whoami

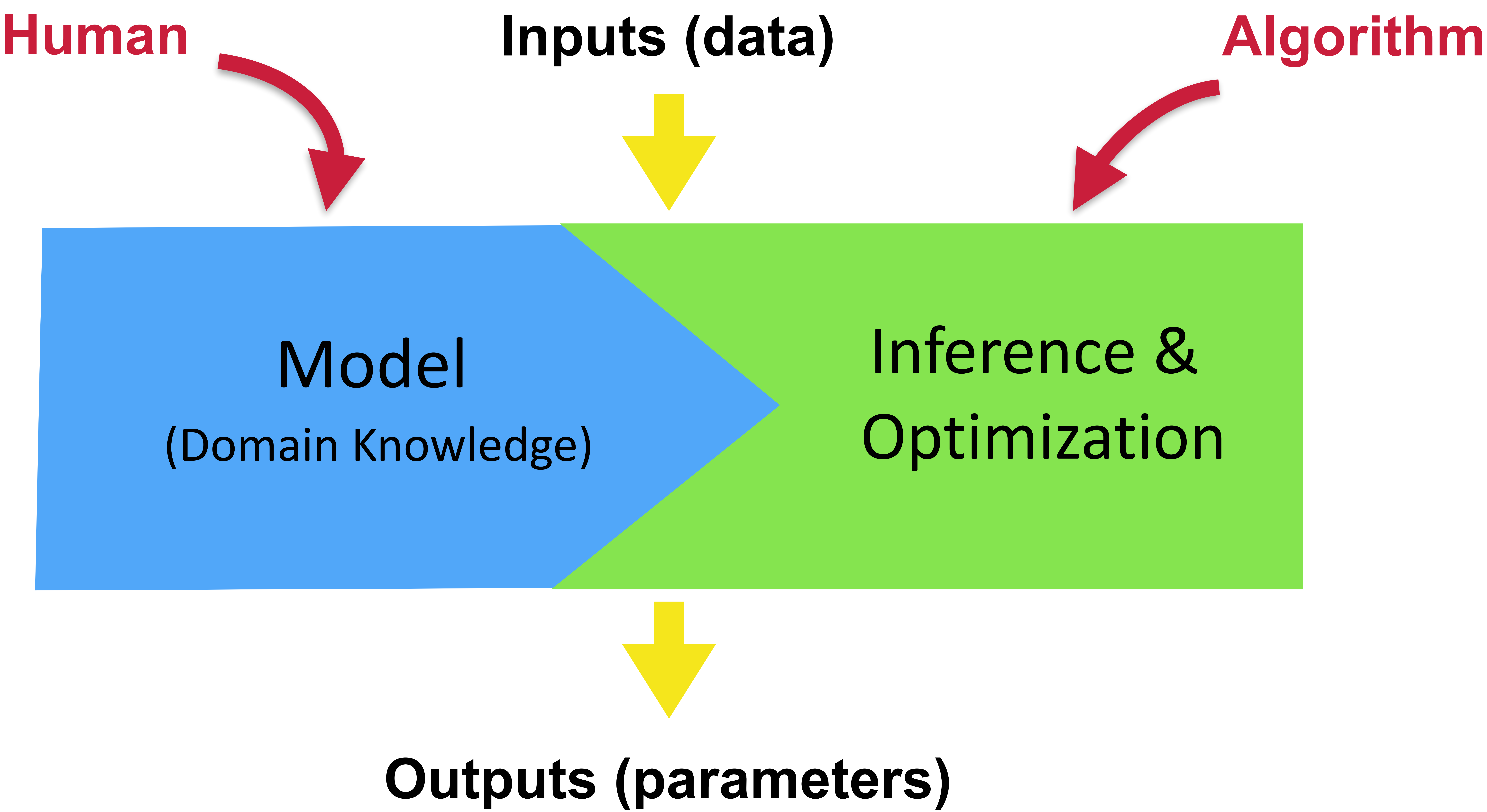
Consulting

- Human-in-the-loop machine learning + MLOps

inquiries@jonathan.industries

Getting the Materials

https://github.com/jonathandinu/spark-livetraining


Human
Inputs (data)
Algorithm
Model
(Domain Knowledge)
Inference &
Optimization
Outputs (parameters)

Types of Learning

Supervised Learning

- Training data includes desired output

Unsupervised Learning

- Training data does not include desired output

Semi-supervised Learning

- Training data includes some desired outputs

Reinforcement Learning

- Rewards from sequence of actions

Types of Learning

Supervised Learning

- Training Data includes desired output

Unsupervised Learning

- Training Data does not include desired output

Semi-supervised Learning

- Training Data includes some desired outputs

Reinforcement Learning

- Rewards from sequence of actions

Iris Dataset

	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)	label
0	5.1	3.5	1.4	0.2	0
1	4.9	3.0	1.4	0.2	0
2	4.7	3.2	1.3	0.2	0
3	4.6	3.1	1.5	0.2	0
4	5.0	3.6	1.4	0.2	0

Features
(feature matrix)

Target

What to learn an unknown target function f()

Input: labeled training set (x_i, y_i)

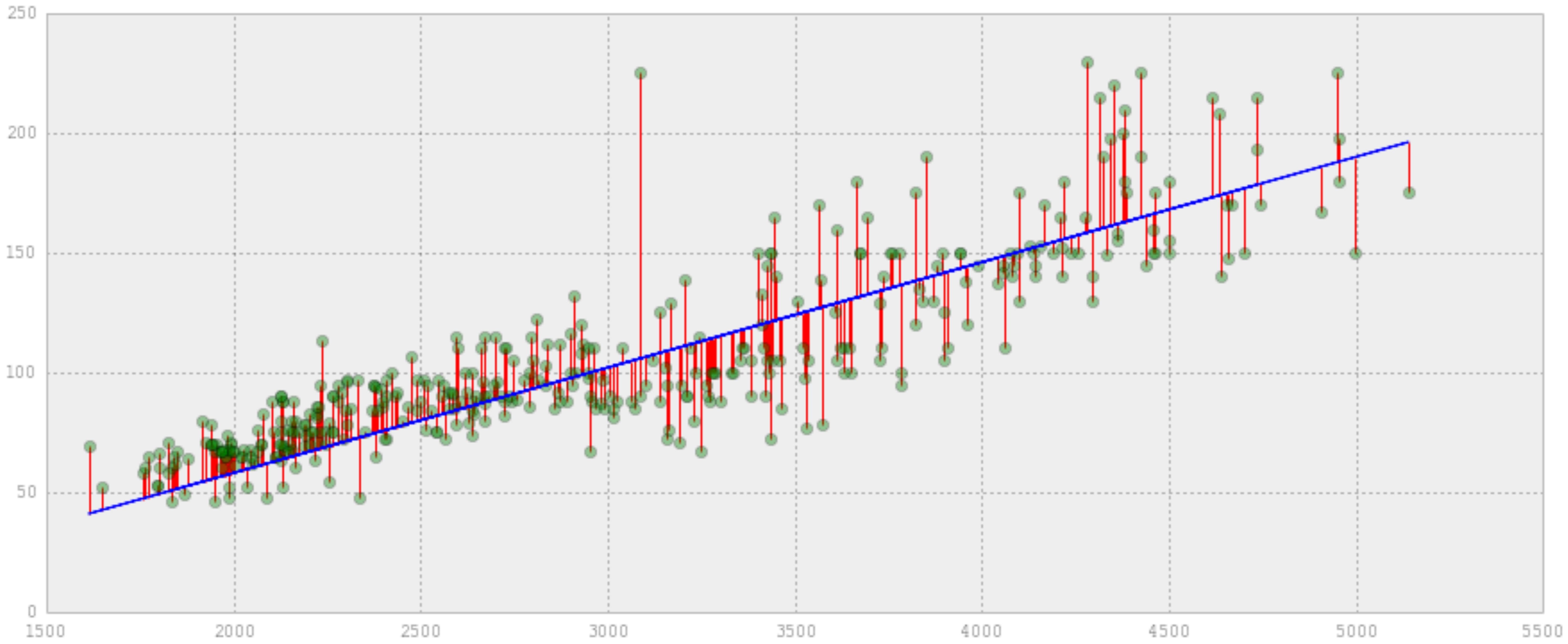
- y_i = f(x_i)

Output: hypothesis h() function “close” to f()

Many possible hypothesis families:

- Logistic
- Linear
- decision trees
- example-based (nearest neighbor)
- etc.

Linear Regression



Parameters

$$A = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_n x_n$$

Logistic Regression

$$\mathrm{A} = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_n x_n$$

$$P(label \mid X) = \sigma(A)$$

$$\sigma = \frac{1}{1 + e^{-A}}$$

(function bound between 0 and 1)

Logistic Regression

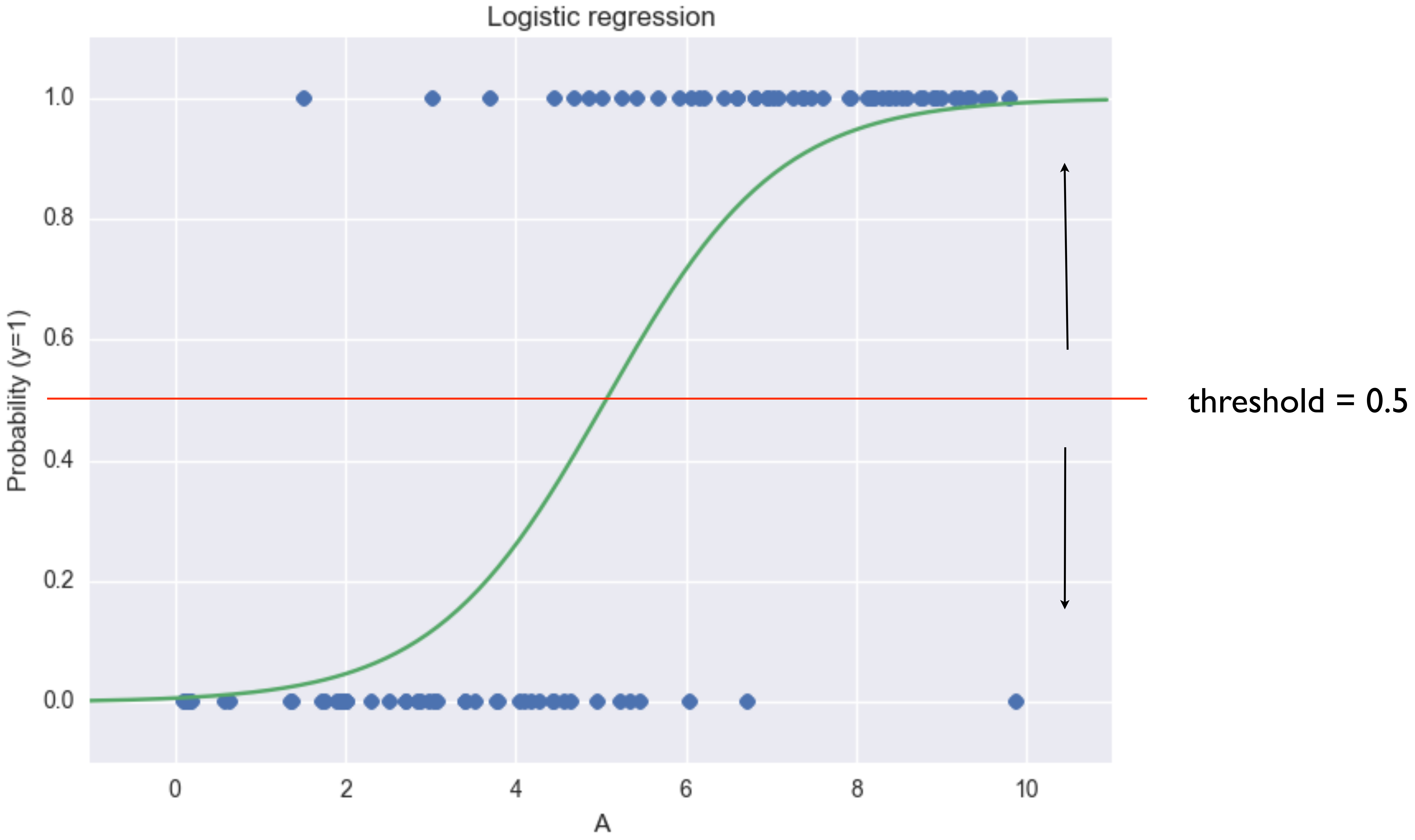
$$\mathrm{A} = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_n x_n$$

$$P(label \mid X) = \sigma(A)$$

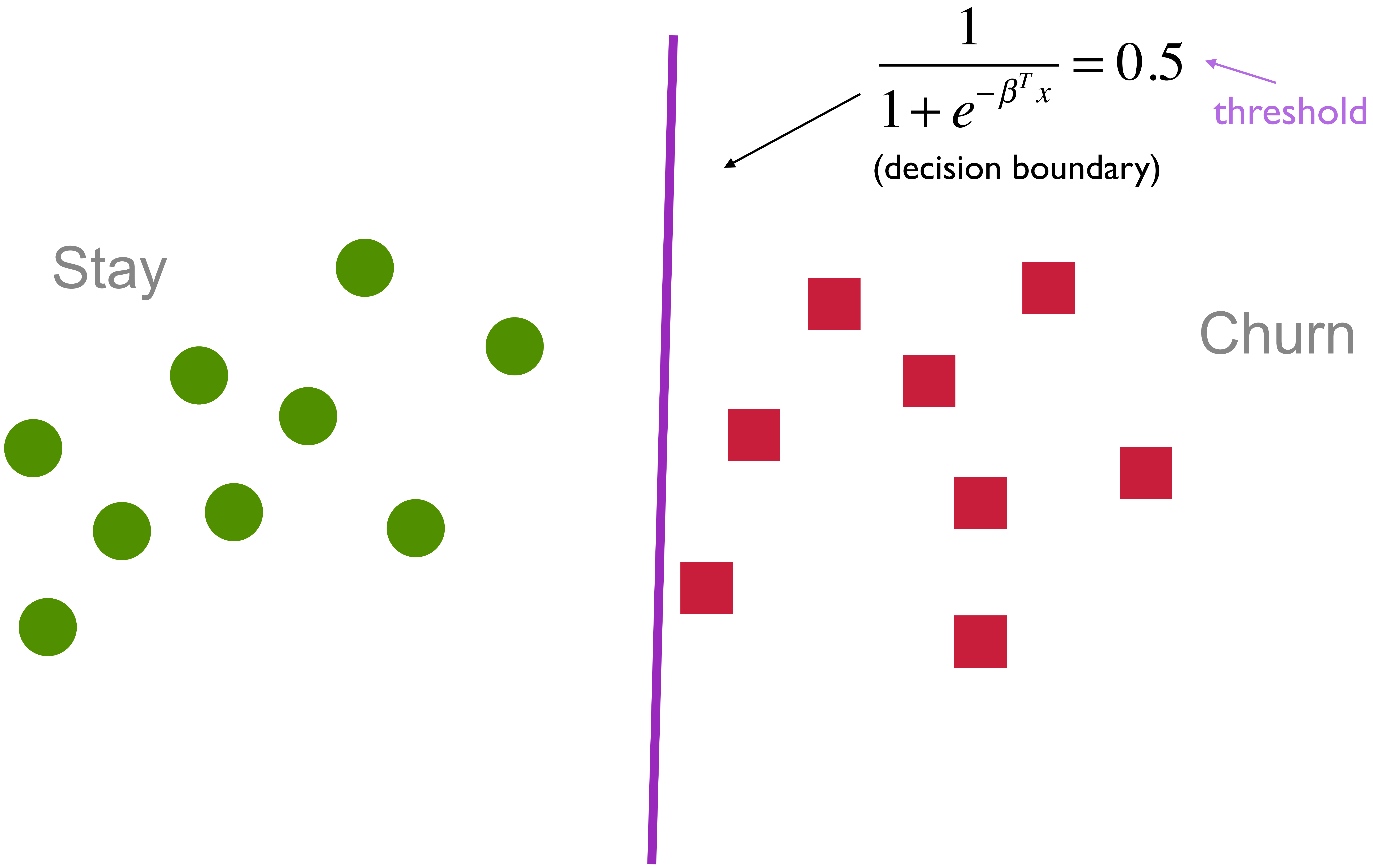
$$p = \frac{1}{1 + e^{-(\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_n X_n)}}$$

Logistic Regression

(contrary to its name... actually used to classify)



Linear Separator



Interpreting Logistic Regression

Log odds

$$p = \frac{1}{1 + e^{-(\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_n X_n)}}$$

$$\frac{p}{1-p} = e^{\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_n X_n} = e^{\beta_0} e^{\beta_1 X_1} e^{\beta_2 X_2} \ldots e^{\beta_n X_n}$$

A one unit change in X_1 increases the odds ratio by e^{β_1}

Solving for Parameters

1. Analytically with differential calculus

2. Computationally with optimization methods

3. Approximately with iterative methods

Solving for Parameters

1. Analytically with differential calculus
2. Computationally with optimization methods
3. **Approximately with iterative methods**

What are the most likely parameters given the data we have?

$$P(\theta \mid x_1, x_2, \ldots, x_n)$$

$$y = f(\mathbf{X}) + \boldsymbol{\varepsilon}$$

$$y = f(\mathbf{X}, \boldsymbol{\beta}) + \varepsilon$$

Cost

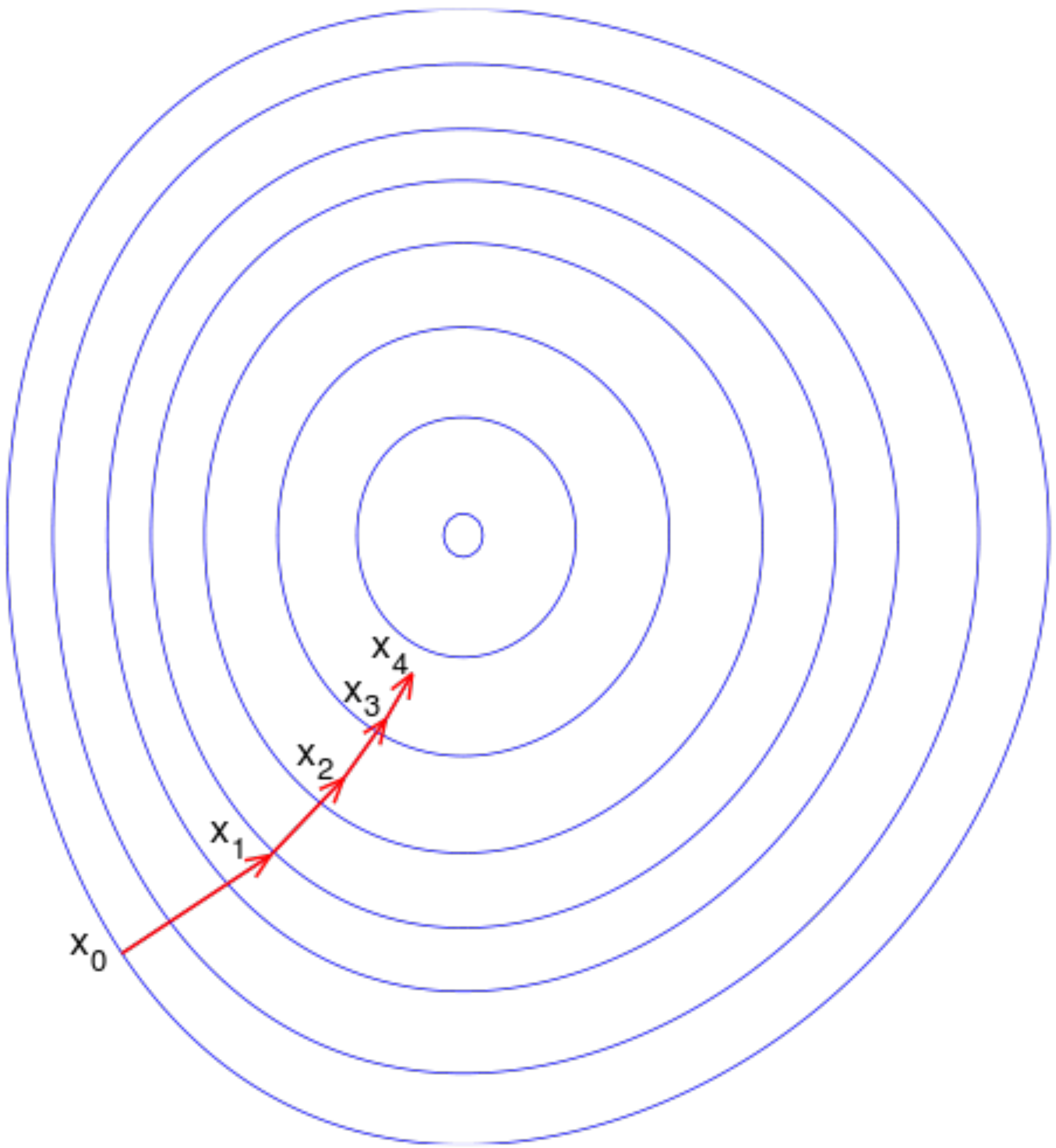
$$\beta_i \leftarrow \beta_i - \frac{\partial}{\partial \beta_i} J(\beta)$$

Repeat until Convergence….

$$\beta_i \leftarrow \beta_i - \frac{\partial}{\partial \beta_i} J(\beta)$$

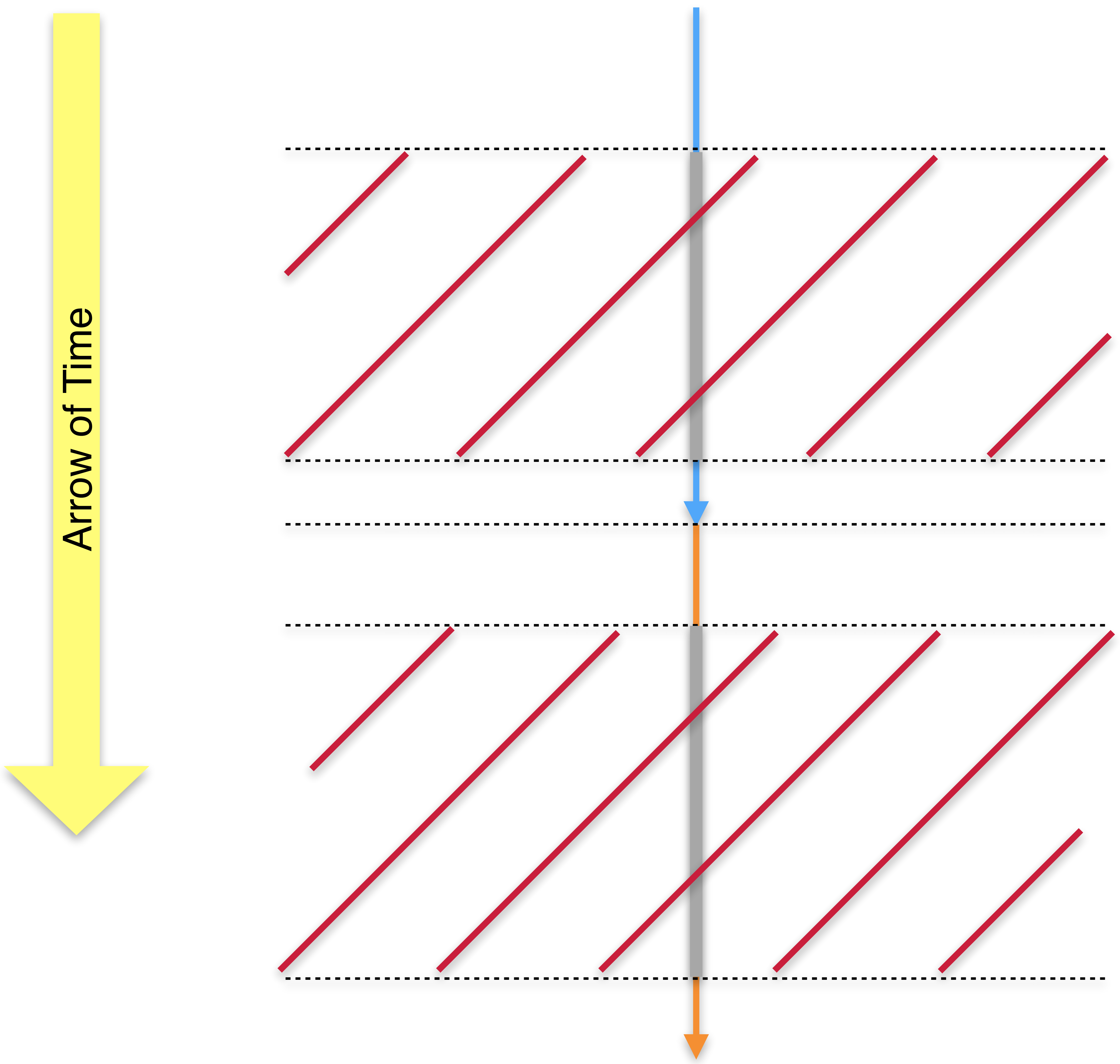
"Learning Rate"

$$\beta_i \leftarrow \beta_i - \alpha \frac{\partial}{\partial \beta_i} J(\beta)$$



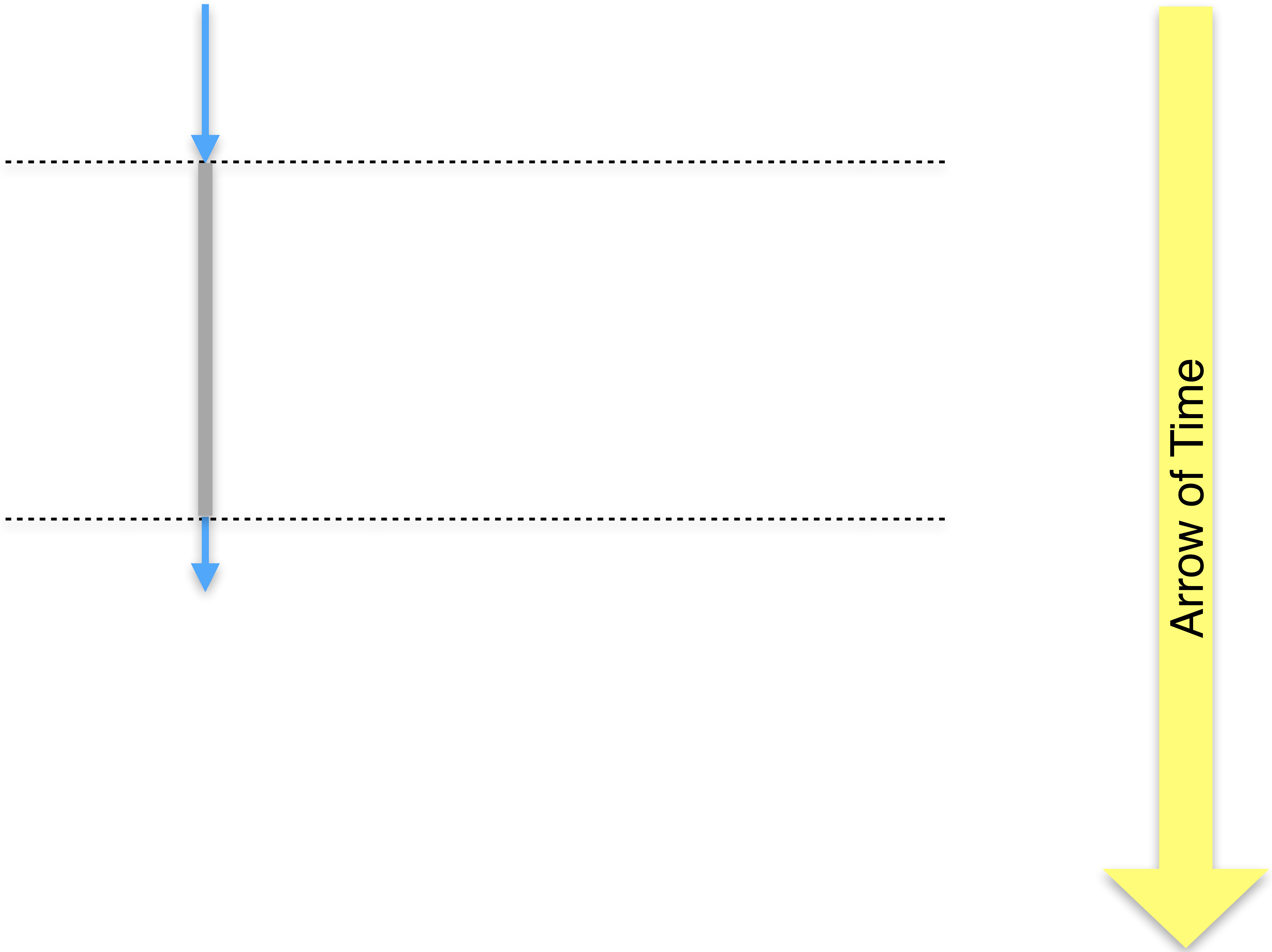
source: Wikipedia

Sequential



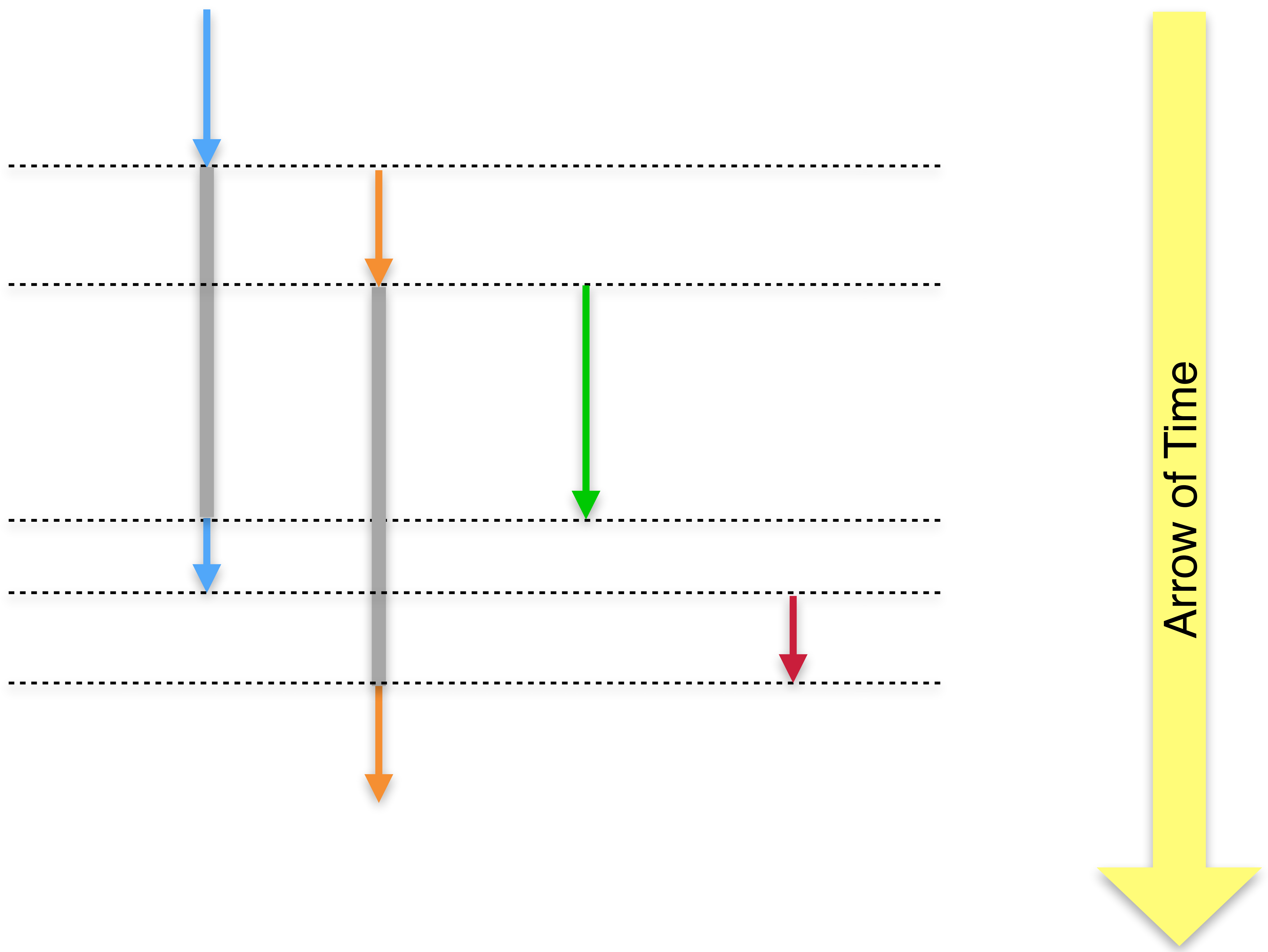
Concurrent

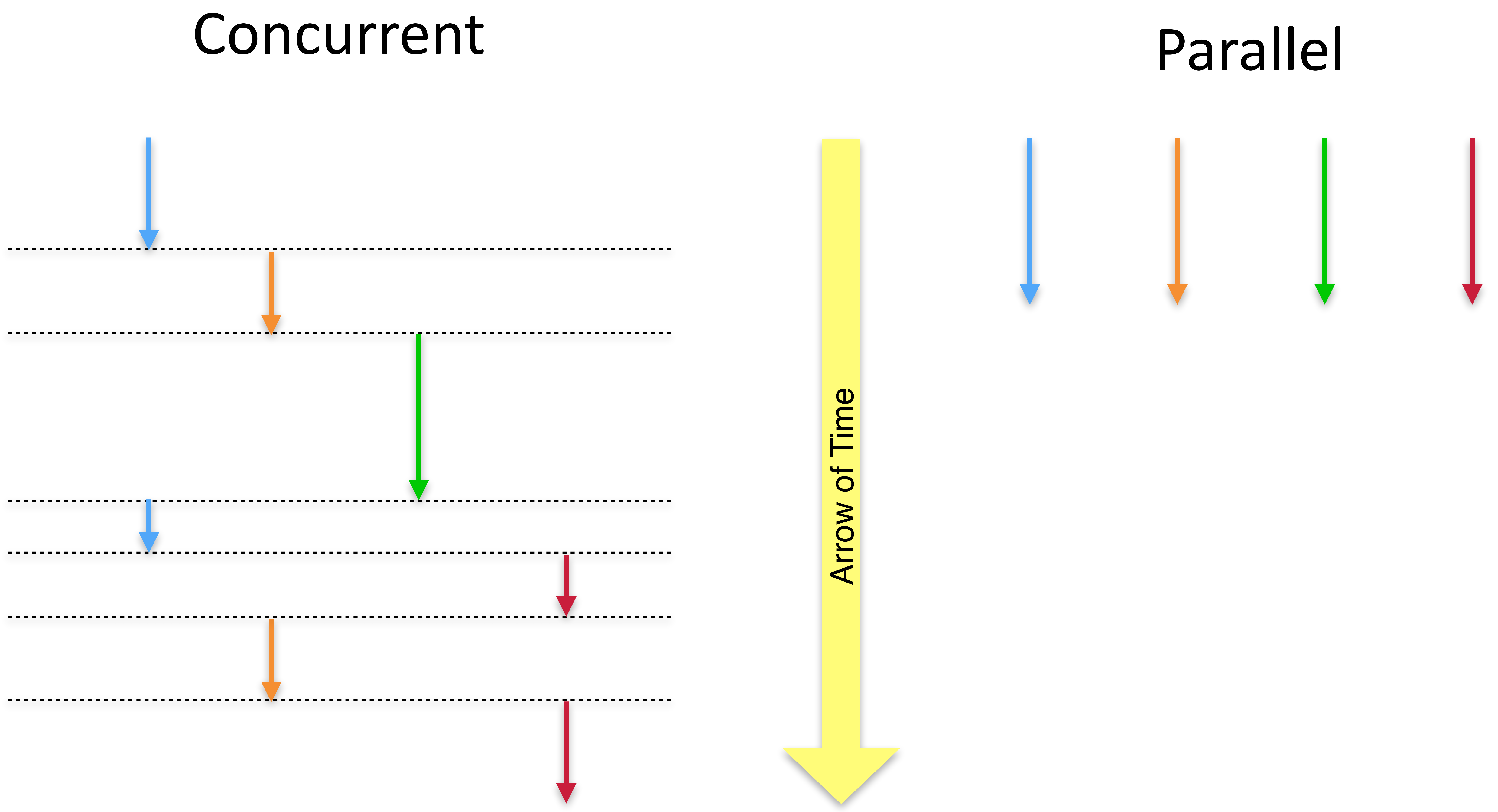
Parallel



Concurrent

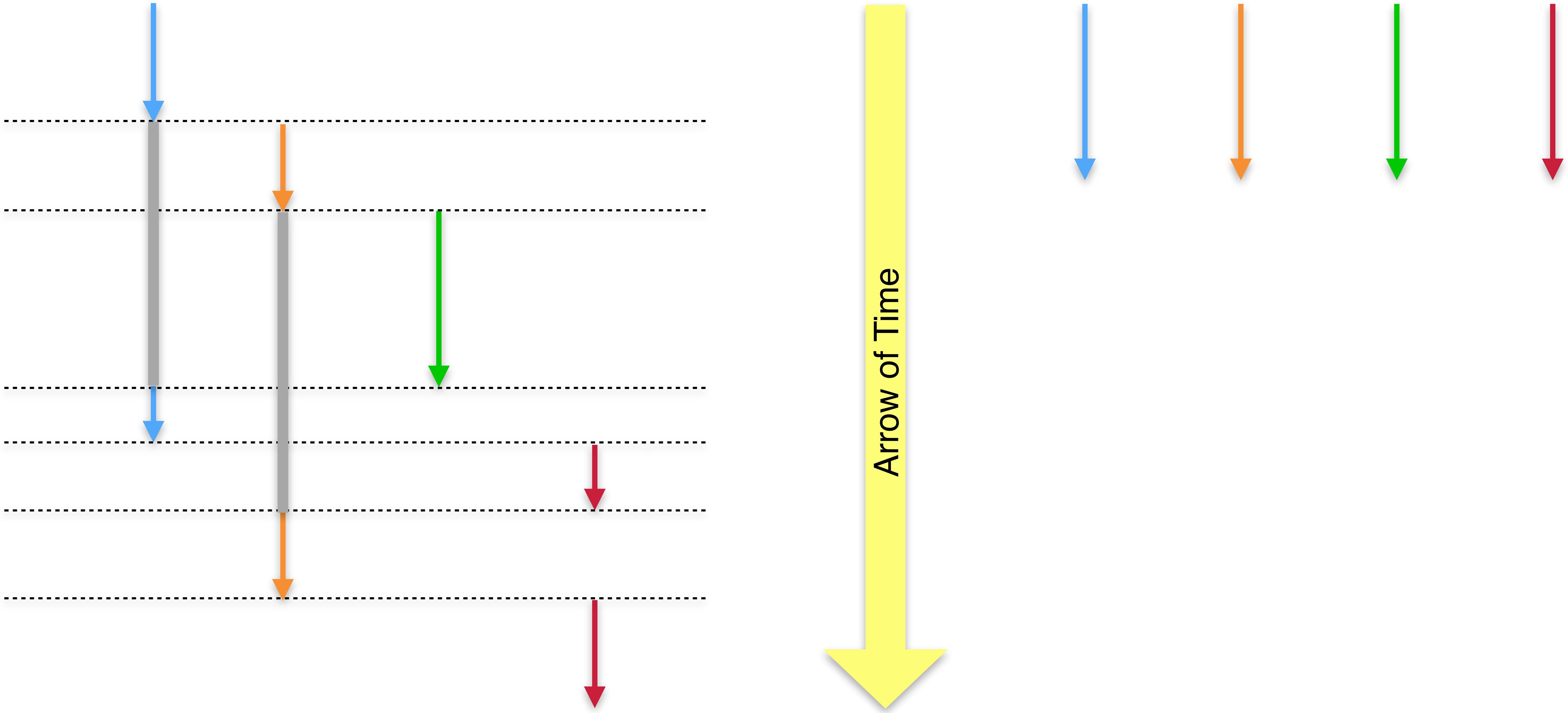
Parallel

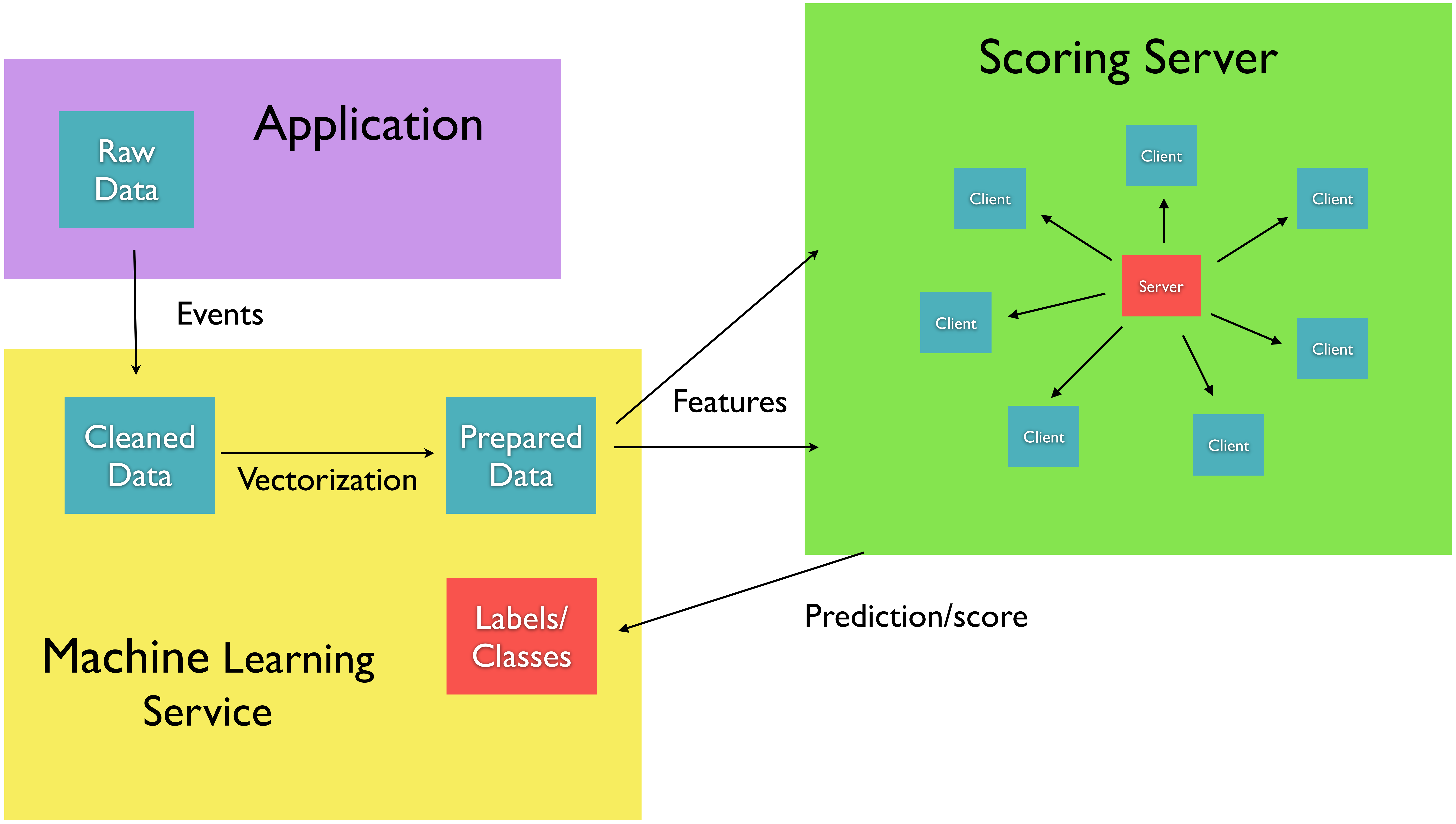



Concurrent
Parallel
Arrow of Time

Concurrent

Parallel

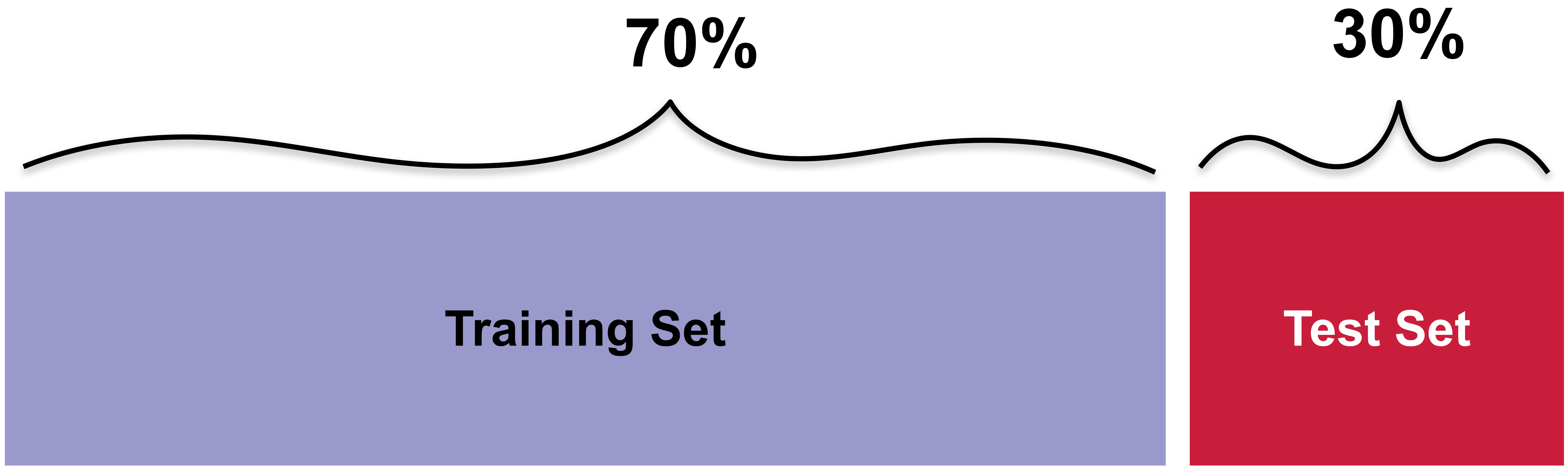



Application
Raw Data
Events
Cleaned Data
Vectorization
Prepared Data
Labels/ Classes
Machine Learning Service
Features
Prediction/score
Scoring Server
Server
Client
Client
Client
Client
Client
Client
Client

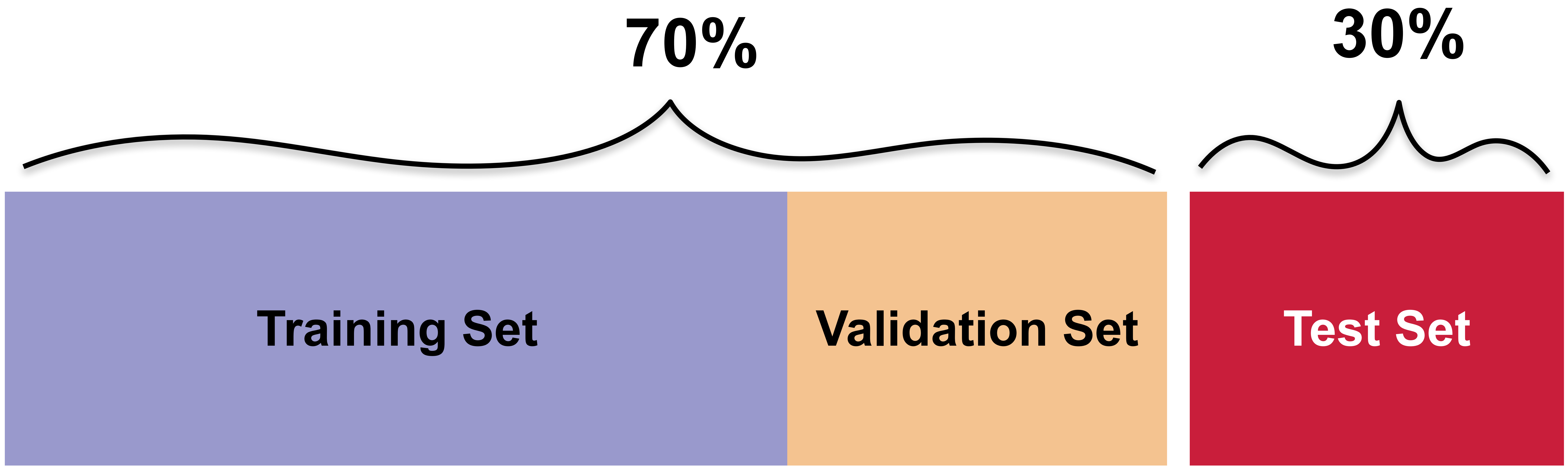
Model Diagnostics

- Overfitting (high variance)
- Underfitting (high bias)
- Training Convergence
- Label Distribution (imbalanced classes)
- Data/covariate Shift

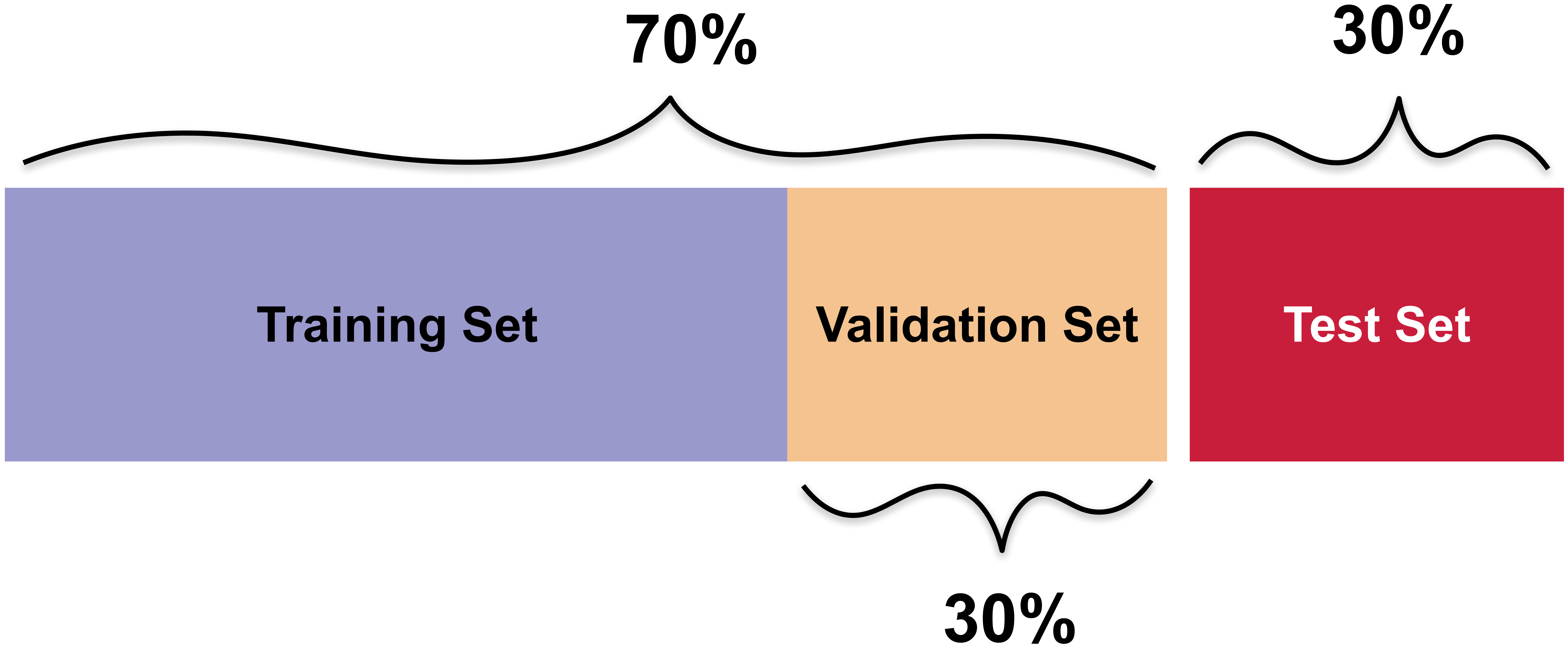
Holdout Cross Validation



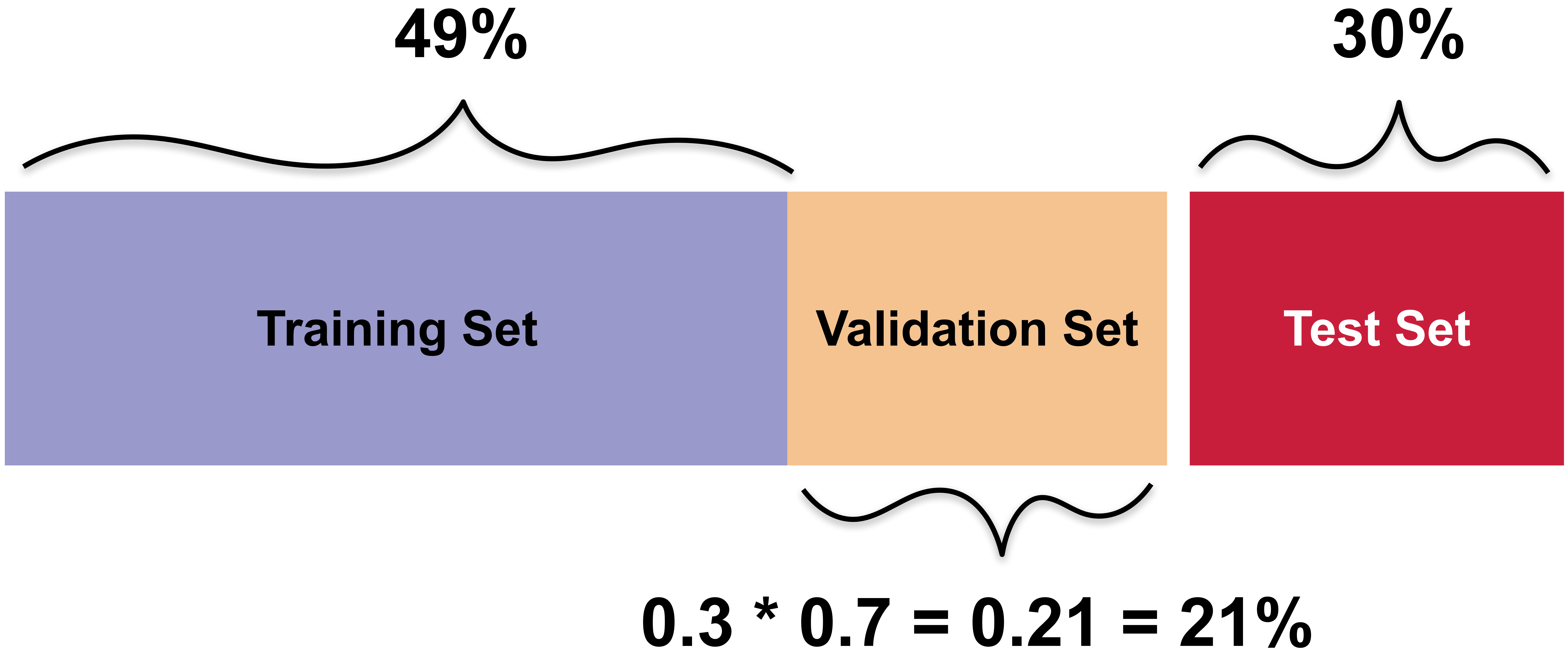
Holdout Cross Validation



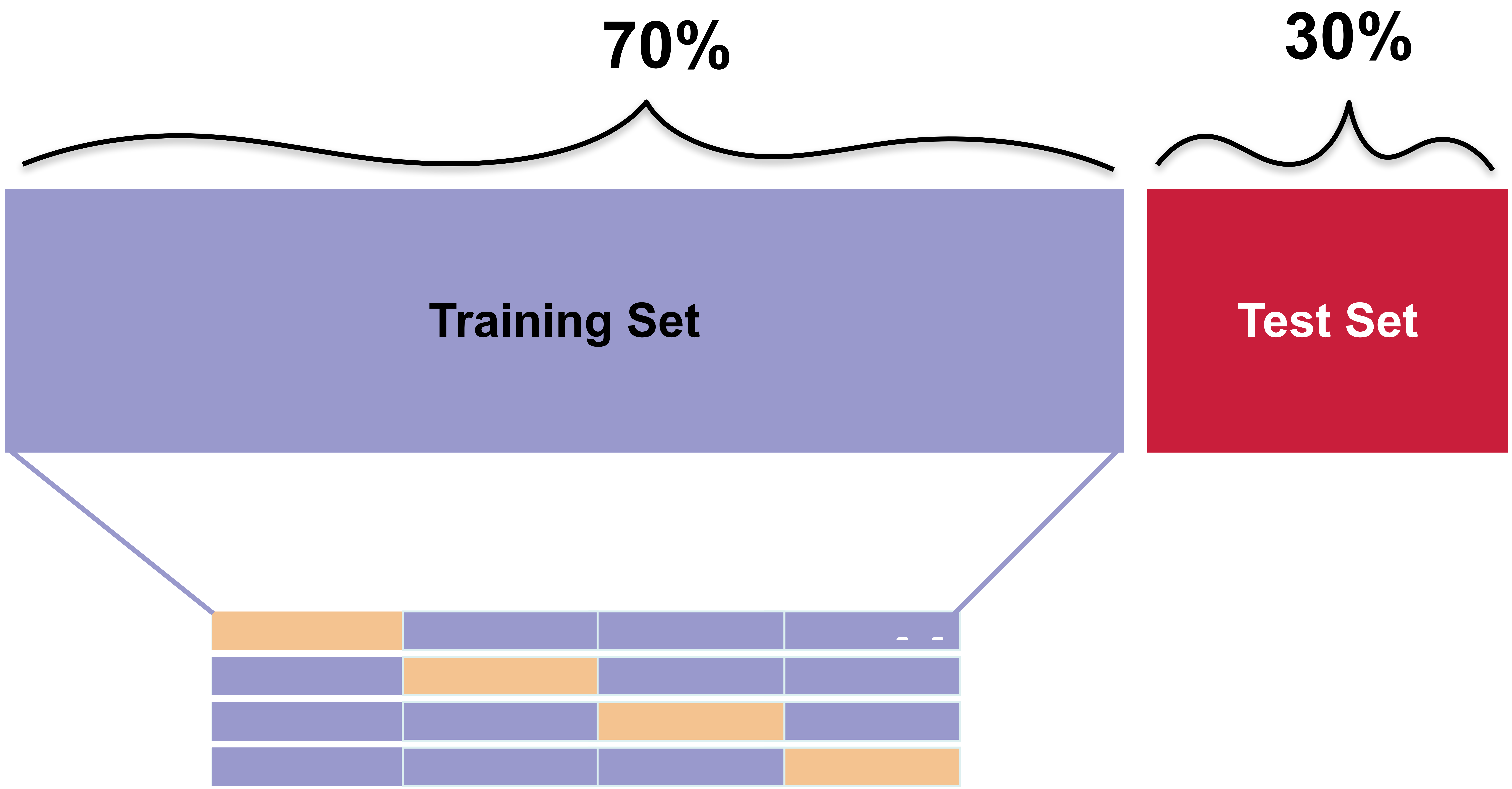
Holdout Cross Validation



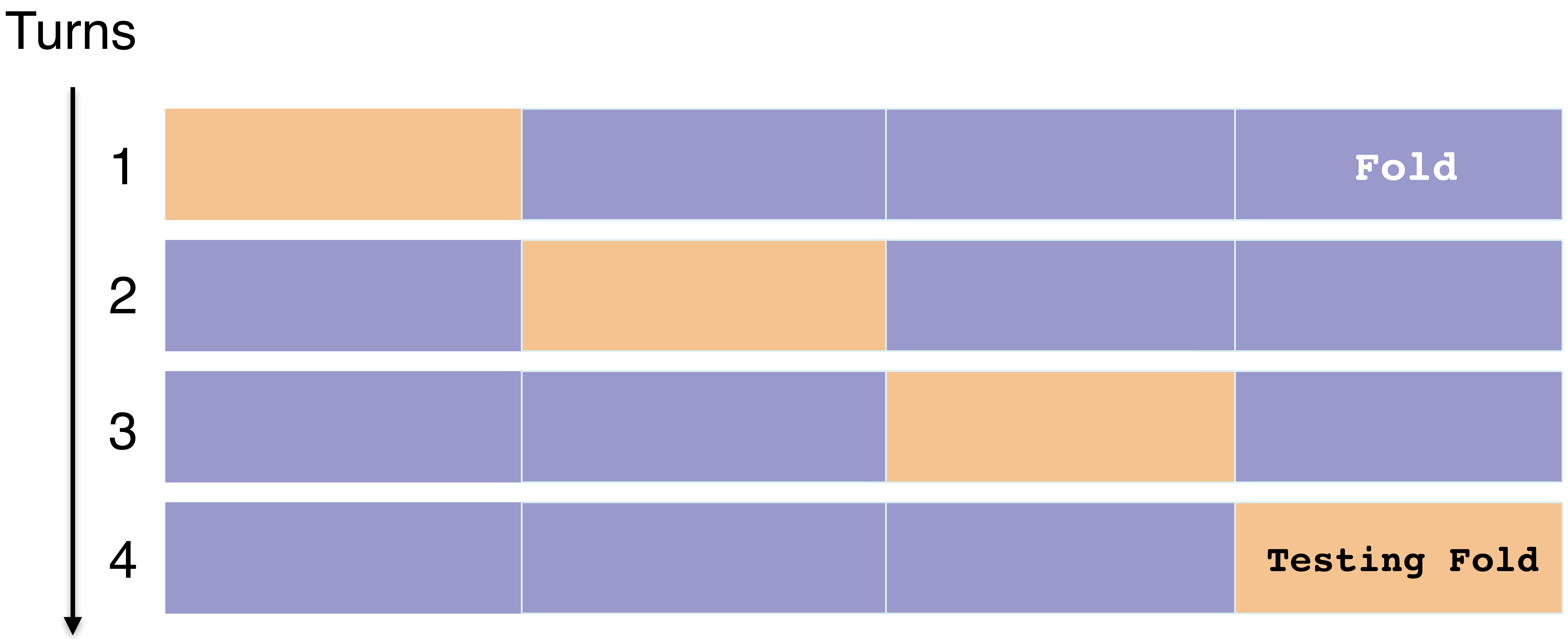
Holdout Cross Validation



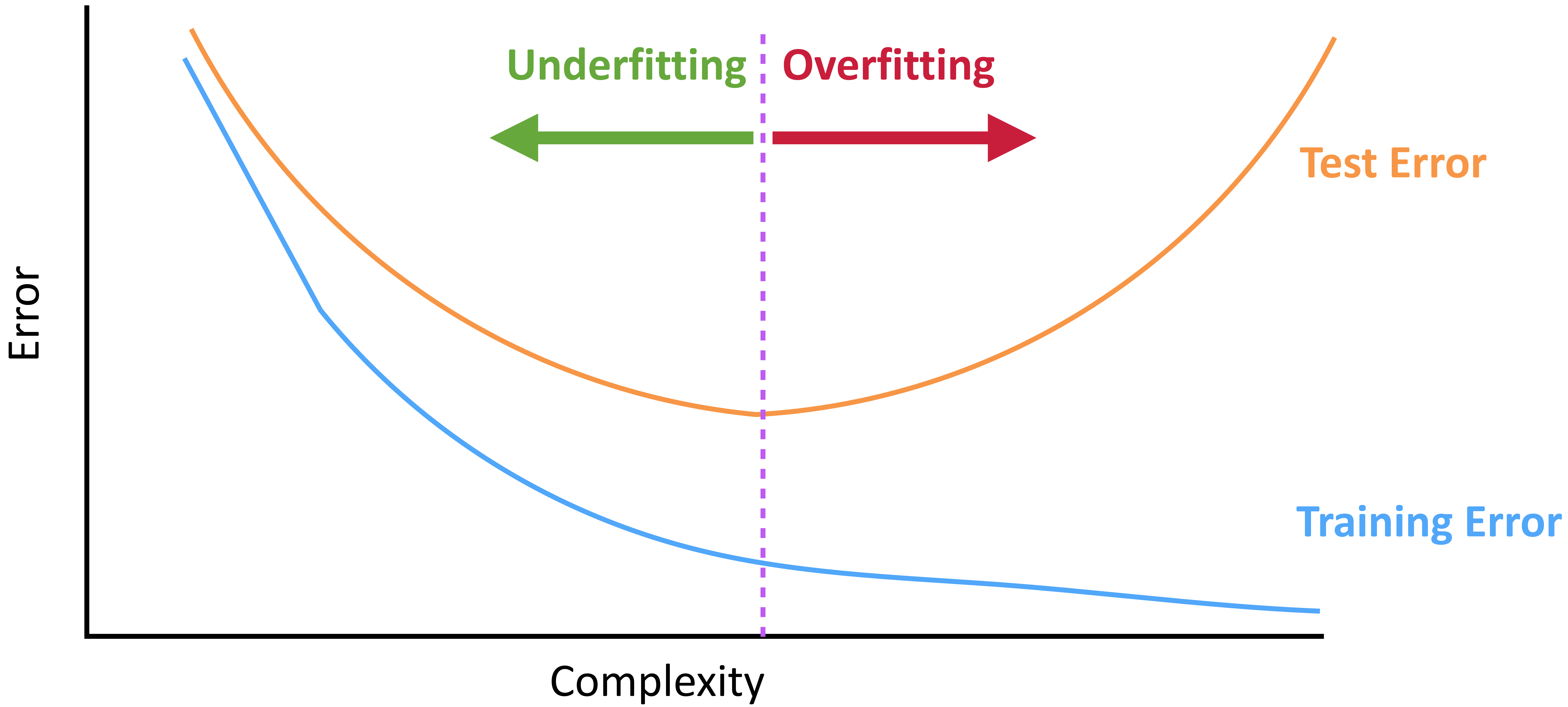
K-fold Cross Validation



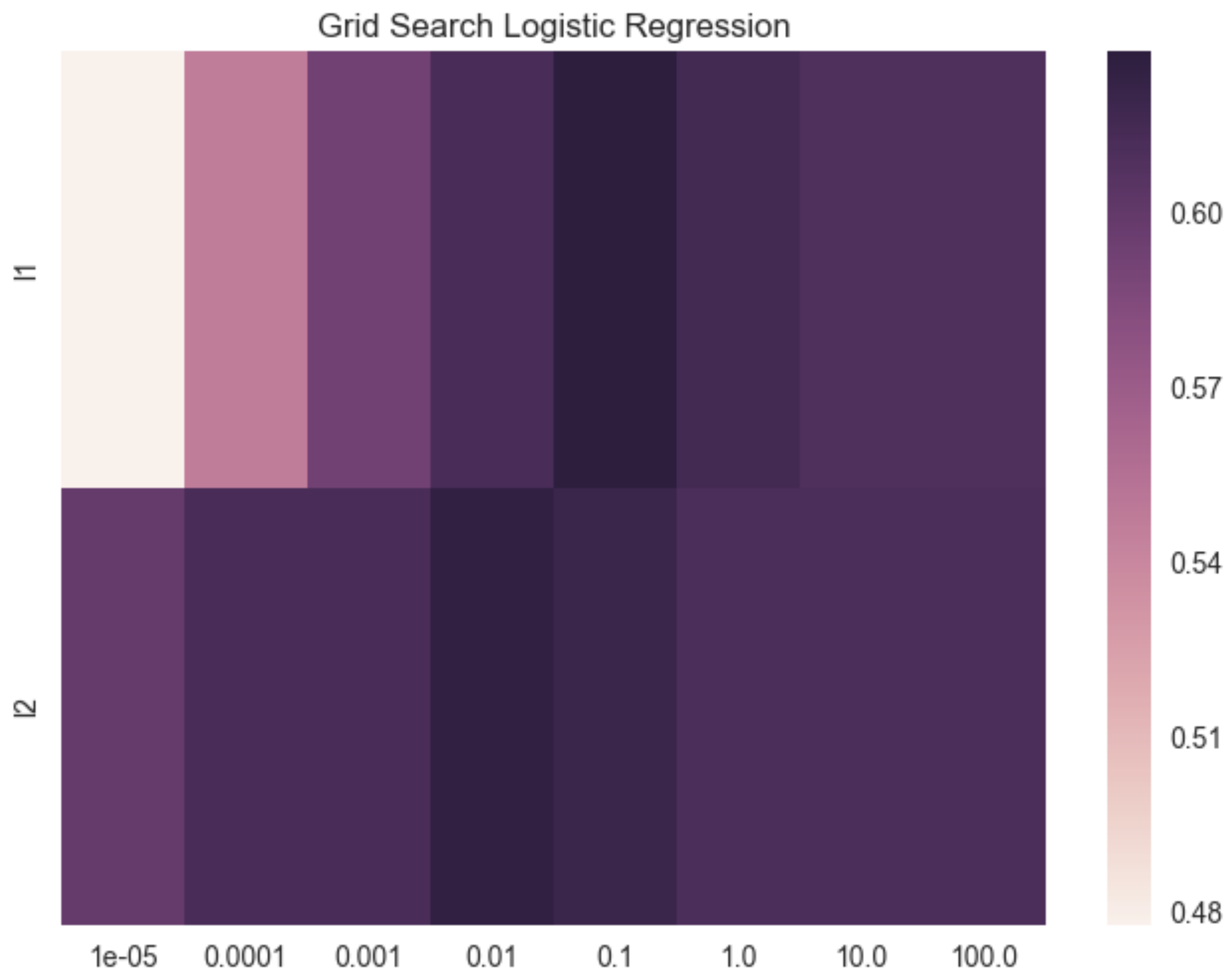
K-fold Cross Validation



Bias-variance trade-off



Grid Search



Grid Search Logistic Regression
l1
l2
1e-05
0.0001
0.001
0.01
0.1
1.0
10.0
100.0
0.60
0.57
0.54
0.51
0.48

Grid Search

- Exhaustive brute force search
- Find optimal hyperparameters or models
- Computationally costly
- But embarrassingly parallel!

Live Code

Short Comings of Accuracy

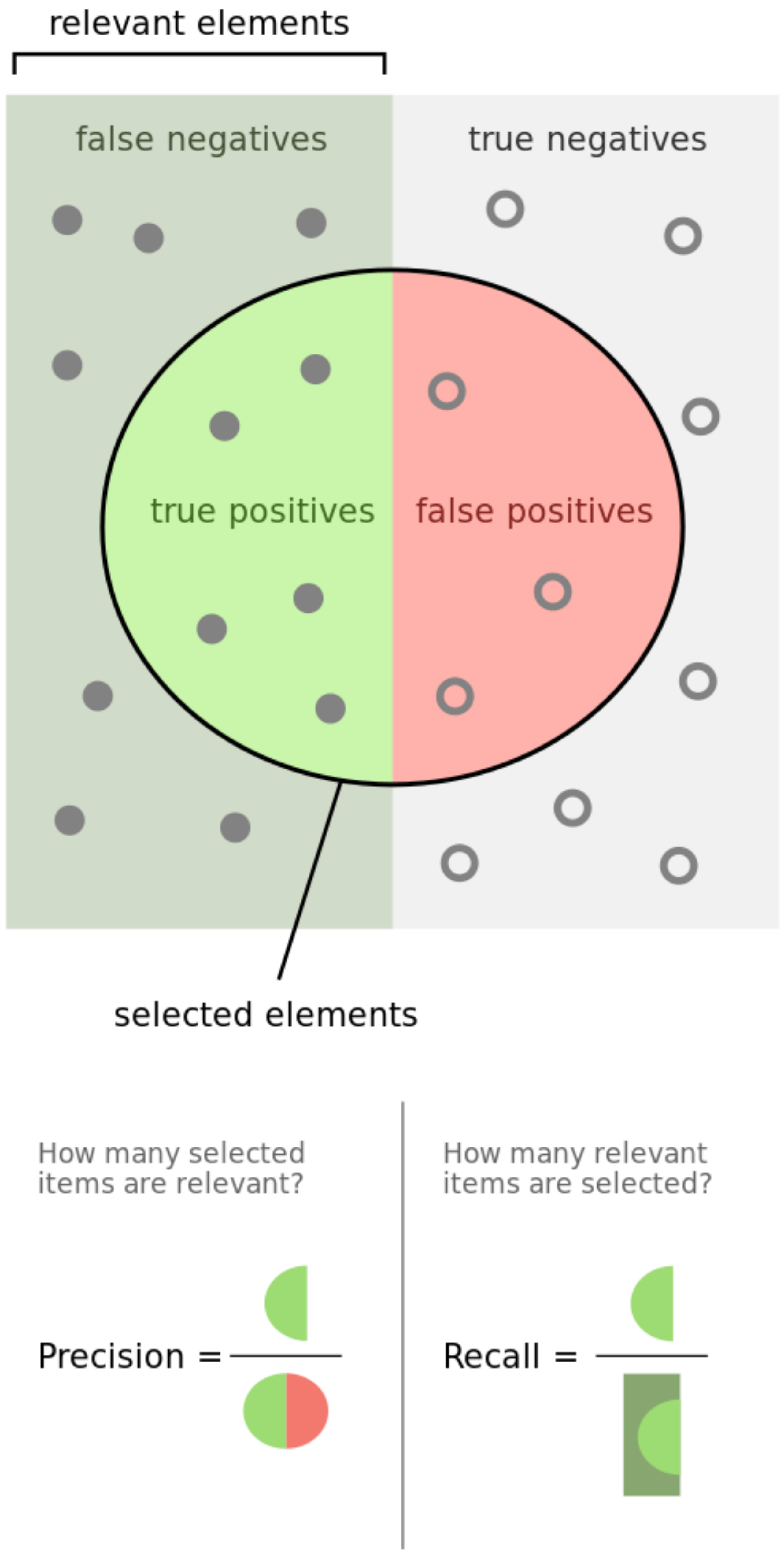
- Sensitive to imbalanced classes
- Misclassifications carry equal weight

$$\text{Accuracy} = \frac{\#\ \text{correct}}{\text{total examples}}$$

Evaluation Metrics

- Accuracy
- Precision/Recall
- F1
- AUC (area under the curve)

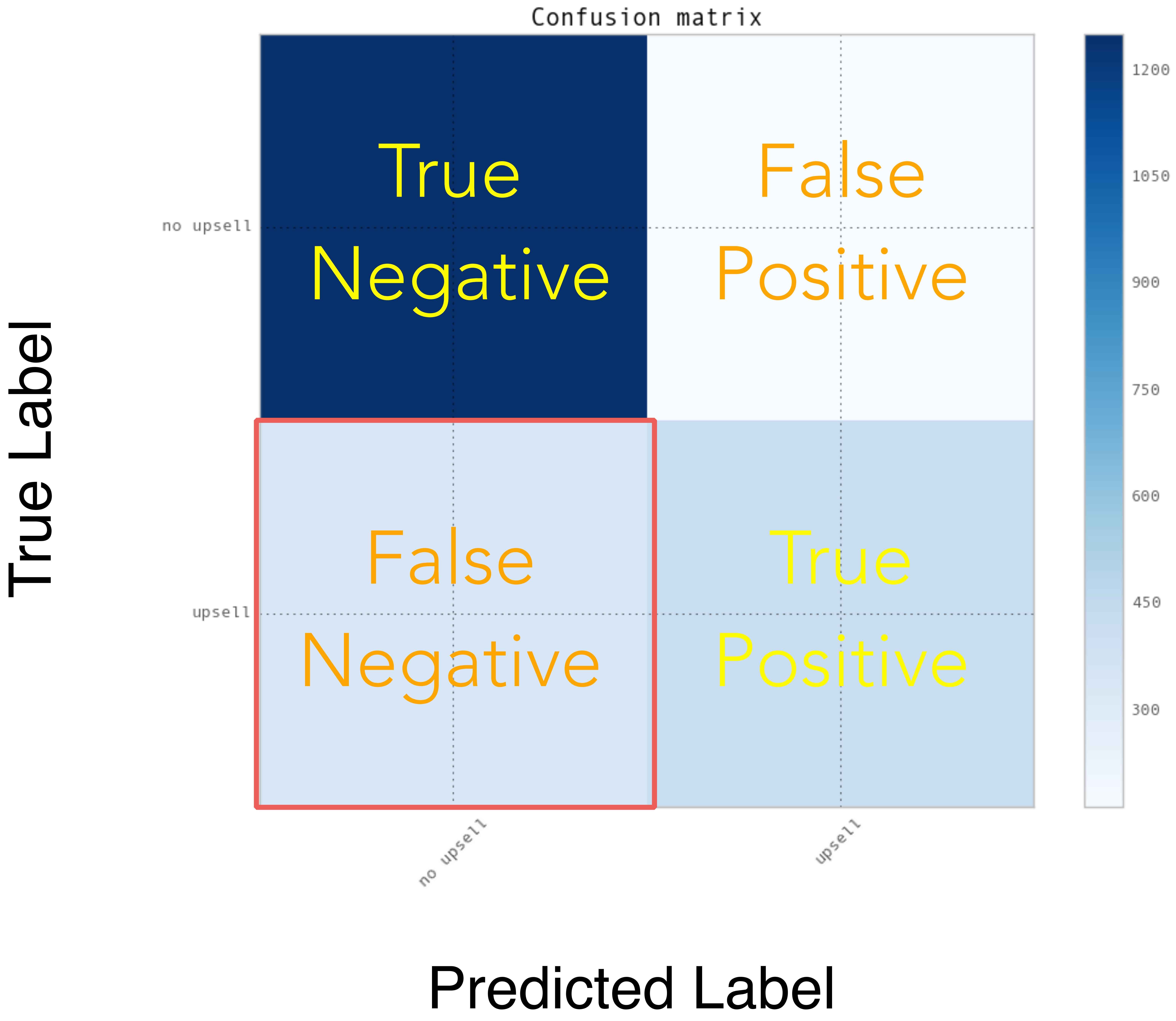
Precision and Recall



$$\text{Precision} = \frac{TP}{TP + FP}$$

$$\text{Recall} = \frac{TP}{FN + TP}$$

$$F1 = \frac{2}{\frac{1}{\text{precision}} + \frac{1}{\text{recall}}}$$

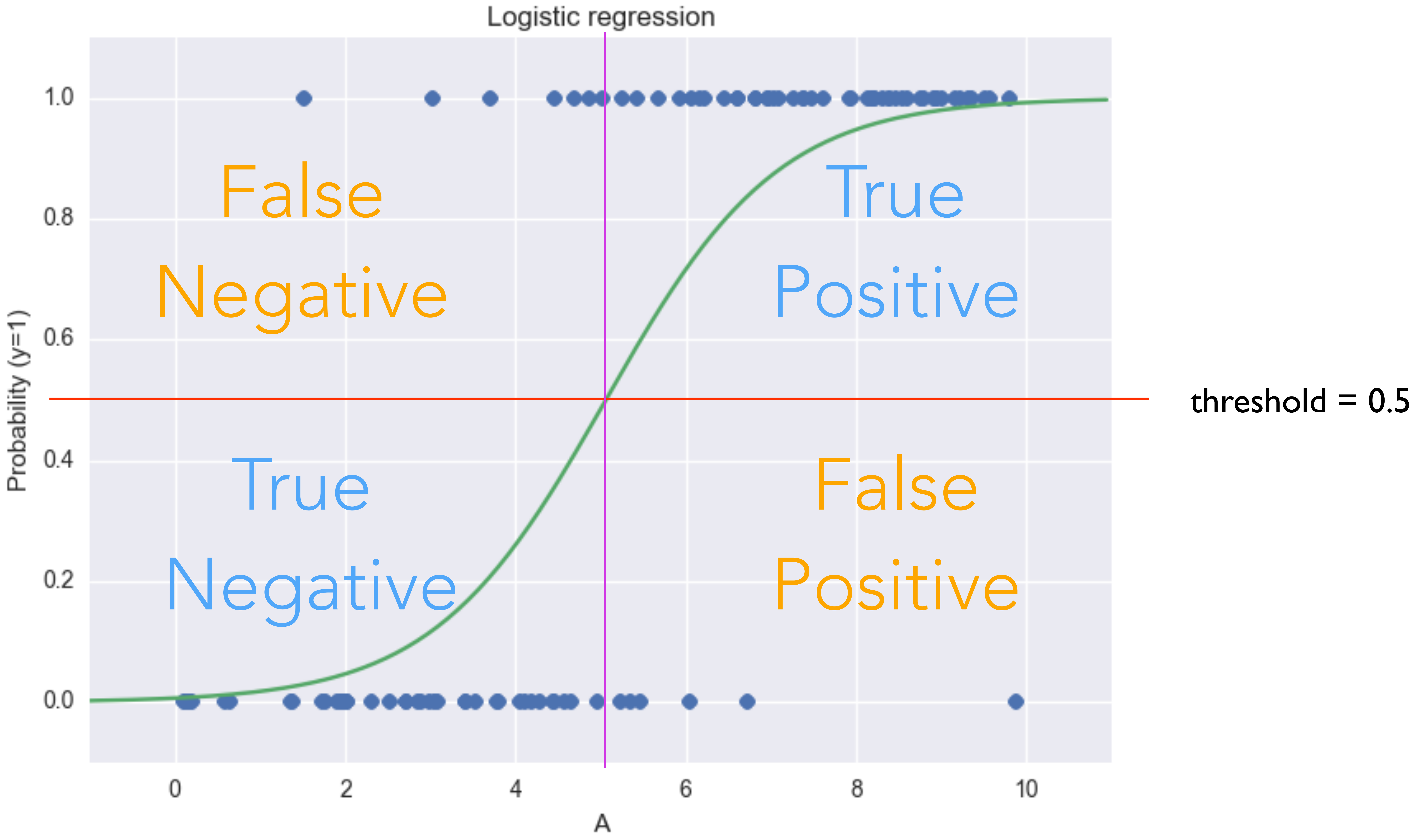


$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

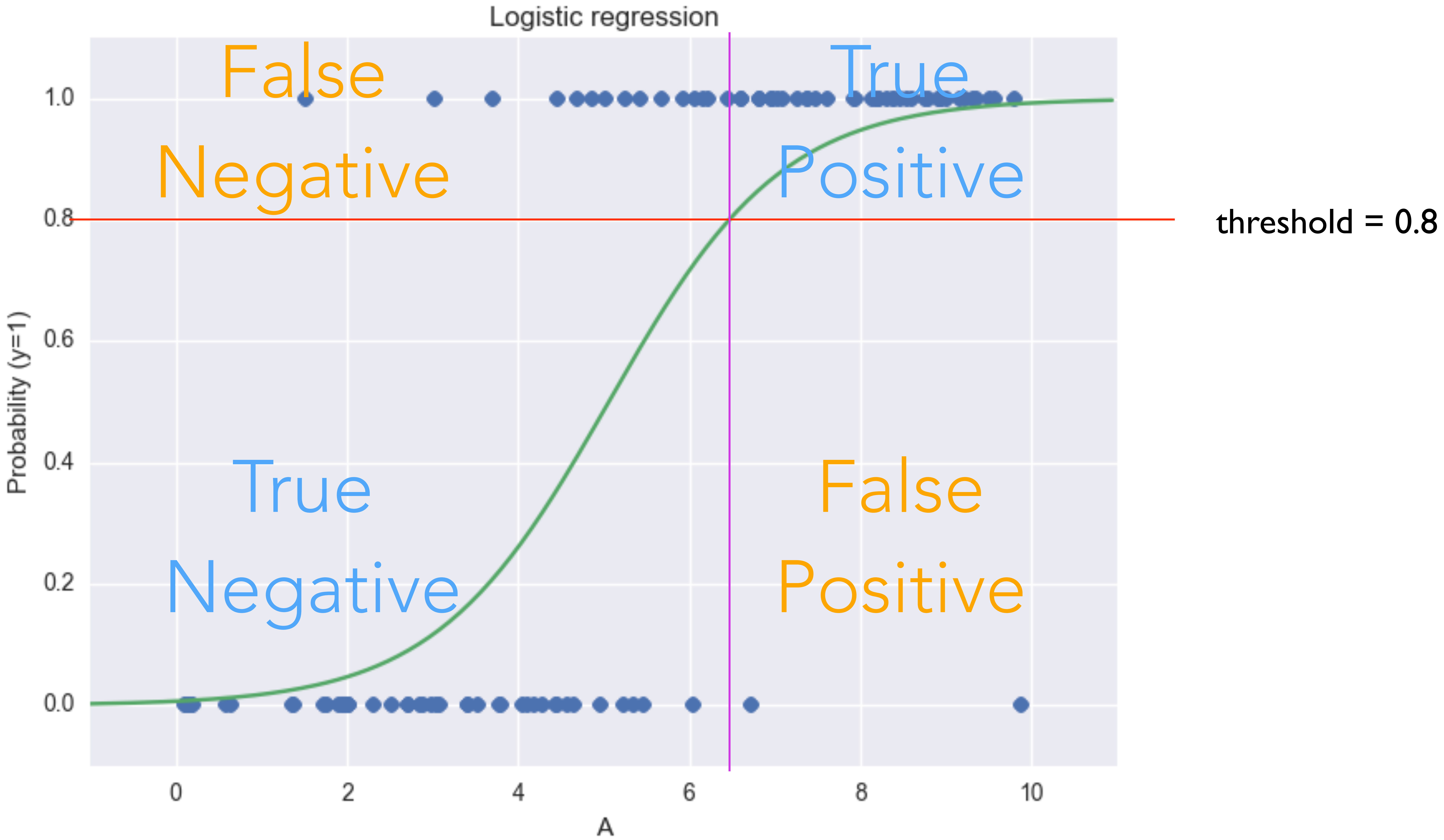
$$\text{True positive rate} = \frac{TP}{TP + FN}$$

$$\text{False positive rate} = \frac{FP}{FP + TN}$$

Logistic Regression



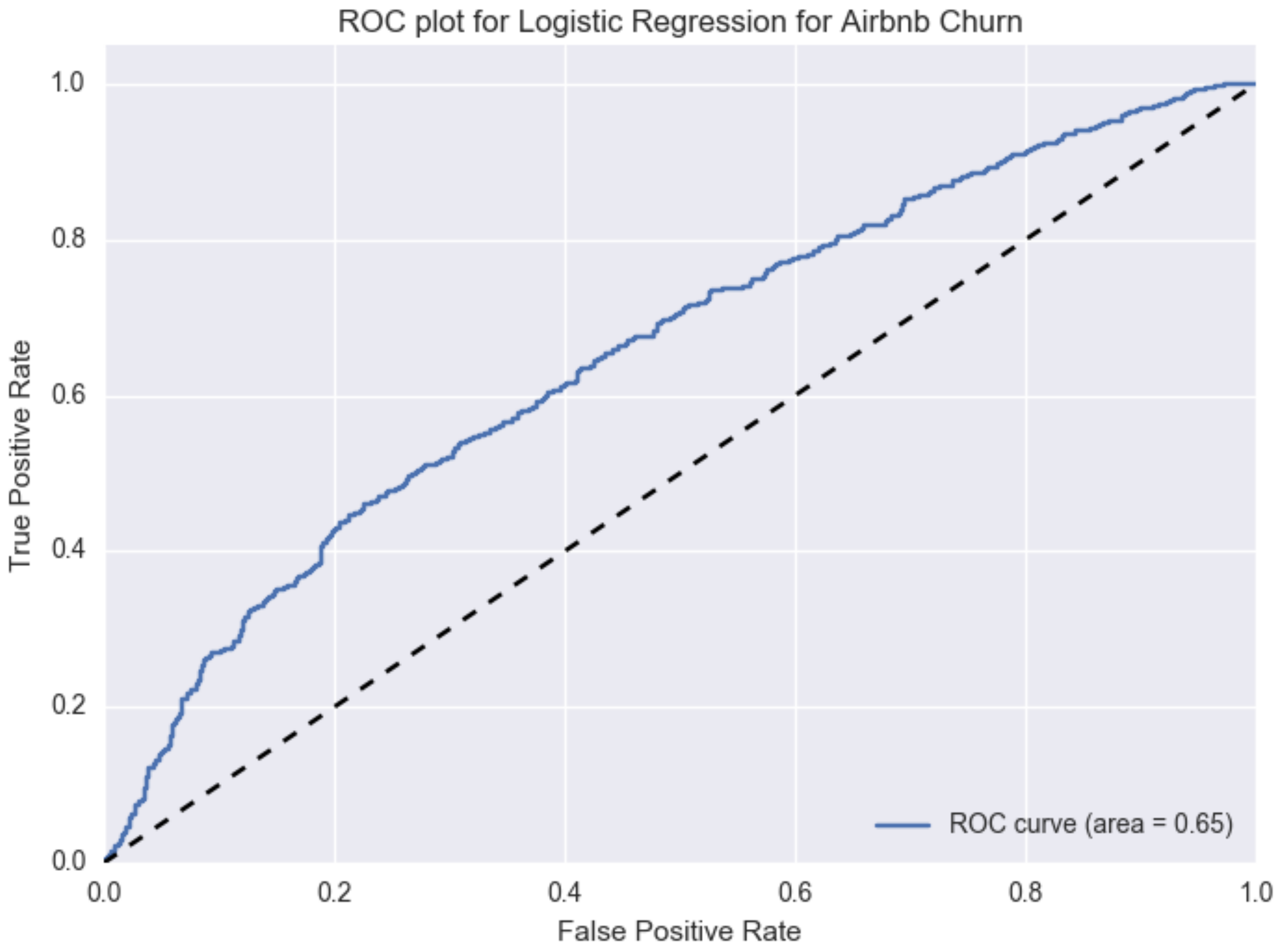
Logistic Regression



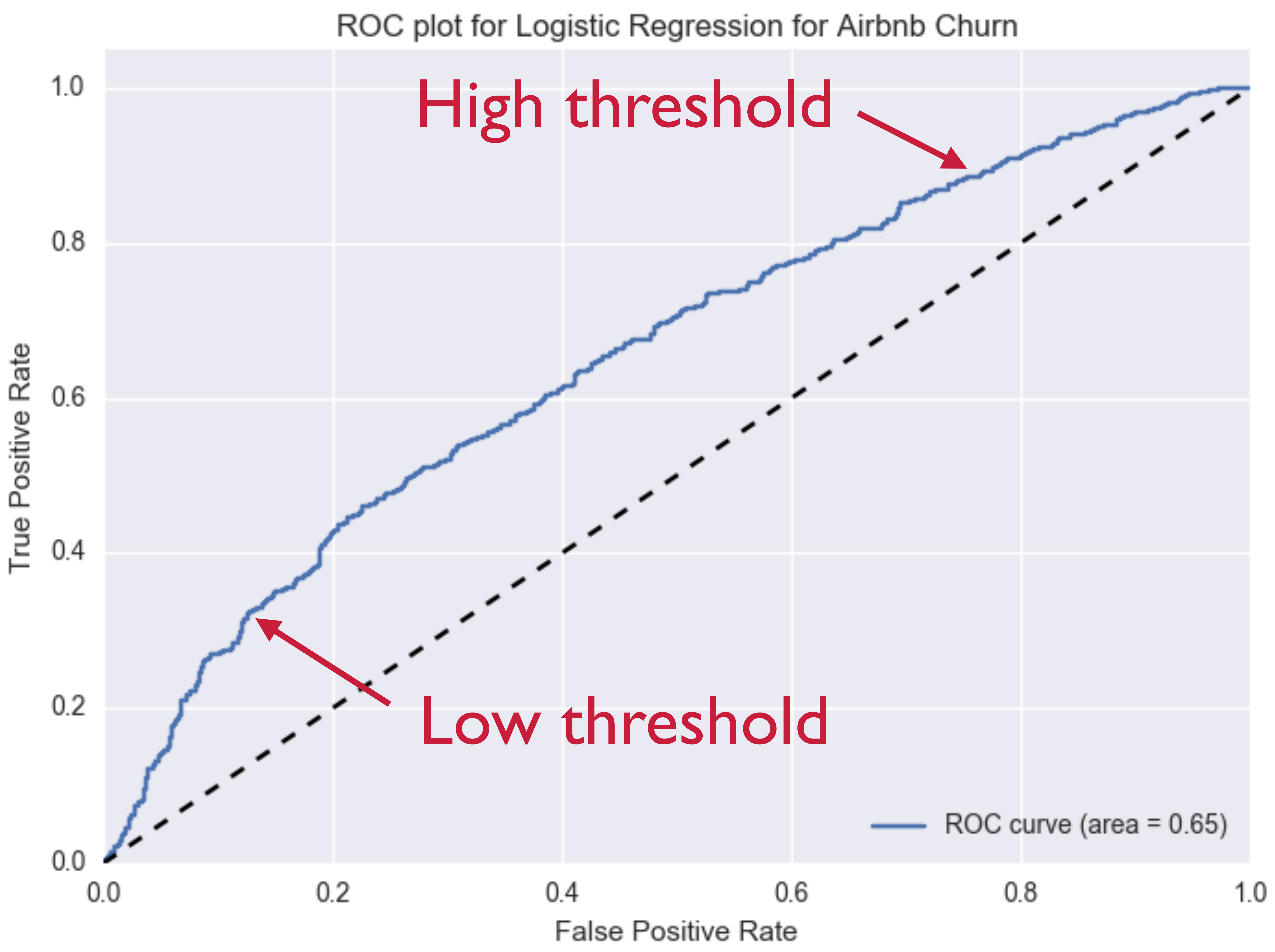
Confusion Matrix vs. Point Metrics

- Confusion matrix has fine-grained information about misclassifications
- Precision/Recall/F1 can be used in automated comparison (grid search)

ROC plots


ROC plot for Logistic Regression for Airbnb Churn
1.0
0.8
0.6
0.4
0.2
0.0
True Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0
False Positive Rate
ROC curve (area = 0.65)

ROC plots



ROC plot for Logistic Regression for Airbnb Churn
High threshold
Low threshold
ROC curve (area = 0.65)
True Positive Rate
False Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0

Live Code

Strategies

Error decomposition

- Input **ground truth** for each component/stage of pipeline.

Ablation analysis

- Remove components/parameters **one at a time.**

See also: http://cs229.stanford.edu/materials/ML-advice.pdf

Improving a Model

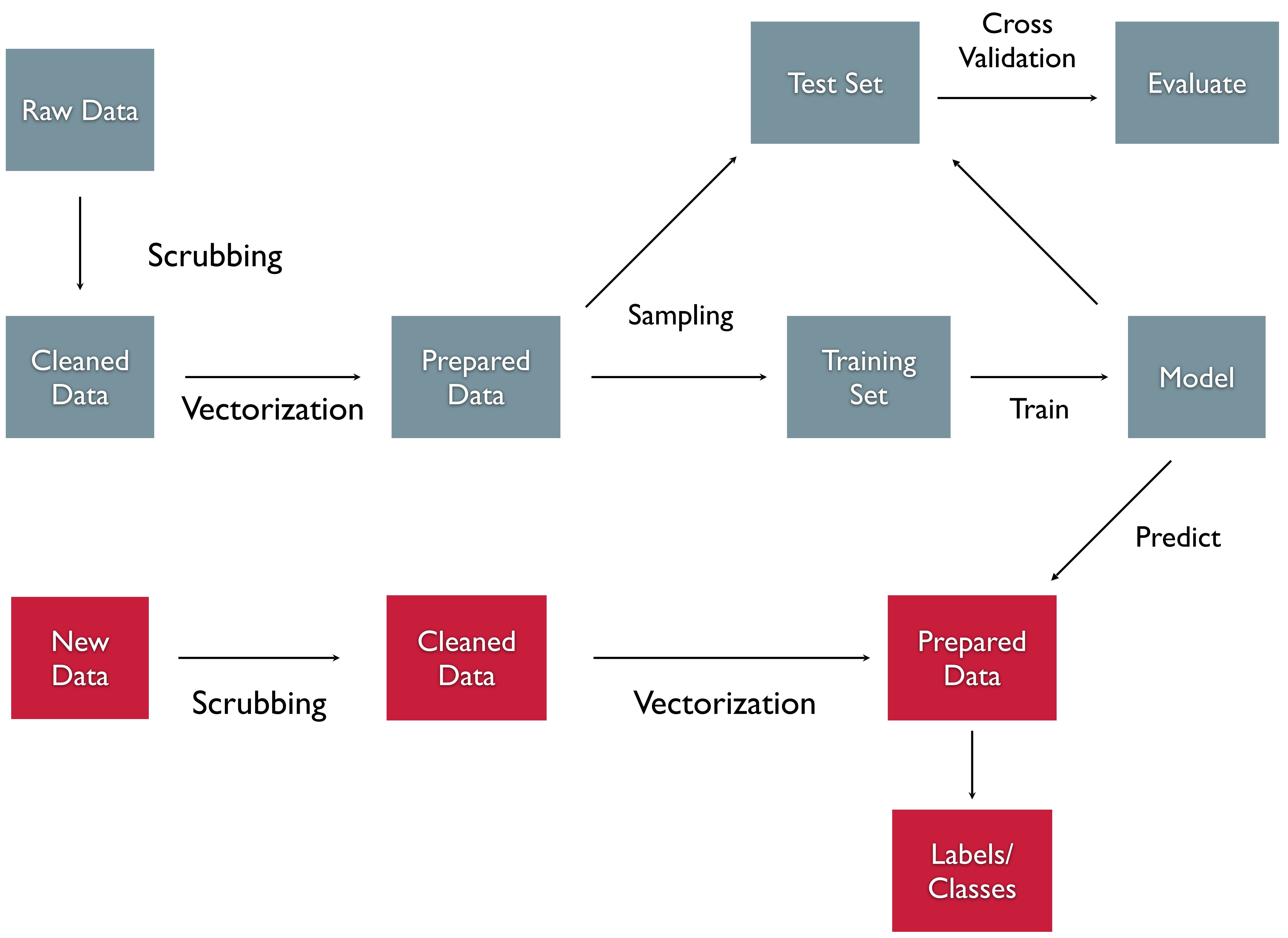
- More data
- Better data (and labels)
- Feature Selection and Engineering
- Regularization
- Model Selection (more/less complex model)

Model Selection

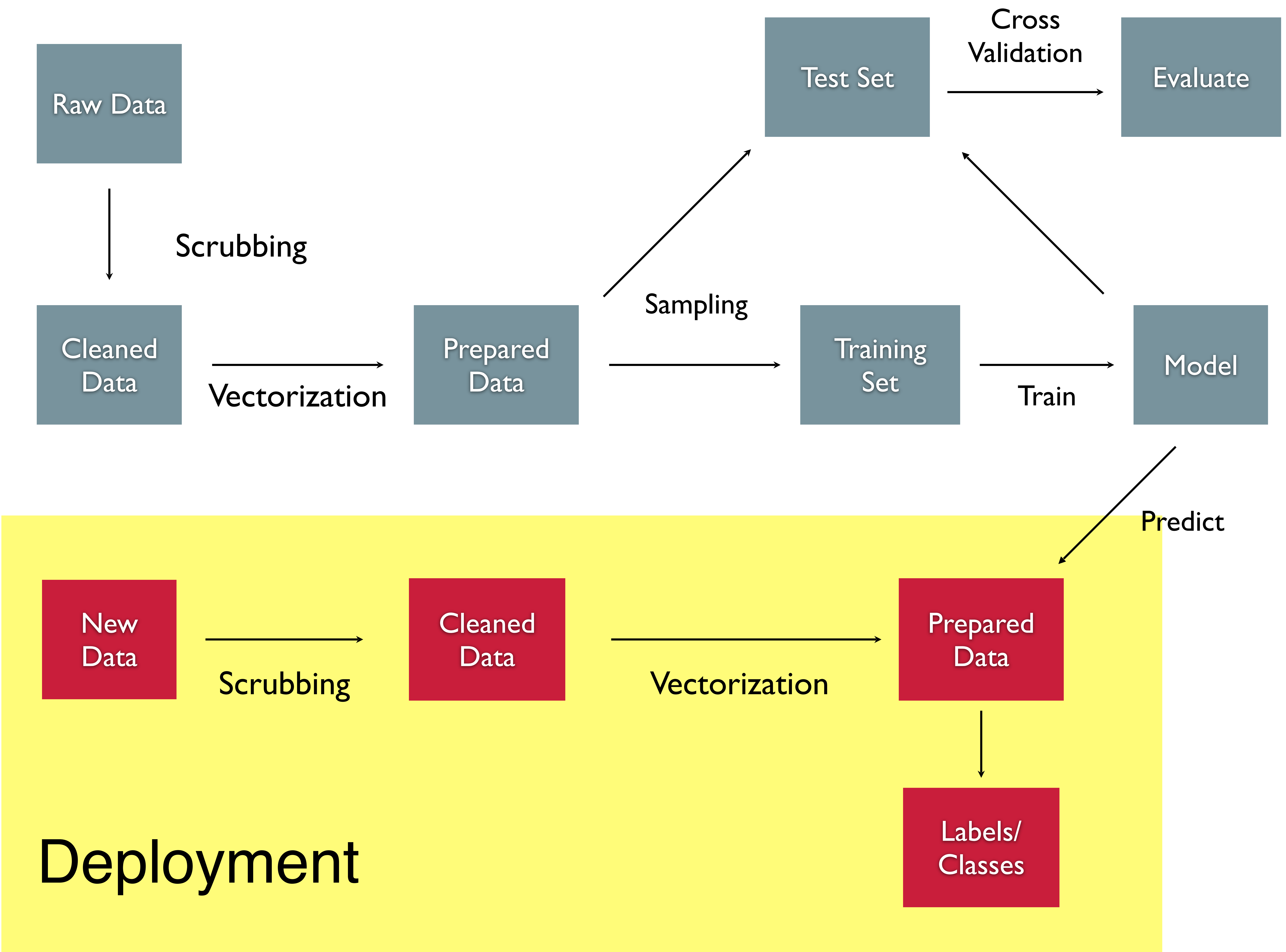
- Performance (optimizing for metric of interest)
- Training time vs testing time
- Online vs. batch
- Interpretability
- Multiclass vs. single class
- High dimensionality
- Nonlinear vs. linear

Machine Learning in the Wild

Overview


Raw Data
Scrubbing
Cleaned Data
Vectorization
Prepared Data
Sampling
Test Set
Cross Validation
Evaluate
Training Set
Train
Model
Predict
New Data
Scrubbing
Cleaned Data
Vectorization
Prepared Data
Labels/ Classes

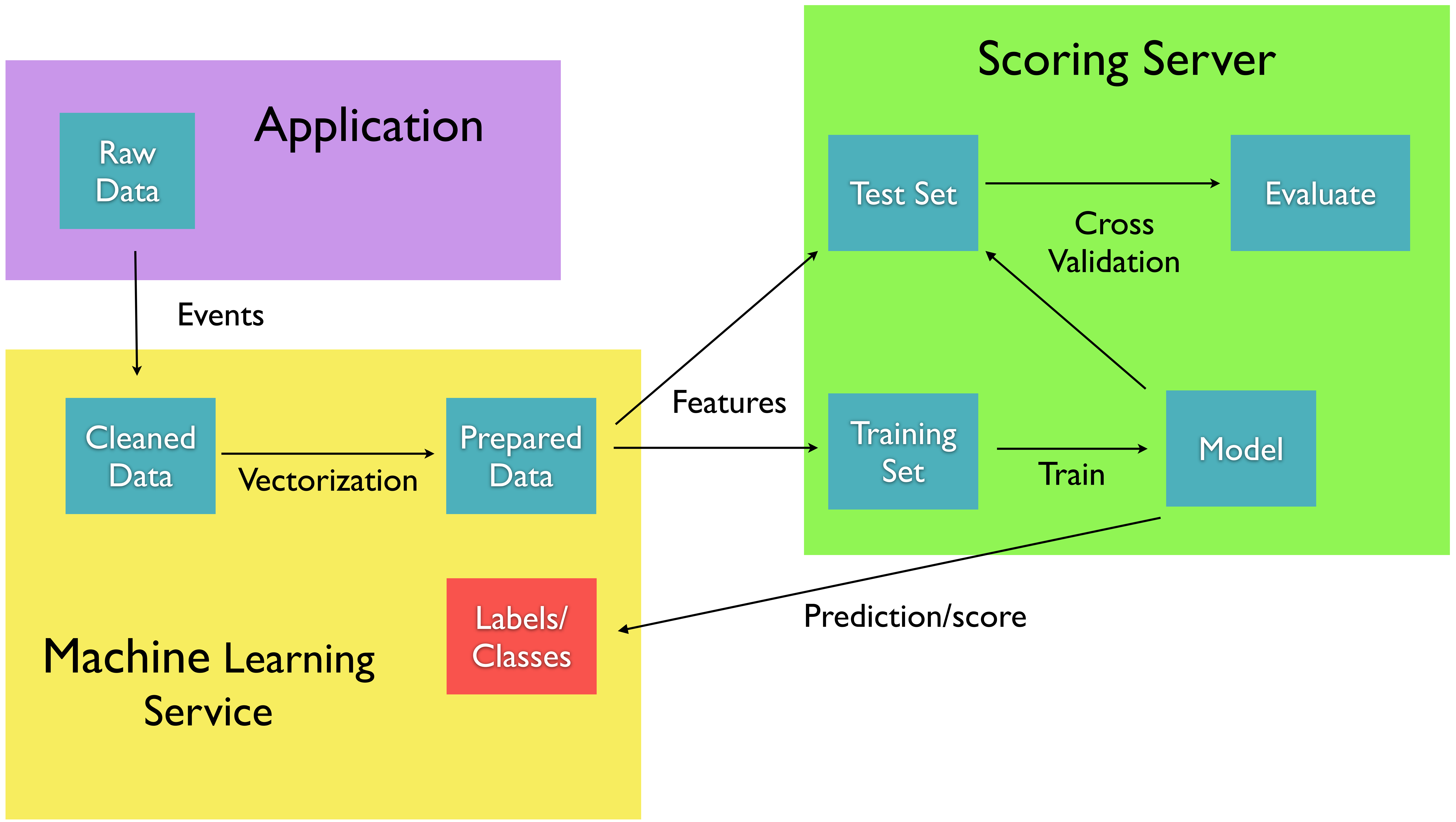
Overview


Raw Data
Scrubbing
Cleaned Data
Vectorization
Prepared Data
Sampling
Test Set
Cross Validation
Evaluate
Training Set
Train
Model
Predict
New Data
Scrubbing
Cleaned Data
Vectorization
Prepared Data
Labels/ Classes
Deployment

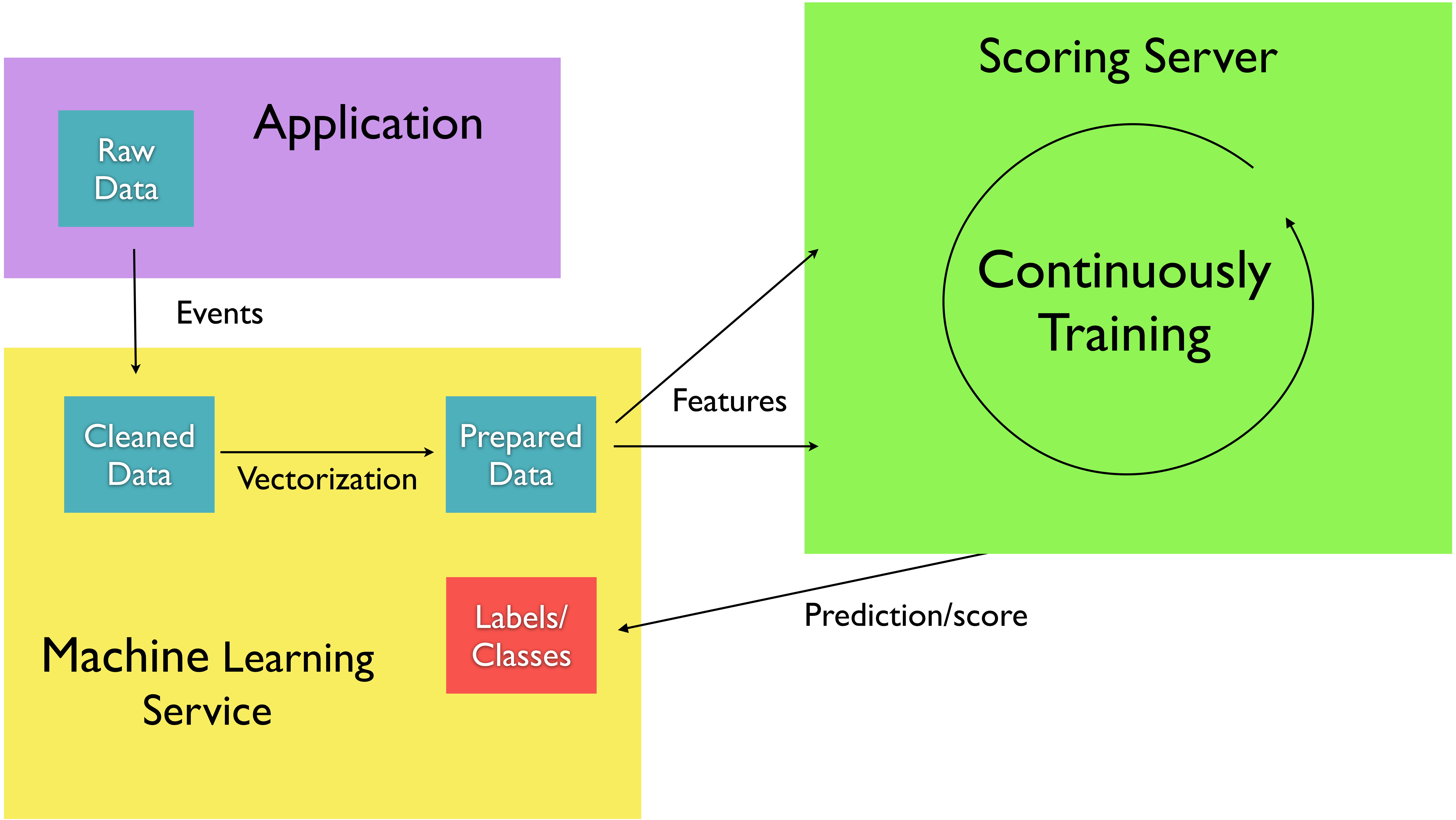
Online Evaluation Strategies

- Continuous Batch Offline Evaluation
- “Live” A/B Testing
- Multi-armed Bandit

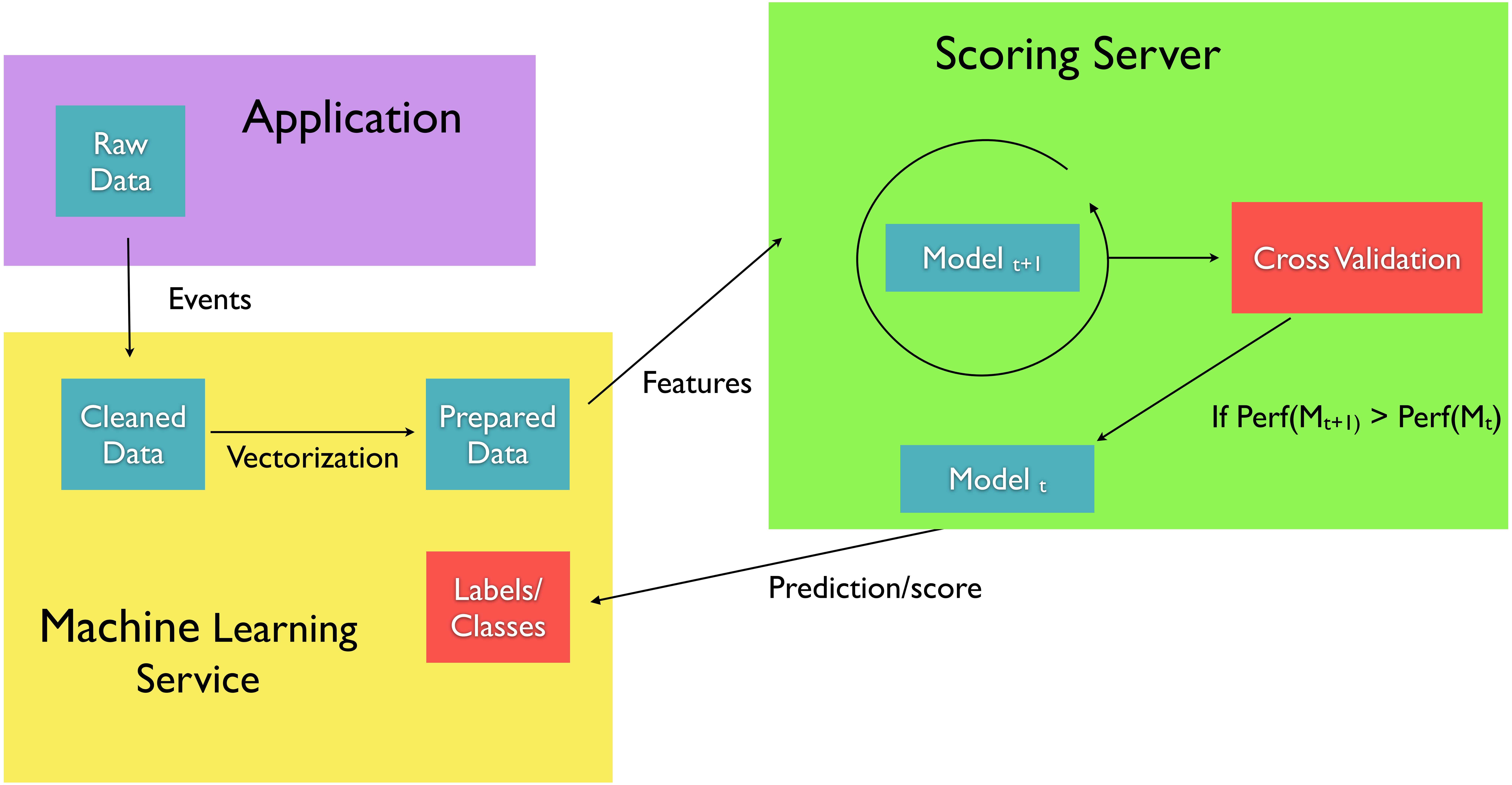
Deploying Models



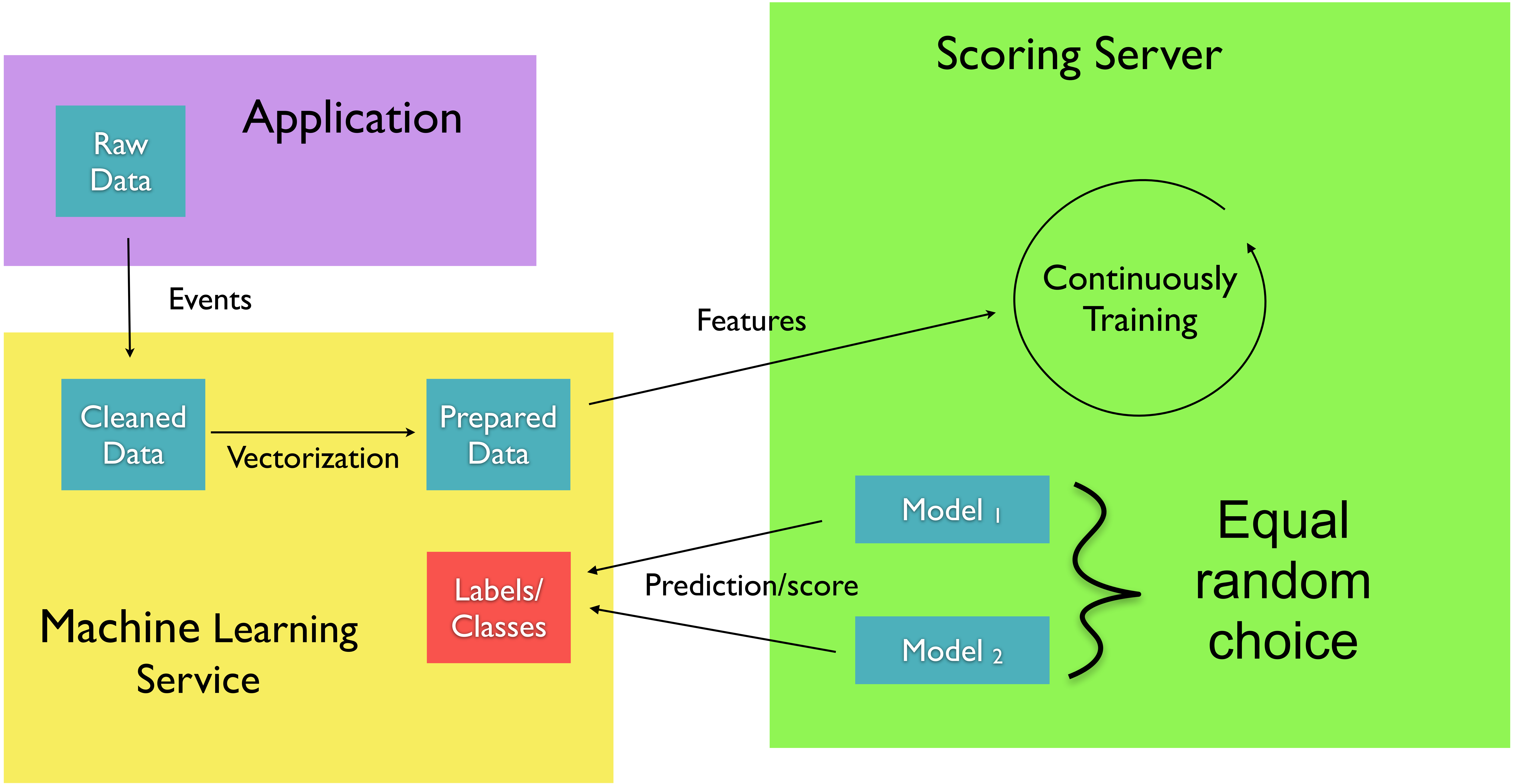
Deploying Models



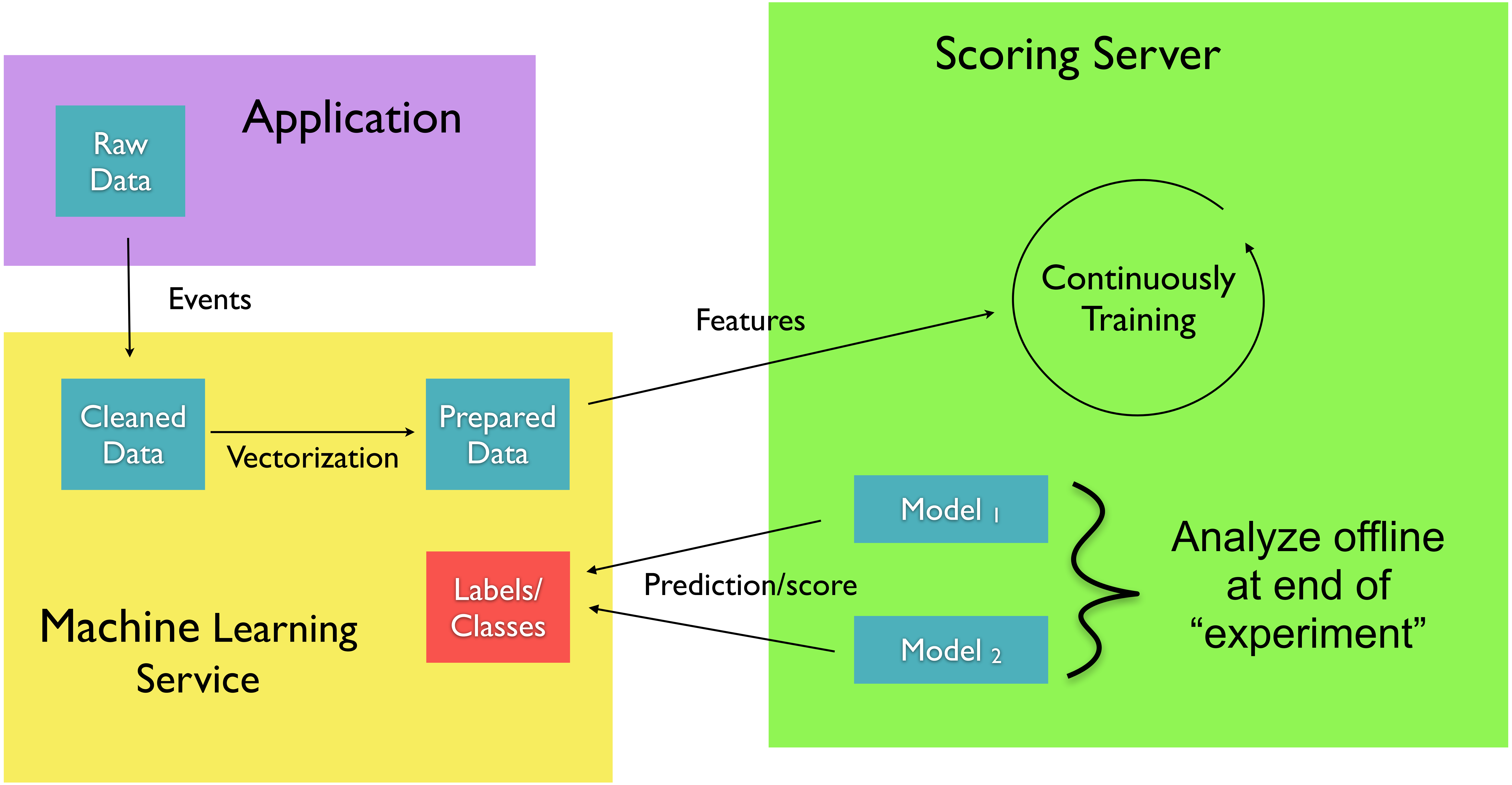
Batch Offline Evaluation



Live A/B Testing



Live A/B Testing



Multi-armed Bandit