



INTRODUCTION TO MACHINE LEARNING

Measuring model performance or error

Is our model any good?

- Context of task
 - Accuracy
 - Computation time
 - Interpretability
- 3 types of tasks
 - Classification
 - Regression
 - Clustering

Classification

- Accuracy and Error
- System is **right** or **wrong**
- Accuracy goes **up** when Error goes **down**

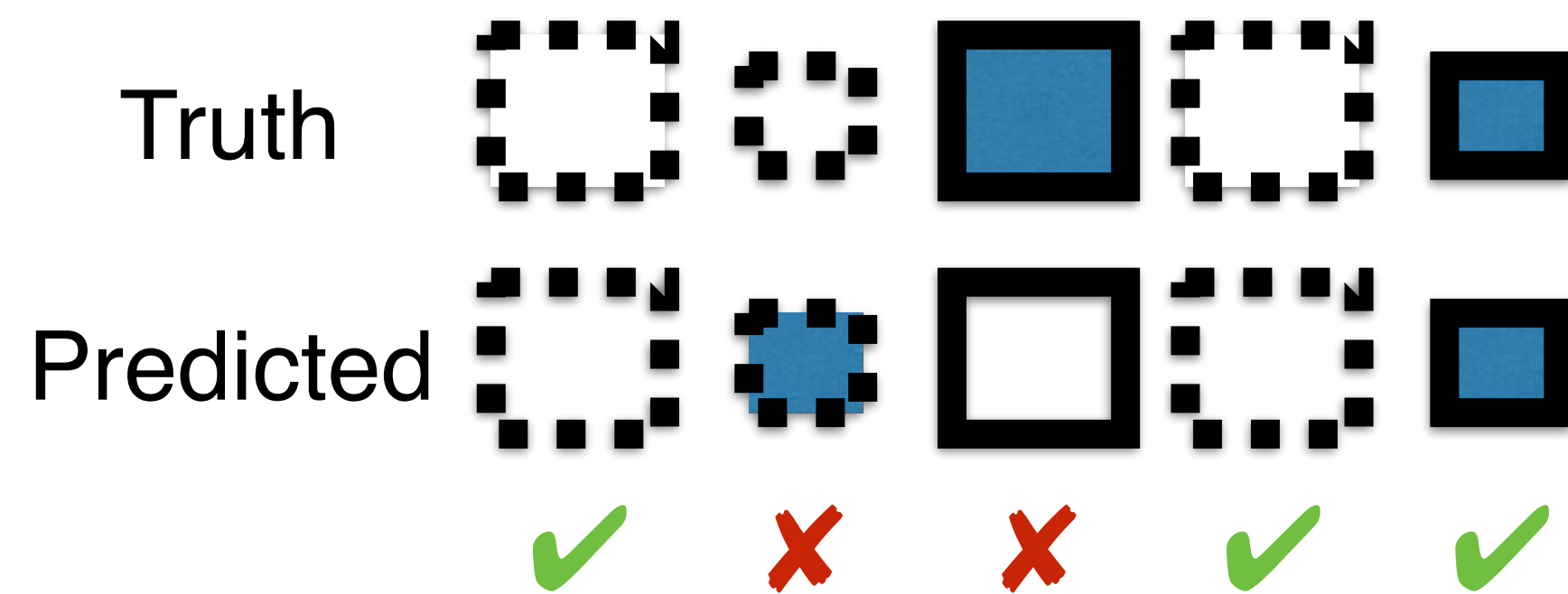
$$\text{Accuracy} = \frac{\text{correctly classified instances}}{\text{total amount of classified instances}}$$

$$\text{Error} = 1 - \text{Accuracy}$$

Example

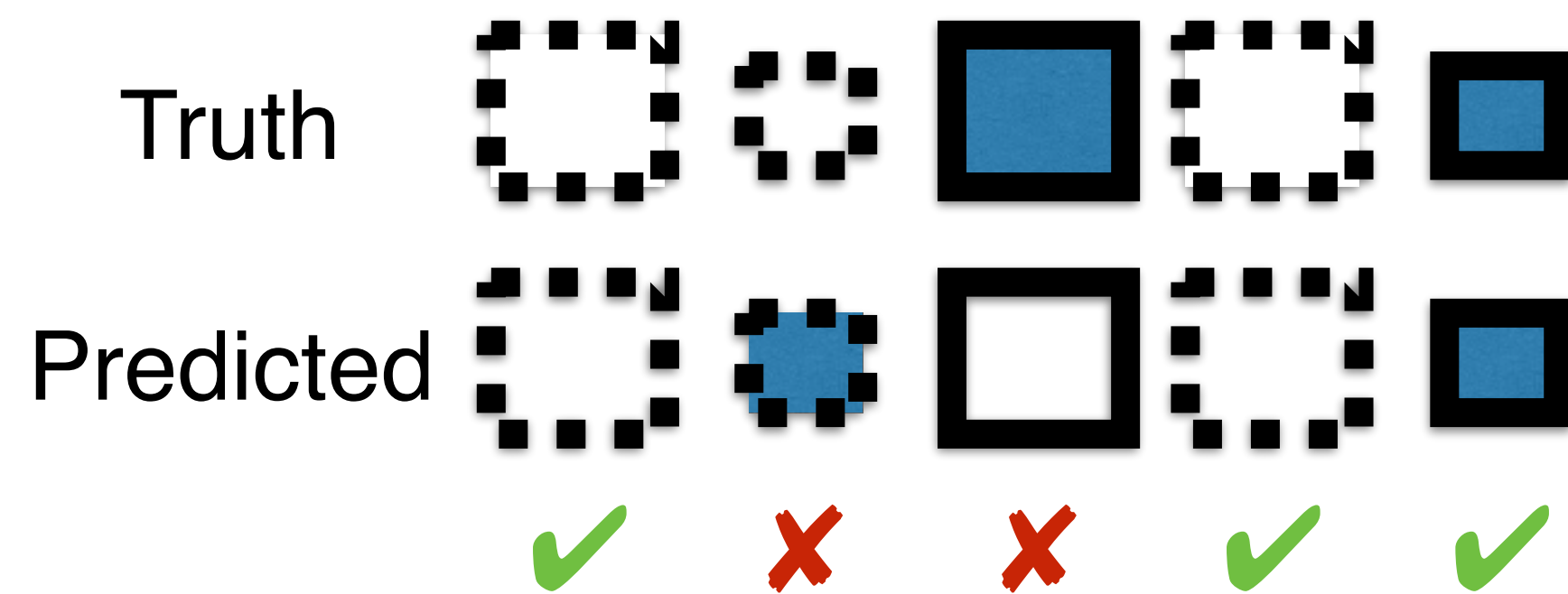
- Squares with 2 features: small/big and solid/dotted
- Label: colored/not colored
- Binary classification problem

Example



$$\frac{\begin{array}{ccccc} & \checkmark & \checkmark & \checkmark & \\ \hline \checkmark & \checkmark & \checkmark & \times & \times \end{array}}{5} = \frac{3}{5} = 60\%$$

Example



$$\frac{\begin{array}{ccccc} & \checkmark & \checkmark & \checkmark & \\ \hline \checkmark & \checkmark & \checkmark & \times & \times \end{array}}{5} = \frac{3}{5} = 60\%$$

Limits of accuracy

- Classifying very rare heart disease
- Classify all as negative (not sick)
- Predict 99 correct (not sick) and miss 1
- Accuracy: 99%
- Bogus... you miss every positive case!

Confusion matrix

- Rows and columns contain all available labels
- Each cell contains frequency of instances that are classified in a certain way

Confusion matrix

- Binary classifier: positive or negative (1 or 0)

		<i>Prediction</i>	
		P	N
<i>Truth</i>	p	TP	FN
	n	FP	TN

Confusion matrix

- Binary classifier: positive or negative (1 or 0)

True Positives
Prediction: P
Truth: P

		<i>Prediction</i>	
		P	N
<i>Truth</i>	p	TP	FN
	n	FP	TN

Confusion matrix

- Binary classifier: positive or negative (1 or 0)

True Negatives
Prediction: N
Truth: N

		<i>Prediction</i>	
		P	N
<i>Truth</i>	p	TP	FN
	n	FP	TN

Confusion matrix

- Binary classifier: positive or negative (1 or 0)

False Negatives
Prediction: N
Truth: P

		<i>Prediction</i>	
		P	N
<i>Truth</i>	p	TP	FN
	n	FP	TN

Confusion matrix

- Binary classifier: positive or negative (1 or 0)

False Positives
Prediction: P
Truth: N

		<i>Prediction</i>	
		P	N
<i>Truth</i>	p	TP	FN
	n	FP	TN

Ratios in the confusion matrix

- Accuracy
- Precision
- Recall

		<i>Prediction</i>	
		P	N
<i>Truth</i>	p	TP	FN
	n	FP	TN

Ratios in the confusion matrix

- Accuracy
- Precision
- Recall

Precision
 $TP/(TP+FP)$

		<i>Prediction</i>	
		P	N
<i>Truth</i>	p	TP	FN
	n	FP	TN

Ratios in the confusion matrix

- Accuracy
- Precision
- Recall

Precision
 $TP/(TP+FP)$

		<i>Prediction</i>	
		P	N
<i>Truth</i>	p	TP	FN
	n	FP	TN

Ratios in the confusion matrix

- Accuracy
- Precision
- Recall

Recall
 $TP/(TP+FN)$

		<i>Prediction</i>	
		P	N
<i>Truth</i>	p	TP	FN
	n	FP	TN

Ratios in the confusion matrix

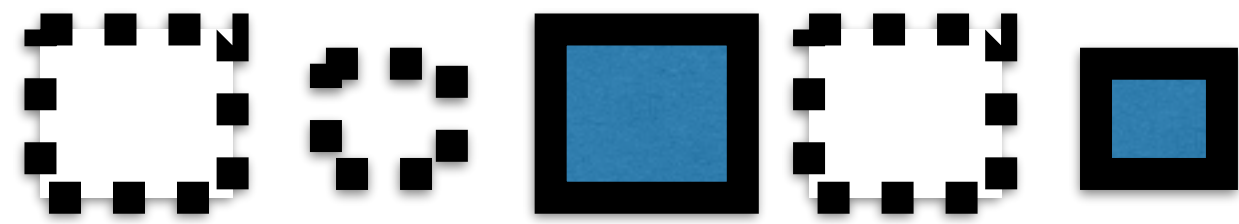
- Accuracy
- Precision
- Recall

Recall
 $TP/(TP+FN)$

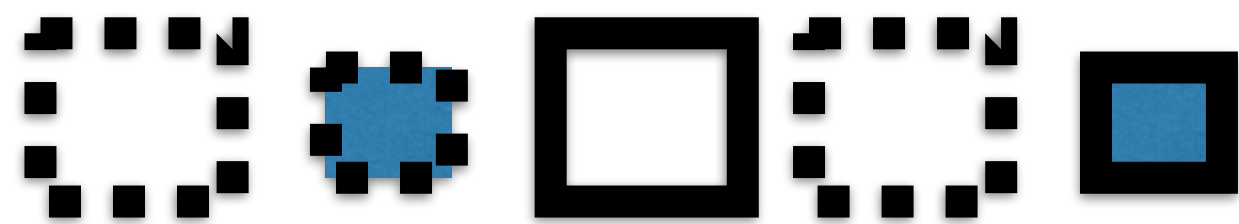
		<i>Prediction</i>	
		P	N
<i>Truth</i>	p	TP	FN
	n	FP	TN

Back to the squares

Truth

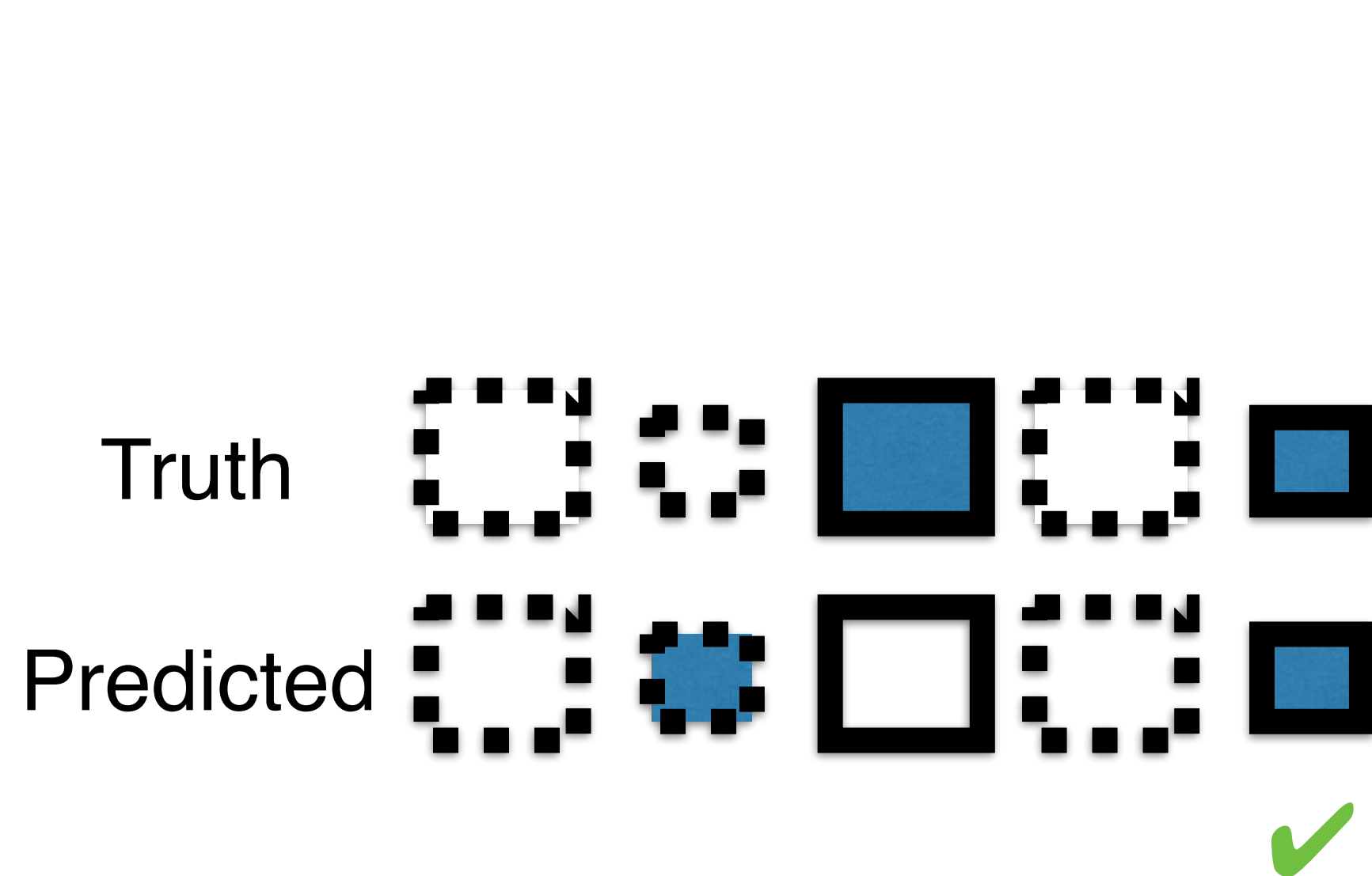


Predicted



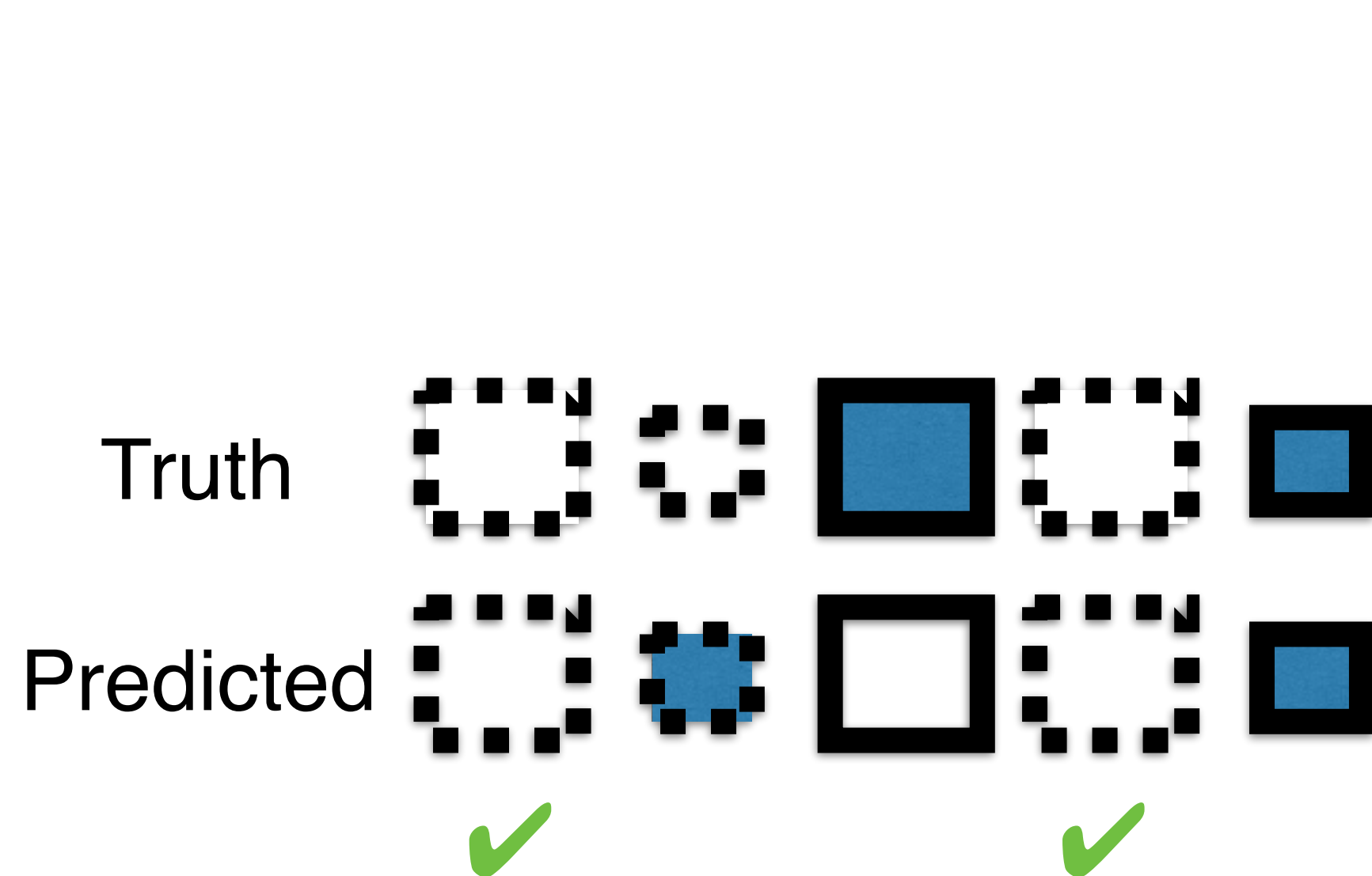
		<i>Prediction</i>	
		P	N
<i>Truth</i>	p	1	1
	n	1	2

Back to the squares



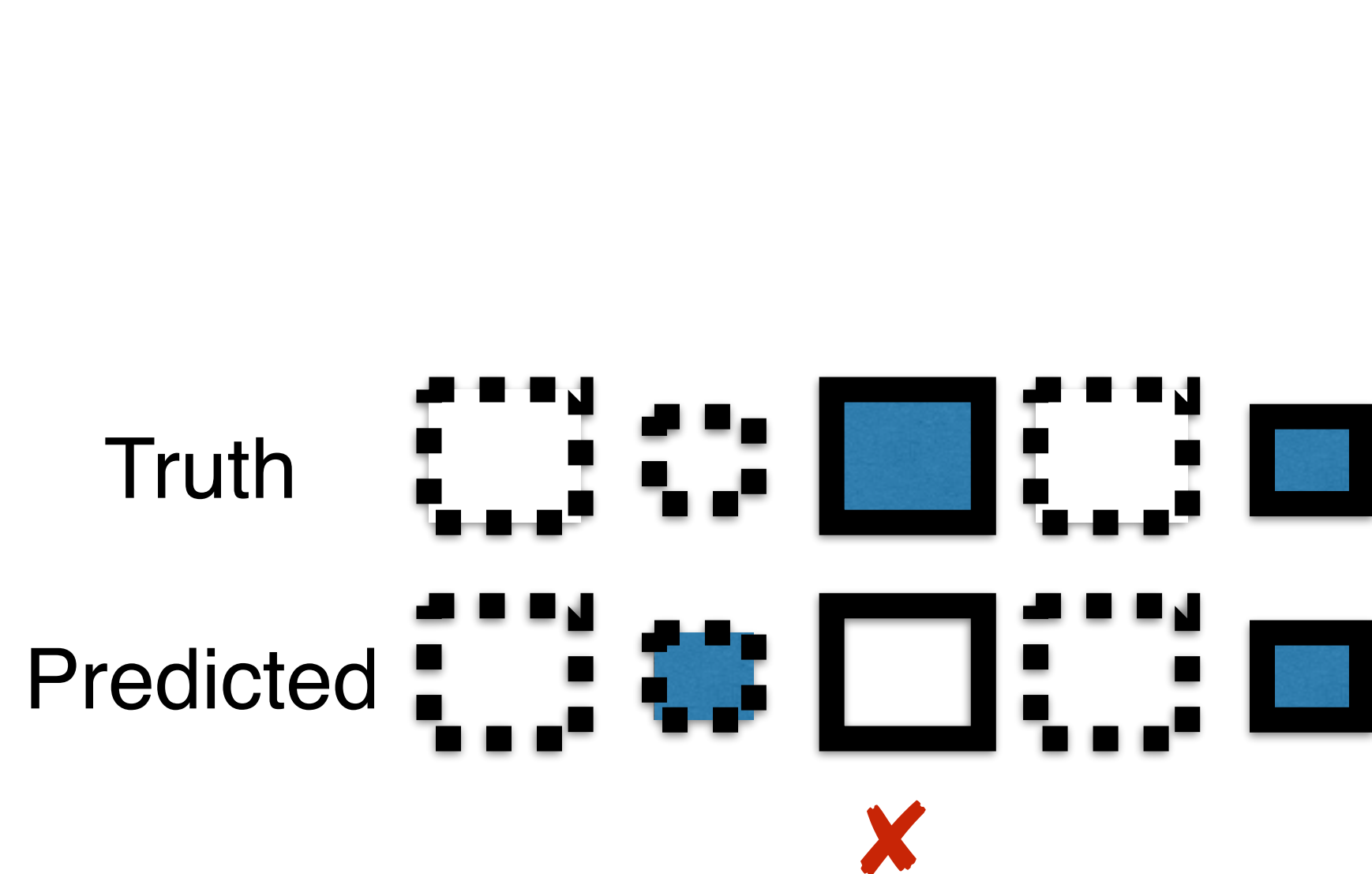
		<i>Prediction</i>	
		P	N
<i>Truth</i>	p	1	1
	n	1	2

Back to the squares



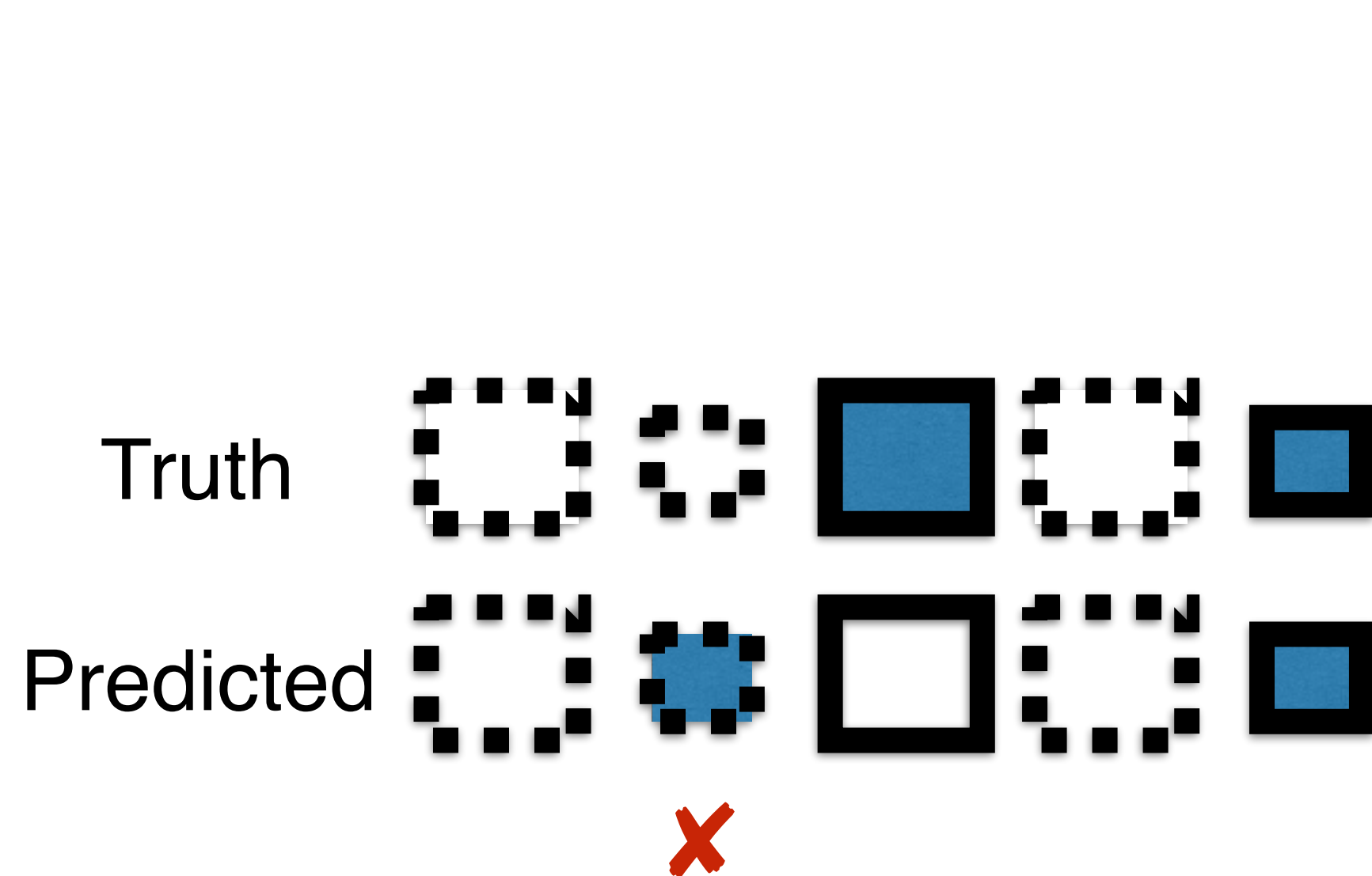
		<i>Prediction</i>	
		P	N
<i>Truth</i>	p	1	1
	n	1	2

Back to the squares



		Prediction	
		P	N
Truth	p	1	1
	n	1	2

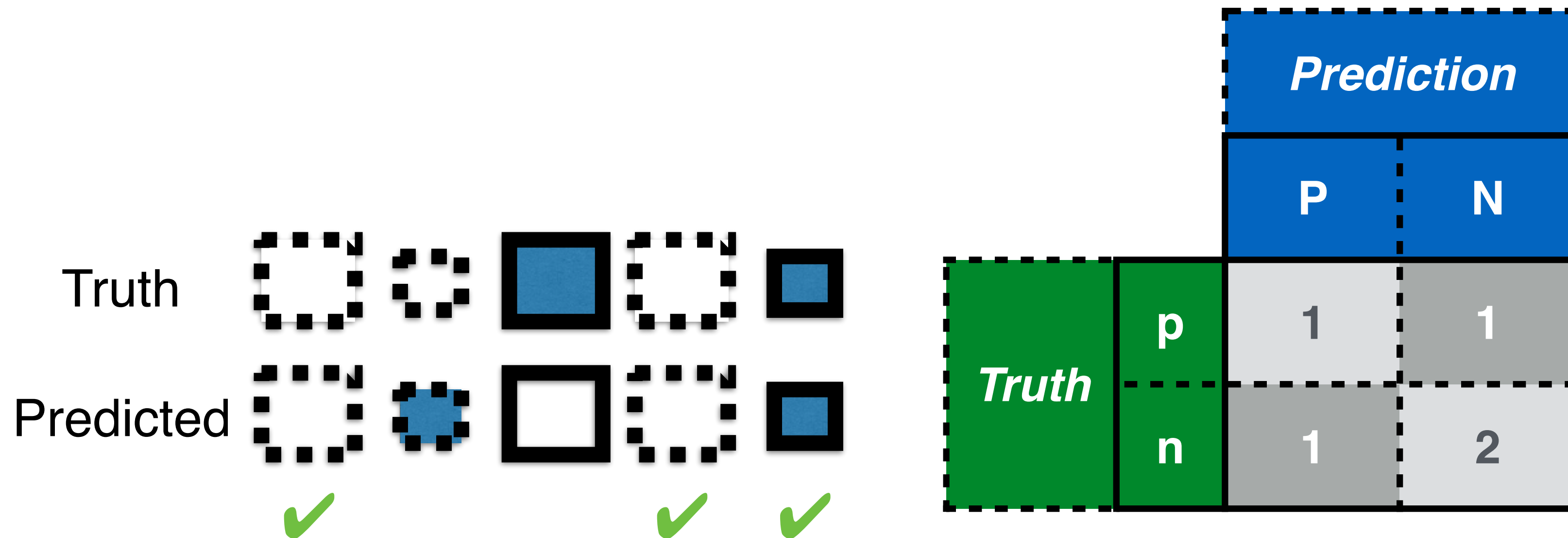
Back to the squares



		Prediction	
		P	N
Truth	p	1	1
	n	1	2

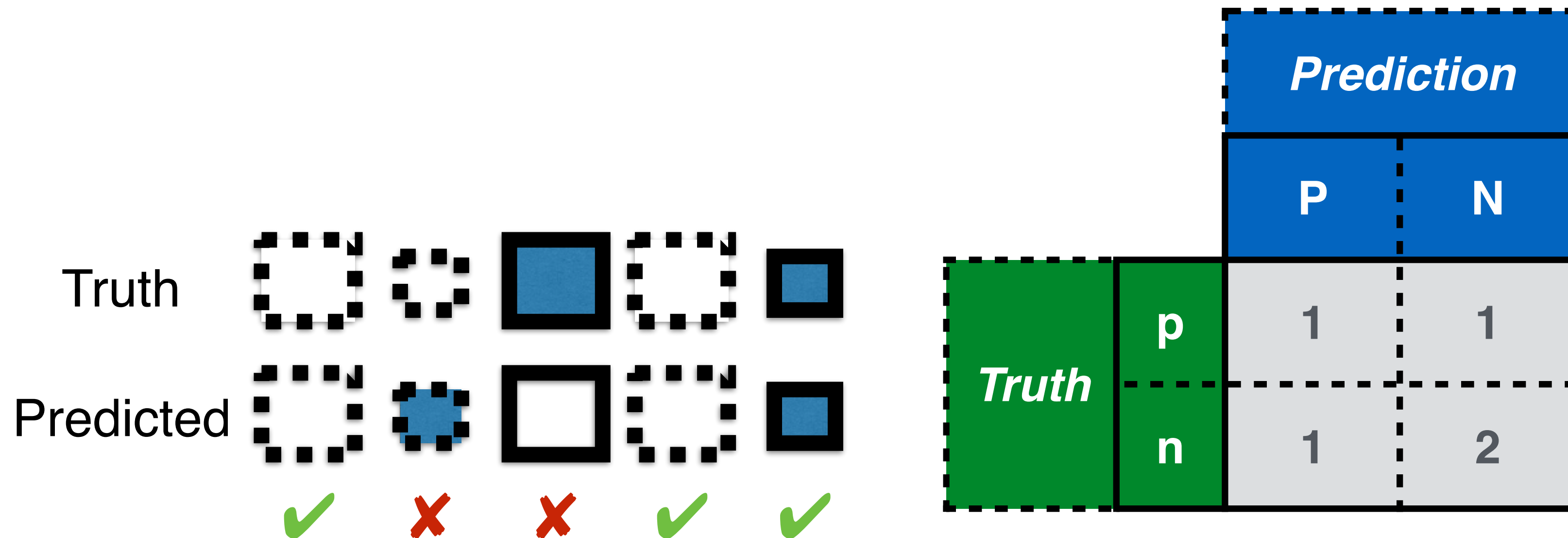
Back to the squares

- Accuracy: $(TP+TN)/(TP+FP+FN+TN) = (1+2)/(1+2+1+1) = 60\%$
- Precision: $TP/(TP+FP) = 1/(1+1) = 50\%$
- Recall: $TP/(TP+FN) = 1/(1+1) = 50\%$



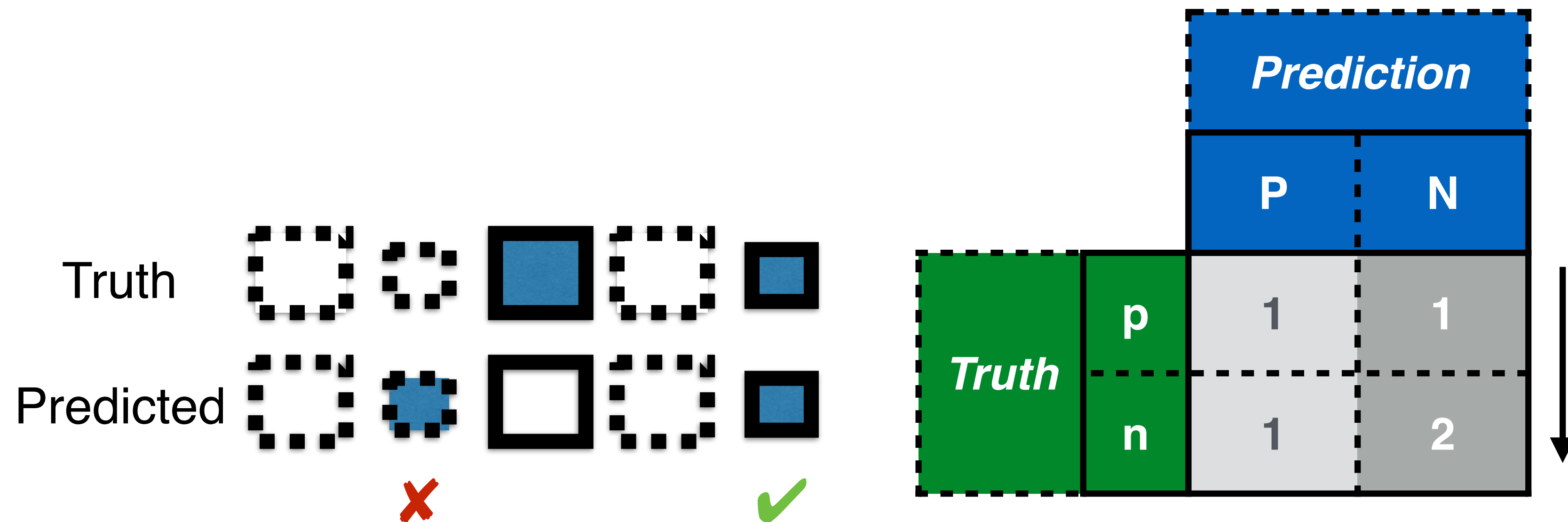
Back to the squares

- Accuracy: $(TP+TN)/(TP+FP+FN+TN) = (1+2)/(1+2+1+1) = 60\%$
- Precision: $TP/(TP+FP) = 1/(1+1) = 50\%$
- Recall: $TP/(TP+FN) = 1/(1+1) = 50\%$



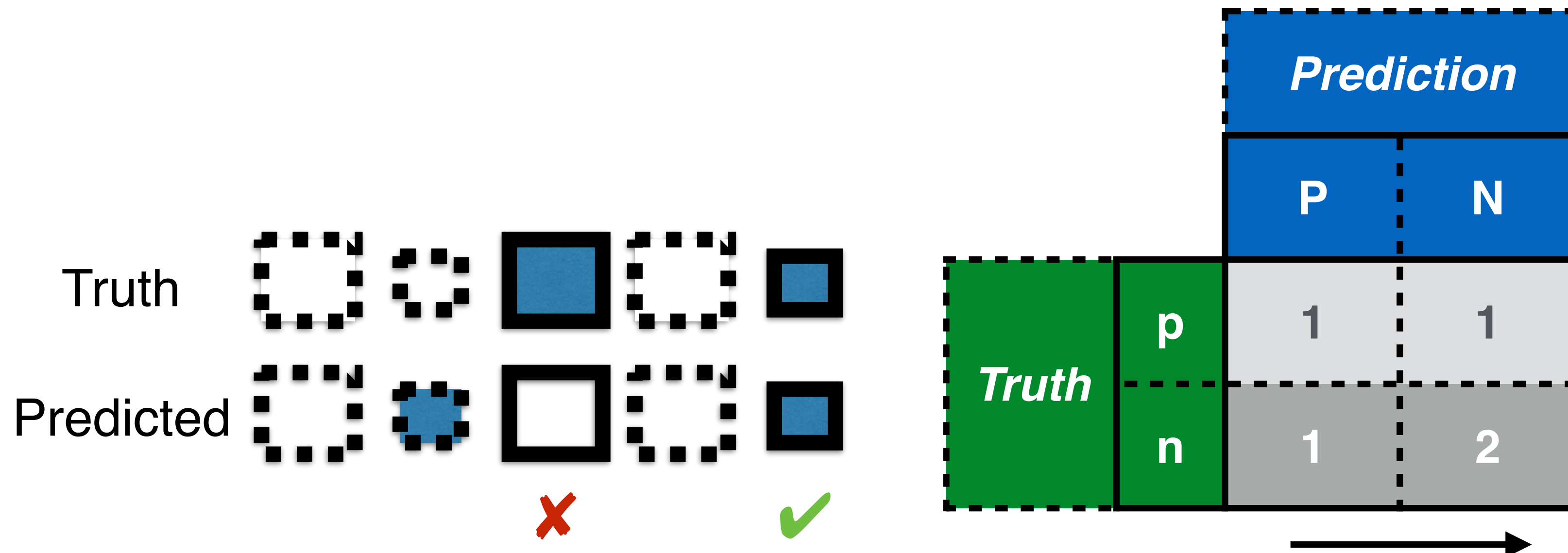
Back to the squares

- Accuracy: $(TP+TN)/(TP+FP+FN+TN) = (1+2)/(1+2+1+1) = 60\%$
- Precision: $TP/(TP+FP) = 1/(1+1) = 50\%$
- Recall: $TP/(TP+FN) = 1/(1+1) = 50\%$



Back to the squares

- Accuracy: $(TP+TN)/(TP+FP+FN+TN) = (1+2)/(1+2+1+1) = 60\%$
- Precision: $TP/(TP+FP) = 1/(1+1) = 50\%$
- Recall: $TP/(TP+FN) = 1/(1+1) = 50\%$



Rare heart disease

- Accuracy: $99/(99+1) = 99\%$
- Recall: $0/1 = 0\%$
- Precision: undefined — no positive predictions

		<i>Prediction</i>	
		P	N
<i>Truth</i>	p	0	1
	n	0	99

Regression: RMSE

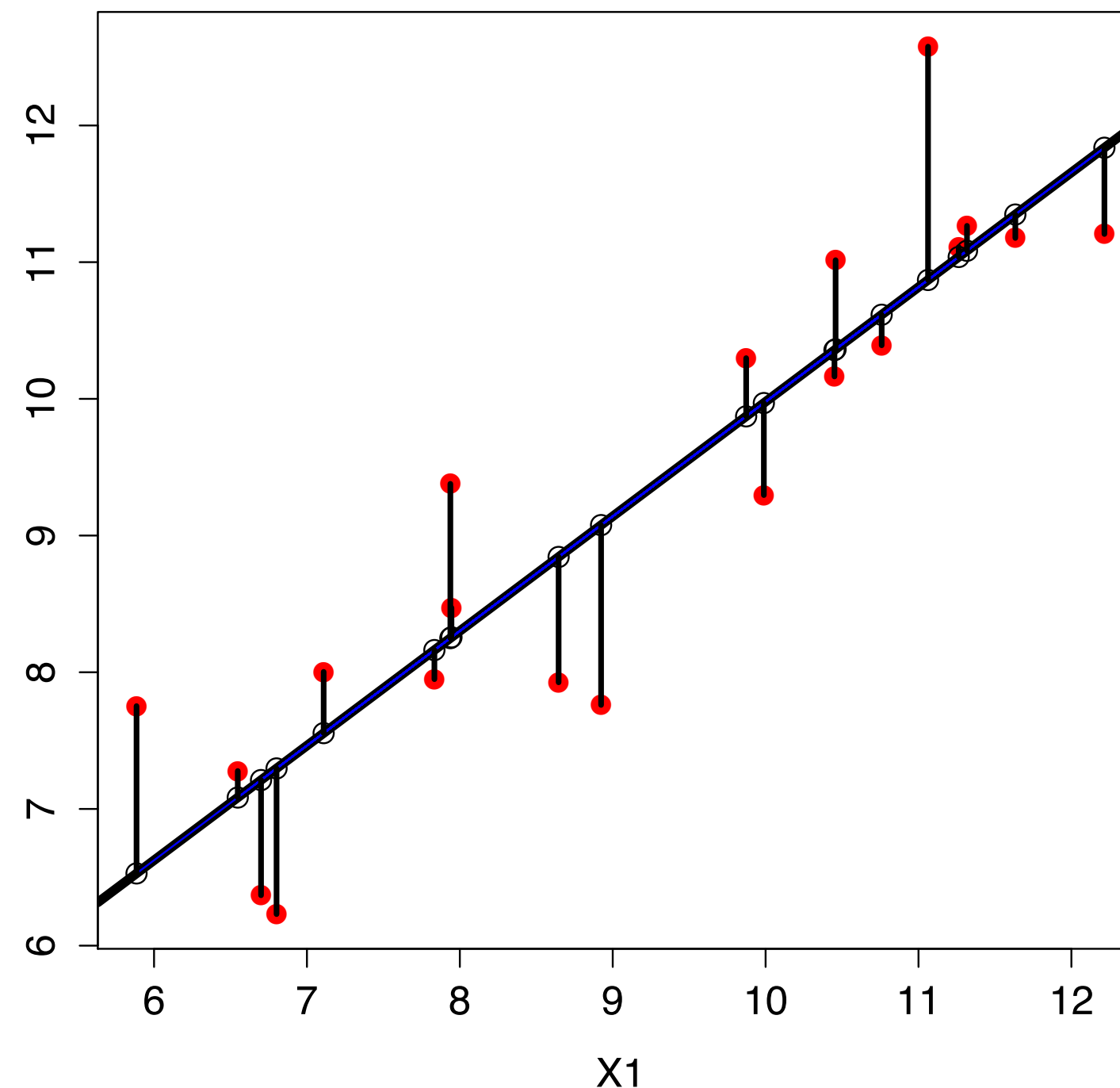
- Root Mean Squared Error (RMSE)
- Mean distance between estimates and regression line

$$\text{RMSE} = \sqrt{\frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2}$$

y_i : actual outcome for obs. i

\hat{y}_i : predicted outcome for obs. i

N : Number of observations



Clustering

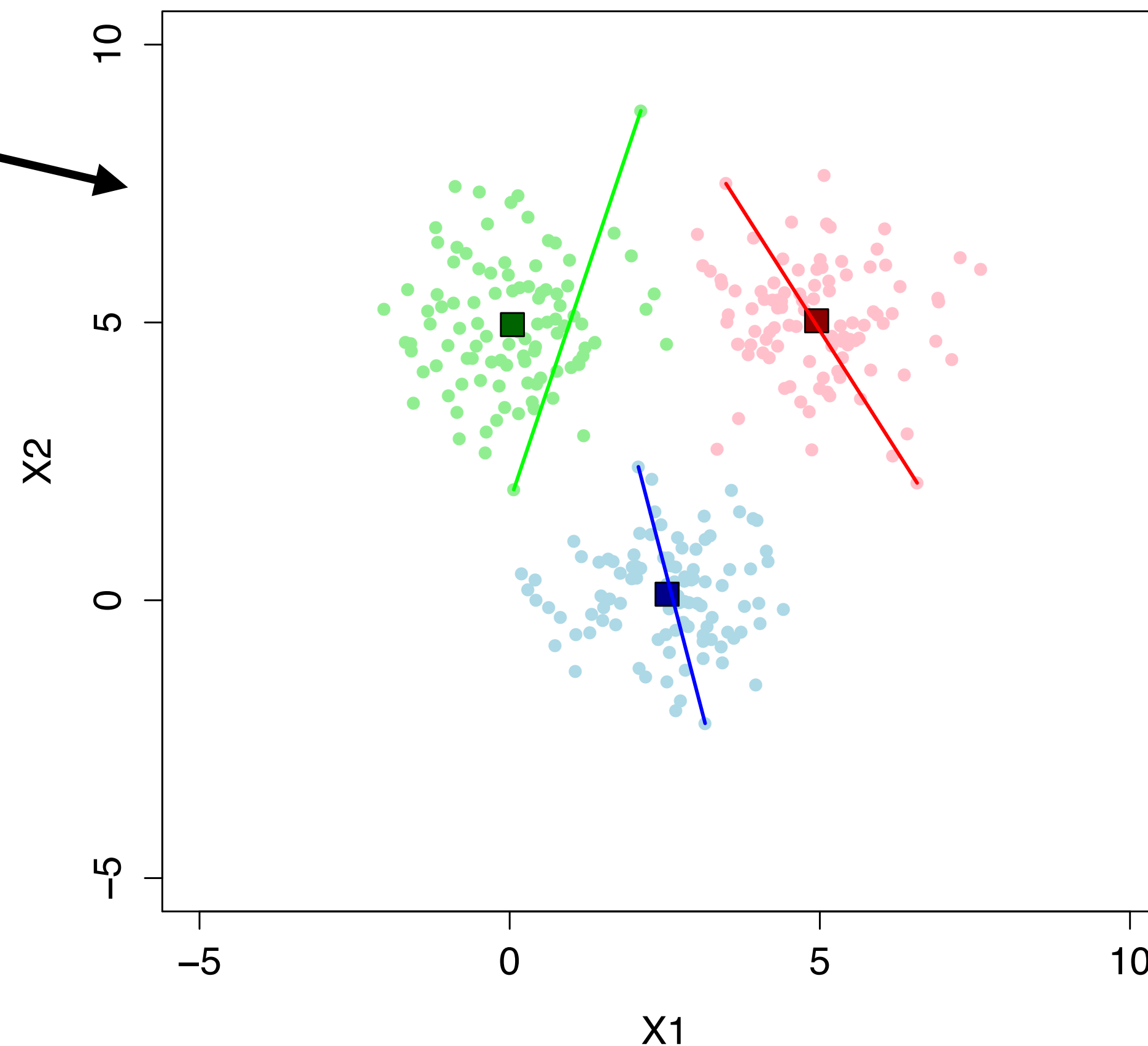
- No label information
- Need distance metric between points

Clustering

- Performance measure consists of 2 elements
 - Similarity within each cluster ↑
 - Similarity between clusters ↓

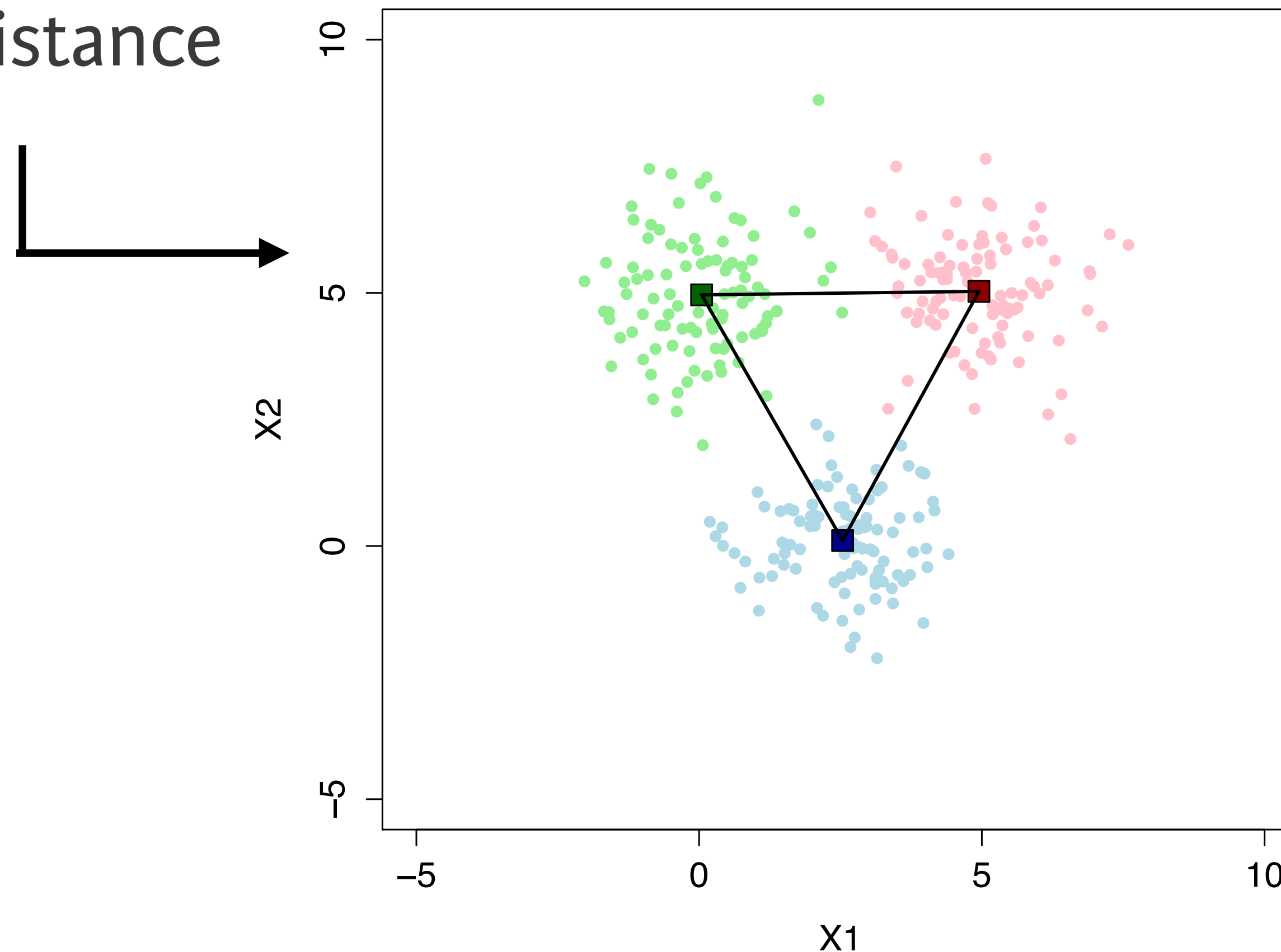
Within cluster similarity

- Within sum of squares (WSS)
- Diameter
- Minimize



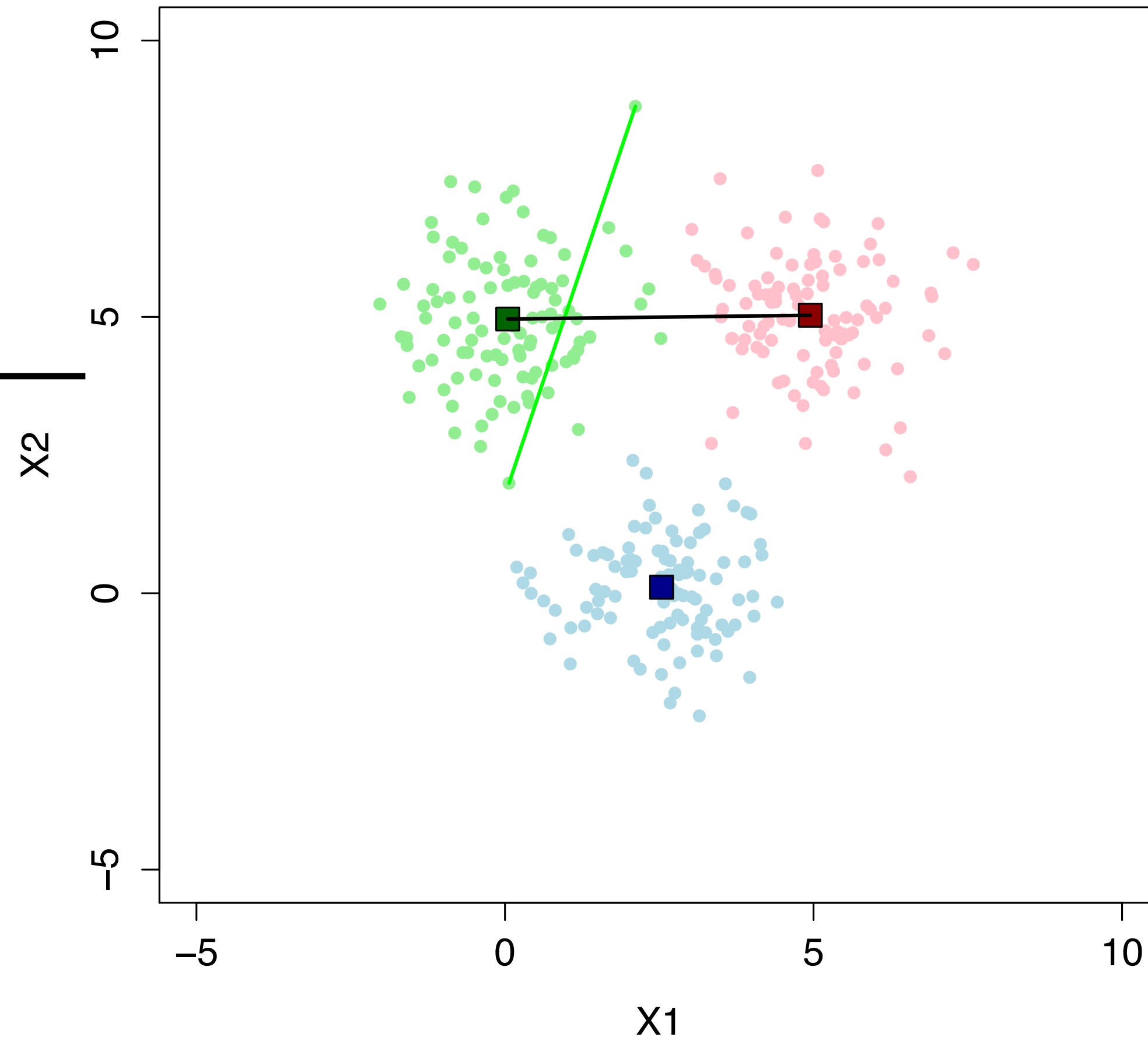
Between cluster similarity

- Between cluster sum of squares (BSS)
- Intercluster distance
- Maximize



Dunn's index

minimal intercluster distance
—————
maximal diameter





INTRODUCTION TO MACHINE LEARNING

Let's practice!



INTRODUCTION TO MACHINE LEARNING

Training set and test set

Machine learning - statistics

- Predictive power vs. descriptive power
- **Supervised learning:** model must predict
 - unseen observations
- **Classical statistics:** model must fit data
 - explain or describe data

Predictive model

- Training
 - not on complete dataset
 - training set
- Test set to evaluate performance of model
- Sets are disjoint: **NO OVERLAP**
- Model tested on unseen observations
 - Generalization!

Split the dataset

- N instances (=observations): X
- K features: F
- Class labels: y

	f_1	f_2	...	f_K	y
x_1	$x_{1,1}$	$x_{1,2}$...	$x_{1,K}$	y_1
x_2	$x_{2,1}$	$x_{2,2}$...	$x_{2,K}$	y_2
...
x_r	$x_{r,1}$	$x_{r,2}$...	$x_{r,K}$	y_r
x_{r+1}	$x_{r+1,1}$	$x_{r+1,2}$...	$x_{r+1,K}$	y_{r+1}
x_{r+2}	$x_{r+2,1}$	$x_{r+2,2}$...	$x_{r+2,K}$	y_{r+2}
...
x_N	$x_{N,1}$	$x_{N,2}$...	$x_{N,K}$	y_N

Training set

Test set

Split the dataset

- N instances (=observations): X
- K features: F
- Class labels: y

	f_1	f_2	...	f_K	y
X_1	$X_{1,1}$	$X_{1,2}$...	$X_{1,K}$	y_1
X_2	$X_{2,1}$	$X_{2,2}$...	$X_{2,K}$	y_2
...
X_r	$X_{r,1}$	$X_{r,2}$...	$X_{r,K}$	y_r
X_{r+1}	$X_{r+1,1}$	$X_{r+1,2}$...	$X_{r+1,K}$	y_{r+1}
X_{r+2}	$X_{r+2,1}$	$X_{r+2,2}$...	$X_{r+2,K}$	y_{r+2}
...
X_N	$X_{N,1}$	$X_{N,2}$...	$X_{N,K}$	y_N

Training set

Test set

Split the dataset

- N instances (=observations): X
- K features: F
- Class labels: y

	f_1	f_2	...	f_K	y
X_1	$X_{1,1}$	$X_{1,2}$...	$X_{1,K}$	y_1
X_2	$X_{2,1}$	$X_{2,2}$...	$X_{2,K}$	y_2
...
X_r	$X_{r,1}$	$X_{r,2}$...	$X_{r,K}$	y_r
X_{r+1}	$X_{r+1,1}$	$X_{r+1,2}$...	$X_{r+1,K}$	y_{r+1}
X_{r+2}	$X_{r+2,1}$	$X_{r+2,2}$...	$X_{r+2,K}$	y_{r+2}
...
X_N	$X_{N,1}$	$X_{N,2}$...	$X_{N,K}$	y_N

Training set

Test set

Split the dataset

- N instances (=observations): X
- K features: F
- Class labels: y

	f_1	f_2	...	f_K	y
x_1	$x_{1,1}$	$x_{1,2}$...	$x_{1,K}$	y_1
x_2	$x_{2,1}$	$x_{2,2}$...	$x_{2,K}$	y_2
...
x_r	$x_{r,1}$	$x_{r,2}$...	$x_{r,K}$	y_r
x_{r+1}	$x_{r+1,1}$	$x_{r+1,2}$...	$x_{r+1,K}$	y_{r+1}
x_{r+2}	$x_{r+2,1}$	$x_{r+2,2}$...	$x_{r+2,K}$	y_{r+2}
...
x_N	$x_{N,1}$	$x_{N,2}$...	$x_{N,K}$	y_N

Training set

Test set

Split the dataset

	f_1	f_2	...	f_K	y
\mathbf{x}_1	$x_{1,1}$	$x_{1,2}$...	$x_{1,K}$	y_1
\mathbf{x}_2	$x_{2,1}$	$x_{2,2}$...	$x_{2,K}$	y_2
...
\mathbf{x}_r	$x_{r,1}$	$x_{r,2}$...	$x_{r,K}$	y_r
\mathbf{x}_{r+1}	$x_{r+1,1}$	$x_{r+1,2}$...	$x_{r+1,K}$	y_{r+1}
\mathbf{x}_{r+2}	$x_{r+2,1}$	$x_{r+2,2}$...	$x_{r+2,K}$	y_{r+2}
...
\mathbf{x}_N	$x_{N,1}$	$x_{N,2}$...	$x_{N,K}$	y_N

Training set

Test set

Split the dataset

	f_1	f_2	...	f_K	y
\mathbf{x}_1	$x_{1,1}$	$x_{1,2}$...	$x_{1,K}$	y_1
\mathbf{x}_2	$x_{2,1}$	$x_{2,2}$...	$x_{2,K}$	y_2
...
\mathbf{x}_r	$x_{r,1}$	$x_{r,2}$...	$x_{r,K}$	y_r
\mathbf{x}_{r+1}	$x_{r+1,1}$	$x_{r+1,2}$...	$x_{r+1,K}$	y_{r+1}
\mathbf{x}_{r+2}	$x_{r+2,1}$	$x_{r+2,2}$...	$x_{r+2,K}$	y_{r+2}
...
\mathbf{x}_N	$x_{N,1}$	$x_{N,2}$...	$x_{N,K}$	y_N

Training set

Test set

Use to predict y : \hat{y}

Split the dataset

	f_1	f_2	...	f_K	y
\mathbf{x}_1	$x_{1,1}$	$x_{1,2}$...	$x_{1,K}$	y_1
\mathbf{x}_2	$x_{2,1}$	$x_{2,2}$...	$x_{2,K}$	y_2
...
\mathbf{x}_r	$x_{r,1}$	$x_{r,2}$...	$x_{r,K}$	y_r
\mathbf{x}_{r+1}	$x_{r+1,1}$	$x_{r+1,2}$...	$x_{r+1,K}$	y_{r+1}
\mathbf{x}_{r+2}	$x_{r+2,1}$	$x_{r+2,2}$...	$x_{r+2,K}$	y_{r+2}
...
\mathbf{x}_N	$x_{N,1}$	$x_{N,2}$...	$x_{N,K}$	y_N

Training set

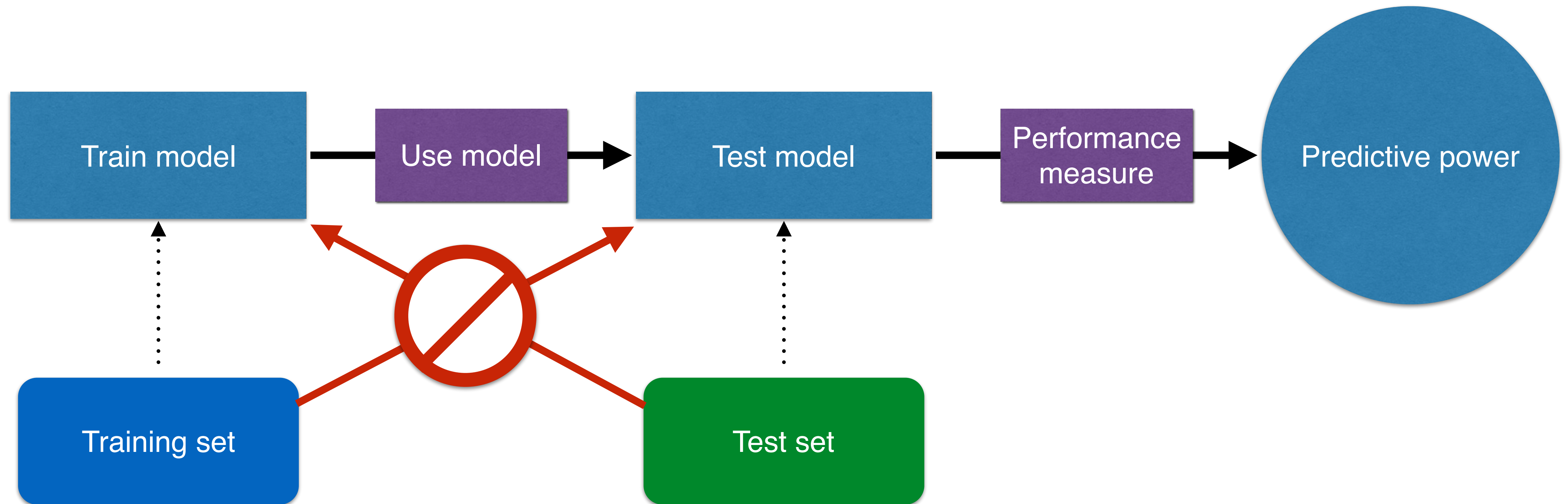
Test set

Use to predict y : \hat{y} \longleftrightarrow real y
compare them

When to use training/test set?

- Supervised learning
- Not for unsupervised (clustering)
 - Data not labeled

Predictive power of model



How to split the sets?

- Which observations go where?
- Training set larger test set
- Typically about 3/1
- Quite arbitrary
- **Generally:** more data = better model
- Test set not too small

Distribution of the sets

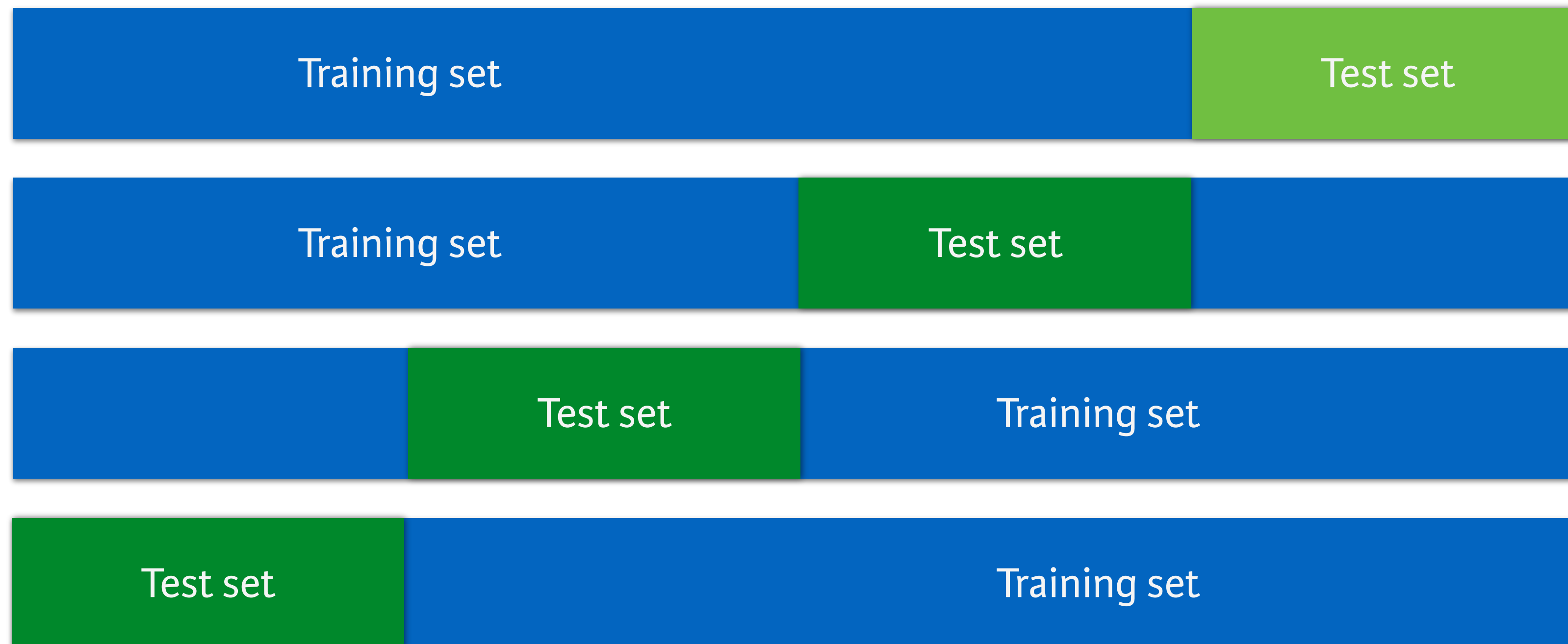
- **Classification**
 - classes must have similar distributions
 - avoid a class not being available in a set
- **Classification & regression**
 - shuffle dataset before splitting

Effect of sampling

- Sampling can affect performance measures
- Add robustness to these measures: **cross-validation**
- **Idea:** sample multiple times, with different separations

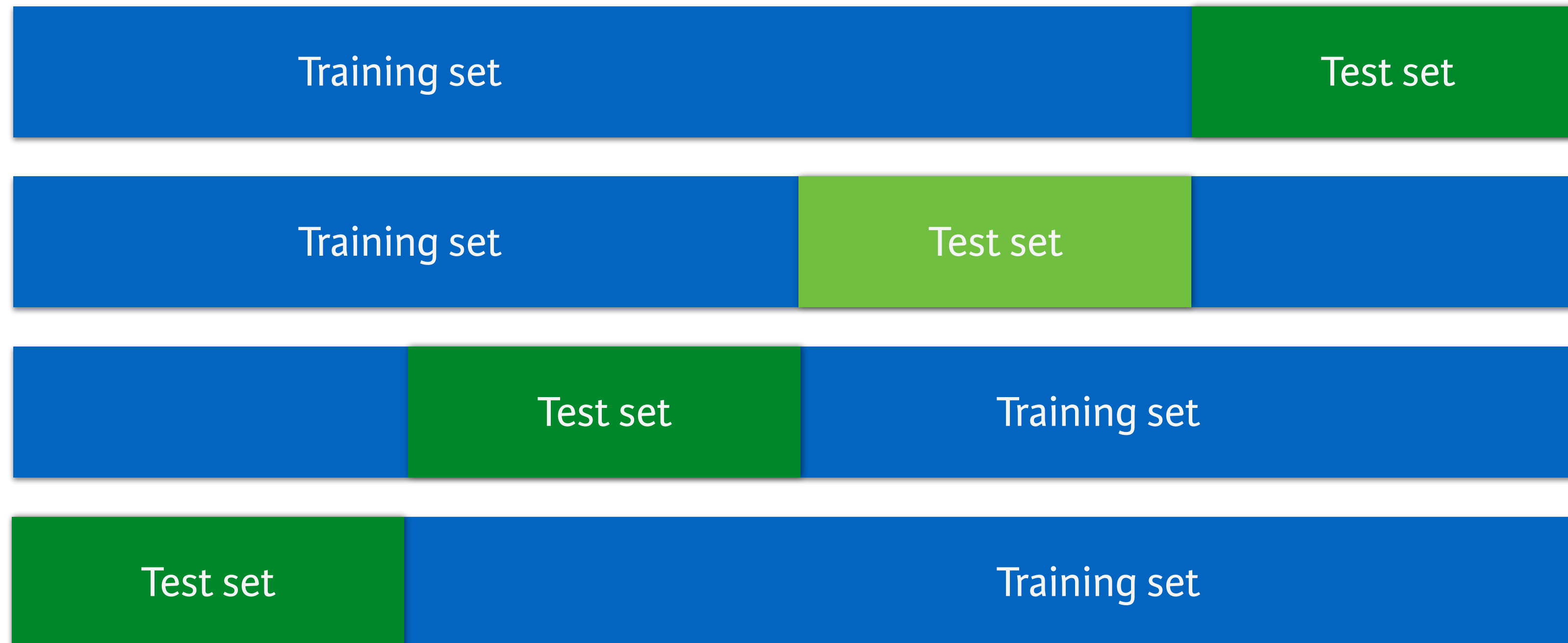
Cross-validation

- E.g.: 4-fold cross-validation



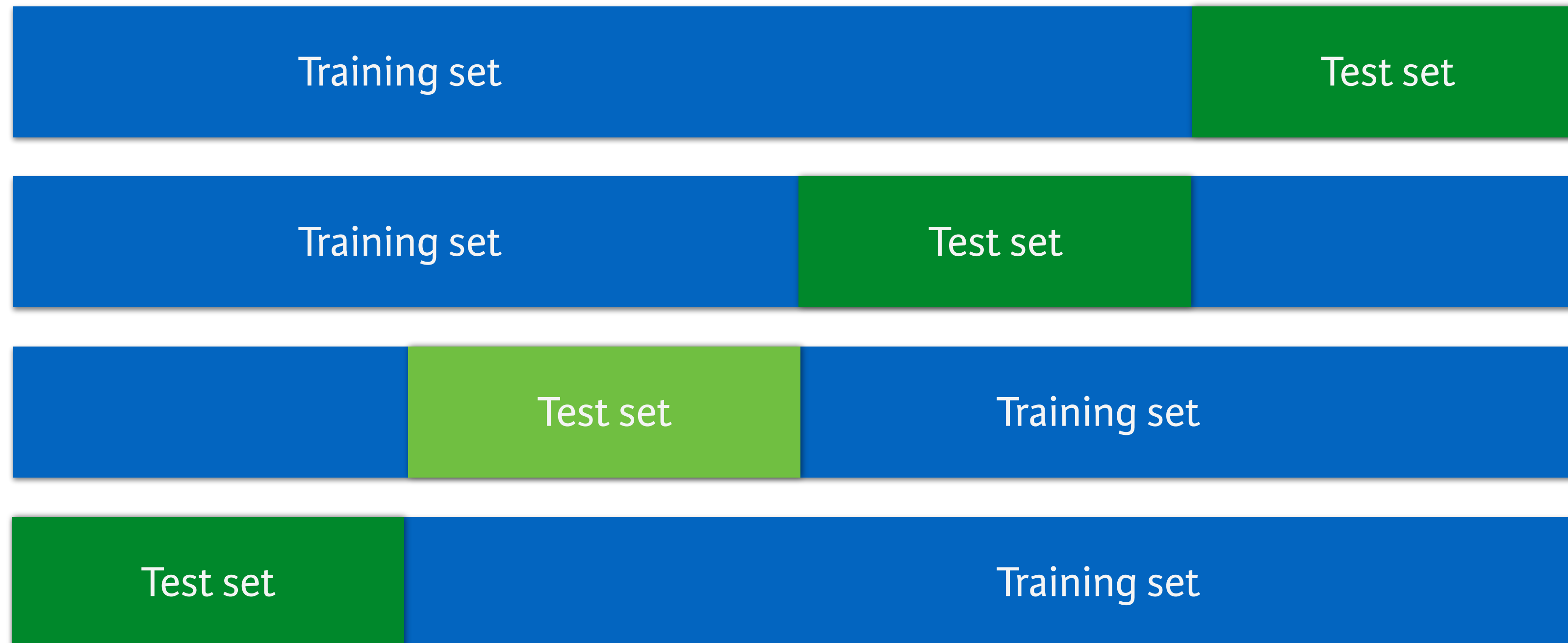
Cross-validation

- E.g.: 4-fold cross-validation



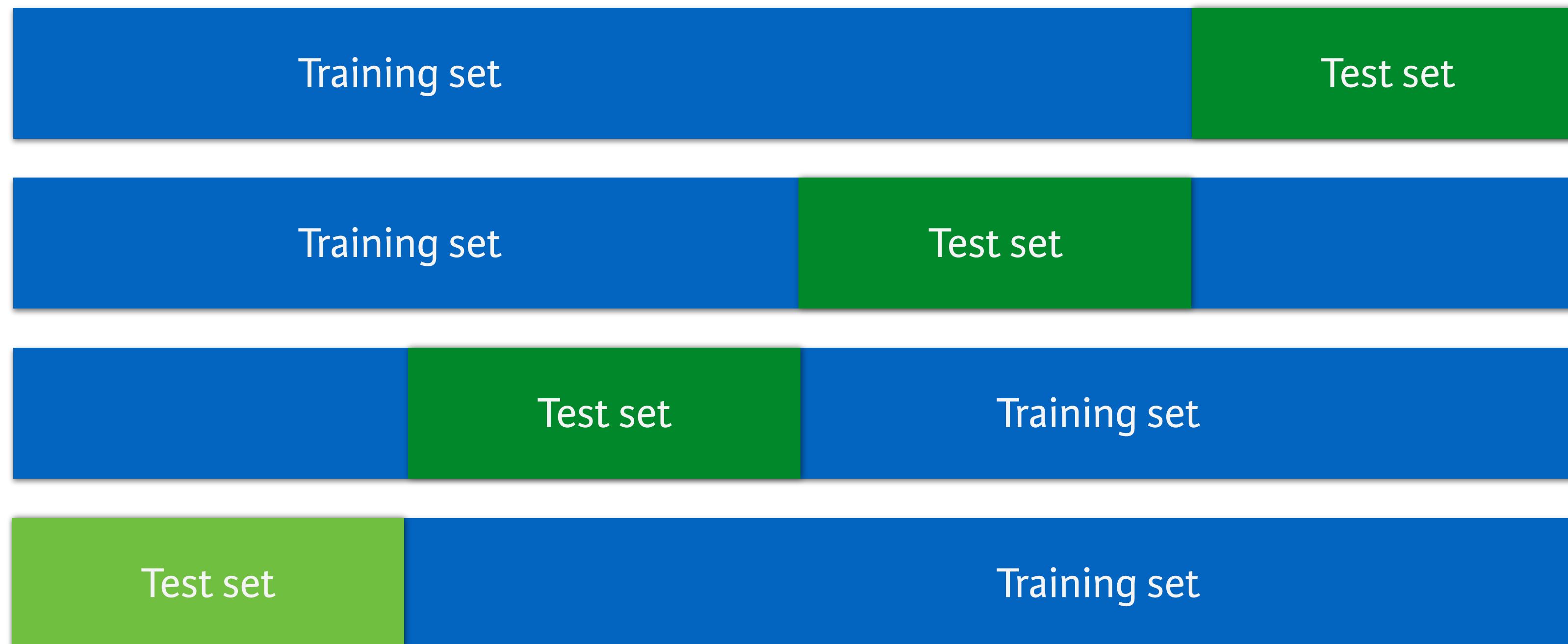
Cross-validation

- E.g.: 4-fold cross-validation



Cross-validation

- E.g.: 4-fold cross-validation



aggregate results for robust measure

n-fold cross-validation

- Fold test set over dataset n times
- Each test set is $1/n$ size of total dataset



INTRODUCTION TO MACHINE LEARNING

Let's practice!



INTRODUCTION TO MACHINE LEARNING

Bias and Variance

What you've learned?

- Accuracy and other performance measures
- Training and test set

Knitting it all together

- Effect of splitting dataset (train/test) on accuracy
- Over- and underfitting

Introducing

BIAS

VARIANCE

Bias and Variance

- Main goal of supervised learning: **prediction**
- **Prediction error** ~ reducible + irreducible error

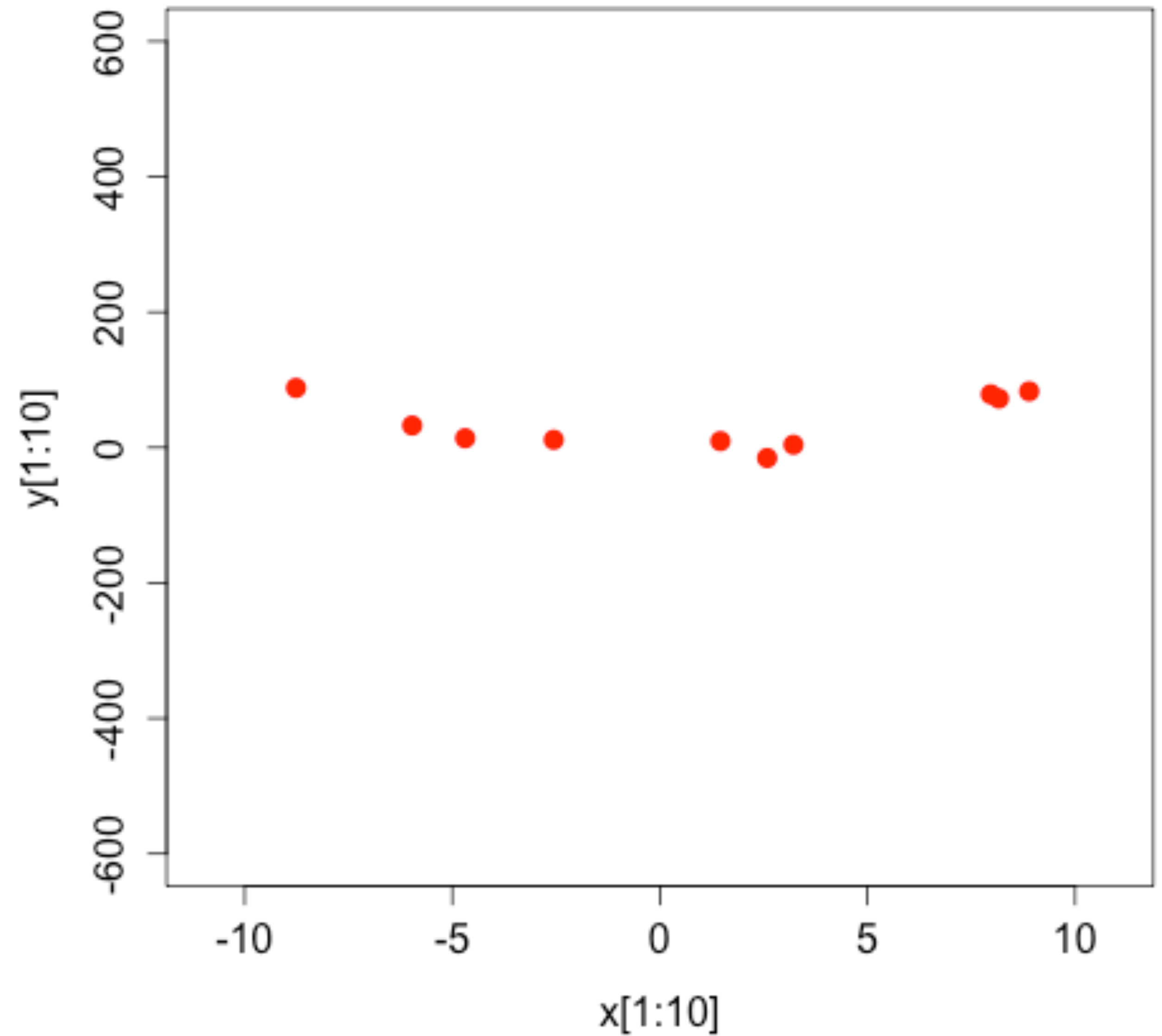
Irreducible - reducible error

- Irreducible: noise — don't minimize
- Reducible: error due to unfit model — minimize
- Reducible error is split into **bias** and **variance**

Bias

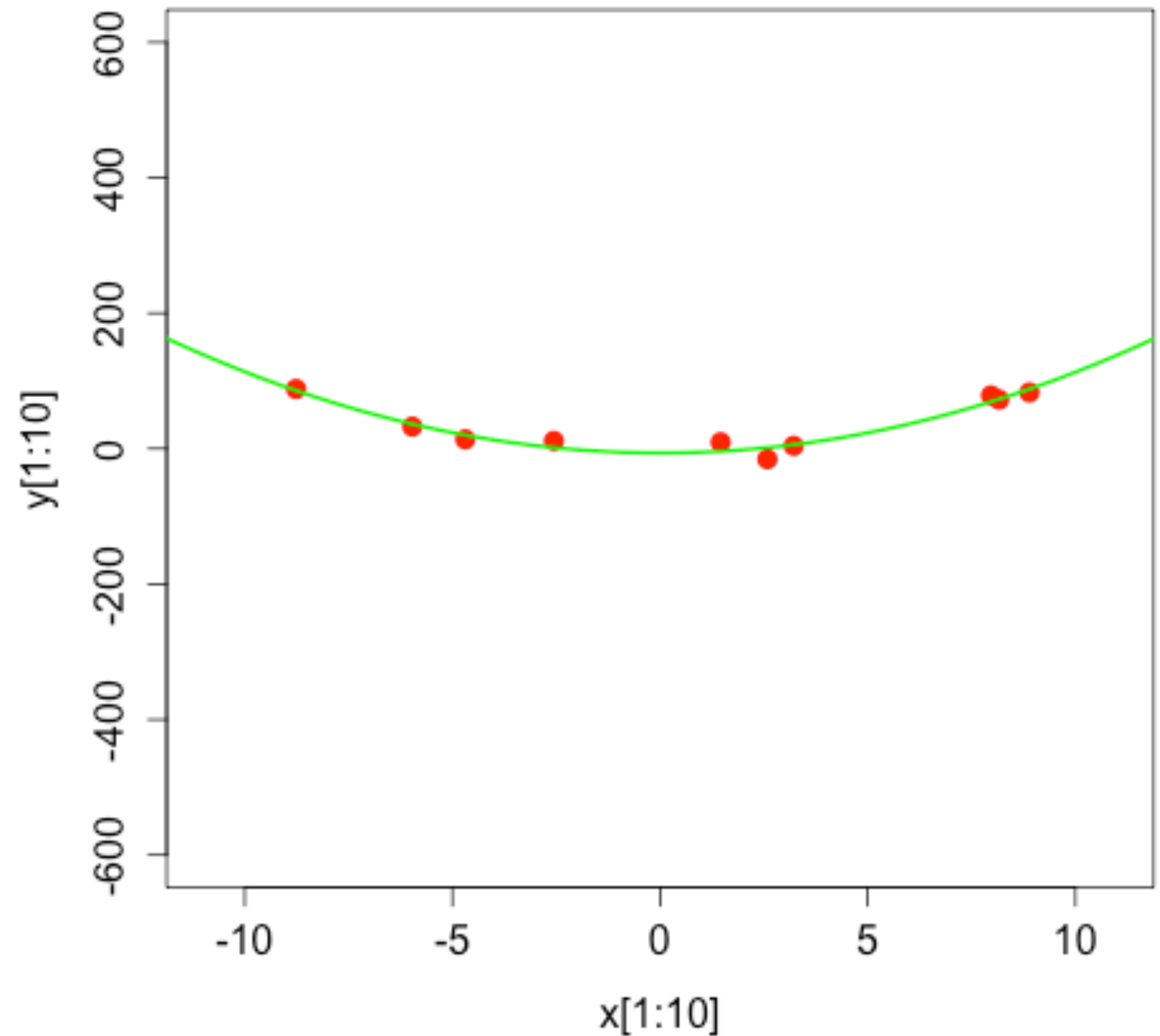
- Error due to **bias**: wrong assumptions
- Difference predictions and truth
 - using models trained by specific learning algorithm

Example



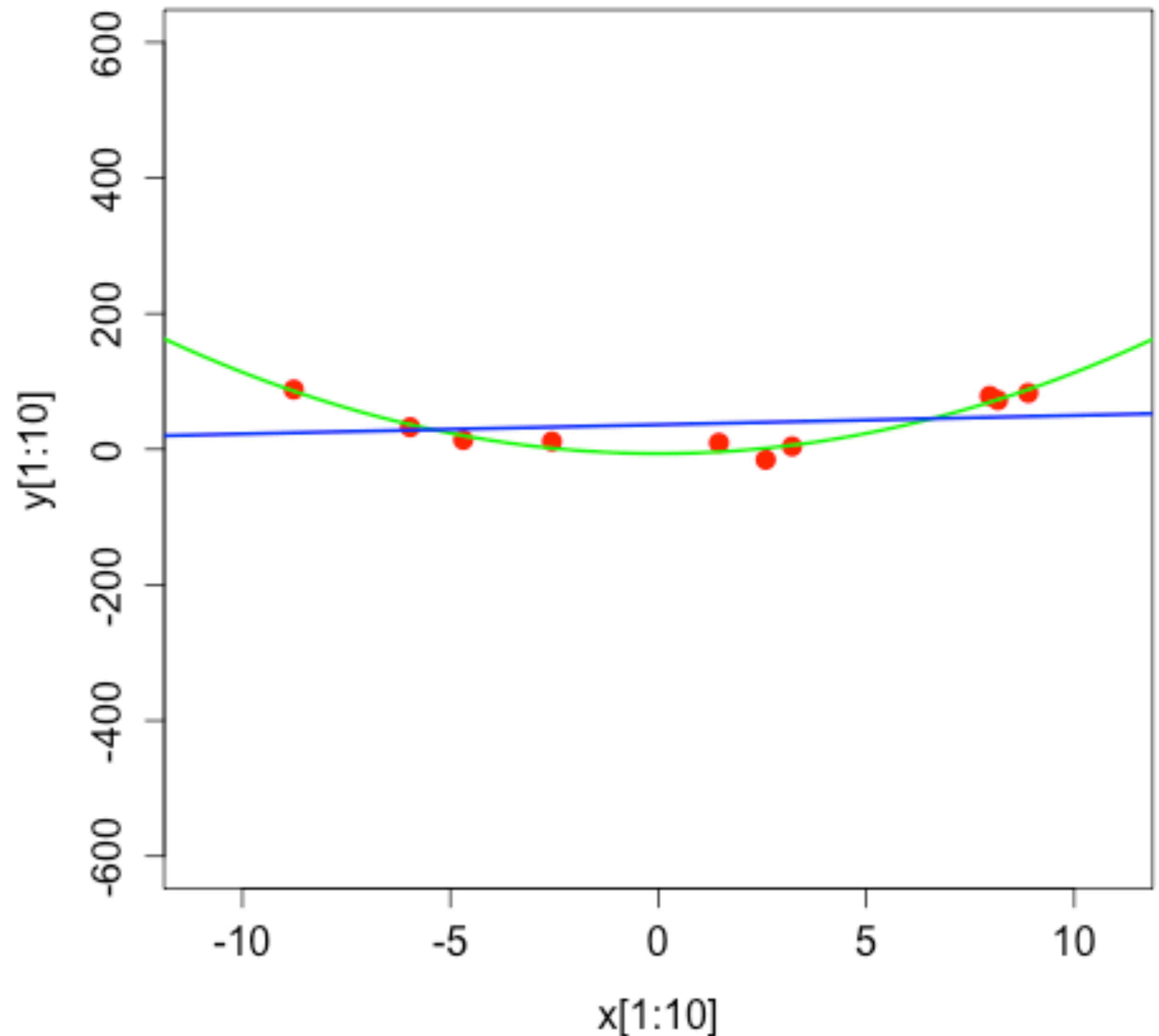
Example

- Quadratic data



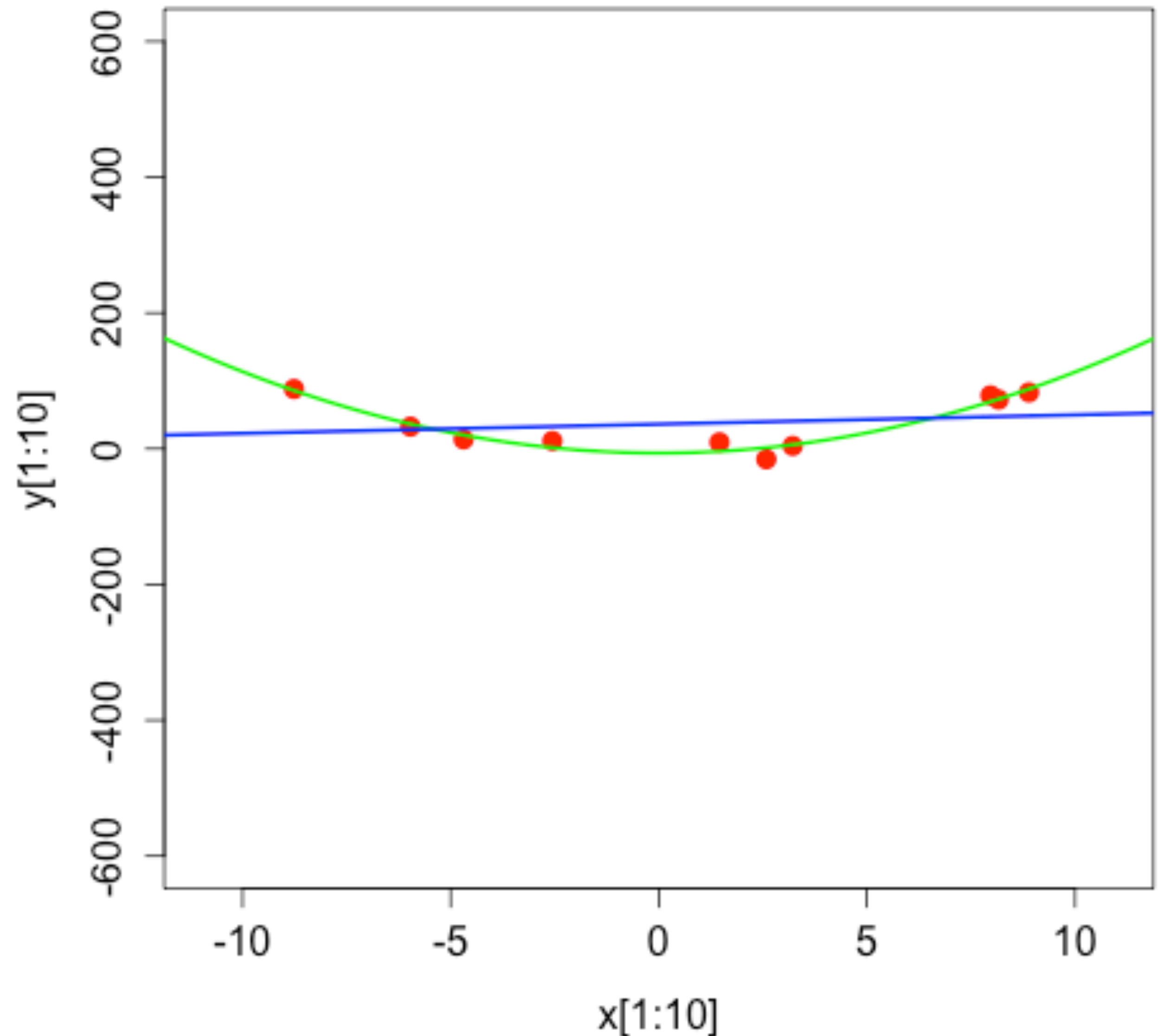
Example

- Quadratic data
- Assumption: data is linear
— use linear regression



Example

- Quadratic data
- Assumption: data is linear — use linear regression
- Error due to **bias** is high: more restrictions on model



Bias

- Complexity of model
- More restrictions lead to high **bias**

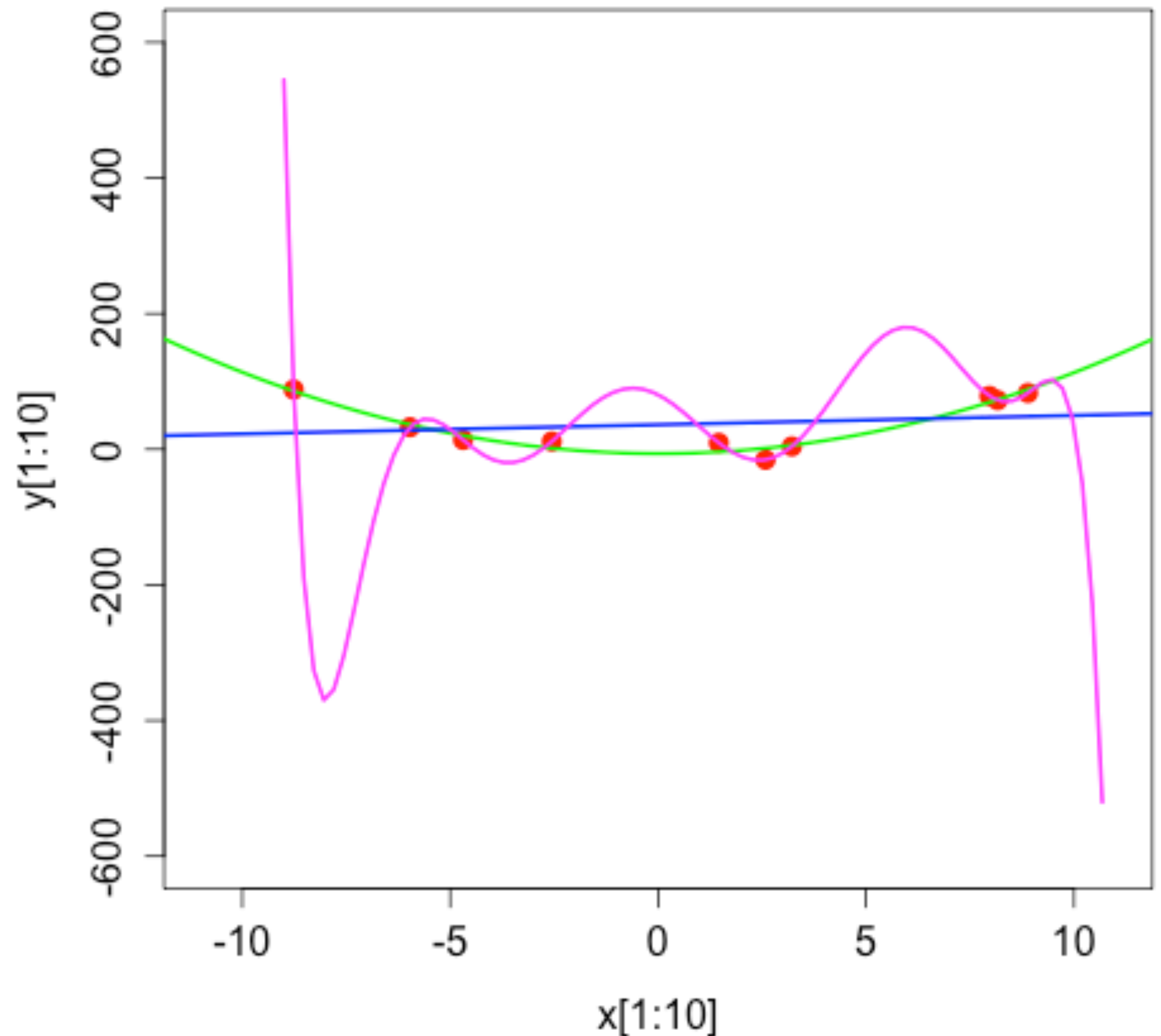
Variance

- Error due to **variance**: error due to the sampling of the training set
- Model with high **variance** fits training set closely

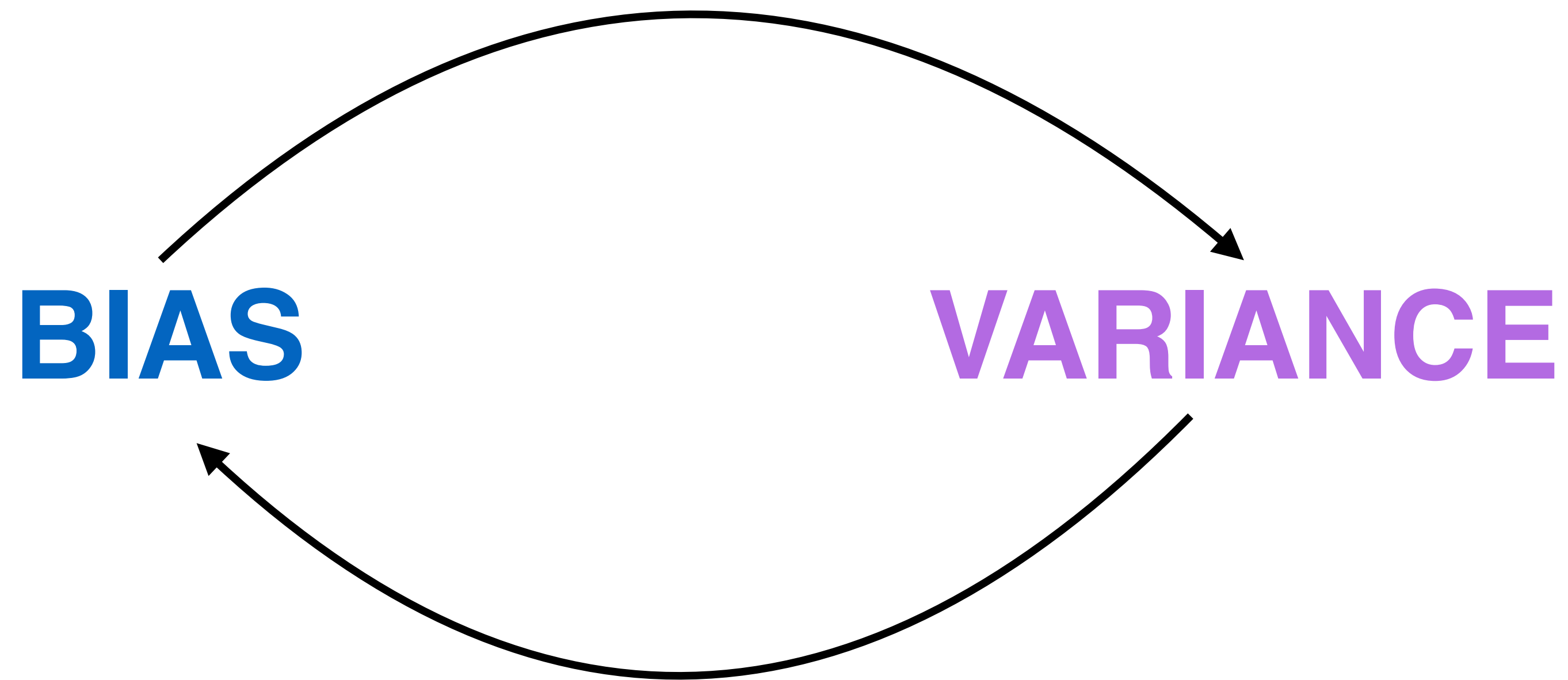
Example

- Quadratic data
- Few restrictions: fit polynomial perfectly through training set
- If you change training set, model will change completely

high **variance** : generalizes bad to test set



Bias-variance tradeoff



low **bias** - high **variance**

low **variance** - high **bias**

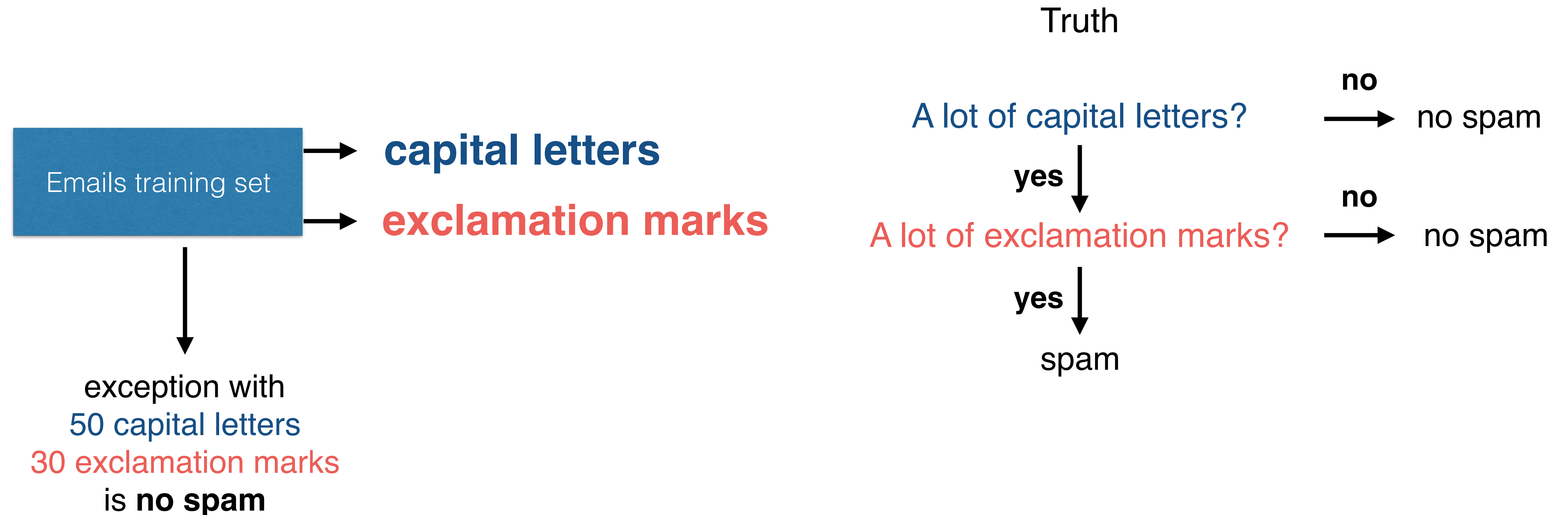
Overfitting

- Accuracy will depend on dataset **split** (train/test)
- High **variance** will heavily depend on **split**
- **Overfitting** = model fits **training set** a lot better than **test set**
- Too **specific**

Underfitting

- Restricting your model too much
- High **bias**
- Too **general**

Example - spam or not?



Example - spam or not?

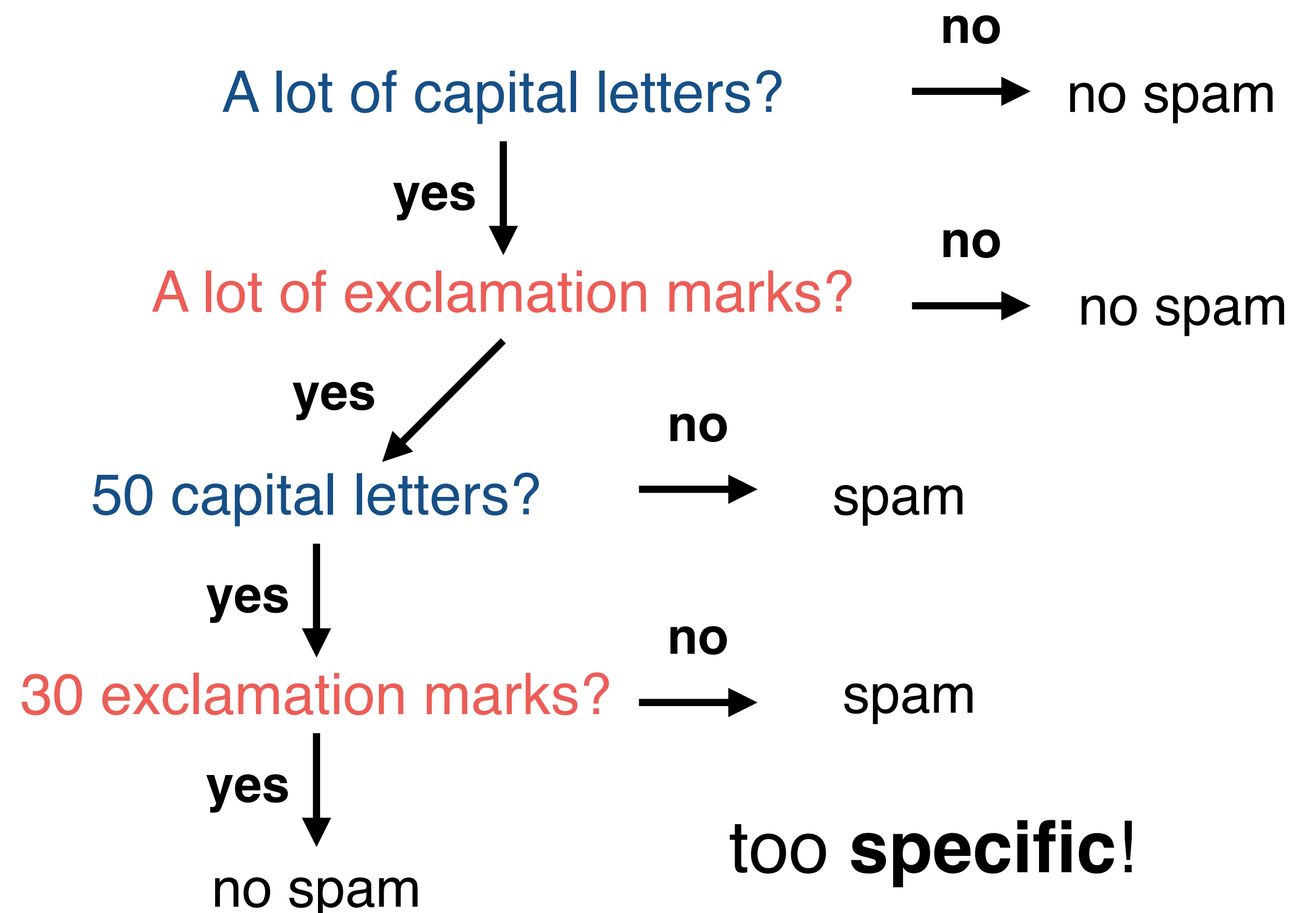
Overfit

Emails training set

capital letters

exclamation marks

exception with
50 capital letters
30 exclamation marks
is no spam



Example - spam or not?

Emails training set

→ **capital letters**

→ **exclamation marks**

Underfit

More than 10 capital letters? **no** → no spam

yes ↓
spam

too general!



INTRODUCTION TO MACHINE LEARNING

Let's practice!