

Machine Learning Engineering

A few key concepts & some learnings



2002-2005



2005-2008



2008-2009

glhuilli.github.io



2009~2010



2010-2011



GROUPON

2011-2015



2015-2020



2020-...

Disclaimer

1. Personal comments (own perspective and biases).
2. I don't know everything/remember well (I'll be wrong in some details).
3. I'll skip important nuances as we have 40min, but please ask!

ML Engineering

Use software engineering principles
to build **robust** and **reliable** machine learning systems.

ML Engineering - Horror Stories

Amazon's one hour of downtime on Prime Day may have cost it up to \$100 million in lost sales

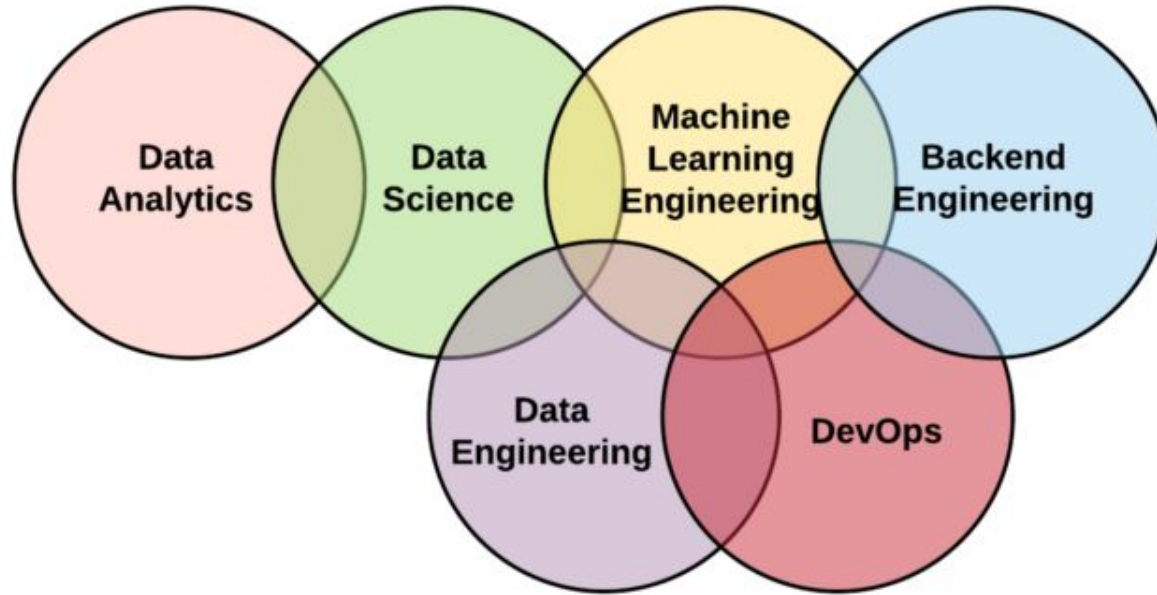
Sean Wolfe

Jul 19, 2018, 10:53 AM

“If their auto-scaling was working, things would have scaled automatically and they wouldn't have had this level of outage,” Caesar said. “There was probably an implementation or configuration error in their automatic scaling systems.”

[CNBC news](#)

Roles Landscape



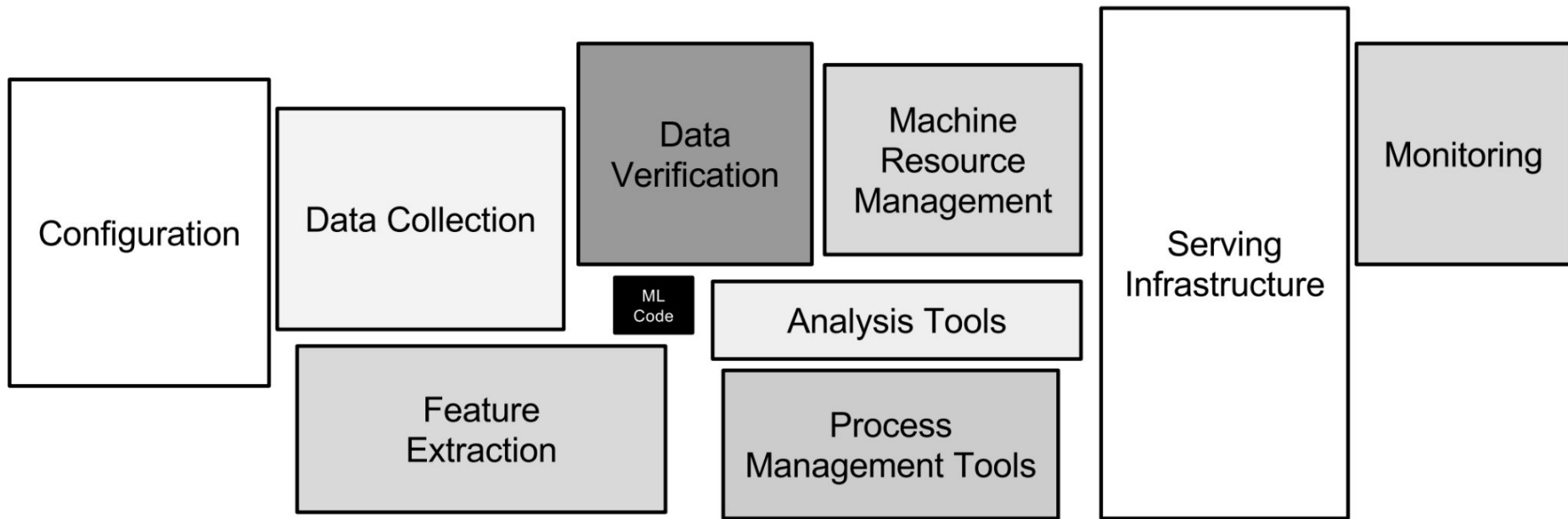
Source: [Medium post](#)

ML Engineering vs ML Research

	Research	Production
Objectives	Model performance	Different stakeholders have different objectives
Computational priority	Fast training, high throughput	Fast inference, low latency
Data	Static	Constantly shifting
Fairness	Good to have (sadly)	Important
Interpretability	Good to have	Important

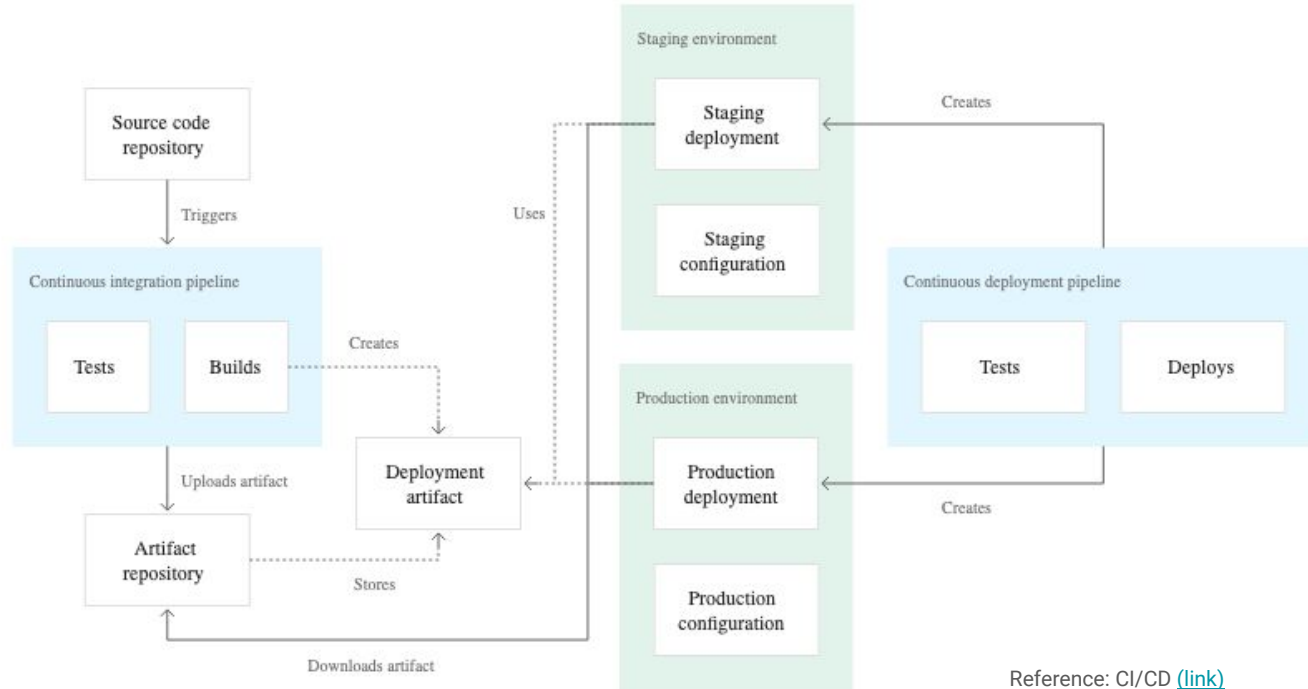
Source: Chip Huyen's ML Eng course ([course](#), [link](#) for content above)

ML Engineering is much more than just ML

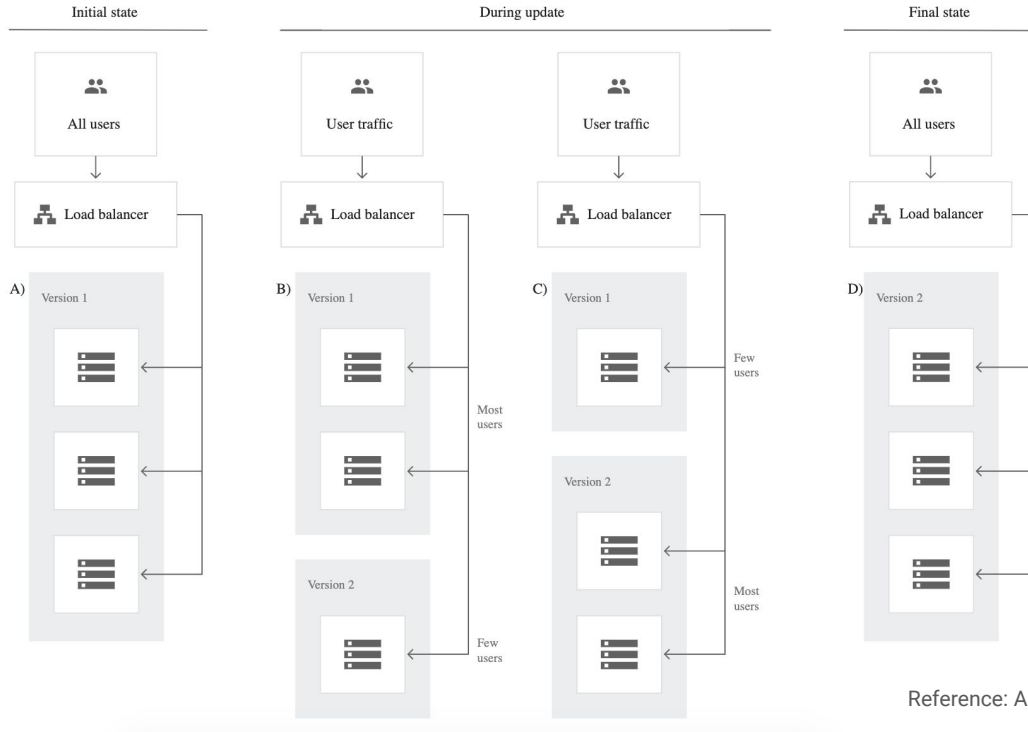


Scully et al., NeurIPS (2004) ([paper](#))

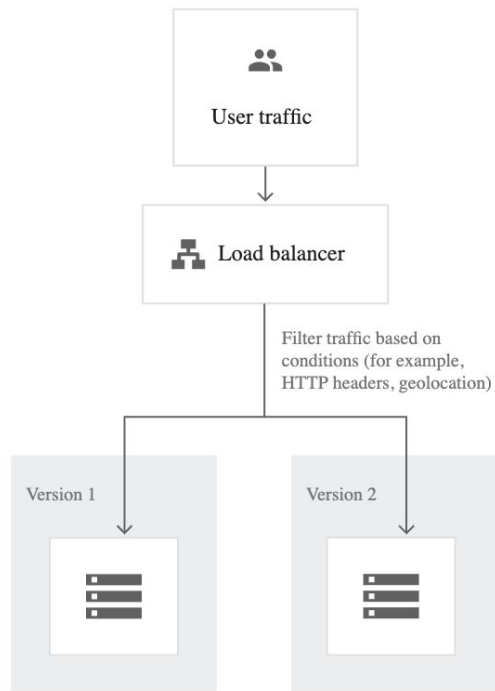
Production Systems Concepts (CI/CD)



Production Systems Concepts (Canary Testing)

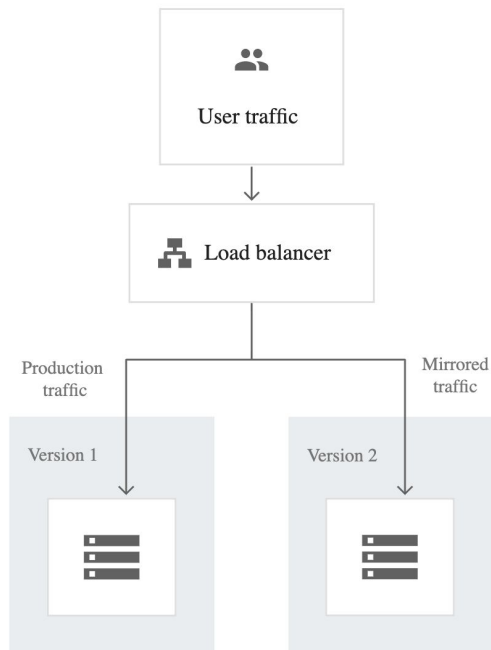


Production Systems Concepts (A/B Testing)



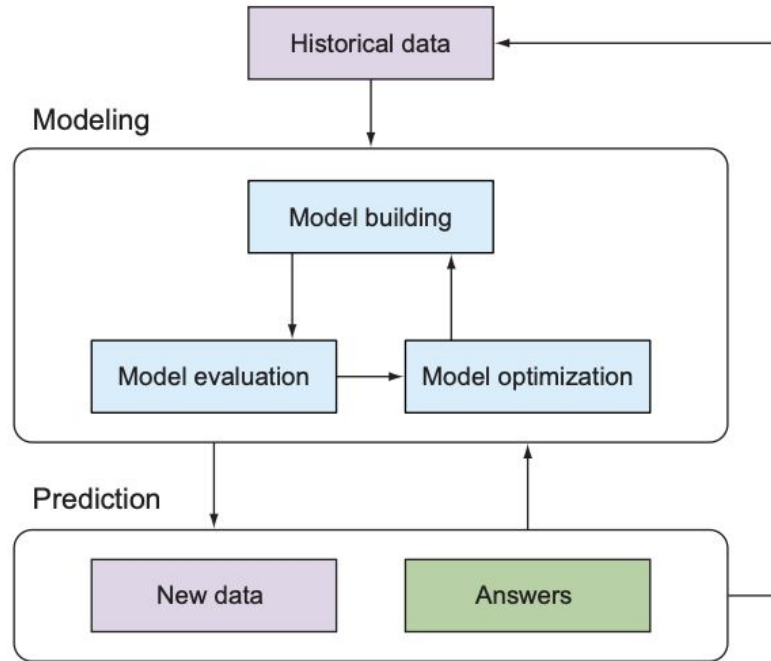
Reference: Applications deployment [\(link\)](#)

Production Systems Concepts (Shadow Testing)



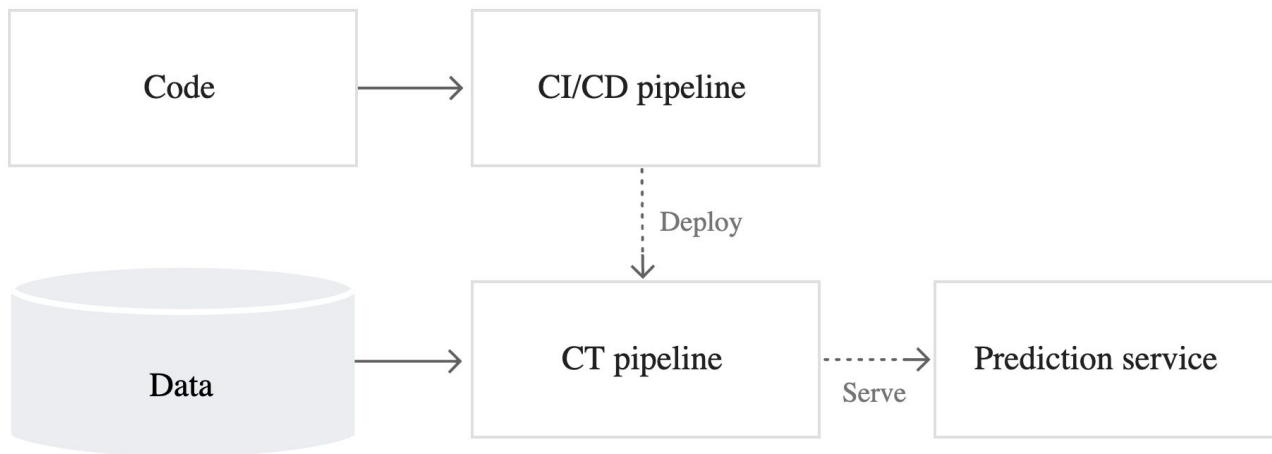
Reference: Applications deployment [\(link\)](#)

Simplest ML System



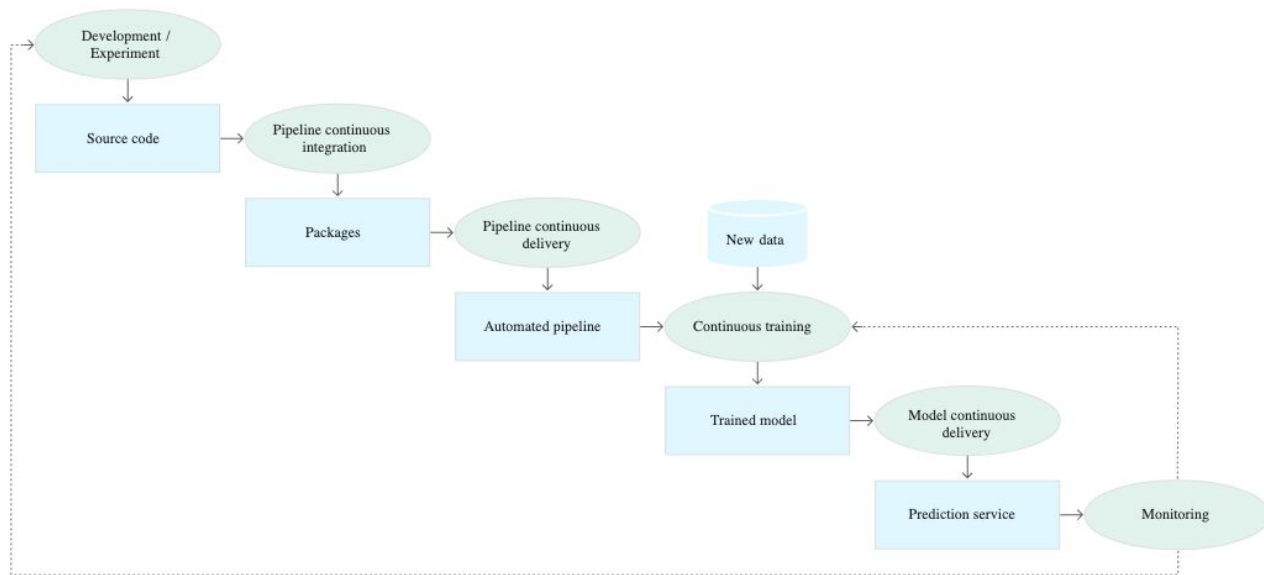
Real-world Machine Learning [\(link\)](#)

CI/CD + Continuous Training (CT)



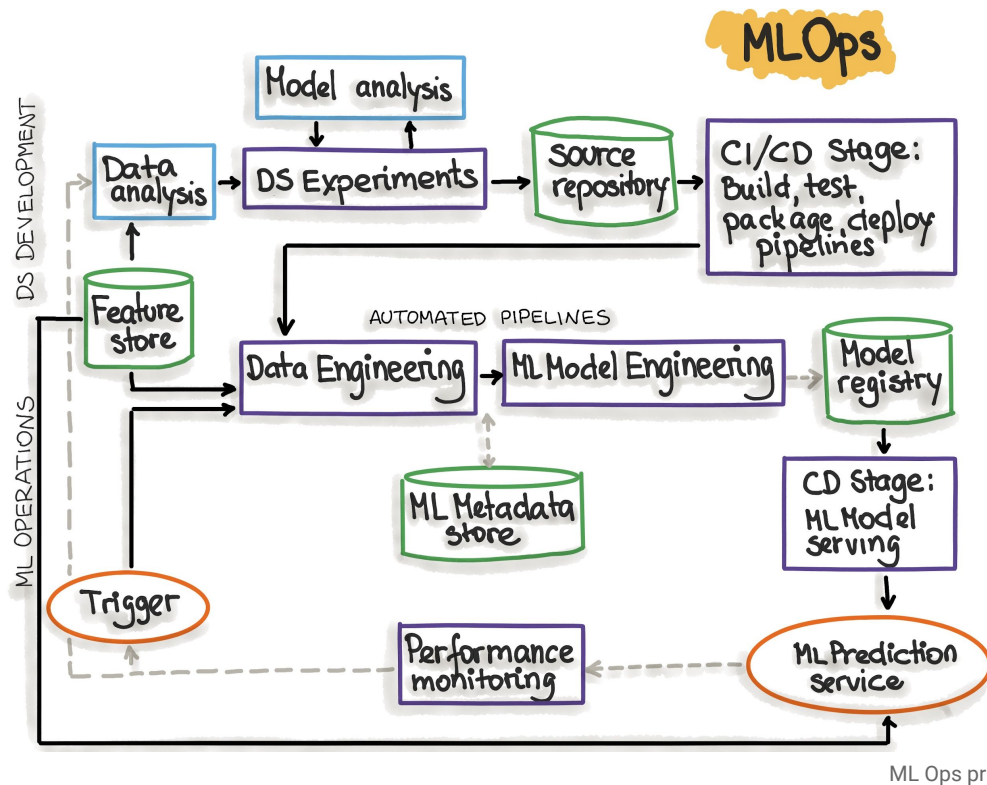
Reference: CI/CD/CT ([link](#))

CI/CD/CT + Model Continuous Delivery (mCD)



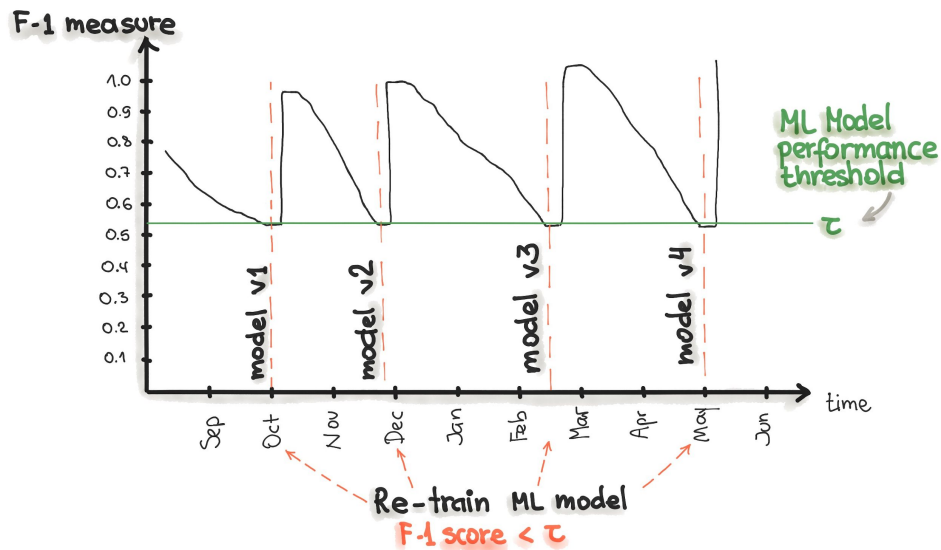
Reference: CI/CD/CT ([link](#))

ML Ops



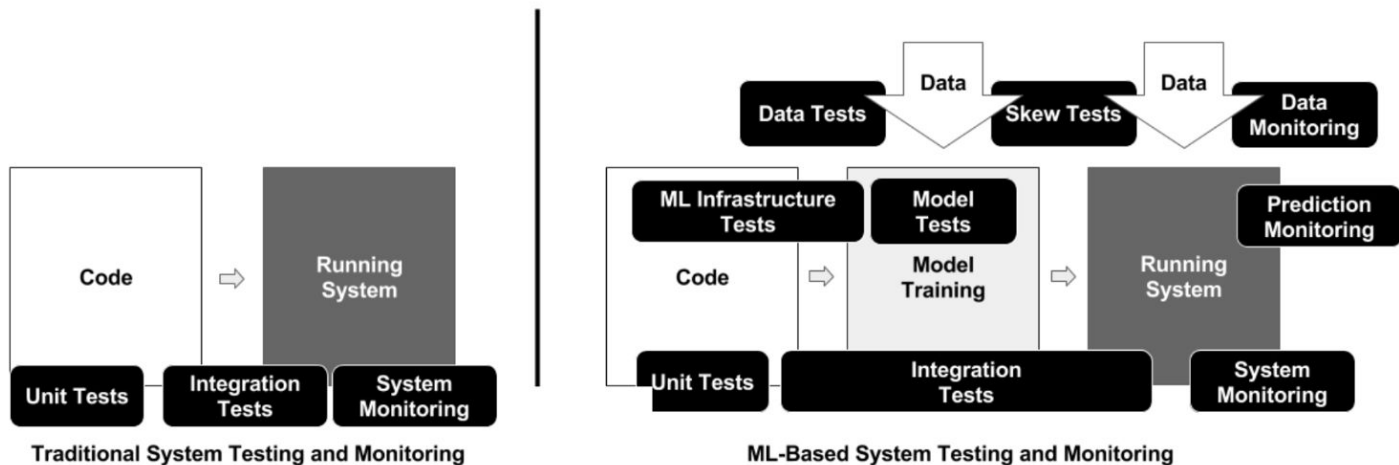
ML Ops

ML MODEL DECAY MONITORING



ML Ops principles [\(link\)](#)

ML Ops



ML Test Score, Breck, et al. (2017) ([link](#))

ML System Inference Paradigms

- Offline Inference
 - All possible predictions in batch
 - Feed predictions to cache/lookup table
 - Pros:
 - Cost/speed of inference not very important.
 - Post-verification on predictions on data before pushing
 - Cons:
 - Only predict what it's known
 - Update latency takes time

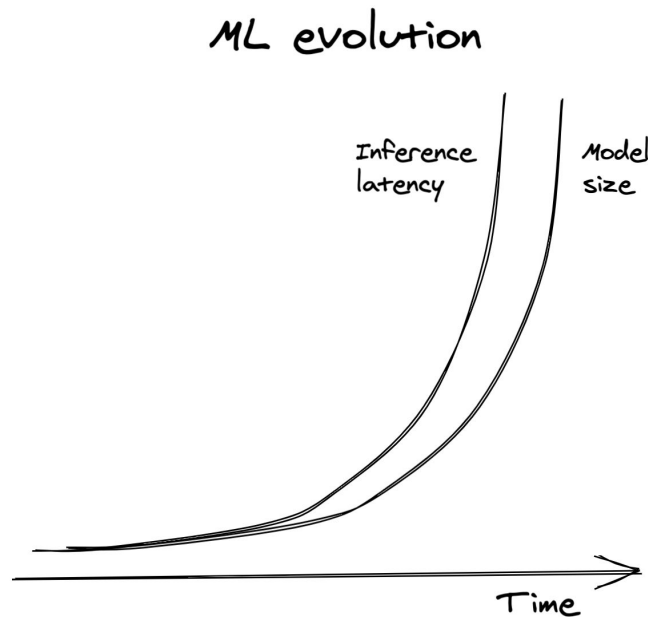
Reference: Google ML crash course [link](#)

ML System Inference Paradigms

- Online Inference
 - Predict on demand.
 - Feed predictions to cache/lookup table
 - Pros:
 - Predicts any item that comes in.
 - Cons:
 - Computational intensive, latency sensitive
 - Limits model complexity
 - Monitoring more intensive

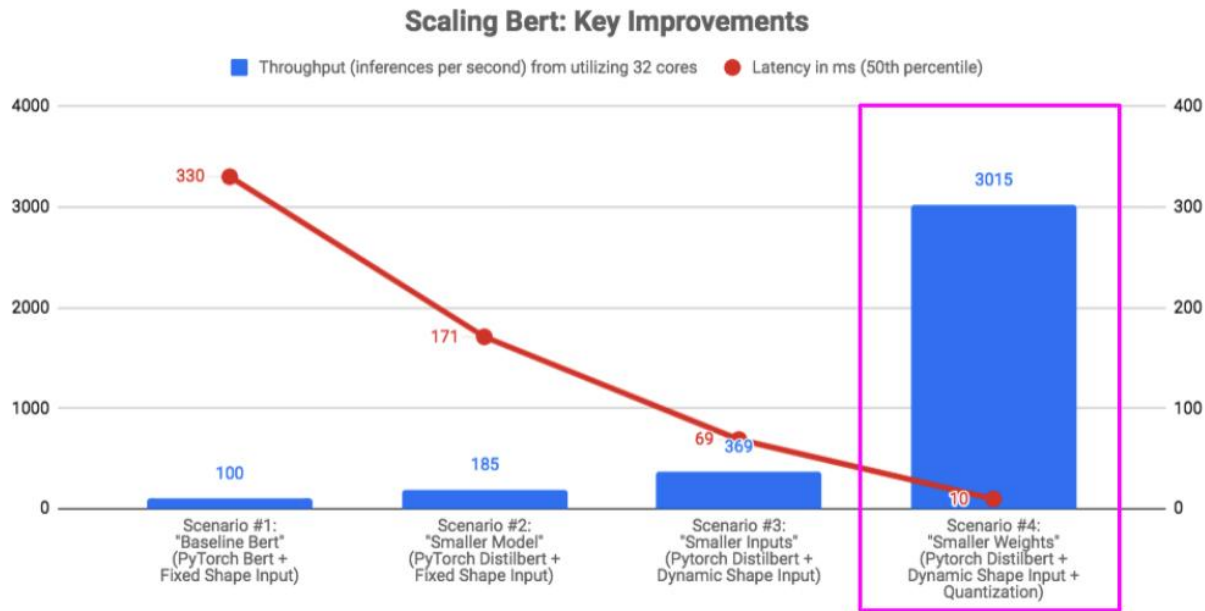
Reference: Google ML crash course [link](#)

Size/Complexity has an effect on latency



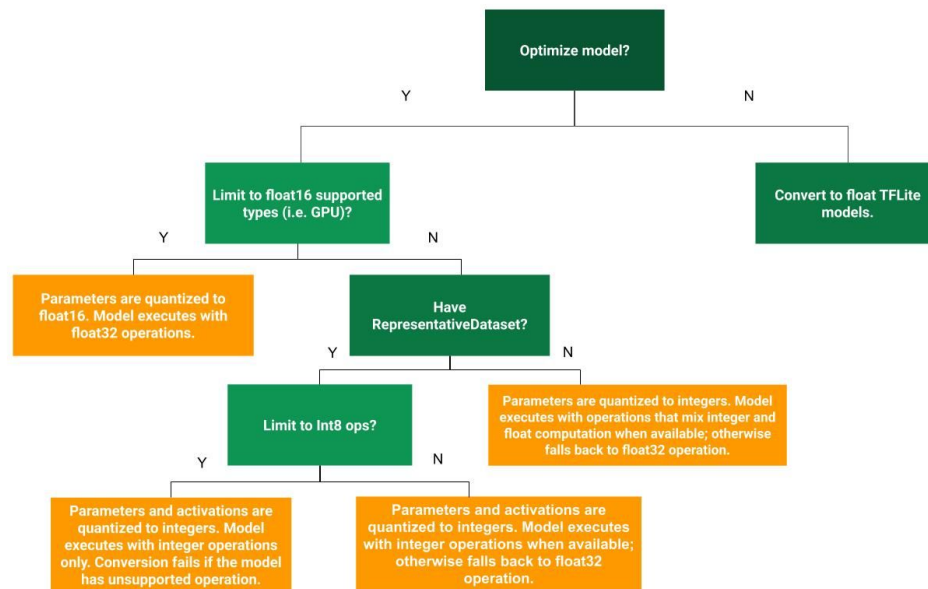
Source: Chip Huyen's ML Eng course ([course](#))

Scaling ML Systems



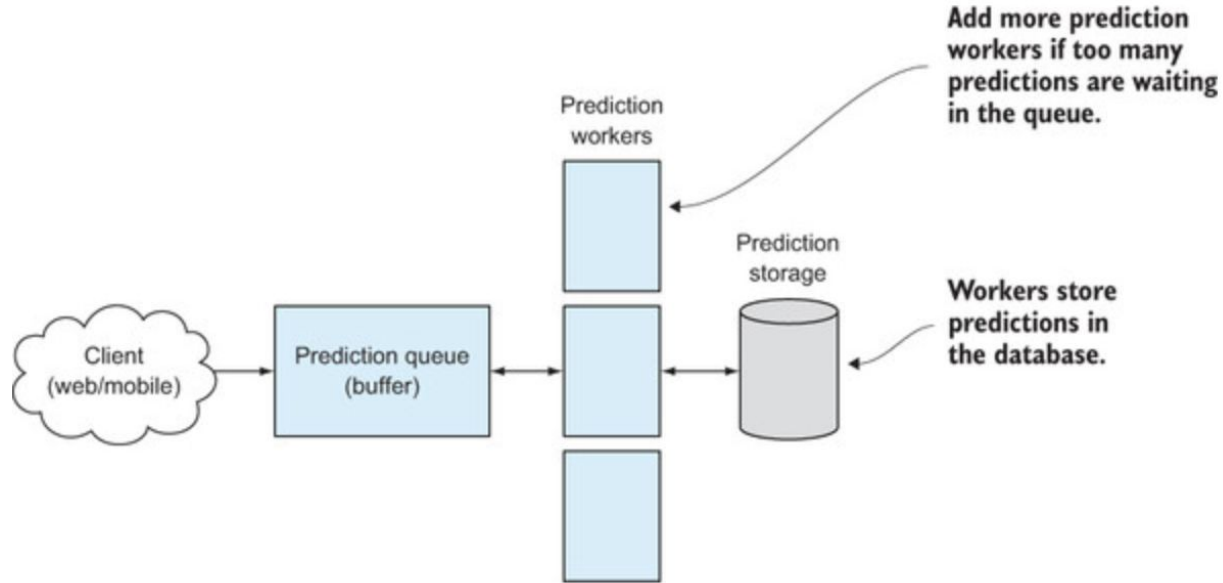
Reference: [How We Scaled Bert To Serve 1+ Billion Daily Requests on CPUs](#) (Roblox, 2020)

Scaling ML Systems



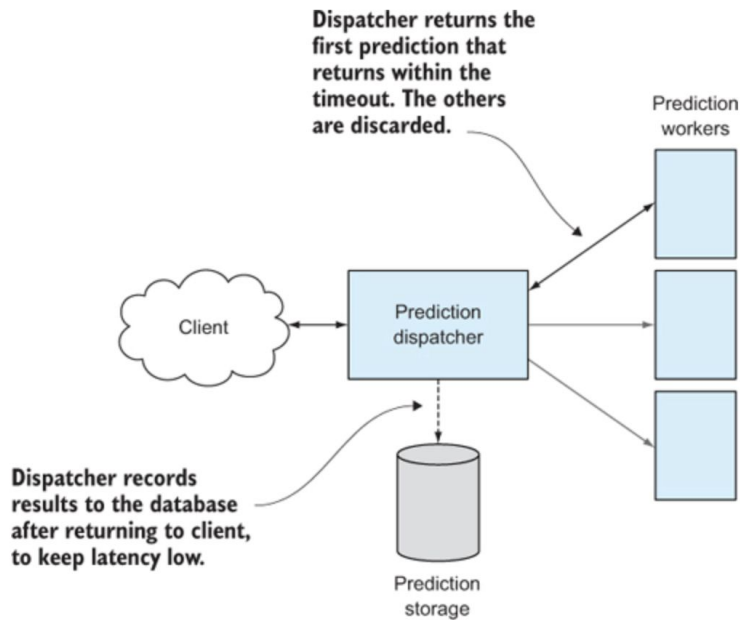
Reference: [Tensorflow Quantization](#)

Scaling ML Systems



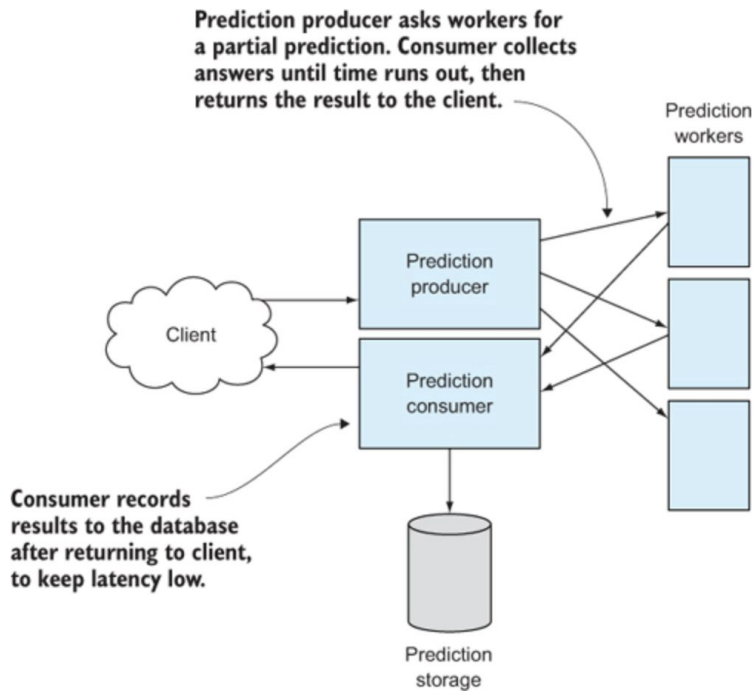
Real-world Machine Learning [\(link\)](#)

Scaling ML Systems



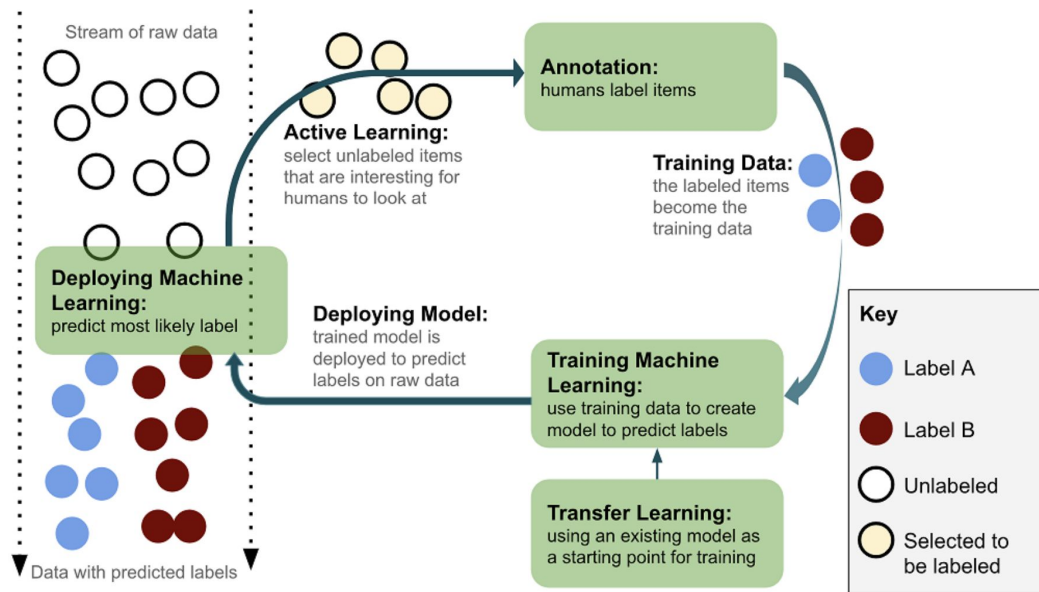
Real-world Machine Learning [\(link\)](#)

Scaling ML Systems



Real-world Machine Learning ([link](#))

Human-in-the-Loop ML

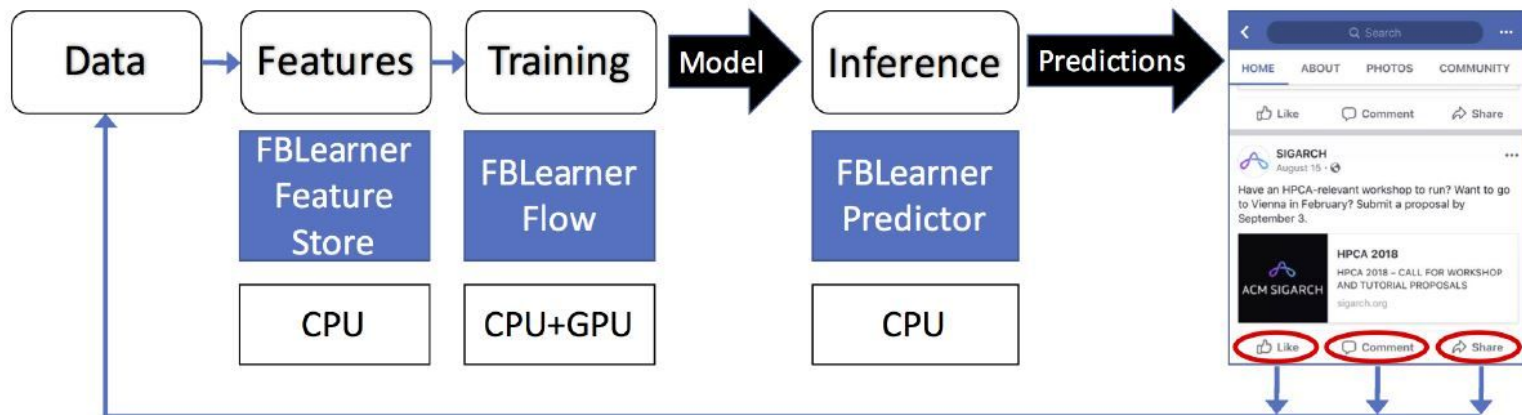


Human in the loop ML [\(link\)](#)

ML Infra

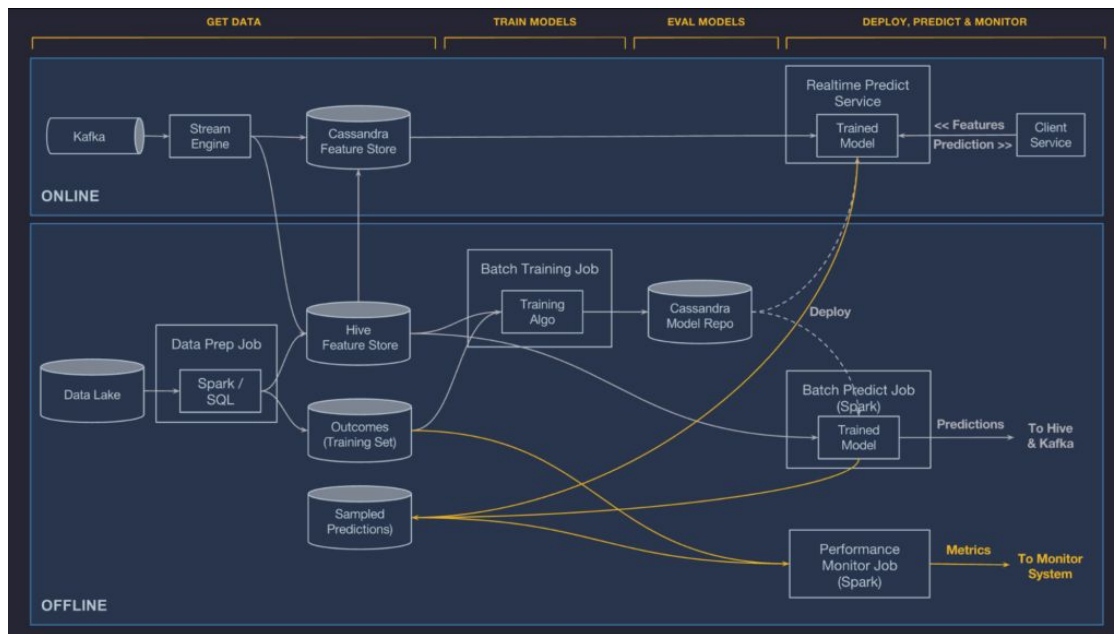
- Different teams with different standards building ML systems into prod.
 - Tensorflow vs PyTorch vs Sklearn vs XGBoost, etc...
 - Deployment process, tracking tools, monitoring, versioning, config, etc... it's a nightmare.
- When big enough, some companies build ML infrastructure to standardize ML best practices
 - Monitoring, deployment, versioning, Feature stores, Concept drift tracking, AB testing & experimentation frameworks, etc.. all following the same standards within a company.

ML Infra (e.g. @ Facebook - FBLearner)



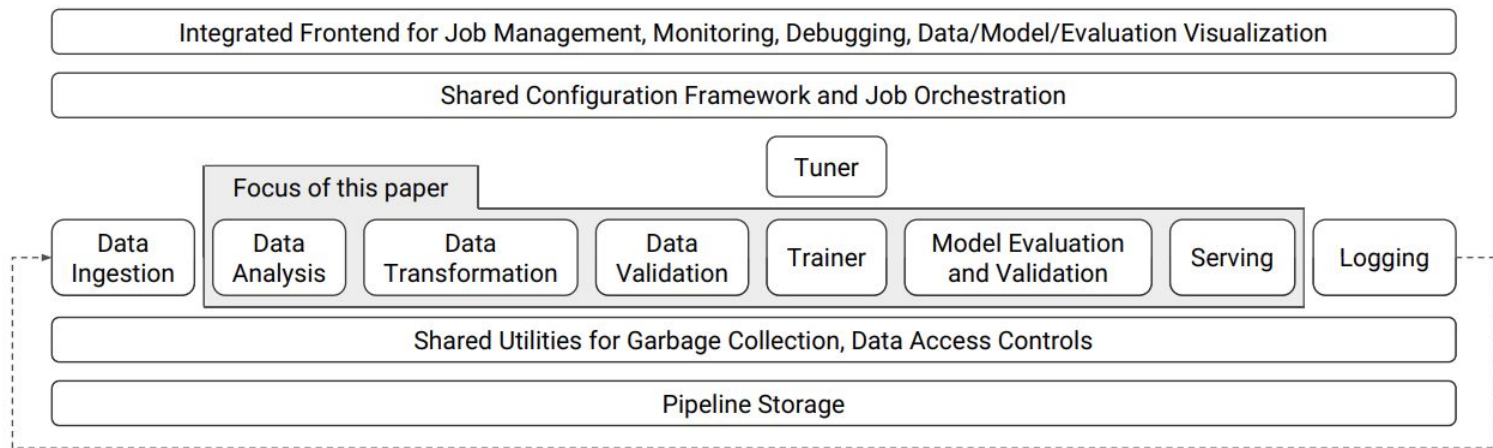
Reference: FBLearner - AI Infra at Facebook [link](#)

ML Infra (e.g. @ Uber - Michaelangelo)



Reference: Michaelangelo [link](#)

ML Infra (e.g. @ Google - TFX)



Reference: TensorFlow Extended [link](#)

Recommended trilogy

Machine Learning: The High-Interest Credit Card of Technical Debt

Sculley, et al., NeurIPS (2015) ([link](#))

Hidden Technical Debt in Machine Learning Systems

Sculley, et al., SE4ML (2014) ([link](#))

What's your ML Test Score? A rubric for ML production systems

Breck, et al., NeurIPS (2016) ([link](#))

ML Eng Projects & Learnings

- Extract Aspects+Sentiment from customer reviews
 - It can take a week to get a prototype but +6 months to get production working.
 - Very obvious: Proving value early on is fundamental.
 - Prototypes are great to minimize the project risk and understand the scope.
 - Classic: 80% of the problem can be solved with a 20% effort.
 - Sometimes scaling horizontally >> scale vertically.
 - You should thrive for this when problems are trivial to parallelize.
- Dynamic pricing
 - Small teams can have massive impact.
 - They need to work on the right problems though...
 - Running MVP online tests can take a big toll to the DS teams...
 - Imagine to collect data + fit a model + run prediction + update production on a daily basis. Far from sustainable.

ML Eng Projects & Learnings

- Search @ www.groupon.com
 - I broke production for the very first time -- countless learnings.
 - Integration testing for ML-based models is particularly hard.
 - A/B testing IS FUNDAMENTAL but can get pretty complex very quickly.
 - Many moving pieces can make the stats analysis very time consuming.
- Improve freshness of rec-sys
 - Online experiments more expensive than offline experiments but more trustworthy.
 - Expensive in every sense (wait time, engineering investment, resource intensive, etc.)
 - Counterfactuals in offline experiments that involves user behavior are **hard**.
 - Creating a gold standard for offline rec sys experiments == death.

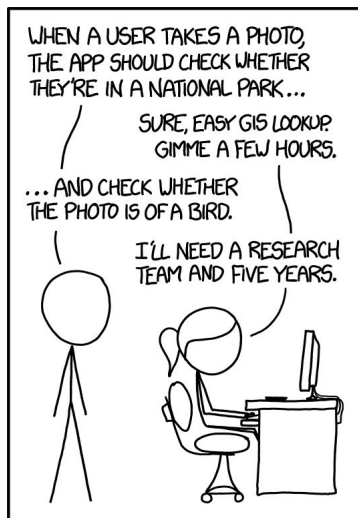
ML Eng Projects & Learnings

- Chatbot for Sales
 - The more complex your ML system the harder is to maintain and the more \$ you pay to AWS.
 - Keep things simple at all costs.
 - Deploying DL-based models (or at least vector size-wise) is **hard** (of course).
 - Particularly pressing if you need a fast iteration environment.
- Graph-based RecSys for Sales
 - Node predictions based on graph embeddings can be a pretty bad idea.
 - If the graph evolves constantly, keeping up retraining the embeddings is painful.
 - ML using user behavior data can be very effective.
 - Not necessarily good and can be very creepy.

ML Eng Projects & Learnings

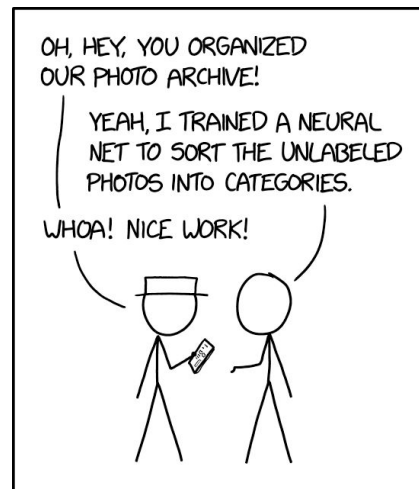
- Very common anti-patterns
 - **“If I just become the most amazing computer scientist, I’ll get very far!”**
 - There’s much more than just coding/modeling/designing. Soft skills are a core skill.
 - Extremely important skill: Communicate your ideas effectively and efficiently.
 - **“I worked SOooo hard on this feature, but no one appreciates my work!”**
 - If you see no value in that feature, say something. Make your own luck.
 - Very easy to fall in love with a problem that could drive Zero value.
 - **“Nah, it was trivial” (*..after working silently the whole weekend on a problem..*)**
 - This will create unrealistic expectations and will most likely burn you out.
 - Working hard is great and really appreciated.
 - **“I’m a 10x Engineer. I can ship everything by myself in 2 weeks.”**
 - “Journey before destination” if you want to be able to maintain a complex architecture.
 - Hero journeys are likely to end poorly. “Bus factor” is life or death for some companies.

ML Engineering - more learnings...



IN CS, IT CAN BE HARD TO EXPLAIN
THE DIFFERENCE BETWEEN THE EASY
AND THE VIRTUALLY IMPOSSIBLE.

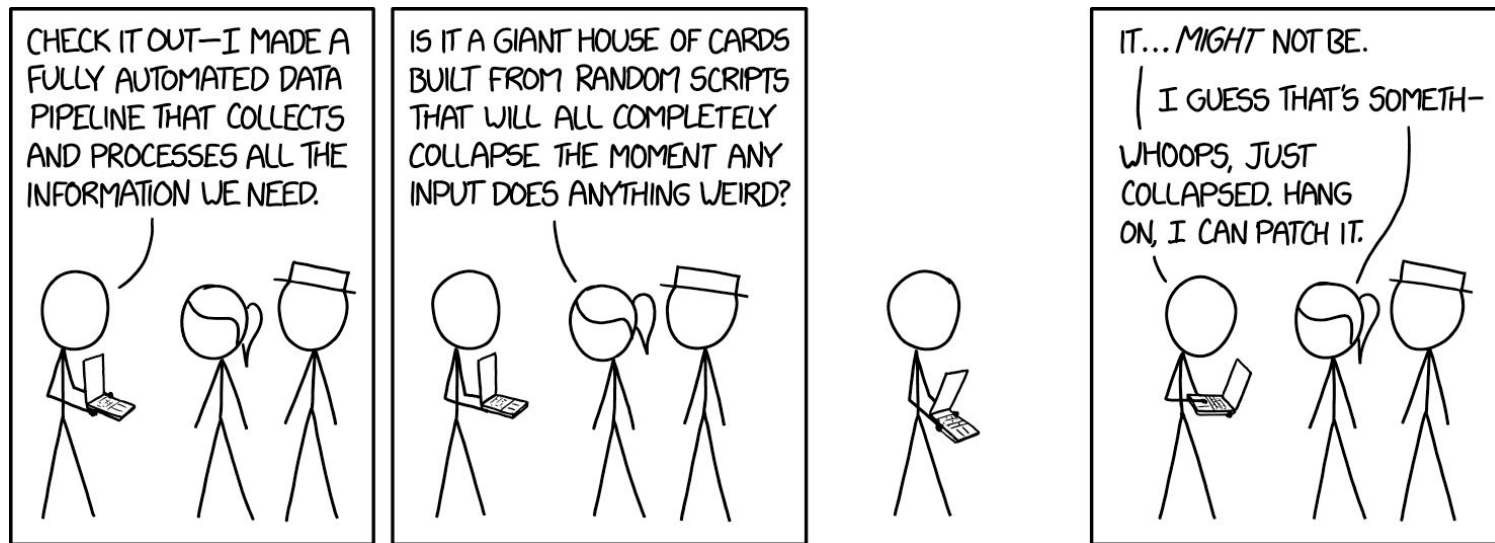
<https://xkcd.com/1425/>



ENGINEERING TIP:
WHEN YOU DO A TASK BY HAND,
YOU CAN TECHNICALLY SAY YOU
TRAINED A NEURAL NET TO DO IT.

<https://xkcd.com/2173/>

ML Engineering - more learnings...



<https://xkcd.com/1425/>

ML Engineering - more learnings...



<https://xkcd.com/1838/>

References

- Machine Learning System Design ([link](#))
 - Good collection of case studies ([link](#))
- ML Engineering Best Practices ([link](#))
- Check the slides!
 - Full of references through the presentation.

Thanks!