (confidence interval = 0.95). In addition, for all years in this study, there were no academic integrity cases. Overall, the average result for students who took the exam in a lab was 75.84%, and for those who took the assessment remotely was 58.63% (this was for both years and for FT and PT mode). Interestingly, 52% of the students sat the assessment in-person. A Mann-Whitney U Test was conducted and a p-value of 0.0208, with a confidence interval of 0.95, where this is was statistically significant.

4 CONCLUSION

The article discusses a fair tool for assessing students' open-book exams in Machine Learning. It presents the advantages of the tool for both students and lecturers, including fairness, higher exam grades, lower levels of anxiety and stress, and no plagiarism detected. The tool is also versatile, being valid for both in-class and online assessments, and can be used for summative exams, formative assessments, or labs. Finally, this tool has been published openly, since the authors aim to facilitate the work of other lecturers as well as to improve teaching globally and for free. All in all, our results are encouraging and provide an indication that the open-book assessment tool developed in this paper promotes a more fair assessment for CS students and will be useful to a wider public in the education sector in the near future.

REFERENCES

[1] Ekaba Bisong. 2019. *Google Colaboratory*. Apress, Berkeley, CA, 59–64. https://doi.org/10.1007/978-1-4842-4470-8_7

[2] Katrina Falkner, Sue Sentance, Rebecca Vivian, Sarah Barksdale, Leonard Busuttil, Elizabeth Cole, Christine Liebe, Francesco Maiorana, Monica M McGill, and Keith Quille. 2019. An international benchmark study of k-12 computer science education in schools. In *Proceedings of the 2019 ACM Conference on Innovation and Technology in Computer Science Education*. 257–258.

[3] Keith Nolan and Susan Bergin. 2016. The Role of Anxiety when Learning to Program: A Systematic Review of the Literature. In *Proceedings of the 16th Koli Calling International Conference on Computing Education Research* (Koli, Finland) *(Koli Calling '16)*. ACM, New York, NY, USA, 61–70. https://doi.org/10.1145/2999541.2999557

[4] Keith Nolan, Susan Bergin, and Aidan Mooney. 2019. An Insight Into the Relationship Between Confidence, Self-efficacy, Anxiety and Physiological Responses in a CS1 Exam-like Scenario. In *Proceedings of the 1st UK & Ireland Computing Education Research Conference (UKICER)*. ACM, New York, NY, USA, Article 8, 7 pages. https://doi.org/10.1145/3351287.3351296

[5] Keith Nolan, Aidan Mooney, and Susan Bergin. 2015. Facilitating student learning in Computer Science: large class sizes and interventions. *International Conference on Engaging Pedagogy* (2015).

[6] Keith Nolan, Aidan Mooney, and Susan Bergin. 2019. An Investigation of Gender Differences in Computer Science Using Physiological, Psychological and Behavioural Metrics. In *Proceedings of the Twenty-First Australasian Computing Education Conference (ACE '19)*. ACM, New York, NY, USA, 47–55. https://doi.org/10.1145/3286960.3286966

[7] Keith Nolan, Aidan Mooney, and Susan Bergin. 2019. A Picture of Mental Health in First Year Computer Science. In *Proceedings of the 10th International Conference on Computer Science Education: Innovation and Technology* (Bangkok, Thailand) *(CSEIT'19)*. Global Science and Technology Fourm.

[8] Keith Quille and Susan Bergin. 2016. Does Scratch improve self-efficacy and performance when learning to program in C#? An empirical study. In *International Conference on Enguaging Pedagogy (ICEP)*.

[9] Keith Quille, Soohyun Nam Liao, Eileen Costelloe, Keith Nolan, Aidan Mooney, and Kartik Shah. 2022. PreSS: Predicting Student Success Early in CS1. A Pilot International Replication and Generalization Study. In *Proceedings of the 27th ACM Conference on on Innovation and Technology in Computer Science Education Vol. 1* (Dublin, Ireland) *(ITiCSE '22)*. Association for Computing Machinery, New York, NY, USA, 54–60. https://doi.org/10.1145/3502718.3524755

[10] Keith Quille, Keith Nolan, Brett A. Becker, and Seán McHugh. 2021. Developing an Open-Book Online Exam for Final Year Students. In *Proceedings of the 26th ACM Conference on Innovation and Technology in Computer Science Education V. 1* (Virtual Event, Germany) *(ITiCSE '21)*. Association for Computing Machinery, New York, NY, USA, 338–344. https://doi.org/10.1145/3430665.3456373

[11] Ethel Tshukudu, Sue Sentance, Oluwatoyin Adelakun-Adeyemo, Brenda Nyaringita, Keith Quille, and Ziling Zhong. 2023. Investigating K-12 Computing Education in Four African Countries (Botswana, Kenya, Nigeria, and Uganda). *ACM Transactions on Computing Education* 23, 1 (2023), 1–29.

[12] http://tiny.cc/hcai-ep-files. 2023. Link for files.