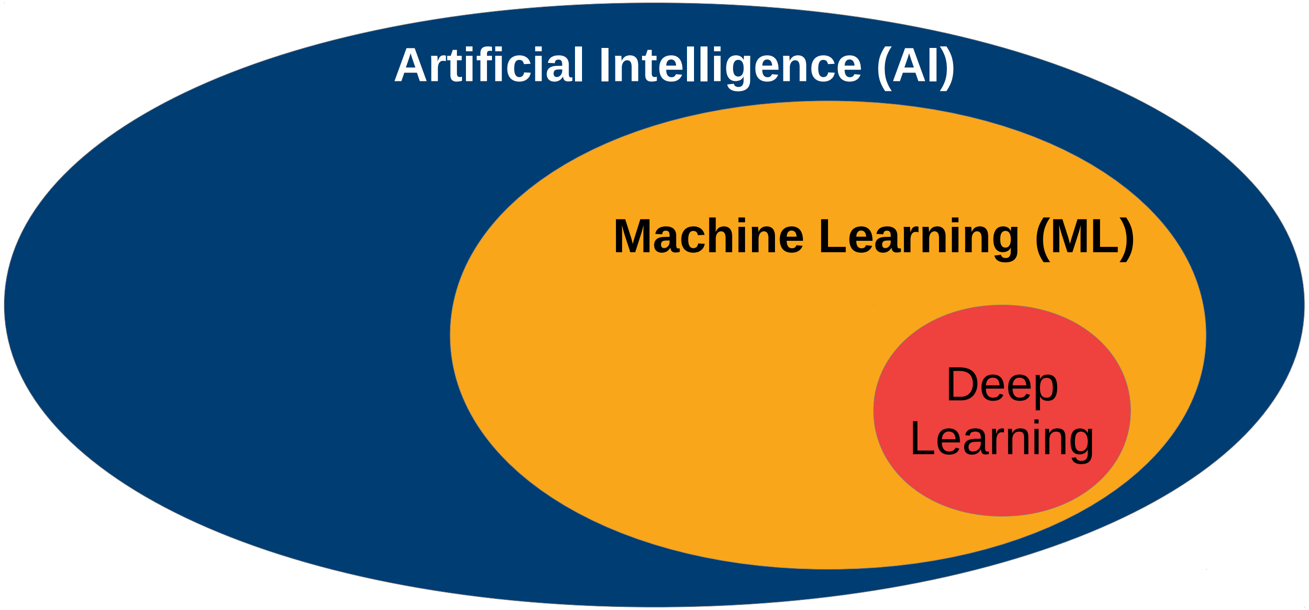
Machine Learning Basics

Janis Keuper

Introduction to ML

INSTITUTE FOR MACHINE LEARNING AND ANALYTICS

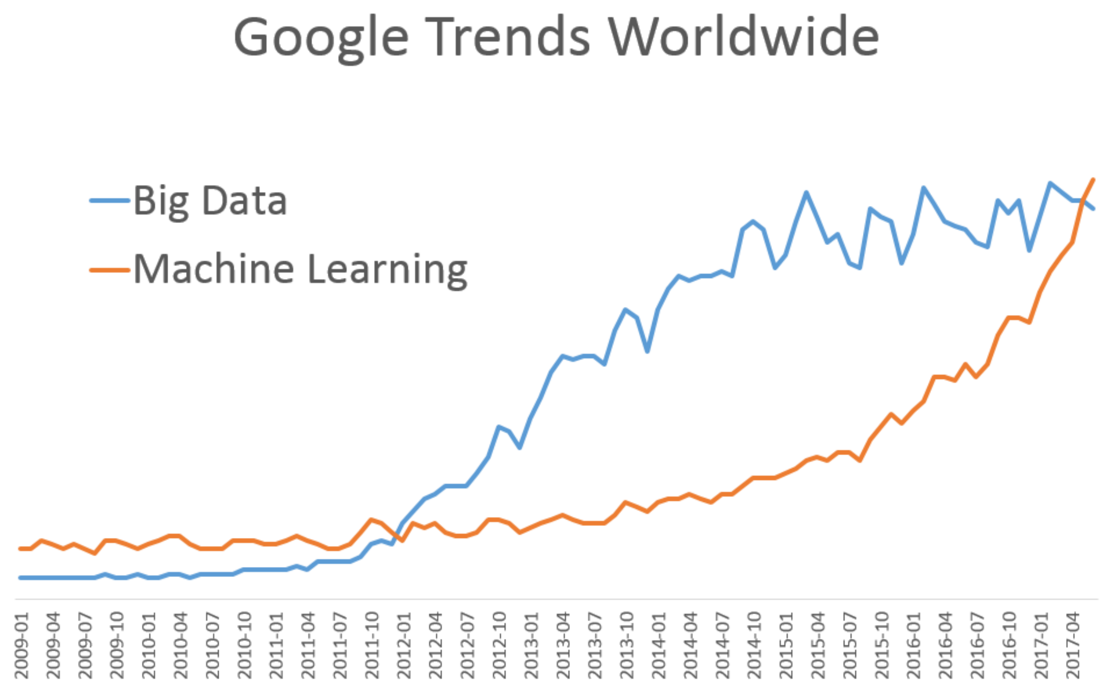
Research and Application Fields



Introduction to ML

INSTITUTE FOR MACHINE LEARNING AND ANALYTICS

The ML Hype



Introduction to ML

INSTITUTE FOR MACHINE LEARNING AND ANALYTICS

Basic Types of Machine Learning Algorithms

- **Supervised Learning**
- **Unsupervised Learning**
- **Reinforcement Learning**

Introduction to ML

INSTITUTE FOR MACHINE LEARNING AND ANALYTICS

Basic Types of Machine Learning Algorithms

Supervised Learning

Unsupervised Learning

Reinforcement Learning

- Labeled data
- Direct and quantitative evaluation

- Learn model from „ground truth“ examples
- Predict unseen examples

Introduction to ML

INSTITUTE FOR MACHINE LEARNING AND ANALYTICS

Supervised Learning

Basic Notation:

Data is given as tuples

$$(X, Y) := \{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}$$

Where X is the actual **data** (sample) and y the associated **label.**

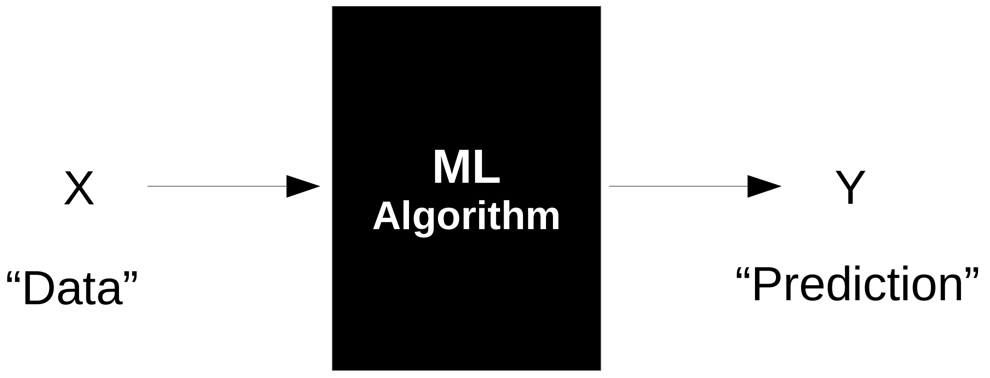
For most ML algorithms **(many Deep Learning algorithms are an exception)**

$$x_i \in \mathbb{R}^n, y_i \in \mathbb{R}$$

The data has to be represented as vectors and the labels are scalars.

Introduction to ML

Supervised Learning as a Black Box



ML algorithms “learns” ***mapping*** from input to output by example tuples

Introduction to ML

INSTITUTE FOR MACHINE LEARNING AND ANALYTICS

Supervised Learning: Example: Classification

ML

“coffee mug”

X: Image

Classification: Y discrete
semantic class label

ML algorithms “learns” ***mapping*** from input to output by example tuples

INSTITUTE FOR MACHINE LEARNING AND ANALYTICS

Supervised Learning: Example: Classification

ML

Scalar

"coffee mug"

X: Image

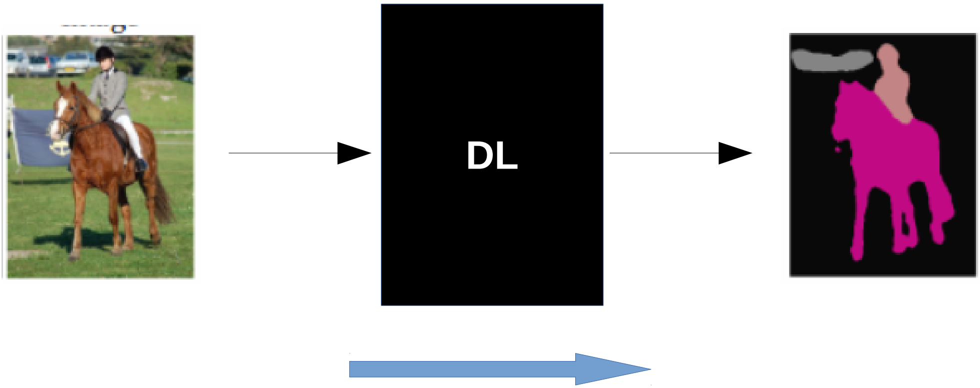
Classification: Y discrete
semantic class label

Vector

ML algorithms "learns" ***mapping*** from input to output by example tuples

Function f(x) → y

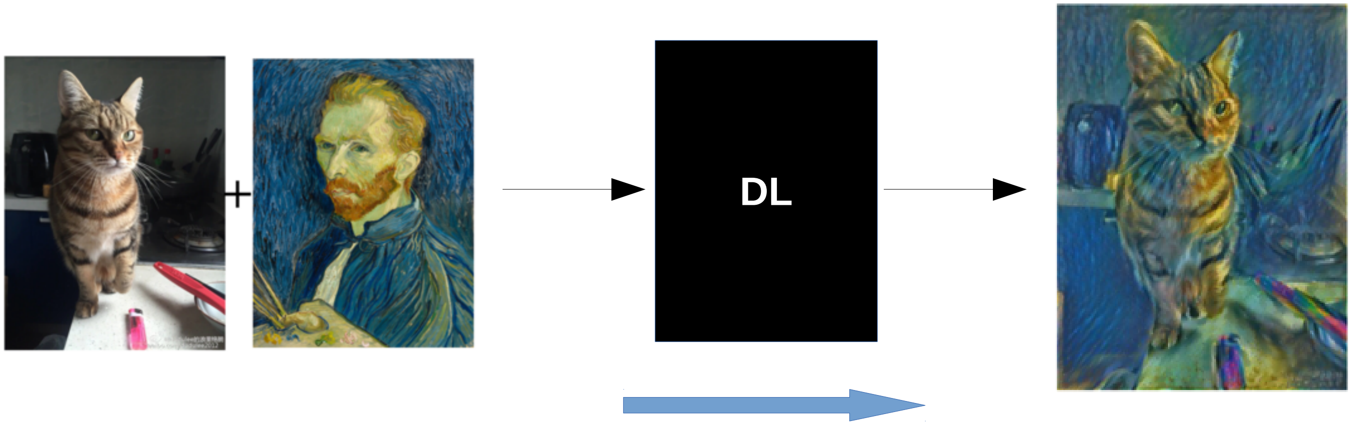
More Examples (here Deep Learning)



Example: semantic segmentation

Introduction to ML

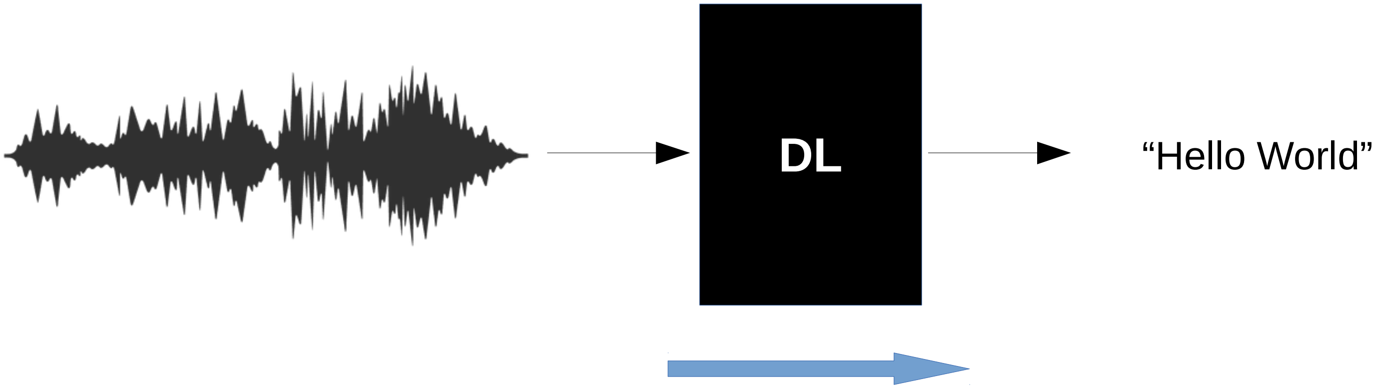
More Examples (here Deep Learning)



Example: contend generation, e.g. style transfer learning

Introduction to ML

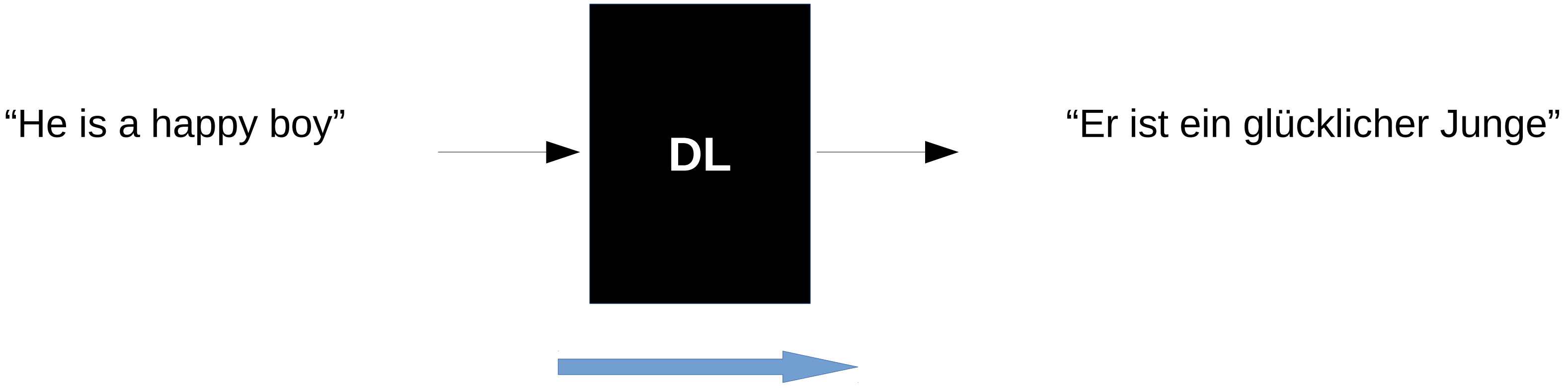
More Examples (here Deep Learning)



Example: Speech recognition

Introduction to ML

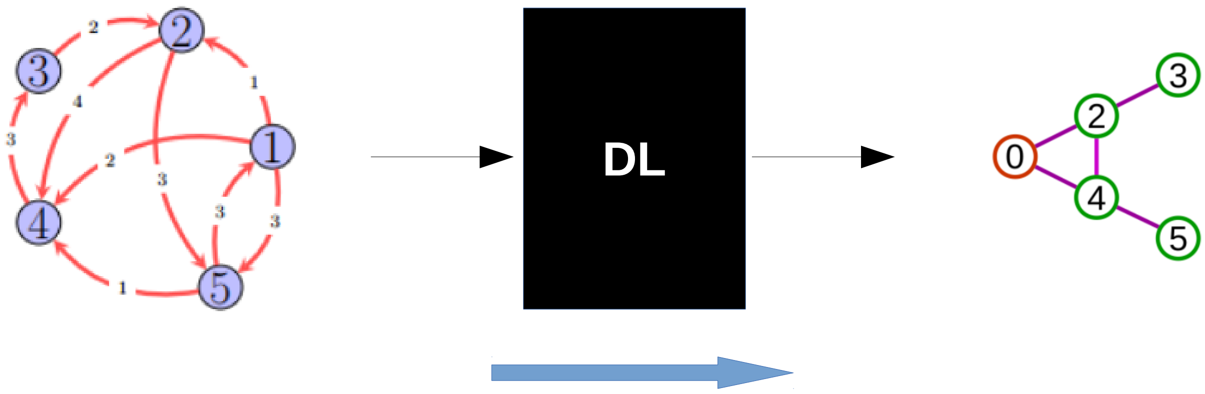
More Examples (here Deep Learning)



Example: Text understanding, e.g. translations

Introduction to ML

More Examples (here Deep Learning)



Example: Graph analysis

Introduction to ML

More Examples (here Deep Learning)

→ DL → “move right”

Video sequence → action

Example: game playing

Introduction to ML

INSTITUTE FOR MACHINE LEARNING AND ANALYTICS

Supervised Learning: Example: Classification

X: Image

→ **ML** →

Y: {“coffee mug”, “non coffee mug}

Introduction to ML

Supervised Learning: Example: Classification

X: Image

pixel by pixel

$$\begin{pmatrix} 5 \\ 8 \\ 15 \\ 23 \\ 21 \\ 30 \\ 28 \\ \dots \\ 5 \end{pmatrix}$$

$$X \in \mathbb{R}^n$$

ML

Y': {-1,1} $Y' \in \mathbb{R}$

1: "coffee mug",
-1: "non coffee mug

Introduction to ML

Supervised Learning: Example: Classification

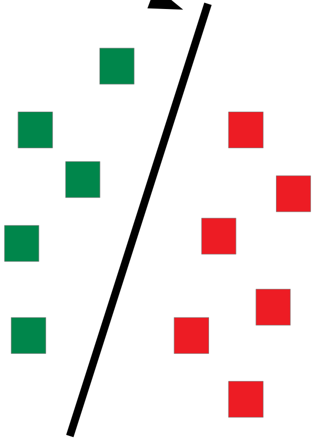
ML Model: function *f* separating mugs from rest

X

$$X = \begin{pmatrix} 5 \\ 8 \\ 1 \\ 5 \\ 2 \\ 3 \\ 0 \\ 2 \\ 8 \\ \dots \\ 5 \end{pmatrix}, \quad X \in \mathbb{R}^n$$

ML → Y': {-1,1}

y=1



y=-1

$X \in \mathbb{R}^n$

Introduction to ML

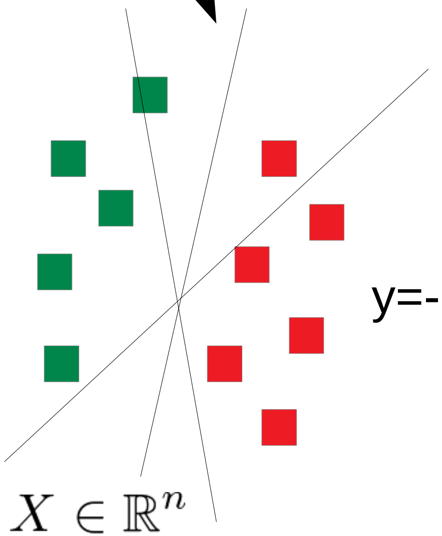
INSTITUTE FOR MACHINE
LEARNING AND ANALYTICS

Supervised Learning: Example: Classification

LEARNING: approximate „best“ *f* for the given data

X

→ (5, 8, 1, 5, 2, 3, 0, 2, 8, …, 5) $X \in \mathbb{R}^n$ → **ML** → Y’: {-1,1}



Introduction to ML

INSTITUTE FOR MACHINE LEARNING AND ANALYTICS

Supervised Learning: Example: Classification

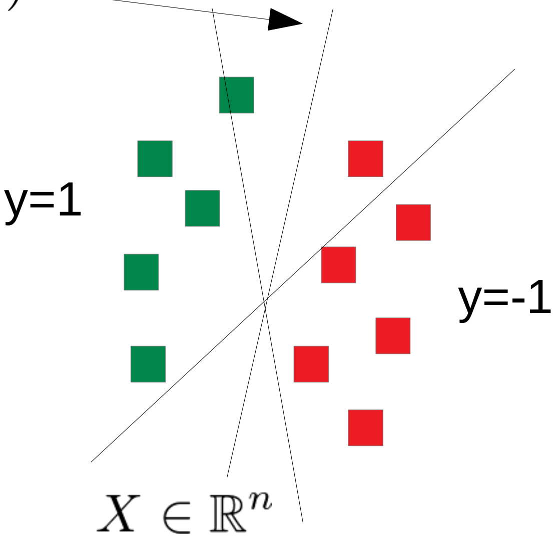
LEARNING: optimization problem:

$$min(\|f(X,Y),Y'\|)$$

X

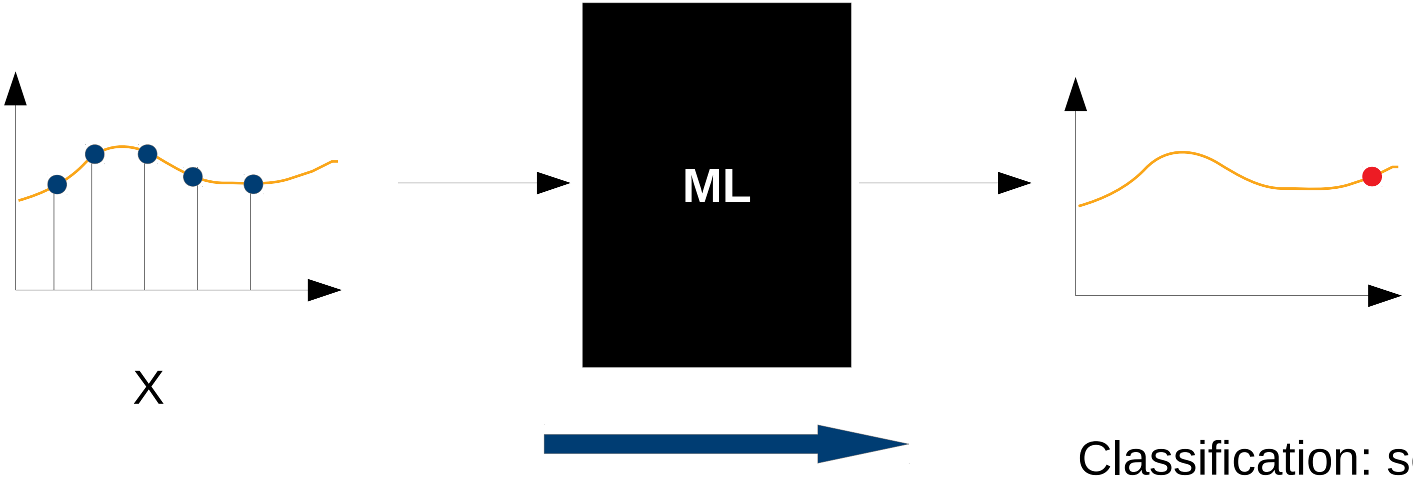
$\rightarrow$ $(5, 8, 1, 5, 2, 3, 0, 2, 8, \ldots, 5)^T$ $\rightarrow$ **ML** $\rightarrow$ Y': {-1,1}

$X \in \mathbb{R}^n$



INSTITUTE FOR MACHINE LEARNING AND ANALYTICS

Supervised Learning: Example: Regression



Y is **continuous**

Regression

Classification: separation function

Regression: continuous mapping
→ harder problem

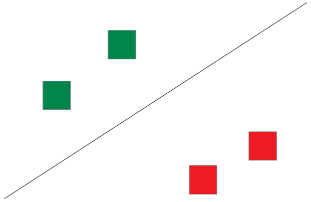
INSTITUTE FOR MACHINE LEARNING AND ANALYTICS

Challenges of Supervised Learning

- Not only need data – also need to have $Y \rightarrow$ human annotation
  - Getting "enough" labeled data is expensive
  - Sometimes impossible

UNDERFITTING

$$min(\|f(X,Y),Y'\|)$$



Training model
On little data

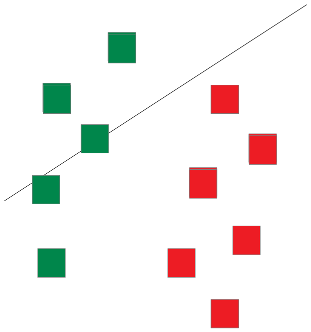
Introduction to ML

Challenges of Supervised Learning

- Not only need data – also need to have *Y* → human annotation
  - Getting “enough” labeled data is expensive
  - Sometimes impossible

UNDERFITTING

$$min(\|f(X,Y),Y'\|)$$



→ bad sampling
Of the data distribution

Introduction to ML

Challenges of Supervised Learning

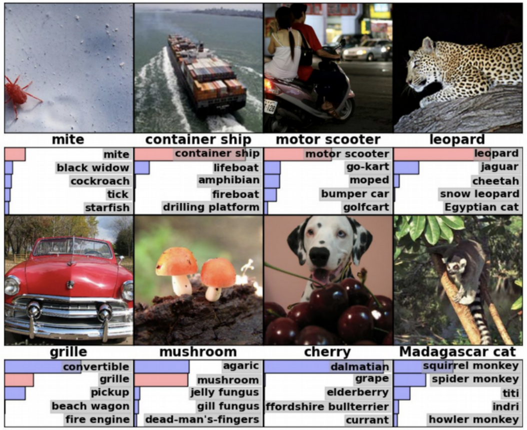
- Not only need data – also need to have Y → human annotation
  - Getting “enough” labeled data is expensive
  - Sometimes impossible

Example:

ImageNet Challenge

IMAGENET

- 1,000 object classes (categories).
- Images:
  - 1.2 M train
  - 100k test.



INSTITUTE FOR MACHINE LEARNING AND ANALYTICS

Challenges of Supervised Learning

- Not only need data – also need to have $Y \rightarrow$ human annotation
  - Getting “enough” labeled data is expensive
  - Sometimes impossible

Example:

CITYSCAPES DATASET

Introduction to ML

INSTITUTE FOR MACHINE LEARNING AND ANALYTICS

Challenges of Supervised Learning

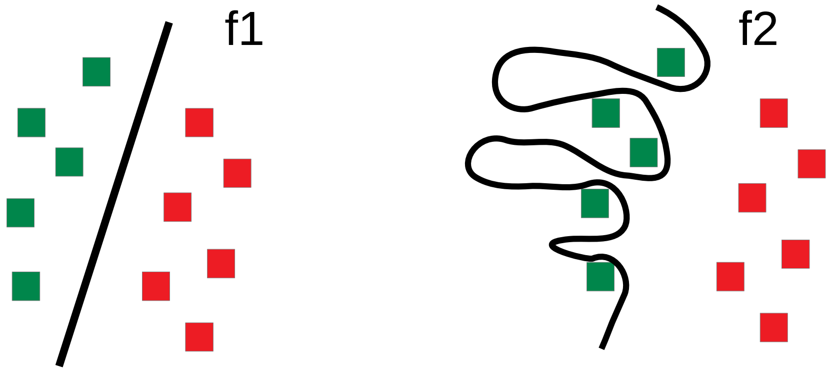
- Not only need data – also need to have Y → human annotation
  - Getting “enough” labeled data is expensive
  - Sometimes impossible
- Training data is **only a sample:** prediction must work on **all data** → **generalization**

Challenges of Supervised Learning

- Training data is **only a sample:** prediction must work on **all data** → **generalization**

Which model is better?

$$min(\|f(X,Y),Y'\|)$$



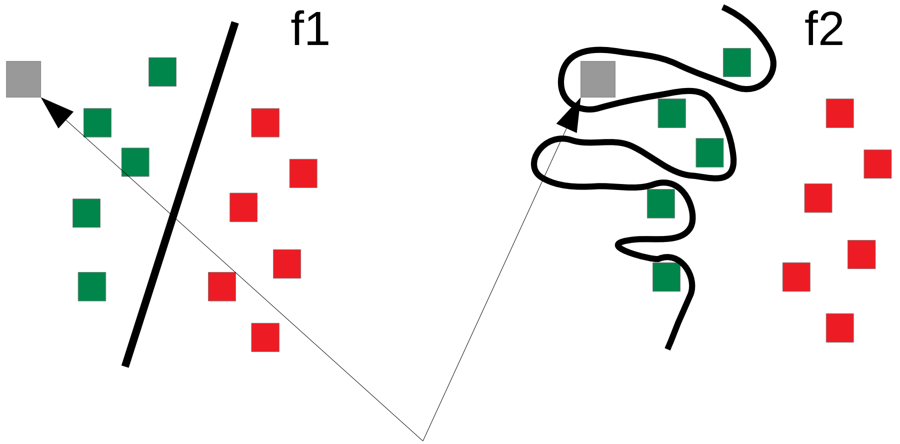
INSTITUTE FOR MACHINE LEARNING AND ANALYTICS

Challenges of Supervised Learning

- Training data is **only a sample:** prediction must work on **all data** → **generalization**

Which model is better?

$$min(\|f(X,Y),Y'\|)$$



Test sample

Introduction to ML

INSTITUTE FOR MACHINE LEARNING AND ANALYTICS

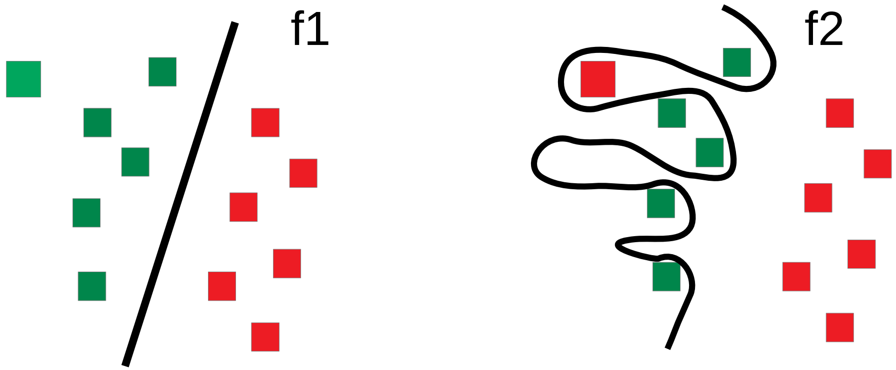
Challenges of Supervised Learning

- Training data is **only a sample:** prediction must work on **all data** → **generalization**

OVERFITTING

Model “to close” to train data

Very likely to happen in practice.
→ we need to work against this...



INSTITUTE FOR MACHINE LEARNING AND ANALYTICS

Data Preparation: Split into Train, Test, and Validate

A basic technique (we will learn more later) to at least detect overfitting is to split the available data into two or three subsets:

- Use unbiased **test set** for final evaluation of a model
- Use **train set** for model training
- **Validation set** (part of train set) can be used to optimize hyper parameters of the model

Caution: sets must be unbiased! (→ random sampling)
In practice it can be hard to guarantee clean train/test sets:
e.g. how to treat possible variance different data sources?
→ statistical analysis needed!

Introduction to ML

Basic evaluation (more techniques to come)

Train error: measure of how well the model predicts the given labels

$$Err_{train} := \frac{1}{|X_{train}|} \sum_{x_i \in X_{train}} |f(x_i) - y_i|$$

low train error is the necessary condition for a "good" model

Test error: **same as train error: low test error is the sufficient condition**

$$Err_{test} := \frac{1}{|X_{test}|} \sum_{x_i \in X_{test}} |f(x_i) - y_i|$$

INSTITUTE FOR MACHINE LEARNING AND ANALYTICS

Basic Types of Machine Learning Algorithms

Supervised Learning

Unsupervised Learning

Reinforcement Learning

- NO Labeled data
- NO Direct and quantitative evaluation

- Explore structure of data

Introduction to ML

INSTITUTE FOR MACHINE LEARNING AND ANALYTICS

Unsupervised Learning

Data without “labels” $(x_1, x_2, \ldots, x_n)$

- **Clustering**
- **Outlier Detection (e.g. Defect or Intrusion detection)**

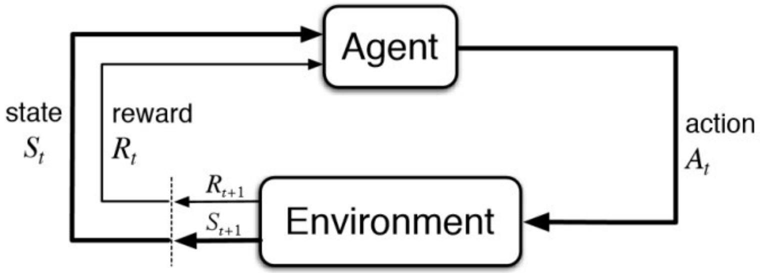
Introduction to ML

Basic Types of Machine Learning Algorithms

Supervised Learning

Unsupervised Learning

Reinforcement Learning



- Learning decisions in an interactive environment
- State ↔ Action learning
- Game playing and robotics
- Hardly use in Data Science

Recall Classification

INSTITUTE FOR MACHINE
LEARNING AND ANALYTICS

Supervised Learning: Annotated Training Data

Vector representation

(X,Y)

$$\begin{pmatrix} 5 \\ 8 \\ 1 \\ 5 \\ 2 \\ 3 \\ 0 \\ 2 \\ 8 \\ \vdots \\ 5 \end{pmatrix}$$

$X \in \mathbb{R}^n$

ML

Y': {-1,1}

Recall Classification

INSTITUTE FOR MACHINE
LEARNING AND ANALYTICS

Supervised Learning: Annotated Training Data

Vector representation

(X,Y)

$$\begin{pmatrix} 5 \\ 8 \\ 1 \\ 5 \\ 2 \\ 3 \\ 0 \\ 2 \\ 8 \\ \ldots \\ 5 \end{pmatrix}$$

$X \in \mathbb{R}^n$

ML

Y': {-1,1}

(feature extraction / engineering)

Recall Classification

Supervised Learning: Annotated Training Data

Vector representation

Feature vector representation
Defines the learning problem!

(X,Y)

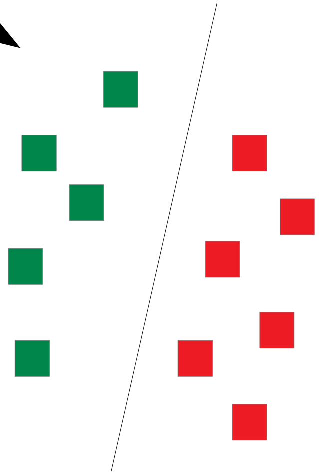
(5, 8, 1, 5, 2, 3, 0, 2, 8, …, 5)

$X \in \mathbb{R}^n$

ML

Y': {-1,1}

y=1



y=-1

$X \in \mathbb{R}^n$

(feature extraction / engineering)

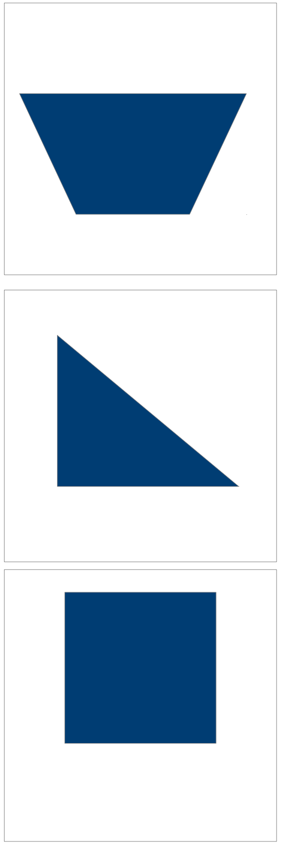
Feature Extraction

INSTITUTE FOR MACHINE LEARNING AND ANALYTICS

A Simple Example

A algorithm that can classify three different shape in an image:

- How to vectorize the data samples?



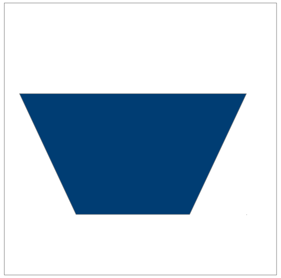
Feature Extraction

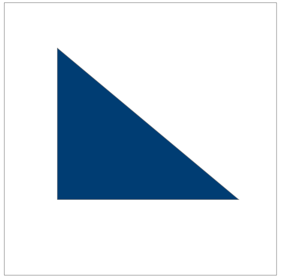
A Simple Example

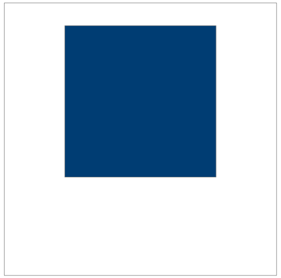
A algorithm that can classify three different shape in an image:

- How to vectorize the data samples?

Naive solution: dump (raw) pixel data row by row into vector.







Feature Extraction

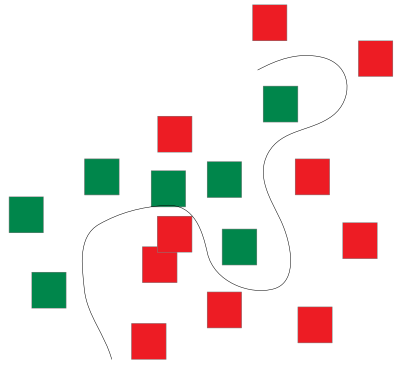
INSTITUTE FOR MACHINE LEARNING AND ANALYTICS

A Simple Example

A algorithm that can classify three different shape in an image:

- How to vectorize the data samples?

Example feature space



Naive solution: dump (raw) pixel data row by row into vector.

→ possible. BUT can become very difficult problem if we have high Variance in the data, e.g. shapes can **translate**, **scale** or even **rotate**.

→ results in the need for much data (to cover the variance) and the high dimensional space. Also needs complex models (danger to overfit).

Feature Extraction

INSTITUTE FOR MACHINE LEARNING AND ANALYTICS

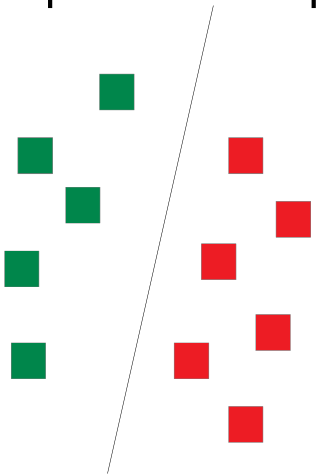
A Simple Example

A algorithm that can classify three different shape in an image:

- How to vectorize the data samples?

Alternative: find Features T(X) that **encode separable properties** of the data.

Example feature space



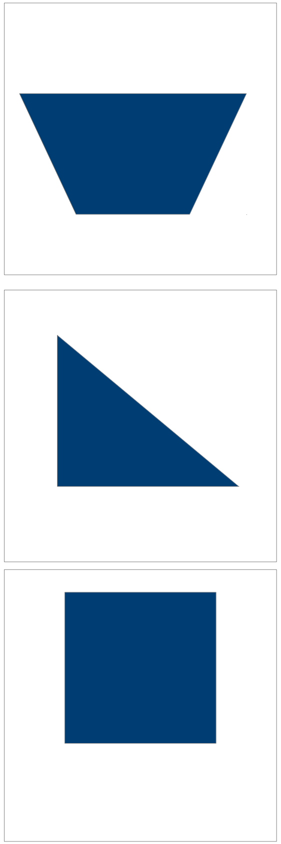
- computed by some function T(X)
- Compact representation of X (low dimension)
- Robust against data variance (detailed definition coming up)
- **Allows much simpler model!**

Feature Extraction

A Simple Example

Possible features for our problem?

- **Number of corners**
  → requires a corner detector (standard computer vision algorithm)
  → does not change under scaling, rotation and translation
  (but occlusion)
  → separates triangle from other two shapes

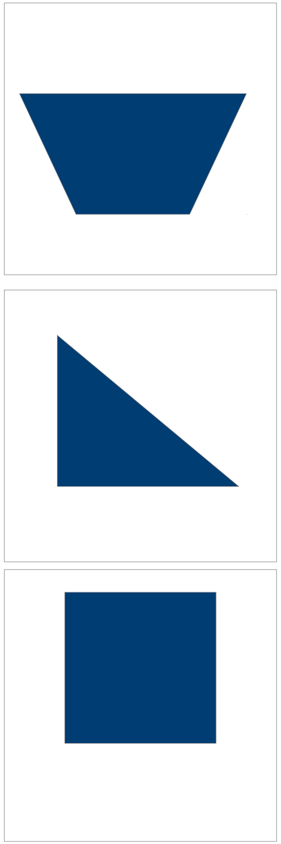


Feature Extraction

A Simple Example

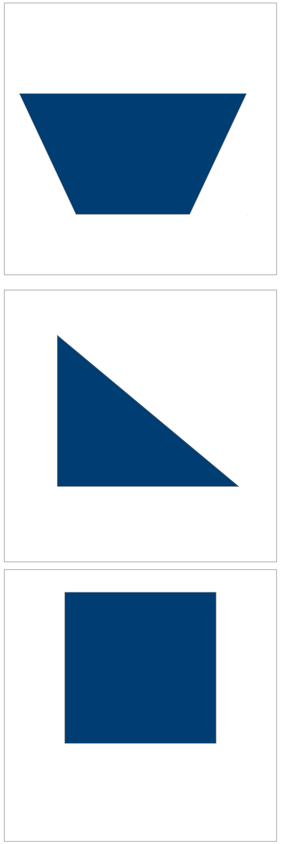
Possible features for our problem?

- **Number of corners**
- **Area**



Feature Extraction

A Simple Example



Possible features for our problem?

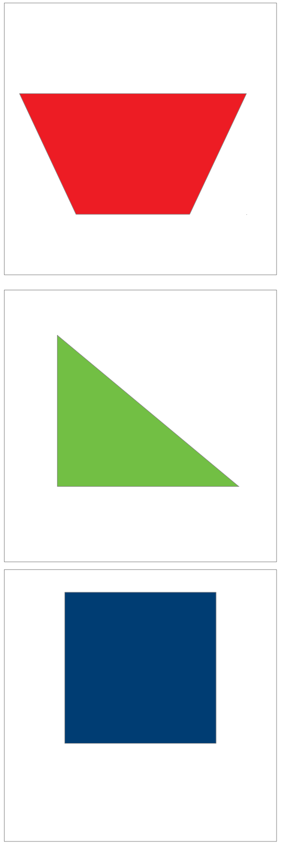
- **Number of corners**
- **Area**
- **Area to circumference relation**
- **Max to min side length relation**
- **...**

Feature Extraction

A Simple Example

Possible features for our problem?

- **Number of corners**
- **Area**
- **Area to circumference relation**
- **Max to min side length relation**
- **…**
- **Color !**



Invariant Theory

Example for invariant features – A classic inspection problem:

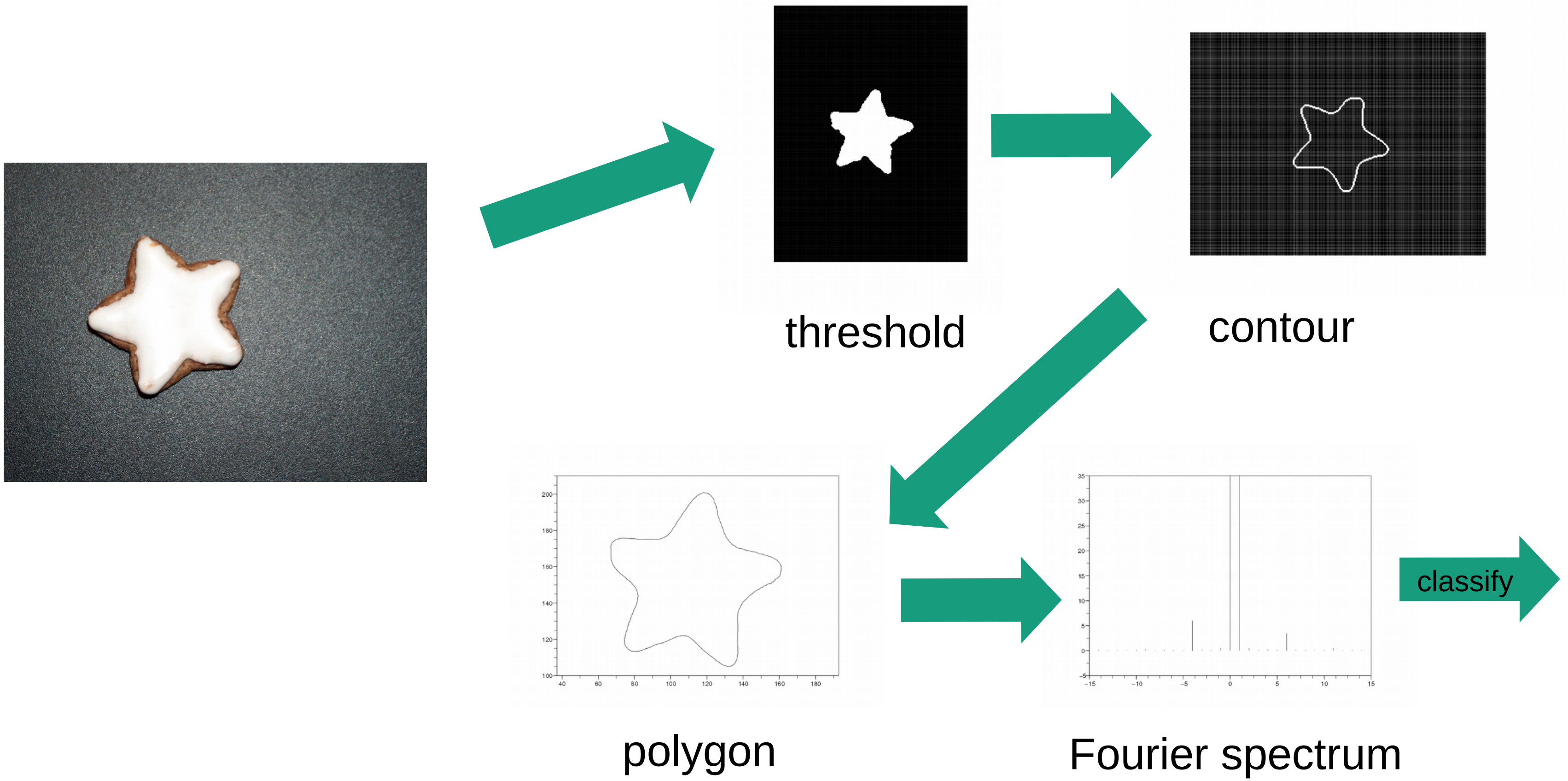
sample data

arbitrary rotations and translations

„Defect"

Invariant Theory

possible solution: invariant features



Feature Extraction

Discussion:

Feature extraction vs learning algorithms

I. if we had perfect features, learning would be trivial

II. if we had perfect classifiers, we would not need features

Features are usually used to introduce prior knowledge about the Structure of the data and variances to the learning algorithm.

Feature Extraction

Discussion:

In practice:

- Good features are hard to find
- Often based on complex mathematical functions
- Depend on the application (domain knowledge needed!)

Generic Approaches:

- **Invariance by differentiation**: set properties into relation / normalization
- **Invaraince by integration**: compute average properties

Feature Reduction

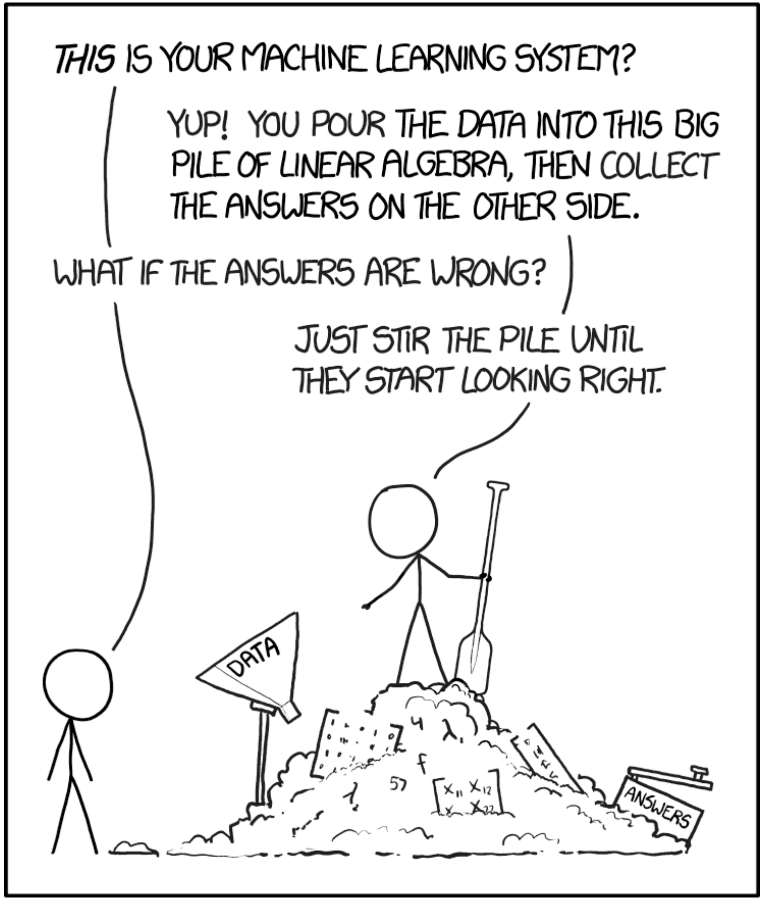
Motivation

- Very high dimensional representations require a lot of data to fill this huge space (curse of dimensionality)

- Danger of overfitting is higher if space is only sparsely sampled

  → **We would like to "compress" our data (with steerable loss) to a lower dimensional representation**

Discussion



https://xkcd.com/1838/