ML in real life

- Unless you work for an ML company, ML is small cog in a complex machine.
- The most important thing that distinguishes a exceptional ML engineer from a very good one is the ability to understand business context.
- Business context is not easy to understand: it takes time, curiosity and non-trivial soft skills.

The Reasonable scale

- The reasonable scale assumption:
  - *If you are not at FAANG, there is a good chance you are working for a commpay at Reasonable Scale.*
- RS does not mean you work for a small company. It means that you work for a company that is subject to a number of constraints that provide the context for your work as a ML engineer.

Constraints often define where you go

Major constraints:

- Financial impact of ML
- Team size and resources
- Data volumes
- Sheer computing costs

Understand the subject you’re working on

- The most common mistake I saw is to start developing something right away.
- Real problems are different from Kaggle competitions.
- Business objective are fuzzy and full of hidden assumptions.
- Data is practically always missing.
- The real world presents some unmovable constraints.

Understand the subject you’re working on

- Understand exactly what we are trying to achieve.
- Come up with a good way to measure it.
- Identify the unmovable constraints:
  - If you’re lucky there is a hack somewhere.
  - If there isn’t at least you saved everybody a lot of trouble.
- See if you have the data you need to start building at least a POC.
- Now you can start developing for real.

An example: real time personalization for ecommerce

- Initial set up:
- We are a b2b company selling recommendation as a service.
- Our clients are businesses that have a website, they want good recommendations, but they are not really up to building them themselves. So they turn to us to provide them a good recommendation engine that they can plug in their website.

An example: real time personalization for ecommerce

- One fine morning, the CEO comes along and says that our product needs to provide personalization for ecommerce recommendation, because:

  “This is what people want”.

What do we mean exactly?

- Although personalization as a value proposition might seem clear to our beloved CEO, it is not really clear what it means.
- From a product perspective the main questions:
  - What are we trying to achieve?
  - Can we measure our progresses?

Fleshing out the request

- There is a qualitative aspect to it: differentiation.
  - The experience of using this product is visibility different from usual.
  - This helps creating a narrative that the company and the sales team can use to promote and sell the product.
- But we also need some quantitative way to measure how we are doing.
- Luckily enough we do have a way to think about Recommendations performances in KPIs:
- Behavioral KPIs:
  - Clicks rate
  - Engagement
  - Bounce rate
- Business KPIs:
  - Conversion rate
  - Average order value

Fleshing out the request

- The goal then is:
- **To build a recommendation engine that personalizes the experience of different users.**
- It needs to be easy to show that personalization is provided.
- It needs to be ideally better than what we provide now - or at the very least it must not be worse: that is KPIs must be affected positively (or worst case scenario not be affected at all).

First look at the data with an SME

- Personalization can mean several things:
- User information:
  - Email address
  - Shopping history
  - Browsing history
  - Loyalty program
  - Geospatial information
- If we could build a **user profile** we could use all these information in our recommendation system to provide more personalization.
- For instance, we could recommend products similar to what the user bought in the past, or belonging to categories that the user is interested in, etc.

Do we have the data?

- Target client for this product:
  - B2C e-commerce websites. E.g. https://www.famousfootwear.com/
- We should have their product catalog and we can track what users do on the website:
  - Clicks
  - Product views
  - Purchases
  - Etc.
- So, ideally we can build a user profiles with the user history and preferences and use those as features for our model.

Movable constraints

- However, looking into the data a bit more thoroughly, it appears that:
  - We do not collect all data from browning. We only have page views, but we do not have the purchases.
- This is a problem for our plan of building a user profile, because we would be missing a very important piece of it: what users actually purchased.
- **We can do something about it:** we can ask our clients to provide us also the purchase data.
- Our plan of building a user profile with all the history and the preference of the users is still within reach.

Unmovable constraints

- However, looking into the data a bit more thoroughly, it appears that:
  - Most b2c ecommerce website only have anonymous users.
- This fact is outside of our control: **there is nothing we can do about them** (i.e. we do not have a way to identify the users through time).
- Our initial plan of building a user profile with all the history and the preference of the users to provide personalization is no longer unattainable.

Hack it until you make it

- We still have data about user browsing anonymously.
- We could approach this problem from a different perspective:
  - Instead of building a user profile, we build a representation of the user browsing session and we try to personalize based on that.
- We build an session-based recommender system.

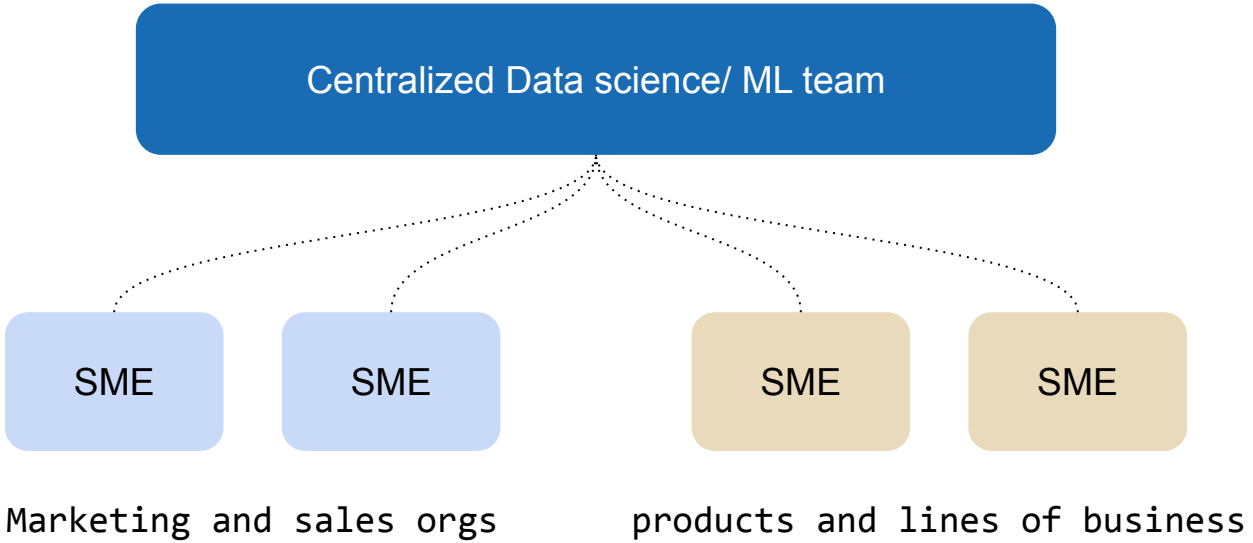
The importance of thin slicing: prototypes should always be end-to-end when possible

vs

Teams organization: centralized

Pros:

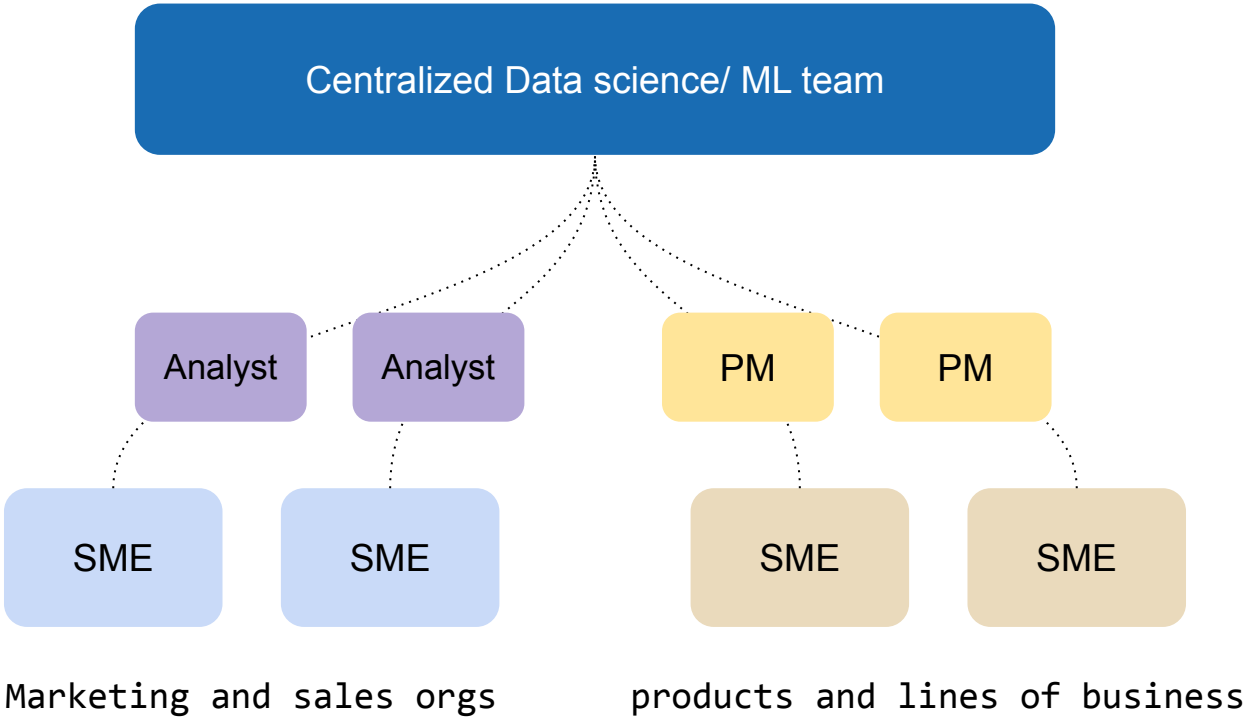
- Teamwork and collaboration.
- Closer feedback loop with your manager.
- Clearer career path.



Teams organization: centralized

Cons:

- Progressive detachment from product and business units.
- Overbureaucratization of the development process.
- Narrower and sometimes shallower skill set.



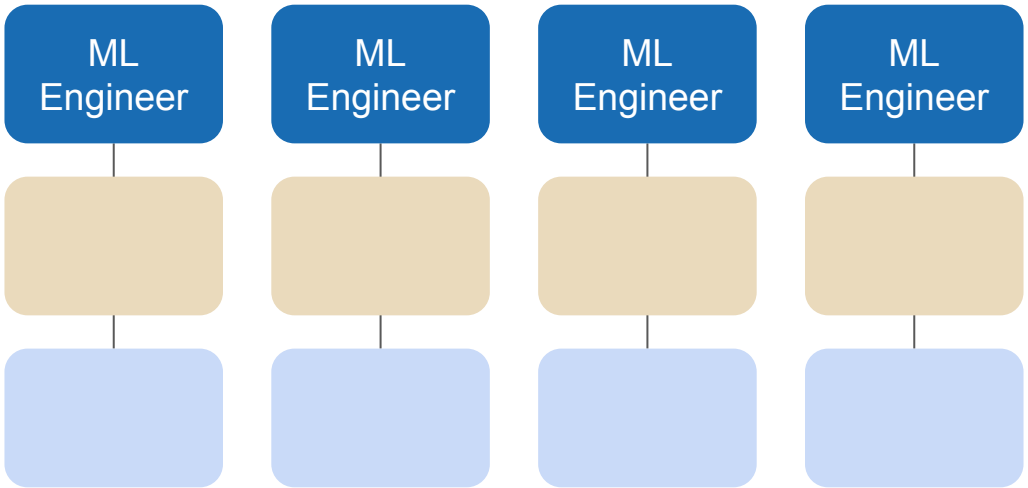
Teams organization: vertical

Pros:

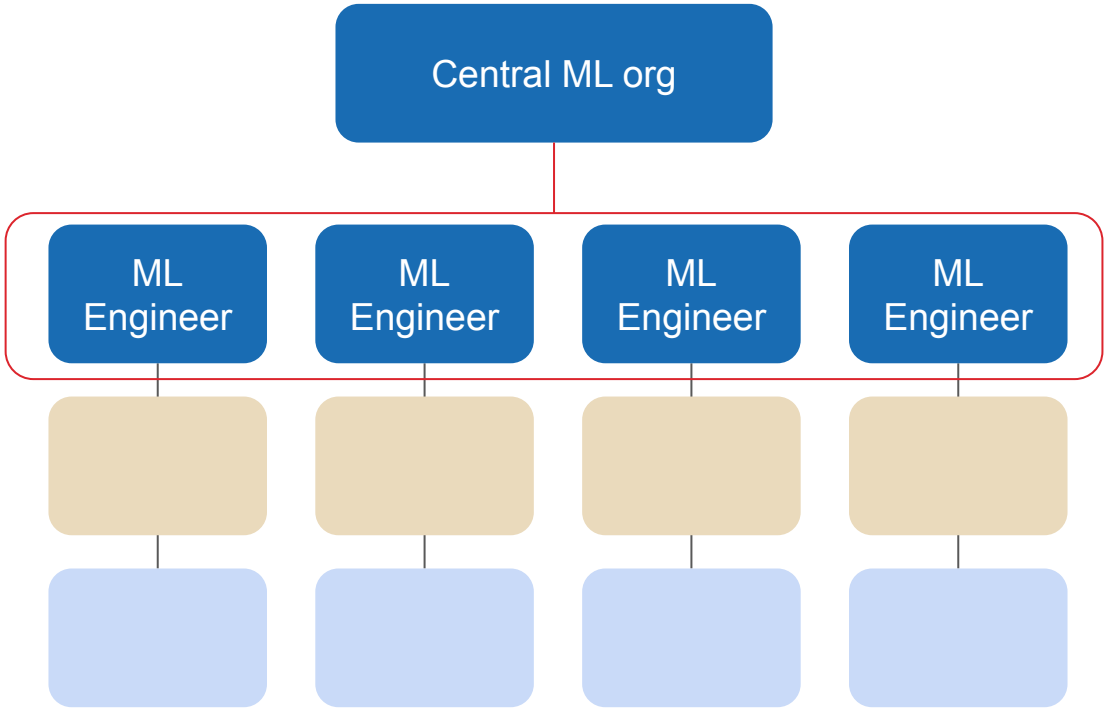
- Faster iteration.
- Higher business impact.
- Concentration of domain expertise.

Cons:

- Isolation.
- Unclear career path.
- Bad reporting structure.



Teams organization: Hybrid



Marketing and sales orgs

products and lines of business