Developing Machine Learning Skills With No-Code Machine Learning Tools

Emmanuel Djaba
Ghana Institute of Management and Public Administration, Ghana

Joseph Budu
https://orcid.org/0000-0002-0003-5807
Ghana Institute of Management and Public Administration, Ghana

INTRODUCTION

Building a machine learning (ML) model involves multiple complex skill sets. In the most minimal form, the standard protocol for building and using ML models involves at least three steps. Firstly, data must be collected and preprocessed. Secondly, a model must be trained using the preprocessed data, and thirdly, the trained model must be deployed in some form of application (Ramos et al., 2020). Generally, each step in this process requires advanced technical skills such as data collection, data preprocessing, model training, and model performance evaluation. In addition to their complexity, these required skills are cognitively demanding to acquire and use and are thus often restricted to experts. If we take into account the fact that, most models have to be updated regularly to accommodate for evolutions in the data (concept drift), the entire ML process then becomes expensive and unattainable for organizations and individuals without extensive resources and skills.

Against this backdrop, No-code ML (NML) tools have emerged as avenues for individuals to build ML models without possessing the requisite technical knowledge and skills (García-Ortiz and Sánchez-Viteri, 2021). This is because NML tools commonly exist as visual programming tools for ML (von Wangenheim et al., 2021). Hence NML tools considerably reduce the cognitive effort needed to create ML models. The reduction in effort is achieved because users focus on the logic of the system being developed instead of the textual elements (programming language syntax and semantics). More specifically, NML tools allow users to train ML models either by using a drag-and-drop interface to place visual elements on a canvas; or by specifying input, output parameters, and values in a few clicks. Furthermore, where visual elements are employed, they are largely used in the form of blocks or flows. This approach improves users' ability to learn by helping them prevent errors and improving their understanding of the concepts at hand. Consequently, NML tools allow one to build models relatively quickly. Additionally, NML tools reduce the financial barriers to using ML. This is mainly due to their features, mode of packaging, and distribution. Quite a substantial number of NML tools are free to download and free to use. Also, some are available over the web as cloud-based tools, and as such, they substantially reduce the necessity of using specialized hardware for training.

Naturally, the emergence of NML tools has generated discourse in academic research. Extant literature highlights the potential of no-code ML tools in developing relevant ML skills, knowledge, and attitudes. For instance, Lao (2020) provides empirical evidence of the use of NML tools in teaching high-school students how to create ML models and troubleshoot their performance. Similarly, García et al. (2020) and Rodríguez-García et al. (2021) identify ML knowledge and computational skills as possible learn-

DOI: 10.4018/978-1-7998-9220-5.ch097

ing outcomes of introducing NML tools to high school students. The foregoing indicates that ML skills can be learned using NML tools. Despite this invaluable knowledge, what remains unclear is how NML tools contribute to the acquisition of ML skills. From a technology affordances perspective, the authors conceive that there is a gap in the explanation of how the affordances generated when learners interact with NML tools, lead to the development of ML skills. An explanation that fills this gap exposes the positive mechanisms embedded in NML tools that lead to the generation of ML skills. Subsequently, developers of NML tools will be able to identify and grow these mechanisms to maximize learning outcomes. Thus, in this chapter, we draw on the technology affordance theory to explain how NML tools afford the development of ML skills.

This chapter has six sections. The first section provides the rationale and motivation for this chapter, while the second provides an overview of NML and reviews relevant literature on the subject. The third section expounds on the theory of technology affordances and constraints while the fourth section outlines our methodology for this work. The fifth, and sixth sections present the core findings of the chapter by covering the results, analysis, and discussions respectively. To conclude, we identify avenues for future research and summarize our arguments and findings.

BACKGROUND

In the introductory section, we have seen that NML tools allow users without computer programming skills and knowledge to develop ML algorithms and projects to solve problems. However, not all NML tools are made the same. While some enable data preprocessing, others focus exclusively on model training and deployment. During model training, an interface is provided for users to load data. The generally accepted forms of data include text, audio, video, and images (Carney et al., 2020). After loading data, the type of ML task to be performed is selected. Here, the user may choose to perform clustering, image classification, sound classification, and so forth. It should be noted that while some NML tools provide access to a wide variety of ML tasks, others focus on enabling very specific tasks. After the task to be performed has been selected, the user can train the model for the selected task using the data provided. Finally, the user can evaluate the model to determine how well it performs. Support for model evaluation is provided in different ways. Some tools enable evaluation by providing detailed graphs on a host of preselected evaluation metrics. Alternatively, other tools allow users to evaluate models by trying them out on sample input data to determine how often the model performs the task correctly (Ozan, 2021).

In supporting model deployment NML tools enable users to export the trained models in easily reusable formats. Usually, this is done by saving models in formats supported by popular ML frameworks and libraries. Thus, it is common to see NML tools export trained models to formats supported by popular ML libraries such as Tensorflow. Consequently, NML tools can support a so-called “low-code” mode of ML deployment where no computer programming is done for training models, but some code is written to incorporate trained models into other computer programs (Carvalho & Harris, 2020).

At the core, NML tools have the potential to enable the wider ML community to overcome three major problems. Firstly, they can address the dire, and yet unsatisfied need for ML experts in various industries. Since its inception, the initial adoption of ML has been by technology-focused organizations such as Google and Microsoft. However, over time, a wide array of organizations have begun to adopt ML. From exemplars in governance, banking, health, and the retail industry (Lamberti et al., 2019; Mikalef et al., 2021), the potential of ML to improve operations and create value in new and interesting ways has been highlighted. However, this rise in adoption has not been accompanied by a similar rise in