DS 1001 Active Learning Modules

R.C. Alvarado

8/23/23

Table of contents

Part I

Preface

This is a Quarto book.

To learn more about Quarto books visit https://quarto.org/docs/books.

1 Rubric Active Learning Lab Sessions

Requirements

- Create three assessments.
- For AY 2023-24.
- Can embedded into undergraduate data science courses.
- Should cover a two-week period.
- Each contains elements described below.
- Assessments are transparent.
- Assessments are authentic.

President and Provost's Fund for Institutionally Related Research: Data Science Active Learning Lab.

1.1 PROJECT DESCRIPTION

Active learning approaches have not been widely applied and tested in data science courses despite significant research supporting their benefits and its successful adoption in computer science. One study notes a 33% increase in pass rates for computer science courses employing active learning as compared to traditional lecture classes [3]. The Academy for Teaching Excellence has thousands of resources to aid in the teaching of computer science content with hundreds of assignments, but essentially no empirically tested resources for data science [4].

A key component of the active learning approach is the design, development, and testing of authentic and transparent assessments. Authentic assignments asks students to either inhabit a role seen in the discipline or do the work of the discipline. Transparent assignments explicity state their purpose, tasks, and criteria. Research has shown that transparent assignments improve a wide range of student success metrics, including a sense of belonging which can be a significant factor in the retention of underrepresented student populations, particularly in STEM.

A major obstacle to creating authentic and transparent active learning assessments is the time and effort required to develop, implement, and test them. **The Data Science Active Learning Lab will provide the resources and structure necessary for collaborative course development.** The Lab will enable faculty and graduate students to work together to develop and empirically test active learning activities in their data science courses. This

will result in the identification and validation of best practices along with practical tools that can be shared broadly and openly across the field of Data Science.

The Lab will reduce the time burden on faculty to create effective assessments. More important, it will establish a trusted research framework and tools that have been empirically tested for effectiveness. Such a resource is currently unavailable in data science pedagogy. We also have a unique opportunity, given the newness of the field, to develop cultural norms around teaching that are known to benefit underrepresented groups and first-generation students. This funding will go a long way to advance that effort by establishing a pedological research lab specifically focused on student success in a historically fast-growing field.

1.2 PROJECT GOALS

In pursuit of the mission to provide resources to enable faculty to create effective active learning assessments for data science, the project has the following goals.

- Be a preeminent **resource** for empirically tested best practices in data science education.
- Establish a **testing and validation framework** for data science oriented active learning tools.
- Create and socialize **teaching norms** in data science that are proven to benefit underrepresented groups and first-generation students.
- Develop and test active learning assessments (labs) specifically for courses in the proposed **data science undergraduate major**. (Projected courses include the Foundation of Data Science, Foundations of Machine Learning, Computation Probability, Data Science Ethics and Policy, and Data Science Systems.)
- **Disseminate open instructional content** in accordance with the School of Data Science open access policy and UVA's Open Scholarship agreement.

The goal of the summer is for each faculty member to create three implementable active learning sessions that can be embedded into classes in the next academic year. These active learning sessions should include the following materials covering a roughly two-week period.

I. Summary of the learning objectives for the session.

II. Pre-reading and/or video material.

III. Two Active learning lecture sessions and associated material

IV. Assessments – assignments, reflections, quizzes, group work etc.

V. Evaluation rubrics for learning outcomes.

VI. Review process to encourage retention.

The remainder of this document will describe each of these elements.

1.3 MATERIALS

1.3.1 Learning Goals

Learning goals should be the overarching goals of the class to which learning objectives could then align.

Examples of Learning Goals for a Machine Learning Class might be:

1. Be able to describe and execute the necessary steps to prepare data for machine learning models.
2. Demonstrate understanding of the mathematical and computation machine requirements for several machine learning approaches and when they should be used.
3. Demonstrate understanding of effective evaluation methods given different machine learning approaches.

1.3.2 I. Learning Objectives.

Learning Session: Data Preparation 1

Learning Objectives:

1.1a) Describe the reasons for and effectively demonstrate partitioning a dataset into train, test and tune.

2.1a) Effectively describe and demonstrate why it is necessary to one-hot encode factor and standardize continuous variables specific to various machine learning models.

3.1a) Execute and effectively describe dealing with missing data

II. **Pre-Reading and/or Video Material** – This should reference specific sections in books known to be of high quality related to the content. I have a tendency to lean more heavily on books that include "theory" explanations aside from simple how-to text given that much of the how-to will be provided in the class.

Videos can be very simple explaining the topics through presentation materials or code at a high level. I would work to limit the length of videos to 10 minutes or less and if several topics are covered break them into individual video segments. As a technic to consider, I often build a "stop and think" question in the video that I then discuss at the beginning of class to encourage students to watch and engage with the content. I also usually place these videos on YouTube as unlisted, but this does still allow you to track the watch time.

III. **Active Learning Sessions** – The definition of active learning is fairly broad, but in it's simplest form it suggests that students should not be passive in their learning. This doesn't eliminate classical lecturing methods but augments them to include directed moments when students are either engaging with each other or with the professor in more robust way than a traditional Q and A session. An example that Pete and I use a lot is "think, pair, share". This includes use asking a question, having the students think on the answer alone, then discussing with a partner and then finally sharing with the class as a team. A good book on methods on Teaching Methods provided by CTE is Teaching at its Best by Linda B. Nelson, I've got a copy and can purchased more if needed (also please feel free to share other references).

As a reference I found this quote from a 2021 paper focused on developing a framework for active learning helpful: "To clarify, we synthesized a working definition of active learning that operates within an elaborative framework, which we call the *construction-of-understanding ecosystem*. A cornerstone of this framework is that undergraduate learners should be active agents during instruction and that the social construction of meaning plays an important role for many learners, above and beyond their individual cognitive construction of knowledge."

The article is available here: https://journals.sagepub.com/doi/pdf/10.1177/1529100620973974

IV. **Assessments** can be the hardest and potentially the most time-consuming portion of course materials. The traditional path in most Data Science oriented classes is an assessment that focuses on implementing a method in code. I would encourage you to continue this practice but also consider adding written or verbal approaches to evaluating learning. This might include prompts in the coding assessments that require further written explanations or reflections on what was the most challenging/enjoyable portion of the assignment or what areas the students believe they need more practice. This information can then be used in a follow-up session that highlights the areas that a majority of students saw as needing more coverage. It is also important to create clear expectations on how the assessment will be evaluated and what the expectations are for the assessment. Below is an example assessment Pete developed for the DS 1001, though not a coding-based assessment the general structure is still relevant.

LOOK Rubric – Systems

DS 1001 – Spring 2023 - Professors Alonzi & Wright

Due Date Target: Noon, April 28; Due date final: Noon, May 10 (last day of reading days)

Submission format: File upload to canvas

Individual Assignment

General Description: This assignment is all about understanding the systems behind popular social media and content apps. You will select a popular app, like Instagram or Netflix, and

do a deep dive on the systems behind it that keep it running. Then you will produce a short report detailing the goal of the app, the software needs to make it work, and the hardware required to make it so. This will focus on the business side. Imagine you are the Chief Data and Technical officer for the company and producing a report for the Chief Executive and Operations officers.

Preparatory Assignments: READ #7-9 and Labs #7-9.

Why am I doing this? In the systems portion of this course, we have been studying hardware and software as well as understanding the scale involved. This assignment puts you in the position of a company that delivers a product at scale through an app. You will need to understand the goal of the company and then the necessary software and hardware to make that happen. This process of studying a company and thinking through their needs will reinforce the learning about hardware, software, and scale.

- LO: Identify the hardware and software components of a computer and describe their function
- LO: Describe the different scales of computer operation

What am I going to do? First you will select a company to study, choosing from the list of Instagram, Facebook, Twitter, or Netflix (if there is another you would like to do get clearance from a professor first). Once you have that chosen you will figure out what it takes to power their app. Put another way you will determine the goal, not the business goal of "make more money", but the technical goal. For example, Netflix streams video content. Then you will research the software and hardware needs of the company to achieve that goal. Finally once you have done that research you will produce a short report detailing the various components.

Tips for success:

- Pick an app that you use.
- Take this opportunity to learn more about something you use, be curious.
- Often apps are very different in different locations, for this assignment you can simplify and stick to the US market.
- Think about yourself and the goal of college. What software do you need to use? What hardware does that software require? Taking a few minutes to think that out can help focus you for the assignment.

How will I know I have succeeded? I will meet spec when I follow the criteria in this rubric.

Spec Category	Spec Details

Formatting	• Submit a single PDF.
	  – Give it a header stating the assignment.
	  – "Look Ahead assignment – Systems."
	  – Name, course, date
	• 3 page maximum, including tables and figures.
	• Executive Summary
	• Goal statement
	• Software requirements
	• Hardware requirements
	• References

Executive Summary	• Goal: A short declarative sentence or two describing the contents of the report and the major takeaways. • List the components of the report. • State the key figures and scale. • Highlight any major takeaways the reader should look out for.
Goal Statement	• • Goal: This is a short paragraph describing the technical goal the app is trying to achieve, for example Netflix streams video. • Include the scale involved, how much data is stored, how much computing power is needed, etc. • This is not a detailed description, just a coherent statement of the mission
Software Requirements	• Goal: Present a detailed description of the software used to achieve the goal. • This is a major section of the assignment and is about half of the length of the assignment. • Mention the major software components and their needs. • Include a table summarizing the software needs. • Include a visualization of the software needs

Spec Category	Spec Details
Hardware Requirements	• Goal: Present a detailed description of the hardware used to achieve the goal. • This is a major section of the assignment and is about half of the length of the assignment. • Mention the major hardware components of the system. • Include a table summarizing the hardware. • Include a visualization summarizing the hardware.
References	• All references should be listed at the end of the document • Use IEEE Documentation style (link)

Acknowledgements: Special thanks to Jess Taggart from UVA CTE for coaching us. This structure is from Streifer & Palmer (2020).

V. **Evaluation rubrics** for learning outcomes –

This step can likely be included in the development of assessments but having it as a standalone emphasizes the need for thoughtful design.

The goal here should be evaluation measures tailored to the assessments but also universal enough to be used in a standard lecture format.

Meaning that the rubrics will be included as part of the experimental design to assess the variances in learning that occurs in an active learning environment when compared to a lecture format.

In the above example the quality of answers to the final question, "How do the answers to the questions make you feel as it relates to the presence of data driven technologies in our everyday lives?", could be a focus on the evaluation rubric as it relates to the specific learning outcome around "understanding the growing influence of data on society".

VI. **Review process** to encourage retention – The idea here is to not compartmentalize the learning objectives but blend them together from week to week to help reinforce the topics throughout the semesters.

One method example is to have quizzes that include questions from all weeks in the class not just the current topic. I think for this use case, simple direct sessions that are short in nature, 10-15 minutes, that review topics from the session makes more sense.

Examples might include a guided back and forth on the key topics from the previous week or a team assignment that is short in nature but requires the students to pull previous information forward.

If the previous week's topic was Decision Trees, I've shown some code and an image of DT in class that had three errors and ask the students to find and describe the errors in 10 minutes, as an example, before moving into the new session for the week.

1.4 III. Additional Notes

- We want to publish these materials online, so when building please consider the goal is to make the materials publicly available.
- We are also hoping to "empirically" test these in a classroom in the Spring of 24, so also be *thinking* about the development of non-active learning materials and where best to measure results.
- You've got great ideas, this is just a framework, so feel free to move as you see fit.
- This process should be useful for future and current SDS faculty. So, keep an eye on the ideal that we are in some ways culture building/establishing best practices, which I hope gets noticed.
- Pete and I will create a Team site with folders for your content and as a placeholder for documentation on the project. You do not need to use these folders, just an option.

Part II

M02 Introducing Python

Topics

- Running Python code
- Python's basic data types
- Python's primary operators associated with each data type
- Python's built-in data structures

Readings

Required

Katz and Katz 2019, Section 1, Preparing the Workspace

Lutz, Learning Python, Part I: Getting Started, Chapter 2

Lutz, Learning Python, Part I: Getting Started, Chapter 3

Lutz, Learning Python, Part II: Types and Operations, Chapters 4–9

Optional

Katz and Katz 2019, Section 1, First Steps in Coding - Variables and Data Types

Built-in Types (Official)

Python Data Types (GFG)

Python Operators (W3S)

Immutable vs Mutable Data Types in Python (Medium)

2 Data and Code

2.1 Code should be simple

An important principle for writing effective and intelligible code is that code should be simple — to quote Einstein, as simple as possible but no simpler.

- A contributing factor to code simplicity is how it is related to the data it is designed to process.
- This relationship depends largely on how the data are structured.
- A program is always written with data in mind — what kind of data it is and how it is structured.

2.2 Simplicity of code follows from the structure of data

There is a view among programmers which, although not orthodoxy, is commonplace.

- It is the idea that the complexity of a program — its algorithms — is a function of the quality of the data structure it processes.
- If a data structure is not well designed, algorithms may be excessively complex and hard to understand.
- However if a data structure is well designed, the algorithms that process them are more robust and intelligible.

2.3 Supporting References

Consider these quotes cited in an essay on Data Structures. by Igor Budasov, reproduced here:

Here's a quote from Linus Torvalds in 2006:

> I'm a huge proponent of designing your code around the data, rather than the other way around, and I think it's one of the reasons git has been fairly successful . . . I will, in fact, claim that the difference between a bad programmer and a good one is whether he considers his [sic] code or his data structures more important. Bad programmers worry about the code. Good programmers worry about data structures and their relationships.

Which sounds a lot like Eric Raymond's "Rule of Representation" from 2003:

> Fold knowledge into data, so program logic can be stupid and robust.

Which was just his summary of ideas like this one from Rob Pike in 1989:

> Data dominates. If you've chosen the right data structures and organized things well, the algorithms will almost always be self-evident. Data structures, not algorithms, are central to programming.

Which cites Fred Brooks from 1975:

> **Representation is the Essence of Programming**
>
> Beyond craftsmanship lies invention, and it is here that lean, spare, fast programs are born. Almost always these are the result of strategic breakthrough rather than tactical cleverness. Sometimes the strategic breakthrough will be a new algorithm, such as the Cooley-Tukey Fast Fourier Transform or the substitution of an n log n sort for an n 2 set of comparisons.
>
> **Much more often, strategic breakthrough will come from redoing the representation of the data or tables.** This is where the heart of your program lies. Show me your flowcharts and conceal your tables, and I shall be continued to be mystified. Show me your tables, and I won't usually need your flowcharts; they'll be obvious.

> **Note**
>
> See video on Canvas.