Hands-on Assignments for Practical Data Science Education to Non-Computing Majors

Xumin Liu, and Erik Golen
{xmlics, efgics}@rit.edu
Golisano College of Computing and Information Sciences
Rochester Institute of Technology

Abstract

It is important to provide non-computing majors with hands-on experience when teaching them data science topics. Meanwhile, this is challenging since those students typically have limited, or no, computing background. This paper describes our experience in offering two types of hands-on assignments in an entry-level data science course for non-computing majors; one with coding tasks and the other, without. Data sets from various domains were used to diversify the types and requirements of those tasks. We evaluated the two types of hands-on assignments and compared how effectively they helped students understand data science topics and improve students' interest in data science and computer science.

Introduction

Data science curriculum is in high demand due to the increasing workforce requirements in many disciplines[1,2,3]. Hands-on practice is an essential component in data science education, as it can not only effectively enhance student understanding of fundamental concepts and techniques learned in class, but also, improve their capability of applying their learning to solve real world problems in various application domains[4,5]. Students are usually required to learn how to program (such as writing code in Python) before taking a data science course. This makes it challenging to offer meaningful data science education to those who have limited, or no, programming background, such as K-12 students or non-computing majors at the college level. In this paper, we describe our experience in offering two types of hands-on assignments in an entry-level data science course for non-computing majors at the Rochester Institute of Technology. In the first assignment type, students were required to write Python code with the support of sample code given through in-class demos, to perform various data science tasks. This allows instructors to teach students computational thinking and coding skills. In the second type of assignment, students were required to perform in-depth data manipulation and analysis tasks on a web-based Data Science Learning Platform (DSLP), where little or no programming is required. The Graphic User Interface (GUI) of the DSLP also maps the tasks to Python code by generating code that can accomplish the tasks. This provides students with hands-on experience

in performing data science tasks without being limited by their coding capabilities.

The remainder of the paper is organized as follows. First, we describe the design of the assignments, with an example of each type. We then present the results of a survey we conducted to evaluate how effectively the hands-on assignments helped students improve their interest and knowledge in data science.

Assignment Design

We designed two types of hands-on assignments for the class; Google Colab Assignments, using Python, and DSLP assignments. Google Colab Assignments were facilitated by the instructor performing Python coding demos in class to teach students which data science libraries and commands should be used for given tasks. The overall purpose of the Google Colab Assignments was thus to provide students with an opportunity to write their own code or modify existing code. The purpose of the DSLP Assignments was to allow students to perform the same or similar data science tasks that may beyond their coding capabilities, as well as reinforce their knowledge in both data science and coding.

Google Colab Assignments

Students were taught data science topics for a typical data to knowledge pipeline, including data querying/selection, cleaning, exploration, visualization, pre-processing, feature engineering, feature selection, and data mining[6]. We designed the two types of assignments for each topic and used several datasets published on Kaggle, including data about the Titanic disaster, iris flowers, the US census, carseat sales, advertising, automobile, house rentals, and COVID vaccinations.

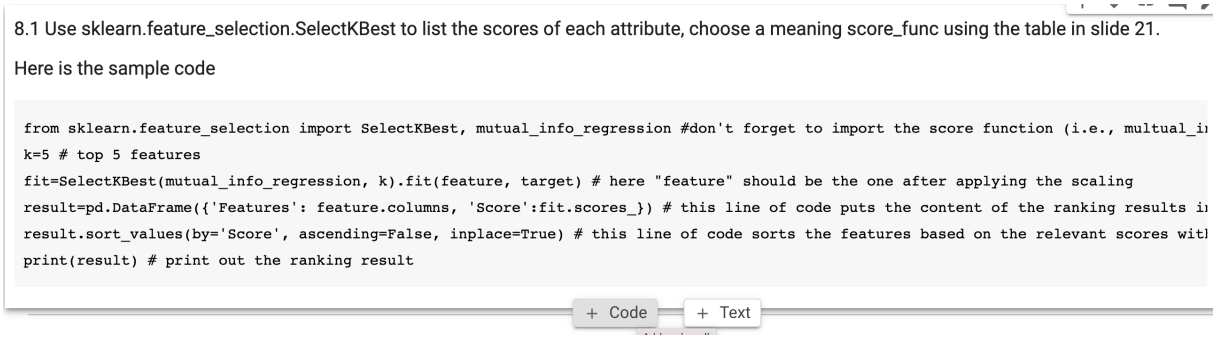

Figure 1: A snippet of Google Colab Assignment for the Feature Selection Topic

In the Google Colab Assignments, students were required to write Python code to accomplish a given task. Each assignment included a template Google Colab Python notebook file, which was read-shared with students. Data science tasks were decomposed into steps, where each step was described in a text block, along with appropriate hints for how to proceed. Expected output was often given to help students verify their answer. Students were asked to add a code block underneath the text block and write their own code for that step. For example, in the feature selection assignment, students were asked to work on the carseat sales dataset[7] to practice some feature selection techniques, including removing attributes with low zero variance, removing

redundant attributes by examining their correlation, applying the sklearn feature selection library, i.e., SelectKBest, to rank the features, and using Principle Component Analysis (PCA) to create a new and lower-dimensional feature spaces. Figure 1 shows a snippet of the assignment, where students were asked to display relevancy scores for the Top 5 most relevant features in the data set by using the specified sklearn library. In addition to the task description, students were also provided assistance, such as a reference to the corresponding lecture slide, sample code, and an explanation of the code. The complete assignment may be found here [1]. Beyond the guides in the assignment, students were also taught how to write the code through in-class demos, where the instructor wrote the code together with the students for a similar task, but with a different data set.

DSLP Assignments

In the DSLP Assignments, students were asked to perform data science tasks using a web-based Data Science Learning Platform (DSLP). The platform was developed as a deliverable for our NSF project[8]. Its purpose is to support student learning of data science topics, regardless of their programming background. As shown in Figure 2, users can perform data science tasks through the web-based Graphic User Interface (GUI). The tasks are automatically translated into Python code by the DSLP through a code examplification component and is then processed in a backend server. Students can see both the generated code, which helps them learn how to code, and the results returned from the backend server, which accomplishes the task specified via the GUI. A code sandbox component allows students to modify and run the generated code for a specific task or try their own code in the current context of the task, such as using a different feature in the dataset.

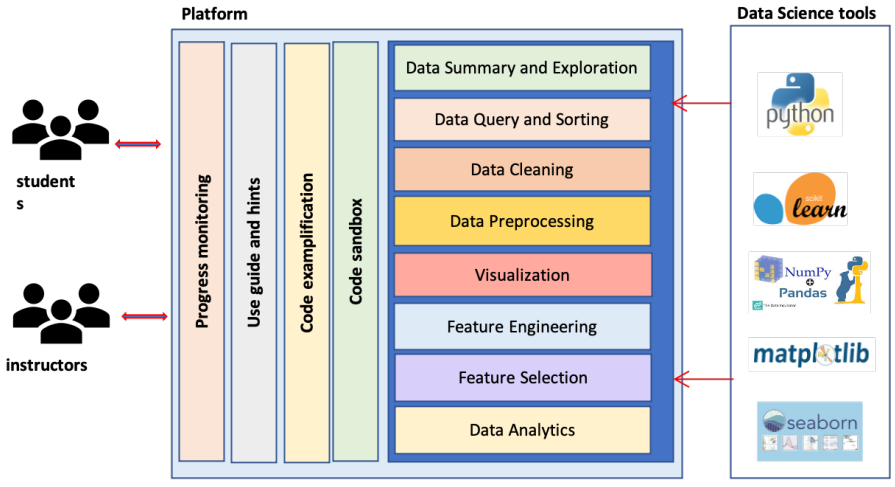

Figure 2: An Overview of the Data Science Learning Platform

In each DSLP assignment, students were given a detailed explanation of the concepts and techniques covered in that assignment, as well as step-by-step instructions for using various DSLP modules to answer questions. For example, in the feature selection assignment, students

[1] https://drive.google.com/file/d/1tCuBAScBGqpzBTVtflifCRgyk8Bt3zIS/view?usp=sharing

were asked to use the DSLP to identify irrelevant features, redundant features, and the most important features. To identify the most important features, students were given the following instructions:

In the Feature Selection lecture, we looked at several methods for comparing the potential utility of features towards the class feature as part of a classification task; these include ANOVA, Chi-Squared, and Mutual Information. All of these methods assume a categorical target variable, but differ in terms of what they expect for the remaining features. ANOVA assumes continuous features, Chi-Squared assumes categorical features, and Mutual Information assumes both continuous and categorical features.

Looking at the remaining features in our dataset (PClass, Sex, Fare, and Embarked), we see a mix of continuous (Fare) and categorical (PClass, Sex, Embarked). Based upon these, we should use Mutual Information. To create a Mutual Visualization analysis, go to the Feature Selection module, click the Select operation button and choose PClass, Sex, Fare, and Embarked as the X variables and Survived as the Y variable. Next, choose "Classification 3: Mutual Information" for the Feature selection technique and Bar as the Plot type. Click Confirm.

We can see clearly from the result that Sex and Fare have provided the most Mutual Information in relation to the target variable, Survived.

Questions were designed for each assignment to test if students followed the steps and understood the output of each step properly. Examples of those questions are:

1. Based upon the ranking result, should we choose Fare or Pclass to continue with in our analysis?

2. Should we continue to use the Embarked feature in our analysis? Why or why not?

The complete DSLP assignment for feature engineering may be found here [2].

As shown in Figure 3, below, after performing these steps using the DSLP, students were given both a feature ranking result visualized in a bar chart and the corresponding Python code, which is editable for students to make any further changes.

Assignment Assessment

We used these two types of assignments in an entry-level data science course for non-computing majors at the Rochester Institute of Technology in Fall 2021, Spring 2022, and Fall 2022. We conducted an end-of-course survey to collect the information on student demographics, past experiences with programming, and assessment of the two types of assignments in the course. 26 of the students consented to have their data included in the research. The evaluation focused on how effectively the assignments helped students (1) understand data science principles and practices, and (2) improve their self-efficacy about, and interest in, data science and computer science.

[2] https://docs.google.com/document/d/1vtdapq_x5ptJwMHScB5nURWeDcw-0ezu/edit?usp=sharing&ouid=117950654858910132940&rtpof=true&sd=true

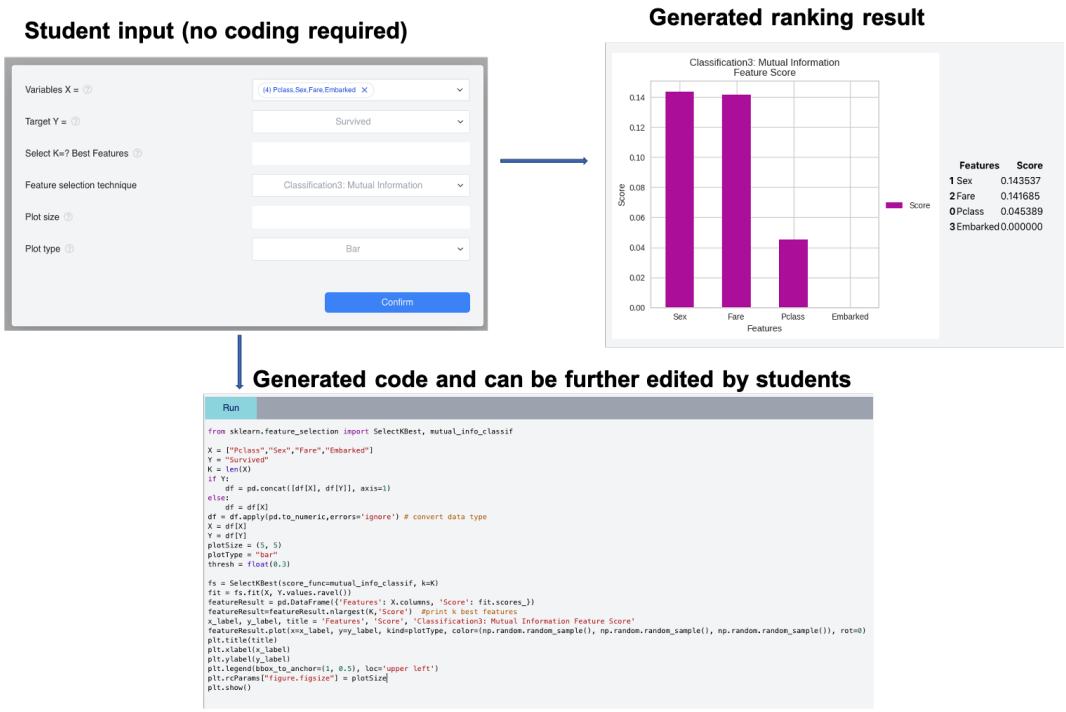

Figure 3: Using the DSLP to Perform Feature Selection Task for Titanic Dataset

Student backgrounds:

- **Student majors and academic levels:** Students who participated in the survey were from various non-computing majors, including Electrical Engineering (3 students), Economics (1 student), Mechanical Engineering (4 students), Imaging Science (1 student), Biochemistry (1 student), Motion Picture Science (1 student), Industrial Engineering (1 student), History/Philosophy (2 students), Biomedical Sciences/History (1 student), Management information System (5 students), Public policy (1 student), Political Science and History (1 student), Film Production Live Action (1 student), Accounting (1 student), School of Individualized Study (1 student), and not specified (1 student). 4 students reported as being in their second year of study, 6 students as in third year, 10 students as in fourth year, 5 as in fifth year, and 1 as not specified. When asked if the course is a required one, 3 students responded yes, 22 students responded no, and 1 student did not respond.
- **Student demographics:** Students were asked to indicate their gender, ethnicity (all that apply), and whether or not they are hard-of-hearing. 21 out of 26 students were Male, 5 were female. 8 students were identified as Asian, 2 as African-American/Black, 1 as American Indian or Alaskan Native, 21 as white, 1 as Native Hawaiian or other Pacific Islander, and 1 as other. One student was identified as Hard-of-hearing students, while 25 students were identified as hearing.
- **Student computing background:** Students were asked to identify all of their past experiences with programming. As survey results showed, 1 student had no prior experiencing with programming, 9 students had informal experiences before college, and 10 students had informal experiences during college. 8 students had formal programming

classes before college and 16 students had formal programming classes offered to non-computing majors during college (such as AP Computer Science Principles (CSP) or a CSP course at the college level). 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.

Students were asked to specify their opinions of the following statements related to the Google Colab Assignments:

- **Colab Q1***: I felt the Google Colab Assignments were easy to follow.*
- **Colab Q2***: The Google Colab Assignments improved my understanding of Data Science.*
- **Colab Q3***: The Google Colab Assignments increased my interest in Data Science.*
- **Colab Q4***: I believe I could perform similar data science tasks to those in the Google Colab Assignments.*

Students were asked the specify their opinions of the following statements related to the DSLP Assignments:

- **DSLP Q1***: I felt the DSLP Assignments were easy to follow.*
- **DSLP Q2***: The DSLP Assignments improved my understanding of Data Science.*
- **DSLP Q3***: The DSLP Assignments increased my interest in Data Science.*
- **DSLP Q4***: I believe I could perform similar data science tasks to those in the DSLP Assignments.*

Table 1: Student Survey Result

Questions	**Agree or Strongly Agree**	**Neutral**	**Disagree or Strongly Disagree**
Colab Q1	60%	30%	10%
Colab Q2	75%	20%	5%
Colab Q3	55%	20%	15%
Colab Q4	75%	20%	5%
DSLP Q1	85%	5%	10%
DSLP Q2	80%	10%	10%
DSLP Q3	40%	50%	10%
DSLP Q4	70%	25%	5%

A summary of results is shown in Table 1. As can be seen results, both Google Colab and DSLP Assignments received positive responses from the majority of students that participated in the survey. The DSLP assignments performed significantly better in terms of ease of use and improving student learning. Additionally, even though using the DSLP for the course project was optional, 35% of students voluntarily used the DSLP for hints or for help with code on their final project.

To better understand student perceptions, we also turned to open-ended comments from students for insight into the above results. Students reported that they found some of the minor bugs in the DSLP to be distracting and sometimes, frustrating. Other students wanted more instruction in coding in Python. Some students expressed concerns about not being able to access the DSLP after taking the course. Although the concern was a misunderstanding, it somehow explained that only 40% of students found that the DSLP Assignments increased their interest in Data Science. Regardless of some complaints about the DSLP bugs, which have been fixed in the current version, students explicitly pointed out the positive impact of the assignments and the DSLP on their learning, *"I feel I got most of my learning done through these [DSLP Assignments], and the DSLP platform was extremely useful."* [Student 2], and, *"Honestly I enjoyed these activities a lot. It really helps visualize the data and its nice that it writes the code out for you so you can learn from that"* [Student 23] and another, *"Really nice intro course to data science, made taking the Business Intelligence class alongside it more manageable."* [Student 9]. This indicates that the quality of the support for hands-on exercises impacts student learning and interest in Data Science.

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 thank Dr. Kimberly Fluet for her contribution in designing the survey questions and collecting/analyzing the survey data. The authors also thank the anonymous reviewers for their feedback.

References

[1] Austin Cory Bart, Dennis G. Kafura, Clifford A. Shaffer, and Eli Tilevich. 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*, pages 1029–1034, 2018.

[2] Lillian N. Cassel, Michael Posner, Darina Dicheva, Don Goelman, Heikki Topi, and Christo Dichev. 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*, page 722, 2017.

[3] Jeffrey S. Saltz, Neil I. Dewar, and Robert Heckman. 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*, pages 952–957, 2018.

[4] Chase Geigle, Ismini Lourentzou, Hari Sundaram, and Chengxiang Zhai. Clads: a cloud-based virtual lab for the delivery of scalable hands-on assignments for practical data science education. In *ITiCSE 2018: Proceedings of the 23rd Annual ACM Conference on Innovation and Technology in Computer Science Education*, page 176–181, July 2018.

[5] Aaron Green and ChengXiang Zhai. Livedatalab: A cloud-based platform to facilitate hands-on data science education at scale. In *Proceedings of the Sixth (2019) ACM Conference on Learning @ Scale*, page Pages 1–2, June 2019.

[6] Pang-Ning Tan, Michael Steinbach, and Vipin Kumar. *Introduction to data mining*. Pearson Education India, 2016.

[7] Kaggle. Carseat sales dataset. In *https://www.kaggle.com/competitions/carseat-sales*.

[8] Xumin Liu and Erik Golen. Developing a hands-on data science curriculum for non-computing majors. In *NSF IUSE program*, 2020-2023.