The AI Hierarchy of Needs

June 12th 2017

TWEET THIS

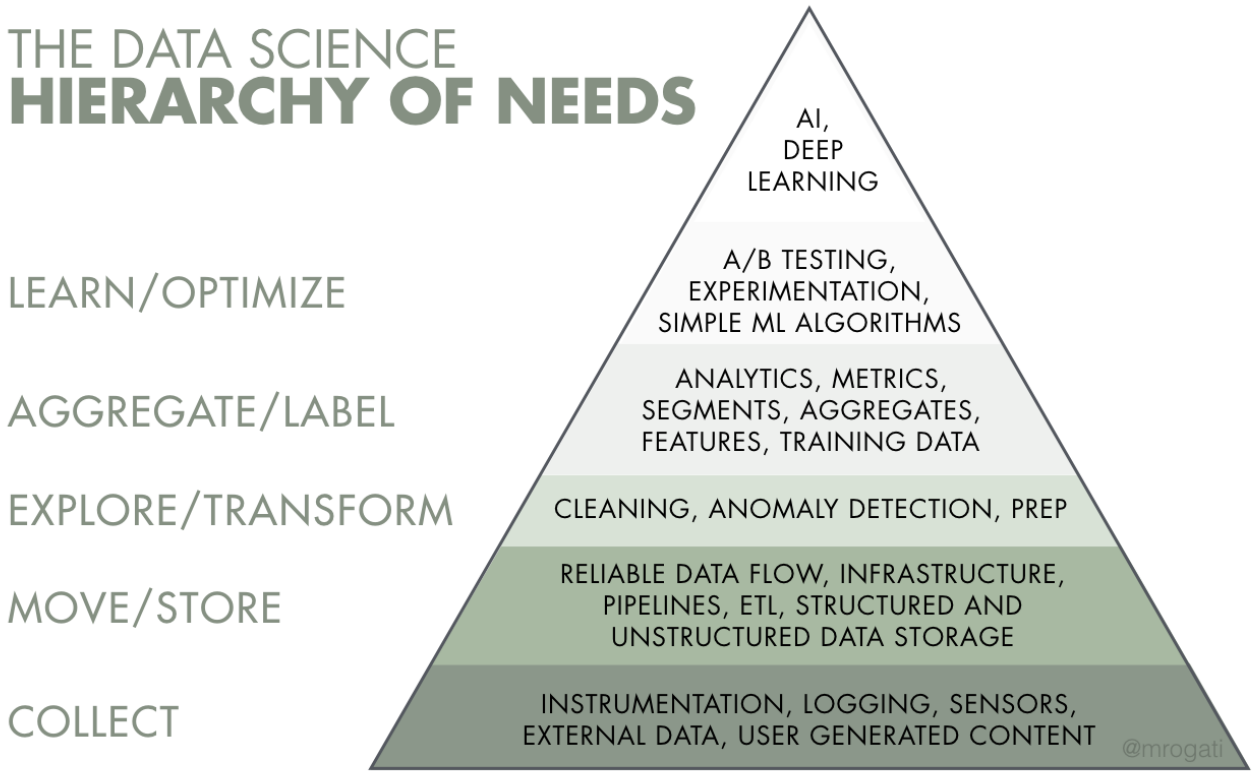
As is usually the case with fast-advancing technologies, AI has inspired massive FOMO , FUD and feuds. Some of it is deserved, some of it not—but the industry is paying attention. From stealth hardware startups to fintech giants to public institutions, teams are feverishly working on their AI strategy. It all comes down to one crucial, high-stakes question: ***'How do we use AI and machine learning to get better at what we do?'***

More often than not, companies are **not** ready for AI. Maybe they hired their first data scientist to less-than-stellar outcomes, or maybe data literacy is not central to their culture. But the most common scenario is that they have not yet built the infrastructure to implement (and reap the benefits of) the most basic data science algorithms and operations, much less machine learning.

As a data science/AI advisor, I had to deliver this message countless times, especially over the past two years. Others agree. It's hard to be a wet blanket among all this excitement around your own field, especially if you share that excitement. And how do you tell companies they're not ready for AI without sounding (or being) elitist—a self-appointed gate keeper?

Here's an explanation that resonated the most:

Think of AI as the top of a pyramid of needs. Yes, self-actualization (AI) is great, but you first need food, water and shelter (data literacy, collection and infrastructure).



You need a solid foundation for your data before being effective with AI and machine learning.

Basic needs: Can you count?

At the bottom of the pyramid we have **data collection**. What data do you need, and what's available? If it's a user-facing product, are you logging all relevant user interactions? If it's a sensor, what data is coming through and how? How easy is it to log an interaction that is not instrumented yet? After all, the right dataset is what made recent advances in machine learning possible.

Next, how does the **data flow** through the system? Do you have reliable streams / ETL ? Where do you store it, and how easy is it to access and analyze? Jay Kreps has been saying (for about a decade) that reliable data flow is key to doing anything with data. *[Aside: I was looking for an exact quote and found it in his 'I love logs' masterpiece. I then noticed that, one paragraph over, he's making this exact Maslow's hierarchy of needs comparison, with an 'it's worth noting the obvious' thrown in there for good measure (thanks Jay!). Speaking of related work, I've also later run (h/t Daniel Tunkelang) into Hilary Mason and Chris Wiggins's excellent post about what a data scientist does. Days ago, Sean Taylor unveiled his own data science pyramid of needs (ironically dubbed the Unconjoined Triangle of Data Science) which, of course, is completely different. Perhaps we should start a tumblr.]*

Only when data is accessible, you can **explore and transform** it. This includes the infamous 'data cleaning', an under-rated side of data science that will be the subject of another post. This is when you discover you're missing a bunch of data, your sensors are unreliable, a version change meant your events are dropped, you're misinterpreting a flag—and you go back to making sure the base of the pyramid is solid.

When you're able to reliably explore and clean the data, you can start building what's traditionally thought of as BI or **analytics**: define metrics to track, their seasonality and sensitivity to various factors. Maybe doing some rough user segmentation and see if anything jumps out. However, since your goal is AI, you are now building what you'll later think of as **features** to incorporate in your machine learning model. At this stage, you also know what you'd like to predict or learn, and you can start preparing your **training data** by generating labels, either automatically (which customers churned?) or with humans in the loop.

OK, I can count. Now what?

We have training data—surely, now we can do machine learning? Maybe, if you're trying to internally predict churn; no, if the result is going to be customer-facing. We need to have a (however primitive) A/B testing or **experimentation** framework in place, so we can deploy incrementally to avoid disasters and get a rough estimate of the effects of the changes before they affect everybody. This is also the right time to put a very **simple baseline** in place (for recommender systems, this would be e.g. 'most popular', then 'most popular for your user segment'—the very annoying but effective 'stereotype before personalization').

Simple heuristics are surprisingly hard to beat, and they will allow you to debug the system end-to-end without mysterious ML black boxes with hypertuned hyperparameters in the middle. This is also why my favorite data science algorithm is division.

At this point, you can deploy a very simple ML algorithm (like logistic regression or, yes, division), then think of new signals and features that might affect your results. Weather & census data are my go-tos. And no—as powerful as it is, deep learning doesn't automatically do this for you. Bringing in new signals (feature creation, not feature engineering) is what can improve your performance by leaps and bounds. It's worth spending some time here, even if as data scientists we're excited about moving on to the next level in the pyramid.

Bring on the AI!

You made it. You're instrumented. Your ETL is humming. Your data is organized & cleaned. You have dashboards, labels and good features. You're measuring the right things. You can experiment daily. You have a baseline algorithm that's debugged end-to-end and is running in production—and you've changed it a dozen times. You're ready. Go ahead and try all the latest and greatest out there—from rolling your own to using companies that specialize in machine learning. You might get some big improvements in production, or you might not. Worst case, you learn new methods, develop opinions and hands-on experience with them, and get to tell your investors and clients about your AI efforts without feeling like an impostor. Best case, you make a huge difference to your users, clients and your company—a true machine learning success story.

Wait, what about MVPs, agile, lean and all that?

The data science hierarchy of needs is not an excuse to build disconnected, over-engineered infrastructure for a year. Just like when building a traditional MVP (minimally viable product), you start with a small, vertical section of your product and you make it work well end-to-end. You can build its pyramid, then grow it horizontally. For example, at Jawbone, we started with sleep data and built its pyramid: instrumentation, ETL, cleaning & organization, label capturing and definitions, metrics (what's the average # of hours people sleep every night? What about naps? What's a nap?), cross-segment analyses all the way to data stories and machine learning-driven data products (automatic sleep detection). We later extended this to steps, then food, weather, workouts, social network & communication—one at a time. We did not build an all-encompassing infrastructure without ever putting it to work end-to-end.

Asking the right questions and building the right products

This is only about how you **could**, not whether you **should** (for pragmatic or ethical reasons).

The promise of machine learning tools

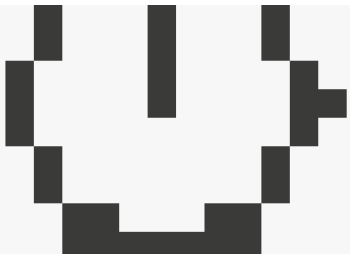
'*Wait, what about that Amazon API or TensorFlow or that other open source library? What about companies that are selling ML tools, or that automatically extract insights and features?*'

All of that is awesome and very useful. (Some companies do end up painstakingly custom-building your entire pyramid so they can showcase their work. They are heroes.) However, under the strong influence of the current AI hype, people try to plug in data that's dirty & full of gaps, that spans years while changing in format and meaning, that's not understood yet, that's structured in ways that don't make sense, and expect those tools to magically handle it. And maybe some day soon that will be the case; I see & applaud efforts in that direction. Until then, it's worth building a solid foundation for your AI pyramid of needs.

Data Science # Artificial Intelligence # Machine Learning # Big Data # Ai

Continue the discussion

More by Monica Rogati



How not to hire your first data scientist

Monica Rogati

Data Science

Hackernoon Newsletter curates great stories by real tech professionals

Get solid gold sent to your inbox. Every week!

Email Address *

First Name

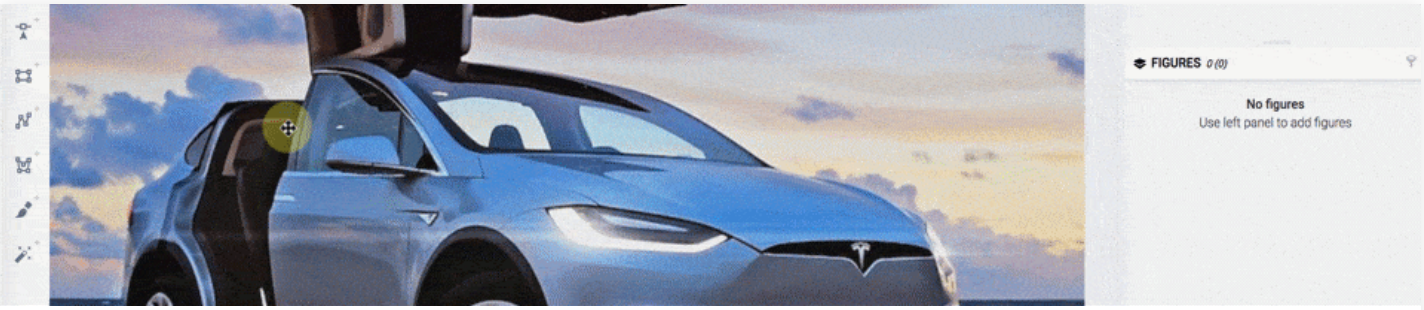
Last Name

TOPICS OF INTEREST

- [x] Software Development
- [x] Blockchain Crypto
- [x] General Tech
- [x] Best of Hacker Noon

Get great stories by email

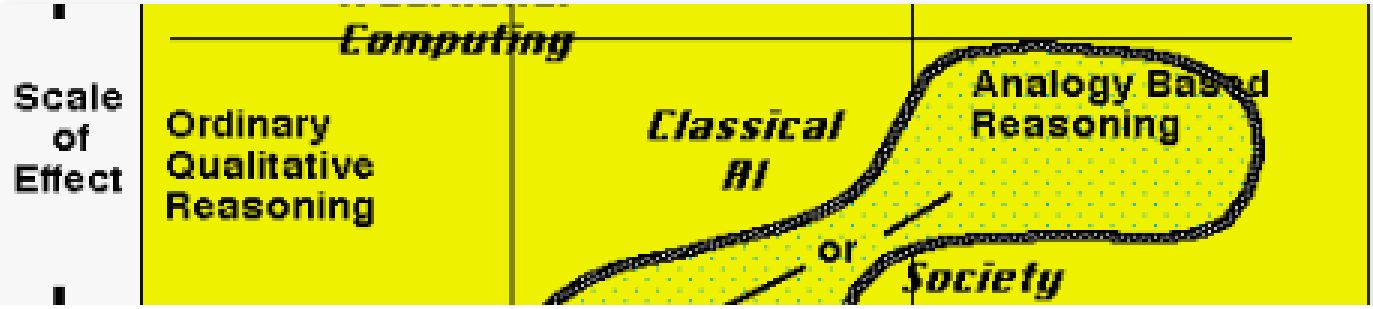
More Related Stories



✏️ Advanced annotation tools in Deep Learning: training data for computer vision with Supervisely

Supervise.ly
Dec 19

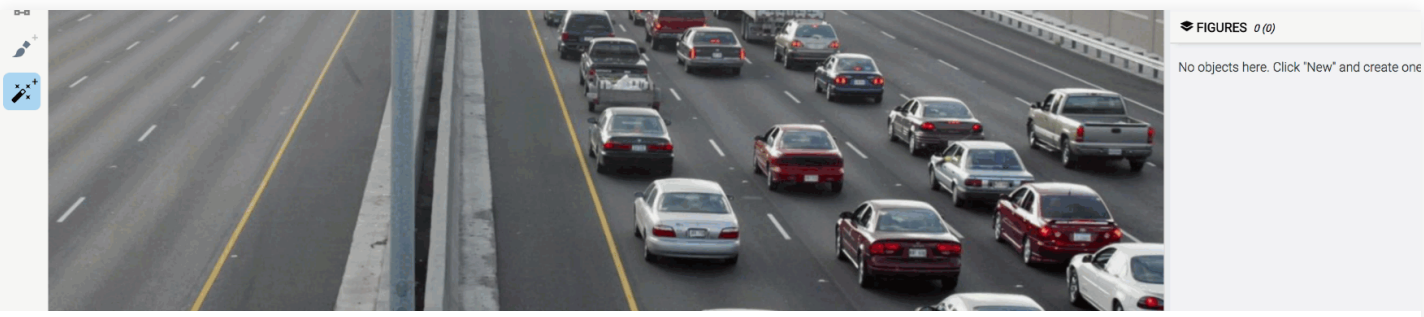
Saas



1—Cool things this week

Shreya Amin
Oct 05

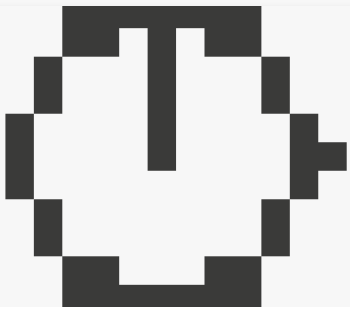
Artificial Intelligence



⚔ Big challenge in Deep Learning: training data

Deep Systems
Nov 21

#Data Science



10 Amazing Articles On Python Programming And Machine Learning Week 3

SeattleDataGuy

#Machine Learning

10 Open Source AI Project Ideas For Startups

Shreya Chawla
Apr 01

#Startup

Help
About
Start Writing
Sponsor:
Brand-as-Author
Sitewide Billboard

Contact Us Privacy Terms