Offering Data Science Coursework to Non-Computing Majors

Xumin Liu
xmlics@rit.edu
Rochester Institute of Technology
Rochester, NY, USA

Erik Golen
efgics@rit.edu
Rochester Institute of Technology
Rochester, NY, USA

Rajendra K. Raj
rkr@cs.rit.edu
Rochester Institute of Technology
Rochester, NY, USA

Kimberly Fluet
kfluet@warner.rochester.edu
University of Rochester
Rochester, NY, USA

ABSTRACT

Data science courses offered by computing departments tend to be inappropriate for non-computing majors due to the emphasis on coding and a long chain of prerequisite courses in computer science and mathematics or statistics. Moreover, courses designed for computing majors by computing faculty do not always match the backgrounds and interests of students majoring in other disciplines. This paper discusses the motivation and challenges of offering an entry-level data science course for students in non-computing disciplines with limited coding experience. Experiences with the teaching of this course at the Rochester Institute of Technology are discussed. Preliminary assessment results have shown this approach to be useful.

KEYWORDS

Data science, computer science principles, non-computing majors, data science learning platform

ACM Reference Format:
Xumin Liu, Erik Golen, Rajendra K. Raj, and Kimberly Fluet. 2023. Offering Data Science Coursework to Non-Computing Majors . In *Proceedings of ACM Conference (Conference'17).* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Data science (DS) has rapidly gained tremendous popularity and broad adoption across the spectrum of application domains, from science to the arts, engineering to business, and so on. The potential for efficiently discovering valuable knowledge from large amounts of data gathered in these areas has created an increasing workforce demand for data science expertise within their own disciplines. As a result, DS curricula have been called for in many disciplines over the past few years [5, 7, 16].

DS education is increasingly viewed as being founded on three pillars: computing, statistics/mathematics, and domain knowledge [10]. The latest report on DS curricular competencies defined by the ACM Data Science Task Force appears to be targeted at computing majors with computing-heavy knowledge areas: "artificial intelligence; big data systems; computing and computer fundamentals; data acquisition, management, and governance; data mining; data privacy, security, integrity, and analysis for security; machine learning; programming, data structures, and algorithms; and software development and maintenance" [9]. It also requires competencies in "calculus, discrete structures, probability theory, elementary statistics, advanced topics in statistics, and linear algebra, among others" [9]. Therefore, many DS courses are offered for computing majors and often have prerequisite requirements on programming (such as Python) and mathematics (such as Linear Algebra and Statistics). They are usually offered in the late years of college or at the graduate level. Such programming-focused, advanced-level courses are not suitable for those having limited or no computing backgrounds, such as elementary to high students or non-computing majors. First, those students may not be ready to take these courses due to the required prerequisites. Second, the courses may not serve their needs of learning DS properly due to the focus on the programming part. Therefore, current DS courses may not cover the education demands of students in non-computing majors.

Some non-computing academic programs, e.g., business and biology, have started to offer courses that cover data analytics topics. However, many other programs lack instructor-related resources and other support to provide DS education to their students. It is also recognized that, just like computing skills, DS concepts should be introduced to students as early as possible to equip them with useful data handling and analytic skills, inspire their interests, and even prepare them to take advanced DS courses. As we are all living in this data era, how data is collected, handled, and used, could make a tremendous impact on our daily lives. Students need to be aware of data ethics challenges and learn how to properly share and handle data, regardless of the career path they will follow. All of these considerations call for an entry-level course that is designed for non-computing majors who will learn important data ethical topics and apply DS techniques to sciences, engineering, business, the humanities, or perhaps others as general education.

The remainder of this paper is organized as follows. We discuss several key challenges of offering an entry DS coursework to non-computing majors and ensuring the success of students in the classroom in Section 2. In Section 3, we describe the design and setting of an entry-level DS course with the incorporation of a web-based learning platform. In Section 4, we describe the deployment

Conference'17, July 2017, Washington, DC, USA

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

and results of the course. In Section 5, we discuss some related work. We conclude this paper in Section 6.

2 WHAT ARE THE CHALLENGES?

In this section, we discuss several common challenges an instructor may face when offering such an entry-level DS course for non-computing majors.

Achieve desirable learning outcomes. Teaching DS to non-computing students introduces the challenge of potentially limited learning outcomes due to the little or no prerequisite knowledge in computing. Besides the high requirement on statistics and probabilistic knowledge to understand data patterns and the rationale of choosing different machine learning models, coding is one of the important components in DS, which is intensively needed for hands-on exercises of using current DS packages in R or Python (e.g., pandas, numpy, and scikit-learn). For students that do not have sufficient background in coding and computational thinking, the instructors need to allocate enough class time to the coding part, potentially leaving less room for DS-specific topics in the class. Other courses described in Section 5, such as the one offered at the University of Sydney [6] and the one offered at the Boston University [17], often cover more programming than our proposed course but might have missed several DS topics that we considered important. As an alternative, the instructors can put less emphasis on implementation details and teach the topics on a purely conceptual level. In a nutshell, due to the tight linkage between hands-on experience and coding assignments, standard convention assumes that students may have limited, or no, chances of applying the knowledge to solve real-world problems, but this does not need to be the case [19].

Solve problems in various domains. Students should not only learn DS skills but also be exposed to problems in various application domains to understand how these newfound skills can be applied in different contexts [15]. A high-quality design and delivery of the course materials usually go beyond the expertise level of individual instructors and require collaboration among a group of faculty with diverse backgrounds that are complementary to each other. For example, instructors from non-computing disciplines can contribute their domain knowledge to design course projects and case studies as well as give guest lectures to explain the general background and the customary data collection process, provide guidelines on understanding the data, interpret the analysis results, and more. Instructors from computing disciplines can teach students to perform those tasks in computational ways while not having to tenuously include domain-specific content that they may not be comfortable with.

Set up accessible lab environments. Hands-on practice is a major component of DS education, where students learn how to apply concepts and techniques to real-world data problems. It is also an effective way to engage students and improve their learning experience. In many DS courses, students are required to use either R or Python to implement certain DS tasks, such as exploring or cleaning data. Students may be given instructions on setting up the lab environment, including downloading, installing, and configuring an IDE to write and run code. For those who have little or no programming background, students can easily be overwhelmed by those steps and issues they may come across, such as the inconsistency between Python versions and the lack of required plugins. Students can also be confused by the instructions if they use different OSs or have different system settings than the ones specified in the document. Some online IDEs, such as Google Colab and myBinder, have gained popularity as a tool to support class demos and hands-on exercises since they allow students to run their Python code without the need of a local Python IDE. But still, the use of languages to code and debug errors or unexpected results can also go beyond the grasp of non-computing majors. Some tools, such as Rattle R, RapidMiner, and Weka, provide different levels of convenience through graphical user interfaces for data analysis and visualization while sacrificing some functionality. However, they are still mainly meant for computing students. Moreover, user interfaces and documentation for these tools are not always friendly to users with a limited background. Non-computing majors often struggle with proper usage of these tools, for example, choosing the right menus and options, providing appropriate input, tuning parameters, understanding and utilizing intermediate results to decide on the next steps, interpreting the final result, and so on.

3 DSP COURSE DESIGN

In this section, we describe the design of the Data Science Principles course, including its overall coverage, a tentative teaching schedule, the hands-on component, and the expected learning outcomes.

3.1 Course coverage and Learning Outcome

The DSP course is meant to provide students, who have completed the AP-CSP (Advanced Placement Computer Science Principles) [8], a high-school course designed for US high school students interested in taking college-level coursework in computing, with additional coursework in computing with data. With the proper support, the course need not be coding intensive. Instead, students are exposed to all the important techniques in the data lifecycle, including data acquisition, preparation and integration, model development and deployment, visualization, and storytelling. Students gain inferential thinking skills with common statistical and data mining techniques as they analyze real-world datasets. It also includes data security and privacy, as well as fairness and biases in data and algorithms. The course learning outcomes are:

(1) Solve a variety of data-oriented problems by exploring different phases of the DS lifecycle.
(2) Communicate the results of data analysis visually or using storytelling;
(3) Explain social, legal, and/or ethical implications of datasets and algorithms used to address problems and challenges;
(4) Reflect on DS concepts and inferential thinking skills learned throughout the term.

3.2 Course topics and teaching schedule

The tentative topic sequence of the course offered in a typical 14-week semester setting is described in Table 1. They cover the complete list of tasks that are commensurate with the standard data to knowledge pipeline [18]. The lecture slides of this course

have been included in OpenDS4All [3], an open-source project that provides DS curricular materials for the public.

The design of the two-phase case studies provides students with intensive hands-on experience after they have learned a set of interrelated DS topics; one set for preliminary investigation and another for in-depth analysis of data. Students will be given several questions to answer through analyzing data and will be encouraged to come up with their own questions and stories from the data. The questions will be designed with different difficulty levels while sufficiently covering the learned topics. As these case studies will be conducted in class, students will get instant support from the instructor when they face coding challenges or other issues.

Table 1: Principles of the Data Science Topic Sequence

Week	Topics
1	**Introduction to Data Science**: Students will learn the definition, motivation, and applications of data science; data fairness, privacy, and ethics. **Basic data types**: Students will learn String, integer, float, boolean, list, array, dictionary, and others used in a typical data file.
2	**Getting started with data**: Students will learn how to load data from formatted files such as CSV, understand a data summary, and perform basic queries on data. **Data cleaning**: Students will learn about data quality issues and their causes, as well as the techniques to address them.
3	**Data exploration and visualization**: Students will learn about summary statistics, probability theory, and basic visualization techniques such as histograms, bar charts, line graphs, and scatter plots.
4-5	**Case Studies-Phase I**: Students will work individually on datasets from two different domains, such as COVID data and US census data. For each domain, students will learn how to apply what they have learned in Weeks 1-3 to perform a preliminary investigation of the data.
6-7	**Feature selection and engineering**: Students will learn the basic idea of data sampling and aggregation, basic techniques for feature engineering, such as attribute renaming and retyping, data transformation, data standardization, data re-scaling, feature selection through correlation analysis, and PCA.
8-11	**Data analytical models**: Students will be exposed to several machine learning models for data analysis, such as data clustering, classification, and regression. Instead of learning the algorithms and theories behind the models, students will focus on how to use the models and interpret the result of applying the models. They will also learn how to evaluate the quality of a model, data imbalance, and modeling overfitting issues.
12-13	**Case studies - Phase II & Guest Lecture**: Students will continue working with datasets they used in Weeks 4-5 and learn how to apply what they have learned in Weeks 6-11 to discover interesting patterns/knowledge from the data for storytelling. They will learn the impact of choosing different techniques in feature selection/engineering and data analysis on model learning. They will learn how to interpret the learning result to tell a story of the data. A Guest lecture will be conducted by an instructor from a non-computing discipline to describe their specific DS problems, how the data is handled and analyzed, and how the analysis results can help generate knowledge and make decisions.
14	**Data ethics**: Students will reinforce and strengthen their knowledge of data fairness, privacy, and ethical challenges in each step of the data-to-knowledge pipeline they have learned in the previous weeks. They will wrap up their storytelling by discussing data ethics in the selected domains.

3.3 Hands-on Component

Hands-on practice is essential to DS education as it is an effective way to reinforce student learning and train their skills in applying knowledge to solve real-world problems. However, as we discussed earlier, it is challenging to provide in-depth hands-on practice to non-computing majors, given their limited computing background. To address this, we have developed a web-based Data Science Learning Platform (DSLP) [2, 14], which works as a middleware between users (i.e., students or instructors) and existing DS libraries to create an accessible lab environment (note: the platform will be free to use and the URL will be released in the final version of the paper). This is inspired by the success of Scratch, a project of the Lifelong Kindergarten Group at the MIT Media Lab [13], and Brockly, a JavaScript library for building visual programming editors developed by Google for Education [11] for teaching kids computational thinking through an interactive framework instead of teaching them how to write code.

Following the same idea, we have developed the DSLP that supports students to focus on the high-level workflow of processing and analyzing data without the need for writing code. DSLP works as a middleware between users (i.e., students or instructors) and Python DS libraries (e.g., Scikit-Learn, Pandas, Matplotlib, and NumPy) due to their great popularity and comprehensive support for data science tasks. It is web-based so it can be accessed through web browsers from computers, tablets, and smartphones.

With the DSLP, students can perform various data science tasks with different levels of coding efforts, including writing the code from scratch using an in-house sandbox, revising a code template for a similar task automatically generated by the DSLP, and writing no code but only using the DSLP interface to perform the tasks. This makes the platform suitable for students with different coding skills and learning expectations. As shown in Figure 1, students can use the interface to specify the variables, the target, and the techniques used to perform a feature selection task. The DSLP then generates the selection result and the corresponding Python code for students if they want to further work on the task.

Hands-on learning consists of in-class demos and take-home assignments. The instructors can use the provided in-class demos to teach students computational thinking and coding skills. Students will be introduced to Python and DS modules, such as *Pandas* and *Matplotlib*. Meanwhile, students will be given detailed instructions on how to directly use the DSLP to perform the same task, which allows them to proceed without the need to write code. Students will strengthen their understanding of what they have learned from classes through take-home assignments. The assignments cover both coding, where Google Colab is used as the IDE, and the use of the DSLP.

3.4 Course Grading

Students in the course are evaluated through traditional grading methods, such as in-class exercises, take-home assignments, quizzes, in-class discussions, midterm/final exams, project presentations, and reports. Students work in groups on a term project, where students choose the dataset that is appropriate for the course and matches their interests. Through the project, students can practice what they have learned from the course in real-world data analysis

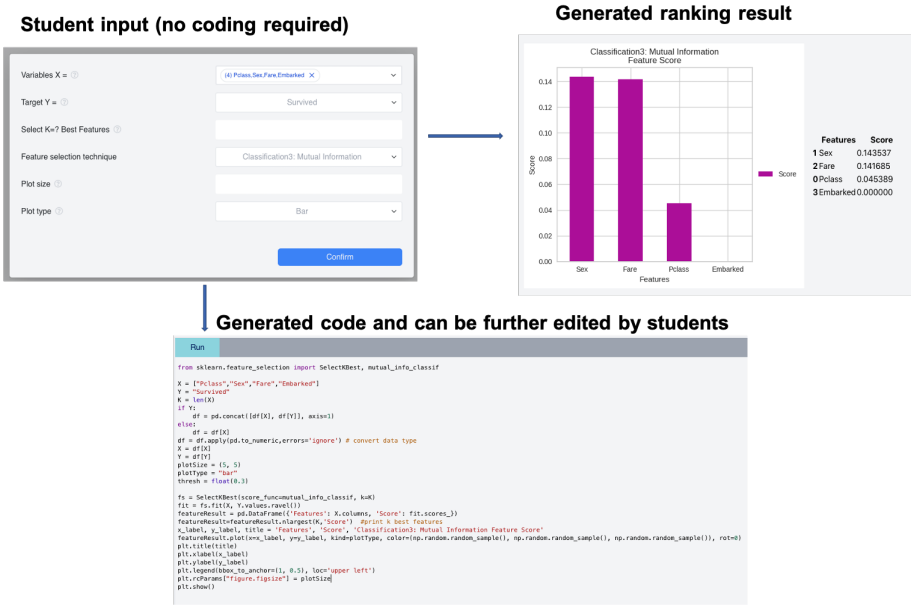

Figure 1: Using the DSLP to Perform Feature Selection Task for Titanic Dataset

problems. They are allowed to use Python, the DSLP platform, or both to implement the DS tasks. An example of such a project is to study the impact of mandated sexual education and other factors on teenage abortion rates. In this project, students collected the data offered by Guttmacher Institute, cleaned the data such as removing those incomplete records and irrelevant columns, filtered the data to focus on several years, visualized the patterns related to different factors such as predominant political parties of states, pregnancy rates, and age groups, preprocessed the data such as correcting data types and normalizing data, engineered some features such as converting numerical values to categorical ones, selected relevant features based on variances and correlations, generated a regression model to learn the impacts, and discussed the data ethical issues related to data privacy and biased models. In the end of the semester, students presented their work in class and discussed the results with other students. The exams focus on whether students can answer questions and solve a problem specific to data science in an open book/open notes setting, to narrow their coding skills gap and avoid the need for pure memorization. In the final exam, with or without giving a concrete problem, students are asked questions such as *what features do we tend to remove through feature selection*, or *list the information needs to be determined before performing KMeans clustering.*

4 COURSE DEPLOYMENT AND RESULTS

The DSP course described above was taught to non-computing majors in the Fall of 2021, the Spring of 2022, and the Fall of 2022 at Rochester Institute of Technology with a total enrollment of 34. 26 students who consented to include their data in the research.

4.1 Student Background

Student demographics are as follows: 81% of the participants were male, while 19% were female. When asked to mark all that apply, 24% identified as Asian, 6% as African-American/Black, 3% American Indian or Alaskan Native, 62% as white, 3% as Native Hawaiian or other Pacific Islander, and 3% as other. Hard-of-hearing students constituted 4% of student respondents, while 96% identified as hearing. Student majors ranged from Accounting to Film Production to Political Science and History. Management Information Systems was the most common major, identified by 5 of the 26 students. The course was required by the majors of only 4 of the 26 students. None of the students were first-year students. Second-year students were 16% of participants, 24% of students were third-year students, leaving 60% of students as fourth- or fifth-year students. When asked to identify all of their past experiences in programming, 4% had no prior experience with programming, 36% had informal experiences before college, and 40% had informal experiences during college. The percentage of participants with formal programming classes before college was 37% while 64% of students had other computing classes in college before this course offering. In terms of proficiency in programming languages, 4 students self-described as not proficient in any language leaving 22 students indicating proficiency in programming languages, in order of most to least mentioned: python, C++, MATLAB, Java, C, SQL, HTML, CSS, Arduino, and R.

4.2 Student Perceptions and Feedback

An end-of-course survey was administered to students to collect: student demographics, and past experiences with programming, and to assess the DSLP modules (lectures and exercises), the Google CoLab exercises, the perceived effectiveness of the DSLP as the course learning platform, and participants interested in additional DS courses. Likert-type questions and open-ended prompts were provided for each course component. The Likert-type questions included a 5-point Likert scale: for "easy to follow," "impact on the understanding of DS," "increase of interest in DS," and "impact on belief in abilities to perform similar DS tasks." Below we report the results from the student surveys, which were encouraging.

Examples of survey questions include *I felt the DSLP Assignments were easy to follow* and *The Google Colab Assignments improved my understanding of Data Science.*

Lectures. At the end of the course, students rated the lectures associated with the modules easy to follow with 56% indicating Agree or Strongly Agree (A+SA), 24% indicating Neutral, and 20% Disagree or Strongly Disagree (D+SD). Results for the lectures as improving students' understanding of DS were 64% indicating A+SA and 32% Neutral, while 4% indicated D+SD. Lectures were rated as increasing interest in DS by 40% (A+SA) of students, while 36% of respondents marked Neutral and 24% indicated D+SD. When asked if students believed they could perform similar DS tasks to those in the lectures 67% indicated A+SA, 33% were Neutral, with 4% indicated D+SD.

DSLP exercises and Google CoLab exercises. In terms of the DSLP exercises, a large majority of students felt that the DSLP exercises were easy to follow (84% A+SA, 8% Neutral, 8% D+SD). They also rated the DSLP exercises as improving their understanding of DS and increasing their ability to perform similar DS tasks equivalently: 68% A+SA, 24% Neutral, 8% D+SD, which are very encouraging results. In terms of the DSLP exercises' impact on their interest in DS students indicated 44% A+SA, 40% Neutral, and 16% D+SD. Alternatively, students rated Google CoLab exercises as easy to follow with 60% A+SA, 32% Neutral, and 8% D+SD. For both improvements in DS understanding and ability to perform similar DS tasks students rated Google CoLab as 68% A+SA, 24% Neutral, and 8% D+SD. Lastly, for Google CoLab's increasing interest in DS, students indicated 48% A+SA, 32% Neutral, and 20% D+SD.

DSLP Learning Platform. Students were also asked to rate the DSLP as a learning platform for ease of use, whether they liked using the DSLP to explore DS if the DSLP improved their understanding of DS, and if it improved their confidence in conducting DS inquiries and analytical tasks. Encouragingly, 68% and 72% of students rated A+SA, respectively, for ease of use and liking the DSLP to explore DS. Similarly, 68% and 64% rated A+SA, respectively, for the DSLP improving their understanding of DS and for the DSLP improving their confidence in similar DS inquiries and analytical tasks. Students were also asked about their use of the, which was voluntary since using the DSLP for the course project was optional. 32% of students used the DSLP to help with the code involved in their project, 12% used it for hints, and 56% didn't use it at all. Of those that used it, 52% found it Very Helpful or Somewhat Helpful to complete their project, 32% were Neutral about its helpfulness, and 16% found it not helpful at all.

Lastly, we assessed the perceived impact of the course on students' interest in future DS courses and student feedback for the course and DSLP improvements. There were strong results in interest in taking a future DS course: 72% Definitely or Probably Interested, 4% Might or Might Not Be Interested, and 24% Probably Not or Definitely Not Interested. In terms of feedback from students, students reported they found some bugs in the DSLP distracting and at times frustrating. Other students wanted more instruction in coding in Python. This indicates that the quality of the support for hands-on exercises impacts students' learning and interest in data science. Regardless of some complaints about the DSLP bugs, students explicitly pointed out the positive impact of the course and the DSLP on their learning, "I feel I got most of my learning done through these [DSLP exercises], and the DSLP platform was extremely useful" [Student 2] and, "Really nice intro course to data science, made taking the Business Intelligence class alongside it more manageable" [Student 9].

5 RELATED WORK

Sullivan [17] describes an introductory data-centric computing course at Boston University for non-computing majors. The course covers the topic of data management, Python programming, data visualization, and data mining. Anderson et al. [4] describe an introductory programming course at four different institutions, which teaches entry-level undergraduate students, including non-computing majors and non-traditional students (e.g., older adult students), basic programming and computing concepts through processing and analyzing real-world datasets. Burridge et al. [6] discuss a similar introductory programming course at the University of Sydney, teaching students in data science and non-computing majors how to write code to perform fundamental data exploration and analysis tasks. Krishnamurthi et al. [12] propose to integrate data science components into an introductory computing course to make it accessible for students with diverse backgrounds. While still covering traditional programming concepts, such as data types and recursion, the course introduces some important data-related concepts, such as data frames and visualization. The UC Berkeley Data Science 8 course has been a well-known effort for teaching basic DS topics to entry-level undergraduate students [21]. The course teaches students computational and inferential thinking, focusing on the programming aspects of DS. It covers various DS topics, such as classification and clustering. Mine Çetinkaya-Rundel et al. describe an entry-level DS course for non-computing majors, covering R programming for the entire DS pipeline, at Duke University [1, 22]. Compared to these courses, our course further reduces the focus on programming concepts but exposes students to additional DS topics, especially those related to data preprocessing, feature engineering and selection, and data analytics. With the use of the DSLP, we believe it can better serve students with diverse backgrounds and interests as students can learn and practice DS concepts without worrying about the programming details.

CODAP (Common Online Data Analysis Platform) is a free online platform that provides a convenient way to browse, visualize, analyze, and simulate data without the need for writing code [20]. It has been a great success in terms of allowing students with no coding background, especially those in middle or high schools, to explore and analyze real-world datasets. Our DSLP targets college students and covers more in-depth DS methods such as those related to feature selection and machine learning models. It also teaches students computational thinking through code exemplification and sandbox.

6 CONCLUSION

We presented the motivation, challenges, and design of an entry-level Data Science Principle course to non-computing majors. The assessment results indicate the success of the course and the effectiveness of using the DSLP platform to break the coding barrier for students to learn DS topics.

ACKNOWLEDGEMENT

This material is based upon work supported by the National Science Foundation under Award IUSE 2021287. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation. The authors also thank the anonymous reviewers for their feedback.

REFERENCES

[1] . [n.d.]. Data Science Course in a Box. https://codap.concord.org/.
[2] . [n.d.]. Data Science Learning Platform. http://dslp.cs.rit.edu:8000/.
[3] [n.d.]. OpenDS4All. https://github.com/odpi/OpenDS4All.
[4] Ruth E. Anderson, Michael D. Ernst, Robert Ordonez, Paul Pham, and Ben Tribelhorn. 2015. A Data Programming CS1 Course. In *Proceedings of the 46th ACM Technical Symposium on Computer Science Education (SIGCSE '15)*. 150–155.
[5] Austin Cory Bart, Dennis G. Kafura, Clifford A. Shaffer, and Eli Tilevich. 2018. Reconciling the Promise and Pragmatics of Enhancing Computing Pedagogy with Data Science. In *Proceedings of the 49th ACM Technical Symposium on Computer Science Education, SIGCSE 2018, Baltimore, MD, USA, February 21-24, 2018*, Tiffany Barnes, Daniel D. Garcia, Elizabeth K. Hawthorne, and Manuel A. Pérez-Quiñones (Eds.). ACM, 1029–1034. https://doi.org/10.1145/3159450.3159465
[6] Joshua Burridge and Alan Fekete. 2022. Teaching Programming for First-Year Data Science. In *ITiCSE '22: Proceedings of the 27th ACM Conference on Innovation and Technology in Computer Science Education*, Vol. 1. 297–303.
[7] Lillian N. Cassel, Michael Posner, Darina Dicheva, Don Goelman, Heikki Topi, and Christo Dichev. 2017. Advancing Data Science for Students of All Majors (Abstract Only). In *Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education, Seattle, WA, USA, March 8-11, 2017*, Michael E. Caspersen, Stephen H. Edwards, Tiffany Barnes, and Daniel D. Garcia (Eds.). ACM, 722. https://doi.org/10.1145/3017680.3022362
[8] College Board. 2023. AP Computer Science Principles. https://apcentral.collegeboard.org/courses/ap-computer-science-principles/course.
[9] Andrea Danyluk, Paul Leidig, Scott Buck, Lillian Cassel, Maureen Doyle, Keegan Hines, Tin Kam Ho, Andrew McGettrick, Suzanne McIntosh, Jian Pei, Weining Qian, Karl Schmitt, Christian Servin, and Hongzhi Wang. 2021. *Computing Competencies for Undergraduate Data Science Curricula*. Technical Report. ACM. https://www.acm.org/binaries/content/assets/education/curricula-recommendations/dstf_ccdsc2021.pdf ACM Data Science Task Force.
[10] Andrea Danyluk, Paul Leidig, Lillian Cassel, and Christian Servin. 2019. ACM Task Force on Data Science Education: Draft Report and Opportunity for Feedback. In *Proceedings of the 50th ACM Technical Symposium on Computer Science Education* (Minneapolis, MN, USA) *(SIGCSE '19)*. ACM, New York, NY, USA, 496–497. https://doi.org/10.1145/3287324.3287522
[11] Google for Education. [n.d.]. Blockly: A JavaScript library for building visual programming editors. https://developers.google.com/blockly/.
[12] Shriram Krishnamurthi and Kathi Fisler. 2020. Data-centricity: a challenge and opportunity for computing education. In *Communications of the ACM*, Vol. 63. 24–26.
[13] MIT Media Lab. [n.d.]. Scratch. https://scratch.mit.edu/.
[14] Xumin Liu, Erik Golen, and Rajendra K. Raj. 2022. DSLP: A Web-based Data Science Learning Platform to Support DS Education for Non-Computing Majors. In *SIGCSE 2022: The 53rd ACM Technical Symposium on Computer Science Education, Providence, RI, USA, March 3-5, 2022, Volume 2*, Larry Merkle, Maureen Doyle, Judithe Sheard, Leen-Kiat Soh, and Brian Dorn (Eds.). ACM, 1181. https://doi.org/10.1145/3478432.3499255
[15] Jeffrey Oliver and Torbet McNeil. 2021. Undergraduate data science degrees emphasize computer science and statistics but fall short in ethics training and domain-specific context. *PeerJ Computer Science* 7 (03 2021), e441. https://doi.org/10.7717/peerj-cs.441
[16] Jeffrey S. Saltz, Neil I. Dewar, and Robert Heckman. 2018. Key Concepts for a Data Science Ethics Curriculum. In *Proceedings of the 49th ACM Technical Symposium on Computer Science Education, SIGCSE 2018, Baltimore, MD, USA, February 21-24, 2018*. 952–957. https://doi.org/10.1145/3159450.3159483
[17] David G. Sullivan. 2013. A data-centric introduction to computer science for non-majors. In *SIGCSE '13: Proceeding of the 44th ACM technical symposium on Computer science education*. 71–76.
[18] Pang-Ning Tan, Michael Steinbach, and Vipin Kumar. 2016. *Introduction to data mining*. Pearson Education India.
[19] Matti Tedre, Peter Denning, and Tapani Toivonen. 2021. CT 2.0. In *Proceedings of the 21st Koli Calling International Conference on Computing Education Research* (Joensuu, Finland) *(Koli Calling '21)*. Association for Computing Machinery, New York, NY, USA, Article 3, 8 pages. https://doi.org/10.1145/3488042.3488053
[20] The Concord Consortium. [n.d.]. Common Online Data Analysis Platform (CODAP). https://codap.concord.org/.
[21] UC Berkeley. [n.d.]. Data 8: Foundations of Data Science: A Data Science Course for Everyone. https://data.berkeley.edu/education/courses/data-8.
[22] Mine Çetinkaya Rundel and Victoria Ellison. 2021. A Fresh Look at Introductory Data Science. *Journal of Statistics and Data Science Education* 29, sup1 (2021), S16–S26. https://doi.org/10.1080/10691898.2020.1804497 arXiv:https://doi.org/10.1080/10691898.2020.1804497