AI4ALL: AI Education for K-12

Yiqiao Yin

Abstract

Many scholars and educators have dedicated their lives in K12 education system and there has been an exploding amount of attention to implement technical foundations for Artificial Intelligence Education for high school and pre-college level students. This paper focuses on the development and use of resources to support K-12 education in Artificial Intelligence (AI). The author and his team have more than three years of experience coaching students from pre-college level age from 15 to 18. This paper is a culmination of the experience and proposed online tools, software demos, and structured activities for high school students. The paper also addresses a portfolio of AI concepts as well as the expected learning outcomes. All resources are provided with online videos and Github repositories for immediate use.

Motivation of Coaching K-12 Students

There is an exploding amount of online tutorials and coaching strategies for pre-college level education in Artificial Intelligence (AI) [2, 8, 14, 3, 5, 6]. Increasingly, product developers, AI researchers, education technology advocates and venture capitalists are turning their attention to education particularly targeting the future of kindergarten through grade 12 (K-12) education [13, 2, 14, 11, 12] as well as AI ethics related topics [15, 12]. Many famous platform has been setting up solid foundations for entry level students to engage the topics of Machine Learning or other related AI education. ReadyAI is the first comprehensive K-12 AI education company that provided online coaching tools for pre-college students and teachers [1]. The program empowers students and teachers the tools to understand AI and put emphasis on the non-technical components of learning. Calypso is another famous online software platform that provides programming to another level [9]. They provided licensed personal usage of human-friendly robot AI framework to allow students to engage robotics and AI in a fun way with a wide range of applications in computer vision, speech recognition, and artificial intelligence. Another company Cognimates sits at the frontier of AI education platform by providing games, programming robots and training AI models with code-free approach [10]. From what we have seen in the past three years, these coding-free and non-technical trend of approaches can no longer satisfy the needs of the students. High school students have been actively preparing for Advanced Placement (AP) courses and take AP exams for variety of reasons [4]. The author has shown high correlation between AP exams and the first-year grades for bachelor students [4]. The conventional assumption that high school students are not equipped to understand the foundational mathematics of AI topics no longer hold true. According to the 2020 Collegeboard Archive [7], there were about 4.7 million students world wide that had taken the AP exams. In U.S. alone, a total 4.59 million students took the AP exams in the year of 2020 (the most recent data released by CollegeBoard). In the year of 2020, the highest ranked exams in terms of proportion is English Language and Comp. The number of exams taken for AP Calculus AB and BC are 266,430 and 127,864, respectively. From our experience, students are fully capable of understanding common machine learning (ML) and deep learning (DL) topics such as "overfitting", "loss function", "gradient descent", and so on. This article proposes to all teachers, in order to achieve true AI4ALL, to not ignore the technical component but to encourage the students to understand the mathematical component behind real AI development. To execute what is advocated, we provide a portfolio of resources to help anyone (teachers and students) to understand AI from not just application and coding but also technical and mathematical perspective.

Proposed Course Materials and Research Topics

The proposed course materials are structured by three components: lecture, coding, and student project.

The lecture is generally provided by the instructors. The instructor (or teacher) provides an educational talk to an audience (students) live in person at a designated classroom or remotely via online video platforms such as Zoom, WebEx, Teams, or Google Meet. The students participate the lectures and receive the first formal presentation intended for a series of subjects. Subjects are discussed in the next subsection (see Section: Lecture Subjects).

The coding component or live coding walk-through is proposed to be the next step following a lecture. The beginning of the learning curve in any programming languages can be extreme challenging. Patience and Passion often work together and it is a common trend that the students with a lot of passion towards a subject would naturally want to spend more time to tackle the problem. However, passion

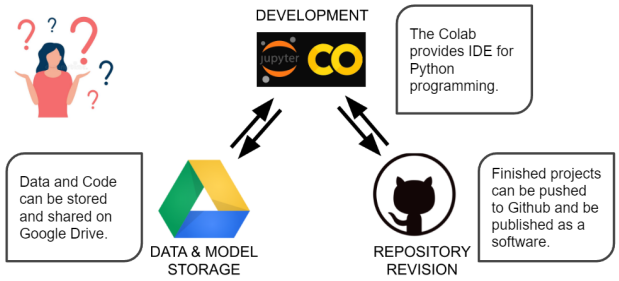
can not be guaranteed for all students and sometimes it is up to the instructor and teacher to guide the students through the difficult roadblocks. The coding component is usually recommended to be live session where the instructor does a live coding walk-through in front of the students with built-in quizzes inside of the code. The materials will be discussed and provided in the resource section (see Sections: Live Coding Walk-through and Resources).

The proposed AI education courses land on a capstone project. The project has the following structure (see 2). The project starts with topic selection. In topic selection, the instructors work with the students to develop a viable plan for the entire program that the instructors can help but the students also show passion for. Our team proposes the three requirements for topic selections: (1) First, the topic needs to have certain social impact and serve a wide range of audience. We do not recommend to have students at pre-college level to focus on a narrow field even if the field has great potential. In the beginning of the learning curve, particular examples can be used but the purpose is to lead students into the field first. (2) Next, the topic needs to pertain certain skill levels that the students might struggle with. The nature of data science is not a simple calculus where the same answer can be achieved for the same setup. It is important to show the students at pre-college level that data science is more of an art than science where caveats and "accidents" can happen. (3) Last, the project needs to be at a level where the students can feasibly finish before certain deadlines. Most pre-college level students are pursuing the study of AI at their leisure time, most of which are using summer period to their AI programs. It is imperative that the instructor team respects this deadline and provides resources for the students to finish before the agreed upon deadlines.

Next, the instructor works with the students to develop a solid game plan to explore the data with the assistance of Machine Learning (ML) or Artificial Intelligence (AI). This steps requires the experiment design to be data-centric (see Figure 2 for global guideline for capstone project and Figure 2 for the data-centric pipeline). The major component of the capstone project is the data-centric pipeline. The pipeline suggests that the instructors work with the students to establish a client-consultant relationship. The student is the client that shows certain curiosity to a field of a problems. The instructor acts as a consultant to share with the client (the student) a list of tools and their feasible application approach towards the proposed ideas. It is much more impactful when the idea comes from the student. This is the most optimal mindset to work with not just for the instructors but also from the students.

Last and most importantly, the proposed course materials lands on a portfolio of deliverable. This paper proposes to have the deliverable structured according to Figure 1. First, there is a location of data and model storage folder. For example, a folder on Google Drive, Microsoft OneDrive, or Amazon AWS S3 Bucket are ideal locations to share files and models. The coding component is carried out using an Integrated Development Environment (IDE) that is shared between the instructors and the students. We have found the Google Colaboratory (also known as Colab) to be a very successful platform and it can also be shared with anyone that has a Gmail account.

Figure 1: **Deliverable**. The workflow structure, database management, and deliverable. The course (lectures and coding work-through) require data. The data is saved on a shared location such as a Google Drive folder. The coding component, just like the data, is an open-source IDE that can also be shared between the instructor and the students. Finally, models ready for deployment and showcase are written as a package and released on Github as a new repository.



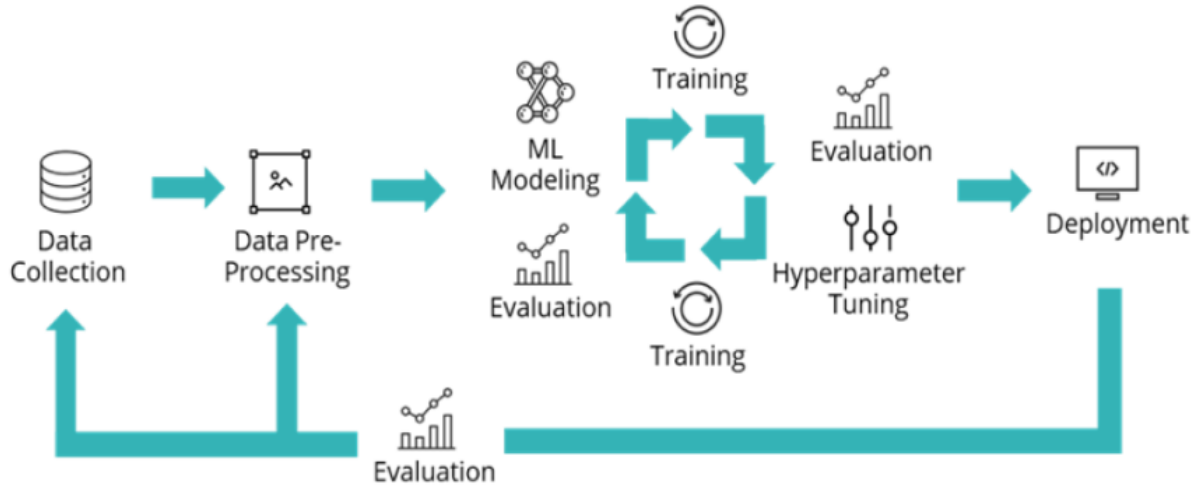
Lecture Subjects

This subsection discusses the lecture subjects and contents. Our team proposes to start with the first chapter to cover the basic statistical learning techniques. The course starts with a small brief of the history including the origins of regression problems and the machine learning hierarchy such as the difference between supervised and unsupervised techniques.

Next, to establish the ground for students to understand loss functions and neural network models from scratch, it is important to set up the tone mathematically and to provide students a list of notations easy to work with. The proposed mathematical language is as the follows. We can describe a data set as $\mathbf{X}$ with n rows (instances or samples) and p columns (features). To compensate the abstract level, Python programming can be a complimentary tool to create a Pandas data frame (a well-known Python library to handle data frames). The explanatory features can be a data frame like this of which we denote as X while the target features is a similar data frame with one column of which we denote as Y. There is an innate relationship between X and Y that exists in nature called $f(\cdot)$. This relationship is what ML or AI scientists aim to model. For the modeled or learned relationship, a hat notation is used $\hat{f}(\cdot)$. Hence, with this set up, an educated guess of $\hat{Y}$ can be found using the learned model $\hat{f}(\cdot)$ and the data (observable) X, i.e. $\hat{y} = \hat{f}(X)$. Since $\hat{Y}$ is the educated or learned estimates, we need to compare this with the ground truth to see how many errors the model $\hat{f}(\cdot)$ is making. Hence, it is desirable to understand a proper loss function, i.e. denote as $\mathcal{L}(\hat{Y}, Y)$. Once the chapter sets up the tone, this statistical learning environment is reinforced whenever a new model is introduced.

Figure 2: **Executive diagram** of proposed capstone project milestone. The proposed capstone milestone project guideline follows data-centric approach. The project starts with topic selection. Topic selection requires instructors to work with the students to develop feasible and optimal plan tailored to the students' needs. The topic selection has three requirements. First, the topic needs to serve a broad range of audience and delivers certain level of social impact. Next, the topic needs to pertain certain level of technical or coding skill sets. Last, the students should have sufficient time to finish the topic on time. Then the students move towards experiment design to explore the data using a variety of different statistical techniques. The major component of the project is constructed using a novel data-centric AI approach (see Figure 3). The project lands on a scholarly writing paper.

Figure 3: **Executive diagram** of proposed training cycle. The proposed training cycle focuses on data-centric AI and each module sees a similar adaption of the proposed model. The proposed pipeline focuses on the value and the needs of the data. The pipeline starts with data collection and data processing. The out-of-the-box thinking process starts with ML modeling and certain evaluation metrics. This is also known as thinking with AI. The intuition built with the assistance of AI can deliver far more values than what is present on the surface from the data. The internal cycle of the proposed diagram suggests to proceed to training of the ML/AI models with certain hyper-parameter tuning. In addition, the instructors need to work with the students to develop practical evaluation strategies to measure the deploy-ability of the model. In the end, the instructors will work with the students to deploy a model in an install-able form as a new software.



Live Coding Walk-through

Every lecture is recommended to have complimentary coding walk-through. The sessions for live coding walk-through can be done in a in-person environment using projectors or using online conference platforms such as Zoom, WebEx, Microsoft Teams, and Google Meet. The code resource is provided in this Github repository [16]. More details about navigating the Github repository is discussed in next section. The benefits of the live coding session is to solidify the lecture contexts in a practical approach. In addition to understand the features X and the target variables Y with certain loss functions $\mathcal{L}(\hat{Y}, Y)$ where $\hat{Y}$ is the estimates from the model $\hat{f}(\cdot)$, it is important to show the students using code that there is a difference between the prediction $\hat{Y}$ and the ground truth Y. From lecture, loss functions such as mean square error (MSE) can be proposed and the MSE takes the following form

$$\text{MSE}(Y, \hat{Y}) = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{1}$$

For students at pre-college level, many of them could be using Python programming languages at the first time. In lecture, notations such as the running index i is introduced and the summation $\sum$ is taught from AP Calculus AB/BC. However, the execution of this function, despite of its mathematical complexity, might actually not require hard coding. It is recommended to show students machine learning librarires such as Sci-kit Learn packages. The equation 1 can be easily implemented using this library. A sample code is provided below.

```
from sklearn.metrics import mean_squared_error
y_true = [3, -0.5, 2, 7]
y_pred = [2.5, 0.0, 2, 8]
mean_squared_error(y_true, y_pred)
```

For intermediate level or advanced level, it is optional to introduce coding from scratch which can be demonstrated by instructors in a live session. A sample code is provided below to demonstrate the difference.

```
import numpy as np
from sklearn.metrics import mean_squared_error
y_true = [3, -0.5, 2, 7]
y_pred = [2.5, 0.0, 2, 8]

# use a function
mean_squared_error(y_true, y_pred)

# from scratch
np.square(np.subtract(y_true, y_pred)).mean()
```

Scenarios could arise where the questions could be raised "what does the *np.subtract* function do". Questions like these are invaluable for the class, because this creates a permanent memory mark in the students head that the chance that this is ever forgotten is highly unlikely. As an opposite, if a lecture were given to the students to discuss *np.subtract* function inside of a whole dictionary of functions, there is no memory mark permanently because the knowledge is forced upon the students whereas the proposed method the knowledge is raised from the students. This tutorial style is against the common sense of the conventional classroom teaching.

Resources

This section provides a whole portfolio of resources for the conference and many of its audience.

First, the main Github repository is created for pre-college level students at a small (10-20 students) classroom size or at 1-to-1 session. This Github repository is here. In regards to the summer course, we propose the course materials to be divided into different modules. We release the 2022 Summer modules and also the 2021 Summer modules. We also provide complimentary code repositories on Google Colab for machine learning topics, called Fundamentals of Machine Learning. In addition, a specialized deep learning portfolio is created and released here.

Together, the family directory is summarized below for convenience.

Table 1: **Family Directory**. This is the family directory of proposed materials for the audience of AAAI track.

Master Portfolio:	
Proposed Summer Program	Lecture, Git
Fundamentals of Machine Learning	lecture, Code, Git
Fundamentals of Deep Learning	Lecture, Code, Git
Capstone Projects	Syllabus, 2022

Acknowledgments

The authors would like to thank the Veritas AI Education platform. Specifically, the author would like to thank Stephen Turban, a Harvard University graduate who is a pioneer in AI4ALL and pre-college AI education, for the support from him and and his team.

References

[1] R. Aliabadi. Readyai. *http://readyai.org/*, -2022.

[2] E. Allensworth and J. Q. Easton. The on-track indicator as a predictor of high school graduation, 2005.

[3] D. Bass. Researchers combat gender and racial bias in artificial intelligence. bloomberg 1–9, 2017.

[4] J. J. Beard, J. Hsu, M. Ewing, and K. E. Godfrey. Studying the relationships between the number of aps, ap performance, and college outcomes. *Educational Measurement Issues and Practice*, 38(4):42–54, 2019.

[5] K. L. Best and J. F. Pane. *Privacy and interoperability challenges could limit the benefits of education technology*. JSTOR, 2018.

[6] E. Brynjolfsson and T. Mitchell. What can machine learning do? workforce implications. *Science*, 358(6370):1530–1534, 2017.

[7] CollegeBoard. Collegeboard. *https://reports.collegeboard.org/ap-program-results/data-archive*, 2020.

[8] B. Cowgill and C. Tucker. Algorithmic bias: A counterfactual perspective. *NSF Trustworthy Algorithms*, 2017.

[9] D. S. David S. Touretzky. Calypso. *Visionary Machines LLC*, -2022.

[10] S. Druga. Cognimates. *http://cognimates.me/home/*, -2022.

[11] S. M. Jayaprakash, E. W. Moody, E. J. Lauría, J. R. Regan, and J. D. Baron. Early alert of academically at-risk students: An open source analytics initiative. *Journal of Learning Analytics*, 1(1):6–47, 2014.

[12] T. D. McFarland and R. Parker. *Expert systems in education and training*. Educational Technology, 1990.

[13] R. F. Murphy. Artificial intelligence applications to support k-12 teachers and teaching. *Rand Corporation*, 10, 2019.

[14] I. T. Sanusi and S. S. Oyelere. Pedagogies of machine learning in k-12 context. In *2020 IEEE Frontiers in Education Conference (FIE)*, pages 1–8. IEEE, 2020.

[15] B. C. Stahl and D. Wright. Ethics and privacy in ai and big data: Implementing responsible research and innovation. *IEEE Security & Privacy*, 16(3):26–33, 2018.

[16] Y. Yin. Introduction to machine learning, big data, and applications. *https://github.com/yiqiao-yin/Introduction-to-Machine-Learning-Big-Data-and-Application*, 2022.