

Printout

Thursday, 1 November 2018

12:02 PM

**Harvard
Business
Review**

PRODUCT DEVELOPMENT

How to Build Great Data Products

by Emily Glassberg Sands

OCTOBER 30, 2018



CINTASCOTCH/GETTY IMAGES

Products fueled by data and machine learning can be a powerful way to solve users' needs. They can also create a “data moat” that can help stave off the competition. Classic examples include Google search and Amazon product recommendations, both of which improve as more users engage. But the opportunity extends far beyond the tech giants: companies of a range of sizes and across sectors are investing in their own data-powered products. At Coursera, we

https://hbr.org/2018/10/how-to-build-great-data-products?utm_medium=email&utm_s... 1/11/2018

use machine learning to help learners find the best content to reach their learning goals, and to ensure they have the support – automated and human – that they need to succeed.

The lifecycle of a so-called “data product” mirrors standard product development: identifying the opportunity to solve a core user need, building an initial version, and then evaluating its impact and iterating. But the data component adds an extra layer of complexity. To tackle the challenge, companies should emphasize cross-functional collaboration, evaluate and prioritize data product opportunities with an eye to the long-term, and start simple.

Stage 1: Identify the opportunity

Data products are a team sport

Identifying the best data-product opportunities demands marrying the product-and-business perspective with the tech-and-data perspective. Product managers, user researchers, and business leaders traditionally have the strong intuition and domain expertise to identify key unsolved user and business needs. Meanwhile, data scientists and engineers have a keen eye for identifying feasible data-powered solutions and a strong intuition on what can be scaled and how.

To get the right data product opportunities identified and prioritized, bring these two sides of the table together. A few norms can help:

- *Educate data scientists about the user and business needs.* Keeping data scientists in close alignment with product managers, user researchers, and business leads, and ensuring that part of their role is to dig in on the data directly to understand users and their needs will help.
- *Have data scientists serve as data evangelists,* socializing data opportunities with the broader organization. This can range from providing the organization with

easy access to raw data and model output samples in the early ideation stages, to building full prototypes in the later stages.

- *Develop the data-savvy of product and business groups.* Individuals across a range of functions and industries are upskilling in data, and employers can accelerate the trend by investing in learning programs. The higher the data literacy of the product and business functions, the better able they'll be to collaborate with the data science and tech teams.
- *Give data science a seat at the table.* Data science can live different places in the organization (e.g., centralized or decentralized), but no matter the structure having data science leaders in the room for product and business strategy discussions will accelerate data product development.

Prioritize with an eye to the future

The best data products get better with age, like a fine wine. This is true for two reasons:

First, data product applications generally accelerate data collection which in turn improves the application. Consider a recommendations product powered by users' self-reported profile data. With limited profile data today, the initial (or "cold start") recommendations may be uninspiring. But if users are more willing to fill in a profile when it's used to personalize their experience, launching recommendations will accelerate profile collection, improving the recommendations over time.

Second, many data products can be built out to power multiple applications. This isn't just about spreading costly R&D across different use-cases; it's about building network effects through shared data. If the data produced by each application feeds back to the underlying data foundations, this improves the applications, which in turn drives more utilization and thus data collection, and the virtuous cycle continues. Coursera's Skills Graph is one example. A series of algorithms that

map a robust library of skills to content, careers, and learners, the graph powers a range of discovery-related applications on the site, many of which generate training data that strengthen the graph and in turn improve its applications.

Too much focus on near-term performance can yield underinvestment in promising medium- or long-term opportunities. More generally, the criticality of high-quality data cannot be overstated; investments in collecting and storing data should be prioritized at every stage.

Stage 2: Build the product

De-risk by staging execution

Data products generally require validation both of whether the algorithm works, and of whether users like it. As a result, builders of data products face an inherent tension between how much to invest in the R&D upfront and how quickly to get the application out to validate that it solves a core need.

Teams that over-invest in technical validation before validating product-market fit risk wasted R&D efforts pointed at the wrong problem or solution. Conversely, teams that over-invest in validating user demand without sufficient R&D can end up presenting users with an underpowered prototype, and so risk a false negative. Teams on this end of the spectrum may release an MVP powered by a weak model; if users don't respond well, it may be that with stronger R&D powering the application the result would have been different.

While there's no silver bullet for simultaneously validating the tech and the product-market fit, staged execution can help. Starting simple will accelerate both testing and the collection of valuable data. In building out our Skills Graph, for

example, we initially launched skills-based search – an application that required only a small subset of the graph, and that generated a wealth of additional training data. A series of MVP approaches can also reduce time to testing:

- *Lightweight models* are generally faster to ship and have the added benefit of being easier to explain, debug, and build upon over time. While deep learning can be powerful (and certainly is trending) in most cases it's not the place to start.
- *External data sources*, whether open source or buy/partner solutions, can accelerate development. If and when there's a strong signal from the data the product generates, the product can be adapted to rely on that competitive differentiator.
- *Narrowing the domain* can reduce the scope of the algorithmic challenge to start. For example, some applications can initially be built and launched only for a subset of users or use-cases.
- *Hand-curation* – where humans either do the work you eventually hope the model will do, or at least review and tweak the initial model's output – can further accelerate development. This is ideally done with an eye to how the hand-curation steps could be automated over time to scale up the product.

Stage 3: Evaluate and iterate

Consider future potential when evaluating data product performance.

Evaluating results after a launch to make a go or no-go decision for a data product is not as straightforward as for a simple UI tweak. That's because the data product may improve substantially as you collect more data, and because foundational data products may enable much more functionality over time. Before canning a data product that does not look like an obvious win, ask your data scientists to quantify answers to a few important questions. For example, at what rate is the product improving organically from data collection? How much low-hanging fruit

is there for algorithmic improvements? What kinds of applications will this unlock in the future? Depending on the answers to these questions, a product with uninspiring metrics today might deserve to be preserved.


Speed of iteration matters.

Data products often need iteration on both the algorithms and the UI. The challenge is to determine where the highest-value iterations will come from, based on data and user feedback, so teams know which functions are on the hook for driving improvements. Where algorithmic iterations will be central – as they generally are in complex recommendation or communication systems like Coursera’s personalized learning interventions – consider designing the system so that data scientists can independently deploy and test new models in production.

By fostering collaboration between product and business leaders and data scientists, prioritizing investments with an eye to the future, and starting simple, companies of all shapes and sizes can accelerate their development of powerful data products that solve core user needs, fuel the business, and create lasting competitive advantage.

Emily Glassberg Sands is the senior director of data science at Coursera.

This article is about PRODUCT DEVELOPMENT

 FOLLOW THIS TOPIC

Related Topics: DATA | TECHNOLOGY

Comments

Leave a Comment

POST

0 COMMENTS

✓ JOIN THE CONVERSATION

POSTING GUIDELINES

We hope the conversations that take place on HBR.org will be energetic, constructive, and thought-provoking. To comment, readers must sign in or register. And to ensure the quality of the discussion, our moderating team will review all comments and may edit them for clarity, length, and relevance. Comments that are overly promotional, mean-spirited, or off-topic may be deleted per the moderators' judgment. All postings become the property of Harvard Business Publishing.