

# SRE for ML

## The First 10 Years and the Next 10

Todd Underwood @tmu ♦  
[tmu@google.com](mailto:tmu@google.com) ♦  
2021-Oct ♦  
SRECon ML Track

# Agenda

## **SRE for ML is not (that) new:**

SRE on ML systems dates back more than 10 years. Looking at what has happened in the last decade might help us decide what we need to do in the next.

What? ML!

Where? Steeltown ML SRE

How? One model

Why SRE? A Platform


What's next? It depends



---

# Greetings

SRE for ML

The background features a dark blue, almost black, field. Overlaid on this are several wireframe cubes of varying sizes and orientations. Each cube is composed of thin, light blue lines. From the vertices of these cubes, bright blue starburst light effects radiate outwards, creating a sense of depth and digital energy. The overall aesthetic is futuristic and technical.

# Basic Questions (the usual)

## Who am I?

Recovering Systems/Network engineer, duped into founding Ads ML SRE (SmartASS SRE!) at Google 12+ years ago without knowing what ML or SRE were. Founded and still lead ML SRE and Pittsburgh site for Google. Writing a Book on Reliable ML for O'Reilly (with a bunch of other people). Some chapters already online. Most importantly: animal decision still pending!

## Who are you?

SREs who have heard the hype about ML and wonder what's real. Have not used ML in anger, on purpose but wonder if it's good for anything yet.

## Why are we here?

To demystify the past and try to decide where we're going.

## So, where are we going?

That, compañeres, is the question. Stick along for the ride!




RF NX © Todd Underwood

---

# ML/AI WAT?

SRE for ML



SRE for ML

# ML vs. AI:

## A brief reminder:



**Mat Velloso is on vacation** 🤪🌴

@matvelloso

Follow



Difference between machine learning and AI:

If it is written in Python, it's probably machine learning

If it is written in PowerPoint, it's probably AI

5:25 PM - 22 Nov 2018

8,481 Retweets 23,533 Likes



# ML vs. AI: No, Seriously

## **AI**

A family of technologies and approaches designed to create machines that can demonstrate “intelligence” that we normally associate with humans. Ideally this would include something like “learning” and “problem solving”.

## **ML**

The study of algorithms that enable computer systems to solve some specific problem/perform some task by learning from data. A subset of AI.

## **WAT?**

Today is about ML. Most of what we do is about ML. We’re concerned with algorithms that learn from examples and how to make them work well.

A quick, concrete reminder:

SRE for ML

# What even is ML?

**ML is software that learns from data**

Lots of things we want computer to do are too hard to program. Are these oranges? Apples? Limes?

ML is a set of strategies for building data structures that are updated by looking at data and then used to categorize or interact with other data.

See my SRECon 2019 Dublin talk for more context (or literally hundreds of better introductions by smarter people).



Photo by [Bruno Scarmgnon](#) on [Pexels](#)

Photo by [Public Domain Pictures](#) on [Pexels](#)



## SRE for ML

# ML: Learn from Examples (simple supervised example)

**Labeled Examples:** A set of instances of the type of thing we're trying to categorize or predict with the “right answer” already determined.

**Features:** Facts about the examples that are relevant to our predictions.

**Model:** A data structure that represents what we have “learned” so far.

**A Learning System:** Reads labeled examples and updates the model.




Photo by [Trang Doan](#) on [Unsplash](#)

---

# Where?

SRE for ML

The background is a dark, deep blue. It features a grid of glowing wireframe cubes, some of which are illuminated from within, creating a starburst effect. The cubes are arranged in a staggered pattern, and the overall scene has a futuristic, digital aesthetic. There are also some faint, glowing particles or dust specks scattered throughout the space.

SRE for ML

# The Beginning: SRE for ML *(in Pittsburgh, PA, USA?)*

## Google Ads Quality: We Use Math

Ads targeting is critical to the success of search.

We should show relevant ads only.

We can charge only when users click on those ads.

It is in everyone's interest to show the best ads.

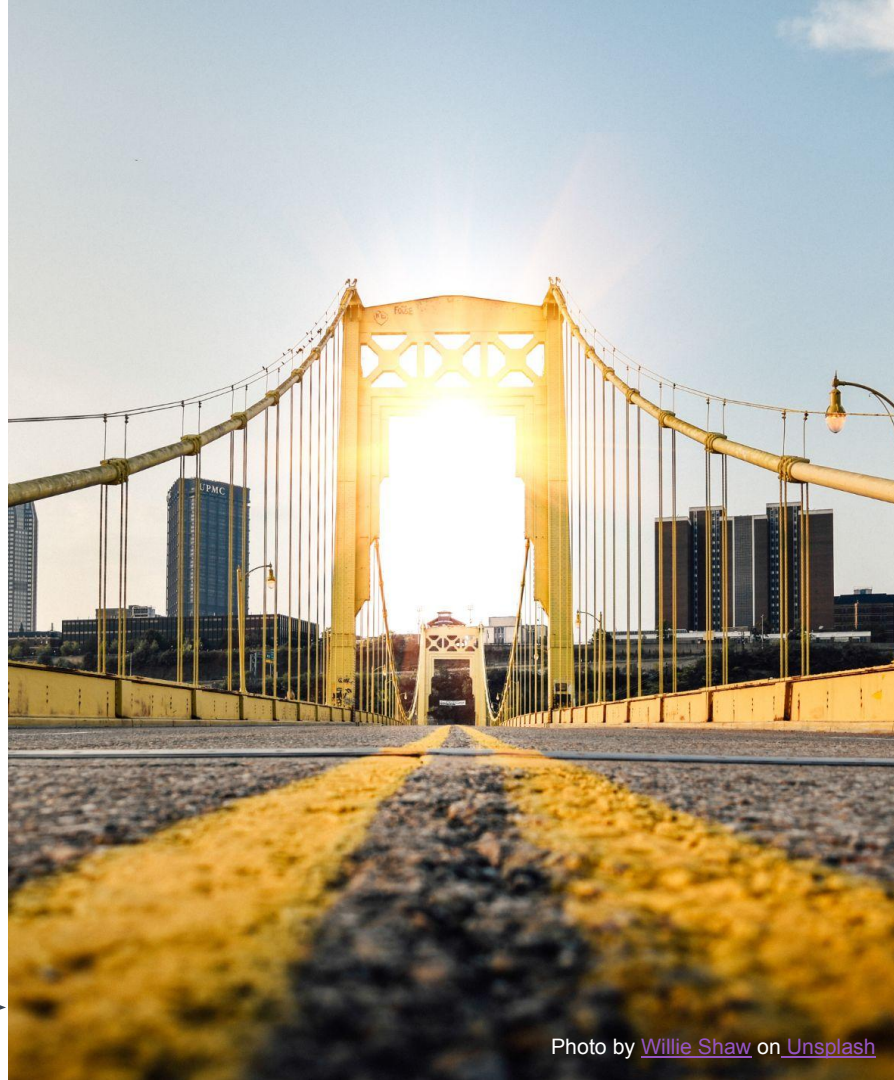
Enter... ML (some kind of fancy math).

Why SRE? Revenue/experience critical:

- Good ads make users happy and get clicks.
- Bad ads cost money to find and serve, anger users, and make zero money.

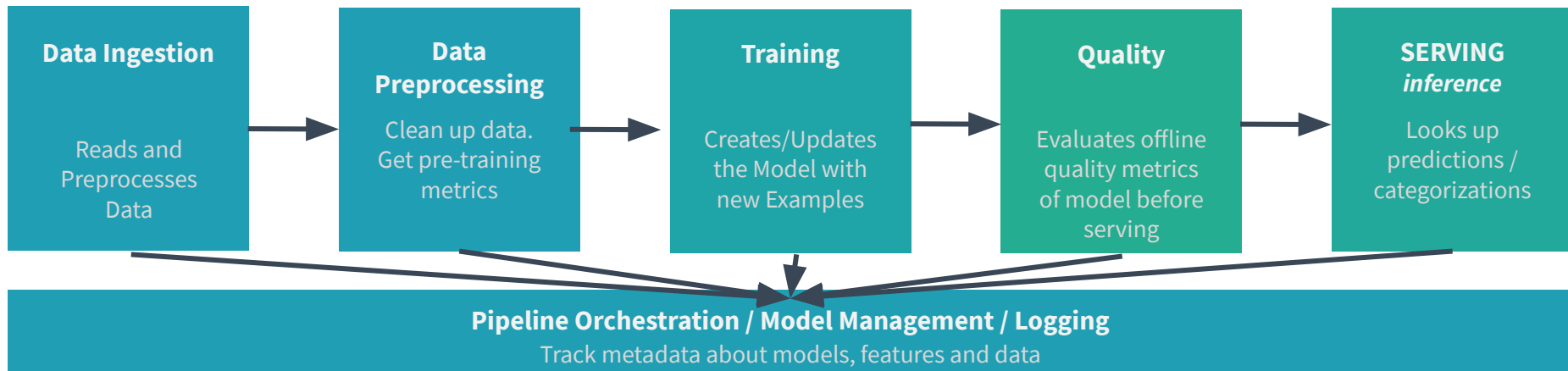
Why Pittsburgh?

**CMU is over there (to the East a bit)** →



SRE for ML

# (our) Basic ML Systems Architecture



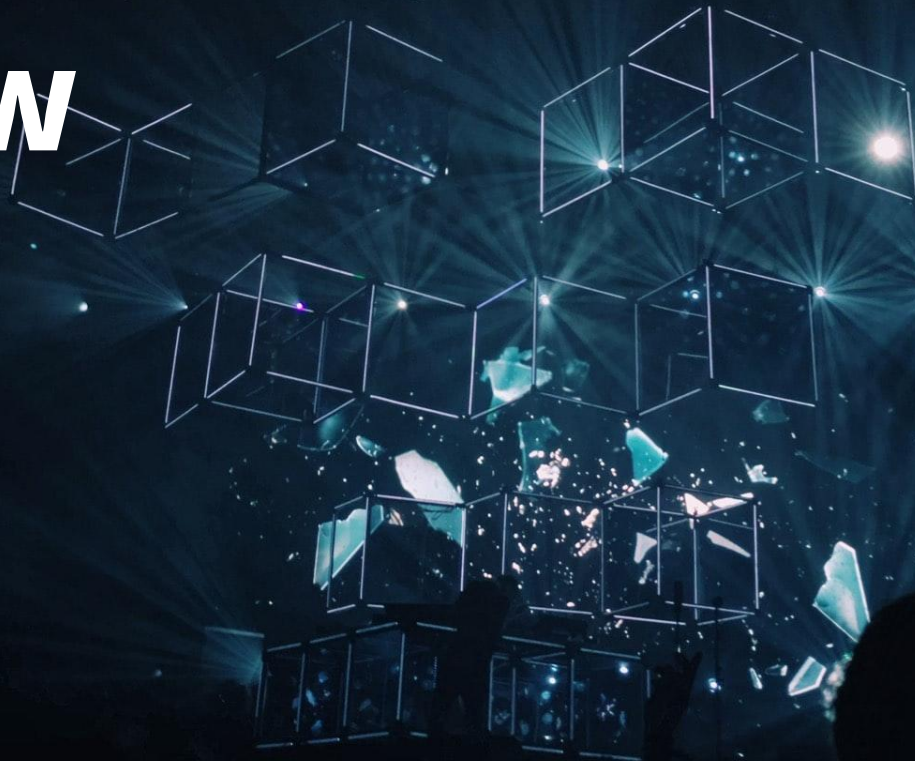
Read data; check data; update fancy data structure; check fancy data structure; serve from fancy data structure.

That's really it. (See Mary McGlohon's *Demystifying Machine Learning in Production* talk for way more detail)

---

# How

SRE for ML



SRE for ML

# Everything Starts with One Model

## There's a Reason to do ML

- We have some compelling application where we believe ML will work well
- We build a model and it works...
- So we iterate, troubleshoot, and improve
- Train on new data
- Deploy repeatedly (and repeatably) to production

Now we have a **SYSTEM**.



Photo by [Clarisse Croset](#) on Unsplash

# New Teams, New Models

## Success is attractive

- Other teams notice the system works
- They want their models trained, tested, deployed, served similarly.
- More teams with more models show up.
- Dozens of teams with hundreds of models, each slightly different.
- They all want production stability with as much flexibility as possible.


If we meet their needs, soon, we have a multi-model, multi-tenant **PLATFORM**.



---

# Why SRE

SRE for ML

The background is a dark, deep blue space filled with a grid of glowing wireframe cubes. Each cube is outlined in a lighter blue, and some of them have bright, multi-pointed starburst light effects emanating from their corners. The overall aesthetic is futuristic and technical, suggesting a digital or data-driven environment.



SRE for ML

# The ML Platform

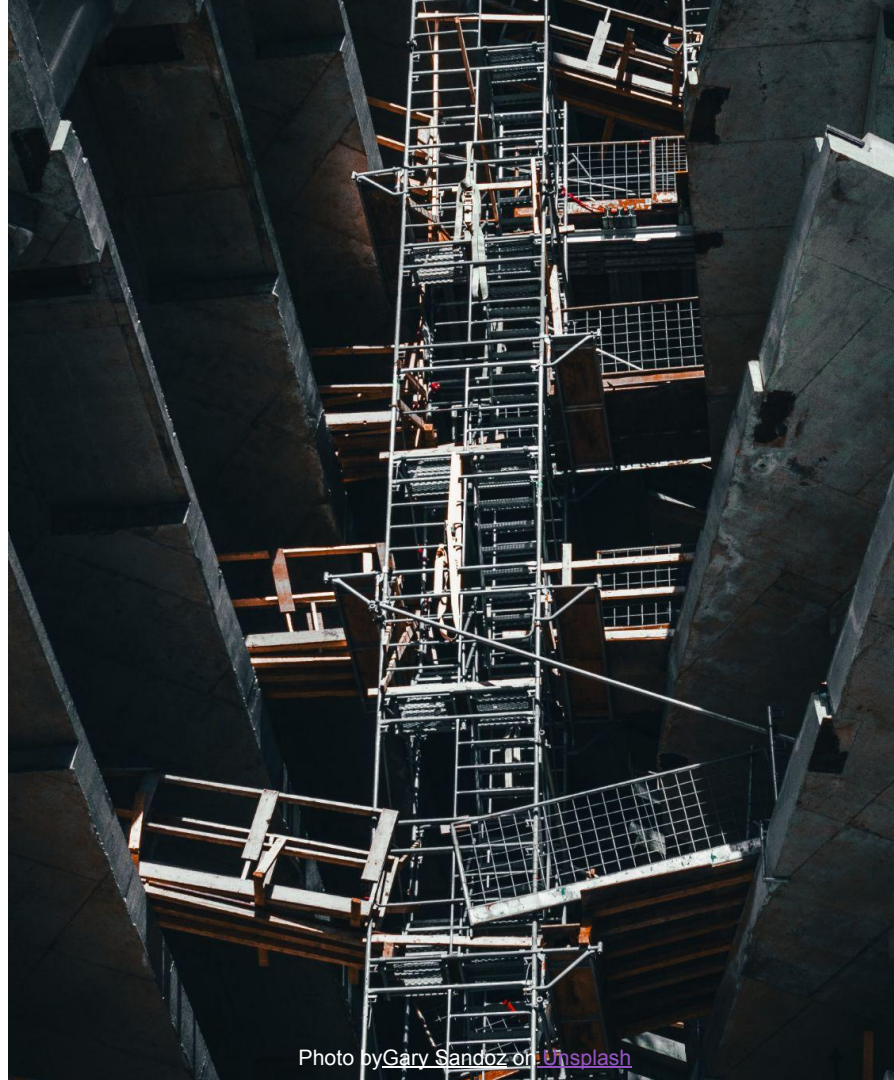
Go see Mary McGlohon's  
talk first...

I hope you saw it yesterday.  
If not, please watch it  
ASAP.[\*]

You should know:

- What an ML Platform looks like
- How it fails.
- A few great techniques for preventing and mitigating those failures.

[\*] <https://www.usenix.org/conference/srecon21/presentation/mcglohon>



# The ML Platform

## The Platform

- (Almost) Just another distributed system.
- Needs:
  - Capacity planning
  - Monitoring
  - Deployments and testing
  - Incident response
  - Hardening/robustification
  - Automation.

...(mostly) Just regular, (boring?) SRE stuff. :-)



SRE for ML

# One Elusive Goal

We want one thing more than everything else

New models/teams need **flexibility**.

Custom model architectures, easy addition of new features, fast deployment, models run directly by teams.

...but then...

They need **fast productionization**. with reliable training & serving, automatic capacity provisioning, SRE incident response, etc.

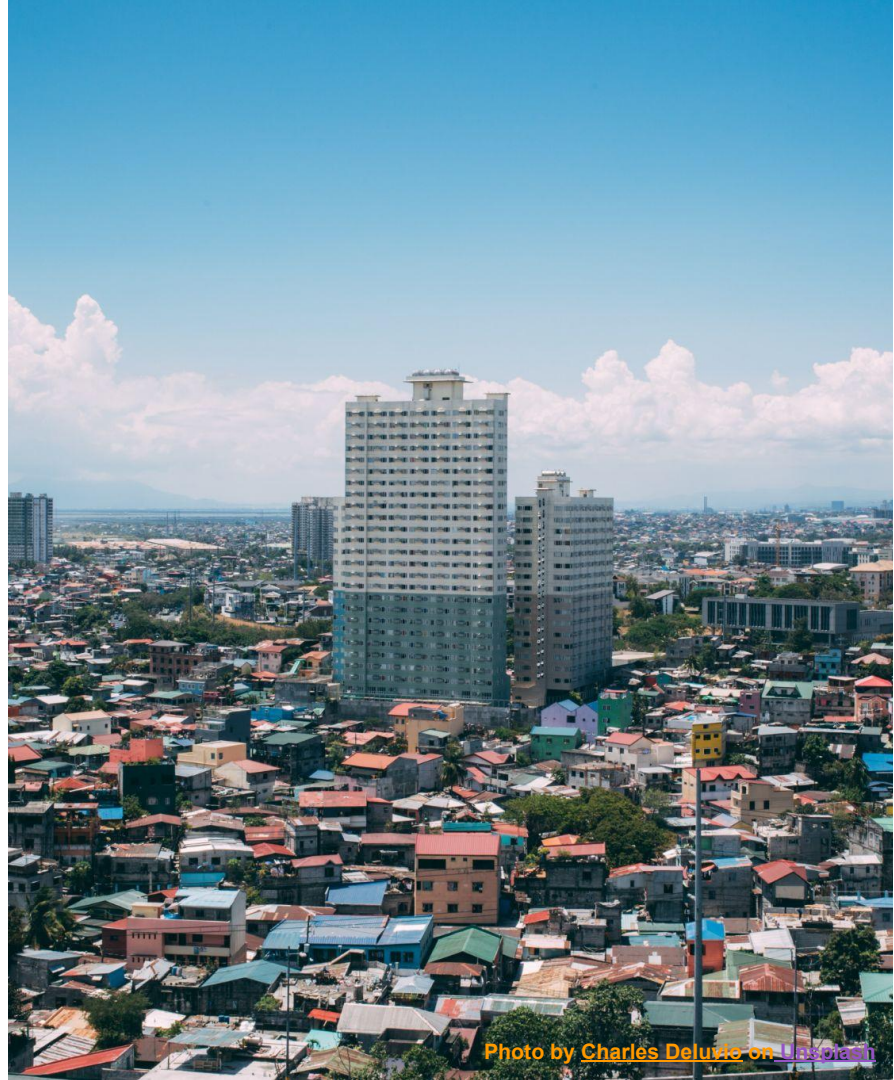


Photo by Charles Deluis on Unsplash

SRE for ML

# Model Quality: The Big Problem

**Who handles model quality problems?**

**Obvious answer: Model developers**

Model developers built the model, understand what it is supposed to do and can evaluate/troubleshoot problems.

**Obvious other answer: ML SREs**

Once models are stable, quality problems are often caused by **systems** issues (resources, pipeline stalls, serving configuration) rather than modeling issues.

**This question is unsolved.**



SRE for ML

# Big Idea

## “Promotion” from Experimentation to Prod

How?

- Like everything else we do: start by hand to make sure we understand the problem.
- Model owners can train, evaluate, and serve models with no external gating (and minimal SLOs!)
- Once some set of things about a model is true, the model is eligible for “promotion” to production, upon request. Get production SLOs.

What things?

- That it works. Basically that it works.



# What Metrics Mean It “Works”?

## Measure what you would respond to

### SLO-violating events:

- Data are incomplete, too big, too small, or has a different distribution on some slice.
- Training too slow or stuck.
- Training requires much more resources
- Significant change in model quality.
- Model will not load in serving.
- Model is much slower in serving.

Set objectives for these (and other model system metrics you care about) and measure models' performance according to them.



# Possible(?) Automation

**Reliability of a model is that of a comparable model**

**Models that are retrained:** Easy.

The predicted reliability of the model is the trailing average of its own reliability over n previous runs.

**All other models:** Harder. Find an equivalence class.

Identify models of similar structure trained on similar features. Predict the reliability based on the delta of relevant aspects of the model.

Future work.



SRE for ML

# Speed Matters

**ML Speed -> Innovation**

**Innovation requires iteration requires speed.**

ML is a game of exploration and iteration. Without trying things, we cannot make the magic.

**Only the best systems have high reliability/speed.**

Without careful engineering, thoughtful monitoring, good abstractions, and well trained incident response, speed isn't possible.

**This is why we need SRE for ML.**






---

# What's Next

SRE for ML

The background features a dark blue, almost black, field. Overlaid on this are several wireframe cubes of varying sizes and orientations. These cubes are illuminated from within, creating a bright cyan or light blue glow. Interspersed among the cubes are numerous starburst light effects, also in shades of cyan and white, which radiate outwards, giving the impression of a complex, multi-dimensional data structure or a futuristic architectural design. The overall aesthetic is high-tech and mysterious.

HACKING ML INTO YOUR ORG

# Make ML Boring

## Five Big Things Will Happen

1. Organizations will accept “older” ML technology, and AutoML approaches. Good enough will be good enough.
2. Platforms will converge in features and stabilize. And mostly fade into the background
3. Training and serving costs will plummet.
4. APIs to integrate ML into applications will stabilize and become ubiquitous
5. ML model quality evaluation will become ubiquitous and trustworthy



HACKING ML INTO YOUR ORG

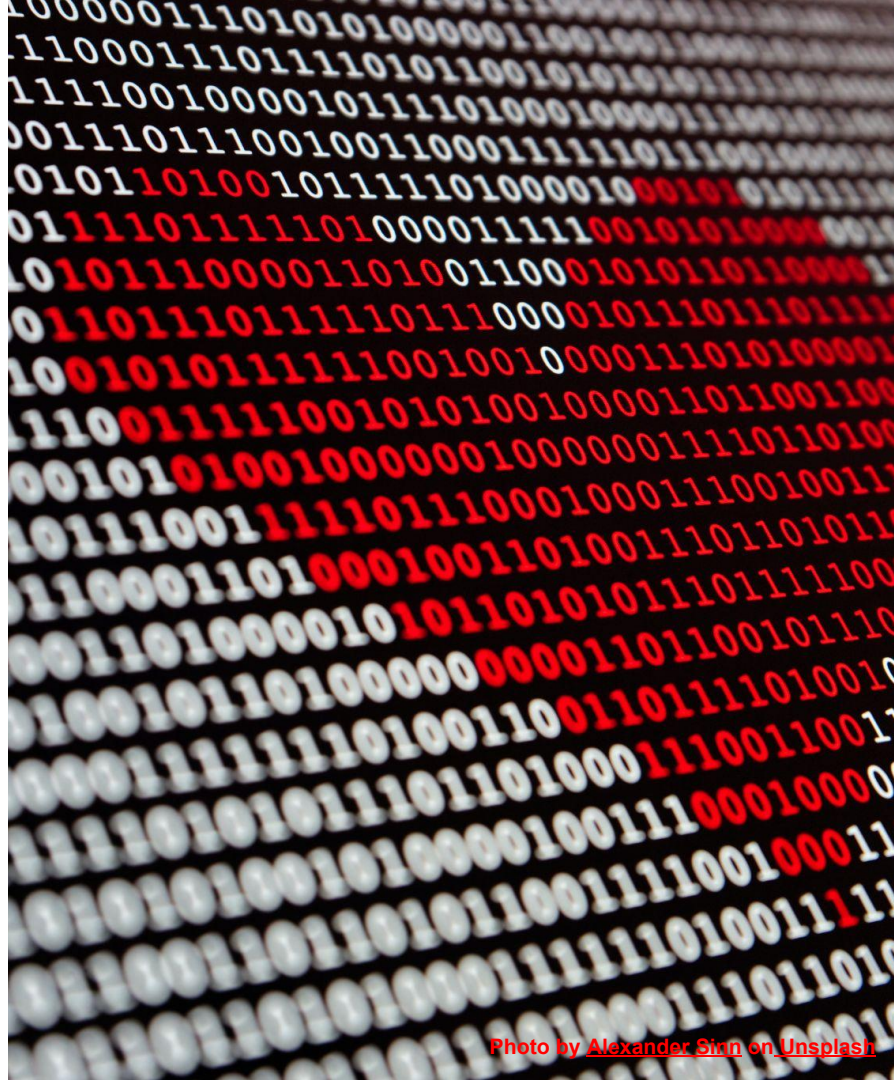
# SRE Teams Will All Do ML

## Boring ML Will Be Everywhere

Stable ML platforms, easy modeling, cheap training and serving will push ML to ubiquity. Almost every SRE team will have portions of their service using at least some ML.

Are we ready for that? Not Remotely.

Time to get to work.



---

# Thank You

SRE for ML

Todd Underwood @tmu ♦  
[tmu@google.com](mailto:tmu@google.com) ♦  
2021-Oct ♦  
SRECon ML Track