

Development of a learning pilot for the remote teaching of Smart Maintenance using open source tools

Maira Callupe, Luca Fumagalli, Domenico Daniele Nucera

Department of Management, Economics, and Industrial Engineering, Politecnico di Milano, Italy.

Abstract

Technology has created a vast array of educational tools readily available to educators, but it also has created a shift in the skills and competences demanded from new graduates. As data science and machine learning are becoming commonplace across all industries, computer programming is emerging as one of the fundamental skills engineers will require to navigate the future and current workplace. It is, thus, the responsibility of educational institutions to rise to this challenge and to provide students with an appropriate training that facilitates the development of these skills. The purpose of this paper is to explore the potential of open source tools to introduce students to the more practical side of Smart Maintenance. By developing a learning pilot based mainly on computational notebooks, students without a programming background are walked through the relevant techniques and algorithms in an experiential format. The pilot highlights the superiority of Colab notebooks for the remote teaching of subjects that deal with data science and programming. The resulting insights from the experience will be used for the development of subsequent iterations during the current year.

Keywords: *Learning nugget; Jupyter notebook; open source tool; online learning; smart maintenance.*

1. Introduction

The 21st century is characterized as the era of digital transformation, as technology has permeated virtually every aspect of our lives and is fundamentally changing the way in which humans interact with their surroundings.

The increasing emergence and adoption of technological advancements has an impact on higher education in two distinctive manners: as an opportunity due to the potential for the development of sophisticated educational tools and approaches, but also as a challenge due to the educational needs created as a result of new expectations from the job market. Technologies such as the Internet of Things (IoT), Big Data, cloud computing, and others, are becoming commonplace in several industries as they are key enablers of what is known as *smart manufacturing* (Wang et al. 2016).

In recent years, a number of publications have discussed the effects caused by the introduction of these technologies into industrial spaces in terms of jobs and competences. These works mainly highlight the evolution of technical skills required from the workforce (Pinzone et al. 2017) as well as the need for a higher level of education among employees (Hecklau et al. 2016). One of the technical skills frequently appearing in these and similar studies is programming, as, due to the increasing digitalization of industrial shop floors, it is becoming ubiquitous in manufacturing and maintenance activities.

2. Computational notebooks: Jupyter and Google Colab

Computational notebooks are interactive environments for writing documents where users can insert regular text and software code that can be executed and which results are incorporated into the document. This combination of code, text, and visualization in a single document makes notebooks well suited for work in computation-intensive areas such as data science and machine learning (Rule et al. 2018). While notebooks have been around for quite some time, it is only recently that the emergence of open-source notebooks has caused a sharp increase in their adoption for research and data analytics.

Among the multiple alternatives currently available, Jupyter notebooks stand out as an excellent educational instrument for the instruction of a wide range of topics in science, engineering and programming. In the last years, there are several publications discussing their use in subjects related to artificial intelligence (O'Hara et al. 2015), manufacturing engineering (Suárez et al. 2018), cyber-security (Shahriar et al. 2019), and data analytics (Cardoso et al. 2019).

Colab (Figure 1) is a free Jupyter notebook environment offered by Google, which, in addition to the features included in regular Jupyter notebooks, also provide a number of extra features which can be summarized into the following advantages (Table 1):

Portability. Unlike Jupyter notebooks which are stored on the user local machine, Colab notebooks are stored in the cloud and are easily accessible from any device through a Google account.

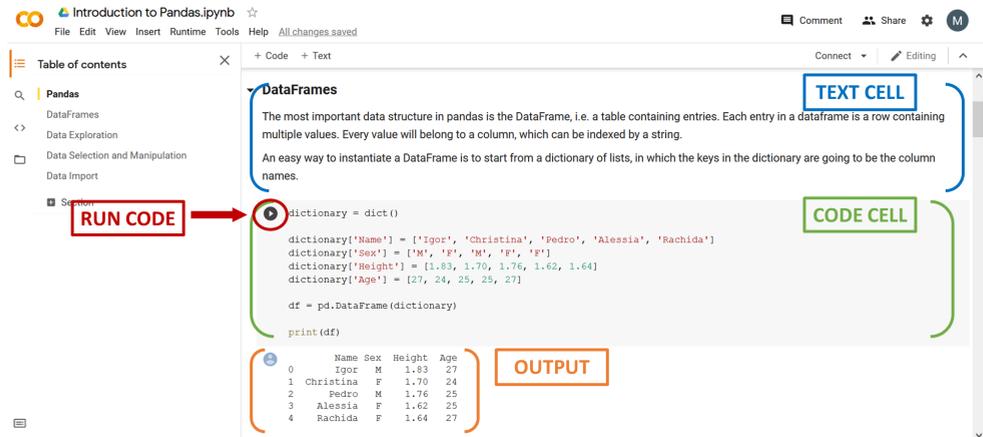


Figure 1. Notebook interface highlighting the main elements.

Computational Power. In addition to cloud storage, Colab also provides their dedicated GPU and TPU computing resources and does not use the processing power of the user local machine.

Organized Interface. Colab includes a table of contents on the sidebar that facilitates the organization and exploration of the workspace, which is particularly significant for long notebooks.

Convenience. Colab does not require the installation of any software or package in the user local machine, and the process for sharing notebooks is quick and straightforward. Moreover, in addition to traditional data libraries included in Jupyter notebooks, Colab also provides pre-installed libraries for computationally-heavy machine learning projects that can run on any machine, as they use cloud computing resources.

Table 1. Comparison of features included in Jupyter and Colab notebooks.

Features	jupyter	colab
Running code	●	●
Note taking	●	●
Interactive	●	●
Cloud storage		●
Computing power		●
Organized interface		●

Source: (Kluyver et al. 2016) and (Google 2020)

These additional features and the resulting advantages are highly relevant in a scenario where lessons are delivered in a remote modality. For this reason, Colab notebooks are used as the primary instructional tool for the development of the learning pilot, as students can easily access, create and share notebooks without experiencing any technical hassles beyond the creation of a Google account.

3. Development of the learning pilot

The context for the design of the learning pilot was given by two main factors defined from the beginning: the learning modality and the learner's level of preparation. Due to the sudden change in the lecture delivery method caused by the onset of the COVID-19 emergency, the structure and contents of the Smart Maintenance course had to be redesigned to allow for an effective remote learning experience. While the main course contents were adapted for emergency remote teaching, the learning pilot was proposed as an accompaniment that could exploit the potential of the online format and include practical contents previously taught only in theory.

Furthermore, the distinct profile of the students expected to enroll in the course was well known from the previous iterations of the course. The large majority are industrial and mechanical engineers in senior positions at large companies belonging to mining and manufacturing industries who seek to pursue specialized postgraduate qualifications. Notably, a key shared characteristic is the possession of limited knowledge and experience in programming or the complete absence of a significant background.

3.1. Pilot Description and Objective

While the main course delivered in parallel focused on the theoretical foundation of Smart Maintenance, such as technologies and implementation frameworks, the learning pilot aims to present students a more practical experience involving the use of programming for the development and implementation of Smart Maintenance algorithms. Through the planned learning activities, students are able to visualize the underlying architecture while experiencing the steps involved in a typical solution.

3.2. Pilot Structure and Contents

The structure and contents of the pilot were defined and developed respectively based on the context provided at the beginning of this section (Table 2). The main learning activities chosen for the pilot are three: learning nuggets, demonstration exercises, and Q&A sessions (Figure 2).

The combination of these activities that involve instruction and guided discovery is expected to accomplish the creation of a deep learning experience for the students. The format of learning nuggets has been chosen due to its suitability for online teaching (Bailey et al. 2006). They are videos about 15 min in length that contain the necessary theory for the demonstration exercises.

The demonstration exercises were developed using computational notebooks through which students are walked through the process alternating between instructions and sections where code is meant to be written and executed. Both learning nuggets and demonstration exercises are delivered asynchronously, encouraging students to take an active role and to trace their learning path at their own pace.

Finally, live Q&A sessions were scheduled twice a week during which students are persuaded to ask questions as they go through the contents of the learning nuggets and the demonstration exercises.

Table 2. Specifications of the Smart Maintenance learning pilot based on moderating variables for online learning design (Means et al. 2014).

Duration	3 weeks
Modality	Online
Pacing	Self-paced
Synchrony	Blended (synchronous + asynchronous)
Instructor role	Passive responder: Answer students' questions when prompted
Student role	Active learner: Revise materials and solve exercises

4. Learning pilot notebooks

The notebooks are intended to aid students in the understanding of the role of programming in the development and implementation of an industrial Smart Maintenance solution. The notebooks follow a clear structure that ensures students are well aware of their position with respect to the process described in the exercises, which is as follows:

- a) **Objective.** Explanation of the significance of the contents within the larger context and expectations from the execution of the code.
- b) **Recap.** Recapitulation of the previous notebooks highlighting their relevance for the current one.
- c) **Exercise.** Instructions alternated with code cells.
- d) **Conclusions.** Exploration of the exercise results. Alternative variables are provided inviting students to modify the code written in the exercise.

The notebooks are designed to be explored sequentially so that the output of a particular notebook becomes the input for the following. The case study is intended to be solved through the successive application of the contents learned in the previous 6 learning modules.

The first two notebooks are intended as a quick introduction to programming in Python, presenting basic concepts and methods required for later exercises. Notebooks 4 to 6 walk students through the process of analyzing data extracted from a number of sensors installed on an industrial robot.

Students are walked through the process of preparing data for further analysis and the application of algorithms that will result in the extraction of meaningful information concerning the behavior of the machine, which can be used to identify undesired conditions that are relevant for the development of a Smart Maintenance solution.

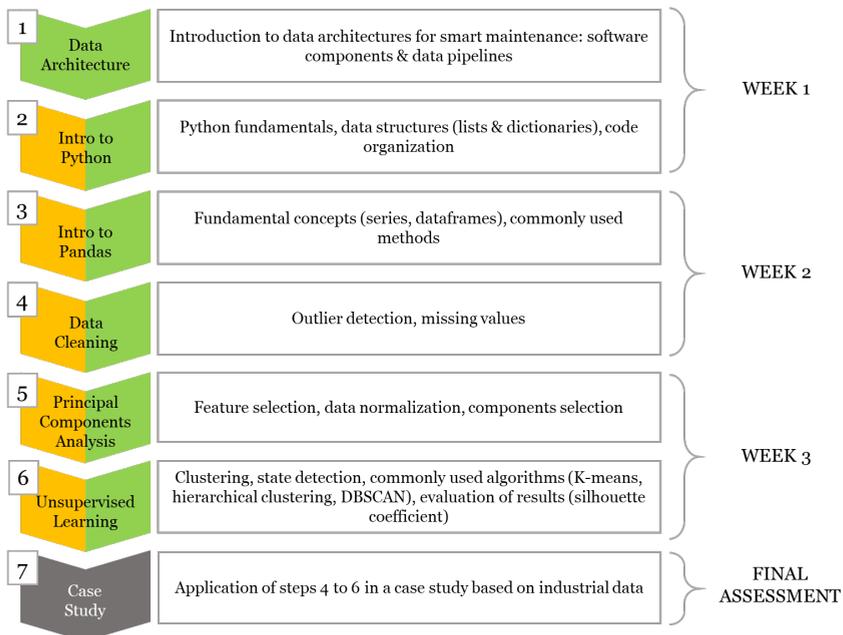


Figure 2. The learning pilot is structured into 6 learning modules and a case study. Learning modules are comprised of learning nuggets (green) and demonstration exercises (yellow).

5. Student feedback and lessons learned

After the conclusion of the learning pilot, students were asked to provide their feedback about the pilot by completing a survey with open-ended questions related to the development of the course. Their answers provided valuable insights that will serve as an input for designing subsequent iterations of the course.

Firstly, a significant number of students reported being confused about seemingly needless programming-based learning contents and hesitant about their practical value for their careers. Despite its inherent technological sophistication, at the university level Smart Maintenance is commonly taught as a topic within larger subjects concerned with the maintenance of industrial equipment or, if a standalone course, by covering theoretical aspects such as its origins, main concepts, standards and frameworks for implementation, etc.

The connection between Smart Maintenance, data science, and programming -as well as its practical implications- must be emphasized to students throughout the course. Most importantly, however, students should be able to understand the need for a shift in the contents usually covered in class in order to address skills gaps already identified in the workforce.

Furthermore, the large majority of students reported that the jump from introduction to the development of algorithms was abrupt. The pilot included only two notebooks dedicated to the introduction of Python where basic concepts are introduced. Difficulties in understanding algorithmic and programming concepts are well documented, and one of the methodologies implemented by educators to address these is Design-Based Research (Barab and Squire 2004). Through the subsequent implementations of the pilot the authors expect to adopt this student-centric approach that will allow for the identification of the shortcomings of each cycle and the subsequent redesign and reevaluation of the contents presented to students.

6. Discussion and Outlook

This article presents and discusses the use of computational notebooks as an apt environment for the teaching and learning of programming in the specific context of Smart Maintenance. While the learning pilot came into existence as a coping strategy during the COVID-19 emergency, it highlighted the potential of notebooks and open source tools to create lessons intended to be delivered online.

Programming in itself is a complex subject that is not easy to learn, and there is abundant literature that offers alternatives for a successful approach. The paper discusses the success of teaching programming starting from a very specific context (i.e. maintenance activities) with which students are already familiar. This contextualization represents an advantage to create a more effective learning experience, as students are able to place the contents learned within the larger scheme of concepts and experiences they already possess as a result of their work experience. While students showed some degree of reluctance when initially presented with the notebooks, they became more confident and eager to learn once they began to understand the importance of the contents visited.

The course is expected to undergo improvements based on the lessons learned and to be adapted for a 2nd and 3rd iterations by adopting a Design-Based Research approach. The respective audiences in these iterations will be high school students and university students following an Industrial Engineering program.

Acknowledgements

This project has received funding from the European Union’s “Erasmus+ Capacity Building in the field of Higher Education” programme under grant agreement No 2019-1949 / 001-001 (correspondent to the project shortly entitled “NePREv”, “NEXt Production REVolution”).

References

- Bailey, Chris, Mohd T. Zalfan, Hugh C. Davis, Karen Fill, and Gráinne Conole. 2006. “Panning for Gold: Designing Pedagogically-Inspired Learning Nuggets.” *Educational Technology and Society* 9(1).
- Barab, Sasha, and Kurt Squire. 2004. “Design-Based Research: Putting a Stake in the Ground.” *Journal of the Learning Sciences* 13(1). doi: 10.1207/s15327809jls1301_1.
- Cardoso, Alberto, Joaquim Leitão, and César Teixeira. 2019. “Using the Jupyter Notebook as a Tool to Support the Teaching and Learning Processes in Engineering Courses.” in *Advances in Intelligent Systems and Computing*. Vol. 917.
- Google. 2020. “Colaboratory – Google.” *Colaboratory Frequently Asked Questions*. Retrieved (<https://research.google.com/colaboratory/faq.html>).
- Hecklau, Fabian, Mila Galeitzke, Sebastian Flachs, and Holger Kohl. 2016. “Holistic Approach for Human Resource Management in Industry 4.0.” *Procedia CIRP* 54:1–6. doi: <https://doi.org/10.1016/j.procir.2016.05.102>.
- Kluyver, Thomas, Benjamin Ragan-Kelley, Fernando Pérez, Brian Granger, Matthias Bussonnier, Jonathan Frederic, Kyle Kelley, Jessica Hamrick, Jason Grout, Sylvain Corlay, Paul Ivanov, Damián Avila, Safia Abdalla, and Carol Willing. 2016. “Jupyter Notebooks—a Publishing Format for Reproducible Computational Workflows.” in *Positioning and Power in Academic Publishing: Players, Agents and Agendas - Proceedings of the 20th International Conference on Electronic Publishing, ELPUB 2016*.
- Means, Barbara, Marianne Bakia, and Robert Murphy. 2014. *Learning Online: What Research Tells Us about Whether, When and How*.
- O’Hara, Keith J., Douglas Blank, and James Marshall. 2015. “Computational Notebooks for AI Education.” in *Proceedings of the 28th International Florida Artificial Intelligence Research Society Conference, FLAIRS 2015*.
- Pinzone, Marta, Paola Fantini, Stefano Perini, Stefano Garavaglia, Marco Taisch, and Giovanni Miragliotta. 2017. “Jobs and Skills in Industry 4.0: An Exploratory Research.” in *IFIP Advances in Information and Communication Technology*. Vol. 513.
- Rule, Adam, Aurélien Tabard, and James D. Hollan. 2018. “Exploration and Explanation in

Computational Notebooks.” in *Conference on Human Factors in Computing Systems - Proceedings*. Vols. 2018-April.

Shahriar, Hossain, Miahcel Whitman, Dan Chia Tien Lo, Fan Wu, Cassandra Thomas, and Alfredo Cuzzocrea. 2019. “Experiential Learning: Case Study-Based Portable Hands-on Regression Labware for Cyber Fraud Prediction.” in *Proceedings - 2019 IEEE International Conference on Big Data, Big Data 2019*.

Suárez, Andres, Miguel A. Alvarez-Feijoo, Raquel Fernández González, and Elena Arce. 2018. “Teaching Optimization of Manufacturing Problems via Code Components of a Jupyter Notebook.” *Computer Applications in Engineering Education* 26(5). doi: 10.1002/cae.21941.

Wang, Shiyong, Jiafu Wan, Daqiang Zhang, Di Li, and Chunhua Zhang. 2016. “Towards Smart Factory for Industry 4.0: A Self-Organized Multi-Agent System with Big Data Based Feedback and Coordination.” *Computer Networks* 101:158–68. doi: 10.1016/j.comnet.2015.12.017.