BIOS 7747: Machine Learning for Biomedical Applications

Supervised learning: classification

Antonio R. Porras (antonio.porras@cuanschutz.edu)

Department of Biostatistics and Informatics
Colorado School of Public Health
University of Colorado Anschutz Medical Campus

Outline

- Supervised learning: classification
- Binary classification: from thresholding to regression
- Logistic regression
- Performance evaluation

Supervised learning: classification

- Supervised learning
  - Learning from a dataset with known labels or outcomes

- Assumptions
  - The training dataset contains the “right” answers.
  - The right answers can be obtained from the available data

Training dataset
(the teacher)

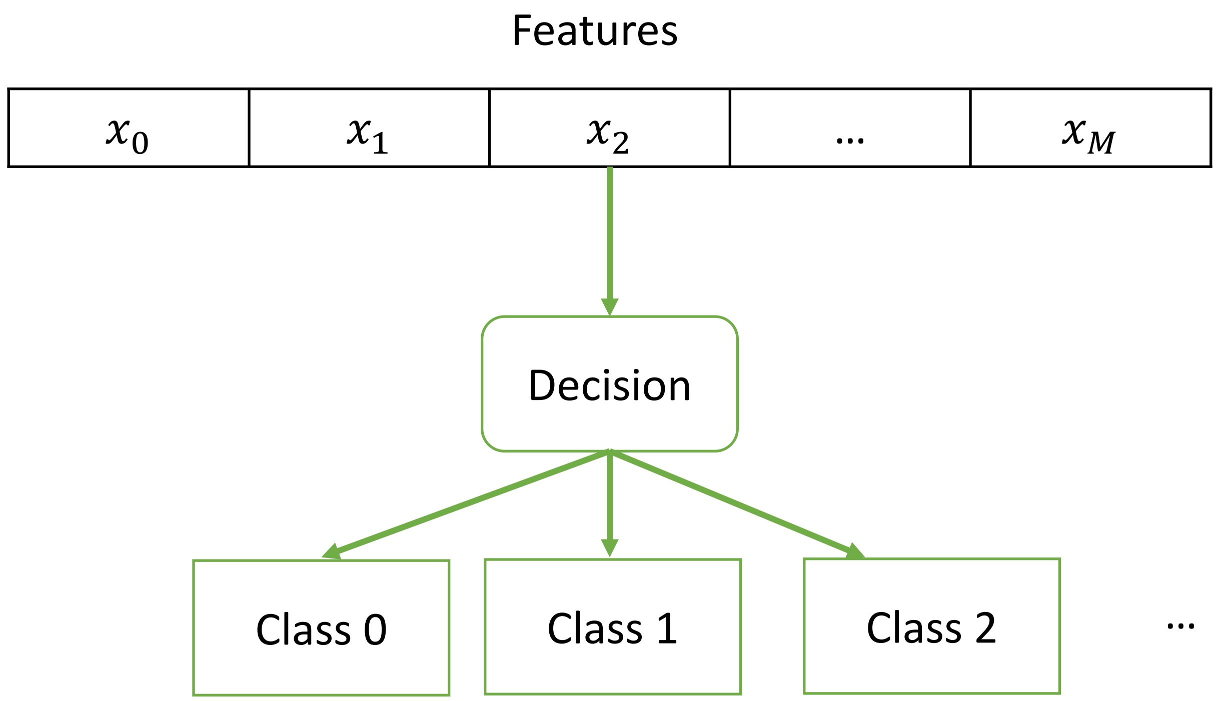
(x, y)

Model
(the student)

$f?$

Supervised learning: classification

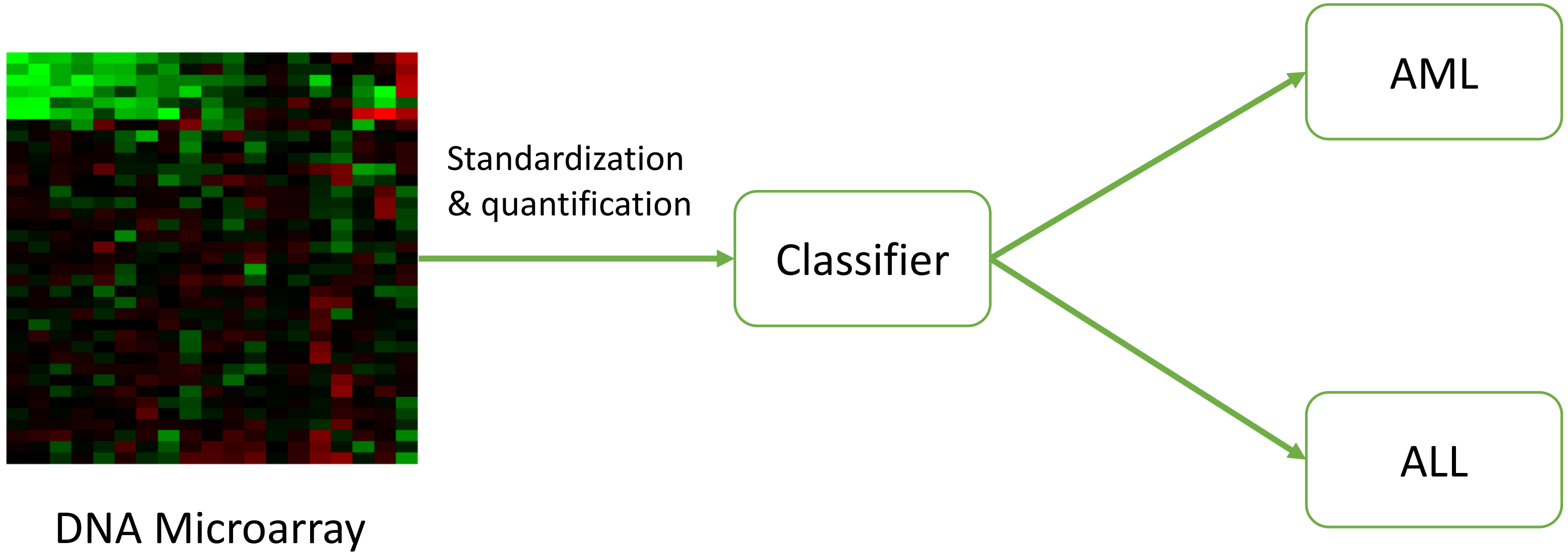
- Classification: models predict **discrete** variables



Supervised learning: classification

❑ Example:

- AML (acute myeloid leukemia) vs. ALL (acute lymphoblastic leukemia)

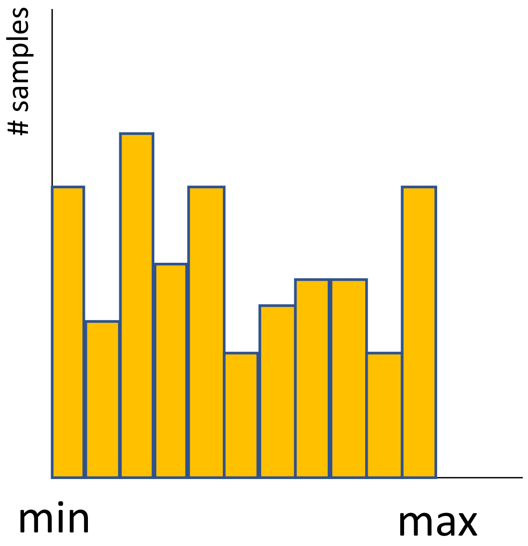


[Gollub et al, Molecular Classification of Cancer: Class Discovery and Class Prediction by Gene Expression Monitoring. Science, 1999]

Supervised learning: classification

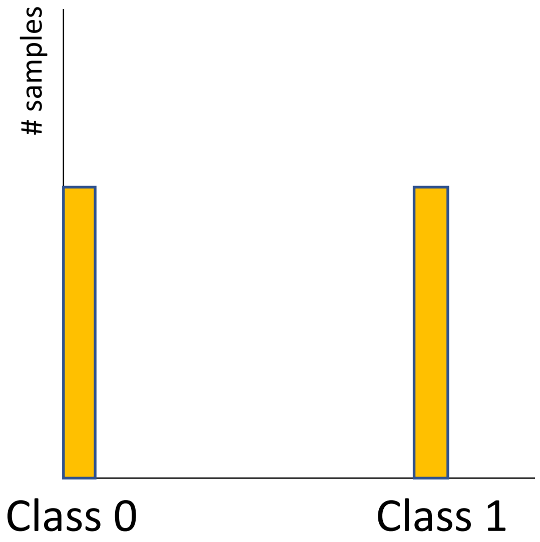
- Classification vs. regression

Regression



Continuous outcome

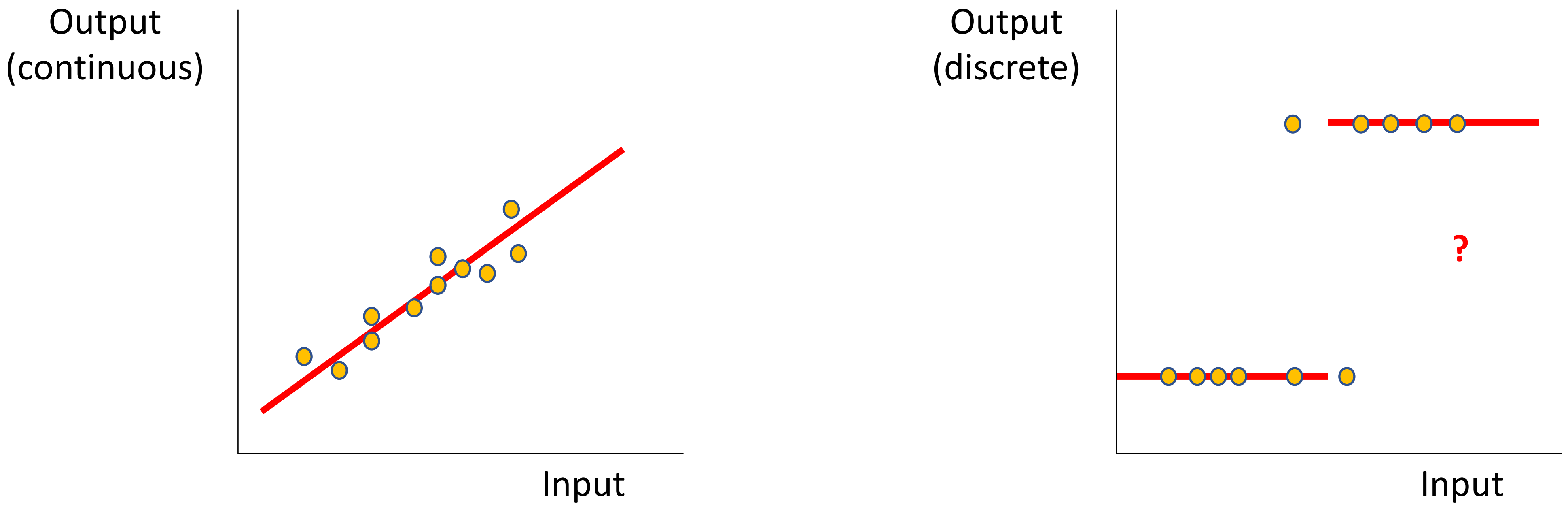
Classification



Discrete outcome

Supervised learning: classification

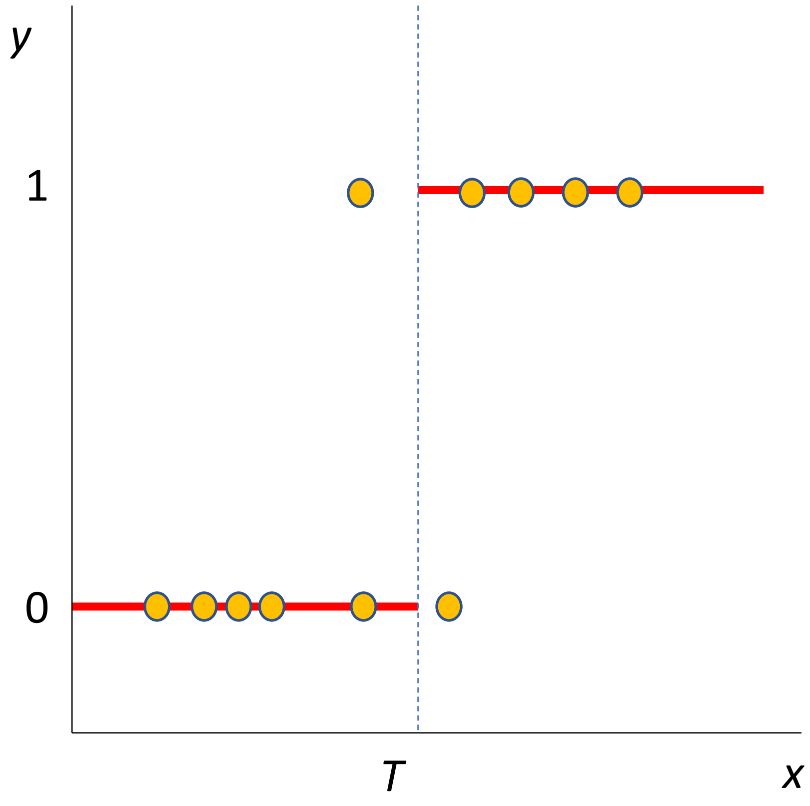
- Classification vs. regression



Thresholding

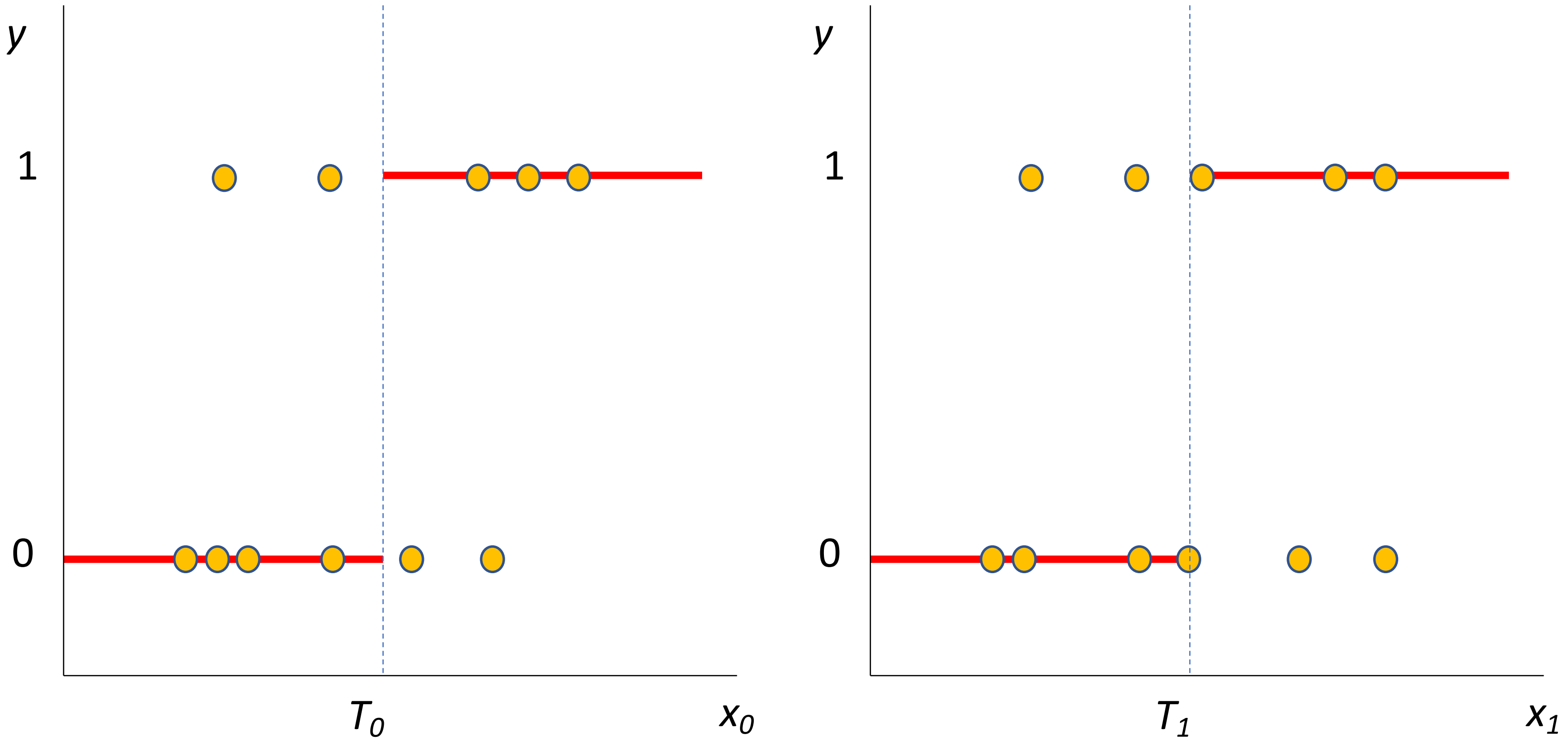
- Thresholding for binary classification

$$f(x) = \begin{cases} 0 \text{ if } x < T \\ 1 \text{ otherwise} \end{cases}$$



Thresholding

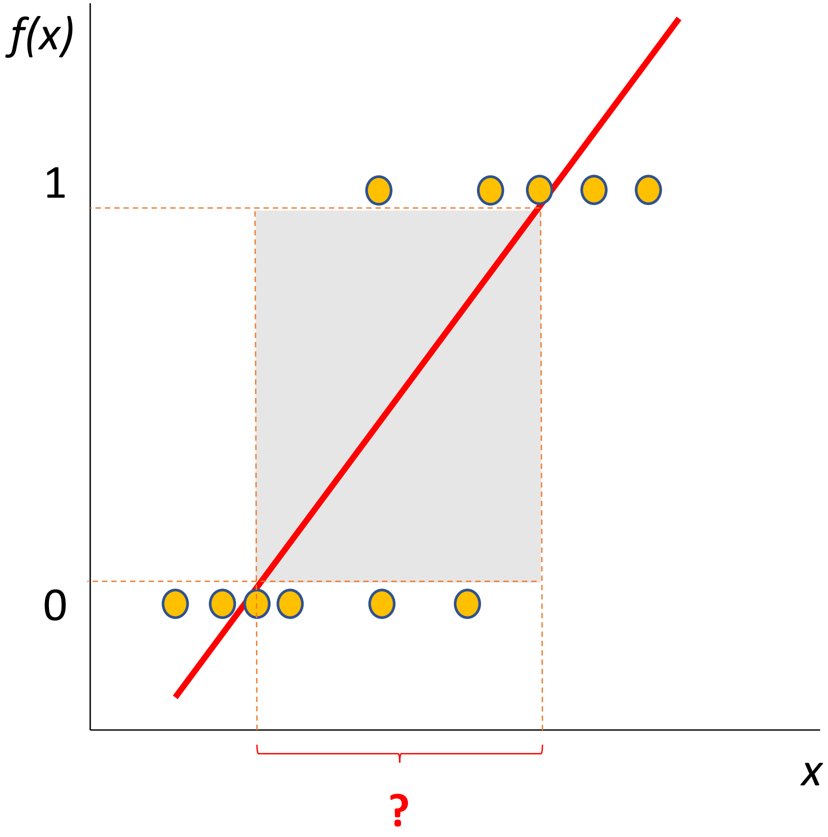
- Thresholding for binary classification
  - Impractical in most problems



- How to find multiple thresholds?
- How to combine multiple decision functions?

Classification as regression

Linear regression



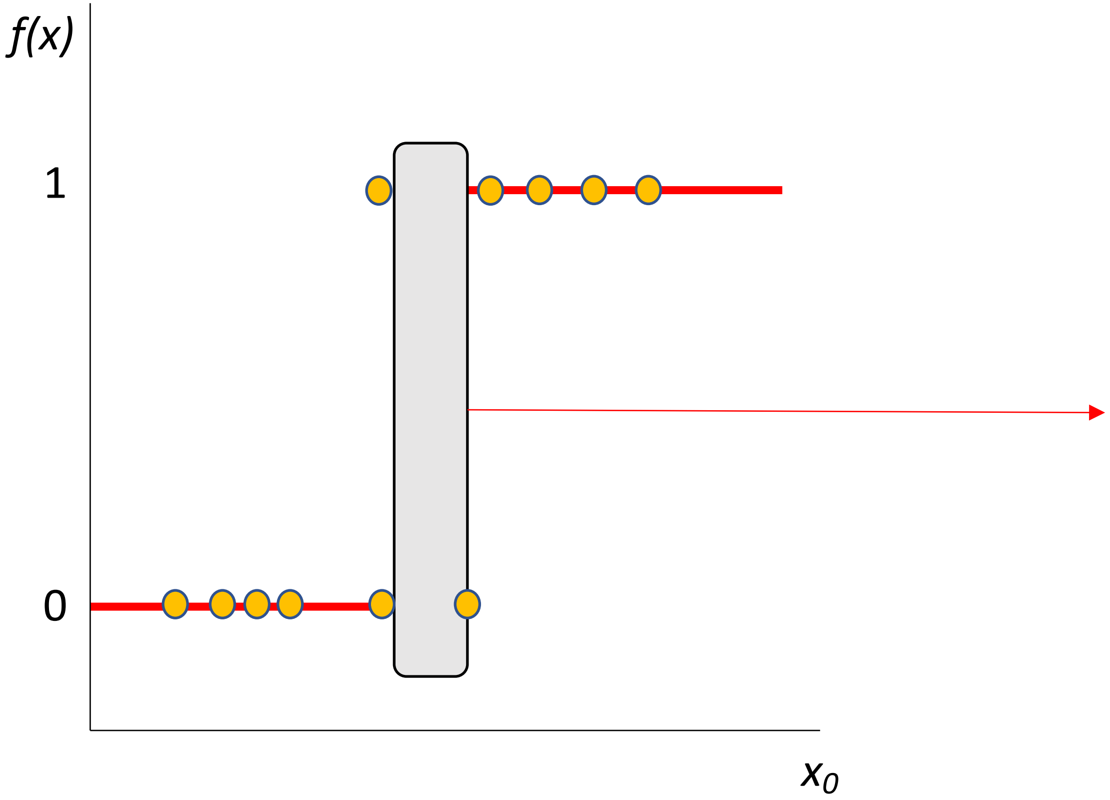
$$f(\boldsymbol{x};\boldsymbol{\theta}) = \boldsymbol{x}\boldsymbol{\theta}$$

$$\boldsymbol{\theta} = \min \sum_{i=0}^{N-1} \left(f\left(\boldsymbol{x}^{(i)};\boldsymbol{\theta}\right) - y^{(i)} \right)^2$$

- Can combine and weight different features
- Creates unbounded predictions with no real interpretation
- Great degree of uncertainty

Classification as regression

Binary regression?



$$\boldsymbol{\theta} = \min \sum_{i=0}^{N-1} \left(\underbrace{f\left(\boldsymbol{x}^{(i)}; \boldsymbol{\theta}\right)} - y^{(i)} \right)^2$$

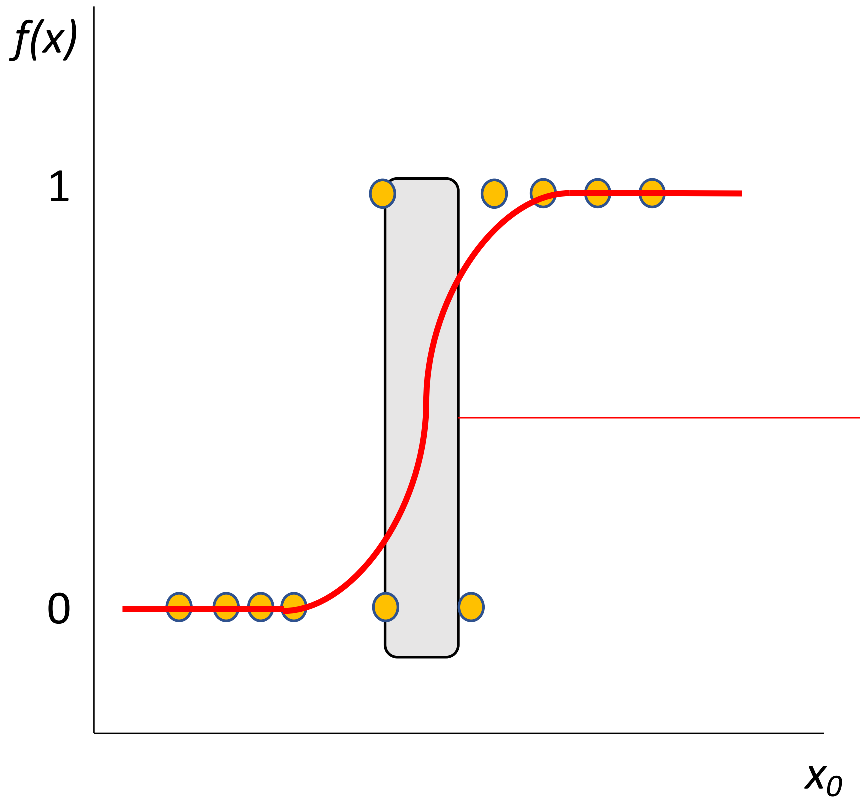
Not continuous and differentiable

- Not suitable for gradient-based optimization

But also, data are not normally distributed…

Classification as regression

Sigmoid regression



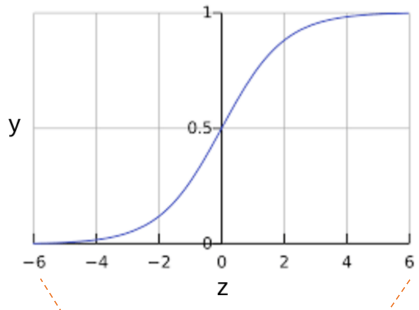
$$\boldsymbol{\theta} = \min \sum_{i=0}^{N-1} \left(\underbrace{f\left(\boldsymbol{x}^{(i)}; \boldsymbol{\theta}\right)} - y^{(i)} \right)^2$$

- Continuous and differentiable
- Small transition area
- Could be interpretable?
- How to combine different predictions?
- Least-square-error fitting: data are still not normally distributed

Classification as regression

Logistic regression

Sigmoid function



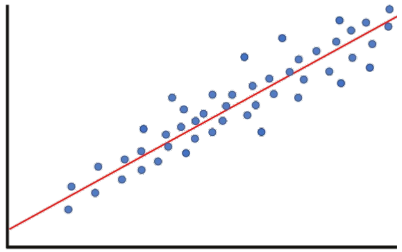
$$y = \frac{e^z}{1 + e^z}$$

$$y = \frac{1}{1 + e^{-z}}$$

- Bounded: [0,1]
- Could be interpreted as a probability
- Continuous and differentiable

+

Linear regression



$$z = \boldsymbol{x\theta}$$

- Combines multiple features
- Allows for probability calibration

Classification as regression

- Probability, odds and log(odds)

Normal coin flip

$$P(heads) = \frac{N_{heads}}{N_{total}} = \frac{50}{100} = 0.5$$

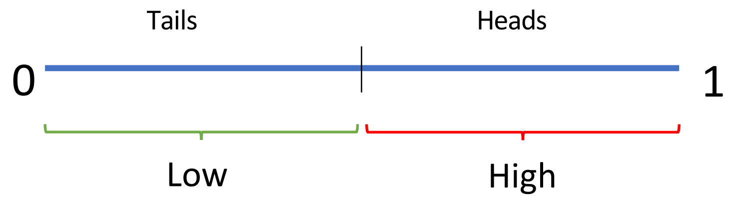
$$Odds(heads) = \frac{P(heads)}{P(not\ heads)} = \frac{P(heads)}{1 - P(heads)} = \frac{0.5}{0.5} = 1$$

Rigged coin flip

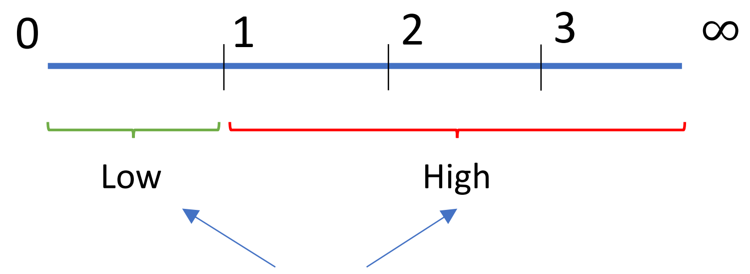
$$P(heads) = \frac{N_{heads}}{N_{total}} = \frac{75}{100} = 0.75$$

$$Odds(heads) = \frac{P(heads)}{P(not\ heads)} = \frac{P(heads)}{1 - P(heads)} = \frac{0.75}{0.25} = 3$$

Probabilities $\in [0, 1]$



Odds $\in [0, \infty]$



Highly asymmetric

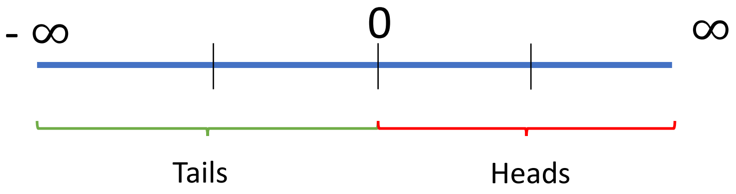
Classification as regression

- Probability, odds and log(odds)

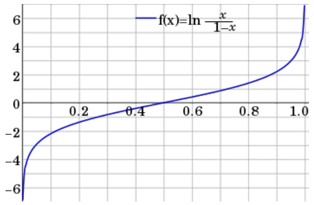
Rigged coin flip

$$P(heads) = \frac{N_{heads}}{N_{total}} = \frac{75}{100} = 0.75$$

Log(odds) $\in [-\infty, \infty]$



Logit function



$$Odds(heads) = \frac{P(heads)}{P(not\ heads)} = \frac{P(heads)}{1 - P(heads)} = \frac{0.75}{0.25} = 3$$

$$\log(Odds(heads)) = \log\left(\frac{P(heads)}{P(not\ heads)}\right) = \log\left(\frac{P(heads)}{1 - P(heads)}\right) = \log\left(\frac{0.75}{0.25}\right) = \log(3) = 1.1$$

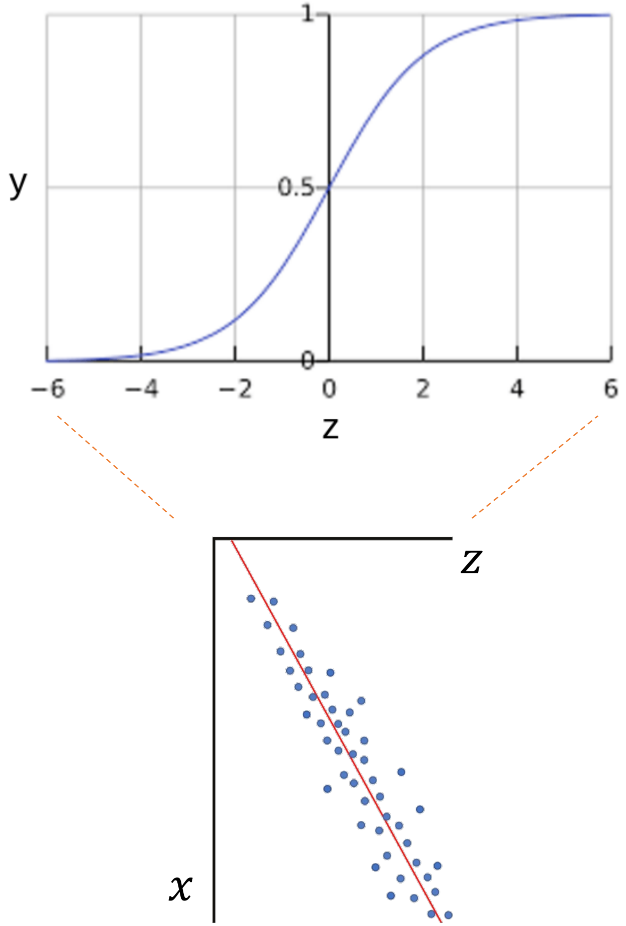
Symmetric

$$\log(Odds(tails)) = \log\left(\frac{P(tails)}{P(heads)}\right) = \log\left(\frac{P(tails)}{1 - P(tails)}\right) = \log\left(\frac{0.25}{0.75}\right) = \log(0.33) = -1.1$$

If we repeated this experiment with random samples and generated a histogram of log(odds), it would have a normal distribution centered at 0

Classification as regression

Logistic regression



$$y = P(class_1) = \frac{e^z}{1+e^z}$$

$$P(class_0) = 1 - P(class_1) = 1 - \frac{e^z}{1+e^z} = \frac{1}{1+e^z}$$

$$Odds(class_1) = \frac{P(class_1)}{1-P(class_1)} = \frac{P(class_1)}{P(class_0)} = e^z$$

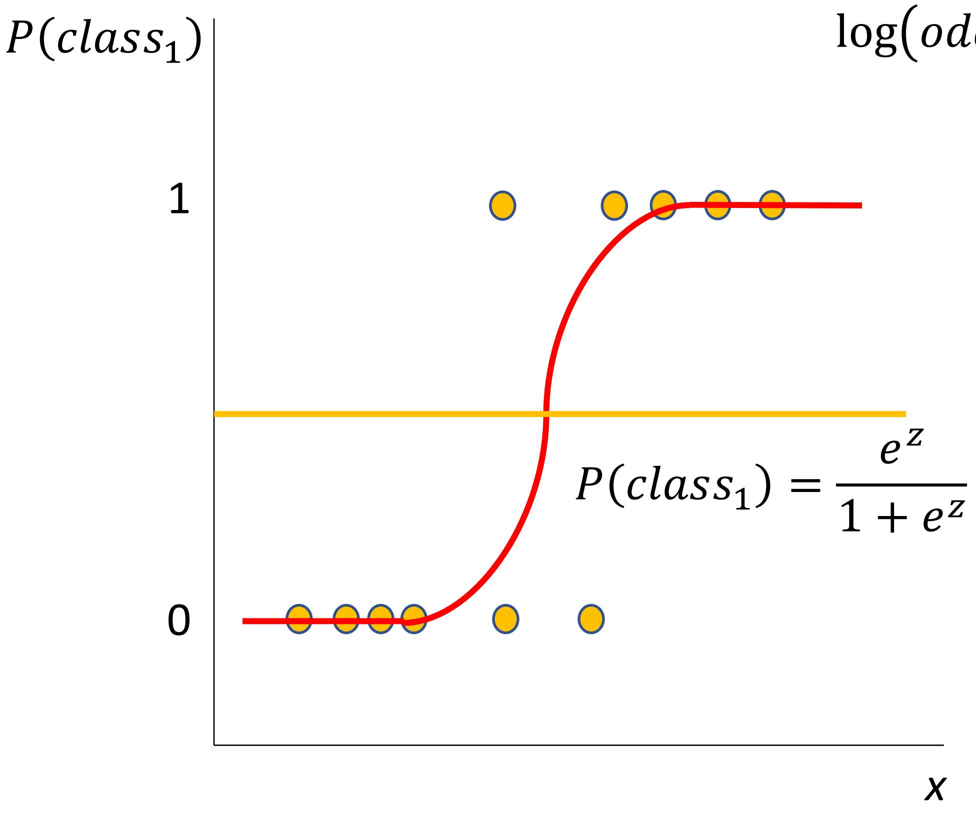
$$\log\big(odds(class_1)\big) = z$$

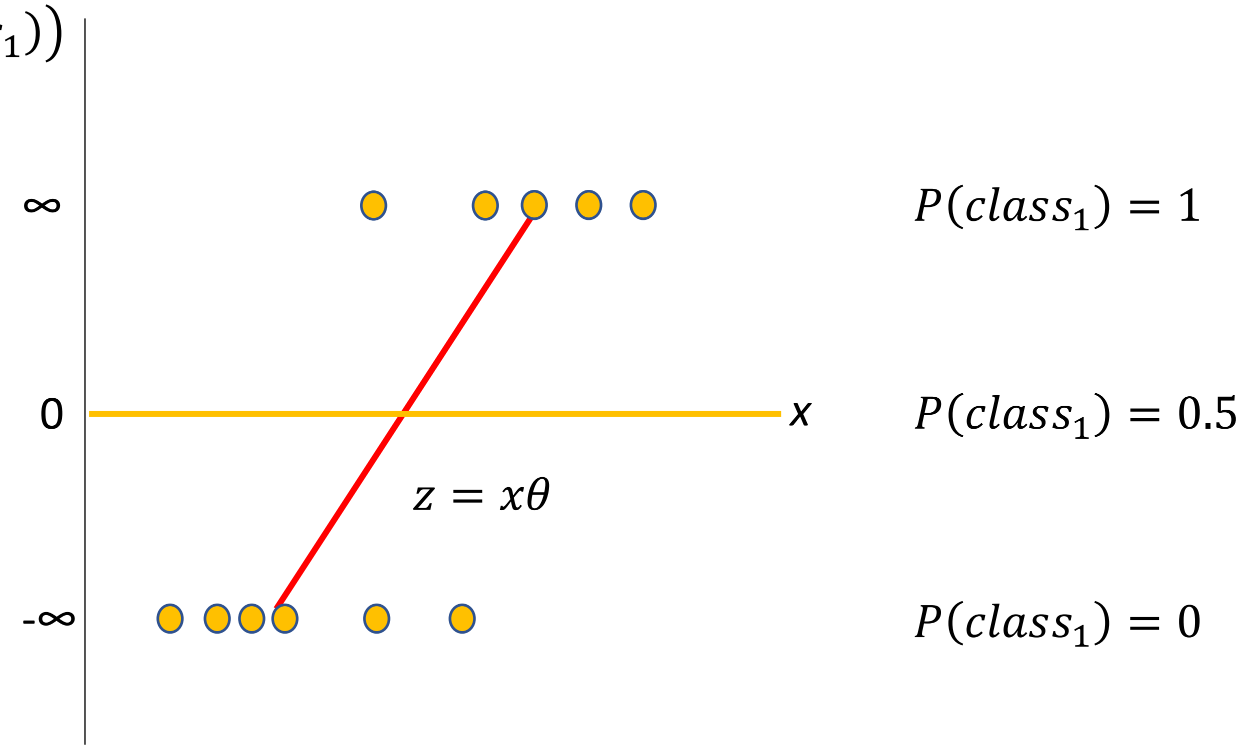
Zero mean

$$Odds(class_0) = \frac{P(class_0)}{1-P(class_0)} = \frac{P(class_0)}{P(class_1)} = e^{-z}$$

$$\log\big(odds(class_0)\big) = -z$$

Classification as regression



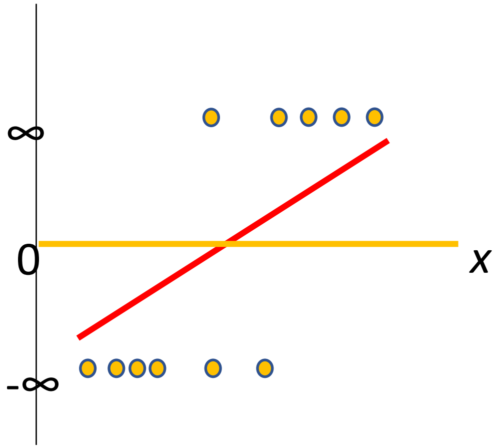


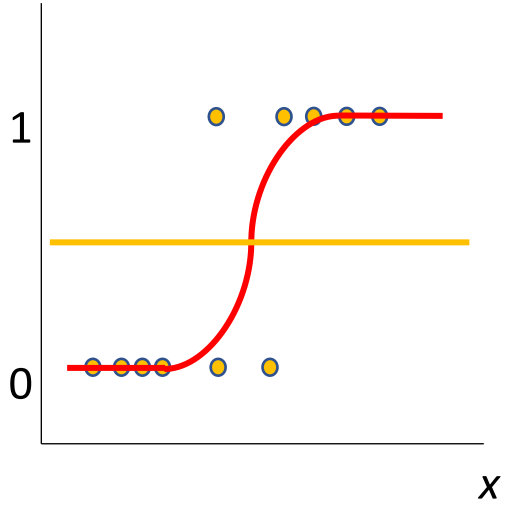
Logistic regression is similar to linear regression, but the coefficients predict the log(odds) ☺

But the residuals are infinity! Can't use least squares to minimize the residuals ☹

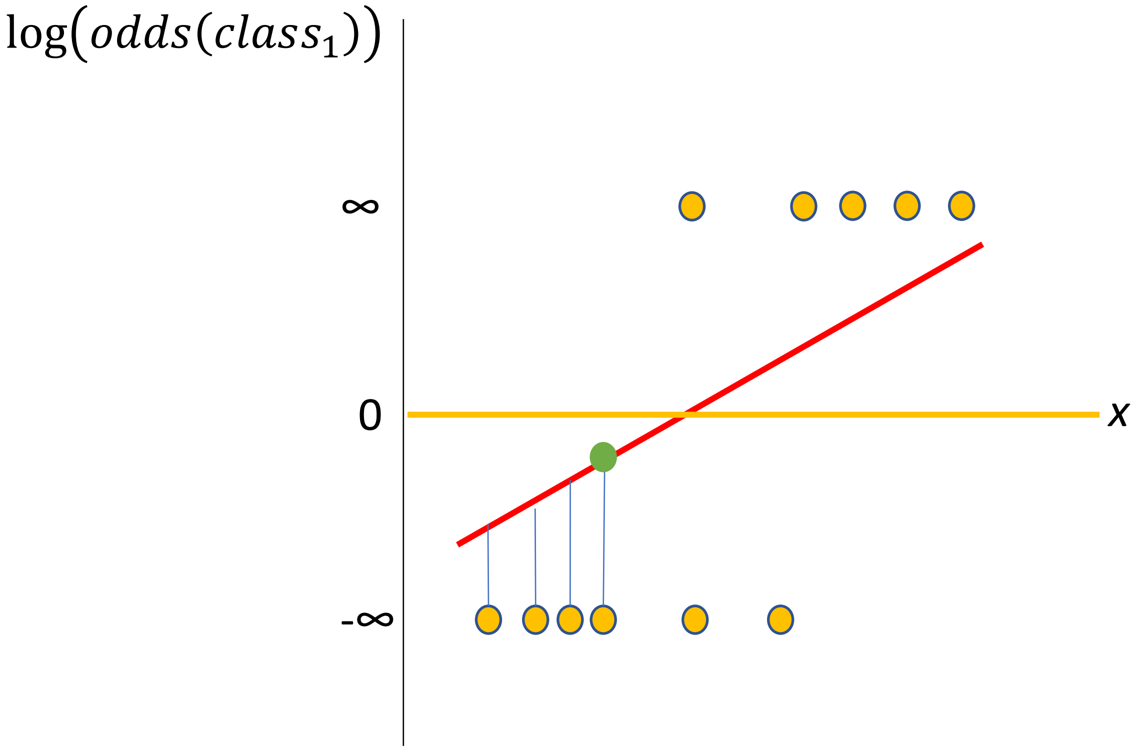
Classification as regression

$$\log(odds(class_1)) = \log\left(\frac{P(Class_1)}{1 - P(Class_1)}\right) \Rightarrow P(Class_1) = \frac{e^{\log(odds(Class_1))}}{1 + e^{\log(odds(Class_1))}}$$





Classification as regression



1. Initialize $\boldsymbol{\theta}$ to calculate best fitting line to $\log(odds_1))$ as: $\hat{z}(\boldsymbol{x}) = \boldsymbol{\theta x}$

2. Estimate likelihood of the data:

   Positive class samples: $P(Class_1) = \frac{e^{\hat{z}(x)}}{1+e^{\hat{z}(x)}}$

   Negative class samples: $1 - P(Class_1)$

3. Update $\boldsymbol{\theta}$ so to maximize the likelihood

Classification as regression

Maximum likelihood estimation

Do until convergence:

1. Calculate log-likelihood of the model

Likelihood: $$L(\boldsymbol{\theta}) = \prod P\left(\boldsymbol{x}^{(i)}\right)^{y^{(i)}} (1 - P\left(\boldsymbol{x}^{(i)}\right)^{1-y^{(i)}}$$ $$P\left(\boldsymbol{x}^{(i)}\right) = \frac{e^{\boldsymbol{x}^{(i)}\boldsymbol{\theta}}}{1 + e^{\boldsymbol{x}^{(i)}\boldsymbol{\theta}}}$$

Log likelihood: $$\mathcal{L}(\theta) = \log\left(\mathrm{L}(\boldsymbol{\theta})\right)$$

2. Calculate derivatives $$\frac{\partial \mathcal{L}(\theta)}{\partial \theta} = \sum \frac{y^{(i)}}{P(x^{(i)})} \frac{\partial P\left(x^{(i)}\right)}{\partial \theta} - \frac{\left(1 - y^{(i)}\right)}{1 - P(x^{(i)})} \frac{\partial P\left(x^{(i)}\right)}{\partial \theta} = \sum x^{(i)} \left(y^{(i)} - P\left(x^{(i)}\right)\right)$$

3. Update parameters: $$\boldsymbol{\theta} = \boldsymbol{\theta} + \frac{\partial \mathcal{L}(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}$$

Performance evaluation

- When is a model good?
  - Provides acceptable accuracy when ”acceptable” can be defined?
  - It performs better than baseline or existing models?
- Context is usually needed to interpret a model
  - Lower bounds: simple baseline model that needs to be improved
  - Upper bound: best possible outcome

Performance evaluation

- Lower bound:
  - No-information (or zero-information) prediction function
    - Classification: always predicts the same class
    - Regression: predicts the average value
  - Single-feature prediction functions
    - Train basic model (e.g., linear regression, threshold…) with one single feature at a time and use as comparator
  - Simple regularized linear model (regression, linear classifier)
    - If your model does not beat simple models, there may not be enough data or parameter running is suboptimal.
- Upper bound:
  - Oracle model: best performing model
    - Train same model on the testing dataset and evaluate fitting performance

Performance evaluation

- Confusion matrix for binary classification

		Actual	
		Class 0	Class 1
Predicted	Class 0	a	b
	Class 1	c	d

Performance evaluation

- Confusion matrix for binary classification

		Actual	
		Class 0	Class 1
Predicted	Class 0	a	b
	Class 1	c	d

Accuracy $\frac{a+d}{a+b+c+d}$ Fraction of correct predictions

Performance evaluation

- Confusion matrix for binary classification

		Actual	
		Class 0	Class 1
Predicted	Class 0	a	b
	Class 1	c	d

Error rate $\dfrac{b+c}{a+b+c+d}$ Fraction of incorrect predictions

Performance evaluation

- Confusion matrix for binary classification

		Actual	
		Class 0	Class 1
Predicted	Class 0	1283	5
	Class 1	0	0

Accuracy: 99%

Accuracy and error rate do not quantify performance on one specific class

Performance evaluation

- Focus on the positive class

		Actual	
		Class +	Class -
Predicted	Class +	TP	FP
	Class -	FN	TN

Let's assume Class 1 is a <u>positive</u> class:

- Less frequent
- More significant
- Example: cancer diagnosis

Precision $\frac{TP}{TP + FP}$ Accuracy in the positive predictions only (aka positive predicted value)

Recall $\frac{TP}{TP + FN}$ Accuracy in the real positive class only (aka true positive rate, or **<u>sensitivity</u>**)

<u>Note</u>: a low value of FP or FN translates into large precision or recall, respectively. Evaluation of a single metric can be misleading

Performance evaluation

- Focus on the positive class

		Actual	
		Class +	Class -
Predicted	Class +	TP	FP
	Class -	FN	TN

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

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

F1-score $$2 \cdot \frac{1}{\frac{1}{\text{recall}} + \frac{1}{\text{precision}}} = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}$$ Harmonic mean between precision and recall

F_β-score $$(1 + \beta^2) \cdot \frac{\text{precision} \cdot \text{recall}}{\beta^2 \cdot \text{precision} + \text{recall}}$$ Weighted harmonic mean between precision and recall

Performance evaluation

- Focus on the negative class (e.g., useful for screening)

		Actual	
		Class +	Class -
Predicted	Class +	TP	FP
	Class -	FN	TN

Negative predictive value $\frac{TN}{TN + FN}$ Accuracy in the negative predictions only

Specificity $\frac{TN}{TN + FP}$ Accuracy in the real negative class only (aka true negative rate)

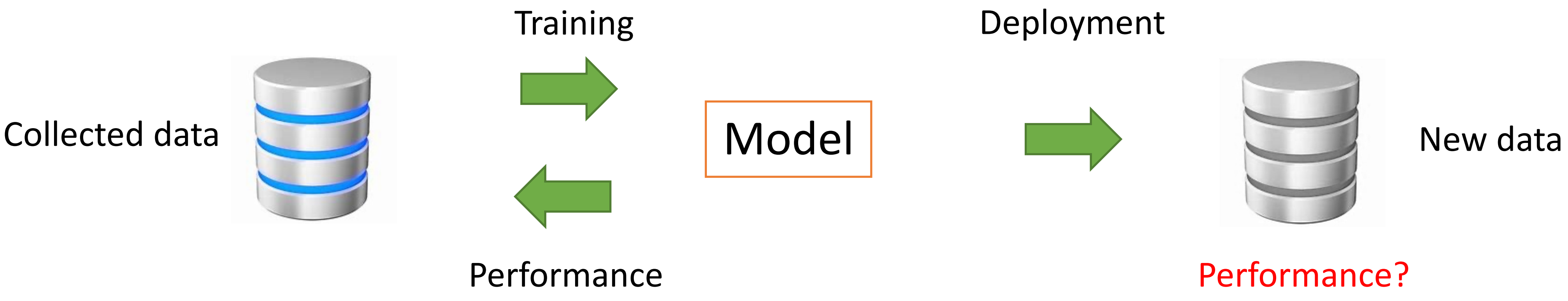
Note: Evaluation of a single metric can be misleading

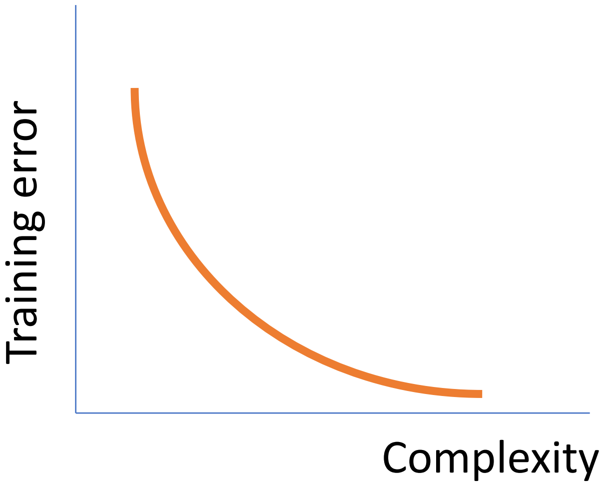
Performance evaluation

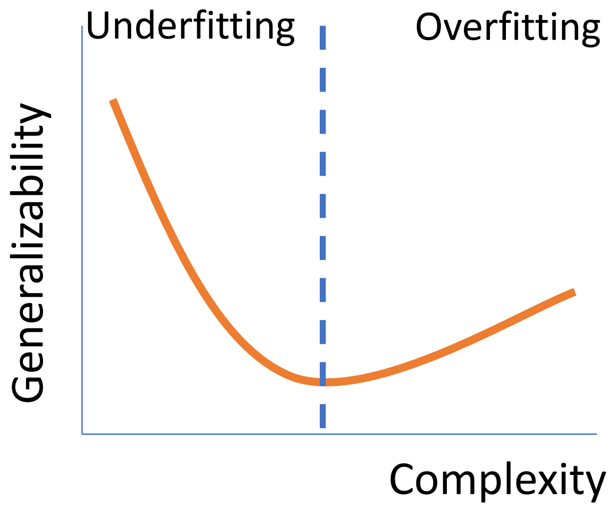
	Positive class focus	**Negative class focus**
Focus on real classes	Sensitivity/recall	Specificity
Focus on predictions	Precision	NPV

		Predicted			
		Positive (PP)	**Negative (PN)**		
Actual	**Positive (P)**	True positive (TP)	False negative (FN)	True positive rate (TPR), recall, **sensitivity** (SEN) = TP/P = 1 − FNR	False negative rate (FNR) = FN/P = 1 − TPR
	Negative (N)	False positive (FP)	True negative (TN)	False positive rate (FPR)= FP/N = 1 − TNR	True negative rate (TNR) **specificity** (SPC) = TN/N = 1 − FPR
	Prevalence = P/P + N	**Positive predictive value** (PPV), precision = TP/PP = 1 − FDR	False omission rate (FOR) = FN/PN = 1 − NPV	Positive likelihood ratio (LR+) = TPR/FPR	Negative likelihood ratio (LR−) = FNR/TNR
	Accuracy (ACC) = TP + TN/P + N	False discovery rate (FDR) = FP/PP = 1 − PPV	**Negative predictive value** (NPV) = TN/PN = 1 − FOR		
		F_1 score = 2 PPV × TPR/PPV + TPR = 2 TP/2 TP + FP + FN			

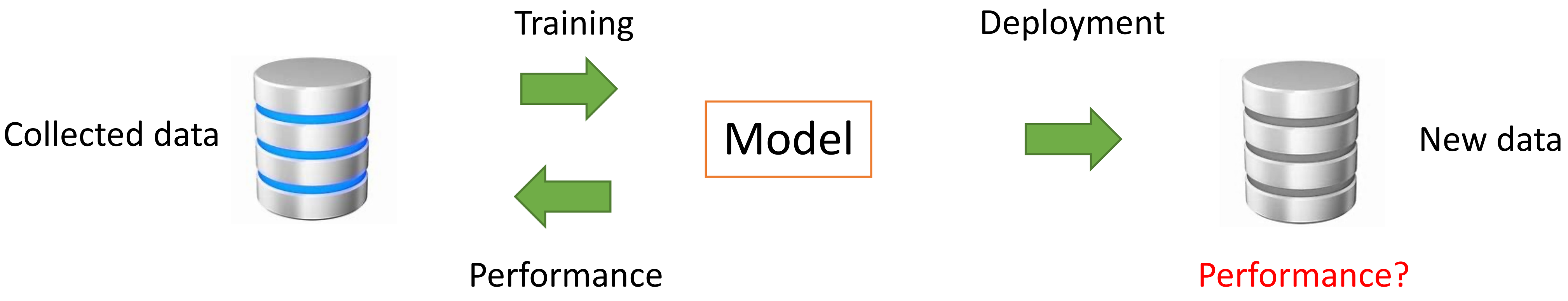
Performance evaluation

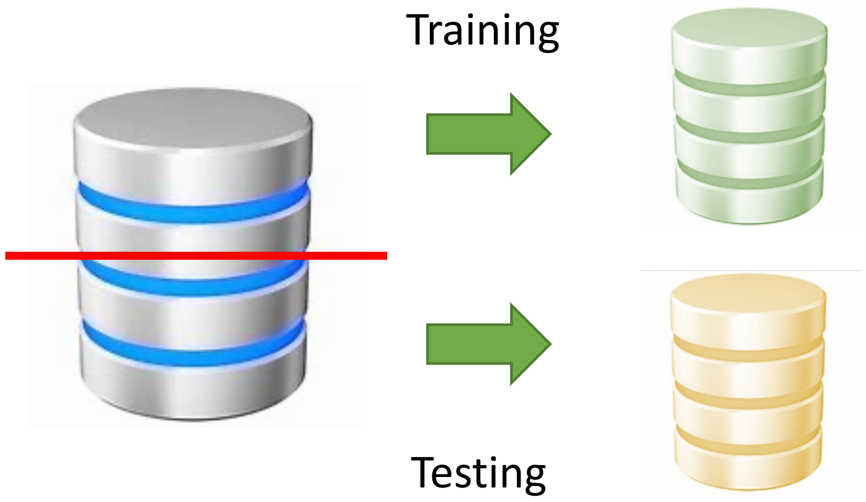






Performance evaluation

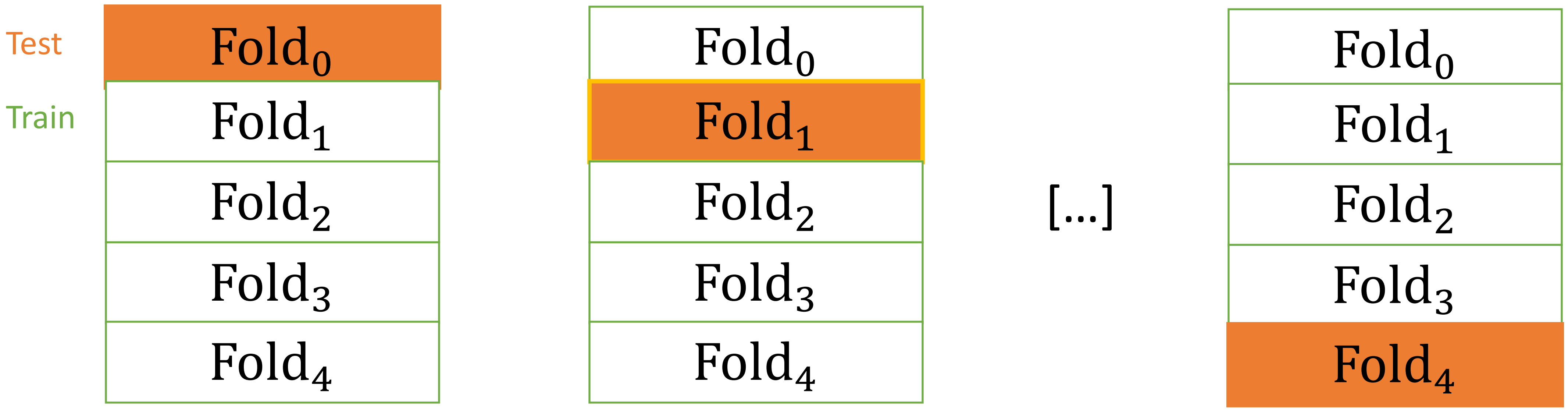




Data split bias

Performance evaluation

- K-fold cross-validation

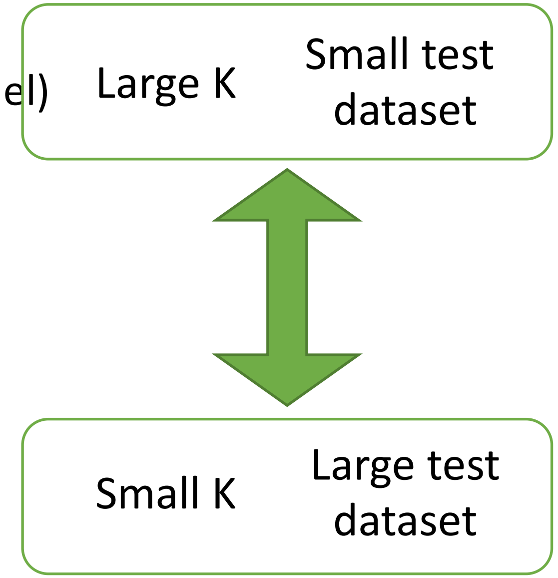


Performance evaluation

❑ K-fold cross-validation

- **IMPORTANT**: Only the dataset can change between folds

- Large K:
    - Each fold is trained using a similar training dataset (similar to production model)
    - Larger training dataset reduces likelihood of overfitting
    - High computational cost
    - Higher variability in the outcomes as a results of more evaluated models



- Small K:
    - Lower computational cost
    - Higher likelihood of overfitting
    - Models tend to have more differences between them depending on complexity

Performance evaluation

❑ K-fold cross-validation

- Class balance:
  - Consider stratified cross-validation: keep class balance in all folds
- Data aggregation
  - Class predictions (e.g., decision trees)
    - Save confusion matrix for every fold
    - Add confusion matrices and evaluate performance
  - Continuous predictions (e.g., logistic regression)
    - Save predictions
    - Evaluate diagnostic ability

Performance evaluation

- Evaluating diagnostic ability of classifiers providing quantitative predictions

Classification thresholds

	Predictions	0.00	0.25	0.50	0.75	1.00
True positives	0.96	TP	TP	TP	TP	FN
	0.40	TP	TP	FN	FN	FN
	0.65	TP	TP	TP	FN	FN
	0.89	TP	TP	TP	TP	FN
True negatives	0.10	FP	TN	TN	TN	TN
	0.52	FP	FP	FP	TN	TN
	0.05	FP	TN	TN	TN	TN
	0.15	FP	TN	TN	TN	TN

Performance evaluation

- Evaluating diagnostic ability of classifiers providing quantitative predictions

0.00	0.25	0.50	0.75	1.00
TP	TP	TP	TP	FN
TP	TP	FN	FN	FN
TP	TP	TP	FN	FN
TP	TP	TP	TP	FN
FP	TN	TN	TN	TN
FP	FP	FP	TN	TN
FP	TN	TN	TN	TN
FP	TN	TN	TN	TN

	Positive class focus	Negative class focus
Focus on real classes	Sensitivity/recall	Specificity
Focus on predictions	Precision	NPV

Threshold	**0.00**	**0.25**	**0.50**	**0.75**	**1.00**
Sensitivity	1	1	0.75	0.5	0
Specificity	0	0.75	0.75	1	1
Precision	0.5	0.8	0.75	1	0

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

$$\text{Specificity} = \frac{\text{TN}}{\text{TN} + \text{FP}}$$

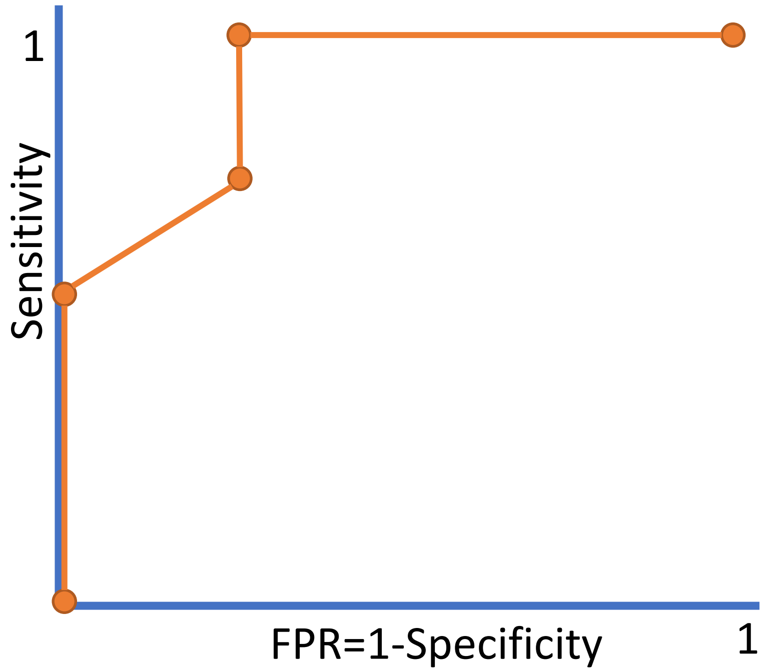
$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

Performance evaluation

- Evaluating diagnostic ability of classifiers providing quantitative predictions

Threshold	**0.00**	**0.25**	**0.50**	**0.75**	**1.00**
Sensitivity	1	1	0.75	0.5	0
Specificity	0	0.75	0.75	1	1

Receiver Operating Characteristic (ROC) Curve

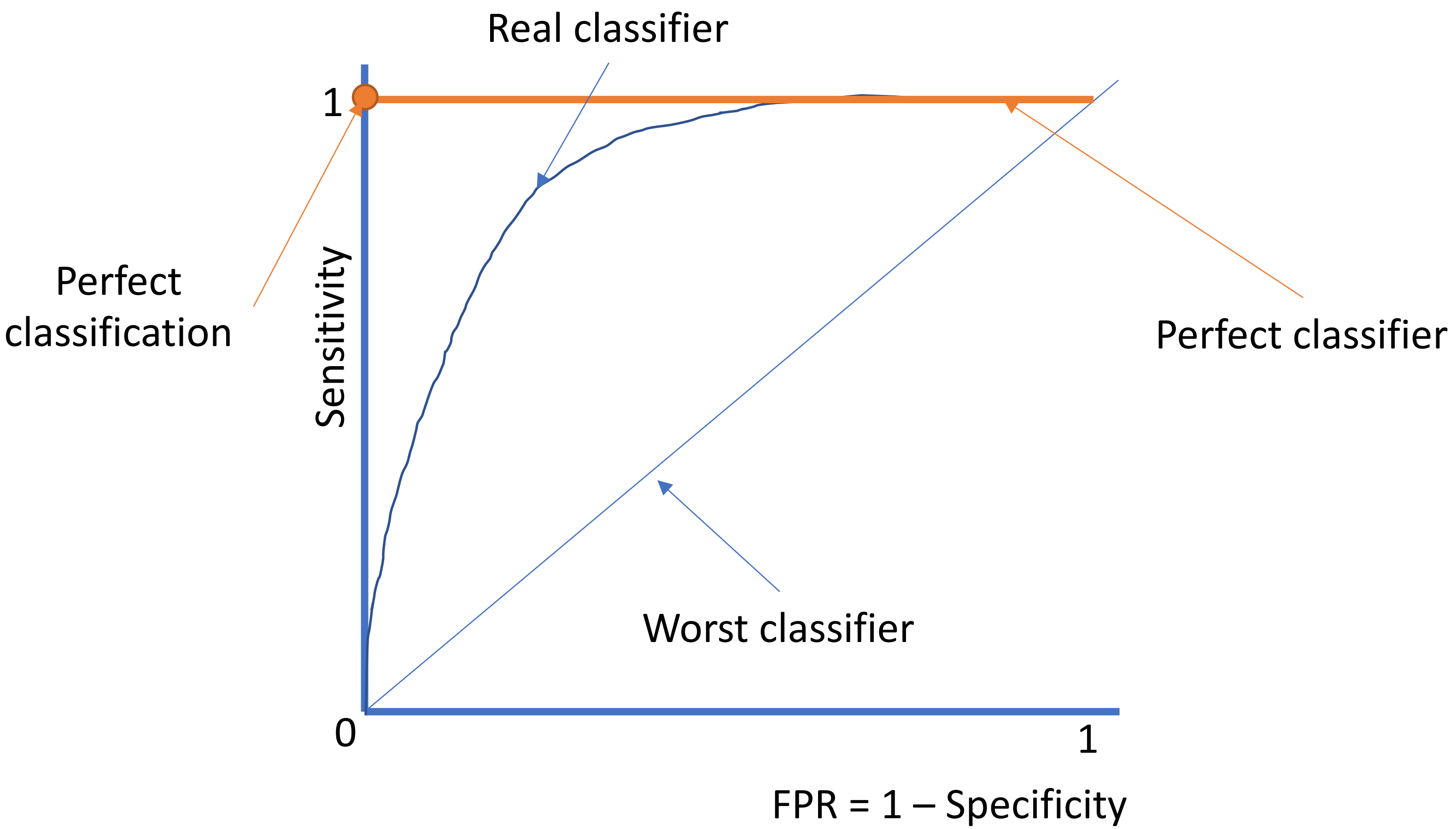


Represents classifier's performance on the true positive and negative classes for different "operating points" or binary thresholds

Performance evaluation

- Evaluating diagnostic ability of classifiers providing quantitative predictions

Receiver Operating Characteristic (ROC) Curve

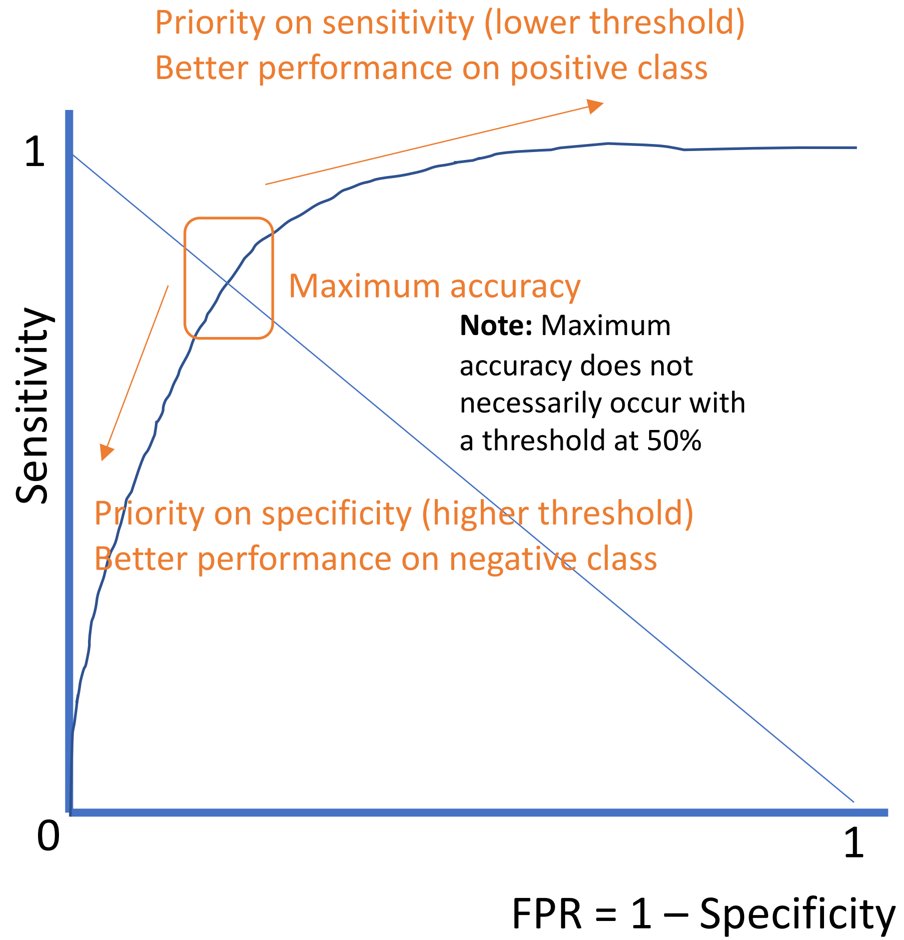


Performance evaluation

❑ Evaluating diagnostic ability of classifiers providing quantitative predictions

Receiver Operating Characteristic (ROC) Curve

- Focus on accuracy (weighted average between sensitivity and specificity)
- Invariant to class imbalance



Area under the ROC curve (AUC)

- Measures the overall performance of the classifier.
- It's equivalent to Mann-Whitney U-test

$$AUC = U(N_0 * N_1)$$

[Mason, S.J. and Graham, N.E. (2002), Areas beneath the relative operating characteristics (ROC) and relative operating levels (ROL) curves: Statistical significance and interpretation. Q.J.R. Meteorol. Soc., 128: 2145 2166. https://doi.org/10.1256/003590002320603584]

Performance evaluation

❑ Evaluating diagnostic ability of classifiers providing quantitative predictions

Threshold	0.00	0.25	0.50	0.75	1.00
Sensitivity	1	1	0.75	0.5	0
Precision	0.5	0.8	0.75	1	

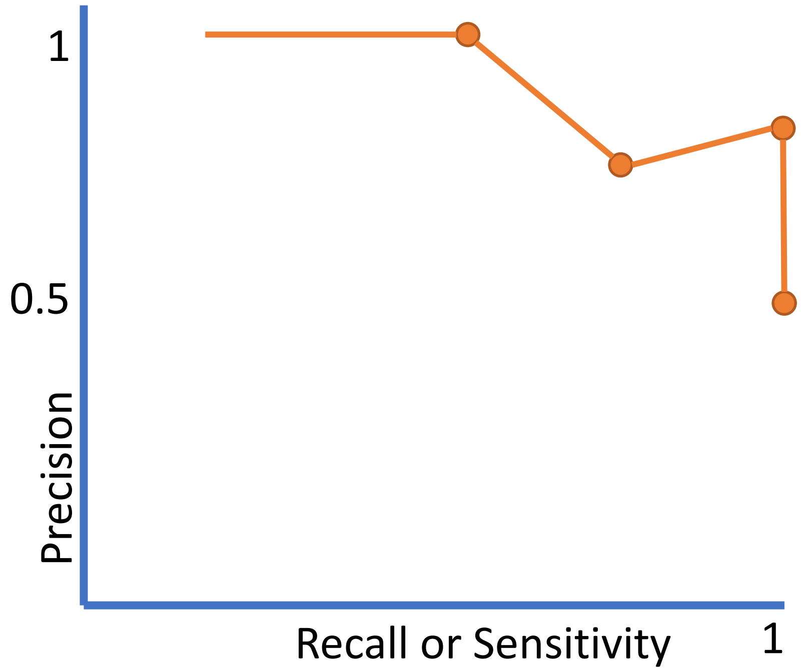
Area under the ROC curve (AUC)

- Measures the overall performance of the classifier.

Represents classifier's performance on the positive class

Precision-Recall curve

- Focus on F1 score (harmonic mean between precision and recall)
- Affected by class imbalance



Next class

- Have a look at:

  - Sklearn.metrics: ROC curve analysis

  - Sklearn.model_selection: documentation on cross-validation

  - Sklearn.linear_model: logistic regression