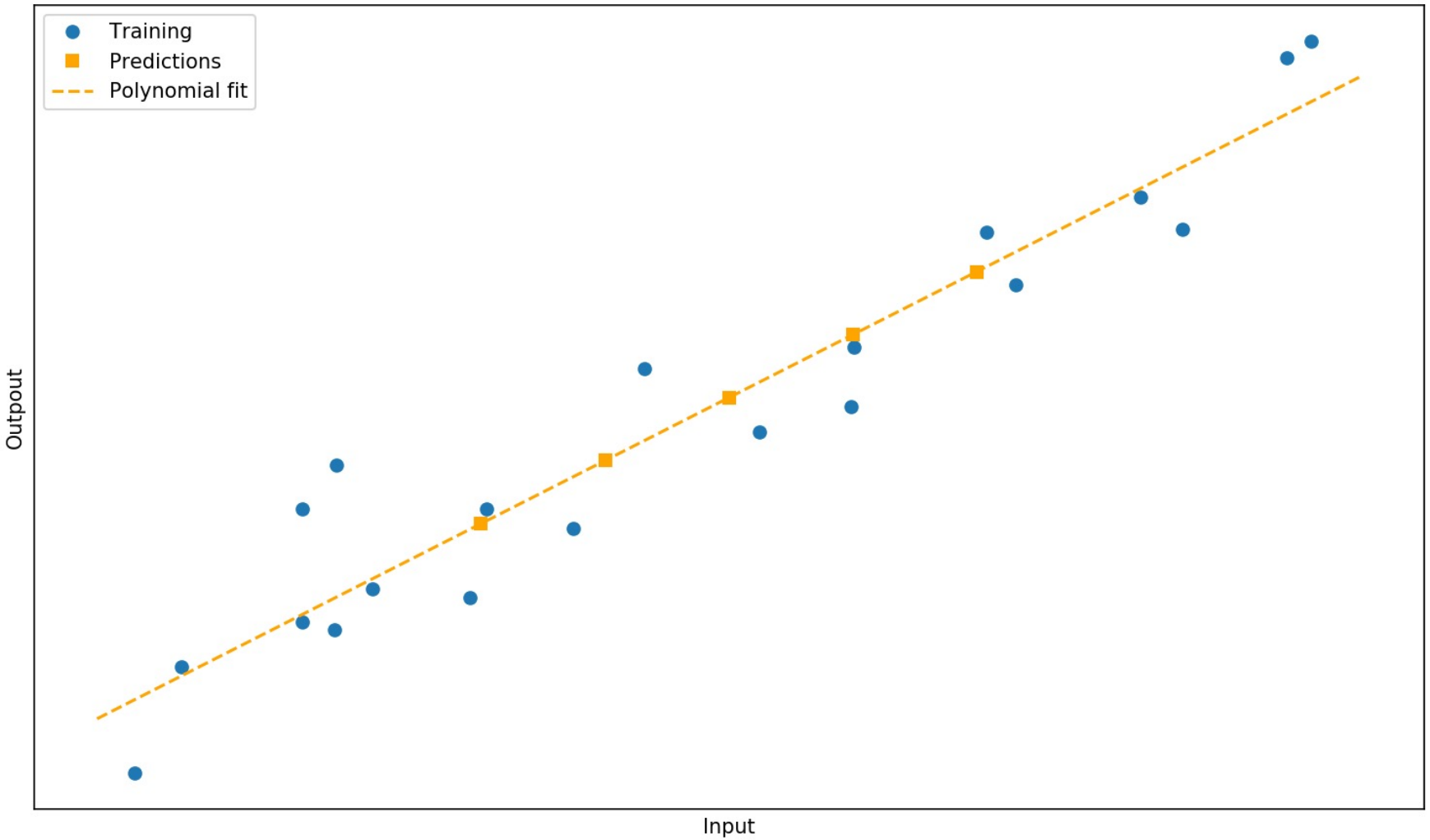
A (very) brief introduction to Machine learning key concepts

Notebook: 08-Intro_ML/ML_intro_short.ipynb

http://www.r2d3.us/visual-intro-to-machine-learning-part-1/

Supervised learning: Regression



Two main categories of ML:

- **Regression**
- Classification/clustering

ML requires a large training set

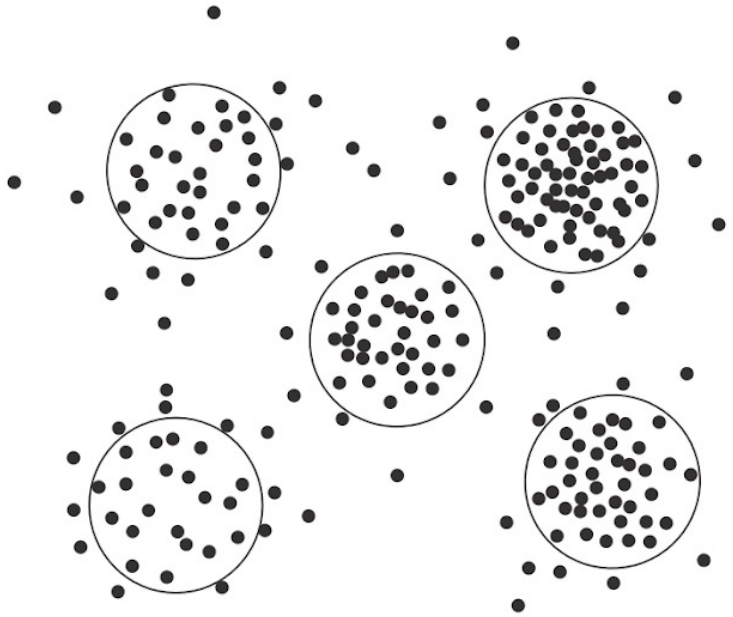
Science enters in big data era such that ML plays and will keep being important in the coming decade

(Un)-Supervised learning: Clustering

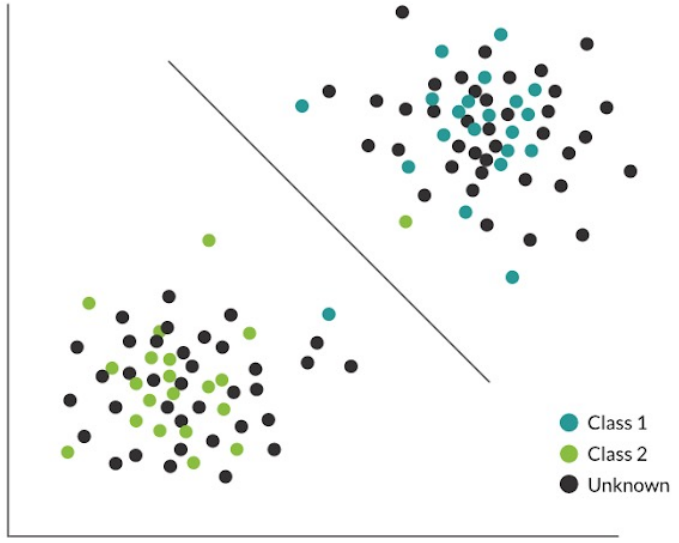
Two main categories of ML:
- Regression
- **Classification/clustering**

Unsupervised

Pattern learned from unlabeled data



Supervised

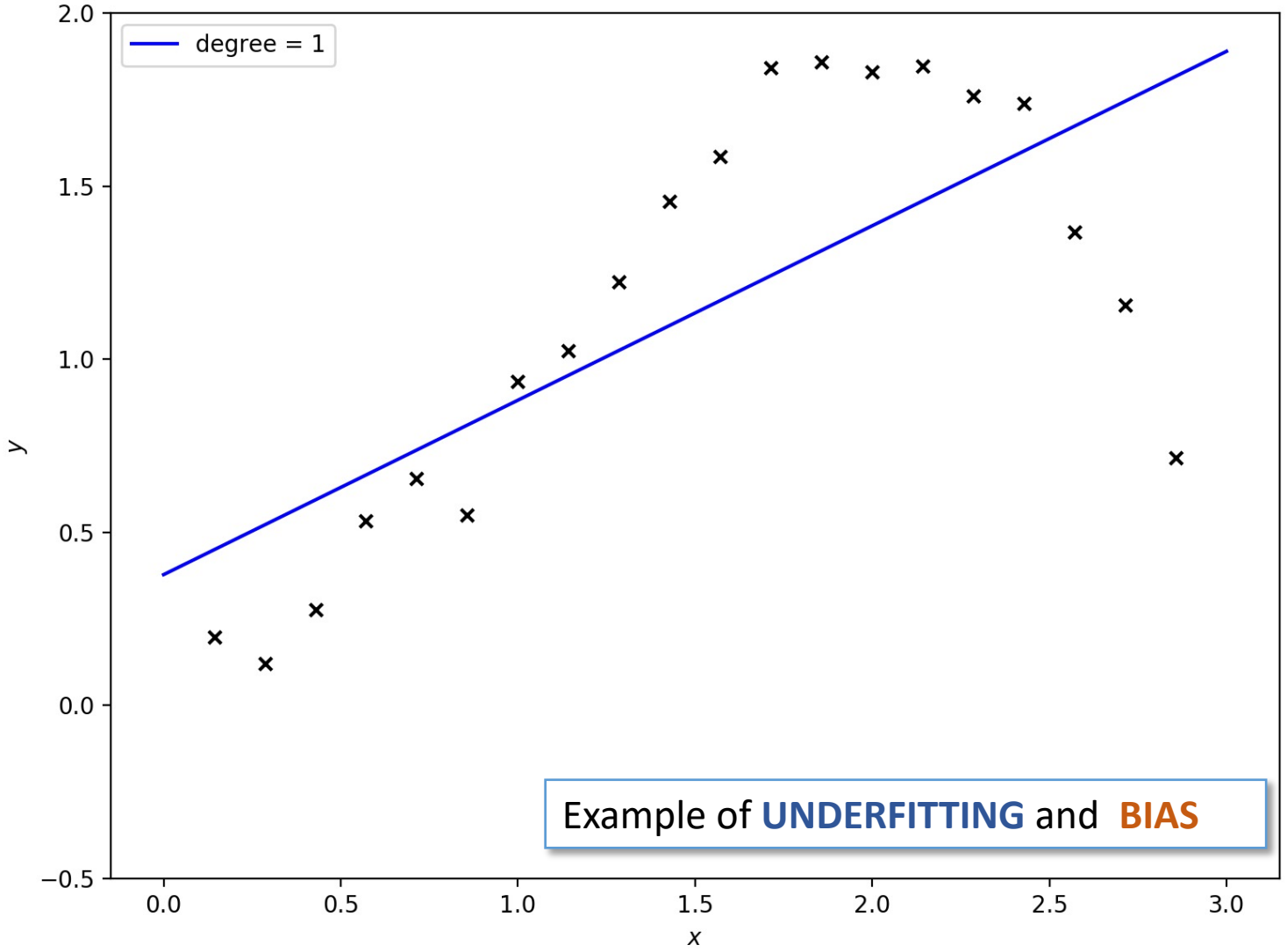


Large variety of ML algorithms:

- Regression trees
- K-nearest neighbours
- Support Vector Machine (SVM)
- Artificial neural Networks (ANN)
- Linear regression
- …

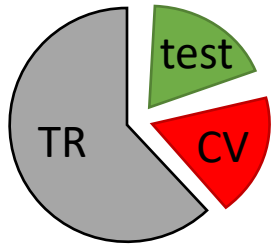
Python: scikit-learn

Bias, variance, overfitting



Example of **UNDERFITTING** and **BIAS**

One never trains a model on full data set



Training set : typically 50-70 % of data set
Cross-Validation (CV): ½ remaining data set
Test set: the other ½ of remaining data set

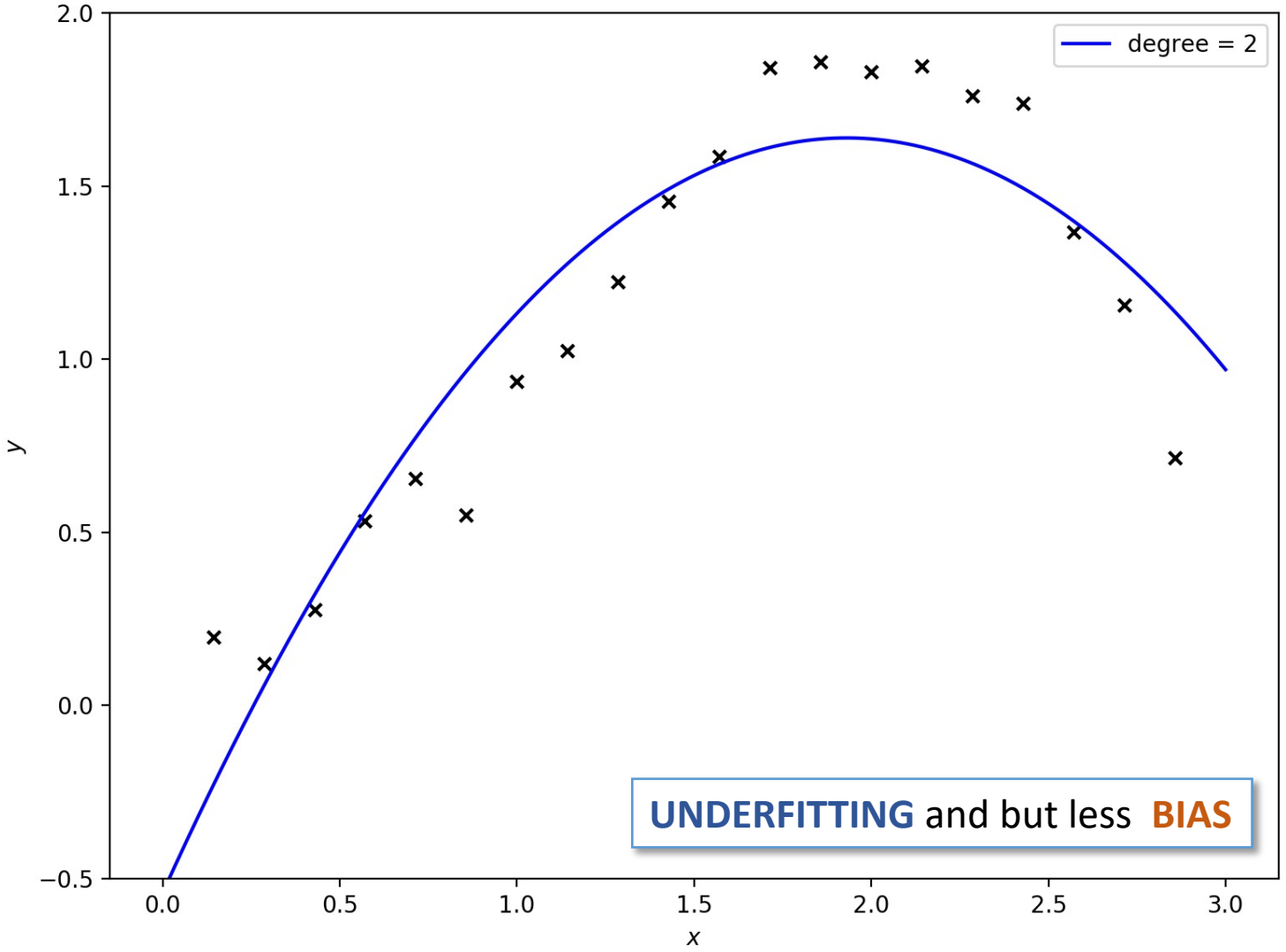
Quality of regression: Merit function / metric

MSE: $$\epsilon = \frac{1}{N}\sum_{i}^{N}(y_i - y_M(x_i))^2$$

BIC: $$BIC \equiv -2\ln[L^0(M)] + k\ln N$$

where $L^0(M)$ = Max. Likelihood of the model

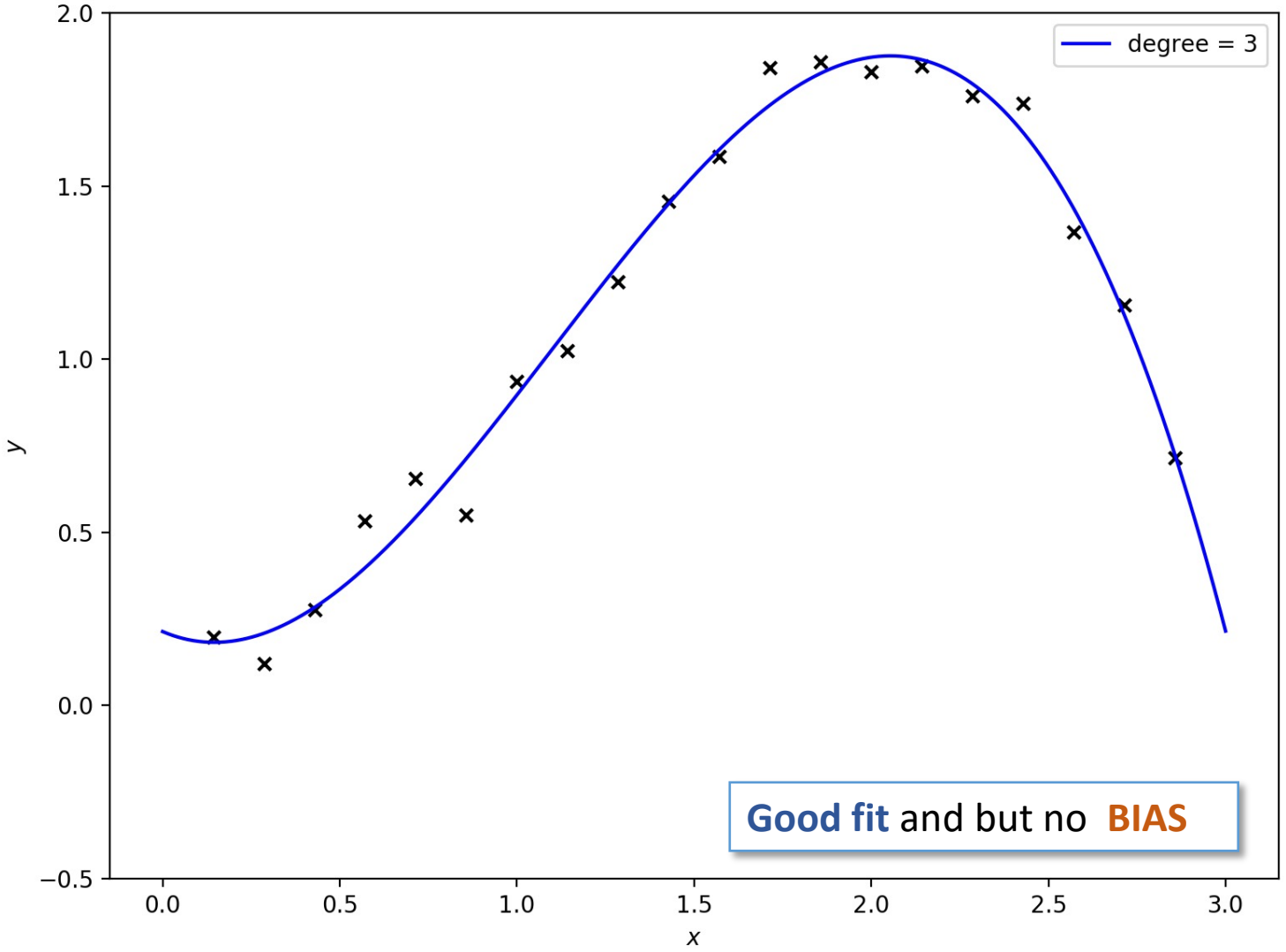
Bias, variance, overfitting



UNDERFITTING and but less **BIAS**

We can increase the degree of the polynomial

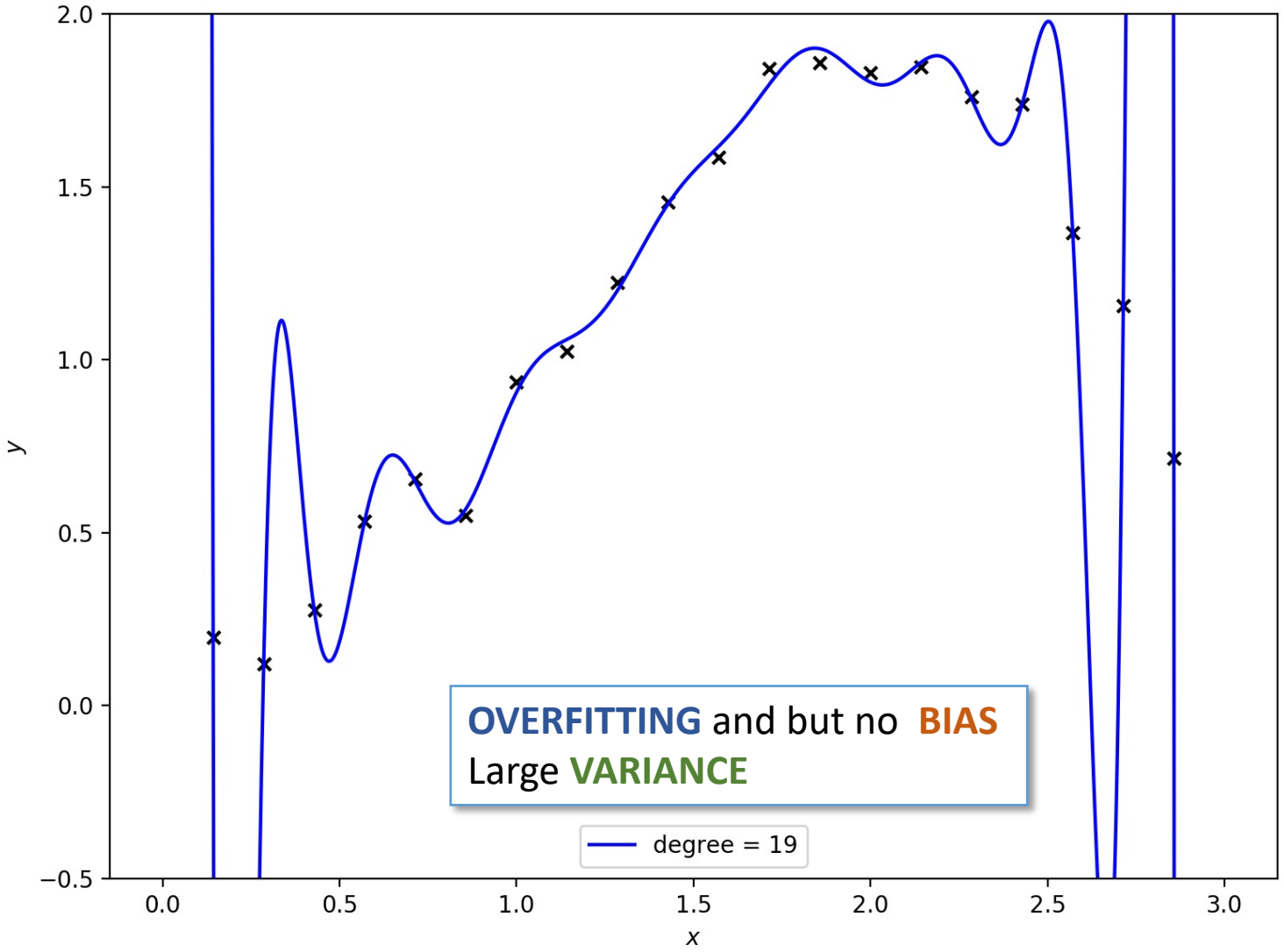
Bias, variance, overfitting



Good fit and but no **BIAS**

We can increase the degree of the polynomial

Bias, variance, overfitting

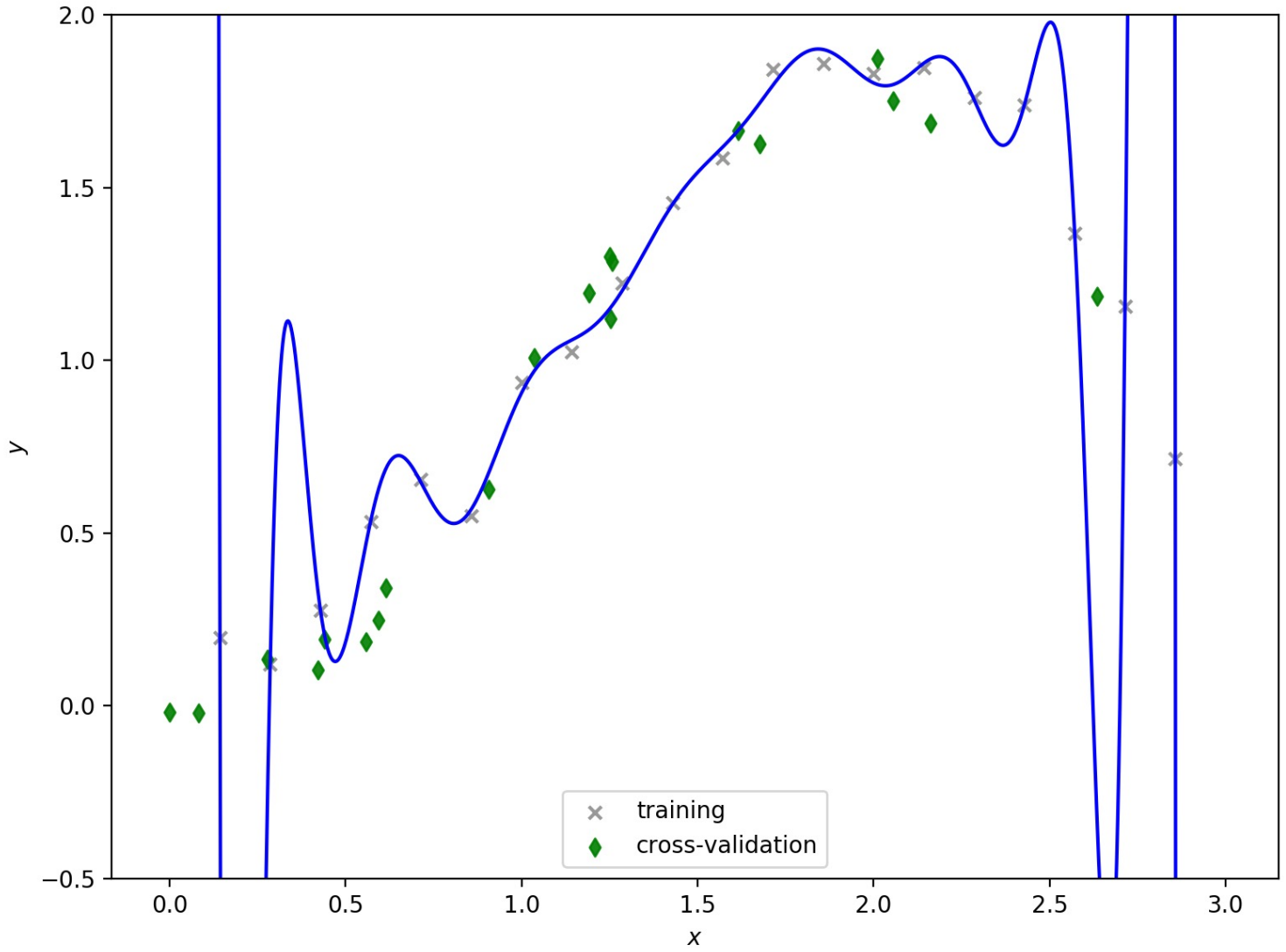


We can increase the degree of the polynomial

BIAS: How well we reproduce the data

Variance: How much the model changes if slightly perturbing the data points

How to choose the best model?

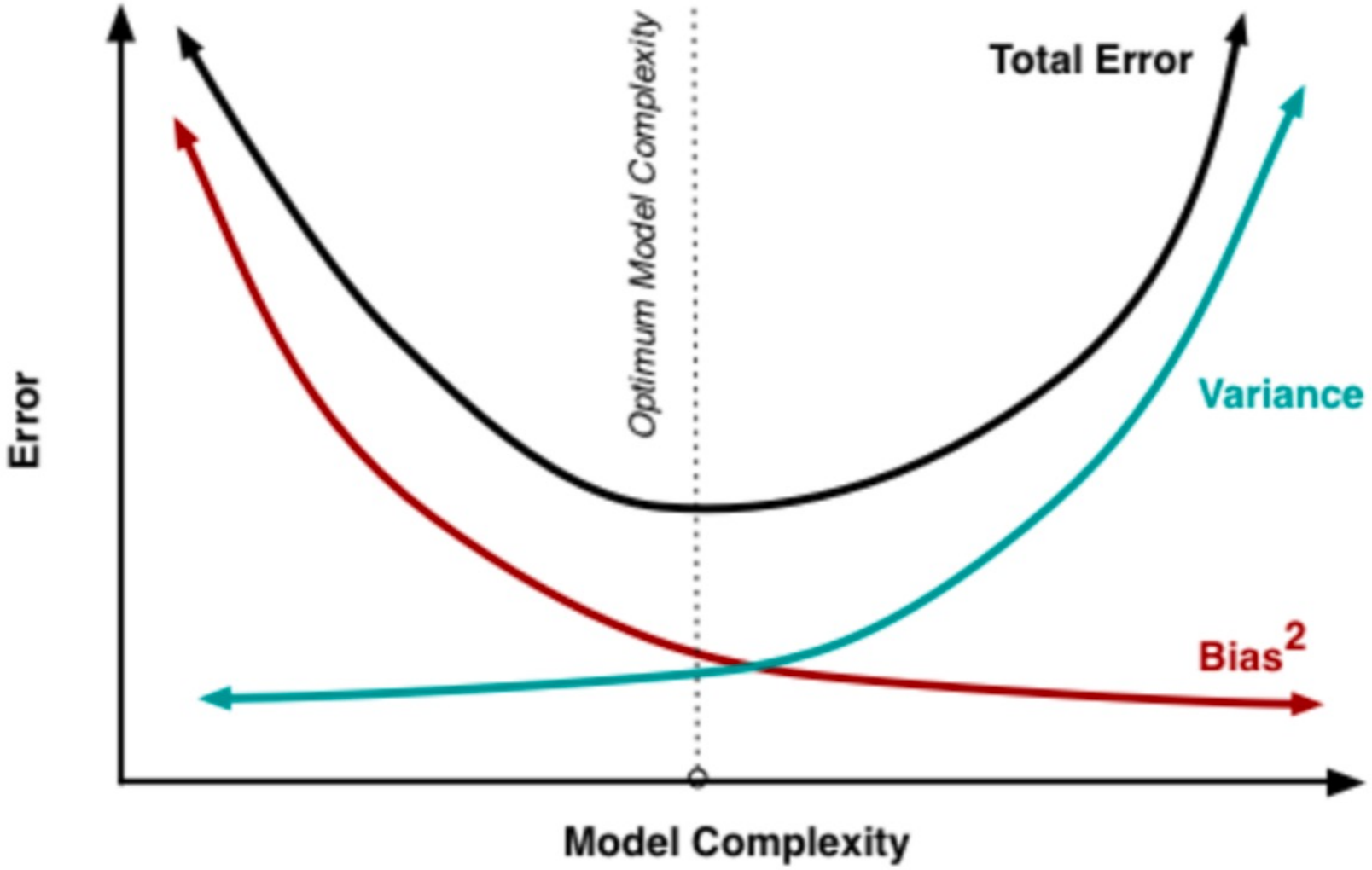


Use **CROSS-VALIDATION** to evaluate the best model

Use some merit function to evaluate the quality of the model: e.g. MSE,
RMS error = $\mathrm{MSE}^{1/2}$

$$\epsilon_{\mathrm{cv/tr}} = \sqrt{\frac{1}{n}\sum_{i=1}^{N_{\mathrm{cv/tr}}}\left[y_i - \sum_{m=0}^{d}\theta_0^{(n)}x_i^m\right]^2}$$

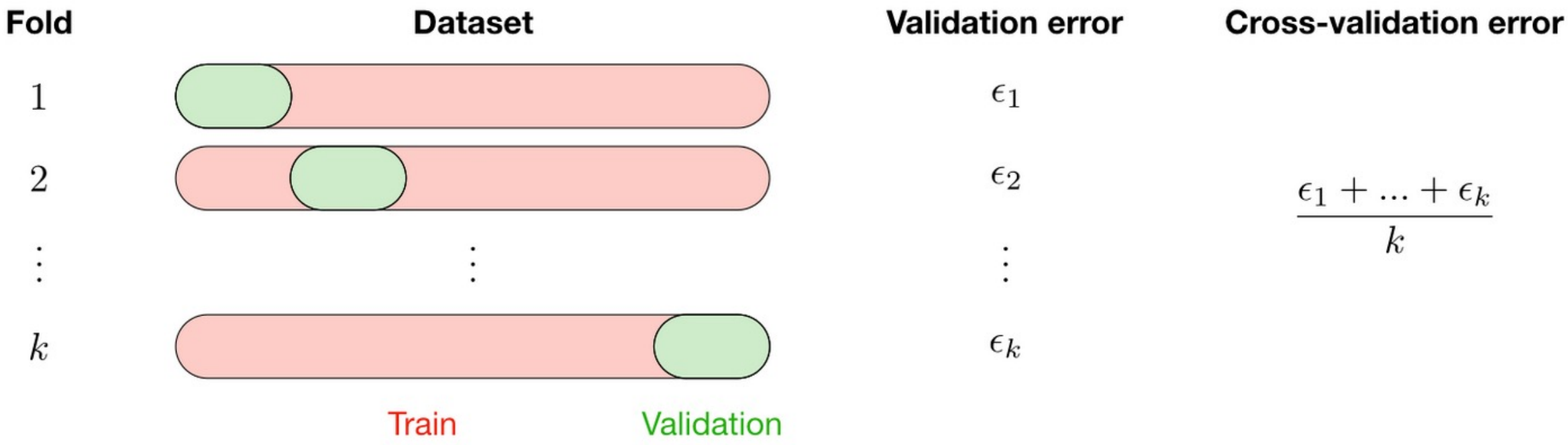
How to choose the best model?



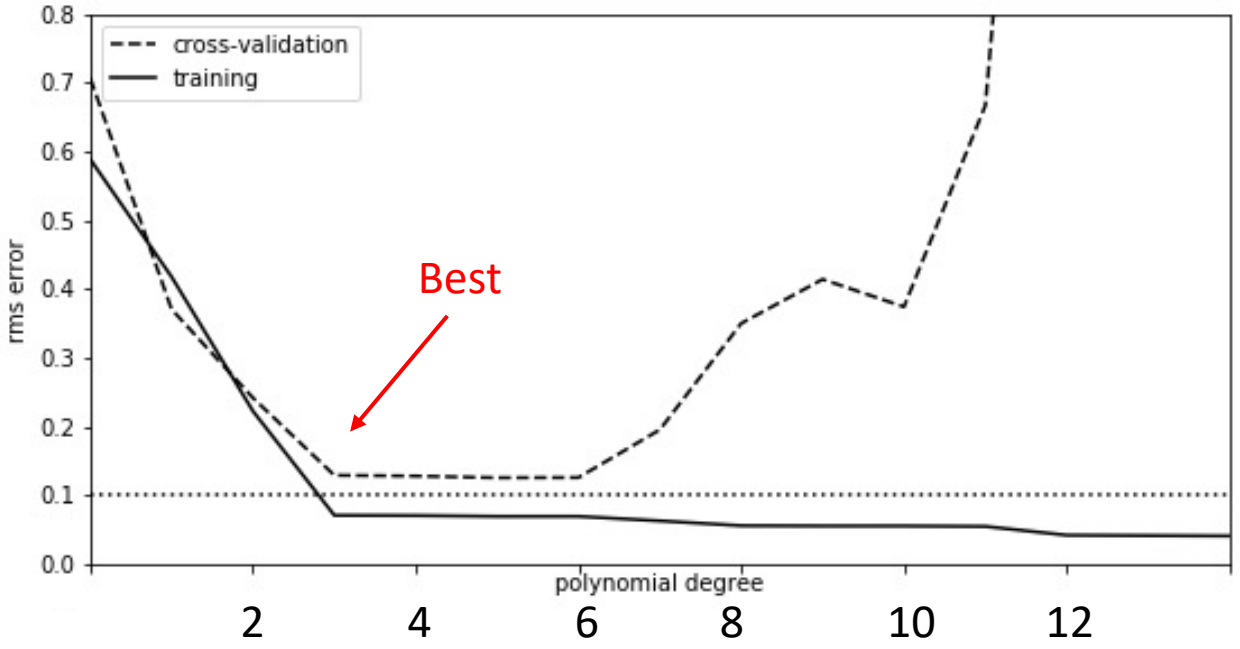
Use **CROSS-VALIDATION** to evaluate the best model

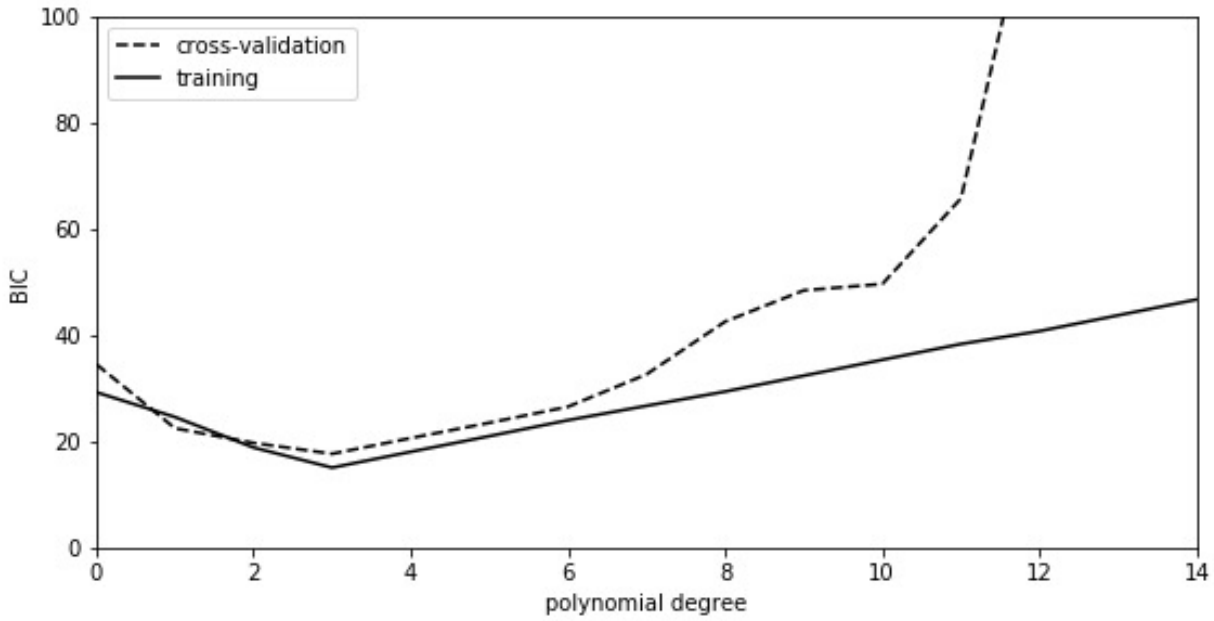
K-fold cross validation

Example of advanced CV scheme



How to choose the best model?





RMS error

$$\epsilon_{\mathrm{cv/tr}} = \sqrt{\frac{1}{n}\sum_{i=1}^{N_{\mathrm{cv/tr}}}\left[y_i - \sum_{m=0}^{d}\theta_0^{(n)}x_i^m\right]^2}$$

$$BIC \equiv -2\ln[L^0(M)] + k\ln N$$

Go to Notebook

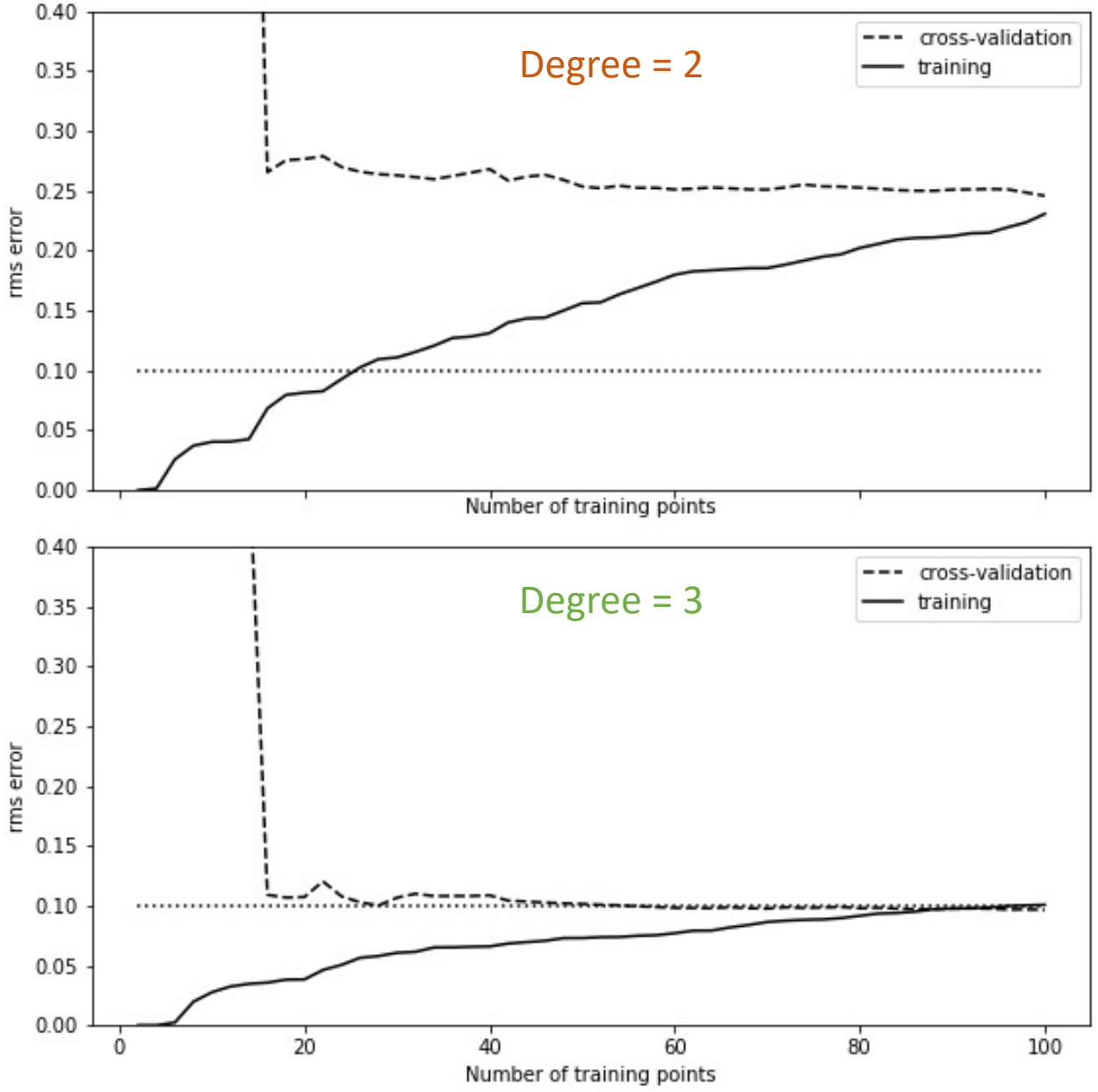
Do we need more data ?

To improve model fitting one can:

- Get **more training data**: does not guarantee a better fit
- Use **more/less complicated** model (especially if more training does not help)
- Use **more/less regularization**
- Increase the **number of features** (think of Tully-Fisher and fundamental plane)

Objective way to find out if more data help: **Learning curve**

Do we need more data ? *Learning curve*



Learning curve:

Merit/Metric calculated as a *function* of # of data points

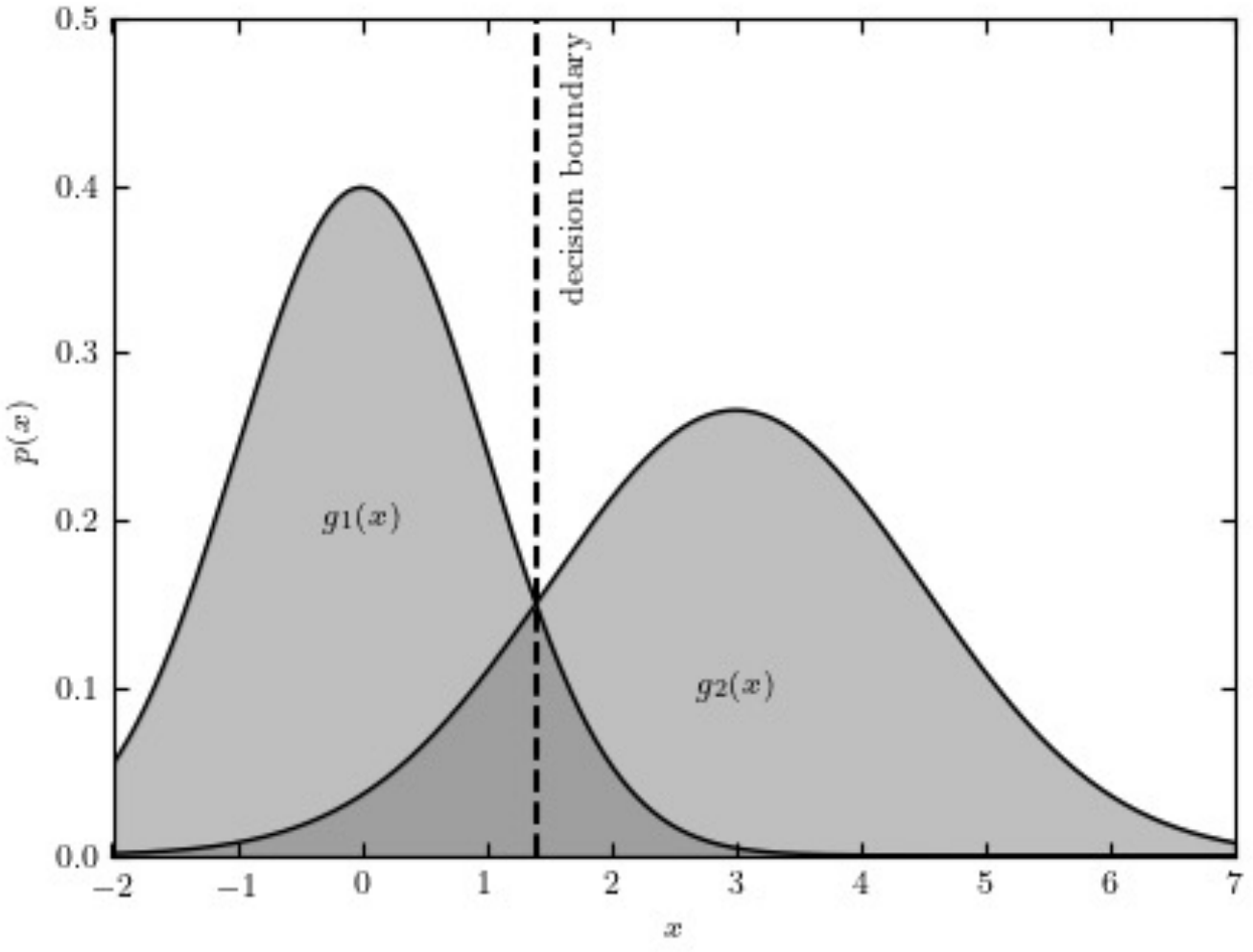
Warning: Use *TOTAL* N_{cv} for RMS error and vary only N_{train}

Training errors increase w. N_{train} (easier to fit less points)

CV errors decrease w. N_{train} (Overfitting less likely w. more points)

Go to Notebook

Classification

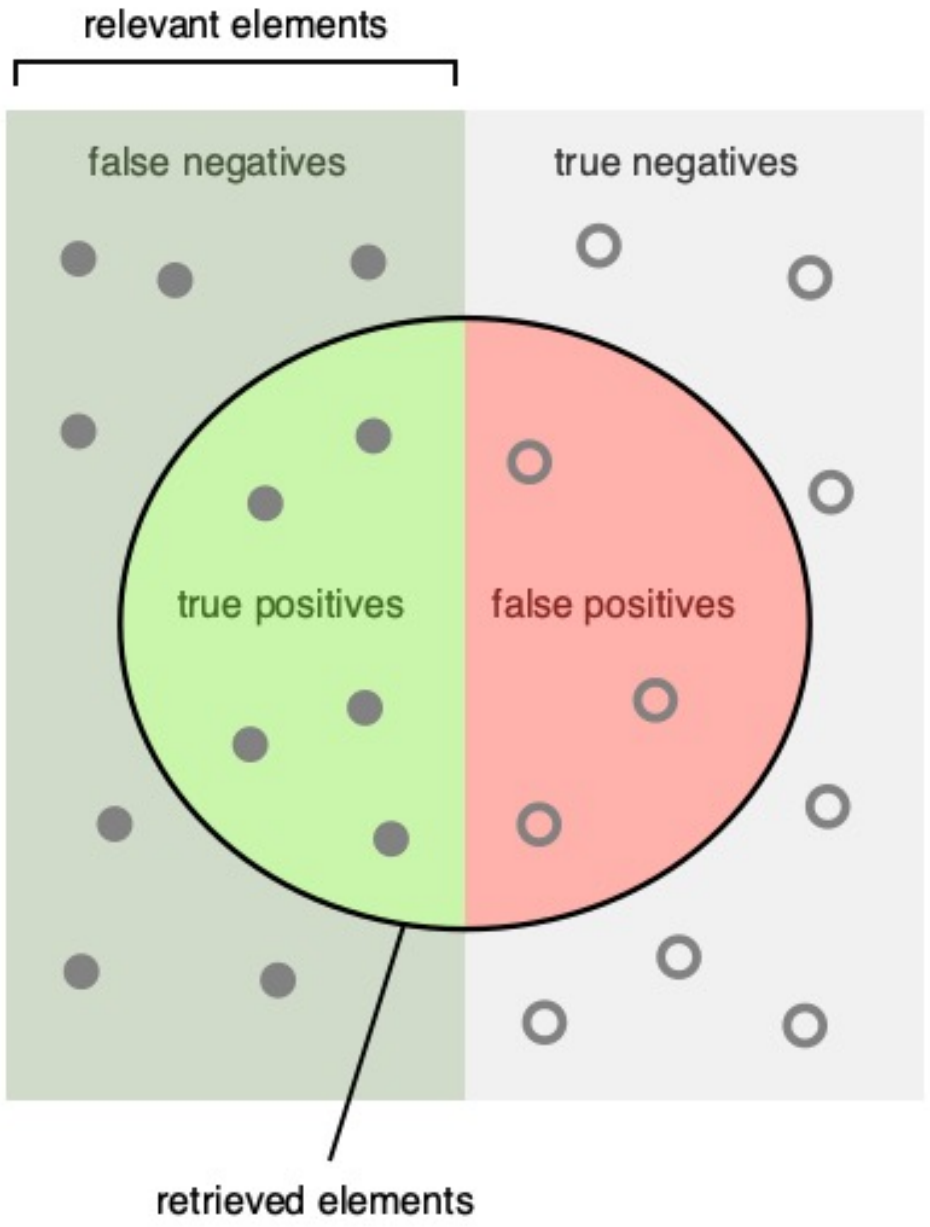


Generative classification: full model (predicting a pdf) for each class. Model 1 is better fit than model 2?

Discriminative classification: More like clustering. Decision boundary at x = 1.4.

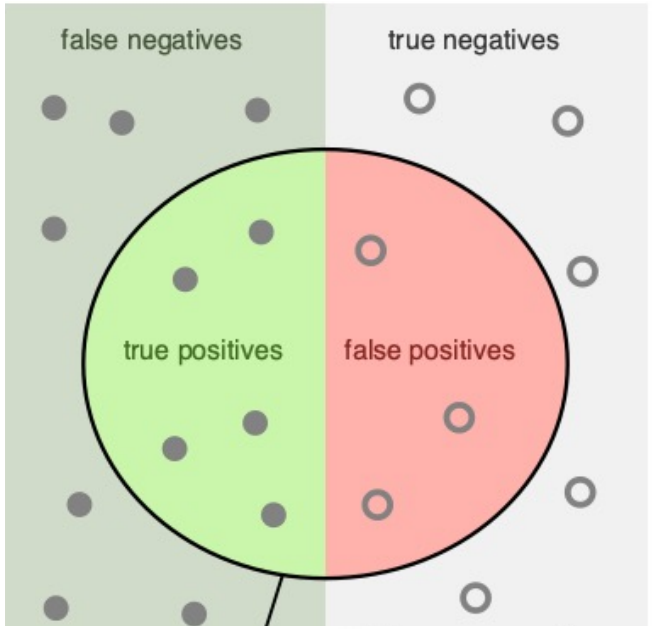
Classification: confusion matrix

		Ground truth	
		+	-
Predicted	+	True positive (TP)	False positive (FP)
	-	False negative (FN)	True negative (TN)



Classification: Completeness, contamination & friends

Name	Formula	Definition	Usage
Completeness or **True Positive Rate (TPR)** or **sensitivity** or **recall**	$\frac{TP}{TP+FN}$	Fraction of positive instances correctly identified	ML, Astro
Contamination of **False DISCOVERY rate**	$\frac{FP}{TP+FN}$	Fraction of positive instances correctly identified	Astro
False positive rate (FPR)	$\frac{FP}{TP+FP}$	Fraction of negative instances falsely classified as positive	ML, Astro
Accuracy	$\frac{TP+TN}{N}$	Fraction of instances correctly classified	ML, Astro
Precision or **efficiency** ≡ 1 - contamination	$\frac{TP}{TP+FP}$	Fraction of positive instances among the predicted positive	ML, Astro



How many retrieved items are relevant?

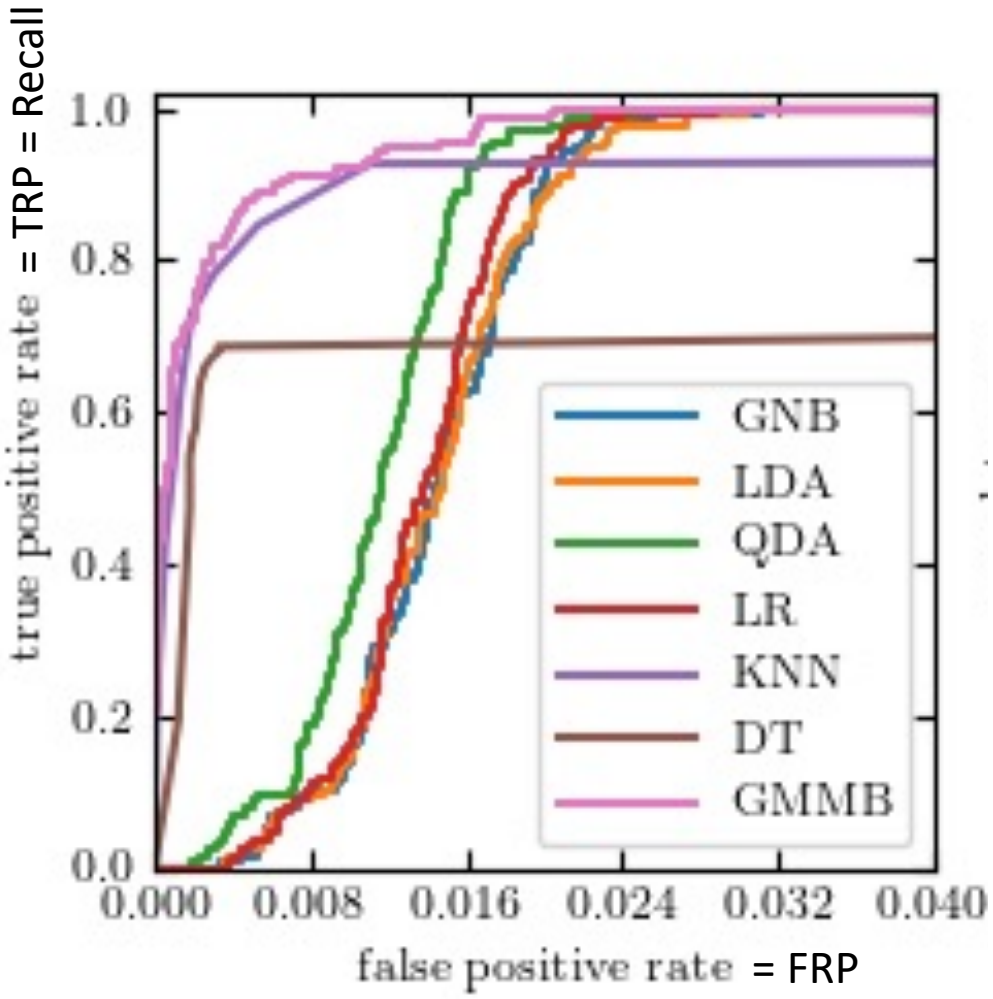
Precision =

How many relevant items are retrieved?

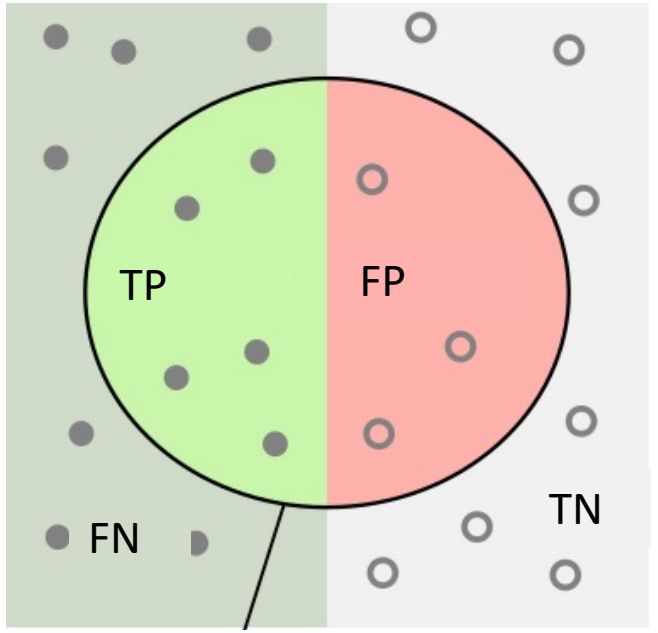
Recall =

Go to Notebook for example

ROC (Receiver Operating Characteristic) Curve



		Ground truth +	Ground truth -
Predicted +		True positive (TP)	False positive (FP)
Predicted -		False negative (FN)	True negative (TN)



$$\mathrm{TPR} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} =$$

← Relevant elements

$$\mathrm{FPR} = \frac{\mathrm{FP}}{\mathrm{TP} + \mathrm{FP}} =$$

← What you retrieved

Go to Notebook for example

To move forward