ML Productionization - The CEO Guide

Credits: Sherry Chen, on unsplash

Productionization

Productionization or operationalization of Machine Learning is the process of making machine learning models run every day, reliably, and integrated into data products. Standalone/adhoc development of models is not compelling anymore (they don't make nearly as much business sense). "*The majority (85%) of respondent organizations are evaluating AI or using it in production*" accordinging to "AI adoption in the enterprise 2020". This is the trend across industries, geographies, and scales.

Productionization has proven to be more difficult than people expected. Very few ML models reach production stage. **The main challenge is robustness**. According to the same report, *"Whether it's controlling for common risk factors—bias in model development, missing or poorly conditioned data, the tendency of models to degrade in production—or instantiating formal processes to promote data governance, adopters will have their work cut out for them as they work to establish reliable AI production lines."*

ML Platforms

All serious companies are building ML platforms to build models reliably and scalably. Uber has Michelangelo, Stripe has RailYard, AirBnB has BigHead and Swiggy has DSP. In fact, Uber shared their motivation for Michelangelo: *"there were no systems in place to build reliable, uniform, and reproducible pipelines for creating and managing training and prediction data at scale. Prior to Michelangelo, it was not possible to train models larger than what would fit on data scientists' desktop machines, and there was neither a standard place to store the results of training experiments nor an easy way to compare one experiment to another. Most importantly, there was no established path to deploying a model into production"*.

Google's engineers detailed out what these platforms achieve (Hidden Technical Debt in Machine Learning Systems) at around the same time (2017).

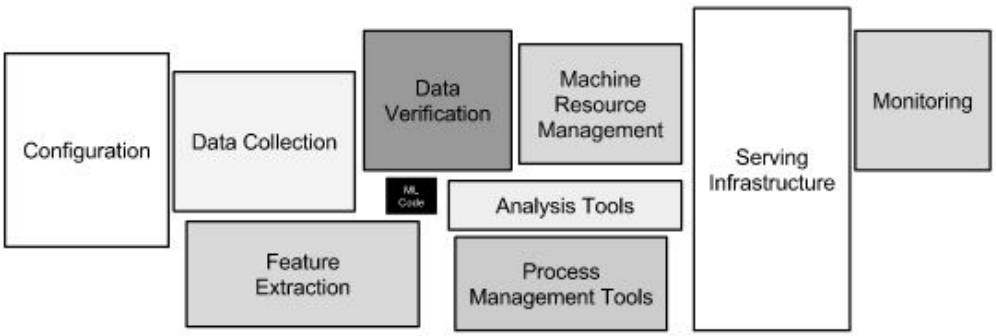

Figure 1: Only a small fraction of real-world ML systems is composed of the ML code, as shown by the small black box in the middle. The required surrounding infrastructure is vast and complex.

The structure of the problem remains the same, whether the model is simple or complex, and whether the development is happening in a small company or large. Platforms are being used to standardize and manage the process of development, deployment, and operation of machine learning models in order to achieve robustness and grow the use.

Uber shared their lessons learnt after multiple years of operation. A couple of them are:

A. Models need to be monitored: *"model monitoring and instrumentation is a key component of real world machine learning solutions"*
B. Data is the hardest thing to get right: *"data engineers spend a considerable percentage of their time running extraction and transformation routines over datasets"*

MLOps (ML Operations)

As every organization is trying to build/buy its own Michelangelo, the way that ML models are developed is changing. MLOps - devops for ML - is the new framing and is growing in importance. The major sub-areas for MLOps are:

1. **DevOps for Models** - Develop and deploy models (e.g., Domino Data)
2. **DevOps for Data** - Preparing and monitoring data (e.g., Tecton)

To this end (#1 - DevOps for Models), Domino Data Lab's workbench capabilities include:

A. Model development including A/B testing
B. Exploration of large datasets
C. Automatic tracking for reproducibility, reusability, and collaboration
D. Scalable compute and deployment management
E. Reports, dashboards, and API for model output
F. Data preparation
G. Integration with major compute platforms such as kubernetes and spark

And as for #2 - DevOps for Data, Tecton.ai is the hottest new company in the space. Their platform's capabilities includes:

A. Feature Pipelines for transforming your raw data into features or labels
B. A Feature Store for storing historical feature and label data
C. A Feature Server for serving the latest feature values in production
D. An SDK for retrieving training data and manipulating feature pipelines
E. A Web UI for managing and tracking features, labels, and data sets
F. A Monitoring Engine for detecting data quality or drift issues and alerting

Because both Scribble and Tecton are informed by the design principles behind Uber's Michelangelo, there is a high overlap between the functionalities offered on both platforms though they operate at different scales.

The ML Productionization Journey

Gartner, McKinsey and Others have articulated the challenges faced by organizations when they get on the ML journey. Here are a few recommendations for extracting business value from ML based on the industry consensus and our experience:

1. **Owning the ML solution process and outcomes**

    a. *Move to a new way of building systems.* ML models and systems are probabilistic in design and operation. Internalizing the uncertainty in ML is critical for success.
    b. *Accept that ML is NOT magic.* Making ML takes effort, often upfront. Increasing performance and accuracy is an iterative process requiring tools, experimentation, and processes that evolve with the context.
    c. *Recognize new risks and opportunities.* ML algorithms and data usability brings organizations into the purview of new privacy and algorithmic accountability laws directly and indirectly. It also enables companies to build new data products at a pace and with differentiation that wasn't possible before.

2. **Skilling for success**

    a. *Pick the right problems and approaches*. A lot of time is wasted by pursuing problems that don't have good RoI potential or that cannot be realistically solved with existing data. Mature teams invest in good problem selection, evaluation metrics, development process, and integration into product. Here, experience makes a difference.
    b. *Build end-to-end discipline*. ML is ultimately linear algebra or some other math. Correct operation of ML requires discipline in all phases of the lifecycle from planning and data collection to model operations. Organizations tend to narrowly focus on the model, ignoring the rest. Even the modeling phase is chaotic. Developing and enforcing discipline is a must.
    c. *Design for learning*. All ML models degrade over time (in fact, the degradation starts from the moment the training is over) and we learn over time what matters - data quality, corner cases etc. Continuous monitoring and improvement should be a core part of the design of any ML solution.

3. **Providing the right infrastructure**

    a. *Use tools for standardization and automation*. ML development and operational processes are iterative, laborious and error-prone. Cutting time and effort at every phase through standardization, simplification, validation, and automation helps.
    b. *Provide checks and balances*. The core value of ML is in the data and the algorithms. Risks to the organization include lost data, lost knowledge when staff leaves and decisions that can't be defended with clients/other stakeholders.. Tools that provide checks and balances during all phases of ML are critical to protecting the value created by ML for the organization.

A sample journey could be as follows:

1. Phase 1 (1 usecase): Select and put basic infrastructure in place and identify one usecase. Design from get-go for continuous usage, along with data and process discipline. Achieve transparency (everyone knows what is happening), reproducibility (repeated execution), predictability (standardize outputs, locations, servers etc.), monitoring (notifications etc.), and consumption interfaces (APIs)
2. Phase 2 (2-10 usecases). Generalize standards and processes by adding new usecases and evolving the compute and process to scale. Also create reusable datasets, processes, and assets.
3. Phase 3 (10+ usecases). Separate out teams to focus on specific phases of the ML. Design APIs, integration mechanisms, monitoring mechanisms etc.

There is an active debate on build-vs-buy across the industries. For a long time there was a strong preference for build, especially on the infrastructure side. What organizations are learning over time is that:

1. *The core value is in data ownership, good people, and end-to-end design*. Organizations are therefore freely discussing their solution design with no fear of loss of competitive edge. They are using transparency to attract good talent.
2. *Time is of the essence*. Product development cycles are shrinking across the board. Organizations are stitching complex solutions with available resources, and not waiting for the perfect product or approach.

3. *Infrastructure is very important but also expensive and time consuming*. Few organizations have the budgets of Uber and Google. It is the new database. Organizations are reducing their build approach here over time.
4. *Complex algorithms will not be easily built or bought*. The algorithm that won the Netflix recommendation prize was not put into production due to RoI considerations. Simplicity and careful thinking is winning over complexity. New requirements of explainability are also pushing organizations in this direction. Again, staff and modeling approach is critical to this.

Summary

The best companies, at every scale, today have understood the need to have the right people, processes and mechanisms by which they can reliably find ML usecases, build models, and use them in production deployments every day.

A thought-through approach (more time spent sharpening axe than the actual chopping of wood) to the ML lifecycle, including the internal processes, standards and tool choices, will allow organizations that are getting on the ML journey to be that much more efficient, and to build serious value internally as well as for their end-customers.

Scribble implements MLOps for data on its flagship platform, Enrich. We're based in Toronto and Bangalore. Our customers are in fintech, e-commerce, and edtech. Get in touch - hello@scribbledata.io