Data Council '19

Building a Lean AI Startup
Lessons learned

or How to start an ML company in your garage

Paul Cothenet
Co-founder & CTO

MadKudu

MadKudu is a Lead & Account Scoring platform that enables B2B companies to build relevant customer journeys at scale

Drift

IBM

Segment

shopify

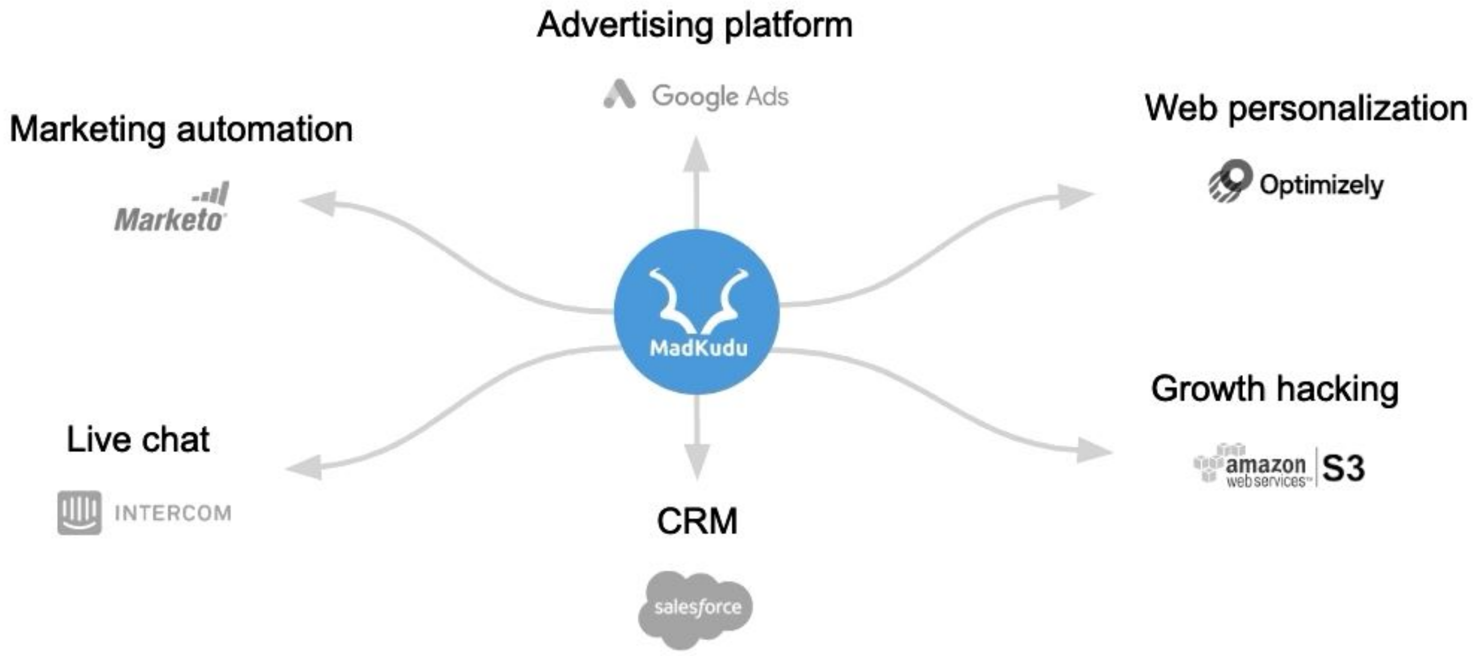
GUSTO

invision

About MadKudu

- "Machine Learning for Sales and Marketing"
- Used by the sales and marketing team at **InVision**, **Shopify**, **Segment**, **Drift**, **IBM**, **Avalara**, **Freshworks...**
- Our assumption: If you're lucky to have good enough Data Scientists and Data Engineers, employ them on what makes your product unique, not on your sales and marketing funnel

Lead scoring everywhere



Before that

- 2 (then 3) data scientist / product managers
- No funding for a year
- Working from an actual garage

What this talk is

- Why being lean is hard in data
- Practical lessons learned building an AI product
- Practical tools and techniques we used
- Focus on the product/engineering, with a side of go-to-market

Target audience:

- Early stage (or aspiring) entrepreneurs
- Data Scientists / Engineers looking at launching new products

Lean Startup vs. DS/ML/AI

The goal is still the same (especially if you don't have a lot of $$)

What's different with AI?

Source: The Lean Startup (http://theleanstartup.com/principles)

What do you need to prove?

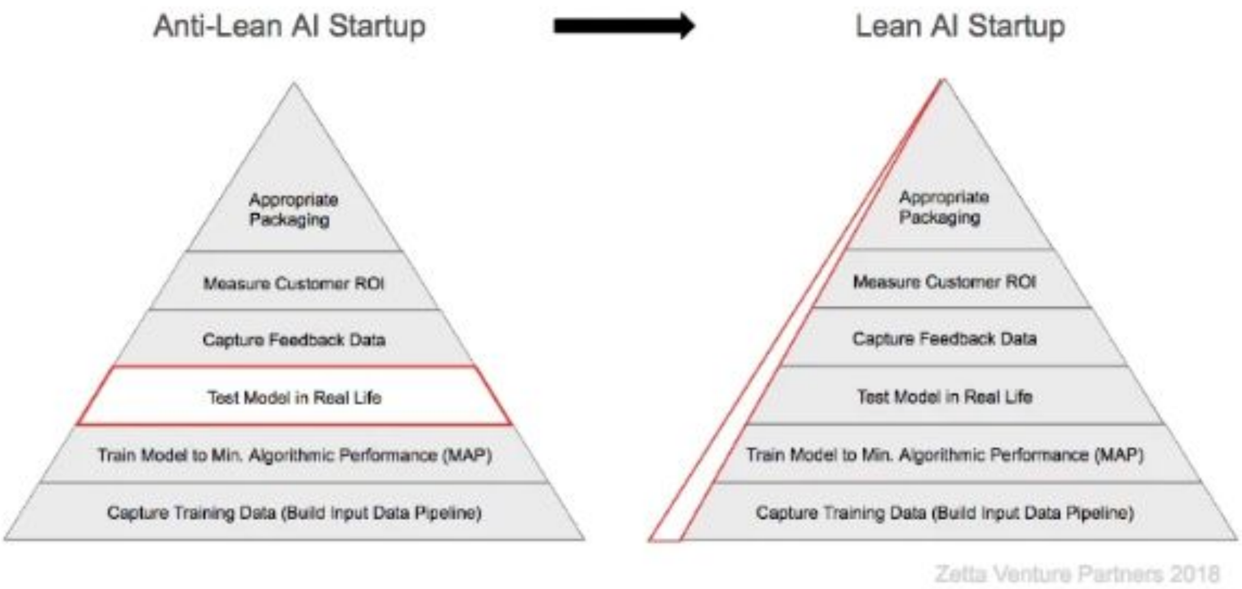
You need to prove that:

1. Your problem is well-suited for AI
   a. You can collect the right data
   b. You can predict with "minimum accuracy"

2. There is a market for your models ("model market fit")

3. You're solving the problem correctly over time

The framework I wish I had seen



(source: Zetta Venture Partners https://venturebeat.com/2018/08/18/the-ai-first-startup-playbook/)

Step 1: Get data!

Step 1: Get data!

You need data:

1. To prove that the problem is solvable (aka your model is predictive)

2. To get feedback (mock-ups won't get you there).

But how do you get customers to initially trust you with their data?

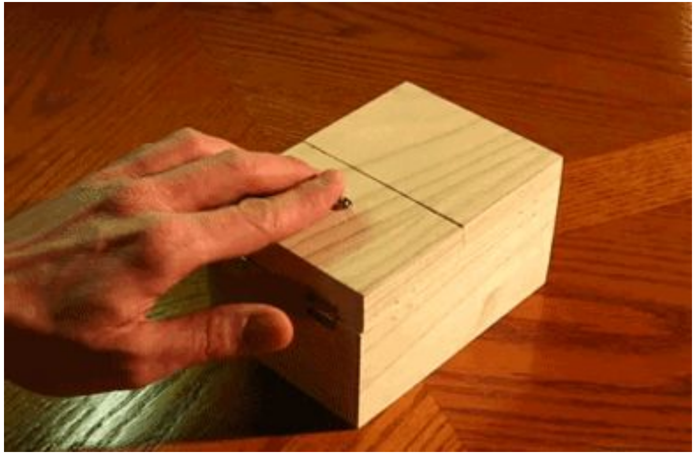
Step 1: Get data!

What worked for us:

- **We spent most of your initial engineering effort making it as easy as possible for customers to send us data.**
- **We partnered with existing data repositories**
- **We asked for slightly more data than we needed**
- **Find potential customers of the right size**

Get data! - Smoke and mirrors

Make it stupidly easy for customers to send you their data (and do the rest manually)



- Our first endpoint was a node.js API dumping data into SQS with a small aggregator to S3 (also in node.js)
- Our first Salesforce "integration" would only save credentials to the database (and we would use it manually behind the scenes)

Get data! - Find the right friends

(If you can) find the right data partners:

- Focus on partners of the right size (that will let you integrate without having to demonstrate your value first)
- Avoid: partners that ask you to demonstrate value upfront (in our case, Marketo, Eloqua...)
- If no partnership possible, ask your customers for their API key

Get data! - Pack the leftovers

We asked for slightly more data than what we thought we needed at the time:

- That let us iterate later (see obstacle 2)
- Not too much so it will prevent customers from giving you access
- Make sure to use your early engineering efforts to adequately protect the data.

Step 2: Get predictive!

Get predictive! - The trap

If you're a data scientist, you will probably spend too much time here

...

Even if you know it's a trap

Get predictive! - The trap

A couple dead-ends we got stuck in for too long

- Trying to predict churn
- Trying

Why?

- Not because we couldn't predict, but because it didn't matter
- We didn't have "Model-Market Fit"
- We didn't go fast enough to the "Capture Feedback Data" and "Measure ROI"

Get predictive! - Minimum predictiveness

- Find the simplest model that seem to do better than the current case scenario ("Minimum Algorithmic Performance)
  - (in our case: trees and regressions)

- Before increasing complexity of the models
  - Can you increase the size of the datasets
  - Can you supplement with other sources of data to increase dimensionality

- Shut up every cell of your brain that tells you to worry about scalability/modularity

Step 3: Get real!

Get real! - Simplify your stack

What is the mininum you can do so you can get feedback?

What worked for us:

- Exclusively SQL + CRON
- Full-refresh first, no incremental
- No real-time, no streaming

Figure this out real-time and incremental only when the need arises

Get real! - Simplify your stack

You probably do not need:

- Spark
- Kafka
- ...

Step 4: Get feedback!

Get feedback!

A mistake we made:

- "Now we're serving the algorithm, can we move on to the next customer?"

Get feedback!

- Ask for customer's $$ early
- Start by presenting your results with a Powerpoint deck
- Serve your model where your customer is going to use it
- Can you embed in your customer's process?

Get feedback!

Listen to all the feedback:

- If the customer has doubts about the prediction (very frequent in lead scoring), they won't use it. The math might say otherwise but they won't use it.

Recommendation:

- Don't fear overriding your model with manual heuristic in order to get to the next objection

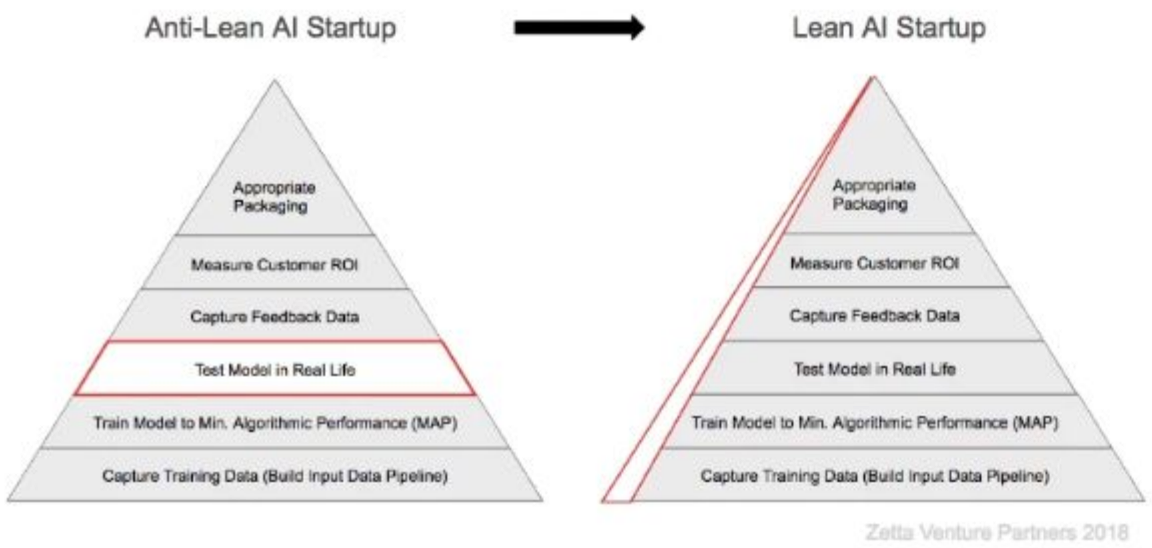
Step 5: Get returns!

Step 5: Get returns!

Honestly, that one is super hard. I can't say that we've found generalizable recommendations yet.

Step 6: Get back!

Get back - Iterate rapidly



Congratulations: it works for one customer, what do you next?

Get back! - Iterate rapidly

What didn't work:

- Create structure and abstractions too early

What did work:

- Erring on the side of the spaghetti

- Rule of three:
  wait until you've done the same
  thing for 3 clients before doing
  any kind of abstraction

Bonus lesson: Team organization

At least one founder that has experience in AI/ML

- Very very hard to get the desired iteration speed if outsourced (or even first hire)

For us:

- Two founders with background in ML
- One with experience in data pipelines and Data Engineering

If you have to make a tradeoff

- Founder has ML expertise (most interaction with customers)
- Hire the Data Engineer

MadKudu

Good luck!

PS: If your company is still making you work on lead scoring, please come talk to me during Office Hours!

paul@madkudu.com
@paulcothenet

Appendix

References

Things I really wish I had read before getting started:

https://machinelearnings.co/why-ai-companies-cant-be-lean-startups-734a289792f5

https://machinelearnings.co/the-ai-first-saas-funding-napkin-2cb138070ffc

http://mattturck.com/the-power-of-data-network-effects/

https://venturebeat.com/2018/08/18/the-ai-first-startup-playbook/

https://techcrunch.com/2018/03/27/data-is-not-the-new-oil/