Engineering Educators Bringing the World Together
2025 ASEE Annual Conference & Exposition
Palais des congrès de Montréal, Montréal, QC • June 22–25, 2025 ASEE
Paper ID #45759

Developing an AI/ML activity for a BME physiology course

Dr. Laura Christian, Georgia Institute of Technology

Laura Christian is a Lecturer in the Biomedical Engineering Department at Georgia Tech. She is excited to combine her experiences in biology teaching with methods used in engineering instruction and to use education research techniques determine methods that work well for these students.

Ophelia Anais Winslett, Georgia Institute of Technology
Alpa Gautam, Georgia Institute of Technology
Dr. Todd M. Fernandez, Georgia Institute of Technology

Todd is the Director of Learning Innovation and a Senior Lecturer in the Wallace H. Coulter Department of Biomedical Engineering at Georgia Institute of Technology. His research interests are engineering faculty and students beliefs about knowledge and education with a special focus on how those beliefs interact with engineering education as a culture.

Developing an AI/ML activity for a BME physiology course

Introduction

The current employment landscape is likely to undergo significant changes as the prevalence of data-driven work increases. The types of engineering jobs available and the skills required for these jobs will be affected [1]. Rather than the traditional computational skills (e.g. writing code, data structures, statistics), critical skills for graduates of engineering degree programs will shift to a higher level - including the ability to conceptualize, identify, organize, and make sense of data using statistical and machine learning (ML) tools. More importantly, how engineers use the results from data to solve engineering problems is constantly changing [1]–[3]. For example, the US Food and Drug Administration (FDA) communication and regulation on using artificial intelligence (AI) in medical technology development continues to be updated [4], [5]. In order to prepare our undergraduate biomedical engineering (BME) students for these changes in the use of data, broadly, our program refers to this field shift as one of "data skills". We see data skills as reflecting the challenges for biomedical engineers, or any engineers, that begin before and end well after the use of any particular AI or ML algorithm or approach. The critical role of data in such work is well established and represents a necessary perspective in preparing BMEs to lead technically competent and morally defensible AI/ML work [6], [7].

Our data skills curriculum is actively evolving. We aim to increase student engagement in data skills-related learning longitudinally. Many programs have created specific course(s), majors, or minors in data science[2], [8]. While adding a major or minor is effective for students who choose this type of coursework, we see a broader need to create a variety of opportunities that are tailored to the needs of different interests and career paths of different students. All students should have some exposure, alongside opportunities for greater depth that flow from this introductory exposure.

Introducing data skills in undergraduate BME education poses particular challenges. Many BME programs are structured so that students receive a broad range of coursework. This breadth of work may come at the cost of depth into topics critical for their future careers in the field [9]. Our department determined that developing a separate introductory course in data skills would not be feasible. This decision was based on limitations on degree credits, space, and other practical challenges. In addition, studies have shown that BME students have a limited view of the types of careers that they may attain after graduation[10], [11] - in part attributed to the presentation of different topics as broad and disparate. We are motivated to integrate data skills content into existing required courses in ways that both further learning in the course itself and address our practical challenges. The result is a program of curricular change where data skills elements are incorporated into each required undergraduate BME course.

In this paper, we describe the development, implementation, and evaluation of one such activity - using ML diagnostics of Atrial Fibrillation (AF) from electrocardiogram (ECG) data to build cardiac systems knowledge in a junior-level Systems Physiology course. We developed and tested two versions of this activity: a longer version requiring access to hosted graphics processing unit (GPU) computing services, and a shorter version that can run on laptops. We used a pre-post survey to measure student perceptions of their skills in and career inclinations towards data skills abilities, and their perceptions of the applicability of data skills in jobs. The activity could be easily adopted in similar classes and includes suggestions on how to scale it to range from 35 minutes to 2 hours of course time. We also report on our general approach for developing this activity so other institutions may consider using this model for teaching data skills to their students. All files for the activity are provided in a GitHub repository.

Activity and Implementation

The activity we developed was designed to serve both data skills and course-specific learning objectives. In this section, we describe the curricular and course context as well as the activity. As noted in the intro, this activity is part of a larger data skills initiative in our undergraduate curriculum. Our effort to change BME undergraduate courses exists within a rapidly increasing ecosystem of AI/ML learning opportunities for students at our university. These include an AI/ML-centric minor available to all engineering students [12]. The minor is interdisciplinary and is available to students in both our College of Engineering as well as our College of Liberal Arts. Learning outcomes include understanding and applying AI in the student's primary discipline and evaluating the ethics of AI and ML uses. There are three required and five total courses – a required AI ethics and Policy course, a statistics course, and an AI/ML focused applications course. Multiple departments have an AI applications course, and students are able to choose from those courses based on the types of AI/ML applications they are interested in – with an encouragement to learn cross-disciplinarily. Other opportunities for students to learn about AI include elective courses, seminars, research, and free computing services through a specialized Artificial Intelligence Makerspace built in partnership with NVIDIA.

Data skills in our BME curriculum

Emerging from ongoing concerns about preparing BME students for modern engineering work, the BME faculty and other stakeholders engaged in a process of discussion, ideation, and definition of what role AI/ML should play in our curriculum. The result was a set of data skills:

1. Implement algorithms for data analysis as a working code in a programming language such as Python, MATLAB, R, or C/C++
    a. Write scripts combining off-the-shelf data processing tools
2. Describe probabilistic models and demonstrate the use of standard tools of statistical analysis and machine learning
3. Use regular operations in spreadsheet programs (e.g. Excel)
4. Use data to solve engineering problems in biology and medicine
    a. Apply statistical analysis and machine learning tools to different datasets and understand their limitations
    b. Justify design decisions, inputs, and constraints
    c. Recognize biases or underlying assumptions within datasets and that their use may pose risks to certain populations
    d. Organize and present data visually to an audience

The data skills were developed with input from, reviewed by, and approved by the department's faculty and external advisory board. The list of skills also maps to those identified by industry leaders in other research [13]. During development, the consensus was that the rapid emergence of AI/ML represented a motivation and framing of the need for our students to develop data skills. However, a narrow focus would likely distract from what students actually need to be effective engineers. Therefore, we defined a set of general data skills, including connections to AI/ML and to other ways data-driven engineering occurs. We believe it is important to situate AI/ML as a continuation of the use of data, rather than as an entirely separate skillset.

In parallel with approving the skills themselves, the faculty approved a requirement that some aspect(s) of these data skills must be integrated into each required undergraduate course. While the scope of that integration was left open-ended, it must be transparent and identifiable to both instructors and students. The integrations were intended to generally adopt the following pedagogical principles:

- Meaningfully link the core content of a course to an authentic application of the use of data, ML, statistics, and/or use of engineering computing tools in biomedical engineering work
- Enhance, rather than replace or compress, learning about existing content in the course
- Not require any pre-requisite knowledge beyond that already required for the course
- Focus on application and exposure over first principles or mastery

Two foundational educational frameworks guide specific interventions enacting these principles. The first was the ICAP framework [14]. The nature of these interventions as application-based and focused on linking led to an emphasis on collaborative and student-directed exploration of AI/ML applications wherever possible. The second was a focus on a higher level and forward-looking reflection as described by Kember and colleagues[15]. Such activities are already common in our program and are largely guided by established frameworks including inquiry and project-based learning, and conceptual change [16].

Course description

The course in which we implemented this activity is an upper-level introductory physiology course that consists of two 1-hour lectures and one 2-hour Problem Solving Studio (PSS) [17] per week, with some content delivered using required course videos. The course is designed to help learners connect disarticulated physiological concepts to solve system-wide problems. The general content covered is typical of introductory physiology courses[18], [19]. However, the course is somewhat different in the extensive focus on integrating knowledge across individual organs and systems to troubleshoot signs and symptoms affecting the entire system. The PSS sessions emphasize evaluative and applicational questions (i.e., higher levels of Bloom's taxonomy[20]). The sessions, which operate at Chi's active level [14], are typically built around medical case studies and typically ask students to make treatment or other decisions in the case studies. We designed a full version of the activity (see next section) that could take the place of a PSS session focused on connections between cardiac anatomy, the cardiac cycle, and electrical recordings via ECG. We implemented the compressed version of our activity (also, see next section) in a PSS session focused on connections between cardiac dysfunction and connections to the respiratory system and respiratory symptoms.

Activity

We designed the data skills activity to integrate manual and ML interpretation of ECG traces to diagnose AF. Neural network models have been used to detect AF in research studies (e.g., [21]–[23]), but we have not found published work that describes the use of this system as an instructional tool. The activity takes the entirety of one PSS session (max 110 minutes) in the systems physiology course [17]. This assumes students come in with the appropriate software installed and configured. Students work in groups of 4 on a worksheet that guides them through the activity. During the fall of 2024, we developed 2 versions of the data skills activity which we implemented with two different groups of students for evaluation (Table 1.) Both versions were based on the same high-level changes to an existing course session on cardiac and/or pulmonary physiology, depending on the material that is relevant to the curriculum at the time of the activity. This section describes the activity development at a high level and then specific activities in each are described in two subsections below. Note that all activity materials are provided in the GitHub repository [24].

Table 1. Description of the design and implementation of 2 new versions of the activity

	Original Activity *(no data skills)*	**Activity Version 1 *(Full)***	**Activity Version 2 (Compressed)**
Time allotted in class	110 minutes	110 minutes	110 minutes
~ Time on data skills	0 minutes	90 minutes	35 minutes
Required resources	Worksheet, pen	GPU-based computation [25] and Google Spreadsheet on student laptops, worksheet, pen, code and data files	MATLAB (with Deep Learning or Statistics and Machine Learning, and Signal Processing Toolkits) on and Google Spreadsheet accessed through student laptops, worksheet, pen, code and data files
Pilot size	-	8 students divided into 3 groups	142 students across 3 course sections. Each section contained up to 48 students in groups of 4
Survey data	No	No	Yes
Observational data	No	Yes	Yes

As noted, the activities were guided by two educational frameworks by Kember and by Chi [14], [15]. We sought to intentionally evaluate and elevate the levels of engagement and reflection. As much as possible, we modified existing activities to move from the active or constructive level to the interactive level and create moments that are likely to elicit critical reflection. Examples of these changes are discussed throughout and also highlight our four pedagogical principles described in the previous section.

Starting from the existing cardiac/respiratory dysfunction activity mentioned above, both versions of the data skills activity integrate training and inference of ML models. In keeping with our principles, no physiology content was removed, although the way students engage with it did change. For example, in the original activity (no data skills), students label the P, Q, R, S, and T parts of a wave on a model ECG of a single heartbeat. In the revised data skills activity, students were asked to use the parts of the wave to mark features indicative of a cardiac diagnosis on a single ECG rhythm strip showing approximately 9 beats, and including noise. Then, focusing on applications, they are asked to extend that knowledge to classify multiple ECGs presented to them as showing symptoms of AF or not. Finally, enhancing and linking this content to other courses, they are asked to consider mathematical features of such waves that could be used in computational analysis. A comparison of the student tasks in the two versions of the data skills activity is shown in Table 2.

Table 2. Comparison of the Full and Compressed Versions of the Data Skills Activity

Student Task	**Compressed version**	**Full version**
Train GPU-based model to distinguish AF vs. not AF		x
Enter manual diagnoses and compare across student groups	x	x
Classify test data and compare model performance information across student groups	x (w/pre-trained model)	X (w/model they train)
Answer physiology-based questions	x	x
Answer questions centered around the clinical use (and consequences of the use) of this AI tool	x	x
Train a model of their choice based from several options highlighting common ways to improve ML performance (e.g., more parameters)		x

Development of the code for the activities

Both the full and compressed versions of the activity are a simplified version of and use the same data as the 2017 edition of the George B. Moody PhysioNet Computing in Cardiology Challenge[26]. The 2017 challenge asked teams of researchers to create classification algorithms to aid clinicians in interpreting ECG signals among four possible categories. The simplified version is based on example code created by MathWorks in the MATLAB software language[27].

While the example code the activity is based on is generally aligned with the activity, it was written to demonstrate an implementation of a set of software functions - not to scaffold an educational activity. Therefore, we (the authors) revised the code before running the activity in four ways. First, it was refactored and organized to align with the activity prompts and questions. Second, comments explaining the code itself were replaced with comments explaining the practical aspects, links to physiology, and modeling decisions that the code reflected. This included obfuscating or glossing over aspects of the code that were extraneous to the activity's main goals (e.g., loading data from physionet). Third, in the full version of the activity, we added an additional model training run in which groups made choices on how to improve their model. Fourth, the example MATLAB code only worked in the most recent releases of MATLAB[1]. That did not align with the varying releases installed by our students and on our institutional compute cluster. To accommodate this, we wrote multiple versions of some sections of code using appropriate functions from all major releases over the last 4 years. The code for the activity is included in supplemental materials. As noted earlier, we prepared two versions. The full version includes the actual training activities as well as the inference activities but requires significant computational resources, described later in the paper. In contrast, the compressed gives students pre-trained models that can be reasonably run on a typical student laptop.

Given common curricular change concerns in engineering education, we want to dissuade others from thinking that the use of pre-existing code is a negative for this activity, or that there is a tension between covering any specific ML model in depth and covering cardiac physiology. Our focus, as laid out in learning objectives and design principles for these activities, was engagement specifically on (1) the evaluation of the output and efficacy of such models, (2) the relationship of ML to the existing course content, and (3) the relationship between ML and clinical decision making. We believe having students write their own code during this activity is an inefficient use of limited time for such an activity in a physiology course. That is, while coding skills are important (see list of data skills above), they are a distraction here from our core goals of using ML tools and big data to further learn about physiology. Further, that time would almost certainly be spent at the lower levels of the ICAP framework where students learn less and would take away time for discussion and reflection. The choice to use an established open data challenge and existing demonstration code was motivated by our interest in showing opportunities for continued growth to students through these activities. Doing so uncovers potential future learning and growth opportunities for students that they otherwise may not know exist. While not directly related to the intended learning outcomes, these types of choices are fundamental to our design principles for these activities. Students who find interest through these introductory exposures have multiple opportunities for future growth as described in the curricular context section. In addition, activities we are developing for other required courses in our major will require students to write code.

Compressed version implementation details

The compressed version illustrates a minimalist implementation of the activity and how such an implementation can integrate into a small portion of a single course session. The compressed version of the activity was implemented in the first 30 minutes of a 110-minute course session, with a 5-minute wrap-up discussion at the end of the session. This version was implemented in the Fall of 2024 in 3 course sections, each with up to 48 students working in teams of 4. In this version, students rely on a pre-trained ML model; run all code locally (i.e., on their laptop); and focus on model inference, model interpretation, and model application. The code and worksheet are divided into parallel ‘sections’ that guide students to run code and answer questions about the code, cardiac system, or results of the code concurrently. Students also enter results produced by their code into a Google Spreadsheet that allows for comparison, summary, and discussion of results across teams. All relevant Supplemental Materials and materials in the GitHub repository are labeled COMPRESSED.

The activity begins by introducing students to a medical case - a 63-year-old woman with a complex medical history who is experiencing declining indicators of cardiac and respiratory function. During the introductory portion of the activity, teams run a section of code that loads data from a web archive, splits that data into test and training data, and plots three ECGs for teams to visually diagnose. Two of the signals (one normal, and one AF) are common among all of the teams to encourage collective decision-making and checking of their answers. The third is selected randomly and could exhibit normal physiology, AF, or another irregularity.

Towards our interactive goal, each group enters their diagnoses of the multiple ECGs into a shared spreadsheet (Figure 1). The groups’ diagnoses are automatically checked against the data set labels for accuracy, sensitivity, and specificity and the results are shown to all the teams (Figure 2). This created two forms of interaction. First, teams could see each other's decisions and check and evaluate their own knowledge when there were

---

[1] Specifically, the example code existed to demonstrate improved functionally of the MATLAB 2024b release of *Deep Learning* and *GPU and Parallel Computing* Toolboxes, which introduced new functions to ease the inference portion of modeling.

differences. Previously such classification was done independently by each group and submitted to the instructor for grading. Second, the shared spreadsheet enabled classwide discussion and debate as the instructor asked for multiple perspectives on what guided their diagnosis.

1	Team Information		Question 1 C						
2	Section	Team #	ECG 4	Diagnosis	ECG 1	Diagnosis	ECG X	Diagnosis (Once you answer don't change it)	Correct? (be honest here)
3	Example	Test	4	Not Atrial Fibrillation	1	Atrial Fibrillation	719	Atrial Fibrillation	INCORRECT
28	Section 1	1	4	Atrial Fibrillation	1	Not Atrial Fibrillation	2974	Not Atrial Fibrillation	CORRECT
29	Section 1	2	4	Atrial Fibrillation	1	Not Atrial Fibrillation	4934	Not Atrial Fibrillation	CORRECT
30	Section 1	3	4	Not Atrial Fibrillation	1	Atrial Fibrillation	141	Not Atrial Fibrillation	CORRECT
31	Section 1	4	4	Not Atrial Fibrillation	1	Atrial Fibrillation	3325	Atrial Fibrillation	INCORRECT
32	Section 1	5	4	Atrial Fibrillation	1	Not Atrial Fibrillation	4972	Not Atrial Fibrillation	CORRECT
33	Section 1	6	4	Atrial Fibrillation	1	Not Atrial Fibrillation	1989	Not Atrial Fibrillation	CORRECT
34	Section 1	7	4	Atrial Fibrillation	1	Not Atrial Fibrillation	5344	Not Atrial Fibrillation	INCORRECT
35	Section 1	8	4	Not Atrial Fibrillation	1	Atrial Fibrillation	3678	Atrial Fibrillation	INCORRECT
36	Section 1	9	4	Atrial Fibrillation	1	Not Atrial Fibrillation	2029	Atrial Fibrillation	INCORRECT
37	Section 1	10	4	Not Atrial Fibrillation	1	Not Atrial Fibrillation	4232	Not Atrial Fibrillation	CORRECT
38	Section 1	11	4	Atrial Fibrillation	1	Not Atrial Fibrillation	4855	Atrial Fibrillation	INCORRECT
39	Section 1	12	4	Atrial Fibrillation	1	Not Atrial Fibrillation	61	Not Atrial Fibrillation	CORRECT

Figure 1. A view of the Google Spreadsheet used by students running the compressed version of the activity. Students select their diagnosis for three, two of which are common among all teams (ECG 4 and ECG 1), and one randomly selected for each team by the code (ECG X). ECG 4 was labeled in the data set as AF by experts, while ECG 1 was labeled normal.

The student worksheet asks teams to discuss their diagnosis accuracy, the visual cues they used in their diagnosis, and to relate AF to the medical case. When most teams have completed this step, the diagnosis indicators of the 'common' ECGs are discussed as a course, alongside class accuracy, and comparisons to the literature on different types of clinicians (Figure 2).

The discussion led by the facilitator was targeted to help students articulate a change in their understanding of how ECG signals relate to cardiac behavior (i.e., Kember's topmost *critical reflection* stage). This discussion is guided progressively - reflect on individual team's diagnoses and knowledge of cardiac physiology, reflect on course level performance to assess mastery, and then reflect on reasons their performance might be higher than certain groups of clinicians but lower than Cardiologists. We found that students were surprised that their accuracy in diagnosing AF[2] was more accurate than most groups of clinicians. As statistics is not a prerequisite for this class, the discussion here also offered a brief conceptual connection to sensitivity and specificity and the implications (e.g., for clinical decision-making) of both false positives and false negatives.

Confusion Chart (class diagnoses summary)		Correct Diagnosis	
		AFib	Not AFib
Class Diagonsis	AFib	31	16
	Not AFib	9	54
Summary of data from BMED3100 Student Diagnoses			
Accuracy	77%	Overall percentage of correct diagnoses	
Sensitivity	78%	Probability of a correct diagnosis of AFib	
Specificity	77%	Probability of a correct diagnosis of NOT AFib	
Accuracy Comparison	Mean Accuracy	Cite: doi:10.1001/jamainternmed.2020.3989	
Our class	77%	This number is built as class goes on.	
All literature	54%	this includes ALL levels of training and experience	
Medical Students	42%	'substantial inconsistency' - range was (6%-79%)	
Residents	56%	Training increased accuracy to 67%	
Practicing physicians	69%	Training increased accuracy to 80%	
Cariologists and fellows	75%	Training increased accuracy to 88%	

Figure 2. Summary of student accuracy from diagnosing randomized ECGs and comparison to accuracy amongst groups of clinicians.

After this introduction, students switch to their laptops to engage in the ML portion of the activity. Having looked at individual ECGs manually, students used pre-trained ML models to classify a large open data set of ECGs. Both the compressed and full activity version models use neural networks to classify ECG signals as

[2] Noting, diagnosis was done by teams and affected by interactive learning in the spreadsheet (see discussion)

either AF or not AF. A classification of "not AF" could mean anything from a completely normal ECG trace to one with ventricular fibrillation, or even an ECG with incorrect electrical connections.

Next, students are instructed to run section 2 of the code, which runs an inference process that classifies all of the raw ECG signals. This is run on a model pre-trained by the instructors on a GPU-based cluster resource, and depending on the student's laptop takes between two and five minutes to complete. To ensure compatibility across different MATLAB versions students may have installed, the code includes the same functional steps implemented using different functions.

While the inference code is running[3], students are given the intentionally vague question: *Translate your visual diagnosis of ECG signals into mathematics and discuss challenges in making a mathematical function from an ECG*. This question was designed to have no concrete answer but rather generate discussion in teams that initiate connections between cardiac physiology, signals processing, and mathematics. We expected students to invoke calculus concepts like derivatives, peaks finding, energy, or frequency. Some of that occurred, but we were surprised by the level to which students struggled with this question or identified such a translation as impossible because the data was 'too complicated'. Faculty interaction during the activity helped enormously, often with just minor prompting, and we found students both surprised and excited to see links between their math and physiology course content. We plan to provide better priming and scaffolding here in future versions.

When the code is complete, it generates a confusion chart, inference run time, and test accuracy (Figure 3). Students then enter that information into the Google Spreadsheet (Figure 4) and a cross-class summary is again generated (Figure 5). Finally, students are asked to put themselves in the shoes of clinicians and asked whether they would rely on this diagnostic tool to guide treatment, and what physiological consequences may result from a treatment choice based on an incorrect diagnosis. We noted that the model trained on raw ECGs performs at a level that is above the level of most clinicians but below the accuracy of our class. The worksheet then continues with related physiology content about AF treatment and links between the cardiac and pulmonary systems.

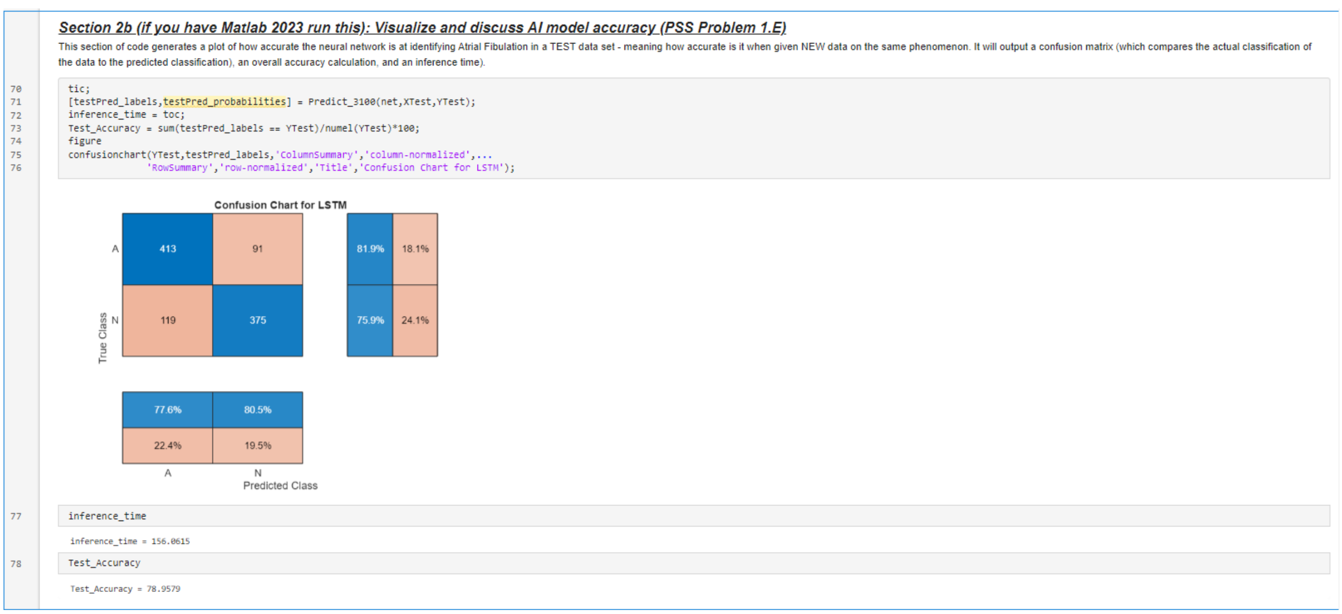

Figure 3. The confusion chart, inference time, and test accuracy reported to students after the raw ECG data is classified by the model provided to them.

[3] As a parallel note, we found it important in several of the activities we are developing to use code run time constructively rather than as having students simply wait. While code (e.g., a model training run) is executing, we use the time to ask students reflective questions about what the code is doing, potential biases, expected outputs, etc. Beyond a better use of class time, this scaffolding both serves to increase engagement *and* allows us to prime students on how to make sense of the output we know they will get.

Team Information		Problem 1 E							
Section	Team #	Inference Time (sec)	ECGs	Sec/ECG	Accuracy(%)	True Positive (A/A)	False Positive (A/N)	False Negative (N/A)	True Negative (N/N)
Example	Test	40	998	0.0401	66%	196	111	308	383
Section 1	1	32.8216	998	0.0329	81.26%		104		
Section 1	2	66.63	998	0.0668	79.36%	399	101	105	393
Section 1	3	41.7158	998	0.0418	81.16%	406	98	90	404
Section 1	4		998	0.0000					
Section 1	5	382.9824	998	0.3837	85.17%	441	85	63	409
Section 1	6	18.938	998	0.0190	82.16%	427	77	101	393
Section 1	7		998	0.0000					
Section 1	8	26.2805	998	0.0263	79.06%	392	97	112	397
Section 1	9	61.4733	998	0.0616	83.17%	427	77	91	403
Section 1	10	51.0033	998	0.0511	80.76%	413	101	91	393
Section 1	11	26.0548	998	0.0261	77.76%	392	110	112	384
Section 1	12	18.96	998	0.0190	79.66%	392	112	91	403

Figure 4. View of the spreadsheet where student teams enter information from their model's outputs (from Figure 3).

Neural Network 1 - Raw Data			
Parameter	**Average**	**Min**	**Max**
Inference Time	0.084 sec/ecg	0 sec/ecg	0.49 sec/ecg
Accuracy	82%	66%	88%
Sensitivity	81%	77%	89%
Specificity	81%	0%	92%

Figure 5. Summary of the performance of a neural network to diagnose AF from raw ECG data. This summary was generated after three class sections to show teams the similarity and variance that jointly exist in machine learning training results.

The choices and structure of this implementation are intended to provide a minimum-impact introduction to the use of ML tools in a physiological space. Specifically, this version was developed to show instructors in the class and across the curriculum the type of minimal implementation that was possible for the introduction of data skills in any course. In doing so, this version of the activity demonstrated to the course instructors how 'extra' or 'different' content, such as statistics related to ECG classification, could further student understanding of core course content (e.g., the cardiac system) without requiring the removal or adjustment of any content. The activity here retained all of the questions of the original activity (i.e., the pre-intervention version). The results discussed below suggest that students' understanding of the core course content actually improved compared to the pre-intervention version. If others were interested, it would be possible to pre-train all models described in the full implementation and run similar inference code locally to expand this activity in order to include the other ML models. We note that we have modified the activity to include some of the ML models described in the full implementation for future offerings of this course.

Full version implementation

This section is meant to expand upon the general structure and compressed version above. Rather than reiterating similar details, it focuses on explaining where the full version of the activity deviates from the compressed version. The full version generally expands on the compressed implementation in two ways: (1) Students train models and run classification on a GPU-based compute cluster as opposed to running inference only and doing so locally. (2) Students train and test three different ML models on ECG. In our efforts at data-skills-based curricular change, this version is meant to show the possibilities of changing a full 110-minute class session. By design, this implementation of the activity is meant to exchange deep work on details of cardiac physiology for a deeper introduction into how cardiac physiology can be linked to other areas of engineering content - e.g., signals processing and ML. It drops the context of a medical case and replaces it with framing in the development of an AF detection product. This version of the activity was piloted with a group of 8 TAs (6 undergraduate and 2 graduate) for feedback and evaluation. In the Supplemental Materials and GitHub repository, all related files are labeled FULL.

The full implementation begins with two extra steps compared to the compressed version, then follows the same structure until the end of the compressed activity, and then continues. First, the students log in and load the data and code on a remote GPU-based compute cluster (see Computational Resources and Collaboration section below for details). All code for the activity is then run on that cluster. Second, students open a highly simplified version (i.e., no comments, no sections, one model) of the code on a single student laptop and run it. This simplified version provides a point of comparison of compute resources during later discussion. Our testing showed that by running the code for the remainder of class, training reached only about 10% for one of the three models that students train on the cluster.

At the beginning of the activity, students perform the same random draw of ECGs, but are shown four random ECGs to diagnose as opposed to one random and two standardized ECGS. After entering their diagnoses into a similar spreadsheet, they are instructed to run the next section of the code, which begins training the first of three ML models. Training the model takes about 11 minutes on the NVIDIA H100 GPUs provisioned to students on the cluster (described below), so we chose to use that time to have students complete other work in the activity. The class has a similar discussion of how they visually classified ECGs. Since the data set has a ratio of approximately 7:1 normal to AF ECGS, the activity then walks students through a theoretical model where all signals are classified as normal - which would produce an accuracy of about 87% (with 0% sensitivity and 100% specificity). This is used to discuss the need for balancing or replication in the data set and the potential for bias from data in ML models.

Once training and inference for the first model are complete, students enter results into their spreadsheet and answer questions similar to the compressed version. In the full version, students have the accuracy results for *both* training and test accuracy, with the model producing higher results for training (~90% accuracy) than test data (~55% accuracy, 52% sensitivity, 51% specificity). This enables discussion and reflection on an aspect of AI (i.e., overfitting) that is not possible in the compressed version. Combined with the manual calculation of the theoretical ‘all normal’ model (87%), the differences in accuracy enabled deeper discussion about ML models of ECG classification than the compressed version. In the reflection prompt and class discussion, students are asked to comment on the reasons and problems of a large difference in training and test accuracy as well as how to make sense of a model with lower accuracy than blind classifying all data as normal.

After analyzing the first model, students then repeat this process for a second ML model. Tracking with the base code from MATLAB, this model uses spectral analysis techniques to perform feature extraction on the ECG signals. It does so by calculating the overall power spectrum, instantaneous frequency, and spectral entropy of each signal. As with the statistics earlier, we note that deep knowledge of signal processing is not prerequisite knowledge for this course. The code first displays a graph with one AF and one normal example of the processed signal (the signals are labeled this time). Students are asked to perform a visual examination similar to the raw ECGs and describe what differences, if any, they can see between AF and normal for the processed signals. They then run the training and inference code that displays the same results as the first model, and they enter the same data into their spreadsheet. Training this model is significantly faster (3 min), and the training (~95%) and test accuracy (~95% accuracy, 95% sensitivity, 94% specificity) significantly higher than the raw model. The reflection prompts for this model differ by focusing on how feature extraction signals processing relates to the cardiac system and the cardiac behavior characteristic of AF. The teams, and class, then focus on comparing the accuracy, efficacy, and practical challenges of using this model with doctors over one trained on raw ECGs using a summary of all teams’ results (Figure 6). The expanded time of this activity also enables a discussion of the sensitivity and specificity and the prioritization of those in product development.

Neural Network 1 - Raw Data			
Parameter	Average	Min	Max
Training Time	11 min	11 min	11 min
Accuracy	60%	58%	66%
Sensitivity	39%	39%	39%
Specificity	78%	55%	78%

Neural Network 2 - Processed Data			
Parameter	Average	Min	Max
Training Time	3 min	3 min	3 min
Accuracy	94%	94%	94%
Sensitivity	97%	97%	97%
Specificity	91%	91%	91%

Figure 6. View of the class-compiled data comparing the two trained networks from the full version of the activity.

The final portion of the full version activity involved students choosing from three options to improve the model that used raw ECGs. Each team could choose to either (1) make the network significantly (100x) bigger by adding another hidden layer and increasing the number of neurons in each layer, (2) resample the data to have 2 AF signals for every normal signal during training, or (3) look at more signals in each training iteration. None of the options meaningfully improve the accuracy of the model, although options 1 and 2 significantly increase training time. The teams entered the results of the models in the Google Spreadsheet similar to the other two training runs, which produces another summary similar to Figure 6. The activity wraps up with an overall reflection and a discussion of diagnostic usefulness.

Computational resources and collaboration

We note that either version of this activity, but especially the full version, would not be possible without our on-campus access to significant computational resources. For those interested in a local version of the activity, we provide all of the pre-trained models[4]. However, we encourage others where possible to provide students the opportunity to perform the training themselves using institutional or commercial compute resources that they are likely to have never interacted with directly before. Below we describe the resources necessary and how we worked with our institutional compute resources team in planning and developing the activity.

The Instructional Cluster Environment (ICE) [25] is an investment by Georgia Tech in AI/ML education, providing students with a set of supercomputing resources exclusively for student learning and exploration. The ICE cluster enables the expansion of AI/ML content in courses and will eventually be open for independent student investigations. ICE is free and accessible to any student enrolled in a course that uses it – whether in class or on assignments – that any instructor at Georgia Tech has developed. To facilitate easy access to advanced computing by students with a wide variety of backgrounds, ICE uses Open OnDemand [28], a graphical web portal for supercomputers. MATLAB and other applications are offered through simple user interfaces that allocate compute resources by interacting with the Slurm scheduler behind the scenes.

The overarching cluster runs on Red Hat Enterprise Linux 9 and is equipped with MATLAB version R2023b. That version includes the Parallel Computing Toolbox to facilitate GPU-based training and the Deep Learning Toolbox to leverage predefined bidirectional long short-term memory (LSTM) architecture support. However, it is different than the version of MATLAB (2024b) that the code we modified is based on and did not contain some functions in the Deep Learning Toolbox implemented in the MATLAB demo[27]. This forced two of the authors to rewrite the inference code using functions available in 2023b (see appendices). The use of 2023b relates to issues with memory leaks we have encountered when operating more recent versions of MATLAB on our architecture.

[4] For those without access to such services, pre-trained model weights are available in the online repository linked in the Supplemental Materials. Those pre-trained models allow others to adopt with compressed version of the activity without access to training hardware.

For the activity, ICE support staff allocated one node per team of four students scheduled for the duration of the in-class session (or out-of-class testing). Students enrolled in the course were automatically provisioned with accounts on ICE, including home directories and scratch storage. To ensure availability during the activity, a reservation was created for the duration of all PSS session with enough nodes for each PSS group. Students were added to a POSIX group unique to the course, and the Slurm reservation used the magnetic feature to automatically place jobs submitted by these students during the reserved window onto the specified nodes, further simplifying the process for students.

Each node included 8 CPU cores, 256 GB of RAM, and one NVIDIA H100 SXM5 80 GB GPU. Under this configuration, a standard training of 30 epochs with a minibatch size of 200 on a preset bidirectional LSTM completes within approximately 10 minutes. The PhysioNet 2017 dataset used in the activity is less than 100 MB and is provided as serialized, single-file train and test sets from MATLAB to the node. Given the datasets' small size and format, we experienced no challenges regarding dataset storage or I/O performance across various storage media.

For readers without similar on-campus resources, various commercial vendors offer similar GPU computing services for MATLAB[29] (Table 3). One good option is Amazon Web Services (AWS) with the ability to integrate with the MathWorks Cloud Center to provide a paralleled experience close to native. Additionally, MathWorks collaborates with Microsoft Azure Marketplace to offer preconfigured MATLAB instances through their software plans. Both AWS and Azure instances support GPU acceleration and can be configured with the Campus-Wide License for academic use. However, neither of these plans includes free trials or free-tier GPU resources. More generic cloud computing options are also available via deploying MATLAB's Deep Learning Container[30]. However, this approach demands additional effort from course instructors to ensure proper configuration, conduct dry runs, and prepare additional instructions for students. We summarized the costs required for a single group to complete the two-hour session. It should be noted that miscellaneous costs, such as network bandwidth and data transfer, may apply but are not accounted for in the provided table. The extra time for the instructors to set and tune the materials beforehand is also not included.

Table 3. Commercial vendors and estimated costs for GPU computing services for MATLAB.

Commercial vendor option	**Cost of one session per group (2 hours, US Dollar)**
AWS EC2 p5.48xlarge**	~$27
Microsoft Azure ND96isr H100 v5**	~$25
Estimated H100 remote rental***	~$4-10
Notes: All instances feature H100 GPU and Red Hat Enterprise Linux (RHEL) operating system (Industry has higher rates with non-free Linux OS (i.e. RHEL). Actual costs likely to be slightly higher due to need for testing, setup, and overrun time. ** Dedicated instances to meet BYOL (Bring Your Own License) requirement *** Does not include MATLAB pre-configuration and would require the use of the docker hosting approach discussed above.	

Activity Evaluation

Methods

We collected multiple forms of data from the first offering of both versions of the activity. From the compressed version, we collected pre-post survey data from students as well as observations from the authors about the activity implementation and the curricular change efforts that surrounded it. From the full version of the activity we collected comments from the TAs who participated as well as observations from the authors who ran the implementation. Our goal at this stage is to address three research questions:

1) How does the compressed version of the activity affect students' confidence in their data skills?
2) Which data skills were most affected by the compressed version of the activity, in terms of confidence, perception of relevance to BME careers, and personal interest, and were those skills relevant to the activity?
3) What can we learn about the different implementations, and the process of introducing data skills into courses, that can aid future efforts?

Population and data collection procedures

For the compressed version of the activity, we sent a pre-post survey to all students enrolled in the class where the activity was implemented. Students were sent a pre-survey recruitment email several days before the exercise. The pre-survey was closed the morning the data skills exercise was scheduled. After the data skills exercise, students were sent a post-survey and given a week to complete the post-survey. Students were offered a small amount of extra credit (1% of their final grade) by completing both the pre- and post-surveys, regardless of their permission for their surveys to be used in this study. Per IRB rules, they could also receive the extra credit by writing a reflection about the data skills exercise for the same credit. Overall, 142 students were invited to participate in the data collection. One student opted to write a reflection to receive the extra credit, due to missing the pre-survey deadline. Of the remaining 141, 133 completed the pre-survey, 128 completed the post-survey, and we matched 115 pre- and post-surveys after removing non-responsive and duplicate responses. Due to the small number of volunteers who participated in the full activity (8 students, 4 teams), we did not collect survey data for the full version.

Our evaluation of the efficacy of the activity also includes informal comments from the course instructors and TAs during the compressed version as well as the participants in the full activity test run. Those are paired with observations from the developers of the activity in the results below. We chose not to collect any responses to the questions in the activity itself at this time, but plan to generally address student performance on the questions in future papers. All components of and the procedure for data collection were approved by the Georgia Tech IRB (protocol # H24396 and 2025-26).

Survey data

As a parallel part of our curriculum-wide effort to improve student data skills, we are in the process of developing a survey-based approach to measuring data skills. We expect to detail the development and testing of that survey in later work. In the absence of existing surveys that address data skills as learning outcomes, we generated a large pool of items based on concrete skills that map to the list of data skills described earlier. As noted earlier in the paper, these data skills aligned with those identified by others as essential in the future of engineering [13]. The survey presents each respondent with 15 of these items. Of the 15, 12 items are pre-selected and shown to all students with another 3 randomly selected from the item pool. Students are asked to respond to each general item in three forms: (1) Their confidence in their ability to apply each item, with guidance, to BME courses or work, (2) Their perception of the general applicability of the items to BME work, and (3) Their personal interest in jobs that make use of each of the items. All forms used a similar four-category response as follows:

- Confidence: No confidence, A little confidence, Some confidence, A great deal of confidence
- Applicability to Biomedical Engineering: None, A little, Some, A great deal
- Personal interest in jobs that use these skills: None, A little, Some, A great deal

We were particularly interested in the analysis of a subset of items most applicable to this activity and course context, specifically those about using preconfigured code and making sense of model credibility and applicability.

Results and Discussion

Research question 1: Growth in data skills confidence

We chose to focus on the confidence scale of the survey because we expect confidence to be most mediated by individual activities, with personal interest and relevance likely to be much more mediated by outside factors [31], [32]. Overall, the pre and post responses to all of the items were well distributed for this type of survey data (Figure 7). We expected, and saw, that the applicability of data skills was the most right-shifted (i.e., higher) while confidence was more centrally distributed. The finding that the combined set of pre-post confidence responses was the least skewed distribution gave us reasonable confidence in applying typical descriptive analytic techniques to the data.

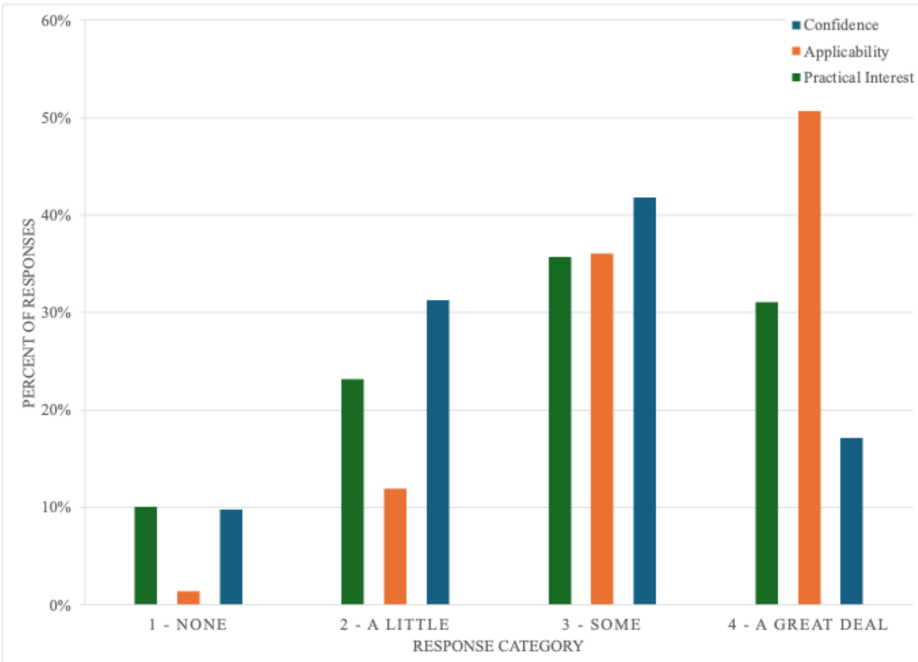

Figure 7 Overall distribution of pre and post survey responses, divided by form (i.e., confidence, applicability, and personal interest)

Overall, the survey results shifted towards greater confidence in their abilities (calculated as the average response to all confidence items) from the pre- to the post-survey (Figure 8). We assigned points to the responses (with *None* being 1 point and *A great deal* being 4). The average response to all of the items about confidence increased by 0.3 points ($p<.001$, from 2.5 to 2.8 - i.e., between *A little* and *Some*) on our confidence scale. While this growth is small within the scoring of the scale, it reflects a medium effect growth ($d=0.53$). From another perspective, 108 of 115 students saw a growth in data skills confidence pre-post. However, we note that while there are meaningful and significant changes, few students had a change in the median response category. That is, 65% (75) of students had the same median confidence response category between the pre and post-survey - even though their mean responses change. Given that we intend for the scale to measure longitudinal changes, we see changes that are smaller than an entire response category step to be likely across individual activities. This, among other reasons, motivated our choice to use means rather than medians as the main descriptive statistic in our analysis. While there is ongoing controversy about the best way to analyze ‘Likert-Style’ data, this choice is well grounded in literature that addresses misconceptions about the use of means (e.g., [33], which details the extensive history on the topic).

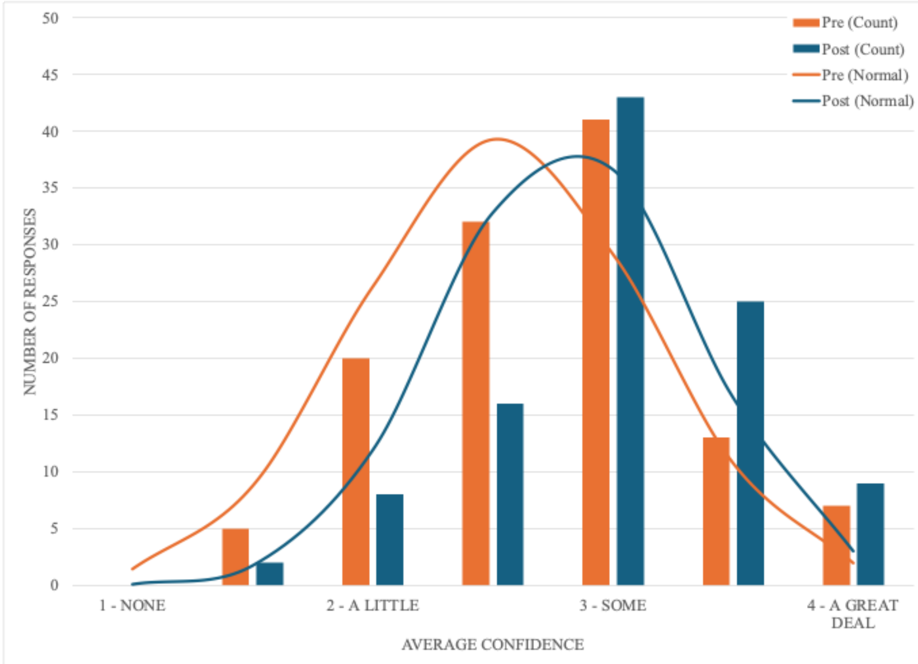

Figure 8 Comparison of average pre post response to confidence portion of data skills survey. Graph shows count of responses in half-response category steps (columns) and also shows a normal distribution of that same data set plotted at the same half-response category step increments (line graph)

Here, we note two things relevant to looking at the scale level change. First, because this is a curriculum-wide project where the intended impact is the result of multiple interventions, AND the survey is intended to measure that movement across a curriculum, we see small changes. Second, any changes to average confidence response will be mediated by the fact that no single activity will cover all of the data skills (i.e., items). For example, while this activity involved the use of MATLAB code and understanding the limitations of analyzing ECG raw data using neural networks (data skills 1 and 4a) it did not ask students to describe the methods or present data visually. For these reasons, when assessing individual data skills activities it is important, and defensible, to look at changes in individual items that are more relevant to a specific activity.

Research question 2: Insights from specific data skills

Next, we decided to examine the two items that showed the most change between pre- and post-survey responses. Our goal was to evaluate whether the change in response patterns was rational in the context of the item text and how that item text related to this specific activity. We did so to evaluate whether pre-post changes have a logical content basis. We plan to further investigate this in future research with data from multiple activities that are aligned with different items to evaluate construct-relevant and -irrelevant variation towards a validity argument for the instrument's use. The usefulness of breaking them out for this activity is not that each shows only positive impacts from the activity. In the remainder of the survey results we look at those two items across all three forms - confidence, applicability, and personal interest.

The item with the largest pre-post mean increase (Figure 9) was the the confidence response to the item *"Find and use prewritten software code, in a software language that you are unfamiliar with, to perform data analysis tasks"*. The average response went from 2.2 (near *A little*) to 3.0 (*Some*) with a similar change in the median category. While limited, we see this as evidence in support of our results' credibility because of its literal connections to the activity. For this activity, students were shown, then used prewritten software code from a source (MathWorks) with a vast library of similar code - and their confidence in using such a resource went up.

For the same item, we saw a no change (2.8 to 2.8, with 3.0 meaning *Some* interest), in interest (Figure 10). For applicability, there was no measurable change with both the pre- and the post-survey responses having an average of 3.1 and a median response of *Some*. We see these results as reinforcing our perception that changes in students' confidence, interest, and sense of applicability are likely to have a complex relationship with individual categories and the curriculum. Notably, the fact that applicability started and remained high suggests that these activities are responding to aspects of engineering practice students find engaging, but that we also need to consider how changes in public perception and media around AI and ML may be informing students' pre-existing knowledge and perceptions of this novel material.

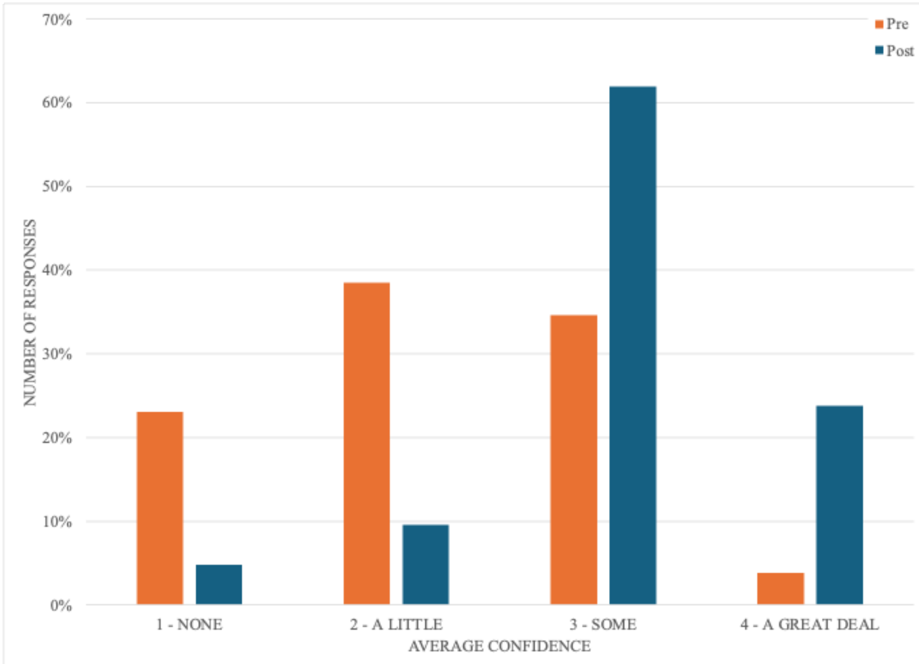

Figure 9 Comparison of the distribution of CONFIDENCE between pre and post-survey to the item *'find and use prewritten software code, in a software language you are unfamiliar with, to perform data analysis tasks'*. The average response increased by 0.8pts, and the median response increased from *A little* to *Some*). The y-axis is in percentage.

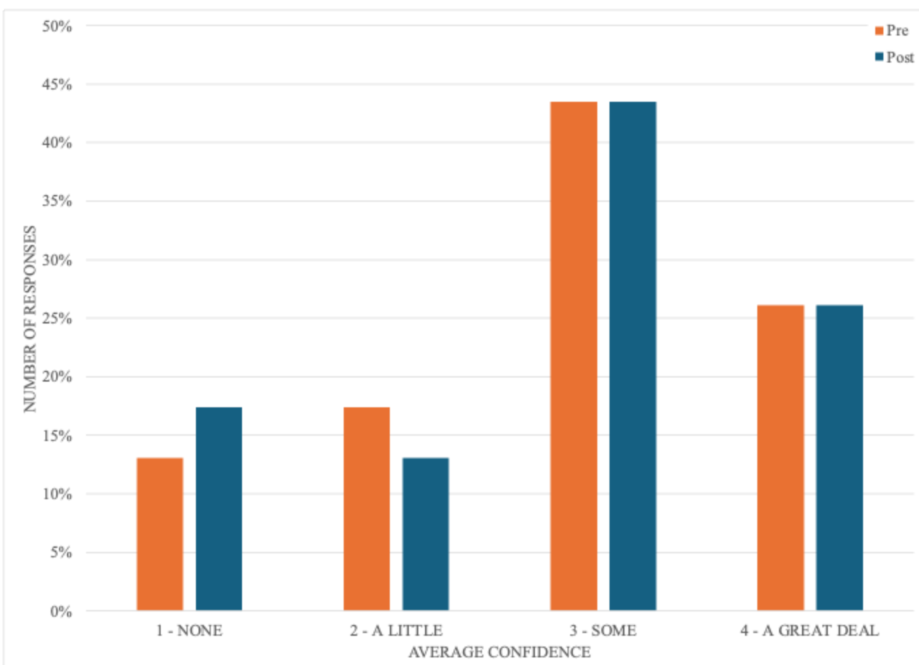

Figure 10 Comparison of the distribution of INTEREST between pre and post-survey to the item *'find and use prewritten software code, in a software language you are unfamiliar with, to perform data analysis tasks'*. The average response (2.8, near the top of *some*) and median response (*some*) stayed the same. The y-axis is in percentage.

In contrast, the item with the second highest pre-post growth in confidence was *justify decisions made in the process of applying statistical and machine learning techniques to data*. We see this item as a cautionary tale of students' perception of their skills and activities like this. Students' confidence increased by 0.8 points (2.1 to 2.9), with the median shifting from *A little* to *Some*. Their sense of the applicability of this skill rose by 0.3 points (3.1 to 3.4), with the median again rising from *A little* to *Some*. Finally, their personal interest rose 0.5 points (2.6 to 3.1), although the median response stayed the same - *Some*. One interpretation of this result is that students interpreted the item differently than we had intended. Students could have been thinking about using the *output* of the model to justify clinical decisions, but we intended the item to ask about decisions made when determining *how* to apply the model. However, we see this as highlighting a limit in activities that are constrained by existing courses and timelines. Students made no modeling decisions in the compressed version of the activity. Only one modeling choice was even shown and explained to them - converting ECGs to the frequency domain. However, their confidence grew significantly ($T(47)=2.9$, $p<.01$).

This reflects the inherent tradeoff in depth vs. connection in this activity, especially the compressed version. Put into educational parlance, this shows the potential risk of decoupling confidence and ability through a focus on explanation. The decoupling of perceived and actual learning has been widely reported and we expect it to be an ongoing challenge when managing the main constraints these activities present[34]. This is a tradeoff we are comfortable with given the educational purpose and context of these activities - curriculum-wide, focused on linking content, and serving as a launching point for a variety of opportunities to learn more. However, this question highlights that phenomenon in an individual activity, and the need to not treat this or any activity as being a single-point solution to introducing new material into a curriculum. This also highlights the importance of and value in isolating and analyzing the students' responses to individual items as part of any scale-level analysis.

Research question 3: Observations and comments from both versions of the activity

As noted, we also made qualitative observations about student actions and feedback in both implementations, with a special focus on the full version. We noted several small technical and logistical details which might be helpful to other instructors. These included explaining to students the need to have at least two laptops for each group, having MATLAB and the required toolkits installed, and (depending on the room) arriving with a fully charged battery. We anticipate future versions having a clear single page of 'preparation' instructions that include details such as the need to install a virtual private network (VPN) to access our cluster (for the full implementation).

Further, we expected students to be familiar with the use of the run sections and inline output of the live editor in MATLAB because it is used in multiple other courses in our department. The challenges students experienced most often were based on improper IDE use, such as looking at the command window not the live editor window for output, or being unclear about when computation was complete. An additional challenge was making progress more visible during the longer training and classification code runs because they were a longer time scale (minutes not seconds) compared to most code our students have written in other contexts. Options here include a printed counter for the classification code and enabling MATLAB's built-in progress window for the training code. However, we disabled the training window as we found that, in combination with the VPN remote desktop, and some software versions, it caused errors when students ran the training on our compute cluster. We also encountered a few MATLAB file-handling issues with Apple systems that we plan to fix.

There was also a need to prep teaching assistants on some of the terms that were glossed over in the worksheet or code. We expected that students would focus on what was asked, but found that the novelty of the activity seemed to motivate more engagement and questions. For example, the first model trained is a Long term-Short Memory (LTSM) neural network. Prior to the activity, we decided that an explanation of what an LTSM was and why we had chosen it was not relevant to the core activity and removed it. However, the TAs noted that if students were interested, it would be important to have the ability to answer some questions in the moment instead of the only option being to go look at the extra resources listed to learn more. We plan to address this in the future by adding a 'machine learning dictionary' to the back of the assignment that covers topics like 'what is a feature' and 'what is an epoch' as a better bridge. In the full activity version, students identified connections to content that used in later courses (e.g., a later lab course studies heart rate variability, or connecting to our statistics course.) A major summative note from students was the appreciation for something that linked the quantitative/math-heavy parts of our curriculum and the more information-heavy portions of our curriculum. While that does happen in later courses, which some of them had experienced, they appreciated the opportunity for it to happen earlier.

Regarding the overall activity design, students who ran the full activity version noted that they perceived the experience of training models on the cluster to be of high value. They also appreciated having the comparison of the cluster to local training efficiency. Conversely, the authors facilitating the full version noted a need for more precise, and more 'higher level' questions. We had kept the number of questions small to not overwhelm students but found that they were often left waiting for training runs to complete - time that could be filled with more reflection or connections to the physiological content. We expect to add questions about where this model could/could not be used in clinical support. We also observed that the options in the third run felt disconnected and random, and plan to restructure those to reflect three options within a single change - balanced, unbalanced, and over weighted - in the future.

Challenges and unexpected victories

There are two primary faculty-related challenges that we experienced in implementing this activity: one practical and one technical. The practical challenge was faculty buy-in. Multiple classes in our curriculum are now in the process of developing interventions from the skills and principles discussed in this paper, with two courses (Biotransport and Biomechanics) prioritized for 2025-2026. In that process, we have generally found that focusing on application without introducing the underlying mathematics was a challenging pedagogical principle for faculty. Specifically, the tradition of theoretically focused engineering teaching that is common in engineering is at odds with a willingness to focus on connecting information within a course, and an applications focus. We found three prominent examples of the challenge of faculty buy-in while planning for our activity:

1. They expressed concerns that students did not know the underlying mathematics of AI/ML tools.
2. They expressed frustration at having to 'fit' new material into the courses.
3. They were concerned about the time it would take for them to design and learn to implement a new activity.

We note these because they situate our efforts at curricular change within fairly common objections within the literature [35]–[37]. Through the development process, we have reinforced the pedagogical principles to faculty as well as situated the work in any individual course in the larger ecosystem of AI/ML learning opportunities for students. We also provided support for the time required to design the activity by using a team of instructors to develop, test, and refine student-facing materials. With an eye towards longer term and larger scale efforts, faculty in the course have recently expressed interest in developing similar activities to support learning of other areas of physiology content in the same course.

The technical challenge is related to the resources needed to replicate this activity. Because of course releases, the two authors who developed the activity were able to attend and assist with all offerings of the compressed and full versions. That lowered any potential barriers to knowledge and training for faculty and TAs involved in the course - both real (including the time it would take to acquire this training) and perceived. We, as faculty at [R1 institution], are also aware that we have computational resources available to us for free that others may not. While the decision to pre-train models in the compressed version was initially meant to be a "backup", meant to reduce the risk of the activity going wrong, we see it as useful to enable others without our resources to replicate the activity with lower cost (Table 3). If others are interested in replicating the activity, we include all of the pre-trained models in our GitHub repository [24].

Although we did not measure student learning of cardiac physiology and ECGs, the instructors of the course informally reported better student performance on exam questions related to ECGs compared to prior semesters without this activity. It would be valuable to measure this learning in the future. We also saw unexpected ways in which learning was created. We were surprised at how students used the Google Spreadsheet in the manual diagnostic portion of the activity. We included the Google Spreadsheet entry as a way to keep track of teams' progress but found students using it for self-directed learning. Once several teams had entered their diagnoses, it was very common for other teams to take these answers into consideration before entering their own diagnoses. They tended to answer the same way as other teams before them. When running the activity with a third group of students, we hid the answers of each team from the others. We noted that there was more variability in student diagnoses in this class. When using this activity, instructors can consider whether they do or do not want to show the work of each team to the whole class while the activity is in progress. We are inclined to make such work visible in the future, which can enable TAs and instructors to engage and ask questions about why teams changed answers to help uncover misconceptions or errors and reinforce learning about cardiac behavior. However, there is also an opportunity to force students to commit to a decision and then justify it. Our only definitive suggestion is to make clear that the diagnoses are not graded for accuracy.

Conclusions

Overall, we see this activity as generally achieving what we intended it to. That is, it made new learning about new material adjacent to physiology a part of the class. This both built the confidence of students in their data skills and abilities and better linked this course to other courses. As noted in the Results and Discussion section, the small changes in student perceptions about data skills make sense given that the respondents only had a 30-minute exposure to the activity. We are primarily interested in results across our curriculum as activities like this multiply. Overall, a critical but secondary implication of this activity was the impact it had on our faculty. Once we overcame the challenges of faculty buy-in for this activity, seeing what was changed grew the interest of a variety of faculty and courses in implementing similar activities. The low instructional overhead, the small course footprint, and the relative ‘coolness’ (to quote a participant) of getting to use the cluster all helped build further buy-in beyond the impact the activity had on students. None of that is to say that this activity was not without challenges or further work to do.

Future directions

As noted in the Results and Discussion section, we have several modifications we plan to make to the activity in future semesters. We plan to continue making the activity a part of the class. In future semesters we plan to take two primary actions related to research and assessment of the activity. The first is to collect students' answers to the activity questions to enable us to more directly evaluate learning. Second, we plan to work with the instructors to either offer the full version of the activity across the class or to have some sections offer each version to allow for a better comparison.

Acknowledgment

We thank the instructors of the course Drs. Bilal Hader, Michelle LaPlaca, Jeff Markowitz, and Shalu Suri as well as the course students, and TAs for their engagement and participation. We also acknowledge that the development and implementation of this activity was funded in part by a grant from the Kern Entrepreneurial Engineering Network (KEEN). Any opinions, findings, and conclusions, or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of KEEN.

References

[1] Committee on Envisioning the Data Science Discipline: The Undergraduate Perspective *et al.*, *Data Science for Undergraduates*. Washington, D.C.: National Academies Press, 2018.

[2] F. G. Yavuz and M. D. Ward, "Fostering undergraduate data science," *Am. Stat.*, vol. 74, no. 1, pp. 8–16, Jan. 2020.

[3] S. C. Hicks and R. A. Irizarry, "A guide to teaching data science," *Am. Stat.*, vol. 72, no. 4, pp. 382–391, Oct. 2018.

[4] H. J. Warraich, T. Tazbaz, and R. M. Califf, "FDA perspective on the regulation of artificial intelligence in health care and biomedicine," *JAMA*, vol. 333, no. 3, pp. 241–247, Jan. 2025.

[5] Center for Drug Evaluation and Research, "Artificial Intelligence for Drug Development," *U.S. Food and Drug Administration*, 20-Feb-2025. [Online]. Available: https://www.fda.gov/about-fda/center-drug-evaluation-and-research-cder/artificial-intelligence-drug-development. [Accessed: 21-Feb-2025].

[6] N. Sambasivan, S. Kapania, H. Highfill, D. Akrong, P. Paritosh, and L. M. Aroyo, "'Everyone wants to do the model work, not the data work': Data Cascades in High-Stakes AI," in *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, Yokohama Japan, 2021.

[7] J. Angwin, J. Larson, L. Kirchner, and S. Mattu, "Machine Bias," *#creator*, 23-May-2016. [Online]. Available: https://www.propublica.org/article/machine-bias-risk-assessments-in-criminal-sentencing. [Accessed: 21-Feb-2025].

[8] S. Sarp, M. Kuzlu, O. Popescu, V. Jovanovic, and Z. Acar, "Development of a data science curriculum for an engineering technology program," in *2023 ASEE Annual Conference & Exposition Proceedings*, Baltimore , Maryland, 2024.

[9] R. Scott, A. Frickenstein, and S. Wilhelm, "Longitudinal analysis of strategies for improving biomedical engineering student knowledge of career paths and desired skillsets," in *2024 ASEE Annual Conference & Exposition Proceedings*, Portland, Oregon, 2024.

[10] Cassandra Sue Ellen Jamison Annie AnMeng Wang Aileen Huang-Saad Shanna R. Daly Lisa R. Lattuca", "BME Career Exploration: Examining Students' Career Perspectives," in *2021 ASEE Virtual Annual Conference Content Access*, 2021.

[11] E. Frow and M. Caplan, "Promoting career reflection among freshman BME students," in *2017 ASEE Annual Conference & Exposition Proceedings*, Columbus, Ohio, 2018.

[12] "Minor in Applications of Artificial Intelligence and Machine Learning," *Georgia Institute of Technology Course Catalog*. [Online]. Available: https://catalog.gatech.edu/programs/minor-artificial-intelligence-machine-learning/. [Accessed: 24-Apr-2025].

[13] P. Mikalef, M. N. Giannakos, I. O. Pappas, and J. Krogstie, "The human side of big data: Understanding the skills of the data scientist in education and industry," in *2018 IEEE Global Engineering Education Conference (EDUCON)*, Tenerife, 2018, pp. 503–512.

[14] M. T. H. Chi and R. Wylie, "The ICAP framework: Linking cognitive engagement to active learning outcomes," *Educ. Psychol.*, vol. 49, no. 4, pp. 219–243, Oct. 2014.

[15] D. Kember, J. McKay, K. Sinclair, and F. K. Y. Wong, "A four-category scheme for coding and assessing the level of reflection in written work," *Assess. Eval. High. Educ.*, vol. 33, no. 4, pp. 369–379, Aug. 2008.

[16] R. A. Streveler, S. Brown, G. L. Herman, and D. Montfort, "Conceptual change and misconceptions in engineering education: Curriculum, measurement, and theory-focused approaches," in *Cambridge handbook of engineering education research*, Cambridge University Press, 2015, pp. 83–102.

[17] J. Doux and A. Waller, "The problem solving studio: An apprenticeship environment for aspiring engineers," *Advances in engineering education*, vol. 5, 2016.

[18] K. L. Ball, "Foundations in physiology: an introductory course using the core concepts," *Adv. Physiol. Educ.*, vol. 47, no. 3, pp. 501–507, Sep. 2023.

[19] K. Semsar *et al.*, "Phys-MAPS: a programmatic physiology assessment for introductory and advanced undergraduates," *Adv. Physiol. Educ.*, vol. 43, no. 1, pp. 15–27, Mar. 2019.

[20] *A Taxonomy for Learning, Teaching, and Assessing: A Revision of Bloom's Taxonomy of Educational Objectives*. .

[21] Z. Xiong, M. Stiles, and J. Zhao, "Robust ECG signal classification for the detection of atrial fibrillation using novel neural networks," in *2017 Computing in Cardiology Conference (CinC)*, 2017, pp. 1–4.

[22] B. Mika and D. Komorowski, "Higher-order spectral analysis combined with a convolution neural network for atrial fibrillation detection-preliminary study," *Sensors (Basel)*, vol. 24, no. 13, p. 4171, Jun. 2024.

[23] S. Nurmaini *et al.*, "Robust detection of atrial fibrillation from short-term electrocardiogram using convolutional neural networks," *Future Gener. Comput. Syst.*, vol. 113, pp. 304–317, Dec. 2020.

[24] *GT-BME-Data-skills-in-Undergrad-Curriculum*. Github.

[25] J. E. Coulter *et al.*, “ICE 2.0: Restructuring and growing an instructional HPC cluster,” in *Proceedings of the SC ’23 Workshops of The International Conference on High Performance Computing, Network, Storage, and Analysis*, Denver CO USA, 2023, pp. 591–597.
[26] G. D. Clifford *et al.*, “AF classification from a short single lead ECG recording: The PhysioNet/computing in cardiology challenge 2017.” PhysioNet, 01-Feb-2017.
[27] “Classify ECG Signals Using Long Short-Term Memory Networks.” [Online]. Available: https://www.mathworks.com/help/signal/ug/classify-ecg-signals-using-long-short-term-memory-networks.html. [Accessed: 09-Dec-2024].
[28] D. Hudak *et al.*, “Open OnDemand: A web-based client portal for HPC centers,” *J. Open Source Softw.*, vol. 3, no. 25, p. 622, May 2018.
[29] “Run MATLAB using GPUs in the Cloud.” [Online]. Available: https://www.mathworks.com/help/parallel-computing/run-matlab-using-cloud-gpus.html. [Accessed: 13-Jan-2025].
[30] “MATLAB Deep Learning Container on Docker Hub.” [Online]. Available: https://www.mathworks.com/help/cloudcenter/ug/matlab-deep-learning-container-on-docker-hub.html. [Accessed: 13-Jan-2025].
[31] B. M. Brisson *et al.*, “Short intervention, sustained effects: Promoting students’ math competence beliefs, effort, and achievement,” *Am. Educ. Res. J.*, vol. 54, no. 6, pp. 1048–1078, Dec. 2017.
[32] I. Estalella, Ó. Román, T. N. Reichenberger, and A. Maquibar, “Impact of a teaching strategy to promote evidence-based practice on nursing students’ knowledge and confidence in simulated clinical intervention choices,” *BMC Nurs.*, vol. 22, no. 1, p. 361, Oct. 2023.
[33] S. E. Harpe, “How to analyze Likert and other rating scale data,” *Curr. Pharm. Teach. Learn.*, vol. 7, no. 6, pp. 836–850, Nov. 2015.
[34] L. Deslauriers, L. S. McCarty, K. Miller, K. Callaghan, and G. Kestin, “Measuring actual learning versus feeling of learning in response to being actively engaged in the classroom,” *Proc. Natl. Acad. Sci. U. S. A.*, vol. 116, no. 39, pp. 19251–19257, Sep. 2019.
[35] Z. Sabagh and A. Saroyan, “Professors’ perceived barriers and incentives for teaching improvement,” *International Education Research*, vol. 2, no. 3, pp. 18–40, 2014.
[36] J. A. White *et al.*, “Core competencies for undergraduates in Bioengineering and Biomedical Engineering: Findings, consequences, and recommendations,” *Ann. Biomed. Eng.*, vol. 48, no. 3, pp. 905–912, Mar. 2020.
[37] B. Allen, A. S. McGough, and M. Devlin, “Toward a framework for teaching Artificial Intelligence to a Higher Education audience,” *ACM Trans. Comput. Educ.*, vol. 22, no. 2, pp. 1–29, Jun. 2022.

Supplemental materials

Due to size and paper length constraints all files for this activity are hosted on a github repository

Link to full repository of all published data skills materials:
https://github.com/owinslett3/GT-BME-Data-skills-in-Undergrad-Curriculum

Link to repository for the activity described in this paper:
https://github.com/owinslett3/GT-BME-Data-skills-in-Undergrad-Curriculum/tree/18539ef771874c06c28b18d2793a41c210ef876b/Systems%20Physiology\

Link to syllabus for the activity described in this paper:
https://github.com/owinslett3/GT-BME-Data-skills-in-Undergrad-Curriculum/blob/18539ef771874c06c28b18d2793a41c210ef876b/Systems%20Physiology/Systems%20Physiology%20Spring%202025%20Syllabus.pdf

Link to google sheets for the activity described in this paper:
https://github.com/owinslett3/GT-BME-Data-skills-in-Undergrad-Curriculum/blob/18539ef771874c06c28b18d2793a41c210ef876b/Systems%20Physiology/Systems%20Physiology%20Compressed%20Assignment/ECG%20Classification%20PSS.xlsx