

Machine learning and physical modelling-1

julien.brajard@nersc.no

October 2019

NERSC

<https://github.com/brajard/MAT330>

Overview of the next lectures

1. Lecture 1 (Tuesday 15 Oct.): Generalities and principles of Machine Learning
2. Lecture 2 (Thursday 17 Oct.): Machine learning process, neural networks, deep learning
3. Practical Work (Tuesday 22 Oct.): **In the lecture room**
4. Lecture 3 (Thursday 24 Oct.): How to train a machine learning algorithm?

julien.brajard@nersc.no

References

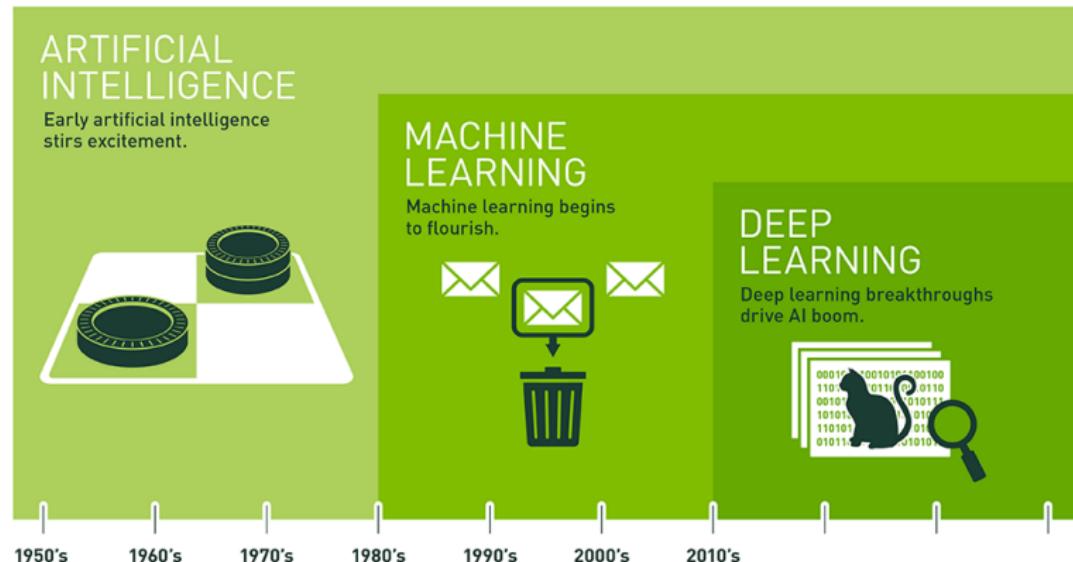
-  Ian Goodfellow, Yoshua Bengio, and Aaron Courville.
Deep Learning.
MIT Press, 2016.
<http://www.deeplearningbook.org>.
-  Jake VanderPlas.
Python Data Science Handbook: Essential Tools for Working with Data.
O'Reilly Media, Inc., 1st edition, 2016.

Table of contents

1. Introduction
2. Generalities on Machine Learning
3. Model selection/validation

Introduction

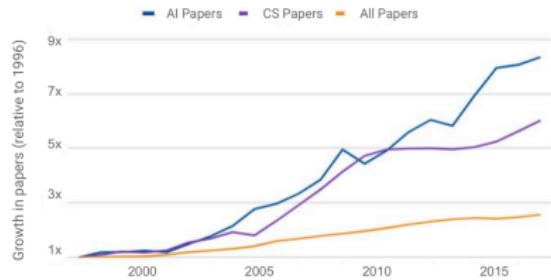
Scope of the lectures: Machine Learning



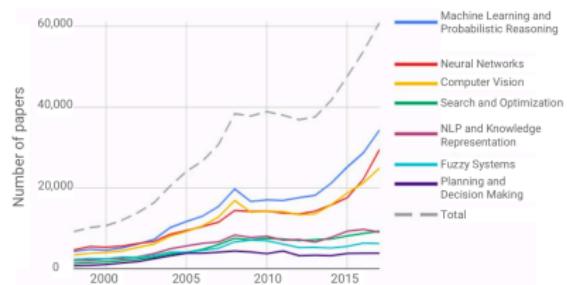
Since an early flush of optimism in the 1950s, smaller subsets of artificial intelligence – first machine learning, then deep learning, a subset of machine learning – have created ever larger disruptions.

A (very) active field

Growth of annually published papers by topic (1996–2017)
Source: Scopus

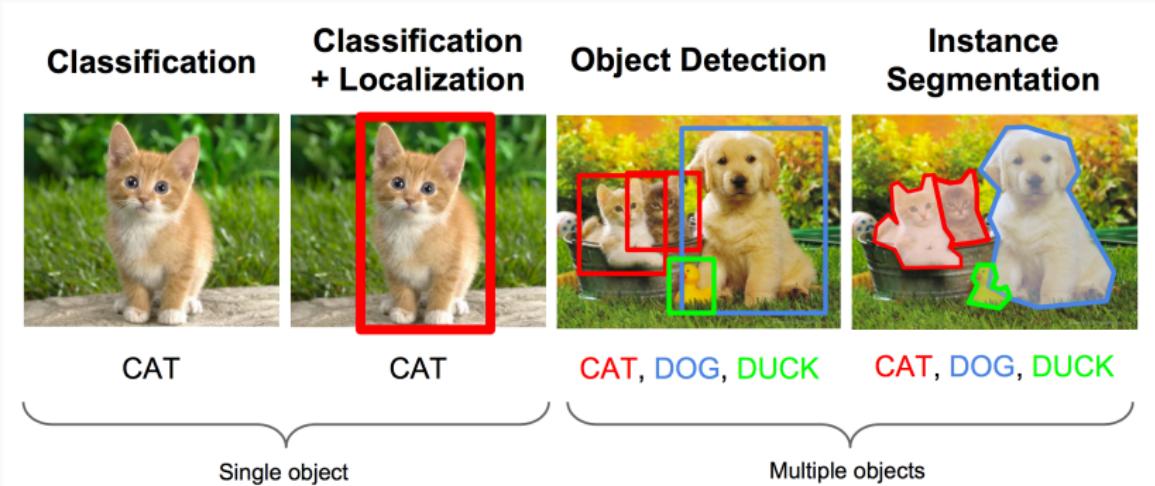


Number of AI papers on Scopus by subcategory (1998–2017)
Source: Elsevier



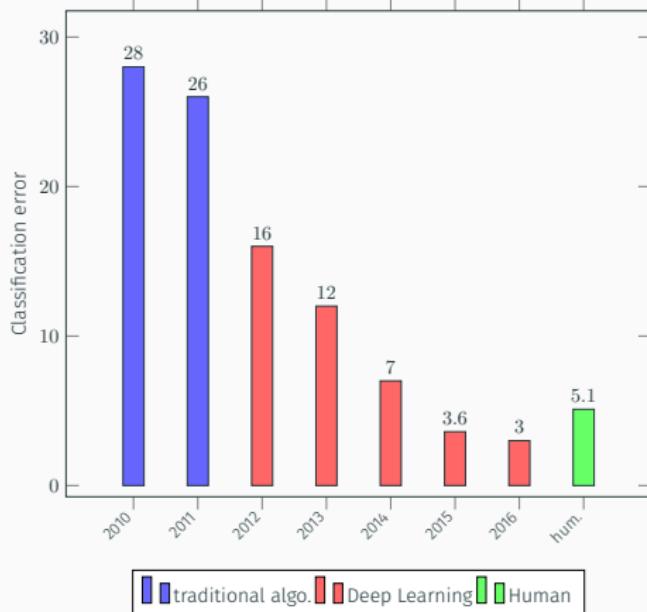
Shoham et al., AI Index 2018 Annual Report

Example 1: Computer Vision



Li, Karpathy and Johnson, 2016, Stanford CS231n course

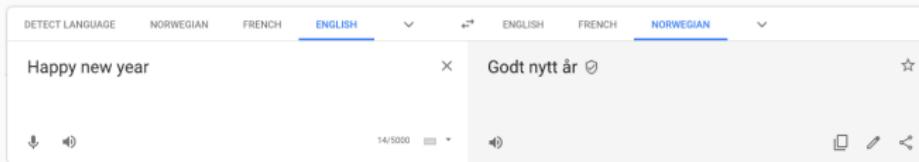
Example 1: Computer Vision



Deep learning architectures were based on Convolutional Neural Networks (CNN).

Machine Translation

Objective : translate a text from a language to another.



- Oct. 2013: Pionneering scientic paper about neural networks in machine translation (Kalchbrenner, N., and Blunsom, P.).
- 2016: Neural machine translation outperform tradiational approaches on public benchmarks
- 2017: Major systems (Google, Systran, WIPO, Microsoft) switch to neural machine translation (using deep recurrent neural networks)

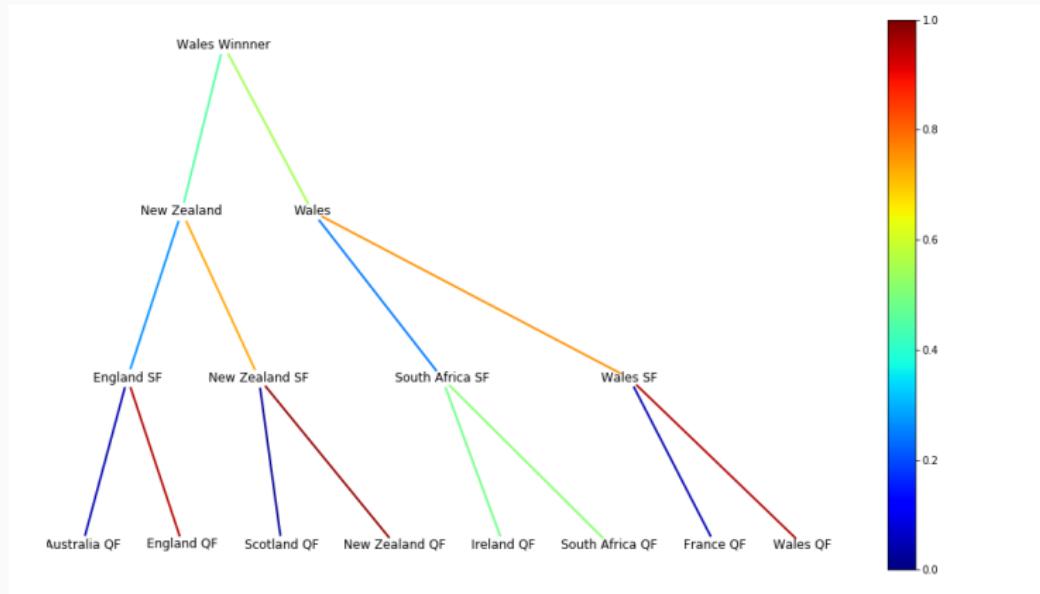
Playing Games

- 1997: Deep Blue defeats Kasparov at Chess.
- 2016: AlphaGo's victory again Lee Sedol at Go.
- 2017: AlphaGo Zero learns how to play Go only by playing against itself. It outperformed previous AlphaGo version
(Reinforcement learning)
- 2017: DeepStack beats professional human poker players.



Predicting sport results?

Rugby World Cup 2019



Prediction based on XGBoost. <https://towardsdatascience.com/predicting-the-winner-of-the-rugby-world-cup-2019-89a27d3f5b8e>

AI Art?

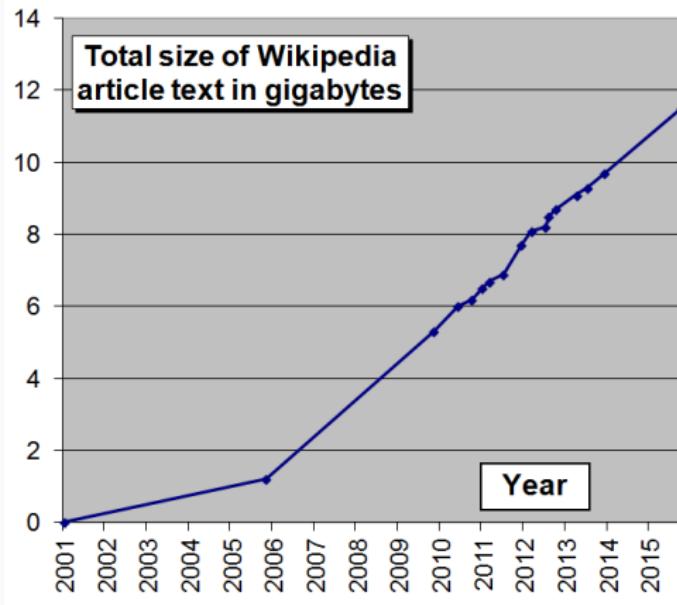


Edmond de Bellamy by Obvious(collective)

Generated using a Generative Adversarial Network.
Selling price (Oct. 2018): \$432,000

Reasons for these recent achievements?

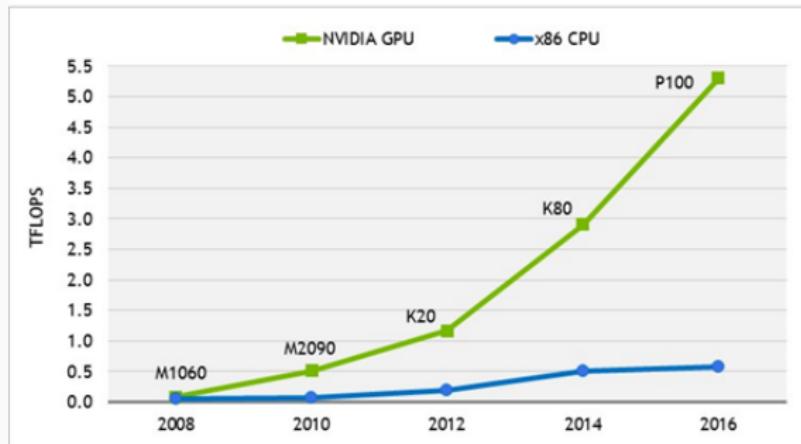
- Increasing of the datasets in size and quality



source: *Wikipedia*

Reasons for these recent achievements?

- Increasing of the datasets in size and quality
- Progress in computing resources.



source: NVIDIA

Reasons for these recent achievements?

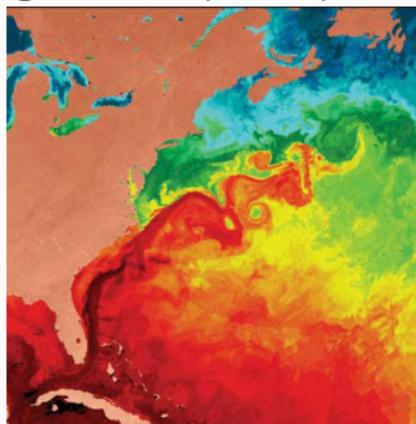
- Increasing of the datasets in size and quality
- Progress in computing resources.
- Scientific research on new algorithms (e.g adapted to image processing)



Apply Machine-Learning to physical modelling?

Why is it a good idea?

- A increasing number of geophysical data (one spatial mission: 24 TB/day)
- Data with highly significant spatial patterns

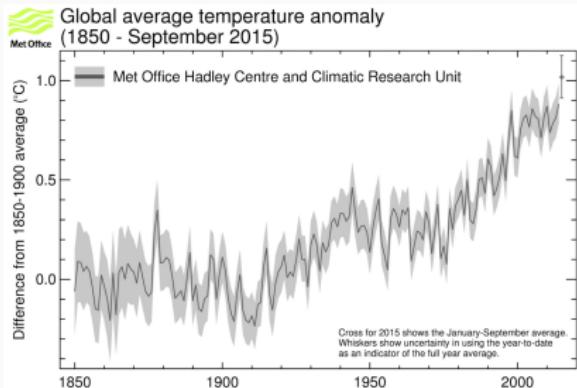
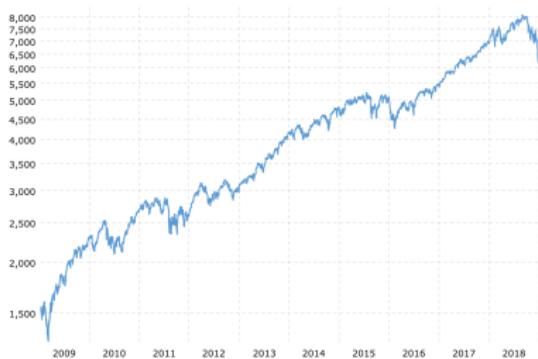


Sea Surface temperature of the gulf stream

source: Talley (2000)

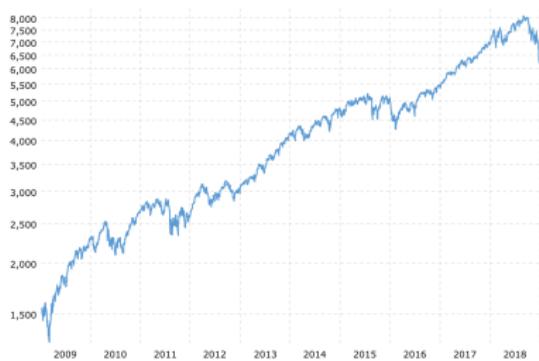
Why is physical modelling specific?

NASDAQ Composite sock market index over the last 10 years

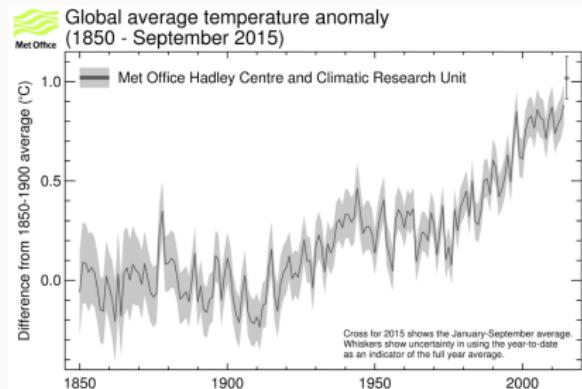


Why is physical modelling specific?

NASDAQ Composite sock market index over the last 10 years



Mostly unknown dynamical processes



Mostly known dynamical processes (based on physical principles)

Generalities on Machine Learning

What is this about ?

Can we extract knowledge, make some predictions,
determine a "model" using this large amount of data ?

What is this about ?

Can we extract knowledge, make some predictions,
determine a "model" using this large amount of data ?



→ Digit $\in \{0, \dots, 9\}$

Base of images

What is this about ?

Can we extract knowledge, make some predictions, determine a "model" using this large amount of data ?



→ Digit ∈ {0, ..., 9}

Base of images

- From high dimensional data (thousands to millions dimensions) to reduced dimensional data (less than 100)
- From disorganized data to comprehensive information
- Can we teach a machine how to do that ?

Two classes of Machine Learning problems

1. **Regression:** Determination of a quantitative variable from a set of data
 - The price of a building from various predictors (Surface, ...)
 - A physical value (Temperature, humidity, ...) in the future knowing the past
 - ...

Two classes of Machine Learning problems

1. **Regression:** Determination of a quantitative variable from a set of data
 - The price of a building from various predictors (Surface, ...)
 - A physical value (Temperature, humidity, ...) in the future knowing the past
 - ...
2. **Classification:** Determination of a class
 - A digit from a image
 - Identification of the content of an image
 - ...

Two types of objectives

1. **Supervised learning:** we have a set of labeled data with examples of targets.

Two types of objectives

1. **Supervised learning**: we have a set of labeled data with examples of targets.
2. **Unsupervised learning**: we only have unlabeled data, we have no examples of what we want to obtain. We want to extract a "useful" representation of these data, or some coherent categories.

Two types of objectives

1. **Supervised learning**: we have a set of labeled data with examples of targets.
2. **Unsupervised learning**: we only have unlabeled data, we have no examples of what we want to obtain. We want to extract a "useful" representation of these data, or some coherent categories.
 - Determine typical behaviors of clients in a supermarket knowing what they have bought.

Two types of objectives

1. **Supervised learning**: we have a set of labeled data with examples of targets.
2. **Unsupervised learning**: we only have unlabeled data, we have no examples of what we want to obtain. We want to extract a "useful" representation of these data, or some coherent categories.
 - Determine typical behaviors of clients in a supermarket knowing what they have bought.
3. **Semi-Supervised Learning**: Only a few subset of the data are labeled

Two types of objectives

1. **Supervised learning:** we have a set of labeled data with examples of targets.
2. **Unsupervised learning:** we only have unlabeled data, we have no examples of what we want to obtain. We want to extract a "useful" representation of these data, or some coherent categories.
 - Determine typical behaviors of clients in a supermarket knowing what they have bought.
3. **Semi-Supervised Learning:** Only a few subset of the data are labeled
4. **Reinforcement Learning:** We can initiate and observe the interaction of an agent with its environment. We want to optimize the behavior of the agent.

Two types of objectives

1. **Supervised learning:** we have a set of labeled data with examples of targets.
2. **Unsupervised learning:** we only have unlabeled data, we have no examples of what we want to obtain. We want to extract a "useful" representation of these data, or some coherent categories.
 - Determine typical behaviors of clients in a supermarket knowing what they have bought.
3. **Semi-Supervised Learning:** Only a few subset of the data are labeled
4. **Reinforcement Learning:** We can initiate and observe the interaction of an agent with its environment. We want to optimize the behavior of the agent.
 - Playing a chess game.

A Machine

$$y = \mathcal{M}(x, \theta)$$

- x : input
- y : output
- \mathcal{M} : a model (named "machine")
- θ : parameters of the model \mathcal{M} .

Machine learning consists in optimizing θ using a set of data.
This is the training process.

The Machine Learning recipe

A Machine

$$y = \mathcal{M}(x, \theta)$$

What are **the ingredients?**

The Machine Learning recipe

A Machine

$$y = \mathcal{M}(x, \theta)$$

What are **the ingredients?**

- Some **data**
 - x, y : supervised learning
 - only x : unsupervised learning
 - x and some subset of y : semi-supervised learning

The Machine Learning recipe

A Machine

$$y = \mathcal{M}(x, \theta)$$

What are **the ingredients?**

- Some **data**
 - x, y : supervised learning
 - only x : unsupervised learning
 - x and some subset of y : semi-supervised learning
- An **objective**
 - y is quantitative: regression
 - y is a class: classification

The Machine Learning recipe

A Machine

$$y = \mathcal{M}(x, \theta)$$

What are **the ingredients?**

- Some **data**
 - x, y : supervised learning
 - only x : unsupervised learning
 - x and some subset of y : semi-supervised learning
- An **objective**
 - y is quantitative: regression
 - y is a class: classification
- A computational architecture (the **machine**)
 - linear
 - non-linear
 - neural networks, random forest, ...

The Machine Learning recipe

A Machine

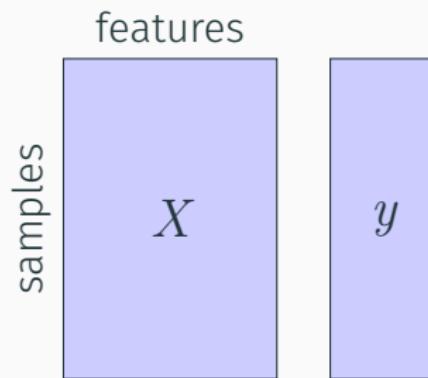
$$y = \mathcal{M}(x, \theta)$$

What are **the ingredients?**

- Some **data**
 - x, y : supervised learning
 - only x : unsupervised learning
 - x and some subset of y : semi-supervised learning
- An **objective**
 - y is quantitative: regression
 - y is a class: classification
- A computational architecture (the **machine**)
 - linear
 - non-linear
 - neural networks, random forest, ...
- A **learning** process
 - Estimation of θ

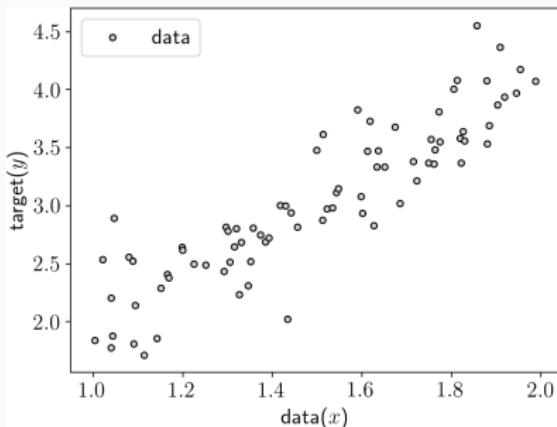
Multidimensional data

Generally, we have multidimensional data X and a one-dimensional target y .



An illustration

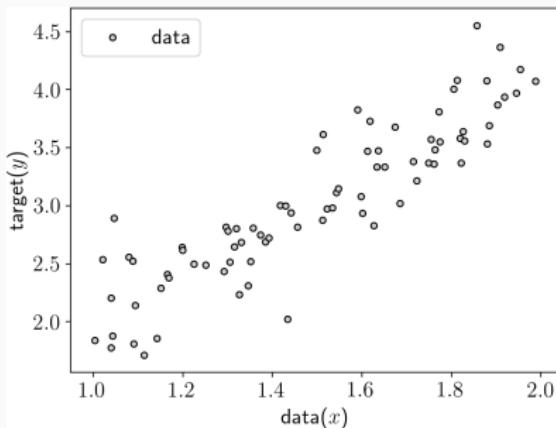
- Some Data



- y is known: supervised learning
- y is quantitative: regression

An illustration

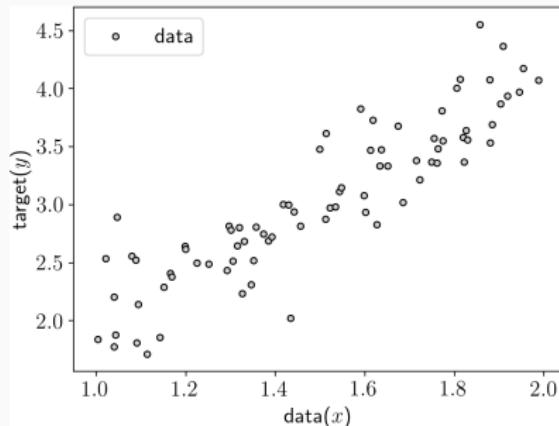
- Some Data



- y is known: supervised learning
- y is quantitative: regression
- An **Objective**: Estimate \hat{y} from x by minimizing $(\hat{y} - y)^2$
(Least-square objective)

An illustration

- Some Data

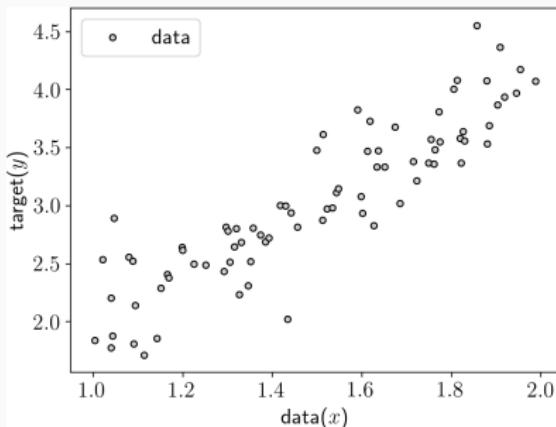


- A model: $y = \theta_1 X + \theta_0$ (linear)

- y is known: supervised learning
- y is quantitative: regression
- An Objective: Estimate \hat{y} from x by minimizing $(\hat{y} - y)^2$
(Least-square objective)

An illustration

- Some Data

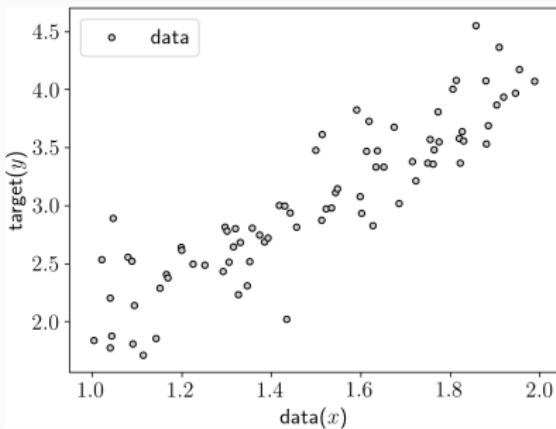


- A **model**: $y = \theta_1 X + \theta_0$ (linear)
- A **learning** process:
$$\theta = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T y$$

- y is known: supervised learning
- y is quantitative: regression
- An **Objective**: Estimate \hat{y} from x by minimizing $(\hat{y} - y)^2$
(Least-square objective)

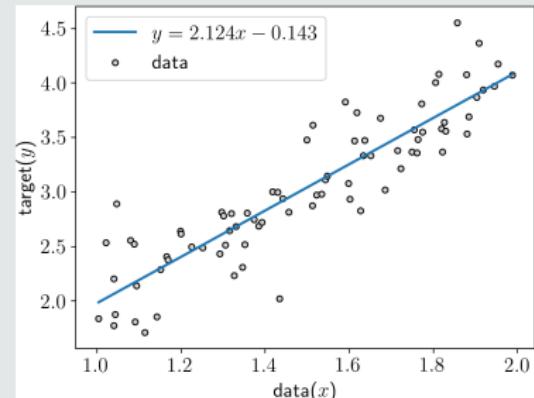
An illustration

- Some Data



- A **model**: $y = \theta_1 X + \theta_0$ (linear)
- A **learning process**:
$$\theta = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T y$$

Result



- y is known: supervised learning
- y is quantitative: regression
- An **Objective**: Estimate \hat{y} from x by minimizing $(\hat{y} - y)^2$
(Least-square objective)

Model selection/validation

Polynomial regression

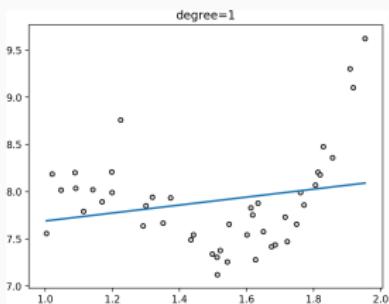
$$y = \theta_0 + \theta_1 x + \theta_2 x^2 + \cdots + \theta_d x^d = \sum_{i=0}^d \theta_i X^i$$

Choice of the model

Polynomial regression

$$y = \theta_0 + \theta_1 x + \theta_2 x^2 + \cdots + \theta_d x^d = \sum_{i=0}^d \theta_i X^i$$

degree = 1 (linear)

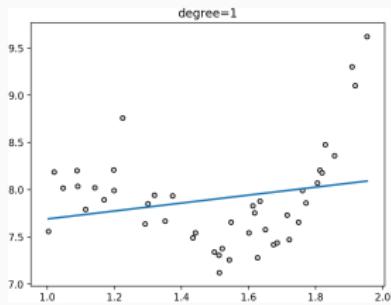


Choice of the model

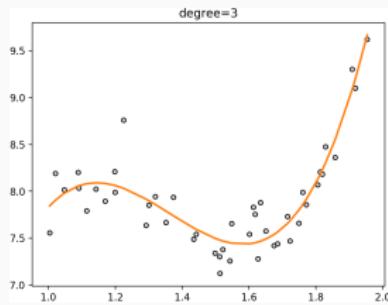
Polynomial regression

$$y = \theta_0 + \theta_1 x + \theta_2 x^2 + \cdots + \theta_d x^d = \sum_{i=0}^d \theta_i X^i$$

degree = 1 (linear)



degree = 3

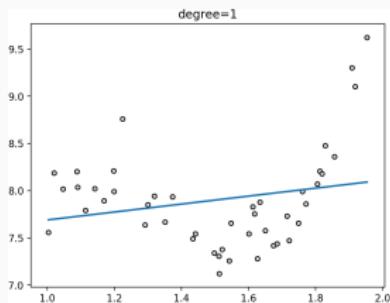


Choice of the model

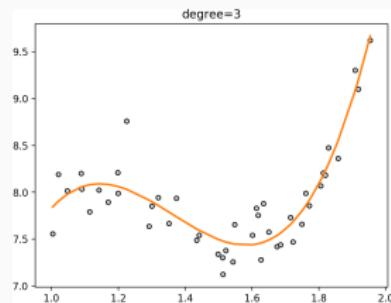
Polynomial regression

$$y = \theta_0 + \theta_1 x + \theta_2 x^2 + \cdots + \theta_d x^d = \sum_{i=0}^d \theta_i X^i$$

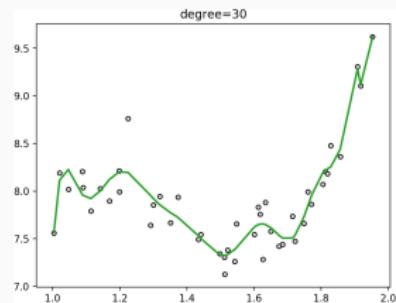
degree = 1 (linear)



degree = 3



degree = 30



What is the best model?

Train/Validation split

The idea

Evaluate a score on a independent dataset

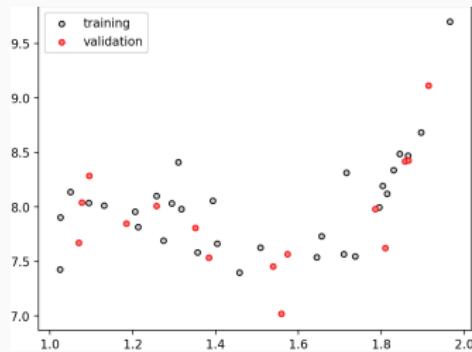
Train/Validation split

The idea

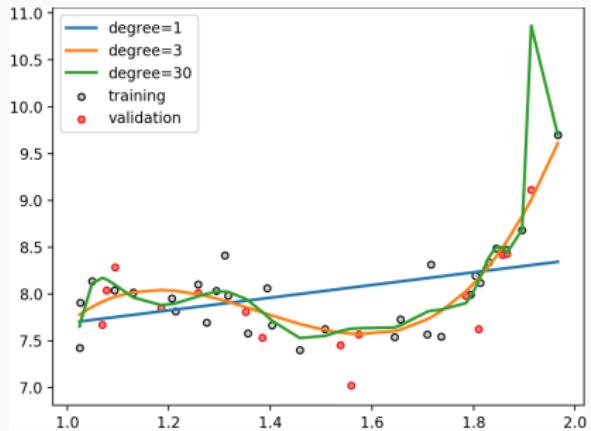
Evaluate a score on a independent dataset

In our example we can randomly divide (X, y) in two datasets:

- The training dataset X_{train}, y_{train} used to fit the model.
- The validation dataset X_{val}, y_{val} used to compute the score (e.g., correlation, mean-squared error)



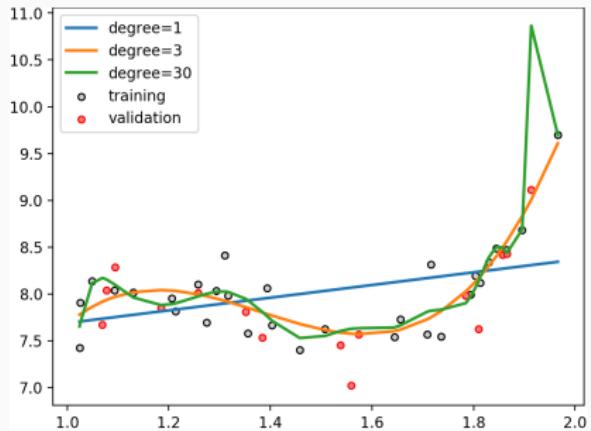
Choice of the model



Score: Mean Square Error (MSE)

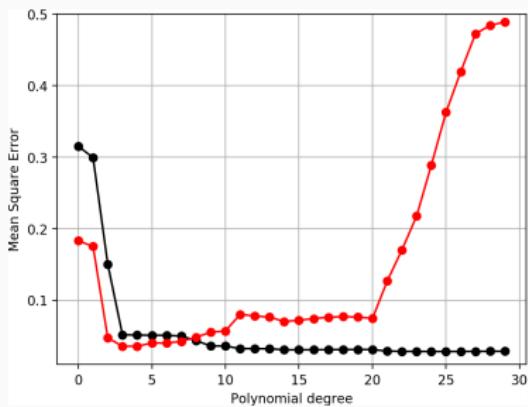
Deg.	Train Score	Val. Score
1	0.17	0.23
3	0.045	0.062
30	0.035	0.27

Choice of the model



Score: Mean Square Error (MSE)

Deg.	Train Score	Val. Score
1	0.17	0.23
3	0.045	0.062
30	0.035	0.27

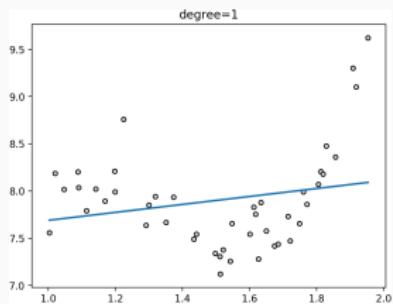


Choice of the model

Polynomial regression

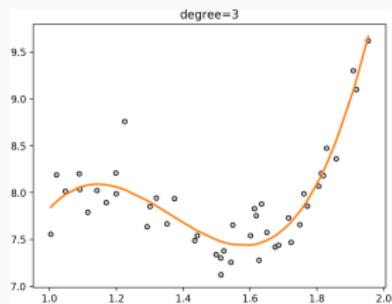
$$y = \theta_0 + \theta_1 x + \theta_2 x^2 + \cdots + \theta_d x^d = \sum_{i=0}^d \theta_i X^i$$

degree = 1 (linear)



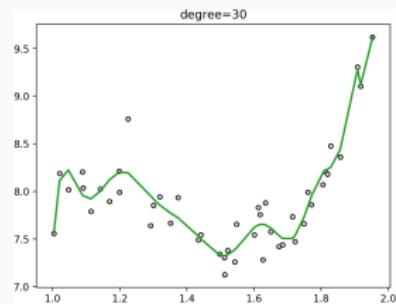
underfitting

degree = 3



good fit

degree = 30



overfitting

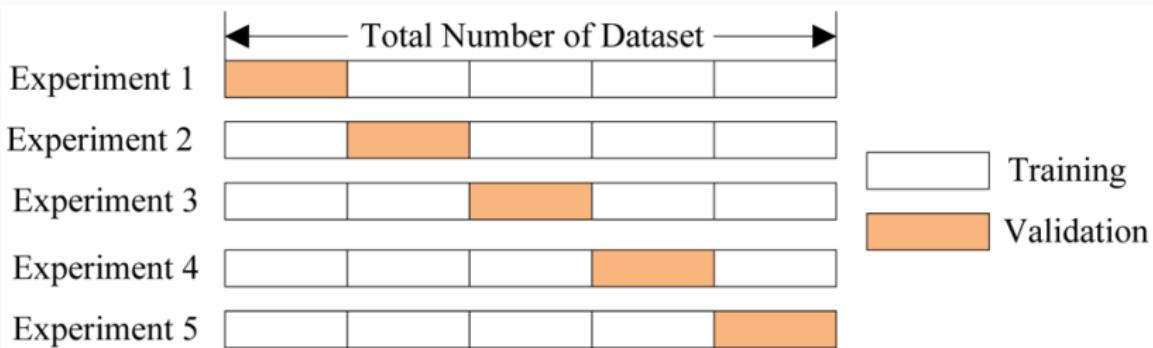
Drawbacks

- drastically reduce the number of samples which can be used for learning the model
- Results can depend on a particular random choice for the pair of (train, validation) sets.

More Robust: cross validation

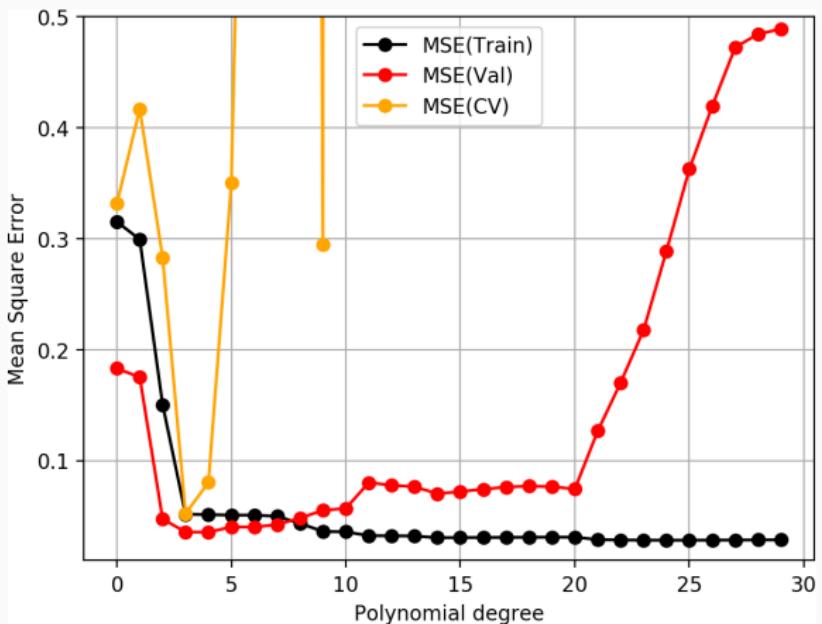
The idea

- Dividing the data in n folds,
- Learning n model (each time with a different training set),
- Compute the mean score over n validation set.



Cross-Validation

Fold	MSE
1	0.052
2	0.043
3	0.137
4	0.025
5	0.048
6	0.144
7	0.011
8	0.025
9	0.010
10	0.028
Mean	0.05



Wrapping up

1. When applying machine learning techniques there are **hyperparameters** to be determined (e.g., degree of the polynomial in polynomial regression).

Wrapping up

1. When applying machine learning techniques there are **hyperparameters** to be determined (e.g., degree of the polynomial in polynomial regression).
2. These **hyperparameters** can be determined by splitting the data into training/validation or by cross-validation.

Wrapping up

1. When applying machine learning techniques there are **hyperparameters** to be determined (e.g., degree of the polynomial in polynomial regression).
2. These **hyperparameters** can be determined by splitting the data into training/validation or by cross-validation.
3. But then... the validation set was used to determine the best machine learning process

Wrapping up

1. When applying machine learning techniques there are **hyperparameters** to be determined (e.g., degree of the polynomial in polynomial regression).
2. These **hyperparameters** can be determined by splitting the data into training/validation or by cross-validation.
3. But then... the validation set was used to determine the best machine learning process
4. To evaluate independantly the performance of our model, we should compute the score on a **third independant dataset: The test dataset.**

Wrapping up

1. When applying machine learning techniques there are **hyperparameters** to be determined (e.g., degree of the polynomial in polynomial regression).
2. These **hyperparameters** can be determined by splitting the data into training/validation or by cross-validation.
3. But then... the validation set was used to determine the best machine learning process
4. To evaluate independantly the performance of our model, we should compute the score on a **third independant dataset: The test dataset.**

Wrapping up

1. When applying machine learning techniques there are **hyperparameters** to be determined (e.g., degree of the polynomial in polynomial regression).
2. These **hyperparameters** can be determined by splitting the data into training/validation or by cross-validation.
3. But then... the validation set was used to determine the best machine learning process
4. To evaluate independantly the performance of our model, we should compute the score on a **third independant dataset: The test dataset.**

More on that in next lecture...

Data Leakage

What is it?

Data Leakage is when information from outside the training dataset is used to create the model.

Data Leakage

What is it?

Data Leakage is when information from outside the training dataset is used to create the model.

Consequence: it can lead to overly optimistic models with no practical predictive skills.

Data Leakage

What is it?

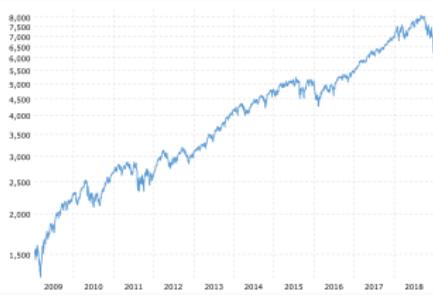
Data Leakage is when information from outside the training dataset is used to create the model.

Consequence: it can lead to overly optimistic models with no practical predictive skills.

Example:

- Using the future to make a forecast.
- Splitting between validation/training is not done properly
- Non-relevant features that can be related to the expected results (e.g. name/date of the files in a training dataset)

Example of time series



Data: x_t : NASDAQ index at the date t .

Objective: predict x_t given the index in the past (x_0, \dots, x_{t-1})

A correct model:

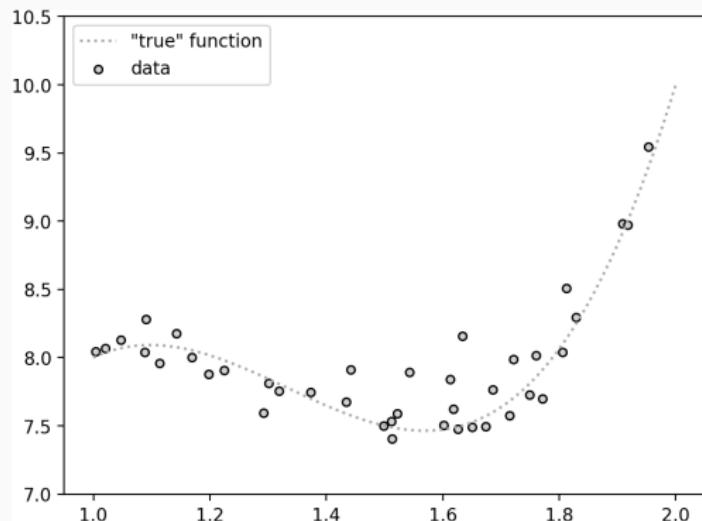
$$\hat{x}_t = \mathcal{M}(x_0, \dots, x_{t-1}, \theta)$$

We can construct an incorrect model:

$$\hat{x}_t = \mathcal{M}^{\text{leak}}(x_0, \dots, x_{t-1}, x_t, \theta)$$

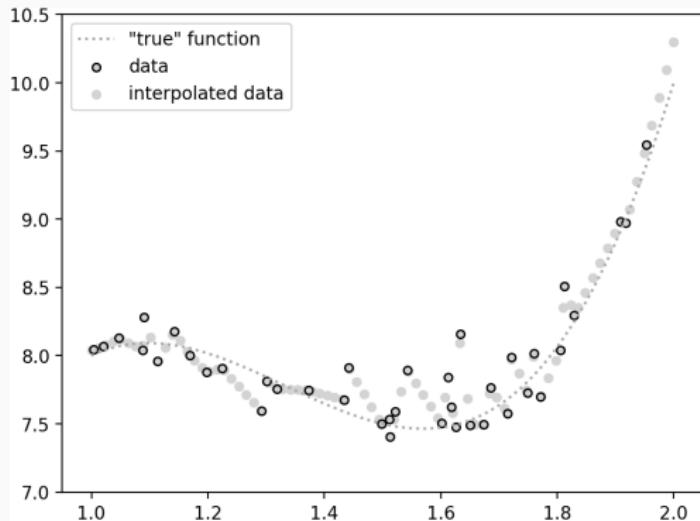
$\mathcal{M}^{\text{leak}}$ is perfect on the training set but is completely useless in practice.

More subtle example: contamination of the validation dataset



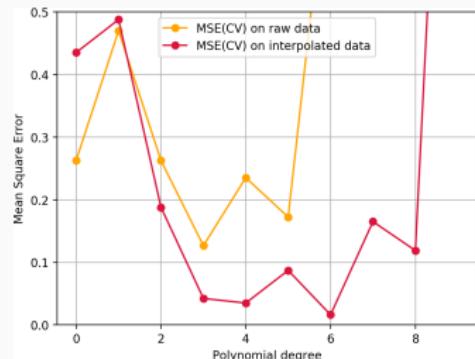
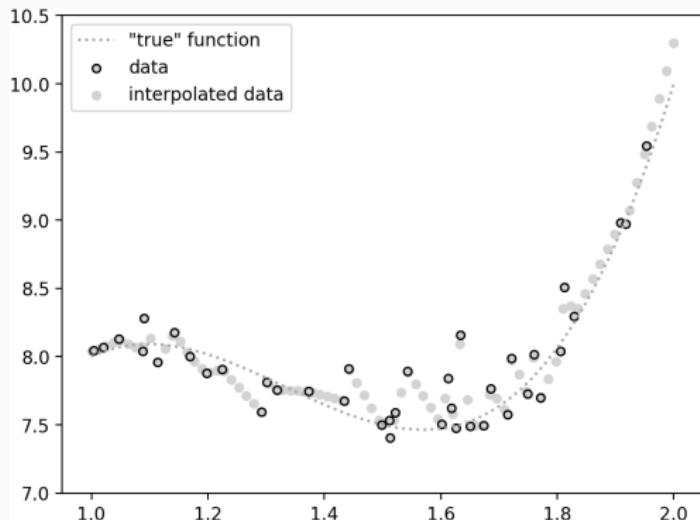
- True polynom: $12x^3 - 48x^2 + 62x - 18$

More subtle example: contamination of the validation dataset



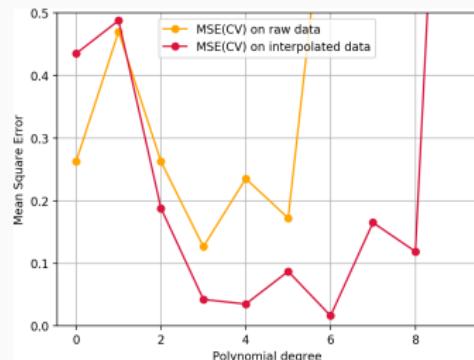
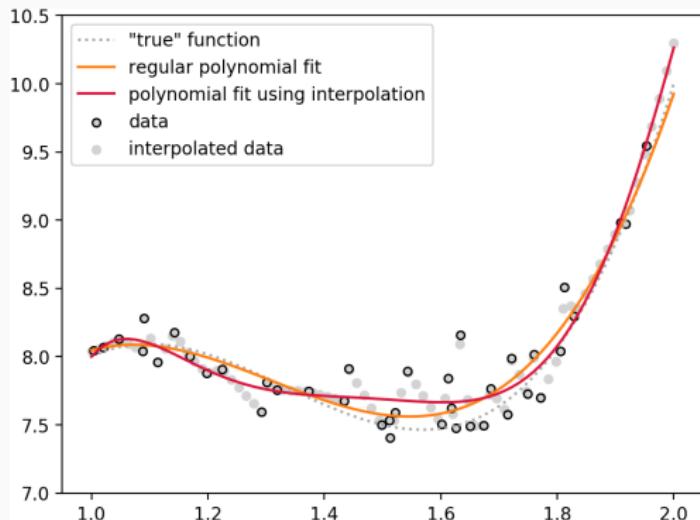
- True polynom: $12x^3 - 48x^2 + 62x - 18$

More subtle example: contamination of the validation dataset



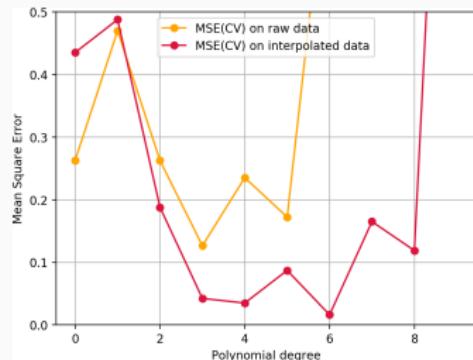
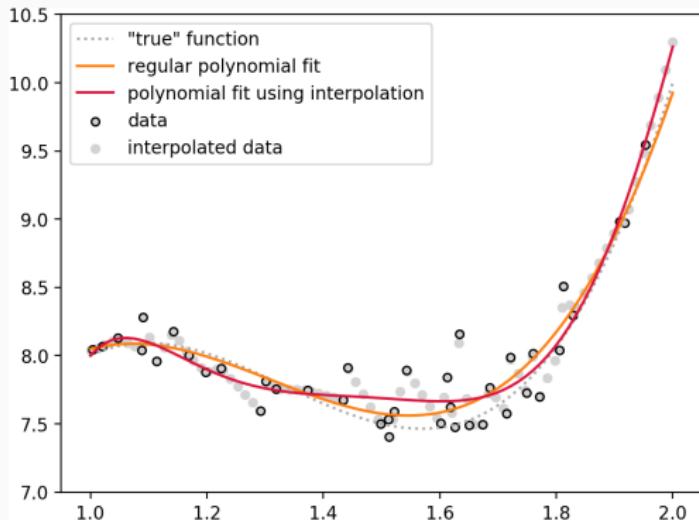
- True polynom: $12x^3 - 48x^2 + 62x - 18$

More subtle example: contamination of the validation dataset



- True polynom: $12x^3 - 48x^2 + 62x - 18$

More subtle example: contamination of the validation dataset



- True polynom: $12x^3 - 48x^2 + 62x - 18$
- regular fit: $12x^3 - 49x^2 + 38x - 9$
- interpolated fit:
$$1143x^6 - 4803x^5 + 8242x^4 - 7446x^3 + 3736x^2 - 986x + 107$$

An example using scikit-learn

Boston House Prices



Questions addressed in this lecture

- What are the more common area of application of machine learning? [GBC16, 1]
- What are the different class of ML problems? [Van16, 5.1]
- What are the different types of learning (supervised, ...)? [Van16, 5.1]
- How to validate/select a model? [Van16, 5.3]
- What is the cross-validation? [Van16, 5.3]
- What is data leakage?

<https://machinelearningmastery.com/data-leakage-machine-learning/>

<https://www.kaggle.com/c/the-icml-2013-whale-challenge-right-whale-redux/discussion/4865#25839>

Refs

[Van16, *n*]: Jake VanderPlas, *Python Data Science Handbook*, section *n*

[GBC16, *n*]: Goodfellow et al., Deep Learning, chapter *n*