

# Machine Learning

## Lecture 1 - Introduction to ML, $k$ -NN and Decision Trees

**Mattias Villani**

Department of Statistics  
Stockholm University

Department of Computer and Information Science  
Linköping University



[mattiasvillani.com](http://mattiasvillani.com)



@matvil



[mattiasvillani](http://mattiasvillani)

# Course overview

- Course [webpage](#). Course [syllabus](#).
- Modes of teaching:
  - ▶ Lectures 1-10 ([Mattias Villani](#))
  - ▶ Lectures 11-12 ([Frank Miller](#))
  - ▶ Computer labs 1-3 ([Karl Sigfrid](#))
- Modules:
  - ▶ **Supervised learning** with **regularized regression** and **classification**.
  - ▶ **Neural networks and deep learning**
  - ▶ **Unsupervised, semisupervised** and **active learning**.
- Examination
  - ▶ Three computer lab reports, 3 credits (in pairs of students)
  - ▶ Exam (5+1 hours computer-based exam. Take home this semester), 4.5 credits

# What is Machine Learning?

Wikipedia (Nov 2, 2019)

- **Machine learning** is the scientific study of algorithms and statistical models that computer systems use to perform a specific task without using explicit instructions, relying on patterns and inference instead.
- It is seen as a subfield of **artificial intelligence**. Machine learning algorithms build a mathematical model based on sample data to make **predictions** or **decisions**.
- Machine learning is closely related to **computational statistics**.
- **Data mining** is a field of study within machine learning, and focuses mainly on exploratory data analysis through unsupervised learning.
- Machine learning for business applications: **predictive analytics**.
- **Data science**. Data science is an interdisciplinary field that uses scientific methods, processes, algorithms and systems to extract knowledge and insights from structured and unstructured data.

# How to become viral on Twitter



Mattias Villani @matvil · Oct 21

...

I will regret this slide in the morning.

## 🔥 Machine Learning vs Traditional Statistics 🔥

	Stats	ML
Parameter inference	😍	🤣
Prediction	😐	😍
Decision making	😳	😍
Interpreting models	😍	😳
Interpreting decisions	😳	😘
Flexible models and regularization	🤔	😍
Rigorous theory	😍	😊
Causality	😐	😐
Programming	😢	😎

40

342

1.4K

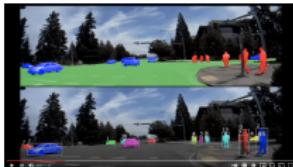


# 🔥 Machine Learning vs Traditional Statistics 🔥

	Stats	ML
Parameter inference	😍	😴
Prediction	😐	😍
Decision making	😳	😍
Interpreting models	😍	😳
Interpreting decisions	😳	😘
Checking model assumptions	😎	😊
Flexible models and regularization	🤔	😍
Rigorous theory	😍	😜
Causality	😐	😐
Programming	😢	😎
Scalability, big data	😱	😎
Real-time/Online	😊	🤓
Data collection and experimental design	😎	🤣

# ML applications

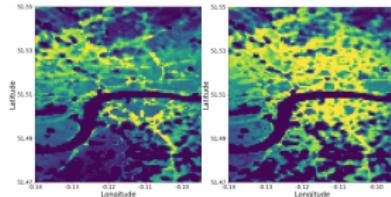
- Finding and tracking objects in traffic (image data).



- Finding bugs in computer code from bug reports (text data).



- Predicting pollution (spatiotemporal sensor data).

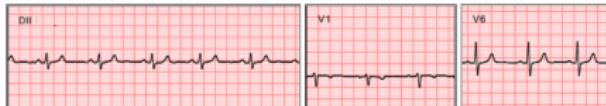


- Predicting defaults on bank loans (standard tabular data).

# ML applications

## Training data

No abnormalities



Atrial fibrillation



Right bundle branch block



## Test data



---

Figures from Lindholm et al (2021).

# Supervised machine learning workflow

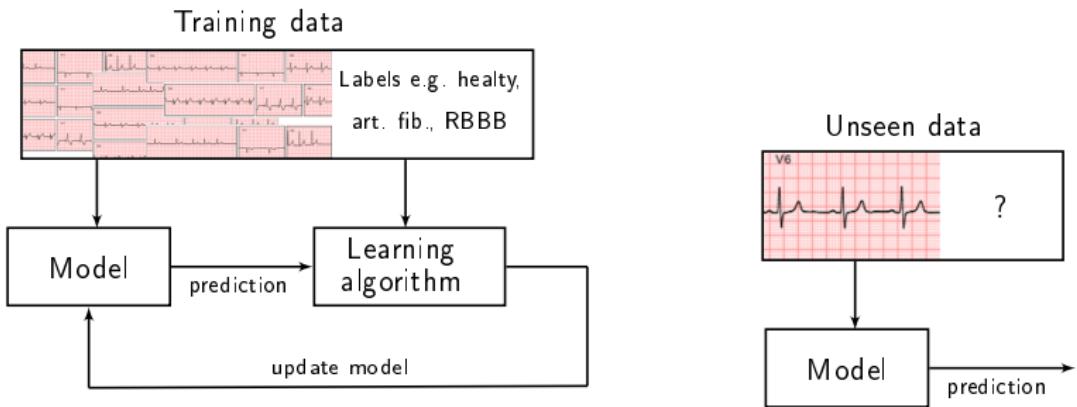
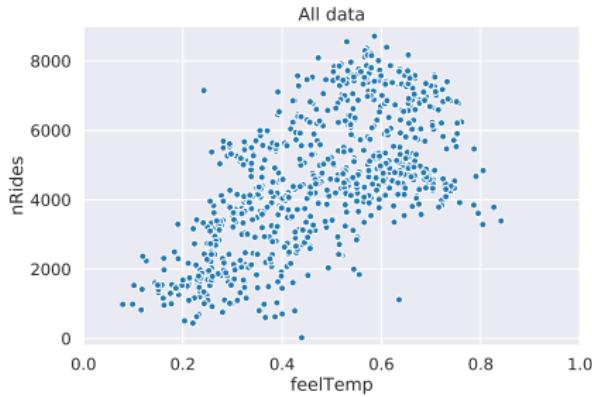
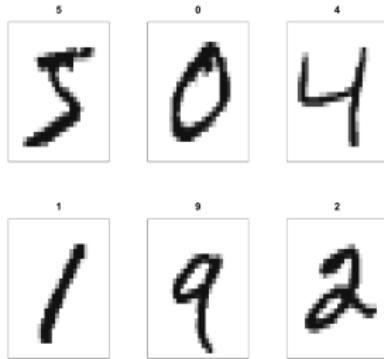


Figure from Lindholm et al (2021).

# Labeling distinctions

- Availability of labels
  - ▶ **Supervised learning**: labeled training data.
  - ▶ **Unsupervised learning**: unlabeled training data.
  - ▶ **Semi-supervised learning**: labels for a subset of training data
  - ▶ **Reinforcement learning**: sequential learning with incremental rewards for good behavior.
- Types of labels
  - ▶ **Regression** - labels are real numbers
  - ▶ **Classification** - labels are discrete



# Data distinctions

- **Regression** data
  - ▶ Real valued
  - ▶ Counts
  - ▶ Proportions
- **Categorical** data (binary, multi-class)
- **Structured non-tabular data** common in ML applications:
  - ▶ **Images**
  - ▶ **Text**
  - ▶ **Sound**
- Other **dependence structures**:
  - ▶ **Time series** (sensor data over time)
  - ▶ **Longitudinal data** (one short time series for many objects)
  - ▶ **Spatial data** (sensors at different locations, images)
  - ▶ **Survival data** (lifetime of hard drives)

# Why Statistics in ML?

- Probability models and statistical inference is a **framework**.
- Principled **way to think** about any problem in ML.
- Can be **evaluated** and critiqued.
- **Quantify uncertainties**. Crucial for **Decision making**.

*As robotics is now moving into the open world, the issue of **uncertainty** has become a major stumbling block for the design of capable robot systems. Managing uncertainty is possibly the most important step towards robust real-world robot systems.*

*from the book Probabilistic Robotics by Thrun et al.*

# Linear/Nonlinear - Parametric/Nonparametric

## ■ Regression:

- ▶ Labels are continuous.
- ▶ Linear regression:  $y = \mathbf{x}^\top \boldsymbol{\beta} + \varepsilon$ ,  $\varepsilon \sim N(0, \sigma^2)$ .

## ■ Classification:

- ▶ Labels are binary or categorical.
- ▶ Logistic regression:  $\Pr(y = 1 | \mathbf{x}) = \frac{\exp(\mathbf{x}^\top \boldsymbol{\beta})}{1 + \exp(\mathbf{x}^\top \boldsymbol{\beta})}$ .

## ■ Linear regression and logistic regression are:

- ▶ **Parametric**
- ▶ **Linear** (log-odds and decision boundaries given by  $\mathbf{x}^\top \boldsymbol{\beta}$ )

## ■ ML

- ▶ **Flexible**, richly parametrized models
- ▶ **Non-linear** models
- ▶ **Non-parametric** models

## *k*-nearest neighbor models

- Nonparametric model for regression and classification.
- Aim: predict  $y_*$  for new  $x_*$ .
- ***k*-NN regression:** predict by average of  $k$  nearest neighbors to  $x_*$

$$\hat{y}(x_*) = \frac{1}{k} \sum_{i \in \mathcal{N}_*} y_i = \text{Average } \{y_i : i \in \mathcal{N}_*\}$$

where  $\mathcal{N}_*$  is the set of  $k$  observations with smallest  $\|x_i - x_*\|_2$ .

- ***k*-NN classification:** majority vote of  $k$  nearest neighbors.
- Any distance measure can be used instead of  $\|x_i - x_*\|_2$ .
- **Input data** ( $x$ ) should be **normalized** (zero mean, unit variance, or scaled so  $x \in [-1, 1]$ ).
- **Hyperparameter**  $k$  selected from predictive performance.

# $k$ -nearest neighbor classification

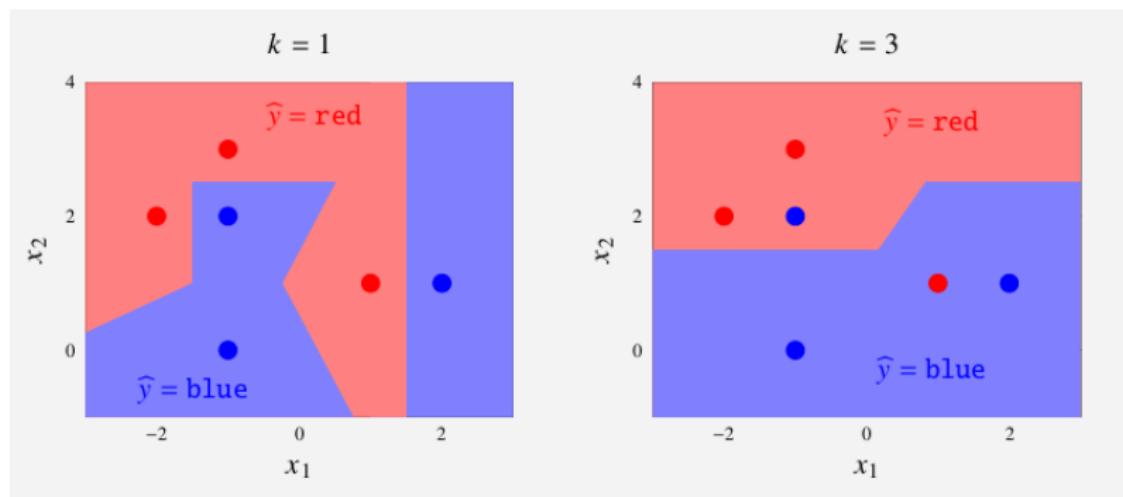


Figure from Lindholm et al (2021).

# Song classification

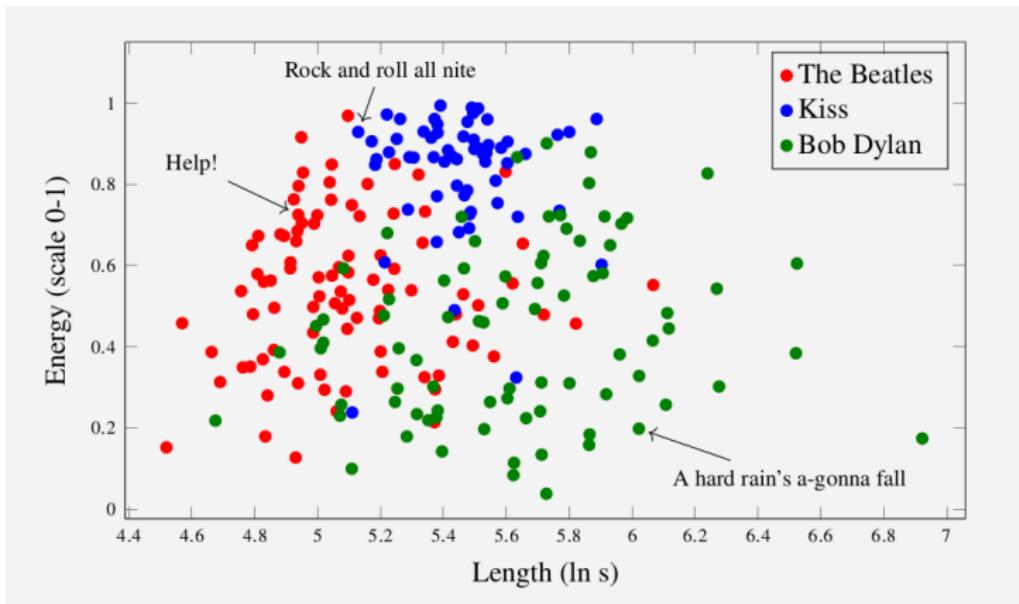


Figure from Lindholm et al (2021).

# Song classification with $k$ -NN

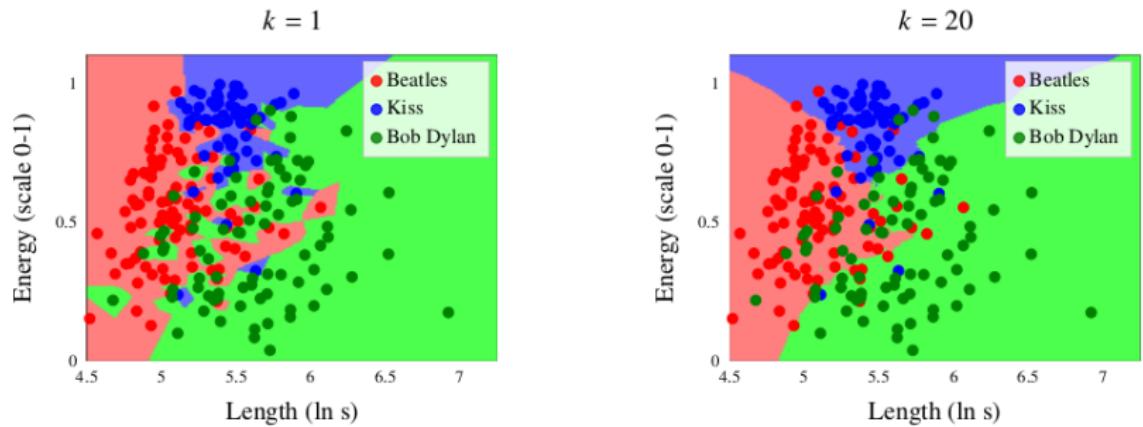
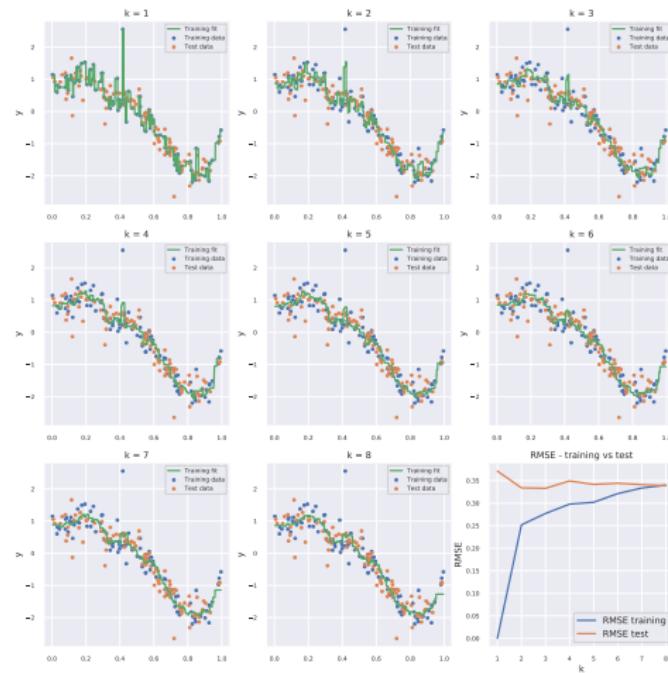


Figure from Lindholm et al (2021).

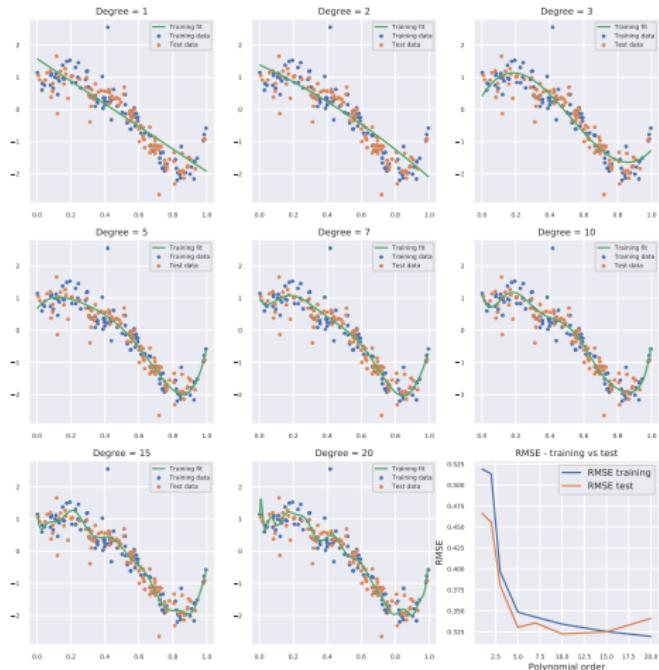
# *k*-nearest neighbor regression



# $k$ -nearest neighbor regression



# Polynomial regression



# Regression trees

- Partition the feature space into  $L$  rectangles,  $R_1, \dots, R_L$ .
- Predict with a model which is constant in each  $R_\ell$ :

$$\hat{y}(x_\star) = \sum_{\ell=1}^L w_\ell \mathbb{I}\{x_\star \in R_\ell\}$$

- Rectangle indicator functions:

$$\mathbb{I}\{x \in R_\ell\} = \begin{cases} 1 & \text{if } x \in R_\ell \\ 0 & \text{if } x \notin R_\ell \end{cases}$$

- Level in rectangle  $R_\ell$  is  $w_\ell$  which can be estimated by

$$\hat{w}_\ell = \hat{y}_\ell = \text{Average } \{y_i : x_i \in R_\ell\}.$$

- Regression trees use binary splits, one feature at the time.
- Computationally efficient and nice interpretation.

# Regression trees

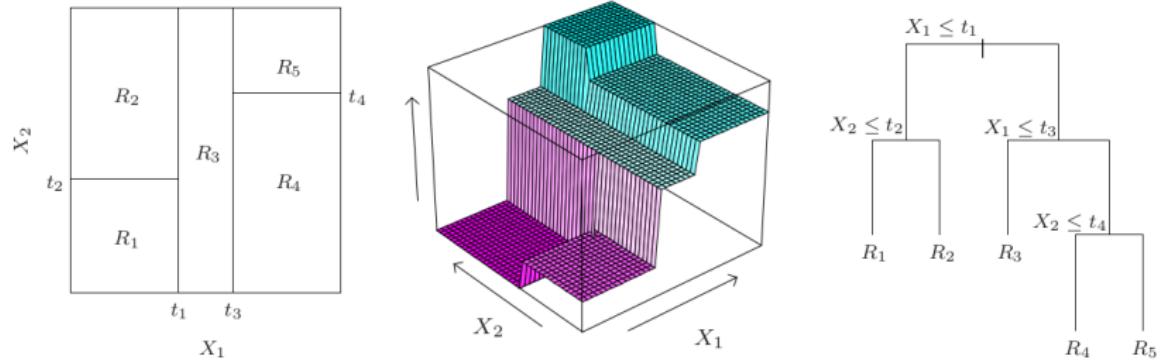


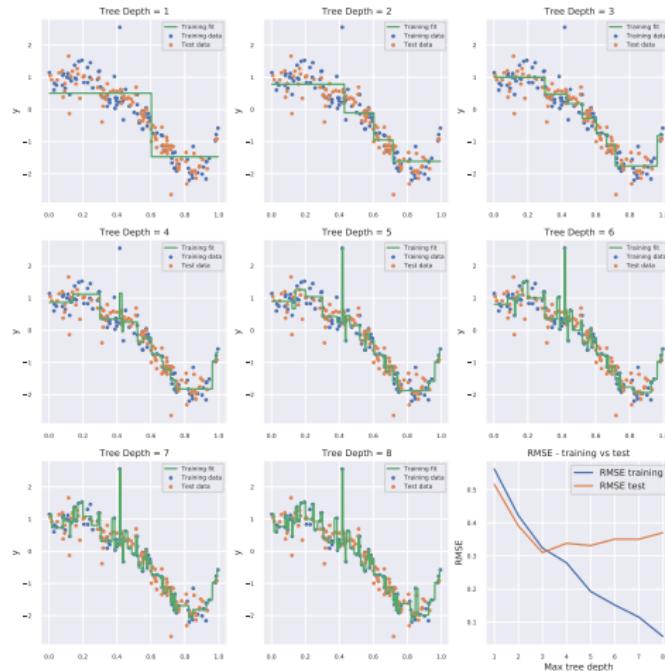
Figure from Hastie et al (2009).

# Fitting regression trees

- **Parameters:**
  - ▶ **split variable** at each stage:  $j_1, j_2, \dots$
  - ▶ **splitting point** at each stage:  $s_1, s_2, \dots$
  - ▶ **level constants**  $c_1, c_2, \dots$
- A given split  $(j, s)$  defines two half-planes in  $x$ -space:
$$R_L(j, s) = \{x | x_j \leq s\} \text{ and } R_H(j, s) = \{x | x_j > s\}$$
- **Level constants.** For a given split  $(j, s)$ , estimate
$$\hat{y}_L(j, s) = \text{Average } \{y_i : x_i \in R_L(j, s)\}$$
$$\hat{y}_H(j, s) = \text{Average } \{y_i : x_i \in R_H(j, s)\}$$
- Greedy for **split variable**  $j$  and **splitting point**  $s$

$$\min_{j, s} \left[ \min_{\hat{y}_L} \sum_{i: x_i \in R_L(j, s)} (y_i - \hat{y}_L(j, s))^2 + \min_{\hat{y}_H} \sum_{i: x_i \in R_H(j, s)} (y_i - \hat{y}_H(j, s))^2 \right].$$

# Regression tree



# Cost-complexity pruning

■ How big tree?

■ Bias (small tree) vs Variance (large tree)

■ Cost-complexity pruning:

▶ grow a large tree (few observation at each leaf)

▶ prune the tree by collapsing non-terminal nodes to minimize

$$\frac{1}{n} \sum_{i=1}^n L(\hat{y}(x_i), y_i) + \eta |T| + \lambda \|w\|_2^2$$

- $L(\hat{y}(x_i), y_i)$  is a loss function (e.g. squared error)
- $|T|$  is the number of leaves in the subtree  $T$ .
- $\lambda \|w\|_2^2$  is a L2-regularization term (more later!)

▶ Hyperparameters  $\eta$  and  $\lambda$  can be set with cross-validation.

■ Variable importance for  $x_j$ : summing the improvement in fit at each node that is split by  $x_j$ .