Automated Feedback to Students in Data Science Assignments: Improved Implementation and Results

Alessandra Galassi*
Pierpaolo Vittorini*
alexandra2g2016@gmail.com
pierpaolo.vittorini@univaq.it
Dep. of Life, Health and Environmental Sciences, University of L'Aquila
L'Aquila (AQ), Italy

ABSTRACT

The automated grading of assignments is a long discussed topic in the field of technology-enhanced learning. In such a large research area, the authors focused on the automated grading of assignments made up of a mix of commands (in R language), their output and comments (in natural language). In particular, the paper discusses several improvements on the automated feedback generated by a tool developed at the University of L'Aquila, to support the students during their study of the subject. The goals of the research are the implementation of a feedback that gives an explanation of the automated grading, also providing students with the causes of the mistakes and suggestions on how to correct them. Accordingly, we designed and developed an automated feedback, used by students during the current academic year to support their homework. We then collected the students' opinions through both standardised and ad-hoc questionnaires, so to evaluate the effectiveness of our proposal and identify the aspects to improve. The results highlight an increased engagement while performing the assessment, the usefulness of the feedback, as well as where the explanation was clear and where improvements are needed.

CCS CONCEPTS

• **Applied computing** → *E-learning*; **Interactive learning environments**; • **Human-centered computing** → *Interactive systems and tools*; *Empirical studies in HCI*; • **Social and professional topics** → **Computational science and engineering education**.

KEYWORDS

Technology-enhanced learning, automated feedback, automated grading, static code analysis

ACM Reference Format:
Alessandra Galassi and Pierpaolo Vittorini. 2021. Automated Feedback to Students in Data Science Assignments: Improved Implementation and Results. In *CHItaly 2021: 14th Biannual Conference of the Italian SIGCHI Chapter (CHItaly '21), July 11–13, 2021, Bolzano, Italy*. ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3464385.3464387

*Both authors contributed equally to this research.

CHItaly '21, July 11–13, 2021, Bolzano, Italy

ACM ISBN 978-1-4503-8977-8/21/06…$15.00
https://doi.org/10.1145/3464385.3464387

1 INTRODUCTION

The manual grading of assignments is a tedious and error-prone task, and the problem particularly aggravates when such an assessment involves a large number of students. The use of artificial intelligence can be useful to address these issues [13]: by automating the grading process, we can assist teachers in the correction and enable students to receive immediate feedback, thus improving their solutions before the final submission. In previous research, we approached the problem of the automated grading of assignments made up of commands, output and comments. We first introduced a distance between the correct solution and the solution given by a student. Then, to calculate such a distance and return a feedback to students, we implemented a system called UTS (*Acronym suppressed for anonymity*) that – among all functionalities – performs: (i) static code analysis for the commands and their output, (ii) natural language processing and machine learning to classify the comments as right or wrong [3, 4, 8, 21].

A point that recently focused our attention is the feedback that the tool returns to students after the automated grading. The scientific literature shows that it can play a fundamental role in the learning process because it may help the students to identify their strengths and weaknesses, as well as to target areas that need further work, encouraging their self-evaluation and increasing their engagement [15, 16, 20].

In such a context, the paper reports on our more recent research finalised to improve the automated feedback. Starting from the suggestions collected from a survey with students that used the initial implementation [10], we structured the improved feedback in terms of (i) correct commands, (ii) partially wrong commands (i.e., commands with a mistake either in call or in the passed data), (iii) completely wrong commands, (iv) missed commands and (v) missed/right/wrong comments. We then implemented such an improved feedback within the UTS system. Finally, we evaluated its impact from a manifold perspective, i.e., engagement, usefulness, clarity, way of use, so to find the strengths and weaknesses, as well as to prioritise the future work.

The paper is structured as follows. Section 2 discusses the application scenario and the foreseen educational impact. Section 3 summarises the related work and Section 4 presents the research objectives. Section 5 describes the novel feedback and its implementation. Section 6 discusses the study and the results, and Section

7 ends the paper by summarising the main results and presenting the future work.

2 APPLICATION SCENARIO AND EDUCATIONAL IMPACT

The course of Health Informatics in the degree course of Medicine and Surgery of the University of L'Aquila (Italy) has a specific topic regarding how to execute statistical analyses in R and how to correctly interpret the results into the corresponding clinical findings. The exercises and the final exam have the same structure: they start with the definition of the dataset and list the analyses and technical/clinical interpretations that should be performed. The analyses must be performed through R commands and can be both descriptive (e.g., mean, sd), inferential (e.g., t.test, wilcox.test) and for testing normality (e.g., shapiro.test). For the interpretation of the results, students must be able to understand e.g. if the test for normality suggests that the distribution should be considered normal or not, or if a test for hypothesis is statistically significant or not.

Let us consider the following dataset:

patient	before	after
1	211	181
2	200	210
3	210	196
4	203	200
5	196	167
6	191	161
7	190	178
8	177	180
9	173	149
10	170	119

The data regards a sample of 10 hypertensive patients (variable "patient") who receive an anti-hypertensive drug. We measure the systolic blood pressure before drug administration (variable "before") and measured also few hours after (variable "after").
You are required to:

(1) calculate the mean of the systolic blood pressure before and after the administration of the drug;
(2) verify if the systolic blood pressure can be considered as extracted from a normal distribution;
(3) comment on the result;
(4) verify if the systolic blood pressure has decreased due to the effect of the drug;
(5) comment on the result.

Submit as solution a text containing the list of R commands with the respective output, as well as your interpretation of the analyses.

Figure 1: A sample exercise

For instance, see Figure 1 and let us take into account the fourth point of the assignment. Since the systolic blood pressure is quantitative, and the same patients are measured before and after the treatment, the student (after a normality test) should use a paired t-test. Such a test is executed in R through the command:

```
1 > t.test(sbp$before, sbp$after, paired=TRUE)
```

which would return the following output:

```
1         Paired t-test
2
3 data:  sbp$before and sbp$after
4 t = 3.0992, df = 9, p-value = 0.01274
5 alternative hypothesis: true difference in means is not
      ↪ equal to 0
6 95 percent confidence interval:
7   4.8613 31.1387
8 sample estimates:
9 mean of the differences
10                   18
```

By looking at the p-value (see line 4 of the output), which is 0.01274, less than 0.05, the student should then conclude that the difference in systolic blood pressure is statistically significant, and therefore it should be caused by the effect of the drug. Such a conclusion is the solution to point 5 of the assignment.

The experience gained by the authors in correcting the assignments enabled the identification of the most common mistakes. With reference to the exercise above, students may not realize that the sample is paired or simply forgot to pass the `paired=TRUE` parameter. As a result, in such a case, the returned p-value would be different and thus leading to completely different conclusions. Further examples of mistakes are the use of the wrong variables in the commands, or – on a minor extent – a wrong import of the dataset.

Accordingly, we started the design of a structured feedback, useful to guide students in understanding their mistakes and revising their commands and comments. In our view, our system should check the student's solution, then provide an immediate feedback, and encourage him/her to continue testing and optimizing his/her preparation. The automated feedback should improve the students' learning experience by encouraging them to improve their solution iteratively until it is correct, and before the final submission of the homework. Moreover, we believe that students may also be better prepared to the exam and thus achieving higher grades.

It is worth noting that the approach we discussed above, besides its contextualisation to the R language and the health setting, can be applied to any course that includes data science assignments, whose solutions are made up of a sequence of commands, interleaved with comments that explain the meaning of the results, written in natural language.

3 RELATED WORK

Several solutions have been previously proposed to perform automated grading of short-text answers and code-snippet answers [5, 17]. For those addressing short-text answers, the common task is to assign either a real-valued score (e.g., from "0" to "1") or to assign a label (e.g., "correct" or "irrelevant") to a student response. These attempts have mainly focused on English, whereas our courses and exams are in Italian. Several existing approaches to short-answer grading rely on knowledge bases and syntactic analyses [14], vector-based similarity metrics [18], transformer-based architectures [9] and neural network classifiers [19]. For code-snippets questions, the automated assessment of student programming assignments was first attempted in the sixties [11] and has produced a large set of results [17]. Currently, the available tools can produce the correction automatically, semi-automatically (i.e., the teacher revises the correction) or manually [6, 7, 12].

As explained in Section 2, our specific problem is the automated grading of assignment with a mix of code, output and short-answers. Starting from the aforementioned literature, in [21], we introduced a general approach valid for assignments whose solutions can be represented as a list of triples containing the command, its output, and a possible comment. In the proposed approach, a solution provided by a student is compared with the correct solution given by the professor. Accordingly, we identify if a student

- provided a correct command that returned the correct output (see the first example in Section 2);
- provided a command with an error either in the call (e.g., a t.test command without the required paired=TRUE option) or in the passed data. In both cases, the returned output is different from that of the professor;
- missed the command;
- correctly/incorrectly interpreted the result of the analysis.

Based on these possibilities, we defined a distance between the two solutions, that represents the final grade: the largest the distance, the lower the grade and viceversa. The aforementioned approach is implemented in a tool with the following characteristics: it provides an automated grading of assignments, supports both the teacher in the correction and the students while revising their homework, uses static source code analysis for the code snippets and a supervised classifier based on sentence embeddings for the open-ended answers [21].

The tool focuses on assignments such as those discussed in Section 2, i.e., with solutions implemented as a set of R commands and comments written in the Italian language, and produces as output both the grade and the correction notes.

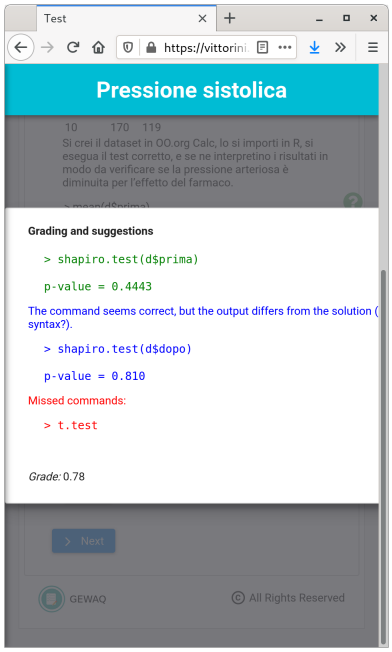
Before the research reported hereafter, the notes consisted in a list of: (i) the correct commands, marked in green with a "Correct" statement; (ii) the commands that appeared correct but returned a wrong output, marked in blue with a "The command seems correct, but the output differs from the solution"; (iii) the missed commands, marked in red; (iv) the student's comments in green or red (if right or wrong, respectively), with a commenting statement. Figure 2 shows such a sample feedback, as returned to a student.

In comparison with the literature, our proposal differs because the available solutions focus on Java, C, or C++, whereas we were interested in the R programming language; and the methods for the analysis of the comments are deployed for the English language, while our courses are taught in Italian.

4 RESEARCH OBJECTIVES

The goals of the research summarised in the paper consist of:

- showing the improvement of the automatic feedback provided to students by the assessment tool. The feedback was extended by providing more complete and detailed messages. The feedback now explains why the student got a certain grade through the display of the possible reasons for the mistakes, exploring a set of common issues that the student may have encountered, and providing suggestions on how to correct the mistake (Section 5);
- verifying through standardised and ad-hoc questionnaires, completed voluntarily by the students (Section 6):
  - if the system improved the student engagement;



English for the Italian sentence: *"Pressione sistolica"* : systolic blood pressure.

Figure 2: Student feedback – old implementation

  - if the feedback was useful (i) in general, (ii) to improve the final solution, (iii) to deepen the comprehension of the subject;
  - how the system was used, i.e., only to read the exercises, to check and submit the solution, or iteratively to refine the solution before the final submission;
  - if they suggested to use similar tools in similar subjects.

5 FEEDBACK

5.1 Design

In [10], we highlighted a set of improvements on the automated grading tool, and in particular on the feedback returned to students. To tackle them, we designed the new feedback as follows:

- for each command given by the student
  - if the command and its output are equal to a certain command and output contained in the correct solution, return a "Correct" feedback;
  - if the command is in the correct solution, but its output differs from the output of the correct solution. We first return the generic message "The command seems correct, but the output differs from the right solution", then we investigate the following two scenarios:
    - the student made a mistake in the command call, by checking if the student used
      - a wrong number of parameters: we either return the sentence "The parameter ... seems missing", or the sentence "The parameter ... seems not needed";
      - a wrong variable: we return the sentence "The variable ... does not seem correct";

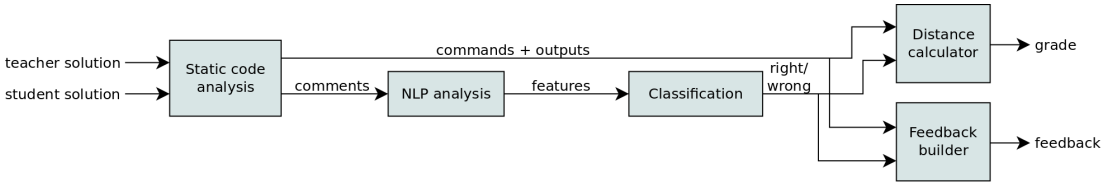


Figure 3: Tool architecture and feedback implementation

· a wrong boolean predicate for selecting a subset of rows: we return the sentence "The boolean predicate ... does not seem correct".

Depending on the case, a further message is added, that tries to suggest the student how to solve the mistake. For instance, in case of a t.test without the expected paired=TRUE parameter, we add the message "You should have used a paired test";

∗ if nothing above applies, we assume that the student incorrectly imported the dataset and the message "Please check if the data was imported correctly" is returned.

– the command is not in the correct solution. In this case:
  ∗ we first return the generic message "Wrong command". Then, we try to find in the correct solution a "similar" command, i.e., an improper choice of the command for calculating the central tendency or dispersion (referring to descriptive statistics) or the hypothesis testing (referring to inferential statistics). Depending on the case, a different message is returned. For example, if the student used the median instead of the mean, we return the message "Another command to calculate the central tendency is in the correct solution. Did you misunderstand the question or the variable type?".
– if the command requires a comment:
  ∗ if the comment is not present, we return the message "No comment was found";
  ∗ if the comment is present:
    · if the supervised classifier rated the comment as correct, we return the message "The interpretation of the analysis seems correct";
    · else, we return the message "The interpretation of the analysis seems incorrect".

• all commands of the correct solution that were not identified in the previous analysis, are listed as "Missed commands".

5.2 Implementation and Feedback interface

Figure 3 depicts the tool architecture. In detail, a solution is analysed as follows. First, the code is parsed, and the commands, outputs and comments are extracted. Then, the comments are first processed by an NLP module to extract all relevant features, then classified as either right or wrong (see [21] for details). Hence, the distance between the student's and the teacher's solution is calculated by the "Distance calculator" module so to estimate the grade. Furthermore, and differently from the previous implementation, the "Feedback builder" module produces the feedback for students according to the aforementioned design.

Figure 4 shows: on the left a possible solution to the exercise discussed in 1, on the right the corresponding feedback produced according to the aforementioned design. In the solution, the student omitted the commands to solve point 1 (i.e., he/she didn't issued two mean commands), executed correctly the shapiro.test, didn't give an interpretation to the normality tests, forgot the paired=TRUE option for the hypothesis testing, nevertheless interpreting correctly the (wrong) result. Accordingly, the tool recognised the two correct normality tests (first two green blocks), but it was unable to find their interpretation (the subsequent red block). It then found the t.test, but – given that the calculated p-value was different than the correct solution – the tool inspected the command call and found the missing paired=TRUE parameter. Hence, the tool reported such a problem in terms of the the three lines that close the blue block, the latest sentence suggesting how to get to the correct solution. Then, the tool automatically classified the comment given to the hypothesis testing as correct, as reported in the subsequent green block. In the last block (in red), the tool reported the missing commands. The feedback is then concluded with an estimation of the final grade.

6 STUDY

6.1 Study design

We conducted a study using data collected from two different cohorts, made up of students of the Medicine and Surgery course, from the 2019-2020 and 2020-2021 academic years (see Figure 5). The two cohorts used the old and new implementation of the automatic feedback, respectively.

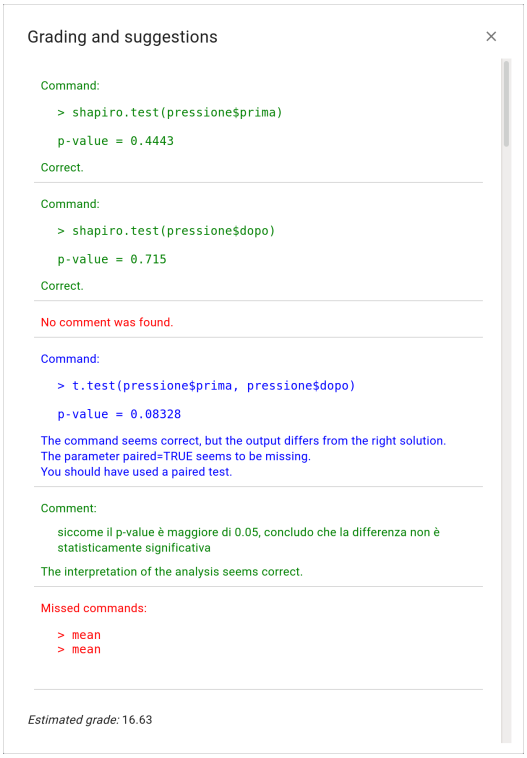
The two cohorts both compiled the User Engagement Assessment Scale (UEAS, [2]) and a question containing a general opinion on the feedback (see Appendix A). The 2020-2021 cohort also answered to two questionnaires. The first is structured in terms of expectation/experience on the following elements of the received feedback: (i) the usefulness of the feedback as a whole, (ii) clarity of the explanation for the incorrect commands, (iii) clarity of the explanation for the partially wrong commands, (iv) usefulness in solving the exercise (see Appendix B). The second contains questions regarding the impact of the feedback, how it was used and if they would recommend similar systems in similar subjects (see Appendix C).

The questionnaires were analysed as follows.

As for the UEAS, we scored the engagement as discussed in [2], then we calculated the average in each cohort.

As for the expectation/experience questionnaire, we followed a similar approach of that proposed by Albert & Dixon [1]. First, we calculated the mean of the expectation and experience, for each

```
> shapiro.test(pressione$prima)

        Shapiro-Wilk normality test

data:  pressione$prima
W = 0.92963, p-value = 0.4443

> shapiro.test(pressione$dopo)

        Shapiro-Wilk normality test

data:  pressione$dopo
W = 0.95392, p-value = 0.715

> t.test(pressione$prima, pressione$dopo)

        Welch Two Sample t-test

data:  pressione$prima and pressione$dopo
t = 1.8648, df = 14.026, p-value = 0.08328
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -2.699551 38.699551
sample estimates:
mean of x mean of y
    192.1     174.1

> # siccome il p-value e' maggiore di 0.05, la differenza non e'
    ↪ statisticamente significativa
```



English for the Italian words/sentences: (i) *"pressione"* : blood pressure; (ii) *"prima"* : before; (iii) *"dopo"* : after; (iv) *"siccome il p-value e' maggiore di 0.05, la differenza non e' statisticamente significativa"* : given that the p-value is larger than 0.05, the difference is not statistically significant.

Figure 4: Student feedback – new implementation: on the left, the solution; on the right, the feedback

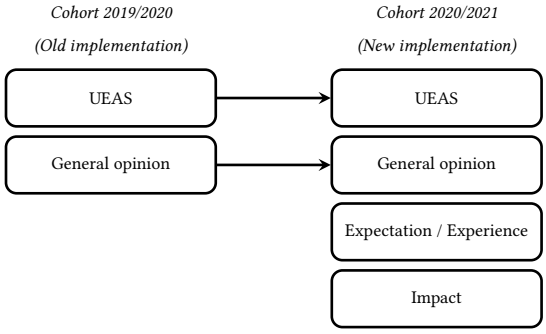


Figure 5: Study design

element. Then, we placed the results in a scatterplot (expectation on the x-axis, experience on the y-axis). As suggested in [1], elements in the top-right quadrant (i.e., good expectation and good experience) can be considered satisfactory; elements on the bottom-right quadrant (i.e., good expectation and low experience) need to be addressed with priority; elements in the top-left quadrants (i.e., low expectation and good experience) show a surprisingly good user experience; elements in the bottom-left (i.e., low expectation and experience) should be addressed as well, although with a lower priority.

Finally, all questions requiring a Likert-scale answer were analysed through averages, whereas, for the multiple-choice questions, we used frequency tables.

The inferential analyses were performed through t-tests or Wilcoxon tests, paired or not, depending on the type of variable (qualitative or quantitative), whether normally distributed or not, in case of paired or independent samples. In the results, when reporting the p-value, we added an index that clarifies the adopted method, i.e., w for the Wilcoxon test, t for the t-test, p for the paired version.

6.2 Results

6.2.1 Engagement (UEAS). 40 students of 19/20 cohort and 16 students of the 20/21 cohort answered the UEAS questionnaire. We observed an increased engagement, from 3.6/5 for the 19/20 cohort, to 4.2/5 for the 20/21 cohort, a difference that is statistically significant ($p_w = 0.002$).

6.2.2 General opinion. The question regarding the usefulness, quality and relevance of the available exercises was answered by 26 students of the 19/20 cohort, and 46 students for the 20/21 cohort. In both cases, the general opinion was rated as 4.7/5.

6.2.3 Expectation/experience. A total of 63 students answered to the expectation/experience questionnaire. Figure 6 summarises the analysis for all questions, i.e., general usefulness (USEFULNESS),

	Element	Expectation	Experience	p_{wp}
+	USEFULNESS	4.25	4.24	n.s.
•	CLARITY C.W.	3.55	3.47	n.s.
▲	CLARITY P.W.	3.55	2.98	0.00043
■	SOLVE	3.90	3.84	n.s.

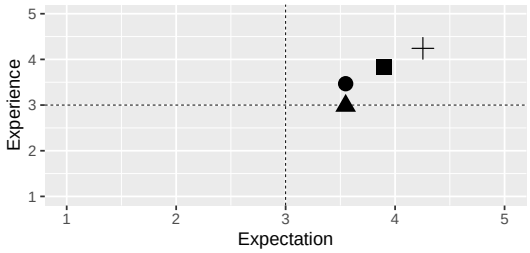


Figure 6: Summary of expectation/experience analysis

clarity of the feedback for the completely wrong commands (CLARITY C.W.), clarity for the partially wrong commands (CLARITY P.W.), and the usefulness for solving the exercise (SOLVE). The USEFULNESS, CLARITY C.W., and SOLVE elements are in the top-right quadrant, so they can be considered satisfactory; on the other hand, CLARITY P.W. – even if borderline – didn't meet the expectations.

6.2.4 Impact. The results show that the automatic feedback provided by the system was useful to students to understand their mistakes (30 students), to understand the correct statistical method to solve the problem (37 students), and to verify the preparation for the final exam (36 students). Furthermore, most of the students used the tool iteratively to improve their solutions (48 students). Only few of them used the tool before submitting the solution (12 students) or just to see the exercises (2 students). Finally, students suggested to use similar tools with a 4.7/5 rate.

6.3 Discussion

The analyses summarised above yield several interesting results.

The first one regards the increased engagement of the 20/21 cohort with respect to the previous one. This result supports our idea that a more detailed and explanatory feedback could raise more attention and participation in students. Nevertheless, the 19/20 cohort followed the lectures in presence, whereas the 20/21 online, and this factor is a clear bias. However, at this point of the research, we do not have two cohorts that can be compared without biases. Accordingly, we consider this result preliminary, needing for verification, but encouraging.

Regarding the analysis of the expectation/experience, three elements of the automated feedback (i.e., usefulness in general, clarity of the explanation for the completely wrong commands, and usefulness for solving the exercise) had average expectations/experiences very positive. On the other hand, the explanation for "the partially wrong commands" has to be improved. It is worth remarking these latter kind of errors are the most deceptive and ambiguous. The student knew which command had to be used, but introduced "something" wrong in the call (e.g., wrong variable, wrong import of data). Therefore, an explanation that leads to solve such a mistake must be very precise and specific to be effective. In other words, explaining a completely wrong command is somehow easy (e.g., the use of a median instead of a mean, can be easily explained in terms of a wrong choice of the central tendency indicator). Nevertheless, a command that appears correct, but returns a different output, may be caused by a multitude of factors, usually difficult to spot also by a teacher. Accordingly, both the initial identification of the possible cause, then the generation of an automated feedback that explains that specific mistake and suggests a way to solve the issue, is actually a difficult task to implement. However, improving this element of the feedback is now the priority of our research.

Finally, the fact that the majority of the students used the tool iteratively, as a guide (confirming or suggesting changes), refining the solution until the final submission, shows the central role that the tool played during the preparation to the exam.

It is worth noting that all data were collected through self-reported questionnaires, but the current pandemic situation did not allow to organise in-person usability testings.

7 CONCLUSIONS

The paper summarises the work done by the authors to design and implement an automated feedback that students may use during their assignments. The feedback is assembled by comparing the solution given by a student with the correct solution given by the professor. The identified mistakes are divided into different categories, and for each category an ad-hoc feedback is returned, that aims both at explaining the mistake, and also suggesting how to correct it.

To understand the quality of the automated feedback, we administered several ad-hoc questionnaires to students. The results of the analyses show that the tool in general was perceived as a good support for the preparation of the exam, both in terms of usefulness and clarity of the explanation. Nevertheless, with specific regards to the latter point, improvements are still necessary for the partially wrong commands.

Accordingly, the future work will articulate into manifold directions. The first is to allow flexible solutions: so far, the tool would not score adequately a student that arrives to the correct conclusions passing through alternative sequences of commands. A second improvement is to capture the case of a correct interpretation of a wrong sequence of commands: so far, the tool would consider such a solution as completely wrong, whereas it might be considered as partially correct. With specific regards to the feedback, we aim at improving the explanation of the partially wrong commands and supporting the propaedeuticity between commands. This latter point is also important for improving further the feedback. For instance, the wrong choice of an hypothesis testing may be caused by the absence of a normality test. So far, the tool only report it as a missing command. On the other hand, by checking

the propaedeuticity, we could relate such a missing command to the wrong choice of the hypothesis testing, thus providing a deeper explanation on the mistake and the right way to correct it. Finally, a comparison of the students' outcomes (with and without new feedback system) is planned with the forthcoming exams, so to substantiate the impact of the tool, as reported by the students.

REFERENCES

[1] William Albert and Eleri Dixon. 2003. Is this what you expected? The use of expectation measures in usability testing. In *Proceedings of the Usability Professionals Association 2003 Conference*. Scottsdale, AZ.

[2] Anna Maria Angelone and Pierpaolo Vittorini. 2019. A Report on the Application of Adaptive Testing in a First Year University Course. In *Communications in Computer and Information Science*, Vol. 1011. Springer Verlag, 439–449. https://doi.org/10.1007/978-3-030-20798-4{_}38

[3] Anna Maria Angelone and Pierpaolo Vittorini. 2019. The Automated Grading of R Code Snippets: Preliminary Results in a Course of Health Informatics. In *Proc. of the 9th International Conference in Methodologies and Intelligent Systems for Technology Enhanced Learning*. Springer.

[4] Angelo Bernardi, Carlo Innamorati, Cesare Padovani, Roberta Romanelli, Aristide Saggino, Marco Tommasi, and Pierpaolo Vittorini. 2019. On the design and development of an assessment system with adaptive capabilities. In *Advances in Intelligent Systems and Computing*. Vol. 804. Springer, Cham. https://doi.org/10.1007/978-3-319-98872-6{_}23

[5] Steven Burrows, Iryna Gurevych, and Benno Stein. 2015. The eras and trends of automatic short answer grading. *International Journal of Artificial Intelligence in Education* 25, 1 (2015), 60–117.

[6] Brenda Cheang, Andy Kurnia, Andrew Lim, and Wee-Chong Oon. 2003. On automated grading of programming assignments in an academic institution. *Computers & Education* 41, 2 (2003), 121–131.

[7] Kenneth M. Dawson-Howe. 1995. Automatic Submission and Administration of Programming Assignments. *ACM SIGCSE Bulletin* 27, 4 (12 1995), 51–53. https://doi.org/10.1145/216511.216539

[8] Giovanni De Gasperis, Stefano Menini, Sara Tonelli, and Pierpaolo Vittorini. 2019. Automated Grading Of Short Text Answers: Preliminary Results In A Course Of Health Informatics. In *ICWL 2019 : 18th International Conference on Web-Based Learning*. Springer. LNCS., Magdeburg.

[9] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Association for Computational Linguistics, Minneapolis, Minnesota, 4171–4186. https://doi.org/10.18653/v1/N19-1423

[10] Alessandra Galassi and Pierpaolo Vittorini. 2021. Improved feedback in automated grading of data science assignments. In *Advances in Intelligent Systems and Computing*, Vol. 1236 AISC. Springer, 296–300. https://doi.org/10.1007/978-3-030-52287-2{_}31

[11] Jack Hollingsworth. 1960. Automatic graders for programming classes. *Commun. ACM* 3, 10 (10 1960), 528–529. https://doi.org/10.1145/367415.367422

[12] David Jackson. 2000. A semi-automated approach to online assessment. In *Proceedings of the 5th annual SIGCSE/SIGCUE ITiCSEconference on Innovation and technology in computer science education - ITiCSE '00*. ACM Press, New York, New York, USA, 164–167. https://doi.org/10.1145/343048.343160

[13] John F. LeCounte and Detra Johnson. 2015. The MOOCs: Characteristics, Benefits, and Challenges to Both Industry and Higher Education. In *Handbook of Research on Innovative Technology Integration in Higher Education*. IGI Global.

[14] Michael Mohler, Razvan Bunescu, and Rada Mihalcea. 2011. Learning to Grade Short Answer Questions Using Semantic Similarity Measures and Dependency Graph Alignments. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1* (Portland, Oregon) *(HLT '11)*. Association for Computational Linguistics, Stroudsburg, PA, USA, 752–762. http://dl.acm.org/citation.cfm?id=2002472.2002568

[15] Ann Poulos and Mary Jane Mahony. 2008. Effectiveness of feedback: the students' perspective. *Assessment & Evaluation in Higher Education* 33, 2 (4 2008), 143–154. https://doi.org/10.1080/02602930601127869

[16] Erhel S. and Jamet E. 2013. Digital game-based learning: Impact of instructions and feedback on motivation and learning effectiveness. *Computers & Education* 67 (2013), 156 – 167. https://doi.org/10.1016/j.compedu.2013.02.019

[17] Draylson M. Souza, Katia R. Felizardo, and Ellen F. Barbosa. 2016. A Systematic Literature Review of Assessment Tools for Programming Assignments. In *2016 IEEE 29th International Conference on Software Engineering Education and Training (CSEET)*. IEEE, 147–156. https://doi.org/10.1109/CSEET.2016.48

[18] Md Arafat Sultan, Cristobal Salazar, and Tamara Sumner. 2016. Fast and easy short answer grading with high accuracy. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. 1070–1075.

[19] Chul Sung, Tejas Indulal Dhamecha, and Nirmal Mukhi. 2019. Improving Short Answer Grading Using Transformer-Based Pre-training. In *Artificial Intelligence in Education*, Seiji Isotani, Eva Millán, Amy Ogan, Peter Hastings, Bruce McLaren, and Rose Luckin (Eds.). Springer International Publishing, Cham, 469–481.

[20] Jaime Urquiza-Fuentes and J. Ángel Velázquez-Iturbide. 2013. Toward the effective use of educational program animations: The roles of student's engagement and topic complexity. *Computers & Education* 67 (September 2013), 178 – 192. https://doi.org/10.1016/j.compedu.2013.02.013

[21] Pierpaolo Vittorini, Stefano Menini, and Sara Tonelli. 2020. An AI-Based System for Formative and Summative Assessment in Data Science Courses. *International Journal of Artificial Intelligence in Education* (12 2020), 1–27. https://doi.org/10.1007/s40593-020-00230-2

A GENERAL OPINION

	1	**2**	**3**	**4**	**5**
1=Useless, 2=Not very useful, 3=Neutral, 4=Useful, 5=Extremely useful					
How do you evaluate the usefulness, quality and relevance of the exercises available on the formative assessment tool for exam preparation?					

B EXPECTATION/EXPERIENCE QUESTIONNAIRE

	1	**2**	**3**	**4**	**5**
1=Useless, 2=Not very useful, 3=Neutral, 4=Useful, 5=Extremely useful					
How did you expect the automatic feedback provided by the platform to be in general?					
How do you generally rate the automatic feedback provided by the platform?					
1=Not at all, 2=A little, 3=Neutral, 4=Very, 5=Completely					
For completely wrong commands, i.e. those highlighted in red, how much did you expect the feedback provided by the platform would be clear?					
For completely wrong commands, i.e. those highlighted in red, how clear did you find the feedback provided by the platform?					
For partially wrong commands, i.e. those highlighted in blue, how much did you expect the feedback provided by the platform would be clear?					
For the partially wrong commands, i.e. those highlighted in blue, how clear did you find the feedback provided by the platform?					
1=Surely not, 2=No, 3=Maybe, 4=Surely yes, 5=Yes					
Did you expect that the explanation of the feedback would allow you to solve the exercise correctly?					
Did the explanation of the feedback later allow you to solve the exercise correctly?					

C IMPACT

					Check all those that apply
The automated feedback provided by the platform allowed me to:					
1. Understand my mistakes					
2. Understand the correct statistical method to solve the problem					
3. Verify my preparation for the final exam					
					Select only one
How did you use the system?					
1. I only used it to see the exercises					
2. I only used it before submitting the solution					
3. I used it iteratively, to improve my solution before the final submission					
	1	**2**	**3**	**4**	**5**
					1=Surely not, 2=No, 3=Maybe, 4=Surely yes, 5=Yes
Would you recommend the use of automatic feedback systems of this type to prepare for similar exams?					