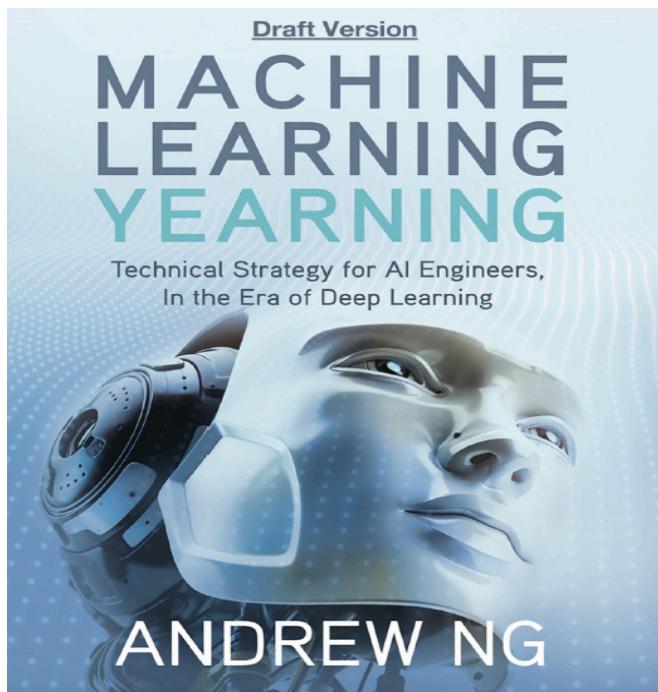


# **Methods for Artificial Intelligence**

Alexander Jung, Computer Science/Aalto

# Reading Material



<https://wwwdeeplearning.ai/machine-learning-yearning/>

<https://arxiv.org/abs/1805.05052>

arXiv.org > cs > arXiv:1805.05052

Computer Science > Machine Learning

**Machine Learning: Basic Principles**

Alexander Jung

(Submitted on 14 May 2018 (v1), last revised 8 Oct 2018 (this version, v8))

This tutorial is based on the lecture notes for, and the plentiful student feedback received from, "Machine Intelligence", which I have co-taught since 2015 at Aalto University. The aim is to provide an accessible introduction to machine learning. Many of the current systems which are considered as (artificially) intelligent are based on machine learning methods. After formalizing the main building blocks of a machine learning problem, some popular machine learning methods are discussed in some detail. In order to improve accessibility of the main concepts, we mostly avoid mathematical notation.

Subjects: Machine Learning (cs.LC); Machine Learning (stat.ML)

Try the Bibliographic Explorer (can be disabled at any time) [s.LG] for this version)  
[Enable](#) [Don't show again](#)

Bibliographic data  
[Enable Bibex (What is Bibex?)]

Submission history

# Some Video Material



## Machine learning in Python with scikit-learn

10 videos • 615,973 views • Last updated on Aug 23, 2016



Learn how to use Python's scikit-learn library to perform effective machine learning:

<https://github.com/justmarkham/scikit-learn>



Data School

SUBSCRIBE

- 1 **What is machine learning, and how does it work?**  
Data School
- 2 **Setting up Python for machine learning: scikit-learn and Jupyter Notebook**  
Data School
- 3 **Getting started in scikit-learn with the famous iris dataset**  
Data School
- 4 **Training a machine learning model with scikit-learn**  
Data School
- 5 **Comparing machine learning models in scikit-learn**  
Data School
- 6 **Data science in Python: pandas, seaborn, scikit-learn**  
Data School

<https://www.youtube.com/playlist?list=PL5-da3qGB5ICeMbQuqbbCOQWcS6OYBr5A>

# (Artificial) Intelligence Involves (Machine) Learning

from Wikipedia:

“Among the traits that researchers hope machines will exhibit are

reasoning, knowledge, planning, learning,  
communication, perception,....“

# AI Principle (Engineer Perspective)

automatically choose (compute) optimal decisions to  
maximize a long-term reward

# Machine Learning for AI

- optimal decisions/planning require insights
- ML extracts high-level insights from raw data
- ML is low-level AI

# What is Machine Learning?

computer programs (algorithms) which use historic data to learn how to make predictions

# When to Harvest?



# When will my bus arrive?



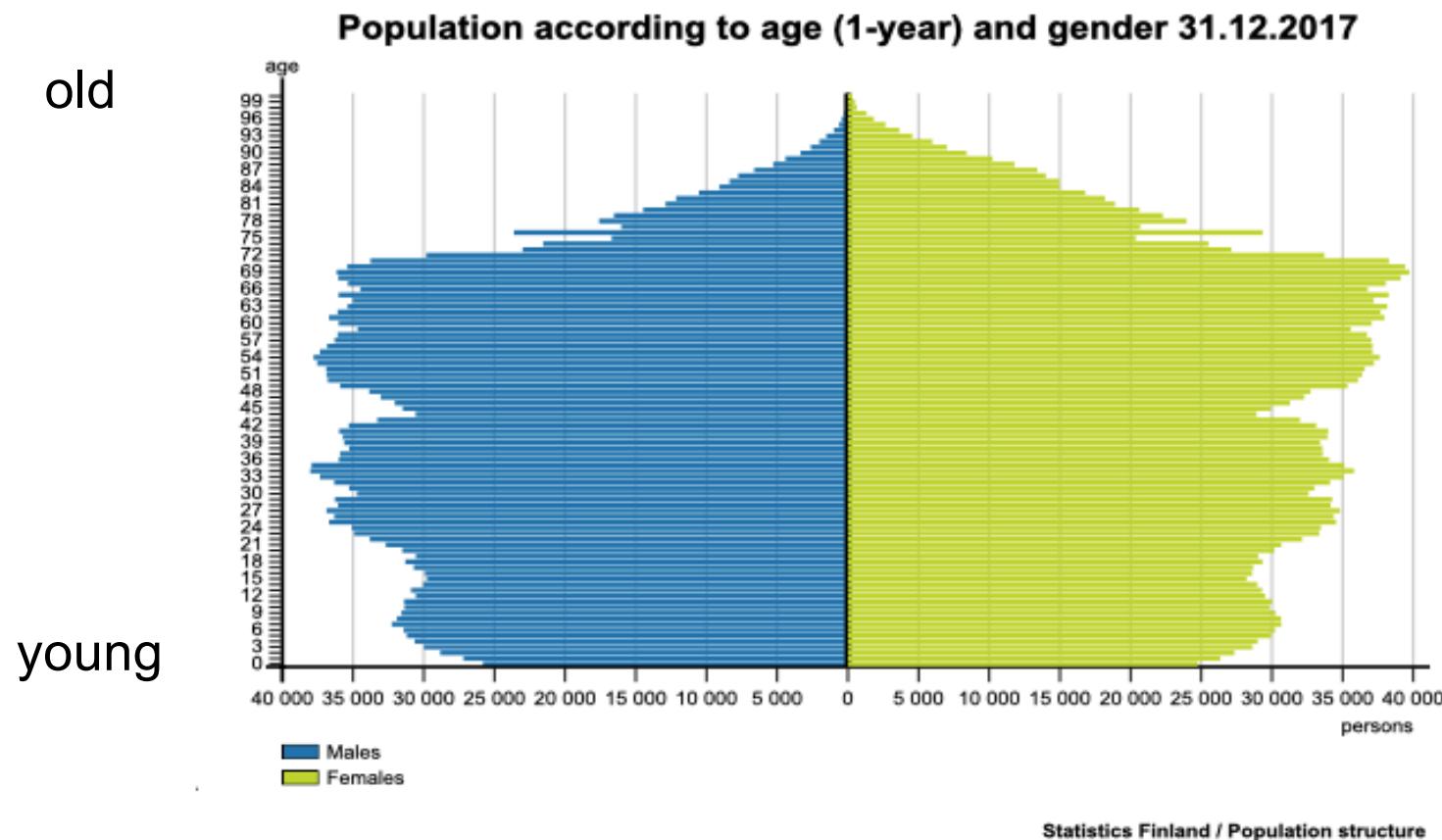
# Is Everything OK? Where is the Patient?



# Say “OK Google...“

- <https://www.youtube.com/watch?v=p5DVeDWtA5U>

# Age Distribution in Finland



# What are Laws of Motion?

<https://users.aalto.fi/~junga1/AppleFalling.mp4>

# I still don't speak Finnish 😞

Finnisch	Englisch
<p>Moi :) Sulla on mielenkiintoinen tutkimusalue. Me just töissä luokitellaan maankäytökarttoja Pakistanin ja testaillaan eri menetelmiä ;)</p> <p>Bearbeiten</p>	<p>Hi :) You have an interesting research area. We just categorize land use maps in Pakistan and test different methods;)</p>

In Google Übersetzer öffnen

Feedback

# **Reliable AI**

How much can we trust the prediction ?

# Understandable AI

- why this particular prediction?
- e.g.: automated decision for KELA benefits
- decisions require detailed justification based on law

# Privacy Preserving AI

- how to learn from data without revealing sensitive data?

## In a Nutshell

ML methods **fit** (complicated) **models** to  
(tons of) **data**

# Nice Ski Day Ahead!



# A ML Problem

- plan a decent ski trip
- which wax should I use?
- depends on temperature during day
- know the min temperature (its 07:00 am)
- what will be max temperature?



# Data

The screenshot shows the 'Download observations' section of the Finnish Meteorological Institute's website. At the top, there is a navigation bar with links for Home, Weather and sea, Climate, Services and products, Scientific themes, Research, and About us. Below the navigation bar, there is a search bar with a placeholder 'Enter location...' and a yellow search button with a magnifying glass icon. On the left side, there is a sidebar with links for Local weather, Weather and sea, Local weather, Marine weather and Baltic Sea, Warnings, Rain and cloudiness, Air quality, Download observations, Guidance to observations, Auroras and space weather, and Mobile weather and service numbers. The main content area has a heading 'Download observations' and a sub-section titled 'Choose parameters'. It includes a map of Finland with various data points and four buttons for 'Weather observations', 'Radiation observations', 'Marine observations', and 'Air quality observations'.

<https://en.ilmatieteenlaitos.fi/download-observations#!/>

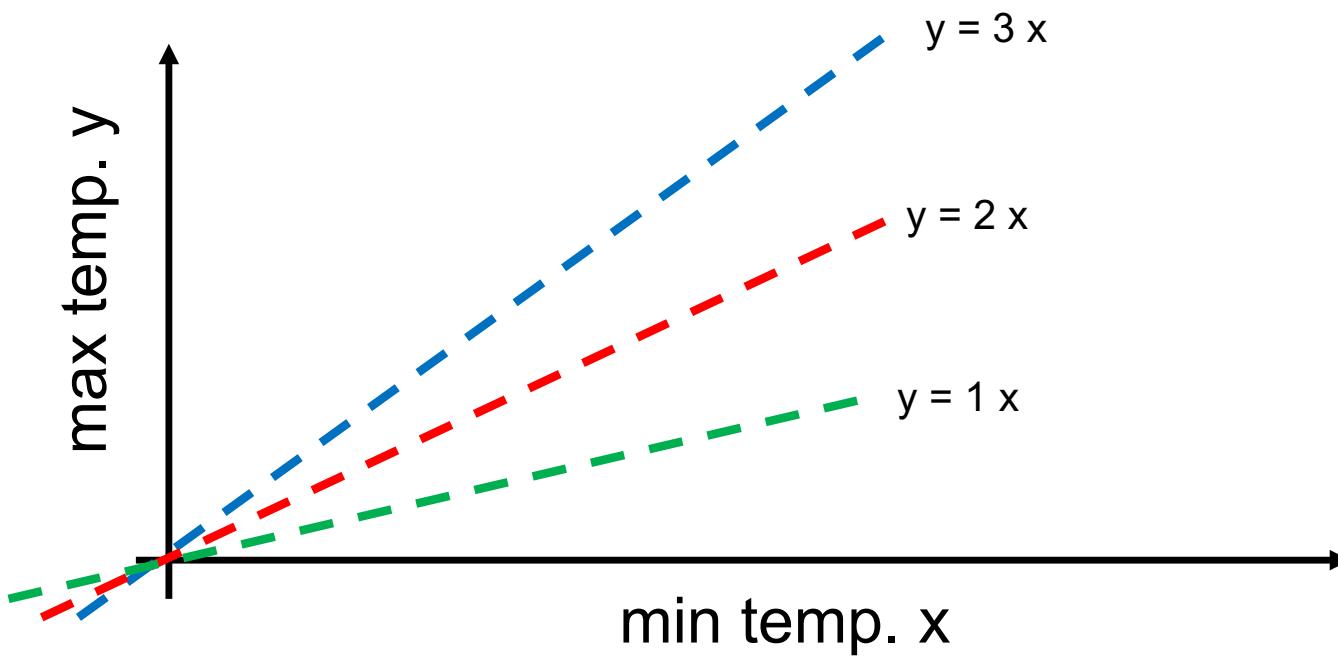
# Historic Data

	Year	m	d	Time	Prec	snow	tmp	maxtemp	mintmp
0	2016	1	2	00:00	-1.0	-1.0	-7.0	-5.5	-7.8
1	2016	1	3	00:00	3.2	-1.0	-8.7	-7.2	-10.0
2	2016	1	4	00:00	-1.0	4.0	-11.2	-4.9	-13.5
3	2016	1	5	00:00	0.6	4.0	-17.6	-13.3	-19.6
4	2016	1	6	00:00	-1.0	4.0	-20.3	-16.4	-21.3

data point

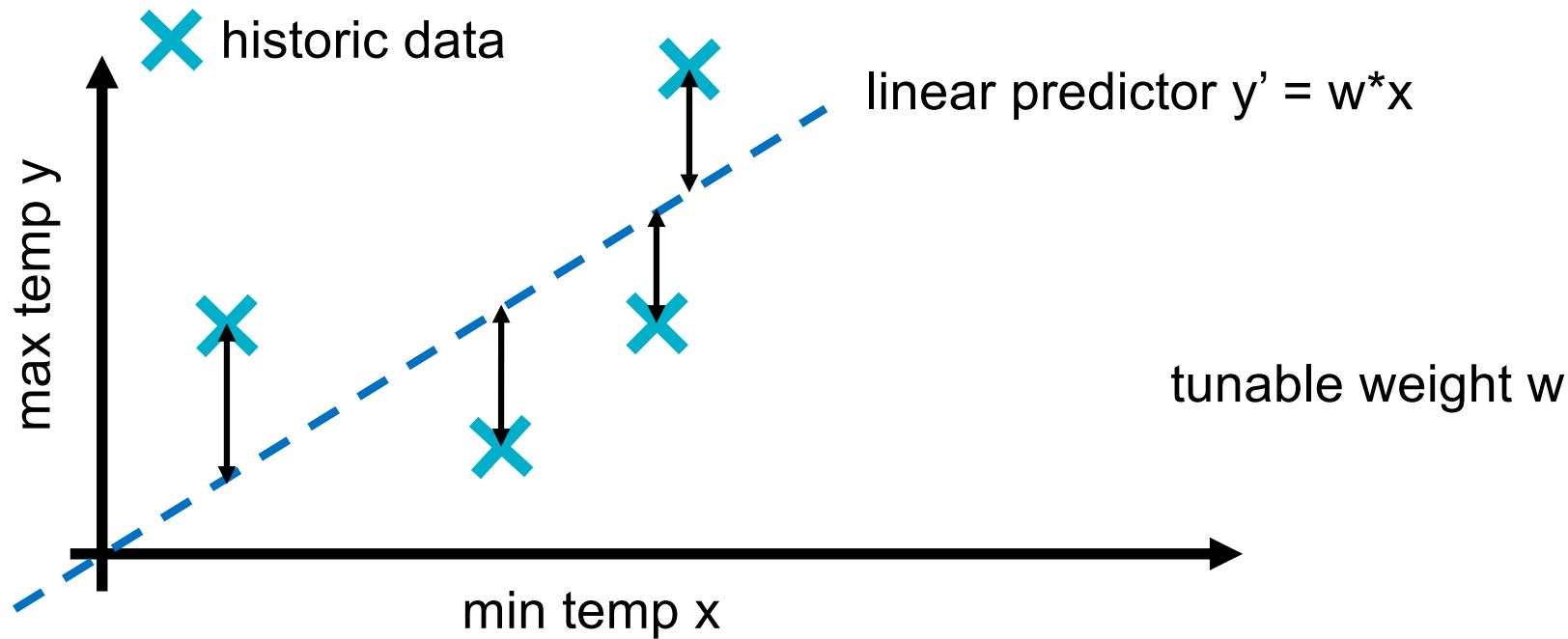
# Model

- hypothesis: linear relation between min and max tmp



# Fitting the Model to Data

- fit linear predictor  $y' = h(x)$  by minimizing mean squared error



**HOW TO FIT ?**

# ML Methods Available in Python/MATLAB/R/...

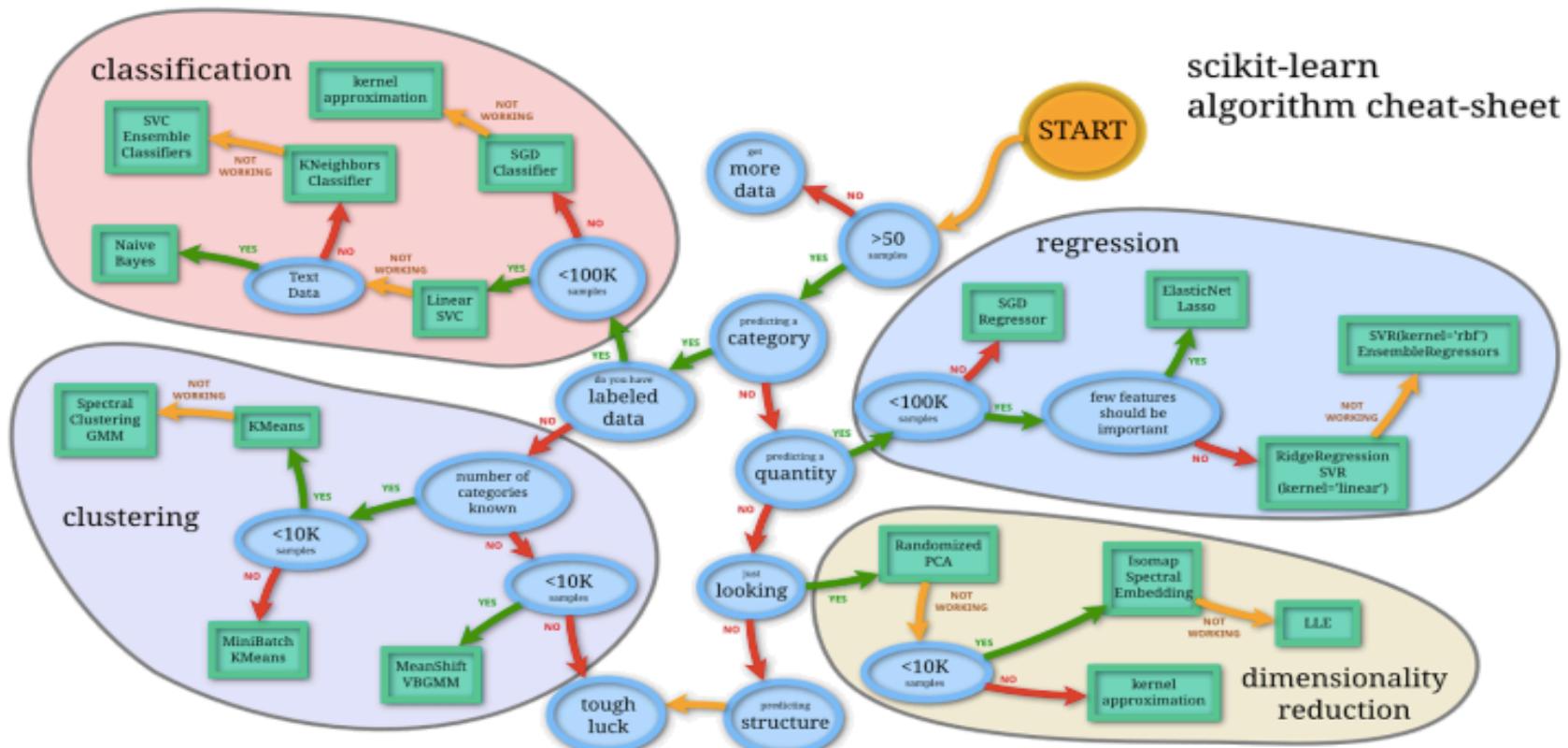


The screenshot shows the official scikit-learn website. At the top, there's a navigation bar with links for Home, Installation, Documentation, Examples, Google Custom Search, and a search bar. Below the navigation is a large blue header with the text "scikit-learn" and "Machine Learning in Python". To the left of the header is a grid of nine small heatmaps illustrating different machine learning datasets. The main content area is divided into several sections: Classification, Regression, Clustering, Dimensionality reduction, Model selection, and Preprocessing. Each section contains a brief description, applications, algorithms, and examples.

Classification	Regression	Clustering
Identifying to which category an object belongs to. <b>Applications:</b> Spam detection, Image recognition. <b>Algorithms:</b> SVM, nearest neighbors, random forest, ... — Examples	Predicting a continuous-valued attribute associated with an object. <b>Applications:</b> Drug response, Stock prices. <b>Algorithms:</b> SVR, ridge regression, Lasso, ... — Examples	Automatic grouping of similar objects into sets. <b>Applications:</b> Customer segmentation, Grouping experiment outcomes <b>Algorithms:</b> k-Means, spectral clustering, mean-shift, ... — Examples
Dimensionality reduction	Model selection	Preprocessing
Reducing the number of random variables to consider.	Comparing, validating and choosing parameters and models.	Feature extraction and normalization. <b>Application:</b> Transforming input data such as

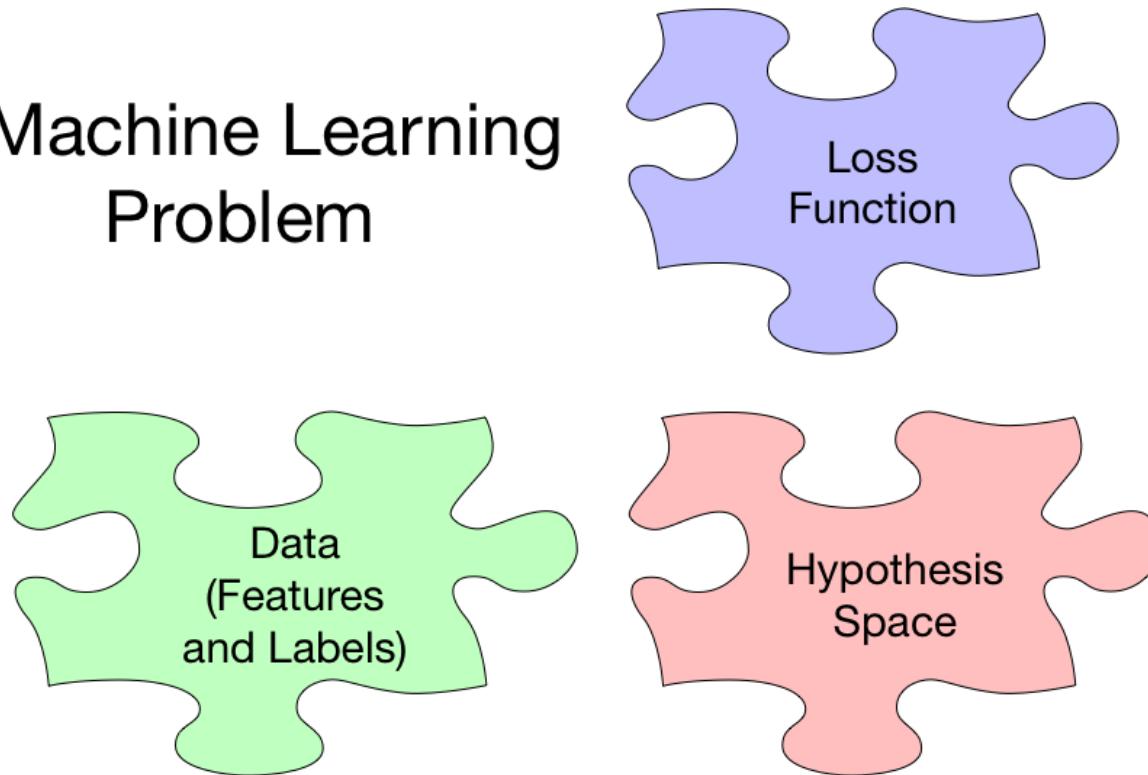
<https://scikit-learn.org/stable/>

# Python Cheat Sheet



# **Elements of Machine Learning**

## A Machine Learning Problem



# The Data

# Data



## Download observations

Instantaneous weather observations are available from 2010, daily and monthly observations from the 1960s onwards (depending on weather station).

Not every parameter is available from every station. If no data is found, please try the nearest stations.

[Frequently asked questions \(FAQ\)](#)

[Observation station list](#)

[Guidance to interpreting observations](#)

1 Choose parameters

Weather observations	Radiation observations	Marine observation
----------------------	------------------------	--------------------

2 Choose time period

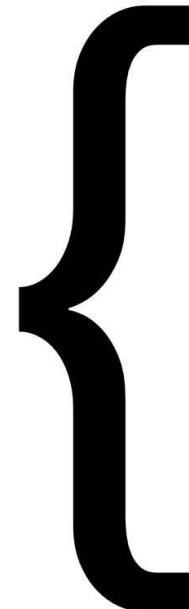
January 30, 2019 12:00 AM – January 30, 2019 11:59 PM

A screenshot of a user interface for downloading observations. It consists of two main sections. The first section, titled 'Choose parameters', contains three buttons labeled 'Weather observations', 'Radiation observations', and 'Marine observation'. The second section, titled 'Choose time period', features a date range selector with the text 'January 30, 2019 12:00 AM – January 30, 2019 11:59 PM' and a small calendar icon.

# Data Points

break raw data into atomic pieces (chunks)  
of data, e.g., individual files stored on hard disk

raw data



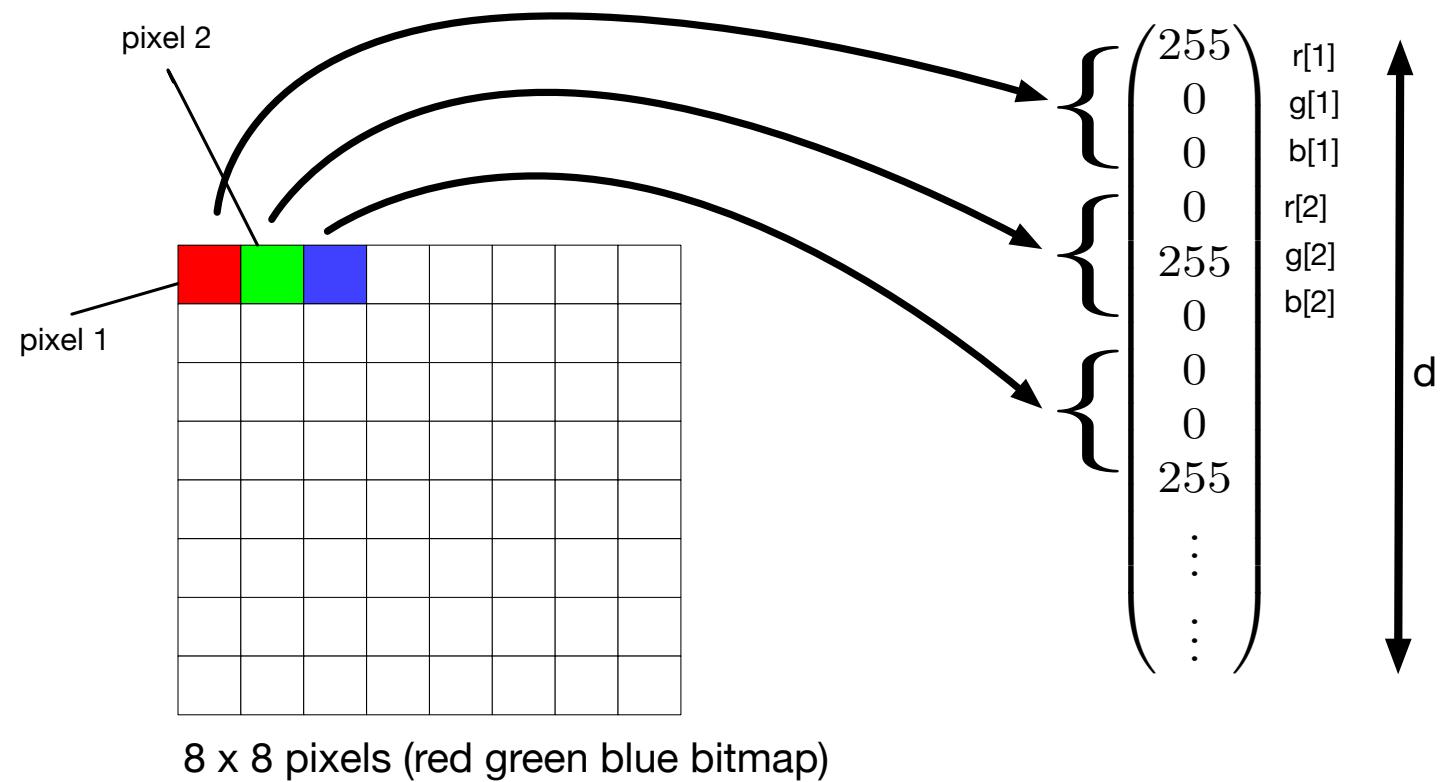
“data point”



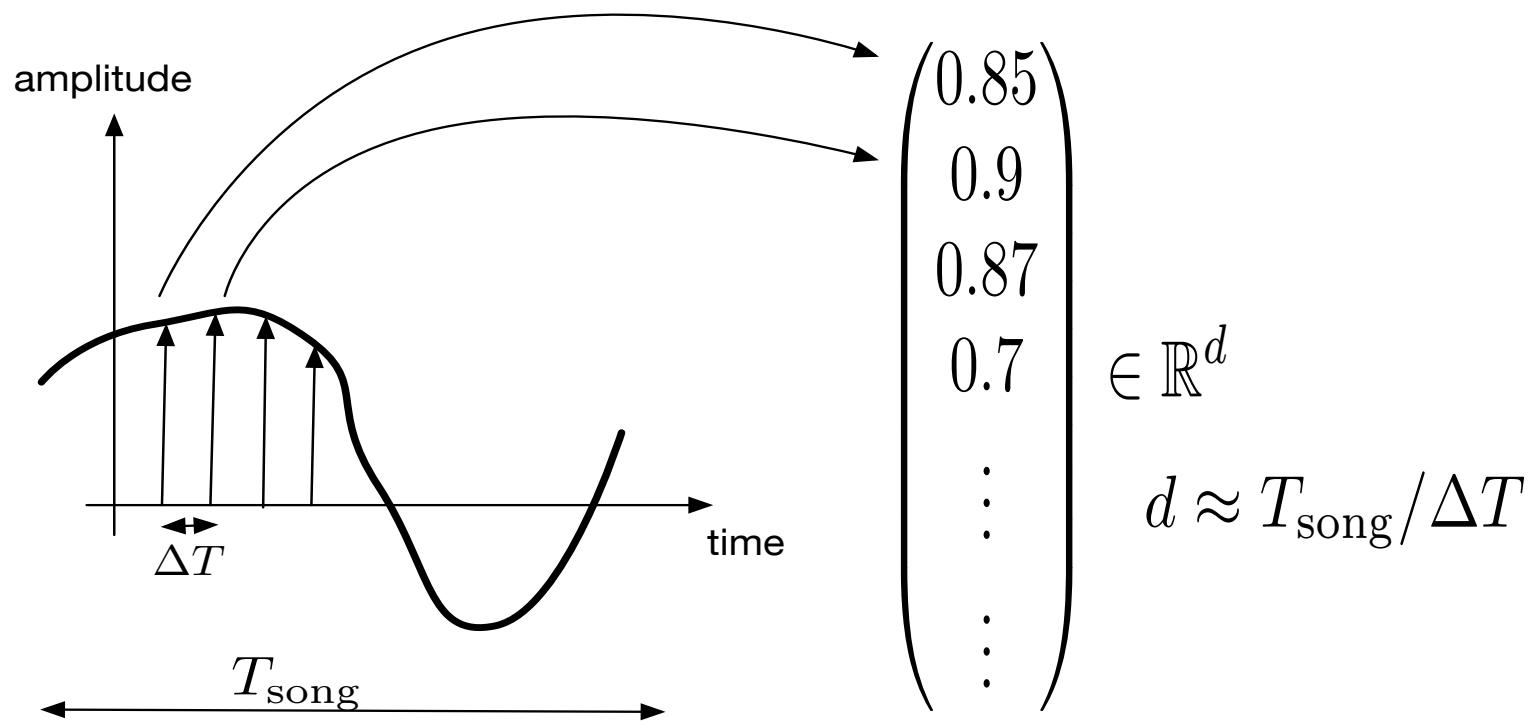
# Data Points

	Year	m	d	Time	Prec	snow	tmp	maxtemp	mintmp
0	2016	1	2	00:00	-1.0	-1.0	-7.0	-5.5	-7.8
1	2016	1	3	00:00	3.2	-1.0	-8.7	-7.2	-10.0
2	2016	1	4	00:00	-1.0	4.0	-11.2	-4.9	-13.5
3	2016	1	5	00:00	0.6	4.0	-17.6	-13.3	-19.6
4	2016	1	6	00:00	-1.0	4.0	-20.3	-16.4	-21.3

# Features for Image Data



# Features for Time Signals



# Features for Text

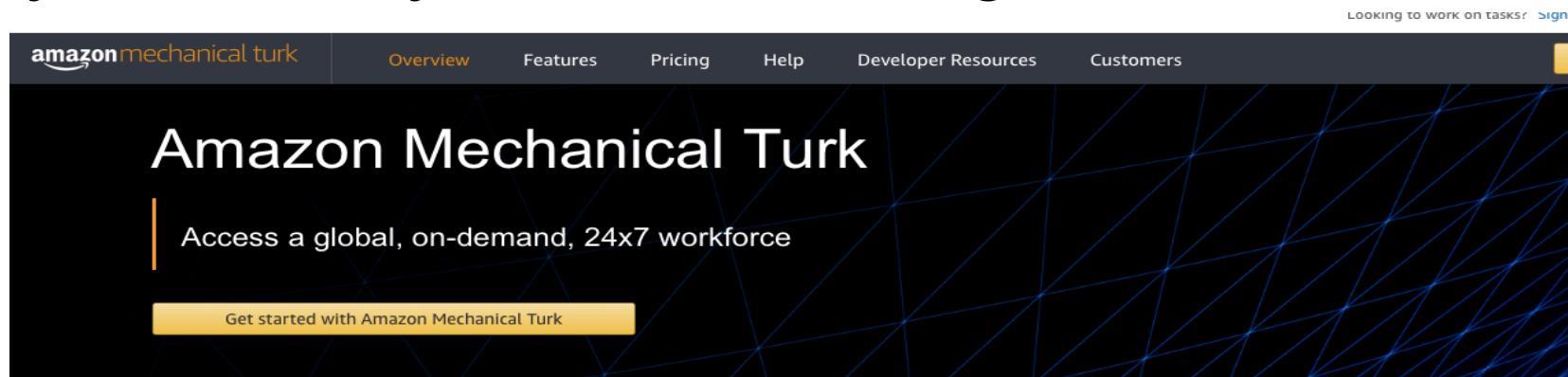
- consider simple language with vocabulary {"have", "a", "good", "great", "day"}
- represent words by "one-hot encoding"
- "have"=(1,0,0,0,0); „a“=(0,1,0,0,0), ...., „day“=(0,0,0,0,1)
- what could be feature vector of text "have have a good great day"?

# Labeled Data

- labeled data consists of data points with known label
- denote labeled data point with feature  $x$  and label  $y$  as  $(x,y)$
- $n$  labeled data points  $(x^{(1)}, y^{(1)}), \dots, (x^{(n)}, y^{(n)})$
- sample size  $n$  is important parameter of the ML problem!
- if data is new oil, then labeled data is premium fuel!

# How To Get Labeled Data?

- you can buy human labelling workforce!



Amazon Mechanical Turk (MTurk) is a crowdsourcing marketplace that makes it easier for individuals and businesses to outsource their processes to a distributed workforce who can perform these tasks virtually. This could include anything from conducting simple data validation and research to subjective tasks like survey participation, content moderation, and more. MTurk enables companies to harness the collective intelligence, skills, and ingenuity of a global workforce to streamline business processes, augment data collection and analysis, and accelerate machine learning development.

While technology continues to improve, there are still many things that human beings can do much more effectively than computers, such as moderating content, performing data deduplication, or research. Traditionally, tasks like this have been accomplished by hiring a large temporary workforce, which

# Labeled Data

	Year	m	d	Time	Prec	snow	tmp	maxtemp	mintmp
0	2016	1	2	00:00	-1.0	-1.0	-7.0	-5.5	-7.8
1	2016	1	3	00:00	3.2	-1.0	-8.7	-7.2	-10.0
2	2016	1	4	00:00	-1.0	4.0	-11.2	-4.9	-13.5
3	2016	1	5	00:00	0.6	4.0	-17.6	-13.3	-19.6
4	2016	1	6	00:00	-1.0	4.0	-20.3	-16.4	-21.3

label

# Missing Data

	Year	m	d	Time	Prec	snow	tmp	maxtemp	mintmp
0	2016	1	2	00:00	-1.0	-1.0	-7.0	-5.5	-7.8
1	2016	1	3	00:00	3.2	-1.0	-8.7	-7.2	-10.0
2	2016	1	4	00:00	-1.0	4.0	-11.2	Nan	-13.5
3	2016	1	5	00:00	0.6	4.0	-17.6	-13.3	-19.6
4	2016	1	6	00:00	-1.0	4.0	-20.3	-16.4	-21.3

there are ML “autofill” tools known as “data imputation”

# Labeled Data

raw data



{

“no cat”



“cat”



“cat”



“no cat”

# Classification Problem

	Year	m	d	Time	Prec	snow	tmp	maxtemp	mintmp
0	2016	1	2	00:00	-1.0	-1.0	-7.0	-5.5	-7.8
1	2016	1	3	00:00	3.2	-1.0	-8.7	-7.2	-10.0
2	2016	1	4	00:00	-1.0	4.0	-11.2	-4.9	-13.5
3	2016	1	5	00:00	0.6	4.0	-17.6	-13.3	-19.6
4	2016	1	6	00:00	-1.0	4.0	-20.3	-16.4	-21.3

will max temp today be more than 0 or not ?

# Classification Problem



does image on the left show “cat” or “no cat” ?

we can encode this by defining label of an image  
as  $y=1$  if it shows cat and  $y=-1$  if not

how can we get the true label  $y$  for an image?

# Regression Problem

	Year	m	d	Time	Prec	snow	tmp	maxtemp	mintmp
0	2016	1	2	00:00	-1.0	-1.0	-7.0	-5.5	-7.8
1	2016	1	3	00:00	3.2	-1.0	-8.7	-7.2	-10.0
2	2016	1	4	00:00	-1.0	4.0	-11.2	-4.9	-13.5
3	2016	1	5	00:00	0.6	4.0	-17.6	-13.3	-19.6
4	2016	1	6	00:00	-1.0	4.0	-20.3	-16.4	-21.3

what will maxtemp be today (mintmp was -5 )?

# Regression vs. Classification

- AI engineer defines what the label is !
- numeric label (e.g., max temp) yield regression problems
- discrete label (e.g., “cat” vs. “dog”) yields classification
- distinction between regression and classification is blurry!
- “cat” vs. “dog” classification can be modeled as regression problem with numeric label  $y$ =“confidence that image shows rather cat than dog”

# The Data – Wrap Up

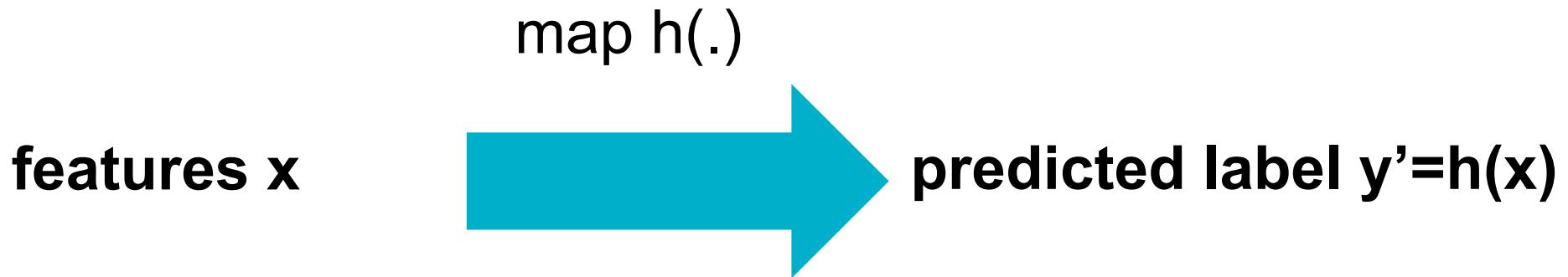
- collection of atomic data units “data points”
- data point characterized by features and labels
- **features** are quantities that can be **determined easily**
- **labels** are difficult to determine (requires **human labor**)
- ML methods **guess label based on features**

# Hypothesis Space (“Model”)

# Hypothesis Space

- need hypothesis (model) relating features (input) and label (output)
- represent hypothesis as map or function  $h(x)$
- input features  $x$  and output predicted label  $h(x)$
- other names for hypothesis map are
  - “predictor” for numeric labels (e.g., temperature next day)
  - “classifier” discrete labels ( e.g., image shows “cat” or “dog”)

**“Hypothesis”/“Predictor”/“Classifier” = Map from Features to Labels**



choose  $h(\cdot)$  s.t. “predicted label  $\approx$  true label”

# Example of a Predictor



## JOB OPENINGS!

The Venngage team is currently looking for creative and self-starting candidates to fill the position of:

### DEVELOPER

#### JOB DESCRIPTION:

A front-end developer specializes in building the front end, or client-side, of a web application, which encompasses everything that a client, or user, sees and interacts with.

### PROGRAM ASSISTANT

#### JOB DESCRIPTION:

A Program Assistant provides operational and administrative assistance to the Program Leader and Program Staff, performs a variety of administrative, coordination and logistical services in support of the operations of the Program, and assists with information management the team.

#### REQUIRED SKILLS:

- Javascript
- CSS
- HTML



#### REQUIRED SKILLS:

- Organized
- Meets deadlines
- Multi-tasking



## APPLY NOW!

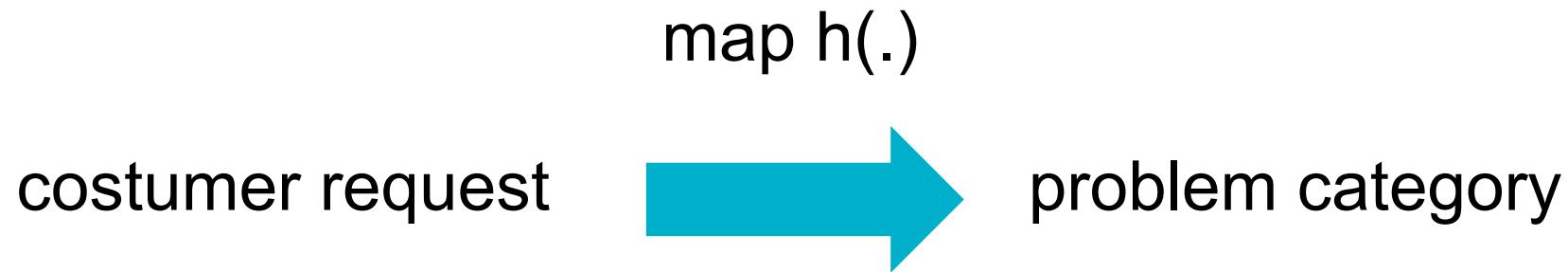
These are NOT real job opportunities at Venngage. This is just a template. Contact us at: [www.venngage.com](http://www.venngage.com)

map h(.)



optimal candidate  
“Kalle Koivunen”

# Example of a Predictor



# Example of a Predictor

current operational  
data of cruise ship



from Wikimedia Commons

map  $h(.)$



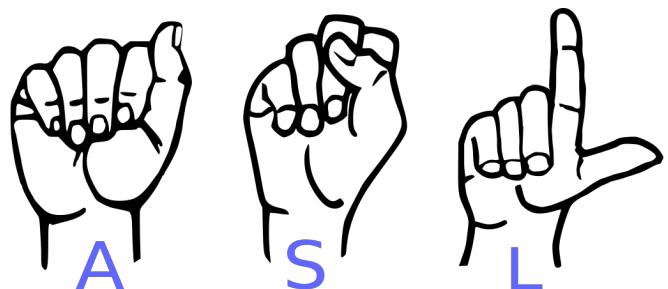
Schnitzel demand  
at next dinner?



from Wikimedia Commons

# Example of a Predictor

sequence of gestures



map  $h(.)$



sentence

“Please google ...”

book

from Wikimedia Commons

# Example of a Predictor Transaction

6.7.18	-62,20	PKORTTIMAKSU RISTORANTE AI DUE L AQUILEIA Viesti: 000432277*****6607 OSTOPVM 180704 MF NRO 74935008186868663596579 VARMENTAJA 400	844431 201807065 EL2148652 J
6.7.18	-218,00	PKORTTIMAKSU ERLEBNIS HOTEL PIRK LATSCACH Viesti: 000432277*****6607 OSTOPVM 180704 MF NRO 74548188186110001211713 VARMENTAJA 400	52100 201807065 EL2284885 J
6.7.18	46,00	PKORTTIMAKSU	570071 201807065 EL20005604 J

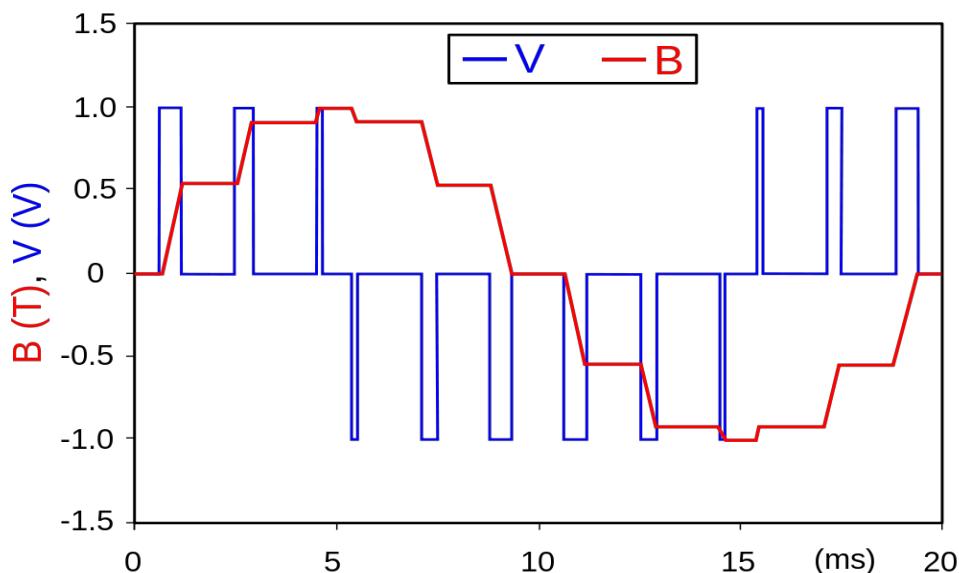
map h(.)



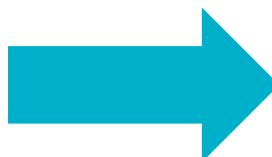
“No Money Laundry”

# Example of a Predictor

measurements



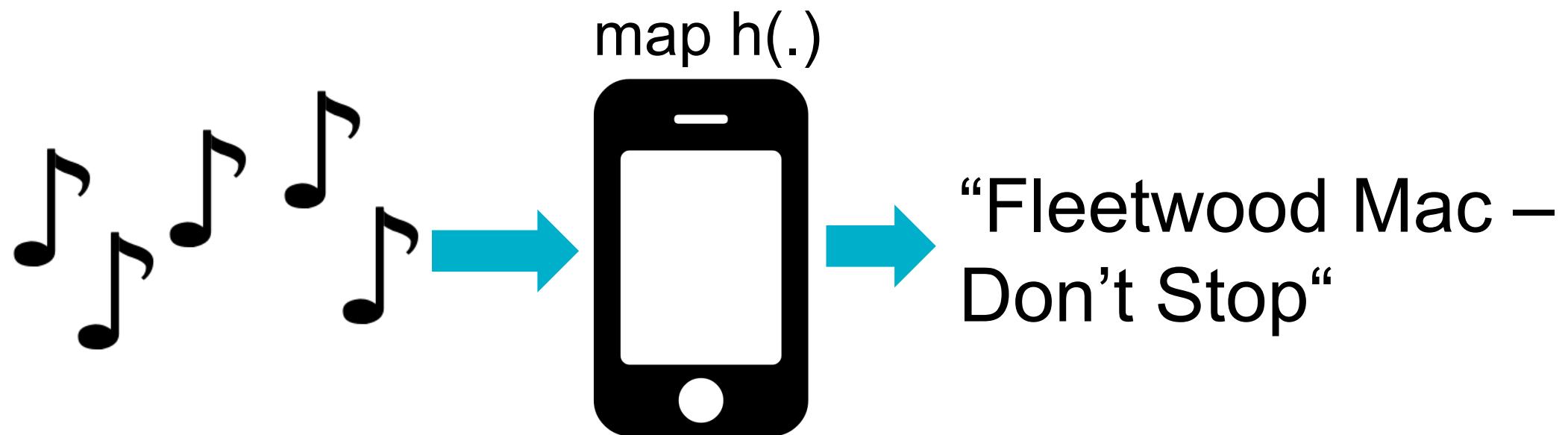
map  $h(\cdot)$



risk of fault  
within next week

60 %

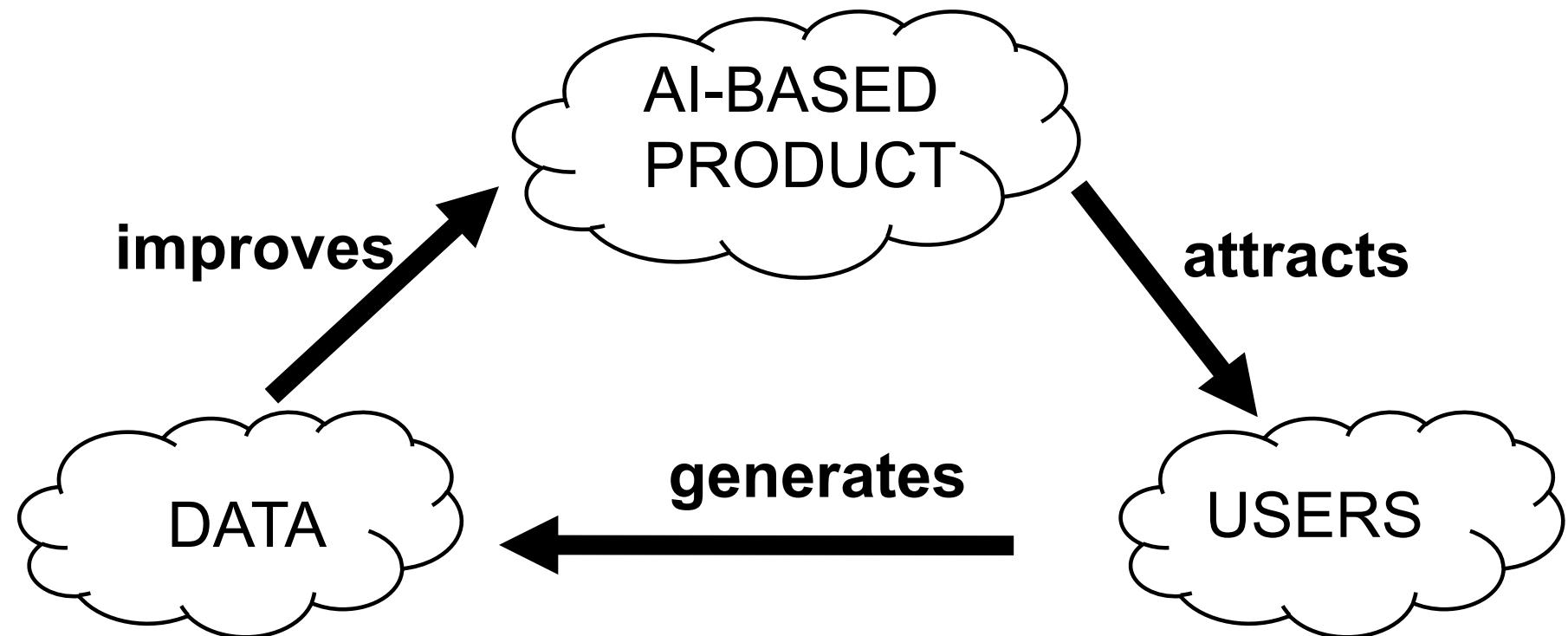
## Example of a Predictor



# From Predictions to Actions

- this module focuses on ML methods that deliver predictions
- predictions are further used e.g. to inform actions/decisions
- actions then result in new data (produced by customers)
- results in a feedback loop “The Virtuous AI Cycle”

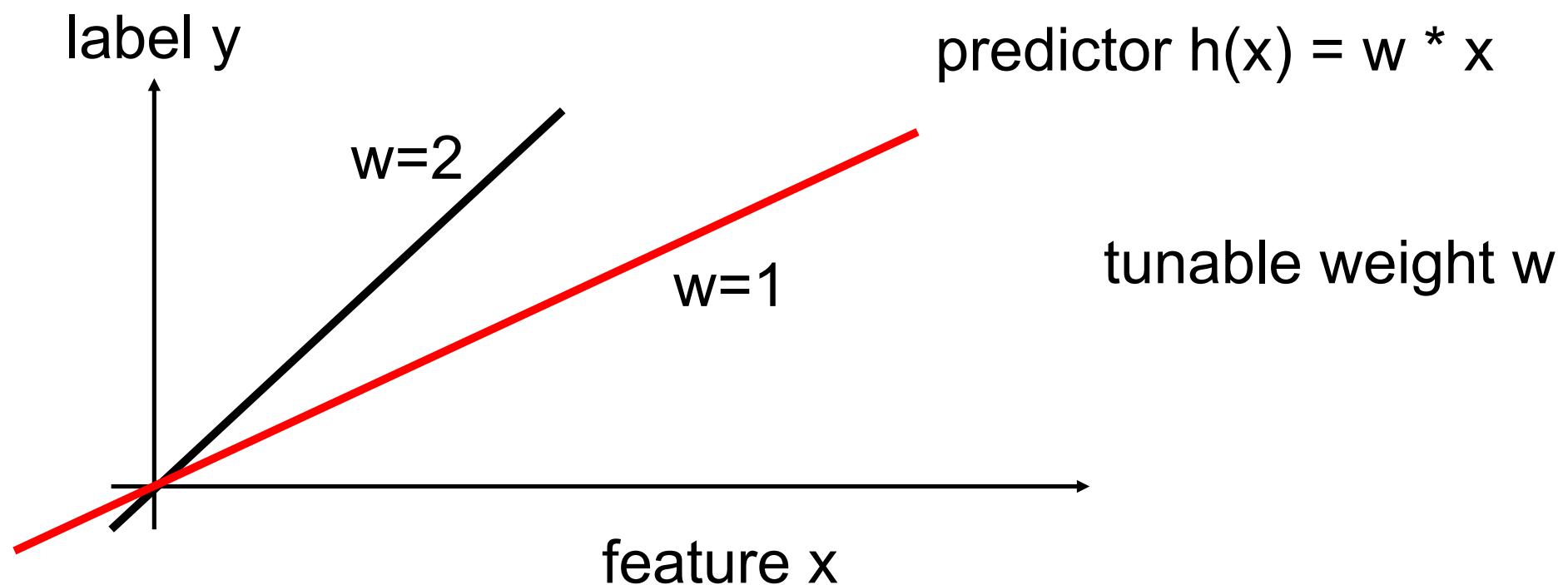
# The Virtuous AI Cycle



# How to Implement Predictors?

- predictor is a map  $h(x)$
- maps feature  $x$  to a predicted label  $y'$
- computing a prediction amounts to evaluation the map
- sounds trivial but can be challenging
- how can we efficiently represent a map  $h(x)$  ?

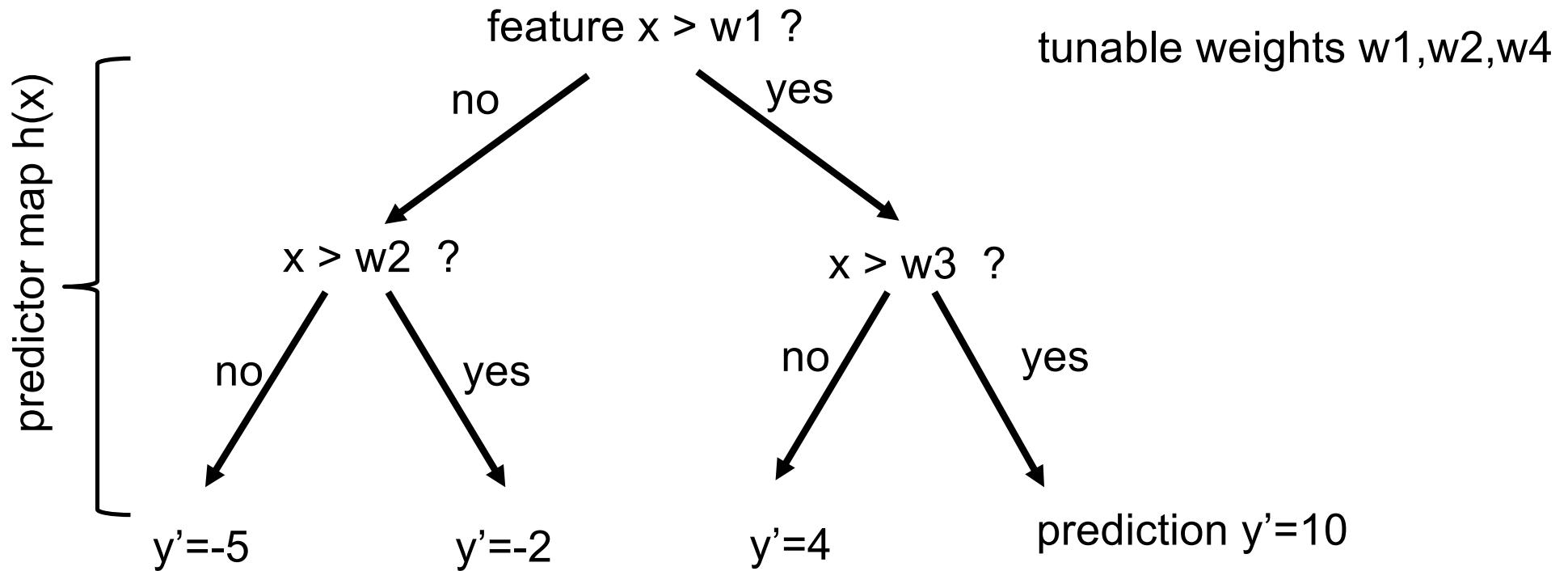
# Linear Predictors (Linear Regression)



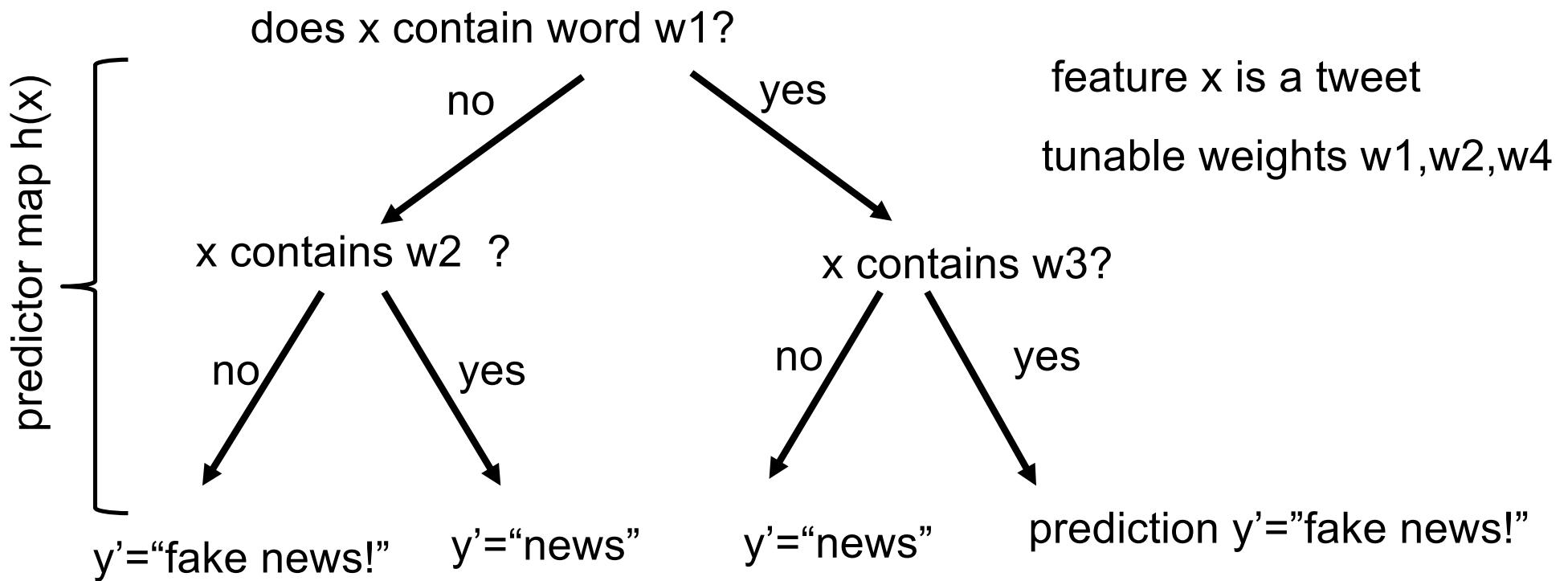
# Linear Predictors

- widely used
- lot of engineering effort to make ML methods efficient
- hardware (GPUs) is tailored to linear methods
- linear ML methods are a mature technology
- can be used directly only with numeric features (vectors)
- how to apply it to text ?

# Decision Trees



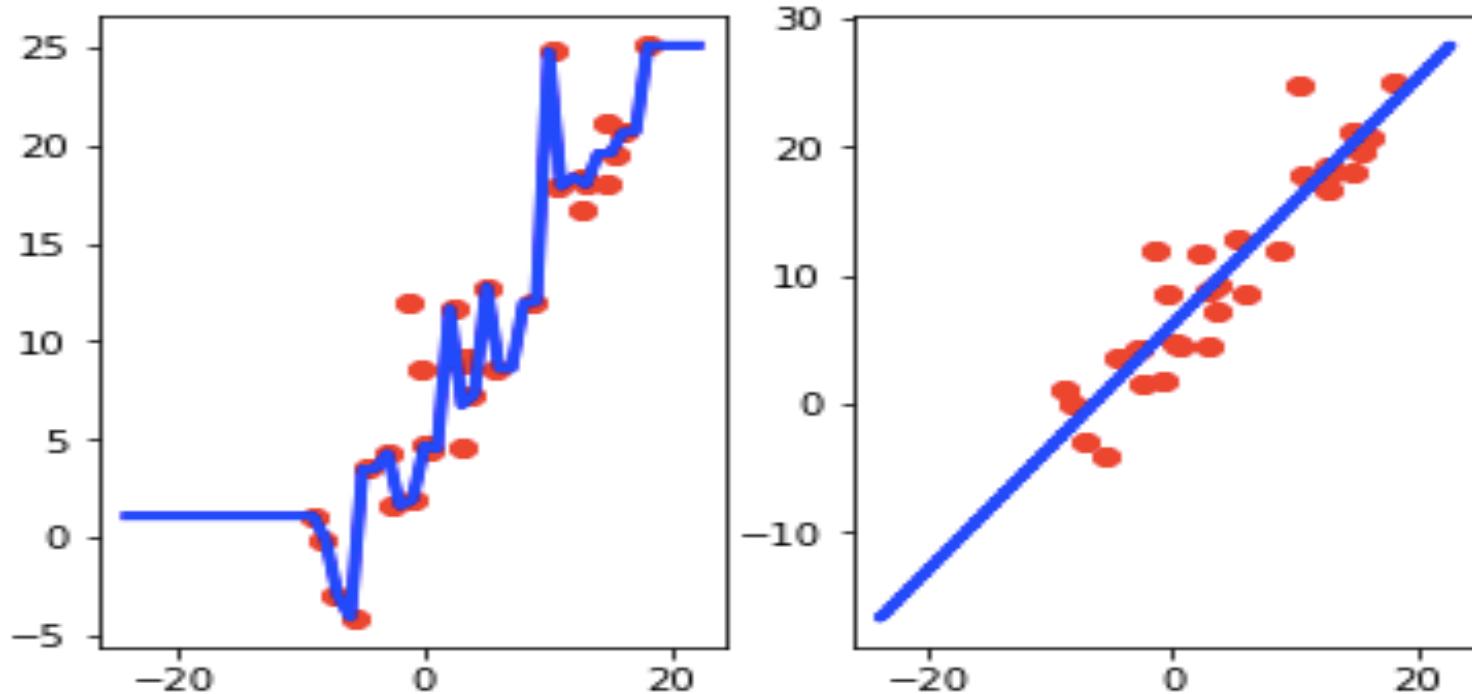
# Decision Trees for Text



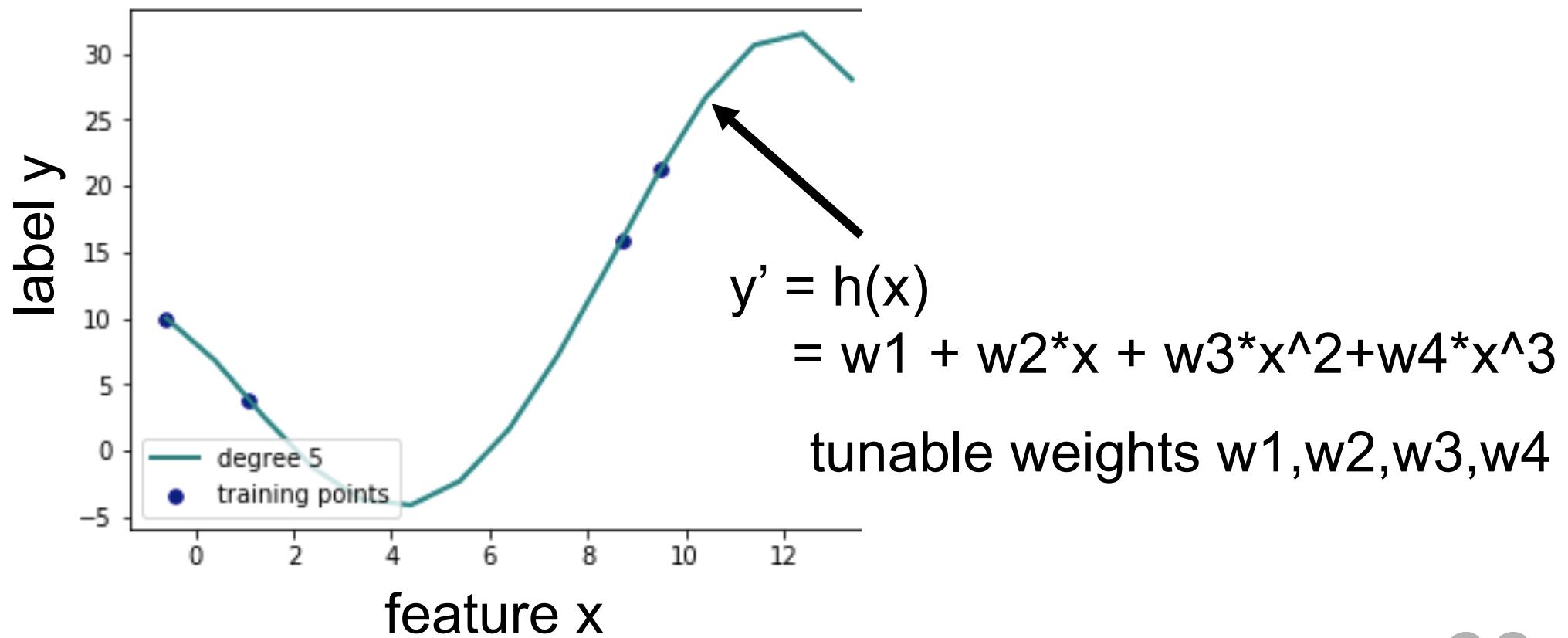
# Decision Trees

- predictions arise from sequence of elementary tests
- tests (e.g.  $x>5?$ ) provide explanation for prediction
- predict “tmp=10” because of  $x>5$  and  $x<20$  ....
- predict “benefit granted“ due to Fact 1 and Law 1 ...

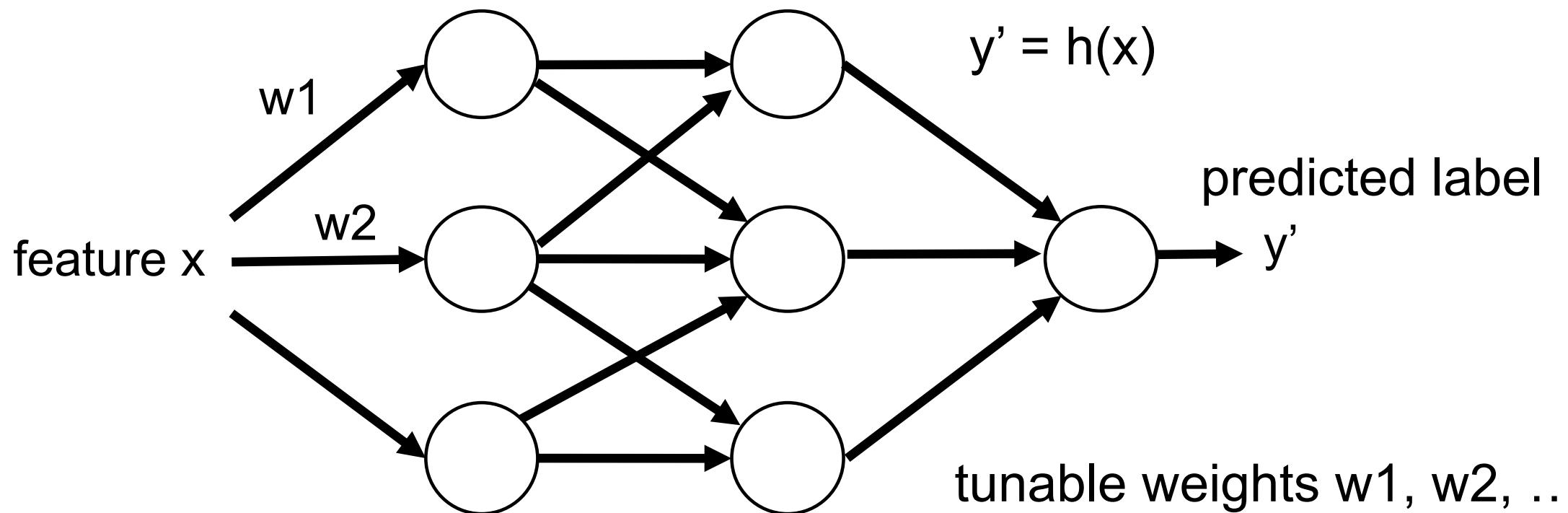
# Decision Tree vs. Linear Predictor



# Polynomial Predictor



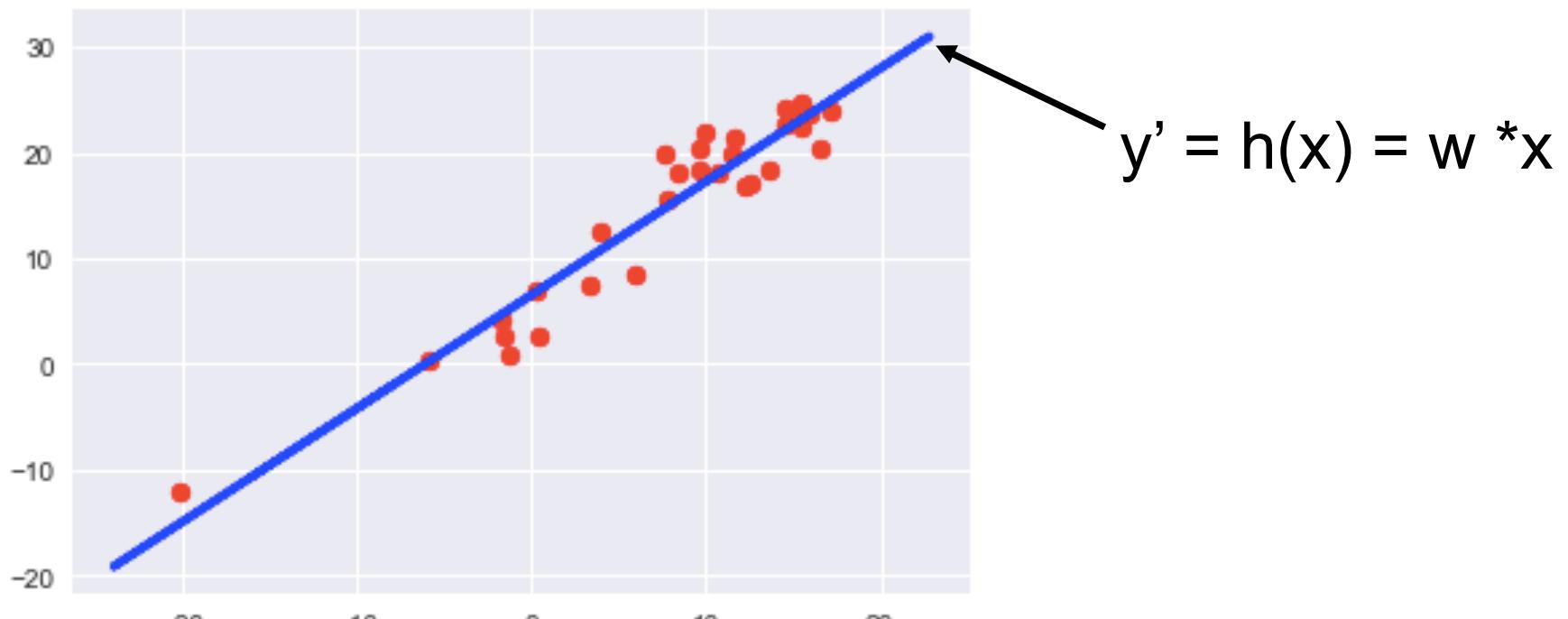
# Artificial Neural Networks



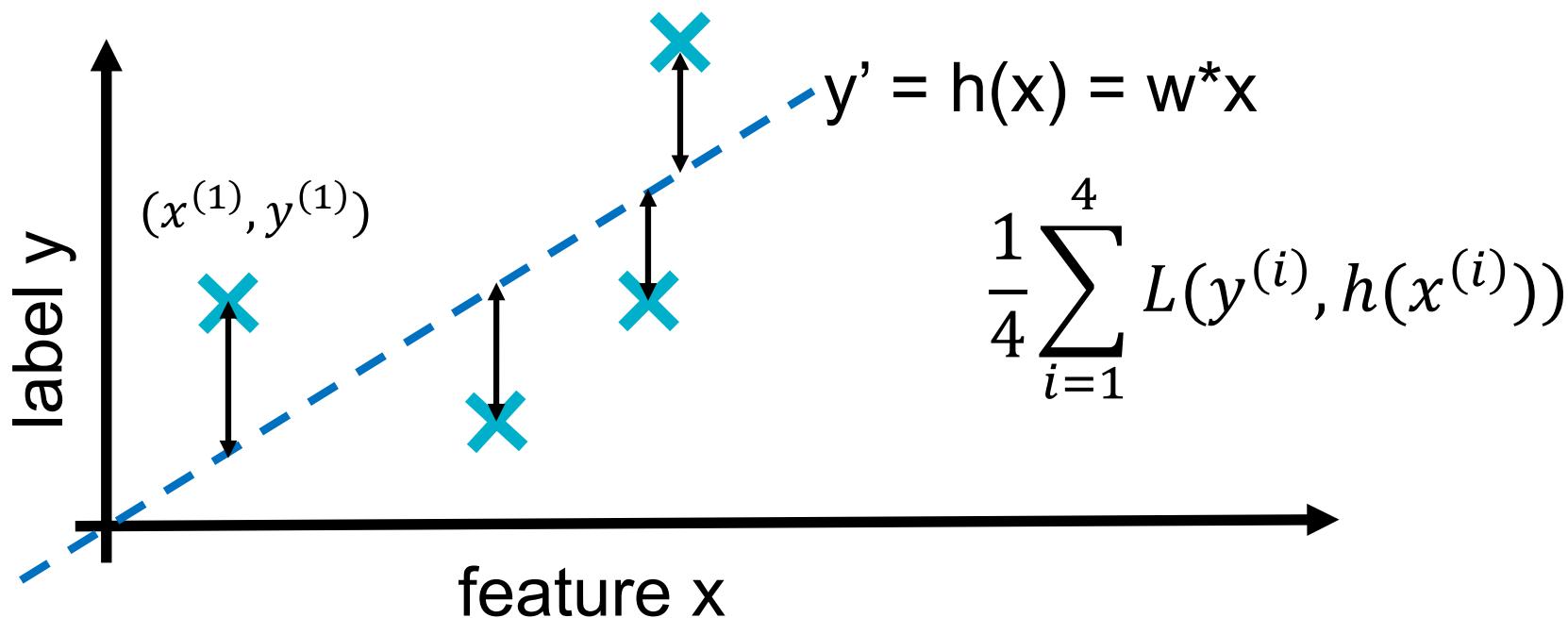
# The Loss Function

# How Good is a Predictor?

- define loss  $L(y, h(x))$  of predicting true label  $y$  by  $y' = h(x)$

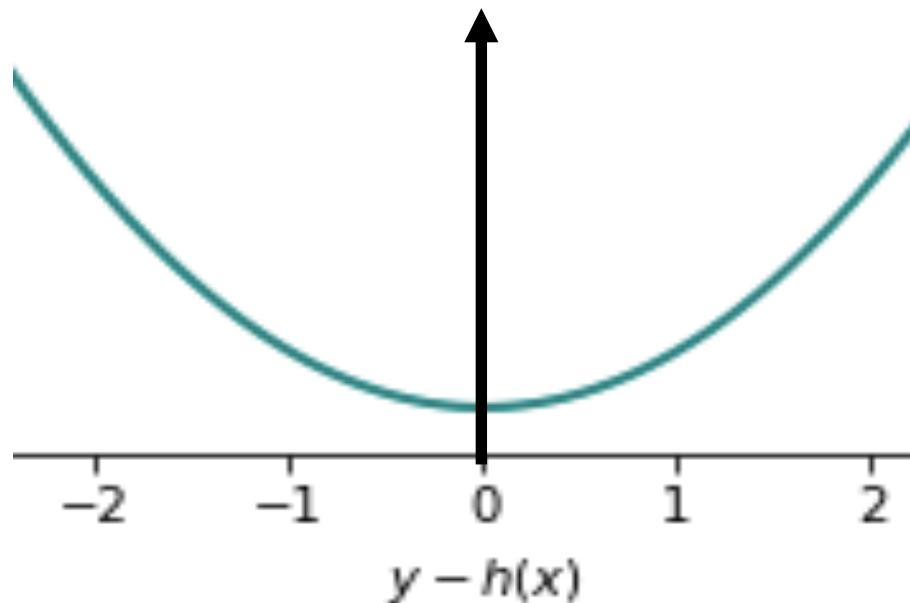


# Average Loss (Empirical Risk)



# Squared Error Loss

$$L(y, h(x)) = (y - h(x))^2$$



# Training Data = Bunch of Labeled Data

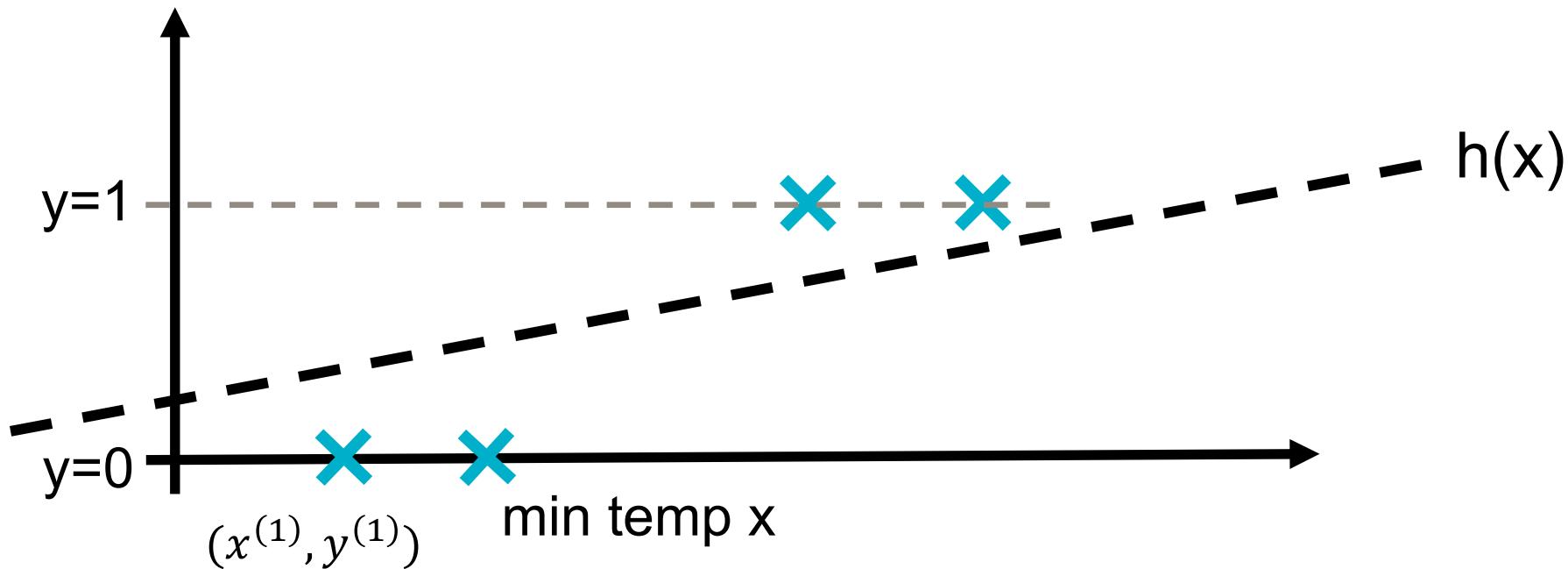
- evaluating loss  $L(y, h(x))$  requires label  $y$  of data point !
- need labeled data points  $(x^{(1)}, y^{(1)}), \dots$  (“training data”)
- assess predictor  $h(x)$  by average loss (or empirical risk)

$$\frac{1}{n} \sum_{i=1}^n L(y^{(i)}, h(x^{(i)}))$$

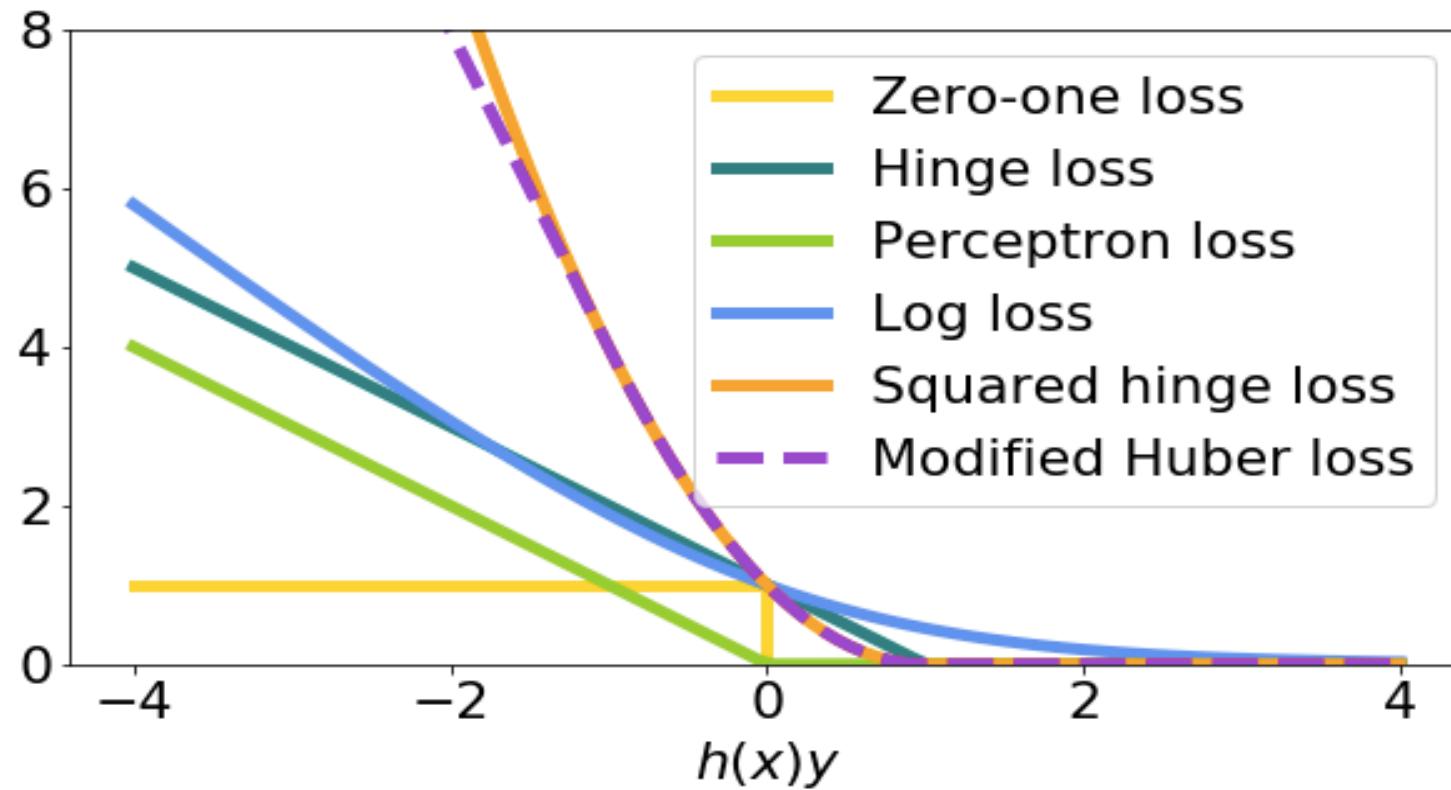
# Choosing Loss Function

- loss function is design choice (specified by AI engineer)
- for numeric labels, squared error loss is widely used
- sq. error has nice statistical and computational properties
- for discrete labels, squared error loss is not well suited

# Squared Error Loss Bad for Binary Labels



# Loss Function for Binary Labels $y=-1$ or $y=1$



# Loss Function for Binary Labels ( $y=-1$ or $y=1$ )

- consider AI for harvester
- camera snapshot is labeled  $y=1$  (-1) if **person present (absent)**
- loss  $L(y=1, y'=-1)$  should be very large since  **$y=1, y'=-1$  means there is ⚡ person but AI misses her**
- loss  $L(y=-1, y'=1)$  can be small (**AI sees person which is not there**)



# Loss Function Selection

## sklearn.tree.DecisionTreeRegressor

```
class sklearn.tree. DecisionTreeRegressor (criterion='mse', splitter='best', max_depth=None,
min_samples_split=2, min_samples_leaf=1, min_weight_fraction_leaf=0.0, max_features=None, random_state=None,
max_leaf_nodes=None, min_impurity_decrease=0.0, min_impurity_split=None, presort=False) [source]
```

A decision tree regressor.

Read more in the [User Guide](#).

**Parameters:** `criterion : string, optional (default="mse")`

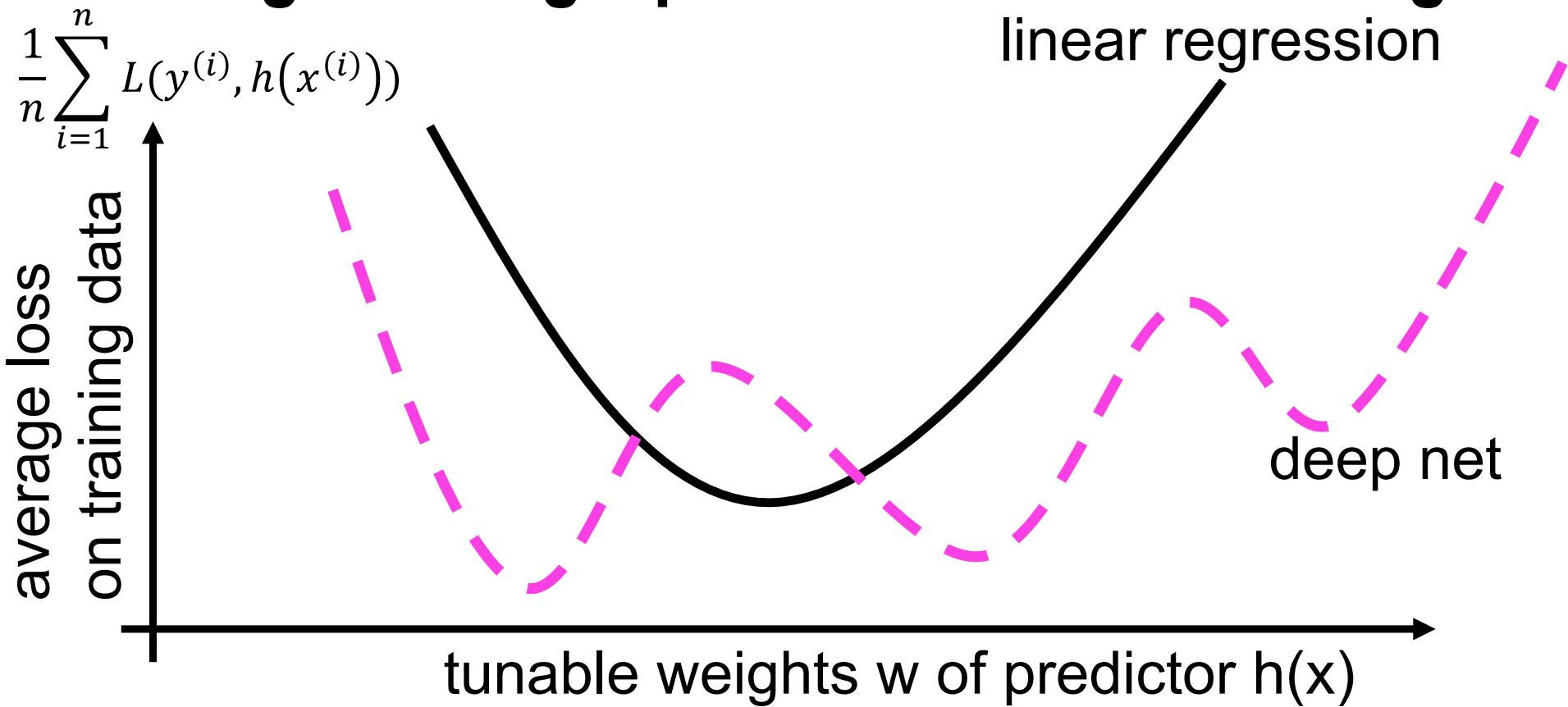
The function to measure the quality of a split. Supported criteria are “mse” for the mean squared error, which is equal to variance reduction as feature selection criterion and minimizes the L2 loss using the mean of each terminal node, “friedman\_mse”, which uses mean squared error with Friedman’s improvement score for potential splits, and “mae” for the mean absolute error, which minimizes the L1 loss using the median of each terminal node.

*New in version 0.18:* Mean Absolute Error (MAE) criterion.

# Putting Together the Pieces

- collect/measure/compute labeled data points
- choose predictor (linear, tree, artificial neural net, ...)
- choose loss function
- tune weights of predictor to minimize the average loss

# Learning/Training/Optimization/Model Fitting



# Model Selection

80

# Model = Choice for Hypothesis Space

- ML method/model = family of predictors with same structure
- linear regression uses  $h(x) = w^*x$  with tunable  $w$
- family of predictors constitutes the hypothesis space

# Example: Linear Regression

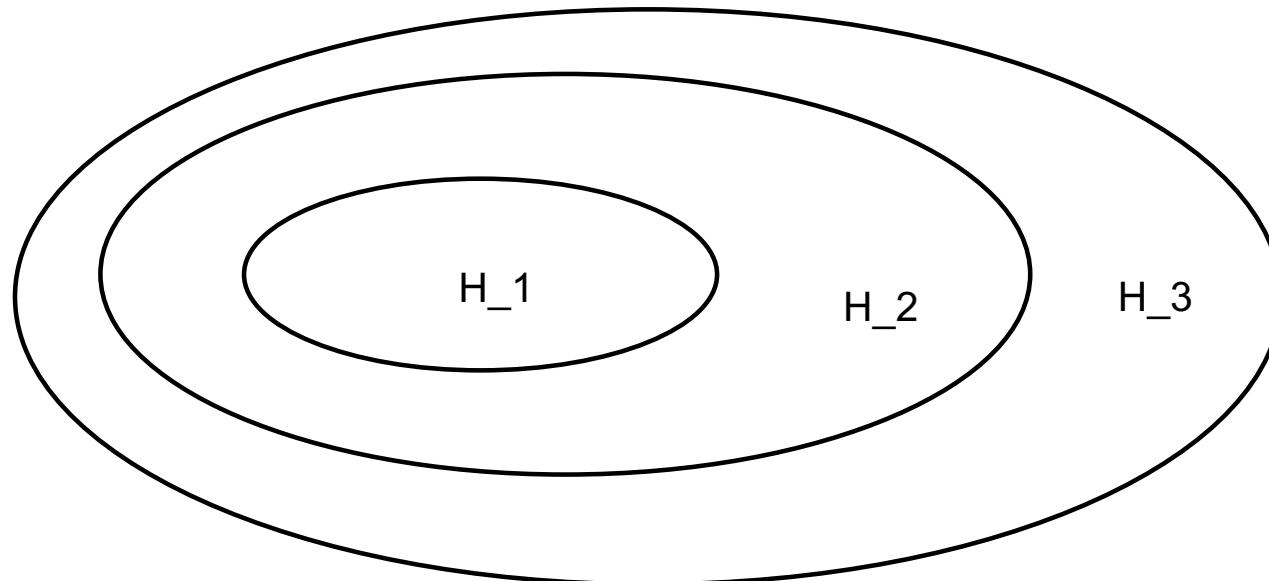
- consider data point having four features  $x_1, \dots, x_4$
- want to predict some numeric label  $y$  of data point
- linear regression relates features and label via  
$$y' = h(x) = w_1 * x_1 + w_2 * x_2 + w_3 * x_3 + w_4 * x_4$$
- different tuning weights  $w_1, \dots, w_4$  yield different predictors !

# Which Model Shall We Use?

- decision trees
- (deep) artificial neural nets
- linear predictors
- polynomial predictors
- ....

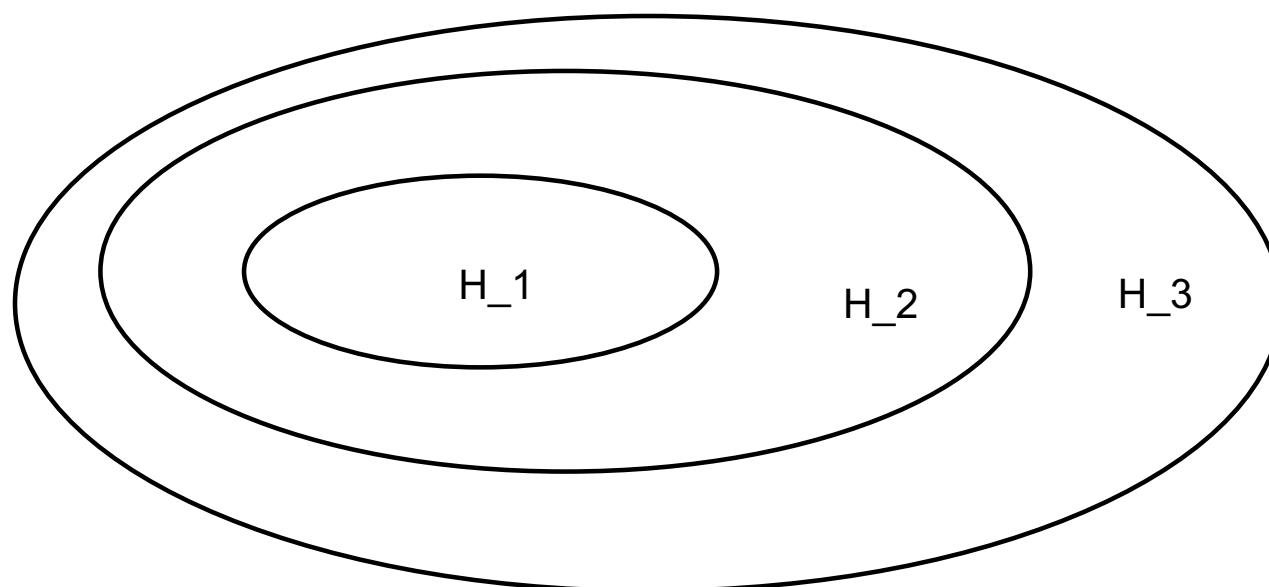
# Nested Models

- $H_d$  = set of linear predictors which use first  $d$  features



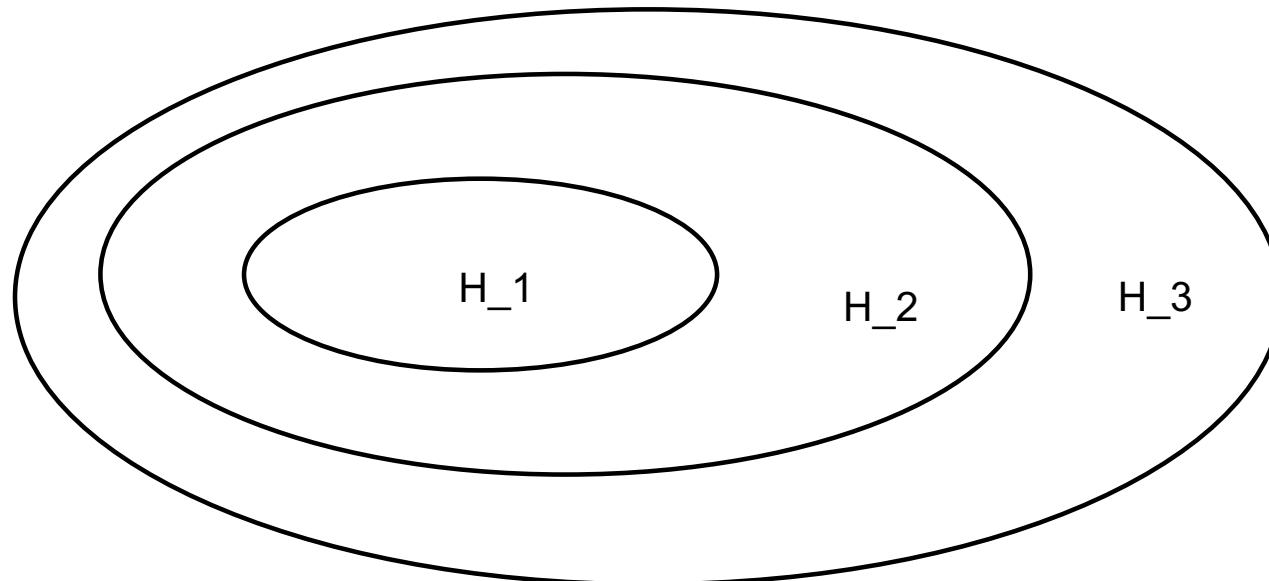
# Nested Models

- define model  $H_d$  by set of polynomials with max. degree  $d$
- $H_d$  included in  $H_e$  if  $d \geq e$



# Nested Models

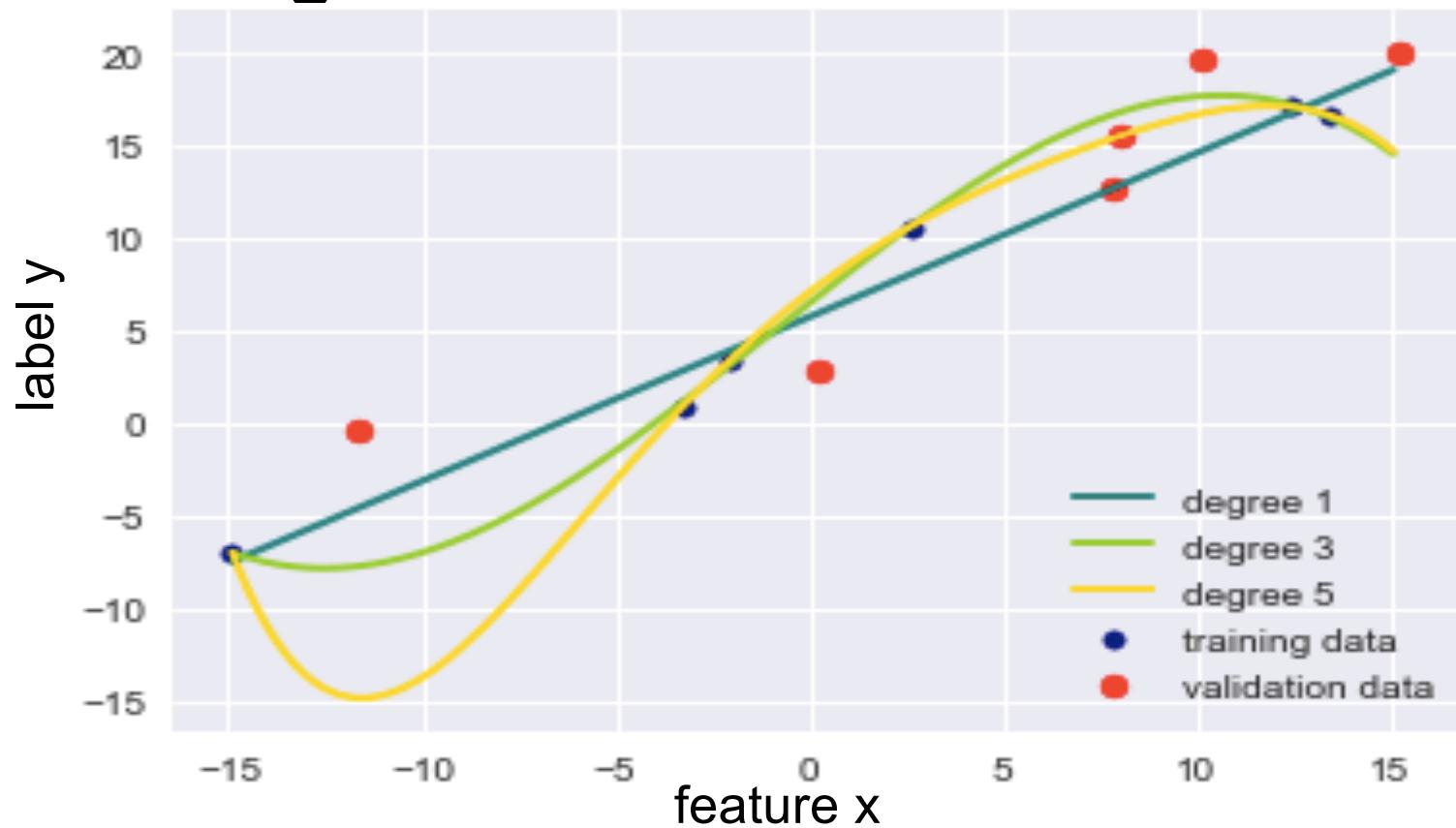
- model  $H_d$  given by ANN with  $d$  hidden units



# Training Error

- consider particular model (e.g., polynomial predictor with max. degree  $d$ )
- model constituted by maps with different choices for tuning weights  $w$
- choose weights by minimizing average loss (training error) over training set
- if we use larger model, we can only reduce training error !
- degree 5 polyn. can fit training data at least as good as degree 3 polyn.
- why not simply using the largest possible model (polyn. with large degree)?

# Overfitting



# Validate!

- average loss  $E_{\text{train}}$  on training set can be misleading!
- indeed, we optimize the predictor based on training set
- after training, try out on different (validation) data!
- compare training error  $E_{\text{train}}$  with validation error  $E_{\text{val}}$
- when  $E_{\text{train}} \ll E_{\text{val}}$ , ML model is overfitting !!!

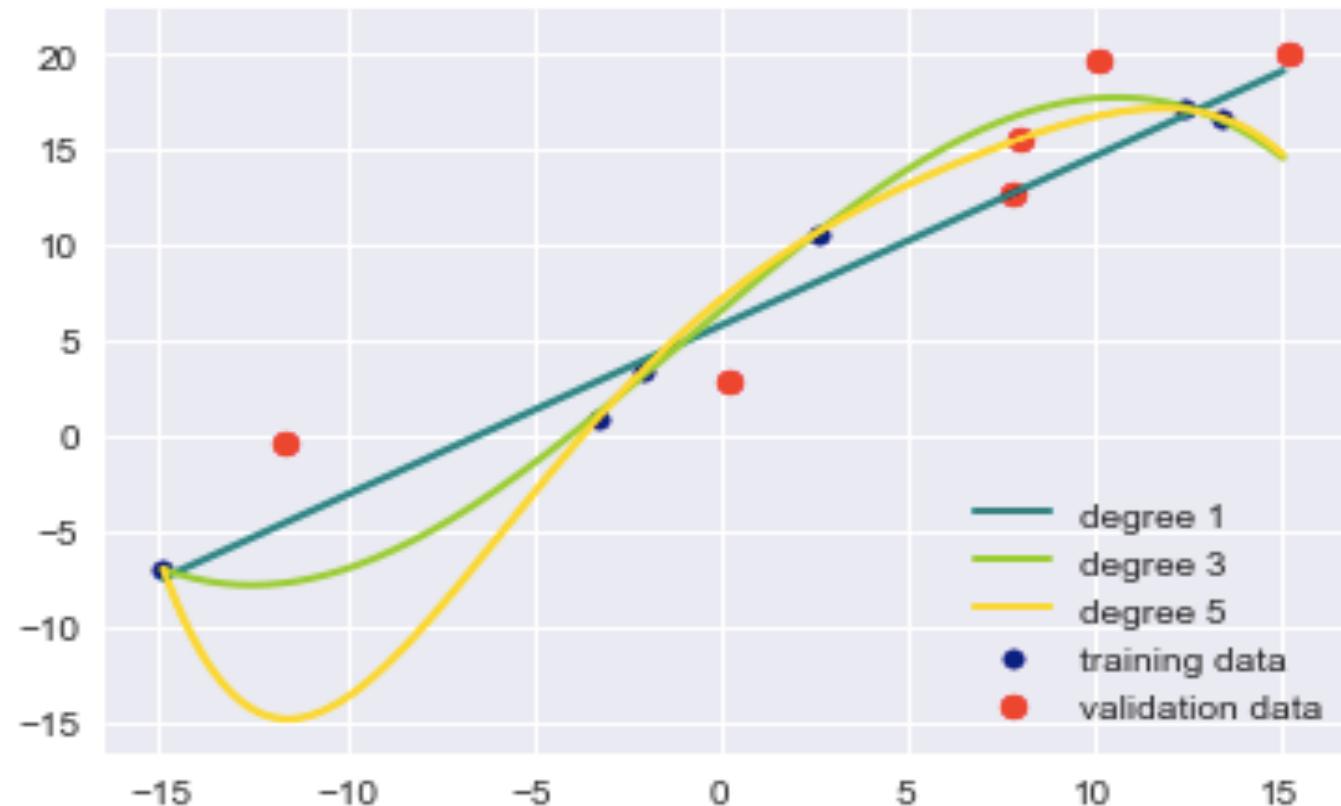
# Labeled Data

	Year	m	d	Time	Prec	snow	tmp	maxtemp	mintmp
0	2016	1	2	00:00	-1.0	-1.0	-7.0	-5.5	-7.8
1	2016	1	3	00:00	3.2	-1.0	-8.7	-7.2	-10.0
2	2016	1	4	00:00	-1.0	4.0	-11.2	-4.9	-13.5
3	2016	1	5	00:00	0.6	4.0	-17.6	-13.3	-19.6
4	2016	1	6	00:00	-1.0	4.0	-20.3	-16.4	-21.3

training

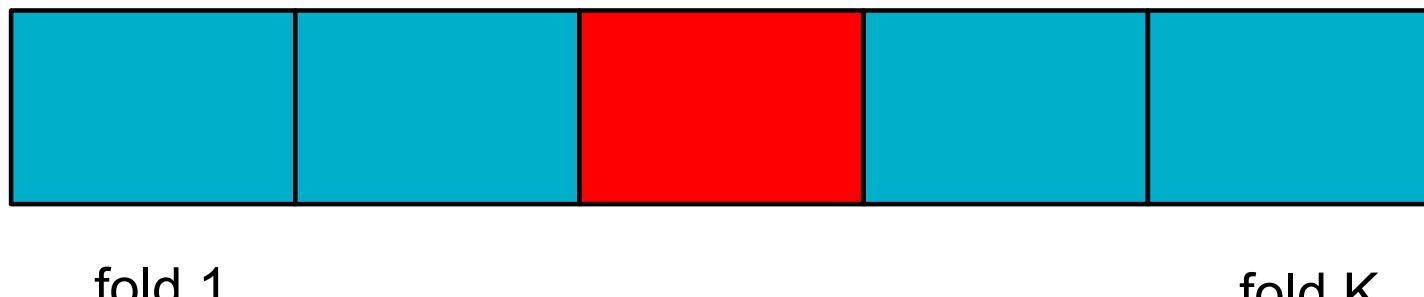
validation

# Training and Validation Error



# K-Fold Cross Validation

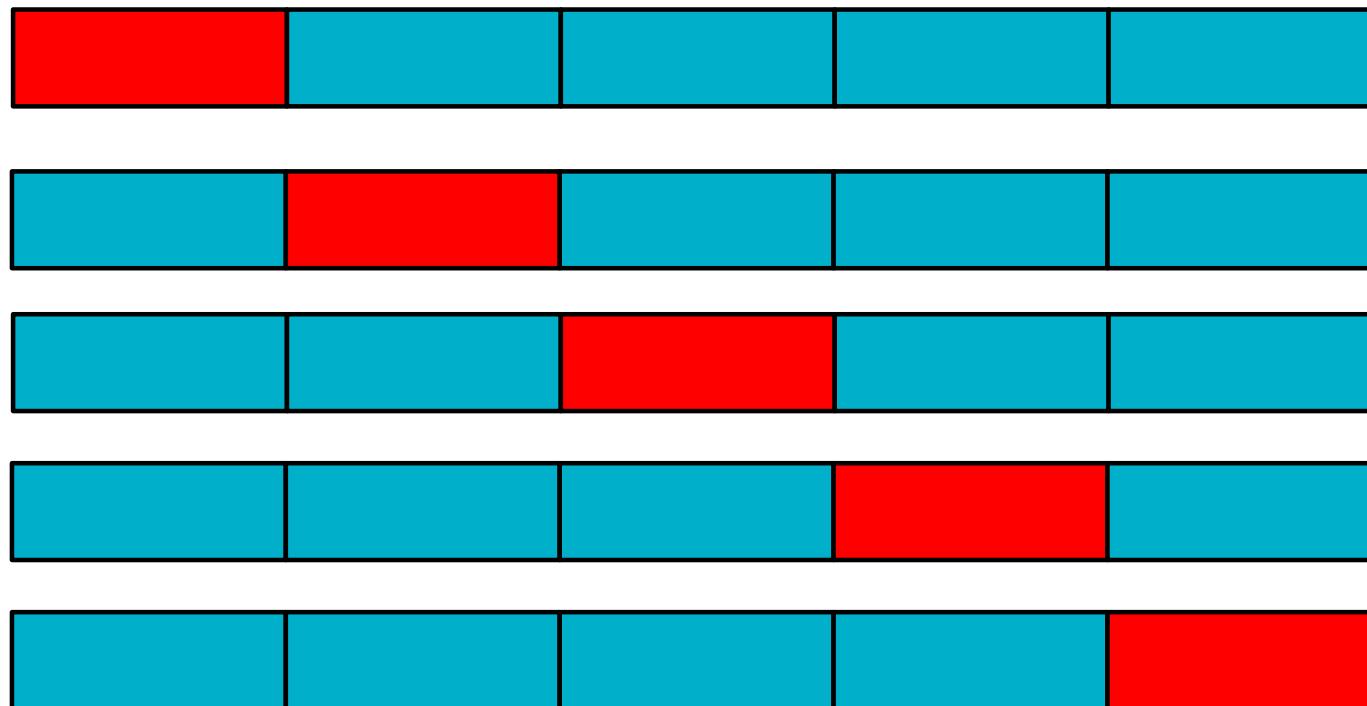
n labeled data points (assume n is integer multiple of K)



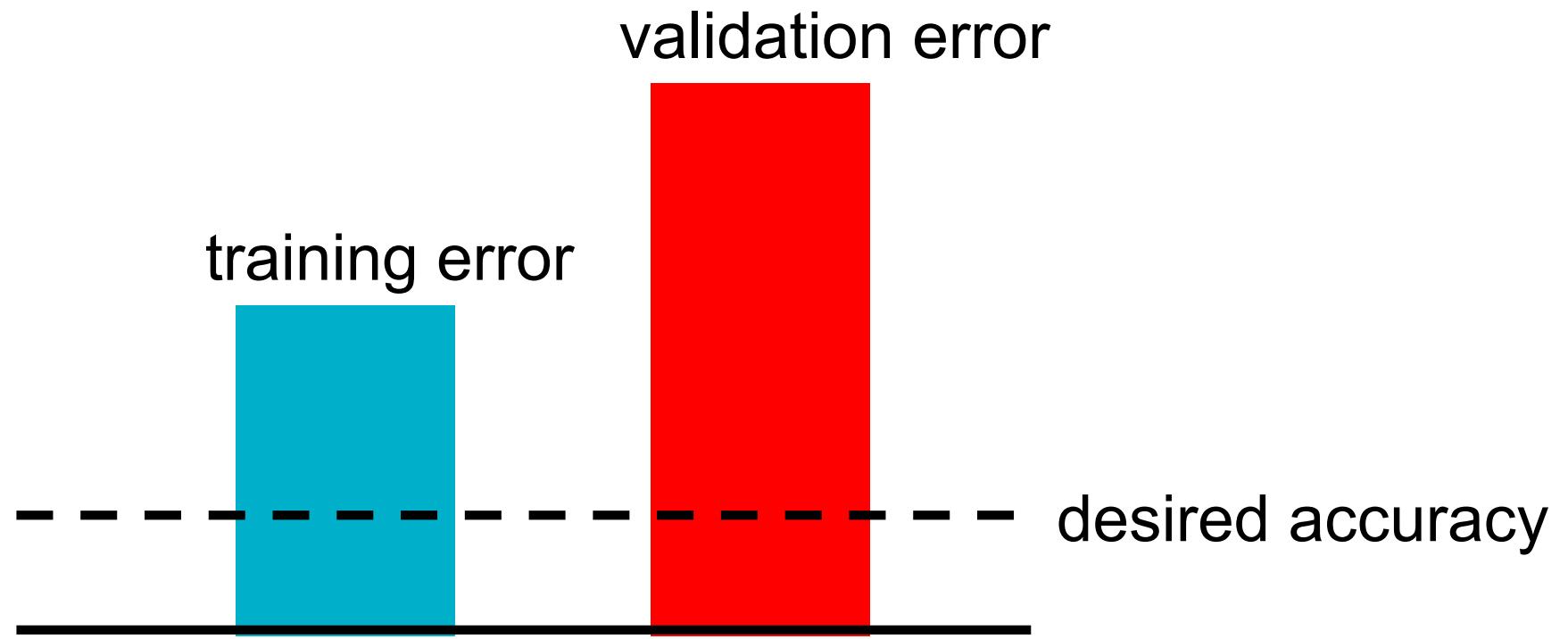
K-fold CV involves K iterations: in each iteration choose one fold as **validation set**; rest as **training set**

average the K validation errors

# K-Fold CV with K=5



# Diagnosing ML Methods



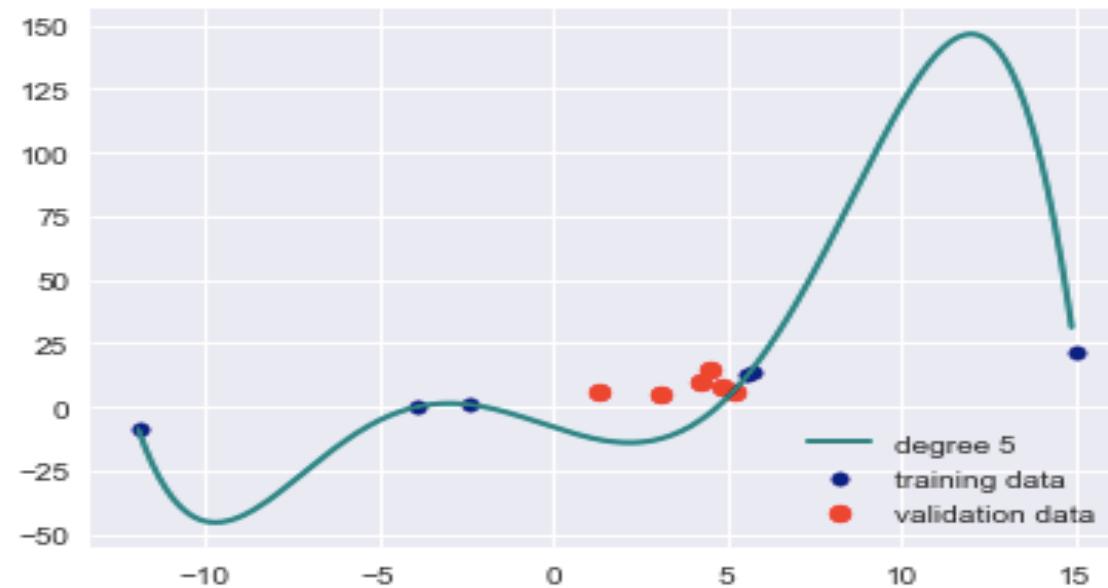
# Diagnosis

- training error larger than desired accuracy:
  - use larger model (larger ANN, higher degree polyn...)
  - use more features
- training error at desired level, but validation error too large:
  - use smaller model (smaller ANN, smaller poly degree.)
  - use larger training set (get more data)
  - use **regularization** techniques
- training error  $\approx$  validation error  $\approx$  desired accuracy: DONE!

# Regularization

consider polynomial predictor with max. degree 5

$$h(x) = w_0 + w_1 * x + w_2 * x^2 + \dots + w_5 * x^5$$



# The Basic Idea of Regularization

- choose weights  $w_0, \dots, w_5$  **not only** by minimizing training error
- choose weights by min. sum of training error and regularization term
- regularization term represents anticipated increase of error on new data
- one widely used choice for reg. term is sum of squares of weights
- choose weights by minimizing

$$\frac{1}{m} \sum_{i=1}^m L(y^{(i)}, h(x^{(i)})) + \rho(w_0^2 + \dots + w_5^2)$$

- tuning parameter  $\rho$  (large  $\rho$  anticipates large increase of error)

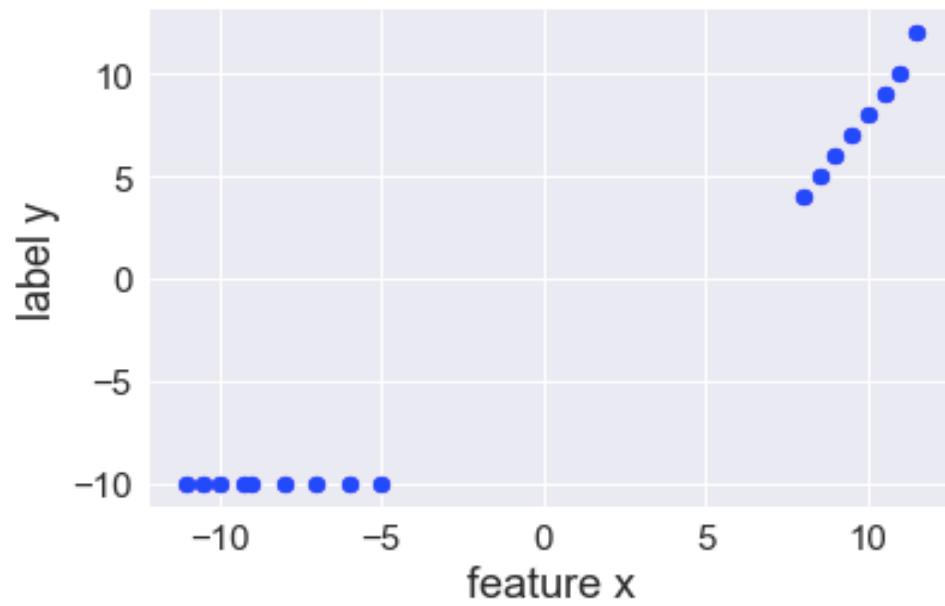
# Lets get Hands Dirty!

- get some data and define features and labels
- choose ML model: linear, polynomial, ANN, decision tree
- compute training and validation error
- compare training, validation error with desired accuracy)
- change model (enlarge/reduce model) and/or get more data
- repeat

# Clustering

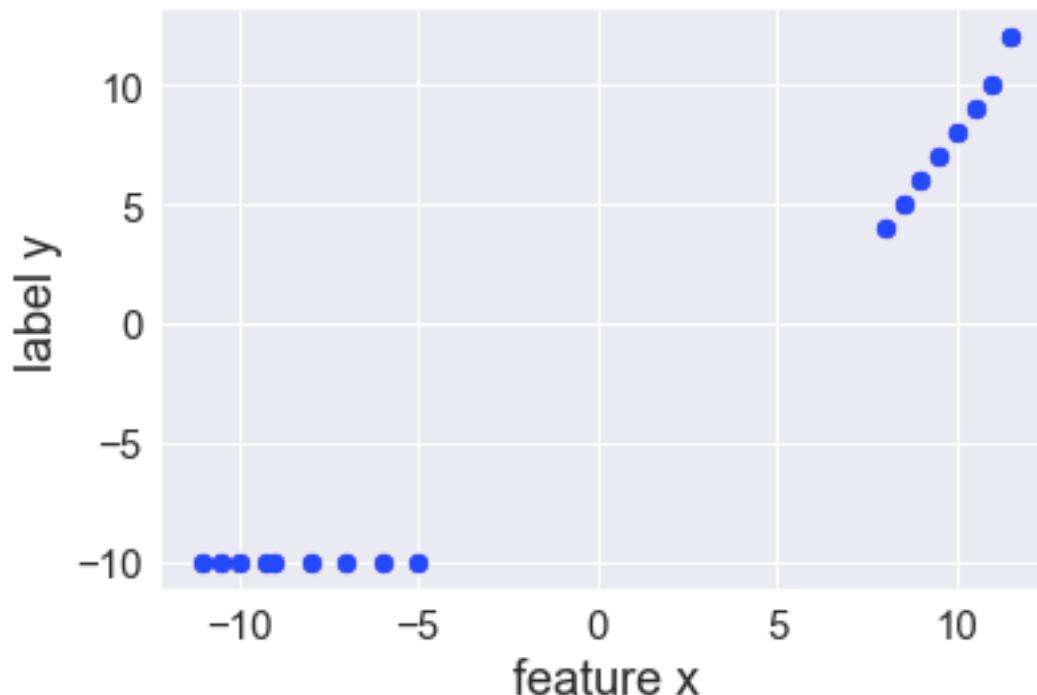
# A Linear Prediction Problem

- consider data points  $(x,y)$  with features  $x$  and label  $y$
- assume you want to use linear predictors to predict  $y$  from  $x$



100

# Clustering = Grouping Similar Data Points

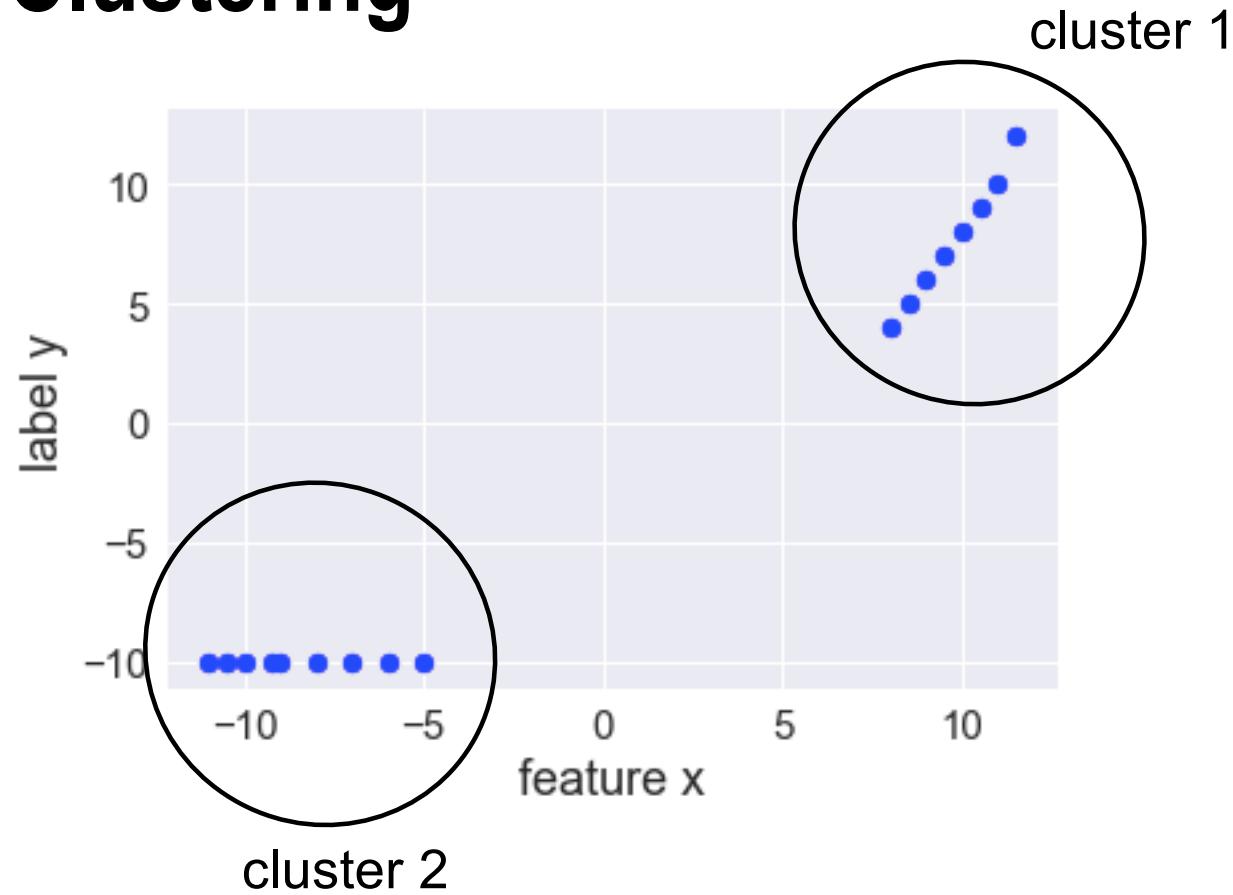


how many cluster (groups)  
do you see?

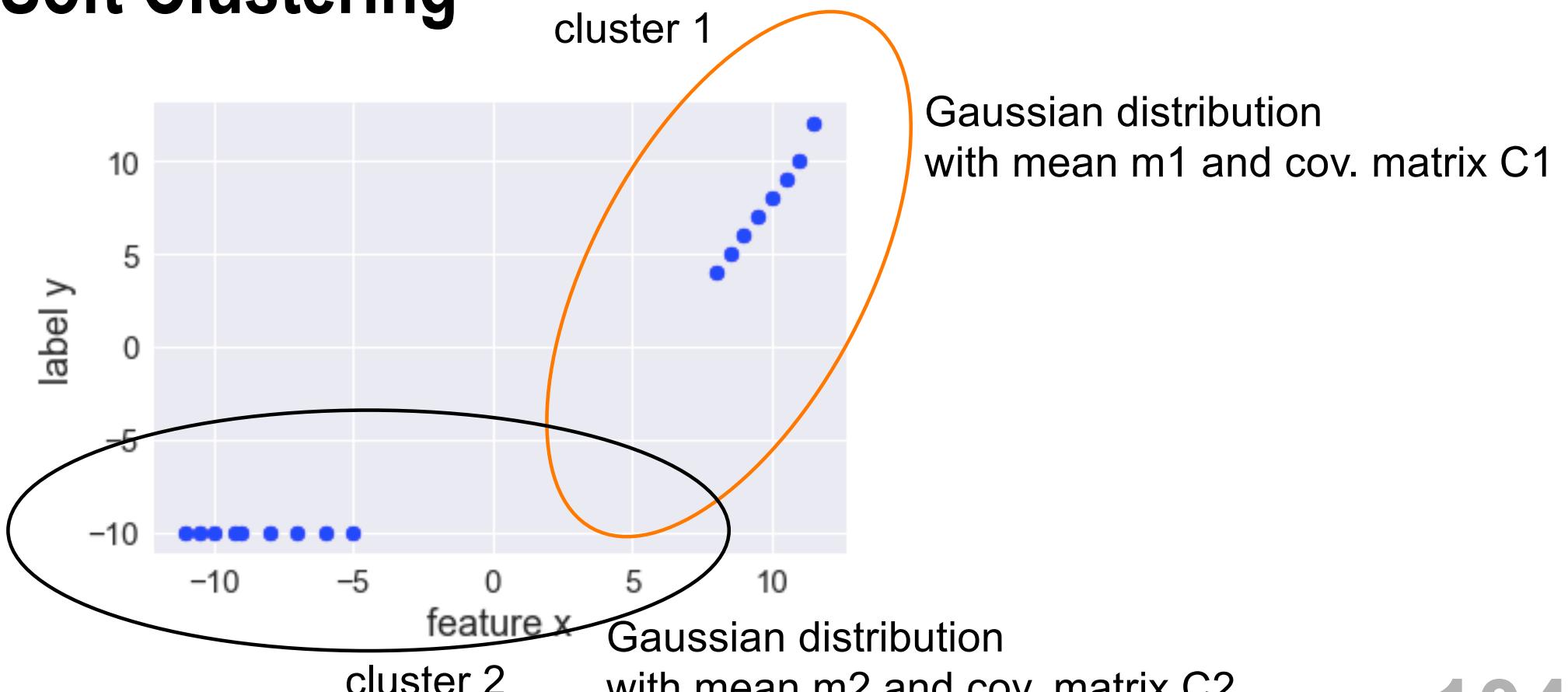
# Two Main Flavors of Clustering

- hard clustering:
  - each data point belongs to exactly one cluster
  - popular method: **k-means**
- soft clustering:
  - data points belong to many clusters
  - data points have degree of belonging to particular cluster
  - popular method: **Gaussian mixture models**

# Hard Clustering



# Soft Clustering



# Feature Learning

105

# Long Vectors

- consider data point represented as vector  $\mathbf{z}$  of length  $D$
- what is  $D$  for a 1000 by 1000 greyscale image
- what is  $D$  for a 5 sec recording of a song ?
- what is  $D$  for a dictionary containing all English words?
- what is  $D$  for a dictionary of English 5-grams ?

# What is Bad About Long Feature Vectors?

- computing with long vectors is time consuming
- risk of overfitting
- consider data points represented by vectors of length  $D$
- linear predictors overfit training data set smaller than  $D$

# Basic Idea

raw data vector  $z$

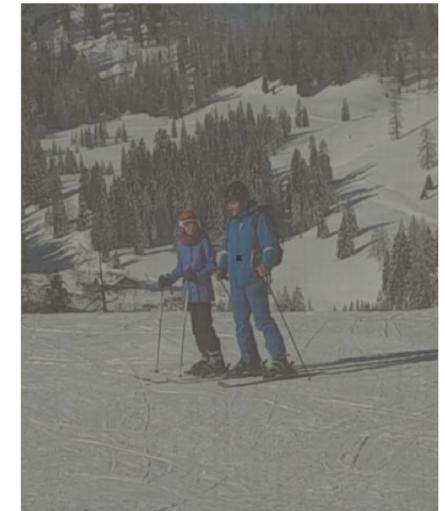


feature learning  
method  $h(z)$



reconstruct

reconstruction  $z'$



map  $h(z)$  involves tunable weights  $w$

choose weights  $w$  to ensure small reconstruction error  $z-z'$

# The Creation (Finnish)



raw data = 254820 pixels



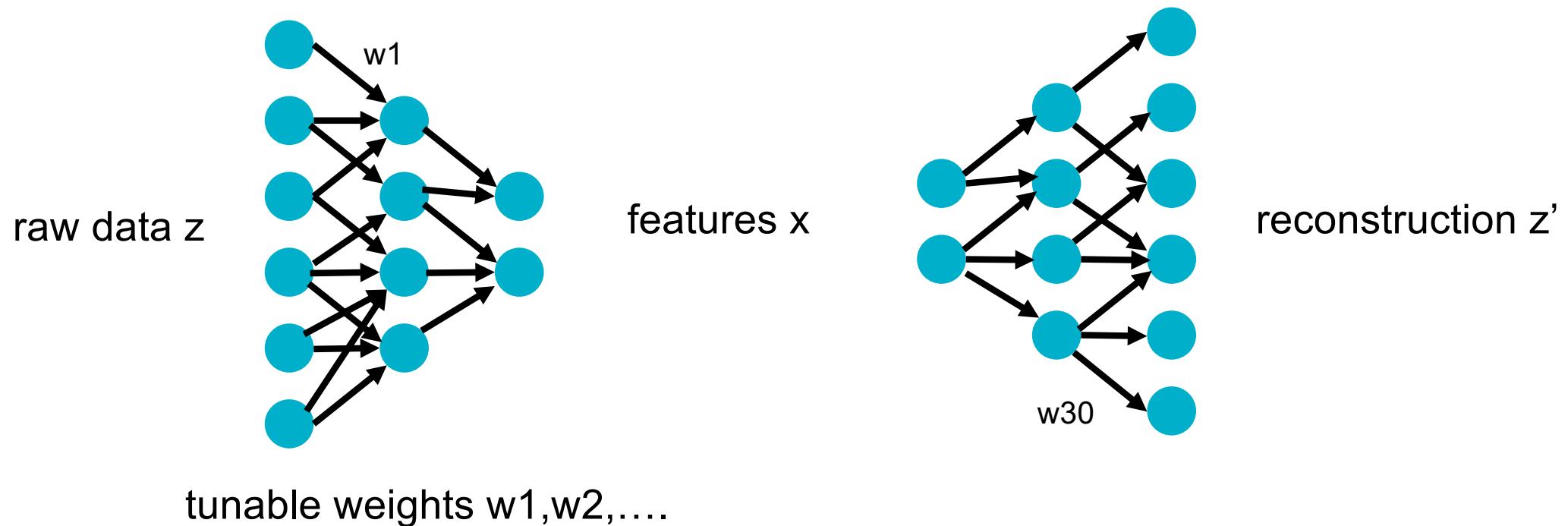
reconstruction obtained from  
17837 features

# Linear Feature Learning



what is best choice for matrix  $W$  such that  $z'$  close to  $z$  ?

# Non-linear Feature Learning – “Autoencoder”



# Transfer Learning

# Basic Idea

- train ML method on different data
- massive amounts of unlabeled data to pre-train ANN
- small amount of high quality labeled data to fine-tune ANN
- allows to compress raw data into pre-trained models

# **Model Based ML Methods**

# Model Based Machine Learning

- consider data points with features  $x$  and labels  $y$
- interpret them as realizations of random variables
- sequence of data points is a random process
- data characterized by probability distribution

# Condition on Data

- specify joint distribution  $p(\dots)$  of all quantities in the model
- quantities include: features, labels of data points and parameters
- compute posterior distribution of variables given observed data
- <https://docs.pymc.io/notebooks/GLM-linear.html>

# Probabilistic Machine Learning

	Year	m	d	Time	Prec	snow	tmp	maxtemp	mintmp
0	2016	1	2	00:00	-1.0	-1.0	-7.0	-5.5	-7.8
1	2016	1	3	00:00	3.2	-1.0	-8.7	-7.2	-10.0
2	2016	1	4	00:00	-1.0	4.0	-11.2	NaN	-13.5
3	2016	1	5	00:00	0.6	4.0	-17.6	-13.3	-19.6
4	2016	1	6	00:00	-1.0	4.0	-20.3	-16.4	-21.3

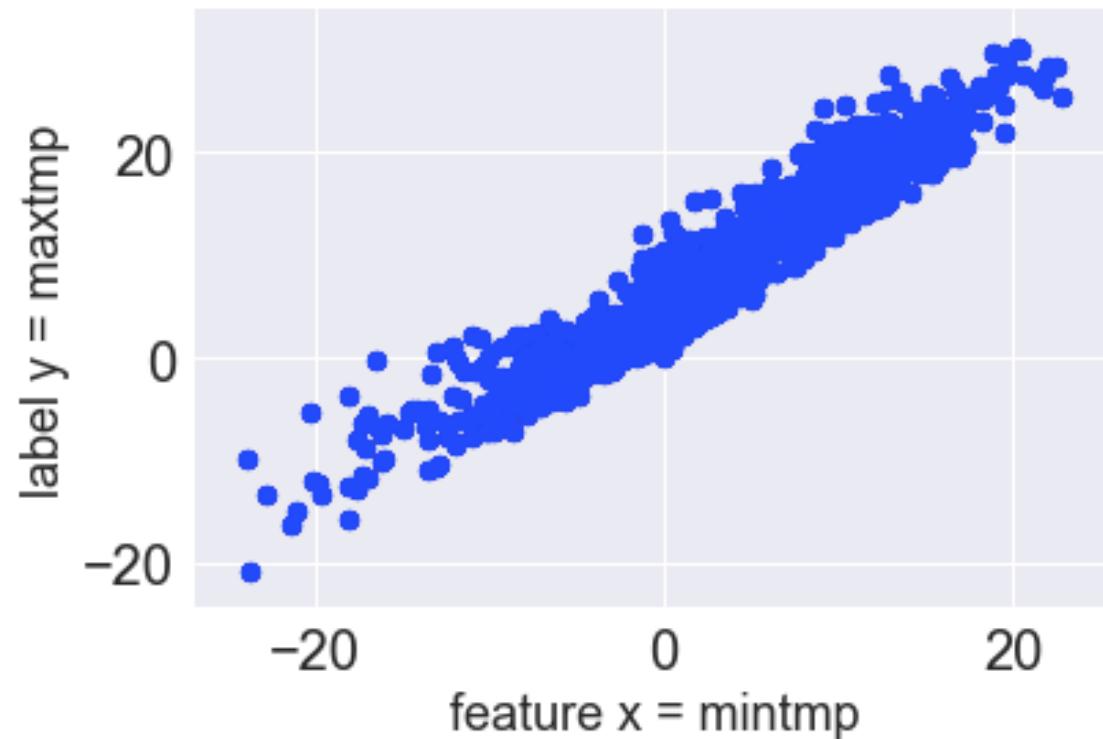
today at 07:00 am: mintmp=-10 maxtmp =?

compute posterior probability distribution  $p(\text{hidden variables} | \text{data})$

# Two Challenges

- Challenge1: how to get joint probability distribution  $p(\text{ data}, \text{ hidden variables})$  ?
- Challenge2: how to compute posterior distribution  $p(\text{ hidden variables} | \text{ data})$  for extremely high-dim. distribution?

# Learning Probability Distribution



what could be a good  
choice for  $p(x,y)$  ?

# Ready Made Tools

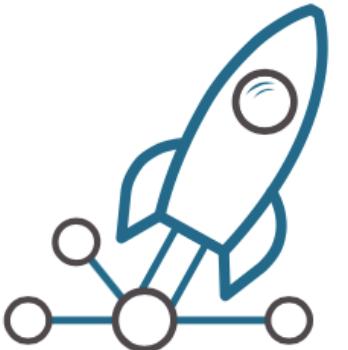
- efficient methods for learning prob. distributions available
- efficient methods for computing posterior  $p(\text{ hidden variables} \mid \text{ data})$  available

---

 PyMC3

Tutorials Examples Books + Videos API Developer Guide About PyMC3

Search... 



# PyMC3

Probabilistic Programming in Python

[Quickstart →](#)

# Reinforcement Learning

# Machine Learning vs. Reinf. Learning

- "ordinary" machine learning methods consider data given
- methods are passive; read in data → output prediction
- reinforcement learning takes effect of predictions into account
- predictions incur decisions/actions
- decisions/actions influence the data which is observed next

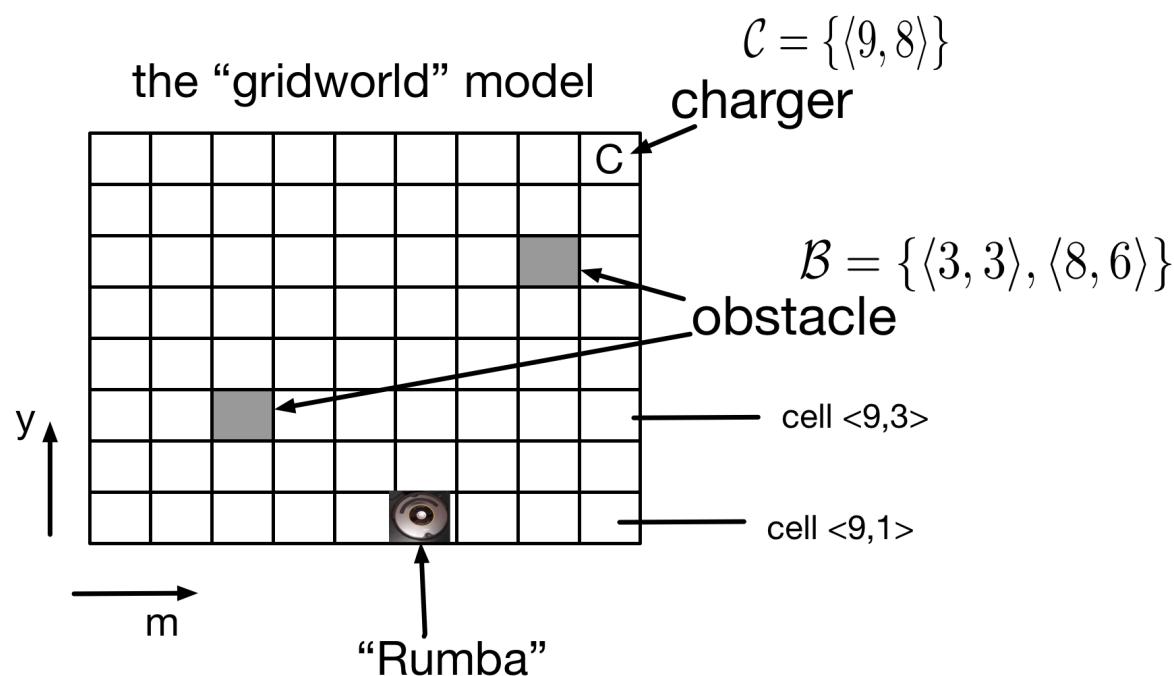
# Cleaning Robot



123

# Office Room

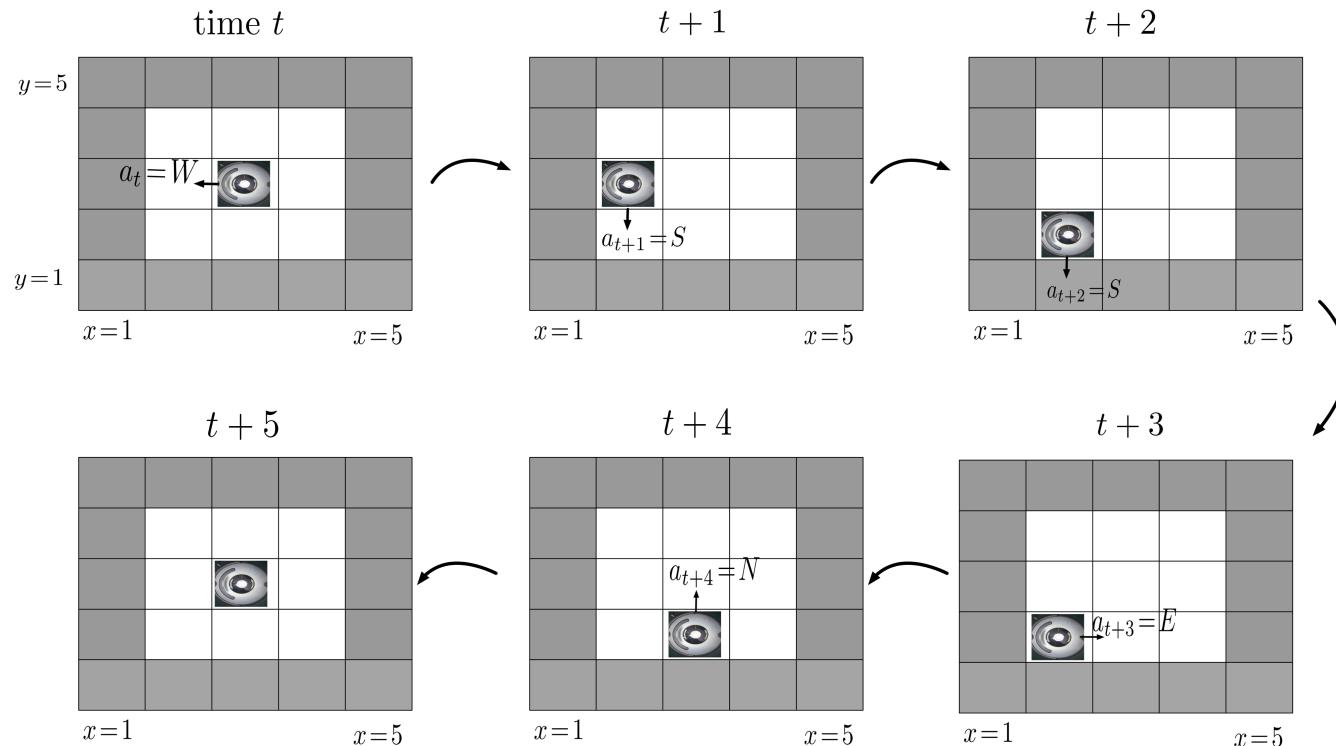
office room “B329”



# Actions

action  $y_t$  at time  $t$  is direction to move next

obstacle



# Reinforcement Learning Problem

given snapshot of onboard camera,  
what is best direction to move next



features  $x$  = pixels of snapshot  
label  $y$  = optimal direction to move next

We do not know the true label here!  
We can only observe the effect of choosing  
some direction!

# Markov Decision Process

- interaction between AI (cleaning robot) and environment (office room)
- at time  $t$ , observe snapshot  $x_t$
- based on snapshot decide for next direction  $y_t$
- measure amount  $r_t$  of dust collected at time  $t$
- sequence  $x_1, y_1, r_1, x_2, y_2, r_2$  modelled as a Markov decision process
- MDP is a special type of random process

# Optimal Policy

- AI cleaning robot implements a policy  $h(x)$
- policy takes input  $x$  (snapshot) and outputs  $y$  (direction of next move)
- policy map  $h(x)$  depends on tunable weights (like predictors !)
- RL amounts to tuning  $h(x)$  such that reward is maximized
- AI engineer defines what the “reward” is (e.g., amount of dust collected)

**THAT'S IT FOLKS!**