

# Using Cloud Tools for Literate Programming to Redesign an AI Course for Non-Traditional College Students

**Maria Hwang, Calvin Williamson**  
{maria\_hwang, calvin\_williamson}@fitnyc.edu  
Fashion Institute of Technology

## Abstract

As more open source educational software applications become available, higher educational institutions have the opportunity to utilize these cost efficient tools to deliver the instruction traditionally taught off line with heavy associated costs. Here we introduce a machine learning course that uses a simple, cloud computing approach to creating course materials. We see this type of serverless, cloud-based, literate programming to be the future of computer science education in non-traditional higher educational institutions in particular serving students who will need the basic literacy for computing and computation but will not pursue the traditional computer scientist path.

## Introduction

With the exploding abundance of available data, artificial intelligence (AI) and machine learning (ML) have taken a new life on its own throughout every possible industry imaginable from genetics to fashion (Dashtban and Balafar 2017; Liu et al. 2019). Not surprisingly, higher educational institutions are also grappling with the inevitable resurgence of AI/ML (Eaton et al. 2018; Iles and Erturk 2017; Wollowski et al. 2016). We report on our own experience with an introductory AI course for non-CS majors. We have redesigned this course around cloud-based *literate programming* to make the course more accessible to students with limited prior computing experience, as well as non-traditional students, such as adult learners and commuting students.

## Is Artificial Intelligence for Everybody?

Because applications of AI/ML are permeating every field, careers that are traditionally non-computationally oriented increasingly require the ability to interact and communicate with data scientists and developers. Therefore, institutions of higher education are encouraging (and occasionally mandating) that even non-CS track students are prepared for this. Students must have basic CS literacy (e.g., programming and data analytics literacy) to be able to efficiently communicate without difficulty with the more technically-oriented workforce. Despite the abundance of online materials available to all, higher educational institutions, and particularly

those non-traditional ones producing workforce that may not be technically-oriented, must adapt to the changing world and incorporate some fundamental CS courses to teach programming literacy. As a trade-school and community college, producing mostly non-computationally oriented students, Fashion Institute of Technology has been dealing with this global demand, in spite of lacking support from the university to provide such programming literacy courses. Based on our experience, we introduce a course design that can be applied in similar non-traditional higher educational institutions. We share our experience on how we have been improving our curriculum and reflect on future challenges facing this sector of AI education.

## AI for Education

In the last four years at Fashion Institute of Technology (FIT), for students those who need statistics requirements (there is no CS major at FIT), we have provided a permutation of a statistics course incorporating ML techniques using the programming language R. In the old design of this course, students would download R on their laptops or on a USB jumpdrive to use on campus computers (a work around the IT Department's restriction of installing software on campus computers). In the R console, students would start learning to code (who typically have never coded before) based on instructors' demonstrations, a series of YouTube tutorials created by one of our own instructors (`\textit{https://www.youtube.com/user/rdjalayer/featured}`), and any other resource they find online. Topics that are typically covered include exploratory data analysis, regression, decision trees (classification and regression), kNNs, and, text mining. However, due to the limitation of the R console interface, often codes are not saved properly, codes are hard to share in real time, and students are less likely to collaborate and share their experiences debugging, arguably the core skill for becoming a competent programmer and problem solver. As educators, we have been struggling to find a solution to address some of the limitations with the R console interface as well as the institutional infrastructure at our university. The following introduces our solution to these problems by leaving the previous generation of installing any sort of software application and moving toward the truly cloud computing and user friendly, literate programming interfaces.

## RMarkdown to Colab and Jupyter Notebooks

In support of an introductory machine learning course, the authors have created utilities for converting RMarkdown (Rmd) (Baumer et al. 2014) to several course-friendly formats. These include Google Colab and Jupyter Notebooks as well as Google Docs (The source repo `\textit{https://gitlab.com/calvinw/rmarkdown-colab-google-docs}` and accompanying pages site `\textit{https://calvinw.gitlab.io/rmarkdown-colab-google-docs}` show several examples of the different formats). Rmd files containing R or Python code are rendered to each of the following formats: html, pdf, md, docx, Colab Notebooks (ipynb), and Binder Jupyter Notebooks (ipynb).

The rendering process uses R's knitr and Rmd packages to convert Rmd to html, pdf, md, docx, and ipynb formats. The notebook versions (ipynb) are created by using Pandoc. Pandoc allows converting from markdown (md) to ipynb. The correct jupyter kernel (R or Python) is specified in each Rmd file header. When Google Colab uploads the resulting ipynb notebook file, it creates a Colab Notebook using the appropriate kernel. We include a Makefile that automates the creation of all the outputs and runs the Pandoc conversions. Finally, we have added tools that automatically convert the ipynb versions to Google Colab Notebooks stored in Google Drive or Binder Jupyter Notebooks runnable in the cloud. Therefore, creating complete sets of content for R and Python programming classes including notebooks and lectures uploaded to Google Drive can all be automated from a single Rmd format with an accompanying build process.

## Benefits of Online Notebooks

Using the approach above it is possible to run a fully cloud-based programming class with all the benefits of sharing and collaboration. Lectures and assignments are delivered as Google Colab Notebooks, all stored in Google Drive for easy access. When an instructor posts these in her/his course management system, students are able to simply click on a link that opens to the assignment document. They “make a copy...” of the instructor’s notebook and immediately share it with him/her to turn it in. No dropbox is needed in this kind of setup, and no code is submitted as text files. To grade assignments an instructor can click on the notebook the student shared with him/her, restart the kernel, and run all the cells. Additionally, the instructor can place comments or suggestions from the time the student starts the assignment, affording real-time feedback and interaction. Every document in the course is fully shareable and accessible from any computer or device, allowing opportunities for distant learners as well as any students with physical commuting restrictions. Last but not least, instructors can easily share the course documents with other instructors and collaborators through the sharing feature and not have to deal with internal approval processes to share course shells as it is done in Blackboard in Fashion Institute of Technology and other institutions that use similar Learning Management Systems.

## Conclusion and Future Work

Our traditional statistics course has evolved throughout the last four years and continues to do so. Our latest cloud based approach has sparked our interest in sharing this with the computer science higher education community. Because no software installation or server infrastructure is needed beyond a Google login and computation is all done on Google Colab servers, there is no support needed from IT, which often hinders widespread technology-integrated curriculum adoption. Therefore, the environment scales to large groups of students or multiple sections easily. Besides providing students a familiar document environment for their work, notebook style lectures and assignments encourage students to practice and utilize *literate programming* in the form of active note-taking and course documentation creation. We share our resources with the AI community—as others have done: (Neller et al. 2019; Sintov et al. 2016; Wollowski et al. 2016)—and will continue to report on our trials and errors with the serverless, cloud-based, machine learning course at Fashion Institute of Technology serving non-traditional students with little to no technical backgrounds. We hope to see more discussion surrounding such curriculum adoption within higher education and encourage collaboration among institutions.

## References

- Baumer, B.; Cetinkaya-Rundel, M.; Bray, A.; Loi, L.; and Horton, N. J. 2014. R markdown: Integrating a reproducible analysis tool into introductory statistics. *arXiv preprint arXiv:1402.1894*.
- Dashtban, M., and Balafar, M. 2017. Gene selection for microarray cancer classification using a new evolutionary method employing artificial intelligence concepts. *Genomics* 109(2):91–107.
- Eaton, E.; Koenig, S.; Schulz, C.; Maurelli, F.; Lee, J.; Eckroth, J.; Crowley, M.; Freedman, R. G.; Cardona-Rivera, R. E.; Machado, T.; et al. 2018. Blue sky ideas in artificial intelligence education from the eaii 2017 new and future ai educator program. *AI Matters* 3(4):23–31.
- Iles, R., and Erturk, E. 2017. What is the vision for ai education resources? how well is it currently being met?
- Liu, L.; Zhang, H.; Ji, Y.; and Wu, Q. J. 2019. Toward ai fashion design: An attribute-gan model for clothing match. *Neurocomputing* 341:156–167.
- Neller, T. W.; Sooriamurthi, R.; Guershoy, M.; Zhang, L.; Talaga, P.; Archibald, C.; Summerville, A.; Osborn, J.; Resnick, C.; Oliver, A.; et al. 2019. Model ai assignments 2019. In *Proceedings of the AAAI*, volume 33, 9751–9753.
- Sintov, N.; Kar, D.; Nguyen, T.; Fang, F.; Hoffman, K.; Lyet, A.; and Tambe, M. 2016. From the lab to the classroom and beyond: extending a game-based research platform for teaching ai to diverse audiences. In *Thirtieth AAAI Conference on Artificial Intelligence*.
- Wollowski, M.; Selkowitz, R.; Brown, L. E.; Goel, A.; Luger, G.; Marshall, J.; Neel, A.; Neller, T.; and Norvig, P. 2016. A survey of current practice and teaching of ai. In *Thirtieth AAAI Conference on Artificial Intelligence*.