Introduction to Supervised Learning

Michela Papandrea
michela.papandrea@supsi.ch

Supervised Learning
Bachelor of Data Science and Artificial Intelligence
University of Applied Sciences and Arts of Southern Switzerland

Overview

1. Introduction
2. Data Representation
3. Classification vs Regression
   - Definition of Classification
   - Definition of Regression
4. Generalization, Overfitting and Underfitting

Machine Learning

- extracting knowledge from data.
- intersection of statistics, artificial intelligence, and computer science (aka *predictive analytics* or *statistical learning*)
- ML applications is ubiquitous

  many modern websites and devices have machine learning algorithms at their core

Example

- automatic recommendations of which movies to watch, what food to order or which products to buy,
- personalized online music streaming
- recognizing friends faces on your photos, inferring age and gender

Why Machine Learning?

Past

intelligent applications involved handcoded rules:

`if-then-else`

decisions to process data

Example: spam filter with blacklist of words

Present

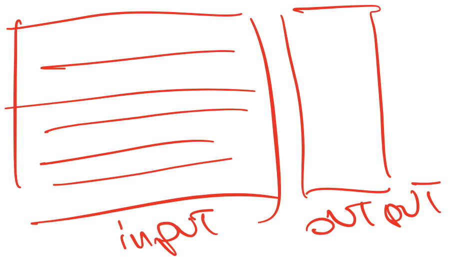
major disadvantages of manually crafted decision rules

1. logic required to make a decision is specific to a single domain and task.

   $\Longrightarrow$ every slight change in the task might require a rewrite of the whole system
2. human expert deep understanding of how a decision should be made is necessary

Example: faces detection in images

Supervised Learning



- automate decision-making processes by *generalizing* from known examples
- (**dataset**) data is provided as pairs of *inputs* and *desired outputs*
- (**model**) the algorithm finds a way to produce the desired output given an input
- (**generalization**) the algorithm generate autonomously an output for an input it has never seen before

Supervised Learning: the meaning

Supervised Learning

- *Supervised Learning* is a subbranch of *Machine Learning*
- algorithms that learn from examples (`<input, desired output>` pairs)
- "*Supervised*" refers to: having a teacher which supervise the whole process
- *supervision* is provided in the form of *desired outputs* for each training example
- require a laborious manual process of inputs and outputs dataset creation
- prediction performances are quantitatively measurable
- **training**: the algorithm will search for patterns in the data that correlate the input with the desired outputs
- **prediction**: the algorithm takes new unseen inputs and determine the output (`<new input, predicted output>`) based on prior training data
- **objective of a SL model**: predict the correct label for newly presented input data

Examples of supervised machine learning tasks

Example

Identifying the zip code from handwritten digits on an envelope

- **input**: a scan of the handwriting,
- **output**: actual digits in the zip code.
- To create a dataset for building a machine learning model, you need to collect many envelopes. Then you can read the zip codes yourself and store the digits as your desired outcomes.

Example

Determining whether a tumor is benign based on a medical image

- **input**: the image
- **output**: whether the tumor is benign (Y/N)
- To create a dataset for building a model, you need a database of medical images. You also need an expert opinion, so a doctor needs to look at all of the images and decide which tumors are benign and which are not. It might even be necessary to do additional diagnosis beyond the content of the image to determine whether the tumor in the image is cancerous or not.

Examples of supervised machine learning tasks

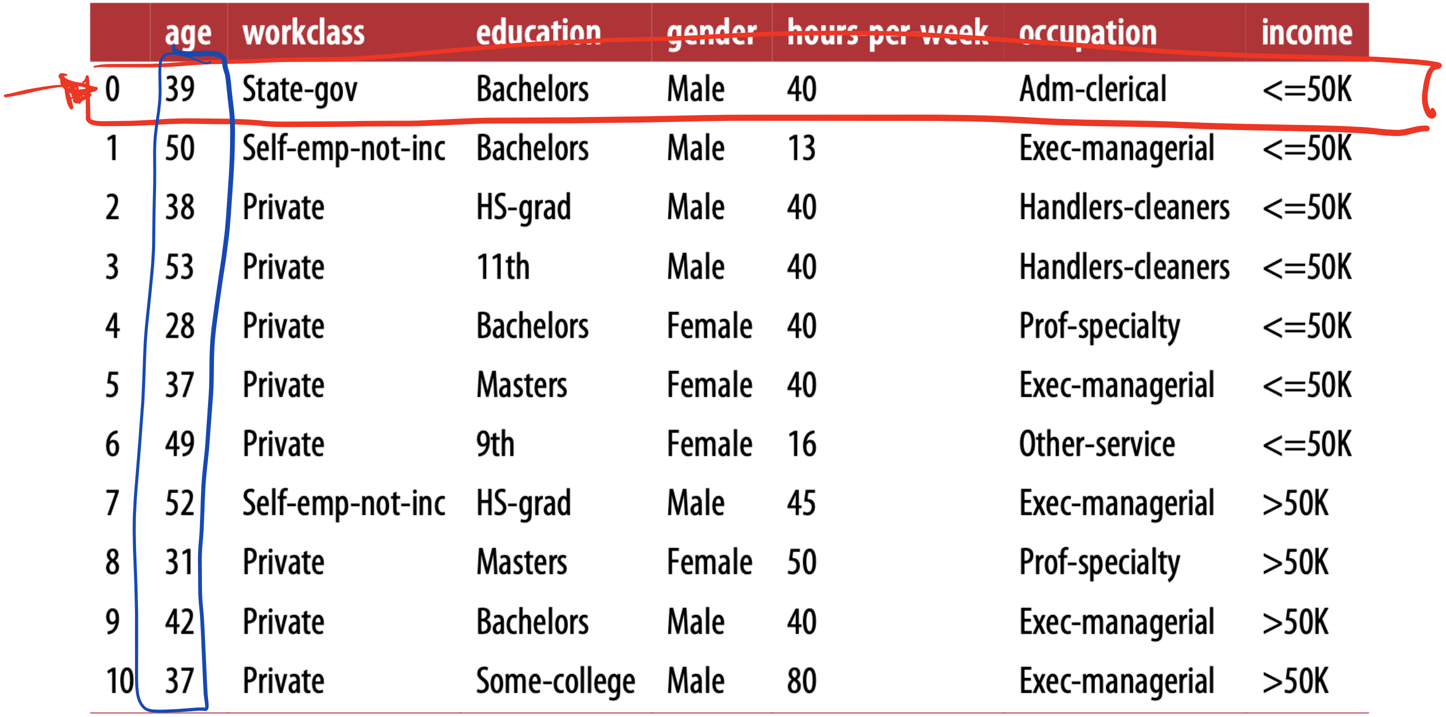
Example

Detecting fraudulent activity in credit card transactions

- **input**: a record of the credit card transaction
- **output**: whether it is likely to be fraudulent or no
- Assuming that you are the entity distributing the credit cards, collecting a dataset means storing all transactions and recording if a user reports any transaction as fraudulent.

Data representation

- Despite the nature of the data, it is important to have a representation of your input data that a computer can understand
- commonly a dataset is representation as a **table**
    - **row** (or **entry**): each **data point** (or **sample**) that we want to reason about
    - **column**: each **property** that describes that data point (**features**)



	age	workclass	education	gender	hours per week	occupation	income
0	39	State-gov	Bachelors	Male	40	Adm-clerical	<=50K
1	50	Self-emp-not-inc	Bachelors	Male	13	Exec-managerial	<=50K
2	38	Private	HS-grad	Male	40	Handlers-cleaners	<=50K
3	53	Private	11th	Male	40	Handlers-cleaners	<=50K
4	28	Private	Bachelors	Female	40	Prof-specialty	<=50K
5	37	Private	Masters	Female	40	Exec-managerial	<=50K
6	49	Private	9th	Female	16	Other-service	<=50K
7	52	Self-emp-not-inc	HS-grad	Male	45	Exec-managerial	>50K
8	31	Private	Masters	Female	50	Prof-specialty	>50K
9	42	Private	Bachelors	Male	40	Exec-managerial	>50K
10	37	Private	Some-college	Male	80	Exec-managerial	>50K

Knowing Your Data, and Your Task

- In ML it is not effective to randomly choose an algorithm and throw your data at it.
- Each algorithm is different in terms of what kind of data and what problem setting it works best for

Main steps in a ML analysis:

1. ✓ Understand the problem we are trying to solve and if the data can solve the problem
2. ✓ Formalize the problem
3. Collect enough data to solve the problem
4. ✓ Identify features and algorithms which allow right predictions
5. ✓ Define metrics for the performances measurement
6. Generate the predictive model and integrate the ML solution within a business product

Supervised Learning: formalism

At its most basic form, a supervised learning algorithm can be written simply as:

$$\overset{\text{output}}{y} = \underline{f}(\overset{\text{input}}{x}) \tag{1}$$

Where:

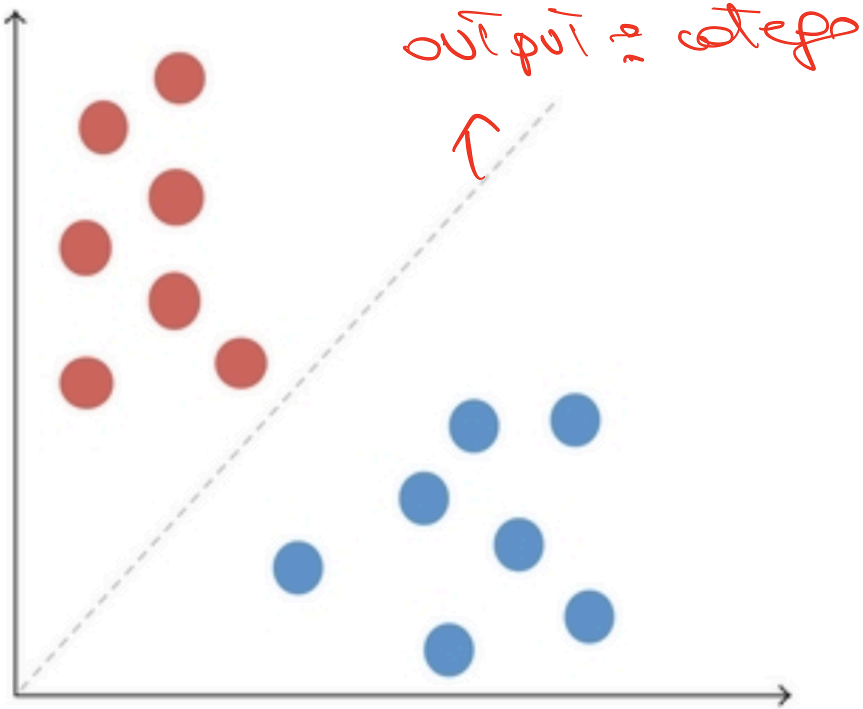
- y is the predicted output that is determined by a mapping function that assigns a class to an input value x.
- The function used to connect input features to a predicted output is created by the machine learning model during training.

Classification VS Regression

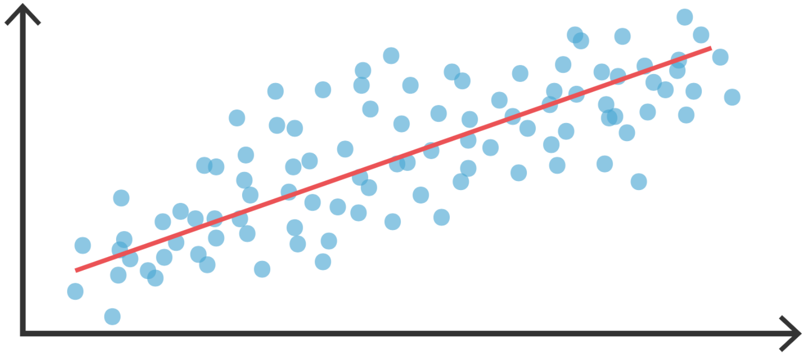
Supervised learning can be split into two subcategories: **Classification** and **Regression**.

output = categorical

output : continuous



(a) Classification



(b) Regression

Classification VS Regression

How to distinguish between Classification and Regression

Question: it there some kind of continuity in the output?
if **YES**, the problem is a regression problem

Example (Regression)

Predict annual income task
There is a clear continuity in the output. Whether a person makes $40,000$ or $40,001$ a year does not make a tangible difference, even though these are different amounts of money; if our algorithm predicts $39,999$ or $40,001$ when it should have predicted $40,000$, we don't mind that much.

Example (Classification)

Recognize the language of a website task
There is no matter of degree. A website is in one language, or it is in another. There is no continuity between languages, and there is no language that is between English and French.

Classification

- **Main goal**: to predict a *class label*, which is a choice from a predefined list of possibilities
- During *training*, the classification algorithm will be given data points with an assigned *category*. The job of the algorithm is to then take an input value and assign it a **class**, or *category*, that it fits into based on the training data provided.

Example of classification: determine if an email is spam or not

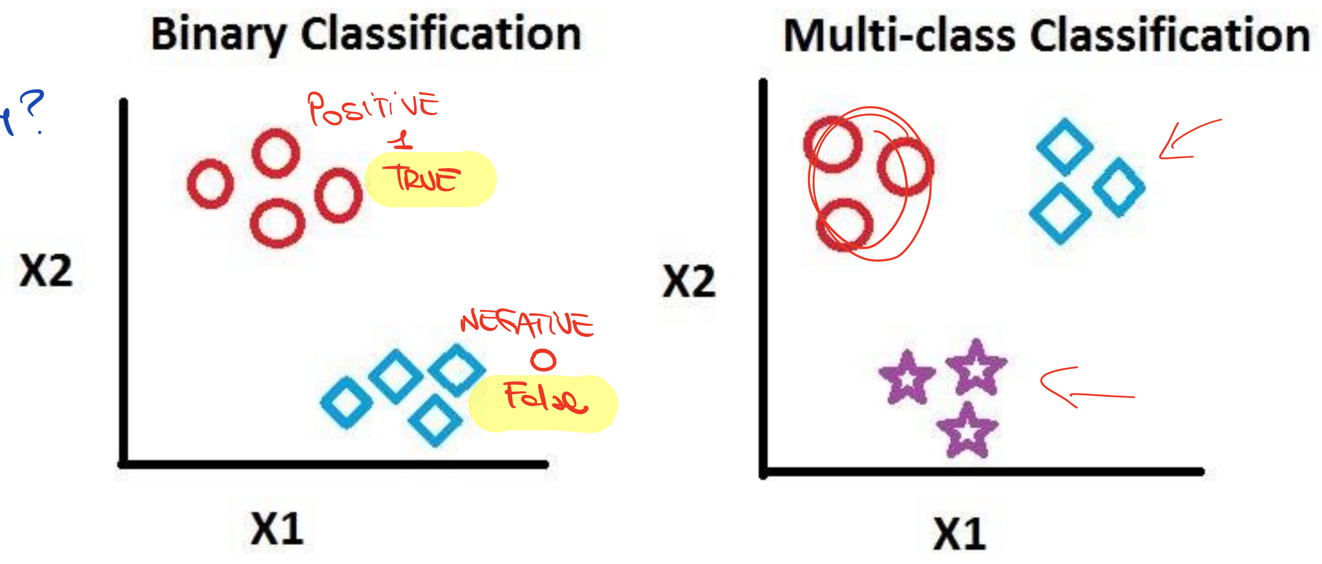
- two classes to choose from (`spam`, or `not spam`)
- the algorithm will be given training data with emails that are both spam and not spam
- the model will find the features within the data that correlate to either class and create the mapping function $y = f(x)$
- when provided with an unseen email, the model will use this function to determine whether or not the email is spam.

Classification

There are two types of classification:

- **binary classification**: distinguishing between exactly two classes
  - we often speak of one class being the **positive** class and the other class being the **negative** class – positive does not have a semantic meaning
  - Example: email spam identification
- **multiclass classification** classification between more than two classes.
  - Example: predict what language a website is in from the text

☑ is this SPAM?



end 19/09

Classification

Classification problems can be solved with a numerous amount of algorithms.
The choise for a specific algorithm depends on the data and the situation.
Here are a few popular classification algorithms:

- Linear Classifiers
- Support Vector Machines
- Decision Trees
- K-Nearest Neighbor
- Random Forest
- Neural Networks
- Naive Bayes
- ...

Regression

Regression

Predictive statistical process where the model attempts to find the important relationship between *dependent* and *independent* variables

Goal:

predict a *continuous number*
(*floating-point number* – programming, *real number* – math)
Example: predicting a person's annual income from their education, their age, and where they live

Regression

$$\hat{y} = \underbrace{\begin{array}{c|cccc} 1 & x_0 & x_1 & \dots & x_i \\ \hline 1 & & & & \\ 1 & & & & \\ 1 & & & & \\ \vdots & & & & \\ 1 & & & & \end{array}}_{n\times m\ X\ (n\text{ rows},\ m\text{ columns})} \cdot \underbrace{\begin{bmatrix} b \\ w_0 \\ w_1 \\ \vdots \\ w_i \end{bmatrix}}_{W_{m\times 1}} = \underbrace{\begin{bmatrix} 1\cdot b + x_{0,1}\cdot w_0 + x_{1,1}\cdot w_1 + \dots + x_{i,1}\cdot w_i \\ \dots \\ \dots \\ \dots \\ \dots \end{bmatrix}}_{\hat{y}_{n\times 1}} = \begin{bmatrix} \hat{y}_1 \\ \hat{y}_2 \\ \vdots \\ \hat{y}_n \end{bmatrix}$$

The equation for basic linear regression can be written as so:

$$\hat{y} = w[0] \cdot x[0] + w[1] \cdot x[1] + \dots + w[i] \cdot x[i] + b \cdot 1 \tag{2}$$

- $x[i]$ are the feature(s) of the data (independent variables)
- $w[i]$ and b are parameters which are developed during training.
- For simple linear regression models with only one feature in the data, the formula looks like this

$$\hat{y} = w \cdot x + b \tag{3}$$

Generalization, Overfitting and Underfitting

Supervised Learning goal

- build a model on the training data and then be able to make accurate predictions on new, unseen data that has the same characteristics as the training set that we used.
- build a model that is able to **generalize**

Generalization

If a model is able to make accurate predictions on unseen data, we say it is able to *generalize* from the training data to the test data.

Generalization, Overfitting and Underfitting

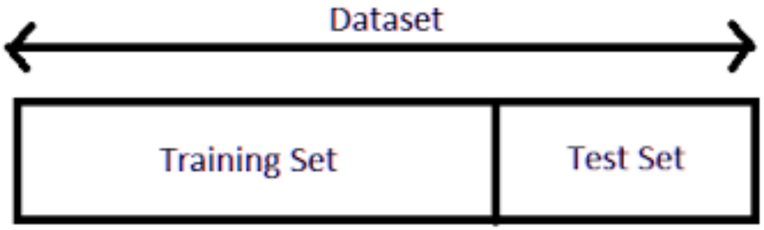
Accuracy

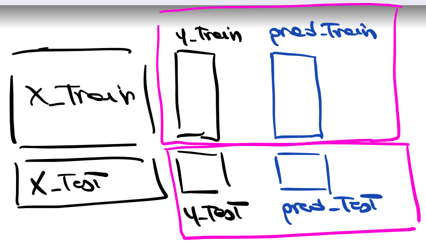
is the easiest way to access model performances.
Model accuracy is defined as the number of prediction correctly made by the models, divided by the total number of predictions made.

$$Accuracy = \frac{number\_of\_correct\_predictions}{total\_number\_of\_predicitons} = \frac{5}{7} \quad \begin{array}{ccc} y\_test & pred & \\ \begin{bmatrix} A \\ A \\ B \\ C \\ B \\ B \\ C \end{bmatrix} & \begin{bmatrix} A \\ A \\ A \\ A \\ B \\ B \\ C \end{bmatrix} & \begin{matrix} \checkmark \\ \checkmark \\ \times \\ \times \\ \checkmark \\ \checkmark \\ \checkmark \end{matrix} \end{array} \tag{4}$$

Evaluating a model

We build a model that can make accurate predictions on the training set.
If the training and test sets have enough in common, we expect the model to also be accurate on the test set.





Overfitting

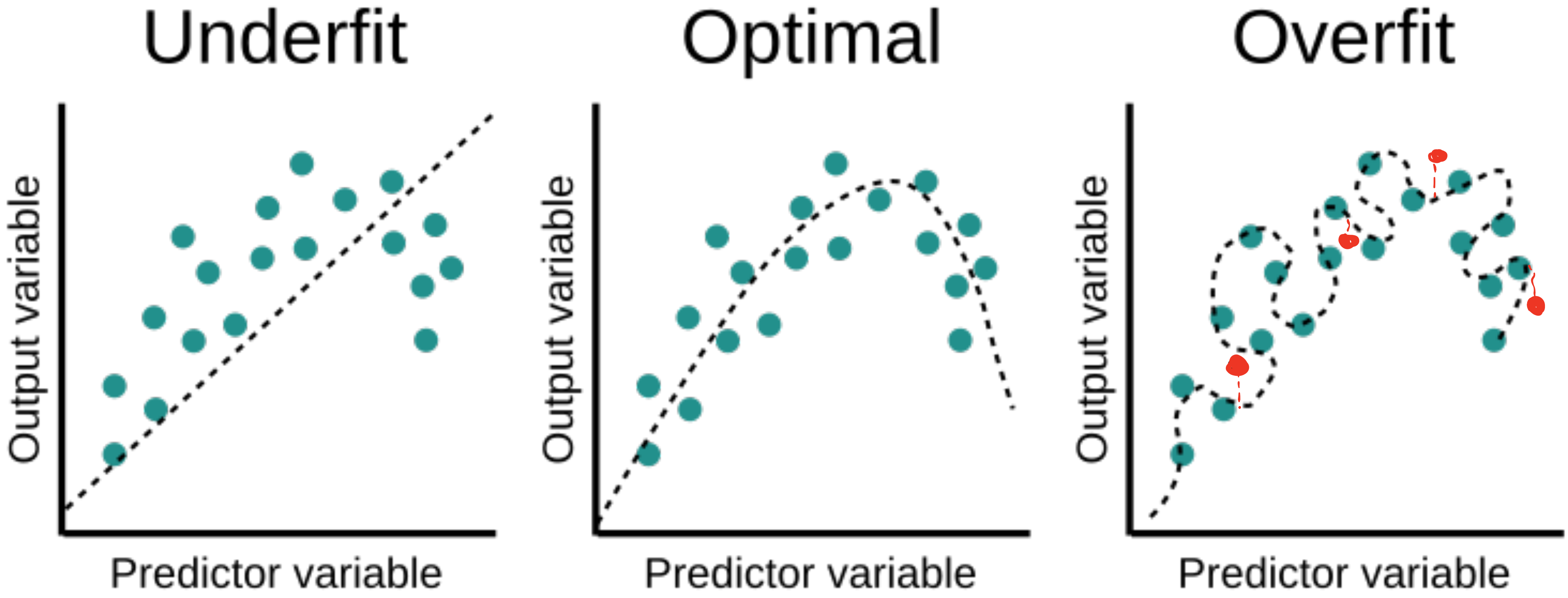
Issue with evaluation on training data

- If we allow ourselves to build very *complex models*, we can always be as accurate as we like on the training set
- The only measure of whether an algorithm will perform well on new data is the *evaluation on the test set*

Overfitting

- We expect simple models to generalize better to new data. Therefore, we always want to find the simplest model.
- Building a model that is too complex for the amount of information we have, is called **overfitting**

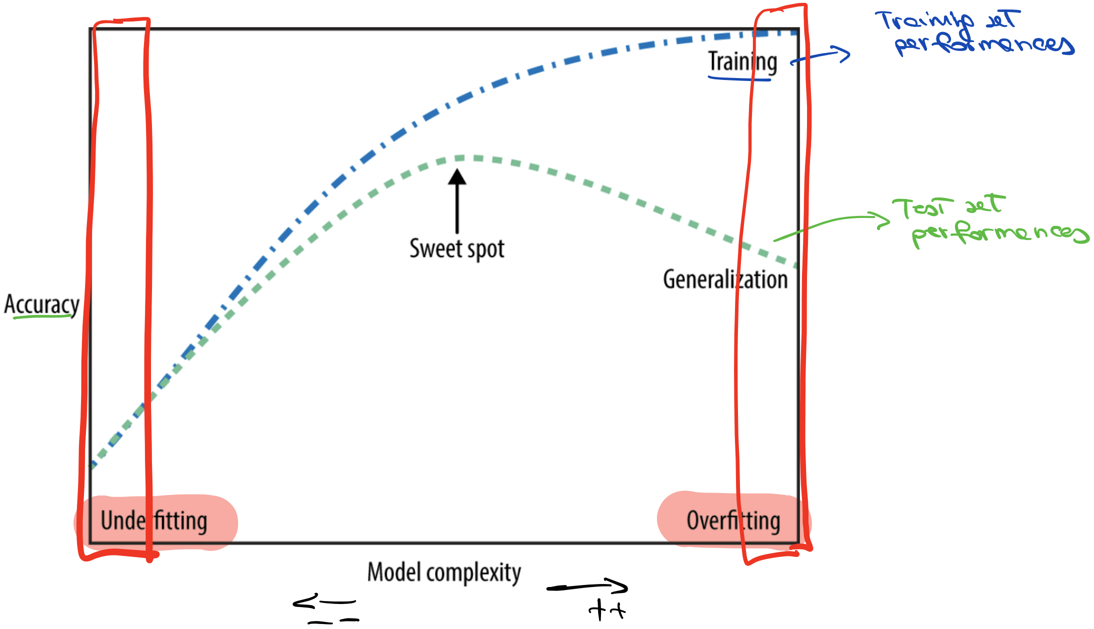
Overfitting and Underfitting



Underfitting

Underfitting

- If your model is too simple, then you might not be able to capture all the aspects of and variability in the data, and your model will do badly even on the training set.
- Choosing a too simple model is called **underfitting**.



There is a *trade-off* between **overfitting** and **underfitting** that yield the *best generalization performance*: this is the model we want to find

Model complexity VS Dataset Size

Model complexity is tied to the variation of inputs contained in the training dataset:

- the larger variety of data points your dataset contains, the more complex the model you can use without overfitting.
- Usually, collecting more data points will yield more variety, so larger datasets allow building more complex models
- However, simply duplicating the same data points or collecting very similar data will not help.

100 rows

1000 rows

References

Andreas C. Müller & Sarah Guido (2017)
Introduction to Machine Learning with Python
Published by O'Reilly Media, Inc.