IG.3510-Machine Learning

Lecture 1: Introduction to Machine Learning

Dr. Patricia CONDE-CESPEDES

patricia.conde-cespedes@isep.fr

September 16th, 2024

Outline

Outline

Introduction

We are in the era of **big data**:

- There are about 1 trillion web pages.
- One hour of video is uploaded to YouTube every second.
- Walmart handles more than 1M transactions per hour.
- ... and so on...

The amount of data increased from 1.2 **zettabyte** (10^{21}) per year in 2010 to 47 **zettabyte** in 2020!

Introduction

We are in the era of **big data**:

- There are about 1 trillion web pages.
- One hour of video is uploaded to YouTube every second.
- Walmart handles more than 1M transactions per hour.
- ... and so on...

The amount of data increased from 1.2 **zettabyte** (10^{21}) per year in 2010 to 47 **zettabyte** in 2020!

... This deluge of data calls for automated methods of data analysis.

What is Machine Learning?

Some definitions:

- *"Machine learning is a set of methods that can automatically detect patterns in data, and then use the uncovered patterns to predict, or to perform other kinds of decision making under uncertainty"* K. Murphy.
- *"Statistical learning refers to a set of tools for modeling and understanding complex datasets."* Hastie and Tibshirani.
- *"Machine learning is essentially a form of applied statistics with increased emphasis on the use of computers to statistically estimate complicated functions and a decreased emphasis on proving confidence intervals around these functions"* Goodfellow et al.
- *"Machine Learning is a young field concerned with developing, analyzing, and applying algorithms for learning from data".* Rose-Hulman Institute of technology

What is Machine learning?

Machine Learning definition

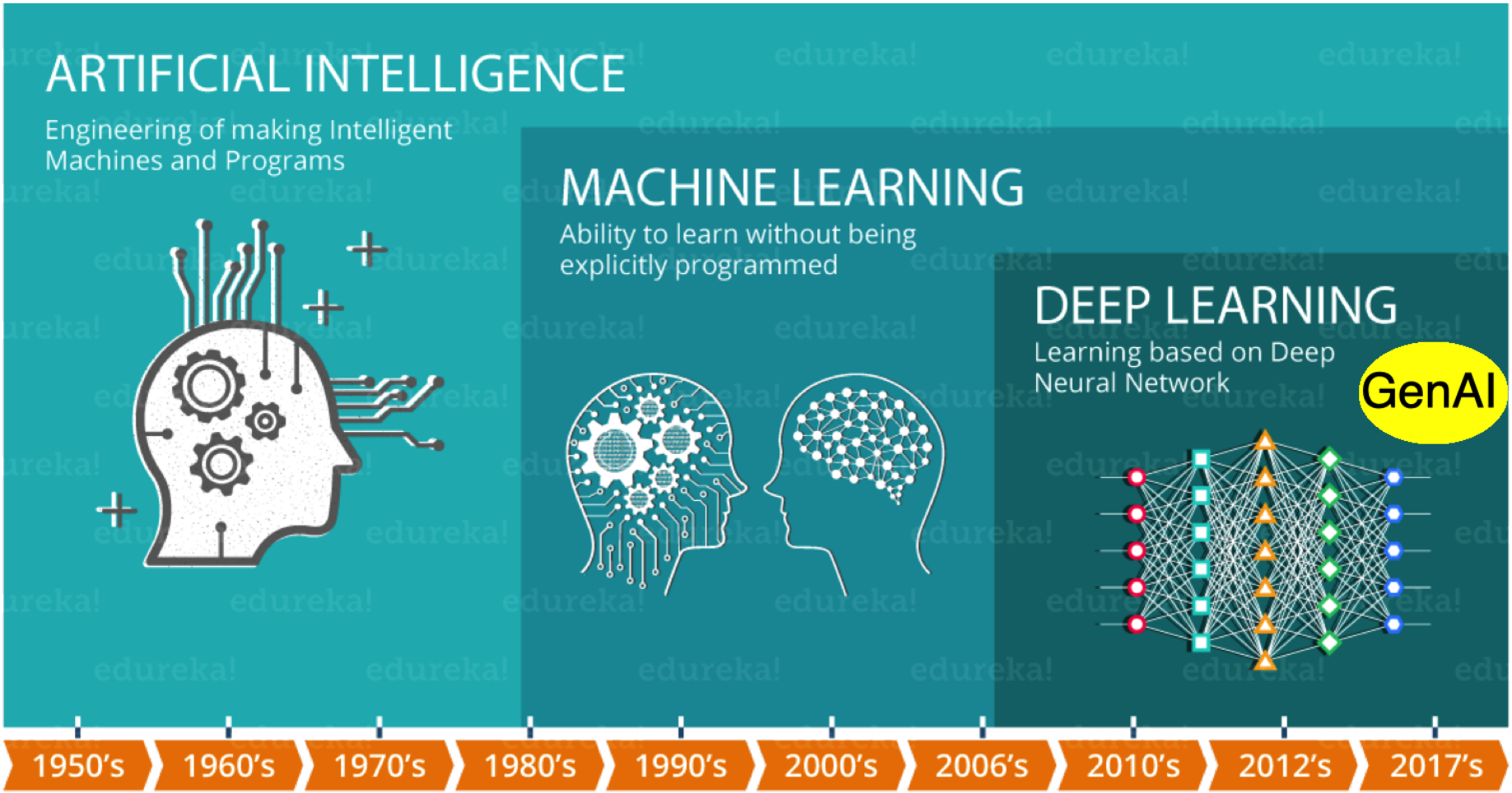
Machine Learning consists in a set of methods and algorithms for analyzing data to automatically extract relevant information for inference or prediction under uncertainty.

Is Machine learning a discipline?

Machine learning is at the cross-road of many disciplines:

- statistics
- applied mathematics
- computer science

AI vs. Machine Learning vs. Deep Learning



Source: `https://www.edureka.co/blog/what-is-deep-learning`

Machine learning applications, example 1

Establish the relationship between salary and demographic variables.

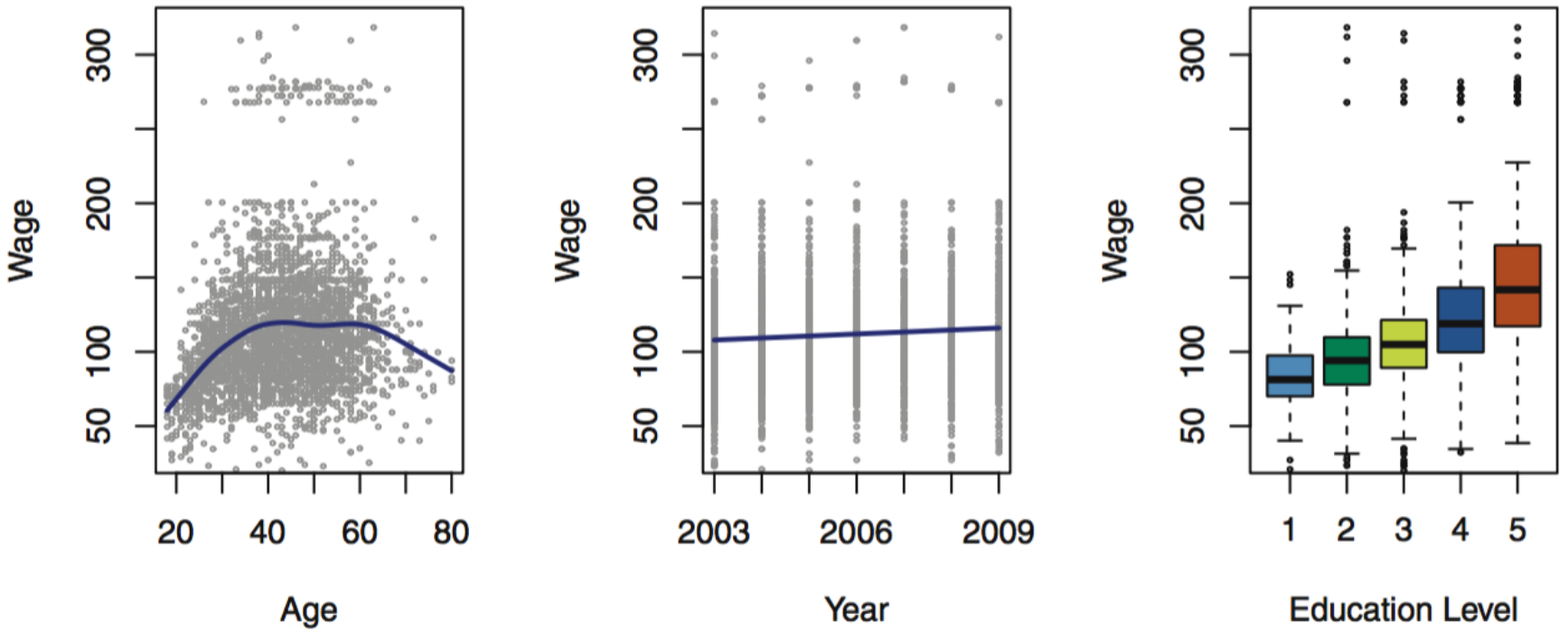


Figure: Income information for men from the central Atlantic region of the US.

We can be interested in predicting the salary of a population depending on the age, education level and some other variables.

Machine learning applications, example 2

Spam Detection. Data from 4601 emails sent to an individual (named George, at HP labs, before 2000). Each email is labeled as spam or email.

- goal: build a customized spam filter.
- input features: relative frequencies of 57 of the **most commonly occurring words** and punctuation marks in these email messages.

	george	you	hp	free	!	edu	remove
spam	0.00	2.26	0.02	0.52	0.51	0.01	0.28
email	1.27	1.27	0.90	0.07	0.11	0.29	0.01

Average percentage of words or characters in an email message equal to the indicated word or character.

Machine learning applications, example 3

Deep Learning in Computer vision (CV)

Image recognition

[Krizhevsky 2012]

Handwriting recognition

[Ciresan et al. 2013]

Object Detection

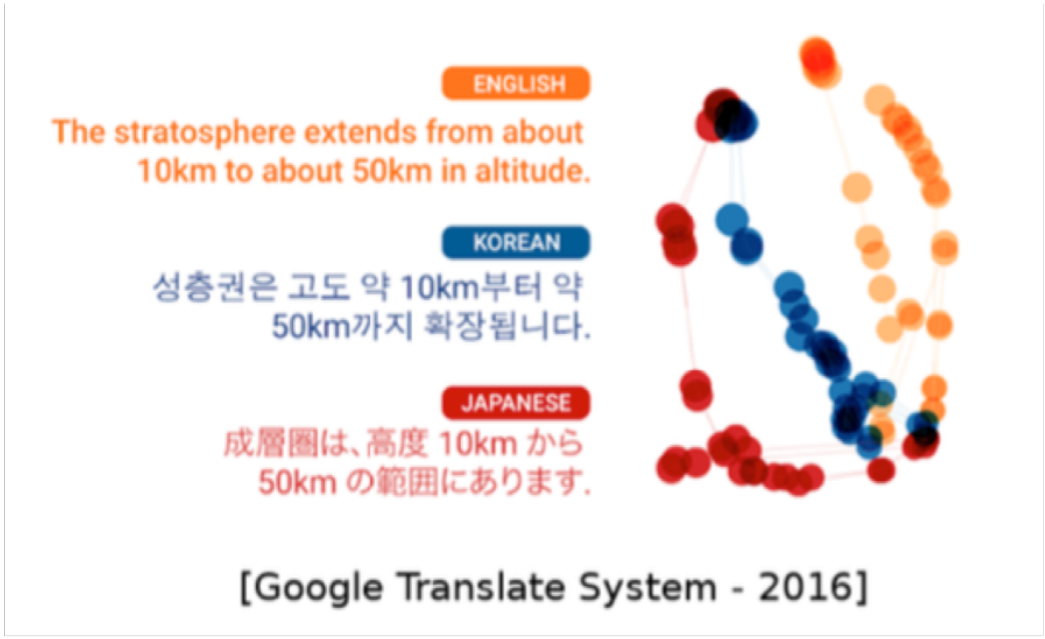
[Faster R-CNN - Ren 2015]

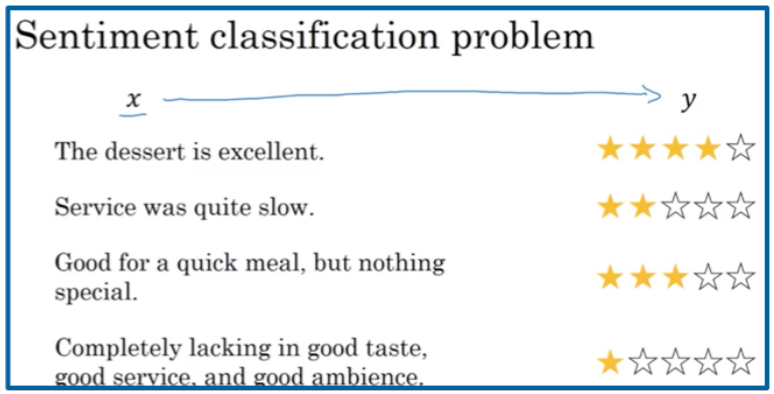
Image Segmentation

[NVIDIA dev blog]

Machine learning applications, example 4

Deep Learning in Natural Language Processiong (NLP)





Outline

Types of variables

Any dataset contains mainly two types of variables:

1. **Quantitative**, also called **numeric**.
   - For example, a person's age, income, the price of a house, etc.

Types of variables

Any dataset contains mainly two types of variables:

1. **Quantitative**, also called **numeric**.
   - For example, a person's age, income, the price of a house, etc.
2. **Qualitative**, also called **categorical** , they take on values in one of different classes/categories or labels.
   They can be ordinal or nominal.
   - Level of education: 1st, 2nd or 3rd year (Ordinal).
   - a single email status: spam/mail, gender: M/F (nominal).

Types of variables

Any dataset contains mainly two types of variables:

1. **Quantitative**, also called **numeric**.
   - For example, a person's age, income, the price of a house, etc.
2. **Qualitative**, also called **categorical** , they take on values in one of different classes/categories or labels.
   They can be ordinal or nominal.
   - Level of education: 1st, 2nd or 3rd year (Ordinal).
   - a single email status: spam/mail, gender: M/F (nominal).

Variables describe units of observation. A unit of **observation** is an object, a person, an email or measurements, etc.

Notation

In the following we will denote:

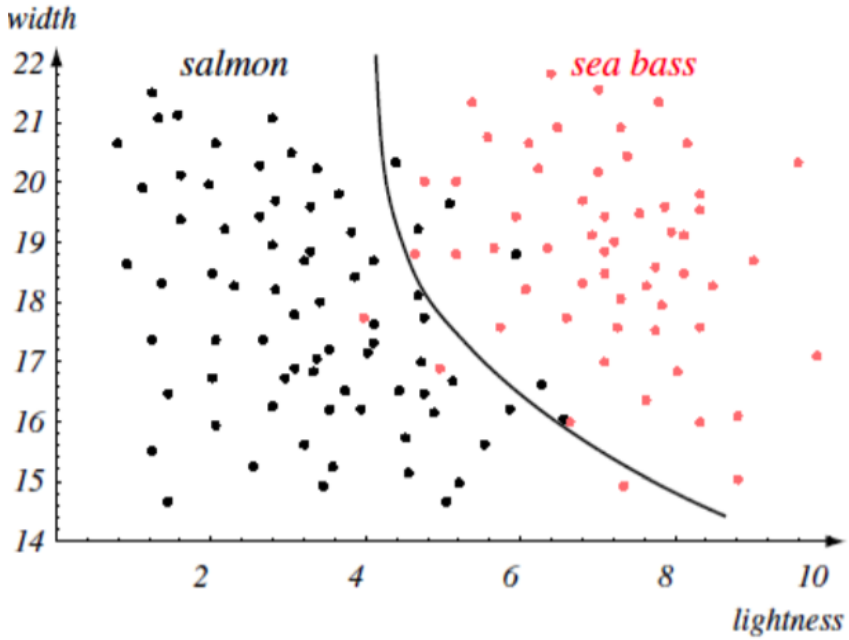
- Y the variable we want to predict (if any), also called *output*, *response*, *target*
- X the predictor, also called *input*, *feature*, *explanatory variable*. If there is more than one predictor, let us say $p > 1$, we will use subscripts and denote each one X_j.

Some examples:

- Y : the income, sales, spam/mail.
- X : the age, education level, gender, the presence/absence of a word in an email

Machine Learning tasks : classification and regression

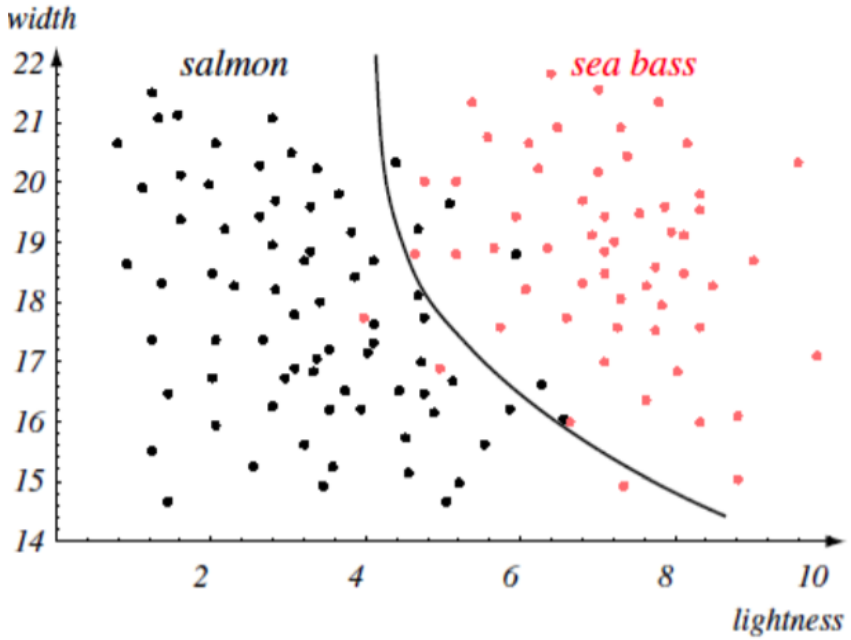
Classification:



Predict fish type depending on the width and weight

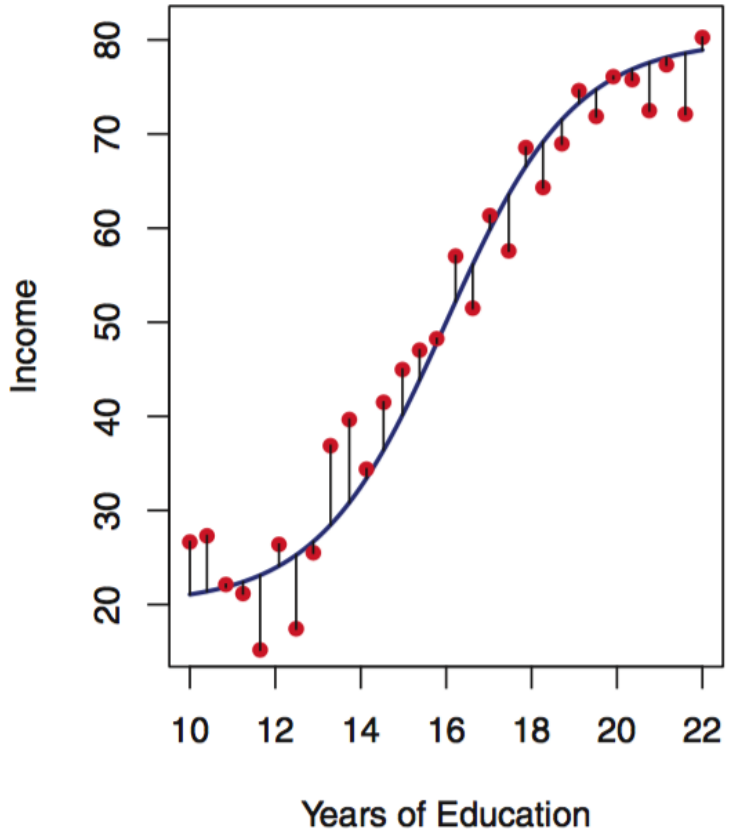
Machine Learning tasks : classification and regression

Classification:



Predict fish type depending on the width and weight

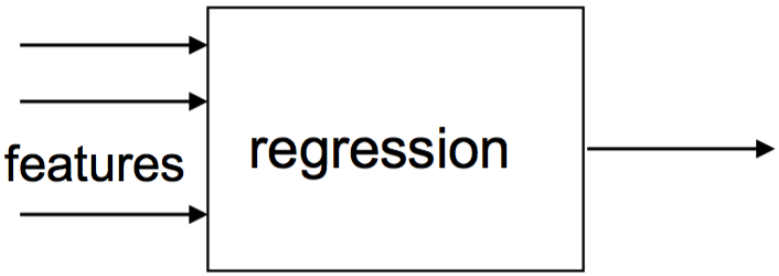
Regression:



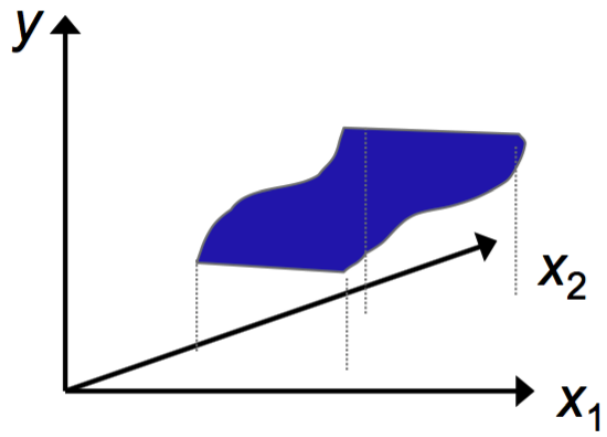
Predict the income based on the years of education

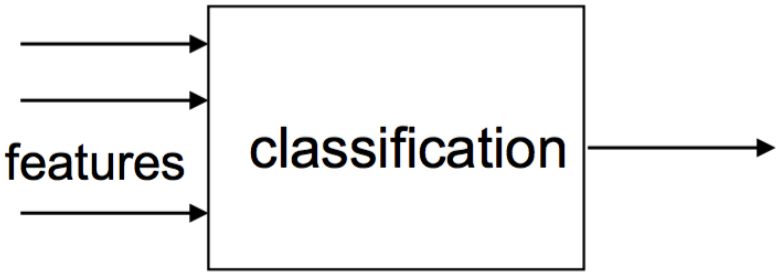
Machine Learning tasks : Classification vs. Regression

The difference is the type of predicted variable

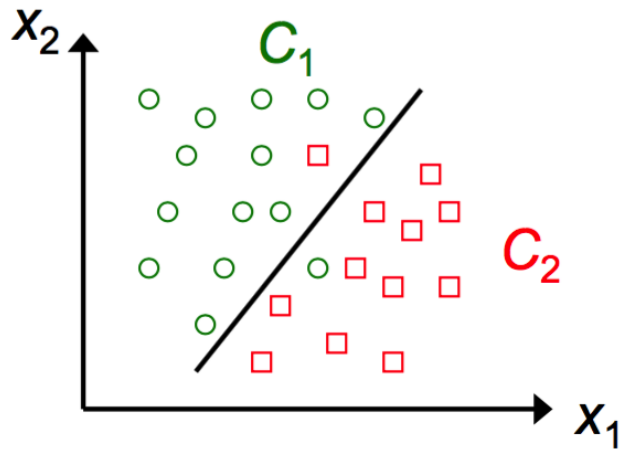


Quantitative variable





Qualitative variable

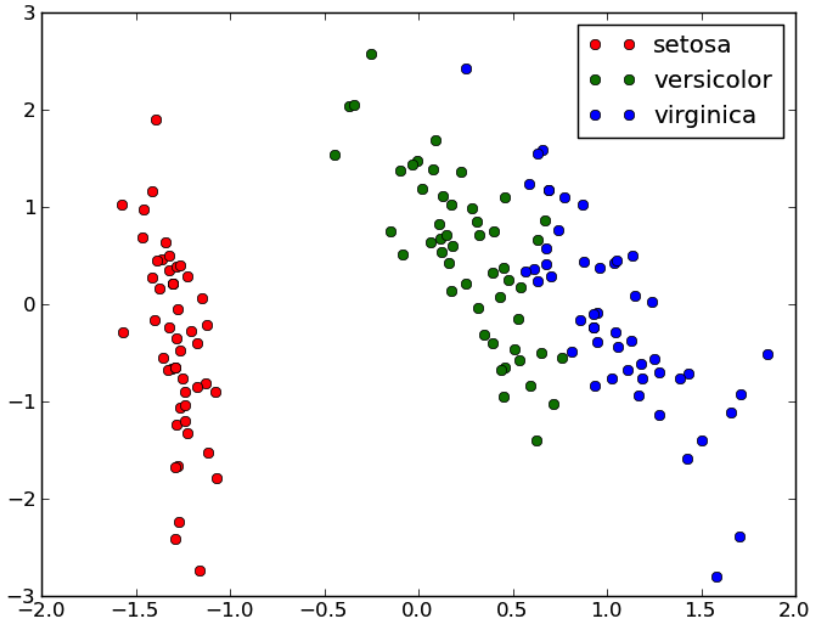


Machine Learning tasks : Clustering

Intuition: Given a dataset of objects described by some features we want to determine groups or clusters of *similar* objects, this is **clustering**.

Machine Learning tasks : Clustering

Intuition: Given a dataset of objects described by some features we want to determine groups or clusters of *similar* objects, this is **clustering**.

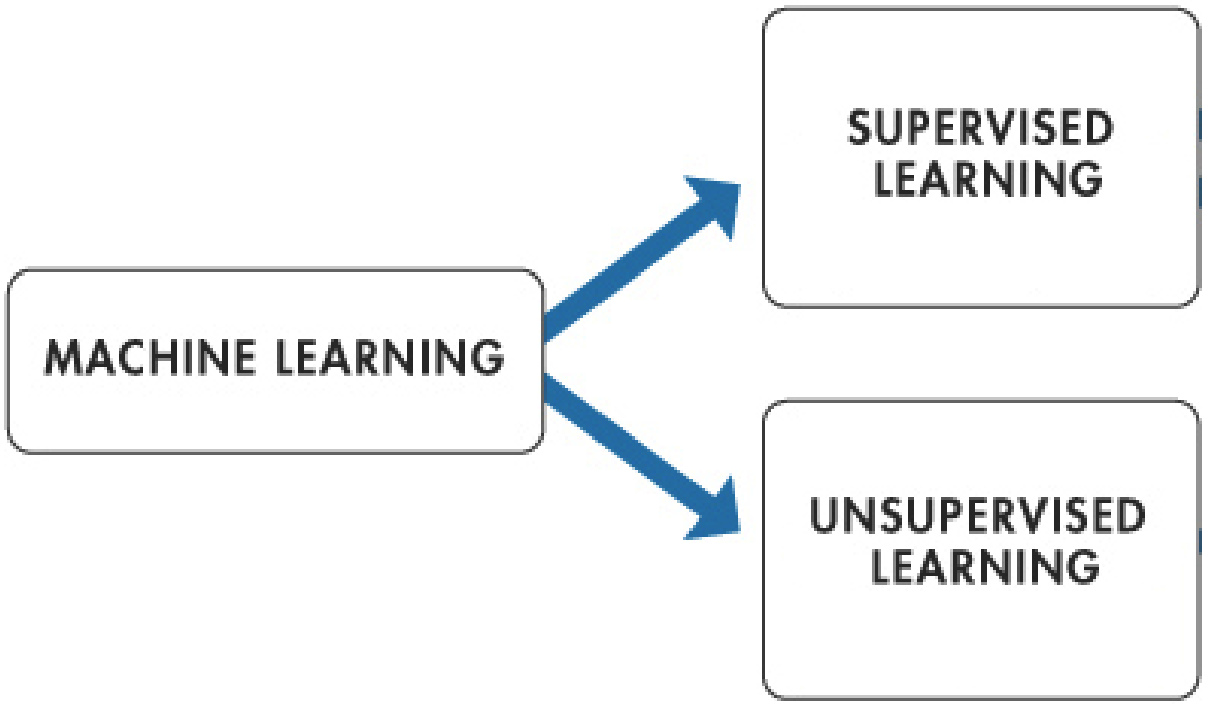


The Iris flower data set studied by Fisher (1936) : 50 samples of 3 species of Iris (setosa, virginica and versicolor) described by the length and the width of the sepals and petals.

We are not interested in predicting a particular output variable !

Typical branches of Machine Learning

Machine Learning algorithms are divided into two big branches.



source: `https://fr.mathworks.com/help/stats/machine-learning-in-matlab.html`

Supervised Learning vs. Unsupervised learning

Supervised Learning

Given a dataset, called the **training set**, consisting in N observations of p features $\mathbf{X}$ and a target variable Y, the purpose is to accurately predict Y for unseen observations, called **test** observations.

- Useful to understand how each input affects the outcome.

Unsupervised Learning

Given a dataset of features variables $\mathbf{X}$ the objective is to learn relationships and structure from data or to find groups of objects that behave similarly.

- There is **no** target variable, so no prediction.
- Data visualization and clustering are unsupervised learning techniques.
- Useful as a pre-processing or exploratory step for supervised learning.

Supervised Learning and unsupervised learning examples

For supervised learning :

- Predict the *iris* type in the Iris dataset.

Supervised Learning and unsupervised learning examples

For supervised learning :

- Predict the *iris* type in the Iris dataset.
- Spam detection.

Supervised Learning and unsupervised learning examples

For supervised learning :

- Predict the *iris* type in the Iris dataset.
- Spam detection.
- Predict the salary based on demographic information

For unsupervised learning :

- In the Iris dataset, detect clusters of flowers that share similar morphologic characteristics.

Supervised Learning and unsupervised learning examples

For supervised learning :

- Predict the *iris* type in the Iris dataset.
- Spam detection.
- Predict the salary based on demographic information

For unsupervised learning :

- In the Iris dataset, detect clusters of flowers that share similar morphologic characteristics.
- Given demographic data, findin clusters of people who belong to the same social classes.

Supervised Learning and unsupervised learning examples

For supervised learning :

- Predict the *iris* type in the Iris dataset.
- Spam detection.
- Predict the salary based on demographic information

For unsupervised learning :

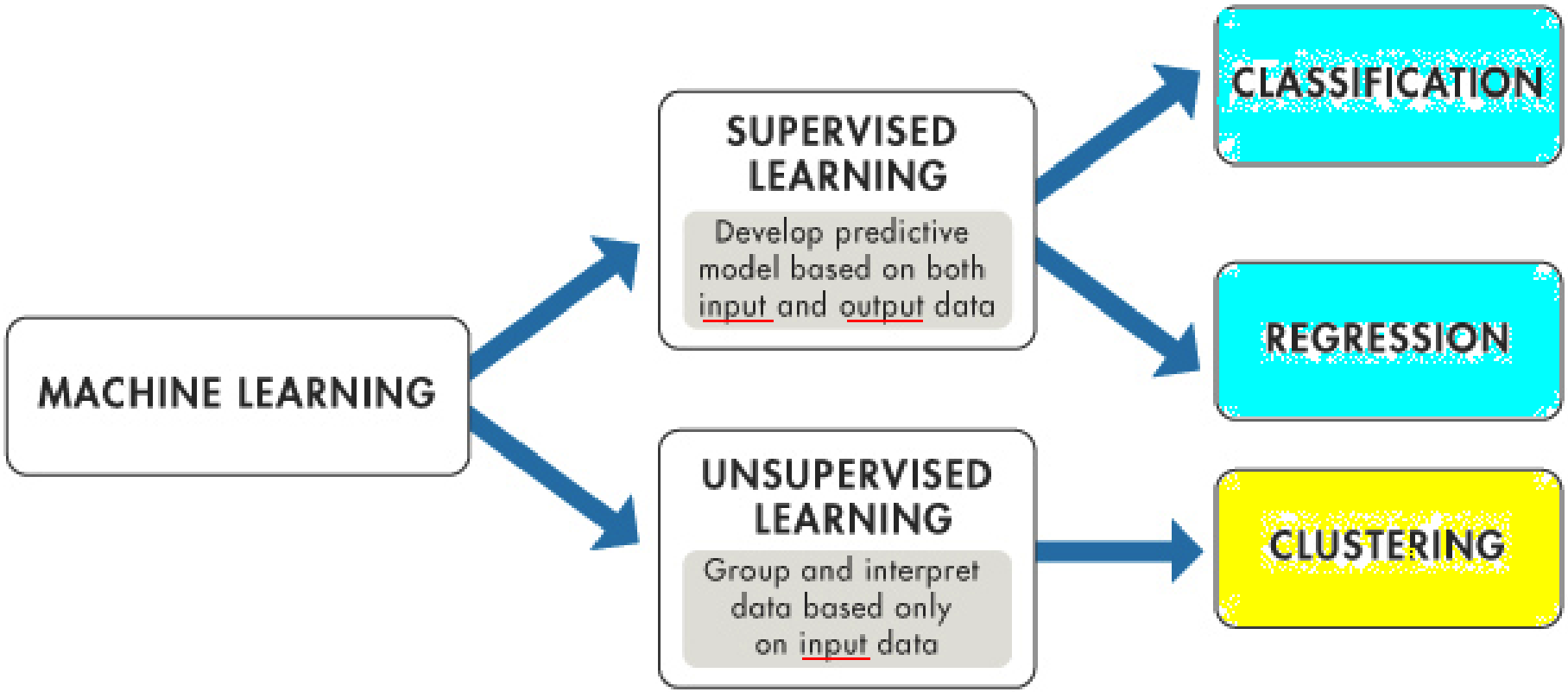
- In the Iris dataset, detect clusters of flowers that share similar morphologic characteristics.
- Given demographic data, findin clusters of people who belong to the same social classes.
- Given the dataset of emails, detect groups of messages that treat related topics.

Supervised vs. **Unsupervised** learning

- Supervised = It is like a teacher that gives classes (supervision),
  - Inputs and outputs.
- Unsupervised = It is more an exploratory analysis.

source: http://dataaspirant.com/2014/09/19/supervised-and-unsupervised-learning/

Supervised vs. Unsupervised



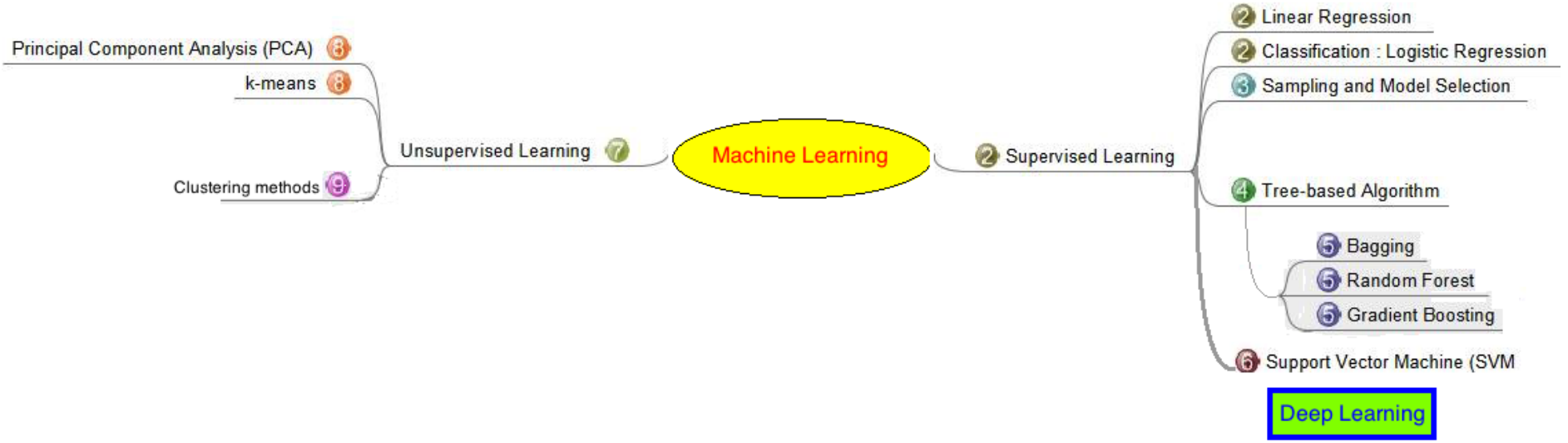
source: https://fr.mathworks.com/help/stats/machine-learning-in-matlab.html

Other Machine Learning methods

Sometimes the question of whether an analysis should be considered supervised or unsupervised is not clear enough.

- **reinforcement learning,** an agent interacts with a dynamic environment in which it must perform a certain goal (such as driving a vehicle or playing a game against an opponent). The program is provided feedback in terms of rewards and punishments. (For example, consider how a baby learns to walk, chess game, AlphaGo game).
- **Semi-supervised learning:** The output variable is know only for a subset of observations. Such a scenario can arise if the independent variables are measured relatively cheaply but the corresponding responses are much more expensive to collect.
- **Transfer learning:** trained models used to solve one problem can be used for solving a different but related problem. For instance, models trained to recognize cars could apply when trying to recognize trucks.

Machine Learning scheme

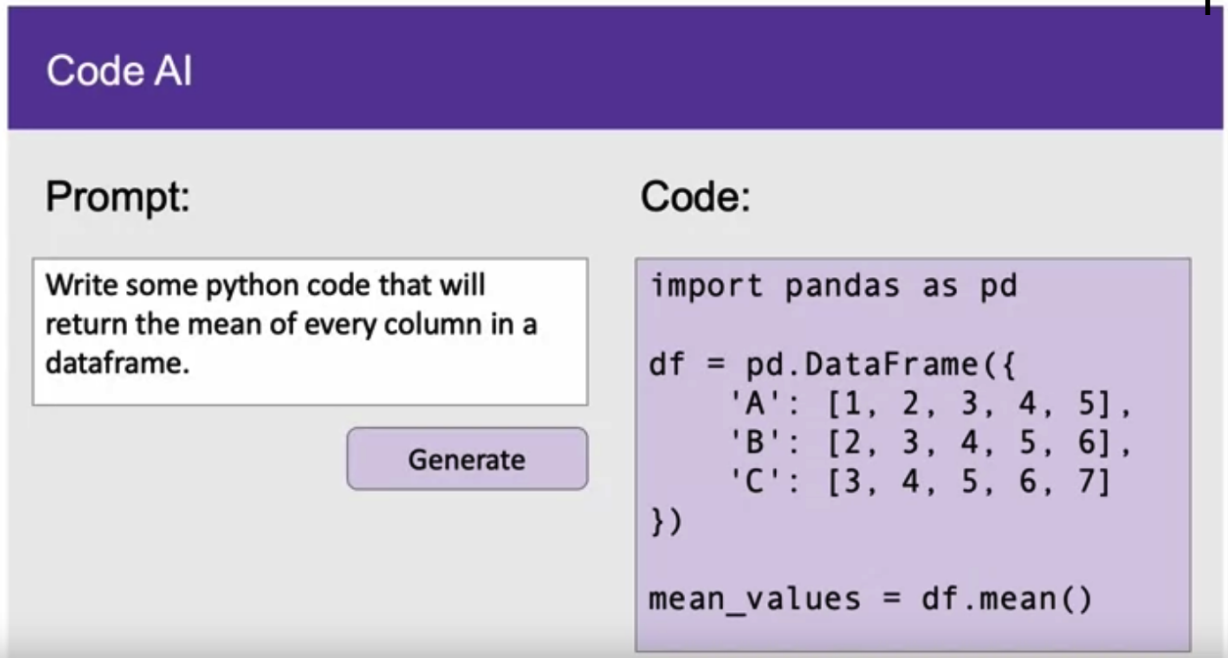


Generative AI (Since 2020s) (1/2)

Generative AI definition

A set of Machine Learning models able to create content (image, text, music or speech) that mimics or approximates human ability.

- LLMs (Large Language Models) are able to take human written instructions and perform tasks such a human would do.

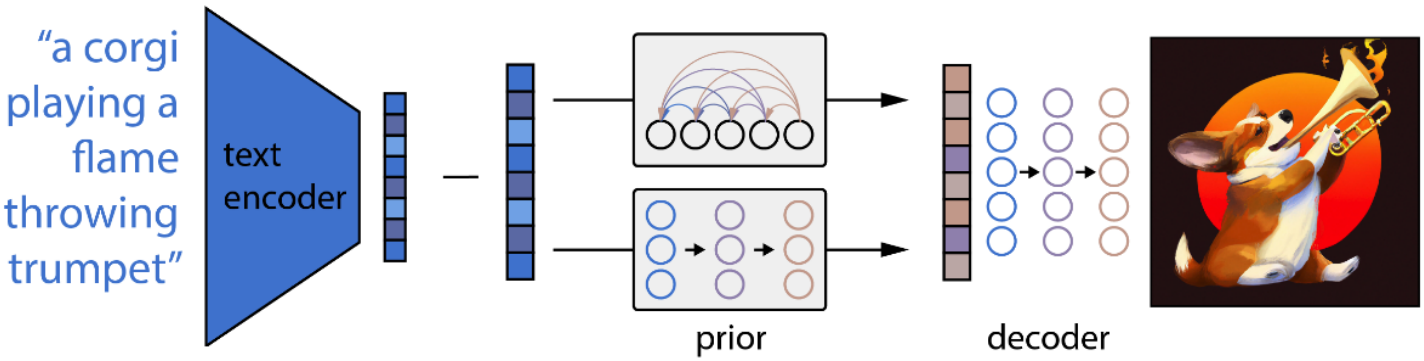


Foundation models :

- ChatGPT (for Chat Generative Pre-trained Transformer)
- LLaMA (Large Language Model Meta AI)
- BLOOMz, PaLM, FLAN.

Generative AI (Since 2020s) (2/2)

- Text to image generation models : take human-written text description as input and produces an image matching that description.
- OpenAI models DALLE-2, CLIP, GLIDE.



Source: Image modified from *Ramesh et Al. (2022)* `https://cdn.openai.com/papers/dall-e-2.pdf`.

"a hedgehog using a calculator"

"a corgi wearing a red bowtie and a purple party hat"

"robots meditating in a vipassana retreat"

"a fall landscape with a small cottage next to a lake"

Source: *Nichol et Al. (2022)* `https://proceedings.mlr.press/v162/nichol22a/nichol22a.pdf`.

Outline

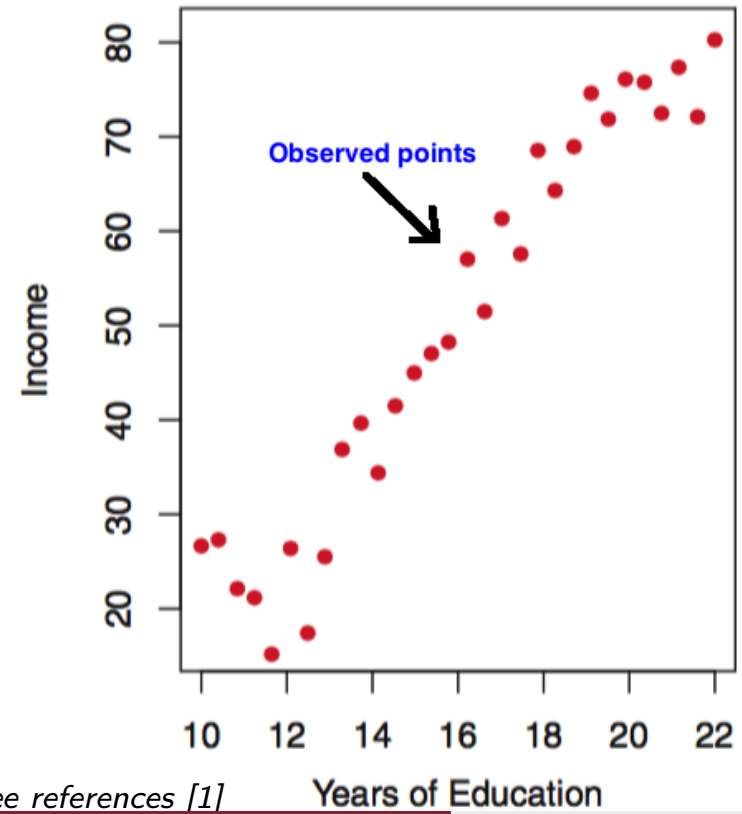
Relationship between Y and X:

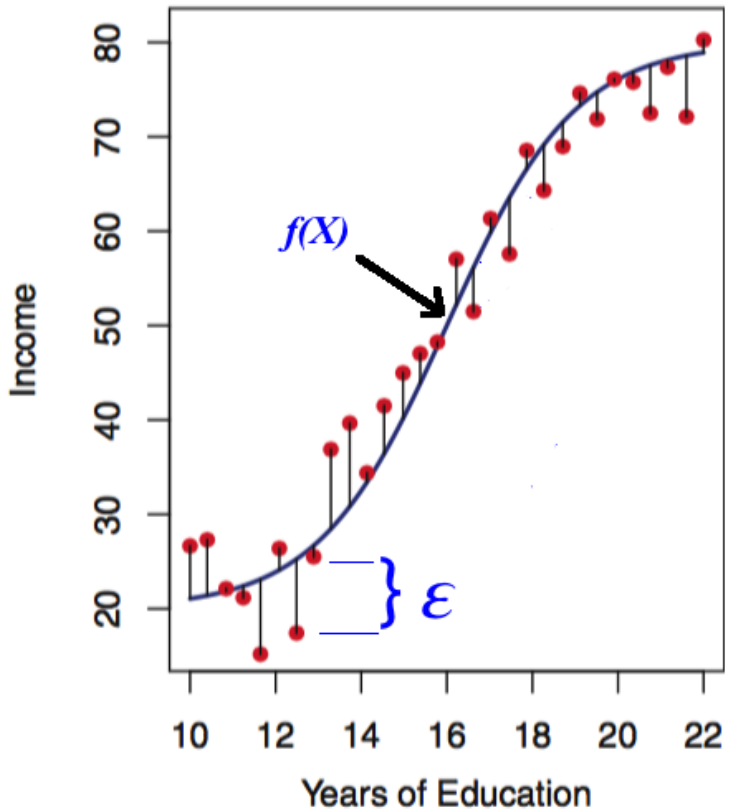
We assume the relationship between Y and $\mathbf{X} = (X_1, X_2, ..., X_p)$ is the following:

$$Y = f(X) + \epsilon \tag{1}$$

where:

- f is some fixed but unknown function of $X_1, ..., X_p$, and
- ϵ is a random error term.





source : *see references [1]*

More about the relationship between Y and X

About $f(X)$:

- f represents the information that X provides about Y.
- $f(x) = E(Y/X = x)$ represents the **expected value** of Y given X.

$\epsilon = Y - f(x)$ is called the **irreducible error**

- ϵ is supposed independent of X and has mean zero.

More about the relationship between Y and X

About $f(X)$:

- f represents the information that X provides about Y.
- $f(x) = E(Y/X = x)$ represents the **expected value** of Y given X.

$\epsilon = Y - f(x)$ is called the **irreducible error**

- ϵ is supposed independent of X and has mean zero.
- *irreducible* : even if $f(x)$ were known, there can be still errors in prediction, since at each $X = x$ there is typically a distribution of possible Y values.

More about the relationship between Y and X

About $f(X)$:

- f represents the information that X provides about Y.
- $f(x) = E(Y/X = x)$ represents the **expected value** of Y given X.

$\epsilon = Y - f(x)$ is called the **irreducible error**

- ϵ is supposed independent of X and has mean zero.
- *irreducible* : even if $f(x)$ were known, there can be still errors in prediction, since at each $X = x$ there is typically a distribution of possible Y values.
- ϵ captures measurement errors and other discrepancies.

Estimator of f : $\hat{f}$

f is unknown, its estimation is based on the observed points (train set).

Let us denote $\hat{f}$ an estimator of f and $\hat{Y} = \hat{f}(X)$, then :

$$E[Y - \hat{Y}]^2 = E[f(X) + \epsilon - \hat{Y}]^2 = \underbrace{E[f(X) - \hat{f}(X)]^2}_{\text{Reducible}} + \underbrace{V[\epsilon]}_{\text{Irreducible}} + c \quad (2)$$

where:

- $E[Y - \hat{Y}]^2$: expected value of the squared difference between the predicted and actual value of Y,
- $V(\epsilon)$ variance associated of the error term ϵ.
- c is a term considered negligent.

In the following, we will focus on minimizing the reducible error.

Outline

Overfitting

- Overfitting is learning (a finite number of) train data so well, that the model is not useful anymore for *new* data (test set).

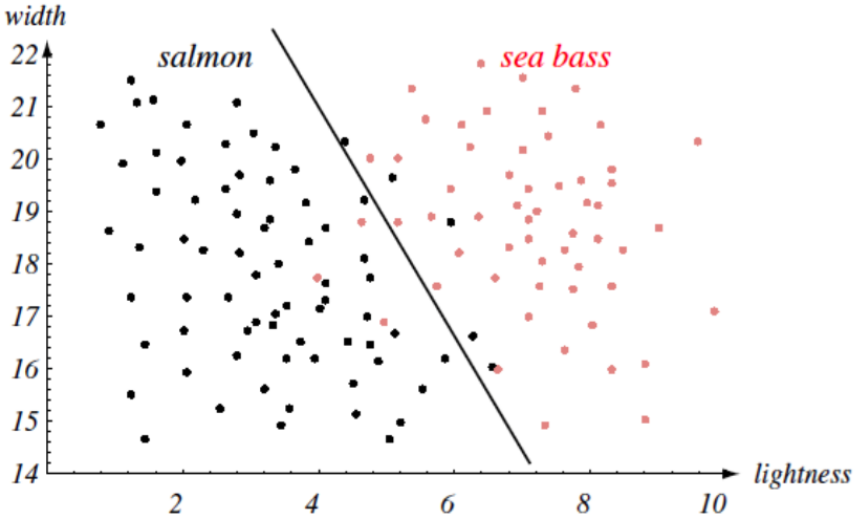
Overfitting

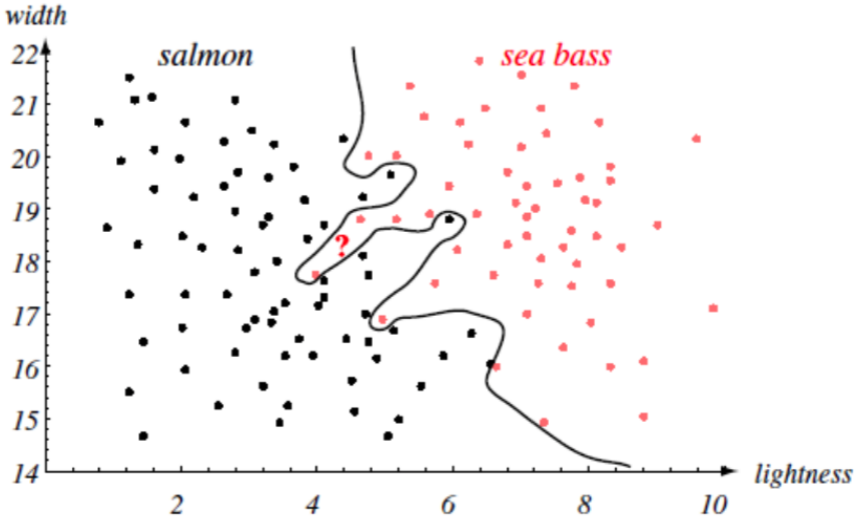
- Overfitting is learning (a finite number of) train data so well, that the model is not useful anymore for *new* data (test set).
- Overfitting the data implies follow the errors, or noise, too closely.

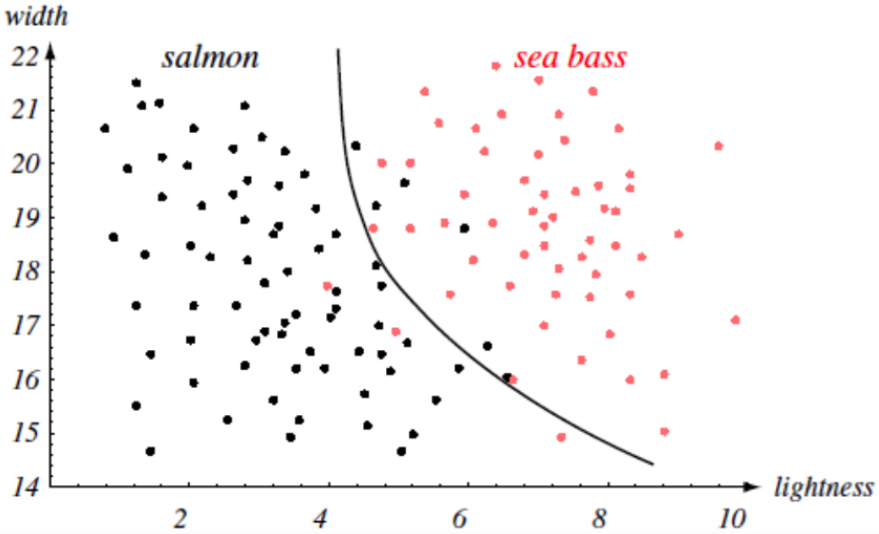
Overfitting

- Overfitting is learning (a finite number of) train data so well, that the model is not useful anymore for *new* data (test set).
- Overfitting the data implies follow the errors, or noise, too closely.
- It is one of the main important concerns in machine learning!

Overfitting in classification 1/2



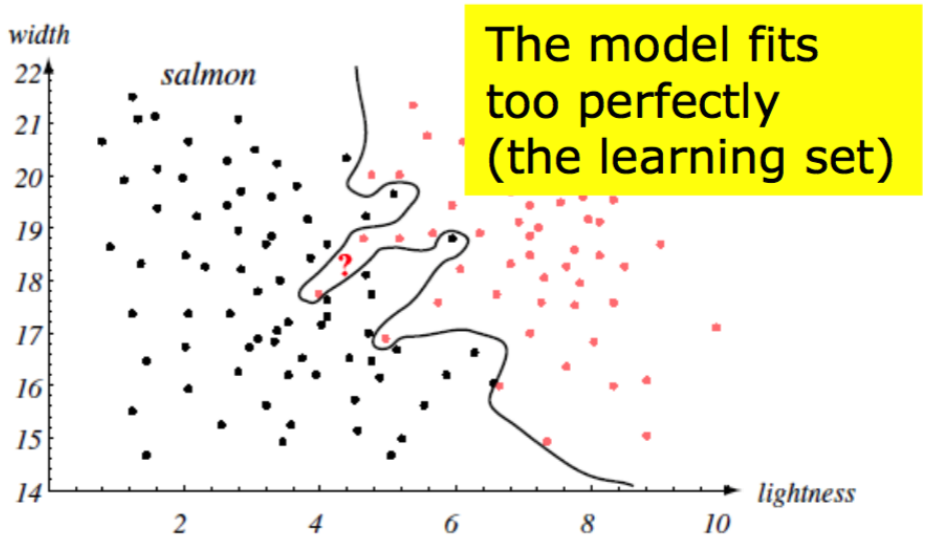


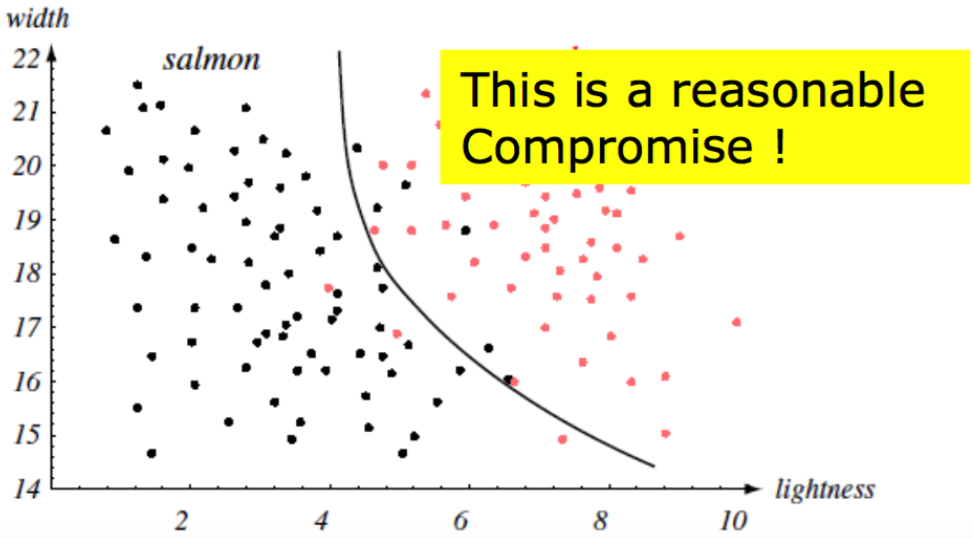


- Which of the 3 models is « best »?

From: Duda et al., Pattern Classification, 2nd ed., Wiley, 2001

Overfitting in classification 2/2





- Which of the 3 models is « best »?

From: Duda et al., Pattern Classification, 2nd ed., Wiley, 2001

Overfitting in regression (1/2)

Let us suppose que $f(X)$ is a polynomial function of X:

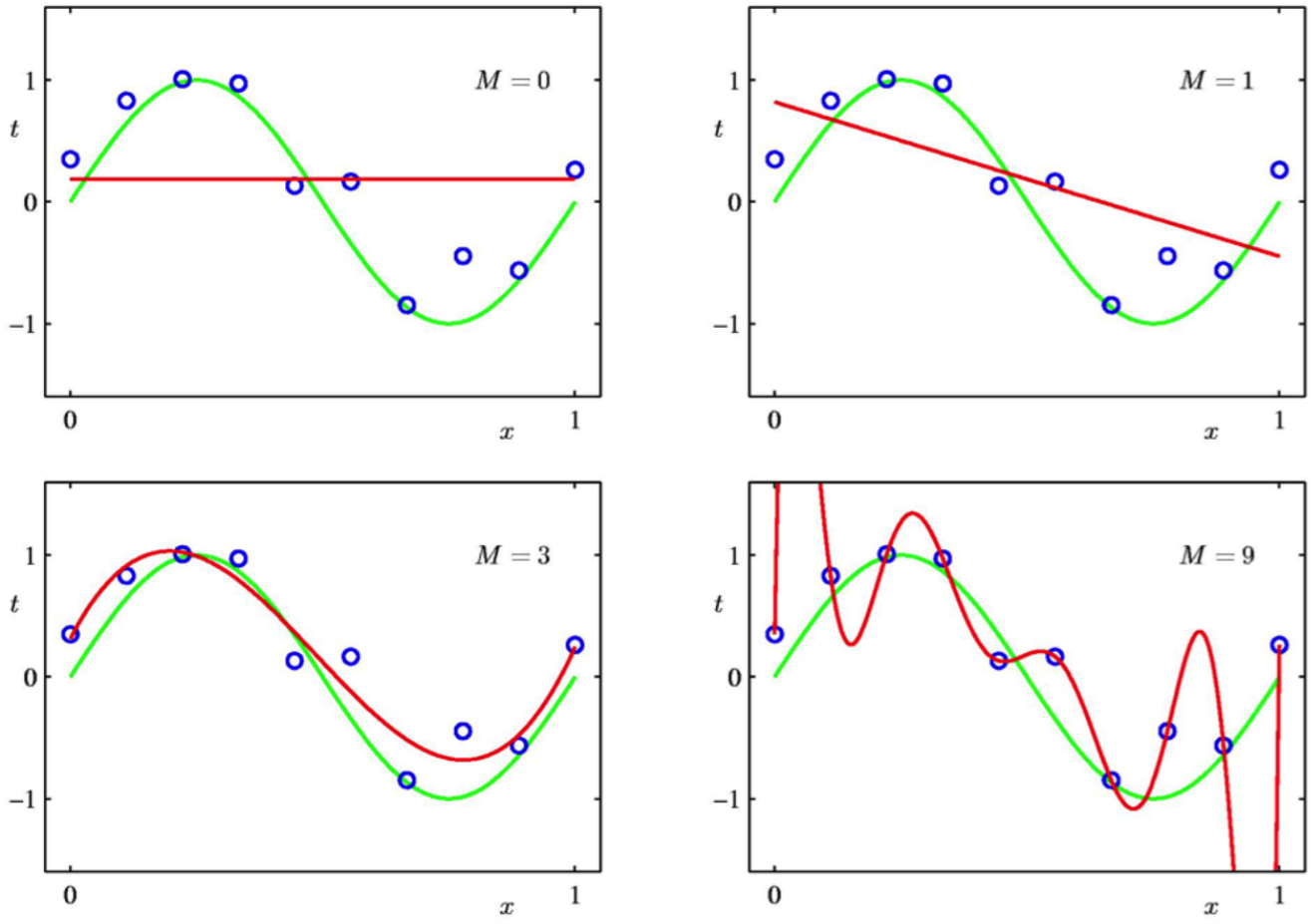
$$f(X) = \beta_0 + \beta_1 X + \beta_2 X^2 + ... + \beta_M X^M \tag{3}$$

where M is the order of the polynomial.

- We estimate the parameters by fitting the model to training data.
- $M + 1$ parameters must be estimated. The number of parameters represents the **degree of freedom**, thus **flexibility** or **complexity** of the model.

Overfitting in regression (2/2)

The following example is about regression with polynomials of order M.



Models with *M=0 and* M=1 are poor,

model with M=9 overfits,

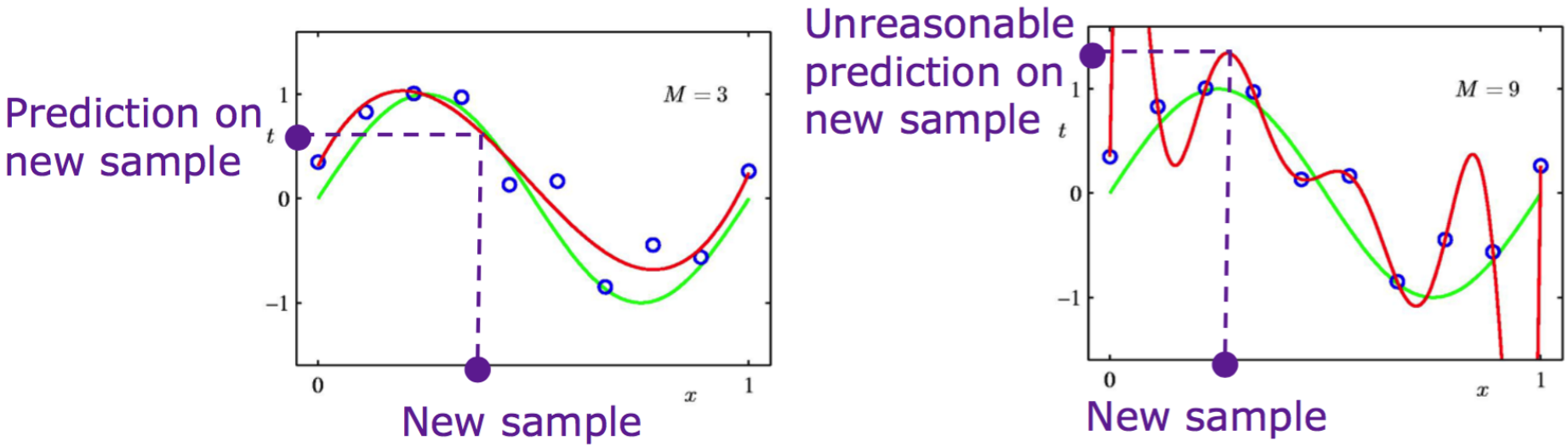
model with M=3 is a good compromise

From: C. Bishop, Pattern Recognition and Machine Learning, Springer, 2006.

Why overfitting is not a good idea?

Why fitting "*as well as possible*" is not a good idea?

- Because we do not care about having a model that reproduces the output on known examples (= training set)
- The real **goal** is to get a model that performs well on new samples!



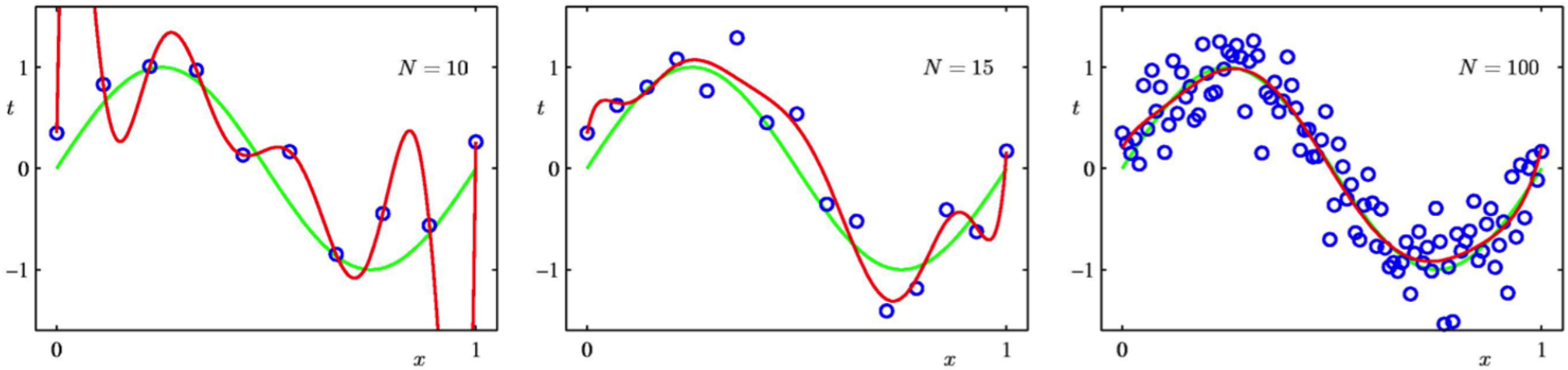
Adapted from: C. Bishop, Pattern Recognition and Machine Learning, Springer, 2006.

Blue dots represent training data and purple dot represents test data.

How to deal with overfitting ?

The risk of overfitting...

- increases with the **complexity** (also called *flexibility*) of the model
- decreases with the size of the learning set
  - example, Polynomial model of order $M = 9$:

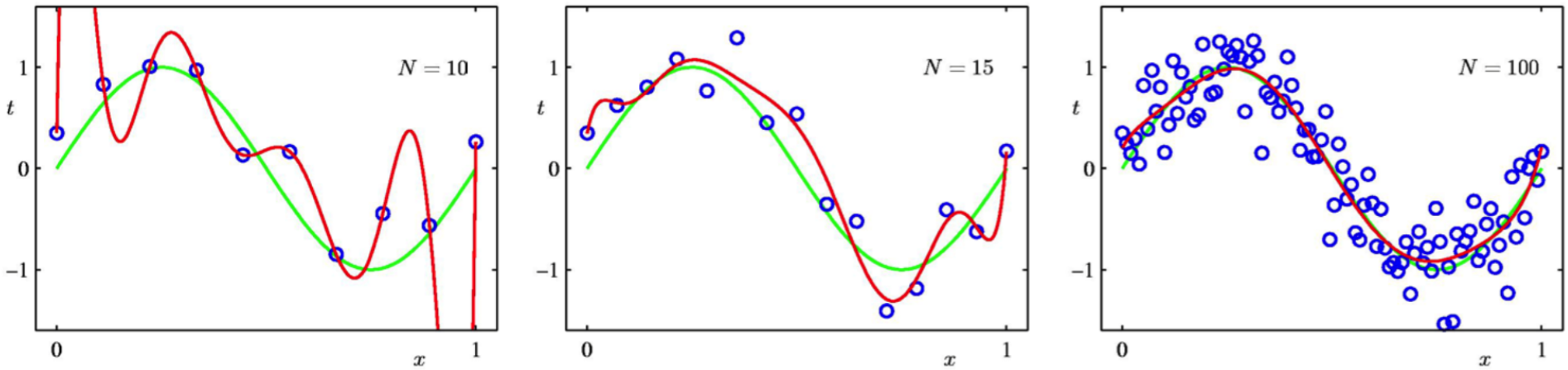


From: C. Bishop, Pattern Recognition and Machine Learning, Springer, 2006.

How to deal with overfitting ?

The risk of overfitting...

- increases with the **complexity** (also called *flexibility*) of the model
- decreases with the size of the learning set
  - example, Polynomial model of order $M = 9$:



From: C. Bishop, Pattern Recognition and Machine Learning, Springer, 2006.

Now, we will see how to avoid overfitting!

Measuring the quality of fit, the **training MSE**

Suppose we fit a model $\hat{f}(x)$ to some training data $\{x_i, y_i\}\, \forall i \in [1, .., n]$, and we want to know how well it performs.

- To measure the quality of fit we can calculate the *mean squared error* (**MSE**):

$$MSE_{Tr} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{f}(x_i))^2, \tag{4}$$

  called the *training MSE*.

The *MSE* measures how close the predicted responses are to the true responses.
In some practical situations we calculate the root MSE (**RMSE**) instead.

Measuring the quality of fit, the **test MSE**

Challenge: the estimator performs well on previously unseen inputs (not just those on which our model was trained).

Measuring the quality of fit, the **test MSE**

Challenge: the estimator performs well on previously unseen inputs (not just those on which our model was trained).

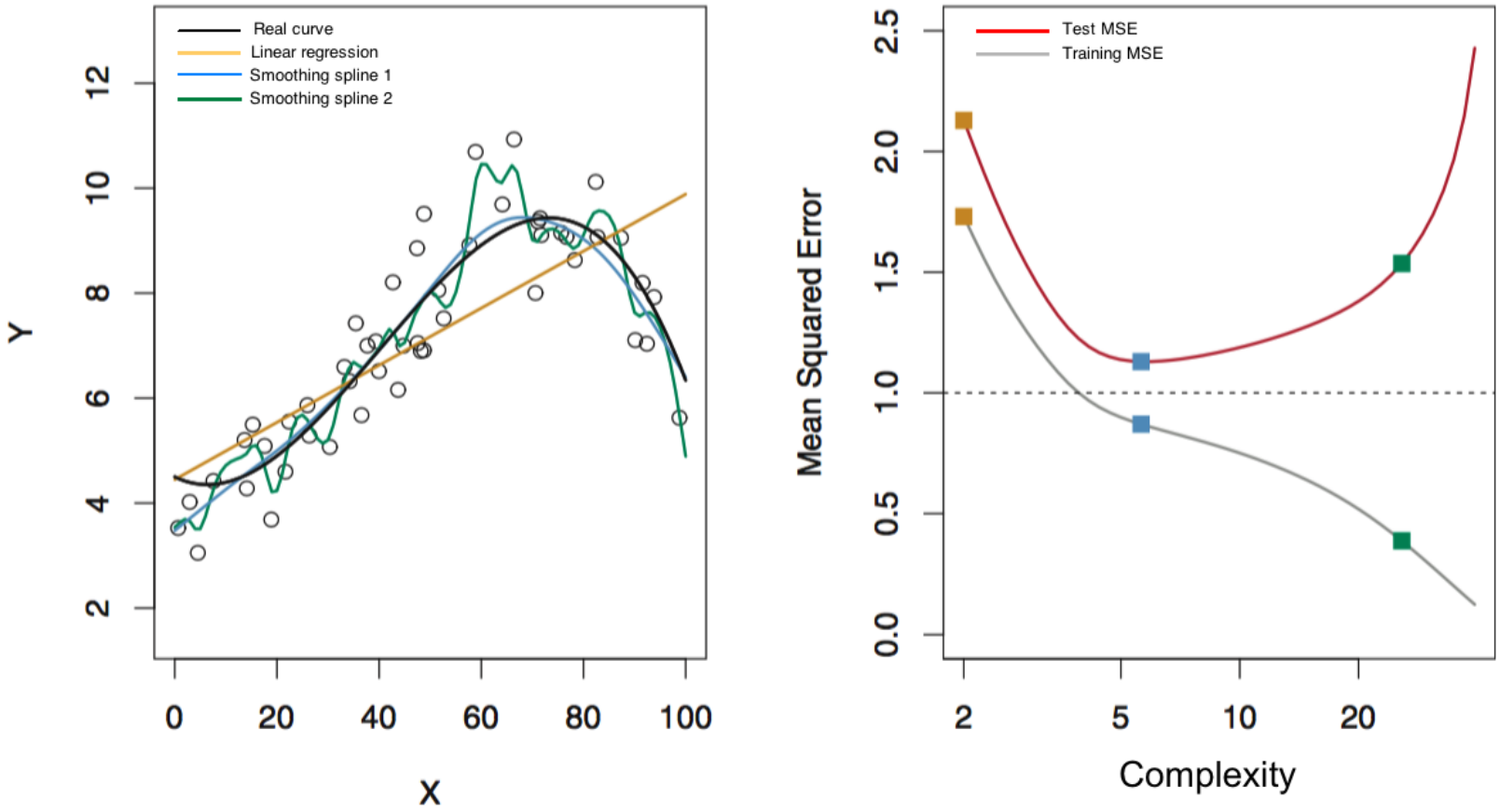
Given a test set $\{x_i, y_i\} \forall i \in [1, .., m]$ we define the *test* MSE:

$$MSE_{Te} = \frac{1}{m} \sum_{i=1}^{m} (y_i - \hat{f}(x_i))^2, \tag{5}$$

We will select the model for which the average of the test MSE is as small as possible.

Training and test MSE versus complexity, example 1

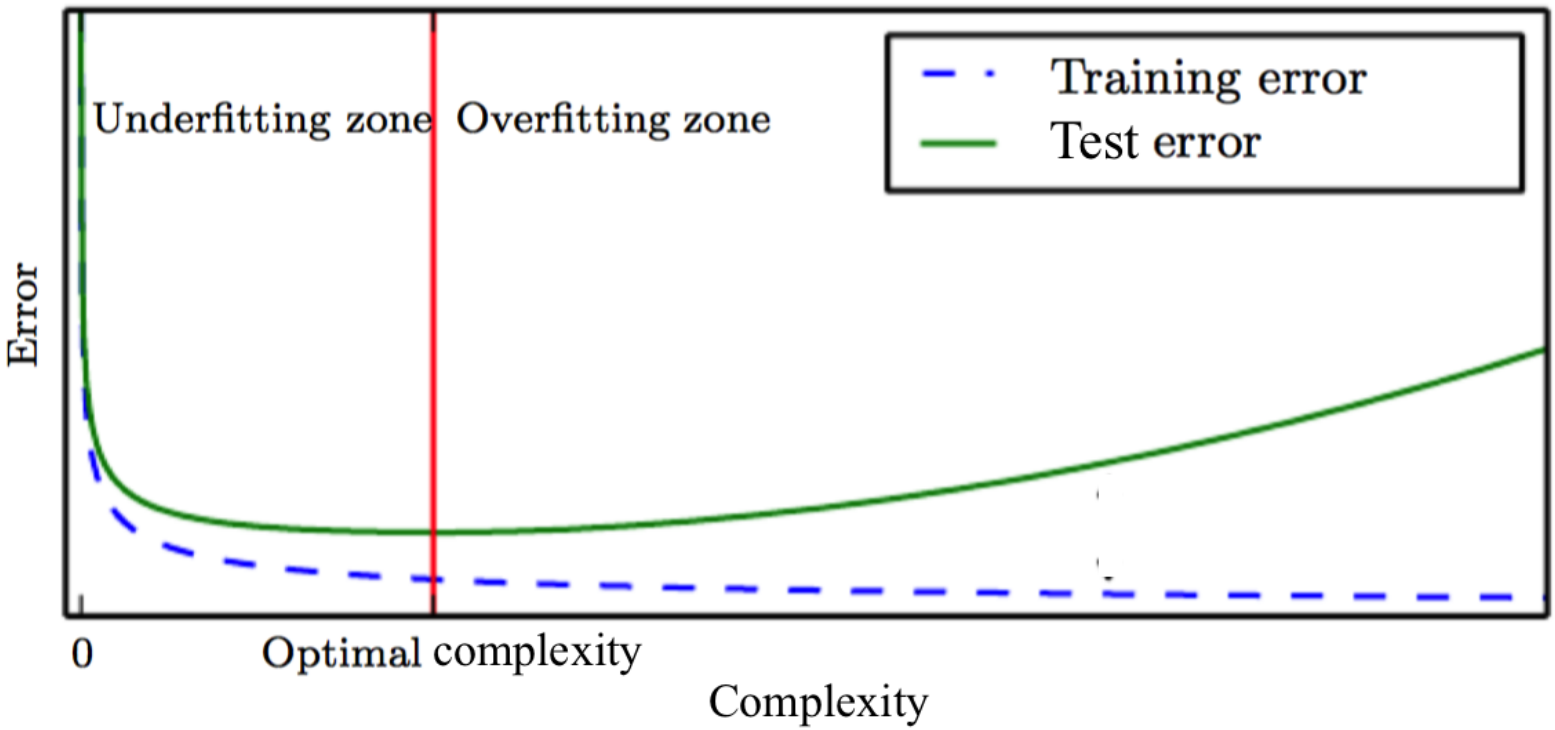
As the complexity of the model increases, the *training MSE* monotone decreases whereas the *test MSE* has a *U-shape*.



Orange, blue and green curves (squares) on the left (right) panel correspond to fits of f of increasing complexity.

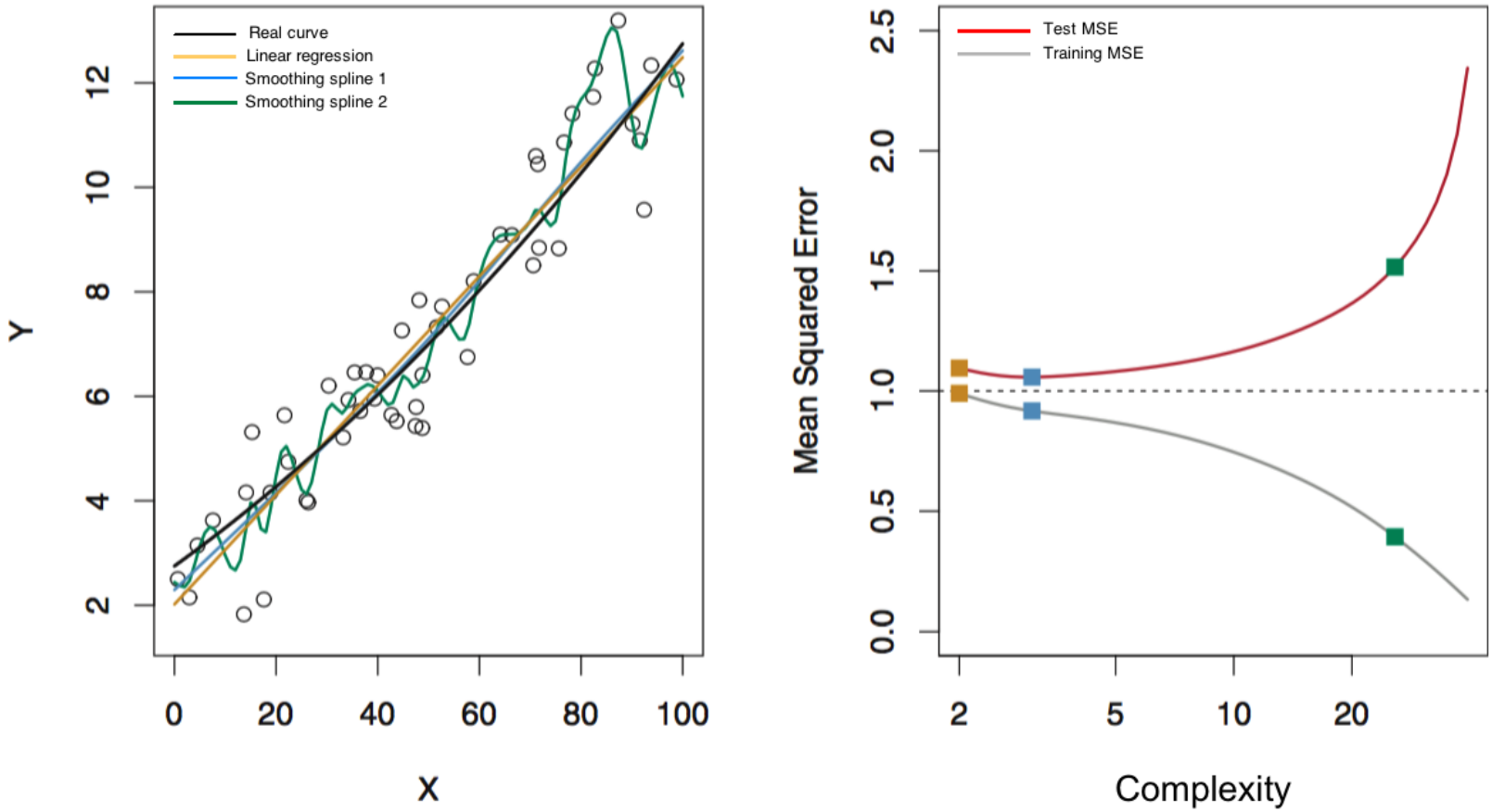
Training and test MSE and overfitting

- Usually if a method yields a small training MSE, the method *overfits* the data.



Training and test MSE versus complexity, example 2

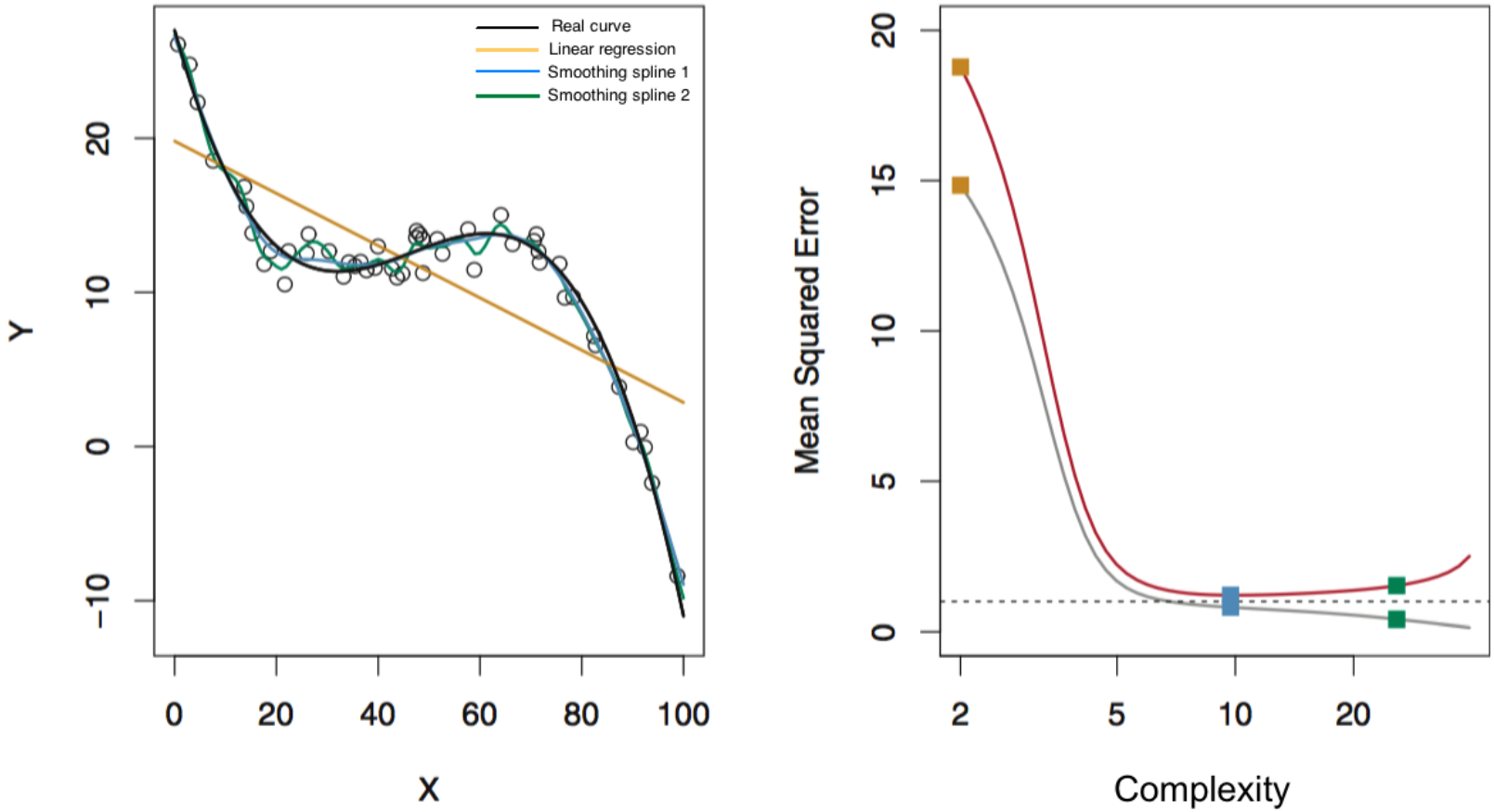
In this example the real curve is close to linear



Orange, blue and green curves (squares) on the left (right) panel correspond to fits of f of increasing complexity.

Training and test MSE versus complexity, example 3

In this example the real f is highly non-linear and the noise is low, so the more flexible fits do the best.



Orange, blue and green curves (squares) on the left (right) panel correspond to fits of f of increasing complexity.

Bias-Variance trade-off

The U-shape observed in the test MSE curves is the result of **two** competing properties of statistical learning methods.

Bias-Variance trade-off

The U-shape observed in the test MSE curves is the result of **two** competing properties of statistical learning methods.

Let $\hat{f}(x)$ be a fitted model to the training data. If the true model is $Y = f(X) + \epsilon$ (with $f(x) = E(Y|X = x)$) for a test observation (x_0, y_0) we have :

$$E(y_0 - \hat{f}(x_0))^2 = \underbrace{V(\hat{f}(x_0)) + [Bias(\hat{f}(x_0))]^2}_{\text{Reducible}} + V(\epsilon). \quad (6)$$

where:

- $E(y_0 - f(x_0))^2$ denotes the *expected test MSE*.
- $Bias(\hat{f}(x_0)) = E[\hat{f}(x_0)] - f(x_0)$.

Bias-Variance trade-off

The U-shape observed in the test MSE curves is the result of **two** competing properties of statistical learning methods.

Let $\hat{f}(x)$ be a fitted model to the training data. If the true model is $Y = f(X) + \epsilon$ (with $f(x) = E(Y|X = x)$) for a test observation (x_0, y_0) we have :

$$E(y_0 - \hat{f}(x_0))^2 = \underbrace{V(\hat{f}(x_0)) + [Bias(\hat{f}(x_0))]^2}_{\text{Reducible}} + V(\epsilon). \quad (6)$$

where:

- $E(y_0 - f(x_0))^2$ denotes the *expected test MSE*.
- $Bias(\hat{f}(x_0)) = E[\hat{f}(x_0)] - f(x_0)$.

Conclusion: In order to minimize the expected test error, our method must **simultaneously** achieve **low variance** and **low bias**.

About Bias-Variance trade-off formula

Concerning formula (6):

- The expected test MSE can never lie below $V(\epsilon)$.

About Bias-Variance trade-off formula

Concerning formula (6):

- The expected test MSE can never lie below $V(\epsilon)$.
- **Variance** refers to the amount by which $\hat{f}$ would change if it is estimated using a different training data set. Generally, more flexible methods have higher variance.

About Bias-Variance trade-off formula

Concerning formula (6):

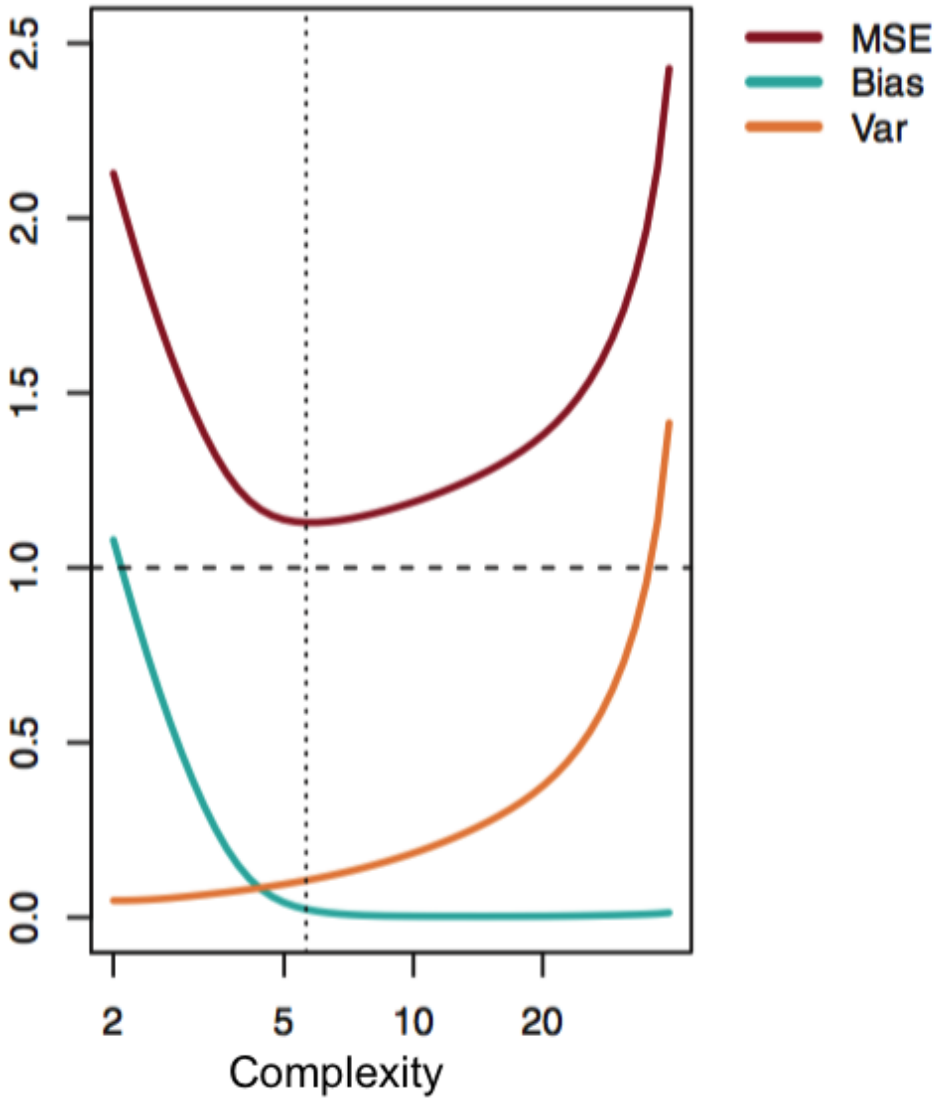
- The expected test MSE can never lie below $V(\epsilon)$.
- **Variance** refers to the amount by which $\hat{f}$ would change if it is estimated using a different training data set. Generally, more flexible methods have higher variance.
- **bias** refers to the error that is introduced by approximating a real-life problem, which may be extremely complicated, by a much simpler model. Generally, more flexible methods result in less bias.

About Bias-Variance trade-off formula

Concerning formula (6):

- The expected test MSE can never lie below $V(\epsilon)$.
- **Variance** refers to the amount by which $\hat{f}$ would change if it is estimated using a different training data set. Generally, more flexible methods have higher variance.
- **bias** refers to the error that is introduced by approximating a real-life problem, which may be extremely complicated, by a much simpler model. Generally, more flexible methods result in less bias.
- The relative rate of change of this two terms determines whether the test MSE increases or decreases. This is known as the **Bias-Variance trade-off**

Bias-Variance trade-off example



-As the flexibility increases, the bias tends to initially decrease faster than the variance increases.

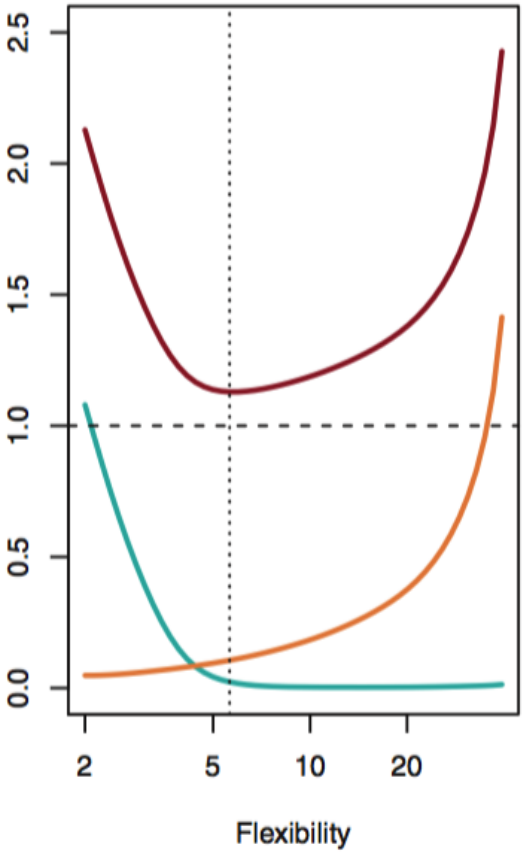
Then, the expected test MSE declines.

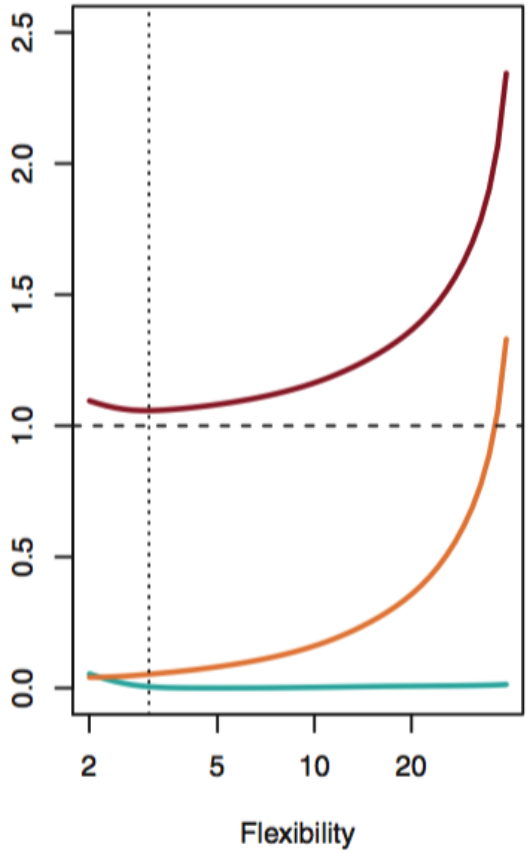
-At some point increasing flexibility has little impact on the bias but starts to significantly increase the variance.

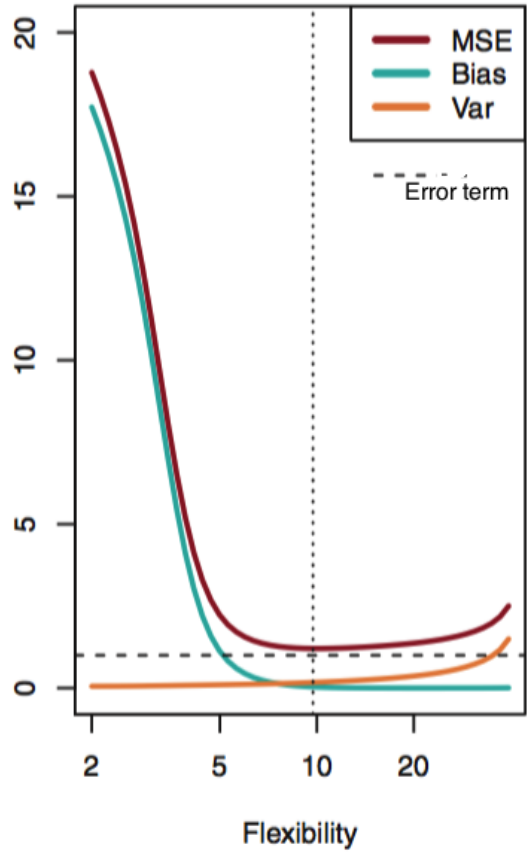
Then, the expected test MSE increases.

Bias-variance trade-off for the three examples

Typically as the complexity of $\hat{f}$ increases, its variance increases, and its bias decreases. So choosing the complexity based on **min** average test error amounts to a bias-variance trade-off.







Outline

References

- James, Gareth; Witten, Daniela; Hastie, Trevor and Tibshirani, Robert. "An Introduction to Statistical Learning with Applications in R", 2nd edition, New York : "Springer texts in statistics", 2021. Site web: `https://hastie.su.domains/ISLR2/ISLRv2_website.pdf`.
- Hastie, Trevor; Tibshirani, Robert and Friedman, Jerome. "The Elements of Statistical Learning (Data Mining, Inference, and Prediction), 2nd edition". New York: "Springer texts in statistics", 2009. Site web : `http://statweb.stanford.edu/~tibs/ElemStatLearn/`.
- Murphy, K. P. (2012). Machine Learning: a Probabilistic Perspective. MIT Press, Cambridge, MA, USA.