Accepted Manuscript in Proceedings of CSEE&T 2024.

Adapting Computational Skills for AI Integration

Hanya Elhashemy
Technical University Of Munich
Massachusetts Institute of Technology
Munich, Germany
hanya.elhashemy@tum.de

Harold Abelson
Massachusetts Institute of Technology
Massachusetts, USA
hal@mit.edu

Tilman Michaeli
Technical University Of Munich
Munich, Germany
tilman.michaeli@tum.de

***Abstract*—In today's data-driven world, the importance of data literacy is paramount. However, software engineering education has not adequately addressed integrating comprehensive data science curricula, leaving students ill-equipped for the future of artificial intelligence (AI), which is built on the foundations of data science. This gap is exacerbated by the lack of tailored courses and the intimidating nature of existing tools for beginners. Consequently, students often miss out on essential skills like data cleanup, real-world application of machine learning (ML) algorithms, and the integration of big data in software products.**

This paper addresses these challenges by proposing a novel approach to applied data science for software engineering students. We argue for a shift from traditional algorithm-focused teaching to a curriculum emphasizing real-world problem-solving, leveraging data science techniques. By empowering students to define and tackle their own data-driven projects, we aim to increase motivation, enhance data literacy, and instill a data-thinking mindset in future software engineers to prepare them for the AI world.

Overall, this paper contributes to the advancement of software engineering education for young learners by offering a comprehensive framework, the data action educational framework (DAEF), and a data science toolkit that enables DAEF by empowering learners to create original data-driven mobile apps.

***Index Terms*—data thinking, data science, software engineering education, computational action**

I. Introduction

Observing the current trends in software engineering, we noticed many software applications integrating ML, AI and generative AI (Gen AI) in their core business logic. Many people experience AI and interact with it without knowing how it functions [9]. All this increases the demand for software engineers who grasp the concepts of ML and AI to be able to efficiently modify the design of their software applications and accommodate the required infrastructure of ML models [1].

We argue that software engineers of the 21st century must understand AI's limitations and potential to be able to leverage its power responsibly in their software applications. In the industry, this need created a new role besides software engineers and data scientists, referred to as ML engineers. These engineers often collaborate with data scientists, software engineers, and domain experts to understand business requirements and develop ML solutions that address specific needs.

In University, when raising the new generation of software engineers in the dawn of AI we need to ensure that the students are equipped with data thinking [2]. Data thinking integrates computational thinking, statistical thinking, and domain thinking [2]. It provides software engineers with the cognitive process or a mindset to approach problems with a data-centric perspective [3], [8]. While a traditional software engineer sees "name", "birthdate", and "address" as user profile data stored in a database and used for login, a data-thinking software engineer sees beyond and understands the potential of this data in generating business value (e.g., customer segmentation, predictive modeling, geospatial analysis, personalization, customer insights, etc.) and would design their system differently to cater for these features.

In traditional software engineering curricula, students usually have one or two courses teaching them about working with data: (1) Algorithms and Data Structures Course and (2) Database Theory Course. In today's data-driven world, these courses no longer provide enough skills for AI integration. They mainly provide knowledge on collecting, storing, accessing, or organizing data but not knowledge on extracting insights from gathered data and identifying patterns [4]. To leverage the power of AI, software engineers need to have data-thinking skills to be able to identify innovative use cases for the application and integration of AI in software products.

II. Related Work

We found several "Intro to Data Science" courses (often targeting graduate students); however, their educational approaches prioritize content over learner-centered design, focusing on what students should learn rather than what they want to learn. They are often limited to theoretical content and algorithmic understanding, not enabling students to engage in hands-on, data-driven projects, and missing the opportunity for students to acquire terminology and skills naturally through practical experience. The closest to our work is Acuña's project-based teaching approach for a data science course to support software engineering students [5]. The main deliverable of his course's project is a data analysis workflow. Adding to his approach, we empower students to build tangible software products to solve a personally relevant real-world problem of their choice while experiencing activities across the entire data lifecycle [6]. Notably, existing curricula lack sufficient coverage of data cleanup, a fundamental aspect of data literacy crucial for ensuring the quality and accuracy of data-driven insights [7]. Educators often provide students with the “perfect” (most fitting) dataset example to focus on algorithmic understanding. Through our approach, which enables students to work on real-world problems, they experience what real-world data looks like and face the challenge of cleaning it.

III. Background & Context

Computational action [11] is a framework empowering learners to apply computational skills to create meaningful and impactful solutions to real-world problems. It involves not just understanding and applying algorithms, but also leveraging these skills to design, build, and deploy applications that address real-world problems. Computational action emphasizes the practical and transformative potential of computing, encouraging individuals to move beyond theoretical knowledge and actively engage in creating technology that can drive social change.

To prepare students for AI integration, the computational action framework should extend to enable students to realize the impact of data and support students in transforming their current computational skills to be data thinkers. We focus on five data literacy skills needed to transform students' computational skills for AI integration based on the data literacy competency model [10].

Firstly, individuals must master data collection methodologies, encompassing qualitative and quantitative data gathering while scrutinizing sources for bias. Secondly, proficiency in data analysis entails preprocessing — data cleanup, hypothesis development, critique, and choosing appropriate tools and algorithms for different types of datasets. Thirdly, evaluating data by assessing its quality, and formulating new research questions based on analysis insights. Fourthly, effective data visualization is crucial for communicating insights to stakeholders. Lastly, ethical considerations such as privacy, data accuracy, and model misuse awareness are paramount, emphasizing the societal impact and responsible use of data-driven projects.

IV. The Data Action Educational Framework (DAEF)

We introduce DAEF, which expands upon the Computational Action Framework [11] with a focus on data-related computations, providing a sustainable data science curriculum, and empowering students to engage in meaningful data-driven projects relevant to their lives and communities. We designed DAEF for K-12 and target (but are not limited to) high school and freshmen students aged 14 to 18 who are familiar with (block-based) programming but have no data science background; this also applies to freshmen software engineering students. It enables them to learn data science concepts while building mobile apps to address real-world problems. Through iterative project-based learning, students experience an entire data lifecycle and acquire skills in problem identification, domain research, real-world data analysis, data visualization, and data-driven app development.

A. Concepts

The framework imparts four foundational concepts to the learner.

1) Defining a real-world problem: Students acquire the ability to identify significant real-world problems that hold personal relevance to them. They further develop skills to define and abstract these problems, enabling them to address and solve them using data science methodologies.
2) Self-acquiring knowledge of the problem's domain: Unlike traditional teaching techniques where educators dictate the knowledge to be transferred to students, here the students themselves define what knowledge they require to solve the problem they identified.
3) Working with real-world data critically: Students engage with authentic, real-world data, often characterized by its ”messiness” and potential errors. They learn to apply various data science algorithms to process this data and critique it.
4) Understanding the impact of data: Through their data-driven solutions, students observe the significant impact of data and the meaningful insights it can yield.

Students practice these concepts by developing mobile apps to solve real-world challenges, collaborating with peers, receiving feedback, and refining their apps iteratively. They gain insights into the challenges of working with real-world data, including data anomalies, errors, and ethical considerations. Students develop a deeper understanding of data science concepts through hands-on experience developing a data-driven mobile app in a software engineering setting and engaging in all phases of a data science lifecycle, including data collection, cleanup, visualization, and prediction [13].

V. Applying Data Action

Recognizing the borders between a software engineering and a data science degree, we do not expect software engineers to be experts in creating ML models but to know when and how they can be integrated into software projects. In the following, we describe an example of employing the Data Action educational framework.

A. Tools

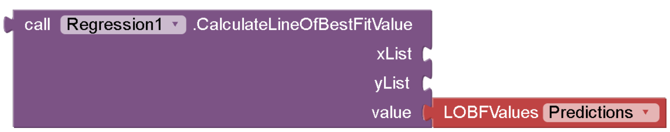
We use App Inventor [18], a visual programming environment that enables learners to create fully functional mobile apps using a drag-and-drop interface. We extend App Inventor with a data science toolkit [14] to abstract the implementation details of ML models and allow software engineering students to focus on problem-solving as data-thinking software engineers.

The toolkit refers to a set of data science features utilizing block-based programming, accessible to students for addressing real-world problems. The data science code blocks snap together like Lego blocks, integrating different data science features in a mobile app.

Solving real-world problems requires working with real-world data, which often exhibits inconsistencies, incompleteness, errors, and varying formats, posing challenges for analysis and interpretation. To interact efficiently with ML models, software engineers must understand such challenges and learn how to prepare and clean data to get the best results from used ML models [12]. For this reason, we introduce the anomaly detection code block (see figure 1). It automatically flags potential anomalies in a given dataset and allows students to use their domain knowledge to decide whether to remove or retain them, while also providing a preliminary visualization feature for initial visual analysis and identification of out-of-context anomalies on mobile devices.

In addition, we introduce regression code blocks so that students can interact with supervised ML algorithms on a high level, produce predictions, and see the direct effect of data cleanup on their prediction results.

(a) The code block enabling learners to detect anomalies in their dataset of choice based on the Z-Score algorithm integrated into their mobile app.



(b) The code block enabling learners to predict the line of best fit for given X and Y values. Besides the prediction values, developers can choose from the drop-down other values to calculate, e.g. correlation coefficient, X-intercept and more.

Fig. 1. Examples of code blocks from the data science toolkit integrated in App Inventor

The various data science code blocks expect as input the different columns of a dataset and as output return the prediction results in a list. App Inventor provides multiple components to store data: CloudDB, DataFile, File, Spreadsheet, and TinyDB. The dataset can also be generated through the app itself, by collecting user input or input from supported mobile sensors (e.g., light sensor, location sensor, accelerometer sensor, etc.).

B. Milestones

While the educational framework gives students the full freedom to choose their topic of interest, we set a list of milestones the students have to reach, to ensure they experience the complete data science lifecycle. Based on the data science lifecycle, we defined the following milestones:

- MS1 Project Proposal: We ask the students to identify real-world challenges that hold personal significance to them and that they wish to solve. As an artifact, they formulate a visionary scenario of their identified project idea and present it to their peers.
- MS2 Domain Research & Data Collection: We ask the students to research their chosen domain and gather data supporting their project and solution idea. As an artifact, they create a dataset in .csv format. The students can either self-generate their dataset using their own collected data or they can use a publicly available dataset.
- MS3 Data Storage and Visualization: The students explore App Inventor's data science blocks and start drafting a data and software architecture for their mobile app. They make design decisions on how to store their data and choose the most fitting data features and graph types to visualize their collected data. As an artifact, we expect them to use App Inventor to create a simple one-screen mobile app displaying their data.
- MS4 Data Cleanup: We ask students to visually explore their data in App Inventor to identify initial irregular phenomena and spikes. The students examine these irregularities more deeply and integrate anomaly detection algorithms to identify more outliers and remove erroneous data points based on their domain knowledge. The artifacts of this milestone are a revised data set and additional anomaly detection features integrated into their mobile apps.
- MS5 Predictions: Students integrate additional ML algorithms in their mobile apps, through which they can offer the app users insightful predictions. Students re-clean their data when necessary to optimize their prediction results.
- MS6 Additional Features: Before the final submission of their mobile apps, we expect the students to add more features to increase the usability of their apps. These additional features do not have to be data-driven.

The students apply their projects in an agile, scrum-based setting [15], with weekly meetings to peer-review each other's project status after each milestone.

C. Examples

We applied this approach with freshmen students. The students identified multiple real-world problems they tried to solve as part of the project proposal milestone. The problems include student debt, lack of motivation for sports, climate change, diabetes complications prevention, and more. In this section, as an example, we will present a fully working data-driven mobile app developed by a freshman student using the App Inventor data science toolkit in the context of applying DAEF.

The Diabetes Logbook App is designed to help diabetes patients manage their glucose levels. Through the app, users can log their blood glucose levels after meals, visualize their past data on a graph, and compare their levels to averages of patients with similar demographics (see figure 2).

The app automatically detects anomalies, such as mistyped data points, and provides options to remove them. Additionally, it offers personalized warnings and suggestions for self-regulating glucose levels based on real-time data inputs and an analysis of the patient's history. Users can also communicate with their doctors directly through the app in critical situations, empowering them to take control of their health easily and conveniently.

In summary, experiencing the framework changes students' perspectives on the world and themselves, as they connect their learning to real-world issues, acquire data science skills, and develop solutions to challenges significant to them and their community. They gain awareness of data science's influence on their lives, understand the importance of data integrity, and feel empowered to create data-driven products. The framework enhances students' critical thinking, problem-solving, and data literacy skills, preparing them to navigate and contribute to an increasingly data-driven society.

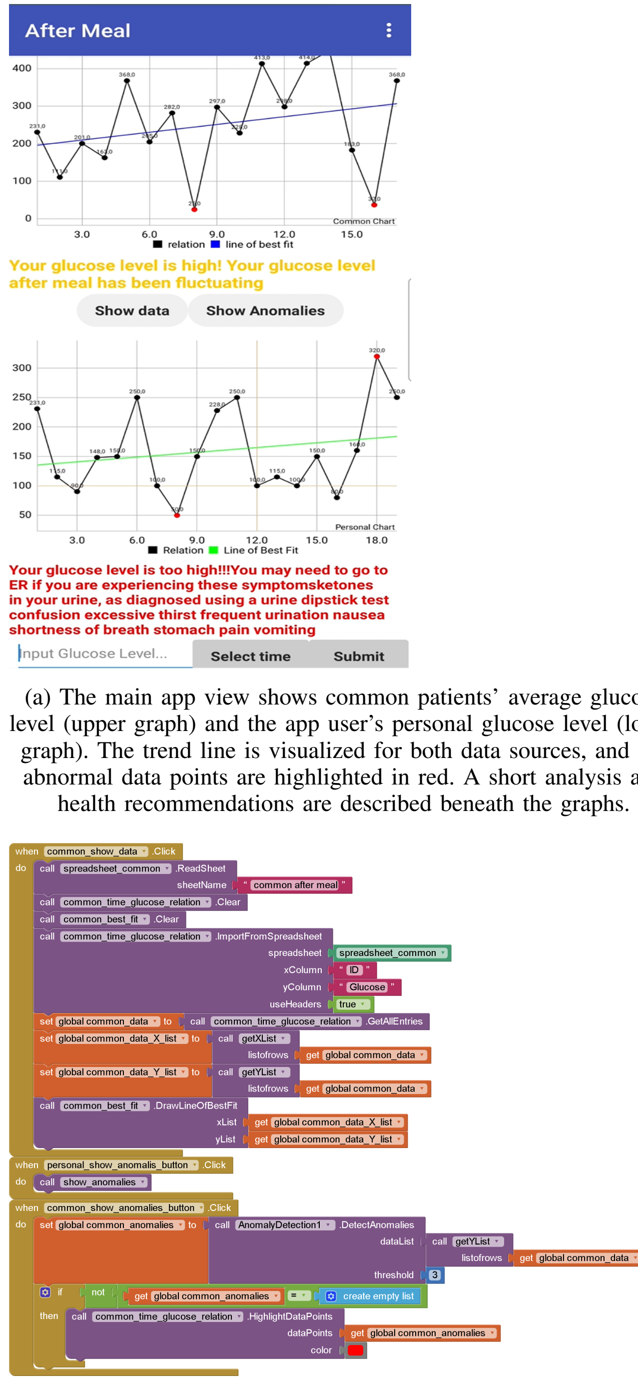
VI. Case Study

Additionally, we investigated the effectiveness of Data Action in a case study. The research questions we investigated were: (RQ1) Is DAEF effective in enabling students to recognize the impact of data science and utilize it to build impactful solutions for real-world problems? (digital empowerment) and (RQ2) Is DAEF effectively empowering students to be data thinkers (self-efficacy)?

Besides applying DAEF with freshmen students for an entire semester, we conducted a three-hour workshop with 13 students who have no prior data science experience as a research study examining the evolution of students' views before and after educational activities of DAEF. The student's views were collected through pre- and post-surveys (see 3 for exemplary items) to evaluate the effectiveness of our educational framework and App Inventor's data science blocks. During the workshop, we focused on two learning objectives of DAEF: (1) Working with real-world data critically and (2) Understanding the impact of data.

For RQ1, our findings [17] include that DAEF changed students' views on the importance of data cleanup and the effect of anomalies on their regression results. More than 45% agree that they would include data science techniques in the tech projects they create in the future. The majority believe that including data science would enhance their mobile apps (see figure 3 b).

In addition, for RQ2, students reported increased confidence in their data literacy skills. We asked the participants to answer four questions before and after the workshop that assessed the students' sense of confidence in (1) data science terminology, (2) working with real-life data, (3) data visualization, and (4) data analysis. For all four categories, we found an apparent increase in participants' confidence in their data literacy skills compared to before the workshop. The majority agree or strongly agree about their data literacy abilities, with 84.62% for confidence in data science terminology, 46.15% for confidence in working with real-life data, 61.53% for data visualization, and 69.23% for data analysis (see figure 3 a for the later). DAEF empowered students to see themselves as individuals who could translate their acquired data literacy skills to solve impactful real-world problems using real-world data. Before the workshop, only around 38% of the participants

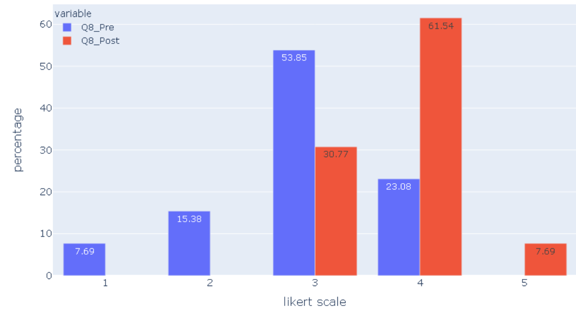


(a) The main app view shows common patients' average glucose level (upper graph) and the app user's personal glucose level (lower graph). The trend line is visualized for both data sources, and the abnormal data points are highlighted in red. A short analysis and health recommendations are described beneath the graphs.

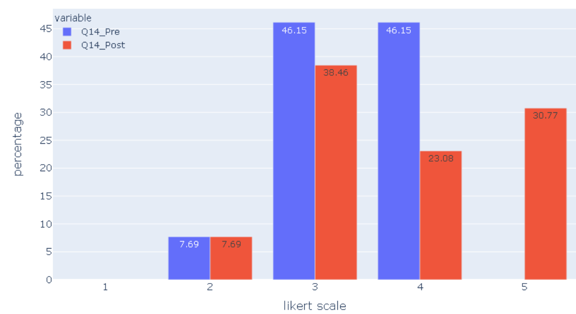
(b) Some of the app's features corresponding code blocks. The upper blocks ("`common_show_data`") are responsible for reading data from a spreadsheet, visualizing it on the graph, and drawing its corresponding line of best fit. The lower code blocks ("`common_show_anomalies_button`") are responsible for detecting anomalies on button click, and coloring them in a distinct color (red, in this example).

Fig. 2. The main view of the production-ready Diabetes Logbook app, showcasing some of the app's features

were confident in their ability to solve real-world problems using data; after the workshop, this increased to around 77%.



(a) Results of Q8: I feel confident in my ability to extract insights from data (identify patterns, identify anomalies, calculate predictions)



(b) Results of Q14: I believe using data science would enhance my apps and bring a positive impact on my community or the world.

Fig. 3. The most interesting results of the pre- and post-survey questions evaluating students' perception of data science and their skills to use data science to solve impactful real-world problem (on a Likert scale)

In summary, the students grasped the significance of data cleanup, recognizing anomalies as valuable insights crucial for addressing identified challenges. Engaging with these challenges fueled their enthusiasm for skill acquisition, with tangible outcomes instilling a sense of accomplishment. Witnessing their apps' real-world impact on societal issues profoundly transformed their perspectives, empowering them to embrace the potential of AI.

VII. Conclusion

In conclusion, the work described in this paper is a promising start for Data Action, empowering students to engage in meaningful, hands-on, data-driven projects that address real-world challenges. The paper presents a novel approach to addressing the challenges faced by software engineering education in the era of AI and data-driven technologies. Recognizing the increasing importance of data literacy and the integration of ML and AI into software applications, we propose a shift towards a curriculum that emphasizes real-world problem-solving using data science techniques. By advocating for cultivating a "data thinking" mindset among software engineering students, we aim to bridge the gap between traditional computational skills and the data-driven industry's evolving demands.

Acknowledgment

Evan Patton (Lead App Inventor Developer), Selim Tezel (Education Team Lead), Robert Parks (Curriculum Developer), Teacher Lisa Miller and her students of Medford Vocational Technical High School (Research Study Participants), Jennet Zamanova (Student & Developer of the Example App - Diabetes Logbook)

References

[1] T. H. Davenport and D. Patil. "Data scientist." In: Harvard business review 90.5 (2012), pp. 70–76.
[2] K. Mike, N. Ragonis, R. B. Rosenberg-Kima, and O. Hazzan. "Computational thinking in the era of data science." In: Communications of the ACM 65.8 (2022), pp. 33–35.
[3] Tedre, M., Denning, P., Toivonen, T. (2021, November). CT 2.0. In Proceedings of the 21st Koli Calling International Conference on Computing Education Research (pp. 1-8).
[4] M. Frank and J. Walker. "Some key challenges for data literacy." In: The Journal of Community Informatics 12.3 (2016).
[5] R. Acuña. "Developing a Data Science Course to Support Software Engineering Students". In 2023 IEEE 35th International Conference on Software Engineering Education and Training (CSEET) (pp. 127-131). IEEE. 2023.
[6] J. M. Wing. "The data life cycle." In: Harvard Data Science Review 1.1 (2019), p. 6.
[7] P. Li, X. Rao, J. Blase, Y. Zhang, X. Chu, and C. Zhang. "CleanML: A study for evaluating the impact of data cleaning on ml classification tasks." In: 2021 IEEE 37th International Conference on Data Engineering (ICDE). IEEE. 2021, pp. 13–24.
[8] R. Bhargava and C. D'Ignazio. "Designing tools and activities for data literacy learners." In: Workshop on data literacy, Webscience. 2015.
[9] R. Williams. "How to train your robot: project-based ai and ethics education for middle school classrooms." In: Proceedings of the 52nd ACM Technical Symposium on Computer Science Education. 2021, pp. 1382–1382.
[10] A. Grillenberger and R. Romeike. "Developing a theoretically founded data literacy competency model." In: Proceedings of the 13th Workshop in Primary and Secondary Computing Education. 2018, pp. 1–10.
[11] M. Tissenbaum, M. A. Sherman, J. Sheldon, and H. Abelson. "From computational thinking to computational action: Understanding changes in computational identity through app inventor and the internet of things." In: International Society of the Learning Sciences, Inc.[ISLS]., 2018.
[12] A. Patcha and J.-M. Park. "An overview of anomaly detection techniques: Existing solutions and latest technological trends." In: Computer networks 51.12 (2007), pp. 3448–3470.
[13] B. Plale and I. Kouper. "The centrality of data: data lifecycle and data pipelines." In: Data analytics for intelligent transportation systems. Elsevier, 2017, pp. 91–111.
[14] Elhashemy, H., Parks, R., Kim, D. Y., Patton, E., Abelson, H. Empowering Learners with a Low-Barrier Mobile Data Science Toolkit.
[15] Schwaber, K. (1997). Scrum development process. In Business Object Design and Implementation: OOPSLA'95 Workshop Proceedings 16 October 1995, Austin, Texas (pp. 117-134). Springer London.
[16] J. Calzada Prado and M. Á. Marzal. "Incorporating data literacy into information literacy programs: Core competencies and contents." In: Libri 63.2 (2013), pp. 123–134.
[17] Elhashemy, H. (2023). From Curious Kids to Data Wizards: An Empowerment Approach to Data Science Education.
[18] Inventor, M. A., Explore, M. I. T. (2017). App inventor. línea]. Disponible en: http://appinventor. mit. edu/explore/.[Accedido: 26-may-2015].