Intelligent Services

Serving Machine Learning

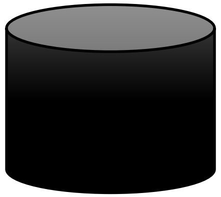
Joseph E. Gonzalez

jegonzal@cs.berkeley.edu; Assistant Professor @ UC Berkeley

joseph@dato.com; Co-Founder @ Dato Inc.

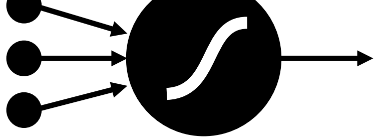
Contemporary Learning Systems

Big
Data



Training

Big
Models



Contemporary Learning Systems

scikit learn

Dato Create

MLbase

TensorFlow

WEKA
The University of Waikato

Spark MLlib

torch

BIDMach

Oryx 2

mlpack

VW

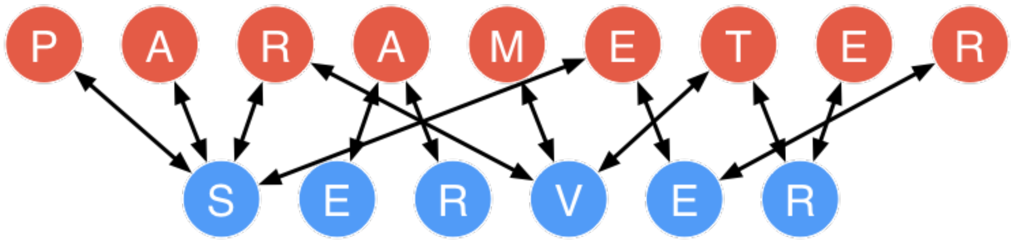
LIBSVM

PredictionIO

DMLC

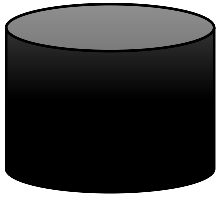
theano

H_2O



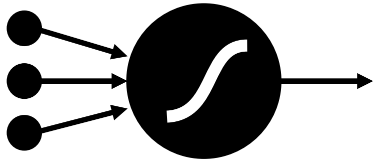
What happens *after* we train a model?

Data



Training

Model




PowerGraph: Distributed Graph-Parallel Computation on Natural Graphs

Joseph E. Gonzalez, Carnegie Mellon University, jegonzal@cs.cmu.edu
Yucheng Low, Carnegie Mellon University, ylow@cs.cmu.edu
Haijie Gu, Carnegie Mellon University, haijieg@cs.cmu.edu
Danny Bickson, Carnegie Mellon University, bickson@cs.cmu.edu
Carlos Guestrin, University of Washington, guestrin@cs.washington.edu

Abstract

Conference Papers

Dashboards and Reports

Drive Actions

What happens *after* we train a model?

Data

Training

Model

Conference Papers

Dashboards and Reports

Drive Actions

Suggesting Items at Checkout

Fraud Detection

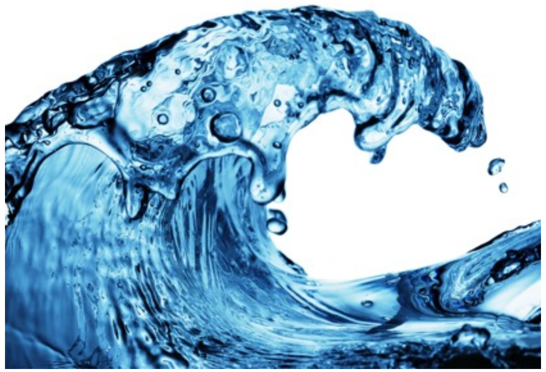
Cognitive Assistance

Internet of Things

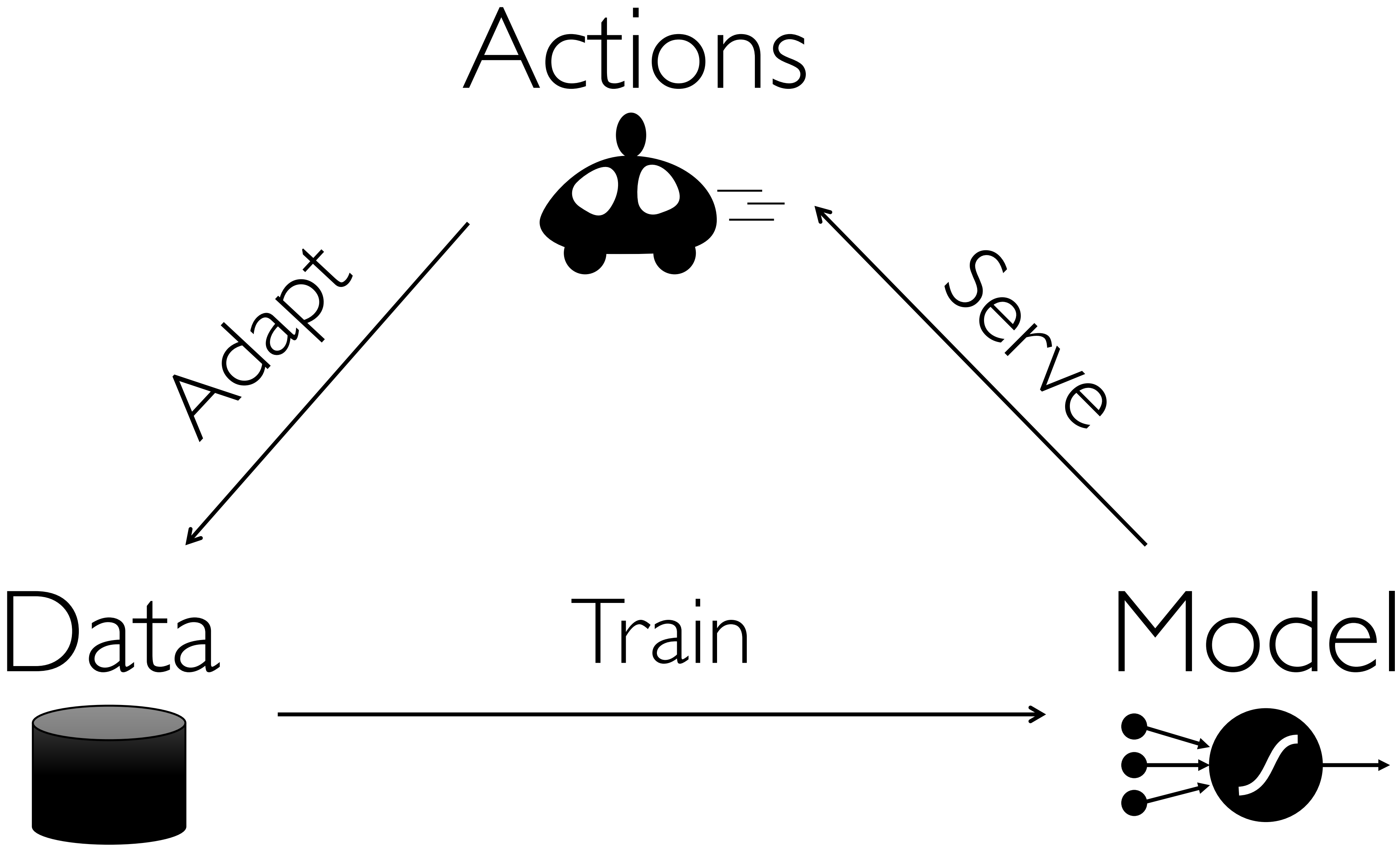
Low-Latency

Personalized

Rapidly Changing




Data
Train
Model

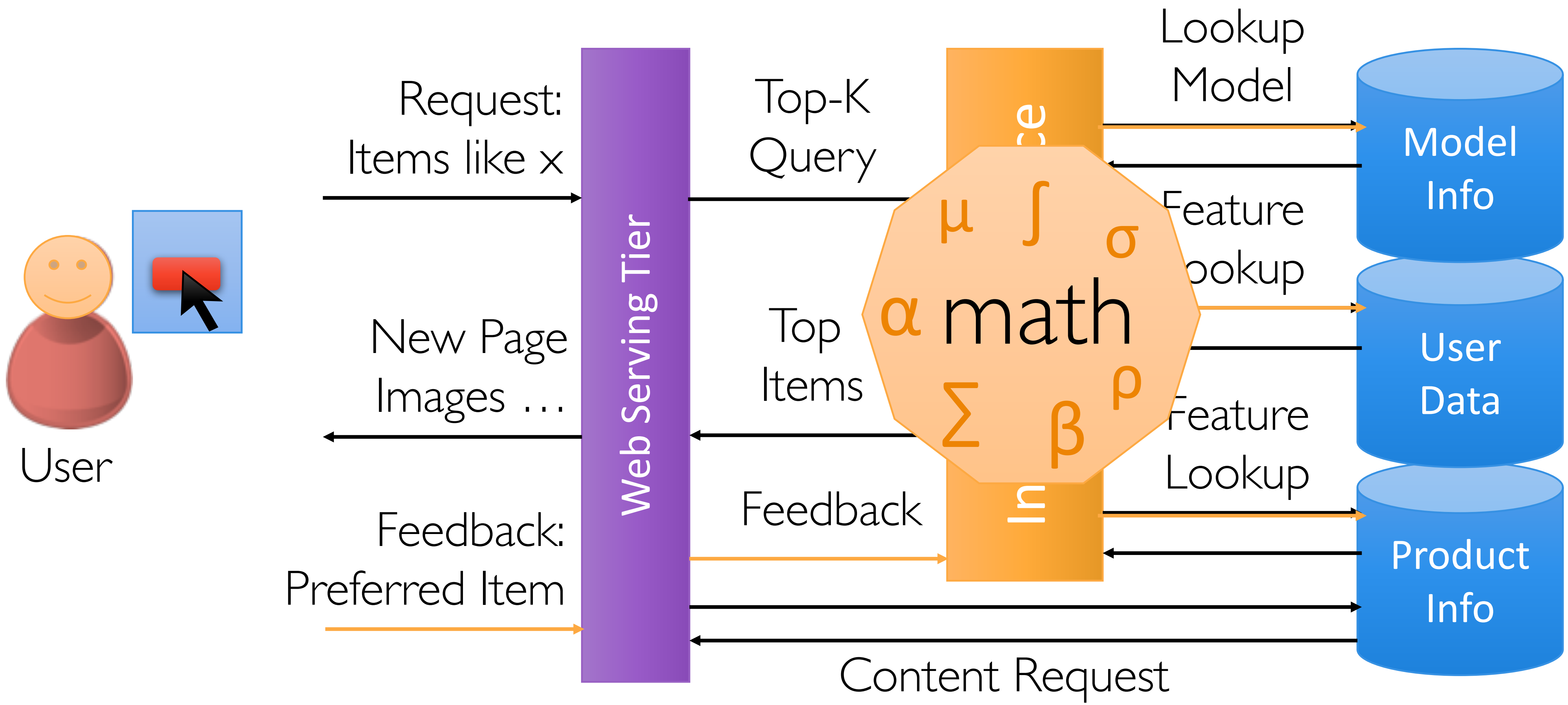


Actions
Adapt
Serve
Data
Train
Model

Machine
Learning

Intelligent
Services

The Life of a Query in an Intelligent Service



Essential Attributes of Intelligent Services

Responsive

Intelligent applications
are interactive

Adaptive

ML models out-of-date the
moment learning is done

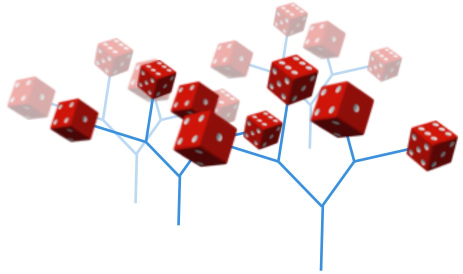
Manageable

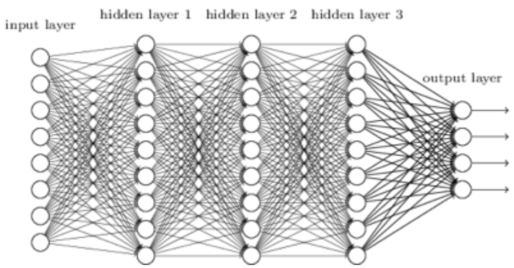
Many models
created by multiple people

Responsive: *Now and Always*

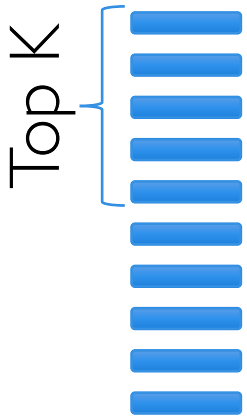
Compute predictions in < 20ms for complex

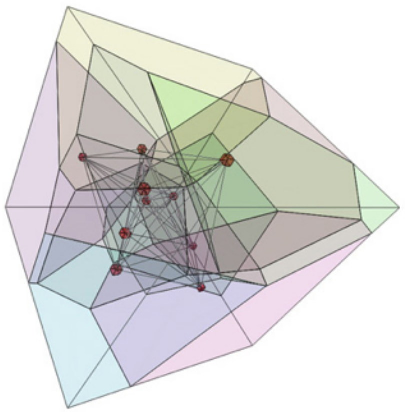
Models





Queries



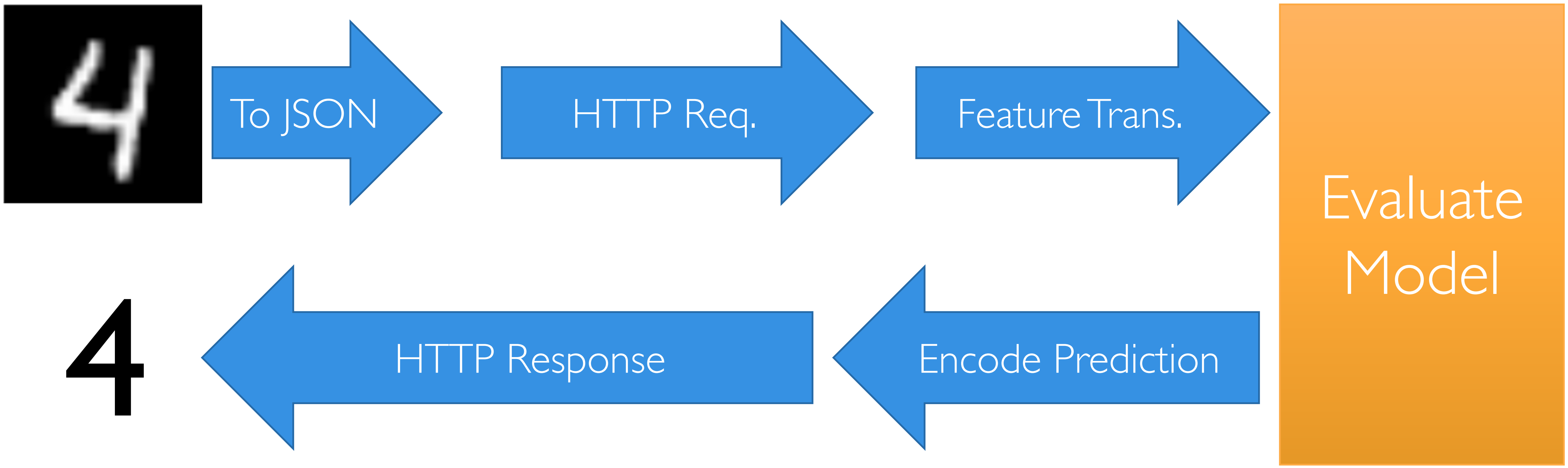


Features

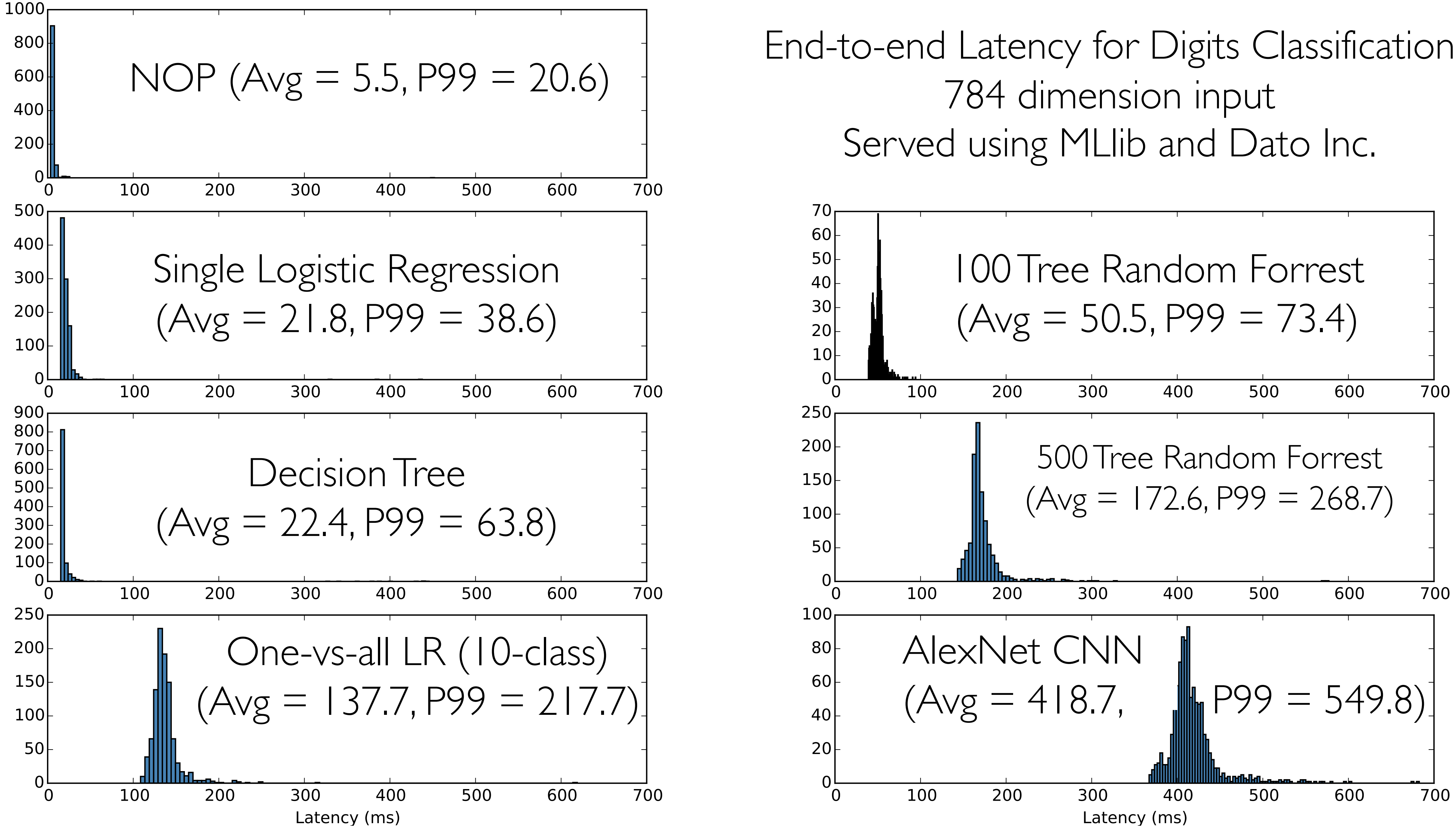
SELECT * FROM
users JOIN items,
click_logs, pages
WHERE …

under heavy *query load* with system *failures*.

Experiment: End-to-end Latency in Spark MLlib

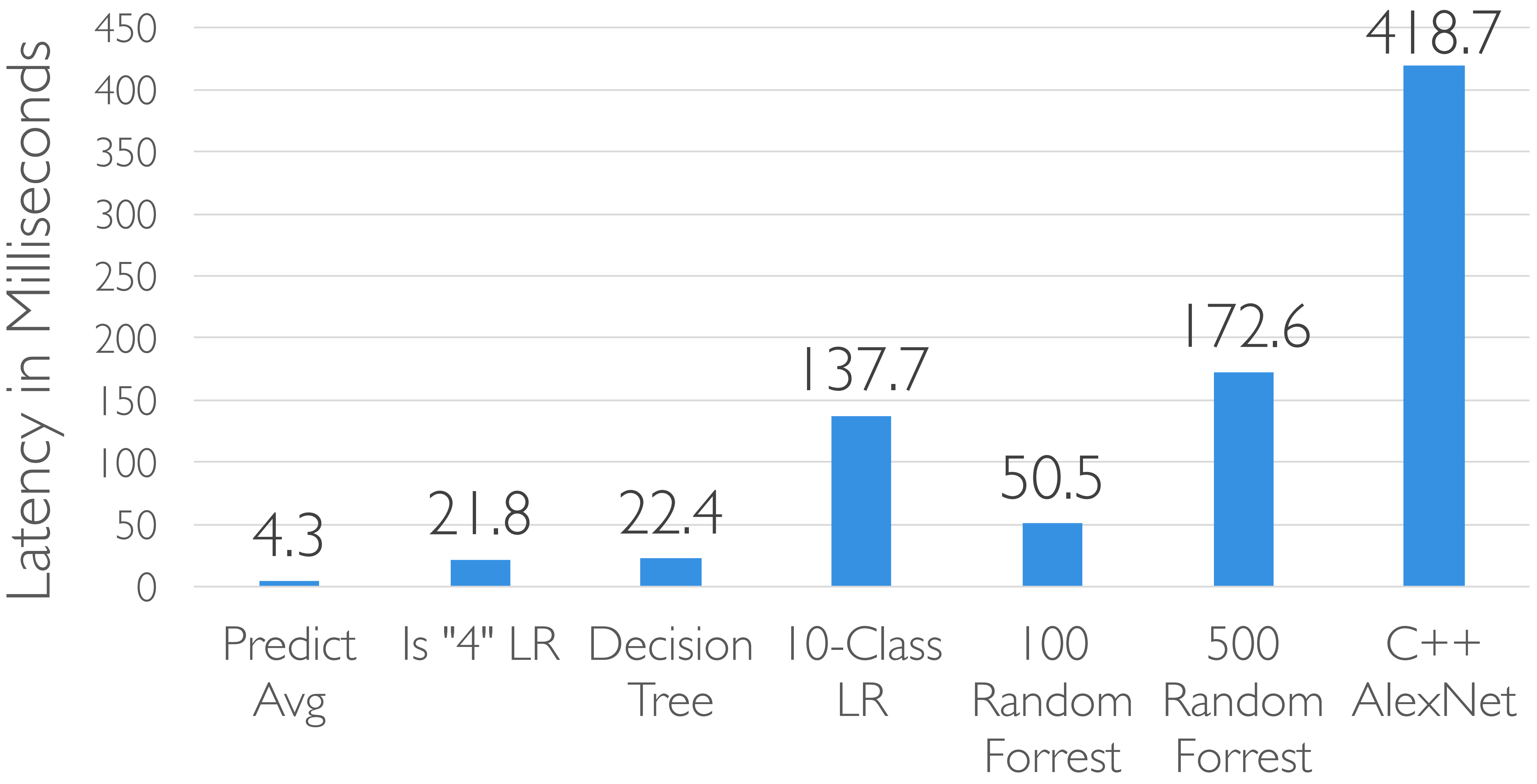


End-to-end Latency for Digits Classification
784 dimension input
Served using MLlib and Dato Inc.



Count out of 1000

Latency measured in milliseconds



Latency in Milliseconds
450
400
350
300
250
200
150
100
50
0
4.3
21.8
22.4
137.7
50.5
172.6
418.7
Predict Avg
Is "4" LR
Decision Tree
10-Class LR
100 Random Forrest
500 Random Forrest
C++ AlexNet

Adaptive *to Change at All Scales*

Population — Granularity of Data — Session

Shopping for Mom

Shopping for Me

Months — Rate of Change — Minutes

Adaptive *to Change at All Scales*

Population — Granularity of Data — Session

Law of Large Numbers
→ Change Slow

Rely on efficient offline retraining
→ High-throughput Systems

Months — Rate of Change — Minutes

Adaptive *to Change at All Scales*

Population — Granularity of Data — Session

Small Data → Rapidly Changing

Low Latency → Online Learning

Sensitive to feedback bias

Shopping for Mom

Shopping for Me

Months — Rate of Change — Minutes

The Feedback Loop

I once looked at cameras on Amazon …

Similar cameras and accessories

Opportunity for
Bandit Algorithms

Bandits present new challenges:

- computation overhead
- complicates caching + indexing

My Amazon Homepage

Exploration / Exploitation Tradeoff

Systems that can take ***actions*** *can* ***adversely bias*** *future* ***data****.*

Opportunity for Bandits*!*

Bandits present new challenges:

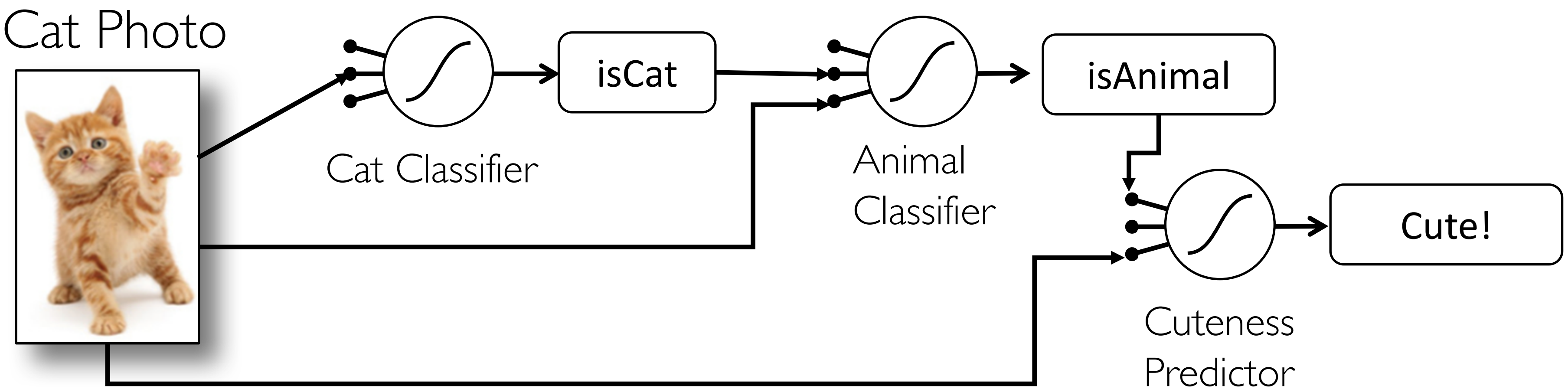
- Complicates caching + indexing
- tuning + counterfactual reasoning

Management: Collaborative Development

Teams of data-scientists working on similar tasks

➢ "*competing*" features and models

Complex model dependencies:



VELOX

UC Berkeley AMPLab

Daniel Crankshaw, Xin Wang, Joseph Gonzalez
Peter Bailis, Haoyuan, Zhao Zhang,
Michael J. Franklin, Ali Ghodsi,
and Michael I. Jordan

Dato

Predictive Services

VELOX

UC Berkeley AMPLab

Daniel Crankshaw, Xin Wang, Joseph Gonzalez
Peter Bailis, Haoyuan, Zhao Zhang,
Michael J. Franklin, Ali Ghodsi,
and Michael I. Jordan

Dato

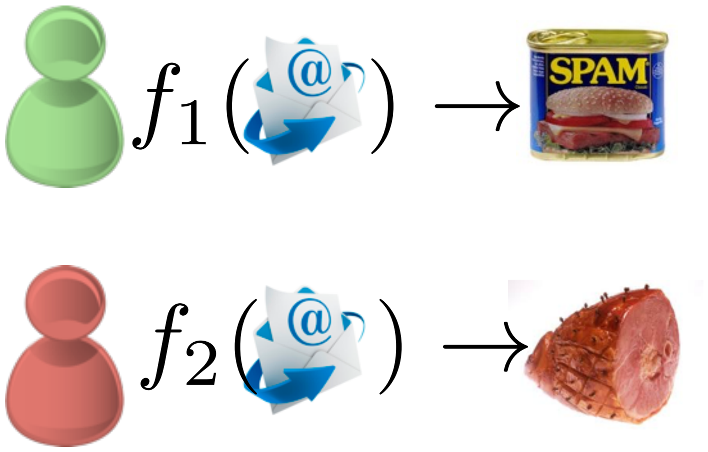
Predictive Services

Active Research Project

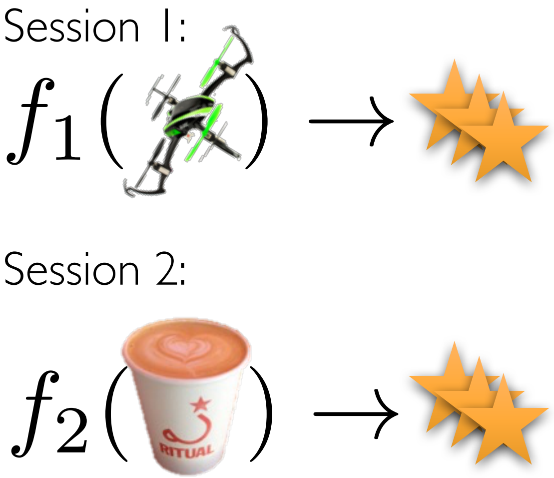
Velox Model Serving System
[CIDR'15, LearningSys'15]

Focuses on the multi-task learning (MTL) domain

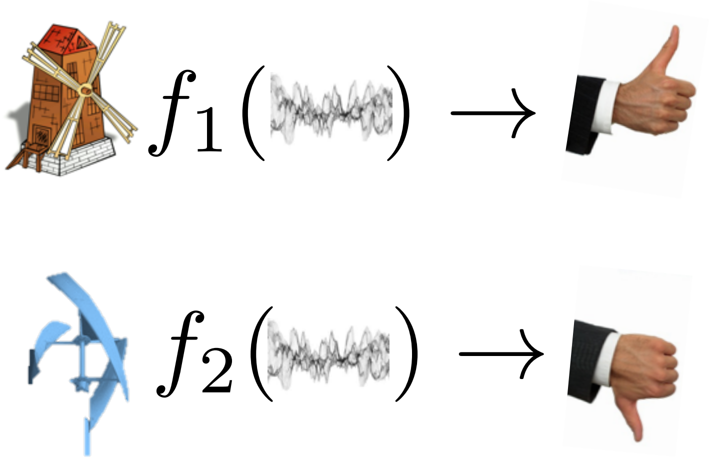
Spam Classification



Content Rec. Scoring



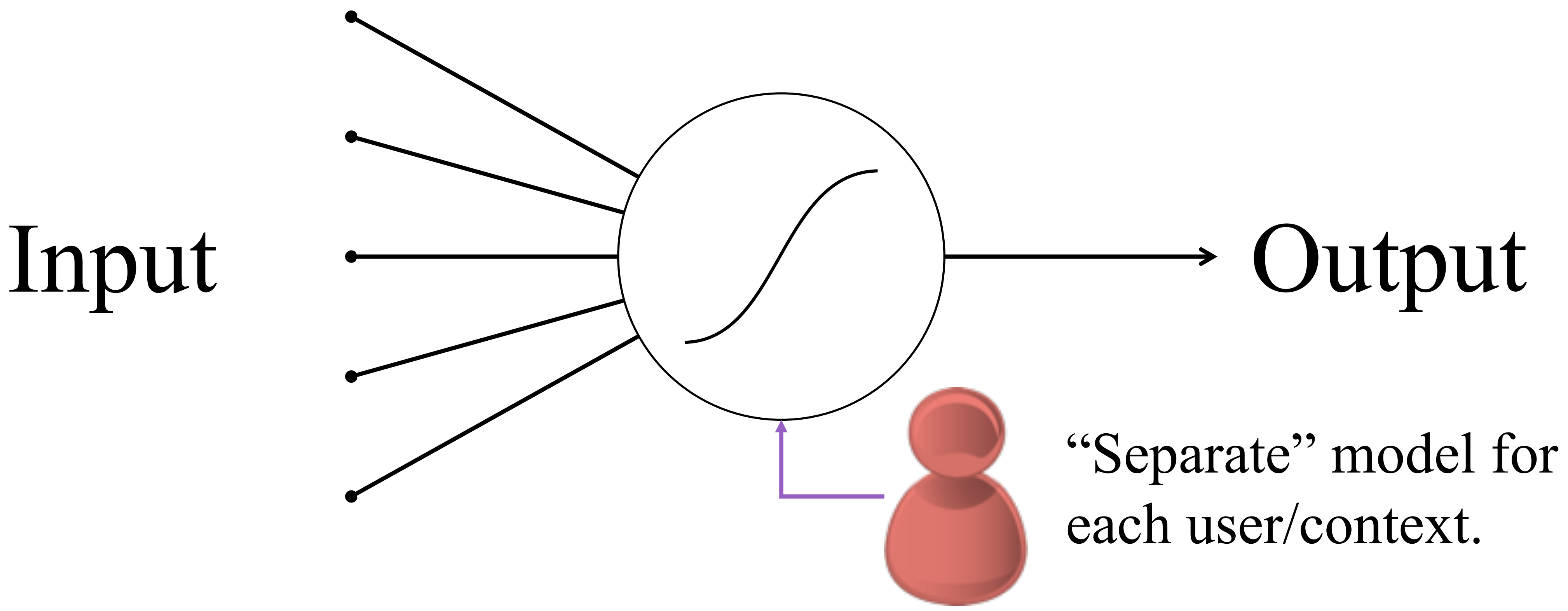
Localized Anomaly Detection



Velox Model Serving System

[CIDR'15, LearningSys'15]

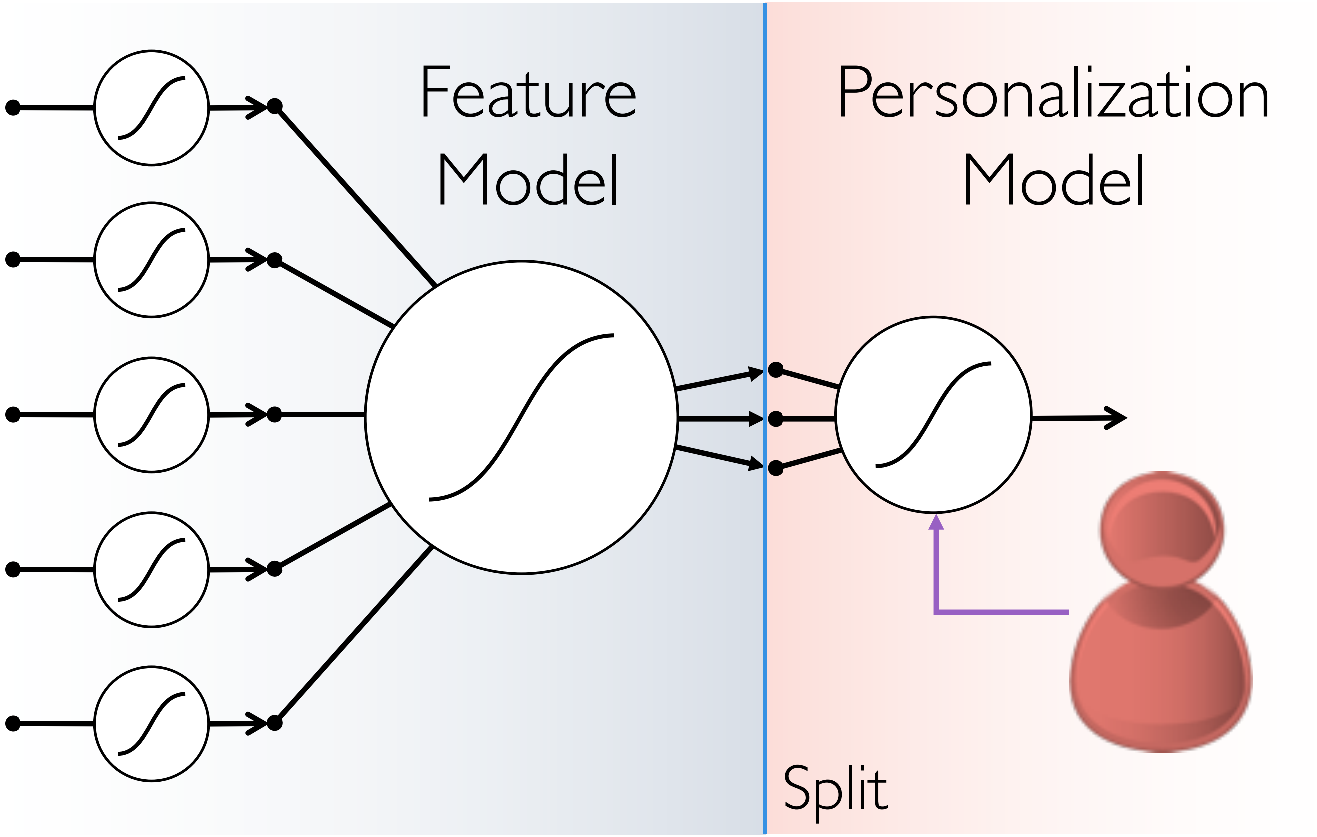
Personalized Models (Mulit-task Learning)



Velox Model Serving System

[CIDR'15, LearningSys'15]

Personalized Models (Mulit-task Learning)



Hybrid Offline + Online Learning

Update feature functions *offline* using batch solvers

- Leverage high-throughput systems (Apache Spark)
- Exploit slow change in population statistics

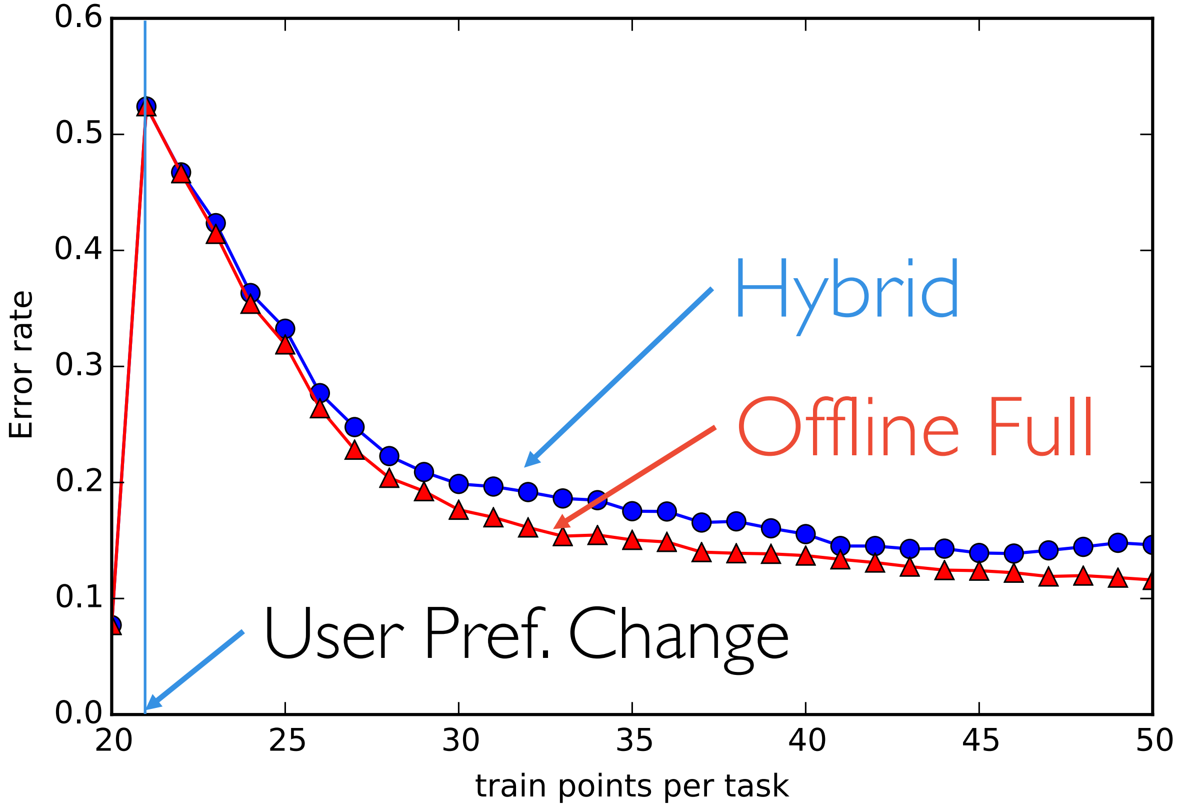
$$f(x;\theta)^T w_u$$

Update the user weights *online*:

- Simple to train + more robust model
- Address rapidly changing user statistics

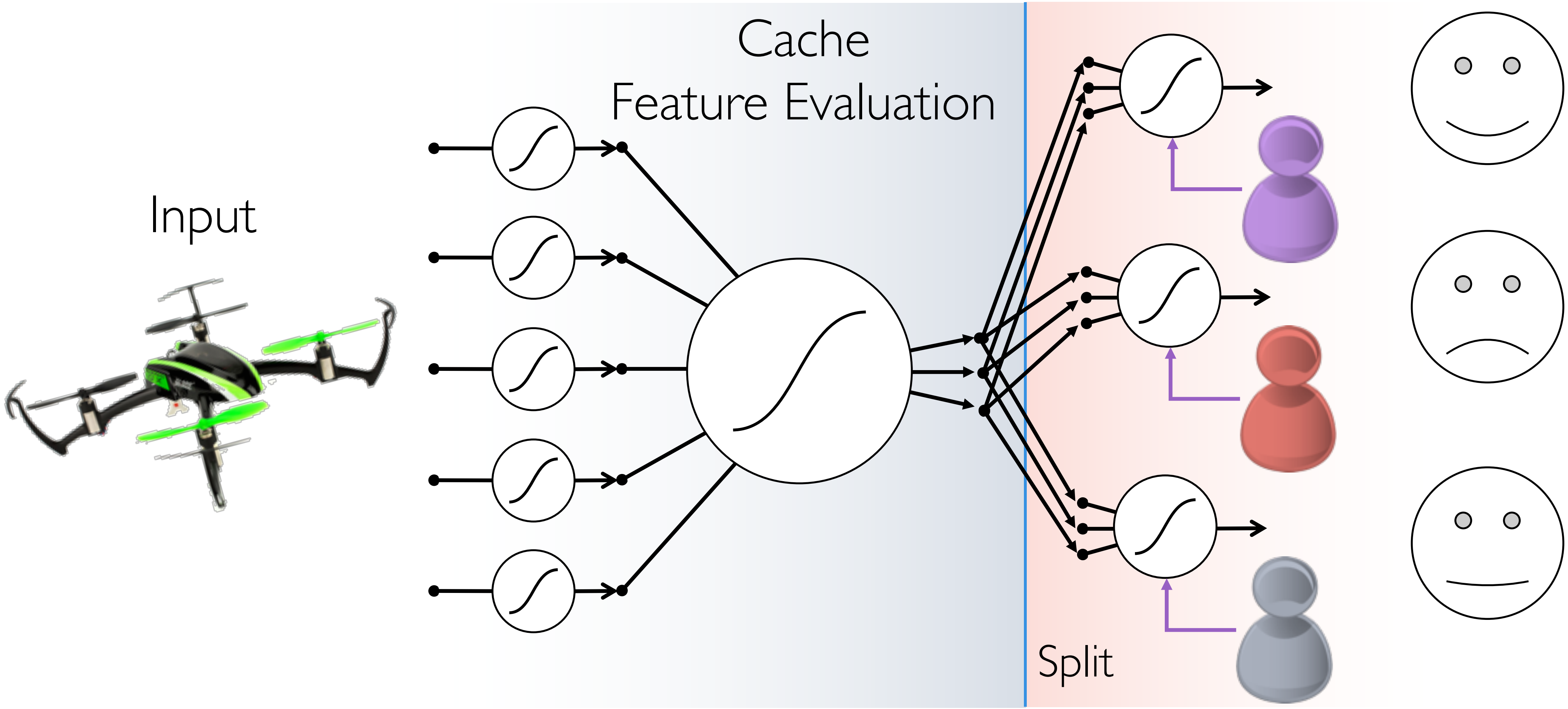
Hybrid Online + Offline Learning Results

Similar Test Error



Substantially Faster Training

Evaluating the Model



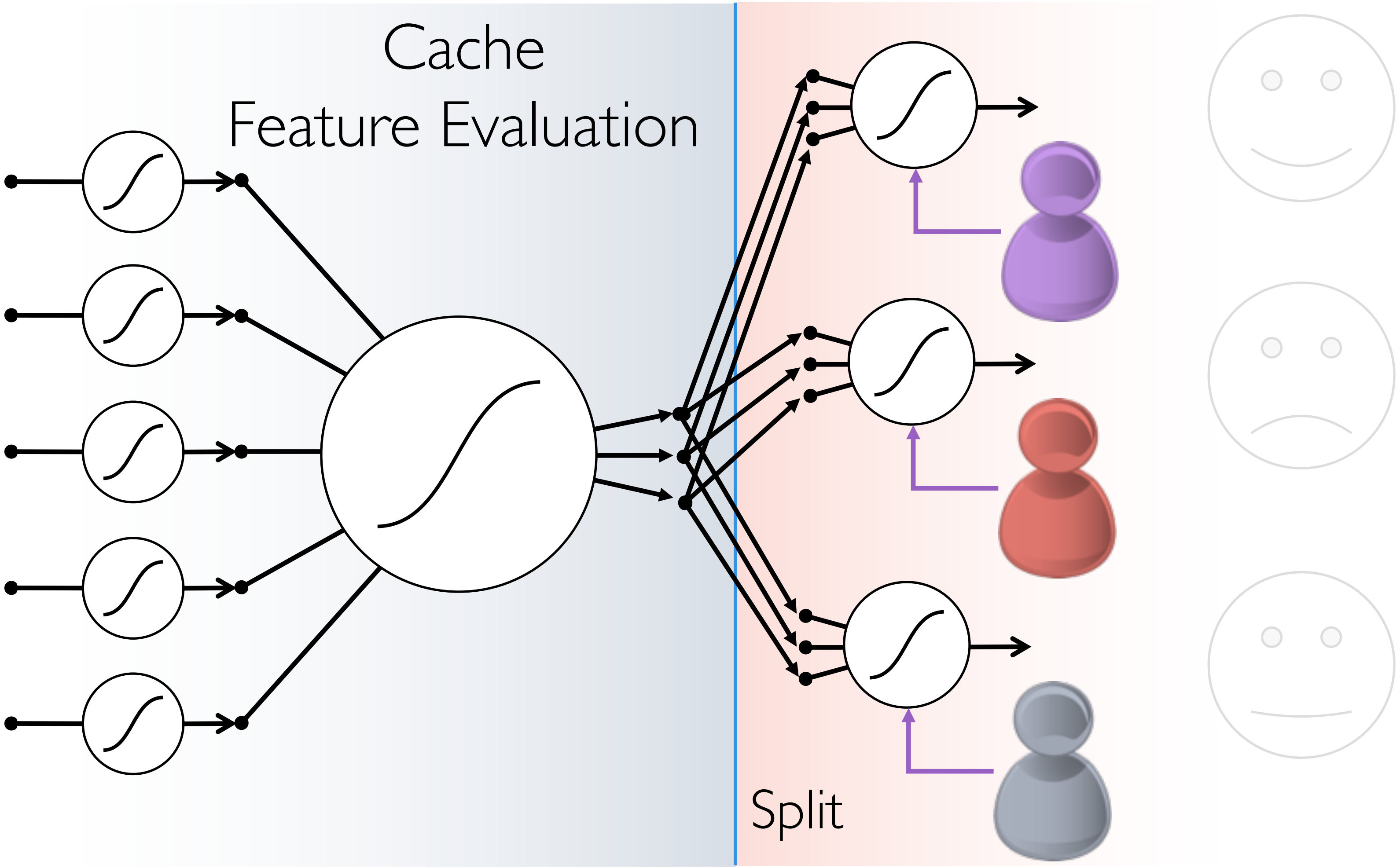
Evaluating the Model

Feature Caching
Across Users

Input

Approximate
Feature Hashing

Anytime Feature
Evaluation



Feature Caching

New input: x

Compute feature: $f(x;\theta)$

Hash input: $h(x)$ Store result in table

			$f(x;\theta)$	

Feature Hash Table

LSH Cache Coarsening

New input $z \neq x$

Hash new input: $h(z)$

False cache collision

Use Wrong Value!

→ LSH hash fn.

			$f(x;\theta)$	

Feature Hash Table

LSH Cache Coarsening

Locality-Sensitive Hashing:

$$x \approx z \quad \Rightarrow \quad h(x) = h(z)$$

Hash new input: $h(z)$

False cache collision

Use Value Anyways!

→ Req. LSH

Locality-Sensitive Caching:

$$f(x;\theta) \approx f(z;\theta) \quad \Rightarrow \quad h(x) = h(z)$$

$f(x;\theta)$

Feature Hash Table

Anytime Predictions

Compute features asynchronously:

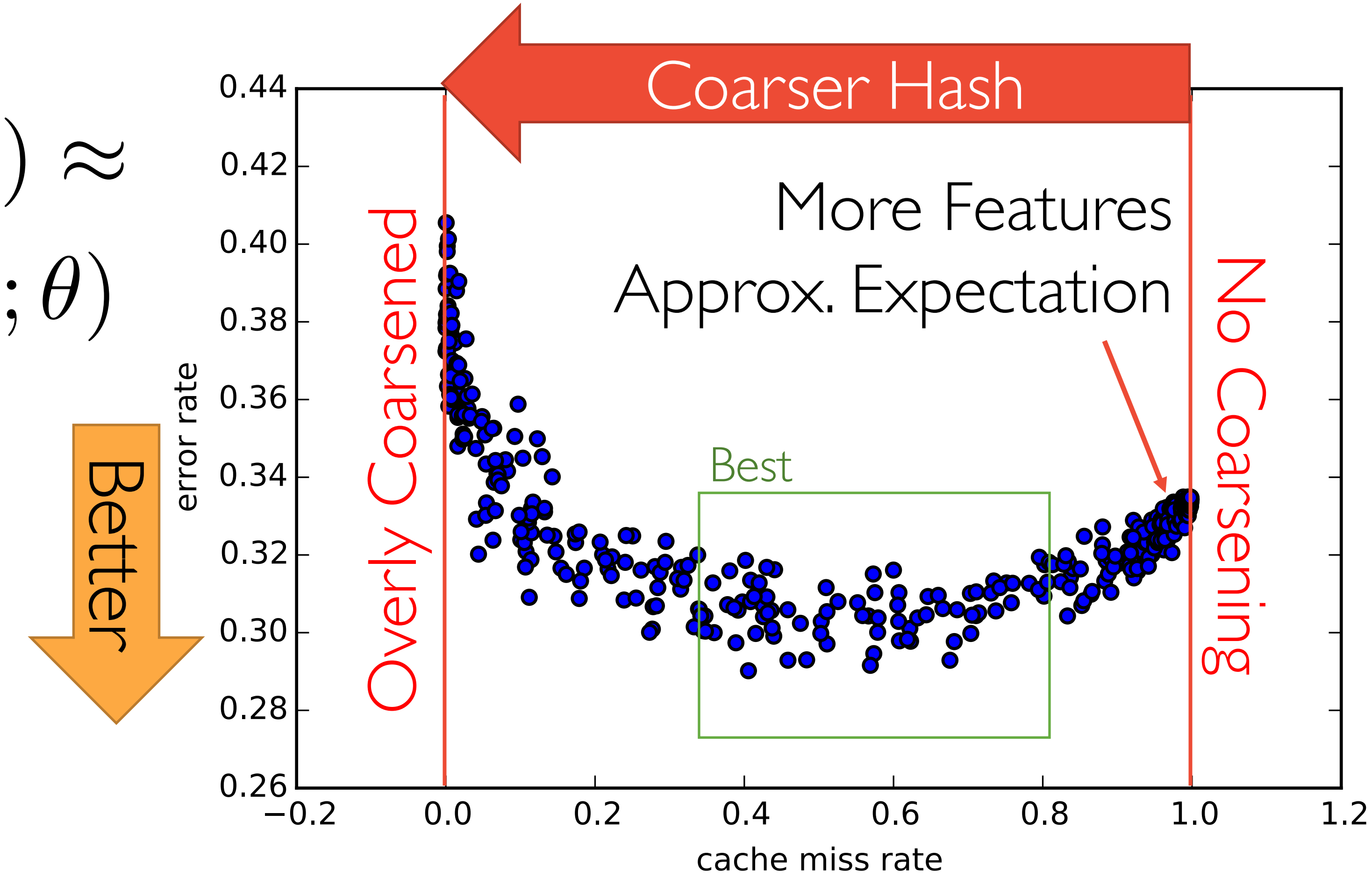
$$\underline{\hspace{3em}}\, w_{u1} + \underline{\hspace{3em}}\, w_{u2} + \underline{\hspace{3em}}\, w_{u3}$$

if a particular element does not arrive use estimator instead

Always able to render a prediction by the latency deadline

Coarsening + Anytime Predictions

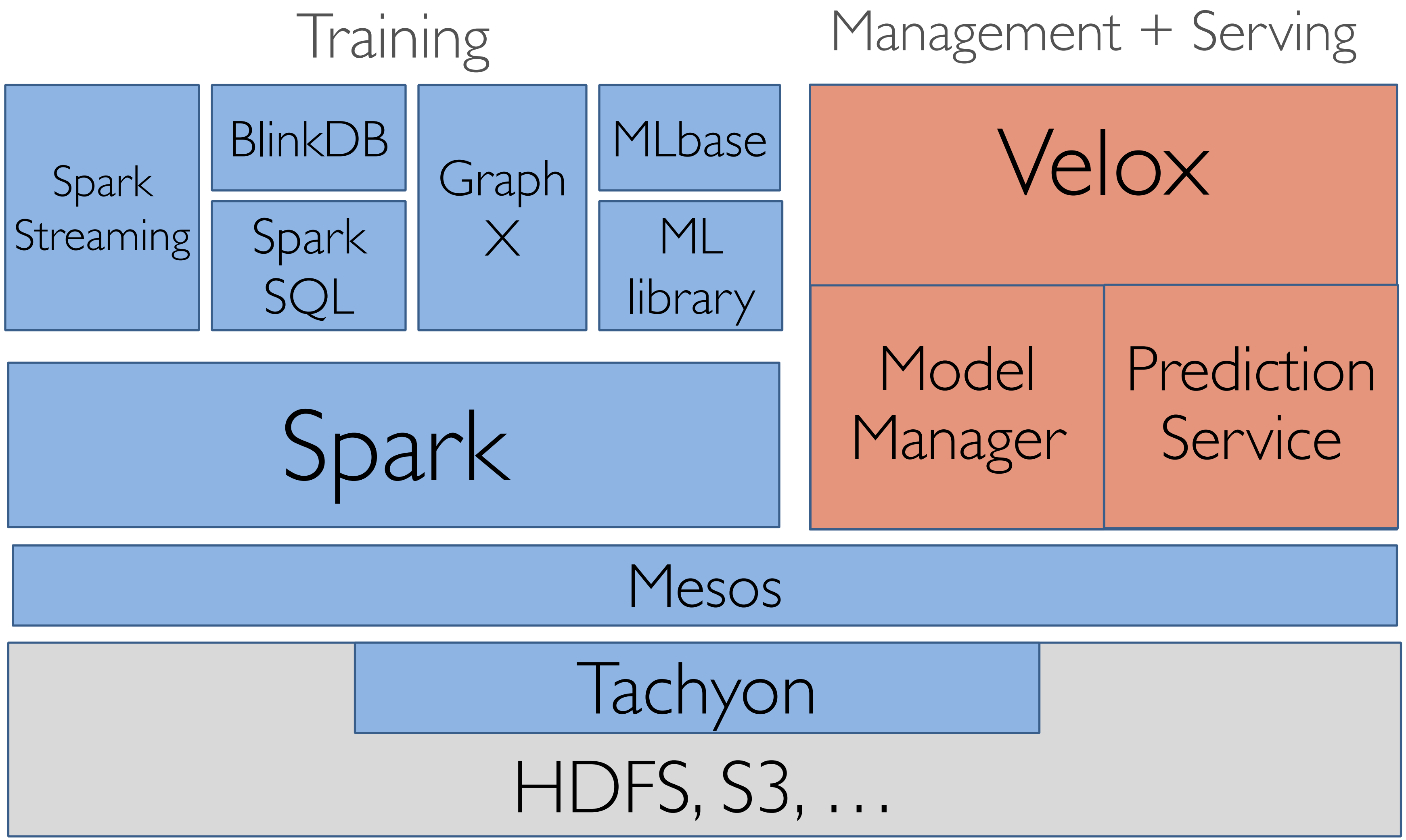
$$f_i(x;\theta) \approx f_i(z;\theta)$$



$$f_i(x;\theta) \approx \mathbb{E}\left[f_i(x;\theta)\right]$$

Checkout our poster!

Part of Berkeley Data Analytics Stack



VELOX

UC Berkeley AMPLab

Daniel Crankshaw, Xin Wang,
Peter Bailis, Haoyuan, Zhao Zhang,
Michael J. Franklin, Ali Ghodsi,
and Michael I. Jordan

Dato

Predictive Services

Dato Predictive Services

Production ready model serving and management system

- Elastic scaling and load balancing of docker.io containers
- AWS Cloudwatch Metrics and Reporting
- Serves Dato Create models, scikit-learn, and custom python
- Distributed shared caching: *scale-out to address latency*
- REST management API: Demo?

VELOX

UC Berkeley AMPLab

Daniel Crankshaw, Xin Wang, Joseph Gonzalez
Peter Bailis, Haoyuan, Zhao Zhang,
Michael J. Franklin, Ali Ghodsi,
and Michael I. Jordan

Dato

Predictive Services

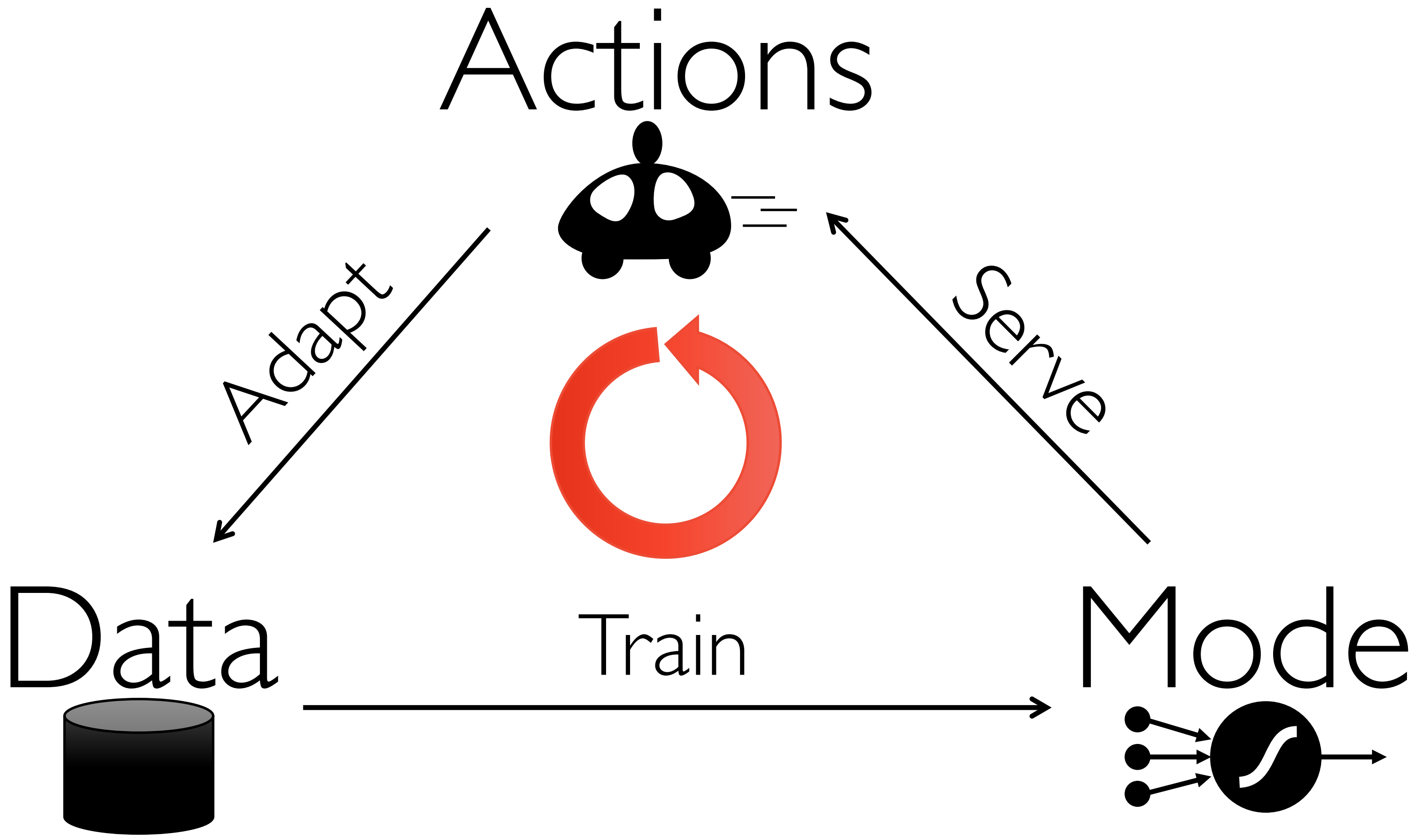
Responsive Adaptive Manageable

Key Insights:

Caching, Bandits, & Management

Online/Offline Learning
Latency vs. Accuracy

Future of Learning Systems



Thank You

Joseph E. Gonzalez

jegonzal@cs.berkeley.edu, Assistant Professor @ UC Berkeley

joseph@dato.com, Co-Founder @ Dato