Data Wrangling and Data Analysis
Machine learning!

Daniel Oberski

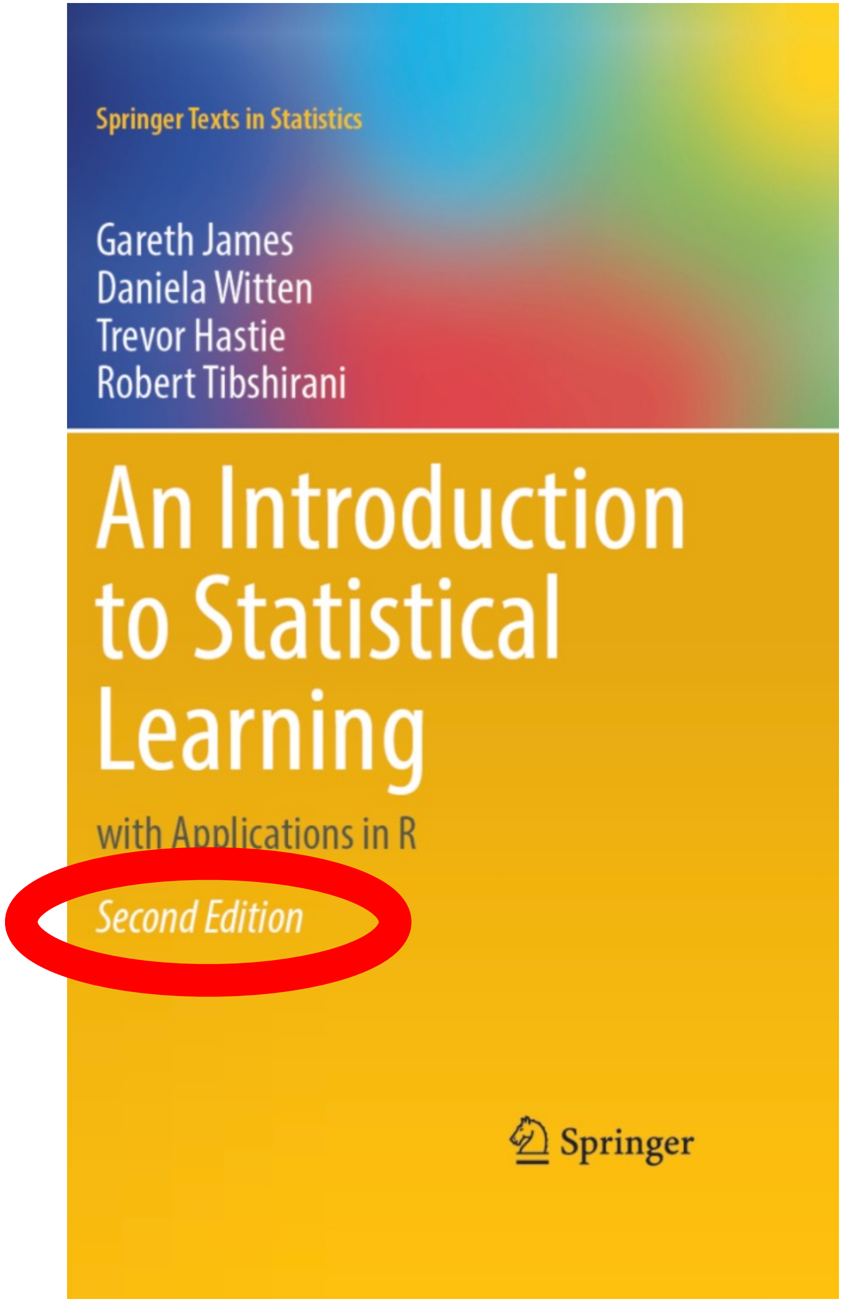
Department of Methodology & Statistics

Utrecht University

VERSION: 2021-10-10

This week:
supervised machine learning

1. **Regression (predicting continuous outcomes)**
2. More regression
3. Classification (predicting discrete outcomes)



Springer Texts in Statistics

Gareth James
Daniela Witten
Trevor Hastie
Robert Tibshirani

An Introduction to Statistical Learning

with Applications in R

Second Edition

Springer

Online free version:
https://www.statlearning.com/

Bird’s eye (high-level) view of machine learning

ML definition (Samuel/Mitchell, 1959)

”A computer program is said to learn from **experience** *E* with respect to some class of **tasks** *T* and **performance measure** *P* if its performance at tasks in *T*, as measured by *P*, improves with experience *E*.”

Some *translation*

”A computer program is said to learn from **experience** *E* with respect to some class of **tasks** *T* and **performance measure** *P* if its performance at tasks in *T*, as measured by *P*, improves with experience *E*.”

	(Loose) translations
Experience	Data; Example; Observation
Task	(Analysis) Purpose; Goal; Aim; Reason; Inference
Performance measure	Error; Loss; Cost; Risk;

Example tasks

1. Identifying customer segments and demographics to help build targeted advertising campaigns
2. Understanding sentiment of Twitter comments as either "positive" or "negative"
3. Identifying financial transactions that are potentially fraudulent
4. Ranking hotels you might like on hotel booking website
5. Recommending a book you might want to buy
6. Detecting brain lesions on an MRI
7. Predicting house prices based on house attributes such as number of bedrooms, location, and size

What could be the experience *and the* performance measure *for each of these?*

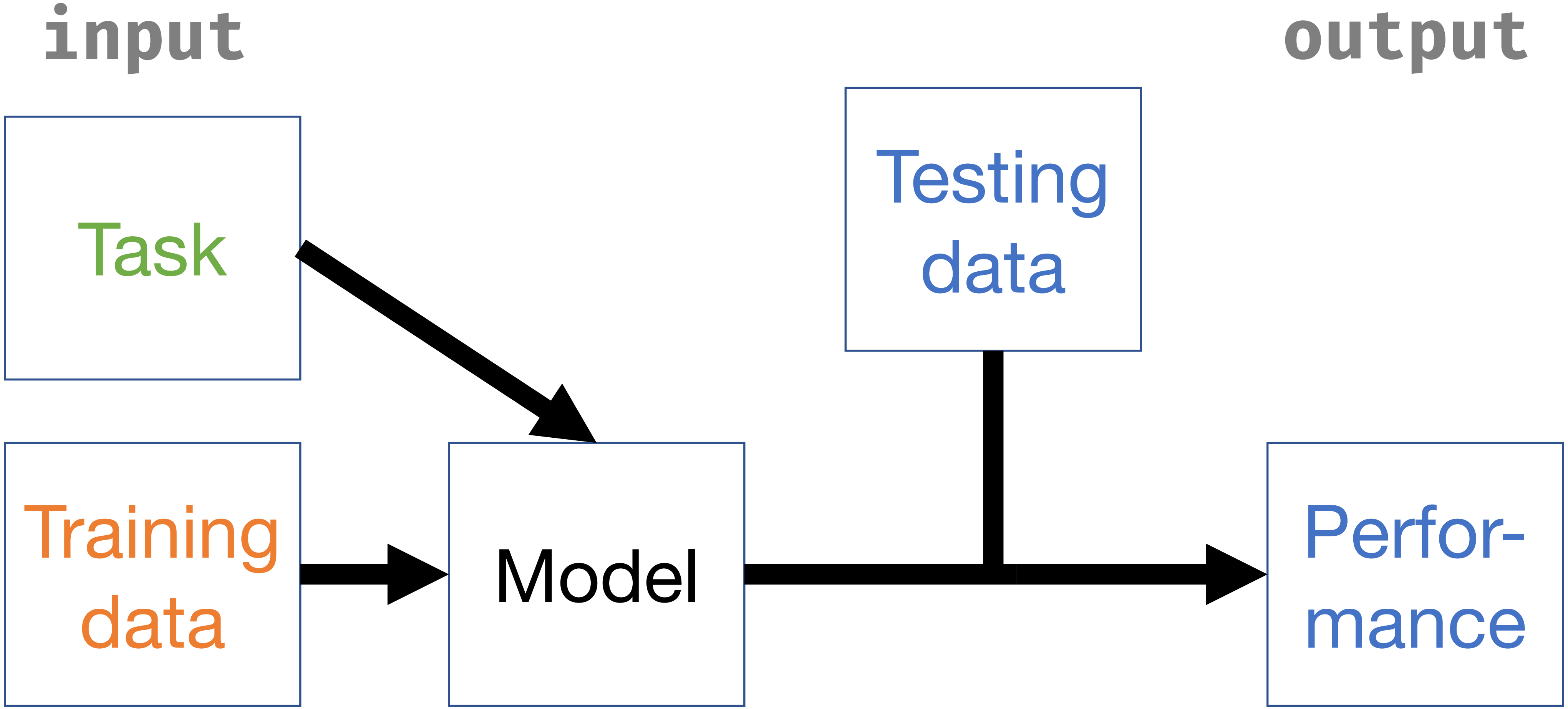
Machine learning definition revisited

What it *says* :

Define your goal (**task**), and let me know when you think I'm doing well at that goal (**performance metric**). Then give me some data (**experience**) and I will figure out how to get you closer to your goal (learn)

What it *does not say* :

You must always do/have <unattainable ideal> when analyzing data. Otherwise your analysis is worthless.



good model. **/**gʊd ˈmɒdl/ *concept.* **1** (*statistics*) a model with perfect input. **2** (*machine learning*) a model with good output, as measured by the performance metric.

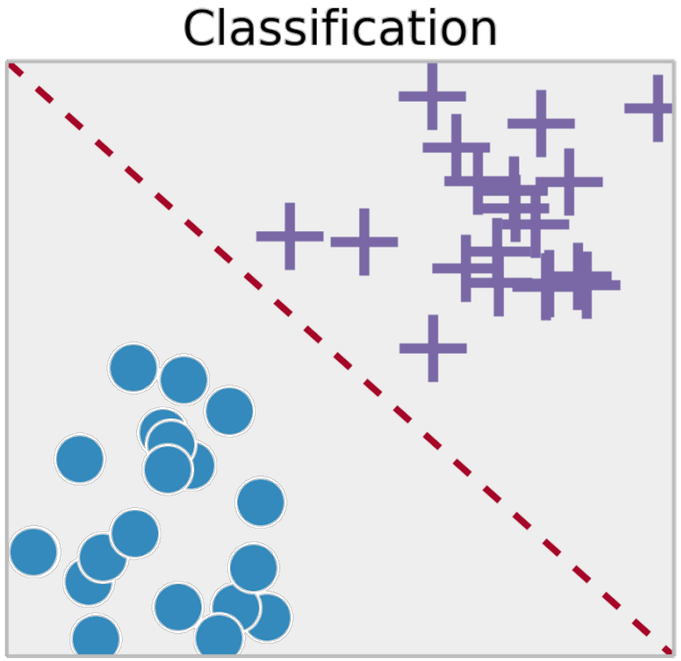
Types of machine learning

We often distinguish **3 types** of machine learning:

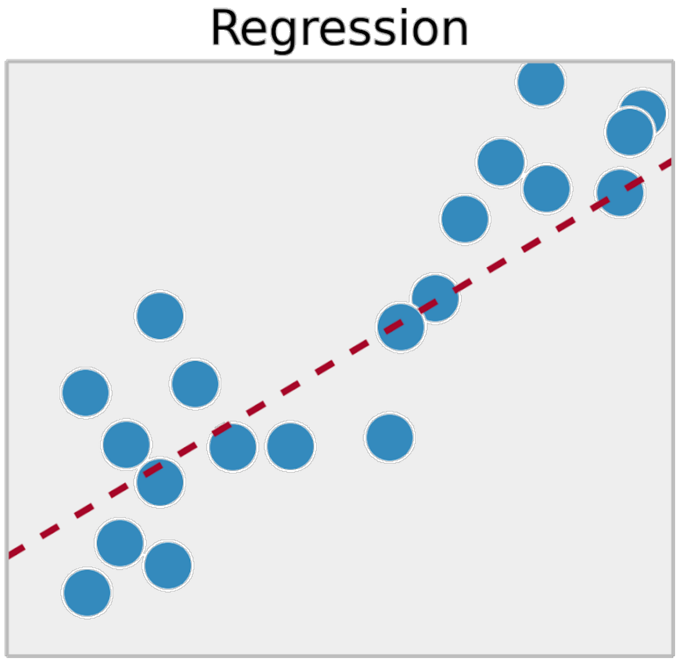
- **Supervised Learning**: learn a model from labeled training data, then make predictions
- **Unsupervised Learning**: explore the structure of the data to extract meaningful information
- **Reinforcement Learning**: develop an agent that improves its performance based on interactions with the environment

Supervised classification vs. regression

Classification: predict a class label (discrete category)

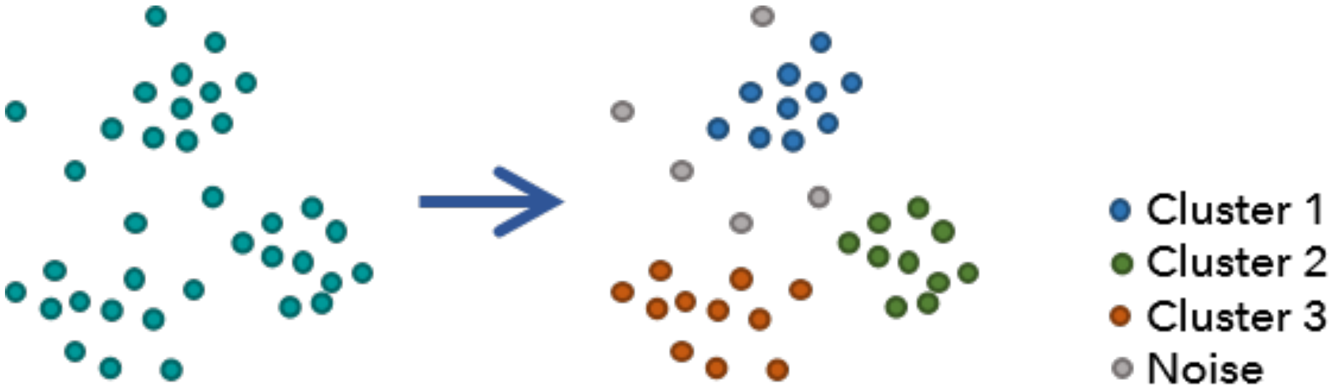


Regression:
predict a continuous value



Unsupervised learning

Learn a model from **unlabeled** training data



Predict “labels” from unlabeled data, or predict new data

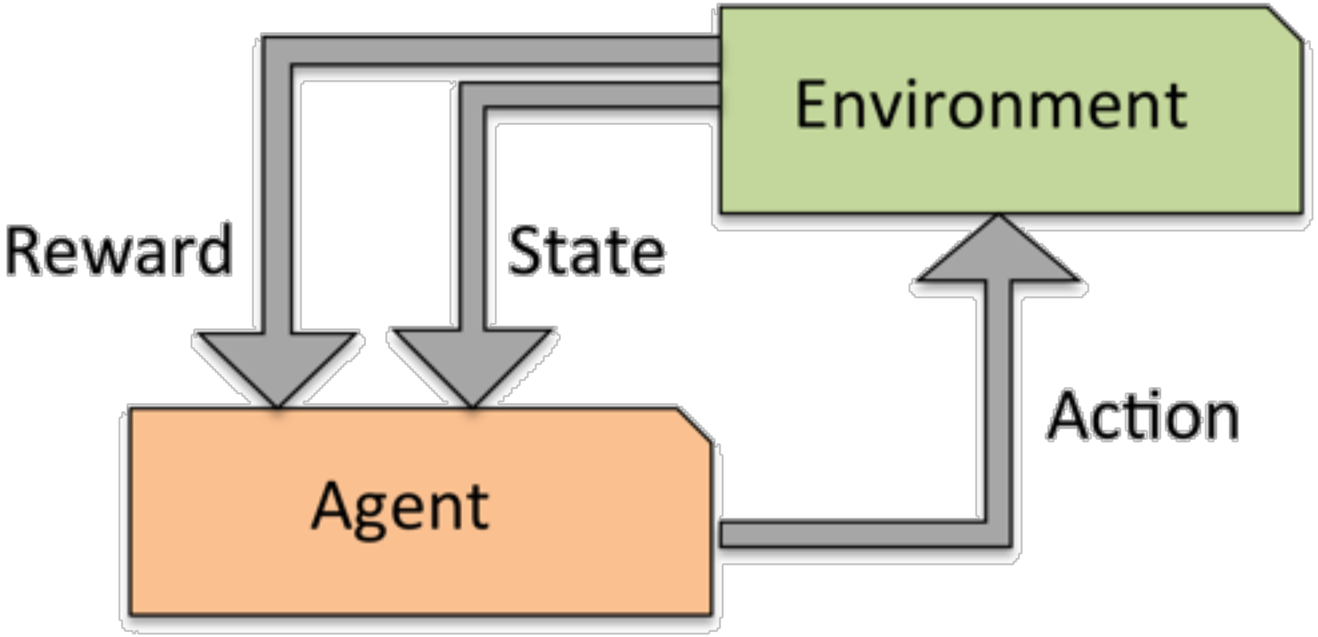
E.g.: PCA, clustering, VAE, GANs, …

Not real people!

Reinforcement learning

(not in this course)

- Develop agent that improves performance based on interactions with environment
- Example: Chess, Go,…
- Reward function defines how well actions work
- Learn actions that maximize reward through exploration



Regression

Regression

$$y = f(\boldsymbol{x}) + \epsilon$$

There are usually a bunch of x’s. We keep notation legible by saying **x** might be a vector of *p* predictors.

Regression

$$y = f(\boldsymbol{x}) + \epsilon$$

y : Observed outcome;

$\boldsymbol{x}$: Observed predictor(s);

$f(\boldsymbol{x})$: *True* prediction function, to be estimated (so **unknown**);

ϵ : Unobserved residuals, just *defined* as the “irreducible error”,

$$\epsilon = y - f(\boldsymbol{x})$$

The higher the variance of the irreducible error, var(ϵ) = σ, the less we can hope to explain with the data ($\boldsymbol{x}$) at hand.

Different goals of regression

Prediction:

- Given $\boldsymbol{x}$ and y, work out $f(\boldsymbol{x})$ as closely as possible.

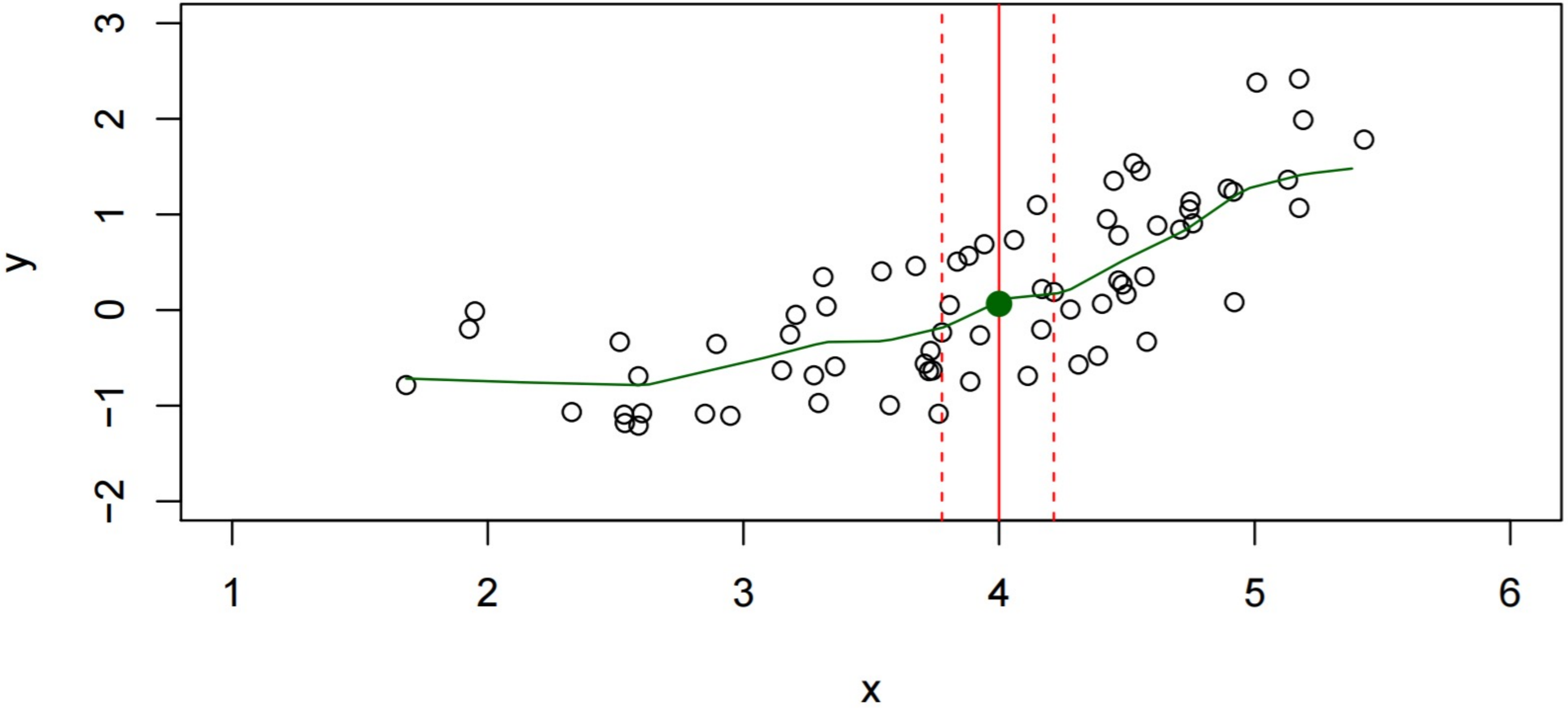
Inference:

- "Is $\boldsymbol{x}$ related to y?"
- "How is $\boldsymbol{x}$ related to y?"
- "How precise are parameters of $f(\boldsymbol{x})$ estimated from the data?"

Galit Schmueli (2010). *To Explain or to Predict?*. Statistical Science.

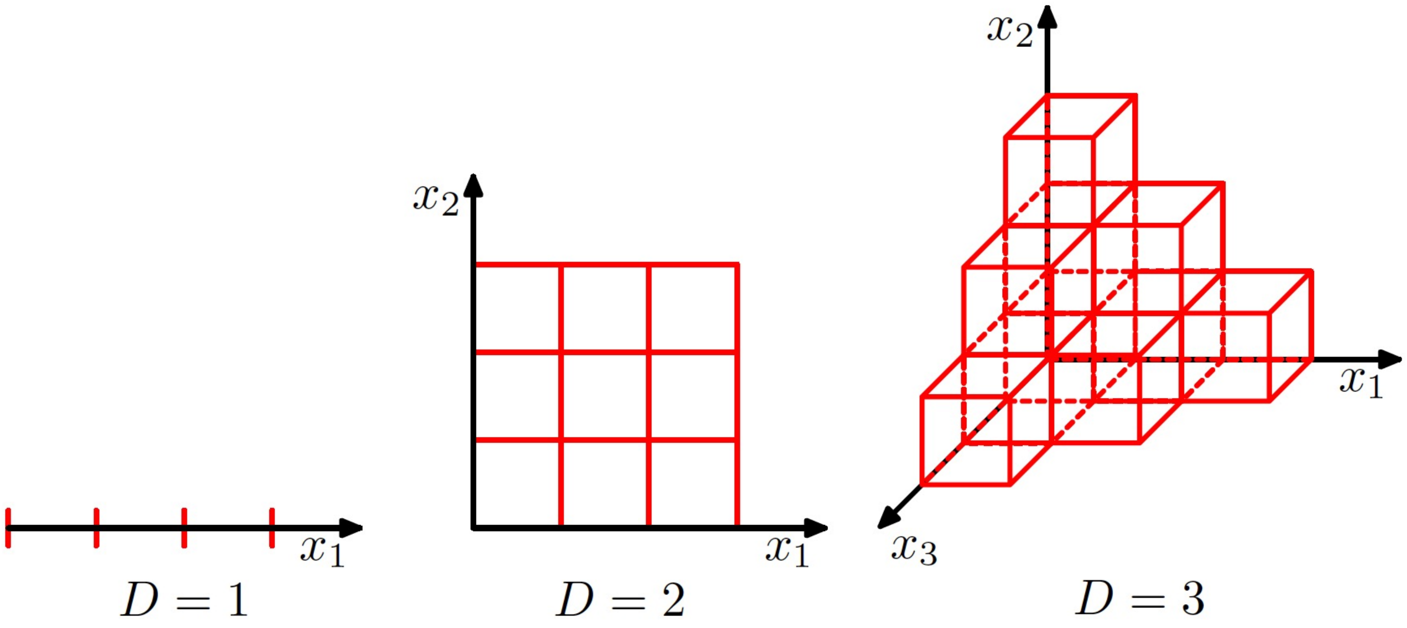
Estimating $f(x)$ with k-nearest neighbors

- Typically we have no data points with $x = 4$ exactly.
- Take “neighborhood” of points around 4 and predict average:

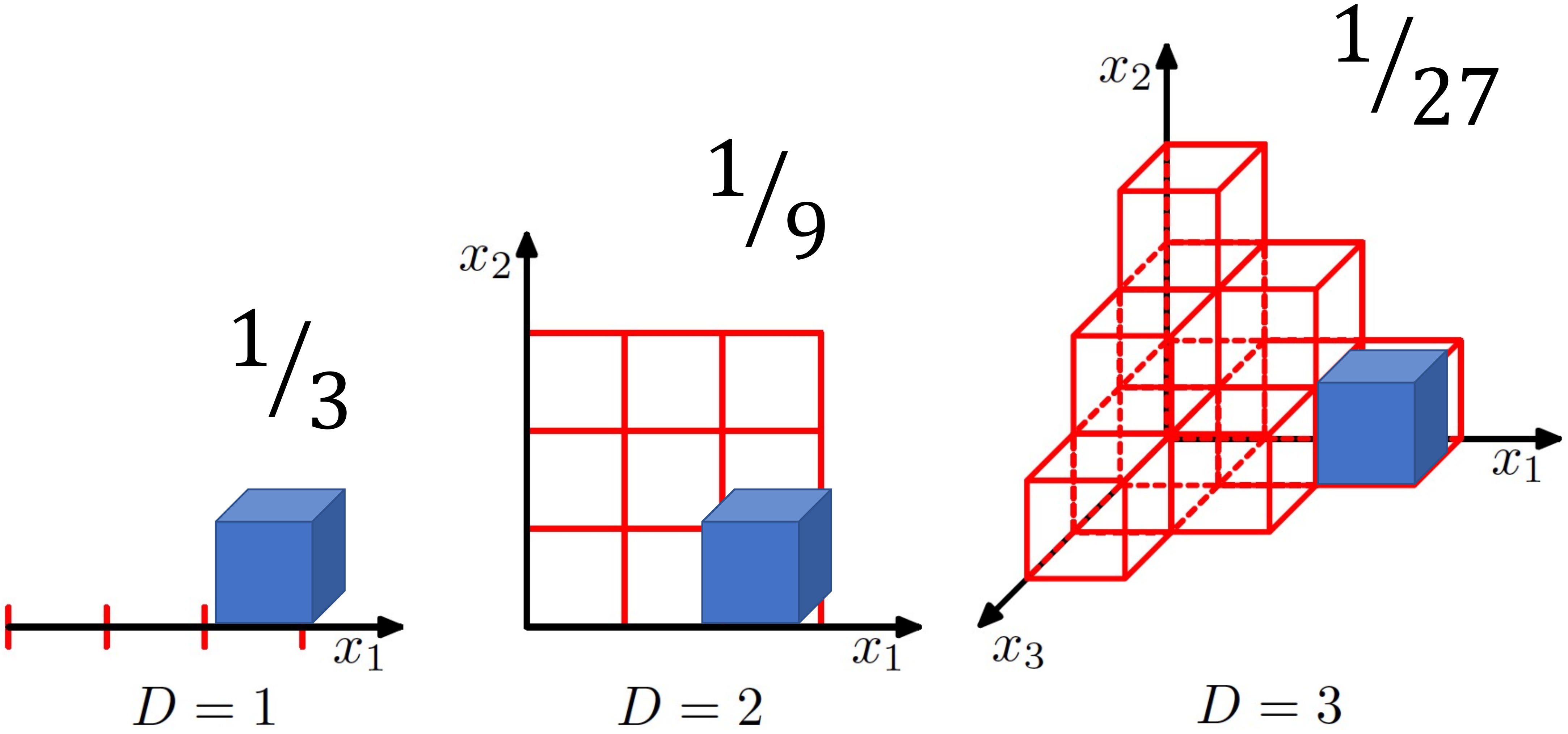


Why not kNN: "Curse of dimensionality"

- kNN is intuitive and can work well with few predictors;
- With "many" (say, 5 or more) predictors, kNN breaks down:
- **Closest** points on many predictors simultaneously are actually **far away!**



Source: Bishop (2006). *Pattern recognition and machine learning.*



$D = 1$

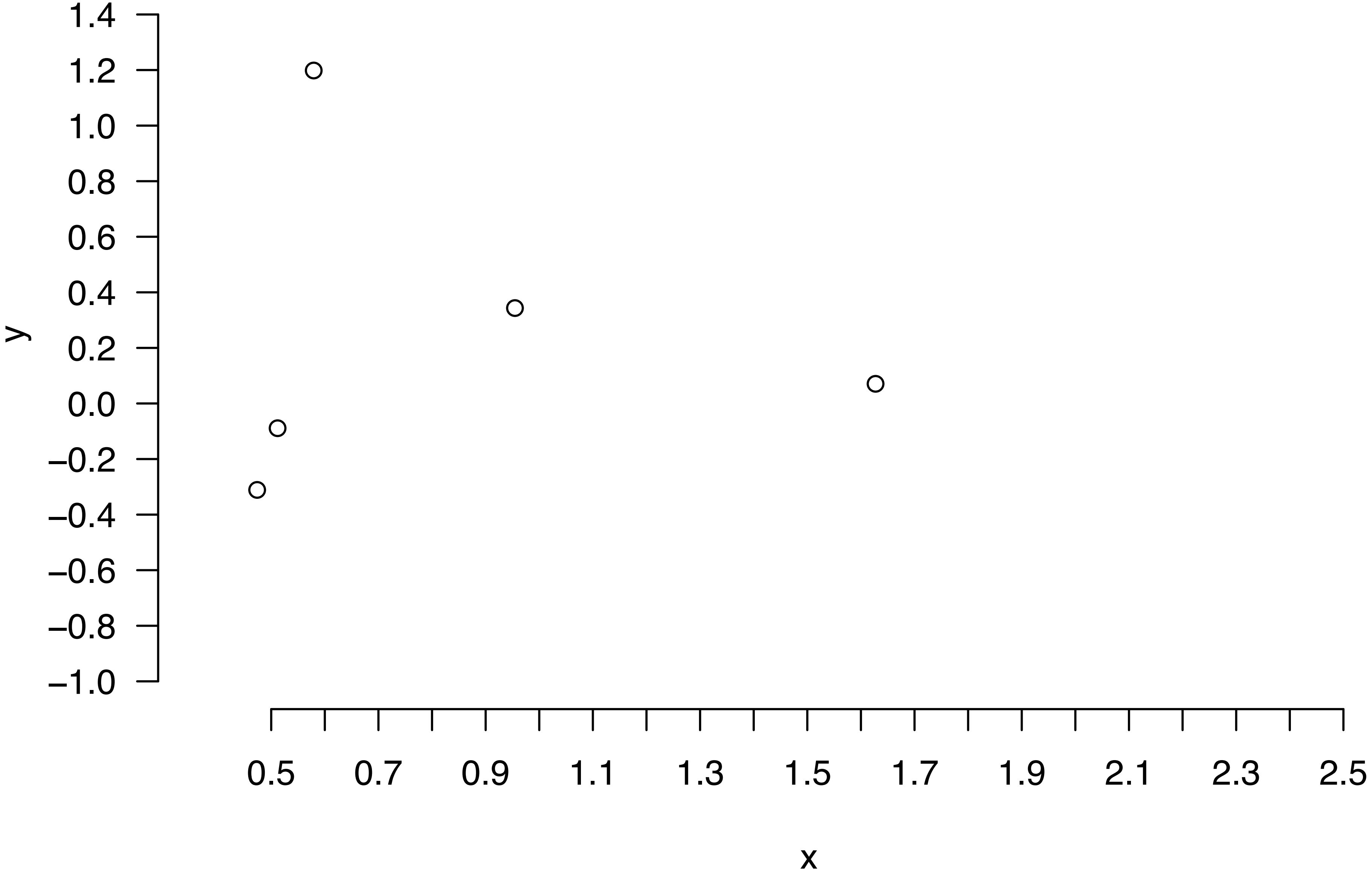
$D = 2$

$D = 3$

High-dimensional space is **lonely...**

An exercise in prediction

- I am going to show you a data set, and we (i.e. you) are going to try to estimate $\hat{f}(x)$ and predict y;
- I generated this data set myself using R, so I know the true $f(x)$ and distribution of ϵ.



Task: predict (however you want): $\hat{y}$ for $x = 0.6, 1.6, 2.0$.

Linear regression reminder

Linear regression says that the i-th outcome y_i is

$$y_i = b_0 + b_1 x_i + \epsilon_i.$$

In other words, $f(x_i) = b_0 + b_1 x_i$.

Linear regression can have quadratic (and other) terms, e.g.:

$$y_i = b_0 + b_1 x_i + b_{12} x_i^2 + \epsilon_i.$$

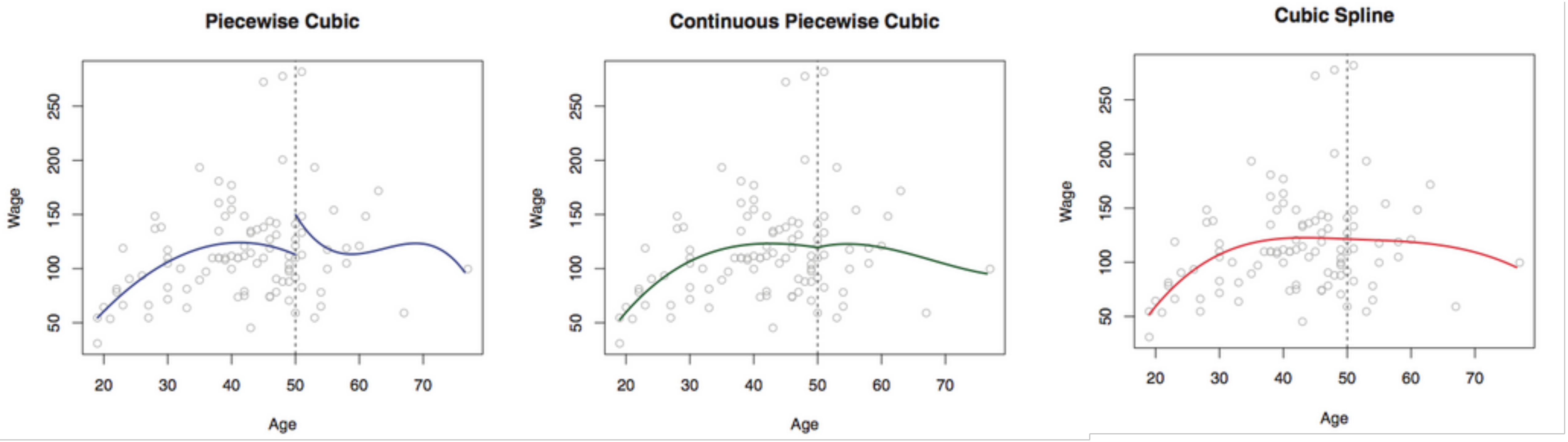
In other words, $f(x_i) = b_0 + b_1 x_i + b_{12} x_i^2$.

“Learning algorithm”: https://www.geogebra.org/m/UxJQorBl

Local ("nonparametric") regression

Nonparametric (e.g. loess, splines):

Idea is to combine "local" linear regressions into one prediction



Model performance

Model performance

- The predictions $\hat{y} = \hat{f}(x)$ differ from the true $y = f\ (x)$;
- We can evaluate how much this happens “on average”.

A few model evaluation metrics

- Mean squared error (MSE):

$$\mathrm{MSE} = n^{-1}\sum_{i=1}^{n}(y-\hat{y})^2$$

- Root mean squared error $\mathrm{RMSE} = \sqrt{\mathrm{MSE}}$
- Mean absolute error (MAE:)

$$\mathrm{MAE} = n^{-1}\sum_{i=1}^{n}|y-\hat{y}|$$

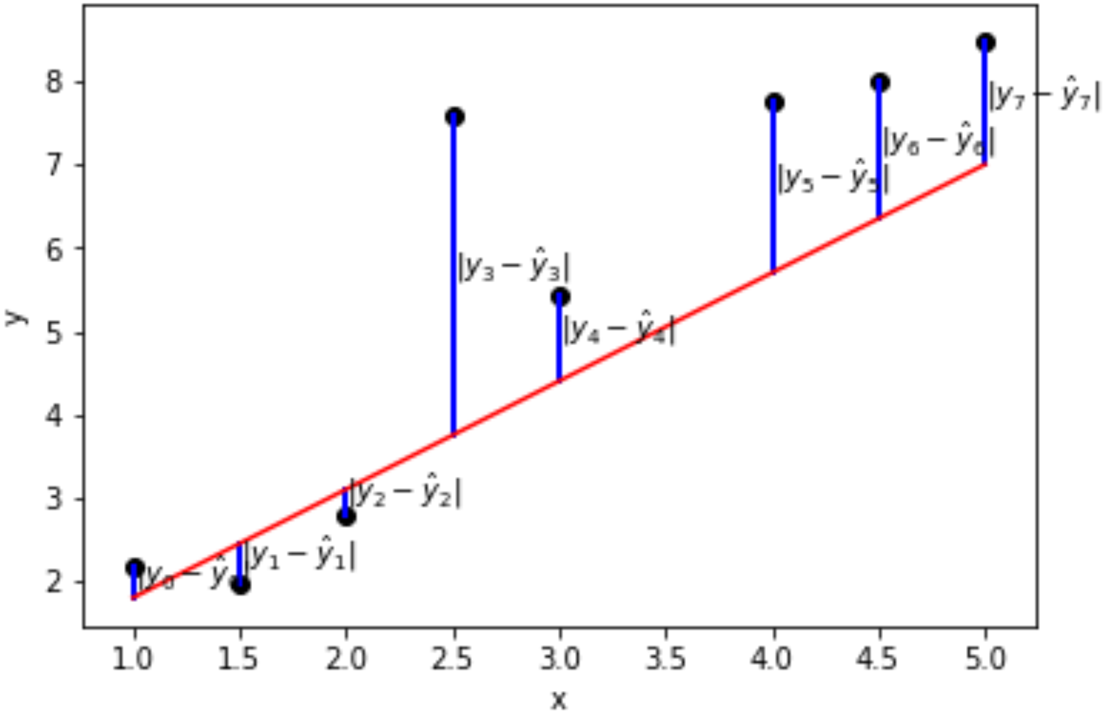
- Median absolute error (mAE):

$$\mathrm{mAE} = \mathrm{median}|y-\hat{y}|$$

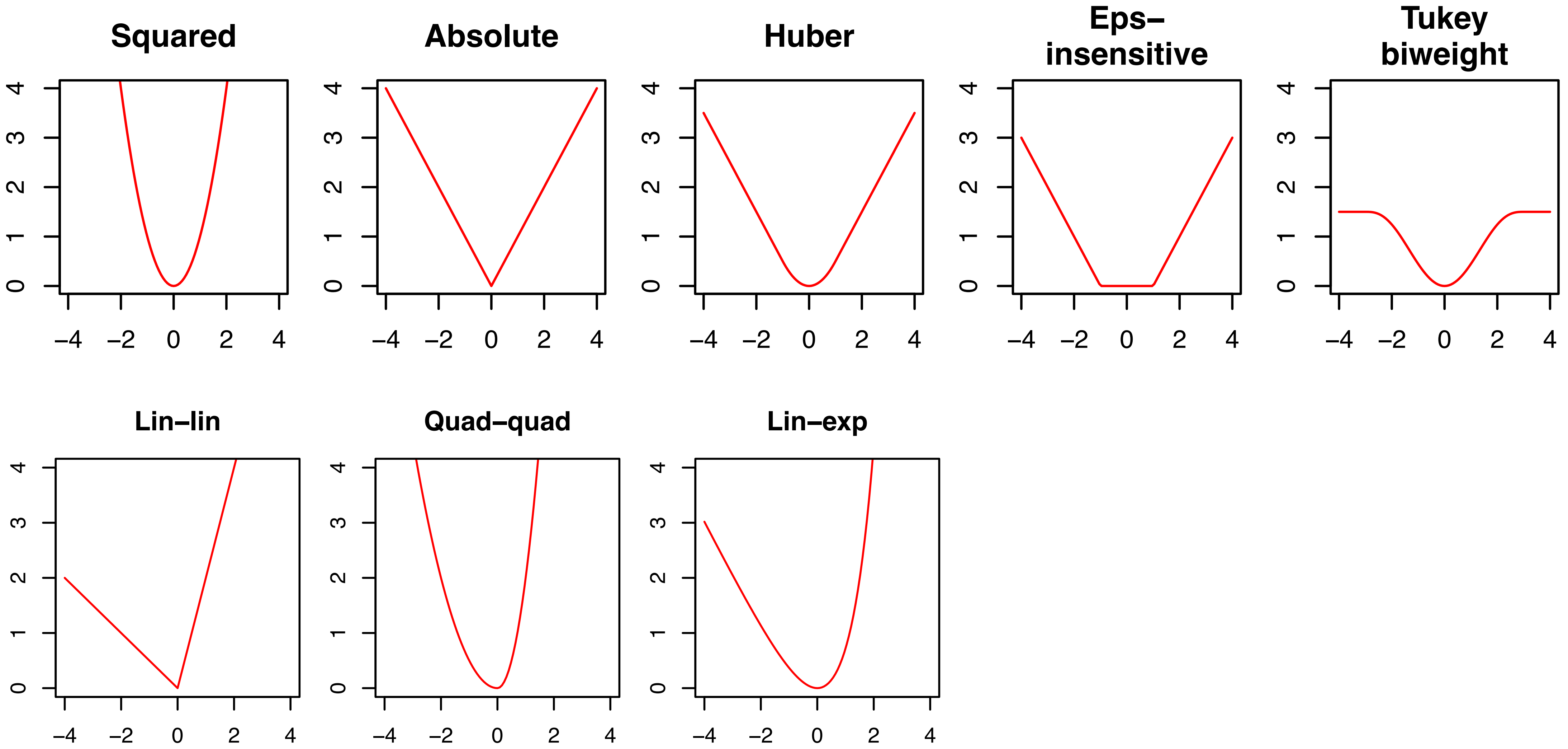
- Proportion of variance explained: (R^2)

$$R^2 = \mathrm{correlation}(y,\hat{y})^2$$

- etc...



Thinking about loss functions



So, who won…?

- Which model appears to fit best to the training data?
- Calculate MSE for one model, relative to truth.
- Which is the best model in terms of MSE?

Thought experiment

- Imagine we had sampled another 5 observations, re-trained all of our models, and predicted again.
- Each time we remember the predictions given.
- We do this a large number of (say, 1,000,000,000) times, and then take the average for the predictions over all samples

Questions

1. Which model(s) would give the right prediction **on average**?
2. Which model(s) would give wildly varying predictions?
3. Which model(s) would you *guess* have the lowest MSE overall?

Unbiased

Model that gives the correct prediction, on average over samples from the target population

- Unbiased in our example: nonparametric, square-root
- Biased in our example: all others

High variance:

Model that easily overfits accidental patterns.

- High variance in our example: nonparam., quadratic, sq-root
- Low variance in our example: linear regression

Bias-variance tradeoff

- Flexibility → less bias
- Flexibility → more variance

Bias and variance are implicitly linked because they are both affected by **model complexity** (“flexibility”, “capacity”)

What do you mean by “complexity”?

- Amount of information in data absorbed into model;
- Amount of compression performed on data by model;
- Number of effective parameters, relative to effective degrees of freedom in data.

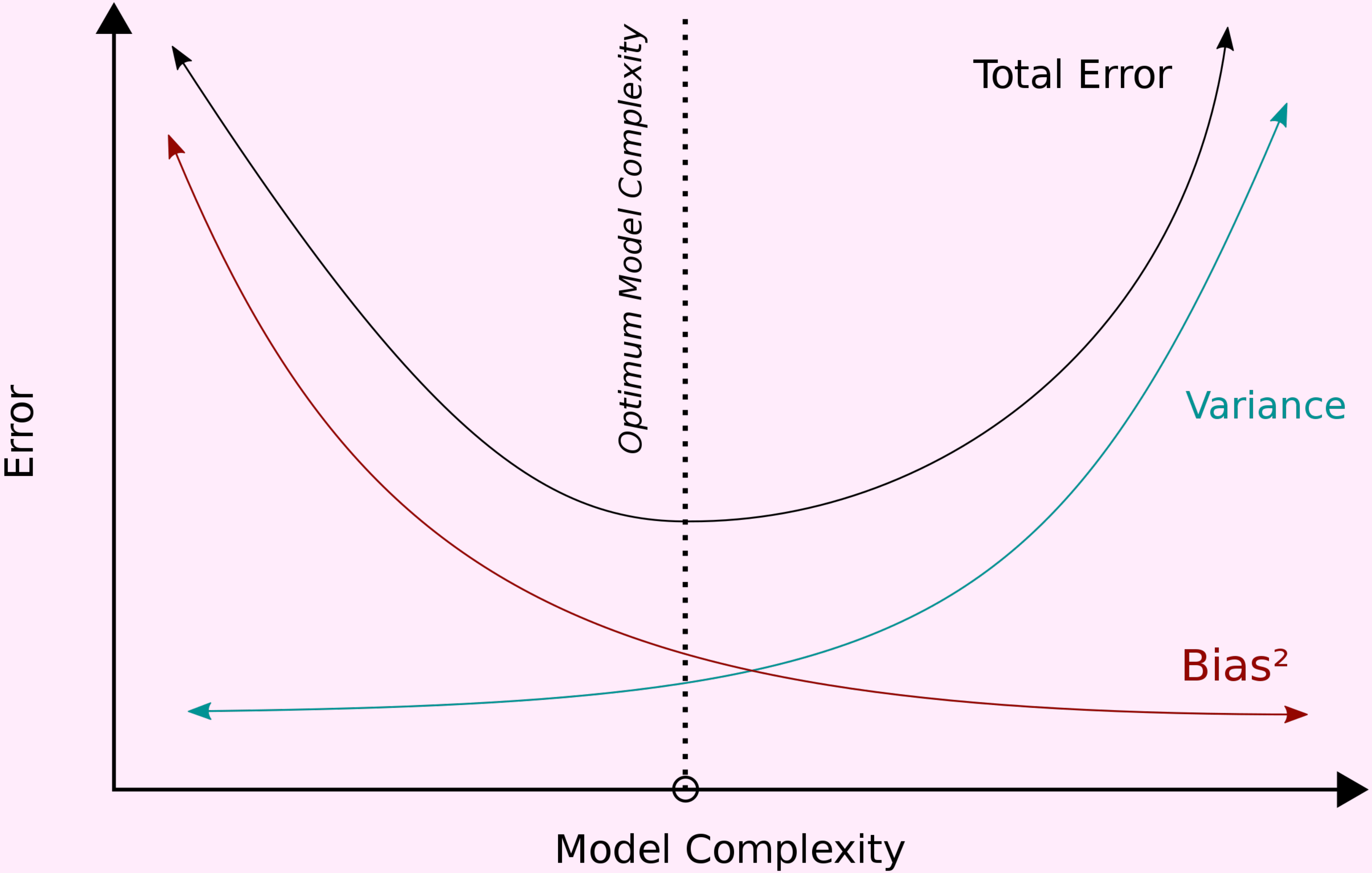
Examples of things that make model **more “complex”**:

- More predictors in linear regression
- Higher-order polynomial in linear regression (x^2, x^3, x^4, etc.);
- Smaller “neighborhood” in kNN
- ...

Question:

Does the bias-variance tradeoff occur with $n = 5$?

Does the bias-variance tradeoff occur with $n = 5{,}000{,}000{,}000$?


Total Error
Variance
Bias²
Optimum Model Complexity
Error
Model Complexity

$$E(\mathrm{MSE}) = \mathrm{Bias}^2 + \mathrm{Variance} + \epsilon$$

Population mean squared error is squared bias PLUS model variance PLUS irreducible variance.

(The E means “on average over samples from the target population”).

Back to reality!

Problem:

- Wait, we don’t actually have the population!
- And our training data were already used to train the model…

Solution:

- Instead, we will take a new, pristine, sample from population:
- The **test data**

Will my model succeed?

These factors should determine your success:

1. How doable the problem is in the first place: irreducible error;
2. How complex the model $\hat{f}(\boldsymbol{x})$ is;
3. How complex the true function $f(\boldsymbol{x})$ is;
4. The sample size.

All tricks of the trade attack one or more of these!

Problem	**Some ideas for plan of attack**	**Example**
Irreducible error	Get more features; Reduce measurement error	LIDAR on car; Multiple rating radiologists
Model complexity	Try models with range of complexity; Include prior knowledge in the model	Download pretrained model and use that as starting point
Task complexity	Choose something easier; Influence the process	Paint road signs for self-driving
Sample size	Get more examples	Why not use all of Wikipedia for NLP?

The train-val-test paradigm

Question: What observations are we supposed to take the "average" over when calculating metrics for $\hat{f}(\boldsymbol{x})$?

A. Observations used to fit $\hat{f}(\boldsymbol{x})$.

B. New observations from the same source.

C. New observations from the intended prediction situation.

D. Other.

Train/dev/test

Training data:

Observations used to train (“fit”, ”estimate”) $\hat{f}(\boldsymbol{x})$

Validation data (or “dev” data):

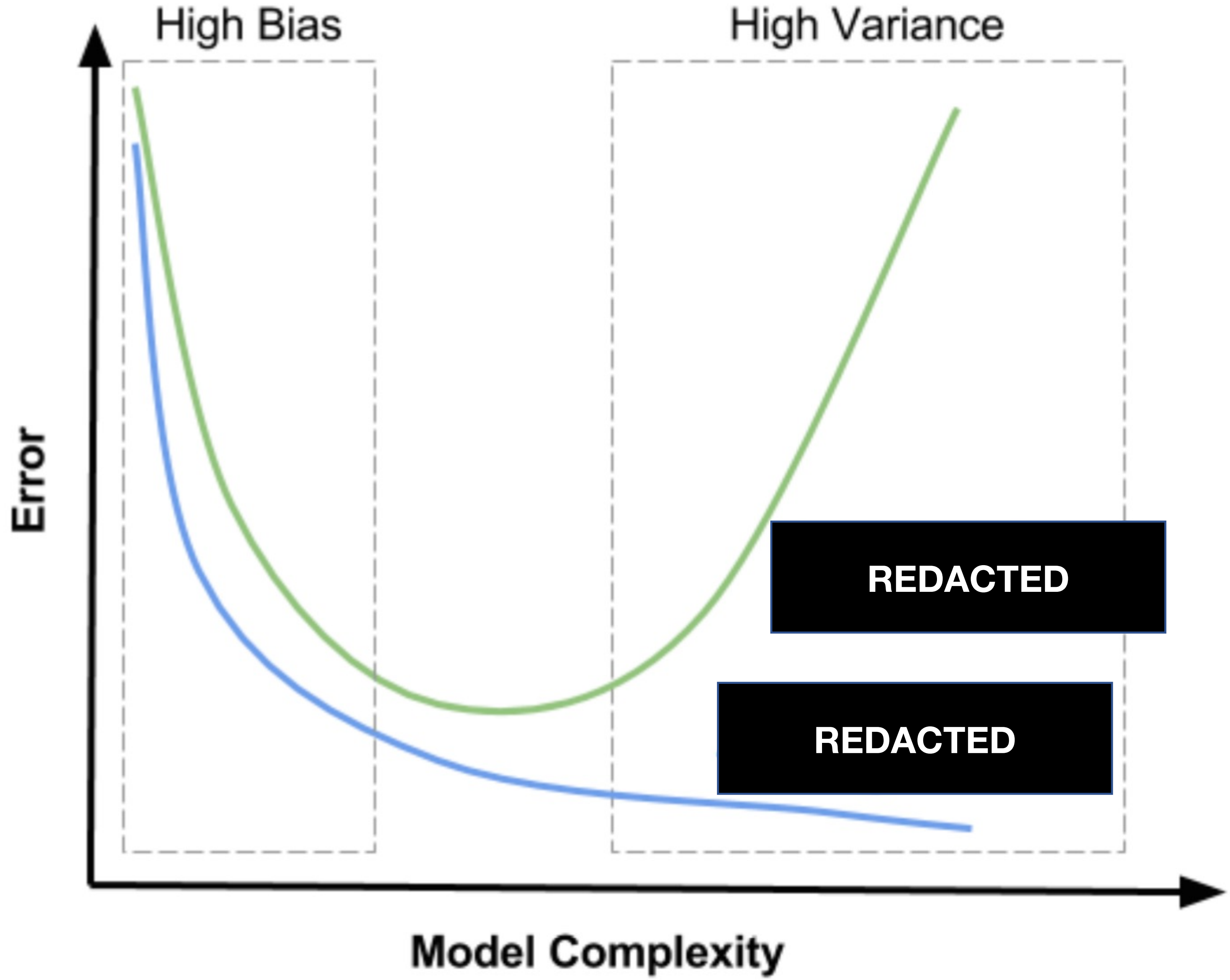
New observations from the same source as training data

Used several times to select model complexity)

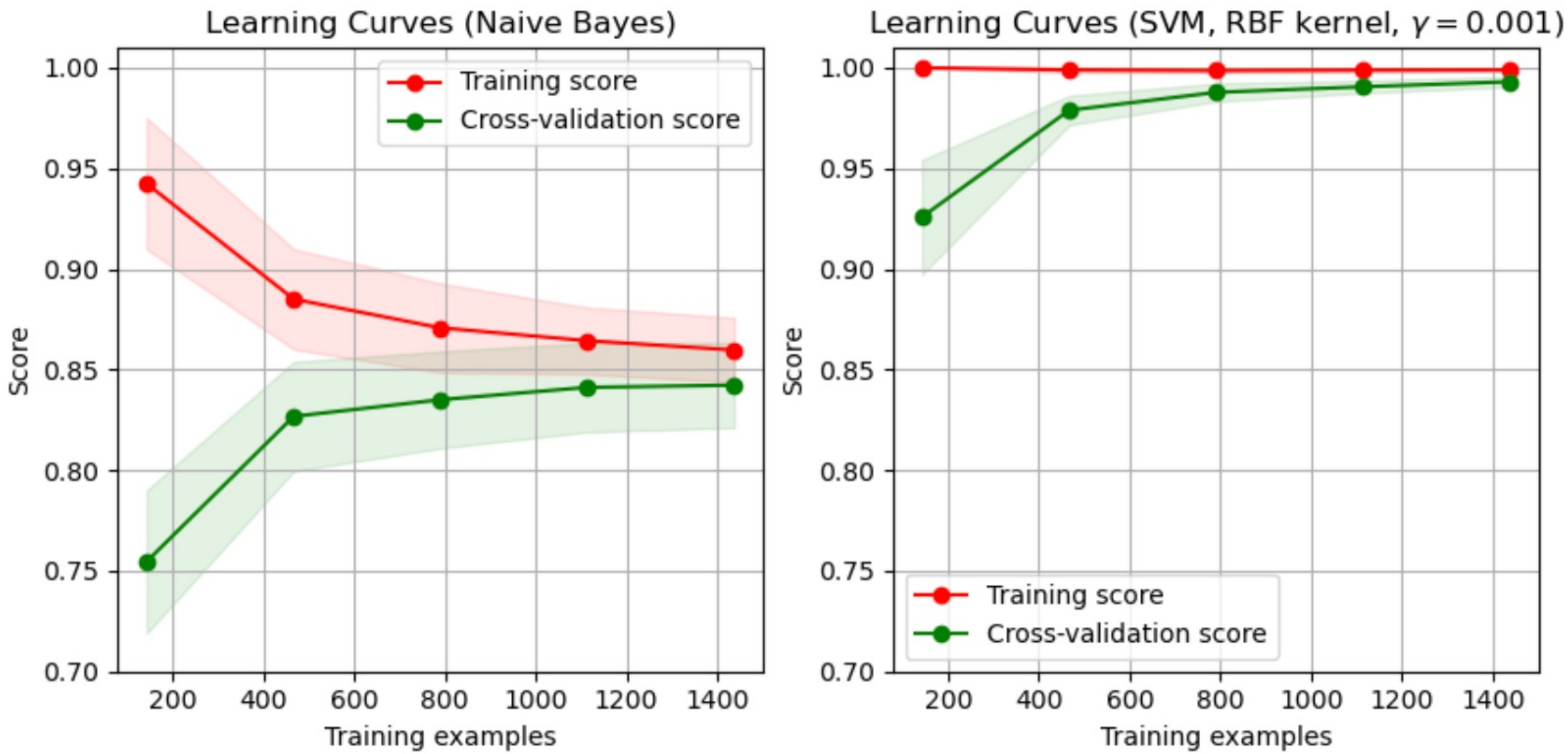
Test data:

New observations from the intended prediction situation

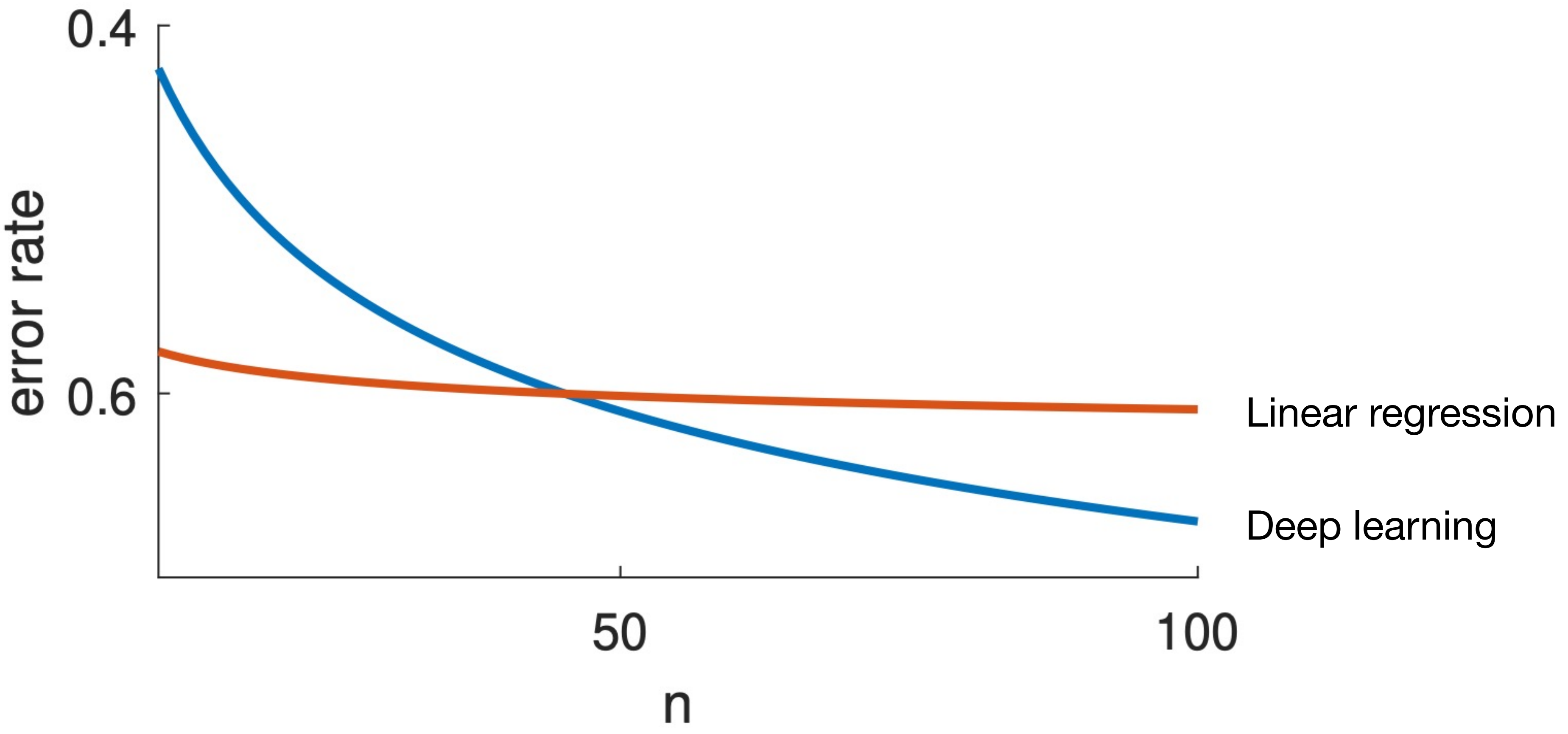
Question: Why don’t these give the same average MSE?


High Bias
High Variance
Error
REDACTED
REDACTED
Model Complexity

Learning curves: *n* vs. performance



Learning curves



Viering & Loog (2021). *The Shape of Learning Curves: a Review*. https://arxiv.org/pdf/2103.10948.pdf

The train-val-test paradigm

- The idea is that the average squared error in the test set $\mathrm{MSE}_{\mathrm{test}}$ is a good estimate of the "Bayes error" E(MSE)
- This only holds when the test set is "like" the intended prediction situation!

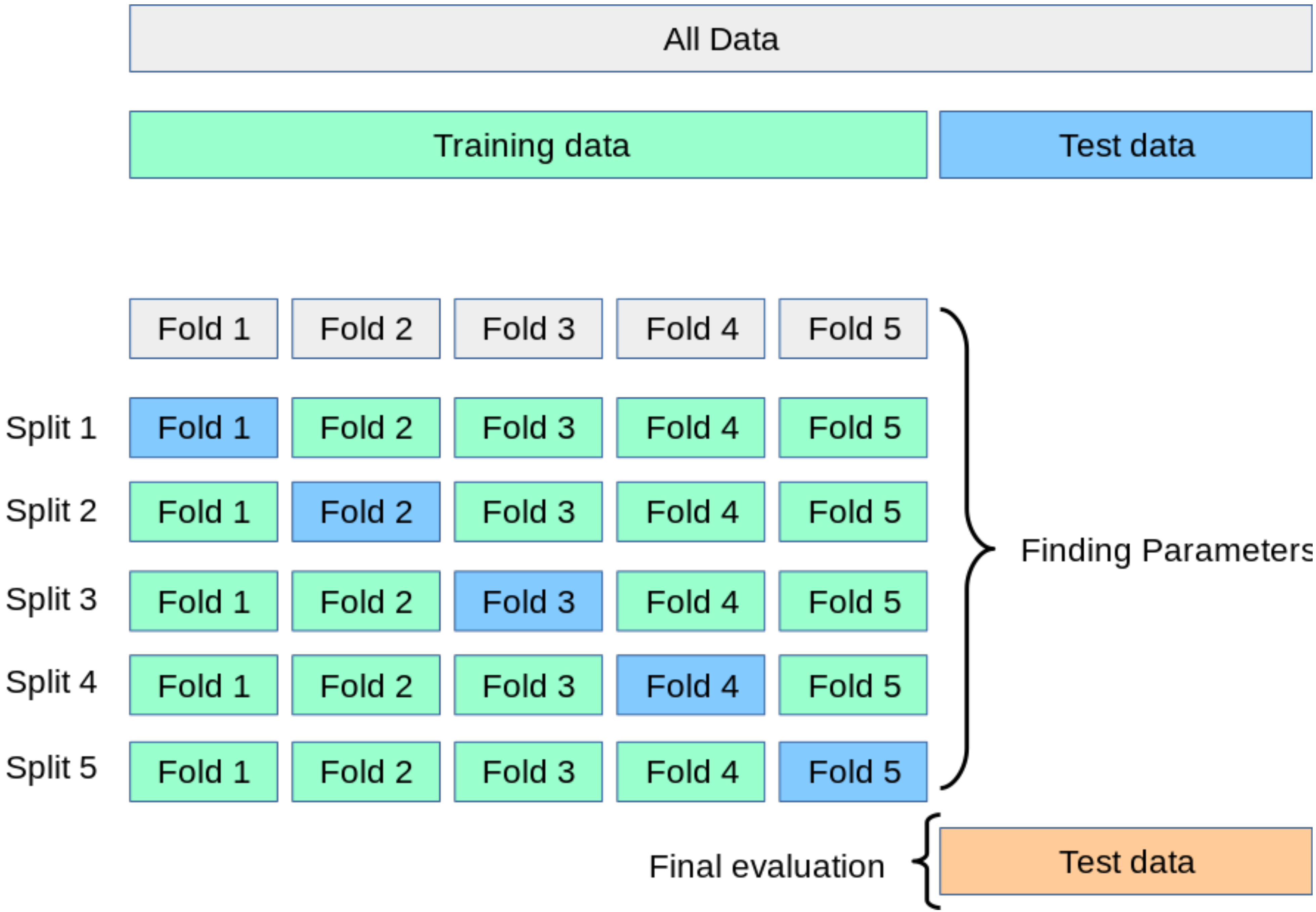
Drawbacks of train/dev/test

- the validation estimate of the test error can be **highly variable**, depending on precisely which observations are included in the training set and which observations are included in the validation set.
- In the validation approach, only a subset of the observations — those that are included in the training set rather than in the validation set — are used to fit the model.
- This suggests that the validation set error may tend to **overestimate the test error** for the model fit on the entire data set.

From `https://lagunita.stanford.edu/c4x/HumanitiesScience/StatLearning`

K-fold crossvalidation

- “Cross-validation” often used to replace single dev set approach;
- Perform the train/dev split several times, and average the result.
- When K = 1, “leave-one-out”;
- Usually K = 5 or K = 10



All Data

Training data | Test data

Fold 1 | Fold 2 | Fold 3 | Fold 4 | Fold 5

Split 1 | Fold 1 | Fold 2 | Fold 3 | Fold 4 | Fold 5

Split 2 | Fold 1 | Fold 2 | Fold 3 | Fold 4 | Fold 5

Split 3 | Fold 1 | Fold 2 | Fold 3 | Fold 4 | Fold 5

Split 4 | Fold 1 | Fold 2 | Fold 3 | Fold 4 | Fold 5

Split 5 | Fold 1 | Fold 2 | Fold 3 | Fold 4 | Fold 5

Finding Parameters

Final evaluation | Test data

Consider a simple regression used to predict an outcome:

1. Starting with 5000 predictors and 500 cases, find the 100 predictors having the largest correlation with the outcome;
2. We then fit a linear regression, using only these 100 predictors.

Class exercise:

- How do we estimate the test set performance of this classifier?
- Can we apply cross-validation in step 2, forgetting about step 1?

Conclusion

- Choose your **data**, **goal**, and **performance metric** with care
- Bias and variance trade off, in **theory** and **practice**;
- We try to estimate this using **train/dev/test** paradigm;
- Getting good test data is difficult problem;
- **Cross-validation** is a useful alternative to separate dev set;
- Beware that *any* procedure that makes decisions based on the data requires validation!
- **Machine learning is not (only) about models/algorithms. It is just as much about data, the reality of your task, etc.**