What's next for ML & you

Emily Fox & Carlos Guestrin
Machine Learning Specialization
University of Washington

Deploying an ML service

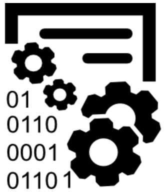
What is Production?

Serving live predictions

Deployment

Measuring quality of deployed models

Evaluation



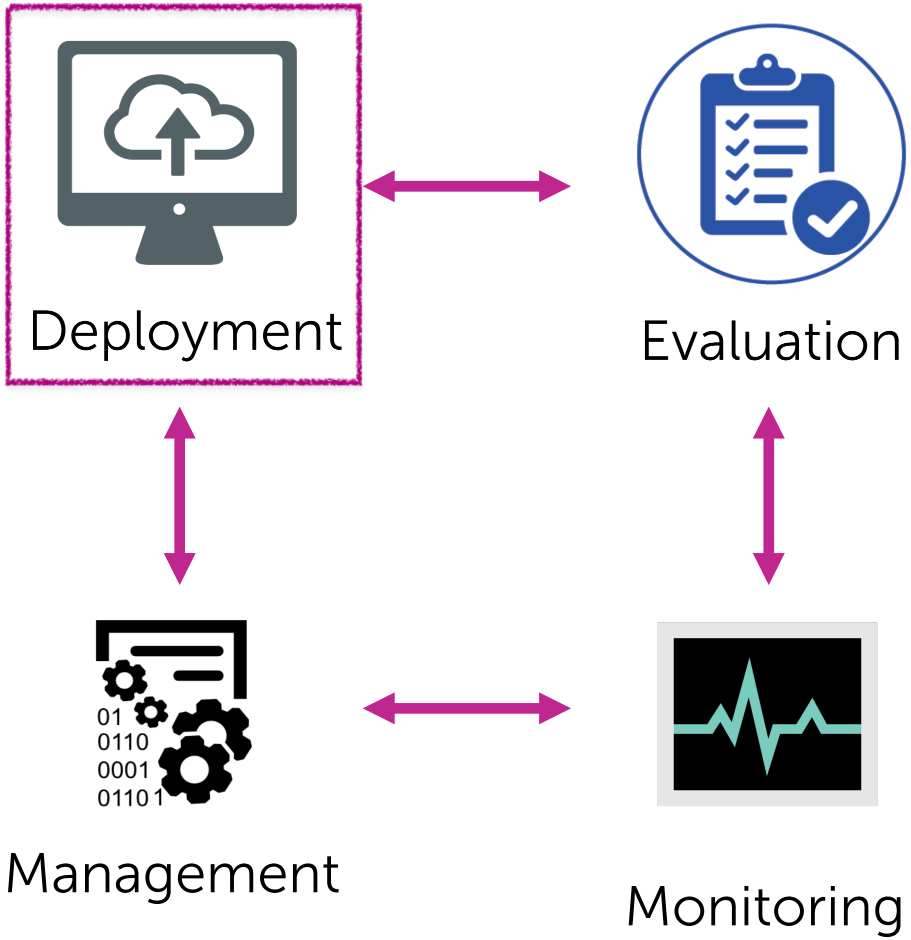
Choosing between deployed models

Management

Tracking model quality & operations

Monitoring

Lifecycle of ML in Production

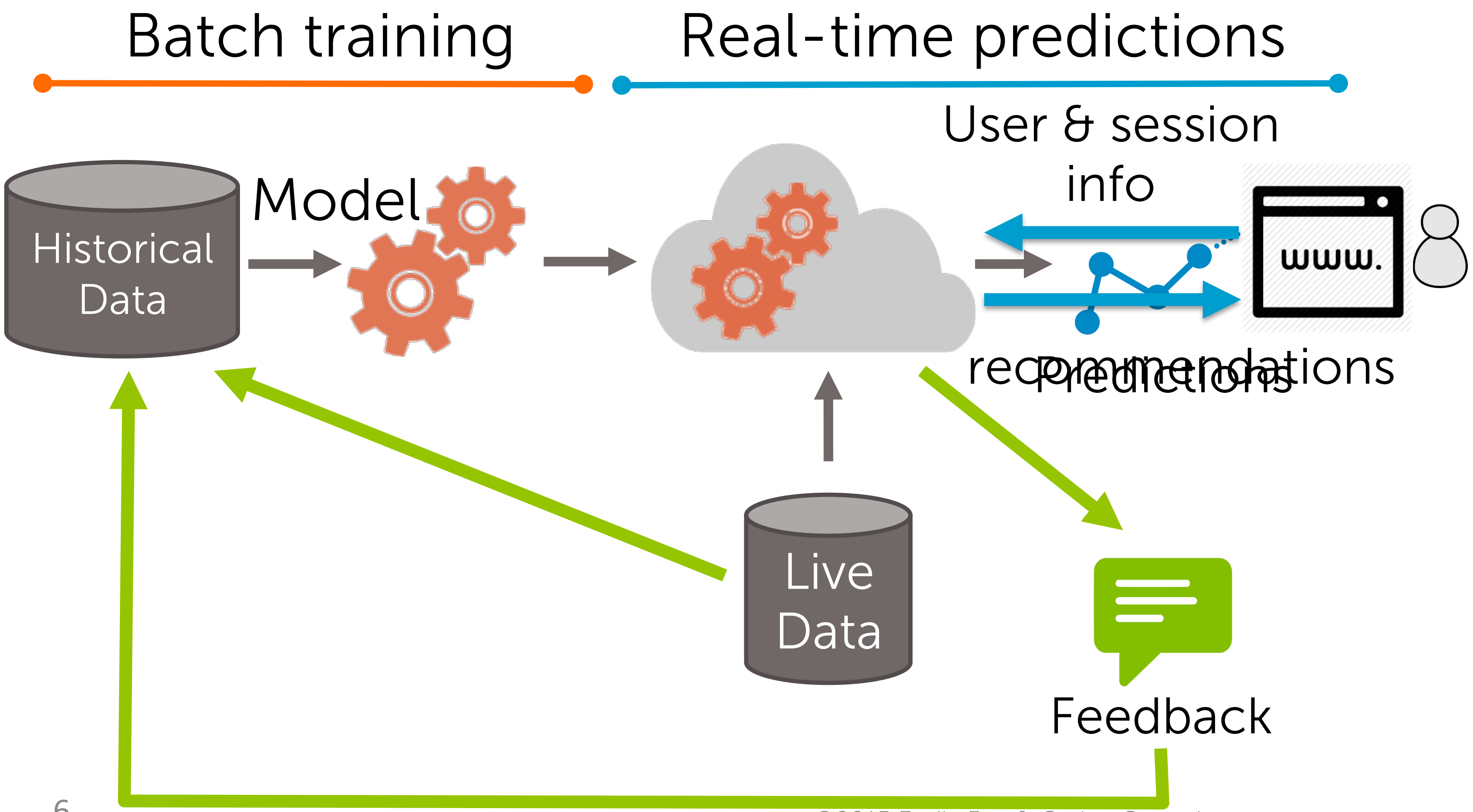


The Setup…

Suppose we are building a website with product recommendations, trained using user reviews.

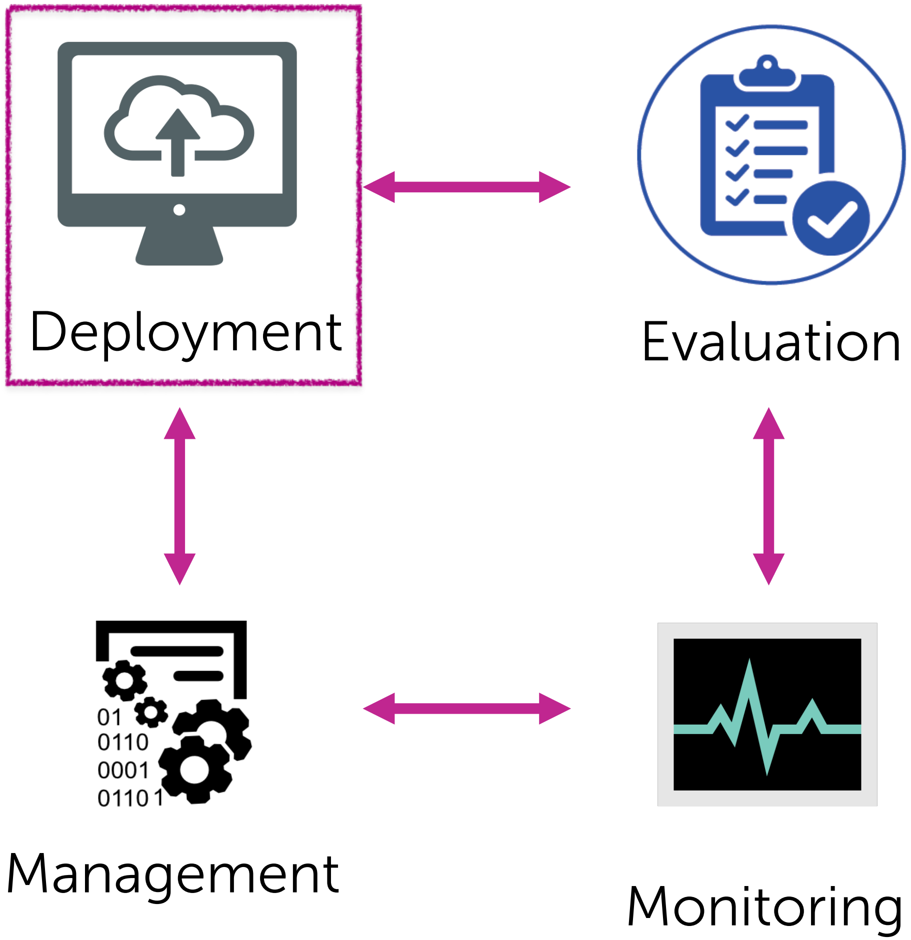
- 34.6M reviews
- 2.4M products
- 6.6M users

Deployment System



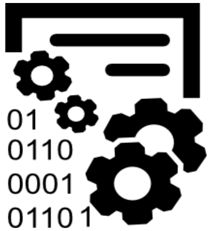
What happens after (initial) deployment

Lifecycle of ML in Production

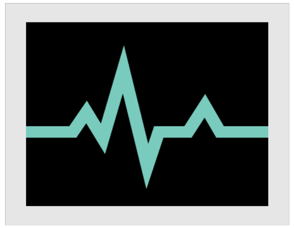


After deployment

Evaluation



Management

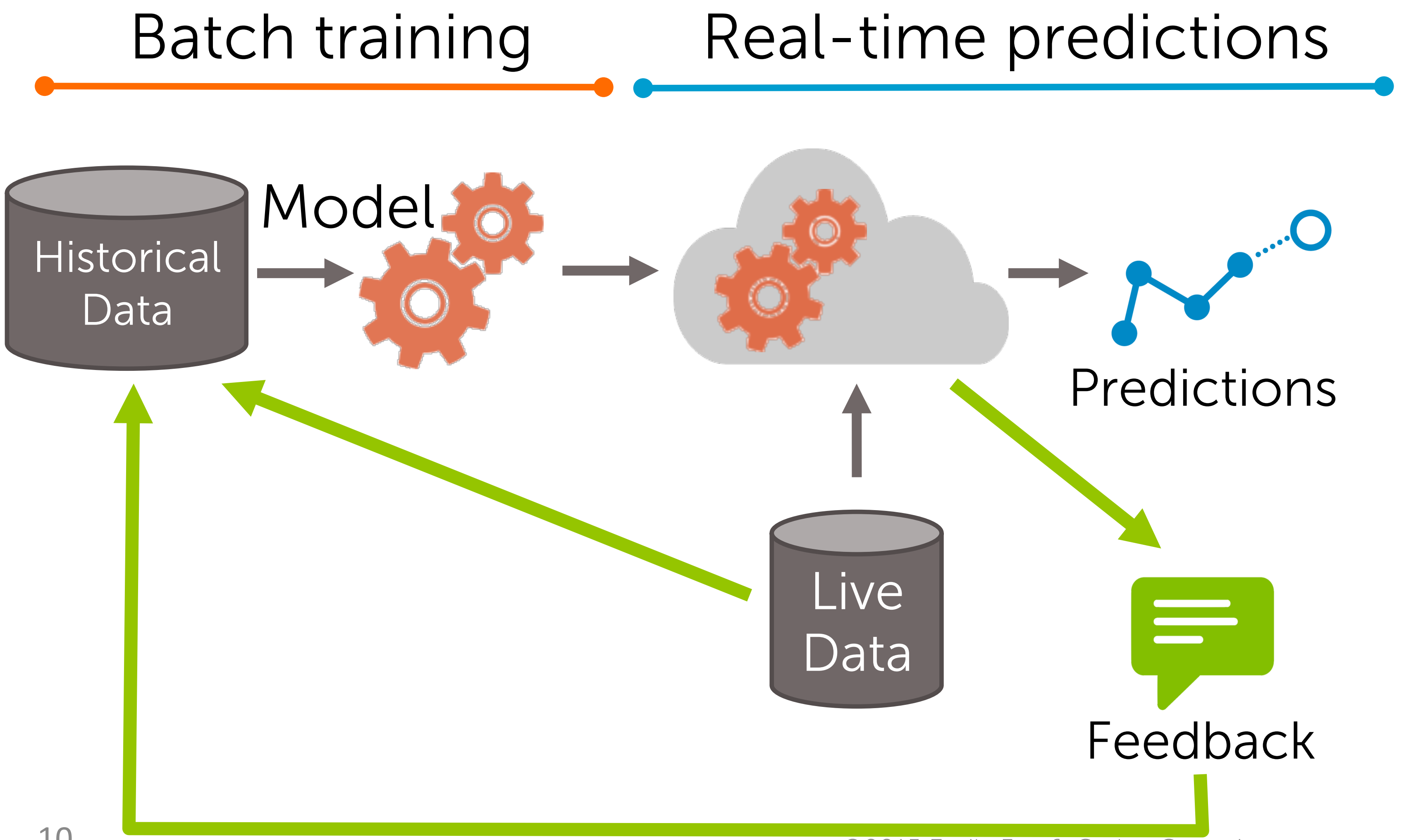


Monitoring

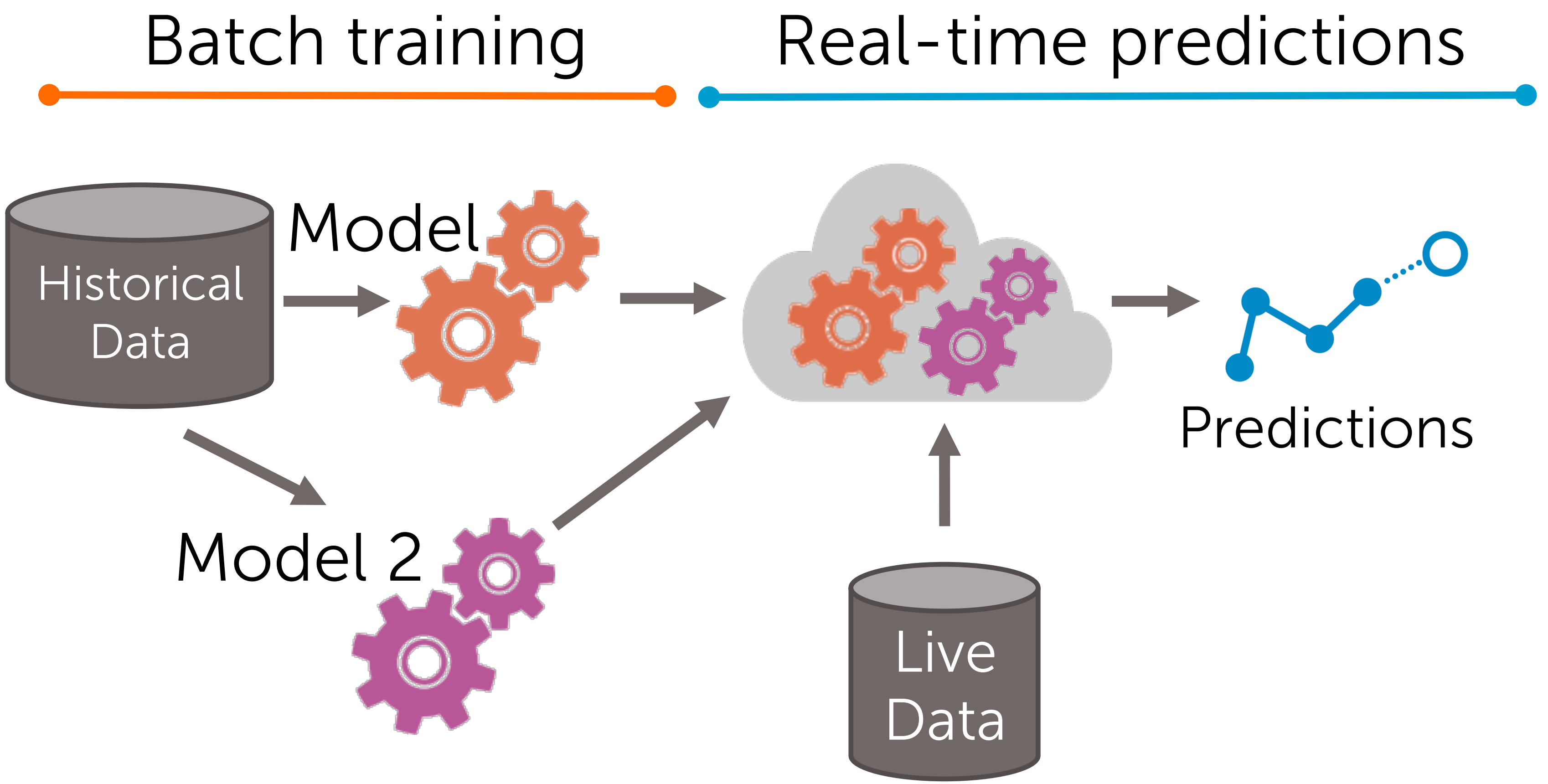
Evaluate and track metrics over time

React to feedback from deployed models

Feedback loop for ML in production



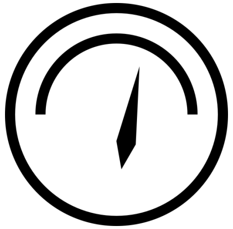
Learning new, alternative models



Key questions

- When to update a model?
- How to choose between existing models?
- Answer: continuous evaluation and testing

What is evaluation?

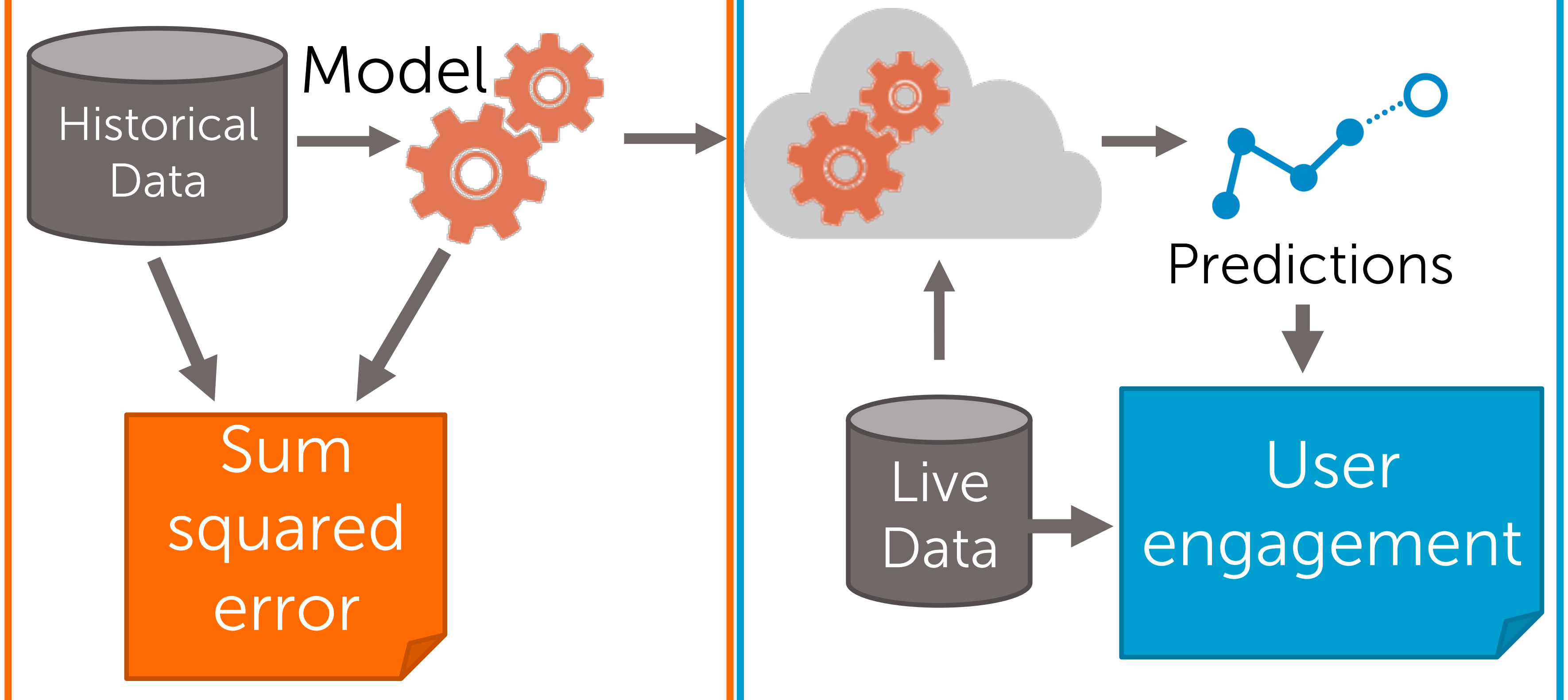
 =  + 

Evaluation = Predictions + Metric

What data?

Which metric?

Evaluating a recommender



Offline evaluation:
When to update model

Online evaluation:
Choosing between models

Updating ML models

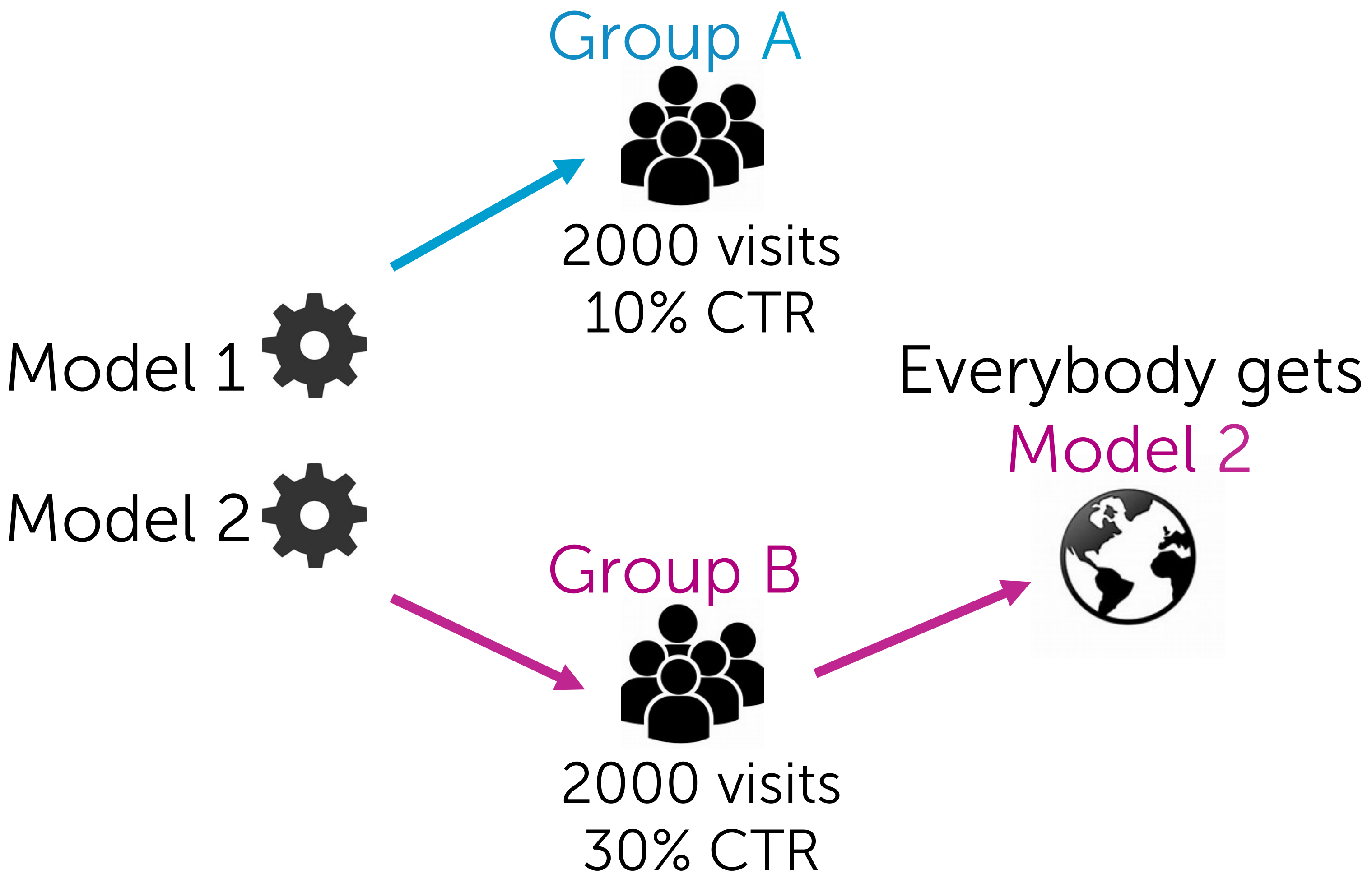
Why update?

- Trends and user tastes change over time
- Model performance drops

When to update?

- Track statistics of data over time
- Monitor both offline & online metrics
- Update when offline metric diverges from online metrics or not achieving desired targets

A/B Testing: Choosing between ML models

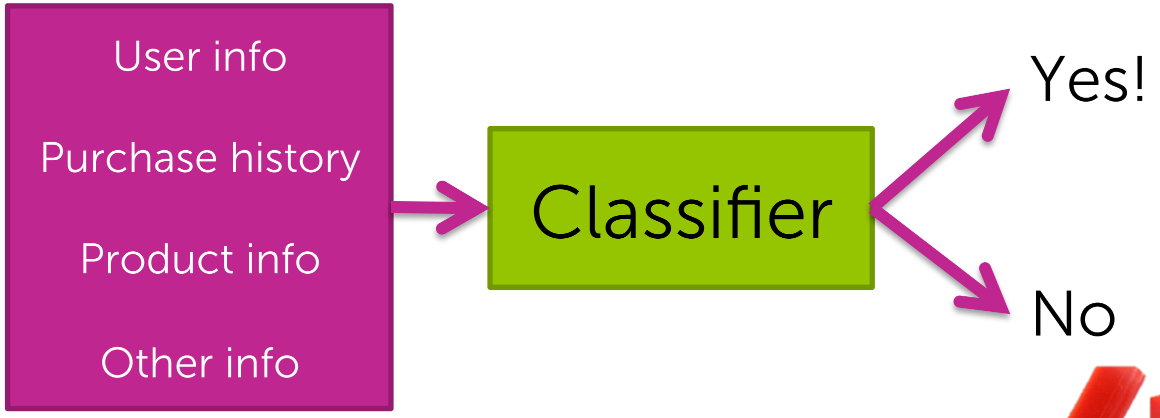


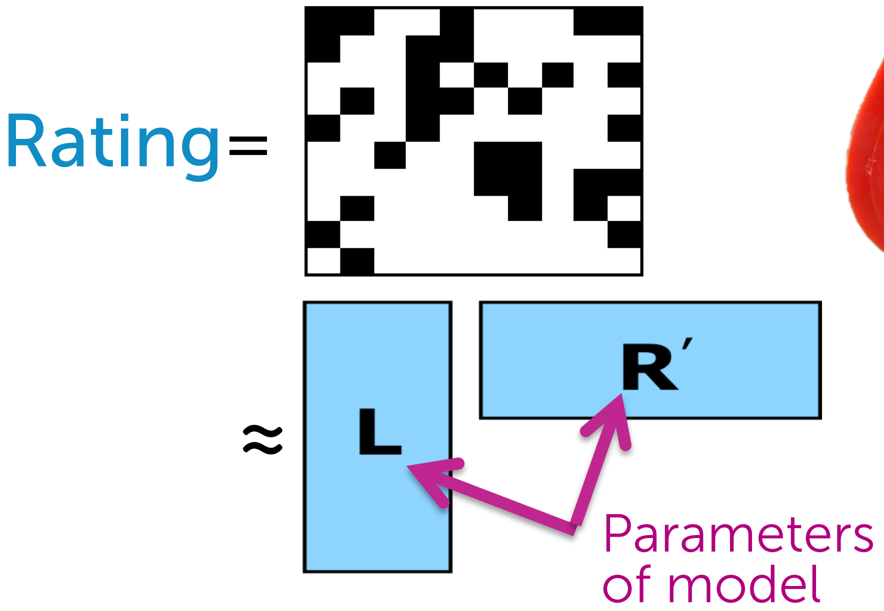
Other production considerations

- A/B testing caveats
  - Also multi-armed bandits
- Versioning
- Provenance
- Dashboards
- Reports
- …

Machine learning challenges

Open challenges: Model selection





Open challenges: Feature engineering/representation

1	0	0	0	5	3	0	0	1	0	0	0	0

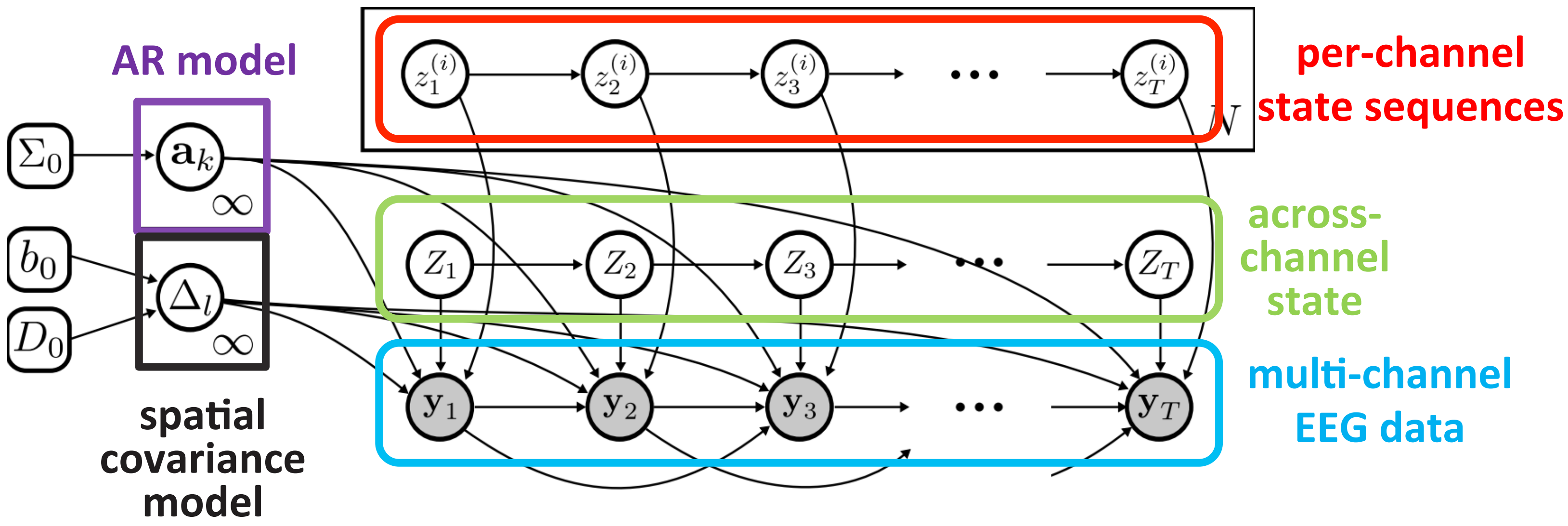
- Bag of word raw counts?
- Normalize?
- tf-idf? (which version???)
- Bigrams
- Trigrams
- …

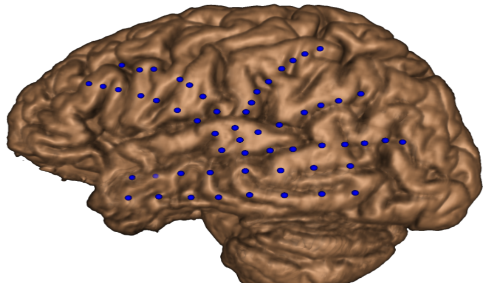
Open challenges: Scaling

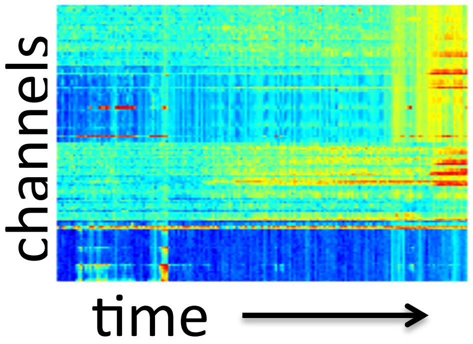
Data is getting big…

Open challenges: Scaling

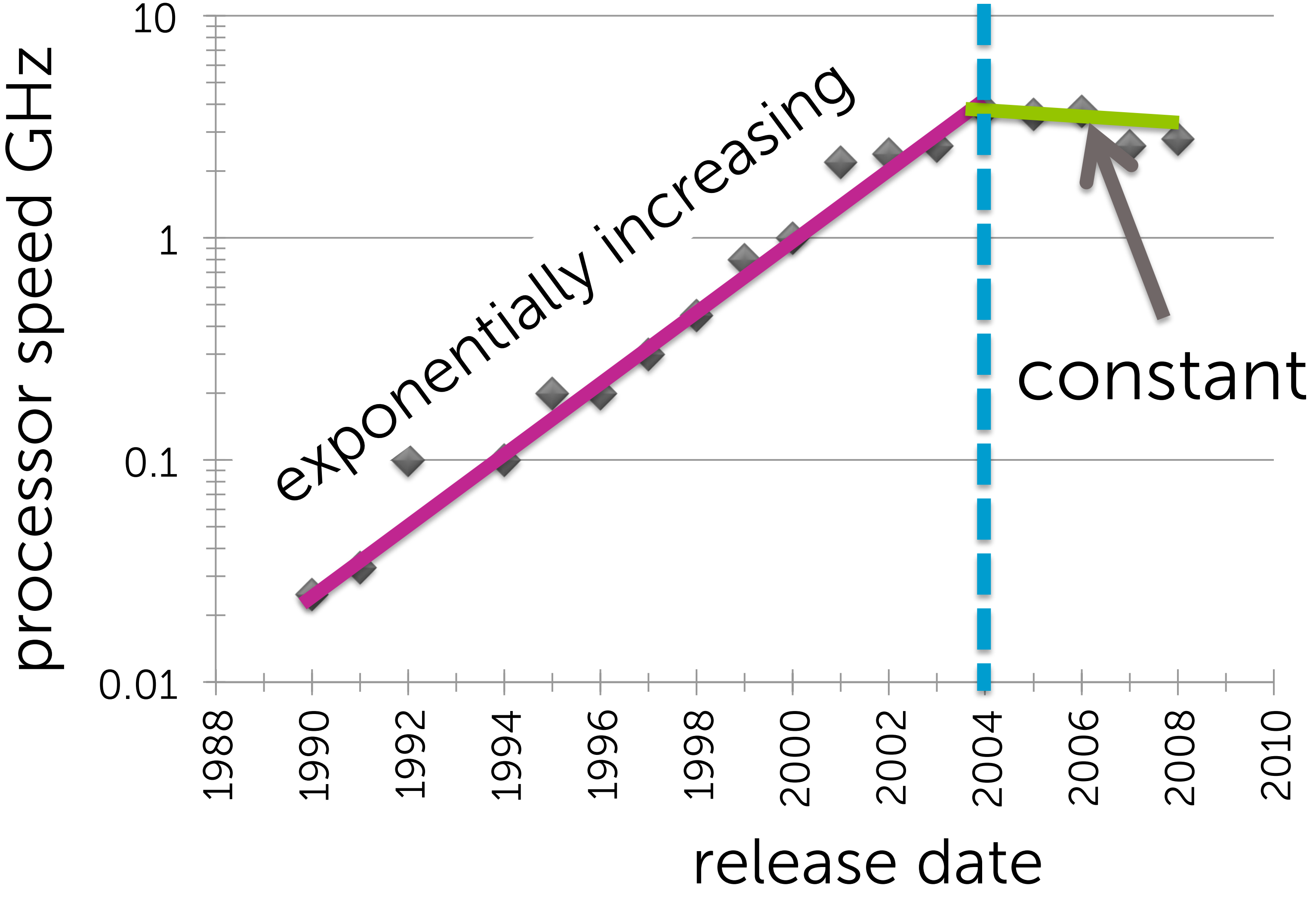
Concurrently, models are getting big…





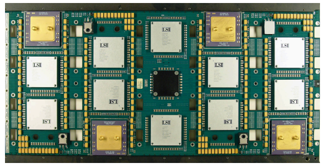


CPUs stopped getting faster…



ML in the context of parallel architectures

GPUs



Multicore

Clusters

amazon web services™

Clouds

Supercomputers

But scalable ML in these systems is **hard**, especially in terms of:

1. Programmability
2. Data distribution
3. Failures

What's ahead in this specialization

2. Regression

Case study: Predicting house prices

Models

- Linear regression
- Regularization: Ridge (L2), Lasso (L1)

Including many features:

- Square feet
- # bathrooms
- # bedrooms
- Lot size
- Year built
- ...

2. Regression

Case study: Predicting house prices

Algorithms

- Gradient descent
- Coordinate descent

$$
\begin{aligned}
\mathrm{RSS}(w_0,w_1) = \; & (\$_{\text{house 1}} - [w_0 + w_1 \text{sq.ft.}_{\text{house 1}}])^2 \\
& + (\$_{\text{house 2}} - [w_0 + w_1 \text{sq.ft.}_{\text{house 2}}])^2 \\
& + (\$_{\text{house 3}} - [w_0 + w_1 \text{sq.ft.}_{\text{house 3}}])^2 \\
& + \ldots \text{ [include all houses]}
\end{aligned}
$$

$$\downarrow$$

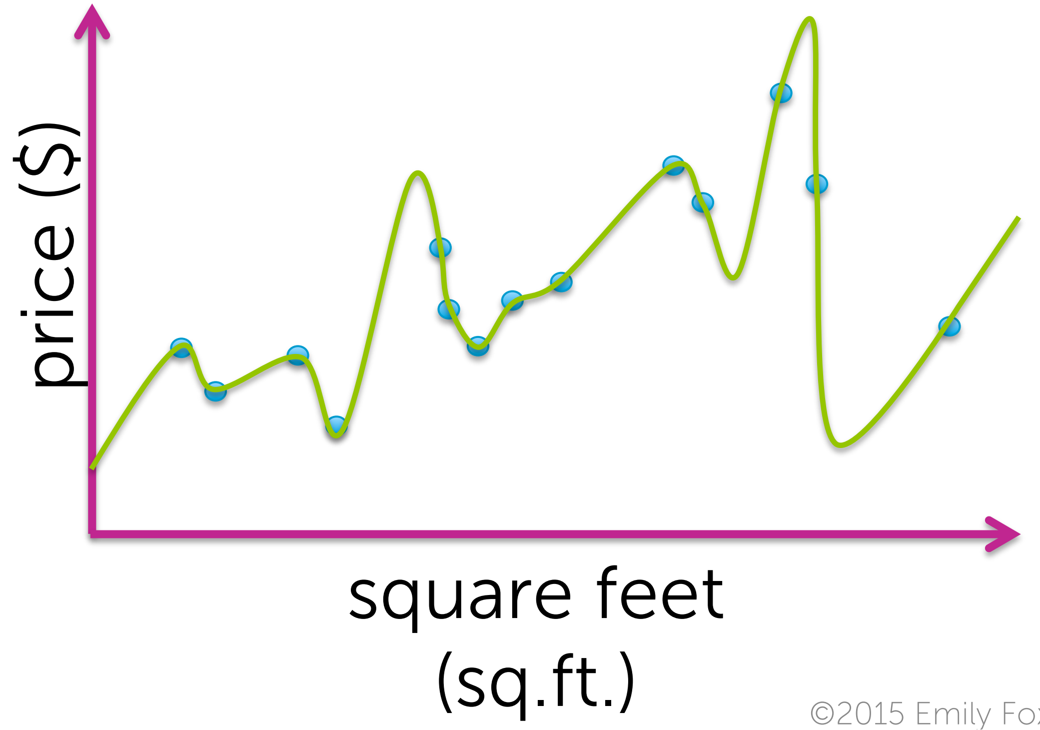
$$\hat{w}$$

2. Regression

Case study: Predicting house prices

Concepts

- Loss functions, bias-variance tradeoff, cross-validation, sparsity, overfitting, model selection

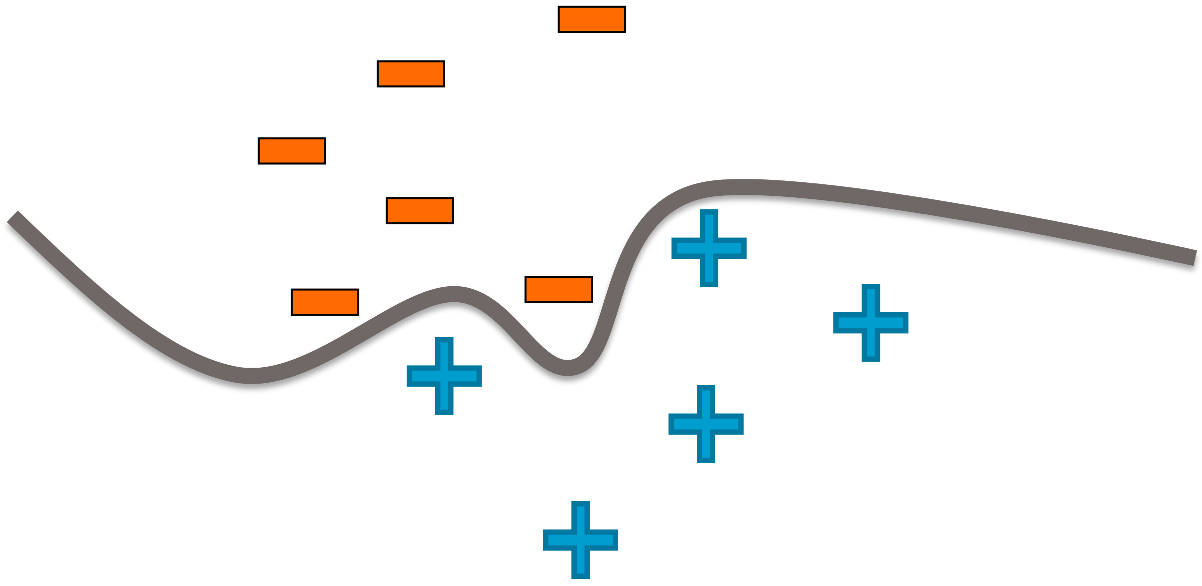


3. Classification

Case study: Analyzing sentiment

Models

- Linear classifiers
  (logistic regression, SVMs, perceptron)
- Kernels
- Decision trees



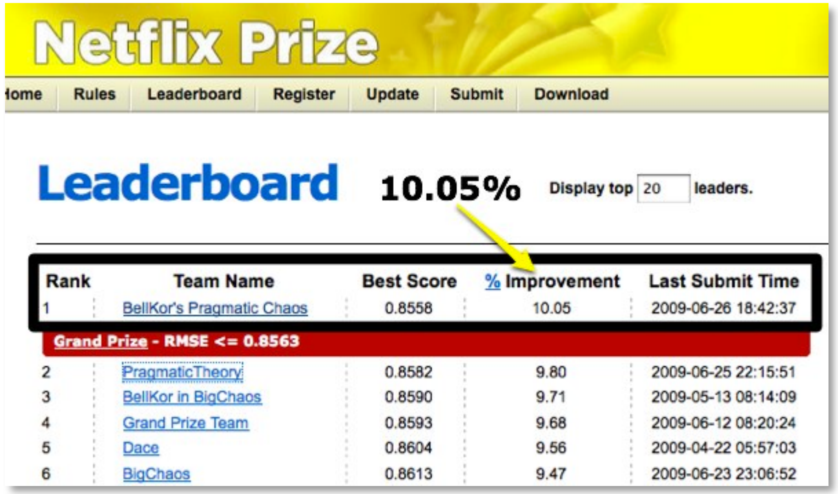
3. Classification

Case study: Analyzing sentiment

Algorithms

- Stochastic gradient descent
- Boosting

Squeezing last bit of accuracy by blending models

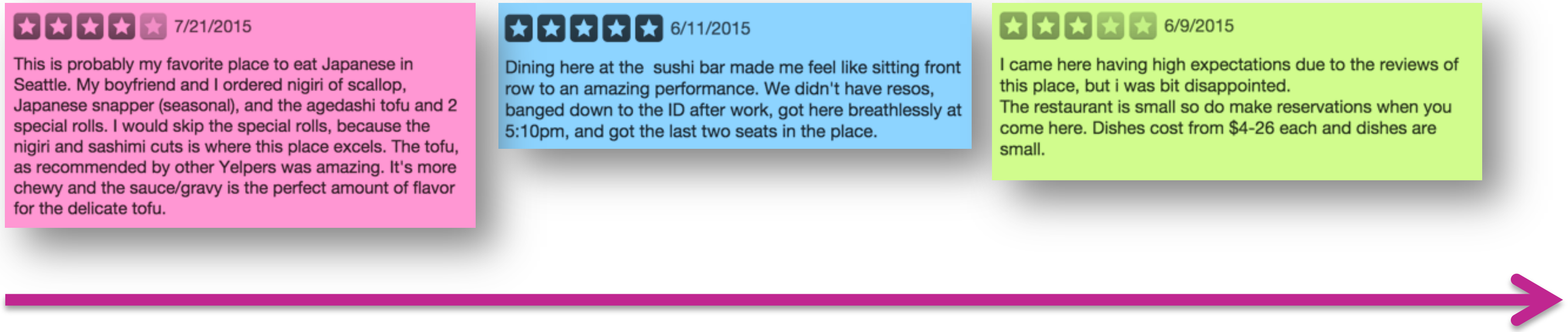


3. Classification

Case study: Analyzing sentiment

Concepts

- Decision boundaries, MLE, ensemble methods, random forests, CART, online learning



Time

4. Clustering & Retrieval

Case study: Finding documents

Models

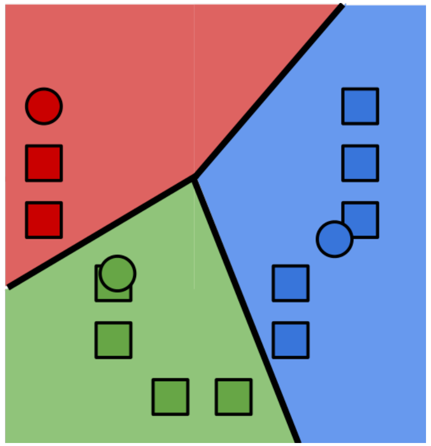
- Nearest neighbors
- Clustering, mixtures of Gaussians
- Latent Dirichlet allocation (LDA)

4. Clustering & Retrieval

Case study: Finding documents

Algorithms

- KD-trees, locality-sensitive hashing (LSH)
- K-means
- Expectation-maximization (EM)



4. Clustering & Retrieval

Case study: Finding documents

Concepts

- Distance metrics, approximation algorithms, hashing, sampling algorithms, scaling up with map-reduce

1	0	0	0	5	3	0	0	1	0	0	0	0

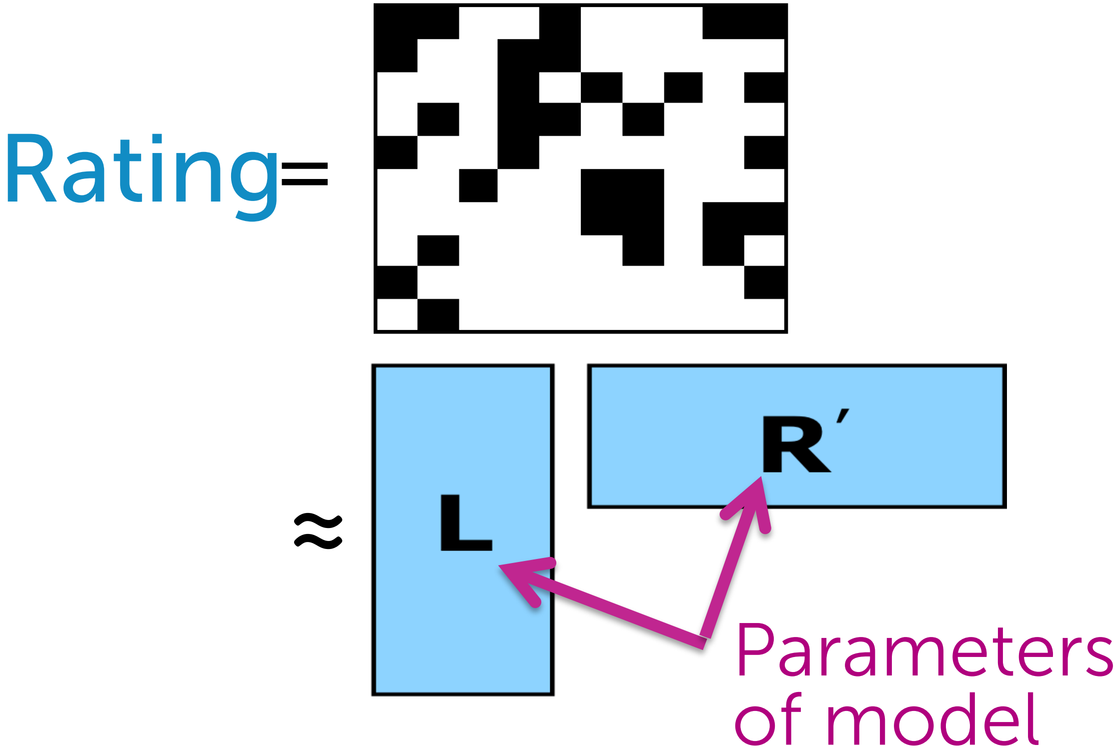
3	0	0	0	2	0	0	1	0	1	0	0	0

1*3
+
5*2
= 13

5. Recommender Systems & Dimensionality Reduction *Case study: Recommending Products*

Models

- Collaborative filtering
- Matrix factorization
- PCA



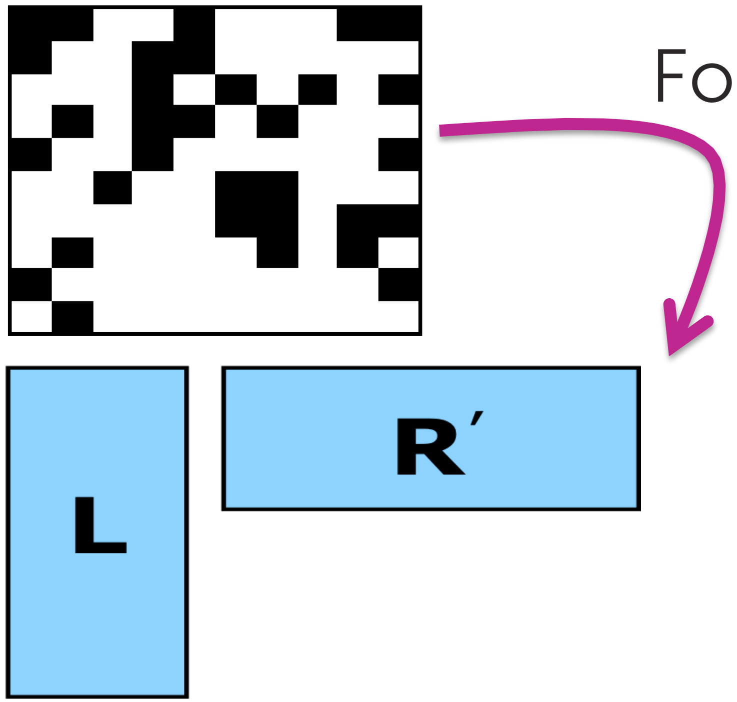
5. Matrix Factorization & Dimensionality Reduction

Case study: Recommending Products

Algorithms

- Coordinate descent
- Eigen decomposition
- SVD

Rating=



$\approx$

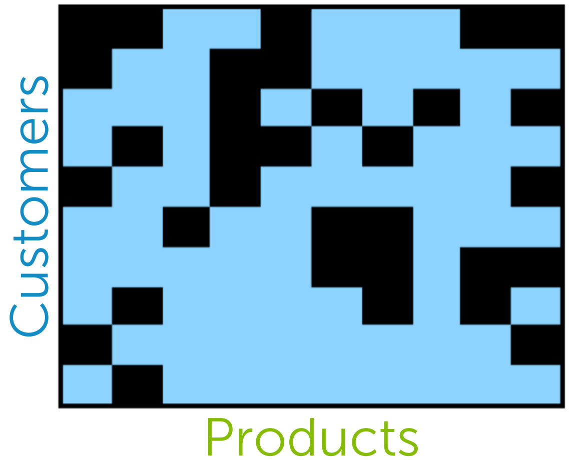
Form estimates $\hat{L}_u$ and $\hat{R}_v$

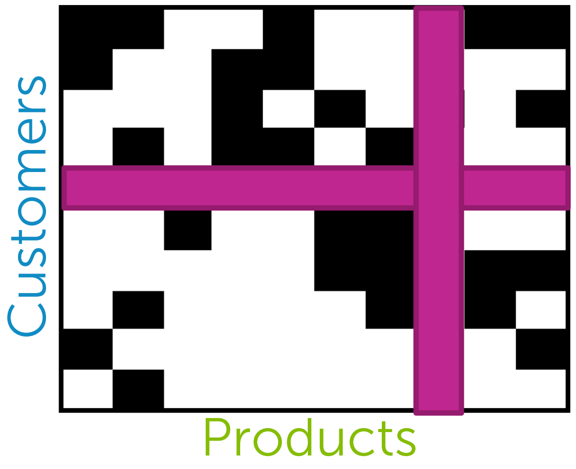
5. Matrix Factorization & Dimensionality Reduction

Case study: Recommending Products

Concepts

- Matrix completion, eigenvalues, random projections, cold-start problem, diversity, scaling up





6. Capstone: *Build and deploy an intelligent application with deep learning*