Machine Learning Pipelines

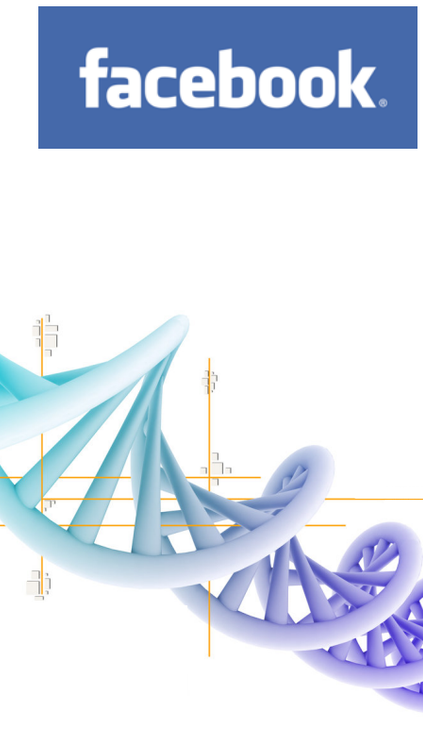
Ameet Talwalkar

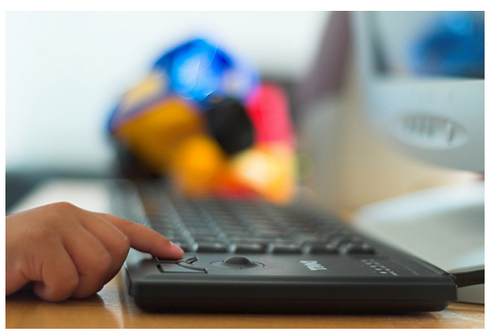
January 13, 2015

Rapid growth of massive datasets

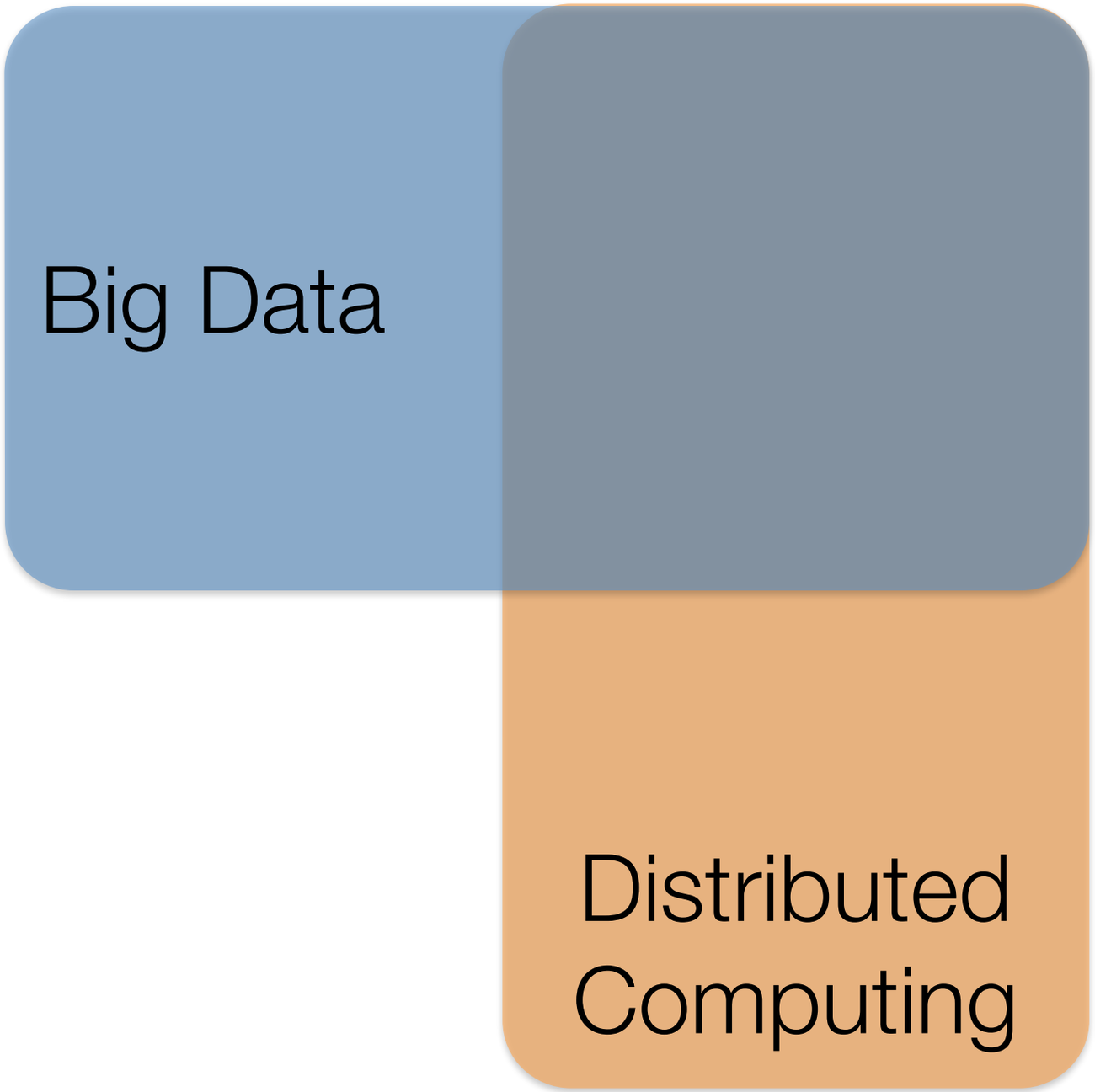
E.g., Online activity, Science, Sensor networks

Big Data





Distributed clusters are pervasive



amazon web services™

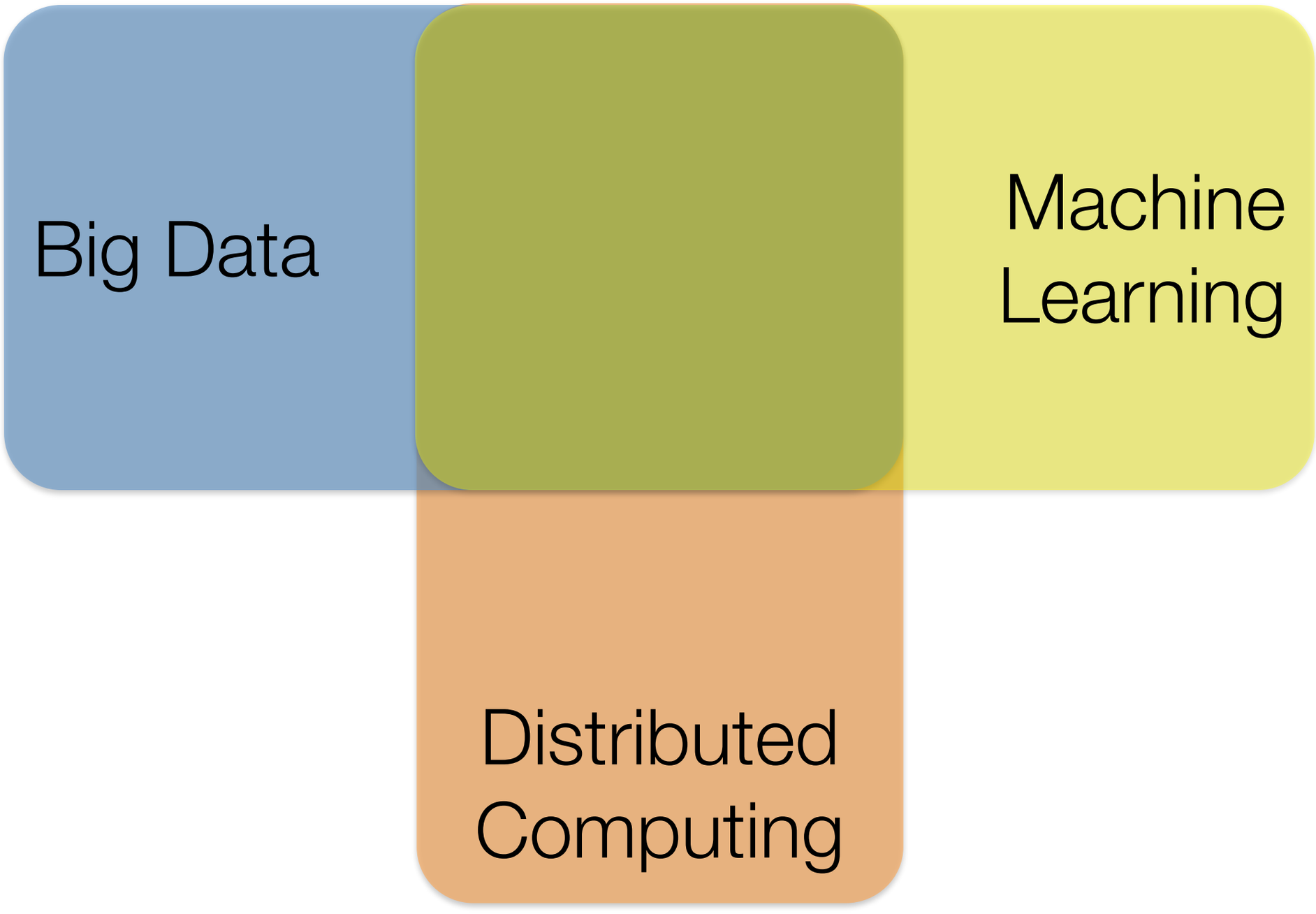
Google Compute Engine

rackspace
the open cloud company

Azure

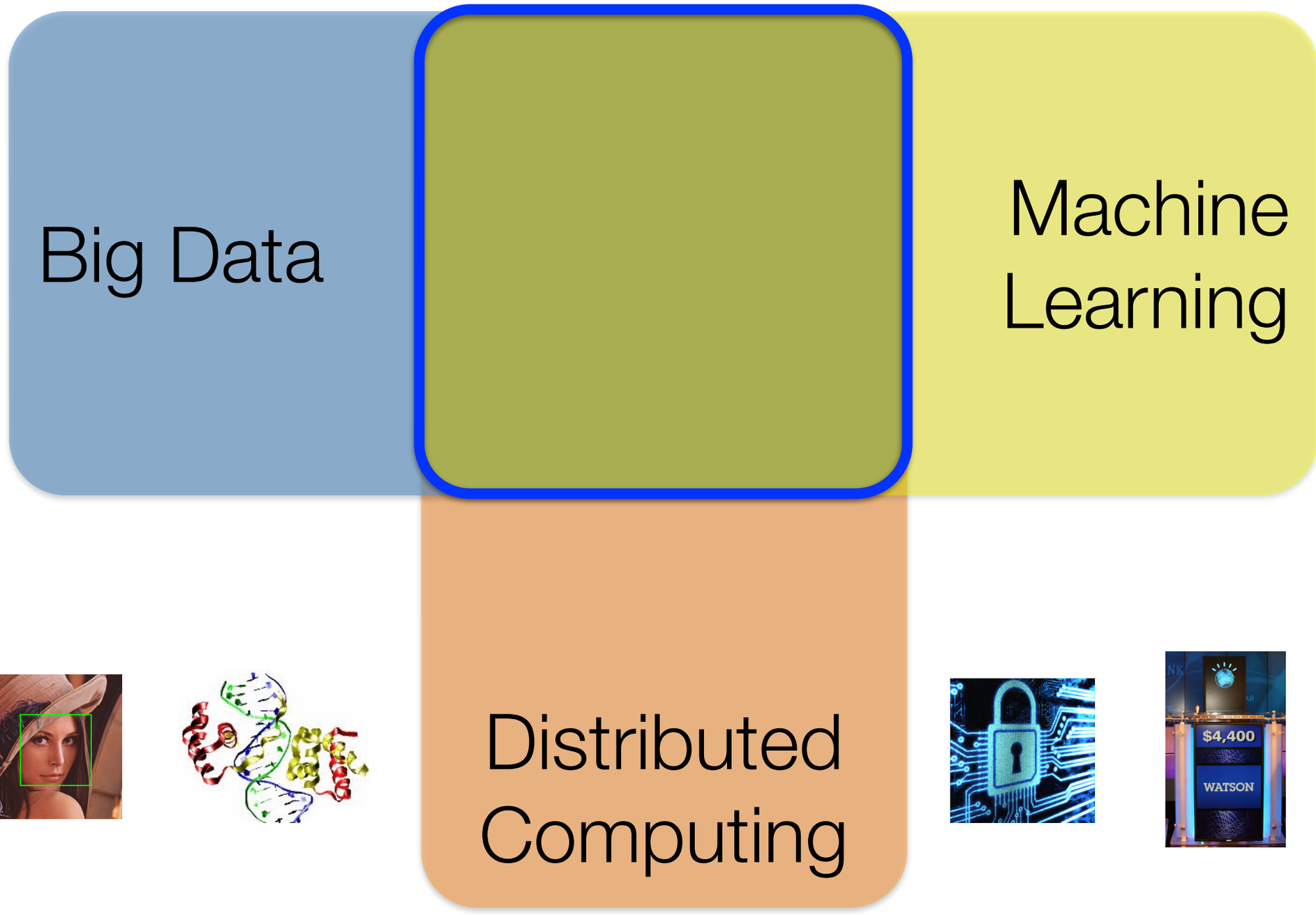
Mature suite of algorithms for common problems

e.g., classification, regression, collaborative filtering, clustering

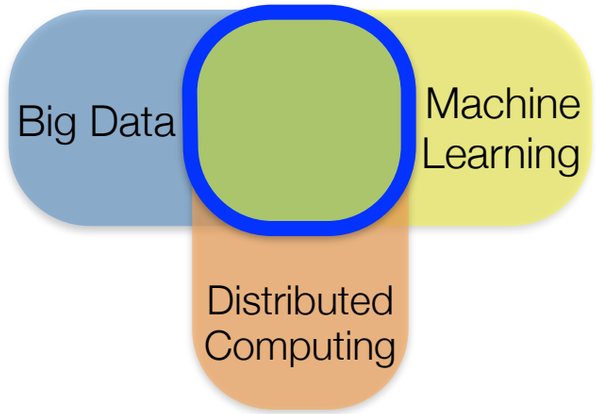


ML is applied everywhere

E.g., Personalized product recommendations, Speech recognition, Face detection, Protein function or structure prediction, Fraud detection, Spam filtering, Playing chess or competing on Jeopardy, Unassisted vehicle control, Medical diagnosis, ...



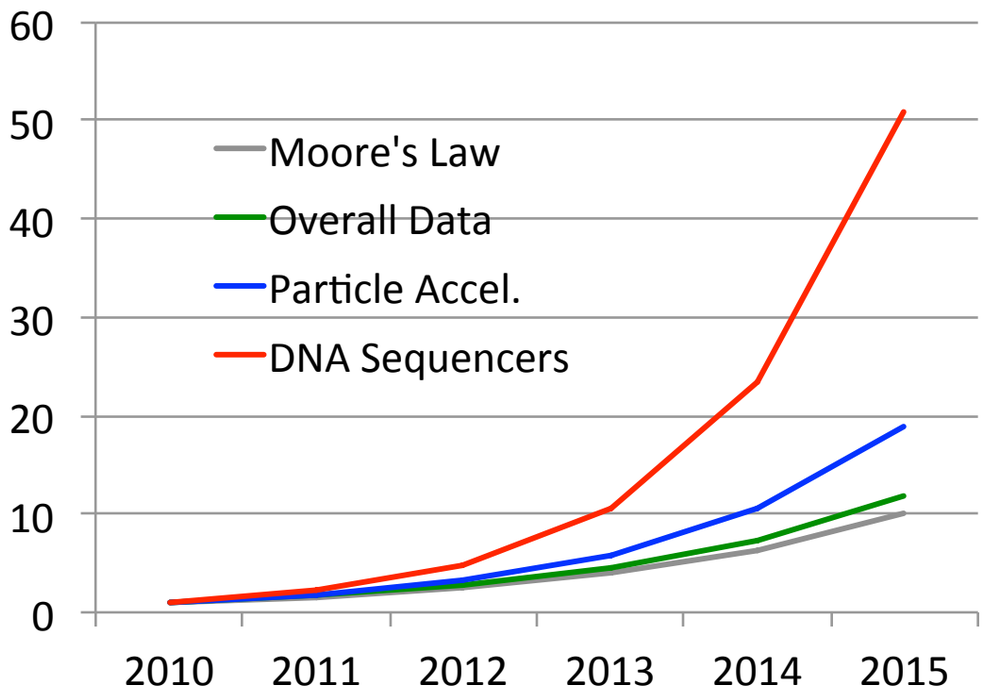
Challenge: Scalability



Classic ML techniques

~~have proven their value at small scales~~

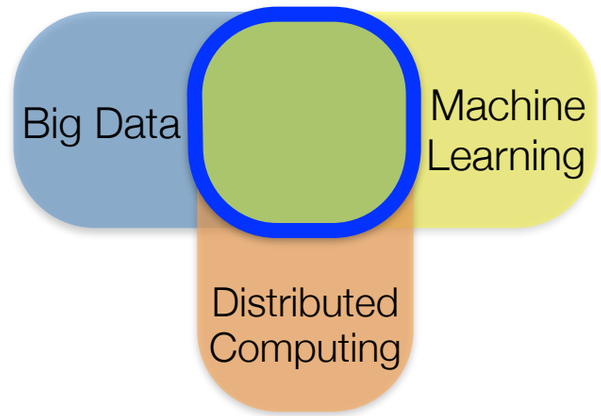
are not always suitable for modern datasets



Data Grows faster than Moore’s Law

[IDC report, Kathy Yelick, LBNL]

Overview



Lecture1: ML Pipeline Basics

- Feature Extraction
- Supervised ML
- Model Evaluation
- Millionsong example

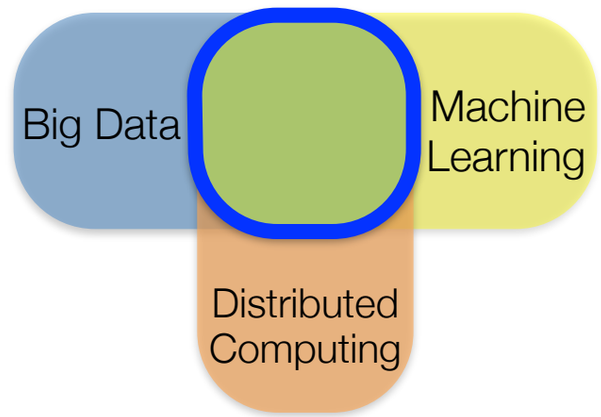
Lecture2: Distributed ML Principles

- Computation, Storage, Communication
- Gradient Descent
- Three Rules of Thumb

Lecture3: Click-through Rate Prediction

- Online advertising
- Logistic Regression
- One-hot-encoding and Feature hashing

Overview



Lecture1: ML Pipeline Basics

- Feature Extraction
- Supervised ML
- Model Evaluation
- Millionsong example

Lecture2: Distributed ML Principles

- Computation, Storage, Communication
- Gradient Descent
- Three Rules of Thumb

Lecture3: Click-through Rate Prediction

- Online advertising
- Logistic Regression
- One-hot-encoding and Feature hashing
- Overview of MLlib [unrelated to CTR]

Typical Pipeline

- Feature Extraction
- Supervised ML
- Model Evaluation
- Spam Classification Example

Millionsong Example

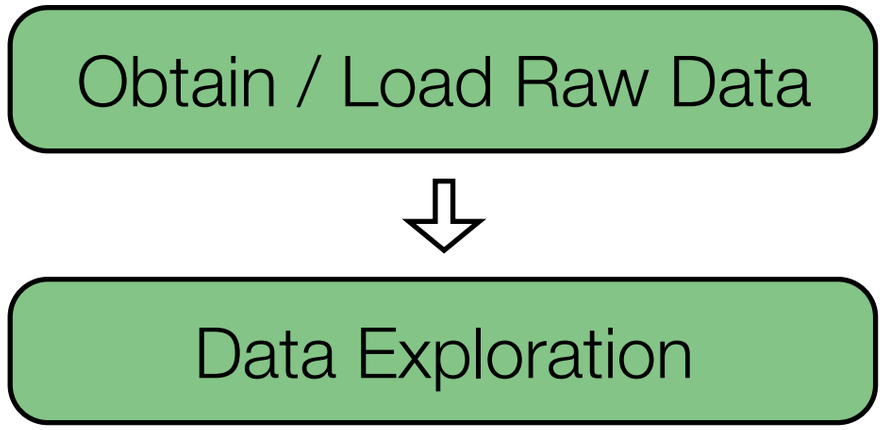
- Polynomial features
- Linear regression and Normal Equations
- Ridge Regression

Typical ML Pipeline

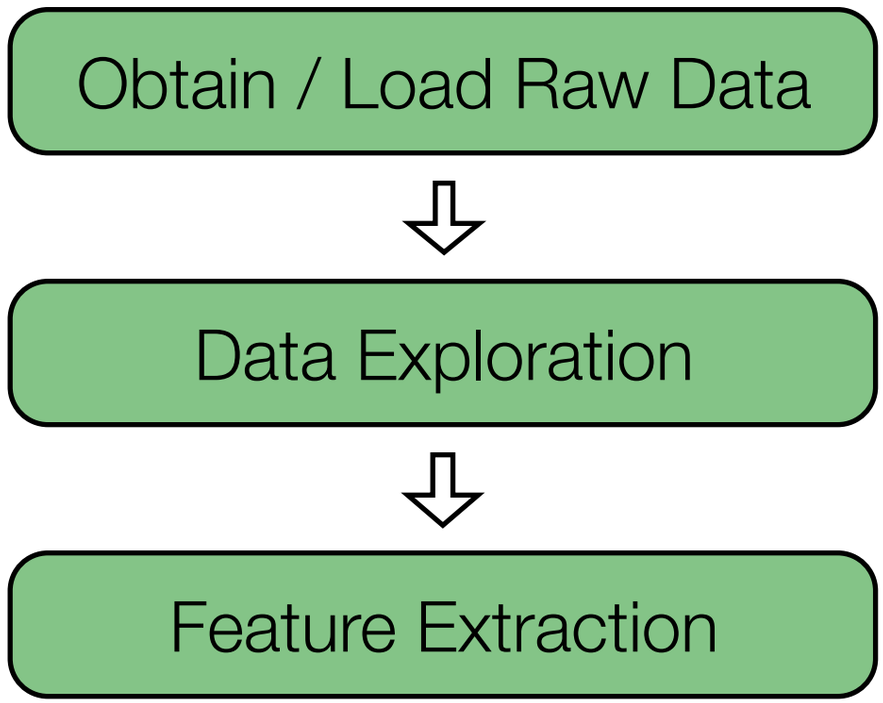
Typical ML Pipeline

Obtain / Load Raw Data

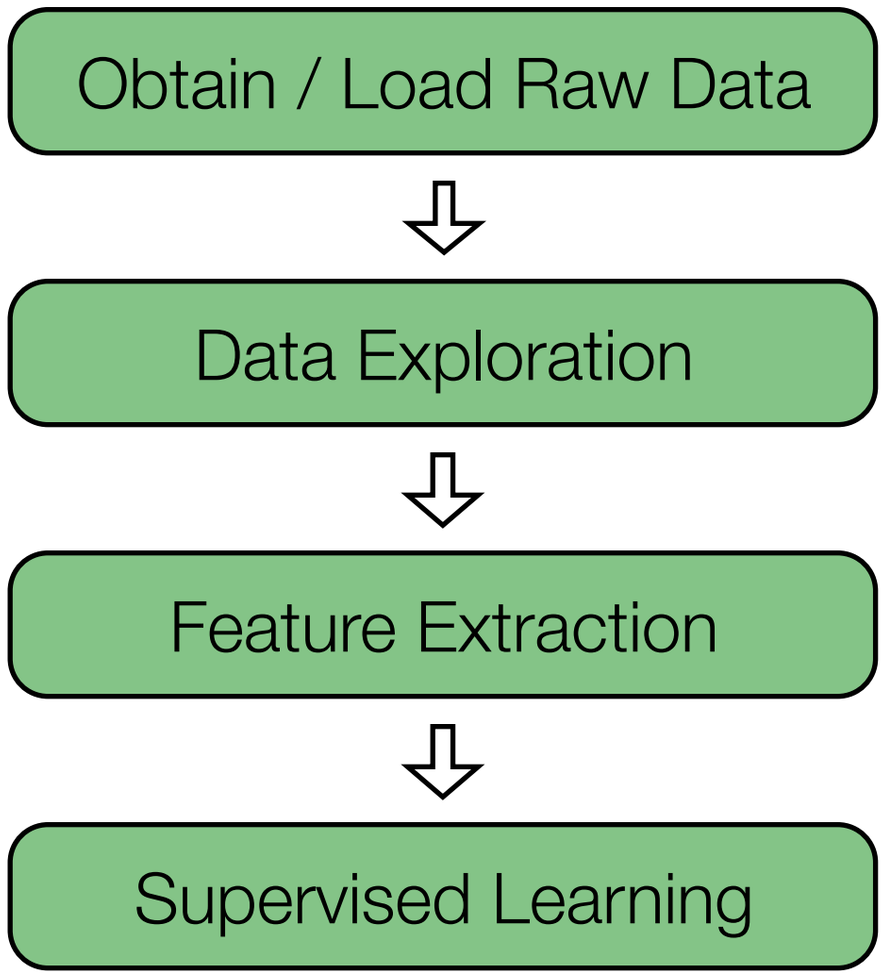
Typical ML Pipeline



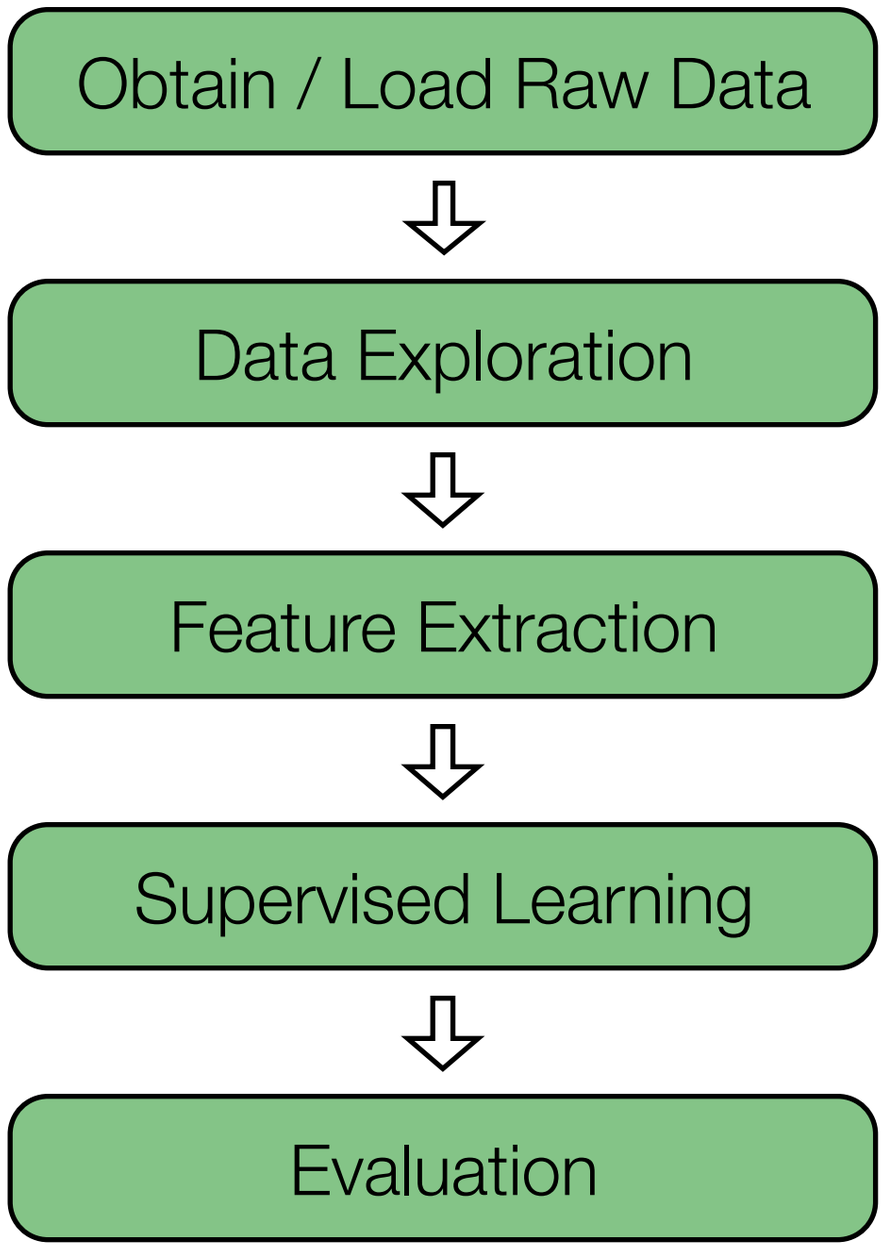
Typical ML Pipeline



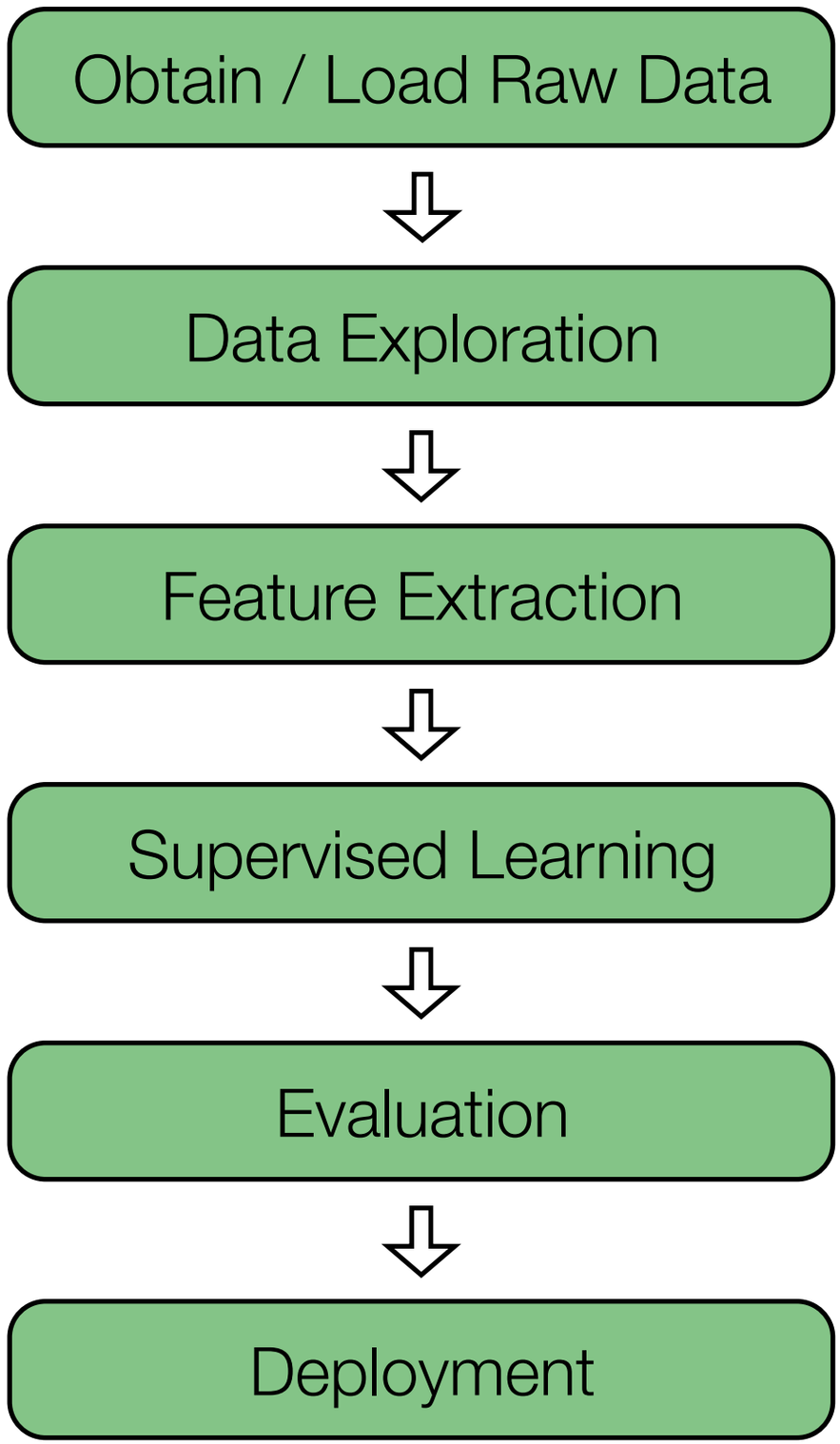
Typical ML Pipeline



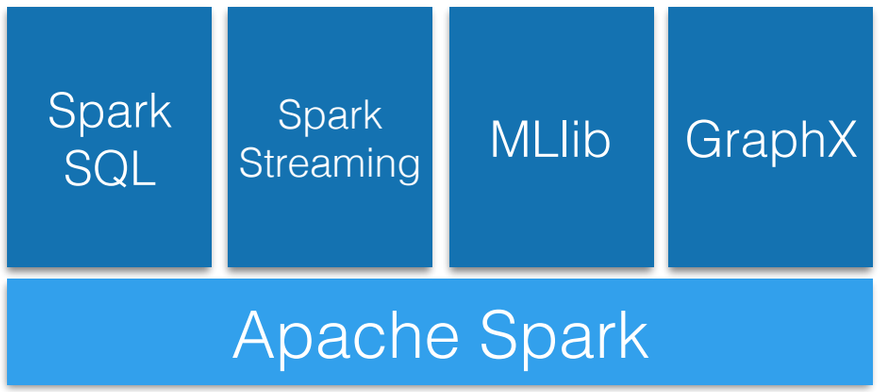
Typical ML Pipeline

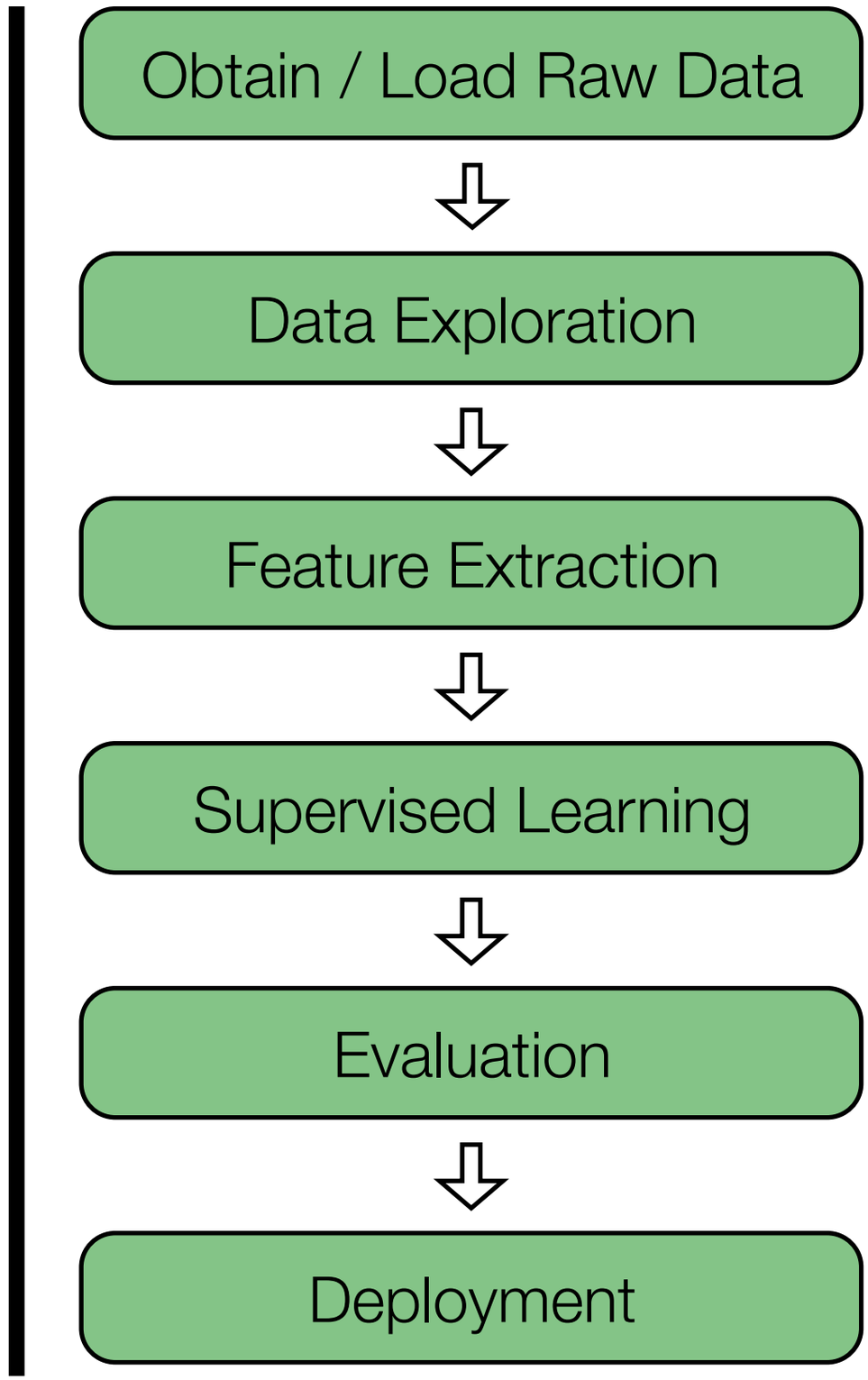


Typical ML Pipeline

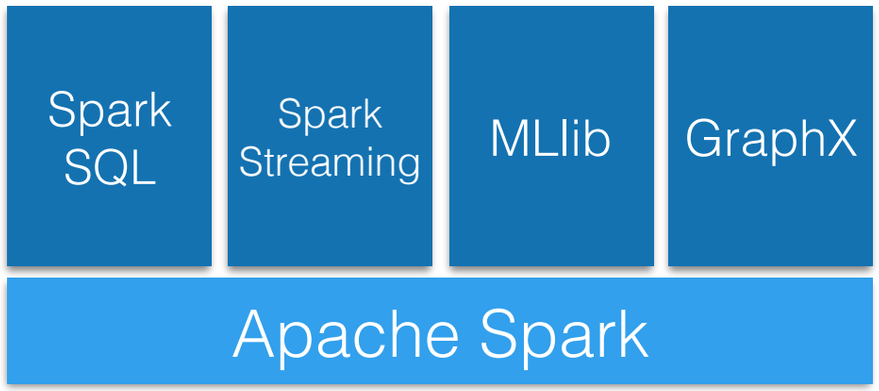


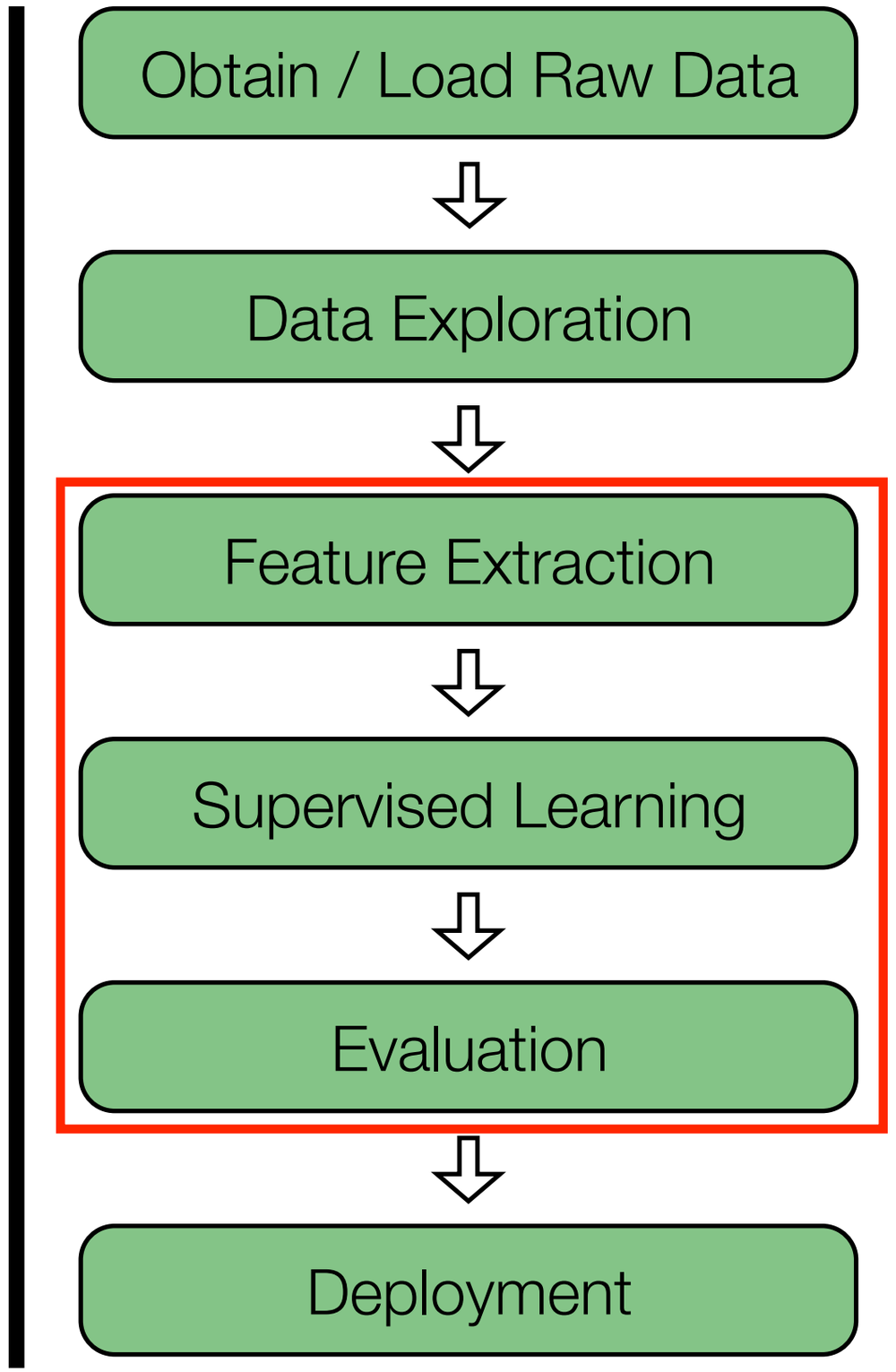
Typical ML Pipeline



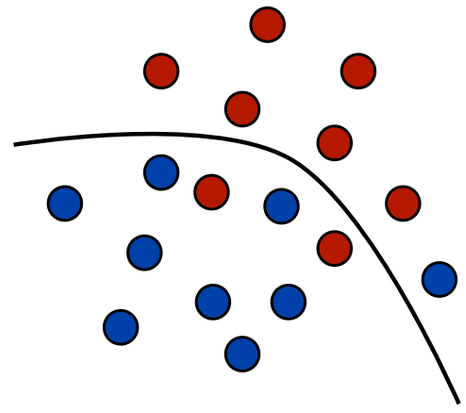


Typical ML Pipeline

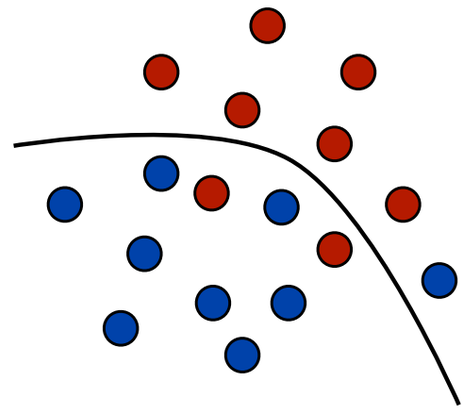




Classification

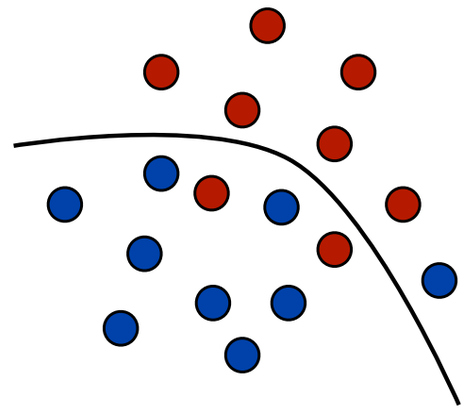


Classification



Goal: Learn a mapping from entities to discrete labels given a set of training examples (supervised learning)

Classification

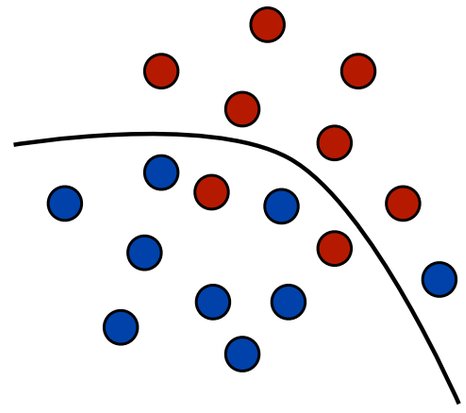


Goal: Learn a mapping from entities to discrete labels given a set of training examples (supervised learning)

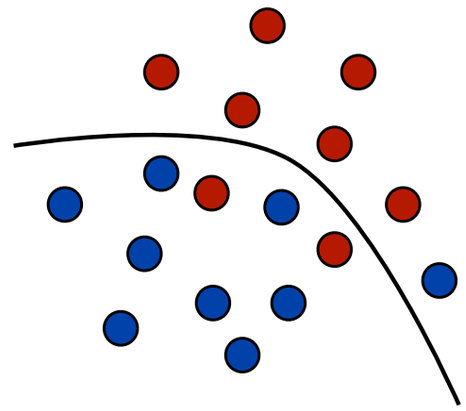
Example: Spam Classification

- Entities are emails
- Labels are {spam, not-spam} (Binary Classification)
- Given past labeled emails, we want to predict whether a new email is spam or not-spam

Classification



Classification

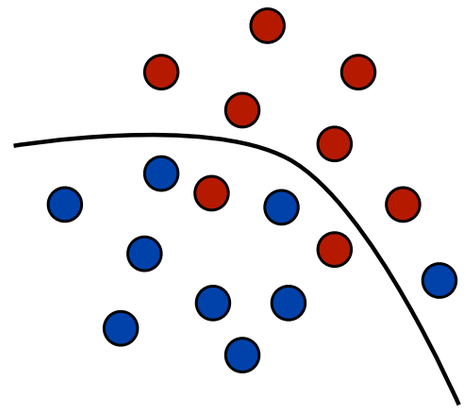


Other Examples:

Fraud detection

- User activity → {fraud, not fraud}

Classification



Other Examples:

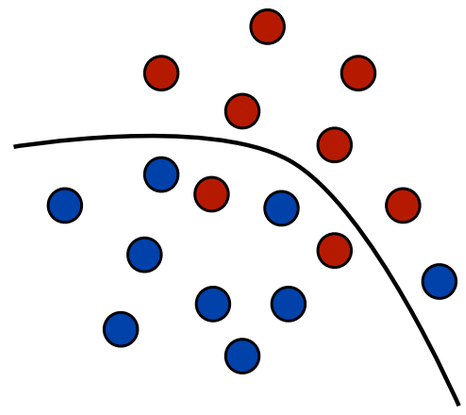
Fraud detection

- User activity → {fraud, not fraud}

Face detection

- Images → set of people

Classification



Other Examples:

Fraud detection

- User activity → {fraud, not fraud}

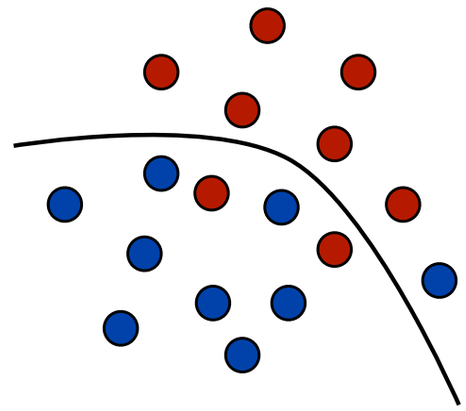
Face detection

- Images → set of people

Link prediction

- Users → {suggest link, don’t suggest link}

Classification



Other Examples:

Fraud detection

- User activity → {fraud, not fraud}

Face detection

- Images → set of people

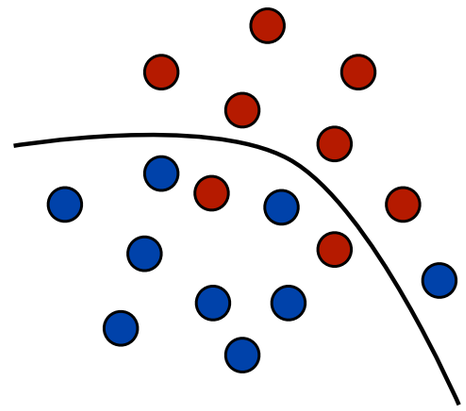
Link prediction

- Users → {suggest link, don't suggest link}

Clickthrough rate prediction (tomorrow's exercise)

- User and ads → {click, no click}

Classification



Other Examples:

Fraud detection

- User activity → {fraud, not fraud}

Face detection

- Images → set of people

Link prediction

- Users → {suggest link, don't suggest link}

Clickthrough rate prediction (tomorrow's exercise)

- User and ads → {click, no click}

Many others…

Classification Pipeline

Classification Pipeline

In the supervised setting, the training data consists of a set of examples

Classification Pipeline

In the supervised setting, the training data consists of a set of examples

Each example is a pair consisting of an input entity and a desired output value (label)

E.g., Spam Classification

Entity

```
From: illegitimate@bad.com

"Eliminate your debt by
giving us your money..."
```

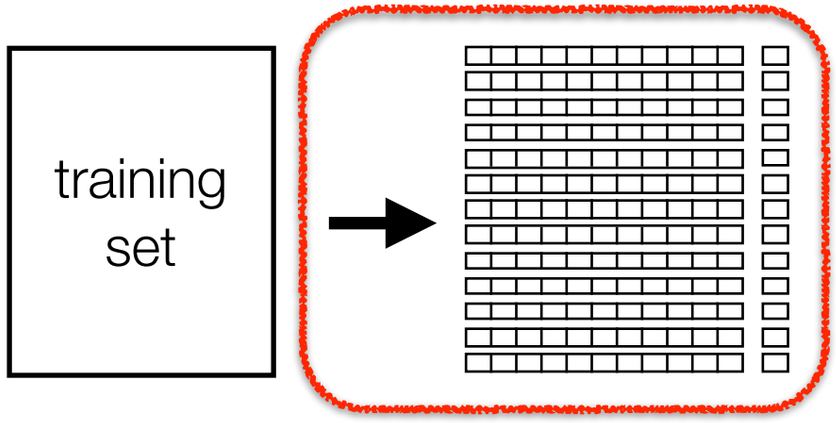
```
From: bob@good.com

"Hi, it's been a while!
How are you? ..."
```

E.g., Spam Classification

Entity	Label
From: illegitimate@bad.com "Eliminate your debt by giving us your money..."	spam
From: bob@good.com "Hi, it's been a while! How are you? ..."	not-spam

Classification Pipeline



Most classifiers require ***numeric*** descriptions of entities (‘features’)

Data Types

```
<html>
    <head> ... </head>
    <body>
        <div id="HEADER"> ... </div>
        <div id="NAVIGATION"> ... </div>
        <div id="LEFTCOLUMN">
            <div class="info">
                <h3>News Teaser</h3>
                <p>Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ...</p>
            </div>
        </div>
        <div id="CENTERCOLUMN">
            <div class="article">
                <h1>New OpenCms concepts</h1>
                <p>
                    Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut
                    labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores
                    et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.
                </p>
            </div>
        </div>
        <div id="RIGHTCOLUMN">
            <div class="toolbox">
                <h3>Toolbox</h3>
                <p>Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ...</p>
            </div>
        </div>
        <div id="FOOTER"> ... </div>
    </body>
</html>
```

Data Types

```
<html>
    <head> ... </head>
    <body>
        <div id="HEADER"> ... </div>
        <div id="NAVIGATION"> ... </div>
        <div id="LEFTCOLUMN">
            <div class="info">
                <h3>News Teaser</h3>
                <p>Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ...</p>
            </div>
        </div>
        <div id="CENTERCOLUMN">
            <div class="article">
                <h1>New OpenCms concepts</h1>
                <p>
                    Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut
                    labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores
                    et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.
                </p>
            </div>
        </div>
        <div id="RIGHTCOLUMN">
            <div class="toolbox">
                <h3>Toolbox</h3>
                <p>Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ...</p>
            </div>
        </div>
        <div id="FOOTER"> ... </div>
    </body>
</html>
```

Web hypertext

Data Types

sample_email.eml - Notepad

File Edit Format View Help

```
X-Receiver: user1@adminstop.com
X-Sender: user1@adminstop.com
Auto-Submitted: auto-generated
Thread-Topic: WinDeveloper Message Recall - evaluation will expire in 10 days
Return-Path:
thread-index: AcwuUbmEZh2DcyZKTcafvEY0B934PQ==
From: <user1@adminstop.com>
To: <user1@adminstop.com>
Subject: WinDeveloper Message Recall - evaluation will expire in 10 days
Date: Sun, 19 Jun 2011 09:23:02 +0200
Message-ID: <D302A151D8B341F881C2457C6D4E52DB@SBS11>
MIME-Version: 1.0
Content-Type: text/plain
Content-Transfer-Encoding: 7bit
X-Mailer: Microsoft CDO for Windows 2000
Content-Class: urn:content-classes:message
Importance: normal
Priority: normal
X-MimeOLE: Produced By Microsoft MimeOLE V6.1.7600.16385

Thank you for evaluating WinDeveloper Message Recall. Please note that the
evaluation period will expire in 10 days.

Once evaluation is over, the product will not process any recall requests and
the system will behave as if the product was not installed.

For assistance on licensing please contact WinDeveloper on:
sales@windeveloper.com

For technical support please email to:
support@windeveloper.com

Licenses may also be ordered from:
http://www.windeveloper.com/recall/recall_register.htm

From product information check:
http://www.windeveloper.com/recall

This notification is being sent by your WinDeveloper Message Recall
installation. Notification settings can be changed at the configuration under
the Status Reporting category.

Regards,

WinDeveloper Message Recall
http://www.windeveloper.com/recall
```

Data Types

Email

sample_email.eml - Notepad

File Edit Format View Help

```
X-Receiver: user1@adminstop.com
X-Sender: user1@adminstop.com
Auto-Submitted: auto-generated
Thread-Topic: WinDeveloper Message Recall - evaluation will expire in 10 days
Return-Path:
thread-index: AcwuUbmEZh2DcyZKTcafvEY0B934PQ==
From: <user1@adminstop.com>
To: <user1@adminstop.com>
Subject: WinDeveloper Message Recall - evaluation will expire in 10 days
Date: Sun, 19 Jun 2011 09:23:02 +0200
Message-ID: <D302A151D8B341F881C2457C6D4E52DB@SBS11>
MIME-Version: 1.0
Content-Type: text/plain
Content-Transfer-Encoding: 7bit
X-Mailer: Microsoft CDO for Windows 2000
Content-Class: urn:content-classes:message
Importance: normal
Priority: normal
X-MimeOLE: Produced By Microsoft MimeOLE V6.1.7600.16385

Thank you for evaluating WinDeveloper Message Recall. Please note that the
evaluation period will expire in 10 days.

Once evaluation is over, the product will not process any recall requests and
the system will behave as if the product was not installed.

For assistance on licensing please contact WinDeveloper on:
sales@windeveloper.com

For technical support please email to:
support@windeveloper.com

Licenses may also be ordered from:
http://www.windeveloper.com/recall/recall_register.htm

From product information check:
http://www.windeveloper.com/recall

This notification is being sent by your WinDeveloper Message Recall
installation. Notification settings can be changed at the configuration under
the Status Reporting category.

Regards,

WinDeveloper Message Recall
http://www.windeveloper.com/recall
```

Data Types

Data Types

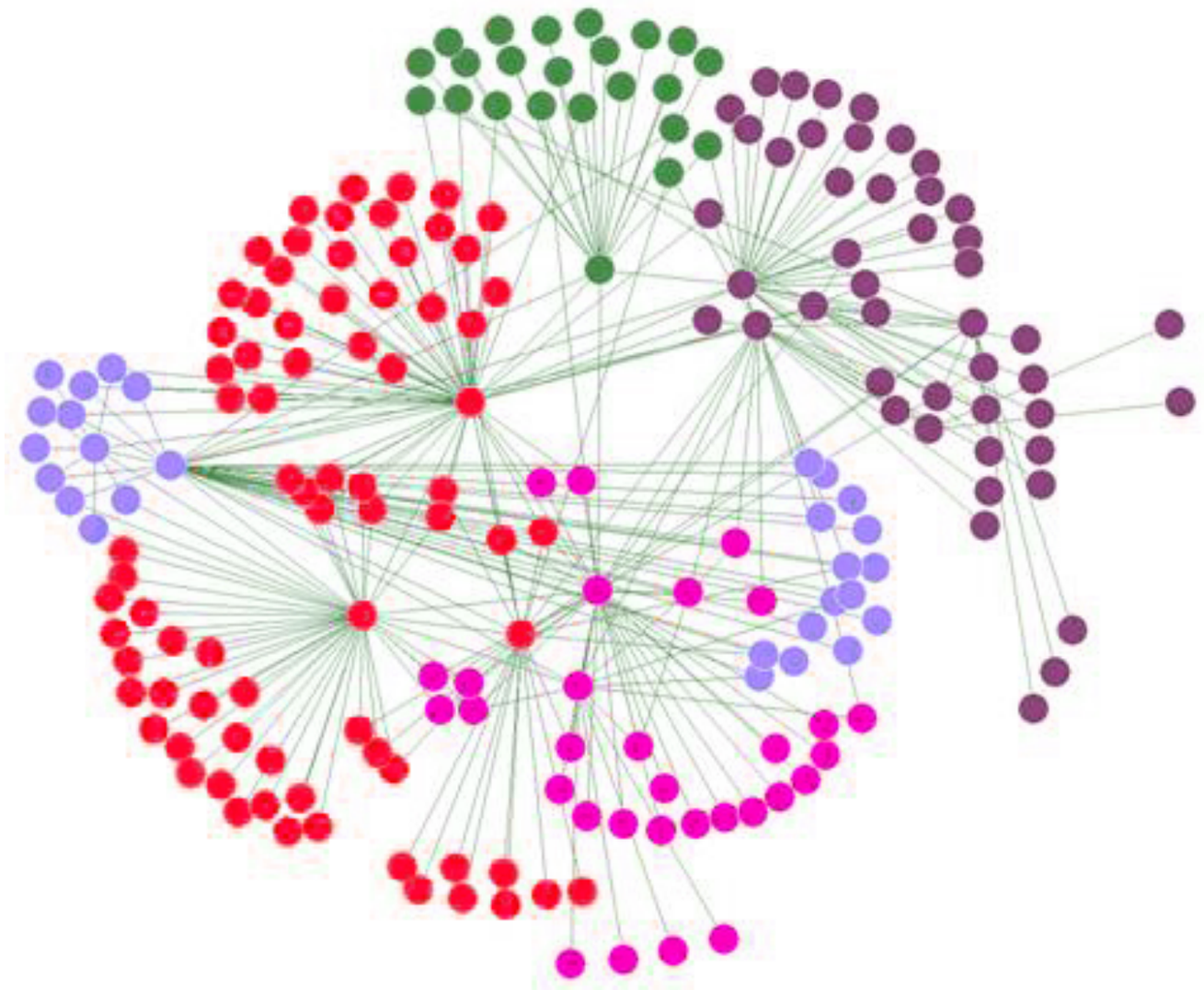
Genomic Data,
e.g., SNPs

Data Types

Data Types

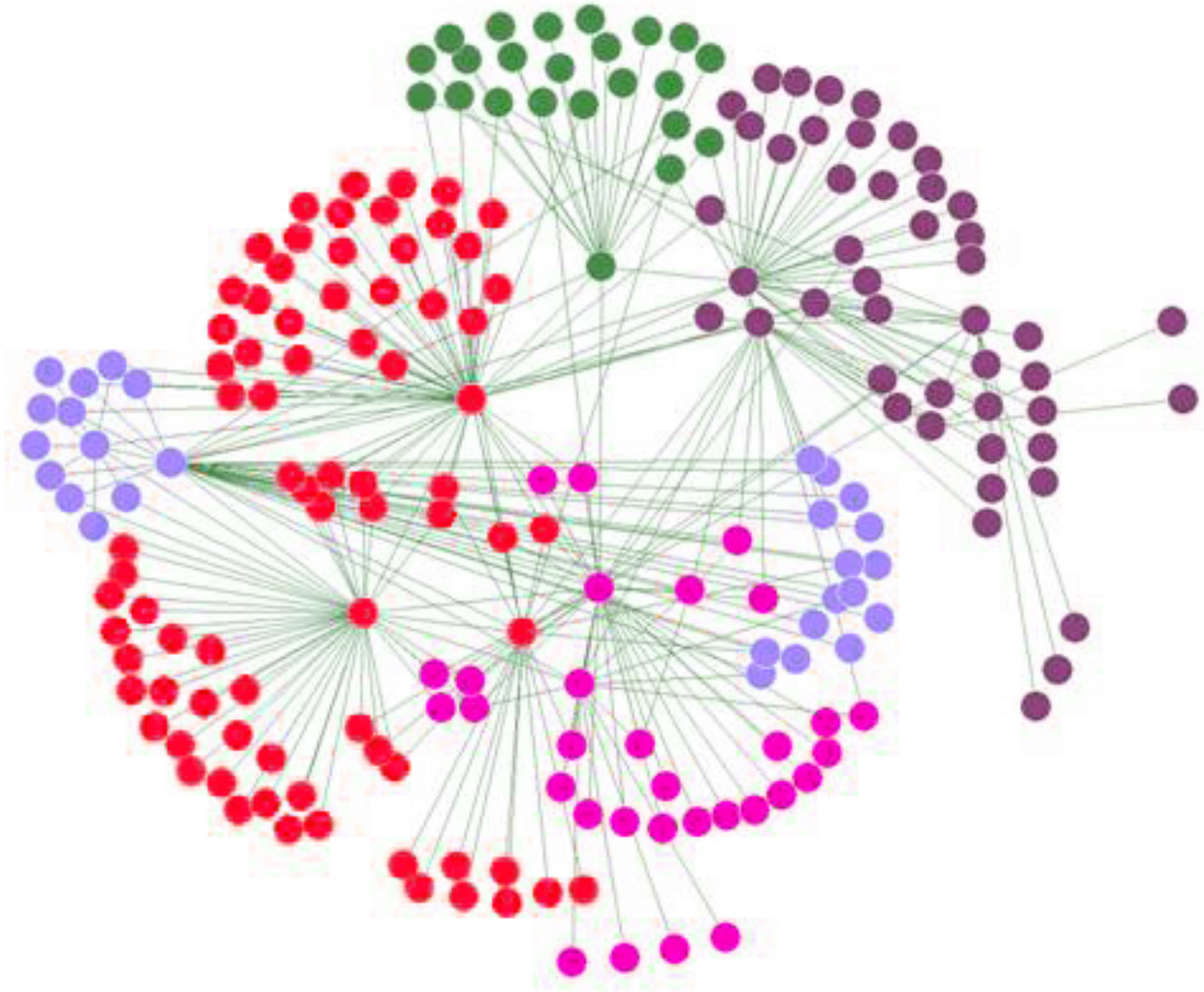
Images

Data Types

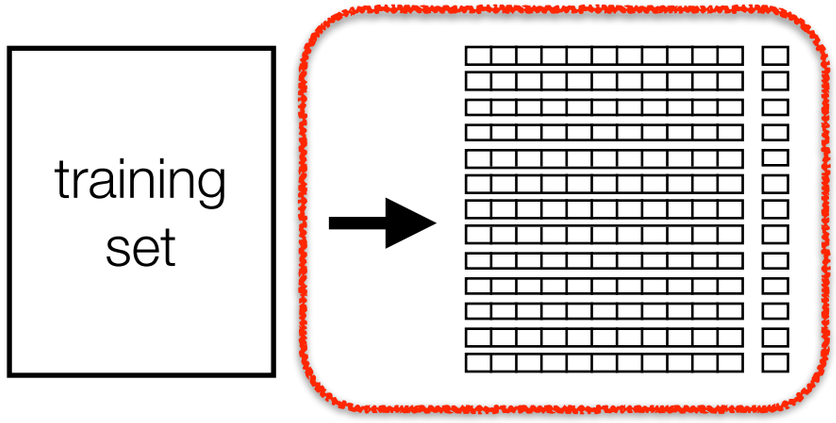


Data Types

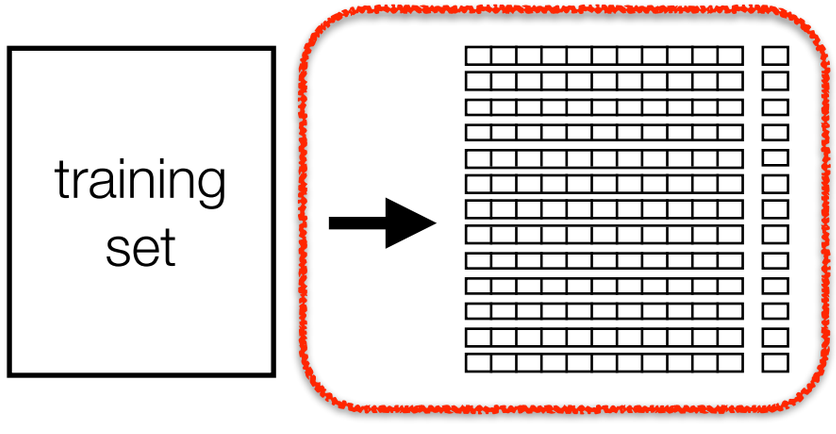
(Social) Networks / Graphs



Classification Pipeline

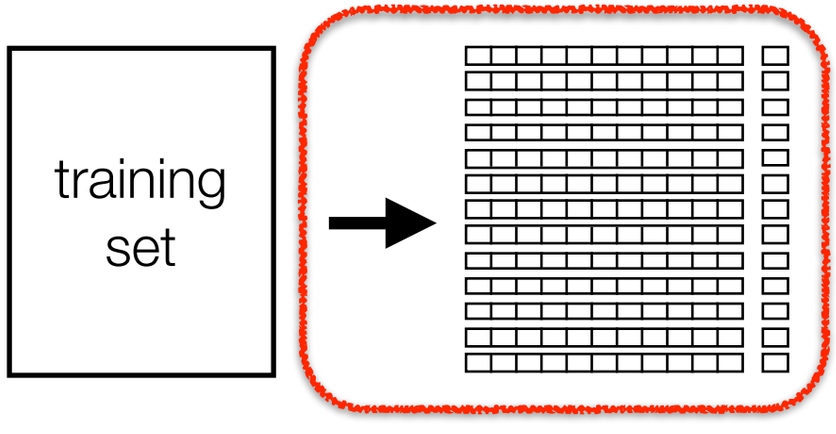


Classification Pipeline



Feature extraction typically transforms each entity into a vector of real numbers (features)

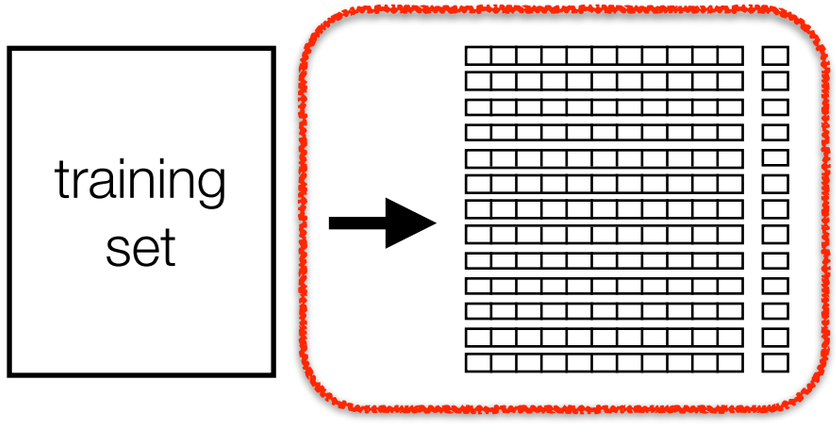
Classification Pipeline



Feature extraction typically transforms each entity into a vector of real numbers (features)

- Opportunity to incorporate domain knowledge
- Useful even when original data is already numeric (as you'll see in your first exercise)

Classification Pipeline



Feature extraction typically transforms each entity into a vector of real numbers (features)

- Opportunity to incorporate domain knowledge
- Useful even when original data is already numeric (as you'll see in your first exercise)

Success or failure of a classifier often depends on choosing good descriptions of objects!!

E.g., “Bag of Words”

Entities are documents

Build Vocabulary

```
From: illegitimate@bad.com

"Eliminate your debt by
giving us your money..."
```

```
From: bob@good.com

"Hi, it's been a while!
How are you? ..."
```

E.g., “Bag of Words”

Entities are documents

Build Vocabulary

```
From: illegitimate@bad.com

"Eliminate your debt by
giving us your money..."
```

```
From: bob@good.com

"Hi, it's been a while!
How are you? ..."
```

Vocabulary

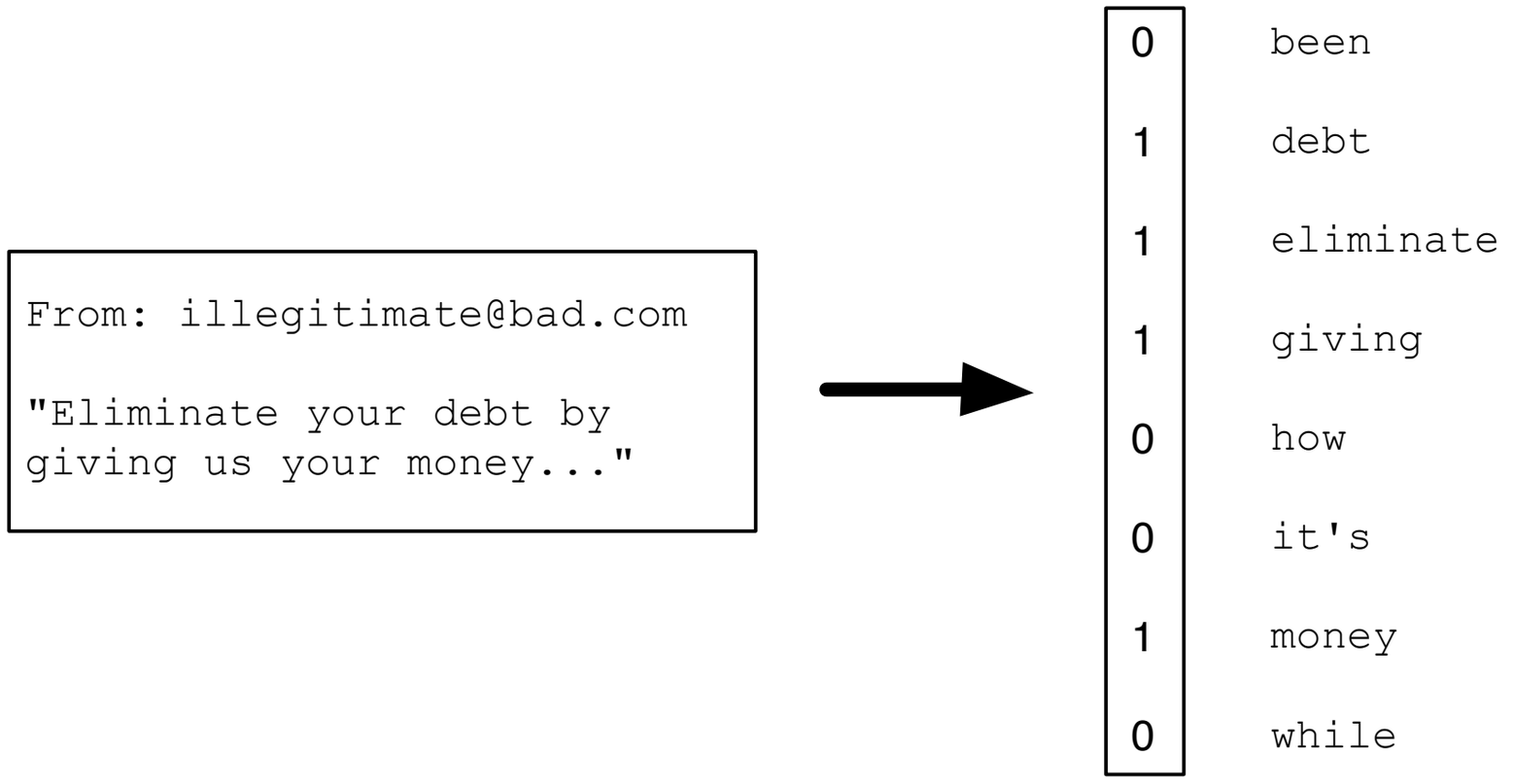
been
debt
eliminate
giving
how
it's
money
while

E.g., “Bag of Words”

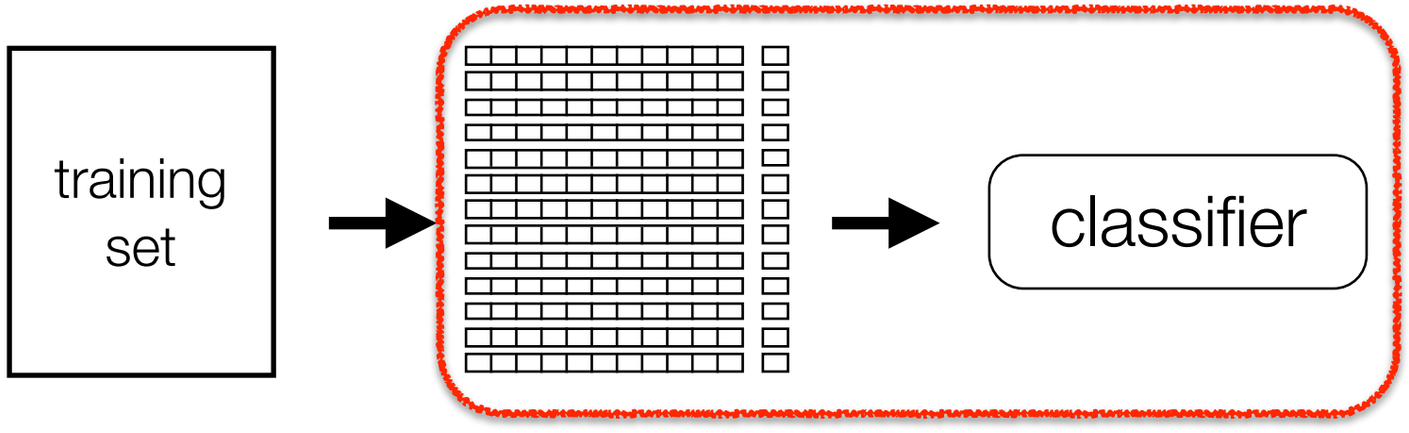
Entities are documents

Build Vocabulary

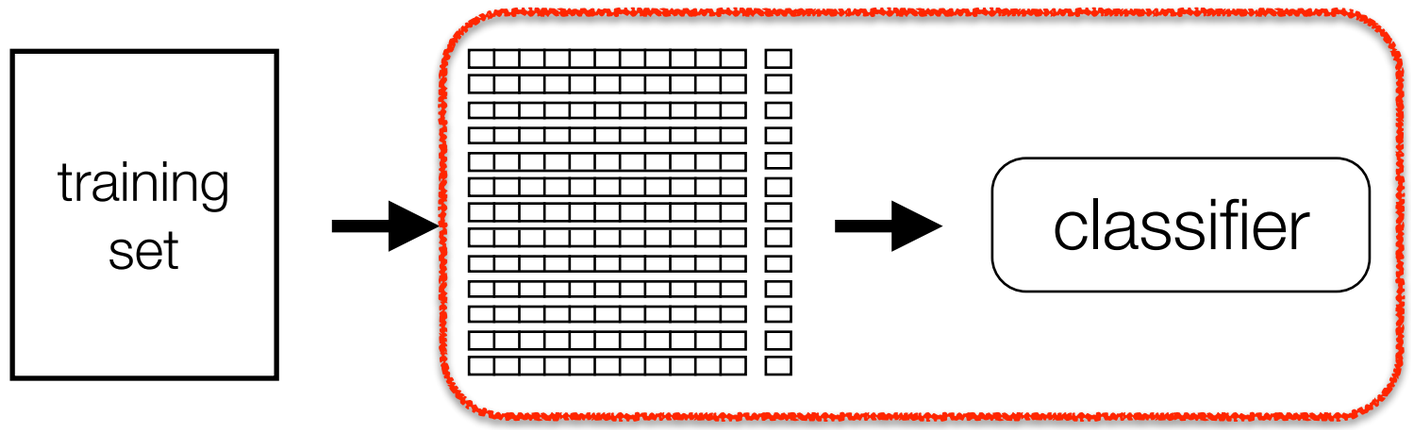
Derive feature vectors from Vocabulary



Classification Pipeline

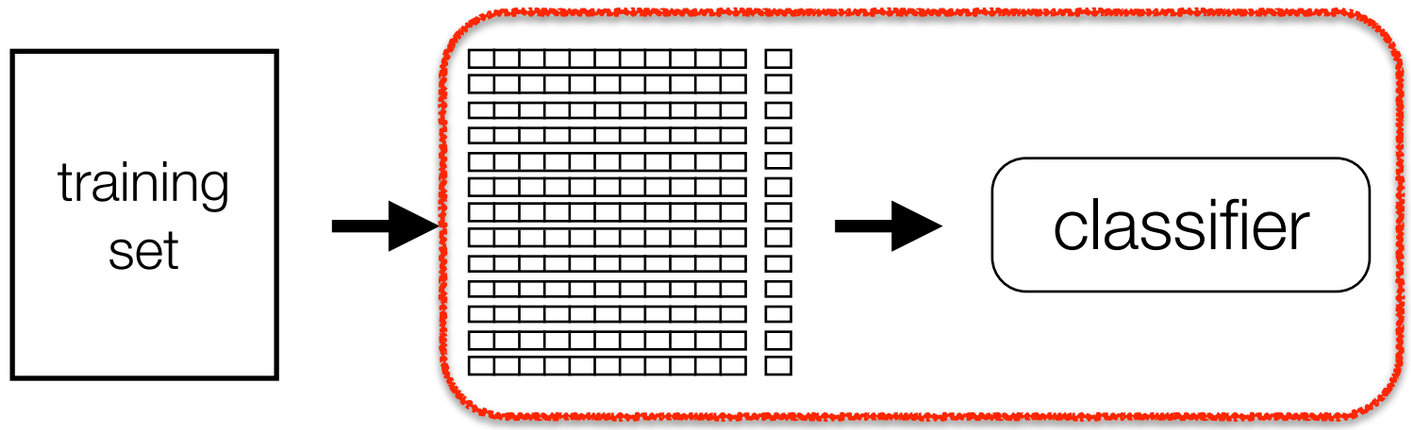


Classification Pipeline



Train a classifier using the training data

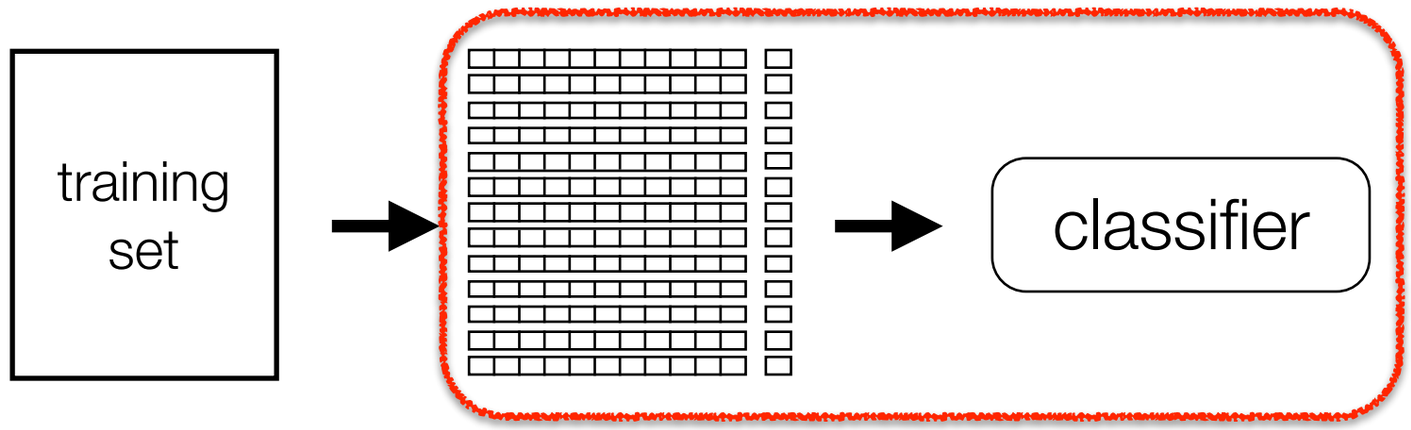
Classification Pipeline



Train a classifier using the training data

- Common classifiers include Logistic Regression, SVMs, Decision Trees, Random Forests, etc.

Classification Pipeline

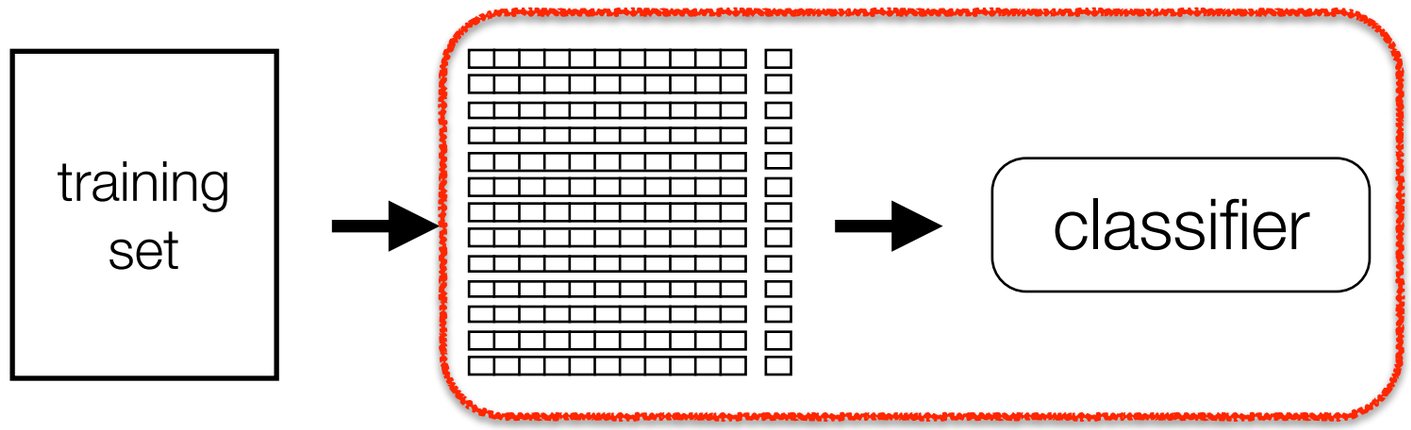


Train a classifier using the training data

- Common classifiers include Logistic Regression, SVMs, Decision Trees, Random Forests, etc.

Training (especially at scale) often involves iteratively optimizing a (convex) function

Classification Pipeline



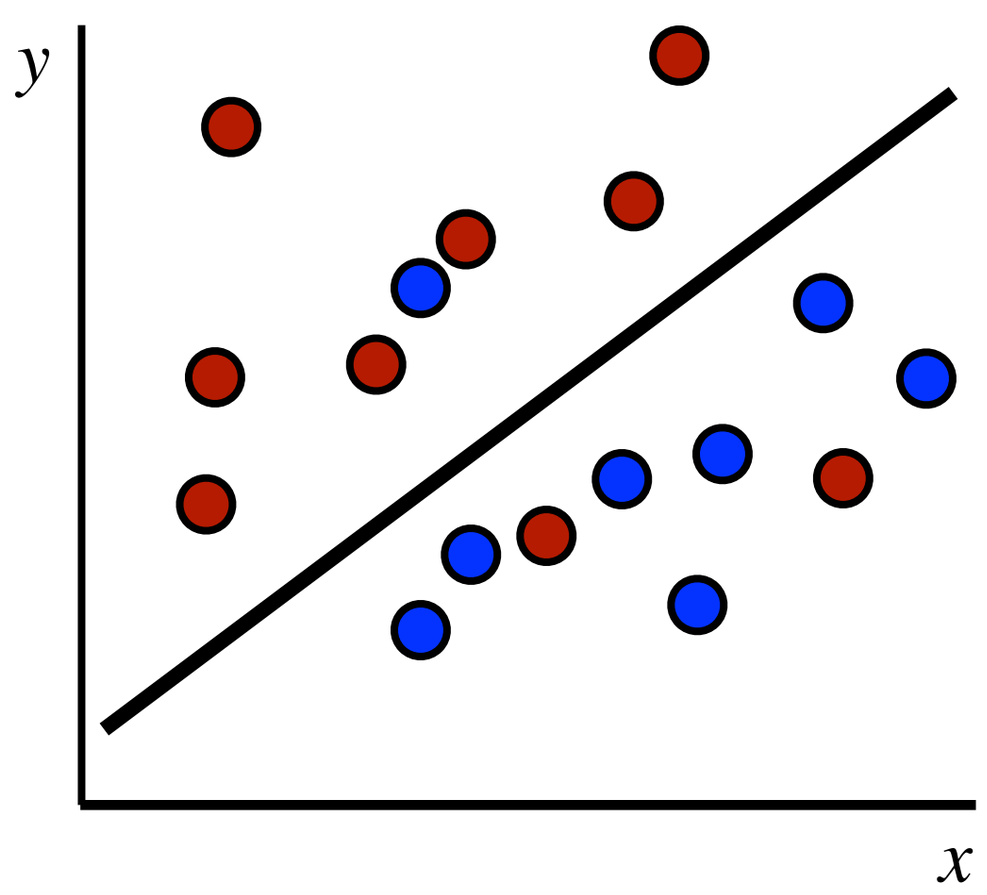
Train a classifier using the training data

- Common classifiers include Logistic Regression, SVMs, Decision Trees, Random Forests, etc.

Training (especially at scale) often involves iteratively optimizing a (convex) function

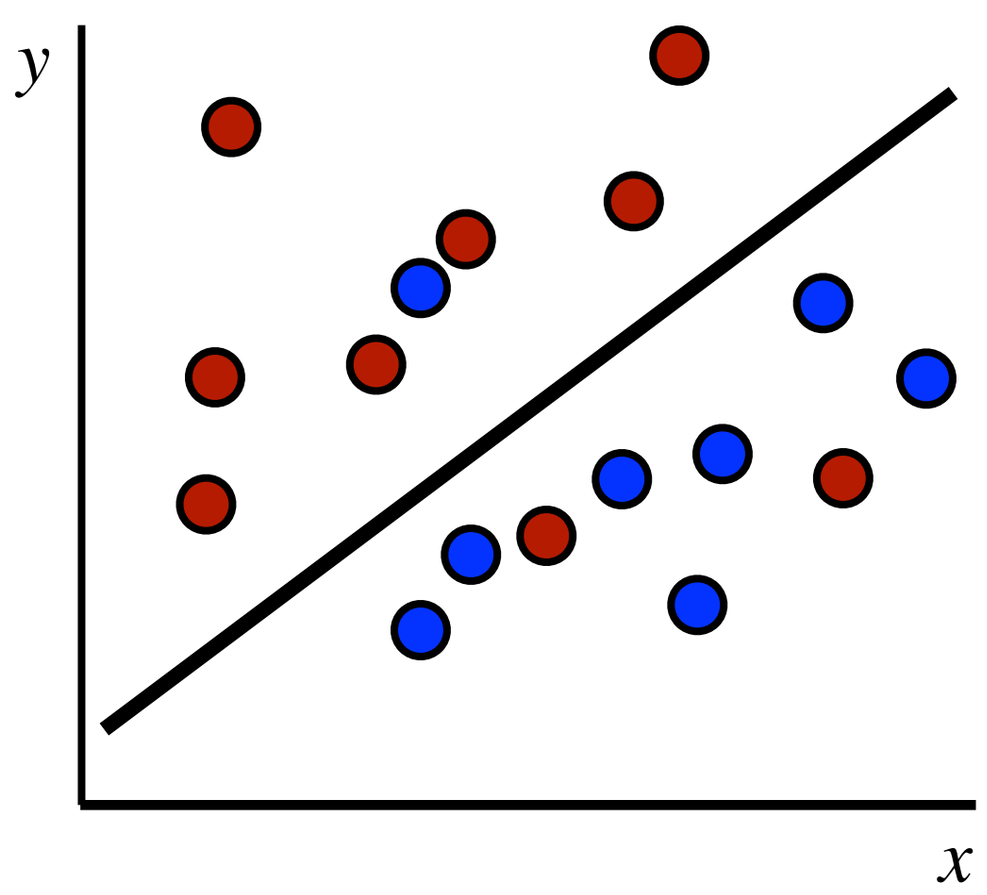
- We'll talk about gradient descent in the next lecture

E.g., Logistic Regression



E.g., Logistic Regression

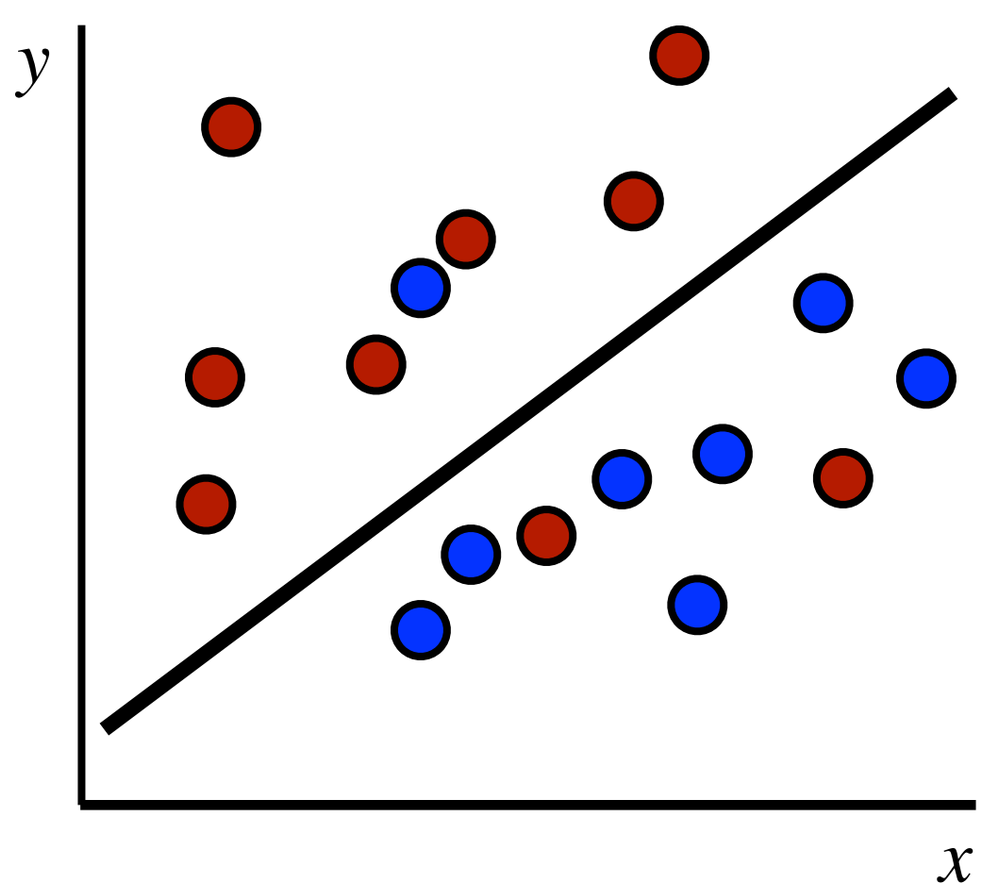
Goal: Find linear decision boundary



E.g., Logistic Regression

Goal: Find linear decision boundary

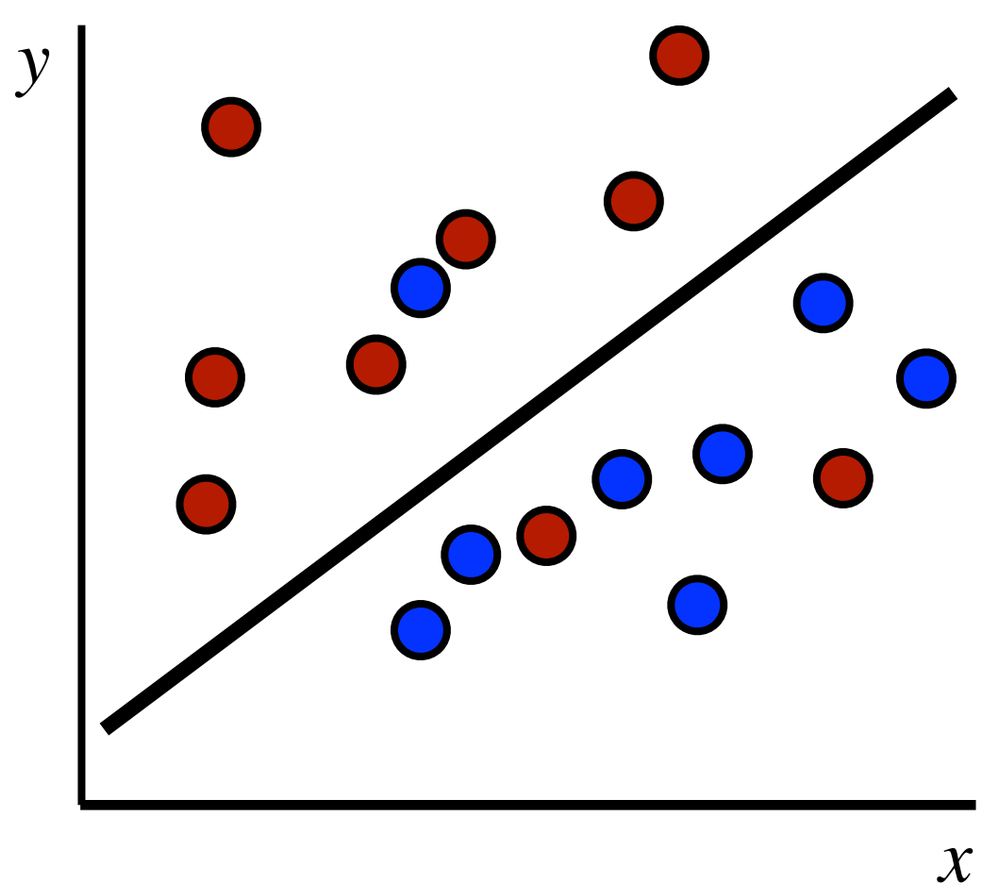
- Model Parameters: slope (w) and intercept (b)



E.g., Logistic Regression

Goal: Find linear decision boundary

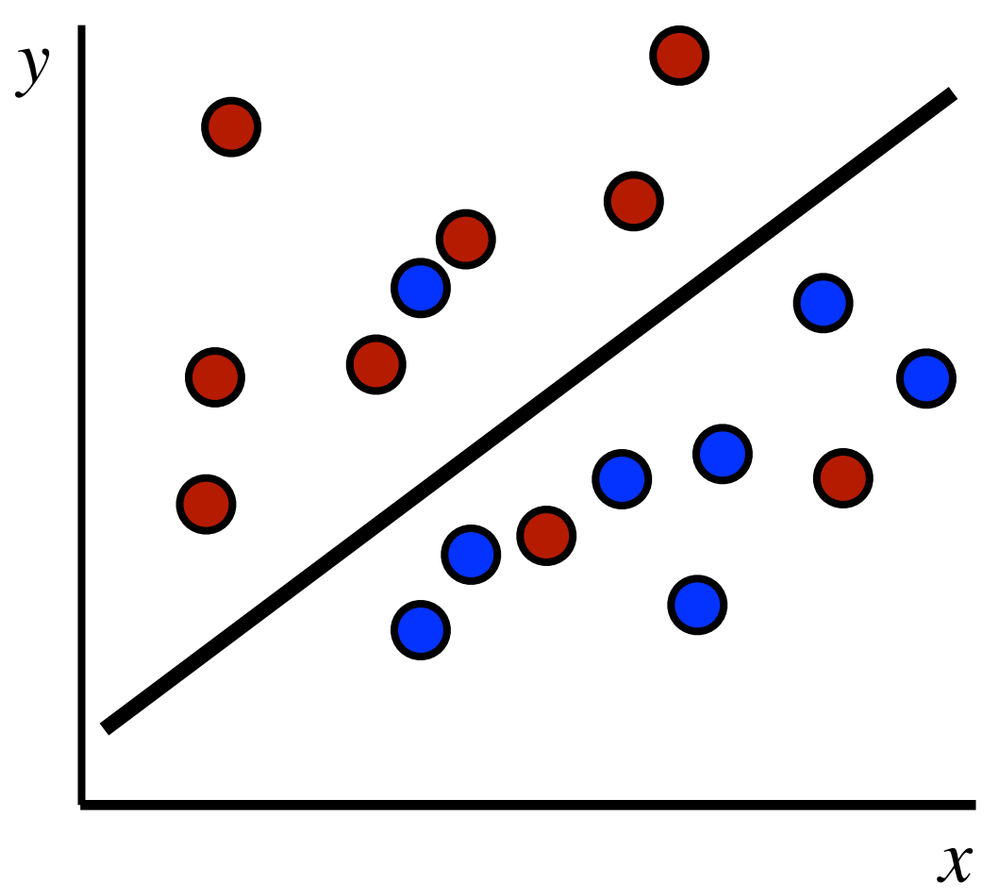
- Model Parameters: slope (w) and intercept (b)
- Nice probabilistic interpretation



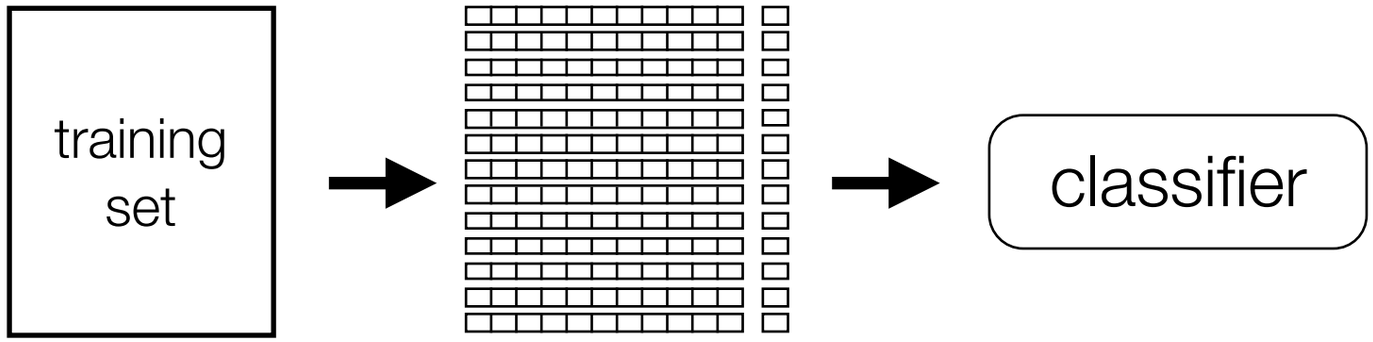
E.g., Logistic Regression

Goal: Find linear decision boundary

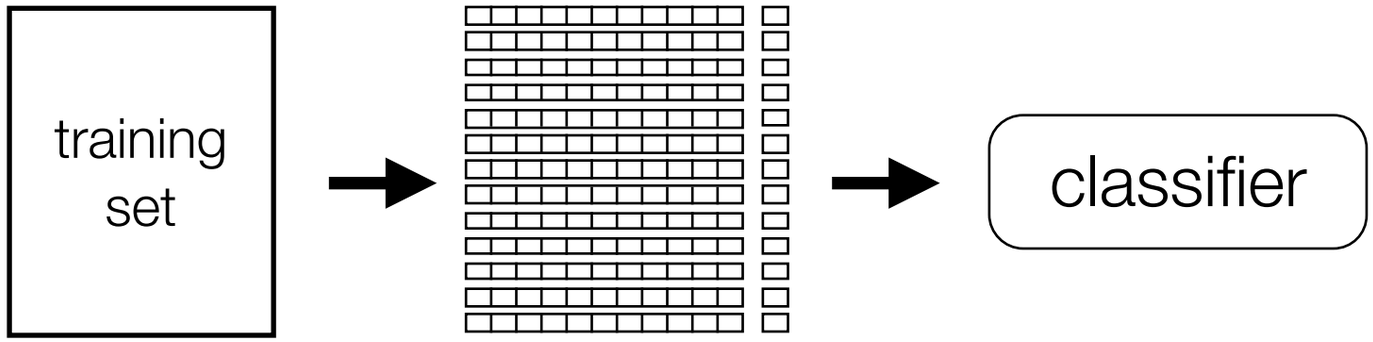
- Model Parameters: slope (w) and intercept (b)
- Nice probabilistic interpretation
- Covered in more detail tomorrow



Classification Pipeline

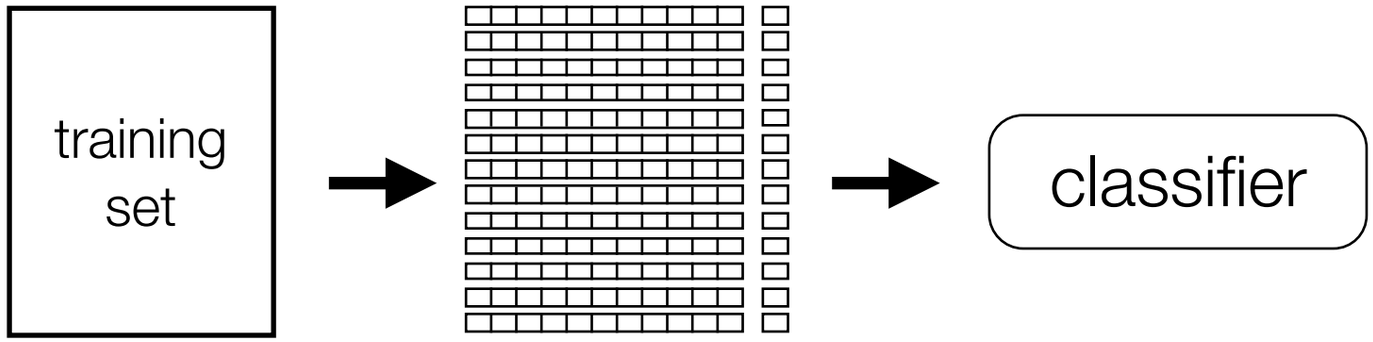


Classification Pipeline



How can we evaluate the quality of our classifier?

Classification Pipeline

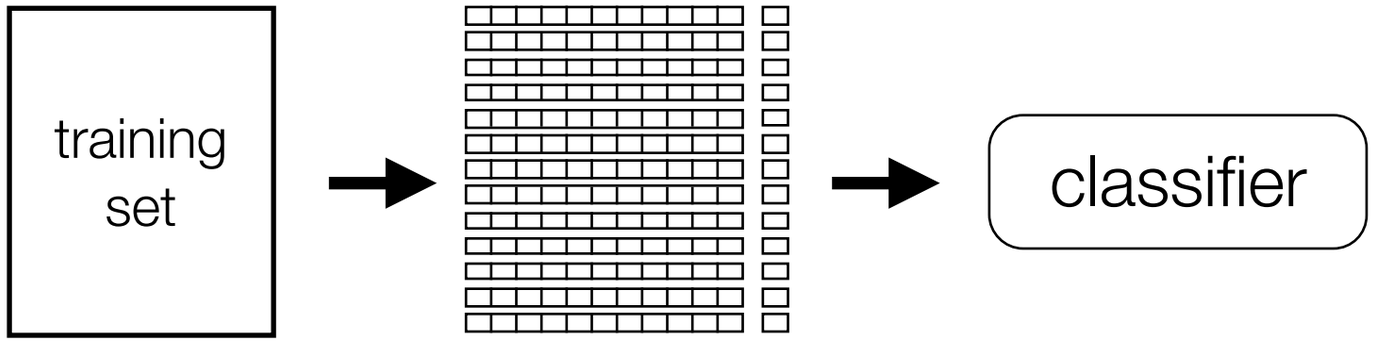


How can we evaluate the quality of our classifier?

We want good predictions on new / unobserved data

- ’Generalization’ ability

Classification Pipeline



How can we evaluate the quality of our classifier?

We want good predictions on new / unobserved data

- ’Generalization’ ability

Accuracy on training data is overly optimistic

- We might be ‘overfitting’

Overfitting and Generalization

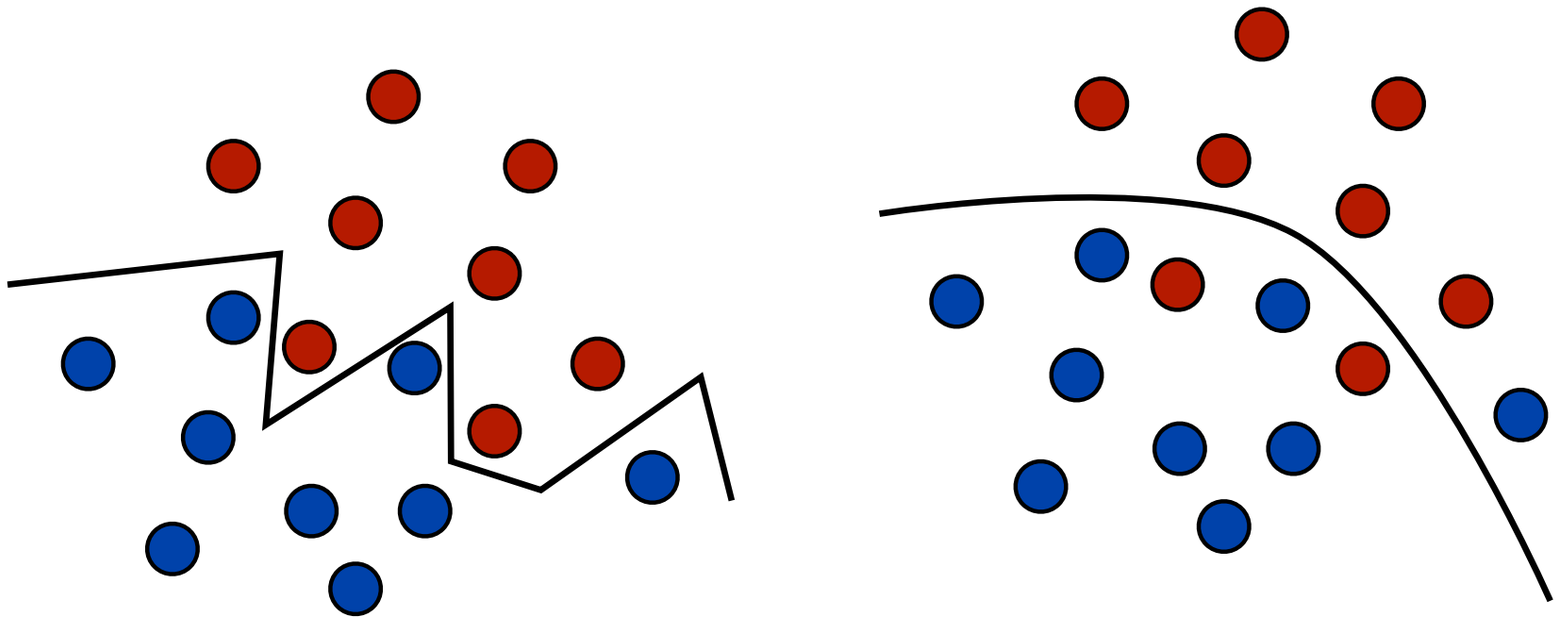


Image Credit: Foundations of Machine Learning, Mohri, Rostamizadeh, Talwalkar

Overfitting and Generalization

We want a model that makes good predictions on new data, i.e., 'generalization'

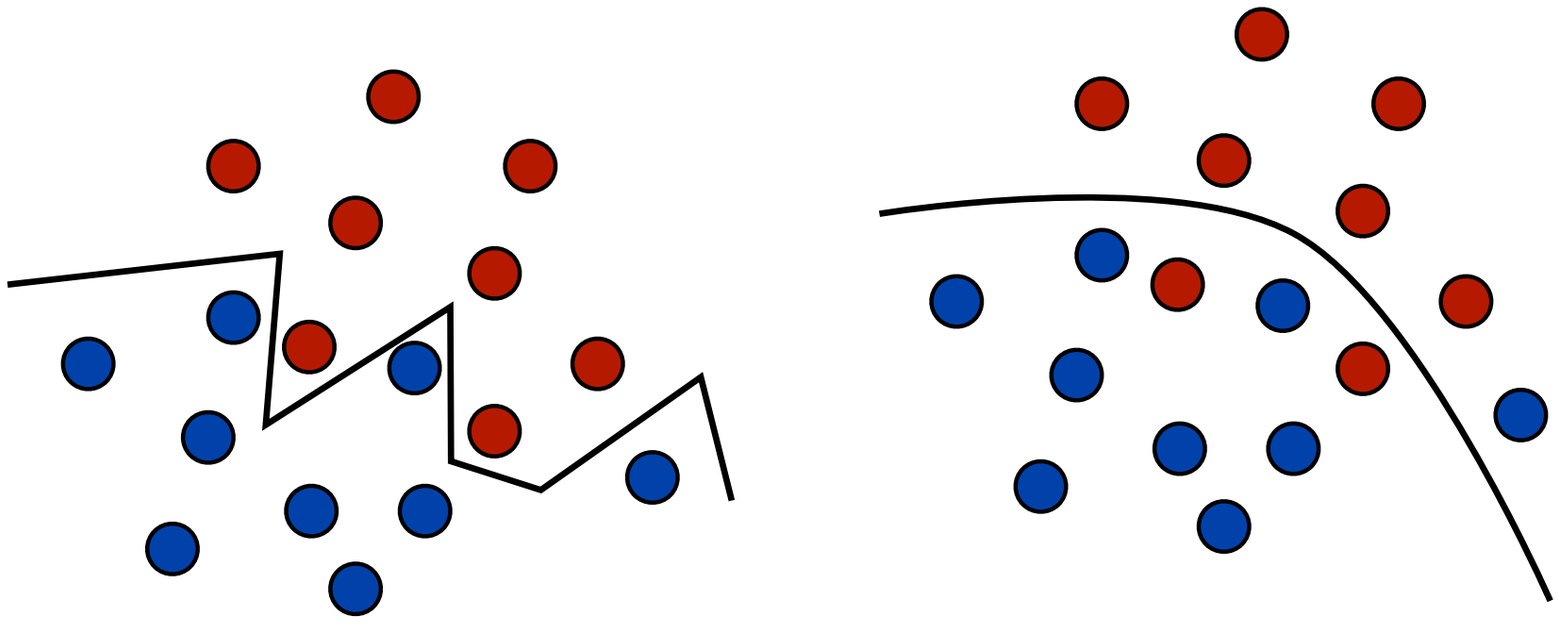


Image Credit: Foundations of Machine Learning, Mohri, Rostamizadeh, Talwalkar

Overfitting and Generalization

We want a model that makes good predictions on new data, i.e., ’generalization’

Fitting training data does not guarantee generalization

- e.g., lookup table
- More complex models are more likely to ‘overfit’
- Occam’s razor

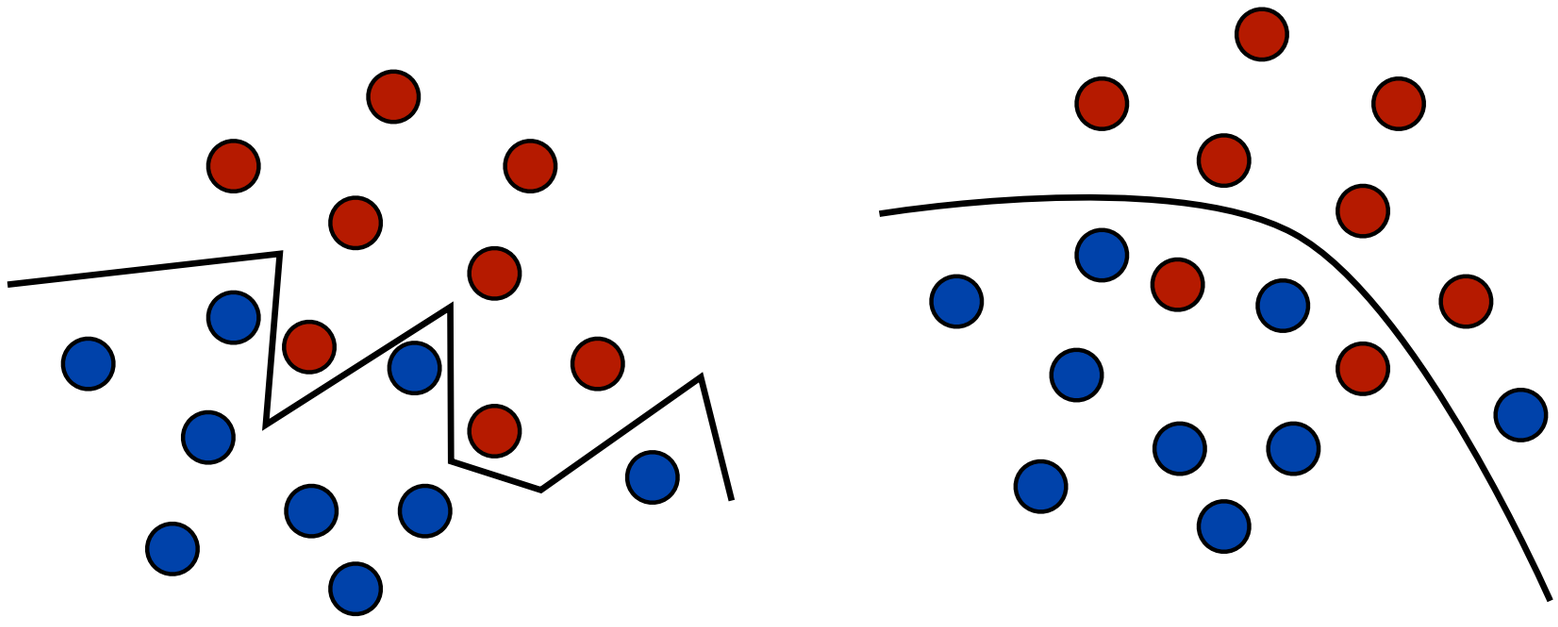
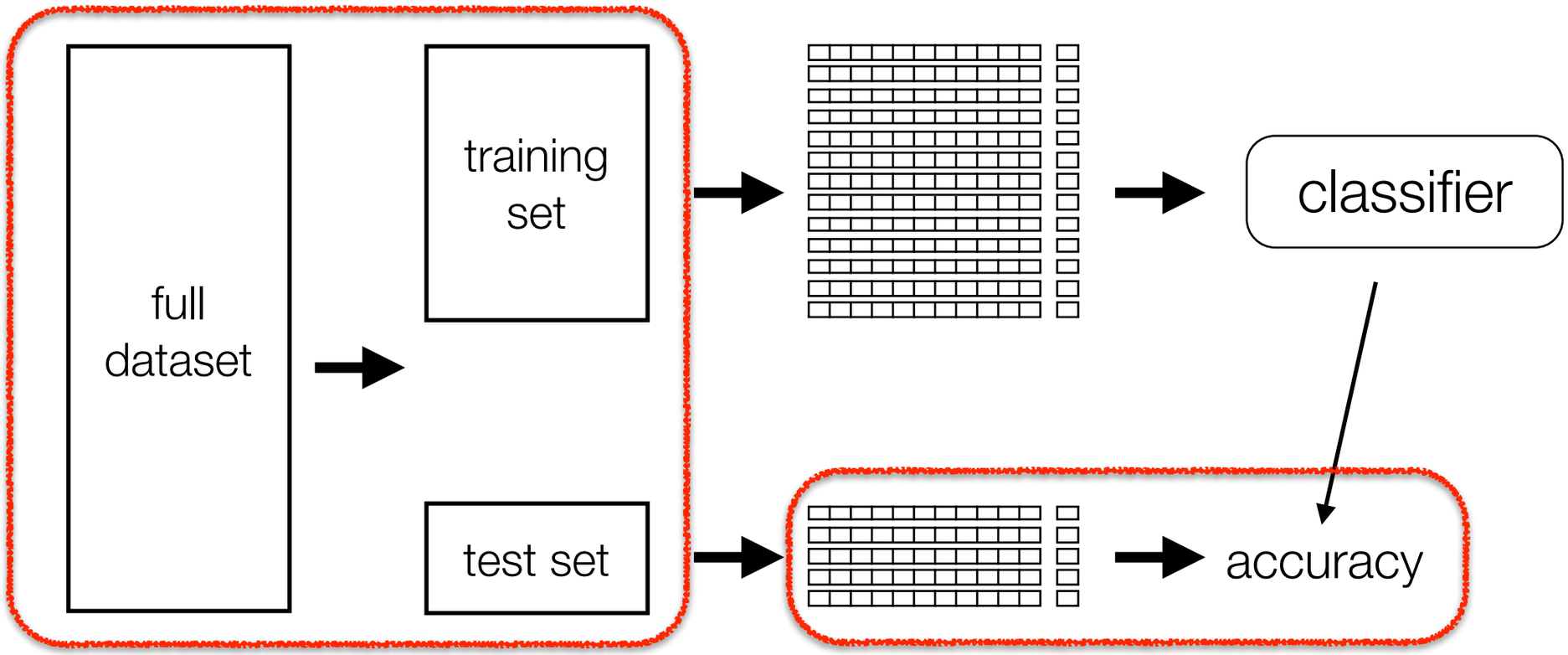
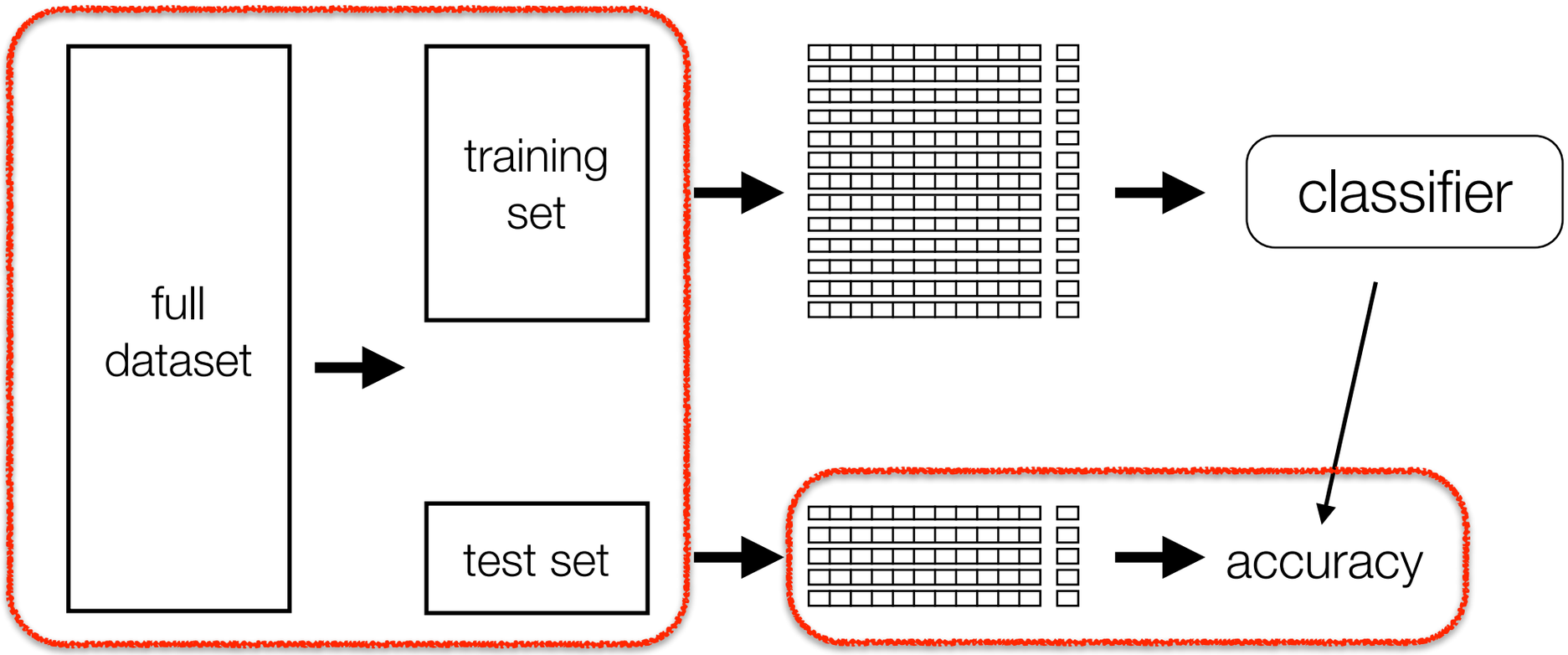


Image Credit: Foundations of Machine Learning, Mohri, Rostamizadeh, Talwalkar

Classification Pipeline



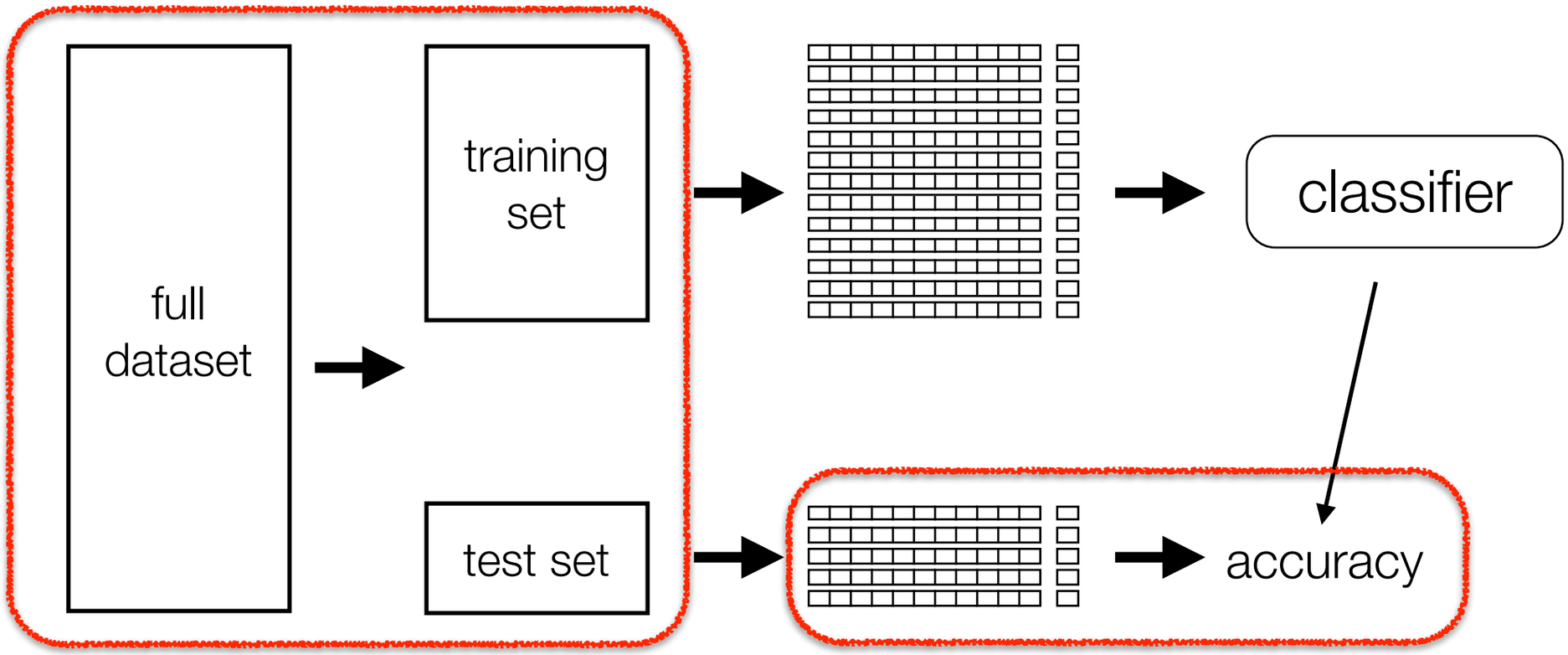
Classification Pipeline



Idea: Split dataset into training / testing datasets

- Test set simulates unobserved data

Classification Pipeline

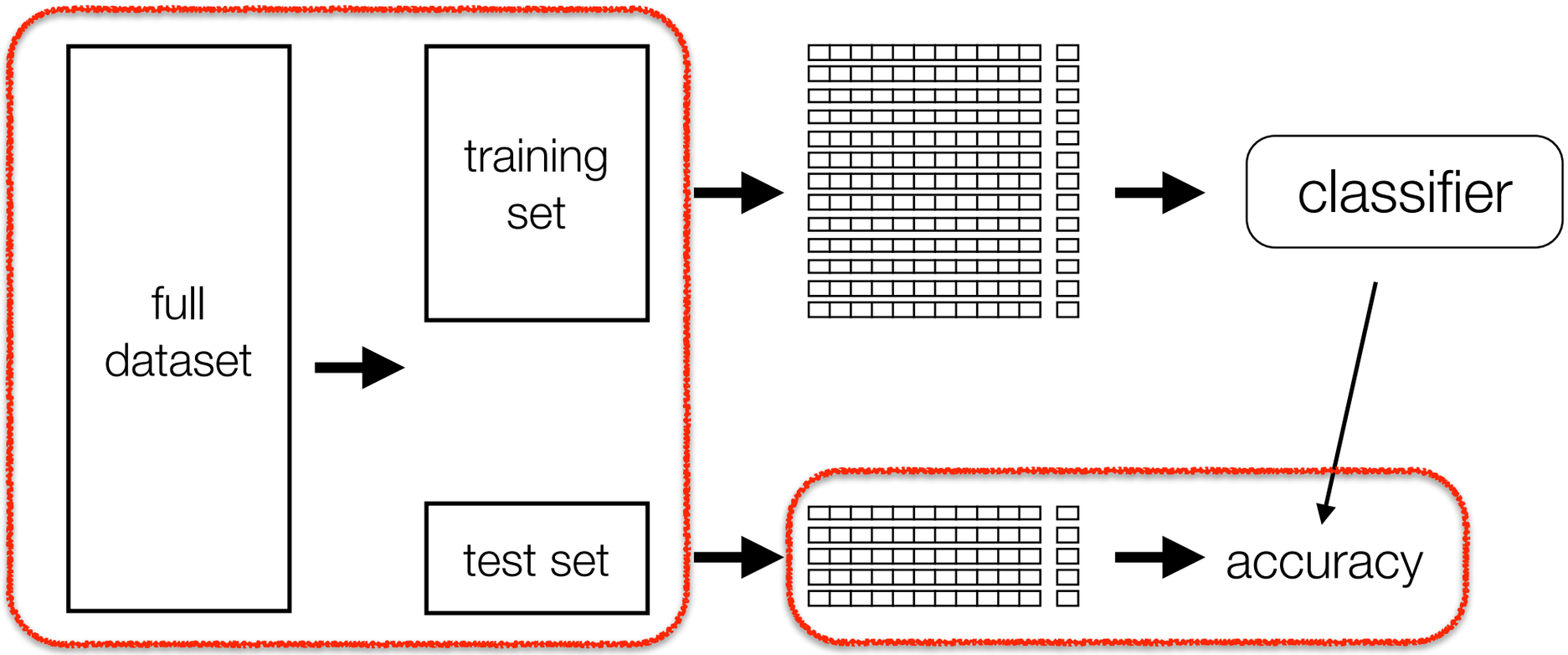


Idea: Split dataset into training / testing datasets

- Test set simulates unobserved data

Evaluation process

Classification Pipeline



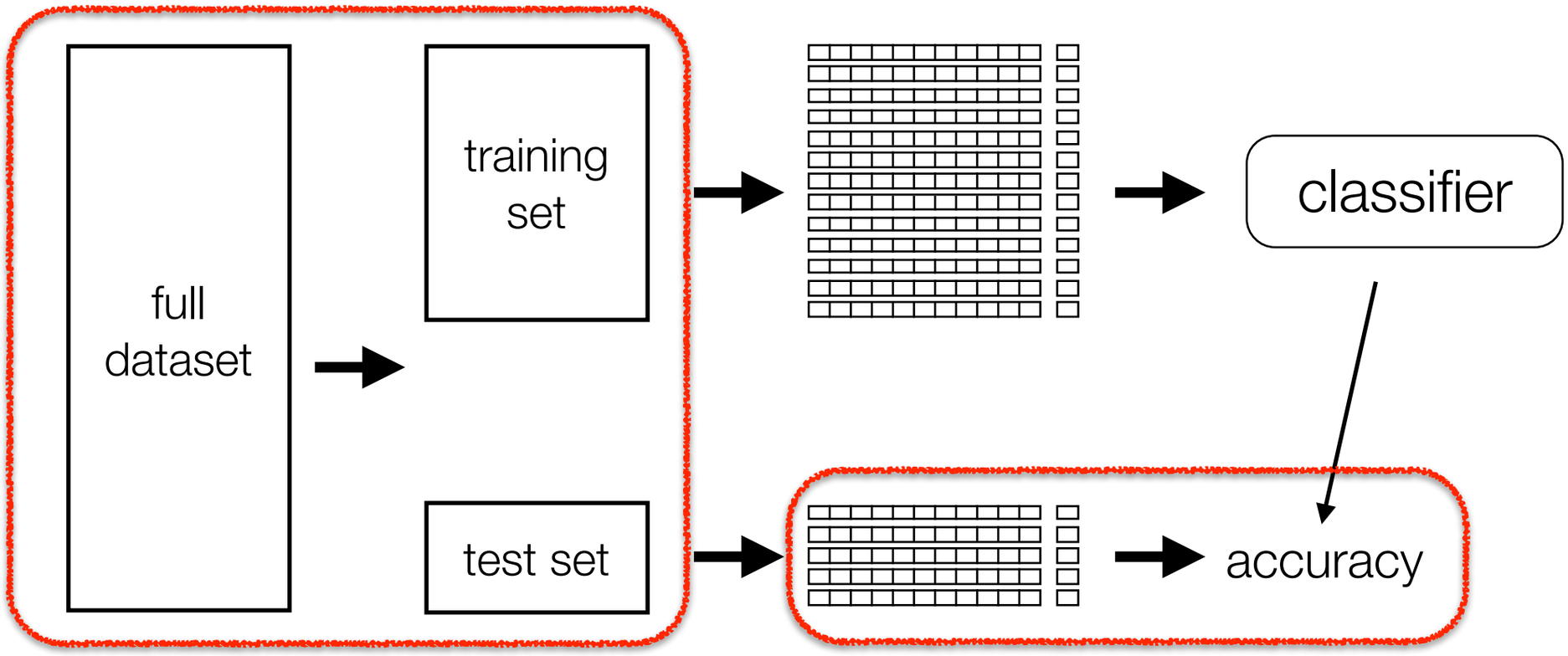
Idea: Split dataset into training / testing datasets

- Test set simulates unobserved data

Evaluation process

- Train on training set (don't expose test set to classifier)

Classification Pipeline



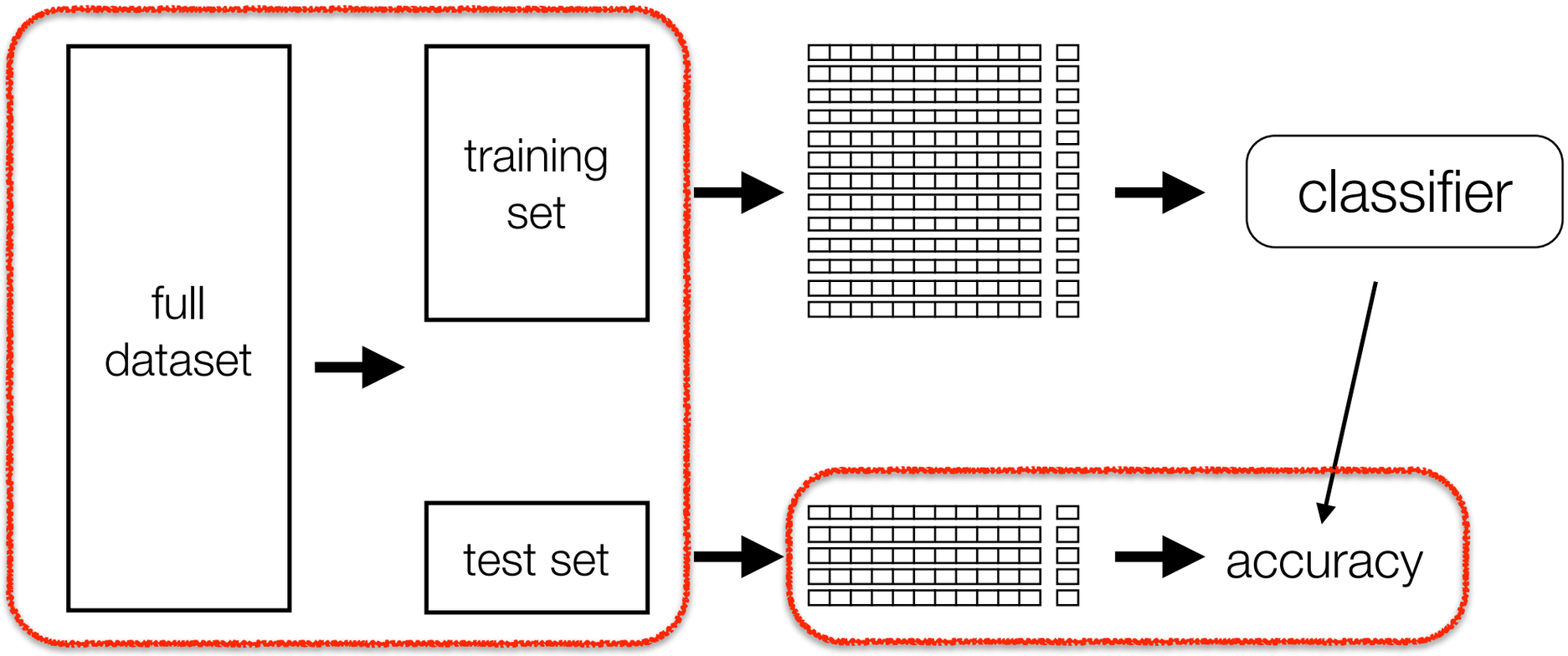
Idea: Split dataset into training / testing datasets

- Test set simulates unobserved data

Evaluation process

- Train on training set (don't expose test set to classifier)
- Make predictions using test set (ignoring test labels)

Classification Pipeline



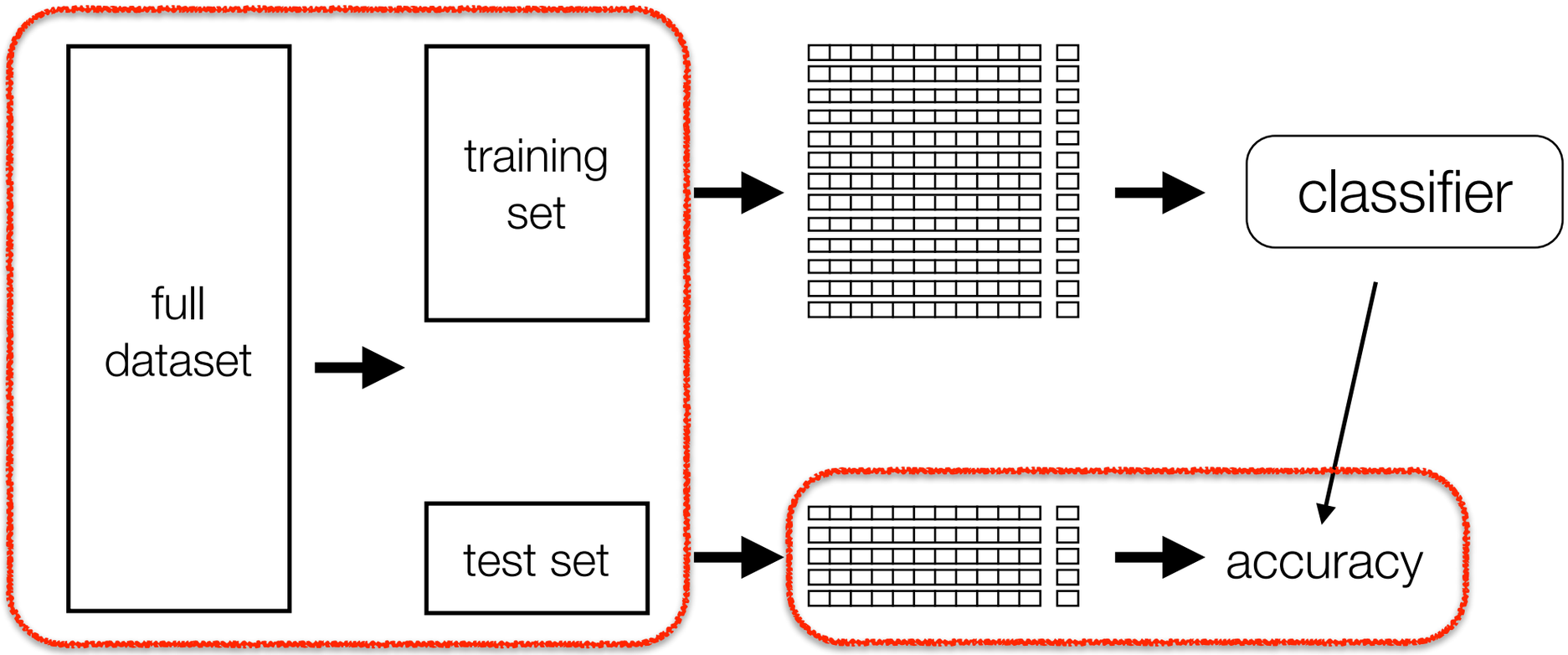
Idea: Split dataset into training / testing datasets

- Test set simulates unobserved data

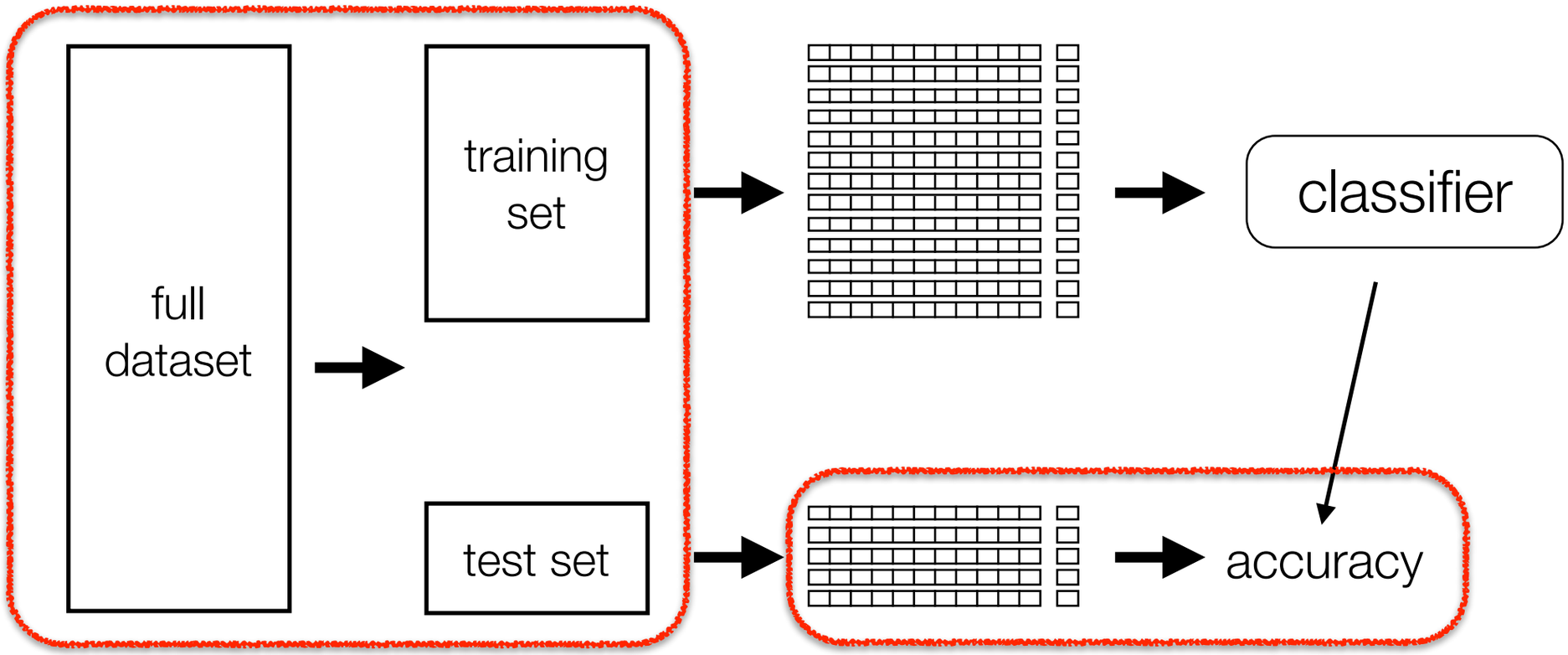
Evaluation process

- Train on training set (don't expose test set to classifier)
- Make predictions using test set (ignoring test labels)
- Compute fraction of correct predictions on test set

Classification Pipeline

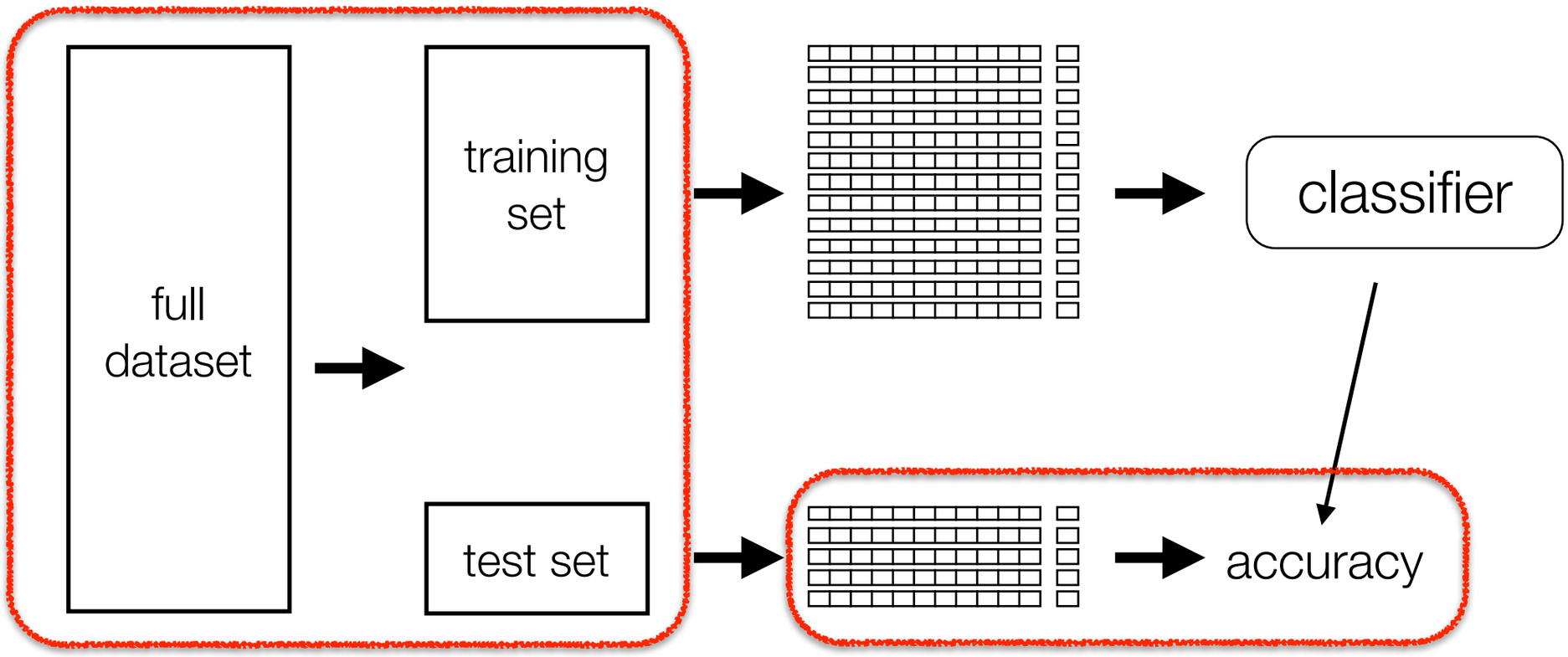


Classification Pipeline



Cross-validation is another common evaluation method

Classification Pipeline

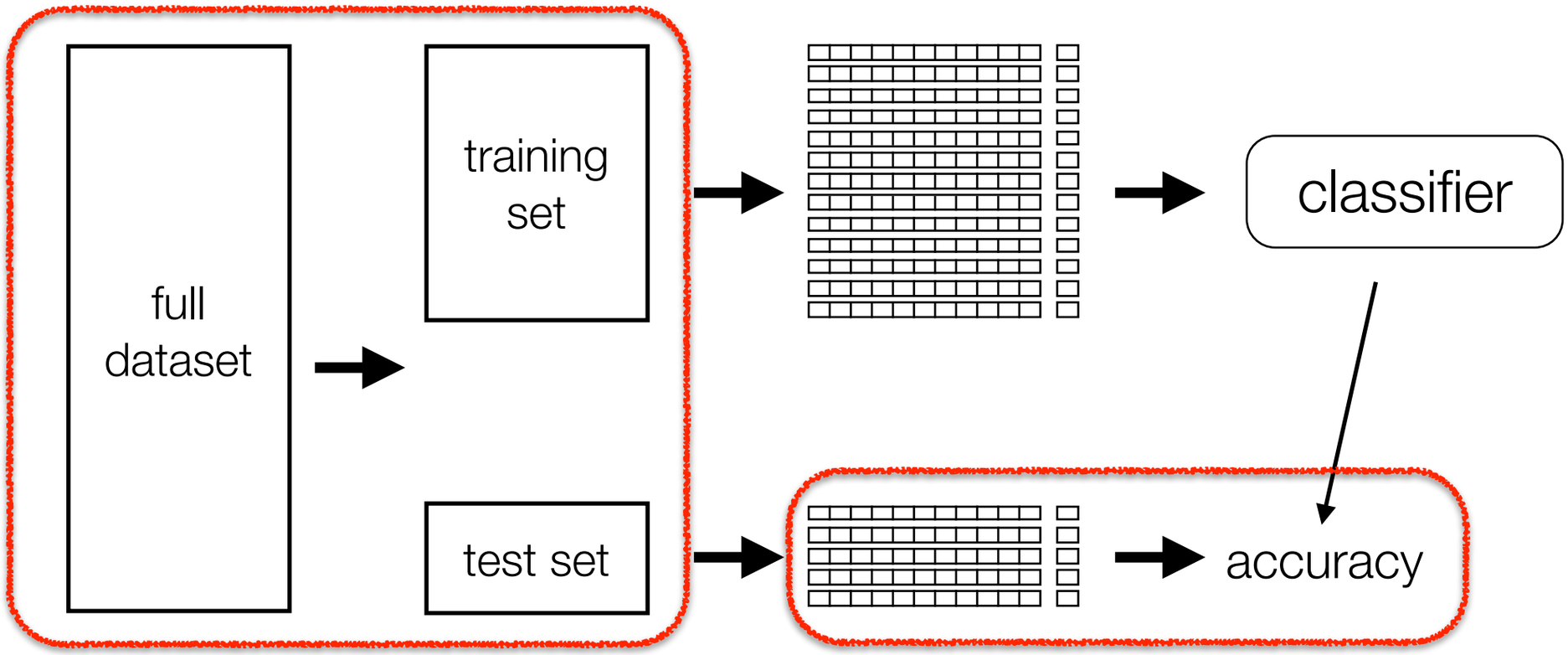


Cross-validation is another common evaluation method

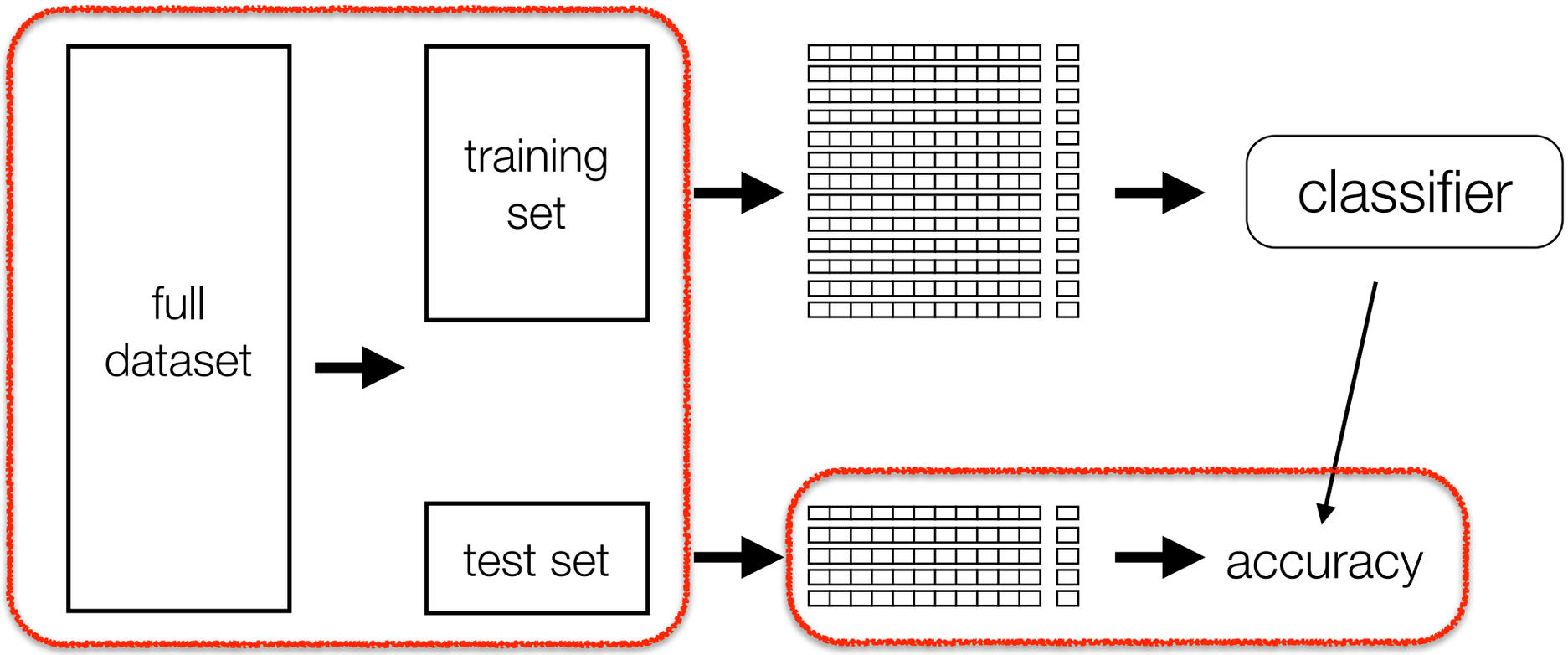
Various metrics for quality, including 0-1 accuracy

- In CTR exercise we'll use logloss instead

Classification Pipeline



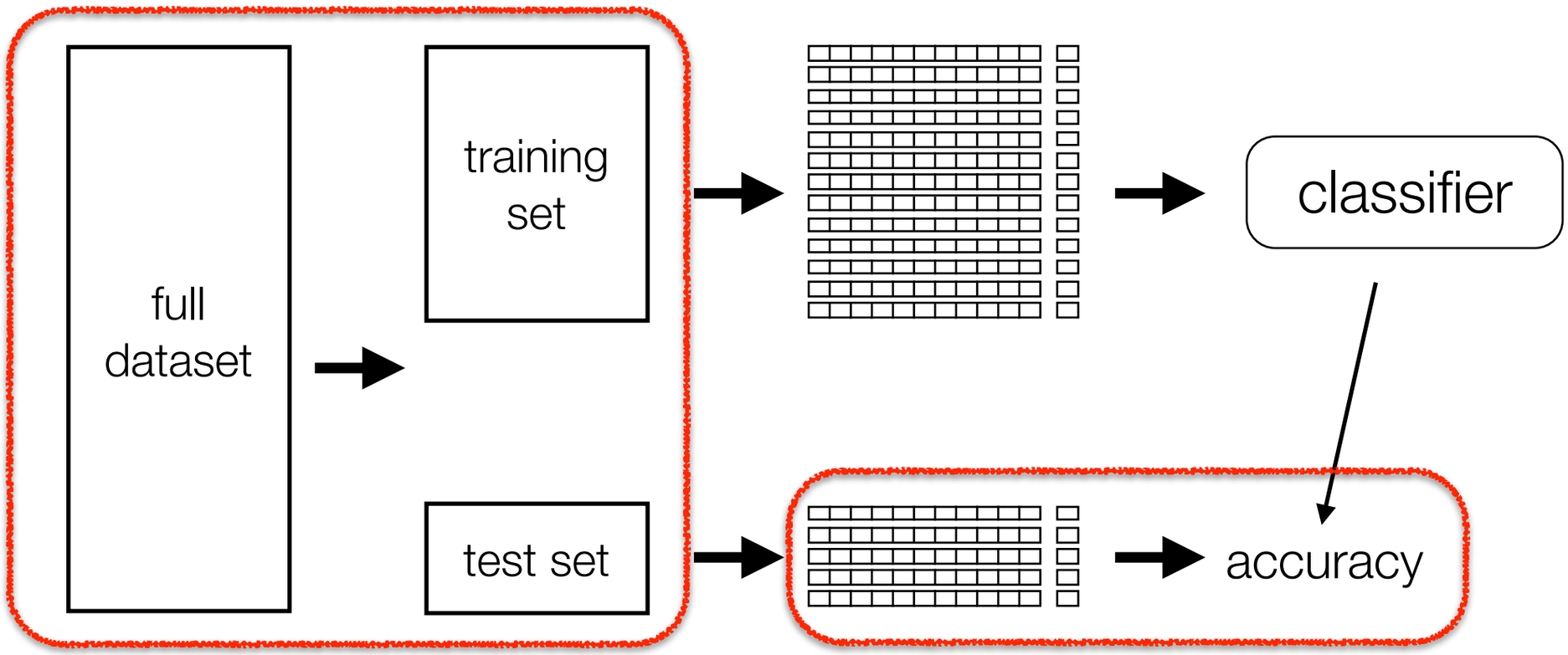
Classification Pipeline



Logistic regression (and most other methods) have free parameters or ‘hyperparameters’ to tune

- e.g., ‘regularizer’ to be discussed later

Classification Pipeline

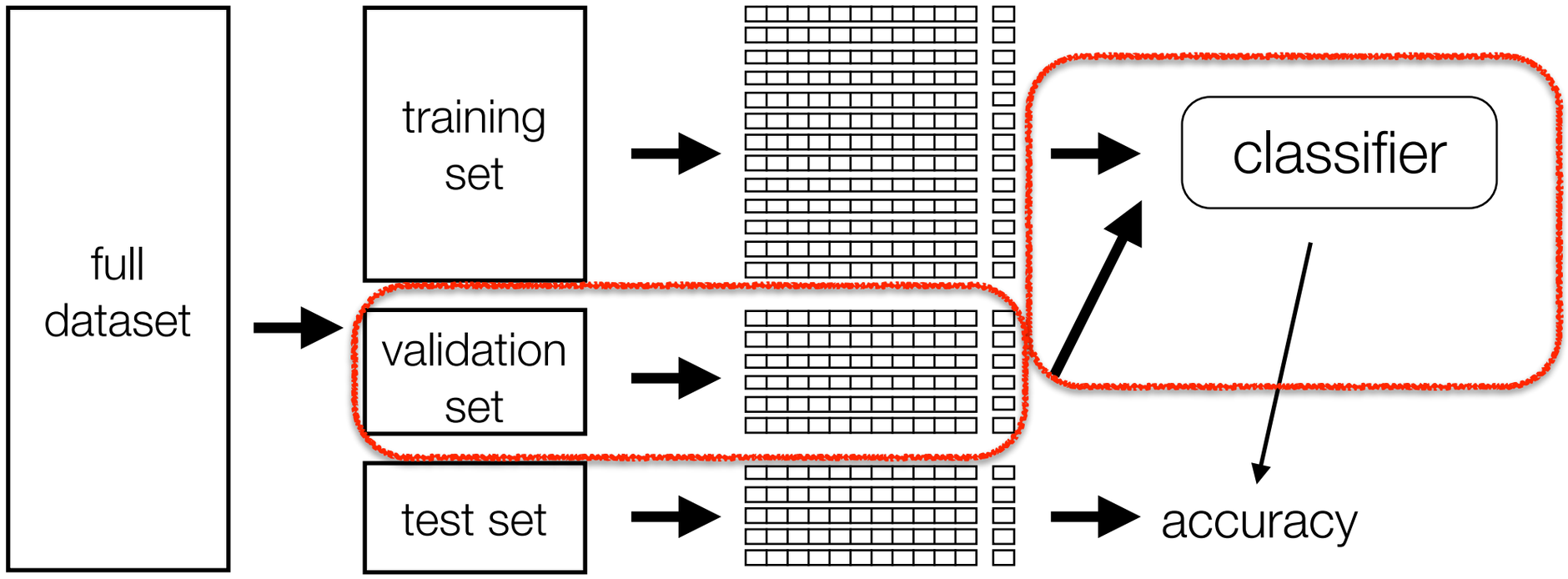


Logistic regression (and most other methods) have free parameters or ‘hyperparameters’ to tune

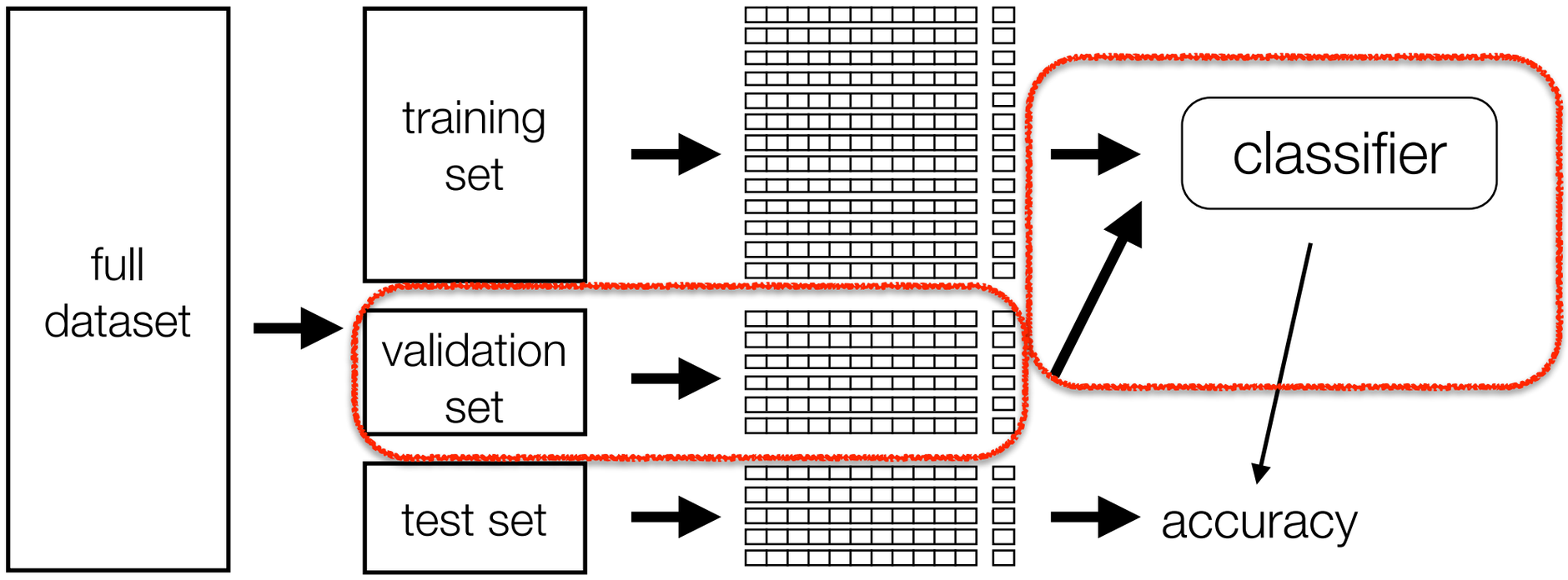
- e.g., ‘regularizer’ to be discussed later

Tuning hyperparameters adds a wrinkle to evaluation, as we don’t want to overfit to the test set

Classification Pipeline

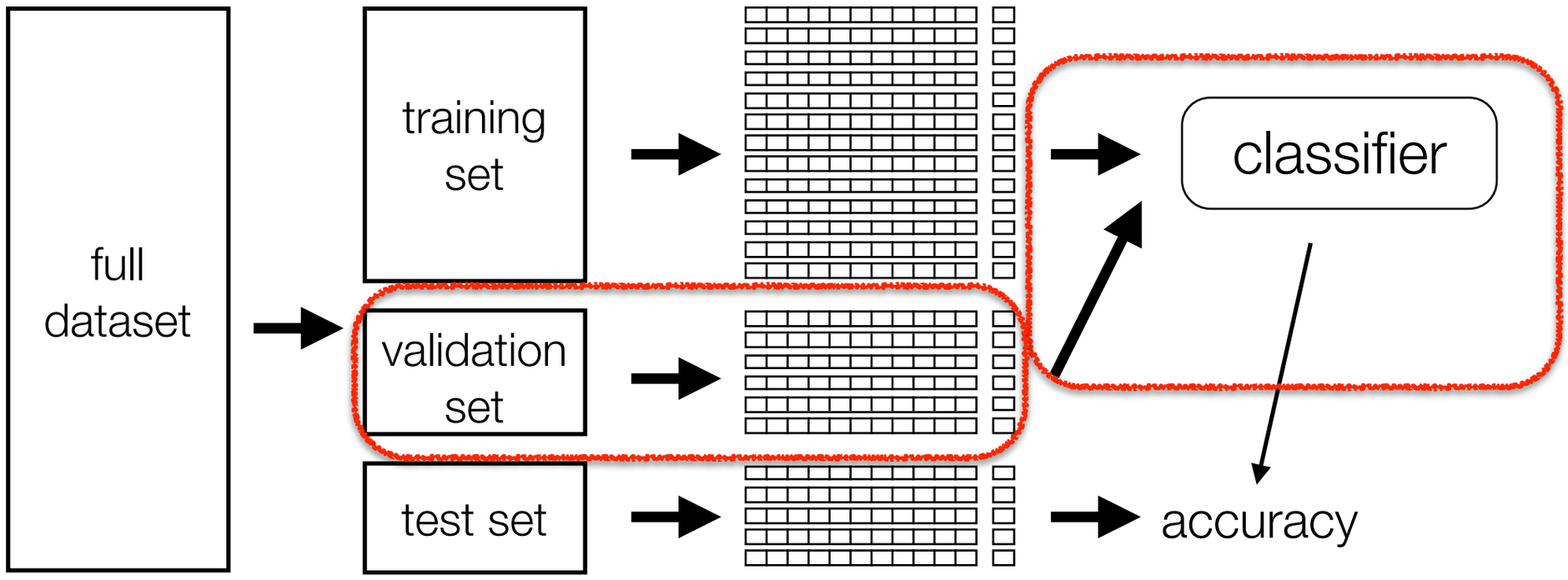


Classification Pipeline



Idea: Create a 3rd dataset to tune hyperparameters

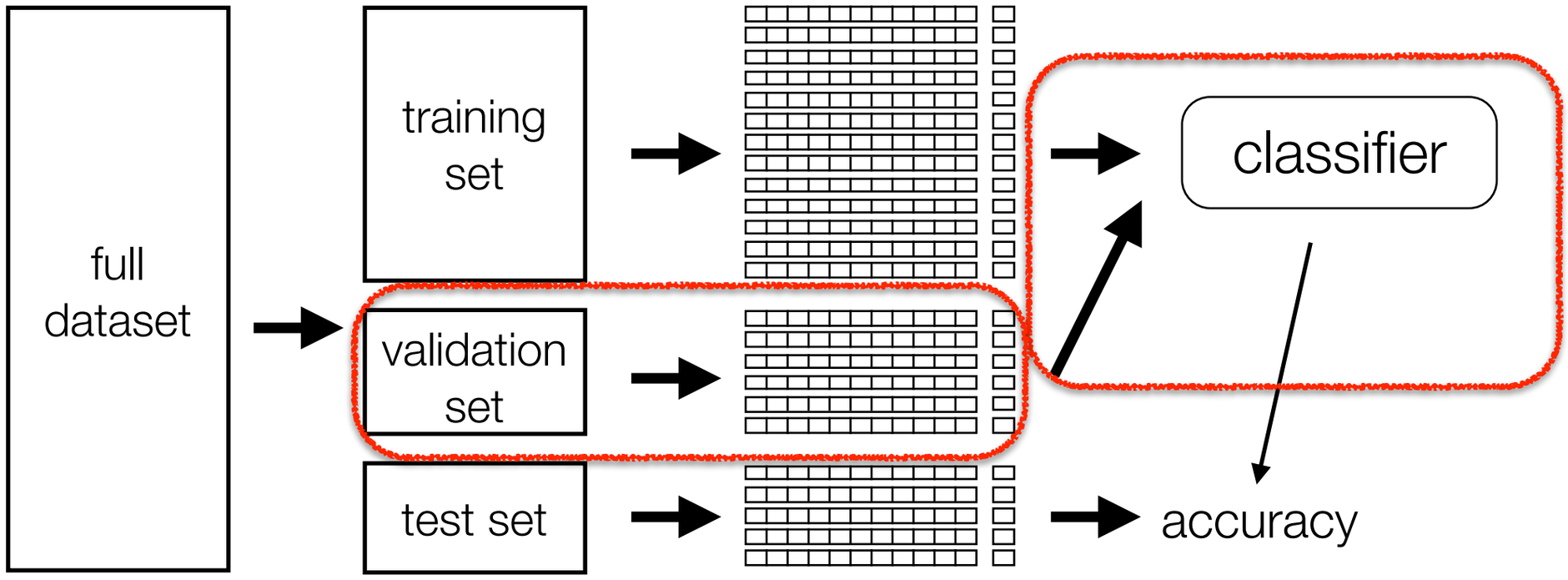
Classification Pipeline



Idea: Create a 3rd dataset to tune hyperparameters

- Training set used to train various classifiers

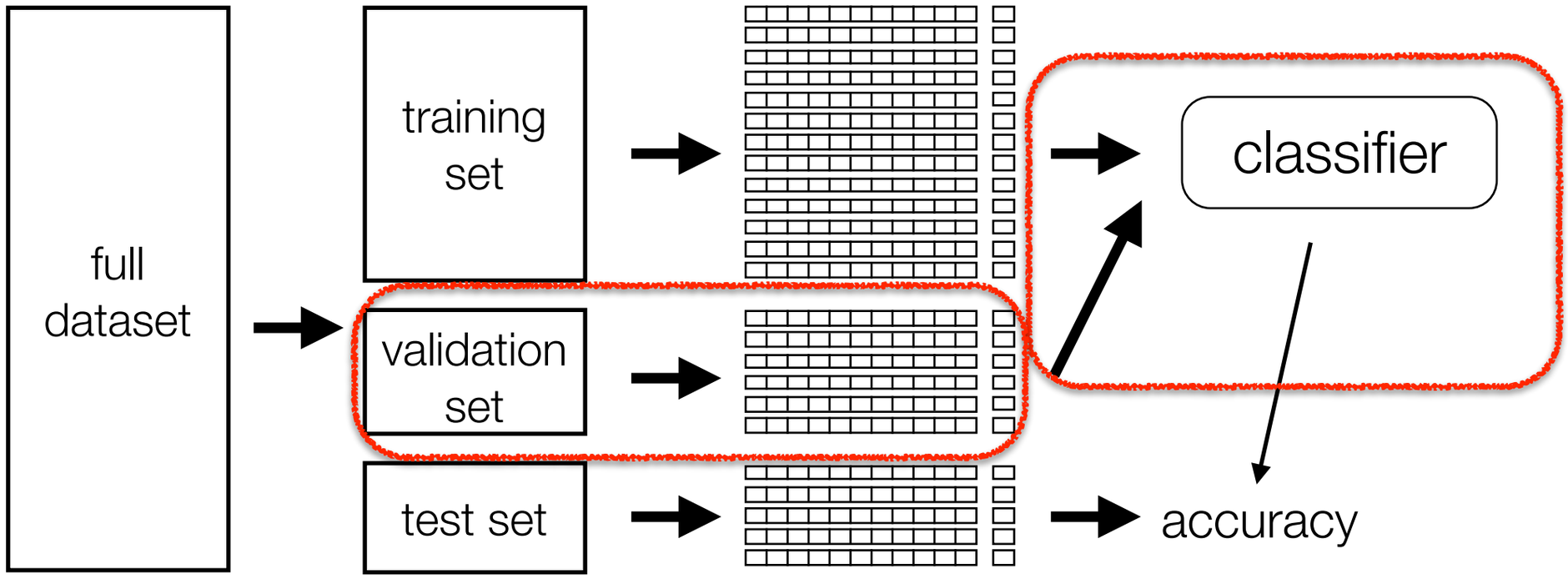
Classification Pipeline



Idea: Create a 3rd dataset to tune hyperparameters

- Training set used to train various classifiers
- Validation set used to find best hyperparameters

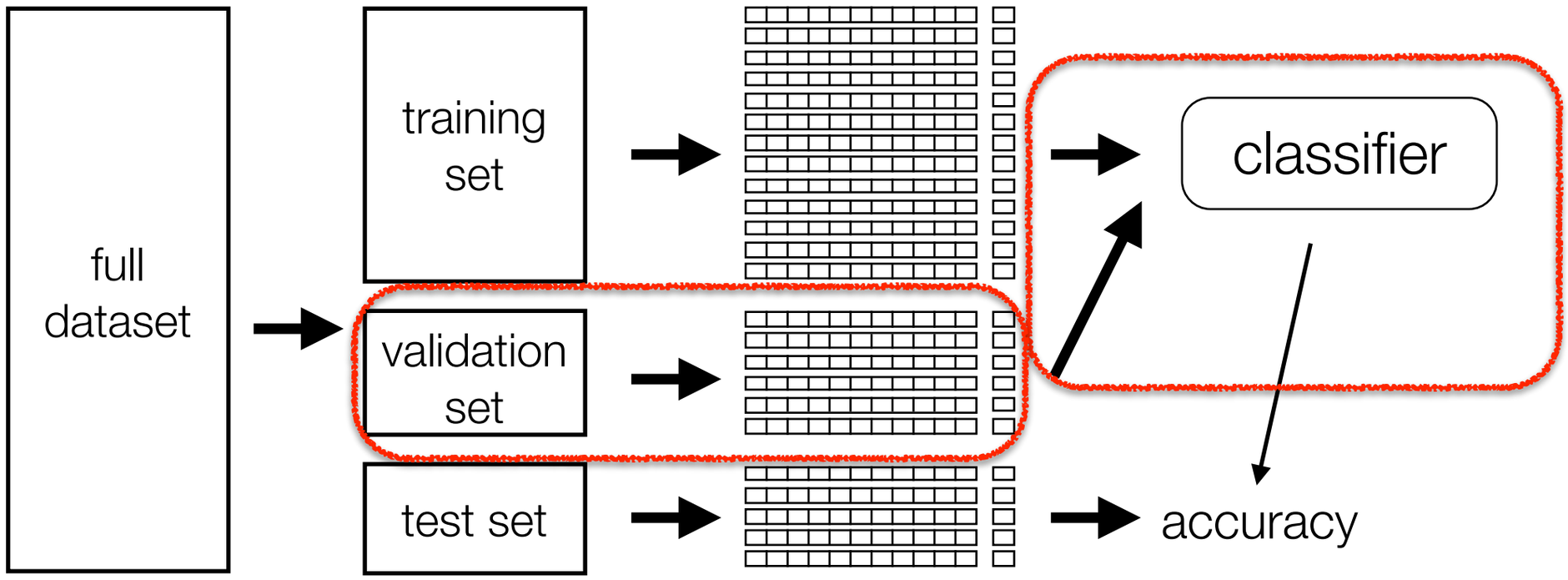
Classification Pipeline



Idea: Create a 3rd dataset to tune hyperparameters

- Training set used to train various classifiers
- Validation set used to find best hyperparameters
- Test set used to evaluate final model's accuracy

Classification Pipeline



Idea: Create a 3rd dataset to tune hyperparameters

- Training set used to train various classifiers
- Validation set used to find best hyperparameters
- Test set used to evaluate final model's accuracy

Hyperparameter values often selected via Grid Search

Grid Search

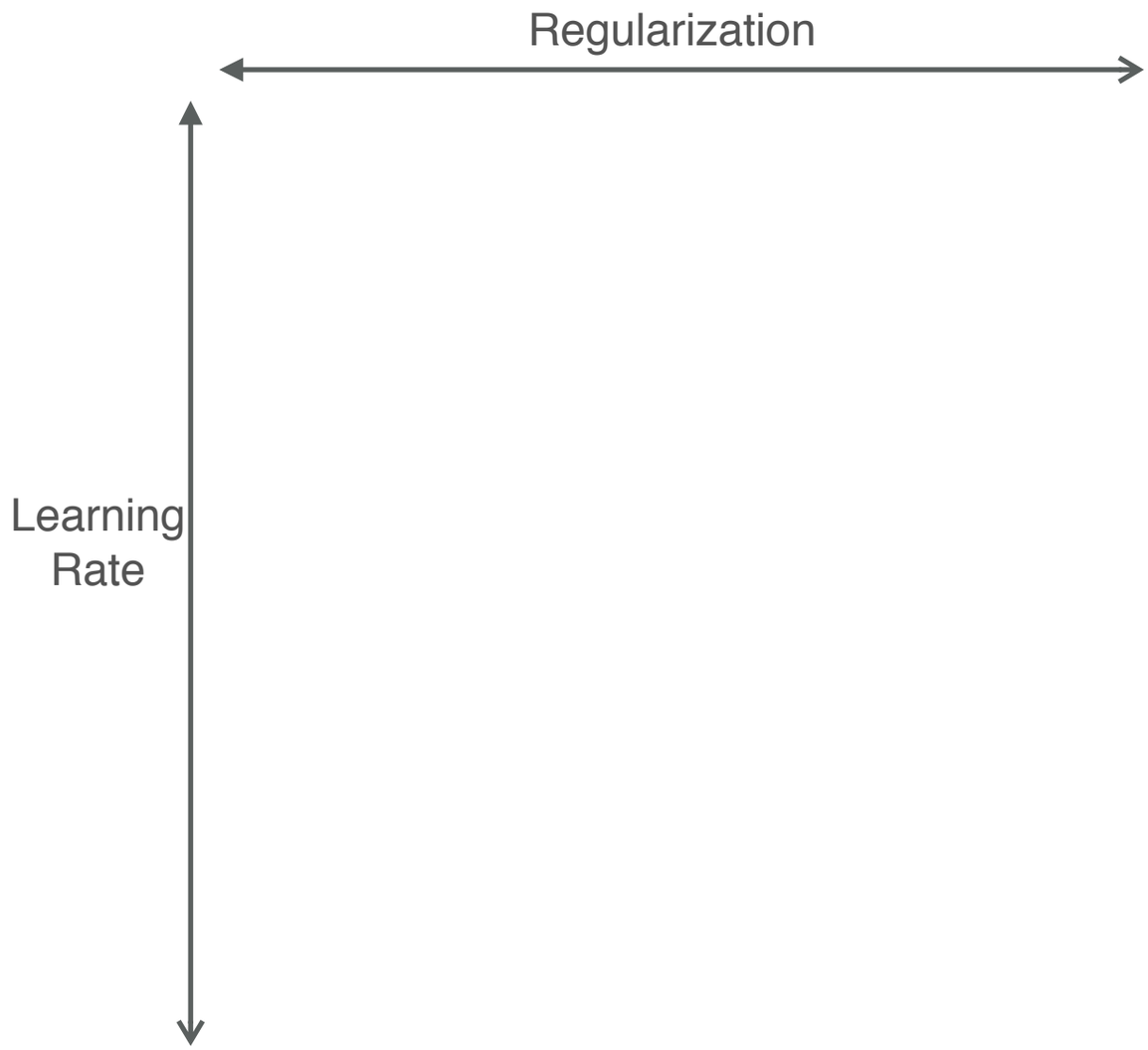
Grid Search

Exhaustively search through
hyper parameter space

Grid Search

Exhaustively search through hyper parameter space

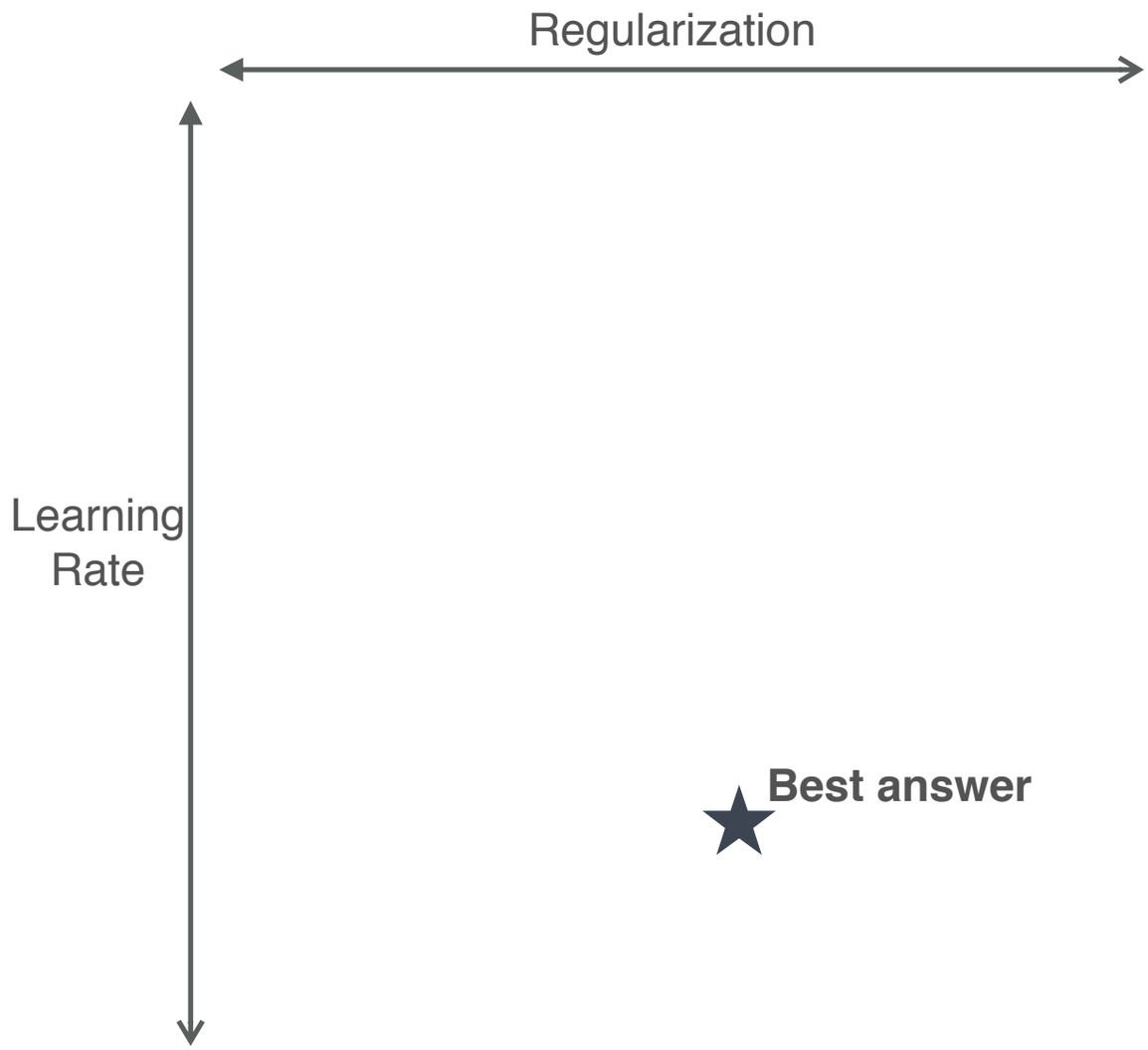
- Define space



Grid Search

Exhaustively search through hyper parameter space

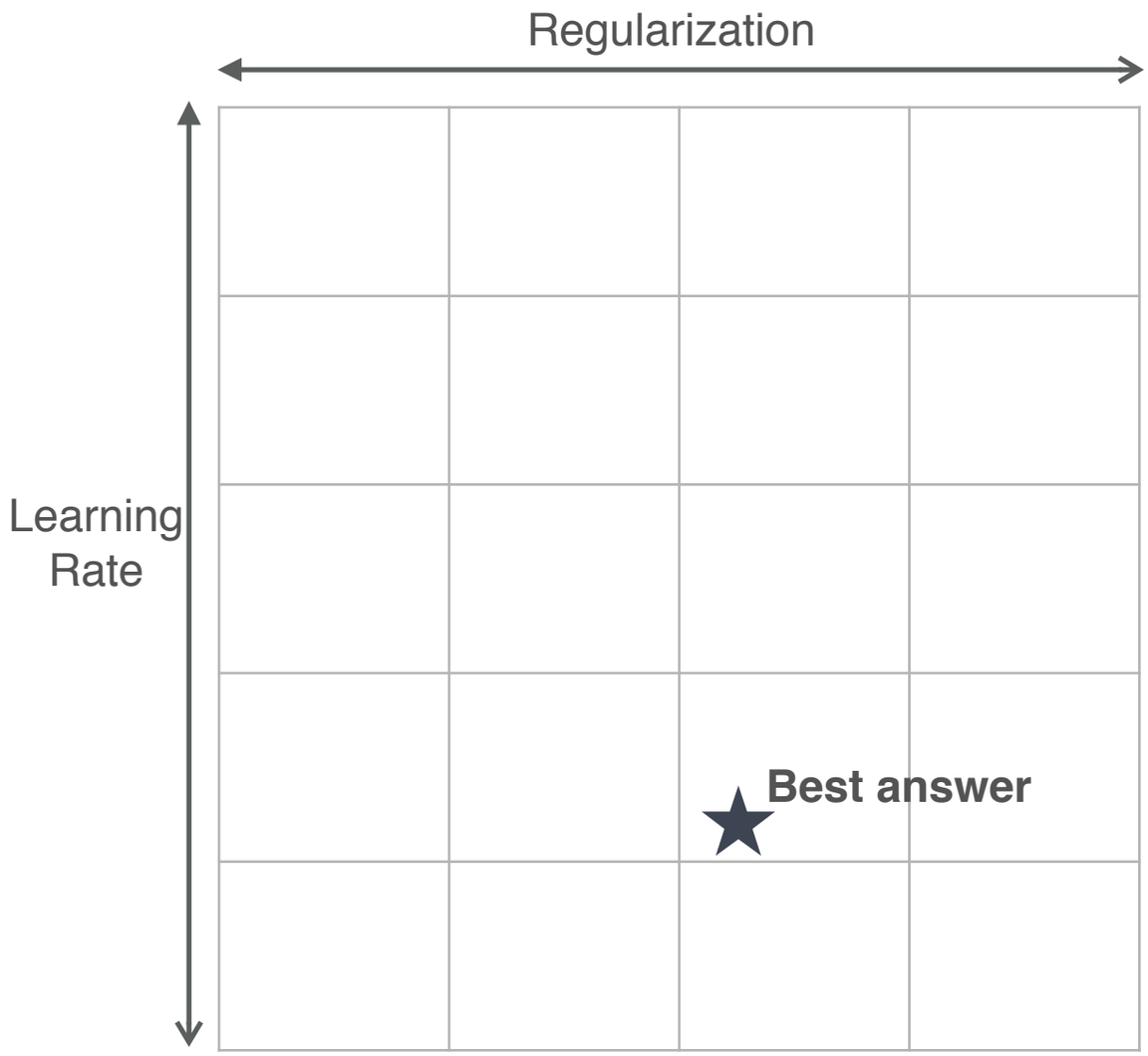
- Define space



Grid Search

Exhaustively search through hyper parameter space

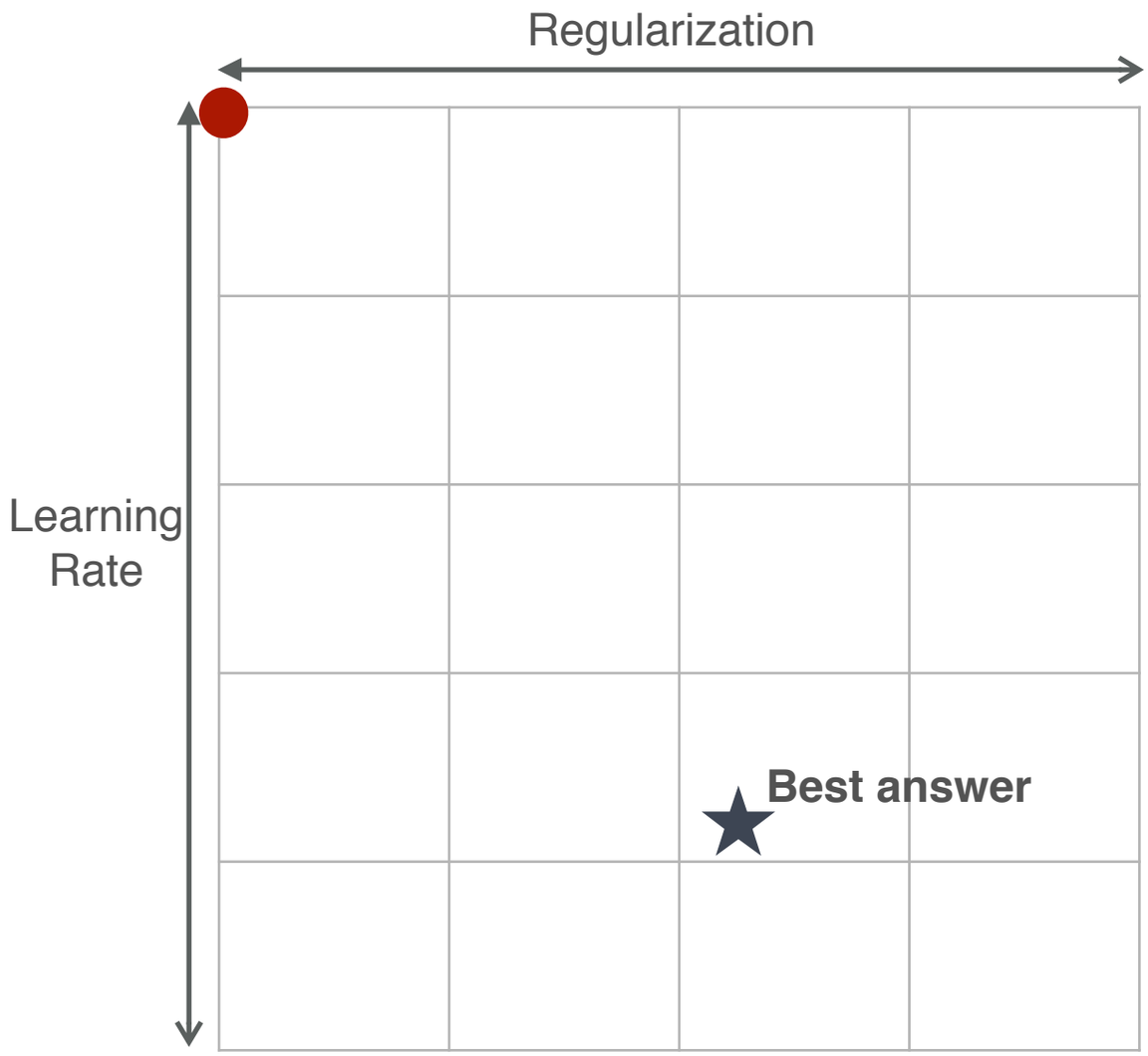
- Define space
- Discretize space



Grid Search

Exhaustively search through hyper parameter space

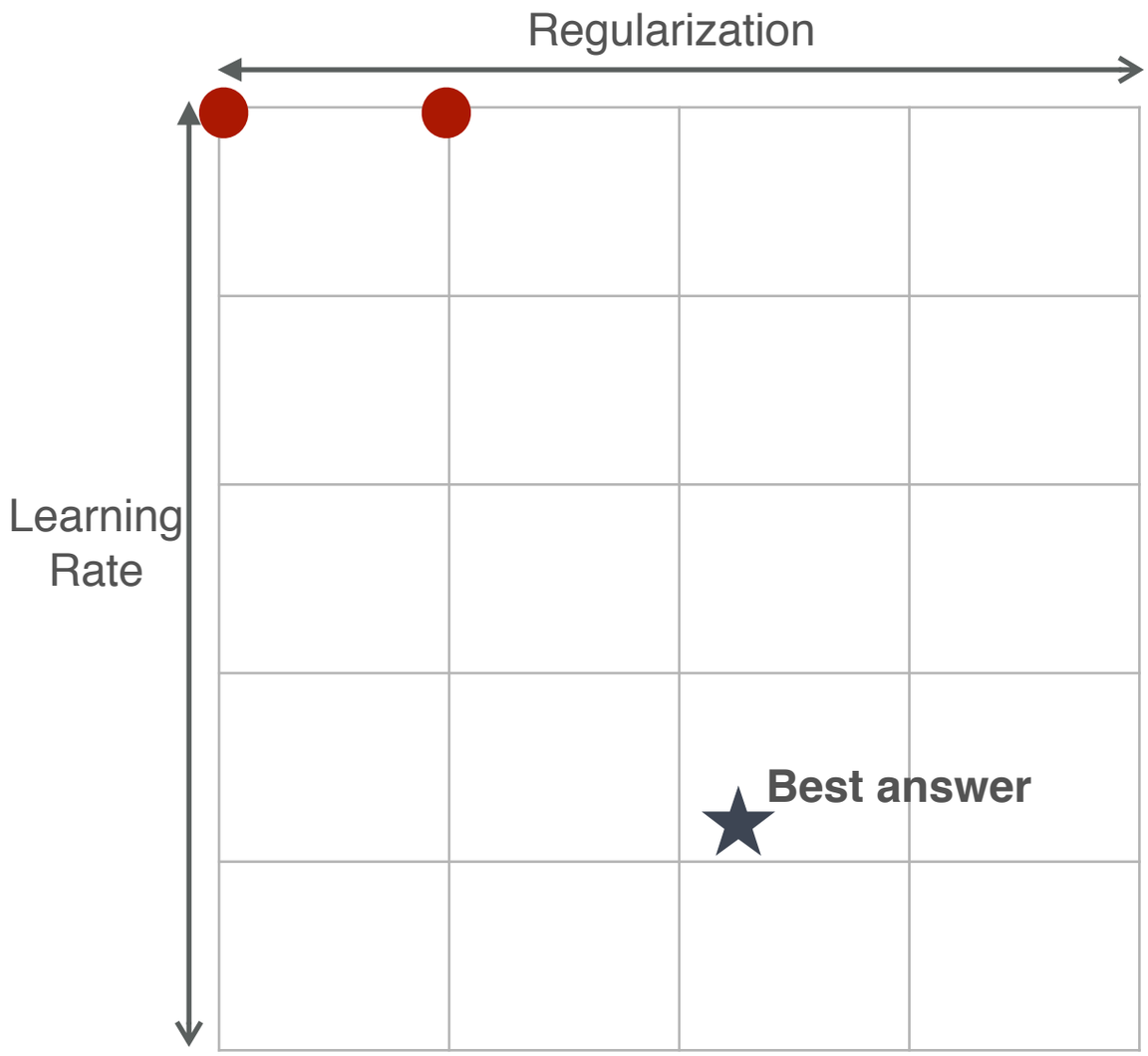
- Define space
- Discretize space
- Evaluate points (via validation error)



Grid Search

Exhaustively search through hyper parameter space

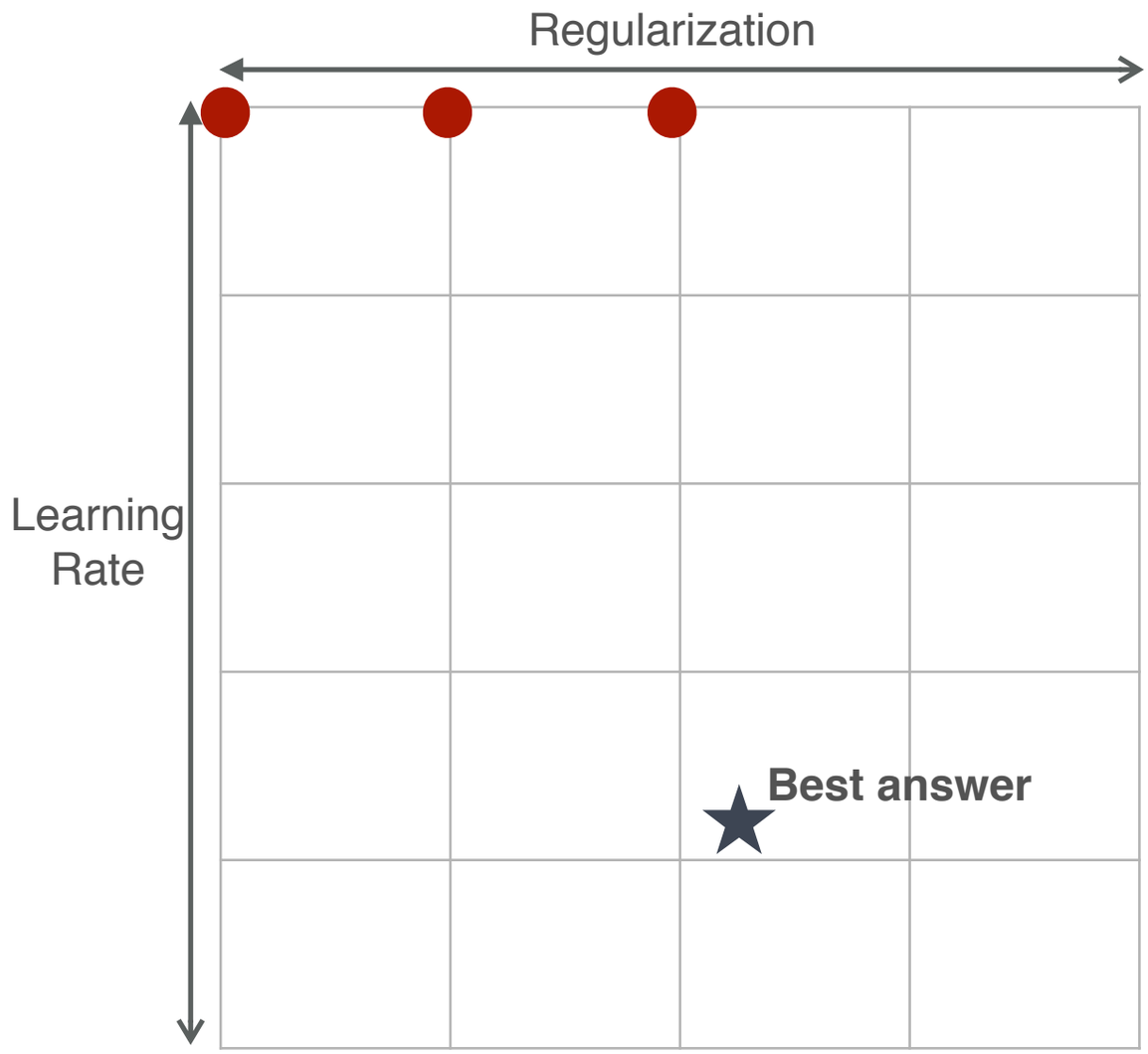
- Define space
- Discretize space
- Evaluate points (via validation error)



Grid Search

Exhaustively search through hyper parameter space

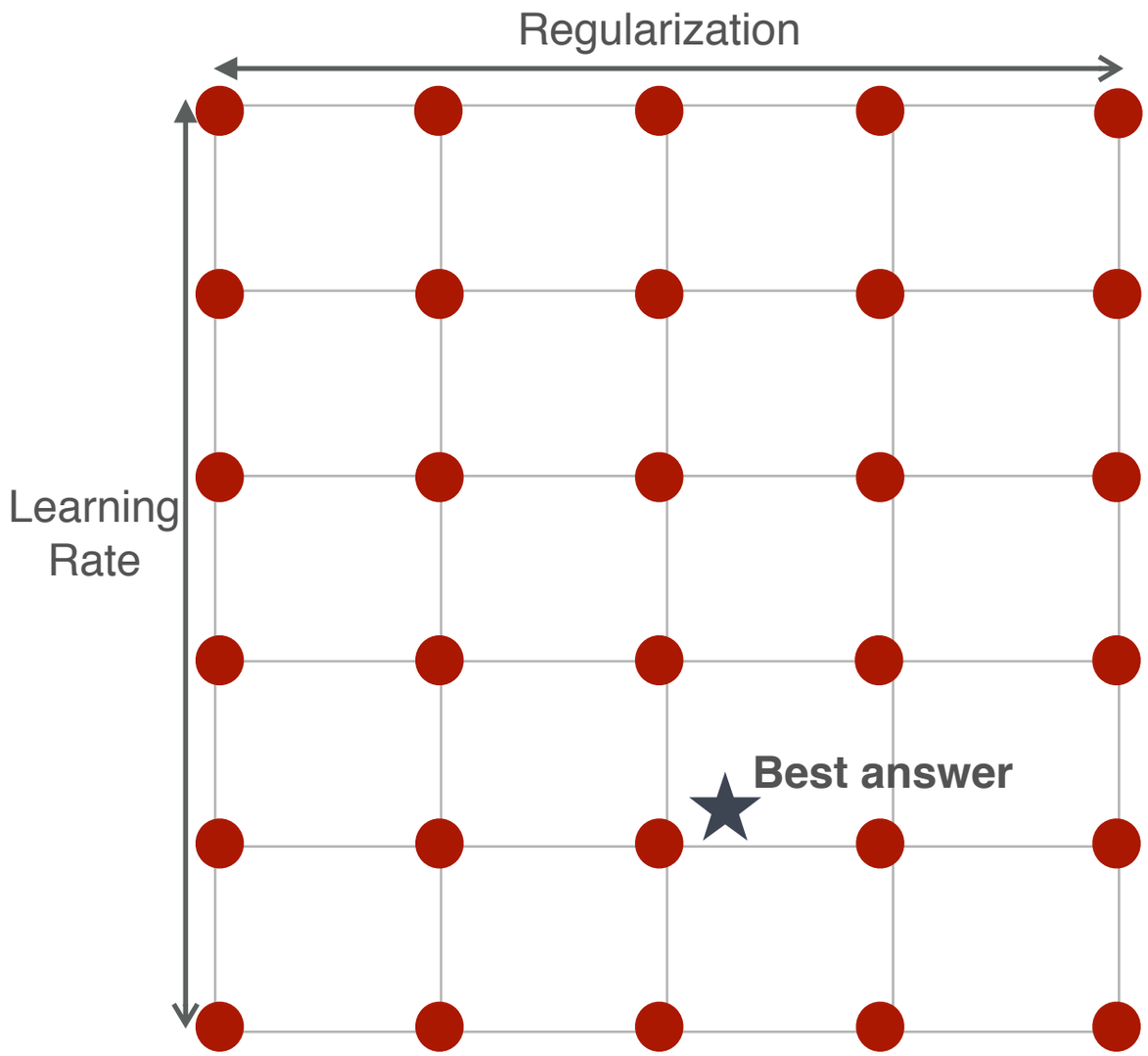
- Define space
- Discretize space
- Evaluate points (via validation error)



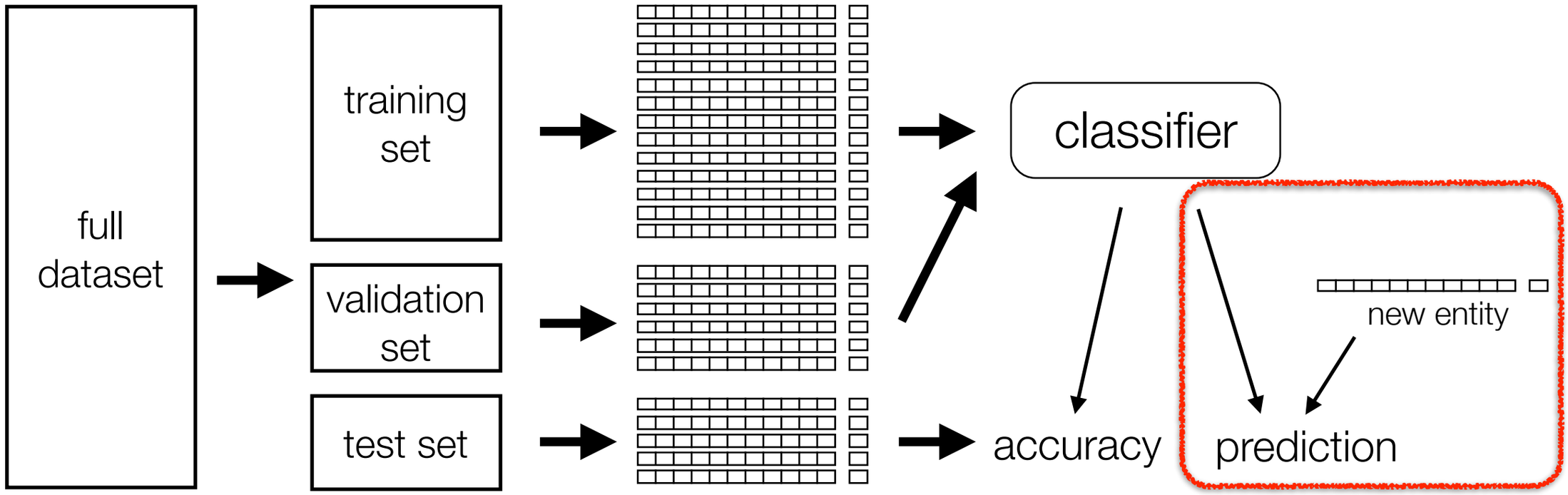
Grid Search

Exhaustively search through hyper parameter space

- Define space
- Discretize space
- Evaluate points (via validation error)



Classification Pipeline



Final classifier can then be used to predict in the wild

Recap Questions

Recap Questions

How does domain knowledge come into play when designing ML pipelines?

Recap Questions

How does domain knowledge come into play when designing ML pipelines?

- Feature extraction!
- E.g., Computer vision folks focus on ways to extract ‘higher-level’ features from raw image pixels
- E.g., Computational biologists extract features from genomic sequences (often times incorporating external data)
- E.g., Google derives features from web index
- Many more examples...

Recap Questions

Recap Questions

Why do we need validation and test datasets?

Recap Questions

Why do we need validation and test datasets?

- To fairly assess generalization ability
- As we train more models during grid search, we could potentially be overfitting to our validation dataset, hence we need the test set to assess the extent to which this occurs.

Recap Questions

Recap Questions

Overfitting: Which of the following statements are true (more than one may be correct)?

Recap Questions

Overfitting: Which of the following statements are true (more than one may be correct)?

- Regularization is used to protect against overfitting.

Recap Questions

Overfitting: Which of the following statements are true (more than one may be correct)?

- Regularization is used to protect against overfitting.
- Overfitting is primarily a concern when training statistical models with large datasets.

Recap Questions

Overfitting: Which of the following statements are true (more than one may be correct)?

- Regularization is used to protect against overfitting.
- Overfitting is primarily a concern when training statistical models with large datasets.
- Overfitting to the training data leads to poor generalization on new data points.

Recap Questions

Overfitting: Which of the following statements are true (more than one may be correct)?

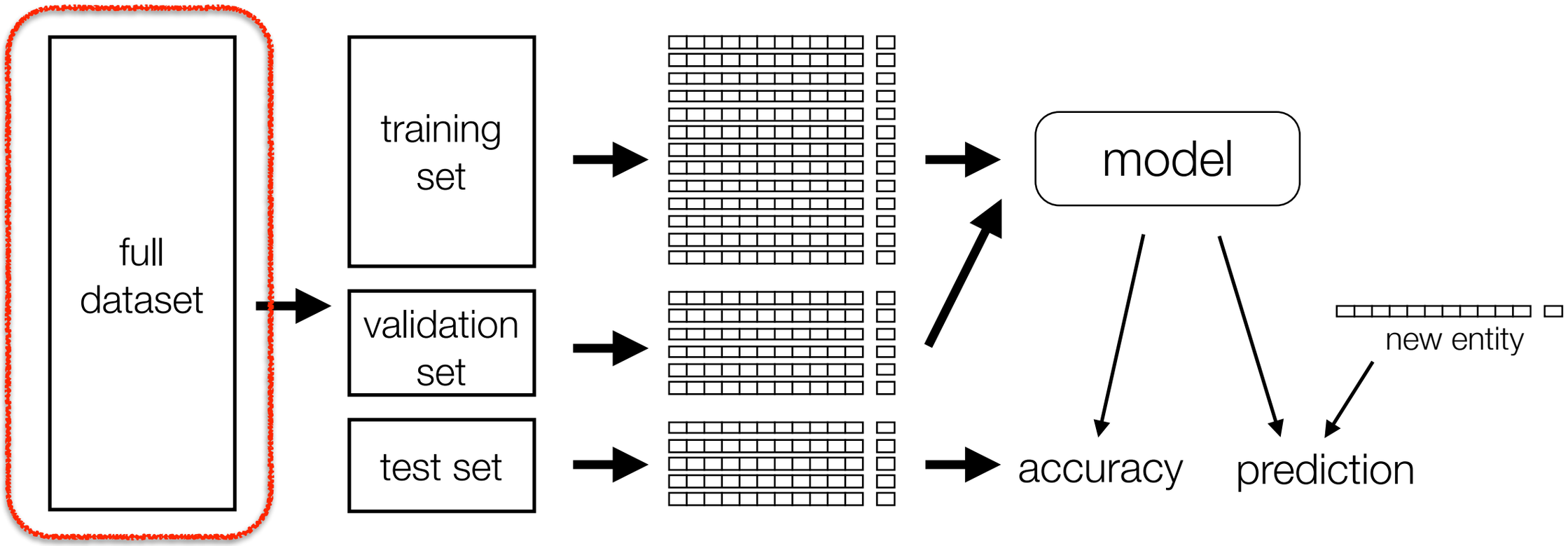
- Regularization is used to protect against overfitting.
- Overfitting is primarily a concern when training statistical models with large datasets.
- Overfitting to the training data leads to poor generalization on new data points.
- Assessing a coin's bias from a single observed coin flip is an example of overfitting.

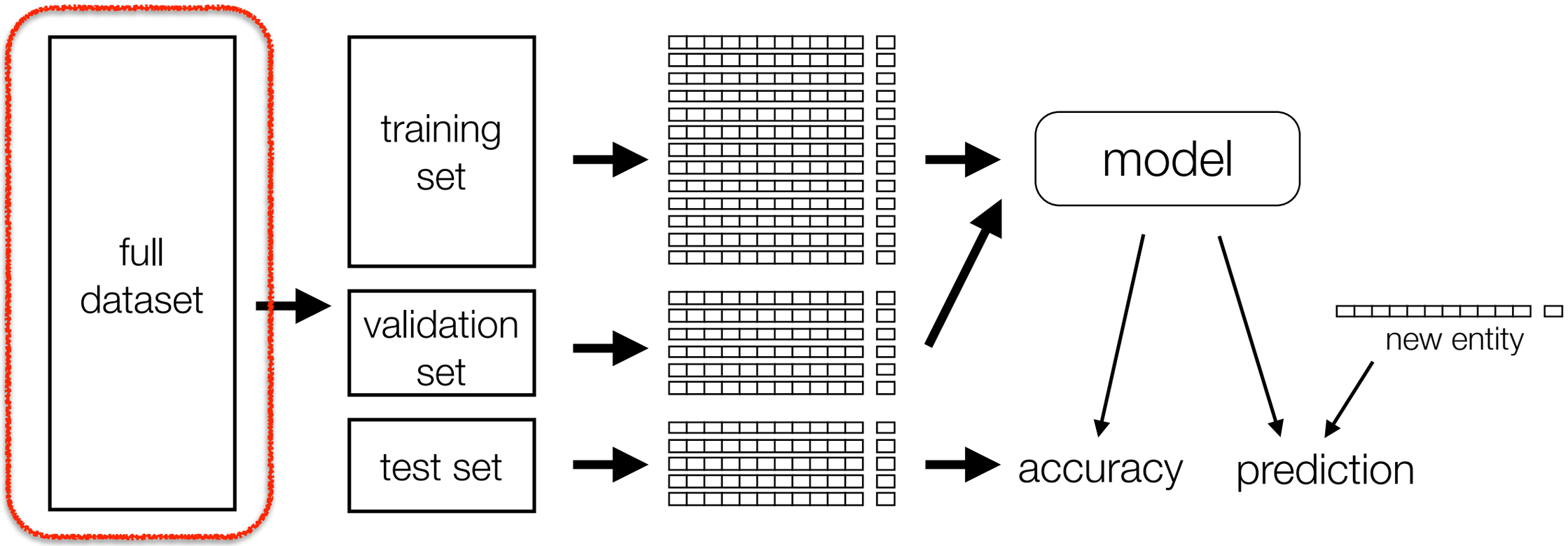
Typical Pipeline

- Feature Extraction
- Supervised ML
- Model Evaluation
- Spam Classification Example

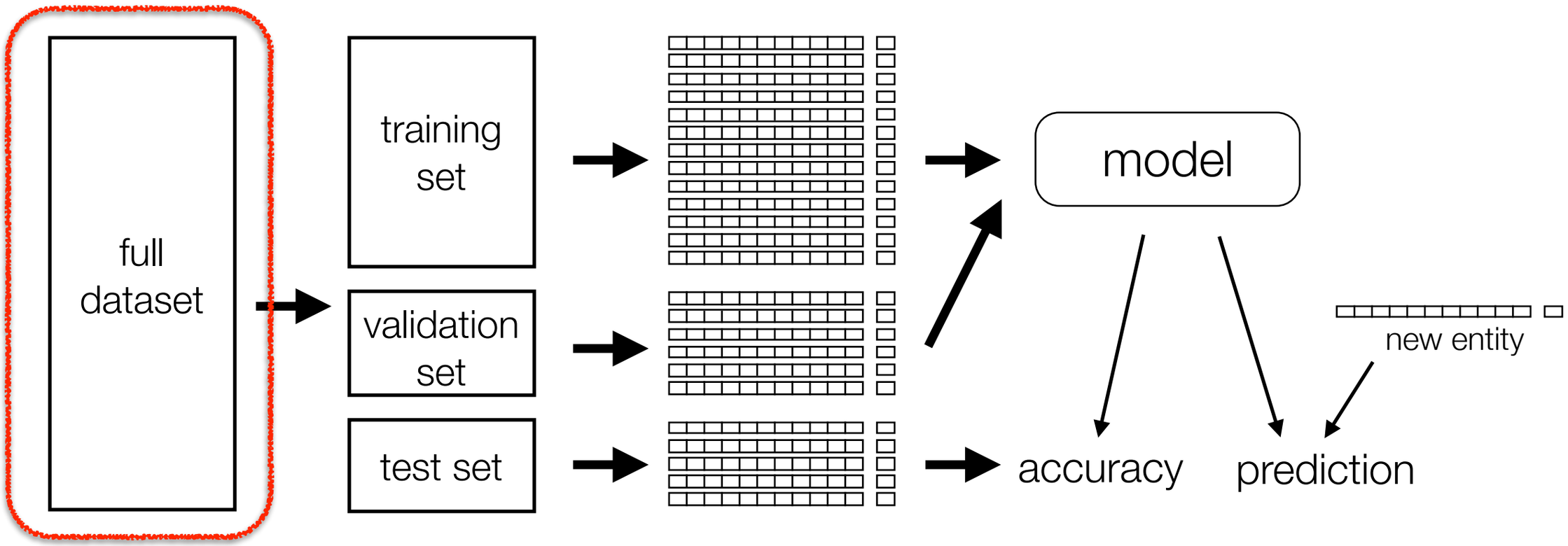
Millionsong Example

- Polynomial features
- Linear regression and Normal Equations
- Ridge Regression


full dataset
training set
validation set
test set
model
accuracy
prediction
new entity



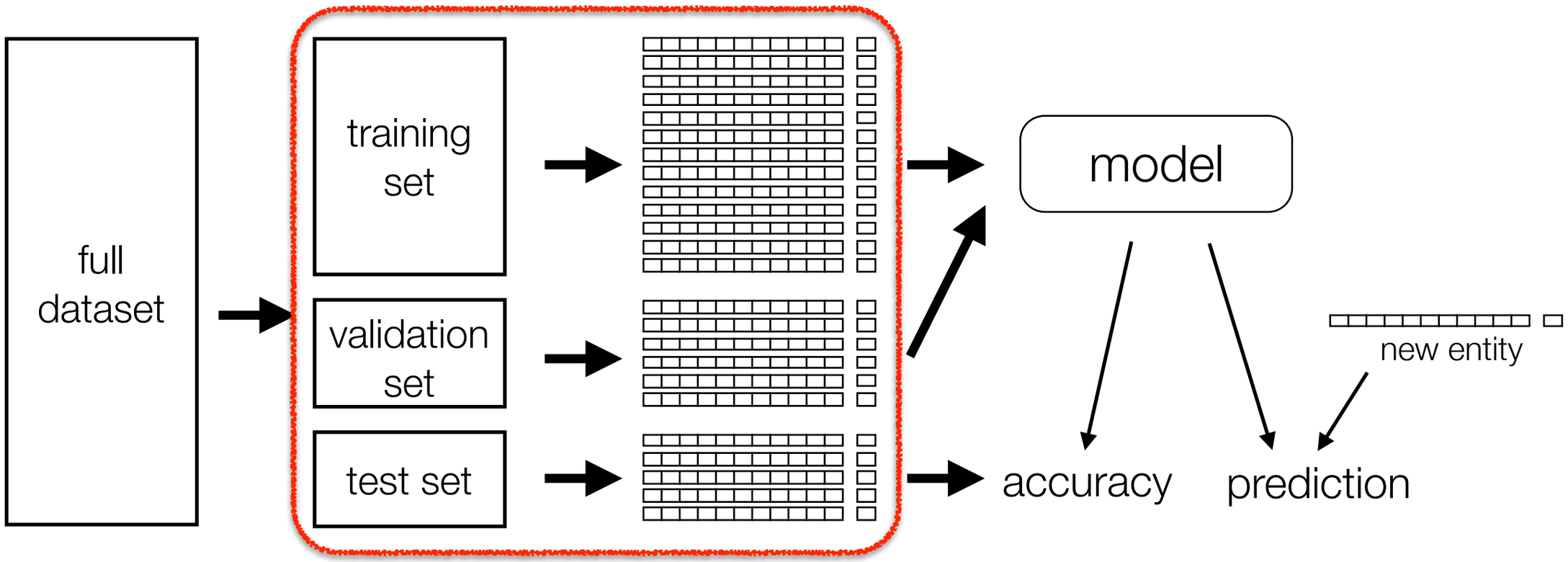
Goal: Predict song's release year from audio features



Goal: Predict song's release year from audio features

Data: Millionsong Dataset from UCI ML Repository

- Western, commercial tracks from 1980-2014
- 12 timbre averages (features) and release year (label)



Quadratic features

- Captures covariance of initial timbre features
- Leads to a non-linear model relative to raw features

$$x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \qquad \Phi(x) = \begin{bmatrix} x_1^2 \\ \sqrt{2}x_1x_2 \\ x_2^2 \end{bmatrix}$$

Given 2 dimensional data, quadratic features are:

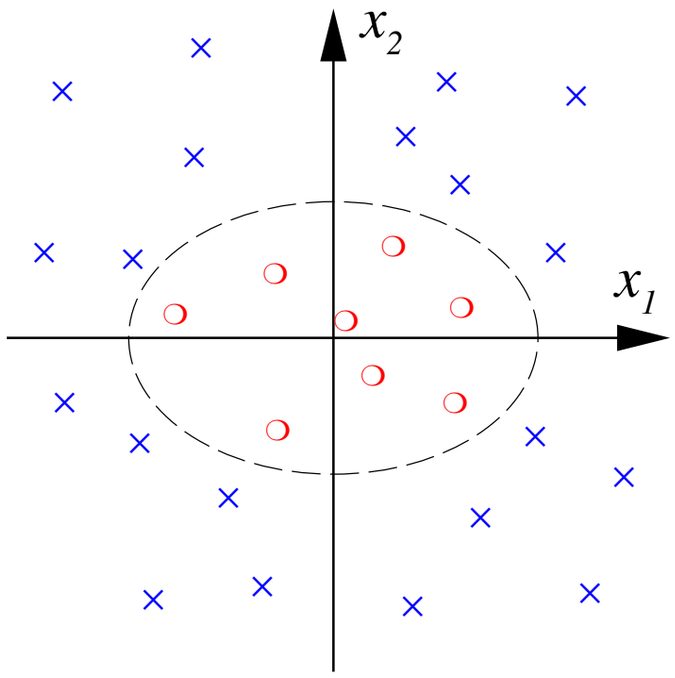
$$x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \qquad \Phi(x) = \begin{bmatrix} x_1^2 \\ \sqrt{2}x_1x_2 \\ x_2^2 \end{bmatrix}$$

Given 2 dimensional data, quadratic features are:

$$x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \qquad \Phi(x) = \begin{bmatrix} x_1^2 \\ \sqrt{2}x_1x_2 \\ x_2^2 \end{bmatrix}$$

Mapping based on definition of quadratic kernel

$$(x_1, x_2)$$

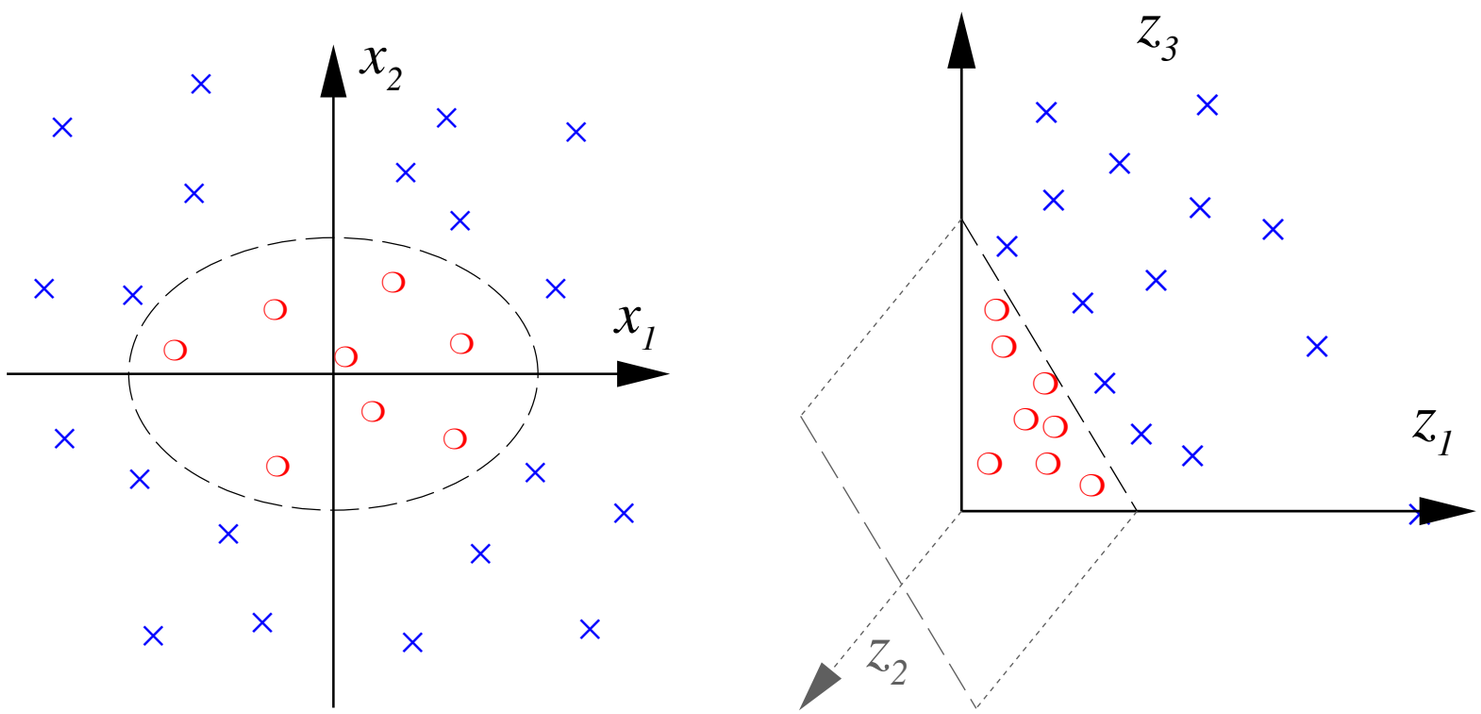


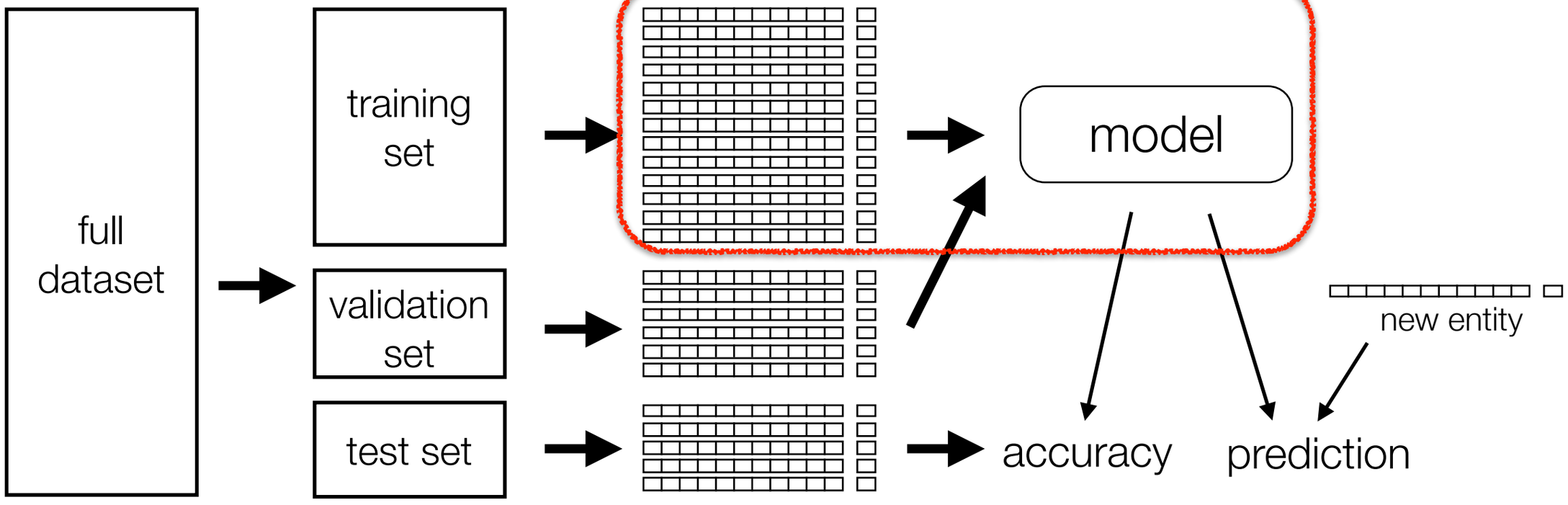
Given 2 dimensional data, quadratic features are:

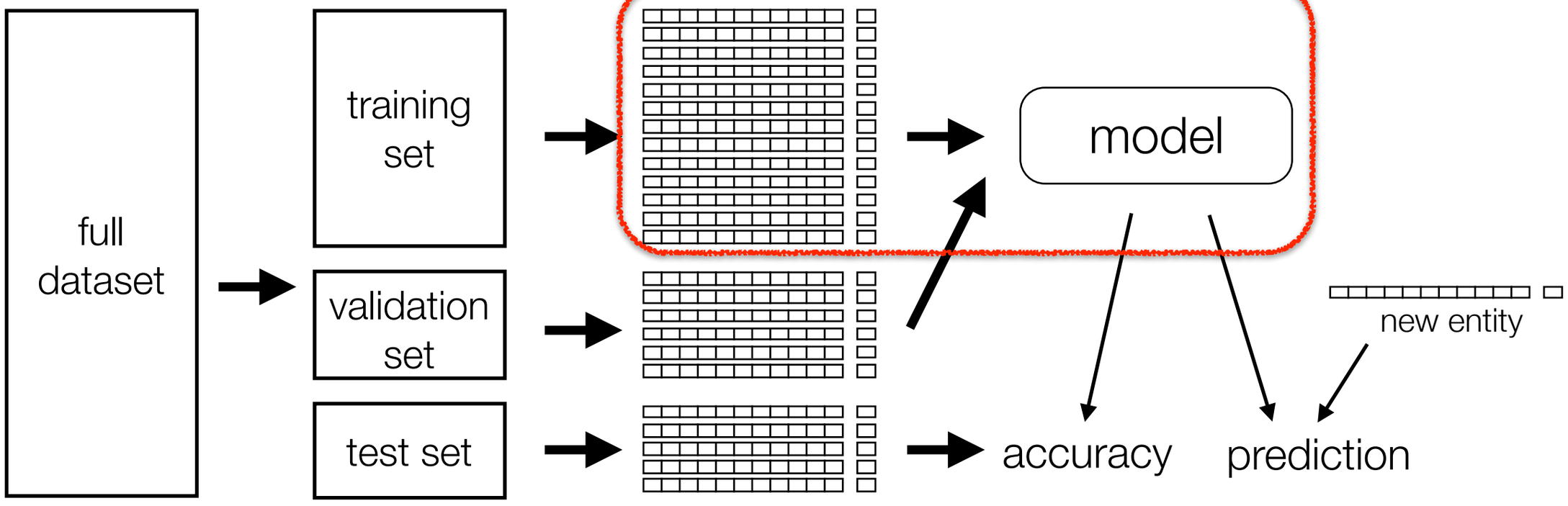
$$x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \qquad \Phi(x) = \begin{bmatrix} x_1^2 \\ \sqrt{2}x_1x_2 \\ x_2^2 \end{bmatrix}$$

Mapping based on definition of quadratic kernel

$$(x_1, x_2) \mapsto (z_1, z_2, z_3) := (x_1^2, \sqrt{(2)}x_1x_2, x_2^2)$$

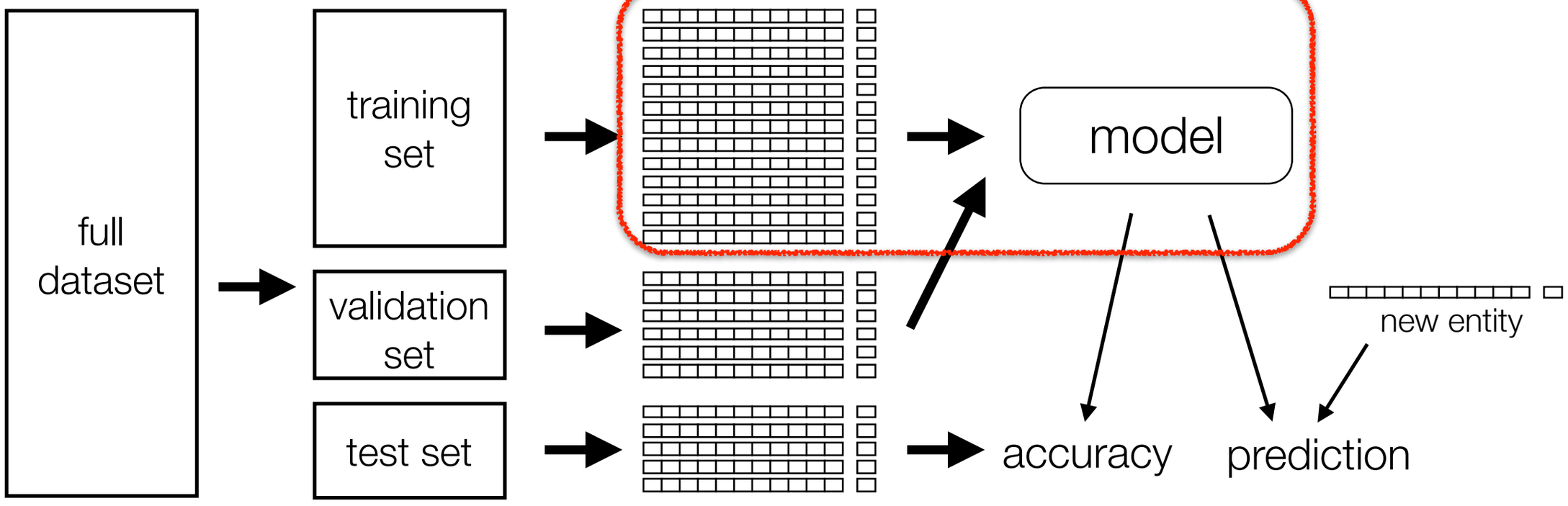



full dataset
training set
validation set
test set
model
new entity
accuracy
prediction



Least Squares Regression: Learn a mapping from entities to continuous labels given a training set

- There is a notion of ‘closeness’ between labels

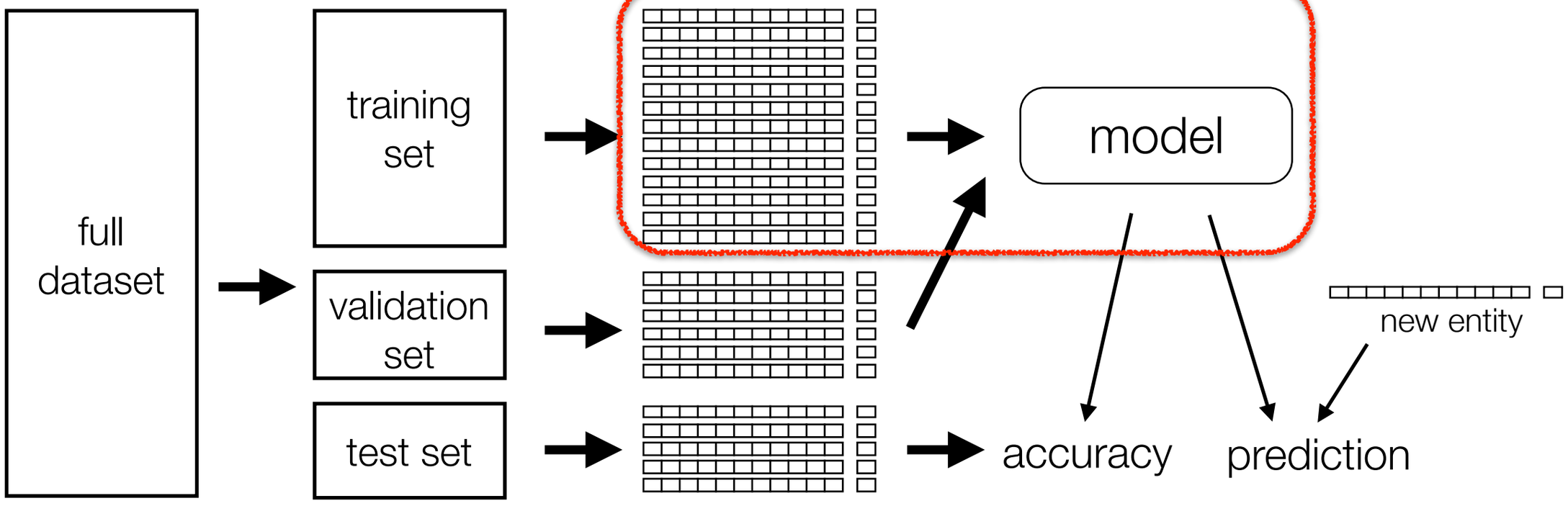


Least Squares Regression: Learn a mapping from entities to continuous labels given a training set

- There is a notion of ‘closeness’ between labels

Examples:

- Timbre features → Song year

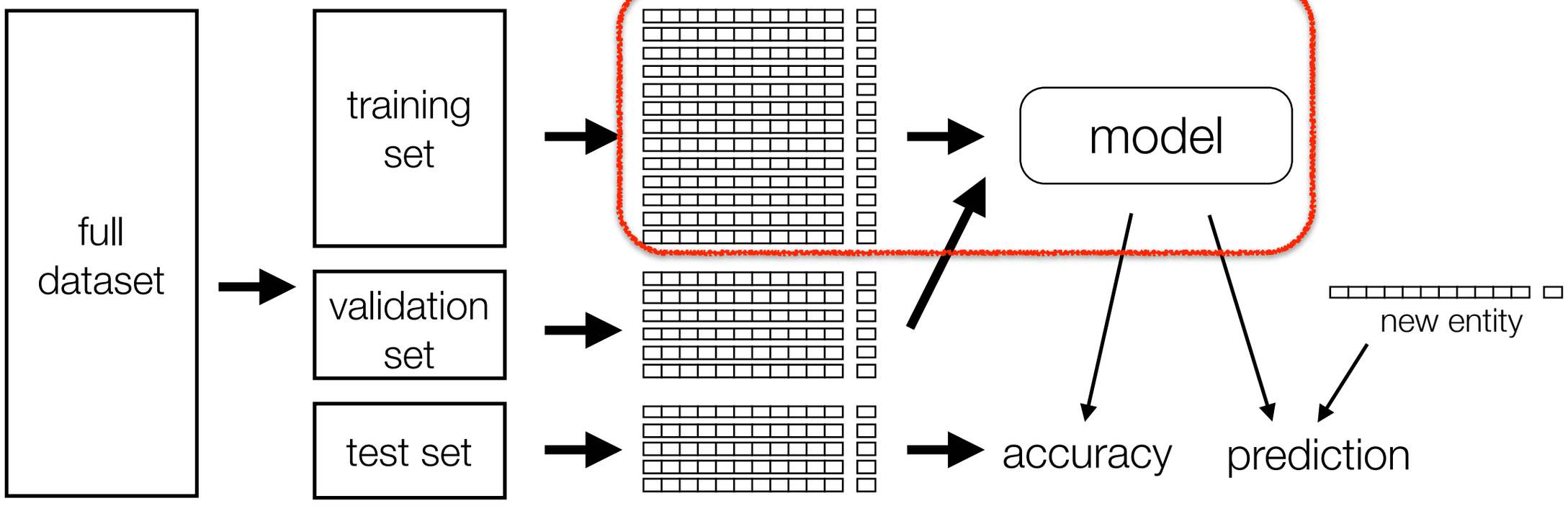


Least Squares Regression: Learn a mapping from entities to continuous labels given a training set

- There is a notion of ‘closeness’ between labels

Examples:

- Timbre features → Song year
- Processes, memory → Power consumption

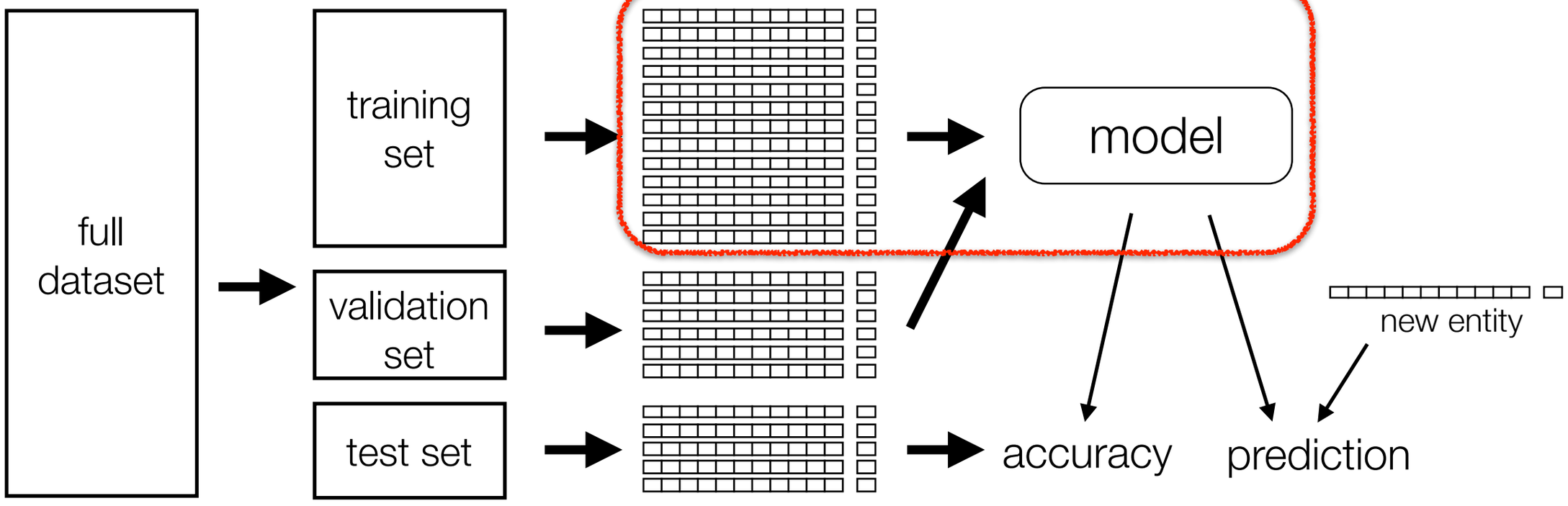


Least Squares Regression: Learn a mapping from entities to continuous labels given a training set

- There is a notion of ‘closeness’ between labels

Examples:

- Timbre features → Song year
- Processes, memory → Power consumption
- Historical finance info → Future stock price



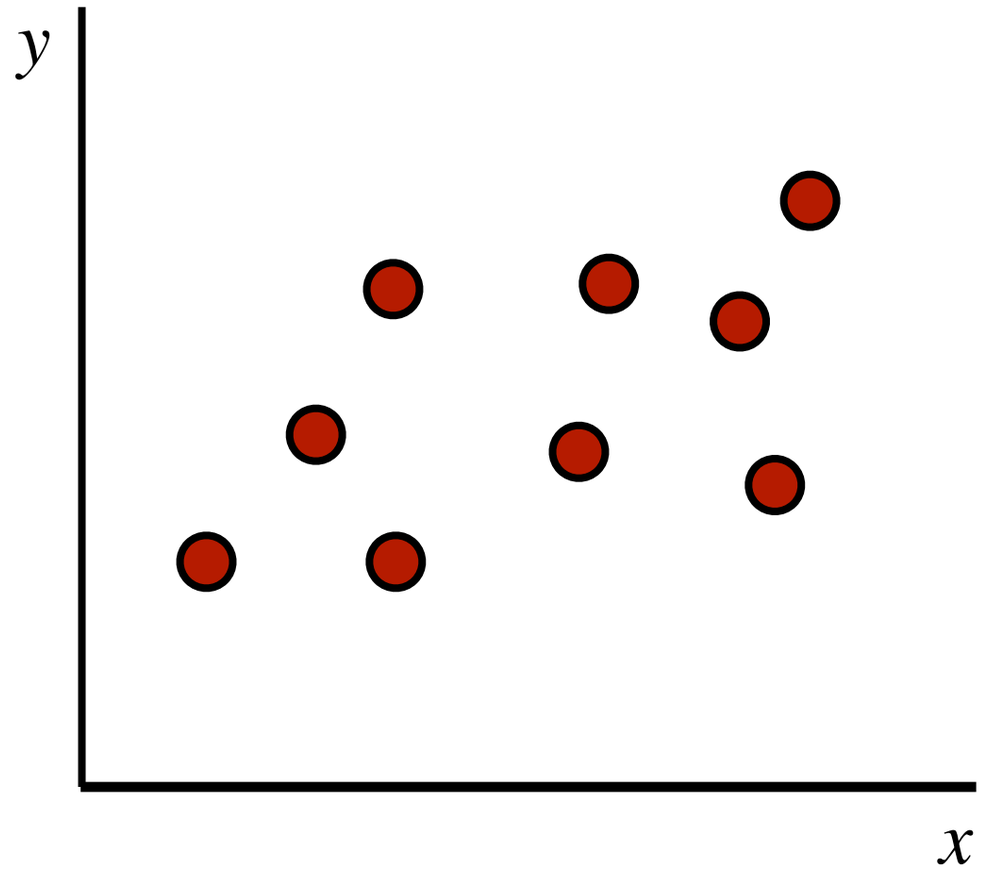
Least Squares Regression: Learn a mapping from entities to continuous labels given a training set

- There is a notion of ‘closeness’ between labels

Examples:

- Timbre features → Song year
- Processes, memory → Power consumption
- Historical finance info → Future stock price
- Many more

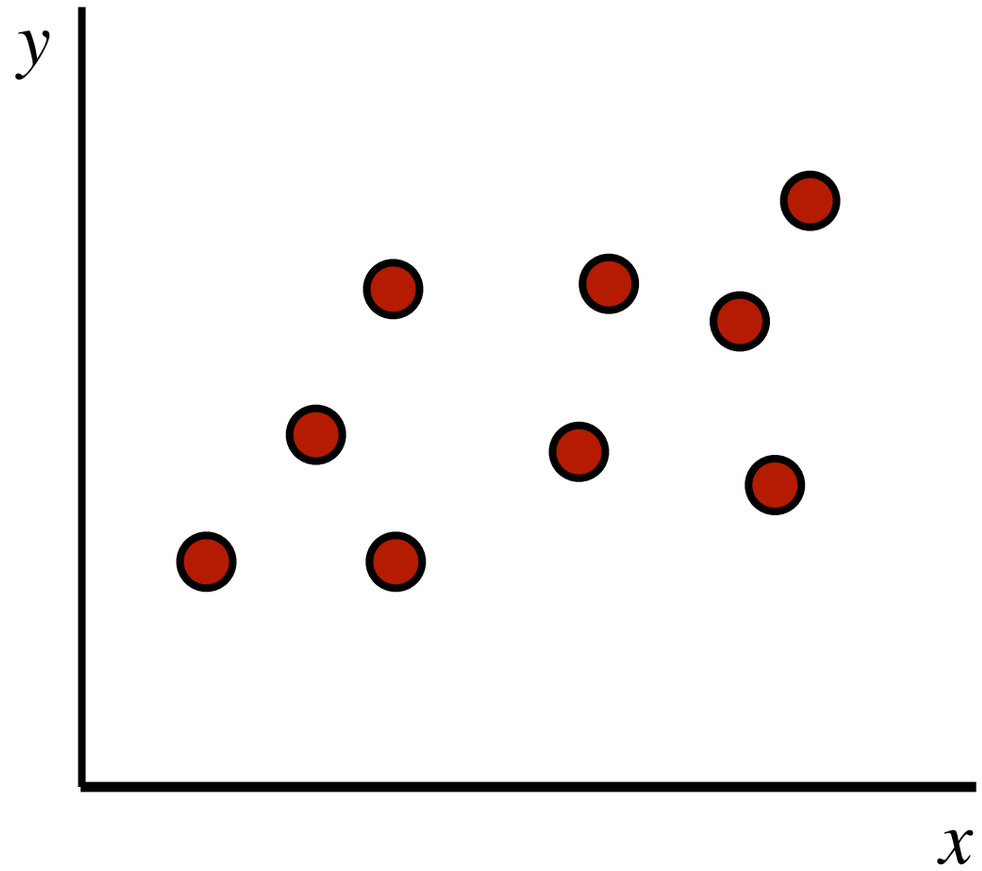
Linear Least Squares



Linear Least Squares

Features: x coordinate

Labels: y coordinate

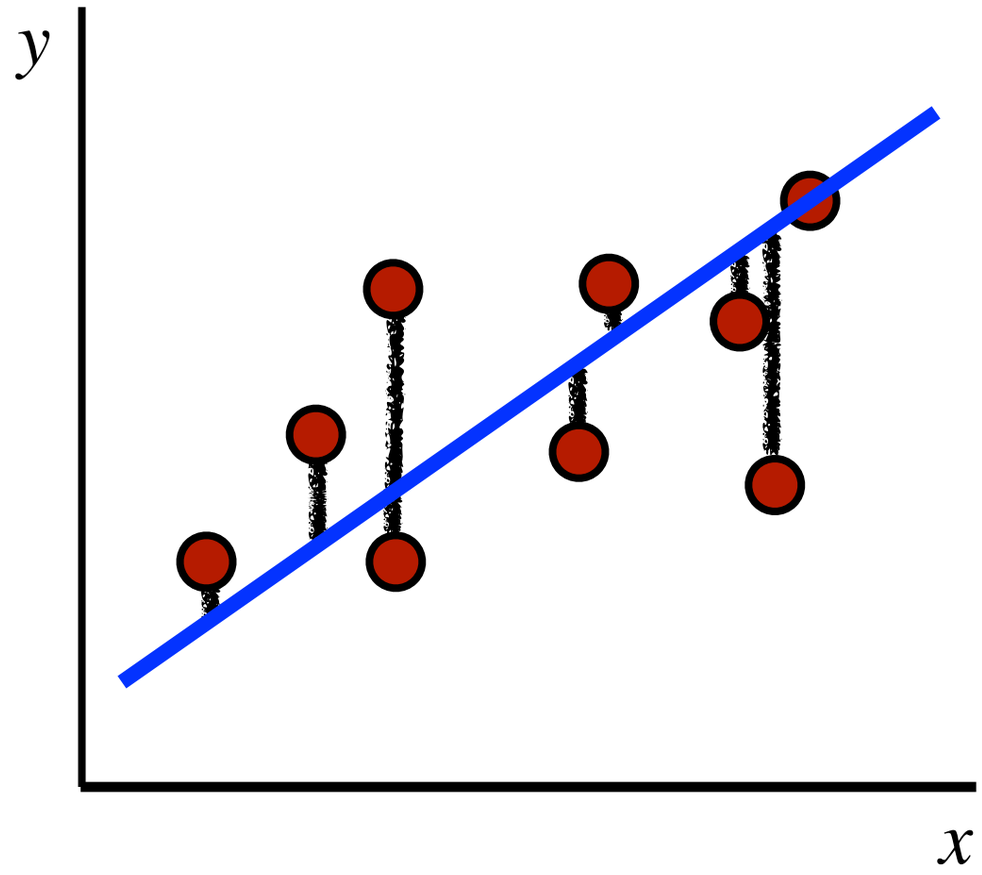


Linear Least Squares

Features: x coordinate

Labels: y coordinate

Goal is to find line of best fit: $y \approx wx + b$

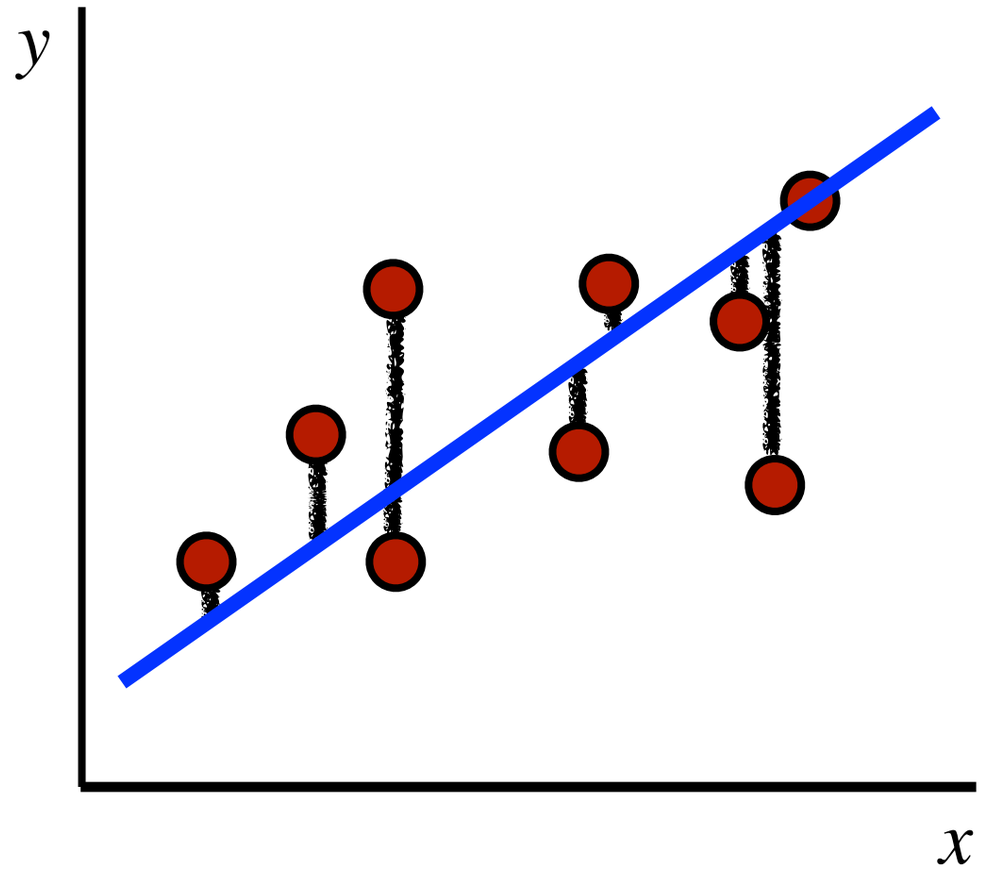


Linear Least Squares

Features: x coordinate

Labels: y coordinate

Goal is to find line of best fit: $y \approx wx + b$



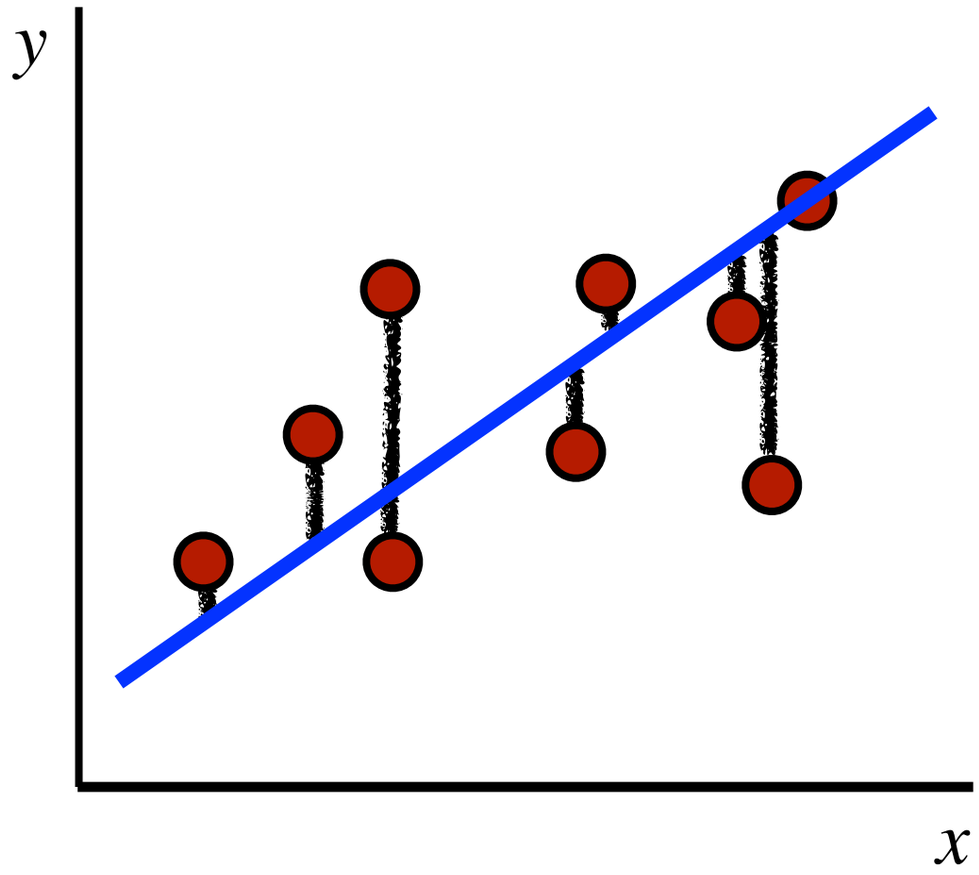
Model Parameters: slope (w) and intercept (b)

Linear Least Squares

Features: x coordinate

Labels: y coordinate

Goal is to find line of best fit: $y \approx wx + b$



Model Parameters: slope (w) and intercept (b)

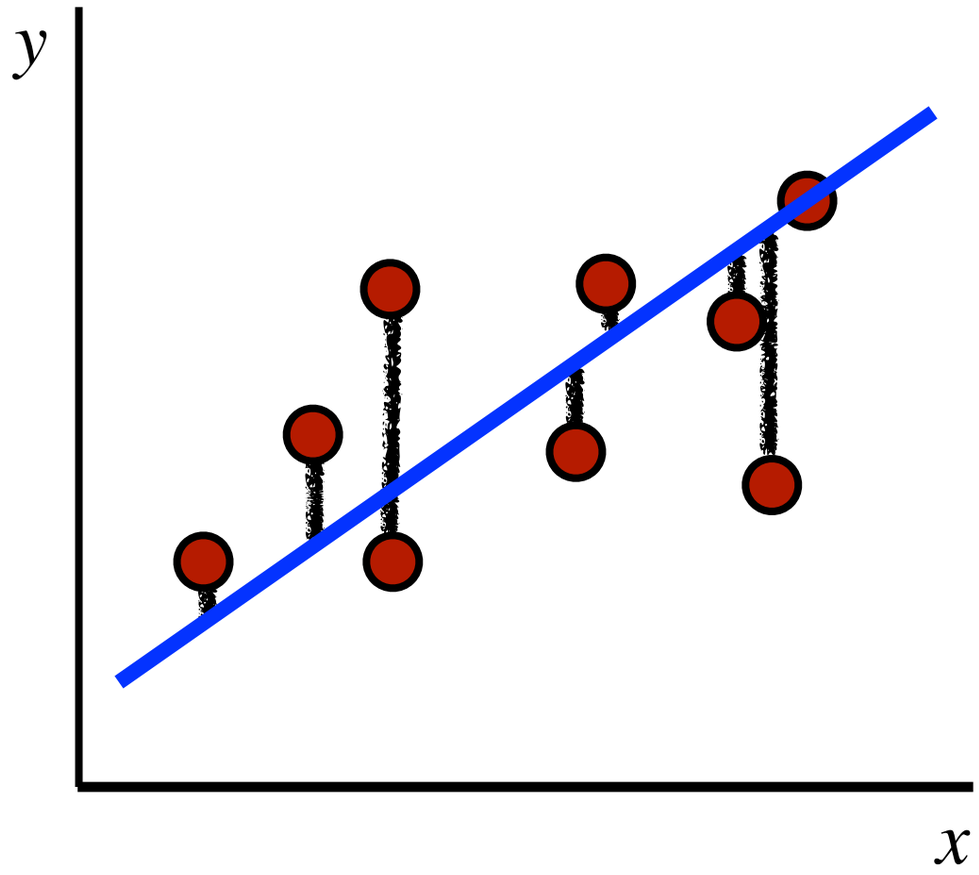
With multiple features: $y \approx b + \sum_{i=1}^{d} w_i x_i$

Linear Least Squares

Features: x coordinate

Labels: y coordinate

Goal is to find line of best fit: $y \approx wx + b$



Model Parameters: slope (w) and intercept (b)

With multiple features: $y \approx b + \sum_{i=1}^{d} w_i x_i$

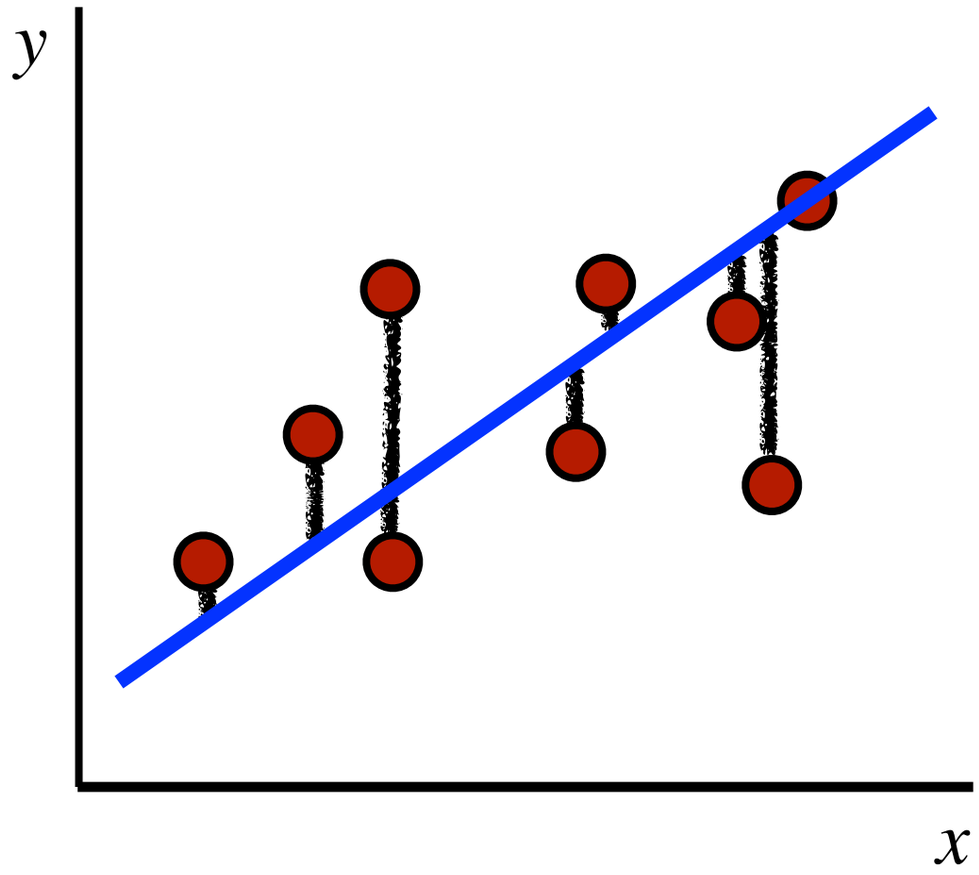
Augment with 1 for intercept

Linear Least Squares

Features: x coordinate

Labels: y coordinate

Goal is to find line of best fit: $y \approx wx + b$



Model Parameters: slope (w) and intercept (b)

With multiple features: $y \approx b + \sum_{i=1}^{d} w_i x_i$

Augment with 1 for intercept

$$y \approx \sum_{i=1}^{d+1} w_i x_i = w^\top x$$

Terminology:

Terminology:

- Given n training points with d features
- X: n by d data matrix storing
- y: real-valued labels for the n points

Terminology:

- Given n training points with d features
- X: n by d data matrix storing
- y: real-valued labels for the n points
- $\hat{y} = Xw$: predicted labels for the n data points
- w: d dimensional regression parameters

Terminology:

- Given n training points with d features
- X: n by d data matrix storing
- y: real-valued labels for the n points
- $\hat{y} = Xw$: predicted labels for the n data points
- w: d dimensional regression parameters

Least Squares Regression: Learn mapping (w) from features to labels that minimizes residual sum of squares:

$$\min_w ||Xw - y||_2^2$$

Terminology:

- Given n training points with d features
- X: n by d data matrix storing
- y: real-valued labels for the n points
- $\hat{y} = Xw$: predicted labels for the n data points
- w: d dimensional regression parameters

Least Squares Regression: Learn mapping (w) from features to labels that minimizes residual sum of squares:

$$\min_w ||Xw - y||_2^2$$

Convex optimization with closed-form solution

- Setting derivative to zero and rearranging yields normal equations: $(X^\top X)w = X^\top y$

Terminology:

- Given n training points with d features
- X: n by d data matrix storing
- y: real-valued labels for the n points
- $\hat{y} = Xw$: predicted labels for the n data points
- w: d dimensional regression parameters

Least Squares Regression: Learn mapping (w) from features to labels that minimizes residual sum of squares:

$$\min_w ||Xw - y||_2^2$$

Convex optimization with closed-form solution

- Setting derivative to zero and rearranging yields normal equations: $(X^\top X)w = X^\top y$
- $w = (X^\top X)^{-1} X^\top y$

Overfitting and Generalization

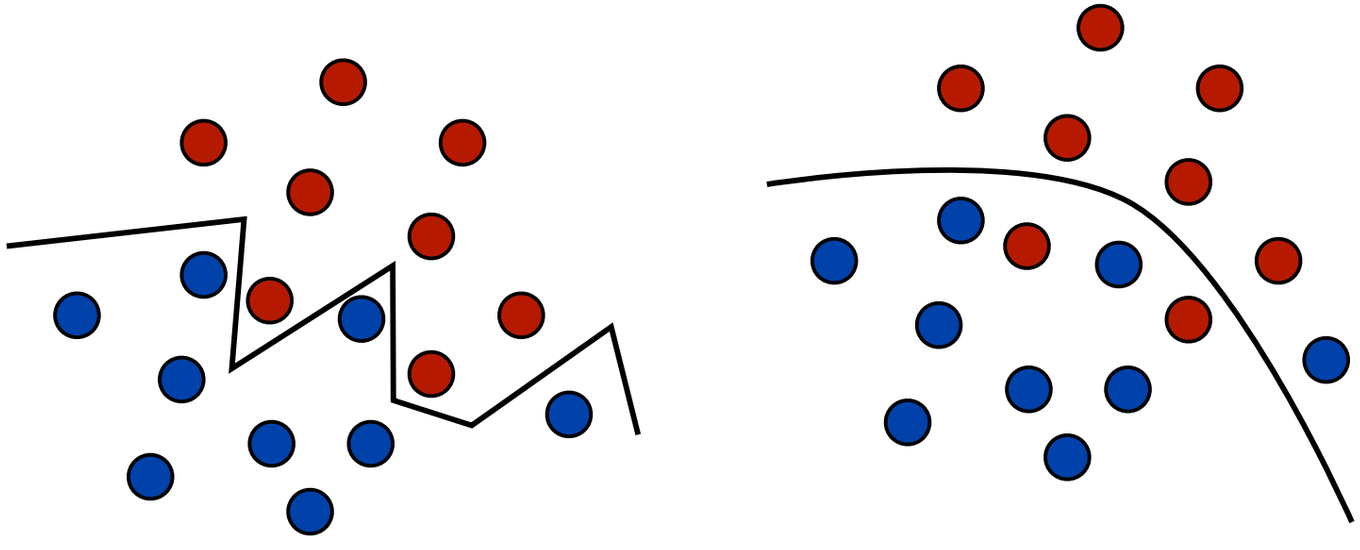


Image Credit: Foundations of Machine Learning, Mohri, Rostamizadeh, Talwalkar

Overfitting and Generalization

We want a model that makes good predictions on new data, i.e., 'generalization'

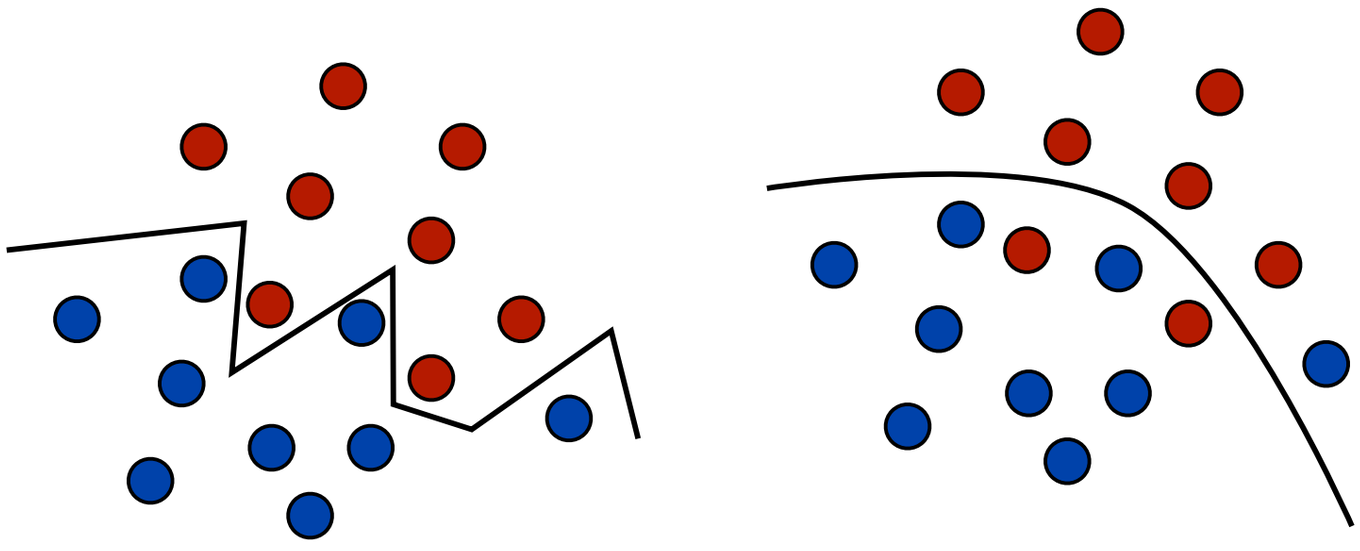


Image Credit: Foundations of Machine Learning, Mohri, Rostamizadeh, Talwalkar

Overfitting and Generalization

We want a model that makes good predictions on new data, i.e., 'generalization'

Least squares regression only aims to minimize error on training data (empirical risk minimization)

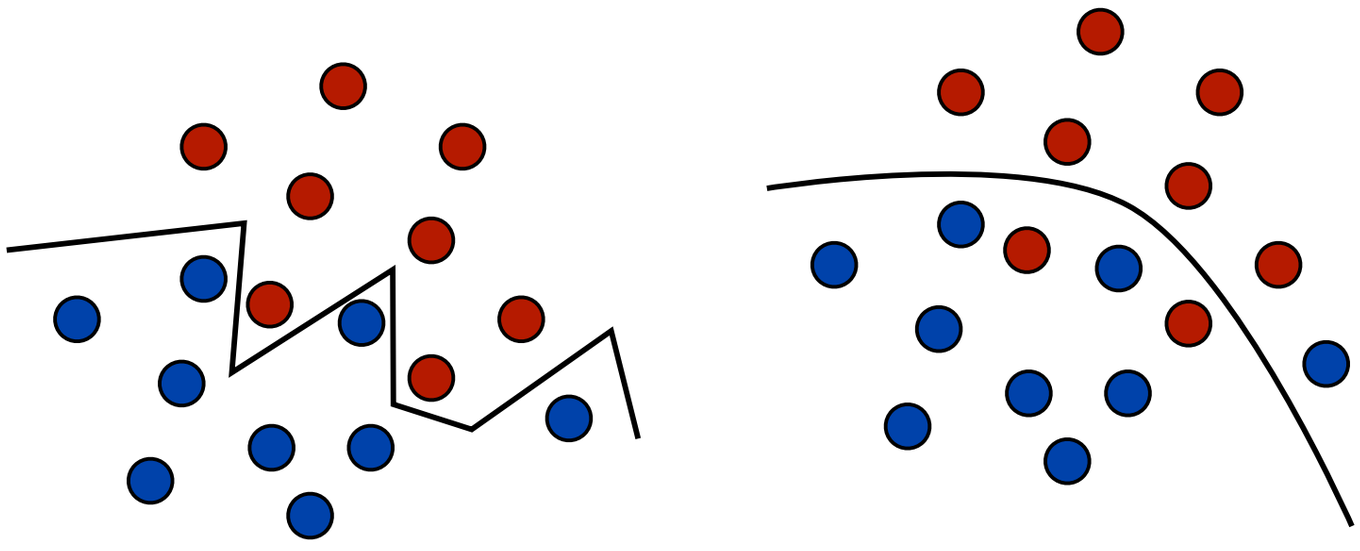


Image Credit: Foundations of Machine Learning, Mohri, Rostamizadeh, Talwalkar

Overfitting and Generalization

We want a model that makes good predictions on new data, i.e., 'generalization'

Least squares regression only aims to minimize error on training data (empirical risk minimization)

Can we also penalize for model complexity?

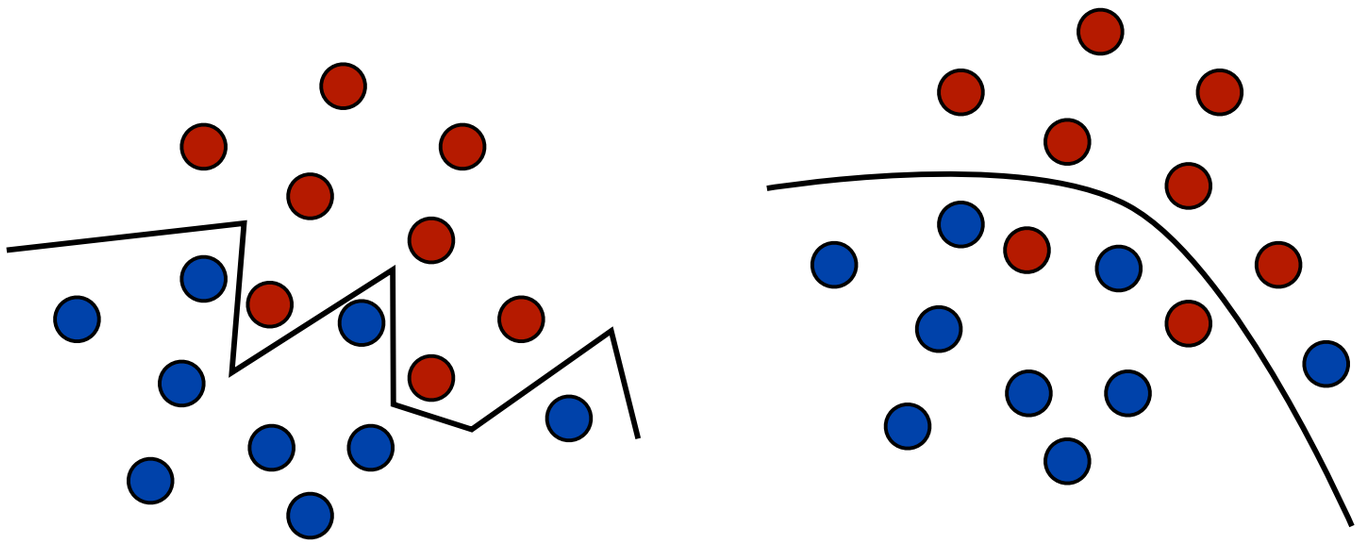


Image Credit: Foundations of Machine Learning, Mohri, Rostamizadeh, Talwalkar

Overfitting and Generalization

We want a model that makes good predictions on new data, i.e., 'generalization'

Least squares regression only aims to minimize error on training data (empirical risk minimization)

Can we also penalize for model complexity?

- Intuitively, models with smaller weights are simpler

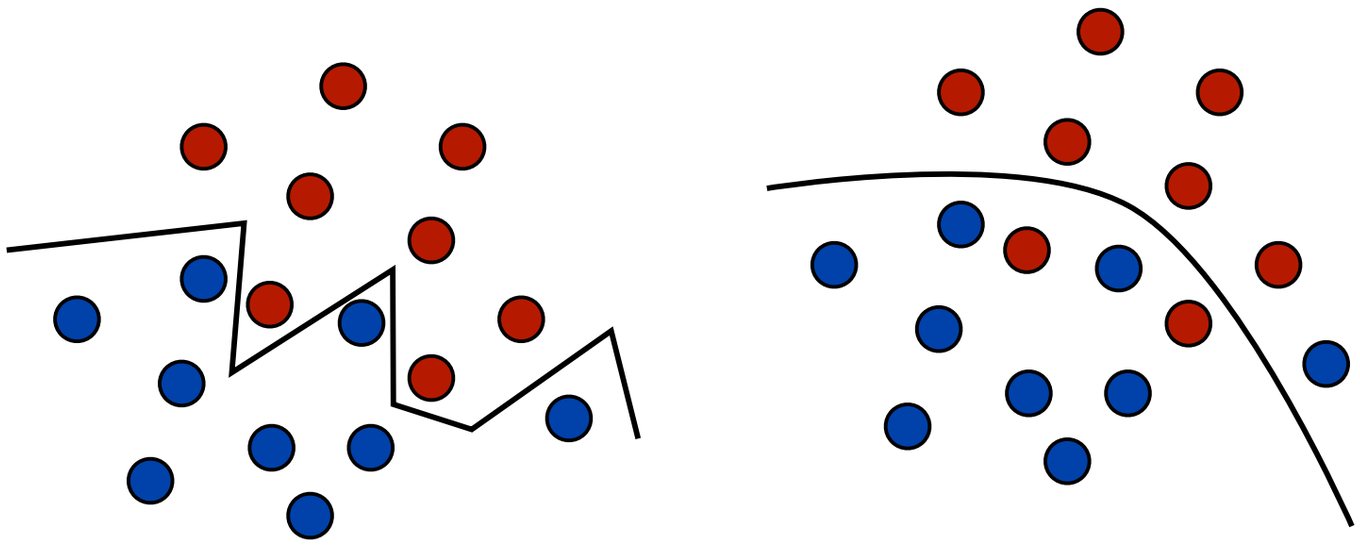


Image Credit: Foundations of Machine Learning, Mohri, Rostamizadeh, Talwalkar

$$\hat{y} = Xw$$

Terminology:

- Given n training points with d features
- X: n by d data matrix storing
- y: real-valued labels for the n points
- $\hat{y} = Xw$: predicted labels for the n data points
- w: d dimensional regression parameters

Terminology:

- Given n training points with d features
- X: n by d data matrix storing
- y: real-valued labels for the n points
- $\hat{y} = Xw$: predicted labels for the n data points
- w: d dimensional regression parameters

Ridge Regression: Learn mapping (w) that minimizes residual sum of squares along with a regularization term:

$$\min_w \underbrace{||Xw - y||_2^2}_{\text{Empirical Risk Minimization}}$$

Terminology:

- Given n training points with d features
- X: n by d data matrix storing
- y: real-valued labels for the n points
- $\hat{y} = Xw$: predicted labels for the n data points
- w: d dimensional regression parameters

Ridge Regression: Learn mapping (w) that minimizes residual sum of squares along with a regularization term:

$$\min_{w} \underbrace{||Xw - y||_2^2}_{\text{Empirical Risk Minimization}} + \lambda \underbrace{||w||_2^2}_{\text{Model Complexity}}$$

Terminology:

- Given n training points with d features
- X: n by d data matrix storing
- y: real-valued labels for the n points
- $\hat{y} = Xw$: predicted labels for the n data points
- w: d dimensional regression parameters

Ridge Regression: Learn mapping (w) that minimizes residual sum of squares along with a regularization term:

$$\min_{w} \underbrace{||Xw - y||_2^2}_{\text{Empirical Risk Minimization}} + \lambda \underbrace{||w||_2^2}_{\text{Model Complexity}}$$

Convex, closed-form involving normal equations

Terminology:

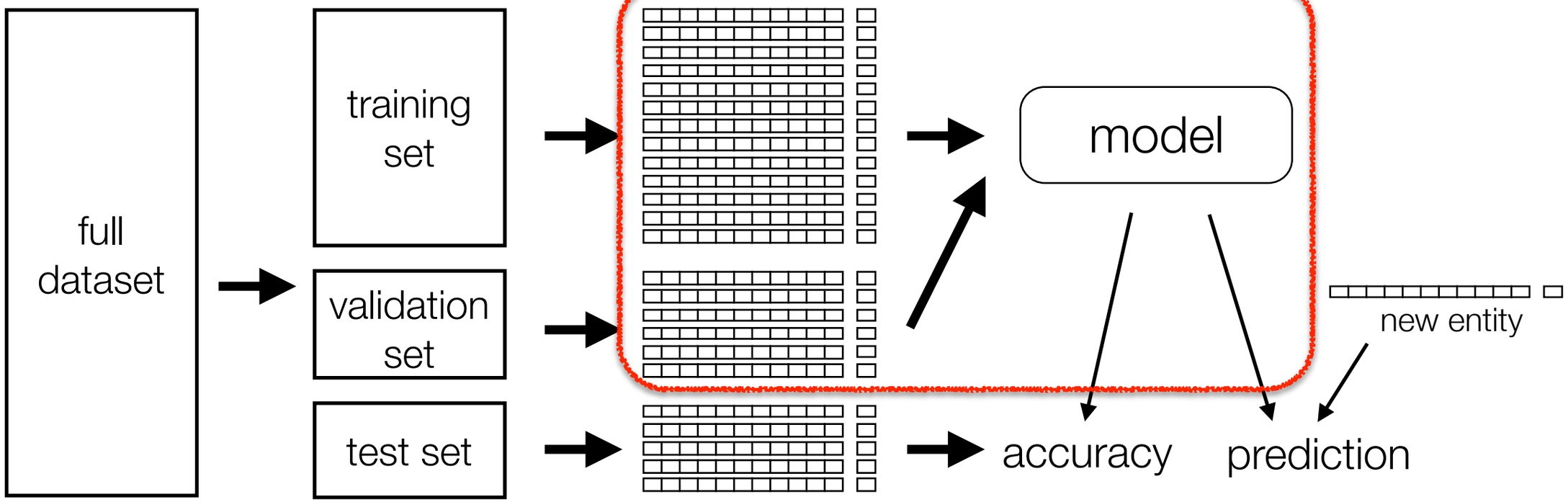
- Given n training points with d features
- X: n by d data matrix storing
- y: real-valued labels for the n points
- $\hat{y} = Xw$: predicted labels for the n data points
- w: d dimensional regression parameters

Ridge Regression: Learn mapping (w) that minimizes residual sum of squares along with a regularization term:

$$\min_w \underbrace{||Xw - y||_2^2}_{\text{Empirical Risk Minimization}} + \lambda \underbrace{||w||_2^2}_{\text{Model Complexity}}$$

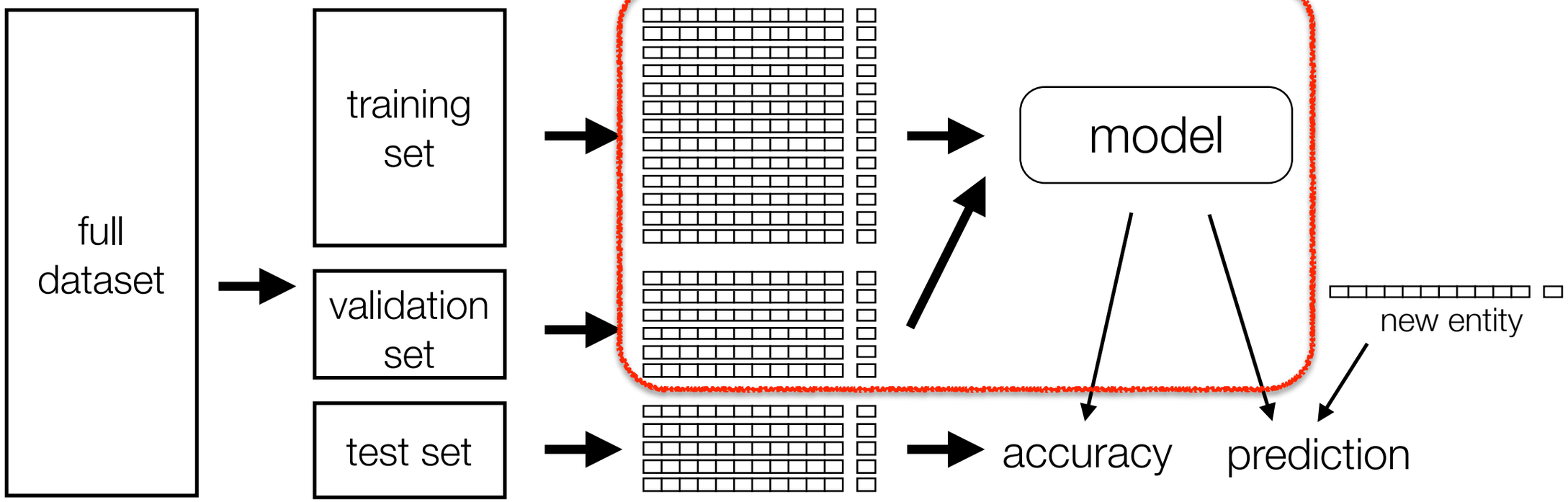
Convex, closed-form involving normal equations

- $w = (X^\top X + \lambda I)^{-1} X^\top y$



regularization hyperparamater

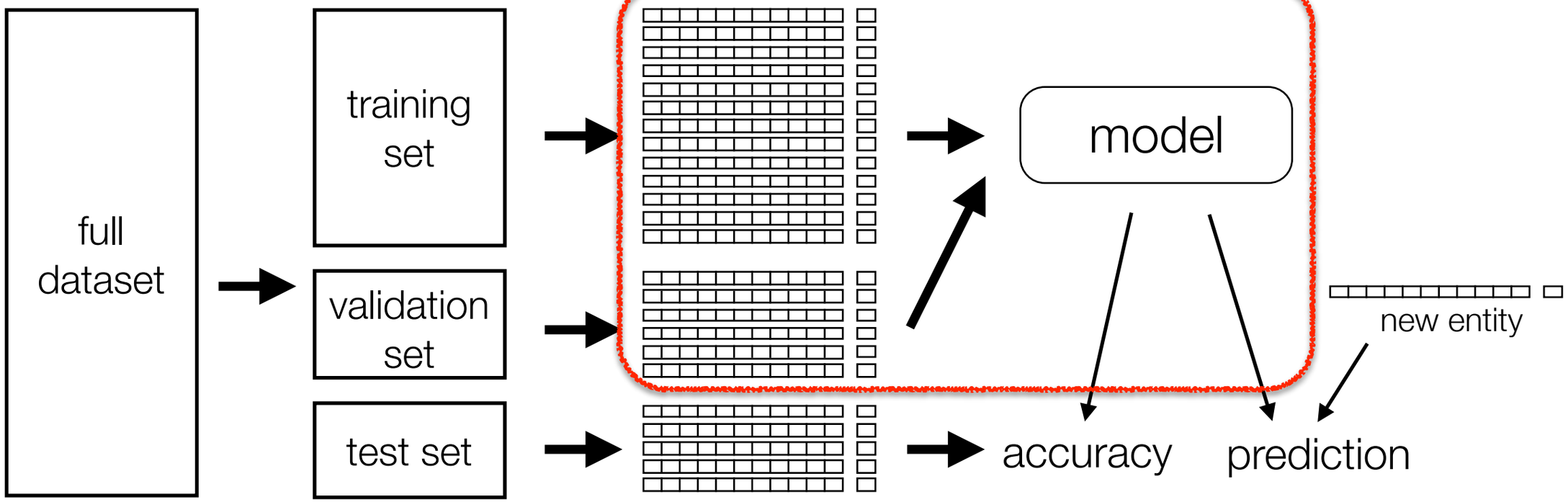
$$\min_{w} ||Xw - y||_2^2 + \lambda ||w||_2^2$$



regularization hyperparamater

$$\min_{w} ||Xw - y||_2^2 + \lambda ||w||_2^2$$

How do we choose the regularization parameter?



regularization hyperparamater

$$\min_{w} ||Xw - y||_2^2 + \lambda ||w||_2^2$$

How do we choose the regularization parameter?

Use validation set and grid search to tune it!

Recap Question

Recap Question

For each problem, determine whether it's an instance of classification or regression.

Recap Question

For each problem, determine whether it's an instance of classification or regression.

- Predicting SAT scores from high-school GPA.

Recap Question

For each problem, determine whether it's an instance of classification or regression.

- Predicting SAT scores from high-school GPA.
- Determining whether an image contains a plant or animal.

Recap Question

For each problem, determine whether it's an instance of classification or regression.

- Predicting SAT scores from high-school GPA.
- Determining whether an image contains a plant or animal.
- Categorizing images into one of 100 classes.

Recap Question

Recap Question

How would you expect the regularization parameter to vary as the size of your dataset (n) increases?

Recap Question

How would you expect the regularization parameter to vary as the size of your dataset (n) increases?

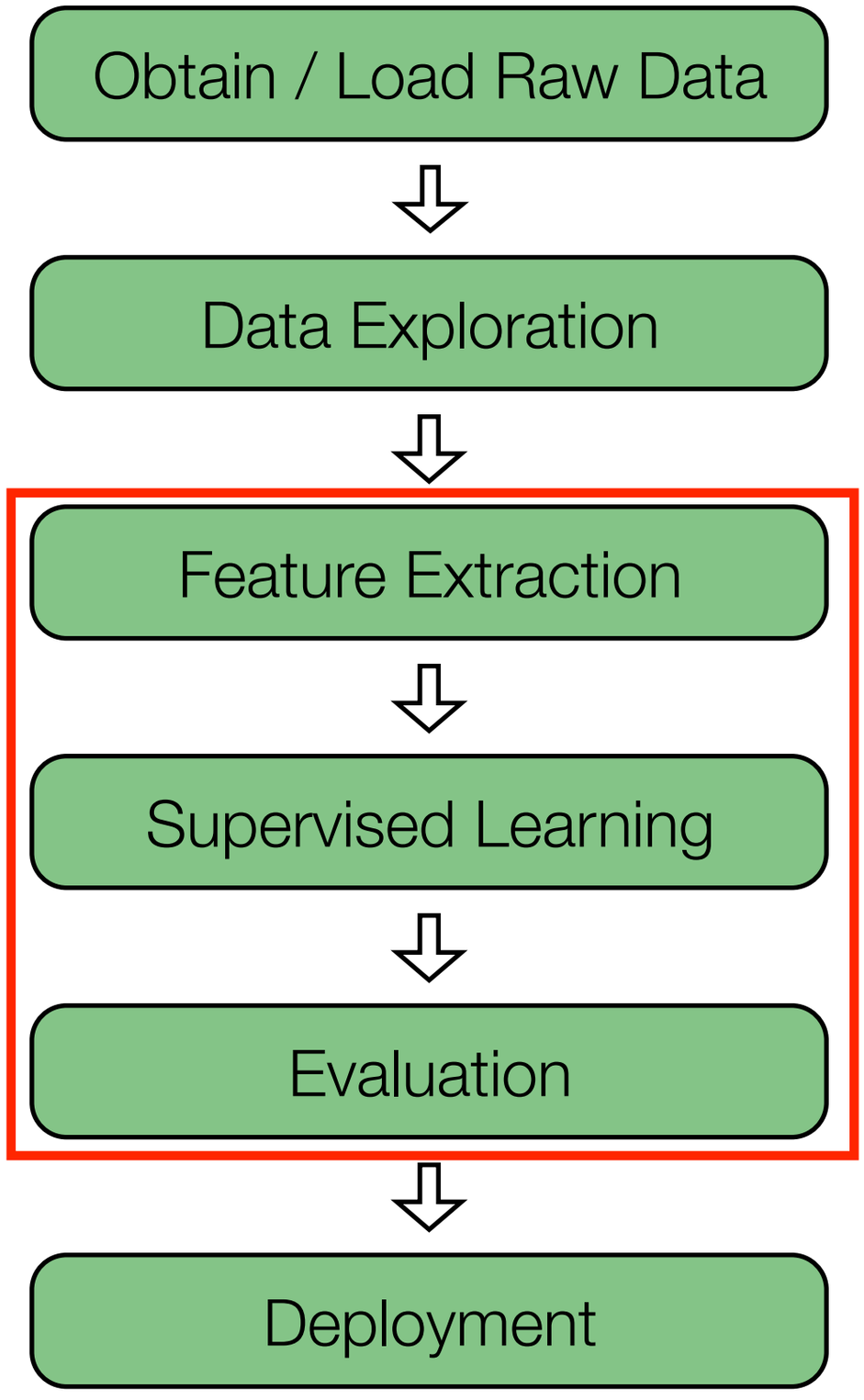
- The regularization parameter should decrease as n increases.

Recap Question

How would you expect the regularization parameter to vary as the size of your dataset (n) increases?

- The regularization parameter should decrease as n increases.
- As you see more and more data, you are less likely to overfit (think about learning a coin's bias from one coin flip versus from 10M coin flips).

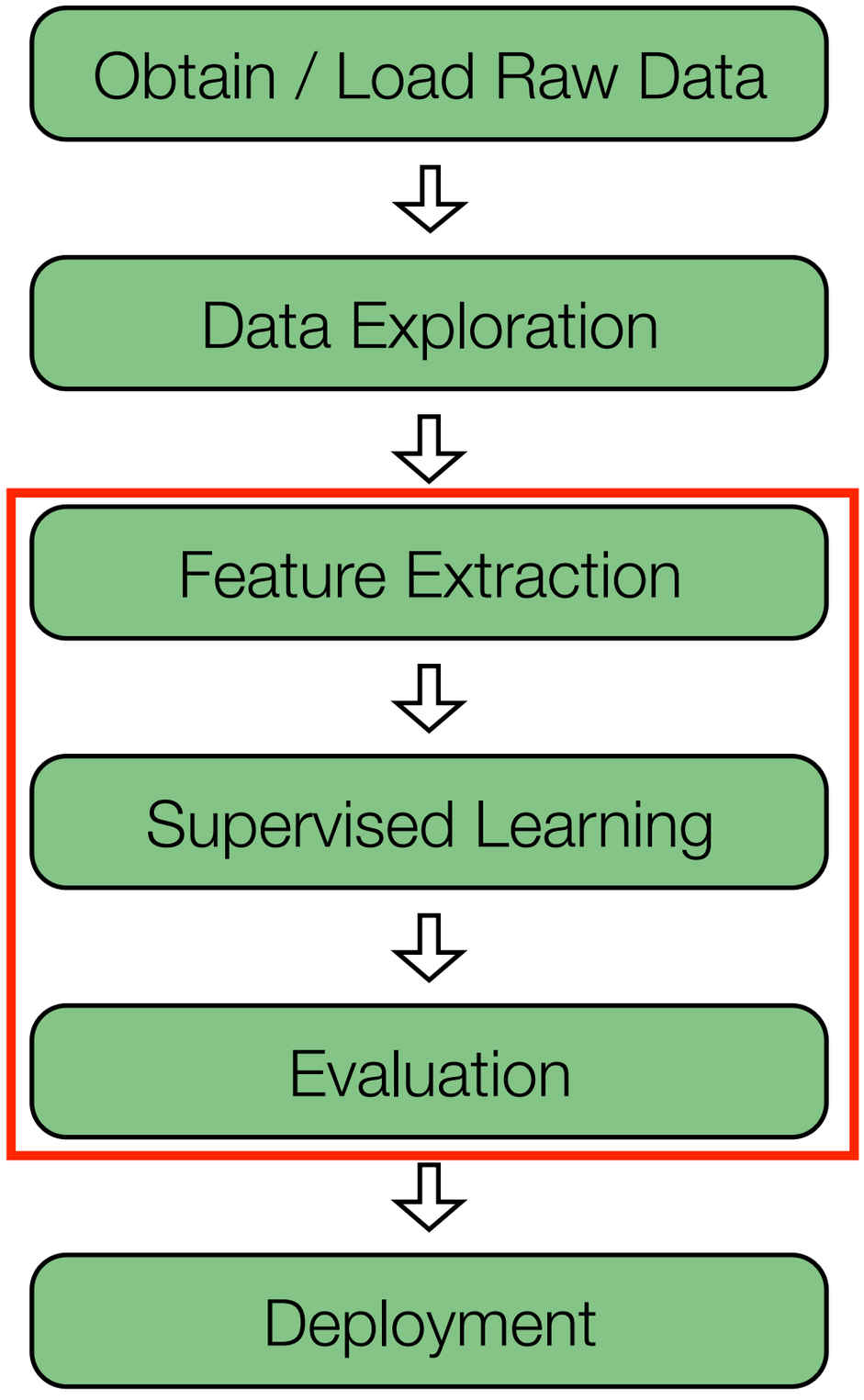
What's Next?


Obtain / Load Raw Data
Data Exploration
Feature Extraction
Supervised Learning
Evaluation
Deployment

What's Next?

Distributed ML principles

- What happens with Big Data?



What's Next?

Distributed ML principles

- What happens with Big Data?

Millionsong pipeline exercise

- Linear regression
- Gradient descent
- Model evaluation
- (Quadratic features)