ML Engineering

Production ML Systems

Includes lots of stuff besides training!

- Data Collection
- Serving
- Monitoring
- Config
- Resource mgt.

etc.

But lots of off-the-shelf solutions exist!

↓

Which to pick?

Depends on...

Static vs. dynamic training

(i.e. offline vs. online)

Static: Model trained then used continuously

Dynamic: Data comes in and updates the model iteratively

	Pros	Cons
Static	Easy to build & test	Still requires monitoring Can go stale
Dynamic	Fresher model	Even more monitoring, validation Rollback capabilities Data quarantine

Offline good when the data won't change much over time, e.g. image recognition.

Online good when underlying distribution might be changing, e.g. seasonality.

Static vs. dynamic inference

Offline (static/batch) — Write to table, lookup (w/cache) at runtime

Pros: Batch quota, cheaper
Post-prediction validation

Cons: May not handle e.g. tail queries
High latency

Online (dynamic) → predict on demand@runtime

Cons: May be expensive if model is costly or slow
Higher monitoring needs

Pros: Handle all inputs, fresh

Data Dependencies

- Feature management
  — Input data determines behavior!
  How to test, etc?
- Questions to ask:
  - Is this signal reliable? (i.e. always present)
  - Is it stable? Does the system that produces it change over time? Can it be versioned?
  - Is the signal necessary? Does usefulness outweigh cost?
  - Correlations — Do we need to separate signals somehow?
  - Feedback loops — Is output affecting inputs? (Stationarity)