JUPYTER NOTEBOOKS FOR TEACHING, LEARNING, AND DOING DATA SCIENCE

Yannik Fleischer, Sven Hüsing, Rolf Biehler, Susanne Podworny, and Carsten Schulte
Paderborn University, Paderborn, Germany
Podworny@math.upb.de

We report on our work with students in our data science courses, focusing on the analysis of students' results. This study represents an in-depth analysis of students' creation and documentation of machine learning models. The students were supported by educationally designed Jupyter Notebooks, which are used as worked examples. Using the worked example, students document their results in a so-called computational essay. We examine which aspects of creating computational essays are difficult for students to find out how worked examples should be designed to support students without being too prescriptive. We analyze the computational essays produced by students and draw consequences for redesigning our worked example.

INTRODUCTION

Every day, we are surrounded by by-products of data science and especially artificial intelligence (AI) models created by using machine learning (ML) methods. These contents have emerged as new topics in today's school curricula (Bargagliotti et al., 2020; International Data Science in Schools Project Curriculum Team, 2019). Because of this new meaning of ML methods in our everyday lives, it is important for students to understand how ML algorithms are constructed and the roles that data and data explorations play in this context. This means that appropriate methods and tools are needed to make these topics accessible to students. An important feature a tool for this purpose needs to provide is the possibility of documenting insights into the creation of the AI tools and insights that arise from data analyses to re-examine them at a later time. In the Project Data science and Big Data in School (ProDaBi, www.prodabi.de/en) project, we are developing teaching modules, together with respective tools that students can use to get engaged with data and machine learning models. Findings by Fleischer et al. (2022) as well as by Hüsing and Podworny (2022) indicate that creating computational essays is a promising method to let students do data explorations and create data-based decision models while documenting them in the same document.

Besides programming and writing code, this approach explicitly integrates documenting and explaining a (programming-) procedure as well as its results. Hence, there is a need for a programming environment that makes it possible to split program code into small snippets while also integrating comments and pictures. For this purpose, we use Jupyter Notebooks (JNB) (Kluyver et al., 2016; Perez & Granger, 2015) because these allow combining Python-code-cells (for writing code) and text cells (for annotations and explanations). Because creating a computational essay is a complex task, students are offered a "working blueprint" to get an idea of how the process of applying an ML method for creating a data-based decision model can be documented appropriately. Using worked examples in this sense lets students work independently by providing them with related examples as a scaffold, from which the students can create their own products. Metaphorically speaking, students are encouraged to "glu[e] together" (Ko et al., 2011, p. 21:16) certain existing elements and features. To let worked examples appear as scaffolds for creating individual products, it must be determined which elements they should contain and what forms of assistance need to be provided.

Building on insights regarding the use of worked examples in order to create computational essays (Fleischer et al., 2022), in this paper, we want to analyze difficulties in creating computational essays guided by worked examples. We want to investigate how worked examples can be designed to support the creation of computational essays for data exploration and machine learning processes.

BACKGROUND

The Concept of a Computational Essay

The term, computational essay, was initially introduced by diSessa (2000, p. 185) as an essay consisting of programming code, producing content for the reader to play with, and integrated explanatory components such as text, hypertext, diagrams, and pictures. This format is described to be ideal for presenting scientific insights (especially for students) or instructing how to solve a task requiring programming and visualizations. McNamara (2019) supports this idea when arguing for the

In S. A. Peters, L. Zapata-Cardona, F. Bonafini, & A. Fan (Eds.), *Bridging the Gap: Empowering & Educating Today's Learners in Statistics. Proceedings of the 11th International Conference on Teaching Statistics (ICOTS11 2022), Rosario, Argentina.* International Association for Statistical Education. iase-web.org ©2022 ISI/IASE

need for technology to support a programmer in creating reproducible data analyses with a narrative and making the process accessible to others. Compared with other tools, reproducibility is a crucial benefit of computational essays when using Juptyter Notebooks (JNBs, or the mark-down language in R) (Kluyver et al., 2016; McNamara, 2019). Reproducibility, in this context, can be understood first as repeatability, and second as comprehensibility. In well-formed JNBs, all steps of creating the final product are documented in the correct order (the sequence of the code cells) so that the associated research process can be repeated with the same (or similar) data, reproducing the same (or analogous) results technically. Explanatory elements such as written comments and visualizations can be added next to the code cells. Ideally, these comments form a coherent narrative describing steps, explaining (statistical) arguments, and assessing results by considering the broader context. That way, the reader can comprehend the decisions and reasoning of the initial (programming-) process (McNamara, 2019). The use of computational essays can provide different pedagogical advantages. During the process of creating computational essays, students can use pre-written code from prepared worked examples in order to adopt, adapt, and enhance it for their own purposes, such as performing data analyses or developing data-based models. Furthermore, when students create their computational essays to document their results, they need to reflect more deeply on the code they have written and on the results of their data analysis when they have to document their reasoning. Nevertheless, the documentation of reasoning in a computational essay in written comments is a great challenge (Rule et al., 2018), so further research is needed to get insights about adequate guidance.

The Educational Use of Worked Examples

We base our approach to worked examples on literature referring to teaching contexts. The effectiveness of worked examples, especially for teaching problem-solving tasks, has been shown (e.g., Atkinson et al., 2000). The examples typically specify the problem to solve, formulate criteria for a solution, offer a concrete answer, and show the steps to find this solution. We created a worked example in the form of a prepared computational essay to guide students in producing their own computational essay about their machine learning modeling. A crucial aspect of design is the degree of guidance the worked example provides. Its purpose is to give enough orientation without prescribing every necessary step narrowly. We considered seven criteria for designing our worked example, three derived from the purpose of the worked example (a–c) and four taken from the literature (d–g). For the creation of our concrete worked examples, we have chosen (a) a different context than the actual project context so that students have enough freedom for their own contributions when adapting the worked example. Our worked examples (b) provide the necessary code as code building blocks for all individual steps of the process. Additionally, (c) explanations provide additional information about the concrete machine learning process and provide an example of the degree of explanation expected from students in their own computational essays. Moreover, we adopted the following four design criteria for worked examples developed by Atkinson et al. (2000). (d) *Integrating example parts* and using (e) *multiple modalities* are already realized by using JNBs because the given code, associated outputs, and text are combined in one document. The (f) *clarity of the subgoal structure* is established by dividing the worked example JNB into seven different sections separated by meaningful headlines that indicate the aims of the respective sections. In terms of (g) *completeness/incompleteness*, Atkinson et al. (2000) suggest that at specific points, the incompleteness of descriptions and explanations can be beneficial because students have to fill the gap with self-explanations. Such explanations have been found to improve the learning of students (Wylie & Chi, 2014). Therefore, aside from completely formulated comments, in some areas, hints are provided, which indicate that a comment is required, and meta-comments are used to indicate what the comment should contain.

Context of the Study and Previous Findings: The ProDaBi Data Science Course

We developed a yearlong data science course for the upper secondary level as part of the ProDaBi project and conducted it in cooperation with two schools in Paderborn for the third run in the 2020–2021 school year with three students aged 17–18. The course aims at teaching students how to use data for predictive modeling in order to solve a real-world problem. With their data project, the students aim to design, implement, and document data-based predictive models that predict the occupancy of different parking spaces for the coming hours in real-time. The educational idea of the course is that students first learn aspects of data exploration and machine learning processes during three well-prepared teaching modules to then use their gained knowledge and skills to work on their data project in more self-regulated phases afterward. The final products of this year's course are a web app (including a predictive model based on machine learning), two computational essays, and a presentation explaining and documenting the data and the methods used. The two computational essays focus on (a) a data exploration of real parking lot occupation data and (b) a first self-created predictive model created by using machine learning methods in the form of a decision tree. To create the second computational essay, the students participate in a teaching module on decision trees (Biehler & Fleischer, 2021) and then get a worked example as a scaffold. The worked example itself is a prepared computational essay containing the necessary programming code to create a decision tree based on data on students' media use (Podworny et al., 2022), thus being explicitly different from the concrete context of the course. It contains different kinds of support for documentation: example comments are given in the form of continuous text so that students can use these as orientation or even adapt them for their own computational essay. Other forms of documentation support are meta-comments that give information on which aspects need to be included in a respective comment, whereas hints represent the weakest form of support by only indicating that a comment is required at a certain point.

A more detailed description of the data science course and the worked example is given in (Fleischer et al., 2022). The authors analyzed the same data regarding the second computational essay and found that the worked example provided good support for technically creating a competitive decision tree using machine learning. Students' comments in large parts showed adequate reasoning about many aspects from data preparation, formulation of quality criteria, evaluation of overfitting, and the evaluation of the model performance on test data. Nevertheless, the documentation regarding the context of the whole endeavor seemed to be more difficult for the students than the technical steps because they also showed weaknesses in explanations. However, the reasons for these narrative issues remained unclear. Fleischer et al. (2022) hypothesized there could either be a lack of understanding or of adequate documentation skills. Further insights into which issues occurred and why this happened may reveal recommendations for the design of the worked examples (or the teaching module), especially regarding the form of support.

RESEARCH QUESTION

This paper focuses on investigating the students' products in greater depth to identify reasons for weaknesses that were identified in the computational essays. Based on this, implications for the design of worked examples supporting students in documenting their programming process and intermediate results are supposed to be derived. Concretely, we want to answer the following research question: (RQ) What types of weaknesses occur regarding students' explanations in computational essays, and how can we improve our worked examples to better support students?

To answer this research question, we focus our analyses on the following questions:

(1) Which types of weakness occurred, and how often?
(2) Are these observed weaknesses due to a lack of substantive understanding of the contents or a lack of adequate documentation?
(3) What are the relations between the outer form in the computational essay and the respective parts in the worked example?
(4) Why do certain types of errors occur?

METHOD

To answer the research question, we investigate three computational essays created by students, which were already analyzed (Fleischer et al., 2022). Further evidence is drawn from recordings of the students' oral presentations of their essays and subsequent interviews regarding their experiences using worked examples and creating computational essays.

In the previous investigations by Fleischer et al., the individual cells of the computational essays were categorized regarding different aspects. For both code and text cells, an assessment of the content of each cell was made so that it could be classified as being very good, having minor weaknesses, major weaknesses, or being deficient. Although it could be observed that the weaknesses appeared in different forms, this aspect was not yet assessed. The analysis made here focuses on the text cells containing the written comments in the students' computational essays and the related support in the worked example. For all text cells, the *text form* (continuous text, bullet sentences, bullet points) is assessed. Furthermore, by comparing the students' computational essays to the worked example, we assess for each text cell whether there is a corresponding cell in the worked example. The type of the corresponding cell is then captured as *type of support in the worked example* (example comment, meta-comment, hint, nothing, as explained above).

For all text cells with weaknesses, the *type of weakness* is categorized. Using an inductive approach, we identified three types of weaknesses in comments (errors in content, incomprehensible, incomplete). With the help of recordings from the students' oral presentations, we assess the *reason for the weakness* (lack of understanding, lack of adequate documentation) by comparing whether their oral statements show a better understanding and go beyond the written comments.

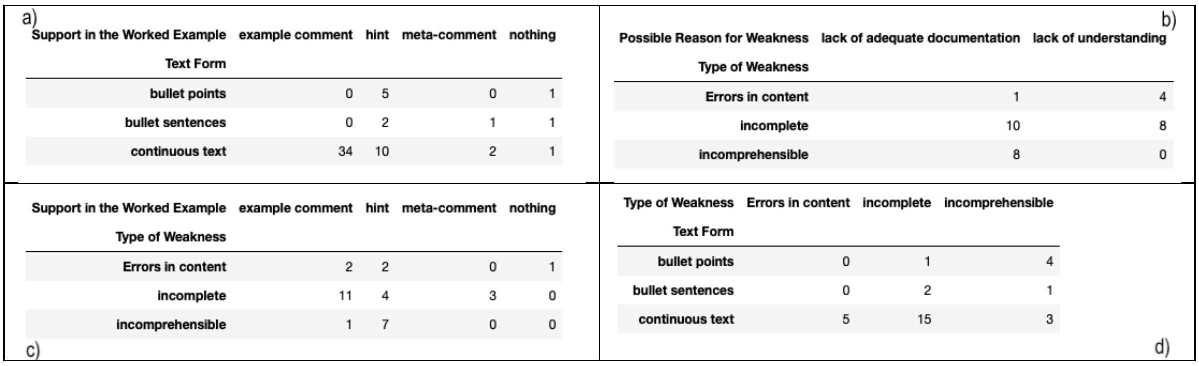
RESULTS

In total, we analyzed 57 comments, 33 of which have weaknesses (25 minor weaknesses, 5 major weaknesses, 3 deficient).

(1) Which types of weakness occurred, and how often?
The evaluation of the frequency of occurrence of different types of weaknesses shows that most errors concerned the completeness of the documentation and did not contain wrong statements (58.1%). Another 25.8% of the weaknesses relate to incomprehensible comments, whereas 16.1% were content errors, which means that the comment is factually wrong.

(2) Are the weaknesses found based on a lack of substantive understanding or a lack of adequate documentation?
As part of the analysis, we observed that in 61.3% of the weaknesses identified in the computational essays, the students explained the respective part correctly in their oral presentations. Only in 38.7% of the cases did students show a weakness in a comment and a lack of understanding in the oral presentation. We conclude that the results hint at a lack of adequate written documentation skills because, in most cases, the students were aware of the correct explanations and interpretations but did not manage to express them in a written form.

(3) What are the relations between the text form in the computational essay and the respective parts in the worked example?



a)

Support in the Worked Example	example comment	hint	meta-comment	nothing
Text Form				
bullet points	0	5	0	1
bullet sentences	0	2	1	1
continuous text	34	10	2	1

b)

Possible Reason for Weakness	lack of adequate documentation	lack of understanding
Type of Weakness		
Errors in content	1	4
incomplete	10	8
incomprehensible	8	0

c)

Support in the Worked Example	example comment	hint	meta-comment	nothing
Type of Weakness				
Errors in content	2	2	0	1
incomplete	11	4	3	0
incomprehensible	1	7	0	0

d)

Type of Weakness	Errors in content	incomplete	incomprehensible
Text Form			
bullet points	0	1	4
bullet sentences	0	2	1
continuous text	5	15	3

Figure 1. Cross tables showing the results regarding the relations between (a) text form and support in the worked example, (b) type of weakness and the possible reason, (c) type of weakness and support in the worked example, and (d) text form and type of weakness

Initially, we expected students to write continuous text in their comments. However, some comments were given in bullet sentences or bullet points. When comparing the different support types in the worked examples to the form of documentation in the computational essays (see Figure 1(a)), it is remarkable that, in all 34 cases, when the continuous text was used in the worked example, the students indeed used continuous text in corresponding comments. This indicates that the students orient themselves on the external format of the respective part of the worked examples when writing their own computational essays. Therefore, it can be helpful to provide example comments.

(4) Why do certain types of errors occur?
Next, we wanted to focus on the occurrences and reasons for different types of errors in the students' computational essays. In this regard, it is first noticeable that most of the weaknesses in the computational essays were due to a lack of adequate documentation skills (61.3%; see Figure 1(b)). In particular, for all incomprehensibility weaknesses in their computational essays, the students explained the respective parts in their oral presentation adequately. They might have missed a more substantially formulated requirement in the worked example of how to formulate a comment in order to present their own (programming-) process in a comprehensible and reproducible way. Our suggestion in this regard is to provide more meta comments on the form of the documentation, including pointing out which aspects should be addressed in which form to enable students to adequately document their knowledge and understanding in their computational essays.

This suggestion is also supported by the analysis of the relation between the types of errors and the support provided by the worked example: in seven of the eight weaknesses related to incomprehensibility, rather than being able to adapt an existing text, only hints were given in the respective parts of the worked examples (see Figure 1(c)).

Another factor that might have influenced the occurrence of incomplete or incomprehensibility errors is related to the text form of documentation in the computational essays: In 10 cases, the students provided documentation in bullet points or bullet sentences. In eight of these cases, the documentation contained weaknesses, which all were related to incomplete or incomprehensible remarks (see Figure 1(d)). Five out of the eight weaknesses related to incomprehensibility appeared in comments in which the students gave only bullet points or bullet sentences.

CONCLUSION

Our analysis investigated different types of weaknesses in computational essays and possible reasons for these, from which implications were derived for the (re-) design of the worked example. Most weaknesses were incomplete comments. As students' oral presentations mostly showed a better performance in these cases (see Figure 1(b)), we concluded that most of these weaknesses occur due to inadequate documentation skills. This indicates that students are unsure about what content should be in a comment. To avoid this type of weakness, the worked example could offer (additional) meta-comments about the required contents and the purpose of a particular comment to give more orientation.

Other weaknesses are incomprehensible comments. This weakness mainly occurs when only a hint is given in the worked example (7/8—see Figure 1(c)), and they often have a form different from the continuous text (5/8—see Figure 1(d)). As before, this type of weakness only occurred due to a lack of adequate documentation and not due to a lack of understanding (see Figure 1(b)). That means a reader could not comprehend the students' thoughts while reading the comment, but the students were able to explain their thoughts orally based on their documentation. Similar findings were made by Rule et al. (2018) regarding documentation in JNB. It seems that students used the notebook in these parts as a "data analysis diary" that makes the reasoning reproducible for themselves, which is already an achievement, rather than communicating it to other readers. If the aim is to make it reproducible for other readers, it might be helpful to provide example comments or meta-comments (or both) to enable students rather than just giving students hints. That way, students can document their reasoning with suggestions for an adequate form or use an example comment as a scaffold. That does not mean that it is never adequate to leave gaps in argumentation, as suggested by Atkinson et al. (2000) and Wylie and Chi (2014), but this should be used carefully.

The third category of weaknesses is errors in content, which is the rarest type of error we found. Most of these weaknesses are due to a lack of understanding (4/5—see Figure 1(b)). The lack of understanding can only partially be addressed in the worked example but can be addressed in the teaching module. For these types of issues, we do not derive any implication for the design of the worked example.

In summary, most of the weaknesses we found were due to difficulties in documentation rather than in understanding, where students could have been supported by designing a worked example slightly differently. From our analysis, we derive the implications that in a worked example, gaps in argumentation (hints) should be used carefully because this provides the lowest degree of guidance or orientation for students. Instead, meta-comments about required content and the purpose of comments might better support students in adequately documenting their thoughts without giving too much strict guidance. In conclusion, a possible way of improving worked examples for documentation in computational essays could be to use more meta-comments indicating the general structure of documentation (to reduce incomprehensibility errors) and aspects that should be included there (to reduce incomplete documentation).

REFERENCES

Atkinson, R. K., Derry, S. J., Renkl, A., & Wortham, D. (2000). Learning from examples: Instructional principles from the worked examples research. *Review of Educational Research*, *70*(2), 181–214. https://doi.org/10.3102/00346543070002181

Bargagliotti, A., Franklin, C., Arnold, P., Gould, R., Johnson, S., Perez, L., & Spangler, D. A. (2020). *Pre-K-12 guidelines for assessment and instruction in statistics education II (GAISE II). A framework for statistics and data science education.* American Statistical Association; National Council of Teachers of Mathematics. https://www.amstat.org/asa/files/pdfs/GAISE/GAISEIIPreK-12_Full.pdf

Biehler, R., & Fleischer, Y. (2021). Introducing students to machine learning with decision trees using CODAP and Jupyter Notebooks. *Teaching Statistics*, *43*(S1), S133–S142. https://doi.org/10.1111/test.12279

diSessa, A. A. (2000). *Changing minds: Computers, learning, and literacy*. MIT Press.

Fleischer, Y., Biehler, R., & Schulte, C. (2022). Teaching and learning data-driven machine learning with educationally designed Jupyter Notebooks. *Statistics Education Research Journal*, *21*(2), Article 7. https://doi.org/10.52041/serj.v21i2.61

Hüsing, S., & Podworny, S. (2022). Computational essays as an approach for reproducible data analysis in lower secondary school. In R. Helenius & E. Falck (Eds.), *Statistics education in the era of data science. Proceedings of the Satellite conference of the International Association for Statistical Education (IASE)*. ISI/IASE. https://doi.org/10.52041/iase.zwwoh

International Data Science in Schools Project Curriculum Team. (2019). *Curriculum frameworks for introductory data science*. http://idssp.org/files/IDSSP_Frameworks_1.0.pdf

Kluyver, T., Ragan-Kelley, B., Pérez, F., Granger, B. E., Bussonnier, M., Frederic, J., Kelley, K., Hamrick, J. B., Grout, J., Corlay, S., Ivanov, P., Avila, D., Abdalla, S., Willing, C., & Jupyter Development Team. (2016). Jupyter Notebooks—A publishing format for reproducible computational workflows. In F. Loizides & B. Scmidt (Eds.), *Positioning and power in academic publishing: Players, agents and agendas* (pp. 87–90). IOS Press. http://doi.org/10.3233/978-1-61499-649-1-87

Ko, A. J., Abraham, R., Beckwith, L., Blackwell, A., Burnett, M., Erwig, M., Scaffidi, C., Lawrance, J., Lieberman, H., Myers, B., Rosson, M. B., Rothermel, G., Shaw, M., & Wiedenbeck, S. (2011). The state of the art in end-user software engineering. *ACM Computing Surveys*, *43*(3), Article 21. https://doi.org/10.1145/1922649.1922658

McNamara, A. (2019). Key attributes of a modern statistical computing tool. *The American Statistician*, *73*(4), 375–384. https://doi.org/10.1080/00031305.2018.1482784

Perez, F., & Granger, B. E. (2015, July 7). *Project Jupyter: Computational narratives as the engine of collaborative data science*. Jupyter blog. Retrieved September 11, 2021, from https://blog.jupyter.org/project-jupyter-computational-narratives-as-the-engine-of-collaborative-data-science-2b5fb94c3c58

Podworny, S., Fleischer, Y., Stroop, D., & Biehler, R. (2022). *An example of rich, real, and multivariate survey data for use in school* [Paper presentation]. 12th Congress of the European Society for Research in Mathematics Education, Bozen, Italy.

Rule, A., Tabard, A., & Hollan, J. D. (2018). Exploration and explanation in computational notebooks. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. Association for Computing Machinery. https://doi.org/10.1145/3173574.3173606

Wylie, R., & Chi, M. T. H. (2014). The self-explanation principle in multimedia learning. In R. Mayer (Ed.), *The Cambridge handbook of multimedia learning* (pp. 413–432). Cambridge University Press. https://doi.org/10.1017/CBO9781139547369.021