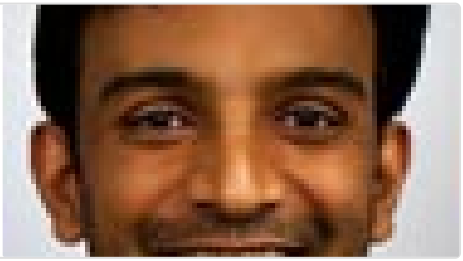
Data Jujitsu

Column	@August 9, 2021 3:25 PM
Due Data	@August 9, 2021
Readings	
Status	Done
Topic	🏆 Product Management

Data Jujitsu: The art of turning data into product

Having worked in academia, government and industry, I've had a unique opportunity to build products in each sector. Much of this product development has been around building data

http://radar.oreilly.com/2012/07/data-jujitsu.html



There’s a method to solving data problems that avoids the big, heavyweight solution, and instead, concentrates building something quickly and iterating.

Smart data scientists don’t just solve big, hard problems; they also have an instinct for making big problems small.

We call this Data Jujitsu: the art of using multiple data elements in clever ways to solve iterative problems that, when combined, solve a data problem that might otherwise be intractable.

jujitsu: “the art or technique of manipulating the opponent’s force against himself rather than confronting it with one’s own force.”

a good definition of a data product is a product that facilitates an end goal through the use of data. It’s tempting to think of a data product purely as a data problem

the fundamental idea is that you shouldn’t solve the whole problem at once. Solve a simple piece that shows you whether there’s an interest

example. A LinkedIn profile includes a tremendous amount of information. Can we use a profile like this to build a recommendation system for conferences? The answer is “yes.” But before answering “how,” it’s important to step back and ask some fundamental questions:

a. Does the customer care? Is there a market fit? If there isn't, there's no sense in building an application.

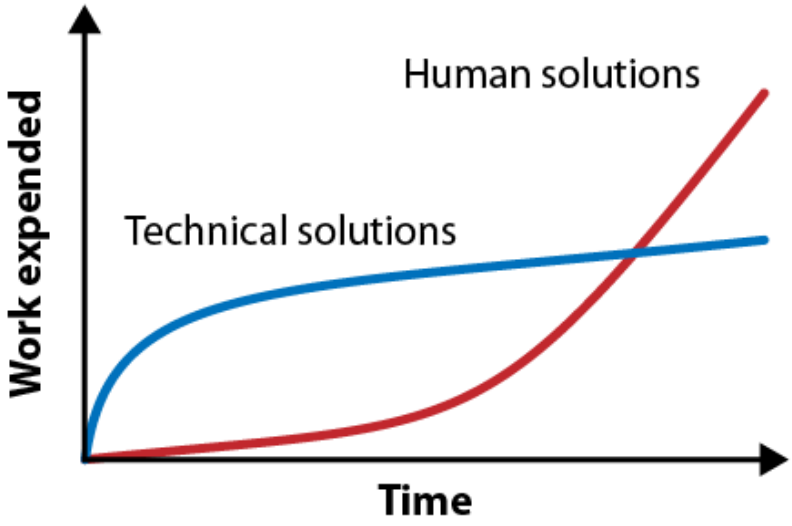
- How long do we have to learn the answer to Question a.?\

Instead, we could build a much simpler system. The key is to start simple and stay simple for as long as possible.

Use product design

One of the biggest challenges of working with data is getting the data in a useful form. It's easy to overlook the task of cleaning the data and jump to trying to build the product, but you'll fail if getting the data into a usable form isn't the first priority

When in doubt, use humans



The point is that technical solutions will always win in the long run; they'll always be more efficient, and even a poor technical solution is likely to scale better than using humans to answer questions.

But when you're getting started, you don't care about the long run. You just want to survive long enough to have a long run, to prove that your product has value.

And in the short term, human solutions require much less work. Worry about scaling when you need to.

Be Opportunistic for wins

I've stressed building the simplest possible thing, even if you need to take shortcuts that appear to be extreme. Once you've got something working and you've proven that users want it, the next step is to improve the product, and create derivative or similar products

Ground your product in the real world

Use analogy between real world and virtual world. The more your digital product resembles the real world product the more successful it is.

eg. People you might know in linkedin

Give Data Back to the user to create additional value

By giving data back to the user, you can create both engagement and revenue.

We're far enough into the data game that most users have realized that they're not the customer, they're the product.

How do you give data back to the user? LinkedIn has a product called "Who's Viewed Your Profile." This product lists the people who have viewed your profile (respecting their privacy settings, of course), and provides statistics about the viewers.

No Data Vomit

As data scientists, we prefer to interact with the raw data. Most of your customers can't do that.

Giving back too much data in a way that's overwhelming and paralyzing is "data vomit."

it's very easy to overestimate the abilities of your users. The product you want may not be the product they want.

The best way to avoid data vomit is to focus on actionability of data. That is, what action do you want the user to take?

Expect unforeseen side effects

unforeseen side effects aren't a joke. One of the best examples of an unforeseen side effect is "My TiVo Thinks I'm Gay."

Outsmarting the recommendation engine once it has "decided" what you want is difficult and frustrating, and you stand a good chance of losing the customer.

it's hard to out think an intelligent agent that has gotten the wrong idea.

Improving precision and recall

Precision — The ability to provide a result that exactly matches what's desired. If you're building a recommendation engine, can you give a good recommendation every time? If you're displaying advertisements, will every ad result in a click? That's high precision.

Recall — The set of possible good recommendations. Recall is fundamentally about inventory: Good recall means that you have a lot of good recommendations, or a lot of advertisements that you can potentially show the user.

an important issue in product design is the tradeoff between precision versus recall.

If you're working on a search engine, precision is the key, and having a large inventory of plausible search results is irrelevant. Results that will satisfy the user need to get to the top of the page. Low-precision search results yield a poor experience.

On the other hand, low-precision ads are almost harmless (perhaps because they're low precision, but that's another matter). It's hard to know what advertisement will elicit a click, and generally it's better to show a user something than nothing at all. We've seen enough irrelevant ads that we've learned to tune them out effectively.

Subjectivity

Another issue to contend with is subjectivity: How does the user perceive the results?

Ask and you shall receive

Take heed not just to demand data. You need to explain to the user why you're asking for data; you need to disarm the user's resistance to providing more information by telling him that you're going to provide value (in this case, more valuable recommendations), rather than abusing the data.

It’s essential to remember that you’re having a conversation with the user, rather than giving him a long form to fill out.

Anticipate failure

data products can fail because of relevance problems arising from the tradeoff between precision and recall. Design your product with the assumption that it will fail. And in the process, design it so that you can preserve the user experience even if it fails.

You can give your data product a better chance of success by carefully setting the users’ expectations.

One under-appreciated facet of designing data products is how the user feels after using the product. Does he feel good? Empowered? Or disempowered and dejected?

In many applications, a design treatment that gives the user control over the outcome can go far to create interactions that leave the user feeling good.

Facebook uses this design technique when they show you an ad. They also give you control to hide the ad, as well as an an opportunity to tell them why you don’t think the ad is relevant.

Putting Data Jujitsu into practice

Data Jujitsu embraces the notion of the minimum viable product and the simplest thing that could possibly work.

Data Jujitsu is all about saying “no” to our inner Rube Goldberg. Rube Goldberg Machine Contests

It’s impossible to overstress this: 80% of the work in any data project is in cleaning the data. If you can come up with strategies for data entry that are inherently clean (such as populating city and state fields from a zip code), you’re much better off