ML: Lecture 1

Introduction to Machine Learning

Panagiotis Papapetrou, PhD
Professor, Stockholm University

Course logistics

- **Responsible teacher:**
  - Panagiotis Papapetrou - *panagiotis@dsv.su.se*

- **Course Assistant:**
  - Zed Lee - zed.lee@dsv.su.se

- **Schedule:**
  - 12 Lectures
  - 5 Lab sessions
  - Written Exam: Mar 20

- **Guest lecturers:**
  - Ioanna Miliou - ioanna.miliou@dsv.su.se
  - Tony Lindgren - tony@dsv.su.se
  - Sindri Magnusson - sindri.magnusson@dsv.su.se

- **Office hours: by appointment only**

Syllabus

Jan 16	**Introduction to machine learning**
Jan 18	Regression analysis
Jan 19	Laboratory session 1: numpy and linear regression
Jan 23	Ensemble learning
Jan 25	Deep learning I: Training neural networks
Jan 26	Laboratory session 2: ML pipelines, ensemble learning
Jan 30	Deep learning II: Convolutional neural networks
Feb 1	Laboratory session 3: training NNs and tensorflow
Feb 6	Deep learning III: Recurrent neural networks
Feb 8	Laboratory session 4: CNNs and RNs
Feb 13	Deep learning IV: Autoencoders, transformers, and attention
Feb 20	Time series classification

Syllabus

Feb 23	Laboratory session 5: Time series classification
Feb 27	Explainable machine learning
Mar 6	Reinforcement learning I
Mar 8	Reinforcement learning II
Mar 13	Exam review
Mar 20	**Examination**
Apr 28	**Re-take Examination**

Course workload

- Assignments 3hp
  - Three programming assignments (python and tensorflow)
- Written Exam 4.5hp

Assignments

- To be done individually
- Will involve python programming
- Five lab sessions for backup
- Tutoring sessions will be offered
- Grading scheme: Pass – Fail
- **To pass:** you should pass each assignment with **full points**

Exam

- Two parts:
  - Part A: multiple-choice questions
  - Part B: free-text questions
- It will examine both your knowlegde on several concepts discussed in the lectures are well as your critical ability on applying what you have learned
- May include some coding tasks
- To pass you need **at least 60% of the points**
- Grade scheme: A – F

Learning Objectives

- Describe, implement and apply machine learning methods and techniques for data exploration and data analysis
- Reason about the selected algorithms and solutions for machine learning
- Implement and apply machine learning methods to large and complex amounts of data
- Describe, implement and apply machine learning methods and techniques for evaluating results
- Become fluent in python and tensorflow

Textbooks

- **Elements of Statistical Learning**
  Edition: 2nd Publisher: Springer
  ISBN: 0387848576

- **Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems**
  Edition: Second Publisher: O'REILLY
  ISBN: 978-1492032649

- **Python Machine Learning**
  Edition: Third
  ISBN: 978-1-78995-575-0

Research papers (pointers will be provided)

Above all

- The goal of the course is to **learn and enjoy**

- The basic principle is to ask questions when you don't understand

- Say when things are unclear; not everything can be clear from the beginning

- **Participate** in the class as much as possible

Today

- What is machine learning?
- What are the different types of machine learning methods?
- What are examples of machine learning tasks in connection to the methods?
- What is the bias-variance trade-off?
- How to evaluate machine learning models?

Machine learning

“Learning is any process by which a system improves performance from experience.”

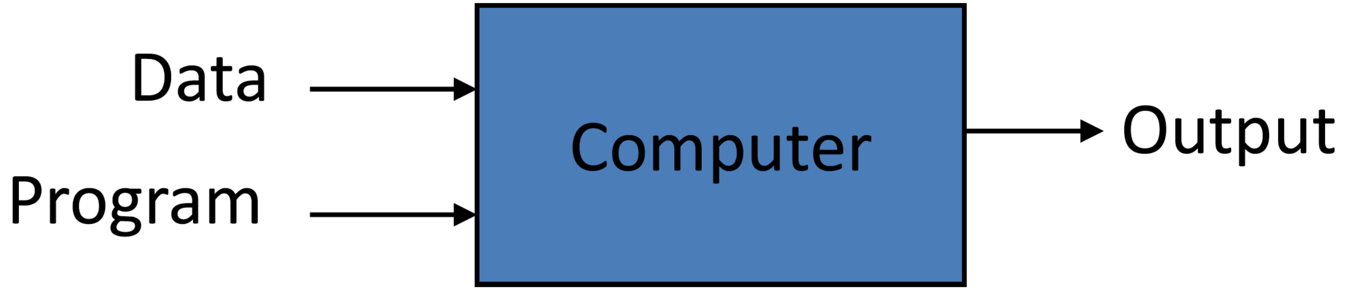
- Herbert Simon

“Leaerning is the study of algorithms that improve their performance with experience given a task.”

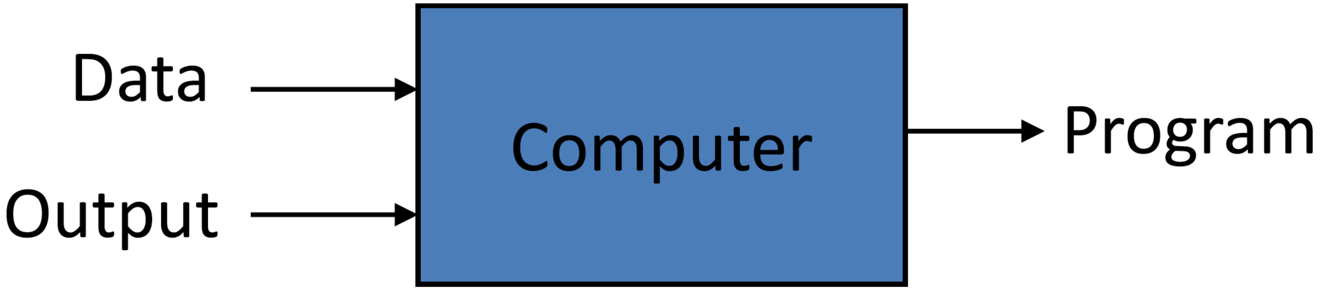
- Tom Michael Mitchell

Programming vs ML

Traditional Programming



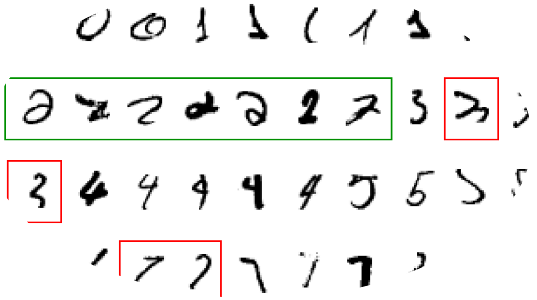
Machine Learning



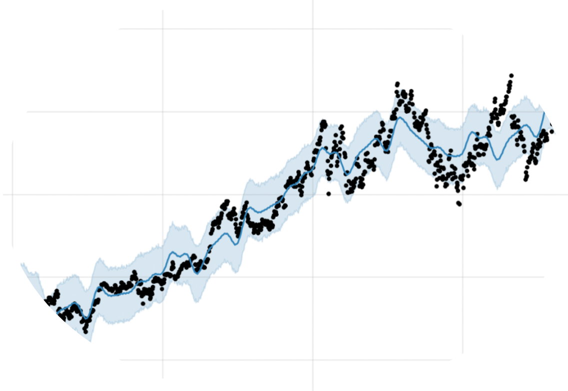
credits to Pedro Domingos

Examples of tasks solved by ML

- Pattern recognition:
  - facial expressions or identities
  - handwritten digits or spoken words
  - medical images
  - patient records



- Pattern generation:
  - generating motion sequences
- Anomaly detection:
  - money laundering

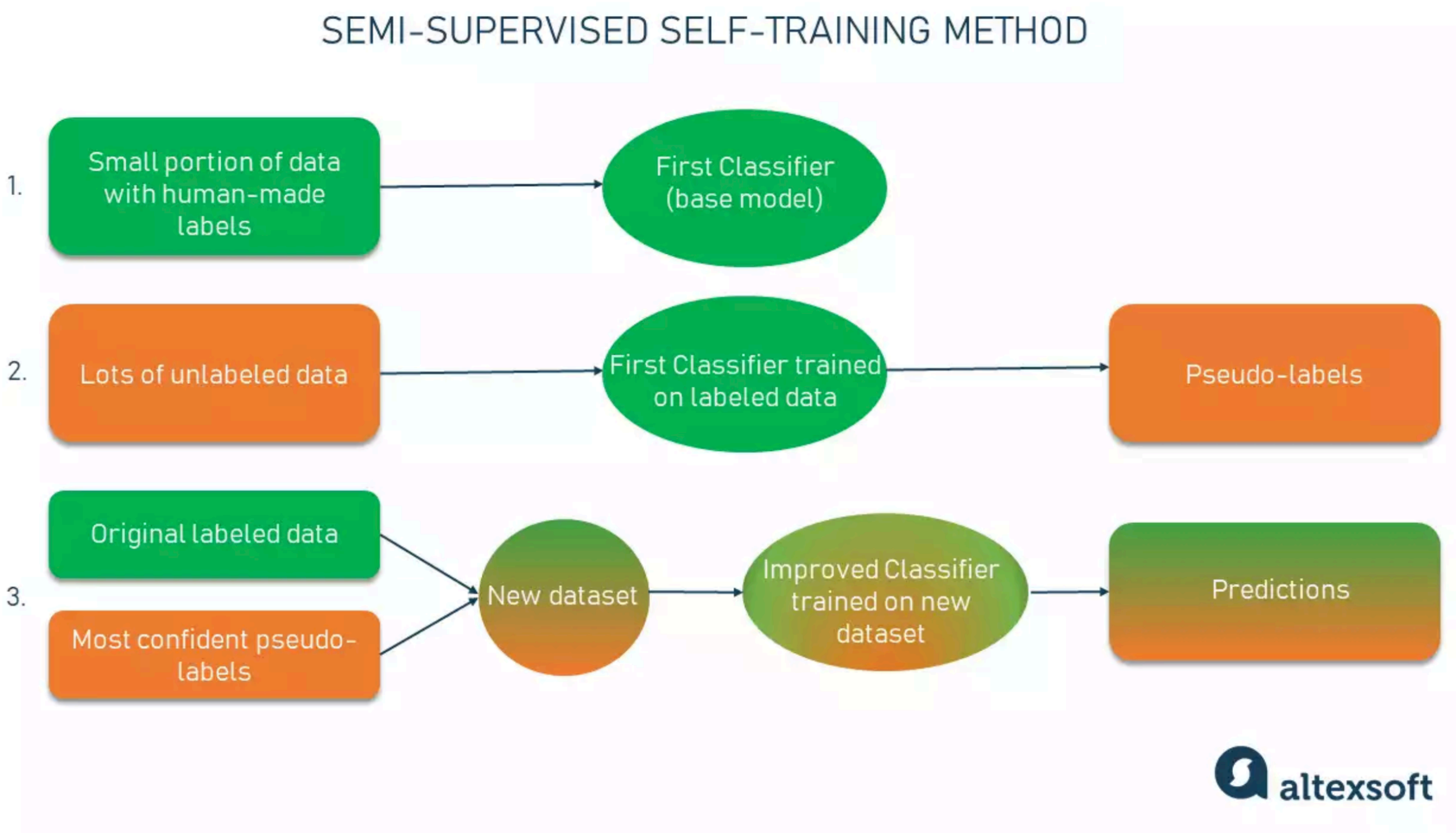


- Prediction:
  - future stock prices, weather phenomena, currency exchange rates

Types of learning

- **Supervised learning**
  - training data with desired output values that are provided before training
- **Unsupervised learning**
  - training data with no desired output values
- **Semi-supervised learning**
  - training data with a small amount of desired output values
  - iteratively train the model using the most confident predictions

Semi-supervised learning (self-learning)

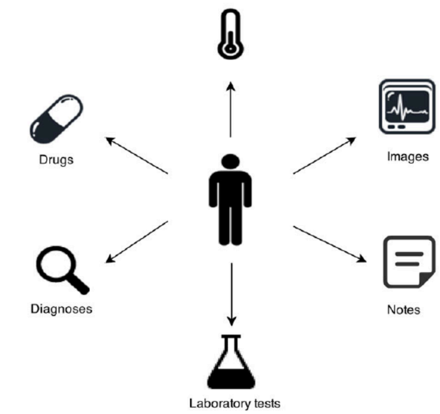


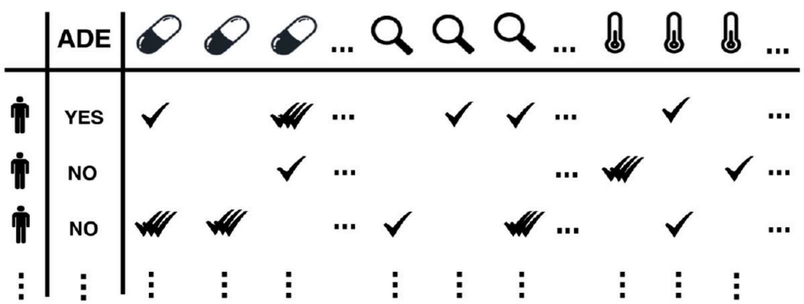
Types of learning

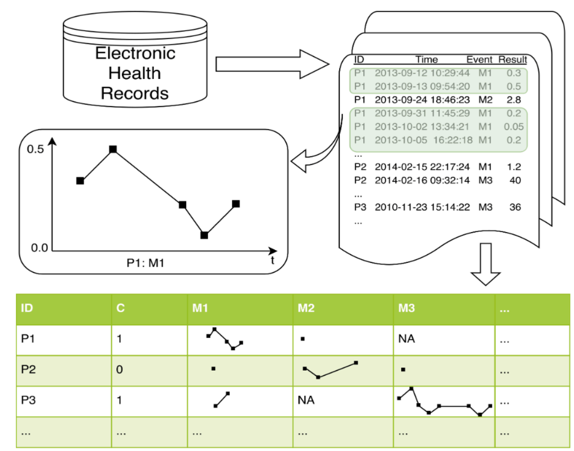
- **Active learning**
  - training data with and without labels
  - interactive labeling of training data by asking a human expert to label the least confident data samples
- **Federated learning**
  - training data is localized
  - learn a central model using distributed local data without sharing the data
- **Reinforcement learning**
  - an agent learns to interprets its environment by interacting with it through actions that are either rewarded or penalized

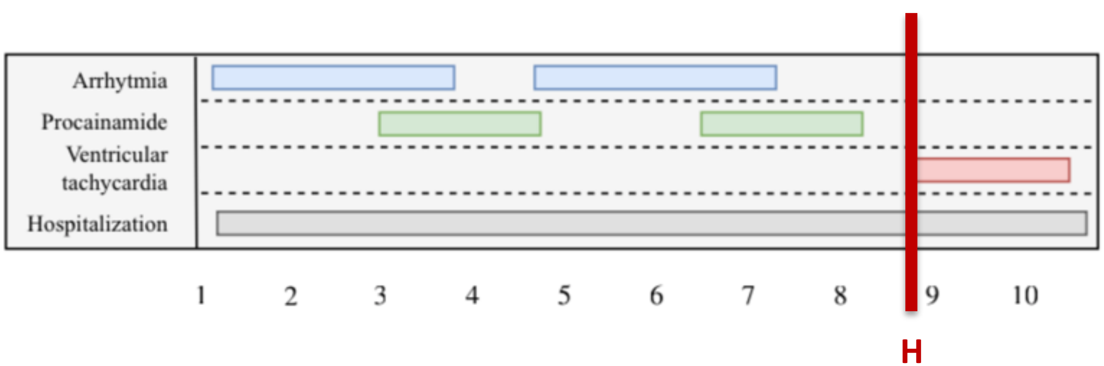
Examples of supervised learning

- Predict whether a patient, hospitalized due to heart attack, will have a second heart attack
- Identify the strongest predictors for heart attack
- Prediction based on many data modalities: demographics, diet, clinical measurements, clinical history



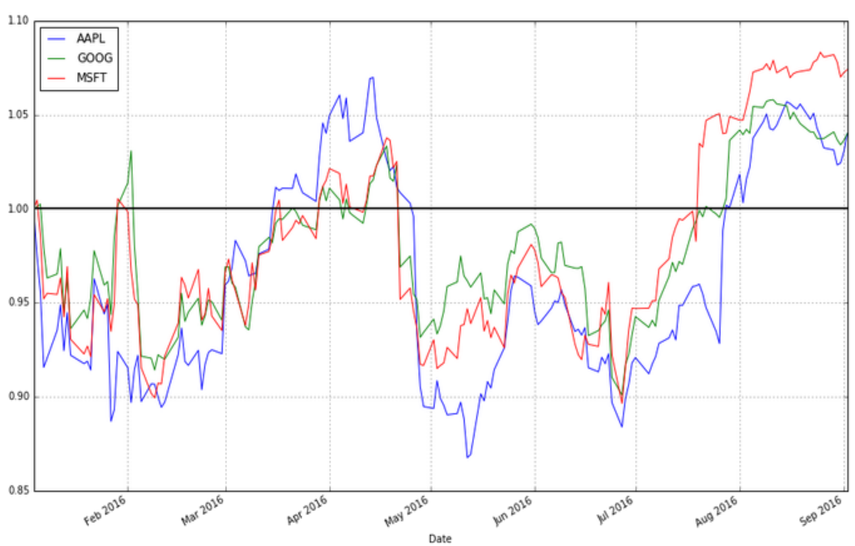


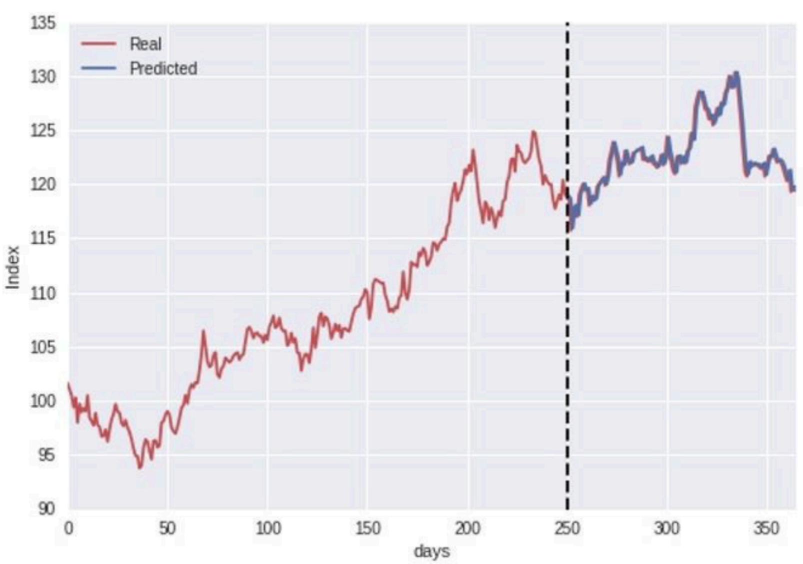




Examples of supervised learning

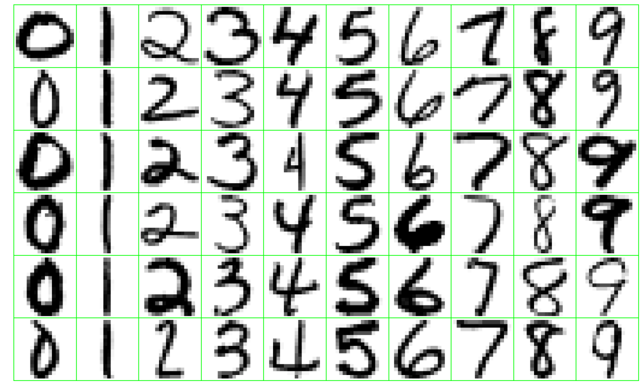
- Predict the price of a stock in 6 months from now, based on
  - the performance of the company (e.g., sales, future outlook)
  - other economic metrics (e.g., market value, Price-to-Earnings ratio)
  - other exogenous variables (e.g., social media)

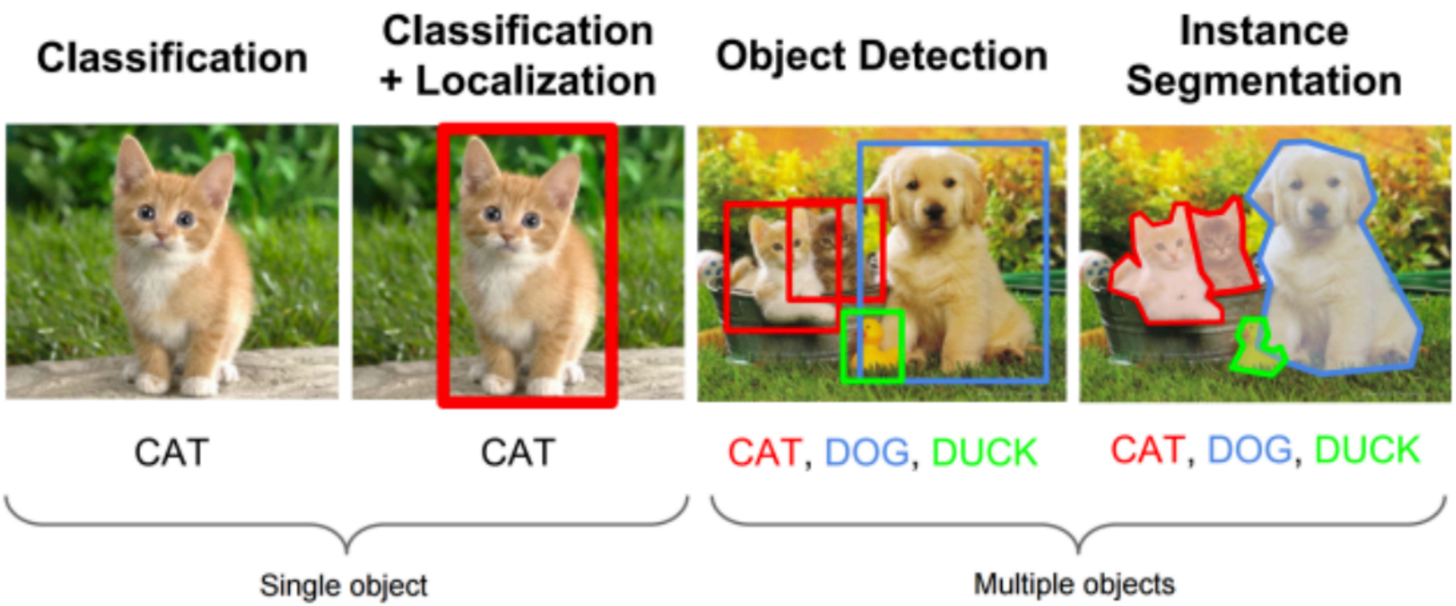




Examples of supervised learning

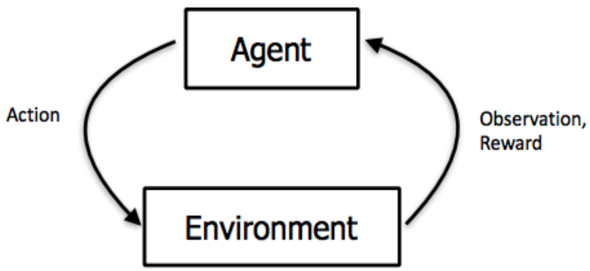
- Identify the handwritten numbers in a a digitized image
- Identify animals in an image

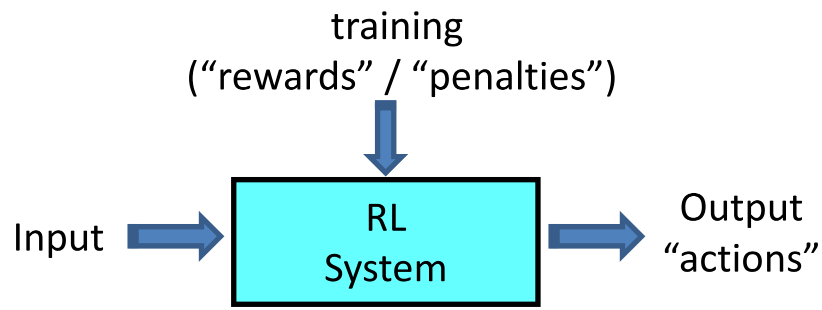


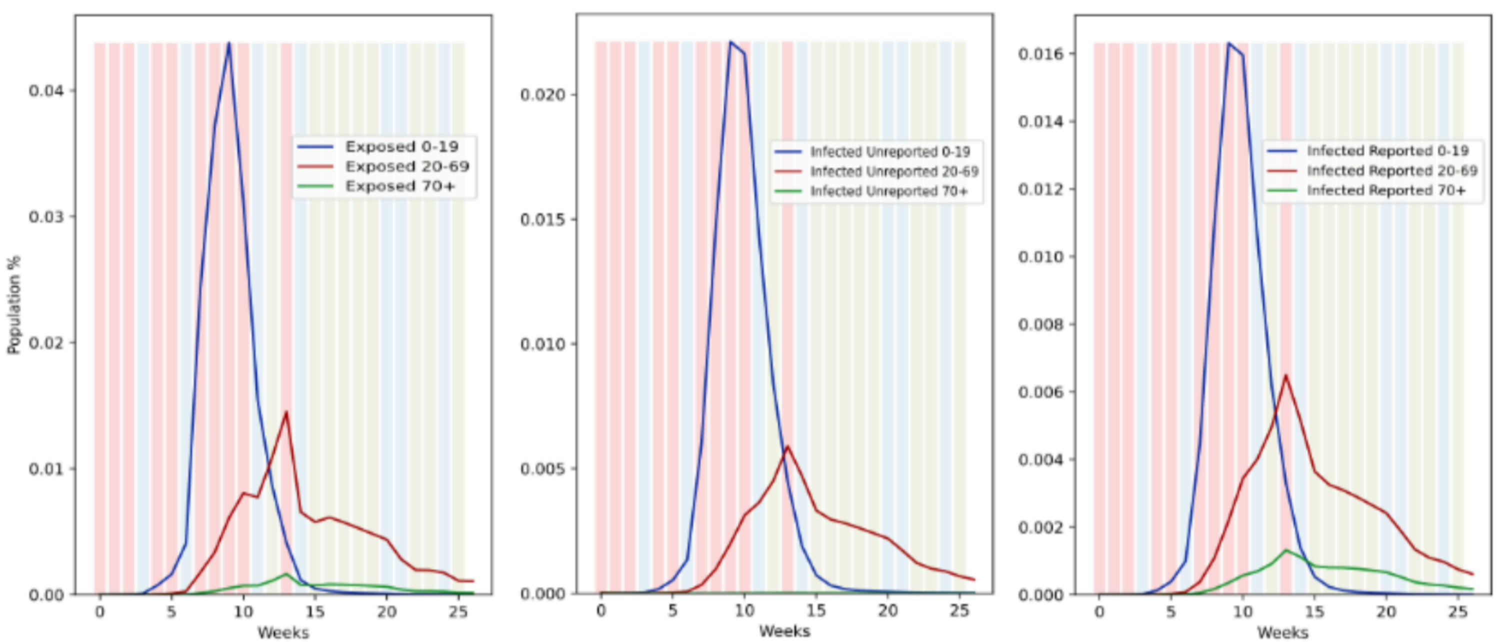


Examples of reinforcement learning

Define the measures that need to be taken by the public health authorities under a pandemic situation

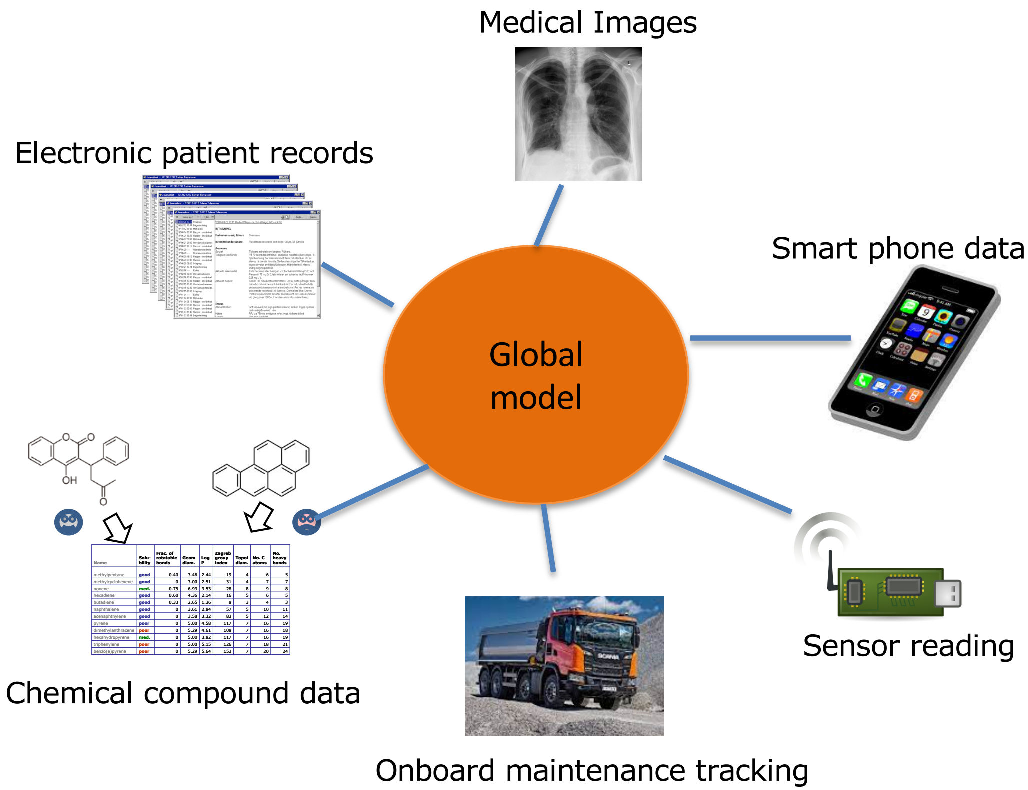


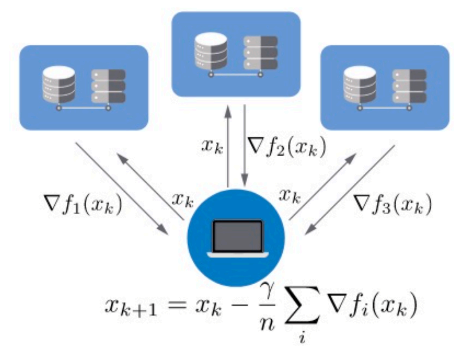




Examples of federated learning

Given data from distributed hospitals learn a central model that can propose the optimal patient treatment without sharing any data





The classical learning setup

Attributes (or variables)
categorical, discrete, ordinal (qualitative)
numercial (quantitative)
denoted as ***X***

Input
predictors or
independent variables

→ Machine Learning function *f* →

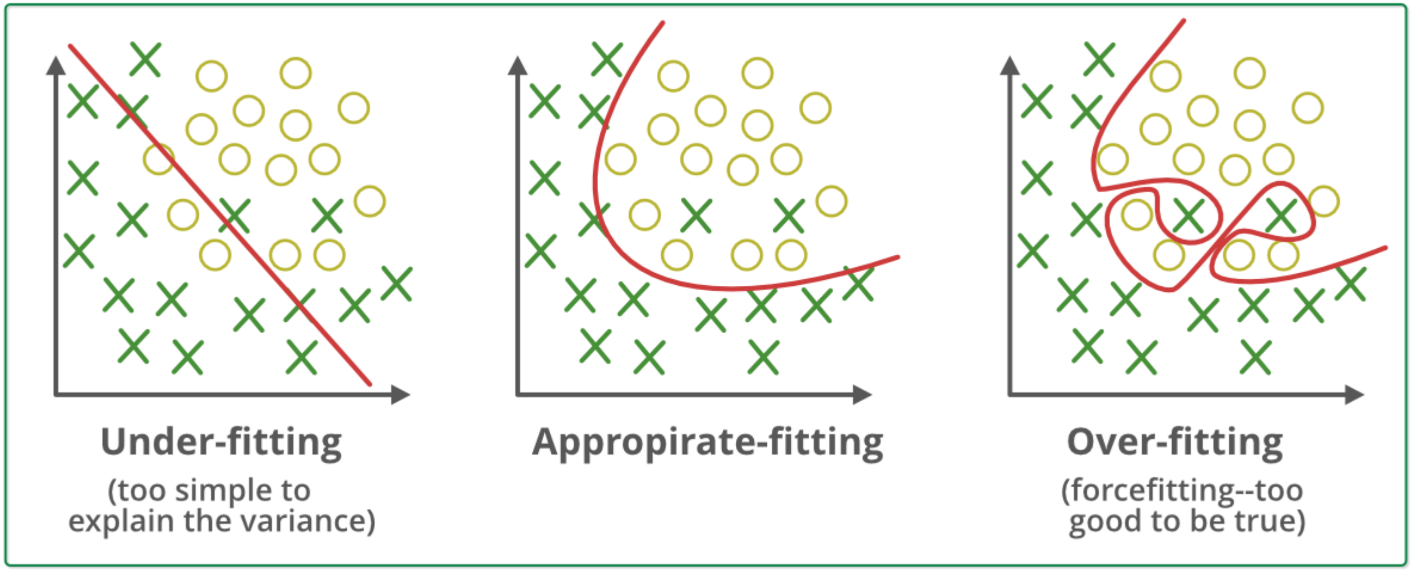
Output
responses or
dependent variables

denoted as ***Y*** (quantitative or qualitative)

Classification is the task of ***learning a target function*** ***f*** that maps an input (attribute set) ***X*** to one of the predefined class labels ***Y***

Regression is the task of ***learning a target function*** ***f*** that maps an input (attribute set) ***X*** to a value in a range of real values ***Y***

How good is f?



What determines a **good fit**?

The bias-variance decomposition

- Assume the data complies to the following statistical model:

$$Y = f(X) + \varepsilon$$

- The relationship between X and Y is *non-deterministic (noise)*
- But we assume:

$$\mathrm{E}(\varepsilon) = 0 \qquad \mathrm{Var}(\varepsilon) = \sigma_\varepsilon^2$$

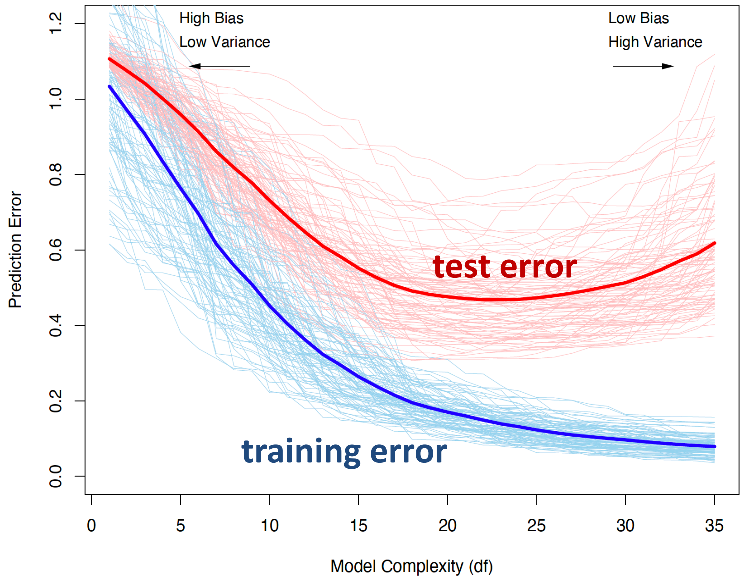
- We also assume that f is fixed and unknown

- The metric to optimize is MSE (mean squared error):

$$\mathrm{MSE} = \mathbb{E}[(y - \hat{f}(x))^2]$$

How good is f?

the U-Shape effect



training error

$$\overline{\text{err}} = \frac{1}{N}\sum_{i=1}^{N} L(y_i, \hat{f}(x_i))$$

- **Training error:** not a good estimate of the test error; why?
  - as model complexity increases, the training error drops to zero
  - **overfitting!**

The bias-variance decomposition

- Bias: over different realizations of the training set, how much does $\hat{f}(X)$ change:

$$\text{bias}[\hat{f}(x)] = \mathbb{E}[\hat{f}(x)] - f(x)$$

- Variance: over different realizations of the training set, how much does $\hat{f}(X)$ deviate from its expectation:

$$\text{var}(\hat{f}(x)) = \mathbb{E}[(\hat{f}(x) - \mathbb{E}[\hat{f}(x)])^2]$$

- **The bias-variance decomposition:**

$$\mathbb{E}[\mathbb{E}[(y - \hat{f}(x))^2]] = \mathbb{E}[\text{bias}[\hat{f}(x)]^2] + \mathbb{E}[\text{var}(\hat{f}(x))] + \sigma_\epsilon^2$$

Proof

$$\mathrm{MSE} = \mathbb{E}[(y - \hat{f}(x))^2]$$

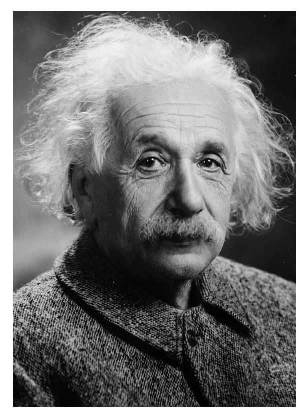
Proof

$$\mathrm{MSE} = \mathbb{E}[(y - \hat{f}(x))^2]$$

$$\mathbb{E}[(y - \hat{f}(x))^2] = \mathbb{E}[(f(x) + \epsilon - \hat{f}(x))^2] \tag{1}$$

Recall:

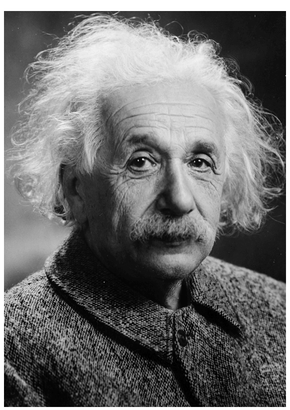
$$y = f(x) + \epsilon$$



Proof

$$\text{MSE} = \mathbb{E}[(y - \hat{f}(x))^2]$$

$$\begin{aligned}\mathbb{E}[(y - \hat{f}(x))^2] &= \mathbb{E}[(f(x) + \epsilon - \hat{f}(x))^2] \\ &= \mathbb{E}[(f(x) - \hat{f}(x))^2] + \mathbb{E}[\epsilon^2] + 2\mathbb{E}[(f(x) - \hat{f}(x))\epsilon]\end{aligned} \tag{1}$$



- square expansion
- linear property of expectation
- independence of random variables ϵ and $\hat{f}$

Proof

$$\text{MSE} = \mathbb{E}[(y - \hat{f}(x))^2]$$

$$\begin{aligned}
\mathbb{E}[(y - \hat{f}(x))^2] &= \mathbb{E}[(f(x) + \epsilon - \hat{f}(x))^2] && (1) \\
&= \mathbb{E}[(f(x) - \hat{f}(x))^2] + \mathbb{E}[\epsilon^2] + 2\mathbb{E}[(f(x) - \hat{f}(x))\epsilon] \\
&= \mathbb{E}[(f(x) - \hat{f}(x))^2] + \underbrace{\mathbb{E}[\epsilon^2]}_{=\sigma_\epsilon^2} + 2\mathbb{E}[(f(x) - \hat{f}(x))]\underbrace{\mathbb{E}[\epsilon]}_{=0} && (2)
\end{aligned}$$

Proof

$$\mathrm{MSE} = \mathbb{E}[(y - \hat{f}(x))^2]$$

$$\begin{aligned}
\mathbb{E}[(y - \hat{f}(x))^2] &= \mathbb{E}[(f(x) + \epsilon - \hat{f}(x))^2] && (1) \\
&= \mathbb{E}[(f(x) - \hat{f}(x))^2] + \mathbb{E}[\epsilon^2] + 2\mathbb{E}[(f(x) - \hat{f}(x))\epsilon] \\
&= \mathbb{E}[(f(x) - \hat{f}(x))^2] + \underbrace{\mathbb{E}[\epsilon^2]}_{=\sigma_\epsilon^2} + 2\mathbb{E}[(f(x) - \hat{f}(x))] \underbrace{\mathbb{E}[\epsilon]}_{=0} && (2) \\
&= \boxed{\mathbb{E}[(f(x) - \hat{f}(x))^2]} + \sigma_\epsilon^2 && (3)
\end{aligned}$$

Let us expand this now

$$\mathbb{E}[(f(x) - \hat{f}(x))^2] = \mathbb{E}\left[\Big((f(x) - \mathbb{E}[\hat{f}(x)]) - (\hat{f}(x) - \mathbb{E}[\hat{f}(x)])\Big)^2\right] \qquad (4)$$

Continuing the decomposition

$$\mathbb{E}[(f(x)-\hat{f}(x))^2] = \mathbb{E}\left[\Big((f(x)-\mathbb{E}[\hat{f}(x)]) - (\hat{f}(x)-\mathbb{E}[\hat{f}(x)])\Big)^2\right] \qquad (4)$$

Continuing the decomposition

$$\mathbb{E}[(f(x)-\hat{f}(x))^2] = \mathbb{E}\left[\left((f(x)-\mathbb{E}[\hat{f}(x)]) - (\hat{f}(x)-\mathbb{E}[\hat{f}(x)])\right)^2\right] \tag{4}$$

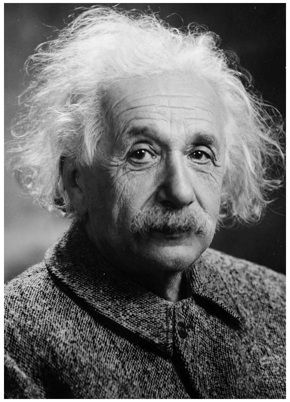
$$\begin{aligned} &= \mathbb{E}\left[\left(\mathbb{E}[\hat{f}(x)] - f(x)\right)^2\right] + \mathbb{E}\left[\left(\hat{f}(x)-\mathbb{E}[\hat{f}(x)]\right)^2\right] \\ &\quad - 2\mathbb{E}\left[\left(f(x)-\mathbb{E}[\hat{f}(x)]\right)\left(\hat{f}(x)-\mathbb{E}[\hat{f}(x)]\right)\right] \end{aligned} \tag{5}$$

Continuing the decomposition

$$\mathbb{E}[(f(x)-\hat{f}(x))^2] = \mathbb{E}\left[\left((f(x)-\mathbb{E}[\hat{f}(x)]) - (\hat{f}(x)-\mathbb{E}[\hat{f}(x)])\right)^2\right] \tag{4}$$

$$= \mathbb{E}\left[\left(\mathbb{E}[\hat{f}(x)] - f(x)\right)^2\right] + \mathbb{E}\left[\left(\hat{f}(x)-\mathbb{E}[\hat{f}(x)]\right)^2\right] - 2\mathbb{E}\left[\left(f(x)-\mathbb{E}[\hat{f}(x)]\right)\left(\hat{f}(x)-\mathbb{E}[\hat{f}(x)]\right)\right] \tag{5}$$

$$= \underbrace{(\mathbb{E}[\hat{f}(x)] - f(x))^2}_{=\mathrm{bias}[\hat{f}(x)]} + \underbrace{\mathbb{E}\left[\left(\hat{f}(x)-\mathbb{E}[\hat{f}(x)]\right)^2\right]}_{=\mathrm{var}(\hat{f}(x))} - 2\left(f(x)-\mathbb{E}[\hat{f}(x)]\right)\mathbb{E}\left[\left(\hat{f}(x)-\mathbb{E}[\hat{f}(x)]\right)\right] \tag{6}$$

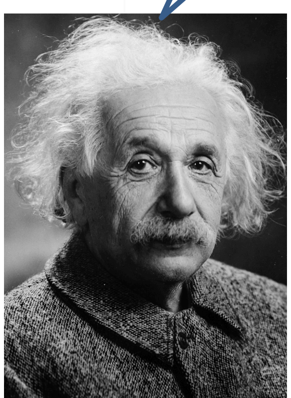


$\mathbb{E}[\hat{f}(x)] - f(x)$: is a constant, hence the expectation is also constant

Continuing the decomposition

$$
\begin{aligned}
\mathbb{E}[(f(x)-\hat{f}(x))^2] &= \mathbb{E}\left[\left((f(x)-\mathbb{E}[\hat{f}(x)]) - (\hat{f}(x)-\mathbb{E}[\hat{f}(x)])\right)^2\right] && (4)\\
&= \mathbb{E}\left[\left(\mathbb{E}[\hat{f}(x)] - f(x)\right)^2\right] + \mathbb{E}\left[\left(\hat{f}(x)-\mathbb{E}[\hat{f}(x)]\right)^2\right] \\
&\quad - 2\mathbb{E}\left[\left(f(x)-\mathbb{E}[\hat{f}(x)]\right)\left(\hat{f}(x)-\mathbb{E}[\hat{f}(x)]\right)\right] && (5)\\
&= \underbrace{(\mathbb{E}[\hat{f}(x)] - f(x))^2}_{=\mathrm{bias}[\hat{f}(x)]} + \underbrace{\mathbb{E}\left[\left(\hat{f}(x)-\mathbb{E}[\hat{f}(x)]\right)^2\right]}_{=\mathrm{var}(\hat{f}(x))} \\
&\quad - 2\left(f(x)-\mathbb{E}[\hat{f}(x)]\right)\mathbb{E}\left[\left(\hat{f}(x)-\mathbb{E}[\hat{f}(x)]\right)\right] && (6)\\
&= \mathrm{bias}[\hat{f}(x)]^2 + \mathrm{var}(\hat{f}(x)) \\
&\quad - 2\left(f(x)-\mathbb{E}[\hat{f}(x)]\right)\left(\mathbb{E}[\hat{f}(x)]-\mathbb{E}[\hat{f}(x)]\right) && (7)
\end{aligned}
$$

linearity of expactation is applied



Continuing the decomposition

$$\begin{aligned}
\mathbb{E}[(f(x)-\hat{f}(x))^2] &= \mathbb{E}\left[\left((f(x)-\mathbb{E}[\hat{f}(x)]) - (\hat{f}(x)-\mathbb{E}[\hat{f}(x)])\right)^2\right] && (4)\\
&= \mathbb{E}\left[\left(\mathbb{E}[\hat{f}(x)] - f(x)\right)^2\right] + \mathbb{E}\left[\left(\hat{f}(x)-\mathbb{E}[\hat{f}(x)]\right)^2\right] \\
&\quad - 2\mathbb{E}\left[\left(f(x)-\mathbb{E}[\hat{f}(x)]\right)\left(\hat{f}(x)-\mathbb{E}[\hat{f}(x)]\right)\right] && (5)\\
&= \underbrace{(\mathbb{E}[\hat{f}(x)] - f(x))^2}_{=\mathrm{bias}[\hat{f}(x)]} + \underbrace{\mathbb{E}\left[\left(\hat{f}(x)-\mathbb{E}[\hat{f}(x)]\right)^2\right]}_{=\mathrm{var}(\hat{f}(x))} \\
&\quad - 2\left(f(x)-\mathbb{E}[\hat{f}(x)]\right)\mathbb{E}\left[\left(\hat{f}(x)-\mathbb{E}[\hat{f}(x)]\right)\right] && (6)\\
&= \mathrm{bias}[\hat{f}(x)]^2 + \mathrm{var}(\hat{f}(x)) \\
&\quad - 2\left(f(x)-\mathbb{E}[\hat{f}(x)]\right)\left(\mathbb{E}[\hat{f}(x)]-\mathbb{E}[\hat{f}(x)]\right) && (7)\\
&= \mathrm{bias}[\hat{f}(x)]^2 + \mathrm{var}(\hat{f}(x)) && (8)
\end{aligned}$$

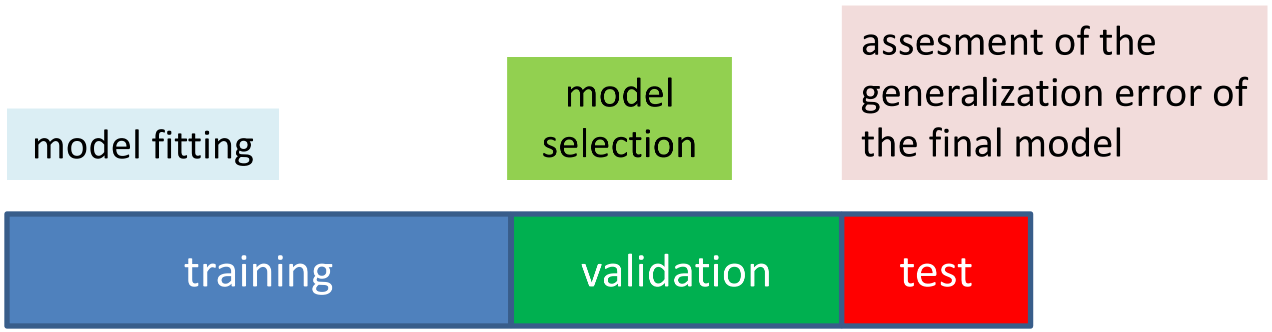
$$\boxed{\mathbb{E}[(y-\hat{f}(x))^2] = \mathrm{bias}[\hat{f}(x)]^2 + \mathrm{var}(\hat{f}(x)) + \sigma_\epsilon^2}$$

Model selection & assessment

Model selection: estimating the performance of different models in order to choose the best one.

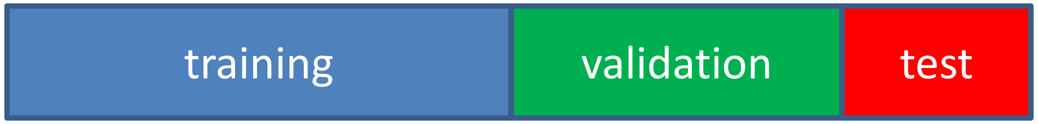
Model assessment: having chosen a final model, estimating its prediction error (generalization error) on new data.

Ideal scenario: plenty of data, so divide it into three compartments: (1) **training**, (2) **validation**, (3) **test**



Hold out & subsampling

- **Holdout (or using a validation set)**
  - reserve **50%** for training, **25%** for validation, and **25%** for testing
- **Random subsampling**
  - repeated holdout

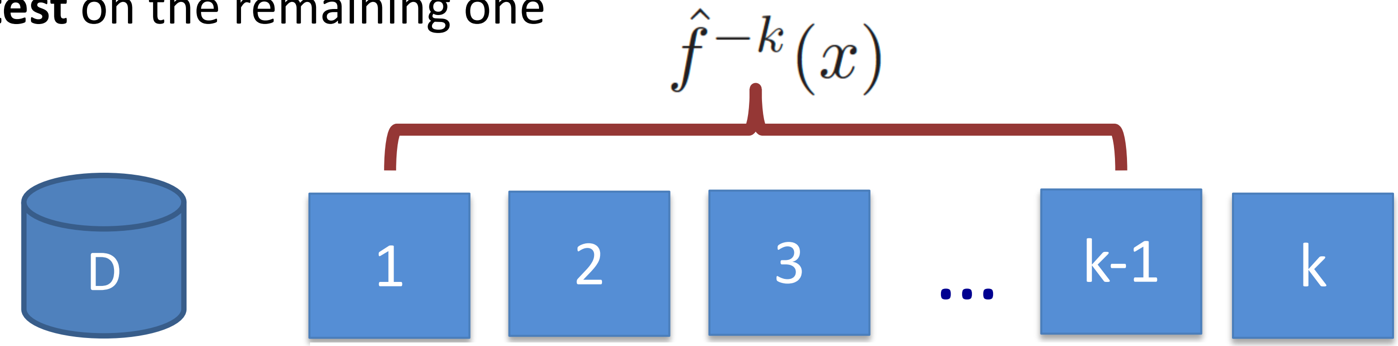


- **Stratified subsampling**
  - preserve the class balance in the samples

- Shortcoming:
  - depends highly on which data points end up in the training set and which end up in the test set

k-fold Cross Validation

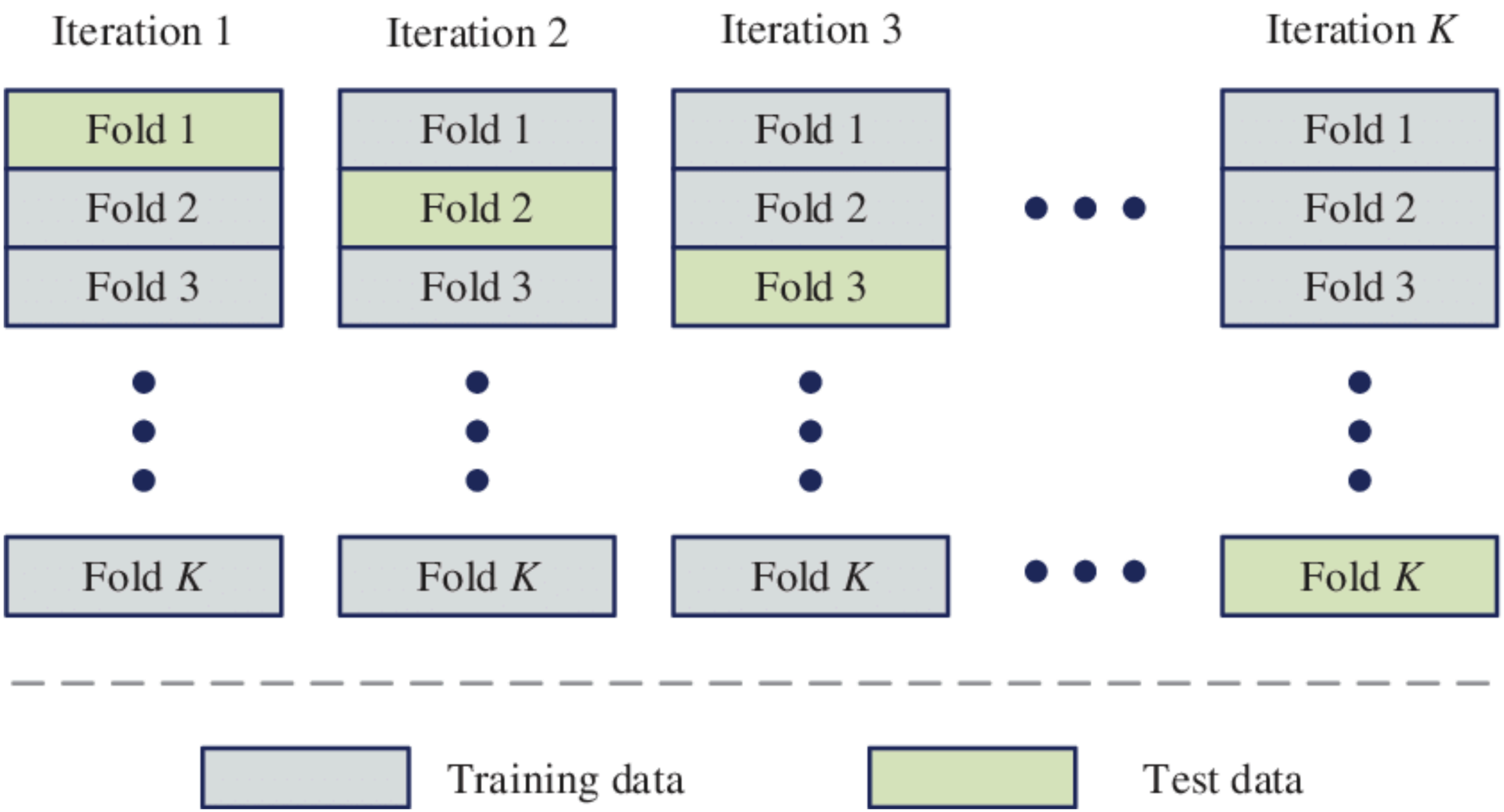
- Partition the ***N*** data samples into ***k*** disjoint subsets
- ***k***-fold:
  - **train** on ***k-1*** partitions
  - **test** on the remaining one



$$\mathrm{CV}(\hat{f}) = \frac{1}{N}\sum_{i=1}^{N} L(y_i, \hat{f}^{-\kappa(i)}(x_i))$$

- Each subset is given a chance to be in the test set
- Performance measure is averaged over the ***k*** iterations
- If ***k=N*** then leave-one-out Cross Validation

k-fold Cross Validation



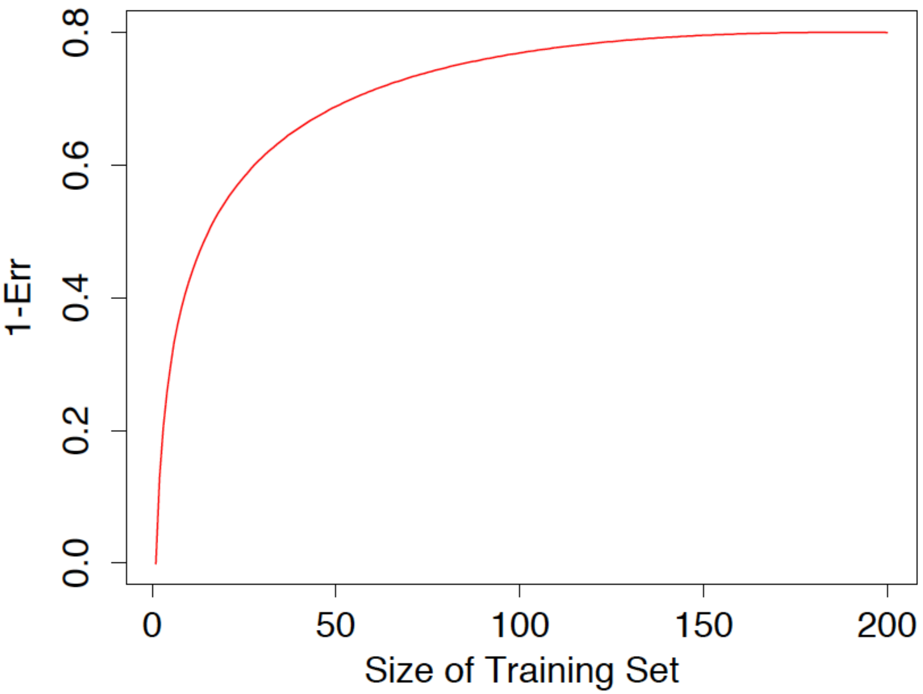
k-fold vs LOOCV

- **LOOCV:**
  - the test error is approximately unbiased to the true (expected) prediction error; training on “almost” the whole training set
  - training on almost identical training sets produces models that are highly correlated
  - the computational burden will be very high: the training algorithm has to be rerun from scratch *N* times
- **k-fold:**
  - less correlated models
  - the computational time is lower
  - the bias of the procedure increases as the training set will contain fewer examples
  - the lower the *k* the higher the bias!

Learning Curve

Two observations:

- **performance** improves as we move to 100 examples
- increasing to 200 brings a very small benefit



What **training set size** works for 5-fold CV?

What if training set = 200 examples?

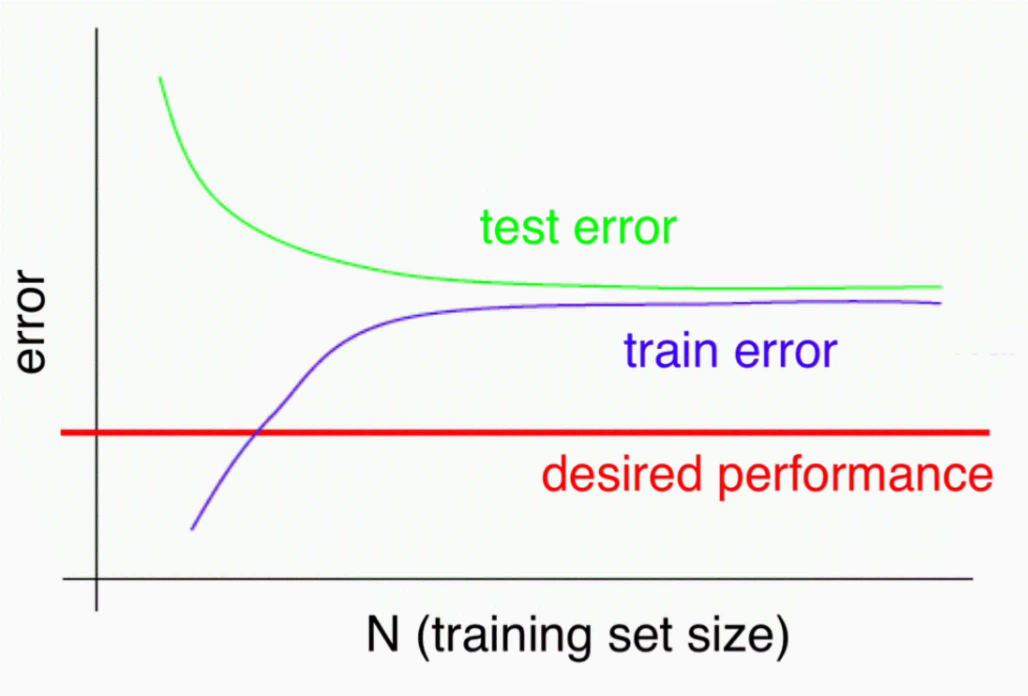
- **5-fold:** training sets of size **160**
- Same as the performance for **200** examples; hence **no bias**

What if training set = 50 examples?

- **5-fold:** training sets of size **40**
- Error will be overestimated; hence **bias**

Comparing Learning Curves

- Indication of high bias



As # of training examples increases:
- training error increases too
- model is not fitting the data with an appropriate curve

Bias leads to comparable errors in cross-validation and training

Adding more data does not help!

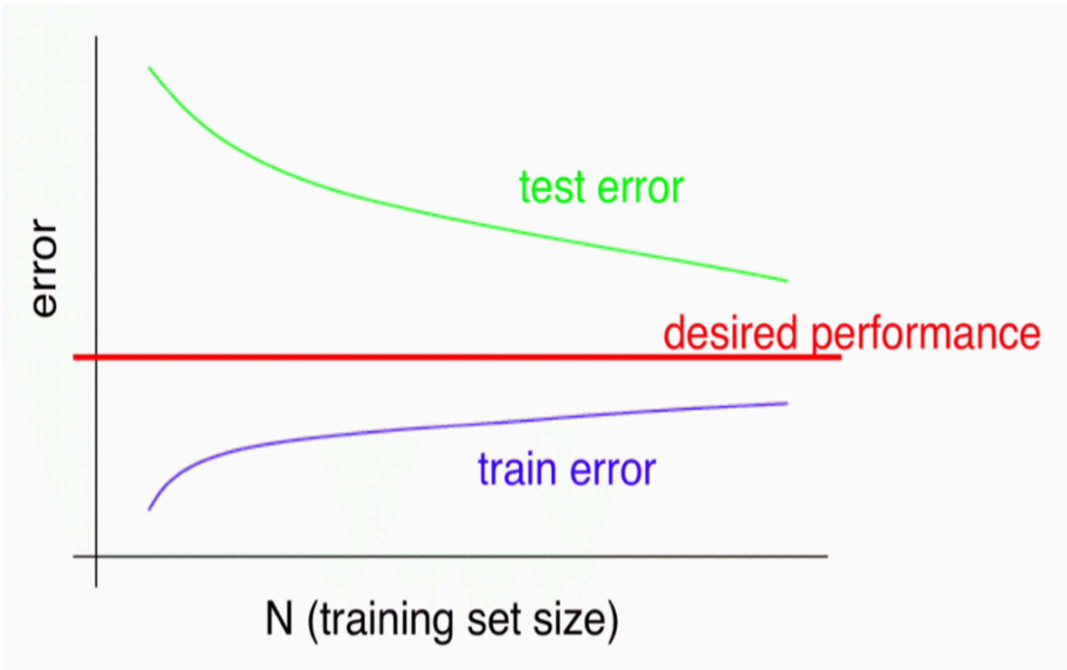
What should one do to reduce bias?

increase the number of features

add more complex features

Comparing Learning Curves

- Indication of high variance



Training error grows slowly but well within the desired range

High variance leads to overfitting and hence high test error

The model is forced to learn more generalized properties

Adding more examples reduces test error and the gap between the curves closes down

What should one do to reduce variance?

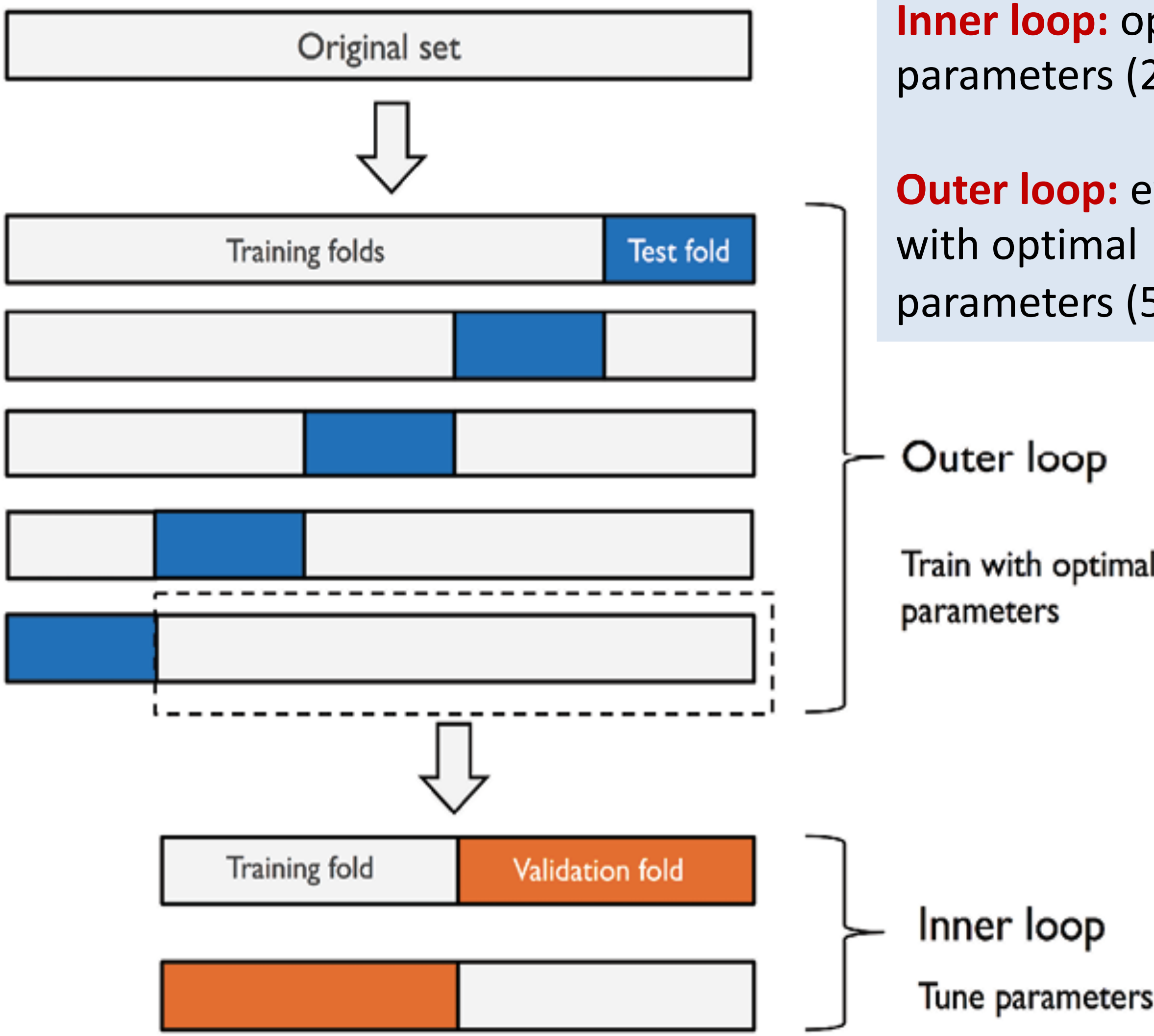
get more data

reduce the number of features

k-fold Cross Validation: pitfall

- Assume a dataset with a large number of predictors (variables)
- You would like to estimate the error of a machine learning model
- Solution strategy:
  - find a subset of "good" predictors with strong correlation to the class label(s) using the whole dataset
  - build a classification model on these predictors using Cross Validation and estimate the tuning parameters of the model as well as its prediction error
  - estimate the prediction error of the final model as the average of all runs
- **Anything wrong?**
  - data leakage!
  - Feature selection should be performed in each fold

Nested Cross Validation (5x2 CV)



Inner loop: optimize parameters (2 splits)

Outer loop: evaluate with optimal parameters (5 splits)

Which model do we finally deploy?

Cross-validation tests a model building procedure and not a single model instance

We **deploy** the feature set obtained by applying the winning procedure to the whole dataset

Bootstrapping

- Create ***B*** replications (of the same size) of a training set of size ***N*** by selecting examples uniformly *with replacement*

$$\begin{aligned} \Pr\{\text{observation } i \in \text{bootstrap sample } b\} &= 1 - \left(1 - \frac{1}{N}\right)^N \\ &\approx 1 - e^{-1} \\ &= 0.632 \end{aligned}$$

Predicted value of model built on the *b^{th}* replication (i.e., bootstrap sample)

$$\widehat{\mathrm{Err}}_{\mathrm{boot}} = \frac{1}{B}\frac{1}{N}\sum_{b=1}^{B}\sum_{i=1}^{N} L(y_i, \hat{f}^{*b}(x_i))$$

- Train on each replication ***b***
- Test on **the whole training set**

Any ***problem*** with that?

Leave-one-out bootstrap

- Train on each replication ***b***
- Test each sample **not in *b***, i.e., *the out-of-bag set **C^{-i}*** against all replications it does not appear in

$$\widehat{\mathrm{Err}}^{(1)} = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{|C^{-i}|}\sum_{b\in C^{-i}} L(y_i, \hat{f}^{*b}(x_i))$$

Leave-one-out bootstrap

- Train on each replication ***b***
- Test each sample **not in *b***, i.e., *the out-of-bag set* C^{-i} against all replications it does not appear in

$$\widehat{\mathrm{Err}}^{(1)} = \frac{1}{N} \sum_{i=1}^{N} \frac{1}{|C^{-i}|} \sum_{b \in C^{-i}} L(y_i, \hat{f}^{*b}(x_i))$$

- **Problems?**
  - the average number of distinct observations in each bootstrap sample is about *0.632 N*
  - its bias will roughly behave like that of two-fold Cross Validation
  - if the learning curve has considerable slope at sample size *N/2*, the leave-one out bootstrap will be an overestimate of the true error

.632 bootstrap

- **Main idea:**
  - extension of leave-one-out bootstrap
  - **intuition:** pulls the leave-one out bootstrap estimate down towards the training error rate, and hence reduces its upward bias

$$\widehat{\mathrm{Err}}^{(.632)} = .368 \cdot \overline{\mathrm{err}} + .632 \cdot \widehat{\mathrm{Err}}^{(1)}$$

training error

Leave-one-out bootstrap error

Coming up next…

Jan 16	Introduction to machine learning
Jan 18	**Regression analysis**
Jan 19	Laboratory session 1: numpy and linear regression
Jan 23	Ensemble learning
Jan 25	Deep learning I: Training neural networks
Jan 26	Laboratory session 2: ML pipelines, ensemble learning
Jan 30	Deep learning II: Convolutional neural networks
Feb 1	Laboratory session 3: training NNs and tensorflow
Feb 6	Deep learning III: Recurrent neural networks
Feb 8	Laboratory session 4: CNNs and RNs
Feb 13	Deep learning IV: Autoencoders, transformers, and attention
Feb 20	Time series classification

TODOs

- **Reading:**
  - Main course book (ESL): Chapters 1, 2, and 7
- **Prepare for this week's lab:**
  - Start reading HML: Chapter 4