A Missing Link in the ML Infrastructure Stack

Josh Tobin
Stealth Startup, UC Berkeley, Former OpenAI
Machine Learning is now a product engineering discipline
Machine learning engineering is 10% machine learning and 90% engineering.

Yeah
How did we get here?

**ML analytics 2000s**
- Simple models run offline on medium to large datasets to produce reports
- Value comes from incorporating model insights into decisions

**ML hype 2010s**
- Complicated models trained on massive datasets to produce papers
- Value comes from marketing potential of high-profile research output

**ML products 2020s?**
- Reproducibility, scalability, and maintainability over complexity
- Value comes from models improving the business’s products or services
ML products require a fundamentally new process

Select problem → Collect data → Clean and label → Train → Report

“Flat-earth” ML
ML products require a fundamentally new process

Select problem → Collect data → Clean and label → Train → Report

Monitor ← Deploy ← Test

ML Product Engineering
ML teams that don’t make the transition die

Uber sells ATG self-driving business to Aurora at $4 billion

By Krystal Hu, Tina Bellon, Jane Lanhee Lee

Buzzy research lab OpenAI debuts first product as it tries to live up to the hype

By Jonathan Vanian

Montreal startup Element AI Inc. was running out of money and options when it inked a deal last month to sell itself for US$230-million to Silicon Valley software company ServiceNow Inc., a confidential document obtained by the Globe and Mail reveals.

Of the 250 industrial firms Plutoshift surveyed,

- over 72% found that they had taken far more time than anticipated to implement the necessary data collection processes for applying machine learning.

- and perhaps as a result, only 17% of those surveyed said they were actually at the full implementation stage of using A.I.,

- while about 70% said they were still studying what resources they’d need, assessing possible business use cases, or conducting small pilot projects only.

Worryingly, almost 20% of companies cited "peer pressure" as the reason they had embarked on A.I. projects.
What does it mean for you?

- Other disciplines will catch up to model training in prestige and pay
- The three Ps (papers, pie charts, PoCs) are no longer enough
Those that make the transition will create amazing things

- Autonomous Vehicles
- Real-time translation
- Drug discovery
- Marketing automation
- Personalization
- Document understanding
- Etc

Cannes: How ML saves us $1.7M a year on document previews
Unlike flat-earth ML, ML products often:

- Run online and in real-time
- Deal with constantly evolving data distributions
- Handle messy, long-tail real world data
- Make predictions autonomously or semi-autonomously
This implies new ops & infra demands

- Run online and in real-time
  Host and serve models with low latency

- Deal with constantly evolving data distributions
  Retrain models frequently, even continuously

- Handle messy, long-tail real world data
  Inspect your data scalable, manage slices and edge cases

- Make predictions autonomously or semi-autonomously
  Quickly catch and diagnose bugs and distribution changes
Is the infrastructure stack keeping up?

Training infrastructure
- Amazon SageMaker
- Determined AI

Experiment management
- W&B
- TensorBoard
- mlflow
- comet

Reproducible pipelines
- DVC
- Pachyderm
- DAGSTER
- KubeFlow

Select problem → Collect → Train → Report → Monitor
Is the infrastructure stack keeping up?

Select problem -> Collect data -> Clean and label -> Train -> Report

What’s still hard?
- Surfacing areas of poor performance
- Managing all your test cases

Model perf exploration
- W&B
- Explainability tools
  - fiddler
  - Arthur
  - truera

CI/CD tools
- GitHub Actions
- circleci
- Jenkins
Is the infrastructure stack keeping up?

- **Select problem**
- **Collect data**
- **Train**
- **Report**

**What’s still hard?**
- Experimentation (AB tests, shadow tests)
- Online / offline consistency

**Feature stores**

**Model serving**

Josh Tobin

A Missing Link in the ML Infrastructure Stack
Is the infrastructure stack keeping up?

What’s still hard?

- Performance monitoring
- Drift is still a bit of an art

Data quality / drift

- Deequ
- TensorFlow Extended
- great_expectations

System monitoring

- DATADOG
- Amazon Cloudwatch
- New Relic
Is the infrastructure stack keeping up?

Data lakes, warehouses

What’s still hard?
- Subsampling data
- Connecting the data back to the model

Collect data → Clean and label → Train → Report

Deploy → Test
Is the infrastructure stack keeping up?

**What’s still hard?**
- What data should I label?
- What data should I train on?

**Labeling tools & services**
- scale
- spaCy
- figure eight
- Labelbox

**Active learning tools**
- ScaleNucleus
- Aquarium

---

Josh Tobin
A Missing Link in the ML Infrastructure Stack
Is the infrastructure stack keeping up?

- Select problem
- Collect data
- Clean and label
- Train
- Deploy
- Test
- Monitor
- Report

**What’s still hard?**
- How do I know when to retrain?
- (Retraining online)
Takeaways

• Many tools emerging to address the problems of ML product engineering

• Problems arise at the boundaries of the tools, especially anything that shepherds data through the process

• At all stages, granular understanding of model performance is lacking
The Evaluation Store

A central place to store and query online and offline ground truth and approximate model quality metrics.
A Missing Link in the ML Infrastructure Stack

Josh Tobin

Eval Store

- Data and prediction profiles
- Metric & slice definitions
- Feedback on model predictions

Training

- Model hub
- Feature store

Production
Querying the evaluation store

What form do queries take?

- Subset of models in the store
- Subset of metrics in the store
- Subset of slices in the store
- Specification of the window of data
Querying the evaluation store

What form do queries take?  

- Subset of models in the store
- Subset of metrics in the store
- Subset of slices in the store
- Specification of the window of data

E.g.,

What is the importance-weighted average drift across all of my features in my production model in the last 60 minutes?

Monitoring
Querying the evaluation store

What form do queries take?

- Subset of models in the store
- Subset of metrics in the store
- Subset of slices in the store
- Specification of the window of data

E.g.,

How much worse is the my accuracy in the last 7 days than it was during training?

Monitoring
Querying the evaluation store

What form do queries take?

- Subset of models in the store
- Subset of metrics in the store
- Subset of slices in the store
- Specification of the window of data

E.g.,

How do all of the metrics compare for model A and model B across all slices in my main evaluation set?

Testing
Querying the evaluation store

What form do queries take?  E.g.,

- Subset of models in the store
- Subset of metrics in the store
- Subset of slices in the store
- Specification of the window of data

E.g.,

- How do my business metrics compare for model A and model B in the last 60 minutes
- AB testing
A digression: approximate performance metrics

• In a perfect world, we would know **right away** how well the model performs **on all data points seen in production**

• In the real world, labels are **unreliable, expensive, and delayed**

• Approximate performance metrics are ways to **guess** which data points may have poor performance

  • E.g., distribution distance between these data points and a reference distribution
  
  • E.g., outlier detection
  
  • E.g., weak supervision (a la Snorkel)
  
  • E.g., metrics about your users (like engagement)
The Evaluation Store

Select problem → Collect data → Clean and label → Train → Report

Monitor → Deploy → Test

Eval Store
The Evaluation Store

- Register data distribution and performance for this model
- Warn us if training data looks too different than prod
The Evaluation Store

- Select problem
- Collect data
- Clean and label
- Train
- Report

Eval Store

- Register performance for this model on all test slices

Test

- Pull historical that has been flagged as “interesting” (e.g., gave another model trouble)
- Pull definitions of slices
The Evaluation Store

- Log data and approximate performance back to the eval store
- Run a shadow test or AB test by pulling the diff in model performance between versions
The Evaluation Store

- Fire an alert when approximate performance on any of our slices dips below a threshold
The Evaluation Store

- Log more data with low or uncertain approximate performance

Select problem -> Collect data -> Clean and label -> Train -> Report

Monitor -> Deploy -> Test

Eval Store
The Evaluation Store

- Inspect & label data with low approximate performance

Select problem -> Collect data -> Clean and label -> Eval Store

- Train
- Report
- Select problem
- Inspect & label data with low approximate performance
- Monitor
- Deploy
- Test
The Evaluation Store

- Select problem
- Collect data
- Clean and label
- Train
  - Retrain when approximate performance dips below a threshold
- Monitor
- Deploy
- Test
- Report
What could an eval store help you with?

- **Reduce organization friction.** Get stakeholders (ML eng, ML research, PM, MLOps, etc) on the same page about metric and slice definitions

- **Deploy models more confidently.** Evaluate metrics and slices consistently in testing and prod. Make the metrics visible to stakeholders

- **Catch production bugs faster.** Catch degradations across any slice, and drill down to the data that caused the degradation

- **Reduce data-related costs.** Collect and label production data more intelligently

- **Make your model better.** Decide when to retrain. Pick the right data to retrain on.
Shouldn’t the feature store do this?

- Feature store is **indexed by feature**, eval store is **indexed by model**
- A model taking a feature as input **doesn’t mean** that it looks at the entire distribution
- A “**poor quality**” feature has **different effects** on **different models**
- **Not all data** will come through the feature store
- The two should talk to each other!
Wait, isn’t this just ML monitoring?

• Yes

• The hard part here is approximating how well your model might be performing right now

• That’s ML monitoring
Wait, isn’t this just ML monitoring?

- No

- Eval store should provide a consistent view of online and offline performance

- Eval store is tightly integrated into the entire MLOps stack

- Eval store keeps track of what data caused questions performance, so it can be used for testing and retraining
ML monitoring

Training → Evaluation → Production → Monitoring
Eval store

- Training
- Evaluation
- Production

Eval store
Case study 1: the Tesla data engine

youtube.com/watch?t=7714&v=Ucp0TTmvqOE
Case study 2: TFX data validation

Case study 3: Overton (Apple)

https://machinelearning.apple.com/research/overton
A Missing Link in the ML Infra Stack?

• To turn ML into a product engineering discipline, we need an infrastructure stack that helps create a data flywheel

• What’s still missing?
  
  • Granular, online-offline understanding of model performance
  
  • Orchestrating data and models throughout the whole loop

• Maybe the Evaluation Store could help