Lessons Learned from Teaching Machine Learning and Natural Language Processing to High School Students

Narges Norouzi, Snigdha Chaturvedi, Matthew Rutledge
University of California, Santa Cruz
{nanorouz, snigdha, mrutledg}@ucsc.edu

Abstract

This paper describes an experience in teaching Machine Learning (ML) and Natural Language Processing (NLP) to a group of high school students over an intense one-month period. In this work, we provide an outline of an AI course curriculum we designed for high school students and then evaluate its effectiveness by analyzing student's feedback and student outcomes. After closely observing students, evaluating their responses to our surveys, and analyzing their contribution to the course project, we identified some possible impediments in teaching AI to high school students and propose some measures to avoid them. These measures include employing a combination of objectivist and constructivist pedagogies, reviewing/introducing basic programming concepts at the beginning of the course, and addressing gender discrepancies throughout the course.

Keywords: Machine Learning, Natural Language Processing, Artificial Intelligence, Summer Course, High School Students

1 Introduction

Throughout the past decade, K-12 Education has evolved to equip students with computer skills, especially programming and topics in data science. Patricia Levesque, CEO of ExcelinEd noted, "States are recognizing that computer skills have become as fundamental to student success as reading, writing, and math. If we are not teaching our children computer science, we are not preparing them to participate in a world of rapidly changing technology and advancement." In the United States, during the first six months of 2019, 33 states have passed legislation and funded 40.1 million dollars to support K-12 computer science instruction (Code.org 2019). With the increase in general computer science instruction starting in elementary schools, in the coming decade, there will be opportunities for more advanced computer science instruction in middle and high schools. Wollowski et al. surveyed practitioners of AI about "up-and-coming AI topics". The most common response was 'machine learning' at 47%, while 'natural language processing' was the fourth most common response at 17% (Wollowski et al. 2016). Therefore, the authors of this work decided to explore instructing machine learning and natural language processing to high school students at the University of California, Santa Cruz.

The University of California, Santa Cruz hosts the COSMOS Summer School for high school students each summer quarter. In Summer 2019, following in the pattern of increased focus on computer science in K-12 education, a new cluster of "Machine Learning and Natural Language Processing" has been added to the program. Even in its first year, the ML/NLP cluster received more than twice as many applications as the other existing nine clusters. Since this was a new cluster, the instructors developed a new curriculum from scratch. Both instructors have prior experience in teaching a variety of machine learning, artificial intelligence, and natural language processing courses at the undergraduate and graduate levels and developed the materials with these experiences in mind.

Traditionally, courses have employed an objectivist curriculum. An objectivist curriculum is completely governed by the instructor. The instructor selects well-defined learning goals from the domain of the class. The explicit learning goals are then taught in a planned sequence. The assessment is given at the end of the course in the form of a final exam. On the other hand, in a purely constructivist curriculum, students control the learning sequence. In the flexible learning environment students influence, or construct, the direction and goals of the learning. Instead of a single final exam, the evaluation or feedback is embedded throughout different learning tasks. Studies on first-year undergraduate students revealed that these students are not quite prepared for a completely constructivist curriculum (Pantic, Zwitserloot, and Grootjans 2005). In order to adapt to the unique needs of high school students, the instructors chose to use a combination of objectivist and constructivist pedagogies. The course started out with more objectivist style lectures in order to build a solid technical foundation and slowly transitioned into a more constructivist group project. The goal of developing this pedagogy was to engage high school students in solving real-world AI problems while retaining intellectual rigor.

While designing the course, instructors incorporated the ethical concerns about AI raised by (Burton et al. 2017). In the paper, Ethical Considerations in Artificial Intelligence Courses, Burton noted that AI systems can be difficult to validate, predict, and explain. These difficulties often arise since AI systems reason in different ways from humans. In our course design, we found it to be helpful to discuss how AI systems work within the context of concrete examples. We also framed the ethical discussions around AI systems within the context of our guest lecturers' research talks. One guest lecturer presented on the ethics of using ML without being able to interpret the results within the 'black box' of Deep Learning (DL). Students engaged in a discussion about potential problems such as how machine learning can inherit biases from its input data. Within these concrete examples students engaged in a lively discussion about the ethics of artificial intelligence. The detailed description of the course content that we specifically designed to target high school students is presented in Section 2.

In addition to designing a condense curriculum for high school students, we also surveyed students in the beginning and at the end of the session to evaluate the effectiveness of the curriculum in teaching ML and NLP as well as building confidence in students to develop ML/NLP models. Section 3 outlines the questions in the entry and exit surveys and Section 4 describes the lessons we learned after analyzing surveys and observing students' performance throughout the course and their contribution to the group project.

Our contributions in this paper are as follows:

- We provide an outline of an AI course curriculum for high school students.
- We evaluate the effectiveness of the curriculum by analyzing the feedback we received from students.
- We identify possible impediments in teaching AI to high school students and propose some measures to avoid them.
- We examine the possibility of completing an AI project by high school students and the appropriate team structure/dynamics to accomplish the task.

2 Course Structure and Outline

The ML/NLP cluster for high school students was designed as an intense one-month course with 20 instruction hours per week. Throughout the program, students were seated in pairs to provide an additional layer of support. For the first week of the course, we randomly assigned new partners at the beginning of each lecture to help prepare students for the collaboration necessary for the final group project. In addition, to help facilitate cooperative work before the lectures began the instructor would lead the class in a non-academic ice breaker activity. For example, the instructor would prompt the class, "Turn to your partner and tell them about the funniest moment in your life." By first developing cooperative skills in a non-academic context all students start on a level playing field without the pressures of making an academic mistake. This non-academic practice helped students learn the cooperative structures, which they can then employ in academic contexts (Kagan 2014). During academic discussions, instructors would employ the cooperative learning structure of Think-Pair-Share. Think-Pair-Share is when students first have a minute to think about the initial question, then a minute to discuss with their partner, and finally a chance to share with the class as a whole. Since Think-Pair-Share allows students time to compose their ideas in their own head and discuss them with their partners, the responses received are often more intellectually involved and concise (Azlina 2010).

We began the course by teaching students the required programming background for them to be able to understand and implement simple ML and NLP algorithms. We chose Colab to teach programming in Python because of the interactive nature of the platform (Google 2018). The interactive nature of Colab allowed students the flexibility of learning at their own pace and to extend their learning by playing with variants of the code. After students were comfortable with the basics of programming, we began lectures on ML and NLP topics. Once the students had a solid foundation of ML and NLP knowledge, instruction transitioned away from objectivist strategies to constructivist strategies. Towards the second half of the course, students were given the chance to work on a course project based on cleaned datasets. Once teams were assigned a topic they were provided with a dataset, and told to get started without any further explicit instruction. Students were given the end goal of the group project: to design ML and NLP algorithms to a specified test accuracy, design a 20-minute slide presentation, and design a conference poster. But students had to collaborate in their groups to figure out how to successfully achieve these end goals. During the group work phase of the class, professors and teaching assistants circulated around the class to answer any questions about coding algorithms or presentation formatting, however all the questions were student-driven. During the last week of the course, guest speakers from different sub-disciplines of ML and NLP had been invited to give talks on their cutting-edge research. We outline the detailed weekly schedule of the course in Tables 1, 2, 3 and 4.

Table 1: Week 1

Welcome - Sharing the schedule of the session, Entry survey, Sharing course materials	Types, Variables, Expressions, Operators, Order of operations
Conditional control flow and loops, Functions, Namespaces, and Scope	Break and Continue keywords, Coding practices
Strings	List and Tuples, Linear and Binary search algorithms
Writing and reading to/from a file, Modules	Dictionaries
Data wrangling and visualization	Data Analytics example

Table 2: Week 2

Introduction to NLP	Introduction to Classification, Linear regression with an example of Wisconsin breast cancer dataset (Wolberg 1992)
Introduction to ML, Training and test sets, Labels, Evaluation metrics, Overfitting/Underfitting, Data cleaning, Training-testing-evaluation phases	NLP preprocessing
Logistic regression	N-grams and related exercises
Neural Networks	Language models
Intro to DL and some variations of it	Assigning projects and forming groups

Table 3: Week 3

Overview of NLP problems	Guest lecture 1: Machine Learning from Natural Language
Stemming and TF-IDF	Work on projects
Work on projects	Guest lecture 2: Language Models and Text Generation
Work on projects	Presenting results to the class
Designing posters	Preparing oral presentations

Table 4: Week

Field trip to Microsoft office	
Guest lecture 3: Graphs and ML	Guest lecture 4: Fairness in ML
Guest lecture 5: Energy data analytics	Guest lecture 6: Language Model and review generation
Guest lecture 7: Ethics in AI	Preparing presentations
Oral presentation day	
Poster presentation day + closing ceremony	

3 Student Feedback

We conducted two major surveys to collect student feedback. We gave the entry survey during the first week of the course and the exit survey on the last day of instruction. In the entry survey, the goal was to identify students' understanding of AI and their prior experience in programming. In the exit survey, we repeated most of the questions to see how their perception of machine learning and natural language processing had evolved and if the course improved their confidence in programming. Another section in the entry and exit surveys inquired about students' concerns coming into the course and if they found those as obstacles throughout the course. The following section outlines the questions in the entry and exit surveys.

3.1 Outline of the Entry and Exit Surveys

Questions in Entry survey

1. Who is a Computer Scientist? What does a Computer Scientist do? Describe in one paragraph.
2. What do you understand by the term 'AI'?
3. How is an AI expert different from a Computer Scientist (if at all)?
4. Give examples of AI projects that you know of. Can you illustrate one or some of the projects using figures? Extra blank pages are provided for your illustrations.
5. In your opinion, what kind of things/tasks will AI be used for in the future? Give examples along with timelines (in how many years do you think these applications will be realized).
6. Why are you taking this course? What excites you the most about this course?
7. What do you hope to achieve/learn by the end of this course?
8. What are your biggest concerns regarding this course? Think of the challenges that you expect to face during this course based on your past experiences. Your concerns could be academic or non-academic.
9. Who do you prefer as a project partner? What are some of their (personal/academic) characteristics that make them suitable as a partner?
10. On a scale of 1 to 5 rate your programming confidence.
11. If you have a programming background, which language(s) are you comfortable with?
12. On a scale of 1 to 5 rate your knowledge about AI.
13. What are some of the courses and school activities that helped you get prepared for joining this cluster and understanding AI?

Questions in Exit survey

1. Who is a Computer Scientist? What does a Computer Scientist do? Describe in one paragraph.
2. What do you understand by the term 'AI'?
3. How is an AI expert different from a Computer Scientist (if at all)?
4. Give examples of AI projects that you know of. Can you illustrate one or some of the projects using figures? Extra blank pages are provided for your illustrations.
5. In your opinion, what kind of things/tasks will AI be used for in the future? Give examples along with timelines (in how many years do you think these applications will be realized).
6. Did you find this course useful and/or exciting? Why or why not?
7. Looking at what you expected to learn in this course, do you think you were able to achieve that goal?
8. Looking at the challenges you were concerned about at the beginning of the course, do you think they affected your performance in the course?
9. If you were to do an ML/NLP project again, who do you prefer as a project partner? What are some of their (personal/academic) characteristics that make them suitable as a partner?

10. On a scale of 1 to 5 rate your programming confidence.
11. On a scale of 1 to 5 rate your knowledge about AI.
12. On a scale of 1 to 5 rate your knowledge about ML.
13. On a scale of 1 to 5 rate your knowledge about NLP.

3.2 Students' Preferences for the Course Project

One of the key constructivist components of the course was to complete a group project on a real-life ML or NLP topic. We allocated about seven days in the course schedule for students to work in teams of three to four students. In order to assign students to different teams and projects, we conducted a survey with the following questions. The goal of the survey was to assign students to the topic they were interested in and make sure that there was a balance of expertise in all teams.

1. What qualities do you think your teammate should have to complement you in the project?
2. On a scale of 1 to 5, how comfortable are you with programming?
3. On a scale of 1 to 5, how comfortable are you with algorithms?
4. On a scale of 1 to 5, how comfortable are you with Machine Learning?
5. On a scale of 1 to 5, how comfortable are you with Natural Language Processing?

In Section 4.6, we described how we used the responses to this survey to assign students to different teams, and then allocate a project topic to each of the teams. Once teams were assigned a topic they were provided with a cleaned data set, and told to get started without any further explicit instruction.

4 Lessons Learned

Throughout the one-month long course we closely observed students, their responses to our entry and exit surveys, their contribution to the team project, and their evaluation of guest speakers' talks. In this section, we share some of the lessons we learned which will help those interested in designing or improving curriculum for teaching AI, ML, NLP, and related topics for high school students.

4.1 Did students demonstrate a more in-depth understanding of AI or what it means to be a computer scientist after taking the course?

While the responses students provided about an example of an AI system were similar in both the exit and entry surveys, suggesting examples of AI systems from movies or news, some of them expanded on their definition of AI in the exit survey. For example, one of the students wrote: "AI is an assortment of code and mathematical principles that can 'learn' patterns from data." Another student, similar to the behaviorists' definition of AI, described an AI system as a system imitating human behavior in the entry survey. In the exit survey, however, they explained that "Artificial intelligence is a machine's ability to process large amounts of data and essentially learn its patterns and trends." In the entry survey students described the general concept of AI, but after taking the course students wrote more detailed definitions of AI, including learning patterns from data.

4.2 Did students become more realistic and explicit with the "kinds of things/tasks AI will be used for in the future"?

In the entry survey, many students mentioned robots, Alexa, or self-driving cars, but then they became more explicit in the exit survey. This suggests that their prior understanding of AI was no more sophisticated than what they must have heard from movies and news. Some of these examples include:

- In the entry survey, one student described robotic functions, but in the exit survey, they mentioned chatbots, energy efficiency, and story generation. Chatbots and story generation were covered in a guest lecture on Natural Language Generation. And energy efficiency was covered in a guest lecture on Energy Grids.
- Similarly, another student expanded their examples in the exit survey with the prediction of criminals, and hip fracture detection. The prediction of a criminal's recidivism is taken from a guest speaker's talk and hip fracture detection is taken from a fieldtrip to a Microsoft office.
- One student expanded their example with their group projects (sentiment analysis using binary classification)

Our analysis indicates that after working on different ML and NLP models throughout the course and hearing about applications of AI in guest speakers' talk, students became aware of more contemporary and real-world uses of AI systems. If logistically possible, guest lectures and fieldtrips make a lasting impact on students.

4.3 Did the student's motivation for taking the course in the entry survey line up with the materials covered in the course? Did the students feel like they achieved their goal for the course?

In responses to the entry survey, most students were excited about working on projects. They were also excited to learn about ML and NLP since they recognized it would will be a huge part of future society. One student predicted, "[Machine Learning] will be a large part of modern life." Students were also excited to learn about how machines learn. A student said, "So I would say the most exciting thing would be finding out more about the process behind letting computers develop intelligence." Another said, "I am most excited to learn how computers use patterns and statistics to learn independently."

In the exit survey, most of the students believed the course provided a solid foundation of theory and vocabulary for future studies. 71% of students mentioned that they met their goal they had at the beginning of the course and 29% said they somewhat met their goal. It is worth noting that the main reason why some of them said they have not fully met their goal was their confidence in independently coding ML

Table 5: Prerequisite recommendation from students based on gender

Prerequisite recommendation from students based on gender Total of 20 student responders, with some multiple responses	Coding Prerequisite	Math Prerequisite	No Prerequisites
Female	6	1	0
Male	10	3	3

algorithms. One student wrote, "This course is the starting point for me, as I still have a long way to go with learning how to code."

These responses indicate that the goals of the course design were successfully met. The course curriculum helped students in getting a good understanding of basic methods in the broad field AI and at the same time obtaining a more realistic birds-eye view of the field. The responses also indicated that students became more aware of the depth of ML and NLP areas and they were excited about learning more about them in the future.

4.4 Do all students recommend some coding or mathematics prerequisites for the course? Were students without proper coding or mathematics background able to successfully complete the course?

Table 5 shows the prerequisite recommendations based on 20 student responses. The majority of students from both genders thought that a coding prerequisite should be required for taking the course. This suggests that having programming lectures during the first week was beneficial in raising the programming skill of all students to the required standard of the course. It is also worth noting that all female students reported that coding and/or mathematics background should be required for the course. On the other hand, 20% of male students mentioned that no prerequisite is necessary for taking the course. This difference between genders and how to address it during instruction is further discussed in Section 4.7.

4.5 How many students were worried about their technical and coding abilities? Did these worries end up affecting their performance in the course?

As a part of the entry survey, the students rated their programming skills on a scale of 1 to 5. As outlined in Table 6, 66% of female students were not confident with their programming skills whereas only 35% of male students were not confident. This suggested that there needed to be a careful assignment of students to teams in order to help all students gain programming expertise in coding ML/NLP algorithms, and at the same time complement each other's capabilities to complete the project as a team. We implemented

Table 6: Programming confidence of students based on gender

Coding Confidence	Not Confident	Confident
Female	66%	34%
Male	35%	65%

some measures on how to assign students to different teams, which will be discussed in Section 3.2.

4.6 Which characteristics (coding/technical ability, strong work ethic, etc.) did students identify as being preferable in a project partner?

Table 7 shows the preferred characteristics of a project partner based on 19 student responses in the entry survey. Some of the students listed multiple characteristics. We found no differences in expectations from project partners by male and female students. Both female and male students seemed to value strong work ethics and collaboration and communication skill of a project partner over their coding ability.

Since there were no specific differences in expectations, we conducted another survey (see Section 3.2) where we asked students for their preferences for project topics out of 6 pre-defined topics. We also asked them to rate their ability and confidence in programming, implementing algorithms, and topics of ML and NLP. The project topics were as follows:

- ML project: Ames Housing Price Prediction (Kaggle.com 2018a)
- ML project: Dogs vs. Cats Classification (Kaggle.com 2014)
- ML project: Heart Disease Classification (Kaggle.com 2018c)
- ML project: Malaria Parasite Detection (Kaggle.com 2019b)
- NLP project: Amazon Fine Food Review Analysis (Kaggle.com 2019a)
- NLP project: Tweet Classification as Relevant or Irrelevant to Disasters (Kaggle.com 2018b)

These topics were chosen to provide students with a good understanding of real-world applications of AI algorithms. However, the instructional staff 'solved' the projects beforehand to ensure that they were neither too difficult nor too easy for the students and that there were enough challenges to learn interesting practical aspects of AI applications.

As students became more aware of their actual technical capabilities, their preferred characteristics of a project partner shifted. Most of the students' concerns were about their programming skills. One student mentioned, "Someone who is willing to help a beginner and someone with a good amount of programming experience. I think I have a general understanding of how to solve a problem, I just need help with translating that to code." Another student said, "I most likely need someone to help me with programming because I am very new to Python. I am willing to learn, I just

Table 7: Preferred characteristics of a team member based on gender

Total of 19 student responders, with some multiple responses	Coding Ability	Strong Work Ethics	Collaboration or Communication Skill
Female	1	6	7
Male	3	9	9

need more practice/experience so I might be a little slow." To address these concerns, we divided up students with a prior programming background in different teams and then allocated a project topic to each team based on the majority preference of the team members. Based on the results we observed at the end of the course projects, all teams were able to complete the assigned task and the majority of students felt like they were able to accomplish the goal they had for the course as described in Section 4.3.

4.7 Effect of gender on student role in course projects

The composition of the groups was not chosen by the students and was determined by the course staff according to their preferences for teammates, their self-reported skills, and the project topics they were interested in. Since we generally observed less confidence in programming and understanding of AI in female students compared to their male peers in the entry survey, care was taken to not have a team with only one female student. In this section, we analyze if gender had any effect on the roles that students played in their project teams.

We realized that while working on the projects and designing presentations and posters, female students tended to spend the majority, if not all, of their time on non-technical or less technical aspects of the project. On the other hand, male team members took the lead on the technical aspects.

To quantify this observation, during the project presentations, we noted the gender of the presenter and the part of presentation they took a lead on during the presentation. Table 8 contains the details of this experiment for groups that contained both men and women. The first two columns list the group number and the composition of the groups (number of females and males). The next two columns list the various sections of the presentation and the gender of the presenter.

From the table, we can see that groups which consisted of an equal number of men and women (groups 1 and 2), the most technical part of the presentation, model description, was led by male students, whereas female students described the less technical portion of the project like pre-processing. Especially, in group 2, female students motivated and described the problem and pre-processing whereas the male students described the model and experiments. This group also had a detailed analysis of the technical problems they faced and how they addressed them, and this section was also described by a male student. On the other hand, in group 3, which consisted of more women than men, the technical

Table 8: Allocation of presentation tasks by gender

Group	Group Composition	Topic of the presentation	Gender
1	2F, 2M	Introduction	M
		Pre-processing	F
		Model description	M
		Experimental results	F
2	2F, 2M	Introduction	F
		Pre-processing	F
		Model description and results	M
		Overcoming practical Machine Learning challenges like overfitting, data skewness, etc	M
3	3F, 1M	Introduction	F
		Pre-processing code	F
		Model 1 description and results	F
		Model 2 description and results	M

part was led by a male and a female student with equal (presentation) contributions. In particular, this group worked on 2 different models and each was presented by a male and a female student. The remaining content was presented by the rest of the team members (all women).

To summarize, this suggests that in mixed-gender groups, male students tend to dominate the technical aspects of projects. However, our observation from a women-majority group indicated that more female-peers might be more encouraging for women. While this conclusion, of course, needs further analysis, there certainly is promise in forming all-women or mostly-women groups when teaching such classes to encourage learning and boost confidence among female students.

4.8 Gender Discrepancy of Course Satisfaction

In the exit survey, only four out of seven female students felt their goals were fully met compared with eleven out of fourteen male students. Why the discrepancy? This discrepancy in course satisfaction could be due to girls working on non-technical aspects of group projects and presentations as analyzed in Section 4.7. Because female students worked on non-technical aspects of the projects they did not feel comfortable with their ability to create their own technical projects in the future. One student reflected, "This course is the starting point for me, as I still have a long way to go with learning how to code." And another student wrote, "I was able to help in the simpler code but was unable to assist [my groupmates] in the more difficult parts, such as implementing linear regression, gradient descent, etc." Instructors need to identify and address these gender discrepancies throughout the group work process.

Female students could end up working on non-technical aspects of projects because female students tend to underrate their technical abilities. Correll analyzed National Education Longitudinal Study 1988 data and found that girls underrate their abilities in mathematics. Correll made sure to control for performance feedback and objective measures of their abilities (Correll 2001). Since female students underrate their technical abilities, they will not advocate for working on the technical aspects of the project. Therefore, instructors could be involved in each group's division of responsibilities to reinforce a more equitable division, between genders and other marginalized demographics.

Some data from the exit survey showed a possible method for addressing the lower satisfaction rate of female students. Three out of the four girls who felt their goals were fully met, were in a group together composed of three female and one male students. On the other hand, all three girls who felt their goals were only somewhat met, were in groups composed of two boys and two girls. This would suggest that female students would feel more satisfied when working in majority-female groups. Parker et al. used datasets from Germany and England to find that mathematics self-concept predicted students' entry into physical sciences, engineering, and mathematics. Self-concept, or a conception of the self based upon the beliefs one holds about oneself and the responses of others, was found to be a more powerful predictor of major choice than standardized tests of ability (Parker et al. 2012). In other words, the student's ability does not explain the gender gap. Instead, the gender gap can be explained by self-concept, which is influenced by the responses of others. In particular, a student's self-concept is influenced by responses from their peers and instructors. In majority-female groups, female students are more likely to have positive influences on their self-concept.

In an effort to foster a more equitable classroom environment instructors need to be aware of their power in the classroom. Students look for an instructor's feedback to form their technical self-concept. Students with positive self-concepts are more likely to successfully learn the technical material and continue working in the STEM fields.

5 Conclusion

In this work, we shared the curriculum we designed to teach ML and NLP to a group of high school students over a one-month period. In addition to sharing the curriculum suitable for teaching ML and NLP to high school students, we shared the impediments we identified in teaching the course and discussed some measures to avoid them. Throughout this paper, we also described the lessons we learned throughout this study. The outline of the takeaways from this work includes:

- The combination objectivist and constructivist curriculum we designed was successful in providing students with a more in-depth understanding of AI and helped them in being more realistic and explicit about the tasks that AI can be used for.
- Reviewing/introducing basic programming concepts in the AI curriculum for high school students is essential. This helped our students gain the expertise required to contribute to the course project.
- The instructional team also realized the necessity of identifying and addressing gender discrepancies throughout the course and, specifically, in group projects in order to assure that all students successfully meet their goals for the course.

References

Azlina, N. 2010. Cetls: Supporting collaborative activities among students and teachers through the use of think-pair-share techniques. *IJCSI International Journal of Computer Science Issues* 7.

Burton, E.; Goldsmith, J.; Koenig, S.; Kuipers, B.; Mattei, N.; and Walsh, T. 2017. Ethical considerations in artificial intelligence courses. *AI Magazine* 38.

Code.org. 2019. 33 states expand access to k-12 computer science education in 2019.

Correll, S. J. 2001. Gender and the career choice process: The role of biased self-assessments. *American Journal of Sociology 106* 6.

Google. 2018. Colaboratory.

Kagan, S. 2014. Kagan structures, processing and excellence in college teaching. *Journal on Excellence in College Teaching* 25.

Kaggle.com. 2014. Dogs vs. cats.

Kaggle.com. 2018a. Ames housing dataset.

Kaggle.com. 2018b. Disasters on social media.

Kaggle.com. 2018c. Heart disease uci.

Kaggle.com. 2019a. Amazon fine food review.

Kaggle.com. 2019b. Malaria cell images dataset.

Pantic, M.; Zwitserloot, R.; and Grootjans, R. J. 2005. Teaching introductory artificial intelligence using a simple agent framework. *IEEE Transactions on Education*.

Parker, P.; Schoon, I.; Tsai, Y.; Nagy, G.; Trautwein, U.; and Eccles, J. 2012. Achievement, agency, gender, and socioeconomic background as predictors of postschool choices: A multicontext study. *Developmental Psychology* 48.

Wolberg, W. 1992. Breast cancer wisconsin (original) data set.

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. *Thirtieth AAAI Conference on Artificial Intelligence*.