j.. / Phase5-... Type / to search

Code Issues Pull requests Actions Projects Wiki Security |

Phase5-Project

Public

1 Branch 0 Tags Go to file t About + e Add fil Code ···

jgoett2 update Readme file

614d933 · 1 minute ago 22 Commits

E...	update ...	2 hours ago
R...	update ...	3 minutes ago
....	update ...	3 days ago
R...	update ...	1 minute ago
i...	update...	10 minutes ago

No description, website, or topics provided.

Readme

Activity

0 stars

1 watching

0 forks

Releases

No releases published

Create a new release

Packages

No packages published

Publish your first package

Languages

● Jupyter Notebook 100.0%

README

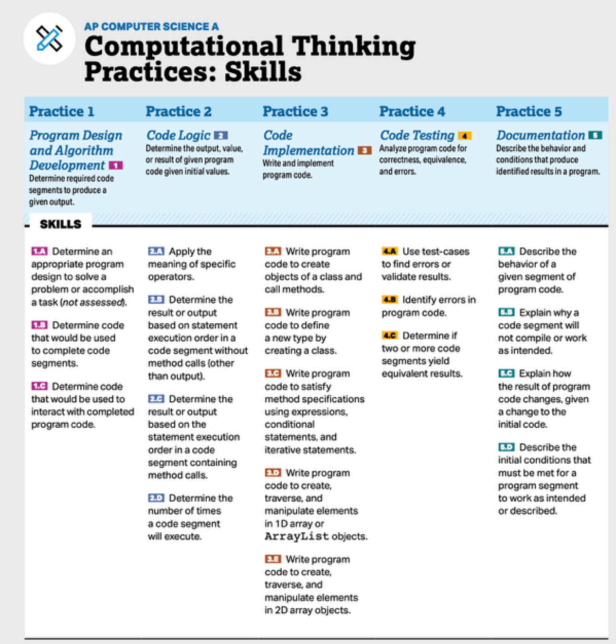
Evaluating Curriculum Rigor

Background

In my experience with high school curriculum, I have found a wide variation in the rigor of course material. This project seeks to develop a tool for evaluating the rigor of a curriculum, by measuring its alignment to the College Board's respective AP Course. This project focuses on the College Board's AP Computer Science A course, which covers a first year Java and Object Orientied Design course.

For this course, the College Board defines a set of "Computational Thinking Practices" (skills) and content that will be assessed on a year-end summative assessment to determine student's mastery of the course.

There are 5 main Computational Thinking Practices identified by the College Board, which it then breaks down into subskills:

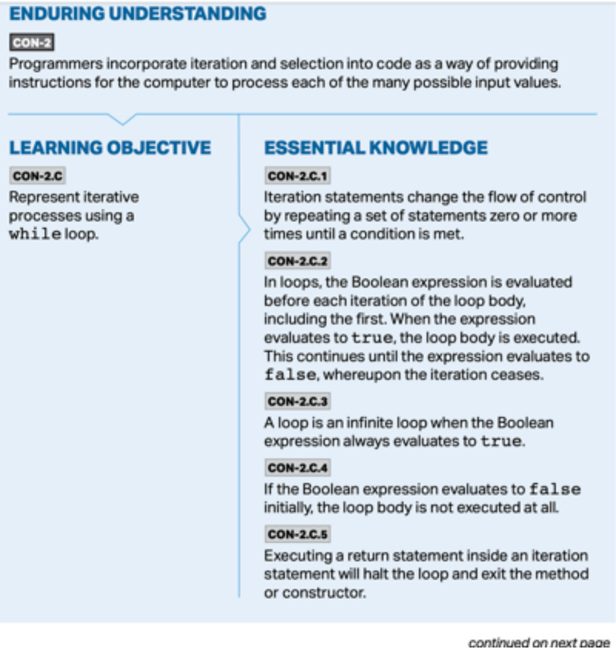


AP COMPUTER SCIENCE A

Computational Thinking Practices: Skills

Practice 1	Practice 2	Practice 3	Practice 4	Practice 5
Program Design and Algorithm Development 1 Determine required code segments to produce a given output.	*Code Logic* 2 Determine the output, value, or result of given program code given initial values.	*Code Implementation* 3 Write and implement program code.	*Code Testing* 4 Analyze program code for correctness, equivalence, and errors.	*Documentation* 5 Describe the behavior and conditions that produce identified results in a program.
SKILLS				
1.A Determine an appropriate program design to solve a problem or accomplish a task (*not assessed*). 1.B Determine code that would be used to complete code segments. 1.C Determine code that would be used to interact with completed program code.	2.A Apply the meaning of specific operators. 2.B Determine the result or output based on statement execution order in a code segment without method calls (other than output). 2.C Determine the result or output based on the statement execution order in a code segment containing method calls. 2.D Determine the number of times a code segment will execute.	3.A Write program code to create objects of a class and call methods. 3.B Write program code to define a new type by creating a class. 3.C Write program code to satisfy method specifications using expressions, conditional statements, and iterative statements. 3.D Write program code to create, traverse, and manipulate elements in 1D array or `ArrayList` objects. 3.E Write program code to create, traverse, and manipulate elements in 2D array objects.	4.A Use test-cases to find errors or validate results. 4.B Identify errors in program code. 4.C Determine if two or more code segments yield equivalent results.	5.A Describe the behavior of a given segment of program code. 5.B Explain why a code segment will not compile or work as intended. 5.C Explain how the result of program code changes, given a change to the initial code. 5.D Describe the initial conditions that must be met for a program segment to work as intended or described.

In addition, the College Board defines a set of "Essential Knowledge" (the content) to be assessed in the course, which it organizes under 5 "Big Ideas." For example, the content for a lesson on iteration is:



ENDURING UNDERSTANDING

CON-2

Programmers incorporate iteration and selection into code as a way of providing instructions for the computer to process each of the many possible input values.

LEARNING OBJECTIVE

CON-2.C

Represent iterative processes using a `while` loop.

ESSENTIAL KNOWLEDGE

CON-2.C.1

Iteration statements change the flow of control by repeating a set of statements zero or more times until a condition is met.

CON-2.C.2

In loops, the Boolean expression is evaluated before each iteration of the loop body, including the first. When the expression evaluates to `true`, the loop body is executed. This continues until the expression evaluates to `false`, whereupon the iteration ceases.

CON-2.C.3

A loop is an infinite loop when the Boolean expression always evaluates to `true`.

CON-2.C.4

If the Boolean expression evaluates to `false` initially, the loop body is not executed at all.

CON-2.C.5

Executing a return statement inside an iteration statement will halt the loop and exit the method or constructor.

continued on next page

Every question on the College Board's end-of-course summative exam is aligned to a particular computational thinking skill and essential knowledge. As a note, some school networks have found the College Board's standards to be very complete, and "backwards plan" their middle school and pre-AP high school courses to prepare students for the AP level work.

As a first step, this project will focus on the assessment questions used in a particular curriculum, and measure how well they align to the College Board's Computational Thinking Practice and Curriculum Framework. (As a note, AP classes in most subjects have an analagous set of thinking practices and framework standards, so one day, this work may be generalized to assess curriculums in other subject areas.)

The three questions to assess are:

1. How accurately does a simple classifier using either a TF-IDF or SentenceTransformer vectorizer identify the Computational Thinking Practice assessed in a question?
2. How accurately does ChatGPT, supplied with only the College Board Framework for Computational Thinking and a few sample questions, identify the Computational Thinking Practice assessed in a question?
3. Can a useful Dashboard be created to improve the quality of an assessment question?

Initial Conclusions:

1. The classifier using a TF-IDF

vectorization accurately identified the Computational Thinking Practice being accessed in 70% of the questions, while the simple classifier using the Sentence Transformed identified the Computational Thinking Practice being assessed in 85% of the questions.

2. ChatGPT had an accuracy of 48% in identifying the Computational Thinking Practice assessed in a question.

Next Steps:

1. Develop better prompts to improve ChatGPT's classification accuracy.
2. Determine whether the classifier can also identify the "Essential Knowledge" assessed by the question, not just the computational skill.
3. Attempt to generalize the classifiers to classify non-assessment questions such as lecture material, lab questions, and homework problems.
4. Create a visualization that shows the distribution of thinking skills and content assessed over the course of the curriculum.

Classifying Questions Using Logistic Regression

As first step, this section will try to classify prompts as assessing one of these two AP Computational Thinking Practices (CTP):

1. **CTP 2.A**: Apply the meaning of specific operators. For example: *Consider the following code segment.*

```
int x = 7;
int y = 3;
if ((x < 10) && (y < 0))

System.out.println(""Value
is: "" + x * y);
else

System.out.println(""Value
is: "" + x / y)
```

What is printed as a result of executing the code segment?

2. **CTP 2.B**: Determine the results or output based on statement execution order in a code segment without method calls (except for output). For example:

Consider the following code segment.

```
int[] arr = {7, 2, 5, 3,
0, 10};
for (int k = 0; k <
arr.length - 1; k++)  {
  if (arr[k] > arr[k + 1])
    System.out.print(k +
"" "" + arr[k] + "" "");
}
```

What will be printed as a result of executing the code segment?

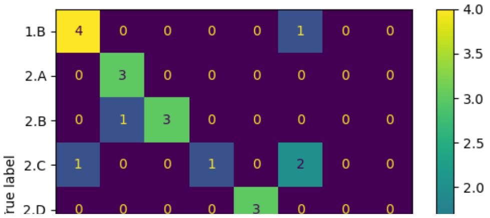
Preprocessing the Data

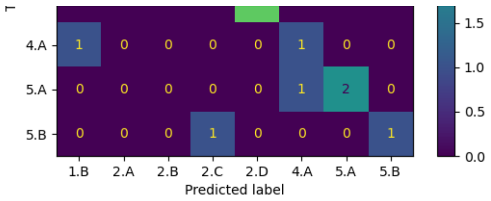
In this first step, we will:

1. Load questions assessing each Computational Thinking Practice.
2. Filter the question set to only consider the most frequently assessed Computational Thinking Practices.

Testing the Logistic Regression Classifier on the Questions

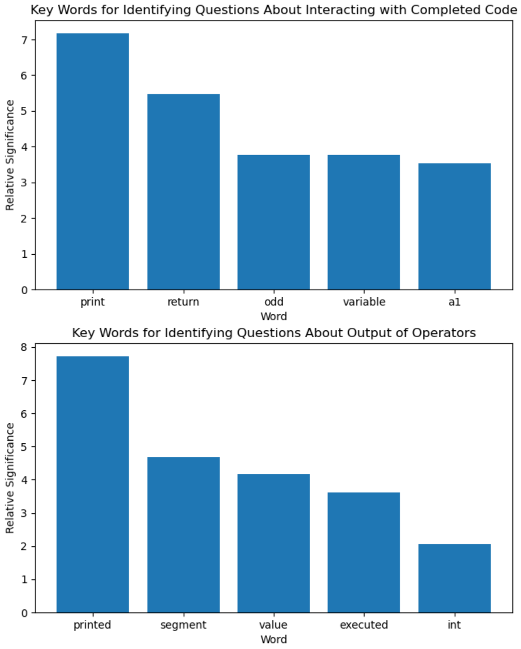
Here, we build and evaluate a classifier that vectorizes questions using the TF-IDF vectorizer, and classifies using a Logistic Regression Classifier.





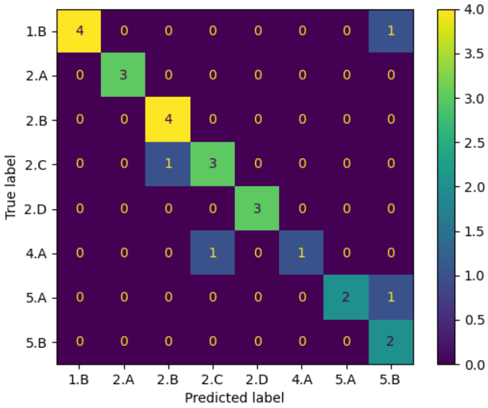
Interpreting the Logistic Classifier to Identify Key Features

Determine the most significant works for identifying each question.



Classification with Sentence Transformer Encoding

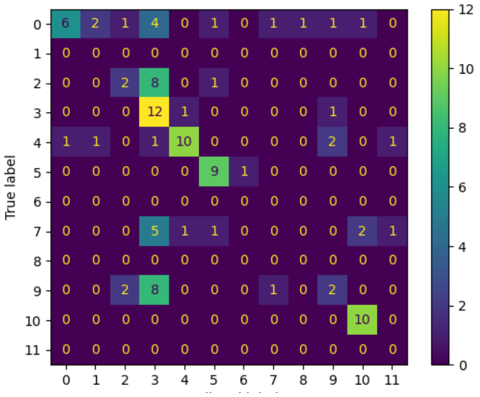
Next, the questions will be vectorized using a Sentence Transformer Embedding, which may capture some of the semantics in the question lost in the earlier TF-IDF vectorier.

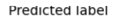


Build and Test a ChatGPT Classifier

This classifier asks ChatGPT to determine the Computational Thinking Skill being assessed in the problem.

1. Create a function call to ask ChatGPT for the classification
2. Test ChatGPT on data set and evaluate classification.





Summary

This project had some success at identifying the skills assessed in a particular question.

1. The classifier using a TF-IDF vectorization accurately identified the Computational Thinking Practice being accessed in 70% of the questions, while the simple classifier using the Sentence Transformed identified the Computational Thinking Practice being assessed in 85% of the questions.
2. ChatGPT had an accuracy of 48% in identifying the Computational Thinking Practice assessed in a question.
3. A simple tool was developed for identifying the skill assessed by a question

Next Steps:

1. Develop better prompts to improve ChatGPT's classification accuracy.
2. Determine whether the classifier can also identify the "Essential Knowledge" assessed by the question, not just the computational skill.
3. Attempt to generalize the classifiers

to classify non-assessment questions such as lecture material, lab questions, and homework problems.

4. Create a visualization that shows the distribution of thinking skills and content assessed over the course of the curriculum.