Lecture 1
Introduction

ISLR 1,2, ESL 1,2

Isabel Valera

Jilles Vreeken

21 October 2021

UNIVERSITÄT DES SAARLANDES

CISPA HELMHOLTZ CENTER FOR INFORMATION SECURITY

Applications of Statistical Learning

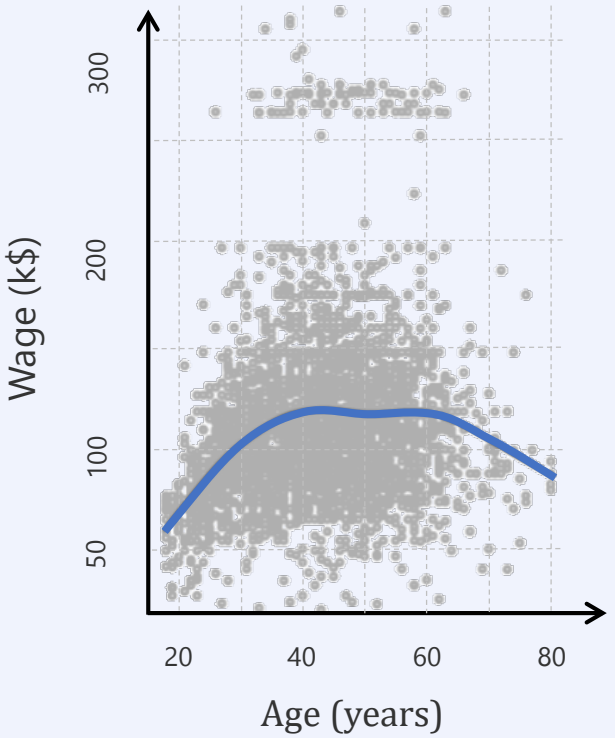
Wage data
continuous output, regression problem

Data 3000 data records relating to wages of males in the Atlantic region of the US

Goal Understand association between age, education, calendar year and wage

Observations

1. wage increases with age before 60 and decreases after 60



Scatter plot
Blue line: smoothed average

Applications of Statistical Learning

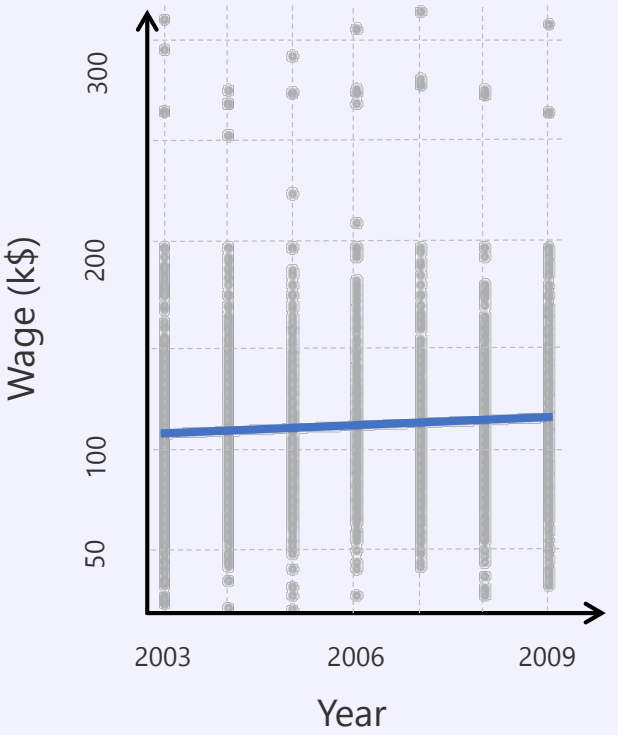
Wage data
continuous output, regression problem

Data 3000 data records relating to wages of males in the Atlantic region of the US

Goal Understand association between age, education, calendar year and wage

Observations

1. wage increases with age before 60, and decreases with age after 60
2. slight linear increase of wage over time ($10,000 over six years)



Scatter plot
Blue line: linear regression

Applications of Statistical Learning

Wage data
continuous output, regression problem

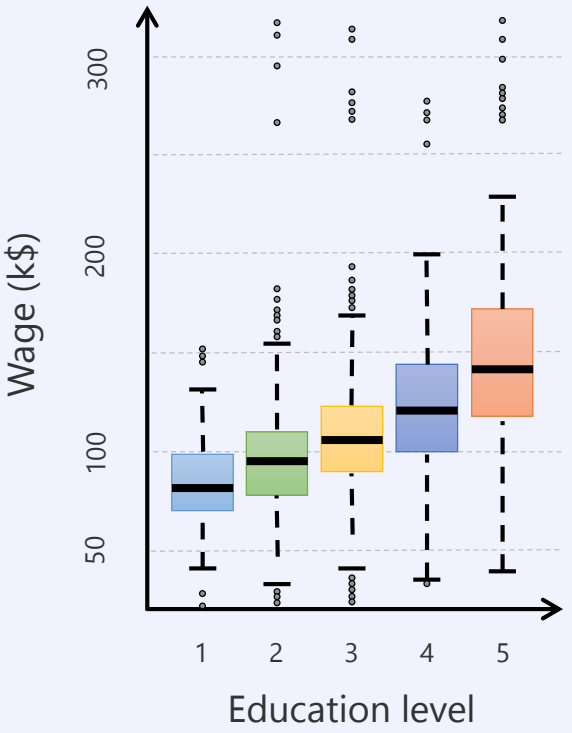
Data 3000 data records relating to wages of males in the Atlantic region of the US

Goal Understand association between age, education, calendar year and wage

Observations

1. wage increases with age before 60, and decreases with age after 60
2. slight linear increase of wage over time ($10,000 over six years)
3. wage increases with level of education

Prediction of wage is best done based on all three features → Chapter 3



Box plots with 25 to 75 percentile as boxes and 5 and 95 percentile as bars

Applications of Statistical Learning

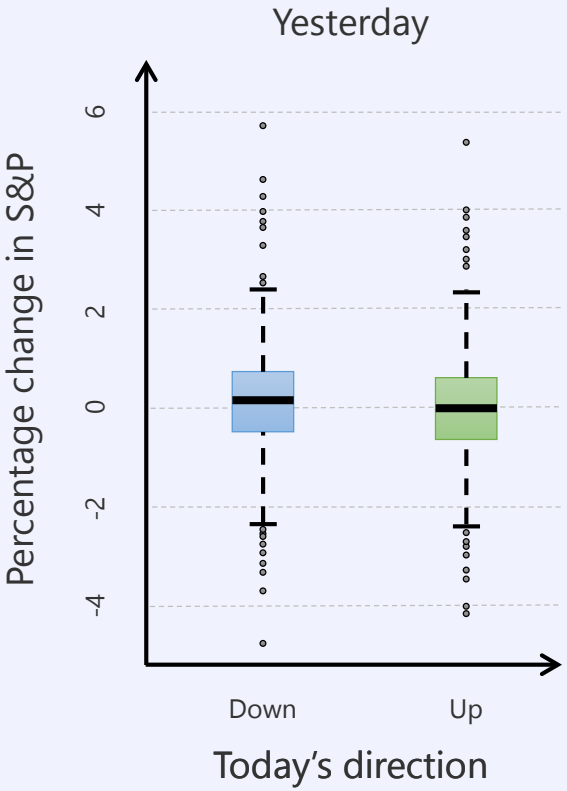
Stock market data
categorical output, classification problem

Data 1250 observations between 2001 and 2005 relating to stock market tendency

Goal predict whether market rises or falls

Observation

1. market increased on 648 days, decreased on 602 days
2. no prediction possible



Applications of Statistical Learning

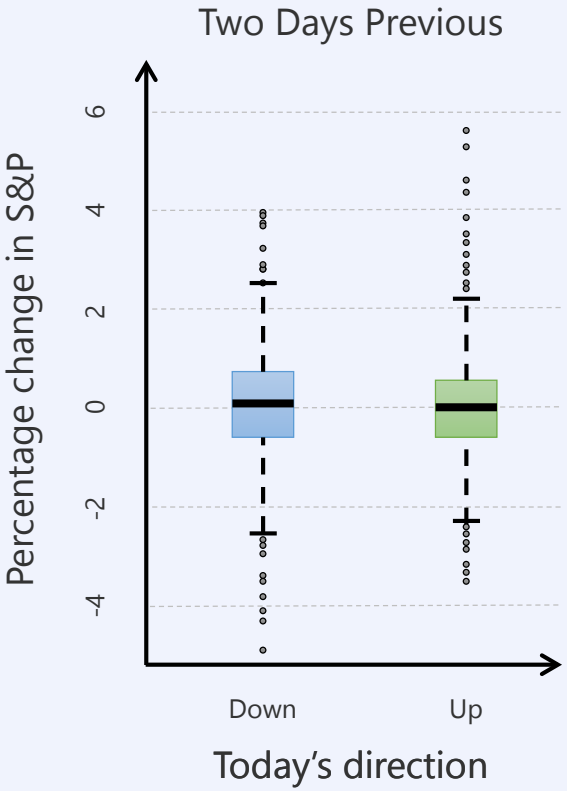
Stock market data
categorical output, classification problem

Data 1250 observations between 2001 and 2005 relating to stock market tendency

Goal predict whether market rises or falls

Observation

1. market increased on 648 days, decreased on 602 days
2. no prediction possible



Applications of Statistical Learning

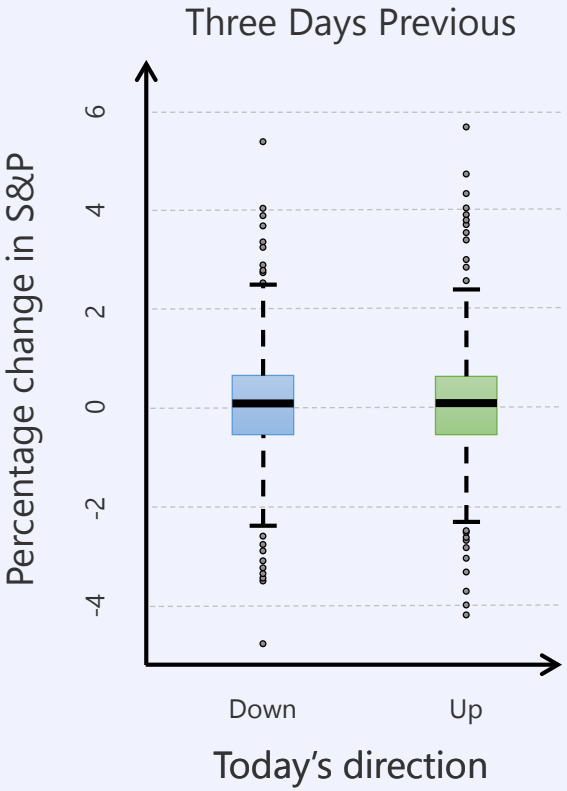
Stock market data
categorical output, classification problem

Data 1250 observations between 2001 and 2005 relating to stock market tendency

Goal predict whether market rises or falls

Observation

1. market increased on 648 days, decreased on 602 days
2. no prediction possible



Applications of Statistical Learning

Stock market data
categorical output, classification problem

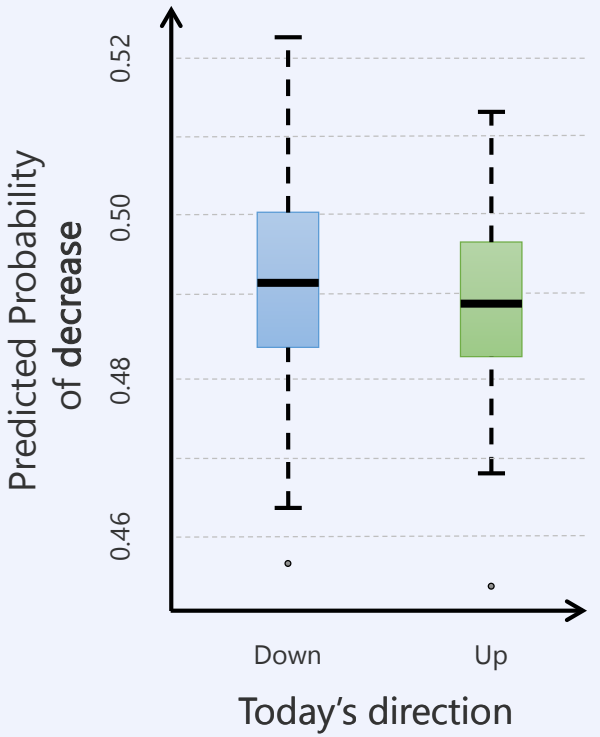
Data 1250 observations between 2001 and 2005 relating to stock market tendency

Goal predict whether market rises or falls

Observation

1. market increased on 648 days, decreased on 602 days
2. no prediction possible

With a more refined analysis of the data we will be able to detect weak trends → Chapter 4. This will allow **predictions** of 60% accuracy (!)



Prediction of stock market tendency with a quadratic discriminant analysis model

Applications of Statistical Learning

Gene expression data

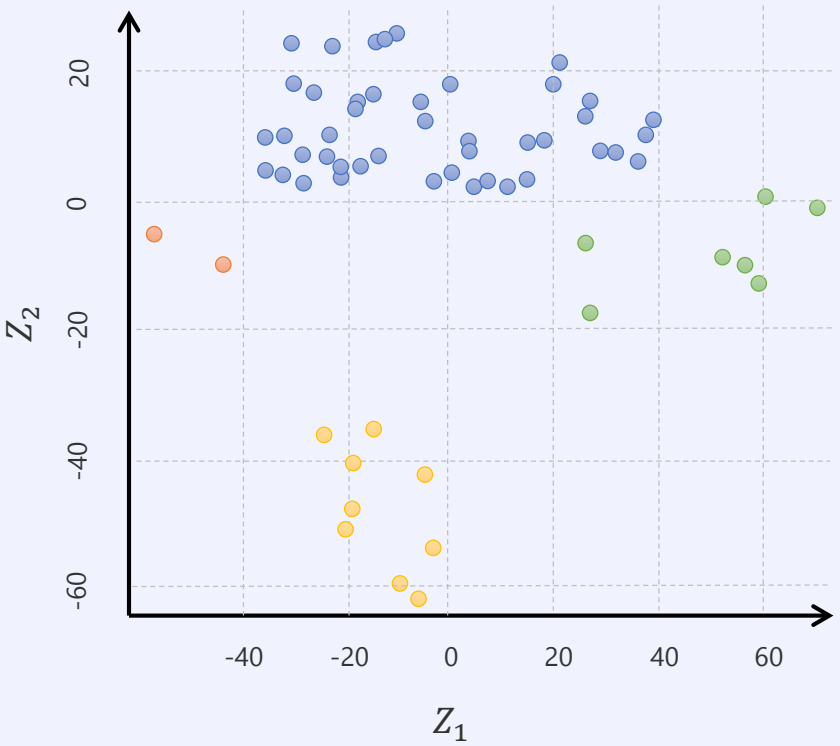
no output variable is available, unsupervised learning problem

Data 64 data records on different cancer cell lines, each comprising 6830 gene expression measurements

Goal find groups of cell lines with similar gene expression profiles

Observations

1. we can naturally group the cell lines into four groups
2. deciding on the number of clusters is often difficult



Plot along the first two principal components. Colors represent grouping

Applications of Statistical Learning

Gene expression data

no output variable is available, unsupervised learning problem

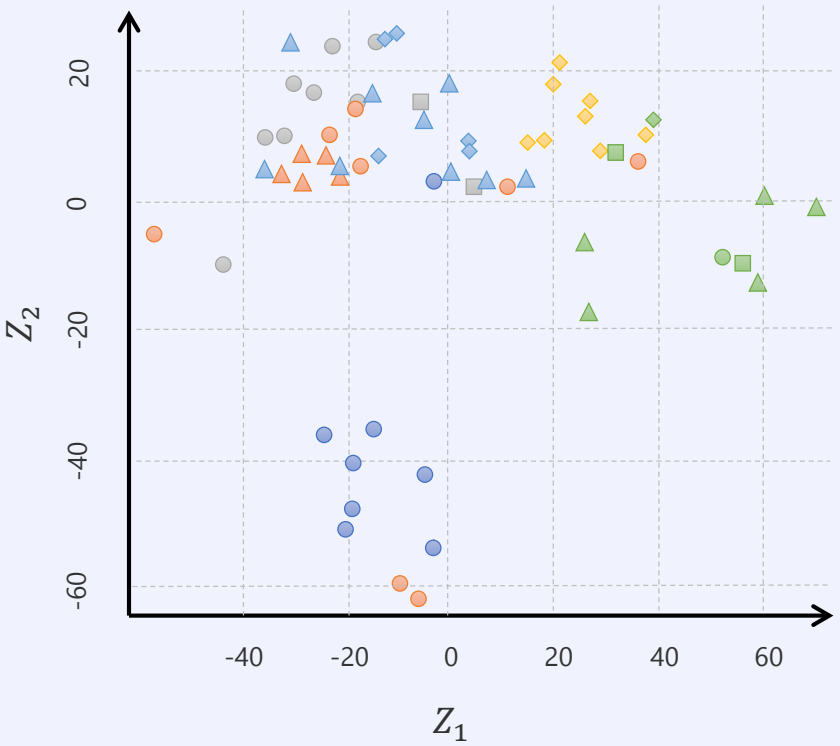
Data 64 data records on different cancer cell lines, each comprising 6830 gene expression measurements

Goal find groups of cell lines with similar gene expression profiles

Observations

1. we can naturally group the cell lines into four groups
2. deciding on the number of clusters is often difficult

Unsupervised learning affords **exploratory data analysis** → Chapter 10



Plot along the first two principal components. Colors represent different cancer types

Introduction

ISLR 2, ESL 2

Example Advertising

Data on sales of a product in 200 markets, and
on advertising budgets via TV, radio and newspaper

Goal adjust advertising budgets to maximize sales

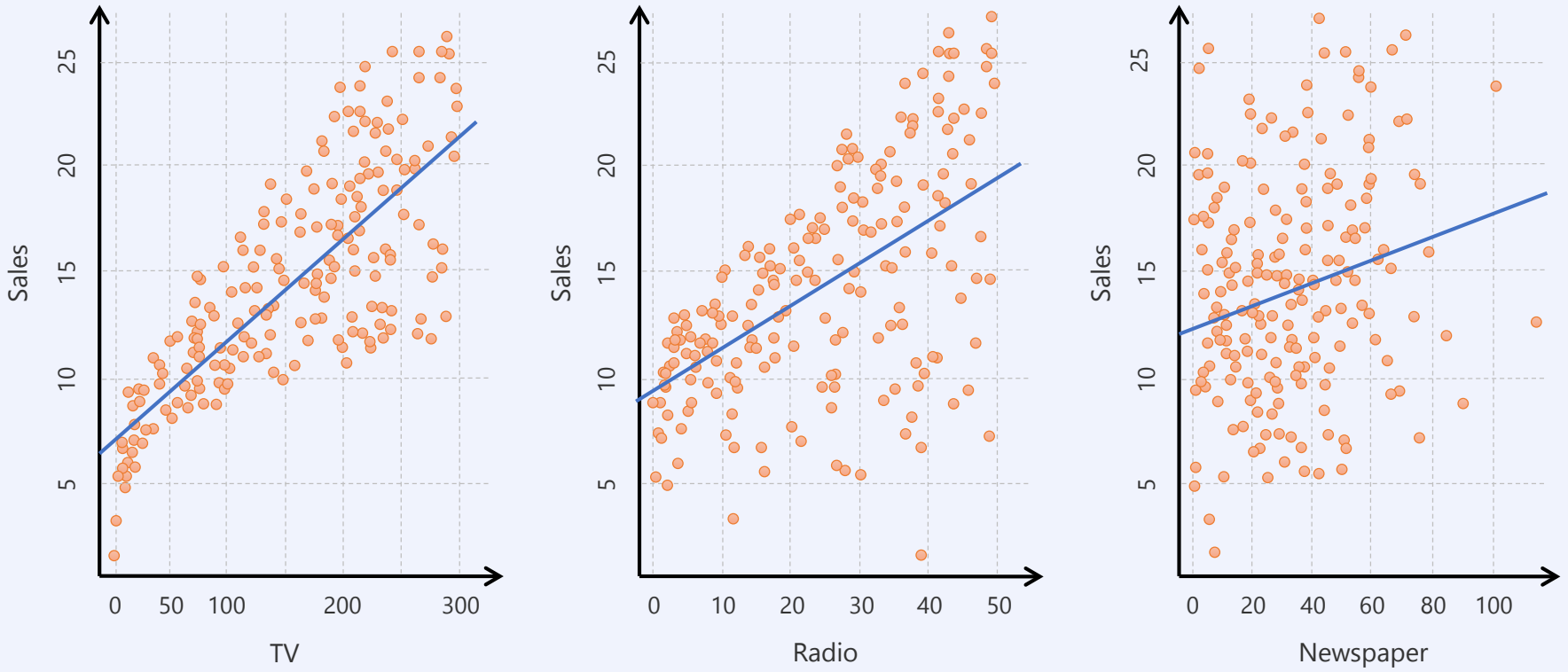
- advertising budgets are **input variables** X
  (aka predictors, features, independent variables)
  - X_1 TV budget
  - X_2 radio budget
  - X_3 newspaper budget
- sales Y is the **output variable** (aka response, dependent variable)

In general, we assume a relationship between X and Y of the form

$$Y = f(X) + \epsilon = f(X_1, X_2, \ldots, X_p) + \epsilon$$

where ϵ is a random additive error term with zero mean

Example Advertising Dataset

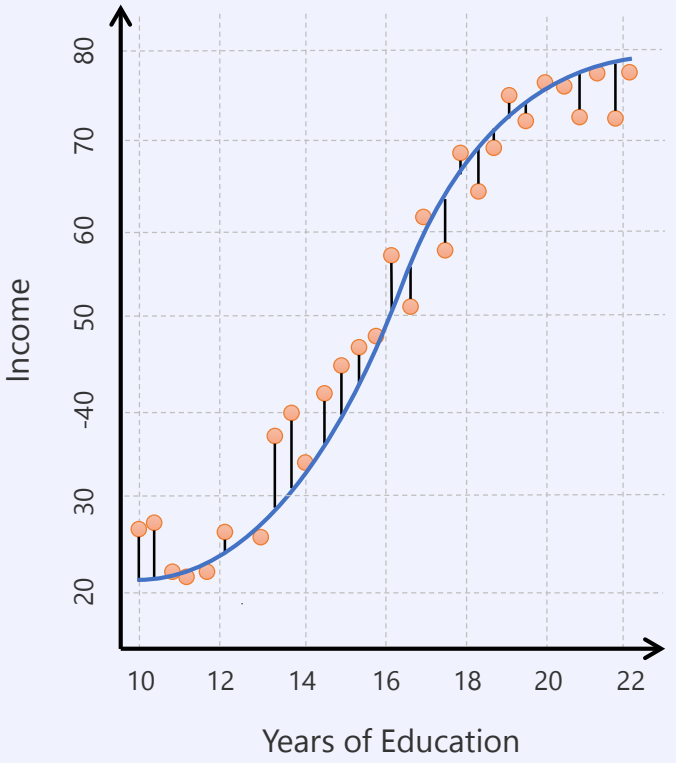


Numbers are in thousands of dollars
In general, sales increase as advertising is stepped up.
The blue lines result from least-squares linear regression to the variable along the x-axis

Example Income – Simulated Dataset

Here the relationship between wage and years of education is **nonlinear**

- this is a **simulated** example (synthetic data set), so the blue line represents the true functional relationship which, in general, is unknown and must be estimated



Why estimate f? Prediction

Often **inputs X are available**, **output Y is not**, but is **desired**

- estimating the output then effects a **prediction**

$$\hat{Y} = \hat{f}(X)$$

We often treat $\hat{f}$ as a black box whose form is not of interest

- **for example, input** is blood profile of a patient, and
  output is the patient's risk of a severe reaction to a drug

Why estimate f? Prediction

Often **inputs X are available**, **output Y is not**, but is **desired**

- estimating the output then effects a **prediction**

$$\hat{Y} = \hat{f}(X)$$

We often treat $\hat{f}$ as a black box whose form is not of interest

- the accuracy of $\hat{Y}$ depends on the **reducible error** and the **irreducible error**
- for fixed X and f we have

$$\mathrm{E}\left[Y - \hat{Y}\right]^2 = E\left[f(X) + \epsilon - \hat{f}(X)\right]^2$$
$$= \underbrace{\mathrm{E}\left[f(X) - \hat{f}(X)\right]^2}_{\text{reducible error}} + \underbrace{Var(\epsilon)}_{\text{irreducible error}}$$

Expectation over all possible training sets

Proof → Homework

The goal of prediction is to minimize the reducible error.
The irreducible error cannot be avoided

Why estimate f? Inference

Often we want to go beyond treating $\hat{f}$ as a black box. Rather, we want to **understand the relation** between input and output

- which predictors strongly associate with the response? Often only few
- what is the relationship between the response and each predictor? Is it positive or negative? Sometimes this depends on other predictors
- is the relationship between the predictors linear or more complicated?

An **example** for inference is the advertising data with questions as:

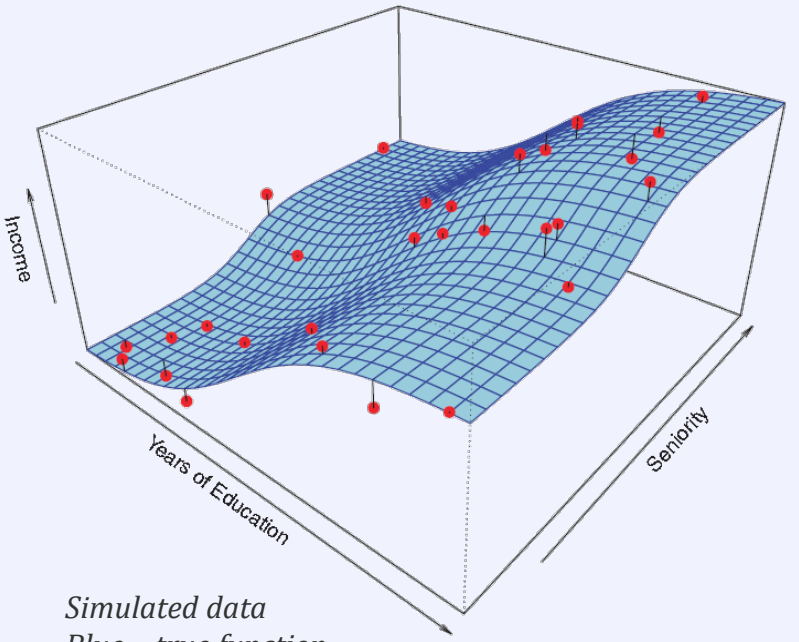
- which media contribute to sales? which generate the biggest boost?
- how much increase in sales is associated with a given increase in TV ads?

Sometimes both prediction and inference are of interest. There is, however, a **tradeoff** between the two. Linear models, for example, allow easily interpretable predictions but may not be very accurate

How to estimate f?

We have a set of n **observations** with inputs and outputs (training data), $\{(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)\}$, and are looking for a function $\hat{f}$ such that $Y \approx \hat{f}(X)$ for any observation (X, Y)

- we distinguish between **parametric** and **nonparametric** methods



Simulated data
Blue = true function

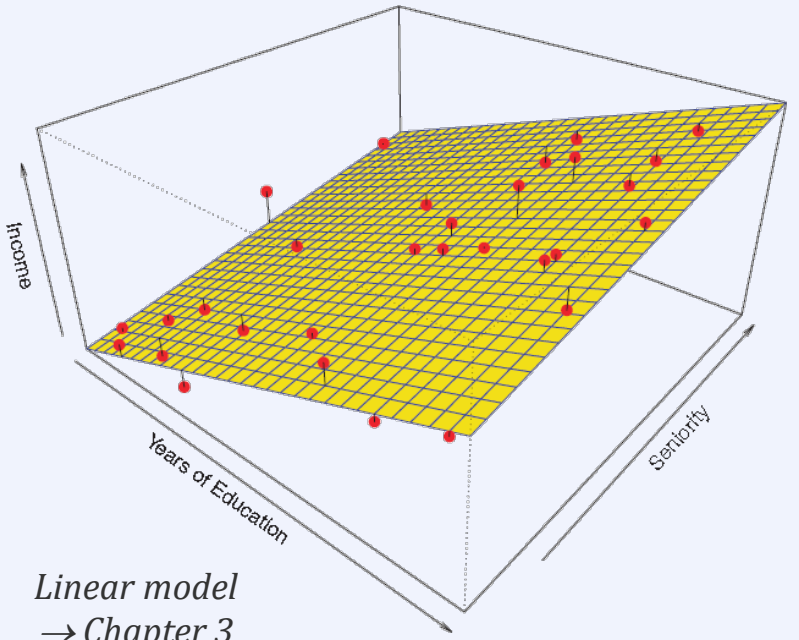
How to estimate f?

We have a set of n **observations** with inputs and outputs (training data), $\{(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)\}$, and are looking for a function $\hat{f}$ such that $Y \approx \hat{f}(X)$ for any observation (X, Y)

- we distinguish between **parametric** and **nonparametric** methods

Parametric Methods

- have a given functional form, usually simple such as a linear model $f(X) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_p X_p$
- estimating $\hat{f}$ means choosing the model parameters
- **problem** the model may not match the true form of f



Linear model
→ Chapter 3

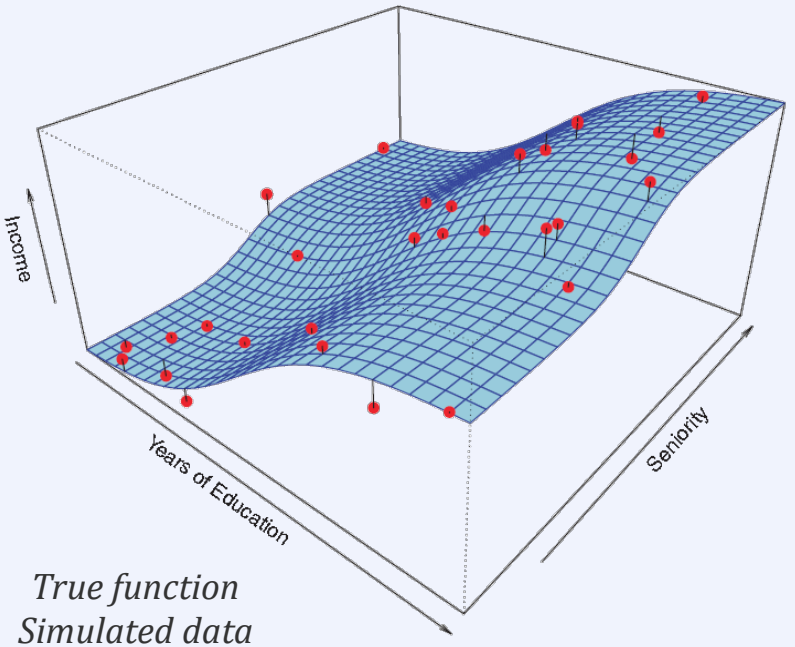
How to estimate f?

We have a set of n **observations** with inputs and outputs (training data), $\{(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)\}$, and are looking for a function $\hat{f}$ such that $Y \approx \hat{f}(X)$ for any observation (X, Y)

- we distinguish between **parametric** and **nonparametric** methods

Nonparametric Methods

- here we aim at finding the form of f
- choosing the form gives us much more freedom
- we have to choose many parameters; this requires many observations
- otherwise, we risk modelling the noise in the training set: **overfitting**



True function
Simulated data

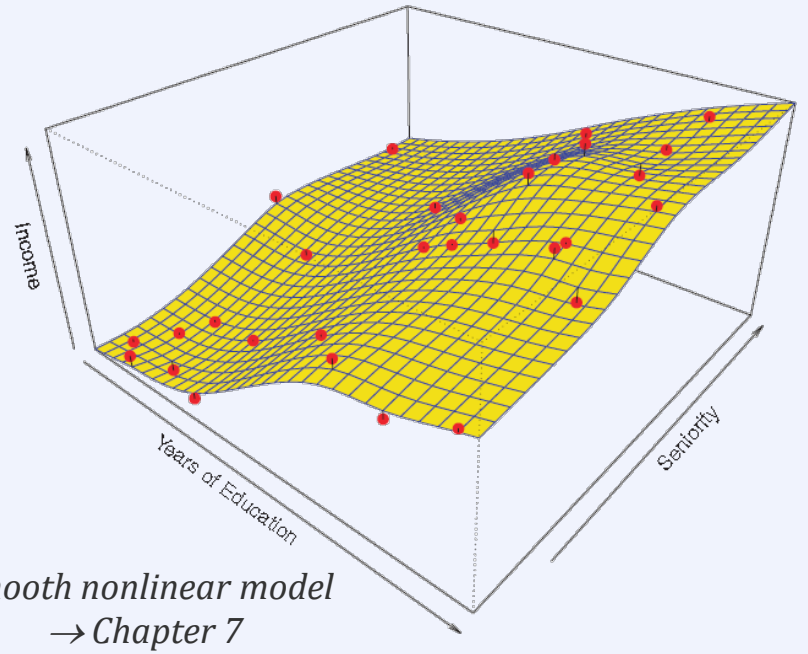
How to estimate f?

We have a set of n **observations** with inputs and outputs (training data), $\{(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)\}$, and are looking for a function $\hat{f}$ such that $Y \approx \hat{f}(X)$ for any observation (X, Y)

- we distinguish between **parametric** and **nonparametric** methods

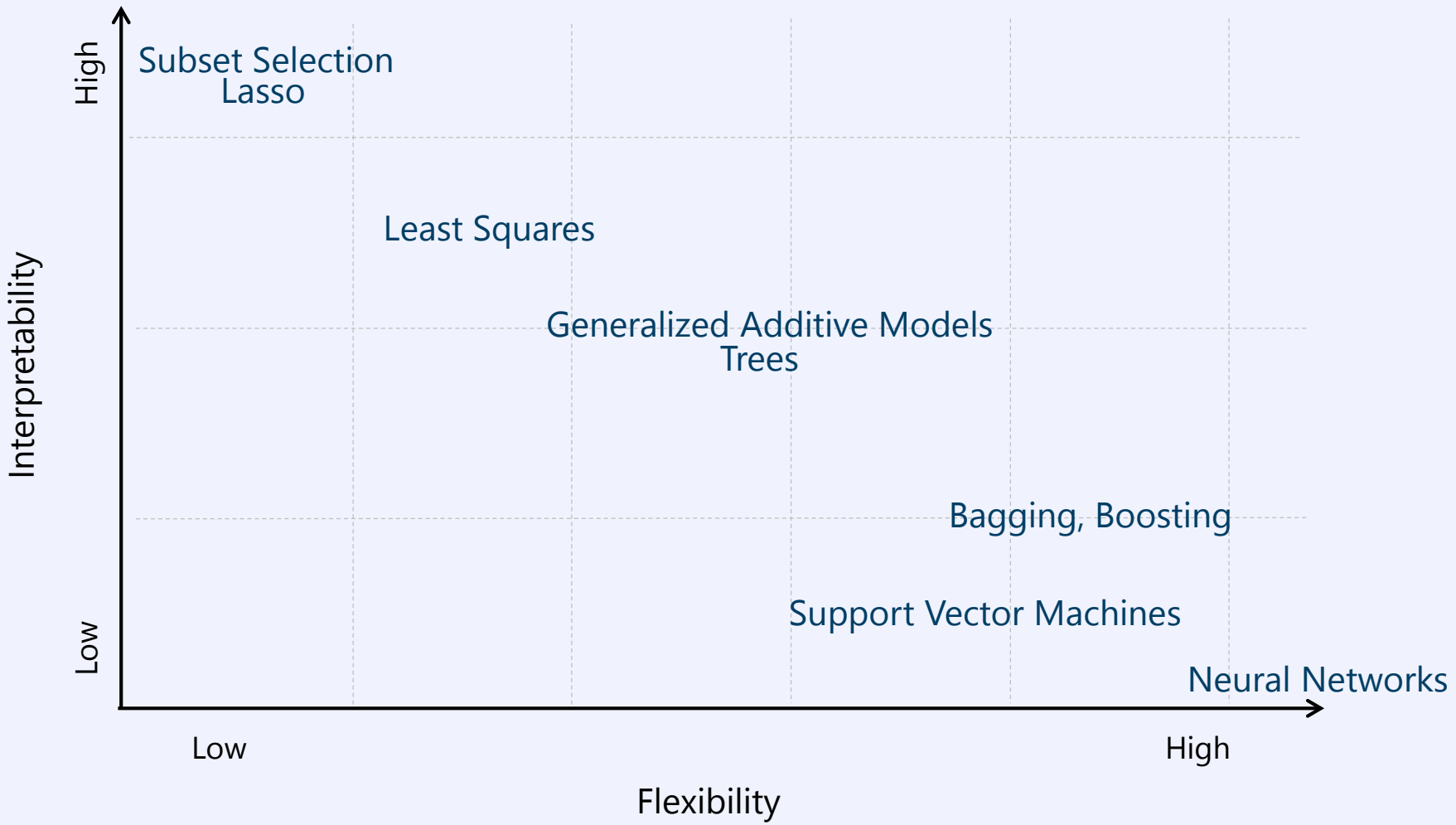
Nonparametric Methods

- here we aim at finding the form of f
- choosing the form gives us much more freedom
- we have to choose many parameters; this requires many observations
- otherwise, we risk modelling the noise in the training set: **overfitting**



Smooth nonlinear model
→ Chapter 7

Accuracy vs. Interpretability



Accuracy vs. Interpretability

Why would we ever prefer a more restricted model over a more flexible one?

A flexible model entails a large number of parameters

1. Estimating all parameters is computationally expensive
2. Complicated models are hard to interpret, so especially when inference is the goal, simple models are preferred
3. If we have only few observations, we do not have enough information to accurately estimate many parameters.
   In such cases flexible models incur a high risk of overtraining

Supervised vs. Unsupervised Learning

Supervised Learning

- **data:** inputs and outputs (x_i, y_i) for observations $i = 1, \ldots, n$ following some unknown functional pattern with noise, e.g. $Y = f(X) + \epsilon$
- **goal:** find function $\hat{f}$ such that $Y \approx \hat{f}(X)$ for every conceivably seen input X
  - setting is like that of a student who learns from a teacher (supervisor) giving examples

Semi-supervised learning

- **data:** inputs x_i for observations $i = 1, \ldots, n$, only some outputs y_i
- **goal:** same as for supervised learning, but also leverages unlabeled data

Supervised vs. Unsupervised Learning

Supervised Learning

- **data:** inputs and outputs (x_i, y_i) for observations $i = 1, \dots, n$ following some unknown functional pattern with noise, e.g. $Y = f(X) + \epsilon$
- **goal:** find function $\hat{f}$ such that $Y \approx \hat{f}(X)$ for every conceivably seen input X
  - setting is like that of a student who learns from a teacher (supervisor) giving examples

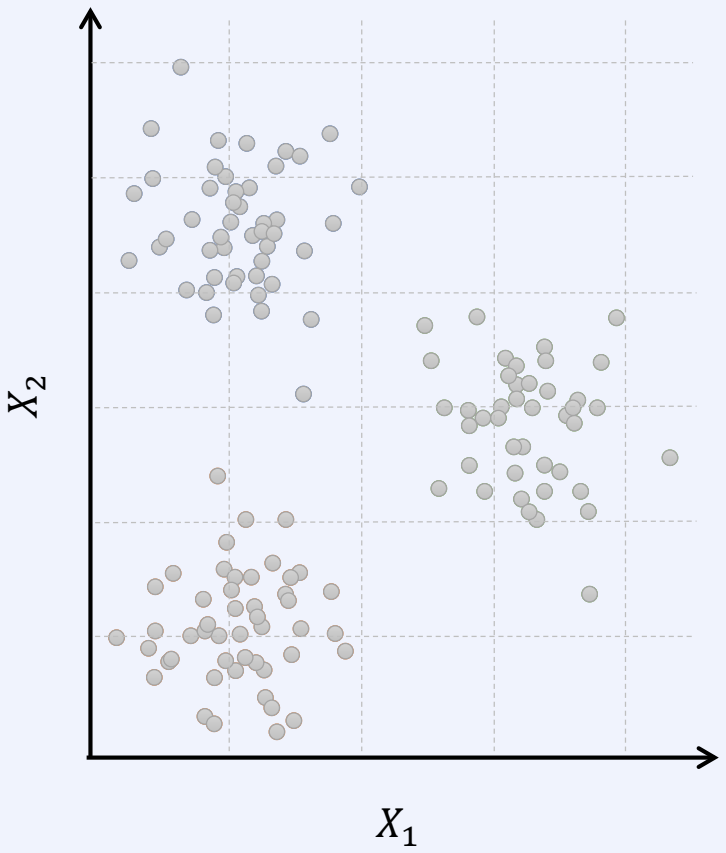
Semi-supervised learning

- **data:** inputs x_i for observations $i = 1, \dots, n$, only some outputs y_i
- **goal:** same as for supervised learning, but also leverages unlabeled data

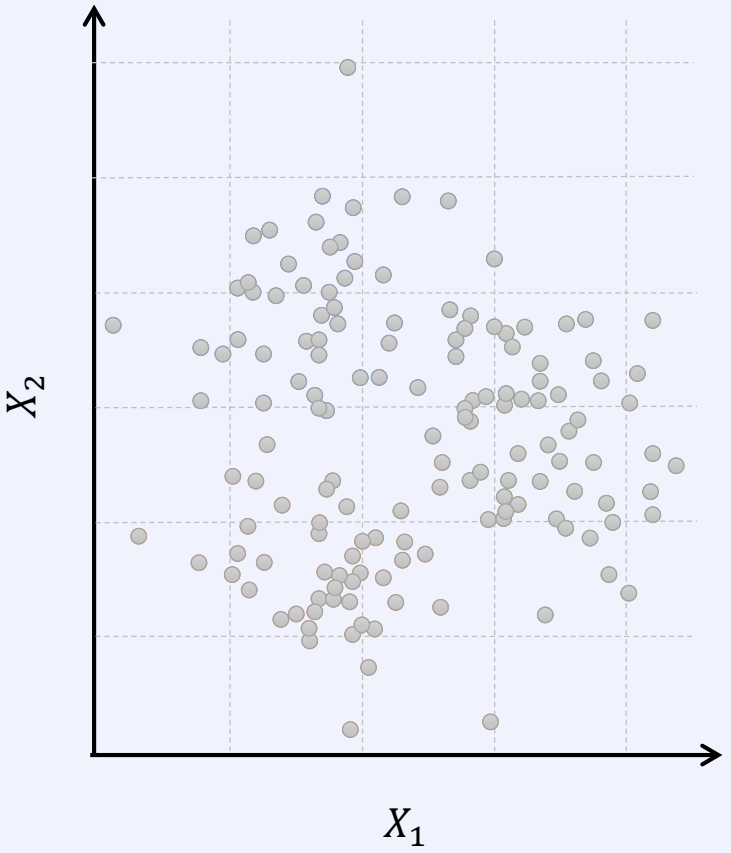
Unsupervised learning

- **data:** inputs x_i for observations $i = 1, \dots, n$, no outputs
- **goal:** elucidate relationships between the variables or the observations
  - often equated with cluster analysis, but many more aspects exist

Example Clustering Problems

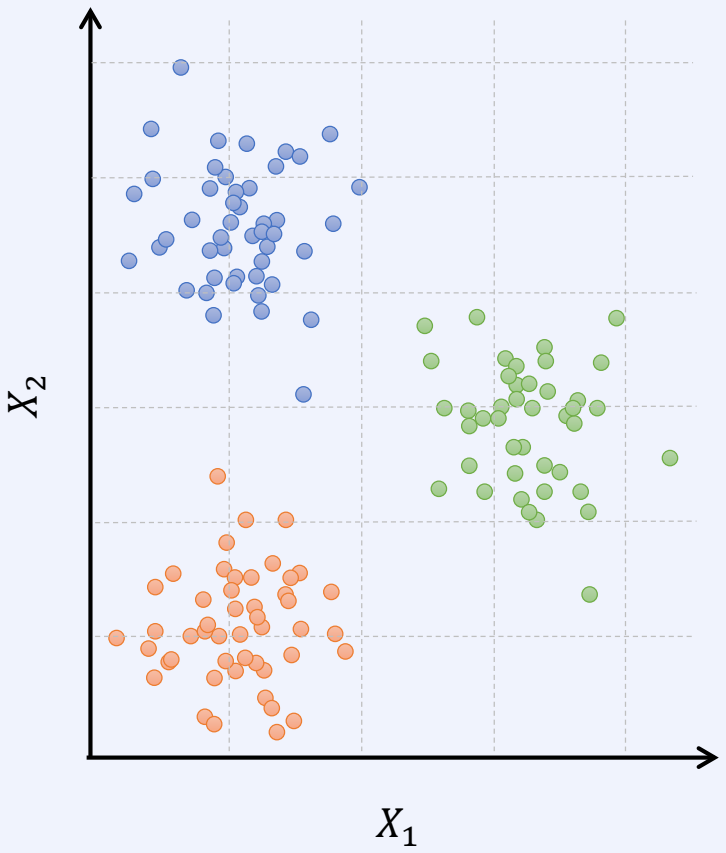


Well separated clusters

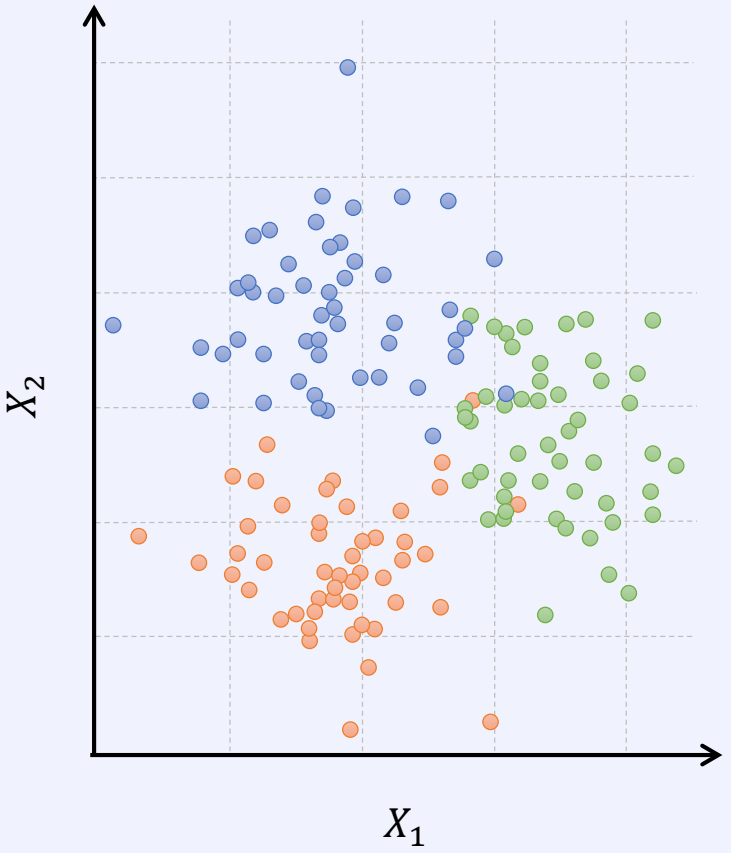


Overlapping clusters

Example Clustering Problems



Well separated clusters



Overlapping clusters

Assessing model accuracy

In regression problems we use the **mean squared error** to assess the quality of fit, here over the training data

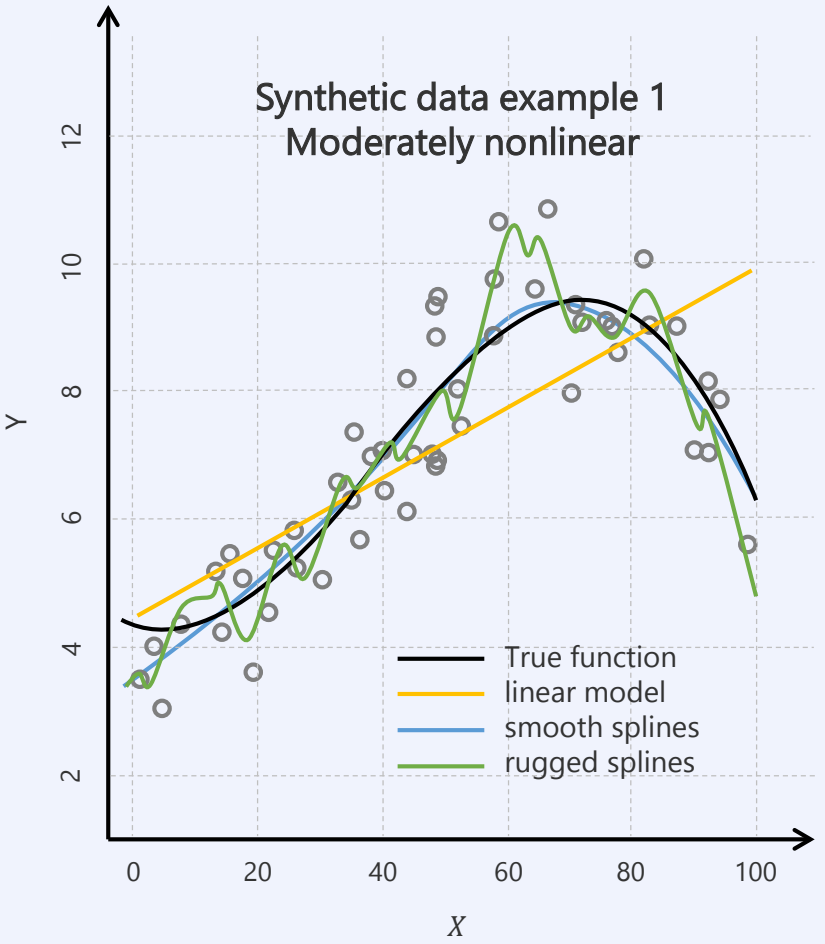
$$MSE = \frac{1}{n}\sum_{i=1}^{n}\left(y_i - \hat{f}(x_i)\right)^2$$

We call this the **training error**

We are more interested in the error over unseen data (x_0, y_0)

$$Ave\left(\hat{f}(x_0) - y_0\right)^2$$

We call this the **test error**, the **generalization error**, or **expected prediction error** (EPE)



Assessing model accuracy

In regression problems we use the **mean squared error** to assess the quality of fit, here over the training data

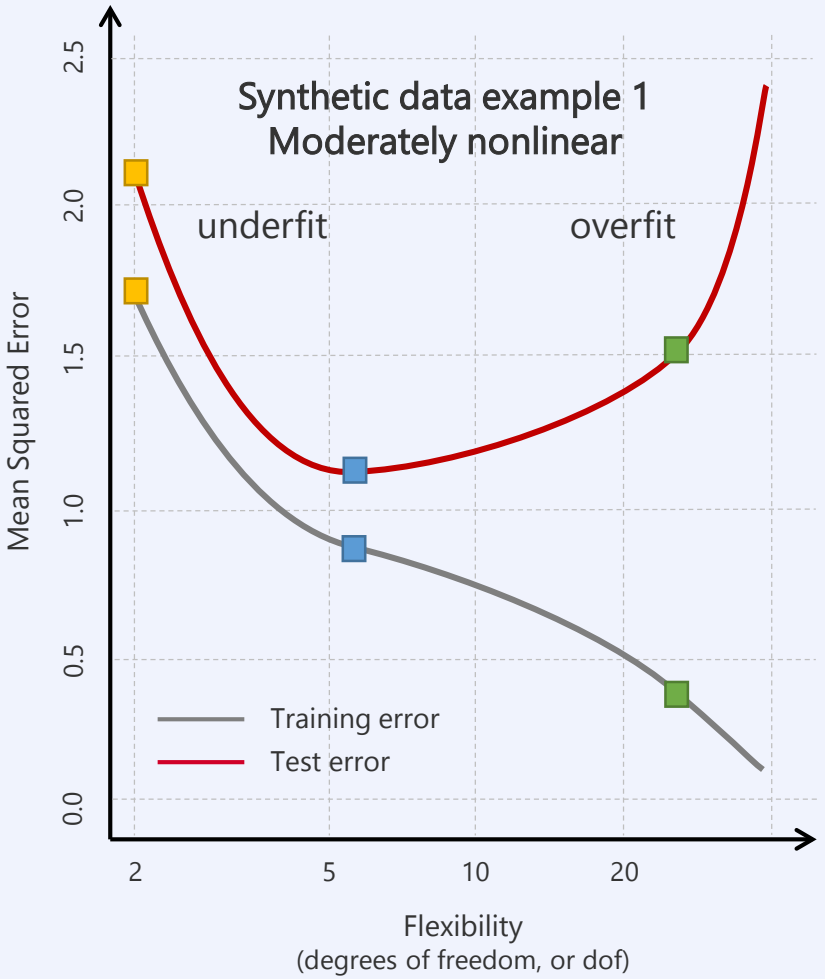
$$MSE = \frac{1}{n}\sum_{i=1}^{n}\left(y_i - \hat{f}(x_i)\right)^2$$

We call this the **training error**

We are more interested in the error over unseen data (x_0, y_0)

$$Ave\left(\hat{f}(x_0) - y_0\right)^2$$

We call this the **test error**, the **generalization error**, or **expected prediction error** (EPE)



Assessing model accuracy

In regression problems we use the **mean squared error** to assess the quality of fit, here over the training data

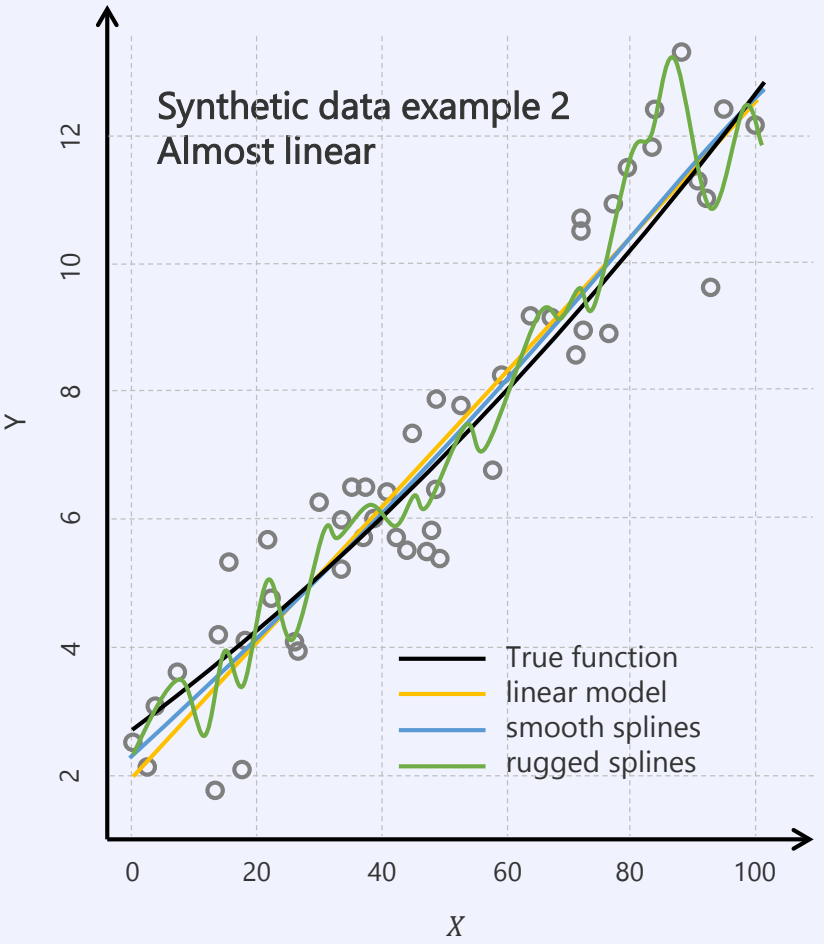
$$MSE = \frac{1}{n}\sum_{i=1}^{n}\left(y_i - \hat{f}(x_i)\right)^2$$

We call this the **training error**

We are more interested in the error over unseen data (x_0, y_0)

$$Ave\left(\hat{f}(x_0) - y_0\right)^2$$

We call this the **test error**, the **generalization error**, or **expected prediction error** (EPE)



Assessing model accuracy

In regression problems we use the **mean squared error** to assess the quality of fit, here over the training data

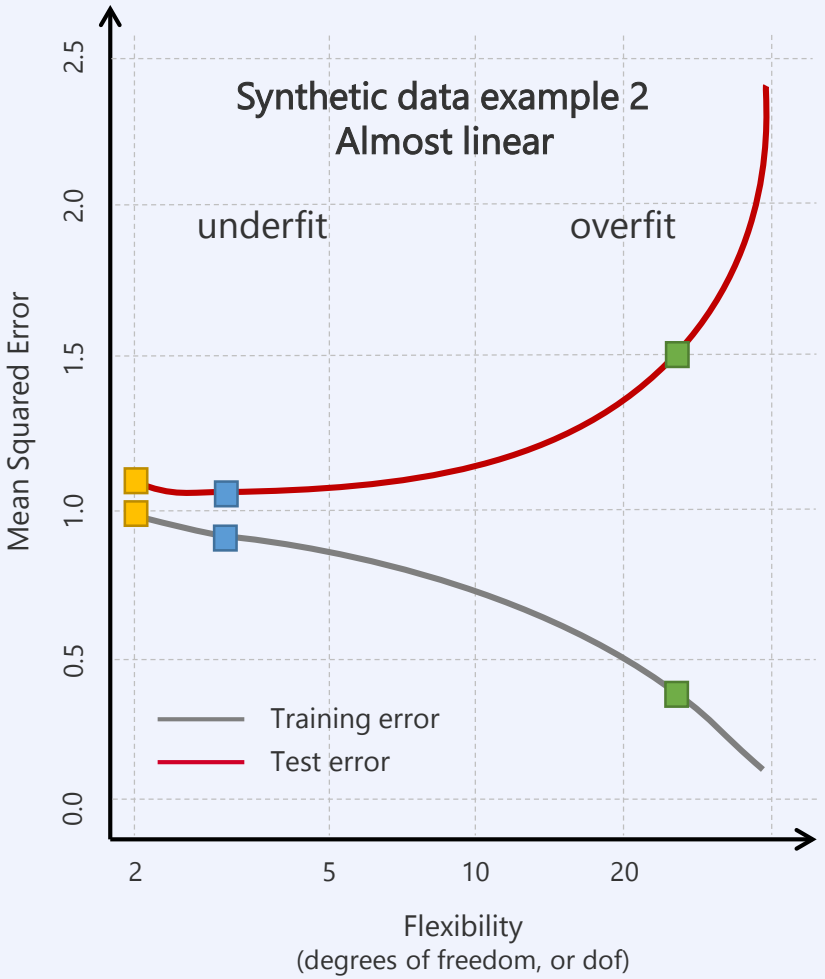
$$MSE = \frac{1}{n}\sum_{i=1}^{n}\left(y_i - \hat{f}(x_i)\right)^2$$

We call this the **training error**

We are more interested in the error over unseen data (x_0, y_0)

$$Ave\left(\hat{f}(x_0) - y_0\right)^2$$

We call this the **test error**, the **generalization error**, or **expected prediction error** (EPE)



Assessing model accuracy

In regression problems we use the **mean squared error** to assess the quality of fit, here over the training data

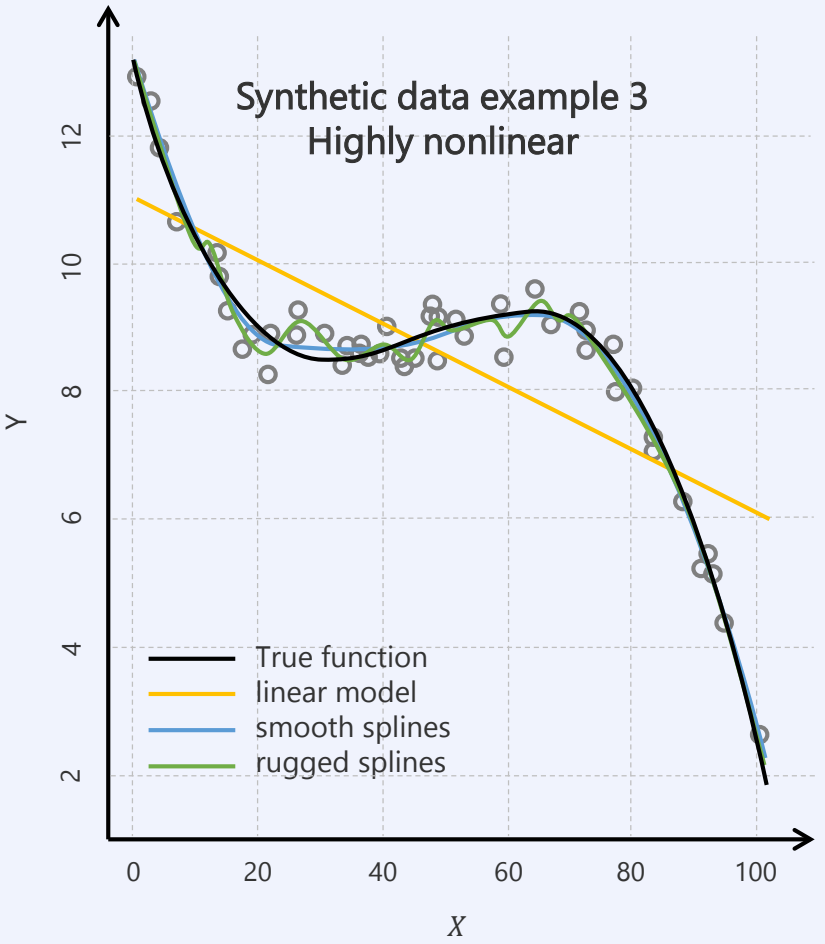
$$MSE = \frac{1}{n}\sum_{i=1}^{n}\left(y_i - \hat{f}(x_i)\right)^2$$

We call this the **training error**

We are more interested in the error over unseen data (x_0, y_0)

$$Ave\left(\hat{f}(x_0) - y_0\right)^2$$

We call this the **test error**, the **generalization error**, or **expected prediction error** (EPE)



Assessing model accuracy

In regression problems we use the **mean squared error** to assess the quality of fit, here over the training data

$$MSE = \frac{1}{n}\sum_{i=1}^{n}\left(y_i - \hat{f}(x_i)\right)^2$$

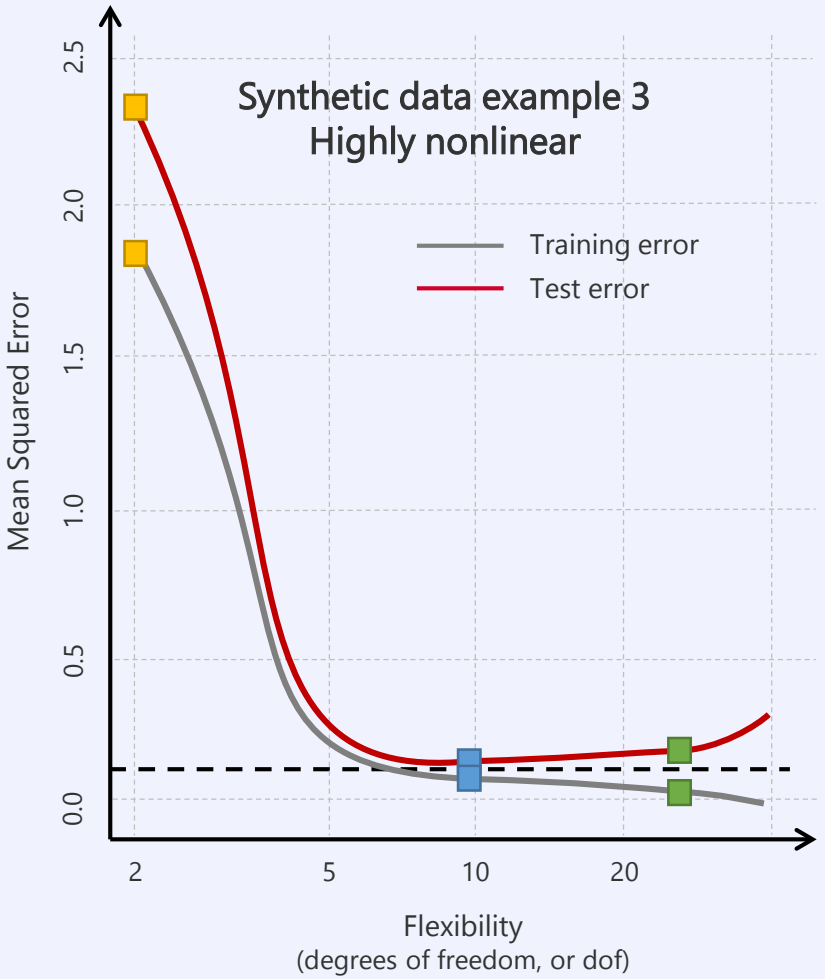
We call this the **training error**

We are more interested in the error over unseen data (x_0, y_0)

$$Ave\left(\hat{f}(x_0) - y_0\right)^2$$

We call this the **test error**

- if the functional dependence between input and output is not known, the test error is hard to estimate



Assessing model accuracy

In regression problems we use the **mean squared error** to assess the quality of fit, here over the training data

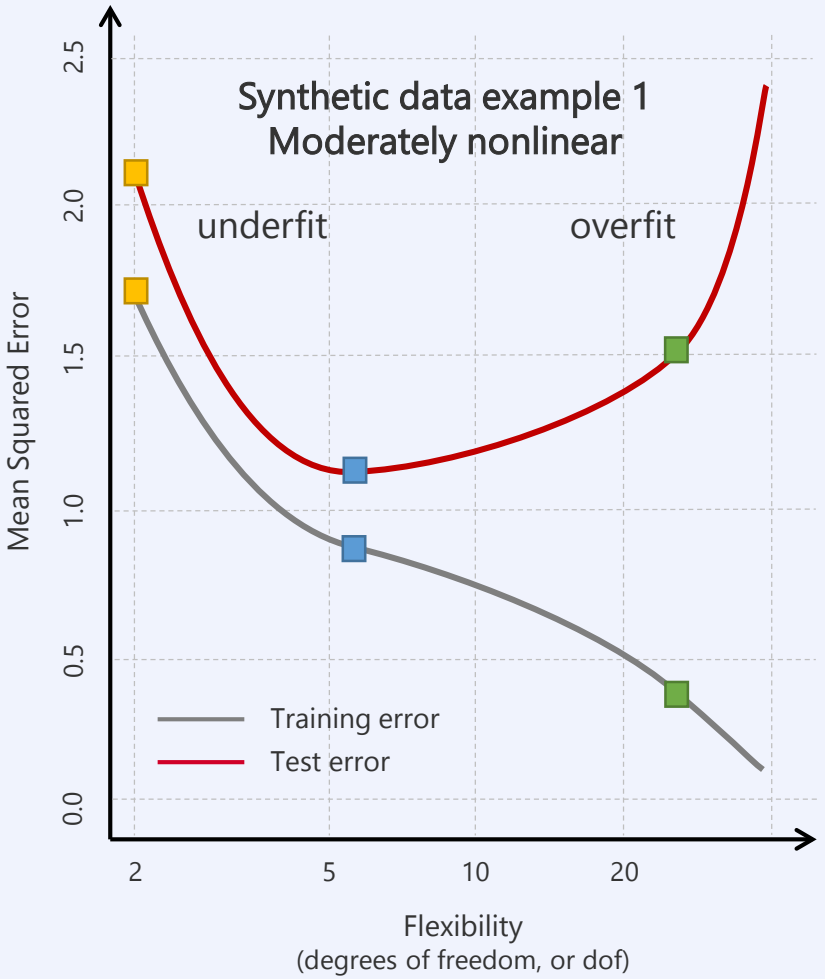
$$MSE = \frac{1}{n}\sum_{i=1}^{n}\left(y_i - \hat{f}(x_i)\right)^2$$

We call this the **training error**

We are more interested in the error over unseen data (x_0, y_0)

$$Ave\left(\hat{f}(x_0) - y_0\right)^2$$

We call this the **test error**, the **generalization error**, or **expected prediction error** (EPE)



Bias-Variance Tradeoff

The shape of the curve for test error is due to a basic tradeoff in the MSE

$$E\left(y_0 - \hat{f}(x_0)\right)^2 = Var\left(\hat{f}(x_0)\right) + \left[Bias\left(\hat{f}(x_0)\right)\right]^2 + Var(\epsilon)$$

Expectation over all possible training sets

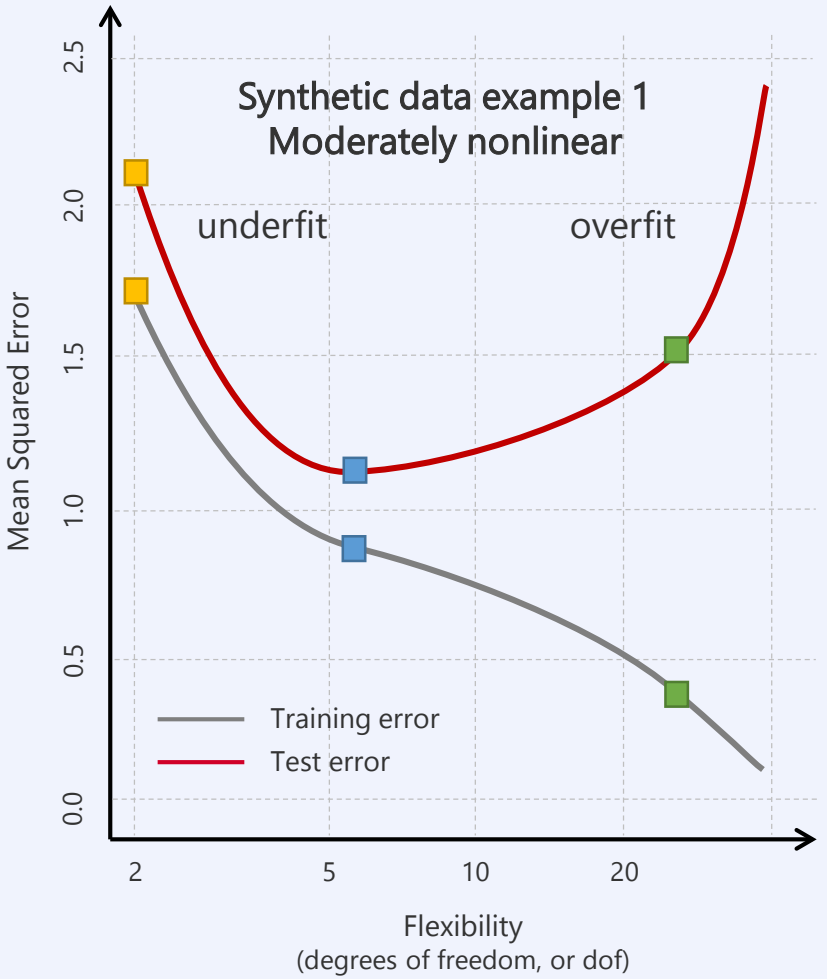
Proof → Homework

Here **bias** is the systematic deviation of the estimate from the true value

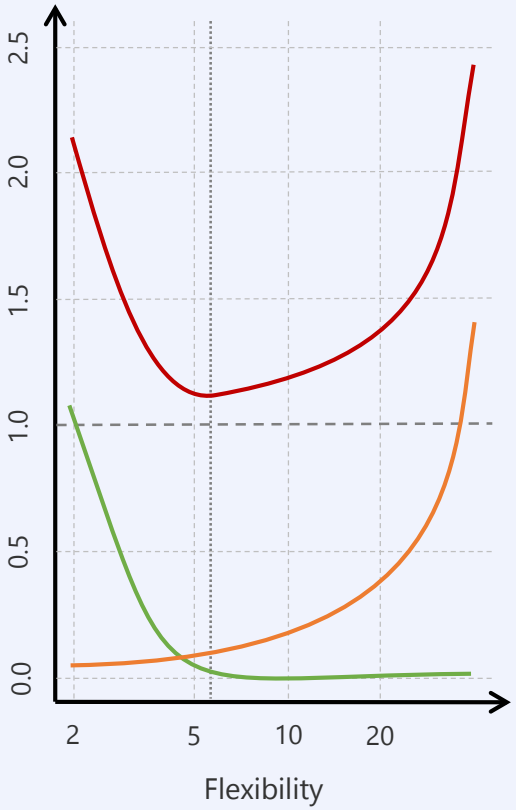
$$Bias\left(\hat{f}(x_0)\right) = E(\hat{f}(x_0) - y_0)$$

and **variance** is the variation of the estimate between different training sets

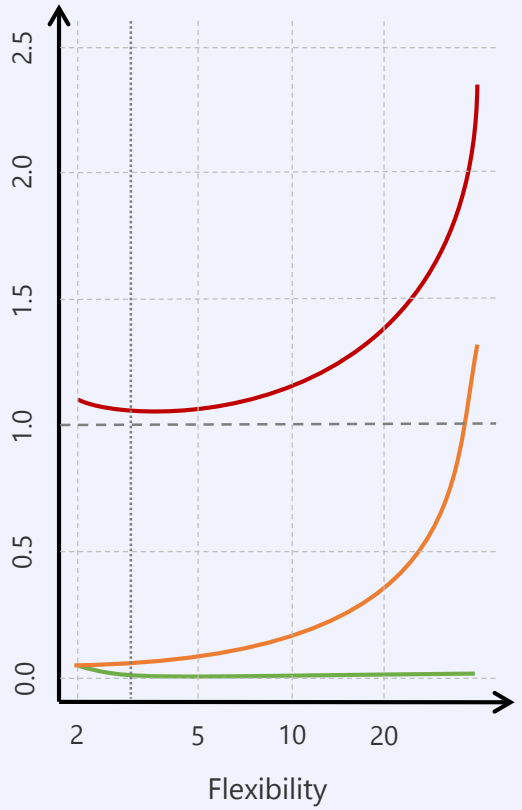
$$Var\left(\hat{f}(x_0)\right) = E\left(\hat{f}(x_0) - E\left(\hat{f}(x_0)\right)\right)^2$$



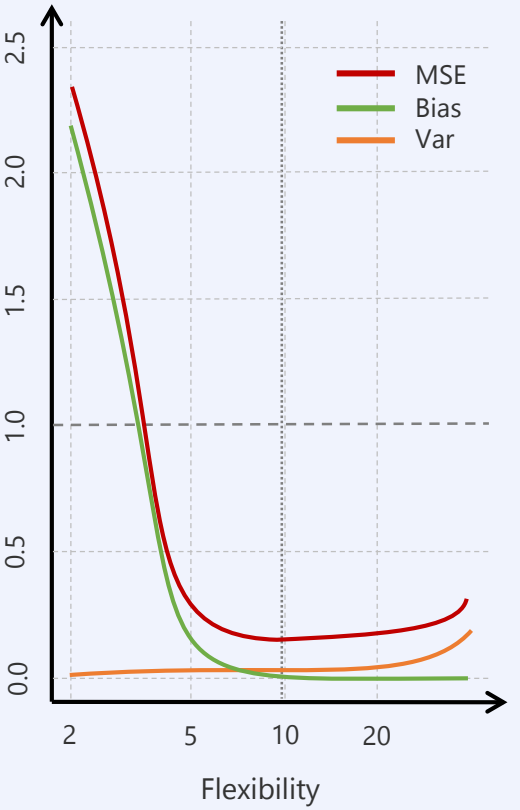
Bias-Variance Decomposition



Synthetic data example 1
Moderately nonlinear function



Synthetic data example 2
Almost linear function



Synthetic data example 3
Highly nonlinear function