



# An Open-Source AI/ML High School Curriculum Using Berkeley Lab STEM Research Data

SAGE MILLER<sup>\*1</sup>, ALISA BETTALE<sup>1</sup>, FAITH DUKES<sup>1</sup>

<sup>1</sup>Lawrence Berkeley National Laboratory, A-LIFT Office, Berkeley, CA 94720, USA

\*[SageMiller@lbl.gov](mailto:SageMiller@lbl.gov)

**Abstract:** Due to the rapid advancements of Artificial Intelligence (AI), it is increasingly important to implement AI and Machine Learning (ML) curriculum into the classroom so that students are well-equipped for future careers in the Science, Technology, Engineering, and Mathematics (STEM) ecosystem. Since 2024, members of the K-12 STEM Education team at Berkeley Lab have been updating our data science curriculum for our high school summer program, the Berkeley Lab Director's Apprenticeship Program (BLDAP): Interdisciplinary Pathways to Machine Learning and Data Science (IPMLDS), to include Machine Learning (ML) challenge Jupyter notebooks. This paper details the three ML challenge notebooks created by scientists, engineers, and members of the K-12 team at Berkeley Lab. Our curriculum has been structured and continuously adapted so that educators can introduce it in their classrooms to address the gap in AI curriculum at the high school level.

**Keywords:** artificial intelligence; machine learning; STEM education; high school curriculum; data science

© 2026 under the terms of the J ATE Open Access Publishing Agreement

## Introduction

Artificial Intelligence (AI) is rapidly changing the workforce [1]. Educational institutions will play a pivotal role in helping people adapt to these rapid advances in technology [2]. Introducing AI in the classroom can help equip students with the knowledge, skills, and tools necessary to address these technological changes. AI literacy has been identified as "crucial" for the next generation of leaders and innovators, emphasizing the importance of introducing this field early on in the classroom [3]. Introducing students to AI at a younger age helps them attain AI literacy and prepares them well for higher education and professional life [3]. Taking the steps towards achieving AI literacy in academia is therefore critical for all students. However, more resources and training opportunities are needed for educators to integrate AI into their classrooms.

At Lawrence Berkeley National Laboratory (Berkeley Lab) we have started to take steps towards achieving AI literacy by adapting our data science curriculum for our high school summer programs, particularly our Department of Energy Pathways Summer Program, the Berkeley Lab Director's Apprenticeship Program (BLDAP): Interdisciplinary Pathways to Machine Learning and Data Science (IPMLDS) [4]. This program is for students who are in their 11th or 12th grade of high school who have little to no coding experience. BLDAP seeks to remove barriers and a lack of access to STEM professionals in order to pique students' interest in pursuing a STEM career and prepare them for the workforce. While the program and curriculum were originally established in 2020, we began to implement AI and Machine Learning into our data science curriculum for BLDAP in 2024. The curriculum has been designed to fulfill two primary objectives. First, it aims to provide a solid foundation in computing concepts, especially for participants with little to no experience in this vital area. Second, the curriculum has been strategically designed to build capacity within the larger STEM education community. While we directly engage and support 20 students per year, 130 since 2020, we understand the importance of reaching a broader audience. To achieve this, we have made our curriculum open source to help educators integrate data science, AI, and ML into their classrooms. This approach allows us to significantly amplify the number of students introduced to critical technical skills



and concepts. It not only enhances individual learning opportunities but also fosters a collaborative environment for sharing resources and best practices, ultimately benefiting a larger community of learners.

As this curriculum is designed for students with little to no experience in coding, we begin by teaching students the very basics and fundamentals of Python and data science that are needed to understand and code the AI and ML challenge notebooks that appear later in the curriculum. By conducting a pre- and post-program coding assessment in BLDAP, we have strong evidence of student learning, proving the effectiveness of our curriculum. The post-program assessment demonstrates an increase in the percentage of questions answered correctly for every data science and ML category.

We have posted all of the lessons and resources on our GitHub so that educators can implement them in their classrooms and/or informal STEM programs [5]. This paper will focus on the AI/ML portions of the curriculum and provide resources for educators to implement this curriculum into their own educational spaces.

Our ML curriculum learning goals for students are as follows:

1. Evaluate how ML can be used as a tool to accelerate scientific research
2. Execute the training, testing, and evaluation of ML models
3. Compare and contrast model performance
4. Experiment with the ML cycle
5. Construct neural networks to get a deeper understanding of ML

## **Methods**

Our first lessons introduce students to the basics of data science and Python. They are important for building foundational coding skills that will be needed later in the ML challenge notebooks. Table 1 outlines the introductory Python notebooks that students complete prior to completing the ML challenge notebooks. Each lesson outlined in Table 1 includes a slideshow presentation, a Jupyter notebook, and checks for understanding. Evaluating student understanding ranges from exit tickets to weekly assessments to interactive review games.

The notebooks listed in Table 1 require varying amounts of time to teach and complete; however, a 2.5-hour block is usually sufficient to complete one notebook and start the next. The exception to this is the 04, 07, and 08 notebooks in Table 1, as these generally took students longer to complete. Students normally finish the 06 notebook in about 20-30 minutes. In our STEM program, students vary in their knowledge, understanding, and experience. As a result, while we observe year-over-year improvement, the pace of each session varies depending on the specific group of students and their overall educational backgrounds. While some years, students got through the introductory notebooks quickly, other groups required much more time. Educators working with students who have some prior coding experience can select which notebooks to use to ensure students are prepared for the ML notebooks. As will be discussed in further detail later in this paper, the prerequisite knowledge in relation to these introductory notebooks will be specified for each ML challenge notebook. There is also more attention now in identifying key AI concepts that should be covered in K-12 CS curriculum, leading to CSTA & AI4K12 developing AI Learning Outcomes [6]. For the Cell Phone Design Challenge and Sustainable Biofuels Challenge, we have identified how the ML concepts covered align with the Prioritized Foundational K-12 AI Learning Outcomes.

Once students complete notebooks 00-08 (Table 1), they will be ready for their first introduction to Machine Learning. Before starting the ML challenge notebooks themselves, students participate in the Machine Learning Unplugged activity developed by Tech Interactive in San Jose, California [7]. The Machine Learning Unplugged activity introduces students to key concepts, including algorithms and classification.



In the Machine Learning Unplugged activity, students categorize butterflies, moths, and other organisms based on the specific set of rules given and then revise this algorithm for the second iteration of the activity [7]. With this activity, students not only learn about the fundamentals of Machine Learning but also gain an understanding of realistic ways Machine Learning can be applied. While the ML challenge notebooks focus on regression rather than classification, this activity is still recommended for students to complete, as it introduces AI, Machine Learning, and algorithms in an accessible way. Once students complete the Machine Learning unplugged activity, they start the ML challenge notebooks below.

**Table 1. Progression of topics for introductory Python modules**

Notebook	Topics Covered
00 - Intro to Python	Basics of Jupyter notebooks, data types, and variables.
01 - Functions	Built-in functions, defining functions, importing libraries.
02 - Lists and Dictionaries	List iteration and slicing, adding/deleting elements, length of lists, operations on lists. Creating dictionaries and adding/deleting elements in dictionaries.
03 - Conditionals	Booleans, comparison operators, and if/else statements.
04 - Loops	For loops, while loops.
05 - NumPy arrays	Creating NumPy arrays, indexing/slicing, operations.
06 - Intro to pandas	Introduction to dataframes, statistics, and histograms.
07 - pandas Dataframes	Indexing and slicing, uniqueness, frequencies, sorting, min/max/range, missing values, and boolean indexing.
08 - Data Visualization	Introduction to matplotlib: line plots, scatter plots, bar plots, histograms. Data visualization exercises using supernovae data and energy data.

### Cell Phone Design Challenge

The Cell Phone Design Challenge is based on the Materials Project [8], a free online database of computed properties for over 130,000 inorganic compounds [9]. It is maintained by more than forty researchers at Berkeley Lab and other universities. The Materials Project allows researchers to identify promising materials from the database to test experimentally, saving them time and cost. The learning objectives of the Cell Phone Design challenge are for students to become comfortable using the pandas library on real datasets and learn how machine learning can be critical in predicting missing, crucial data. This notebook follows the Prioritized Foundational K-12 AI Learning Outcomes as it has students train two ML models on data, evaluate these models, and compare and contrast how well the two different models performed [6].

These slides first go over a general introduction to data science and then introduce the Materials Project before students work on the first part of the Cell Phone Design Challenge [10]. The Cell Phone Design Challenge, part 2, also has a set of slides that introduces students to machine learning.



```
# EXERCISE - Fill in numbers below that meet the transparency, conductivity, cost, and scarcity requirements above:

transparent_filter = (
    (material_data["transparency"] > ...)
    & (material_data["conductivity"] > ...)
    & (material_data["volume_cost"] * 1 < ...)
    & (material_data["scarcity"] < ...)
)

material_data[transparent_filter].sort_values("density")
```

*Fig. 1. Skeleton code that students use to filter the DataFrame to return all materials that meet the necessary requirements for their transparent coating material.*

### **Suggested Prerequisites for the Cell Phone Design Challenge**

1. Notebook 00 (Data Types and Variables)
2. Notebook 03 (Conditionals)
3. Notebooks 06 and 07 (Pandas)
4. Notebook 08 (Data Visualization)
5. The Tech Interactive's Machine Learning Unplugged activity

### **Workflow of the Cell Phone Design Challenge**

The Cell Phone Design Challenge notebook consists of two parts. For the first part, students work with a dataset compiled from the Materials Project database of calculated material properties to design a cell phone that meets criteria such as low cost, lightweight, durability, etc. Students first clean the dataset to remove any incorrect data entries and then design their cell phones.

When designing their cell phones, students use pandas DataFrames and filtering by using boolean indexing to find materials that meet desired criteria, as pictured in Fig. 1. For example, students will have to meet certain requirements (i.e., volume cost and scarcity) when choosing a protective casing material, a battery material, and a conductive transparent coating material for their cell phones. In the end, students can test their phones to see how well they will do in the market.

The second part of the challenge introduces students to regression algorithms in machine learning in order to predict the hardness of a material based solely on its density. As an introductory machine learning activity, students are guided through each process of the machine learning cycle: preprocessing the data, specifying a model, training the model, and evaluating the model. Students have an opportunity to test different regression models from the Python machine learning library scikit-learn to compare results. Students will first be using the Random Forest Regressor model [11]. Students will calculate the root mean square error (RMSE) to evaluate how well the model performed at predicting the hardness based on the density. At the end of the notebook, students will train and evaluate the Ordinary Least Squares model and calculate the RMSE to see how well this model performed compared to the Random Forest Regressor model. The smaller the RMSE, the better the model performed.

### **Machine Learning Application: Sustainable Biofuels**

The main goals of the Sustainable Biofuels notebook are to train a machine learning model to predict the flash point of fuels using their spectra, apply the train-test split technique to validate the machine learning model, and use feature selection as a way to avoid the machine learning model from overfitting. By completing this notebook, students will not only train and test different Machine Learning models but also learn how ML is being used in research to combat real-world issues such as climate change. This notebook satisfies the Reasoning and the Building and Using AI Models Foundational K-12 AI Learning Outcomes by using ML to help solve a real-world problem and training and evaluating multiple models to assess their



strengths and limitations [6]. Given that aviation is responsible for a large portion of Greenhouse Gas emissions, sustainable aviation fuel is a possible solution to help mitigate carbon emissions [12].

Machine learning can help accelerate alternative jet fuel research, and through this notebook, students will gain hands-on experience training a machine learning model capable of predicting fuel properties early in the development cycle.

### Suggested Prerequisites for the Sustainable Biofuels Notebook

1. Notebook 00 (Data Types and Variables)
2. Notebook 01 (Functions)
3. Notebook 02 (Lists and Dictionaries)
4. Notebook 03 (Conditionals)
5. Notebook 04 (Loops)
6. Notebooks 06 and 07 (Pandas)
7. Notebook 08 (Data Visualization)

### Workflow of the Sustainable Biofuels Notebook

In this notebook, students use Fourier Transform Infrared (FTIR) spectroscopy to predict the flash point of fuels. Since this notebook has several components and can be challenging, it is often split into two days. Day one consists of performing Exploratory Data Analysis (EDA), cleaning the data, doing a train-test split, and finally doing feature selection to avoid overfitting.

Students first use pandas to import the flash point and spectra data CSV file and perform EDA to become familiar with the dataset they are working with for this notebook. Students then normalize their data so that they can compare their spectra on a relative scale rather than directly to account for instrumental error and noise. The spectra are the features that are used to predict the target variable, which is the flash point of these fuels. Since it is critical to test how well our model is performing, students will need to split their data: reserving 30% for testing and 70% for training.

```
from sklearn.feature_selection import RFE
import time
tic=time.time()
rfe=RFE(estimator=RandomForestRegressor(random_state=42),n_features_to_select=...,step=...)

rfe.fit(X_train,y_train)
rfe_X_train=rfe.transform(X_train)
rfe_X_test=rfe.transform(X_test)
rfe_model=RandomForestRegressor(random_state=42)
rfe_model.fit(rfe_X_train,y_train)
toc=time.time()

print('Mean Absolute Error: '+str(mean_absolute_error(... , ...)))
```

*Fig. 2. Recursive feature elimination to avoid overfitting. Students will input the number of features they wish to select and the step size.*

One problem when training a model is having too many features (in this case, spectra) that aren't useful in predicting the target variable accurately. If we have too many features, then our model will learn our training set too well, resulting in very good predictions for our training set but failing to predict new data accurately. This is known as *overfitting*. To avoid overfitting, students need to perform feature selection, which cuts down on the number of features used to train their models and only uses the features that help predict the



flash point accurately. Students will be introduced to two types of feature selection: binning and recursive feature elimination, so that they will be able to reduce the number of features used to train the model. Binning aggregates multiple values into the same bin by taking the average of those values. This helps smooth the spectra, eliminating small peaks that aren't important in predicting the flash point of the fuel. Recursive feature elimination is another type of feature selection which removes the least important features (spectra) and is shown in Fig. 2. It is able to do this by training a model that ranks the features based on importance and then removes the least important features until students are left with the number of most important features they defined in `n_features_to_select` in Fig. 2. Note that in Fig. 2, `step` is equal to the number of features removed at each iteration until the `n_features_to_select` is reached. This wraps up day one.

Day two is when students train their models. Students will be testing out two different models: Linear Regression and Random Forest Regressor. After training these two models, students will see how well the models predict the training and test sets. To evaluate how well the models are performing, students will find the mean absolute error (MAE). The smaller the MAE, the more accurate the model is at predicting the target variable (in this case, the flash point of the fuel). Students will then be able to compare which model performed better. Day two finishes off with a challenge where students predict the flash point of a new fuel with their model, seeing how close the model gets to predicting the actual flash point of this new fuel.

The slides walk through the entire notebook and take students step-by-step through the process [10]. The *Science in Motion: Accelerating the Search for Sustainable Jet Fuel* video provides a good overview of the project and is recommended for showing to students at the start of this activity [12].

## Neural Networks

Taking inspiration from the brain, artificial neural networks are made from simple building blocks interconnected by varying strengths, called weights [13]. Throughout this coding notebook, students will learn how powerful neural networks are, building a foundation for one of the most fundamental subsets of machine learning.

### Suggested Prerequisites for the Neural Networks Notebook

1. Notebook 00 (Data Types and Variables)
2. Notebook 01 (Functions)
3. Notebook 02 (Lists and Dictionaries to learn about indexing)
4. Notebook 03 (Conditionals)
5. Notebook 04 (Loops)
6. Notebook 05 (Arrays)
7. Notebook 08 (Data Visualization)

### Workflow of the Neural Networks Notebook

The goals of this notebook are to teach students the foundational concepts and vocabulary needed to understand the basics of neural networks (input and output neurons, weights, bias, feedforward, hidden layer, etc.) and for each student to generate an image of their own neural network. Fig. 3 shows the steps students take to create a simple neural network that has two input neurons ( $x_1$  and  $x_2$ ) and one output neuron ( $y$ ). Students start this notebook by defining the values of the input neurons, weights, and bias, computing the value of the output neuron using the equation  $y = w_1 \times x_1 + w_2 \times x_2 + b$ , and applying the activation function.



```
# inputs
x1 = ...
x2 = ...

# weights
w1 = ...
w2 = ...

# bias
b = ...

# output
y = ...

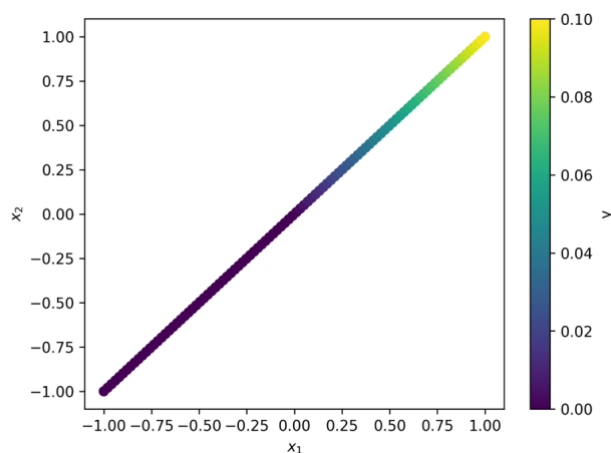
print(f"The value of my output neuron, y, before activation = {y}")

# activation function
if y < 0:
    y = ...

# final output
print(f"The value of my output neuron, y, after activation = {y}")
```

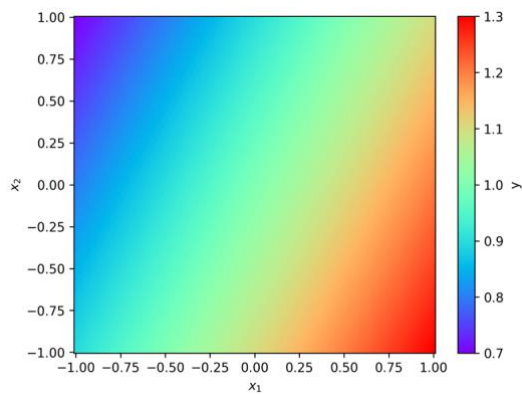
*Fig. 3. Students will first fill in the missing code to create a simple neural network.*

As students progress through the exercises, they will plot their results by having the coordinates of  $x_1$  and  $x_2$  be colored depending on the value of  $y$ , as pictured in Fig. 4. Students will eventually incorporate loops and arrays as their neural networks become more sophisticated. When hidden layers are introduced, students will learn about deep neural networks, learning key concepts such as the feedforward algorithm. By exercise 5, students will make an image by generating neurons in a matrix. Students are able to do this because the two input neurons,  $x_1$  and  $x_2$ , can be interpreted as the positions of the pixels in an image, where each pixel corresponds to a different input sample, as seen in Fig. 5.



*Fig. 4. Each coordinate of  $x_1$ ,  $x_2$  is colored depending on the value of  $y$ .*

In order for students to be able to visualize more complex neural networks, we have made the code more general so it can be used to add multiple hidden layers. When the code gets more complicated in exercises 6 through 9, most of the code will already be given, and students will only have to fill in some critical details, such as input and output layer size, max weight, max bias, number of neurons in each hidden layer, and image size. Once hidden layers are introduced in exercise 7, students may not understand every single line of code, but the emphasis is for students to understand the underlying logic at this point in the notebook.



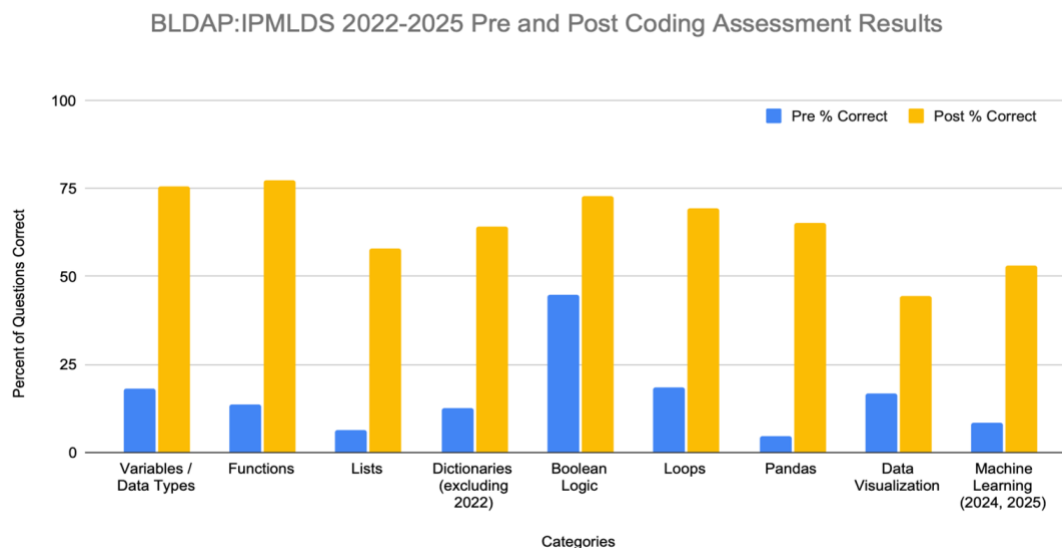
*Fig. 5. Each pixel corresponds to a different input sample, which is color-coded depending on the output of that sample.*

In the final exercise, students will get to choose the parameters of their neural network, such as the number of hidden layers, the number of neurons in each hidden layer, and the maximum weight and bias. By the end of this exercise, students will generate an image of their own neural network.

The slides are helpful for guiding students through the notebook and explaining the key concepts that this notebook is built on [10]. Although the notebook itself doesn't have students train the neural network, the slides review training. There are also diagrams in the slides that are recommended for visualizing a neural network.

## **Results and Discussion**

Starting with the 2022 cohort, students take pre- and post-program data science and ML assessments to evaluate student learning. Key concepts from modules throughout BLDAP are shown in Fig. 6.



*Fig. 6. Pre- and Post- Program Assessment Results (2022-2025).*



## BLDAP 2024-2025 Pre- and Post-Program Assessment Machine Learning Results

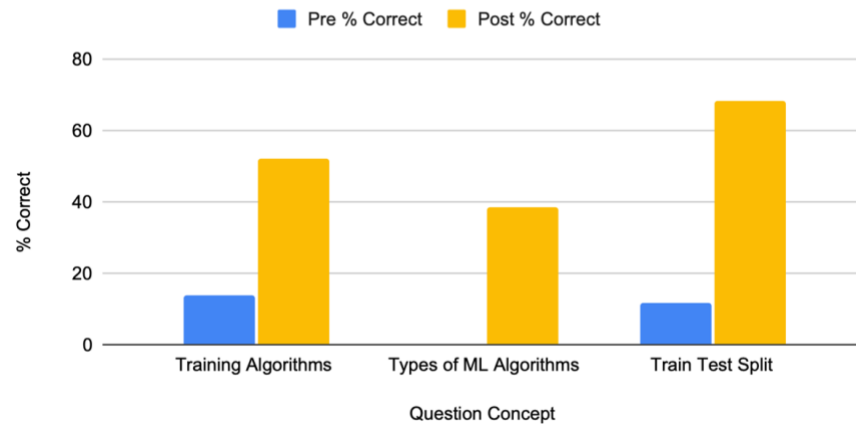


Fig. 7. Pre- and Post- Program Assessment Results for ML Questions (2024-2025).

## TRIO 2025 Pre and Post Coding Assessment Results

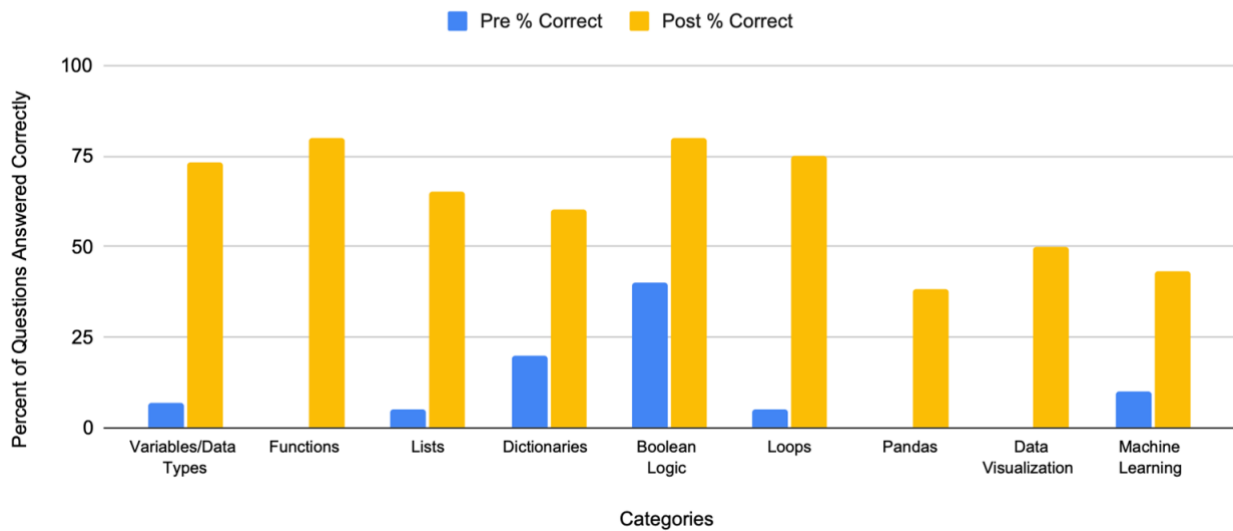


Fig. 8. UC Berkeley TRIO program's pre- and post-program coding assessment results. These results are quite similar to previous years of BLDAP.

Fig. 6 depicts the results of the pre-program and post-program assessments from BLDAP cohorts 2022-2025. Students are given the exact same set of questions before and after the program. Note that BLDAP does not give any homework or tests throughout the program, so the post-program results are purely what students have retained while in the program from the presentations and the Jupyter notebooks.

Across all data science and ML categories, students have an increase in the percentage of questions they got correct in the post-program assessment compared to the pre-program assessment. We conducted a paired t-test for each of the categories listed in Fig. 6. The categories Variables/Data Types (0.0006), Functions (0.0016), Lists (0.0298), Dictionaries (0.0094), Boolean Logic (0.0115), Loops (0.0004), and



pandas (0.0006) had a statistically significant increase from the pre-program to the post-program percent of questions answered correctly.

Machine Learning was not introduced into the pre- and post-program assessment until 2024, so there is less student data compared to the other categories in the pre- and post-program assessment. Nonetheless, there was still an increase in the percentage of questions answered correctly in the post-program assessment compared to the pre-program, as Fig. 7 shows. Three out of the twenty questions in the pre-program and post-program assessments revolve around ML concepts. Fig. 7 depicts these three concepts and the pre- and post-program percent of questions answered correctly. Note that for the category Types of ML Algorithms, 0% of students answered correctly for the pre-program assessment. Examples of questions asked include why we perform a train/test split when building an ML model and what is important to keep in mind regarding data used to train ML algorithms.

We set out to achieve several learning goals. Both the Cell Phone Design Challenge and Sustainable Biofuels notebook use real Berkeley Lab data and showcase how data science and ML are being used to accelerate scientific research. Students learn how ML can be used to predict missing data, and how ML can contribute to combating environmental degradation through the accelerated research of sustainable biofuels. In both the Cell Phone Design Challenge and Sustainable Biofuels notebook, students learn how to train, test, and evaluate different ML models such as Random Forest Regressor, Ordinary Least Squares, and Linear Regression. Students calculate the RMSE to evaluate these models. With both of these notebooks, students experiment with the ML cycle by preprocessing the data, choosing, training, and testing a model, and comparing and contrasting the model's performance with other models. Finally, we wanted students to get a hands-on activity with a critical subset of ML, neural networks. This is accomplished through the Neural Networks notebook by having students build their own neural networks. Through this notebook, students investigate and visualize one of the fundamental mechanisms used to train large, complex ML models.

As of 2025, parts of this curriculum have been adopted by two high school programs: the UC Berkeley TRIO program and the Pritzker School of Molecular Engineering After School Matters STEM Lab Internship Program. In particular, two UC Berkeley students who led the UC Berkeley TRIO program utilized the same pre-program and post-program assessment to determine whether BLDAP's data science and ML curriculum was transferable. After the data was analyzed, it was determined by these two leads that "LBNL data science and machine learning curriculum is accessible and transferable to satellite programs." The UC Berkeley TRIO program's pre- and post-assessments are shown in Fig. 8.

## **Conclusion**

This curriculum has been made open source and can be found on our GitHub [5]. One of our main goals is that this curriculum can be implemented in other educators' classrooms to help bridge the gap in data science and ML curriculum in schools. As has been demonstrated by the two satellite programs, BLDAP's curriculum can be adapted for external high school programs. We are continuously improving our curriculum and welcome any feedback. Our aim is to support educators and students alike by providing an open-source curriculum that supports the introduction of data science and ML skills to the broader STEM education community.

**Acknowledgements.** This work was supported by the following collaborators.

*Introduction to Python / Data Science:* UC Berkeley's Data Science Modules Program, College of Computing, Data Science, and Society. Kseniya Usovich, Karla Palos, Eric Van Dusen, Rachel McCarty



(UC Berkeley); Baishakhi Bose, Samanvitha Murthy, Arianna Formenti (Berkeley Lab); Laurel Hales (Stanford).

*Cell Phone Design Challenge*: Alex Ganose (Imperial College London); Ryan Kingsbury (Princeton University); Barbara Bonfim (WWF-Brazil); Jianli Cheng, Rishabh Guha (Schrödinger Inc.); Ruoxi Yang (Tesla); Oxana Andriuc (UC Berkeley); Jingyang Wang (Beijing Normal University); Aaron Kaplan, Roberta Pascazio (Berkeley Lab).

*Sustainable Biofuels for Aviation Machine Learning Challenge*: Ana Comesana, Vi Rapp (Berkeley Lab). DOE's Bioenergy Technologies Office for funding this work. Kyle Niemeyer at Oregon State University as the PI and collaborator for this work. Christopher Hagen at Oregon State University for his early support with FTIR spectra, Joshua Heyne at WSU-PNNL Bioproducts Institute for providing experimental property data, and Sharon Chen for her efforts in collecting data.

*Neural Networks*: Arianna Formenti.

*UC Berkeley TRIO Program*: Pavithra Arun Anand, Wayland La, David Gonzalez, Bhavani Bindiganavile,

*Department of Energy Office of Science Workforce Development for Teachers and Scientists Pathway Summer Schools.*

*Berkeley Lab Undergraduate Research (BLUR)*

**Disclosures.** The authors declare no conflicts of interest.

## References

[1] G. R. Lokesh, K. S. Harish, V. S. Sangu, S. Prabakar, V. Santhosh Kumar, and M. Vallabhaneni, "AI and the future of work: Preparing the workforce for technological shifts and skill evolution," in Proc. 2024 Int. Conf. Knowledge Engineering and Communication Systems (ICKECS), 2024, pp. 1–6. doi: [10.1109/ICKECS61492.2024.10616486](https://doi.org/10.1109/ICKECS61492.2024.10616486).

[2] L. Amini, H. F. Korth, N. Patel, E. Peck, and B. Zorn, "Empowering the future workforce: Prioritizing education for the AI-accelerated job market," CRA Quadrennial Paper, 2024- 2025. [Online]. Available: [https://cra.org/wp-content/uploads/2025/02/2024-2025-CRA-Quad-Paper\\_-\\_Empowering-the-Future-Workforce-Prioritizing-Education-for-the-AI-Accelerated-Job-Market.pdf](https://cra.org/wp-content/uploads/2025/02/2024-2025-CRA-Quad-Paper_-_Empowering-the-Future-Workforce-Prioritizing-Education-for-the-AI-Accelerated-Job-Market.pdf)

[3] Y. Walter, "Embracing the future of artificial intelligence in the classroom: The relevance of AI literacy, prompt engineering, and critical thinking in modern education," Int. J. Educ. Technol. High. Educ., Feb. 26, 2024. [Online]. Available: <https://educationaltechnologyjournal.springeropen.com/articles/10.1186/s41239-024-00448-3>

[4] A-LIFT: Pre-College Programs, "Director's Apprenticeship Program: Interdisciplinary Pathways to Machine Learning and Data Science." Accessed: Mar. 9, 2026. [Online]. Available: <https://k12education.lbl.gov/programs/high-school/BLDAP>

[5] LBNLnext, "BLDAP-Python-Data-Science," GitHub repository. Accessed: Mar. 9, 2026. [Online]. Available: <https://github.com/LBNLnext/BLDAP-Python-Data-Science>

[6] CSTA and AI4K12, AI learning priorities for all K-12 students. Computer Science Teachers Association, 2025. [Online]. Available: <https://csteachers.org/ai-priorities/>

[7] The Tech Interactive, "Machine Learning Unplugged." Accessed: Mar. 9, 2026. [Online]. Available: <https://www.thetech.org/education/education-resources/lessons/machine-learning-unplugged/>



[8] Materials Project, “The Materials Project.” Accessed: Mar. 9, 2026. [Online]. Available: <https://next-gen.materialsproject.org/>

[9] A. Jain et al., “Commentary: The Materials Project: A materials genome approach to accelerating materials innovation,” *APL Materials*, vol. 1, no. 1, p. 011002, 2013. doi: 10.1063/1.4812323.

[10] LBNLnext, “BLDAP-Python-Data-Science Slides,” GitHub repository. Accessed: Mar. 9, 2026. [Online]. Available: <https://github.com/LBNLnext/BLDAP-Python-Data-Science/tree/main/Slides>

[11] L. Breiman, “Random forests,” *Machine Learning*, vol. 45, no. 1, pp. 5–32, Oct. 2001, doi: 10.1023/A:1010933404324.

[12] J. Nuss, “Science in motion: Accelerating the search for sustainable jet fuel,” YouTube, Oct. 19, 2023. [Video]. Available: <https://www.youtube.com/watch?v=rFal8qRGUjI>

[13] J. Schmidhuber, “Deep learning in neural networks: An overview,” *Neural Networks*, vol. 61, pp. 85–117, Jan. 2015, doi: 10.1016/j.neunet.2014.09.003.