**How to create products that rely on machine learning**

**A framework for product teams to balance the particular risks with machine learning and increase the likelihood of developing successful products**

*Creating products that rely on technologies such as machine learning comes with different considerations, risks and constraints than normal products. To succeed in your product efforts, the development process should acknowledge these inherent challenges and face them head-on — even though this may not sit well with your development team*



Photo by [Austin Distel](https://unsplash.com/@austindistel?utm_source=medium&utm_medium=referral) on [Unsplash](https://unsplash.com?utm_source=medium&utm_medium=referral)

You’re really excited about the new project at work. You are working with a team of your company’s strongest engineers and scientists to develop a cool new product, which at its core is using a bespoke machine learning approach to deliver the outcomes. To make it concrete it could be a tool determining when to restock your retail stores and automatically planning the truck transportation, or it could be a new app helping your customers preview how your furniture would look in their home before buying, an automated fault-inspection system for your company’s production line or something completely different.

Everybody starts working on the project. Lots of technical design and development work. Three months in, the data scientists are working on a complicated new deep learning architecture and are having trouble making it work. The engineers have created a pretty bland app but don’t really know how to progress before the model is ready. In the mean time, management is asking for status and the energy in the team, once very high from the excitement, is dropping as frustration takes over.

With much delay, the pilot product is eventually launched, and yet somehow the intended users are not really picking it up. Management is not happy, and neither are the technical teams. Several of the engineers and scientists feel like a failure, and some have even experienced that their status has dropped within the company.

If you have experienced anything like the above scenario, you’re not alone. **Successfully building products which are using machine learning [1] as a core technology *is different* than other digital product initiatives.** The indicators are many: the failure rate of “AI projects” is quoted as being significantly above 50 % (although hard numbers prove difficult to obtain) [2,3], going beyond a pilot or a proof of concept has proved challenging for many [4,5], and overall retention of AI talent is a big challenge [6]. While the technology itself has gone from exciting to pretty mainstream over the last decade — with many developments improving accessibility — there is still not a shared understanding among product folks and technical folks on how to approach machine learning product development, even though there seems to be some understanding that the technical requirements and risk profile of machine learning product development is different than normal digital products, and that some adjustments are needed.

**A key reason for failed product efforts is that data scientists and product folks do not share the same understanding of the risks and disagree on the order in which to address these.** Being primarily concerned with the “core intelligence”, data scientists will intuitively focus on the applied R&D of model development to explore technical feasibility; this is a time-consuming and exploratory effort which often progresses non-linearly. Product leaders, on the other hand, will be more concerned with creating strong and user-friendly value propositions in a way that works for the company (consider the combination of value, desirability and viability together with the technical feasibility [7]) and will want to iterate quickly to explore this. This is also a non-linear discovery process, just with different control knobs and success criteria. And often, nobody is thinking about how to ensure that the right data at the right quality can be sourced consistently and at a speed that is required by the new product.

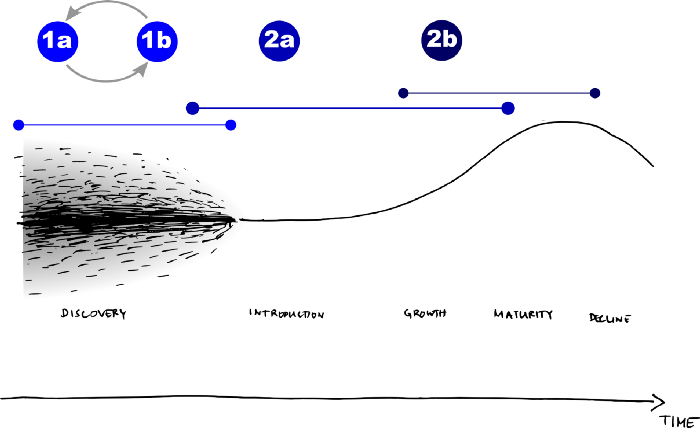
Discovering the right products that work for both the user and the company while constantly balancing and minimizing the four risks of value, desirability, viability and feasibility is always hard, even for products based on familiar technologies. However, these four high-level risks have some particular manifestations for machine learning-powered products which are perhaps not obvious. These do not replace the four big risks, they are just key points among those four that require particular attention. These are

* **Spending too little time early on model feasibility assessment: Technical feasibility of machine learning is limited by data and skill.** Improving the machine learning model afterwards is easy if the foundation is strong, but impossible if the basics are not in place. It will kill your whole product if the model you hypothesize is scientifically impossible to develop. You should therefore follow the #monkeyfirst principle [8]: do the hard things first, and if they are impossible, you must evaluate if another approach is feasible or abandon the product idea.   
    
  The limiting factors for machine learning are a) do you have consistently and timely access to the right data, and b) do you have the skills to develop and deploy the right models, and it is therefore imperative to establish the right level of confidence in these aspects early in product development.
* **Spending too much time on the machine learning model early (before validating the product).** Improving model performance is time-consuming but it may be a significant waste of time unless it has been validated that the product we are giving to our users also improves from the effort. So, whereas the first risk was about making sure to assess deep enough if the machine learning is likely to work, the second risk is to over-invest too early and disregard the user’s needs (risks of value and desirability).
* **Disgruntled data scientists.** Data scientists really like to work on machine learning models, and given the choice, they would prefer to work on more advanced types of models that allows them to grow their skills and experience. On the hand, many data scientists are not as strong as the software engineering counterparts in building the systems to deploy the machine learning models, and data scientists also often feel that they should not be doing that work.   
    
  If the first two risks above are taken to heart, data scientists would justifiably be concerned about whether they get to use their core competencies. There is time for that once the product MVP has proven value to users. Nonetheless, this can lead to frustration and high turnover unless addressed early and appropriately.

Coming to a shared understanding of the desired outcome and the risks of getting there, and agreeing to a development methodology which allows the team to address these is key. What is often forgotten by technical teams is that the intended outcome is a great user experience solving important, valuable problems — working machine learning technology itself is not enough. In fact, handling exceptions and how the user is to take action on the machine learning predictions are keys to success but often sit entirely outside the machine learning technology itself. It can be hard to be super crisp on the risks and challenges, especially early on when the team is still learning about the space of value propositions and technical challenges. Having an open conversation about both desired user outcomes and the development methodology early, and following this up with regular check-ins to reassess and adjust if needed is nonetheless crucial.

**A development approach for machine learning-powered products**

The approach which follows is an amalgamation of first-hand experiences combined and observations from other teams trying to strike a good balance. As in any other creative pursuit, you should not aim to have a perfect process but instead attempt to create great (“perfect”) outcomes, and respect that you will learn along the way: so, expect that this will at times get messy and that mistakes will be made. That being said, the hope is that the approach below will highlight how the most critical risks change through the development process and hopefully provide some perspective for both technical teams and product leaders.



*Reflecting the facts that the biggest risks are whether the product will resonate with users and the intended use of machine learning is feasible, steps 1a and 1b are used during discovery to address these risks while iterating towards a viable solution. Steps 2a and 2b focus on improving or extending the product once initial traction has been reached. Many different ideas are tested during the discovery phase, and the development can take many different outcomes. When an MVP has been developed, the course is much more narrow. Illustration by author.*

The approach is really just about recognizing which of the product risks mentioned previously that are biggest during which phase, and some suggestions for how to address those risks. Hopefully this should allow you to move faster and increase your confidence and likelihood of success as you are developing your new product. During the exploration of a new product that potentially could be using machine learning, two important risks are the technical feasibility and the desirability by the user of the proposed product, i.e. whether the product is solving an important problem in an appealing way (first two risks highlighted above). The best way to address this during product discovery is to be move with the right balance of speed and detail to identify a valuable product which appeals to users, and which is also technically feasible. These first two steps (1a and 1b, see figure) are about executing a suitable number of discovery/development cycles to find a strong product candidate that we have confidence we can build technically. The outcome of iterating through steps 1a and 1b is a minimum viable product (MVP) which we can launch to learn from.

The bulk of the technical development will happen in Step 2a which is all about maturing the product. This is where a lot of the detailed machine learning optimization, data pipeline optimization etc. will happen. Finally, Step 2b is the generic product aspect of expanding with more functionality.

**Step 1a: Is it feasible to address the problem with technology?**

When the product manager comes with an idea, it is the job of the technical team to assess whether it is at all feasible to address it through technology. Although you shouldn’t zero in on a particular technical solution too soon in early discovery, you still need to establish confidence that a technically feasible solution to the problem *can be developed*. Otherwise you wouldn’t know how to build the product, and you should shift focus to another way of serving the user need.

This is not to say “we have technology, now let’s look for a problem to solve”, but even when putting the user problem at the center, the technical capabilities of the team and the context of the problem jointly play big parts in the technical feasibility of delivery a solution.

Therefore, the aim with this first step is to think through the details of potential technology solutions at a sufficient level of detail while still being able to retain appropriate speed. Specifically this means assessing whether it is realistic *for your team and your product* to build the machine learning technology to deliver the required solution to the user problem (incl. some rough understanding of model performance), ensuring that the data for the machine learning can be consistently sourced for the product when the predictions are to be made and establishing feasible ways to measure performance.

As one important example, not being able to consistently get and clean the data needed for the model to run is often a big challenge because no path exists for how to acquire it. A frequent reason for this is models trained on static datasets, which fail when converted to products because insufficient thought given to how to take control of the data generating process [9]. Thinking through the feasibility of establishing a good data generating process already during the discovery phase is essential: not only this can save a lot of time, money and headache [10], but the absence of a data acquisition strategy will halt scaling and thus effectively kill you product before it takes off.

Feasibility assessment of machine learning models and data acquisition is typically done through technical proofs of concepts, but how you go about it can of course vary. However you do it, it’s about establishing confidence within the product team that these challenges can be addressed. While it could happen in some cases, I would be surprised if advanced mathematical modeling is performed at this stage; rather, it is about establishing whether the problem can be formulated in a way that could even be solved with mathematical modeling, establishing some early confidence that the model performance (accuracy, time to serve results, etc.) could be brought to the required level through fine tuning, and whether the required data material could be consistently available.

**Step 1b: Focus on proving the value proposition to users**

This second step is the user-facing parallel to Step 1a: product discovery, the process of identifying how to shape the product so that it is both valuable and appealing to users. I have named these steps 1a and 1b to indicate that they should ideally be addressed at the same time [11]. Expect that doing 1a and 1b will be iterative cycles towards product-market fit.

The goal is to explore the value drivers for users to understand how to design and build the product. That is, this is the process of identifying which ideas we believe in to develop a MVP to test with users, all with the purpose of identifying a strong value proposition. A key question is concerned with identifying the actual value drivers of the machine learning in the eyes of the users, including how to present the outcome of the machine learning in a way that allows the user to take the right action as easy as possible. Details will of course be very dependent on the context, but a good example is offered by car navigation systems: to be successful, such systems need to be able to deliver a good plan via GPS (machine learning and optimization) and timely verbal updates (interpretation of results relative to movement) are the keys to success. The former requires good planning capabilities and decent calculation speed, while low latency for the latter to constantly be able to assess where the car is relative to the plan is the difference between a useful and a useless product.

The key to success at this stage is being able to operate with speed in assessing whether ideas will work, and the signal we’re looking for is whether we’re giving users what they really need. The upshot is that this is the time for simple, off-the-shelf machine learning models (or perhaps a mocked model, i.e. no actual model at all!); once it’s been established that the problem can be solved with machine learning (the concerns of Step 1a have been addressed), there’s no more work to be done.

However, as part of exploring possible value-propositions, it often relevant to explore how to present the results to users. That is, which metrics best convey the message to users? How should the results be presented (in a graph? As a recommended action?) and should the user be able to explore the results interactively (e.g. move around in the map for car navigation system in the aforementioned example). What is the relative importance of accuracy versus time to serve result? What is the right granularity level to show the results at? A GPS that can only guide you from city to city, but doesn’t have resolution to guide you within the cities, is not particularly valuable to most. All these questions sit at the interface of product and data science and finding good solutions will make or break the final product.

**Step 2a: Maturing the product: Improve machine learning performance and value delivery**

When you’ve discovered a product that’s both valuable, desirable and feasible, the science and engineering development work can begin to improve the machine learning capabilities. This is the detailed improvement work where the solution is hardened, and where new machine learning architectures and model types are thoroughly explored, edge cases are dealt with etc. To be clear, maturing the technology and user experience applies to all aspects of the product, not just the machine learning; in what follows I will only focus on some aspects specifically important for the machine learning parts which are foundational for scaling and longevity.

In this phase monitoring and retraining strategies are developed from the rough sketches from Step 1a so that they support the product, a field also know as MLOps [12–14]. Most machine learning — as well as other types of applied mathematical engines — will deteriorate over time, and MLOps is what ensures that we monitor the performance and proactively adjust the models so they continue to deliver for the product to work. Without MLOps, the machine learning models will quickly stop producing useful results and the product will be useless. Product managers should take an active role in defining MLOps strategies together with the technical team to ensure the user value proposition is the primary focus, and key machine learning performance metrics should be part of the KPIs reviewed by the product manager daily.

One part of MLOps requires special mention because the best solution is often through product design: data deterioration. It sometimes happens that one or several of the data sources start to shift after the product has gone live. In marketing situations, this can happen when a new product is introduced that was not previously part of the dataset which shifts behaviors (could even be the introduction of our own product), it could be new regulations (e.g. GDPR), and in other contexts this could result from the deterioration of a physical sensor, changes in context in the images (mess in the background) of a production line, etc. MLOps will allow for proactively addressing changes in data, but it will not guarantee that the quality of the machine learning model is retained. The best way to address this is to design the product in such a way that you are in control of the data generation, or alternatively that the product is positioned in such a way that the value proposition does not deteriorate if sometimes the model cannot be run. Determining the strategy for addressing these issues is a product question, and it should be an active choice.

**Step 2b: Evolve product, but be on the lookout for unintended accidents from models gone awry**

Before discussing the details of this section, some aspects of machine learning models needs to be clarified. As part of the development process, these models are “trained” (fitted) on historical, representative data to tailor their predictive capabilities to the specific problem and data you are working on. However, performance typically drops if it is applied outside the domain of data the model was trained on, and this can happen both if there are changes in the data generating process (e.g. different user behavior) and in how we measure it (e.g. reading one data source using a different sensor). These aspects have some consequences for how to scale your machine learning-powered products.

If things are going well, the product will at some point require scaling. Scaling the serving and operations of machine learning models is a technical challenge (that can be challenging in its own right) as long as it happens within the same user base, but scaling into new user groups or segments comes with additional complexity because the data generating process could change. In the situations, close monitoring is often a good strategy: since machine learning models are often used to help with decision making, and since the decisions and/or behaviors of new user groups may be different than the previous segments, these changes will reflect in the data and performance monitoring. If there is reason to believe the new target user group has different behavior, or if parts of the data generating process is different for this new group, it should be expected that adding this new group will require tweaking the machine learning model itself (essentially going through steps 1a, 1b and 2a) and not just flipping on the switch of the existing model on this new segment. As an example within the GPS case we’ve used before: adding the UK as a new country will require significant tweaking of the routing engine because all routing decisions are affected by driving on the left side. The key callout is for the team to be aware of this important work, and factor it in at the right point in time.

As with any product, once we have a product launched with good traction and scale, the “final” step is continuing to evolve the offering through adding more features. This involves continuous cycles of discovery, prototyping and user validation towards identifying product-market fit for each new feature, as is well-described elsewhere [7]. Some of these new features could be new machine learning-powered solutions that have no technical relation to the core offering, and if so, their development would follow the steps outlined above. Here it is important to note that any new features that changes user behavior will also affect (potentially undermine) existing machine learning models powering legacy features. As an example, HR at one global energy company scored the likelihood of future success with the company of each new hire, but at the same time used this score together with performance and other variables to determine which employees would be offered growth opportunities; because of this (likely unintended) negative feedback loop, people who were not scored high when joining were not receiving the same opportunities, resulting in an overall less diverse talent pool and higher employee turnover. It is important to be on the look-out for these non-intuitive dependencies, and it is also important to be aware that these may be found only in some segments or may even vary across segments. In addition to awareness of the risk itself, diligently measuring the impact on user behavior and performance of existing models and doing controlled experiments when introducing new features are best practices to avoid unintended subversion of your previous success.

**Acknowledgement**

A thorough review by Simon Kiilerich Vedel (no relation) including ideas which have significantly improved this post is happily recognized ([LinkedIn profile](https://www.linkedin.com/in/simonkv/)).

**References**

[1] What I describe in this post applies to all cases using advanced mathematical modeling as the premise for solving problems, and is not limited only to machine learning problems (or even to traditional data science problems). That is, similar challenges will be found in mathematical modeling, optimization, and related areas. I will use the term machine learning throughout for brevity, but the reader should remember that the topics discussed apply more broadly.

[2] <https://venturebeat.com/2020/12/16/the-future-of-ai-deployments-reaching-production-is-bright-in-2021/>

[3] Irving Wladawsky-Berger: Why Some AI Efforts Succeed While Many Fail, Wall Street Journal, January 24, 2020, <https://www.wsj.com/articles/why-some-ai-efforts-succeed-while-many-fail-01579901883>

[4] <https://www.mckinsey.com/industries/semiconductors/our-insights/scaling-ai-in-the-sector-that-enables-it-lessons-for-semiconductor-device-makers>

[5] <https://www.bcg.com/publications/2019/travel-time-push-ai-beyond-pilot-phase>

[6] <https://www2.deloitte.com/content/dam/Deloitte/us/Documents/technology-media-telecommunications/us-tmt-how-do-you-retain-your-data-scientists.pdf>

[7] Marty Cagan: Inspired, <https://svpg.com/inspired-how-to-create-products-customers-love/>

[8] Astro Teller: Tackle the monkey first, 2016 <https://blog.x.company/tackle-the-monkey-first-90fd6223e04d>

[9] Will Douglas Heaven: Google’s medical AI was super accurate in a lab. Real life was a different story. MIT Technology Review April 27, 2020, <https://www.technologyreview.com/2020/04/27/1000658/google-medical-ai-accurate-lab-real-life-clinic-covid-diabetes-retina-disease/>

[10] Andrew Ng: AI doesn’t have to be too complicated or expensive for your business, Harvard Business Review July 29, 2021, <https://hbr.org/2021/07/ai-doesnt-have-to-be-too-complicated-or-expensive-for-your-business>

[11] This post focuses on cases where proving the value to users is part of the challenge, since this is a prevalent situation faced from startups to enterprises. It should be noted, however, that there are pockets where the value proposition is clear and work processes/product development efforts are designed to cater to relying on machine learning, e.g. hearing aids and other embedded signal processing, options pricing and fraud detection in banks, etc. In those cases, the issues Step 1b seeks to mitigate are not as big of a concern.

[12] For specialists: I am using the term MLOps liberally so it also includes DataOps (the data equivalent of DevOps). There are important differences between these, but for the purpose of this blog post they are two sides of the same question of how to operationalize machine learning and data products at scale.

[13] <https://en.wikipedia.org/wiki/MLOps>

[14] Terence Tse et al., The Dumb Reason Your AI Project Will Fail, Harvard Business Review, January 8, 2020, <https://hbr.org/2020/06/the-dumb-reason-your-ai-project-will-fail>