AI Literacy and Informed Skepticism Part 2: What Can AI Do?

An Aside: The New Department of Labor Guidance on

As fate would have it, on Feb 13th the US Department of Labor (DOL) released a high-level AI Literacy Framework. As far as I know this is the first federal attempt to establish a common definition of what is meant by AI Literacy and the competencies it entails. In my last post, I gave a definition of AI Literacy taken from a widely cited paper by Long & Magerko

"a set of competencies that enables individuals to critically evaluate AI technologies; communicate and collaborate effectively with AI; and use AI as a tool online, at home, and in the workplace" (Long & Magerko, 2020, p. 2))

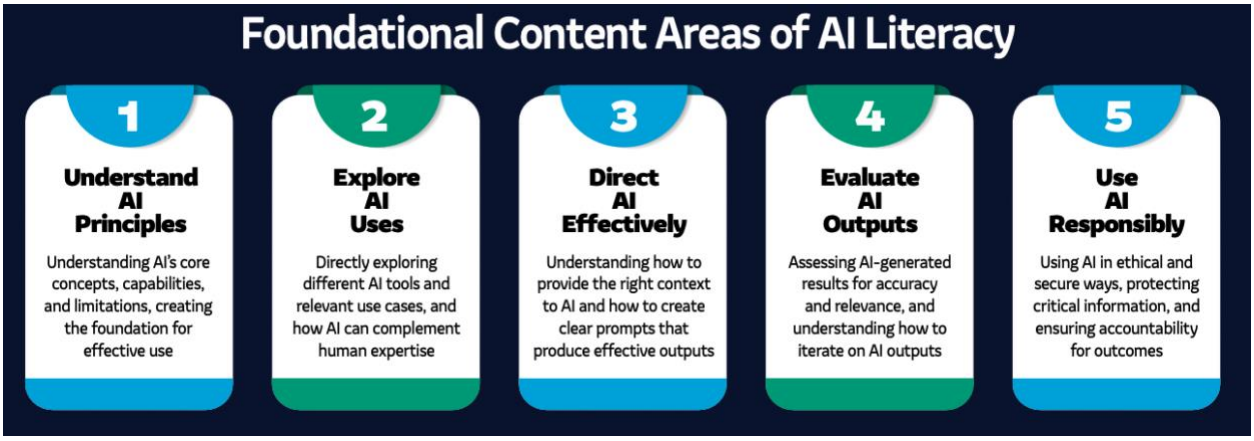
The DOL framework defines AI Literacy as

"a foundational set of competencies that enable individuals to use *and* evaluate AI technologies responsibly, with a primary focus on generative AI, which is increasingly central to the modern workplace."

The definitions seem pretty similar…I could go down a rabbit hole about some subtle distinctions between the two, but let's agree there is a lot of overlap. Each references a "set of competencies" necessary to achieve some foundational level of literacy.

The competencies I had in mind as the basis for this four-part series are those that that make it possible to answer the following questions: What is AI? What can AI do? How does AI work? How can AI be used ethically?

The DOL competencies are shown in the graphic below.



Again, a lot of overlap. The main difference is that the DOL framework has "Evaluate AI Outputs" as a distinct competency, but I think there is some logical circularity given that, according to their own definition, the competencies are supposed to be the thing that "enable"

a person to evaluate. Therefore, it seems nonsensical to speak of needing to learn to evaluate AI outputs in order to be able to…evaluate AI outputs. So, I'm going to stick with my plan to have a post for each of my original four competencies. But you'll see me infuse a lot of what I write, as I have already, with the goal that we should all strive to be critical users and evaluators of AI. And for sure there will be a future post—probably many—that have this explicit focus.

So, What Can AI Do?

One of the defining features of LLMs is their generality. We are rapidly reaching the point where any task a human can carry out through interaction with computer software is a task that an AI agent can replicate, usually much faster. Whether AI can do the task as well (or better) than a human is another story.

Casey Newton and Kevin Kruse, hosts the fabulous podcast Hard Fork make a distinction between East Coast and West Coast mentalities when it comes to AI capabilities. Tech enthusiasts in Silicon Valley focus on all the cool things AI can do, New Yorkers focus on all the ways it falls short. As Ethan Mollick puts it, as of today we still have a very jagged frontier.

My advice to those skeptical (or just curious) about what AI can do for them is to conduct little experiments in domains where they have experience and expertise. Let me illustrate. (If you aren't a psychometrician the specifics of this example may not resonate, but I promise you'll still appreciate the punchline.)

Can Claude Help Me Teach the EM Algorithm?

I regularly teach a doctoral seminar here at the University of Colorado that focuses on psychometric modeling in general, and item response theory in particular. It's a class I've taught on about 10 different occasions. The most technically demanding part of the course is the week spent on the topic of item parameter estimation. There are all kinds of good ways to introduce and explain joint maximum likelihood estimation (JMLE), but when it comes to marginal maximum likelihood estimation (MMLE), I inevitably do a whole lot of hand waving. To be honest, I think part of the problem might be that I never learned it well enough myself to feel sufficiently confident in my ability to teach it. There's a lot to pieces that need to be put together between the role of Bayes Theorem in computing a posterior distribution, use of quadrature for numerical integration, and use of "artificial data" in the expectation step of the EM algorithm.

Each time I would teach the course, I would dutifully re-read the assigned materials and try my best to modify or create that one slide or activity to unlock an "a-ha!" moment from my students. What I kept trying to find was a middle ground between sending students to pore over page after page of partial derivatives in Chapter 6 of Baker & Kim's (excellent) 2004 textbook *Item Response Theory: Parameter Estimation Techniques*, vs. giving them a very high-level overview in chapter 6 of Embretson & Reise's 2000 book *Item Response Theory for Psychologists*.

So, last week I turned to Claude Code (invoking Opus 4.6) for help.

> I have uploaded two new documents. The book 'Item Response Theory Parameter Estimation Techniques' (pdf file) and the PowerPoint deck 'Item_Parameter_Estimation.pptx'. For years I have struggled to figure out a good way to help my students understand some key technical details about marginal maximum likelihood estimation. You can see what I have attempted on my slides 13 to 31. The expert source for this topic is Chapter 6 in the book I uploaded. Read that book and look at my slides. Adopt my role as a professor teaching IRT item parameter estimation to students with a variety of math backgrounds (some know advanced calculus, some do not). Can you suggest improvements to my approach and bring them to life in an R Markdown file? If you have multiple ideas let me know so we can work through this together.

After reading the files I had uploaded, Claude responded with (1) an analysis of my current slides on MMLE/EM: what was working well and where students were likely to get lost; and (2) four ideas for an improvement via an R Markdown file. About an hour later, I had found my middle ground between Baker & Kim and Embretson & Reise. You can see the transcript of our iterations here (insert link to conversation_transcript.txt). What Claude created for me was not a watered-down regurgitation of Baker & Kim or a glorified rearrangement of text from my slides. It was—to my eyes at least—a novel approach that explained and illustrated IRT item calibration in a way I had never seen before.

Before I share it, two key points about my interaction with Claude

1) My prompt contains a lot of crucial context. I have explained the task ("to figure out a good way to help my students understand some key technical details about marginal maximum likelihood estimation"); I've given Claude a role ("professor teaching IRT item parameter estimation to students with a variety of math backgrounds"); I've given Claude examples to build upon (the Baker & Kim book, my slides); I've given concrete instructions ("Suggest improvements to my approach and bring them to life in an R Markdown file. If you have multiple ideas let me know so we can work through this together.") All of this falls under the umbrella of prompt engineering.
2) I feel comfortable saying that the final R Markdown was something Claude and I generated together because after getting the initial draft, I read through it and identified several ways to improve the document, and I found one mistake. Claude and I made a good team. I'm not embarrassed to admit that Claude understands the mathematics of parameter estimation better than me. But, at least for now, I'm still far better equipped to put myself in the role of my students than Claude.

Here is what we came up as an R Markdown document [insert link to tem_Parameter_Estimation_MMLE.html] , and here are the slides I presented for a deep dive into the EM algorithm in class [insert link to MMLE_EM_Slides.pdf].

At the very least, my "vibe evaluation" via expert judgment is that the new learning resources developed with Claude are a huge improvement over my old resources. The bigger question: did the availability of these new resources and my use of them in class have a positive effect on my students' learning? In the absence of a randomized experiment, it's hard to say, and I'll have to wait until the oral exam I do with my students at the end of the semester to probe what it is that they can now explain.

From Ben Domingue to Claude Code

If you haven't yet had the experience of working with something like Anthropic's Claude Code or OpenAI's Codex to complete a complex multistep coding task, I think it can be hard to convey the mix of emotions you are likely to experience. Looking at the conversation transcript after the fact doesn't do it justice. Maybe this back story will help. Relatively early in my career I had the good fortune of lucking into one of the most talented doctoral advisees I've ever had—Ben Domingue, now a tenured Associate Professor at Stanford. Ben was (and still is) one of the fastest coders I've ever seen. Dude is Mozart on a keyboard. I think he picked up R in a month and midway through his first semester in graduate school he was some order of magnitude better at coding than I was. Working with Ben on projects actually required some adjustments to the expectations I had in place for professor-RA interactions. For past RAs, my approach had been to assign them project tasks, and ask them to provide status updates towards completion during weekly meetings. But with Ben, I could express an idea to him during a meeting at 3, and by 6 he had figured out a way to code it and was emailing me with results and questions about next steps just as I was getting home from my afternoon workout. It was a great problem to have! But it also took some getting used to, because it forced me to change my pace and to rearrange both my schedule and my expectations.

Claude Code is like Ben raised to the power of 10. I'd like to use the analogy that Claude Code is the steam-powered drilling machine and Ben is John Henry, but in that mythical tale John Henry is actually able to defeat the drilling machine before dropping dead of exhaustion. But in this matchup, with all due respect to Ben, it's no contest. When it comes to coding, what Ben can do in a few hours, Claude can do in a few minutes. What Ben can do in minutes, Claude can do in seconds. So, to experience this in real time brings back the sense of exhilaration and destabilization—AI Vertigo—I felt when I first started working with Ben, but on whole new level. Exhilaration because I realize I can make progress on a project at a much more rapid pace than I had planned; destabilization because I feel the urge to make immediate decisions with just minutes (or seconds) of reflection instead of hours or days. In the MMLE/EM example above, Claude analyzed my pre-existing lesson plan, came up with four alternatives for improving it within about 5 minutes, and then gave me a menu of options for my next step. All that was required to take the next step was to select the numeric option. Make your choice and then Claude is off to the races as it starts

> Tomfoolering, Meandering, Germinating, Leavening, Sketching, Churning, Shenanagigging, Embelishing, Crystalizing, Shimying, Hyperspacing, Architecting,

> Considering, Elucidating, Swirling, etc. (these are just 15 of an endless array of creative adjectives that appear pulsating on the Claude Code terminal along with snippets of Claude's reasoning process happening in real time)

This happens for seconds to minutes before you are presented with a new set of options. It can begin to feel like playing a video game, and the urge to keep going is addictive. And what the transcript of human to agent interactions can't capture is what happens any time Claude encounters a technical obstacle. You can see red error messages appear as Claude is trying to get code to compile. A very familiar and frustrating sight for human coders. How many hours have I wasted in my past trying to debug code? But Claude never gets frustrated, and never gets tired. The only thing that stops it is me hitting my usage limits. Claude either figures out a solution, or, in rare cases, gives me instructions for what I need to do to unblock a bottleneck.

But Hold Your Horses

On February 6th a blog post from Matt Shumer (and surely an AI co-author since the AI tells strike me as overwhelming) entitled "Something Big is Happening" [insert link to https://shumer.dev/something-big-is-happening] went viral. You should read it for yourself if you haven't already. The basic thesis is that most people in white collar occupations don't realize just how fast AI capabilities are advancing, and they aren't prepared for the disruptive effect it is going to have on them professionally. To illustrate, Shumer writes

> **I am no longer needed for the actual technical work of my job.** I describe what I want built, in plain English, and it just... appears. Not a rough draft I need to fix. The finished thing. I tell the AI what I want, walk away from my computer for four hours, and come back to find the work done. Done well, done better than I would have done it myself, with no corrections needed. A couple of months ago, I was going back and forth with the AI, guiding it, making edits. Now I just describe the outcome and leave.
>
> Let me give you an example so you can understand what this actually looks like in practice. I'll tell the AI: "I want to build this app. Here's what it should do, here's roughly what it should look like. Figure out the user flow, the design, all of it." And it does. It writes tens of thousands of lines of code. Then, and this is the part that would have been unthinkable a year ago, it **opens the app itself**. It clicks through the buttons. It tests the features. It uses the app the way a person would. If it doesn't like how something looks or feels, it goes back and changes it, on its own. It iterates, like a developer would, fixing and refining until it's satisfied. Only once it has decided the app meets its own standards does it come back to me and say: "It's ready for you to test." And when I test it, it's usually perfect.
>
> I'm not exaggerating. That is what my Monday looked like this week.

I agree with a lot of what Shumer is saying in his piece, but on this, at least based on my experiences, I call bullshit.

In virtually every task/project I've had Claude Code complete, it's true that my first reaction is "Holy Shit! This looks perfect!" But once I get over the dopamine hit, I always discover something in need of correction and revision. Sometimes the fix can be relatively minor. In the MMLE/EM tutorial, Claude had included a formula for one of the estimating equations that was based on the 3PL when the running example that had preceded it was for the 2PL. An easy fix (but something that would have confused my students). But other times the problem can be more dramatic. After enlisting Claude to craft the MMLE/EM tutorial, I decided for the sake of completeness that I should have Claude create a JMLE tutorial with parallel structure. Claude wrote the necessary JMLE functions in R and then applied them to the same 5-item toy dataset (taken from a classic LSAT6 example used by Bock & Aitken). Unfortunately, this led to an epic failure. On one of the items, estimates of discrimination and location blew up and failed to converge, for the other items, estimates of discrimination were lower than what had been estimated using MMLE, which was in direct contradiction to the point Claude was making in its text (use of JMLE leads to an upward bias is discrimination—a consequence of the Neyman-Scott problem).

Did Claude notice this problem and contradiction in the tutorial it produced? No. When I pointed it out and asked it to investigate, Claude dutifully noted that indeed, something was going very wrong in the applied example. It then spent the next half hour trying to figure out why until I interrupted to point out that in a subsequent example it had presented using a test with 31 items that I had made available, the JMLE vs MMLE comparison illustrated the theoretical point perfectly. Once I gave it that hint, Claude figured out the problem was with the data it had been using. In hindsight, this should have been obvious. But it shows the present-day limits of AI meta-cognition.

There are many more examples I could give, but let me end with this one. I wanted to see how Claude Code would do on a IRT empirical analysis project I'd assigned to my students. It's a project that I'd expect would require my students somewhere between 5 and 8 hours (or maybe more) to complete, because they have to actively learn IRT content to complete the task. Claude already has all that content effectively stored in its model parameters, and thus finished the assignment in 18 minutes. Generally speaking, it was way better than what my students would turn in.

But it had two curious shortcomings. The first was that Claude failed to notice or mention that one of the items it flagged in its writeup for a low point-biserial correlation had led to a convergence problem when using MMLE to calibrate the 3PL. The second, more humorous problem was that Claude had named the author of the document as…David Briggs. Putting aside for now discussion about the ethics of me being listed as the author (in my view to the extent an author is listed on this kind of fully generated output, it should also include the name of the AI agent), where did "David" come from? After all, anytime I log into the general Claude chatbot, it greets me with messages like "Back at it, Derek!" Apparently, Claude Code must not reference the personalization settings in the more general Claude chat bot. And in this instance, since the root file directory contained no documents with my name, Claude Code must have

guessed that from the “briggsd” in my file directory that Briggs was my last name and the first initial of my first name was d. From there I suppose “David” must be the most common first name associated with Briggs. All understandable. But why not ask me about such a consequential decision?

It's Not What Can AI Do, but How Well Can AI Do It?

I hope after reading this you come to the same important realization as me: the question is not what AI can do, but how well it can do it. In this sense, an *understanding* of AI use cases is iteratively connected to taking an active approach to the *evaluation* of AI use cases.

My Advice

Find something you know a lot about already. Invite AI to work on a meaningful project or task in that space. Interrogate the result. Always keep the counterfactual as a frame of reference: how does the output I get from AI, the decisions it helps me make, compare to what I would have done in its absence.

And be prepared to regularly revisit these experiments with the release of new models, because an AI failure in the past or present is no guarantee of a failure in the future. As Ethan Mollick puts it, assume that the AI you are interacting with today is the worst AI you will ever use.

Resources

If you’re a student learning about psychometrics, or a professor teaching it, you might like to see the full set of tutorials on IRT parameter estimation I created with Claude Code. If you find them to be helpful (or notice an error that I missed), please let me know.

Insert links to pdf versions of

Person_Parameter_Estimation.html
Item_Parameter_Estimation_JMLE.html
Item_Parameter_Estimation_MMLE.html

Insert link to PowerPoint file I used to teach about item parameter estimation before my collaborationg with Claude Code

5_Item_Parameter_Estimation.pptx