# **Como gerenciar produtos de Machine Learning**

## Parte I: Por que gerenciar produtos de machine learning é tão difícil? E por que você deveria se importar?

Bastiane Huang

#### [Bastiane Huang](https://medium.com/@Bastiane?source=post_page-----386e7011258a--------------------------------)

#### [30/09/2019](https://towardsdatascience.com/how-to-manage-machine-learning-products-part-1-386e7011258a?source=post_page-----386e7011258a--------------------------------)

## Resumo: isso é o que eu quero que você lembre sobre esta série de artigos:

1. Gerenciar produtos de ML (Machine Learning) é mais desafiador do que gerenciais produtos de software normais porque envolve mais incertezas requer mudança não apenas técnicas, mas também organizacionais.
2. ML é mais adequado para tomar decisões ou fazer previsões.
3. Defina claramente o problema, escope os requisitos, defina as métricas e dê aos engenheiros e cientistas espaço e flexibilidade suficientes para explorar antes de decidir sobre o caminho a ser seguido.
4. Pense na sua estratégia de dados desde o primeiro dia.
5. A construção de produtos de ML é interdisciplinar. Pense além de ML.

Image for post



No meu [artigo anterior](https://medium.com/swlh/redefining-robots-demystify-next-generation-ai-enabled-robotics-fec64bfeb66c), falei sobre a maior diferença que o Machine Learning (ML) traz: **o ML permite um afastamento de ter que programar a máquina em direção a uma verdadeira autonomia (autoaprendizado)**. As máquinas fazem previsões e melhoram os insights com base em padrões que elas identificam nos dados, sem que humanos lhes digam explicitamente o que fazer. É por isso que ML é especialmente útil para problemas desafiadores que são difíceis de explicar para as máquinas. Também significa que ML pode fazer seus produtos serem mais personalizados, mais automatizados e mais precisos. Algoritmos avançados, dados massivos e hardware barato estão permitindo que o ML se torne o principal driver do PIB.

A adoção de ML tem avançado rapidamente em vários setores de negócios. Quase metade das empresas incorporaram uma ou mais capacidades de inteligência artificial nos seus processos e outros 30% estão pilotando projetos de IA, de acordo com a pesquisa recente da [Mckinsey](https://www.mckinsey.com/featured-insights/artificial-intelligence/ai-adoption-advances-but-foundational-barriers-remain). Não é difícil entender por que se espera que o ML seja ainda mais transformador do que a tecnologia móvel. No entanto, a transição para ML também pode ser mais do que 10 vezes mais difícil do que a transição para mobile. Antes de falarmos sobre o porquê disso, vamos analisar o básico.

# O básico de Machine Learning

## O que são IA e ML?

Não há uma definição de IA universalmente aceita e a definição muda o tempo todo. Uma vez que uma tarefa é realizada por uma máquina, essa tarefa já não está mais no escopo de IA. ML é um subconjunto de IA. CMU professor Tom M. Mitchell defined Machine Learning to be a study of computer algorithms that allow computer programs to automatically improve through experience.

## Tipos de ML

Existem três tipos principais de machine learning:

* **aprendizado supervisionado**: O tipo de aprendizado mais comum e o mais utilizado. Os algoritmos aprendem com dados rotulados, ou seja, dados de treinamento que foram marcados com o resultado que o modelo está tentando prever. Em resumo, é sobre prever resultados.
* **aprendizado sem supervisão**: por outro lado, algoritmos de aprendizado não supervisionados aprendem a identificar padrões nos dados sem que eles tenham sido rotulados. Pode ser utilizado em problemas de agrupamento, de associação e de detecção de anomalias. Há também o aprendizado semi-supervisionado, que é essencialmente um híbrido entre a aprendizagem supervisionada e não supervisionada.
* **aprendizado por reforço (Reinforcement Learning)**: os algoritmos aprendem conforme recebem feedback sobre previsões. O RL é usado em domínios de controle como robótica ou carros autônomos.

## Tipos de Produtos de Machine Learning

Dependendo dos tipos de produtos e de onde vem o valor, você vai precisar de diferentes habilidades no seu time, e se concentrar em partes diferentes dos produtos.

* **Produtos B2C vs. B2B**

Produtos de ML B2C, como alto-falantes inteligentes, têm um componente social mais forte do que seus paralelos em segmentos B2B. Portanto, a experiência do usuário (UX) desempenha um papel mais crítico na concepção de produtos de ML B2C, e o ML tende a se tornar um fator facilitador para melhores experiências. Por exemplo, o NLP (processamento de linguagem natural) é usado para melhorar a interação entre o Alexa e seus usuários. Por outro lado, o valor para segmentos B2B, especialmente dos produtos de ML industriais, como softwares de manutenção preditiva, tende a vir do desempenho funcional (por exemplo, precisão) das suas previsões. Isso não quer dizer que UX não é importante para produtos de ML B2B. No entanto, isso é algo a considerar quando você tem recursos limitados e precisa focar na otimização de pedaços dos seus produtos.

* **Você está construindo um produto de ML ou aplicando ML ao seu produto?**

Se o valor principal do seu produto vem de modelos de ML, então você provavelmente está construindo um produto de ML. Por outro lado, se o ML é usado apenas para melhorar a experiência ou o desempenho do seu produto, então você provavelmente está aplicando ML ao seu produto. Neste caso, é essencial entender as entradas e saídas dos modelos, mas não os detalhes técnicos como a arquitetura ou se os modelos de ML são baseados em CNN (Rede Neural Convolucional) ou R-CNN. Por exemplo, o modelo usa dados demográficos dos usuários para prever os seus gastos mensais na plataforma. Muitas empresas ou times também aproveitam soluções existentes para não reinventar a roda. Por outro lado, a construção de produtos de ML geralmente exige que gerentes de produto sejam mais técnicos para ajudar o time a tomar decisões importantes e fazer trade-offs.

As estruturas das organizações também variam. Para empresas que constroem produtos de ML, ou grandes corporações com investimentos pesados em ML, como o Facebook e o Google, é comum contratar pesquisadores/cientistas de ML e pareá-los com engenheiros de ML. Por outro lado, para empresas que estão aplicando ML aos seus produtos, ou empresas menores com restrições de recursos, é provavelmente melhor contratar engenheiros de ML multidisciplinares ou treinar seus engenheiros de software para aprenderem ML em vez de contratar pesquisadores/cientistas de ML.

* **A construção de produtos de ML é frequentemente interdisciplinar.**

Mesmo que você esteja construindo um produto de ML, raramente ele vai envolver apenas ML. Isso geralmente é interdisciplinar e envolve não só modelos de ML, mas também engenharia de software, infraestruturas de back-end, análise de dados, design de UX/UI e às vezes até hardware. Gerentes de produto precisam ser capazes de gerenciar times multifuncionais e lidar com interdependências e potenciais conflitos entre times. O ML é fundamentalmente diferente de outras disciplinas, como explicaremos melhor no próximo parágrafo. Fica ainda mais complexo se você estiver construindo produtos de ML para o mundo físico, como robótica ou carros autônomos. Gerentes de produto precisam saber o que pode e o que não pode ser feito com ML, além de quando devemos e quando não devemos usar ML.

## Outros conceitos importantes de ML para entender

* **Overfitting**: é um tipo de erro que acontece quando os modelos são muito ajustados a um conjunto específico de pontos de dados. Modelos de ML robustos terão um bom desempenho não apenas em "datasets de treinamento", mas também em "datasets de validação". No entanto, no caso de overfitting, o desempenho em cima dos dados de treinamento aumenta mas a performance sobre os dados inéditos para o modelo (validação) piora.
* **Deep Learning (DL)**: é usado principalmente para classificação de imagens. O DL usa uma rede neural profunda e recebe imagens rotuladas como entradas. Cada camada da rede neural transforma a entrada em uma representação ligeiramente mais abstrata e composta. Eventualmente, o modelo aprende a reconhecer objetos nas imagens.
* **Natural Language Processing (NLP)**: um campo da ciência da computação para fazer as máquinas entendam as linguagens humanas. Não necessariamente envolve ML. O NLP é usado para chatbots, assistentes de voz ou pré-processamento de dados.

# Challenges in Managing ML Products

## 1. Experimentation is a crucial part of ML.

Just because ML involves code and data doesn’t make it similar to software engineering. In fact, the two disciplines couldn’t be more different. Unlike software engineering, developing machine learning products takes a lot more experiments and therefore involves more uncertainties and variabilities. Software engineering is a deterministic process of writing rules for machines to follow while machine learning is more probabilistic as it automates the task of writing the rules.

For example, if you want to teach a machine to recognize a cat. For example, if you want to teach a machine to recognize a cat. With software engineering, you may come up with rules like “a cat has 4 legs and 2 pointy ears.” But how is that different from a dog? If you use deep learning, instead of explicit rules, you will feed the machine with a bunch of cat photos (labeled images) and let the machine learn by itself. By doing so, you let machines write the rules by themselves. What you and your team do is to define the problem, prepare data, build a set of models, test, and iterate until you have a model that delivers desired results.

That’s why teams generally need to take more risks when developing ML products. It’s important for PMs to help set the right expectations to avoid potential clashes among teams. For instance, software engineers may feel that ML team is not giving them clear enough requirements without appreciating the nature of ML products. It’s also crucial to have engineers work closely with researchers/scientists so they can balance each other. More importantly, it’s better to have end-to-end systems working sooner to make sure that the algorithms that ML teams have been working on actually aligned with business goals.

## 2. Developing ML is a highly iterative process.

As mentioned before, ML is well suited for solving problems that are too complicated for humans to program explicitly. Models need to be trained, tested, and tuned. Often times scientists have to test a few approaches before choosing a satisfying one. That’s why it’s often more difficult to define milestones and estimate the timeline for ML products. Due to the nature of ML products, it’s critical for product managers to clearly define requirements and metrics and ensure that the team frequently test models against desired metrics.

## 3. There are more structural challenges beyond technical ones.

Because ML is so different from software engineering, it requires some fundamental organizational changes: experimental culture, data analytics-driven mindset, and more openness towards uncertainties, to name a few. Incumbents could face “innovator’s dilemma” if they treat ML as a purely technical problem and overlook the associated organizational changes. It is particularly challenging for companies such as robot makers who used to pursue high precision to develop ML products internally. In addition, ML products need large datasets for training. Companies need to build their own data pipeline and infrastructure to support the scaling of ML products.

## 4. ML is still a new field and it will keep evolving.

The term “software engineering” first appeared in 1965, 15 years after programming languages started to appear. Almost 20 years later, the Software Engineering Institute was established to manage the software engineering process. And today we have generally accepted best practices for software engineering. Machine learning, on the other hand, only started to flourish as a separate field in the 1990s. Deep learning, a subset of ML that has set new records in accuracy for many problems including image recognition and NLP, wasn’t widely discussed until the rise of AlexNet in 2012. Compared to software engineering, ML is still in its infancy and therefore lacks industry standards, metrics, infrastructure, and tools. Companies are still exploring best practices and kill applications.

## 5. Explainability and interpretability issues with ML products.

Many ML algorithms lack transparency, acting like a black box that takes input (e.g. images) and outputs predictions (e.g. what/who the objects/people in the images are). This makes it difficult for product managers to explain how ML models work and get buy-ins from users and stakeholders. Especially in critical domains like healthcare, accountability and transparency are extremely important. It’s challenging to ensure alignment between ML work and customer problems without a clear understanding of how an algorithm actually works.

With all these challenges, how should we go about managing ML products? Where do good PM instincts go bad for ML products? In Part II, I will talk more about my learnings and best practices.

## Connect Deeper: If you enjoy reading this article, please [subscribe to my personal blog here!](https://bastiane.substack.com/)

## **[How to Manage Machine Learning Products — Part II](https://towardsdatascience.com/how-to-manage-machine-learning-products-part-ii-3bdabf91eae4)**

### [Best practices and things I’ve learned along the way.](https://towardsdatascience.com/how-to-manage-machine-learning-products-part-ii-3bdabf91eae4)

#### [towardsdatascience.com](https://towardsdatascience.com/how-to-manage-machine-learning-products-part-ii-3bdabf91eae4)

*Bastiane Huang is a Product Manager at OSARO, a San Francisco based startup building software-defined robotics. She has worked for Amazon in its Alexa group and with Harvard Business Review as well as the university’s Future of Work Initiative.* [*She writes about ML, robotics, and product management. Follow her here*](https://bastiane.substack.com/).

*This post has been published on* [*www.productschool.com*](http://www.productschool.com/?fbclid=IwAR2D6uU6IAZmeNGLtBWs2POMn-pk9i8C7O4u24emInePlGJSyYIvgR7Mhfk) *communities.*