Foundations of Machine Learning
CentraleSupélec — Fall 2017

4. Model evaluation & selection

Chloé-Agathe Azencott
Centre for Computational Biology, Mines ParisTech
chloe-agathe.azencott@mines-paristech.fr

Practical matters

- You should have received an email from me on Tuesday
- **Partial solution to Lab 1** at the end of the slides of Chapter 3.
- Pointers/refreshers re: **(scientific) python**
  - `http://www.scipy-lectures.org/`
  - `https://github.com/chagaz/ml-notebooks/`
    → `lsml2017`
- Yes, I only put the slides online **after** the lecture.

Generalization

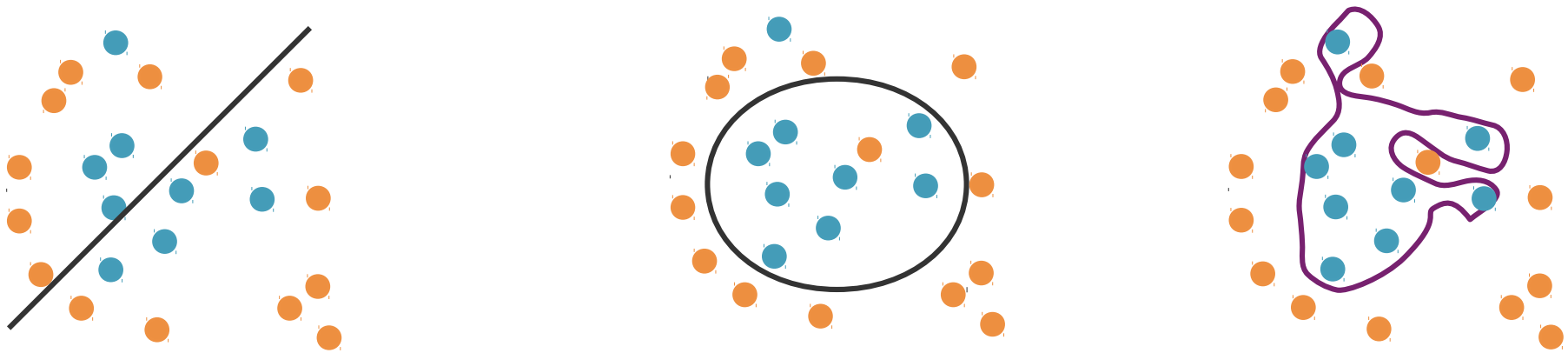
A good and useful approximation

- It's easy to build a model that performs well on the training data
- But how well will it perform on **new data?**
- "Predictions are hard, especially about the future" — Niels Bohr.
  - Learn models that **generalize** well
  - Evaluate whether models generalize well.

Noise in the data

- Imprecision in **recording the features**
- **Errors in labeling** the data points (**teacher noise**)
- **Missing features** (**hidden** or **latent**)
- Making no errors on the training set might not be possible.

Models of increasing complexity



Noise and model complexity

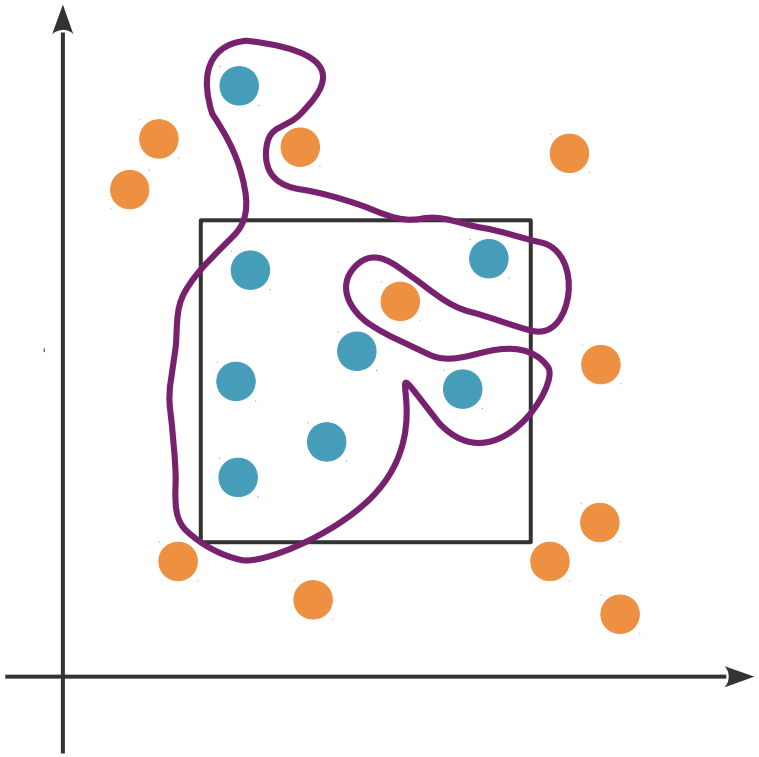
- **Use simple models!**
  - Easier to **use**

    lower computational complexity
  - Easier to **train**

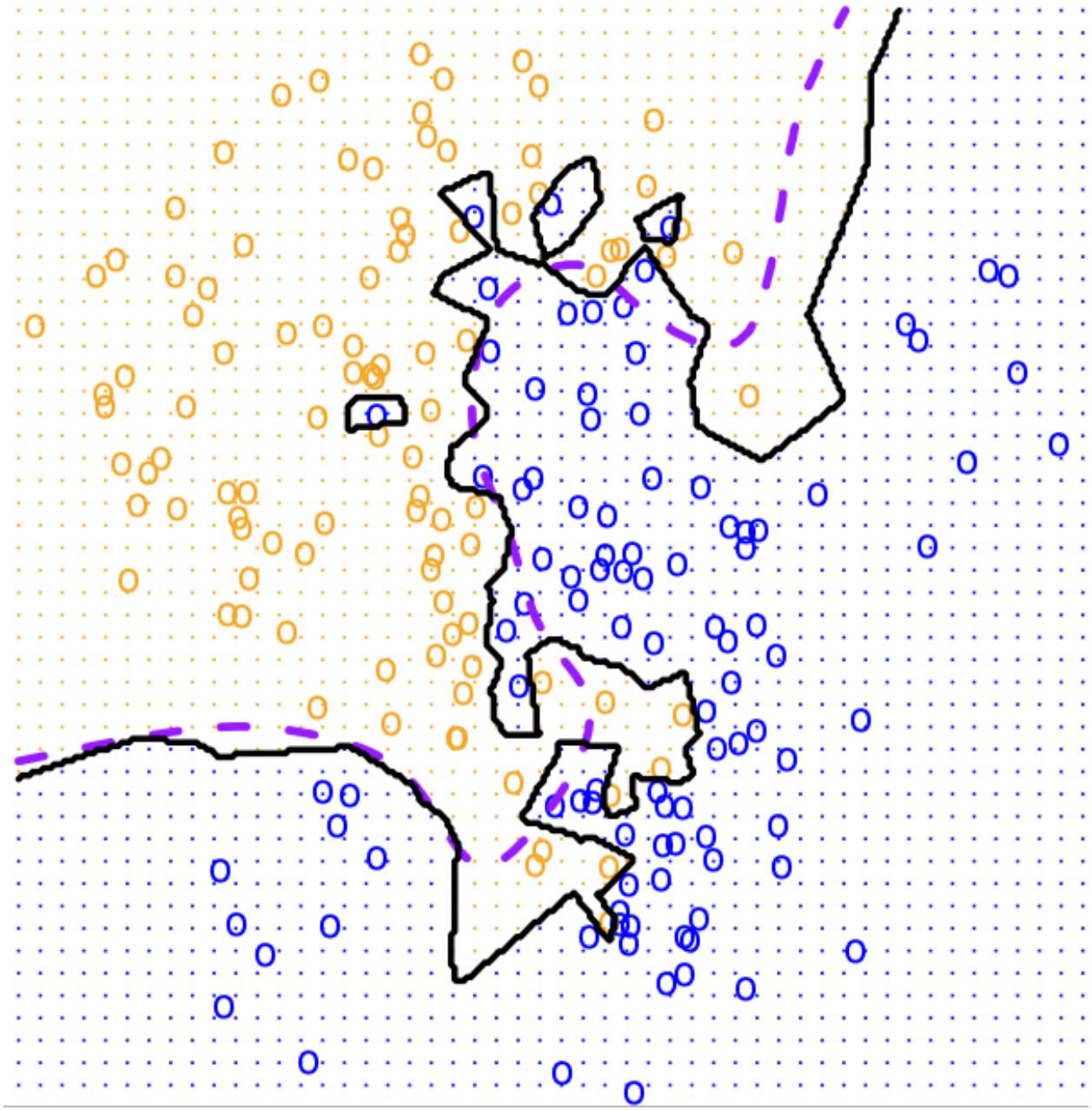
    lower space complexity
  - Easier to **explain**

    more interpretable
  - **Generalize better**

    Occam's razor: simpler explanations are more plausible.

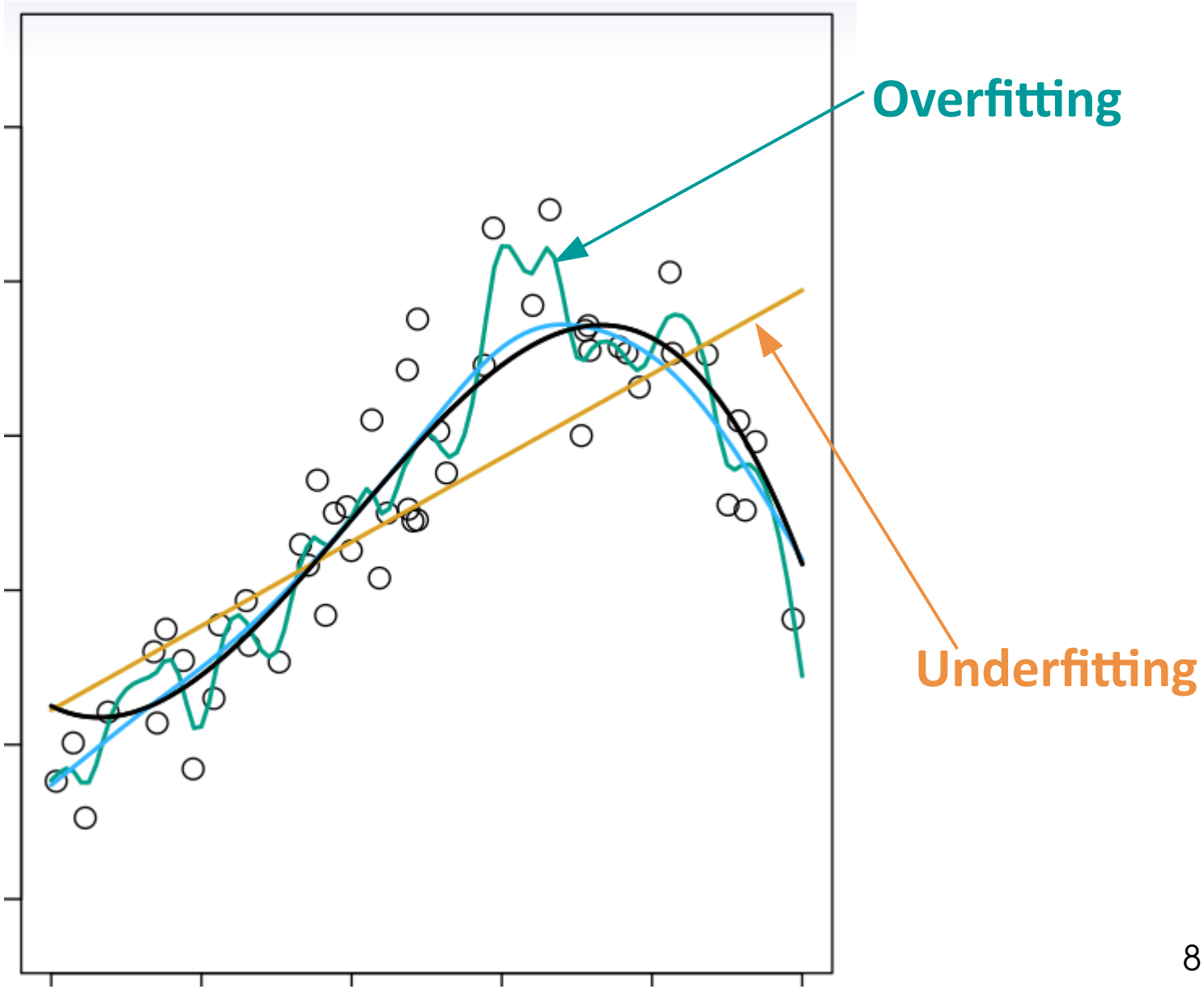


Overfitting

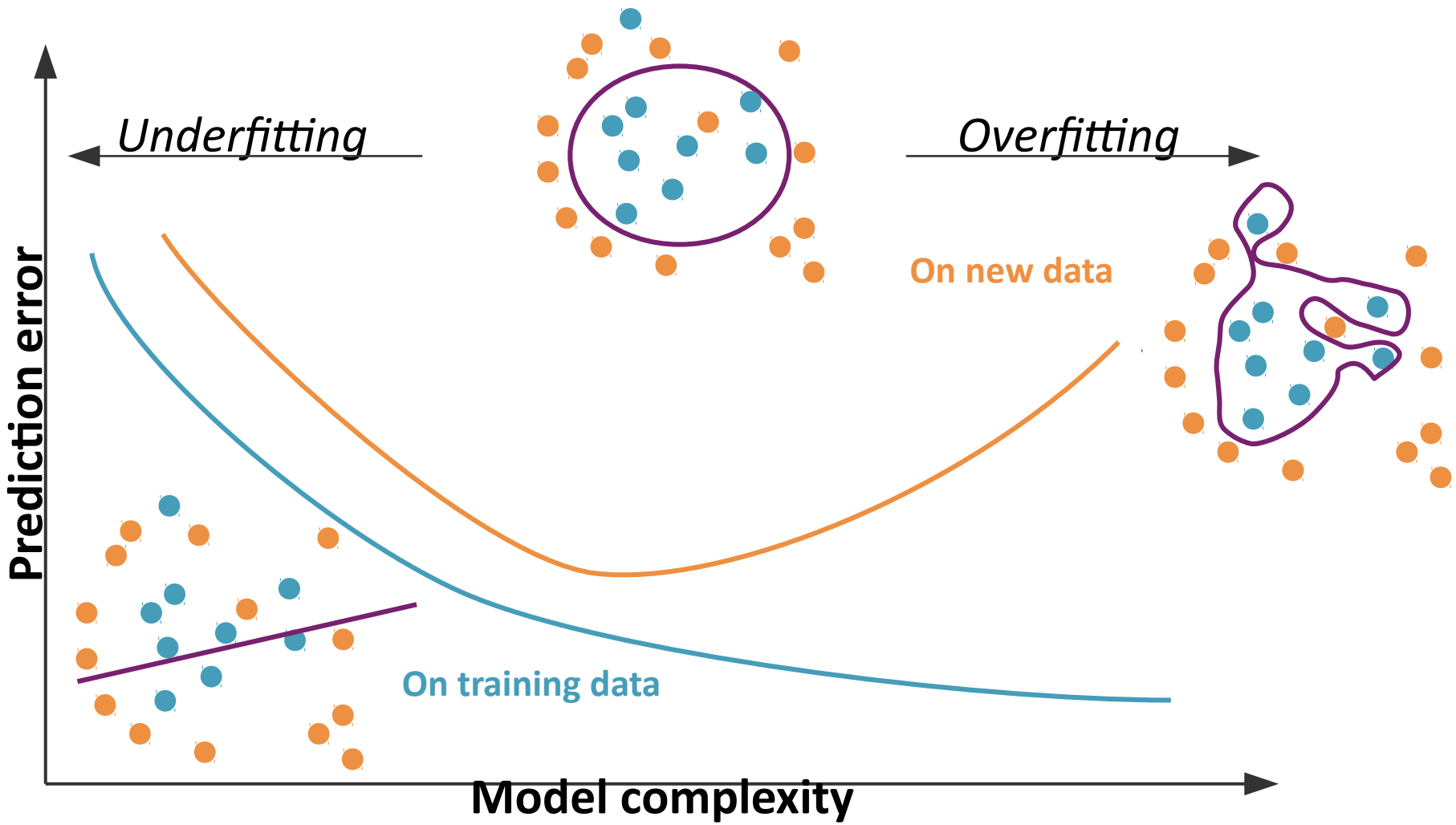


- What are the empirical errors of the black and purple classifiers?
- Which model seems more likely to be correct?

Overfitting & Underfitting (Regression)



Generalization error vs. model complexity



Bias-variance tradeoff

- **Bias:** difference between the expected value of the estimator and the true value being estimated.

$$\mathrm{Bias}(f(\boldsymbol{x})) = \mathbb{E}[f(\boldsymbol{x}) - y]$$

  - A simpler model has a higher bias.
  - **High bias can cause underfitting.**
- **Variance:** deviation from the expected value of the estimates.

$$\mathrm{Var}(f(\boldsymbol{x})) = \mathbb{E}[(f(\boldsymbol{x}) - \mathbb{E}(f(\boldsymbol{x}))^2]$$

  - A more complex model has a higher variance.
  - **High variance can cause overfitting.**

Bias-variance decomposition

- $\mathrm{Bias}(f(\boldsymbol{x})) = \mathbb{E}[f(\boldsymbol{x}) - y]$
- $\mathrm{Var}(f(\boldsymbol{x})) = \mathbb{E}[(f(\boldsymbol{x}) - \mathbb{E}(f(\boldsymbol{x}))^2]$
- **Mean squared error:**

$$\begin{aligned} \mathrm{MSE}(f(\boldsymbol{x})) &= \mathbb{E}[(f(\boldsymbol{x}) - y)^2] \\ &= \mathrm{Var}(f(\boldsymbol{x})) + \mathrm{Bias}^2(f(\boldsymbol{x})) \end{aligned}$$

- Proof

Bias-variance decomposition

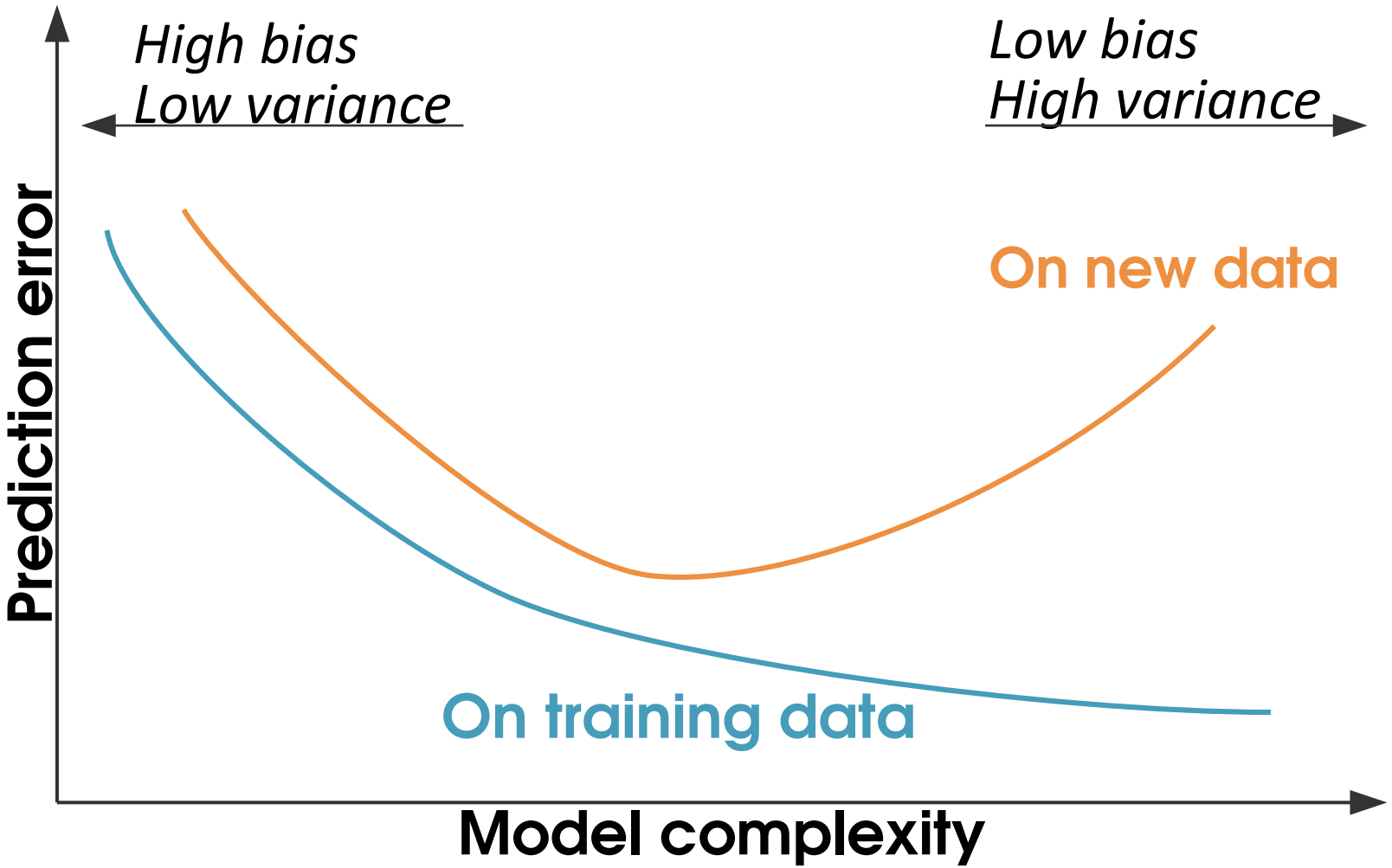
- $\mathrm{Bias}(f(\boldsymbol{x})) = \mathbb{E}[f(\boldsymbol{x}) - y]$
- $\mathrm{Var}(f(\boldsymbol{x})) = \mathbb{E}[(f(\boldsymbol{x}) - \mathbb{E}(f(\boldsymbol{x}))^2]$
- **Mean squared error:**

$$\begin{aligned}\mathrm{MSE}(f(\boldsymbol{x})) &= \mathbb{E}[(f(\boldsymbol{x}) - y)^2] \\ &= \mathrm{Var}(f(\boldsymbol{x})) + \mathrm{Bias}^2(f(\boldsymbol{x}))\end{aligned}$$

$$\begin{aligned}\mathbb{E}[(f(\boldsymbol{x}) - y)^2] &= \mathbb{E}[(f(\boldsymbol{x}) - \mathbb{E}[f(\boldsymbol{x})] + \mathbb{E}[f(\boldsymbol{x})] - y)^2] \\ &= \mathbb{E}[(f(\boldsymbol{x}) - \mathbb{E}[f(\boldsymbol{x})])^2] + \mathbb{E}[(\mathbb{E}[f(\boldsymbol{x})] - y)^2] + 2\mathbb{E}[(f(\boldsymbol{x}) - \mathbb{E}[f(\boldsymbol{x})])(\mathbb{E}[f(\boldsymbol{x})] - y)]\end{aligned}$$

$\mathbb{E}[f(\boldsymbol{x})]$ and y are determinist.

Generalization error vs. model complexity



Model selection & generalization

- **Well-posed problems:**
  - a solution exists;
  - it is unique;
  - the solution changes continuously with the initial conditions

Hadamard, on the mathematical modelisation of physical phenomena.

- Learning is an **ill-posed problem**:

  data helps carve out the hypothesis space

  but data is not sufficient to find a unique solution.

- Need for **inductive bias**

  assumptions about the hypothesis space

  model selection: choose the "right" inductive bias?

How do we decide a model is good?

Learning objectives

After this lecture you should be able to

design experiments to select and evaluate supervised machine learning models.

Concepts:

- training and testing sets;
- cross-validation;
- bootstrap;
- measures of performance for classifiers and regressors;
- measures of model complexity.

Supervised learning setting

- **Training set:** $\mathcal{D} = \{\boldsymbol{x}^i, y^i\}_{i=1,\ldots,n}$
- **Classification:** $y^i \in$?

Supervised learning setting

- **Training set:** $\mathcal{D} = \{\boldsymbol{x}^i, y^i\}_{i=1,\ldots,n}$
- **Classification:** $y^i \in \{0, 1\}$
- **Regression:** $y^i \in$?

Supervised learning setting

- **Training set:** $\mathcal{D} = \{\boldsymbol{x}^i, y^i\}_{i=1,\ldots,n}$
- **Classification:** $y^i \in \{0, 1\}$
- **Regression:** y^i ?
- Goal: Find $f \in \mathcal{F}$ such that $f(\boldsymbol{x}^i) \approx y^i$

Supervised learning setting

- **Training set:** $\mathcal{D} = \{\boldsymbol{x}^i, y^i\}_{i=1,\ldots,n}$
- **Classification:** $y^i \in \{0, 1\}$
- **Regression:** $y^i \in \mathbb{R}$
- Goal: Find $f \in \mathcal{F}$ such that $f(\boldsymbol{x}^i) \approx y^i$
- **Empirical error** of f on the training set, given a **loss**:

Supervised learning setting

- **Training set:** $\mathcal{D} = \{\boldsymbol{x}^i, y^i\}_{i=1,\ldots,n}$
- **Classification:** $y^i \in \{0, 1\}$
- **Regression:** $y^i \in \mathbb{R}$
- Goal: Find $f \in \mathcal{F}$ such that $f(\boldsymbol{x}^i) \approx y^i$
- **Empirical error** of f on the training set, given a **loss**:

$$E(f|\mathcal{D}) = \frac{1}{n}\sum_{i=1}^{n} \mathcal{L}(y^i, f(\boldsymbol{x}^i))$$

  - E.g. (classification)

Supervised learning setting

- **Training set:** $\mathcal{D} = \{\boldsymbol{x}^i, y^i\}_{i=1,\ldots,n}$
- **Classification:** $y^i \in \{0, 1\}$
- **Regression:** $y^i \in \mathbb{R}$
- Goal: Find $f \in \mathcal{F}$ such that $f(\boldsymbol{x}^i) \approx y^i$
- **Empirical error** of f on the training set, given a **loss**:

$$E(f|\mathcal{D}) = \frac{1}{n}\sum_{i=1}^{n} \mathcal{L}(y^i, f(\boldsymbol{x}^i))$$

  - E.g. (classification)

$$\mathcal{L}(y^i, f(\boldsymbol{x}^i)) = 1_{y^i \neq f(\boldsymbol{x}^i)}$$

  - E.g. (regression) ?

Supervised learning setting

- **Training set:** $\mathcal{D} = \{\boldsymbol{x}^i, y^i\}_{i=1,\ldots,n}$
- **Classification:** $y^i \in \{0, 1\}$
- **Regression:** $y^i \in \mathbb{R}$
- Goal: Find $f \in \mathcal{F}$ such that $f(\boldsymbol{x}^i) \approx y^i$
- **Empirical error** of f on the training set, given a **loss**:

$$E(f|\mathcal{D}) = \frac{1}{n}\sum_{i=1}^{n} \mathcal{L}(y^i, f(\boldsymbol{x}^i))$$

  - E.g. (classification)

$$\mathcal{L}(y^i, f(\boldsymbol{x}^i)) = 1_{y^i \neq f(\boldsymbol{x}^i)}$$

  - E.g. (regression)

$$\mathcal{L}(y^i, f(\boldsymbol{x}^i)) = \left(y^i - f(\boldsymbol{x}^i)\right)^2$$

Generalization error

- The empirical error on the training set is a poor estimate of the **generalization error** (expected error on new data)

  If the model is overfitting, the generalization error can be arbitrarily large.

- We would like to estimate the generalization error on **new** data, which we do not have.

Validation sets

- Choose the model that performs best on a **validation set separate from the training set**.

Training	Validation

- Because we have not used the validation data at any point during training, the validation set can be considered "new data" and **the error on the validation set is an estimation of the generalization error.**

Model selection

- What if we want to choose among k models?
  - Train each model on the train set
  - Compute the prediction error of each model on the validation set
  - Pick the model with the smallest prediction error on the validation set.
- What is the generalization error?
  - We don't know!
  - Validation data was used to select the model
  - We have "cheated" and looked at the validation data: it is not a good proxy for new, unseen data any more.

Validation sets

- Hence we need to set aside part of the data, the test set, that remains untouched during the entire procedure and on which we'll estimate the generalization error.

- Model **selection:** pick the best model.
- Model **assessment:** estimate its prediction error on new data.

Training	Validation	Test

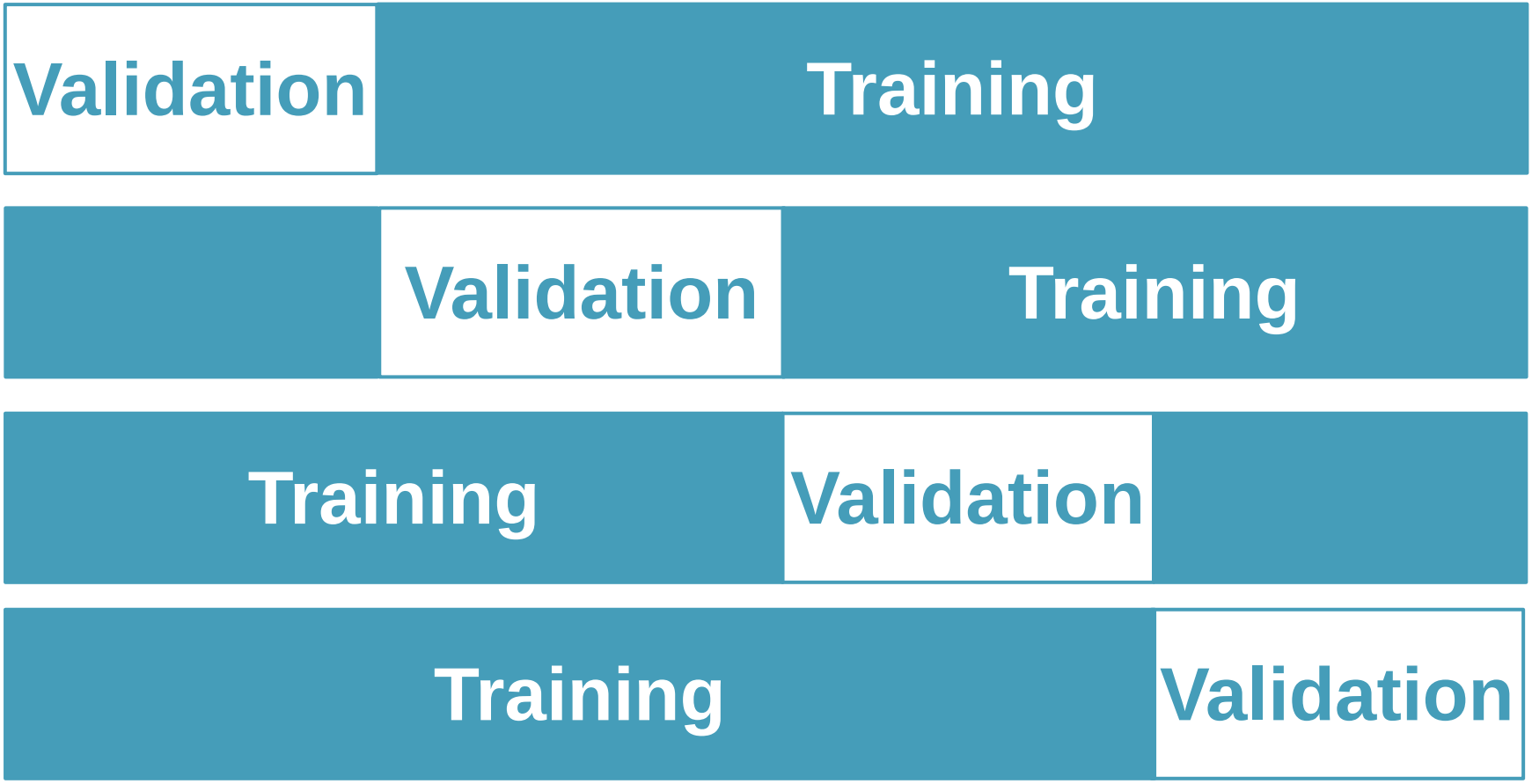
- **How much data** should go in each of the training, validation and test sets?
- How do we know we have **enough data** to evaluate the prediction and generalization errors?

- **Empirical evaluation with sample re-use**
  - cross-validation
  - bootstrap
- **Analytical tools**
  - Mallow's Cp, AIC, BIC
  - MDL.

Sample re-use

Cross-validation

- Cut the training set in k separate **folds.**
- For each fold, train on the (k-1) remaining folds.



Cross-validated performance

- Cross-validation estimate of the prediction error

$$CV(f) = \frac{1}{n}\sum_{i=1}^{n} L(y^i, f_{k(i)}(\boldsymbol{x}^i))$$

Computed with the k(i)-th part of the data removed. k(i) = fold in which i is.

or: $$CV(f) = \frac{1}{k}\sum_{l=1}^{k} E(f|D_l)$$

Fold l

- Estimates the **expected prediction error**

$$\mathrm{Err} = \mathbb{E}[L(Y, f(X))]$$

Y, X: (independent) test sample

Issues with cross-validation

- **Training set size** becomes (K-1)n/K

  Why is this a problem?

Issues with cross-validation

- **Training set size** becomes (K-1)n/K
  - small training set ⇒ biased estimator of the error
- **Leave-one-out cross-validation:** K = n
  - approximately **unbiased estimator** of the expected prediction error
  - potential **high variance** (the training sets are very similar to each other)
  - **computation** can become burdensome (n repeats)
- In practice: set **K = 5 or K = 10.**

Bootstrap

- **Randomly draw datasets with replacement** from the training data
- **Repeat B times** (typically, B=100) ⇒ B models
- **Leave-one-out bootstrap error:**
  - For each training point i, predict with the $b_i < B$ models that did not have i in their training set
  - Average prediction errors
- Each training set contains

Bootstrap

- **Randomly draw datasets with replacement** from the training data
- **Repeat B times** (typically, B=100) ⇒ B models
- **Leave-one-out bootstrap error:**
  - For each training point i, predict with the $b_i < B$ models that did not have i in their training set
  - Average prediction errors
- Each training set contains **0.632.n distinct examples**

  ⇒ same issue as with cross-validation

$$\begin{aligned} Pr(i \in X_k) &= 1 - (1 - \tfrac{1}{n})^n \\ &\sim 1 - e^{-1} \\ &= 0.632 \end{aligned} \qquad e^x = \lim_{n \to \infty} \left(1 + \frac{x}{n}\right)^n$$

Evaluating model performance

Classification model evaluation

- **Confusion matrix**

		True class	
		-1	+1
Predicted class	-1	**True Negatives**	**False Negatives**
	+1	**False Positives**	**True Positives**

- False positives (false alarms) are also called **type I errors**
- False negatives (misses) are also called **type II errors**

- **Sensitivity** = **Recall** = True positive rate (TPR)

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

 # positives

- **Specificity** = True negative rate (TNR)

$$\mathrm{TNR} = \frac{\mathrm{TN}}{\mathrm{FP} + \mathrm{TN}}$$

- **Precision** = Positive predictive value (PPV)

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

 # predicted positives

- **False discovery rate** (FDR)

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

- **Accuracy**

$$\mathrm{Acc} = \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{TP} + \mathrm{FN} + \mathrm{FP} + \mathrm{TN}}$$

- **F1-score** = harmonic mean of precision and sensitivity.

$$\mathrm{F1} = \frac{2\mathrm{TP}}{2\mathrm{TP} + \mathrm{FP} + \mathrm{FN}}$$

Example: Pap smear

- 4,000 apparently healthy women of age 40+
- Tested for cervical cancer through pap smear and histology (gold standard)

	Cancer	No cancer	Total
Positive test	190	210	400
Negative test	10	3590	3600
Total	200	3800	4000

- **What are the sensitivity, specificity, and PPV of the test?**

- **Sensitivity** = **Recall** = True positive rate (TPR)

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

- **Specificity** = True negative rate (TNR)

$$\mathrm{TNR} = \frac{\mathrm{TN}}{\mathrm{FP} + \mathrm{TN}}$$

- **Precision** = Positive predictive value (PPV)

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

	Cancer	No cancer	Total
Positive test	190	210	400
Negative test	10	3590	3600
Total	200	3800	4000

- In this population:

Sensitivity = 95.0 % Specificity = 94.5 % PPV = 47.5 %

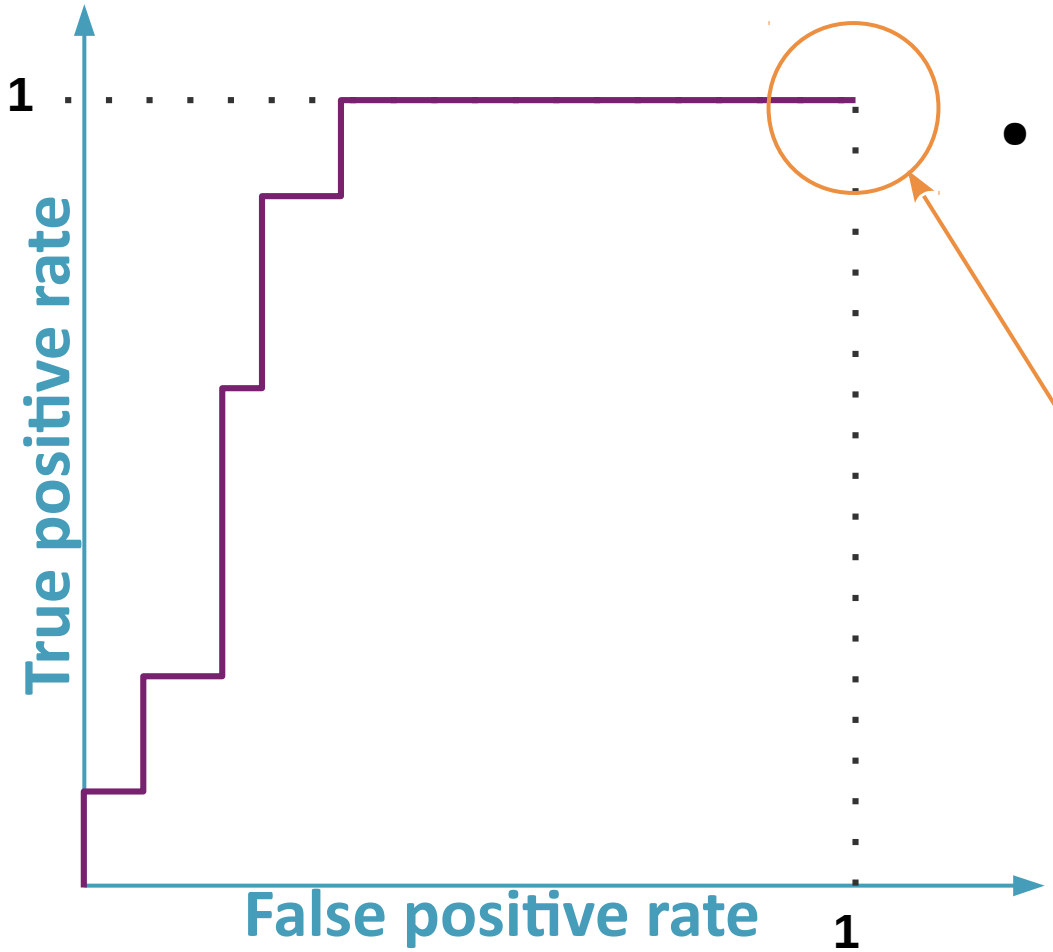
	Cancer	No cancer	Total
Positive test	190	210	400
Negative test	10	3590	3600
Total	200	3800	4000

- **Prevalence** of the disease = 200/4000 = 0.05
- P(cancer|positive test) = PPV = **47.5 %**
- P(no cancer|negative test) = 3590/3600 = **99.7 %**

- Poor **diagnosis** tool
- Good **screening** tool

ROC curves

- ROC = Receiver-Operator Characteristic.
- Summarized by the area under the curve (AUROC).

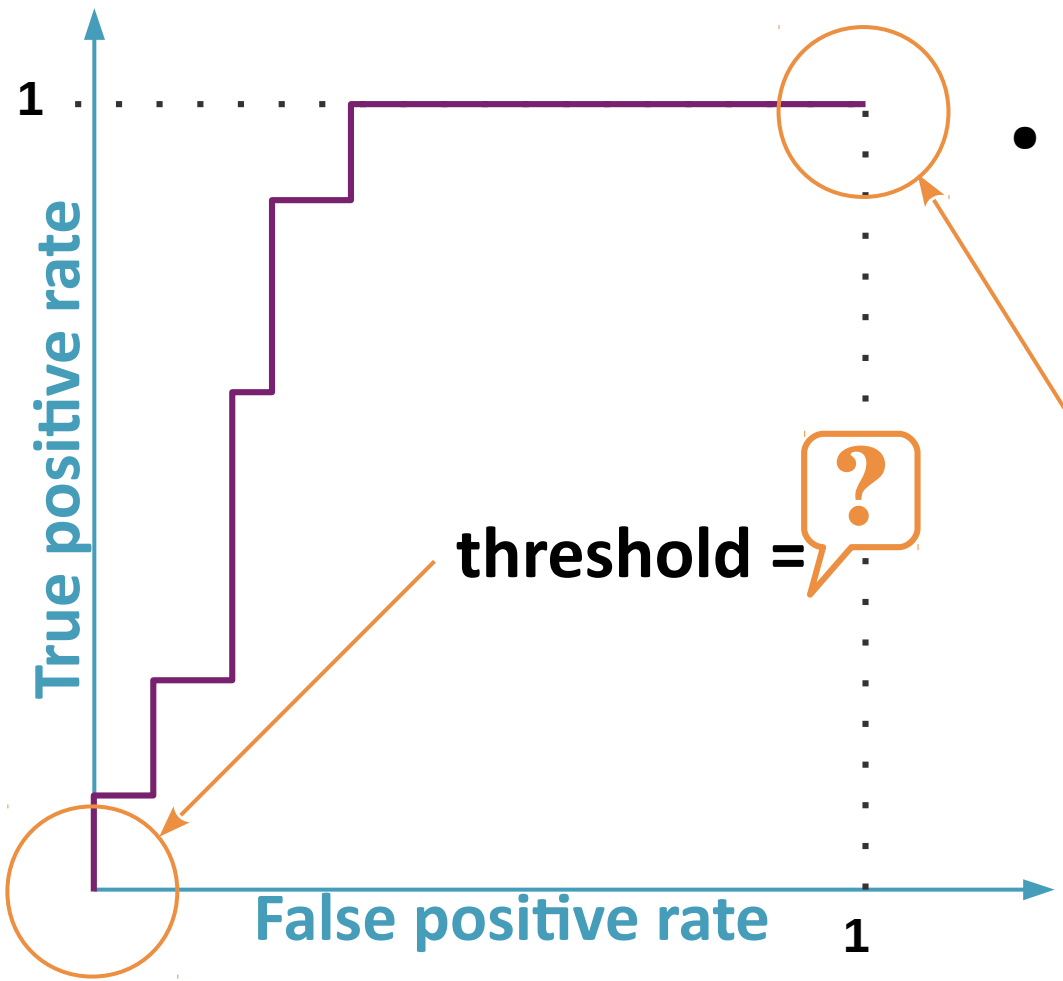


- Plot TPR vs FPR for all possible thresholds.

threshold = ?

ROC curves

- ROC = Receiver-Operator Characteristic.
- Summarized by the area under the curve (AUROC).

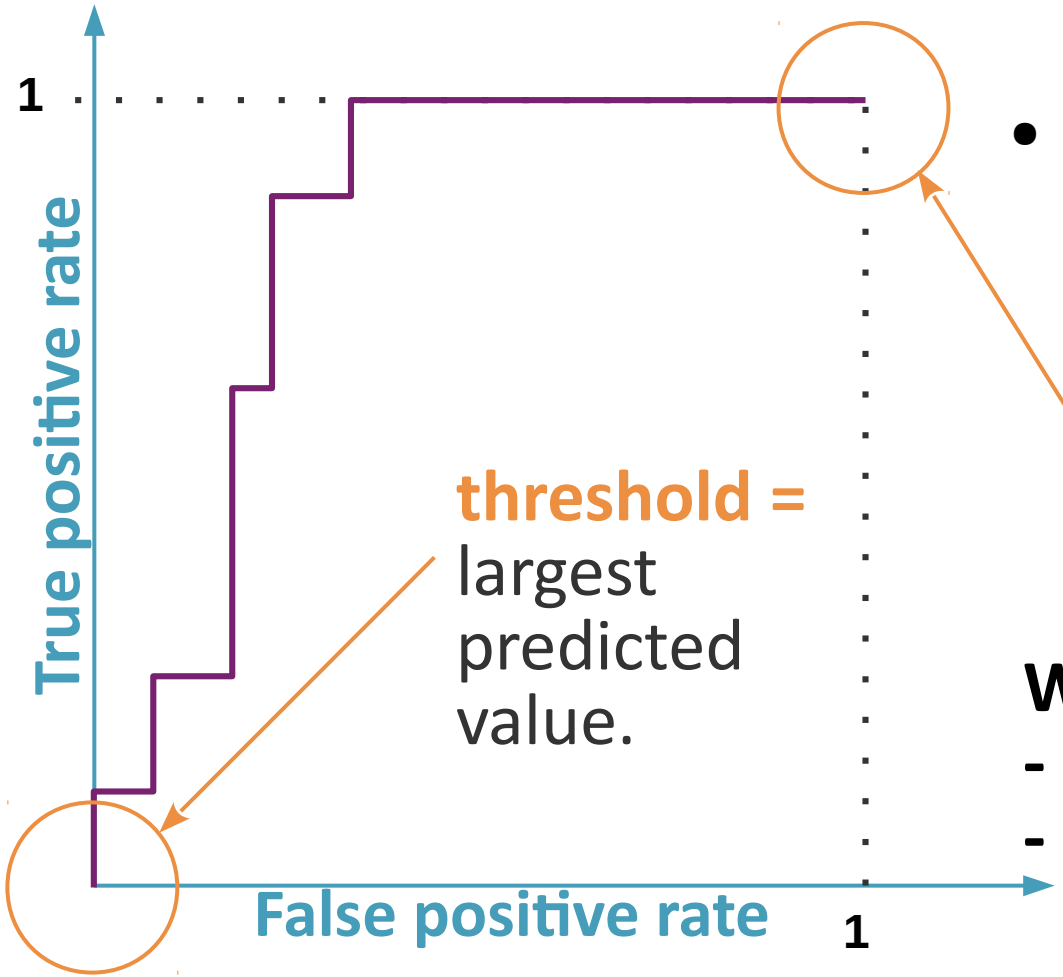


- Plot TPR vs FPR for all possible thresholds.

threshold = smallest predicted value.

ROC curves

- ROC = Receiver-Operator Characteristic.
- Summarized by the area under the curve (AUROC).



- Plot TPR vs FPR for all possible thresholds.

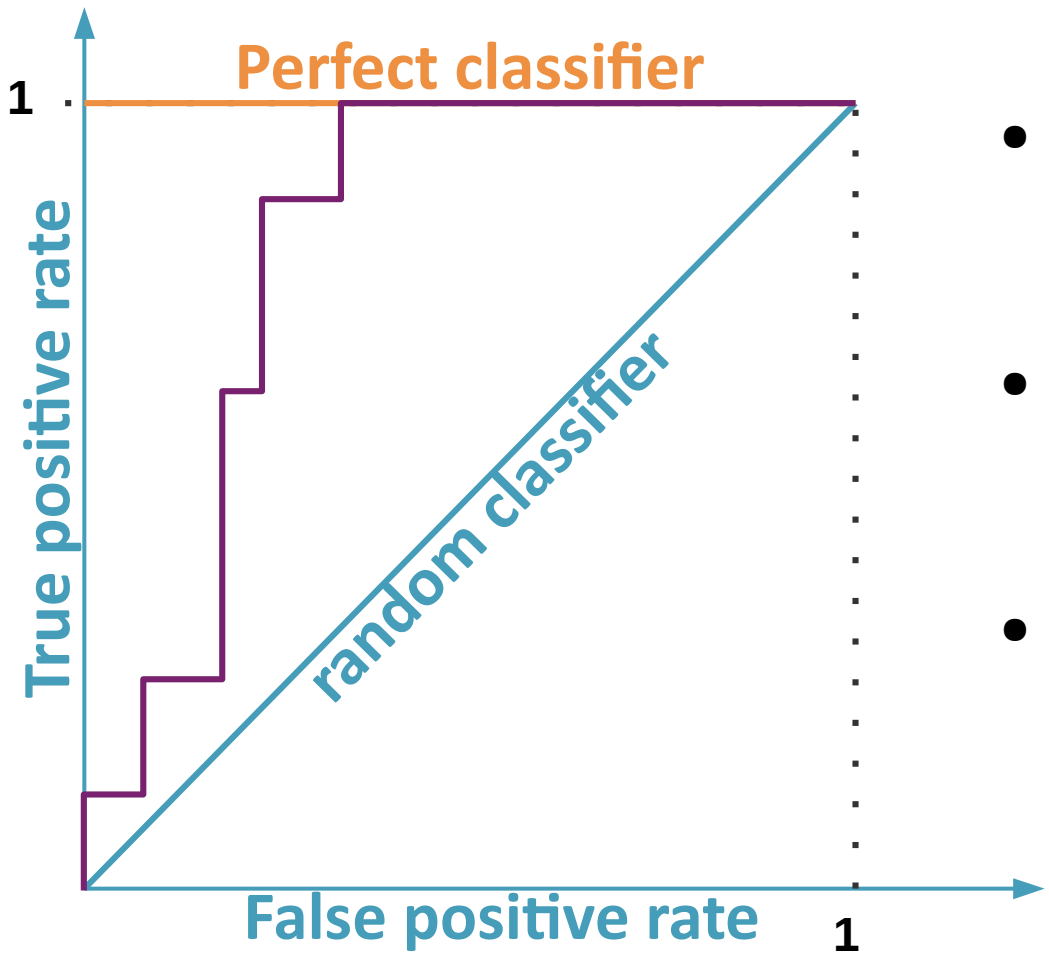
threshold = largest predicted value.

threshold = smallest predicted value.

What is the ROC curve of:
- a random classifier?
- a perfect classifier?

ROC curves

- ROC = Receiver-Operator Characteristic.
- Summarized by the area under the curve (AUROC).



- **Perfect classifier:**

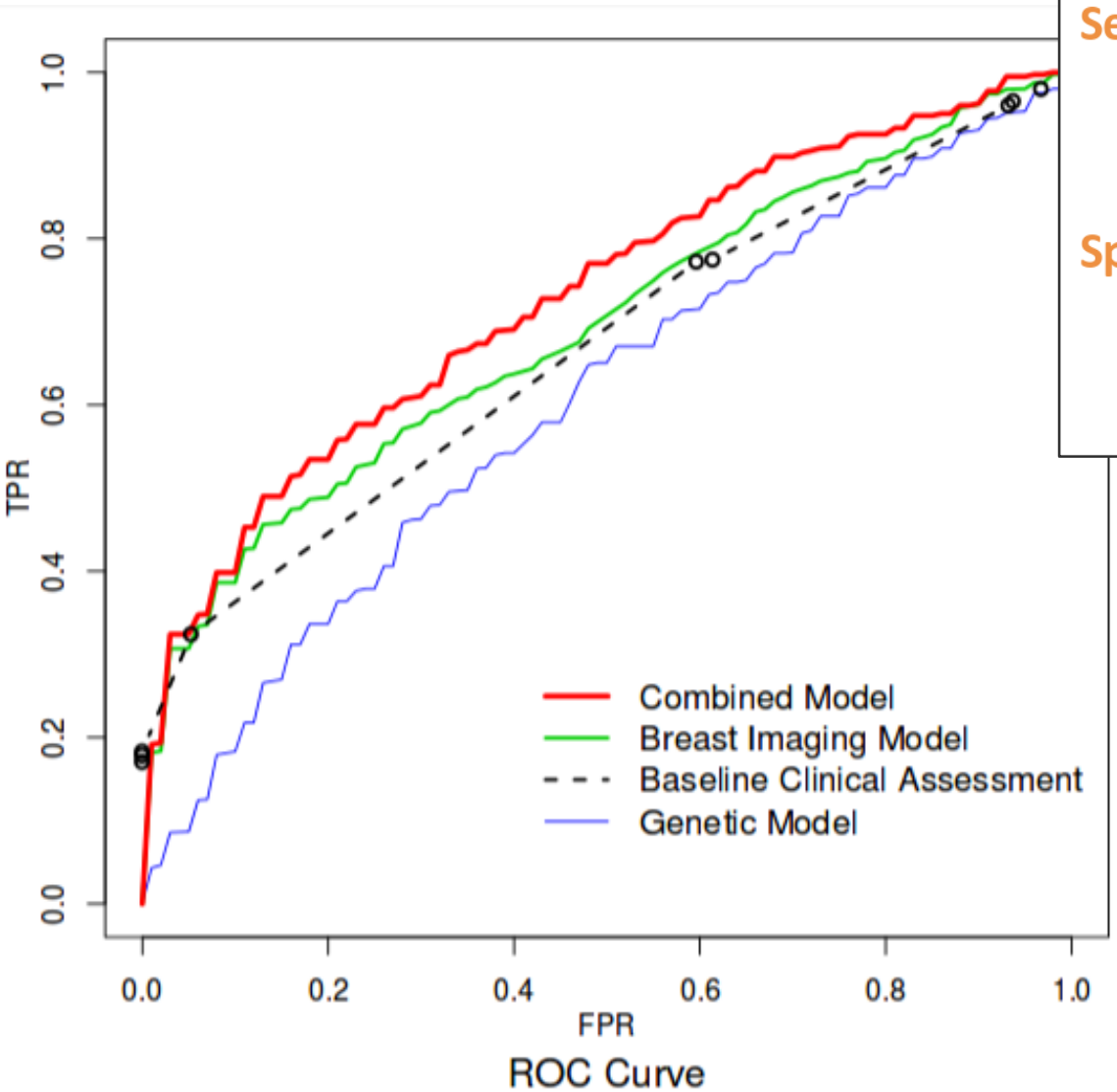
  AUROC = 1.0
- **Random classifier:**

  AUROC = 0.5
- **Our classifier:**

  0.5 < AUROC < 1.0

Predicting breast cancer risk based on mammography images, SNPs, or both.

Liu J, Page D, Nassif H, et al. (2013). **Genetic Variants Improve Breast Cancer Risk Prediction on Mammograms.** *AMIA Annual Symposium Proceedings.* 876-885.



Sensitivity = **Recall** = True positive rate (TPR)

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

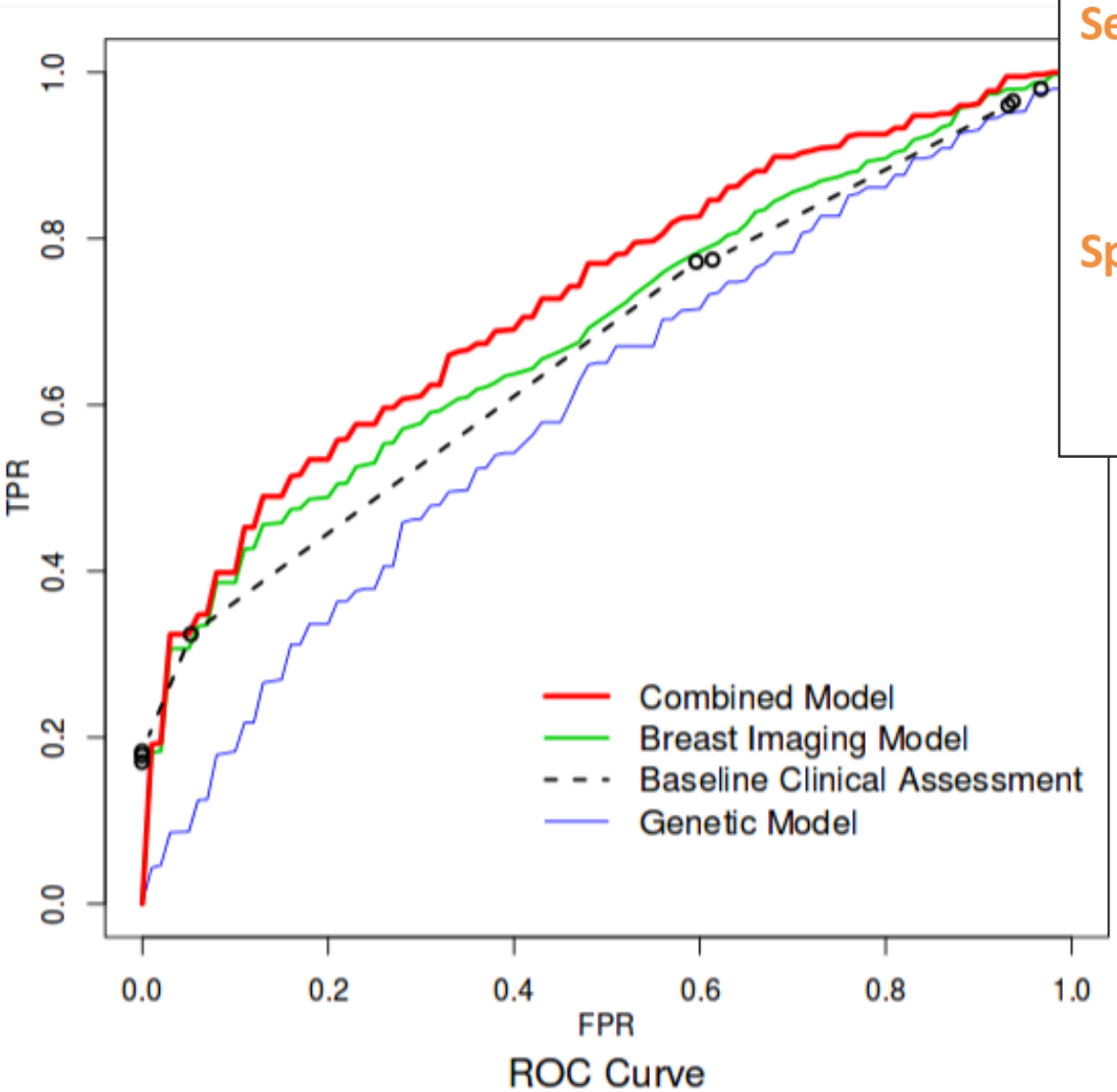
Specificity = True negative rate (TNR)= 1 - FPR

$$\mathrm{TNR} = \frac{\mathrm{TN}}{\mathrm{FP} + \mathrm{TN}}$$

- **Which method outperforms the others?**
- **Is a low FPR or high TPR preferable in a clinical setting?**

Predicting breast cancer risk based on mammography images, SNPs, or both.

Liu J, Page D, Nassif H, et al. (2013). **Genetic Variants Improve Breast Cancer Risk Prediction on Mammograms.** *AMIA Annual Symposium Proceedings.* 876-885.



Sensitivity = **Recall** = True positive rate (TPR)

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

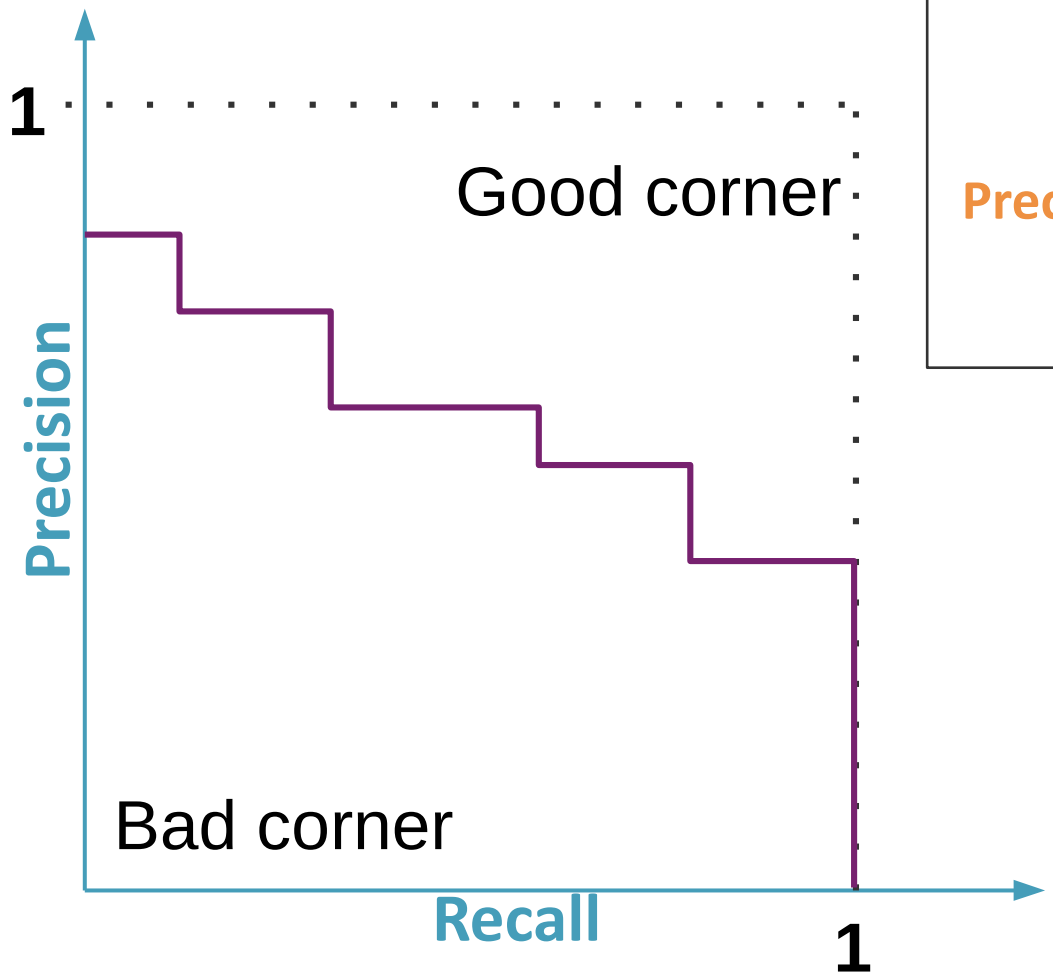
Specificity = True negative rate (TNR)= 1 - FPR

$$\mathrm{TNR} = \frac{\mathrm{TN}}{\mathrm{FP} + \mathrm{TN}}$$

High recall = fewer chances to miss a case

High specificity / low FPR = fewer false alarms

Precision-Recall curves



Sensitivity = **Recall** = True positive rate (TPR)

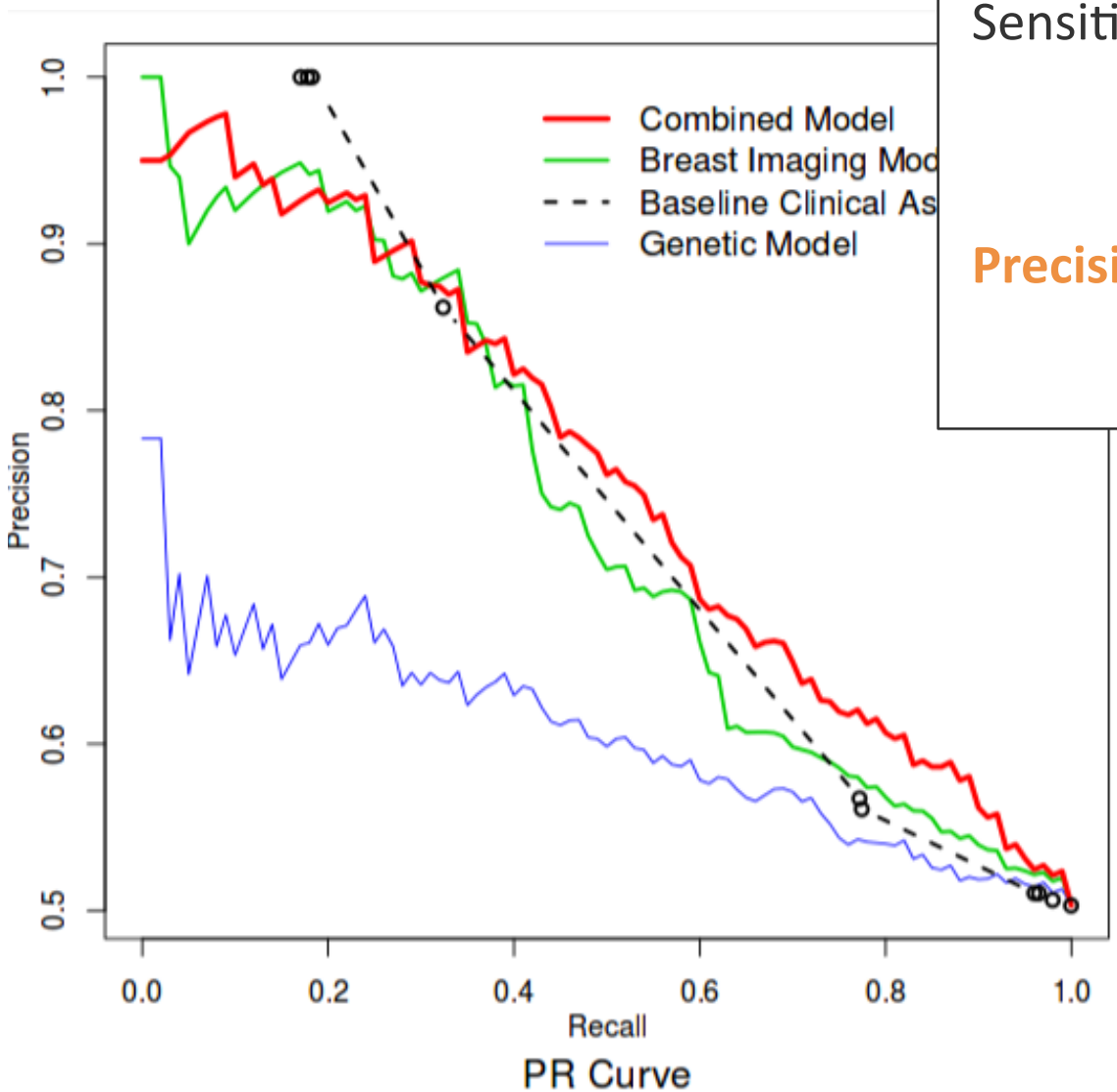
$$\mathrm{TPR} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}}$$

Precision = Positive predictive value (PPV)

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

Predicting breast cancer risk based on mammography images, SNPs, or both.

Liu J, Page D, Nassif H, et al. (2013). **Genetic Variants Improve Breast Cancer Risk Prediction on Mammograms.** *AMIA Annual Symposium Proceedings.* 876-885.



Sensitivity = **Recall** = True positive rate (TPR)

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

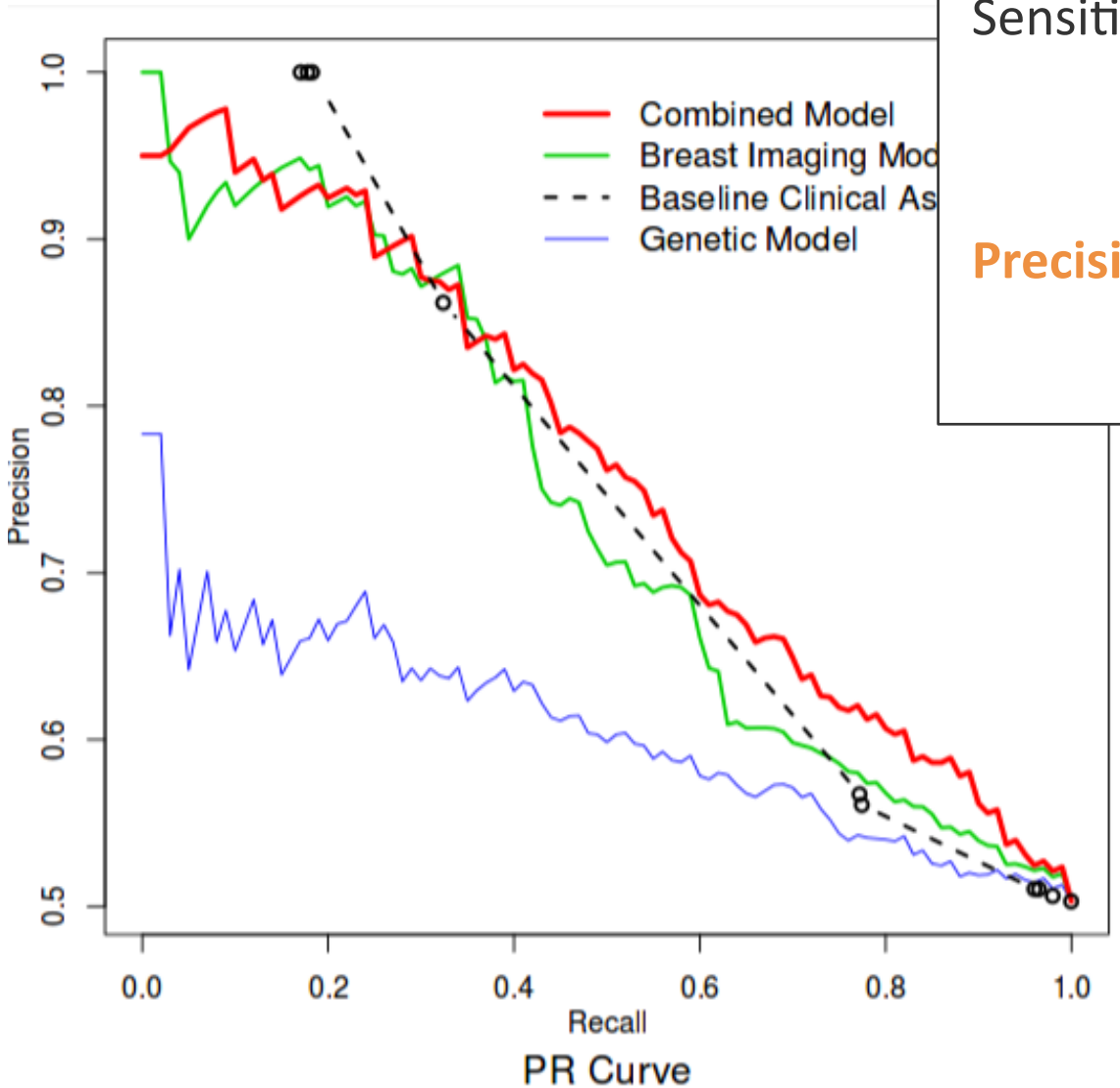
Precision = Positive predictive value (PPV)

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

- **Which method has the highest area under the PR curve?**
- **Is a high recall or high precision preferable in a clinical setting?**

Predicting breast cancer risk based on mammography images, SNPs, or both.

Liu J, Page D, Nassif H, et al. (2013). **Genetic Variants Improve Breast Cancer Risk Prediction on Mammograms.** *AMIA Annual Symposium Proceedings.* 876-885.



Sensitivity = **Recall** = True positive rate (TPR)

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

Precision = Positive predictive value (PPV)

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

High recall = fewer chances to miss a case

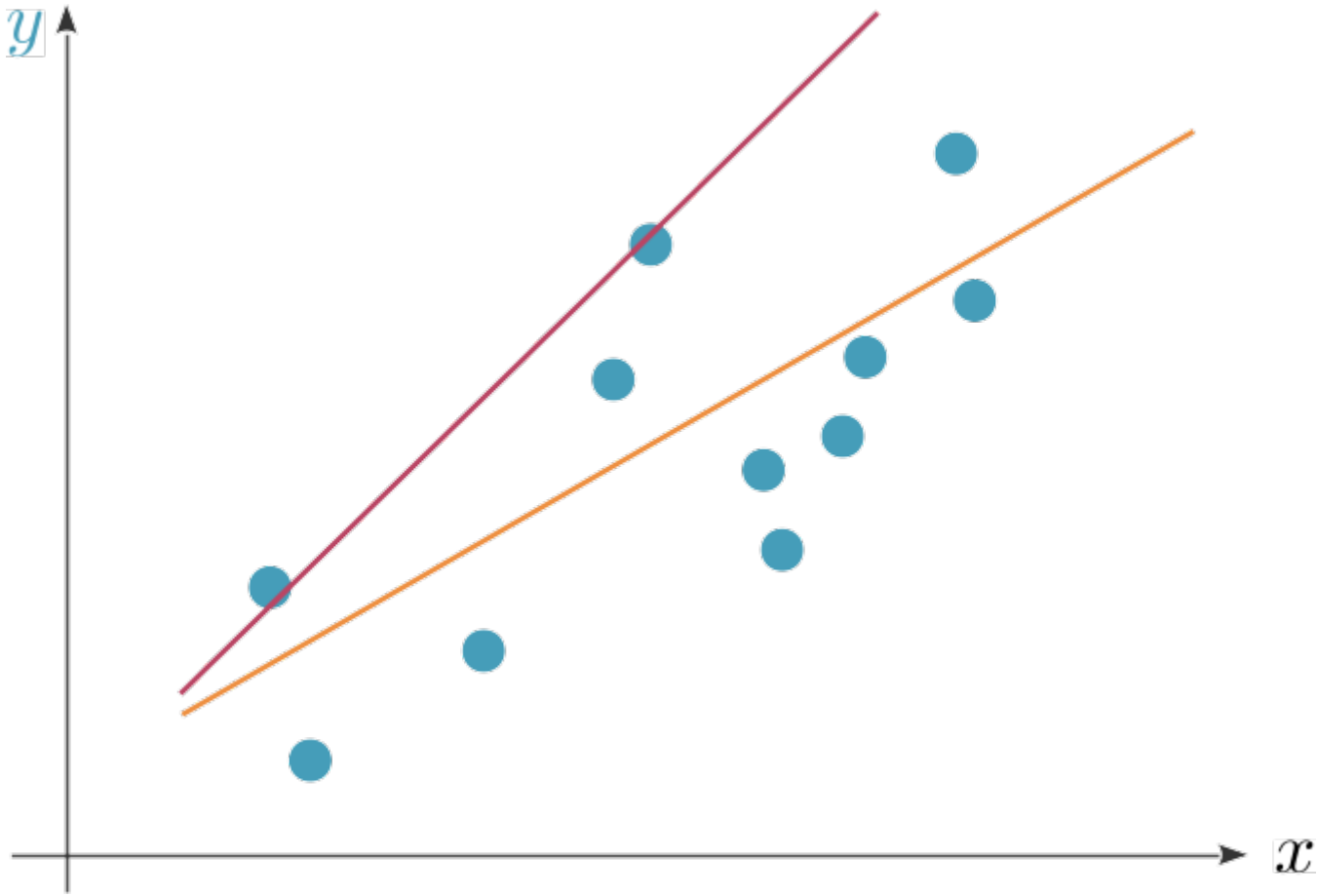
High precision = substantially more true diagnoses than false alarms

Regression model evaluation

- Counting the number of errors is not reasonable

Regression model evaluation

- Counting the number of errors is not reasonable
  - What does error even mean for numerical values?
  - Not all errors are created equal.



Regression model evaluation

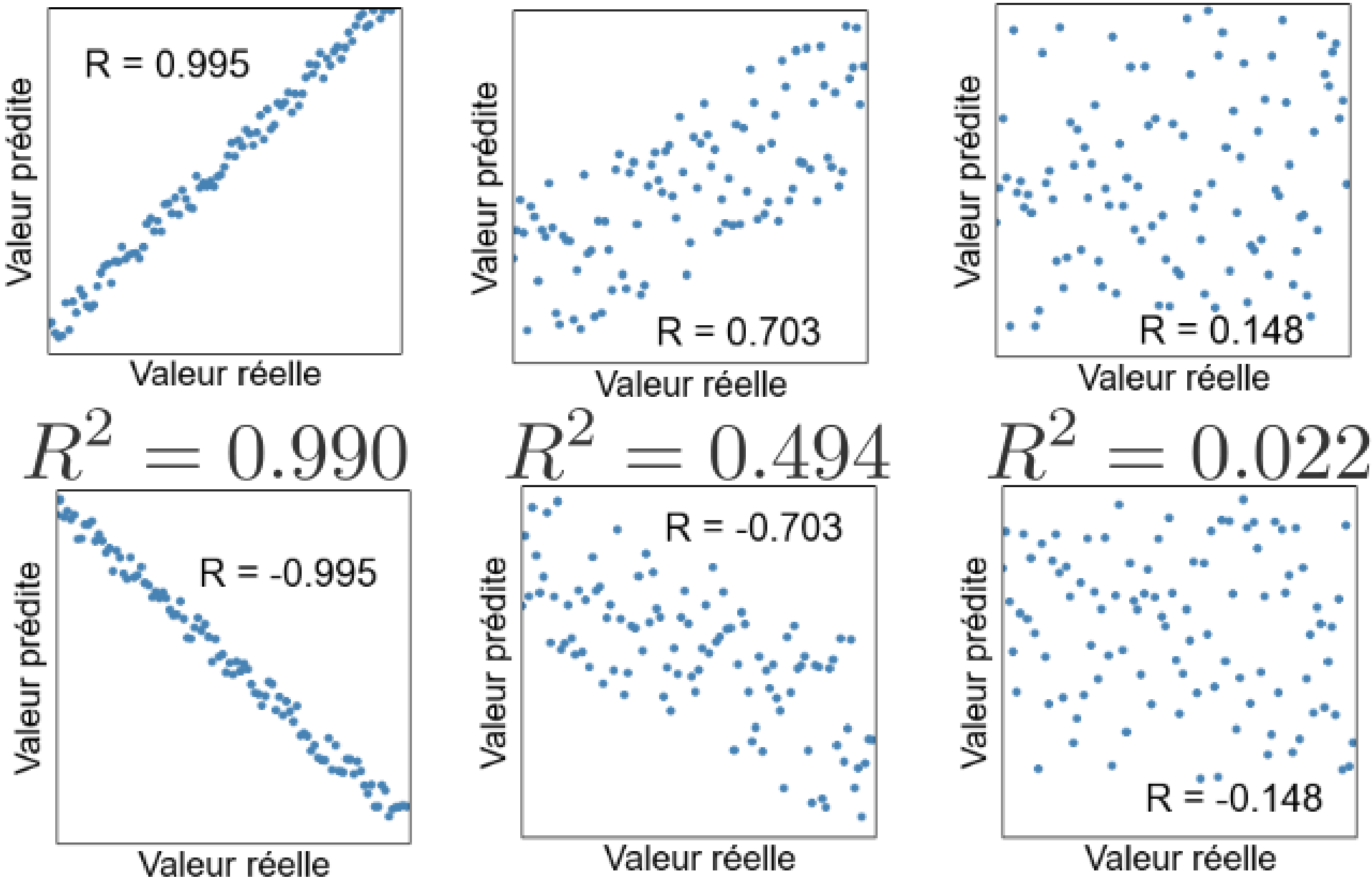
- **Residual sum of squares** $\mathrm{RSS} = \sum_{i=1}^{n} (y^i - f(\boldsymbol{x}^i))^2$
- **Root-mean squared error**

$$\mathrm{RMSE} = \sqrt{\frac{1}{n} \sum_{i=1}^{n} (y^i - f(\boldsymbol{x}^i))^2}$$

- **Relative squared error** $\mathrm{RSE} = \frac{\sum_{i=1}^{n} (y^i - f(\boldsymbol{x}^i))^2}{\sum_{i=1}^{n} (y^i - \bar{y})^2}$
- **Coefficient of determination**

$$R^2 = 1 - \mathrm{RSE} = \frac{\sum_{i=1}^{n} (y^i - \bar{y})(f(\boldsymbol{x}^i) - \overline{f(\boldsymbol{x})})}{\sqrt{\sum_{i=1}^{n} (y^i - \bar{y})^2} \sqrt{\sum_{i=1}^{n} (f(\boldsymbol{x}^i) - \overline{f(\boldsymbol{x})})}}$$

Correlation between true and predicted values



Analytical tools and model complexity

Optimism terms

- Correct the empirical error with an **optimism term**
- Theoretical estimate of the **discrepancy between training and test error**

Augmented error = empirical error + optimism term

- For **linear models**, optimism terms proportional to:
  - **Mallow's Cp:**

$$\frac{d}{n}\hat{\sigma}^2$$

Variance of the residuals on the train set

Squared standard error of the mean of the residuals

parameters
= # non-zero coefficients

  - **Akaike Information Criterion (AIC):** d
  - **Bayesian Information Criterion (BIC):** $d \ln(n)$

Minimum description length (MDL)

- **Shortest code to transmit a random variable z:**

  $-\log_2 P(\boldsymbol{z})$ [Shannon's source coding theorem]

  Consider discrete variable **z**

  – Equiprobable case: use a **fixed-length code**

  $$a \mapsto 00 \qquad b \mapsto 01 \qquad c \mapsto 10 \qquad d \mapsto 11$$

  – Otherwise: use a **variable-length prefix code** in which frequent values get shorter codes

  $$a \mapsto 1 \qquad b \mapsto 10 \qquad c \mapsto 110 \qquad d \mapsto 111$$

The prefix separates codes

Minimum description length (MDL)

- **Shortest code to transmit a random variable z:**

  $-\log_2 P(\boldsymbol{z})$ [Shannon's source coding theorem]

- Assume
  - Parametric model f_θ
  - receiver knows inputs X, model family f.
- To transmit outputs y, need

$$\underbrace{-\log_2 P(\boldsymbol{y}|\theta, f, X)}_{\text{average code length to transmit the difference between model prediction and true outputs.}} \underbrace{-\log_2 P(\theta)}_{\text{average code length to transmit } \theta.}$$

- Choose the model with smallest Kolmogorov complexity (=MDL)

Summary: model selection techniques

- **Empirical:**

  Estimate quality of generalization with

  – **cross-validation**

  – **bootstrap**

- **Theoretical:**

  – Estimate the difference between train error and generalization error with an optimism term

  E.g. Mallow's Cp, Akaike's / Bayesian Information Criteria

  – **Minimum description length (MDL)**

  Choose simplest model (according to Kolmogorov complexity)

References

- *A Course in Machine Learning.*
  `http://ciml.info/dl/v0_99/ciml-v0_99-all.pdf`
  - **Noise:** Chap 2.3
  - **Overfitting:** Chap 2.4
  - **Bias-variance tradeoff:** Chap 5.9
  - **Train and test sets:** Chap 2.5
  - **Cross-validation:** Chap 5.6
  - **Performance measures:** Chap 5.5
- *The Elements of Statistical Learning.*
  `http://web.stanford.edu/~hastie/ElemStatLearn/`
  - **Overfitting:** Chap 7.1
  - **Bias-variance tradeoff:** Chap 2.9, 7.2–7.3
  - **Cross-validation:** Chap 7.10
  - **Bootstrap:** Chap 7.11
  - **Mallow's Cp, AIC, BIC:** Chap 7.7
  - **MDL:** Chap 7.8
- **Entropy encoding:**
  `http://lesswrong.com/lw/o1/entropy_and_short_codes/`

References for prerequisites

- **Linear algebra:**

  http://ocw.mit.edu/courses/mathematics/18-06-linear-algebra-spring-2010/video-lectures/

- **Statistics & probabilities:**
  - **Probability theory: A primer (Jeremy Kun)**
    http://jeremykun.com/2013/01/04/probability-theory-a-primer/
  - **Probability Primer (Jeffrey Miller)**
    https://www.youtube.com/playlist?list=PL17567A1A3F5DB5E4

Practical matters

- Make sure you have turned in **HW01**
- **HW02** is online, due Oct. 9
- **HW03** is online, due Oct. 13

- **Lab**

  `https://github.com/chagaz/ma2823_2017`

Lab 2 – pointers

$$f_2(\mathbf{x}) = 2x_1^2 + 5x_2^2$$
$$= \mathbf{x}^T\mathbf{D}\mathbf{x}$$

where $\mathbf{D} = \begin{bmatrix} 2 & 0 \\ 0 & 5 \end{bmatrix}$ and $\mathbf{x} \in \mathbb{R}^2$.

```
f2 = lambda x : x.T.dot(np.diag([2, 5])).dot(x)
```

Question: Write a function `df2()` for the Jacobian (vector of partial derivatives) of `f2()` below:

```
def df2(x):
    # TODO: return vector of partial derivatives
    return (np.diag([2, 5]) + np.diag([2, 5]).T).dot(x)
    # equivalent to: return (np.diag([4, 10]).dot(x)
```

Question: Write a function `ddf2()` for the Hessian (matrix of second partial derivatives) of `f(2)` below:

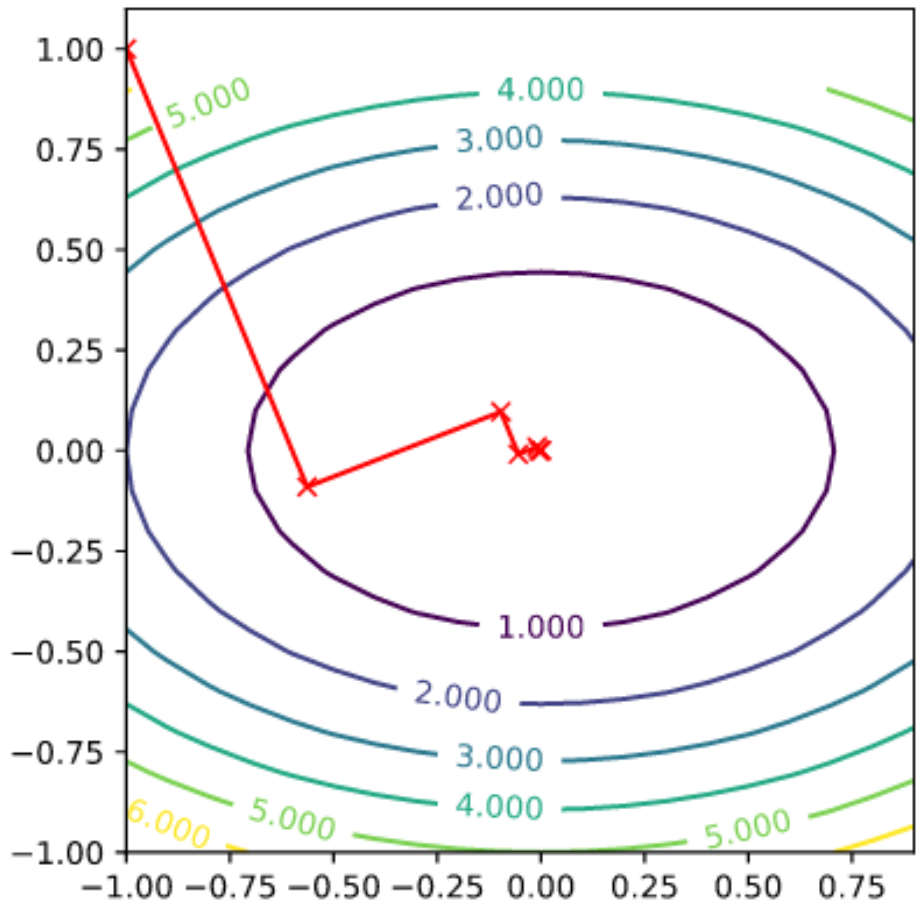
```
def ddf2(x):
    # TODO: return Hessian matrix of second partial derivatives
    return (np.diag([2, 5]) + np.diag([2, 5]).T)
    # equivalent to: return np.diag([4, 10])
```

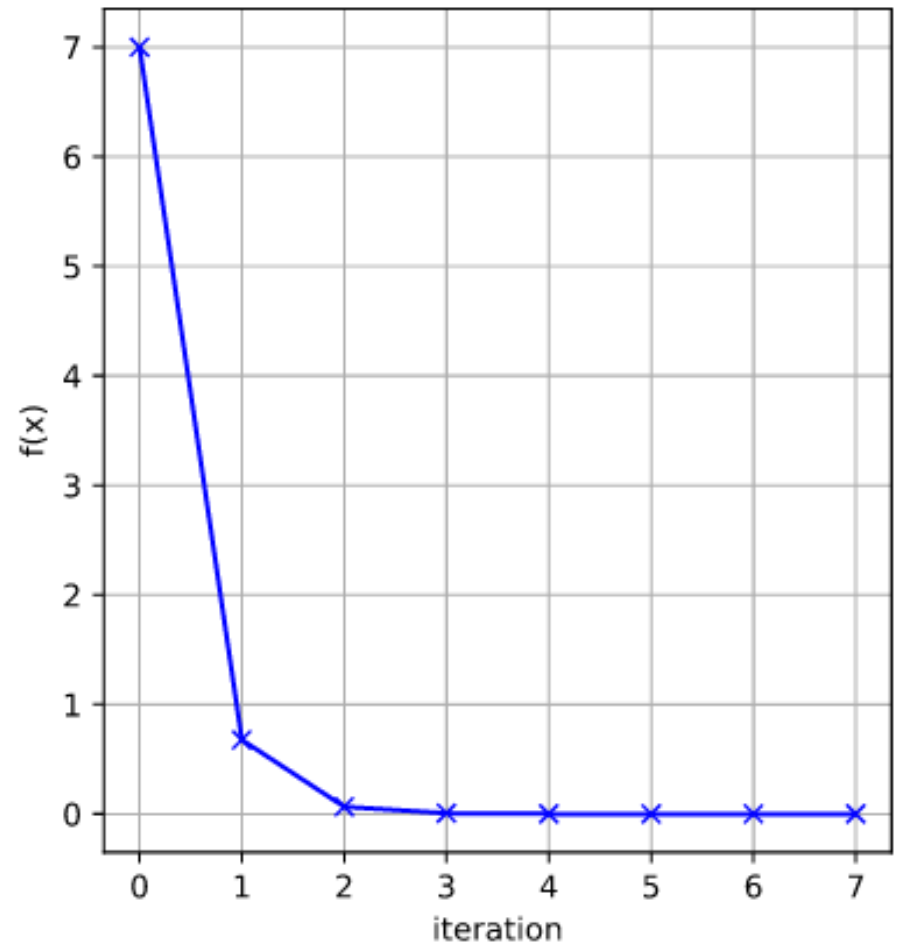
Question: Prove our Hessian matrix, $\mathbf{H} = \partial^2 f / \partial \mathbf{x}^2$, is positive-definite everywhere.

Answer: For any $\mathbf{x} = [x_1, x_2]$, $\mathbf{x}^T\mathbf{H}\mathbf{x} = 4x_1^2 + 10x_2^2 > 0$. **Or:** the eigenvalues of $\mathbf{H}$ are clearly positive

Minimization with Newton's method

```
plot_iterations(f2, np.array(ncg_data))
```





Answer: The descent steps are all at right angles. Minimising $f(\mathbf{x_k} + \alpha \mathbf{d}_k)$ is equivalent to finding α such that $\nabla_\alpha f(\mathbf{x}_k + \alpha \mathbf{d}_k) = \mathbf{d}_k^T \nabla f(\mathbf{x}_k + \alpha \mathbf{d}_k) = 0$. Hence, the next descent direction is orthogonal to the current one (or the gradient is 0).