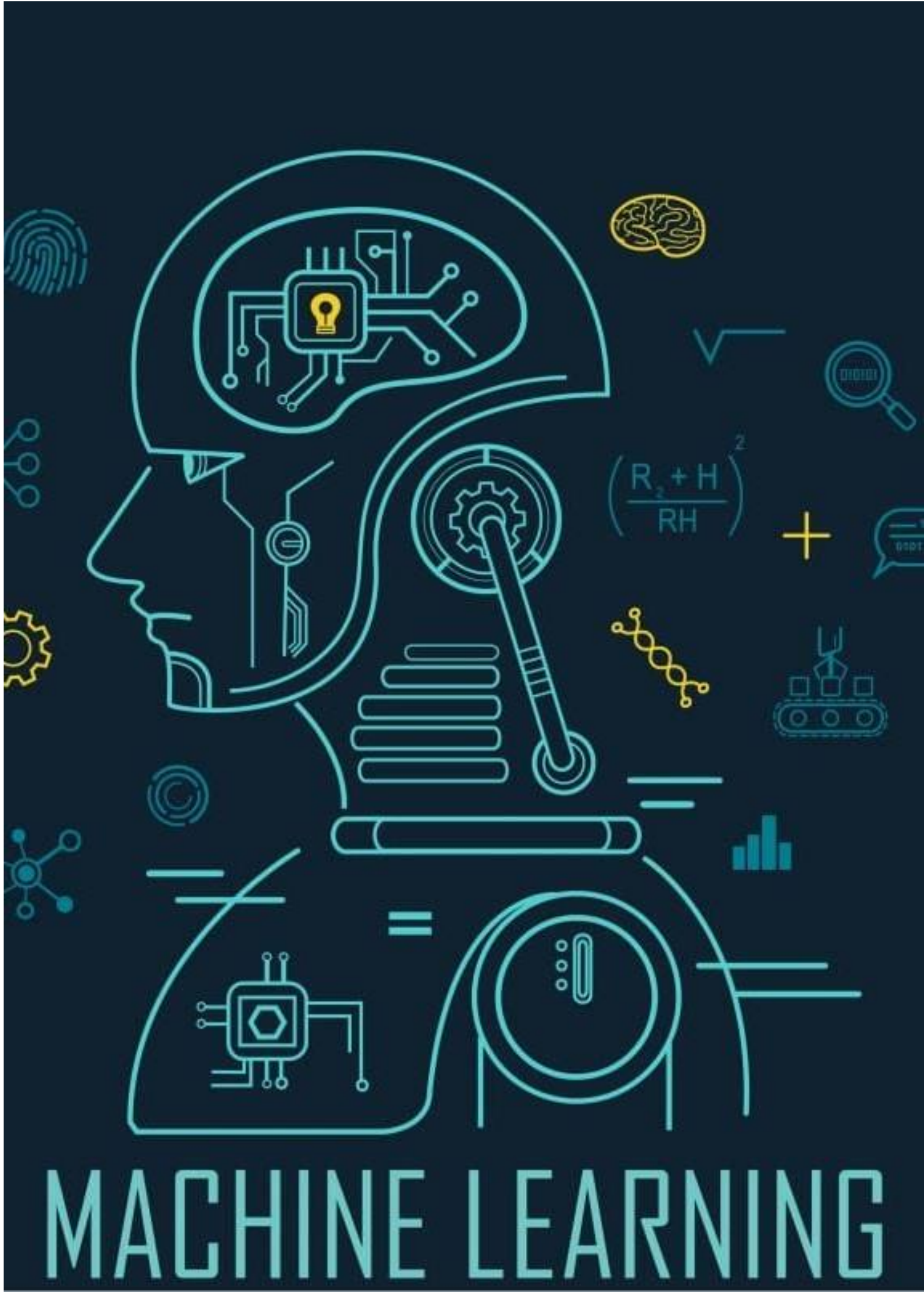
CSCI 5622-001 Machine Learning
Welcome!

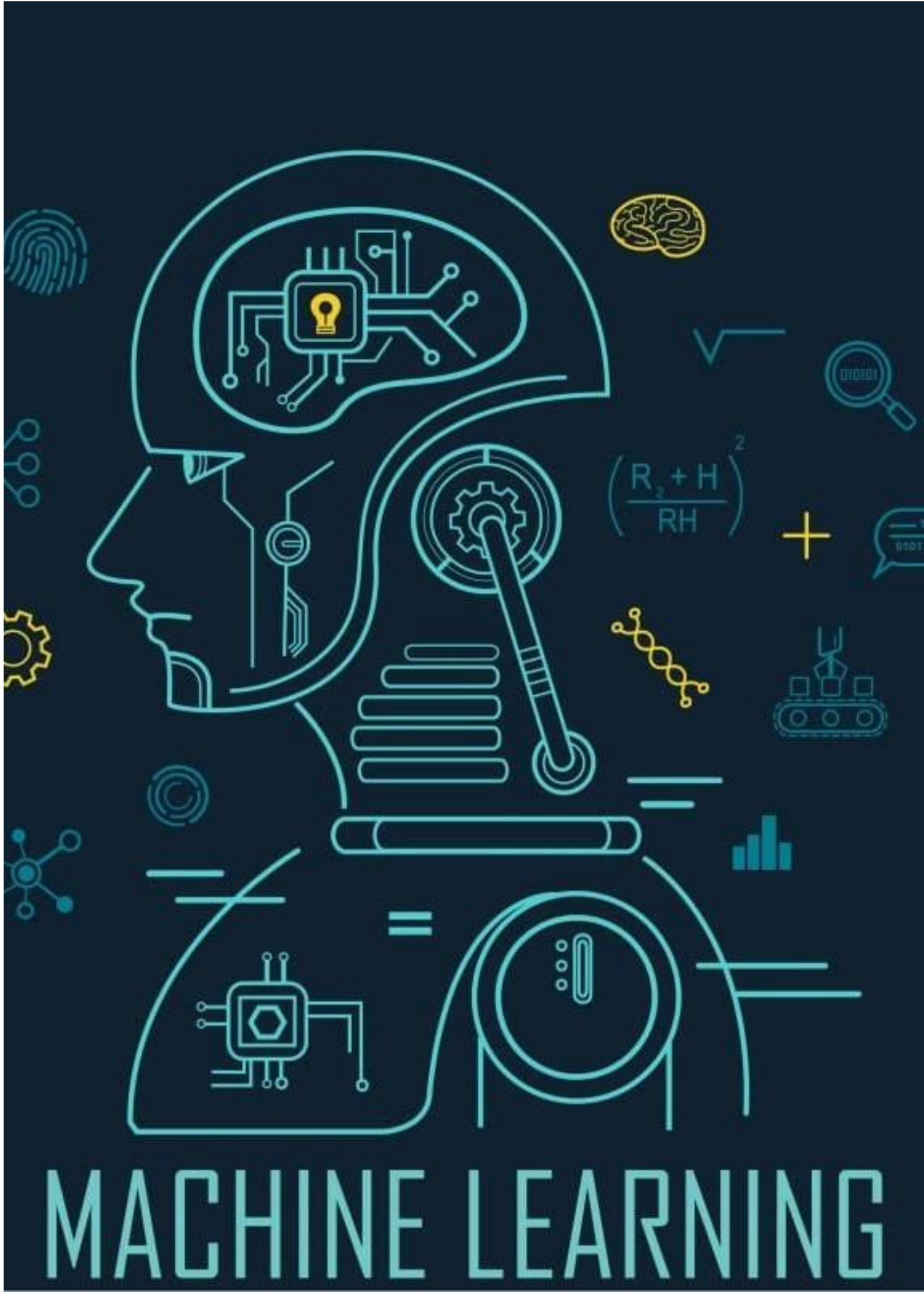
Welcome!

- About this class
- Introduction to machine learning (definitions, basic concepts, challenges)



Welcome!

- About this class
- Introduction to machine learning (definitions, basic concepts, challenges)



Welcome to CSCE 5622!

- Instructor
  - Esther Rolf
  - Email: esther.rolf@colorado.edu
  - Office Hours: Monday, 11am-12pm at ECES 122; and by appointment

- Teaching Assistants
  - Jen MacDonald (jen.macdonald@colorado.edu)
  - Office hours: Wednesdays 5:30-6:30pm and Thursday 10:30am-11:30am ECOT 832
  - Julia Romero (Julia.romero@colorado.edu)
  - Office hours: TBD

- Course manager
  - Sharath Soundarrajan Vanisri
  - Email: sharath.soundarrajanvanisri@colorado.edu

Class websites

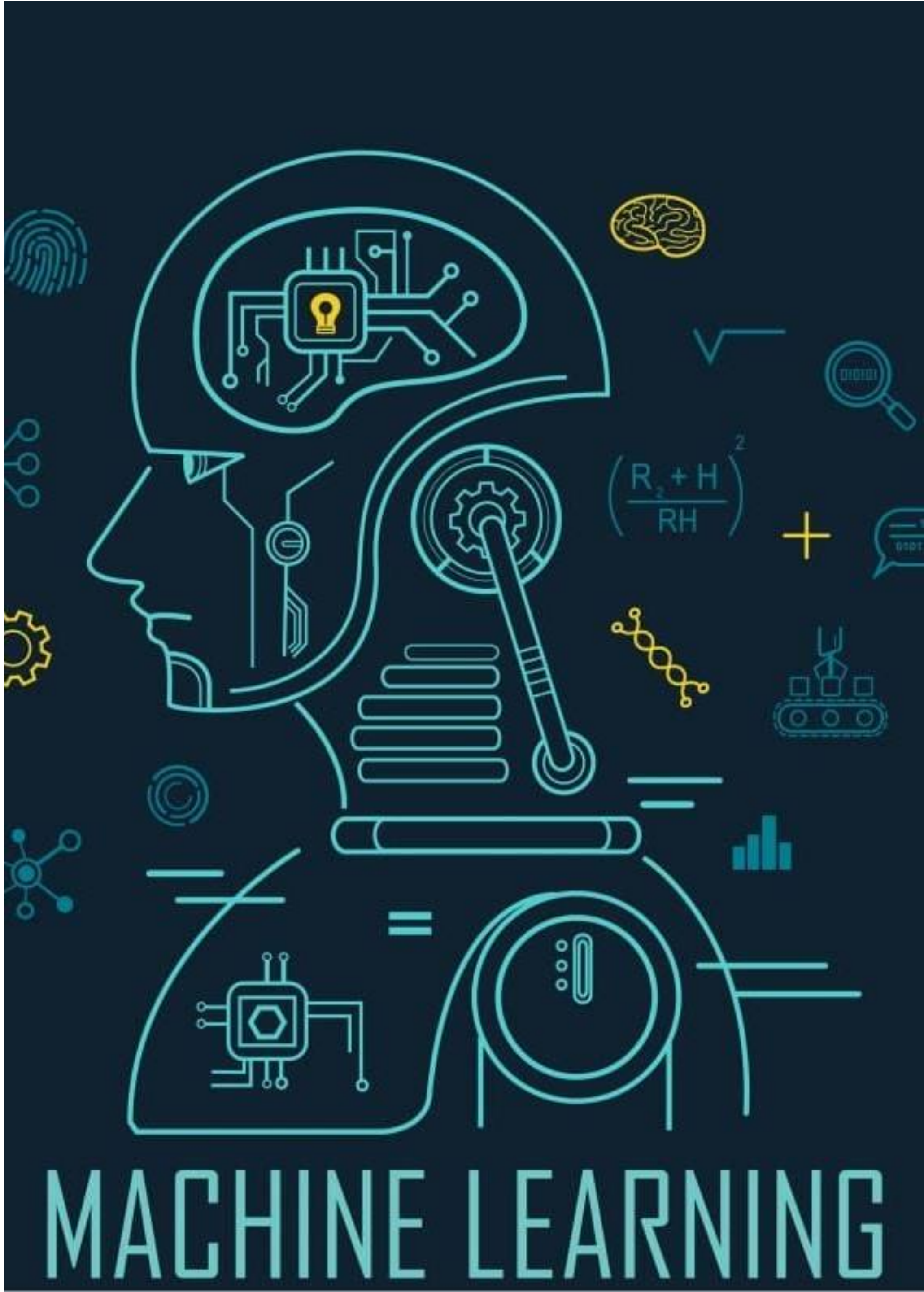
- **CANVAS**
  - Class logistics/announcements
  - Slides
  - Homework posting, solutions, submissions
  - For sending private messages to me, the TA, and the course manager
  - Class recordings
- **Piazza (via CANVAS)**
  - Class discussions
  - You can post your questions anonymously!
- **Class roadmap (to be updated throughout the semester)**
  - https://docs.google.com/spreadsheets/d/1F7pSKzxpn1zziVyjTOxsJoMxA2mzTTC44TDaMSNlaLY/edit?usp=sharing

Textbook and course material

- Lecture notes and supplemental material (on CANVAS)
- Textbooks
  - Introduction to Machine Learning (4th Edition), Ethem Alpaydin, https://mitpress.mit.edu/9780262043793/introduction-to-machine-learning/
  - Learning from Data, Yaser S. Abu-Mostafa, Malik Magdon-Ismail, and Hsuan-Tien Lin, 2012, https://amlbook.com/
  - The Elements of Statistical Learning (2nd Edition), Trevor Hastie, Robert Tibshirani, Jerome Friedman, Springer Series in Statistics, https://hastie.su.domains/ElemStatLearn/printings/ESLII_print12_toc.pdf

Learning outcomes

- Obtain a good understanding of the core issues and challenges in machine learning, encompassing aspects such as data handling, model selection, model complexity.
- Develop insight into the advantages and limitations of popular machine learning methodologies.
- Explore inherent mathematical relationships within supervised and unsupervised algorithms.
- Design and implement various machine learning algorithms in a range of real-world applications.
- Explore ethical implications of deploying machine learning algorithms in real-life.



Class structure

- 5 homework assignments (40 points)
  - Late submissions accepted with a 1-week grace period after the deadline and 1 point penalty (1 out of 8 points will be deducted)
- 6 Quizzes (20 points)
  - Each quiz carries 4 points. Grades are based on top 5 quizzes
  - Unfortunately, there are no opportunities for quiz make-up
- 2 exams (40 points)
  - Exam 1: March 3 (during class time)
  - Exam 2: April 30th (during class time)

Total: 100 points

Homework Submission

- All homeworks will be submitted as a **single pdf** on CANVAS
  - The executable code (when required) needs to be included at the end of the pdf
- Programming assignments
  - Recommended language is Python
- Math assignments
  - Please submit solution produced in Latex
  - Or **very clear** handwritten solution
  - This will help with grading a lot. **Solutions that are not clearly written will not be graded.**

Active Learning

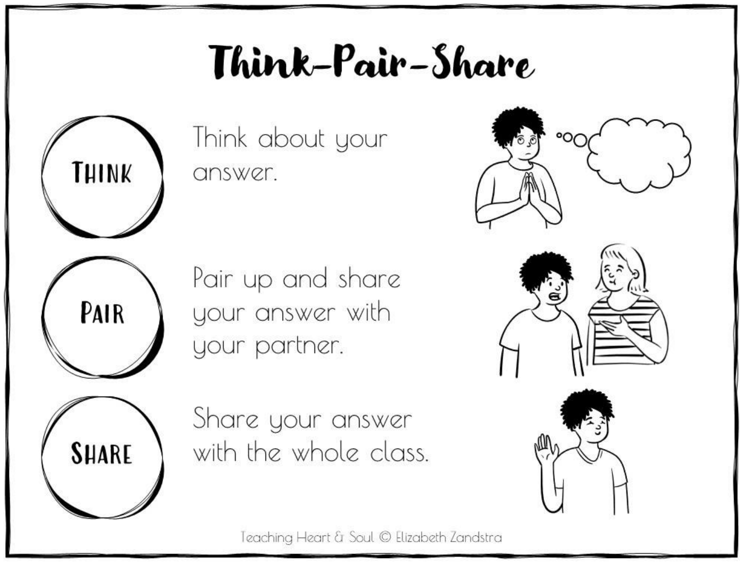
- Would you ever take a cardio class without actually participating in it?
- So why take a CS course without practicing the material in class?

Active Learning

- "Anything that involves students in doing things and thinking about the things they are doing" (Bonwell & Eison, 1991)
- "Anything course-related that all students in a class session are called upon to do <u>other than simply watching, listening and taking notes</u>" (Felder & Brent, 2009)
- Audience attention starts to wane after 10-20 mins
- Research suggests that incorporating active learning techniques
  - encourages student engagement
  - reinforces important material, concepts, etc.
  - builds self-esteem
  - creates a sense of community

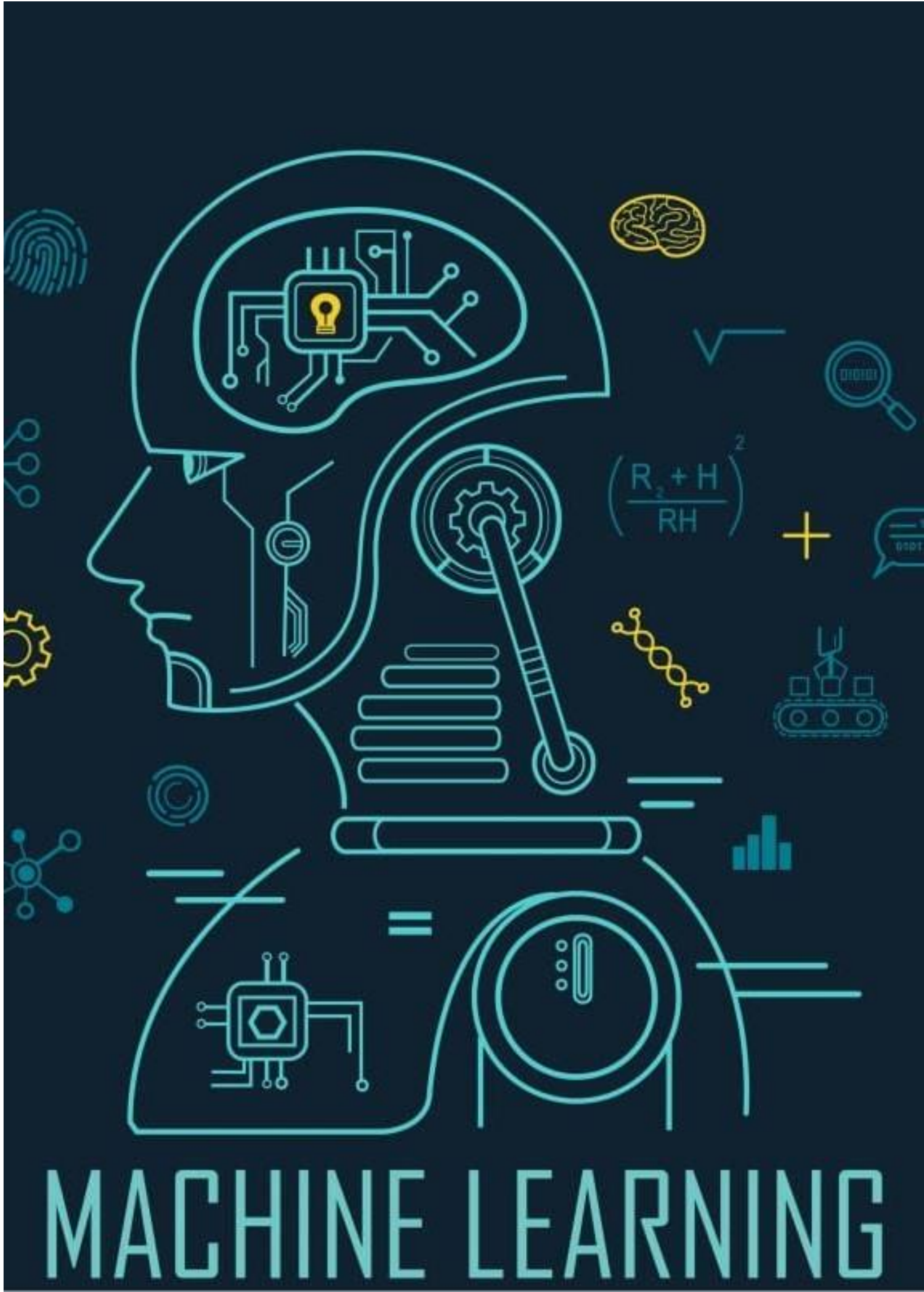
Active Learning

- In-class multiple choice questions
- In-class problem solving and practice questions
- In-class coding demos
- Class discussions

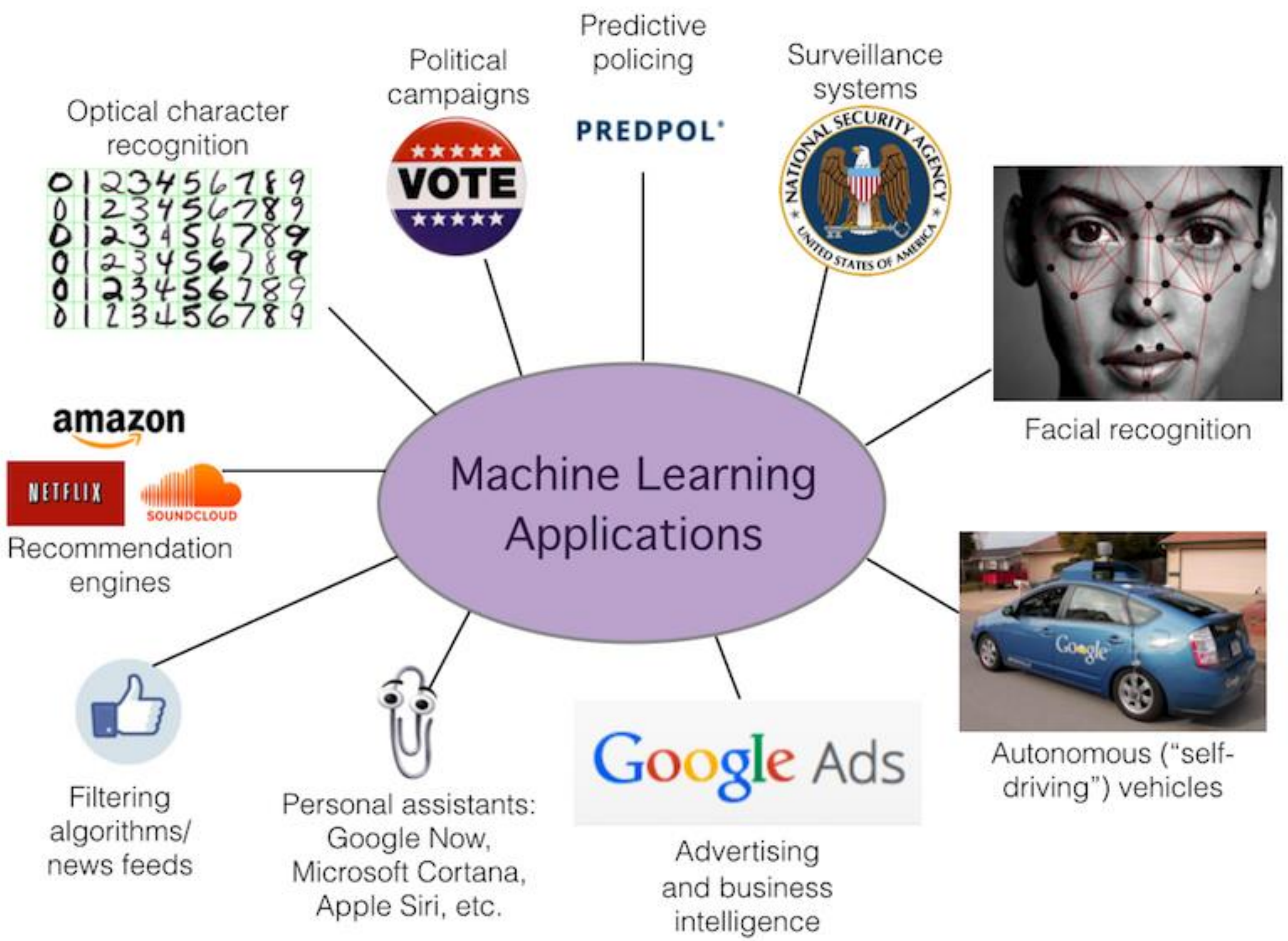


Welcome!

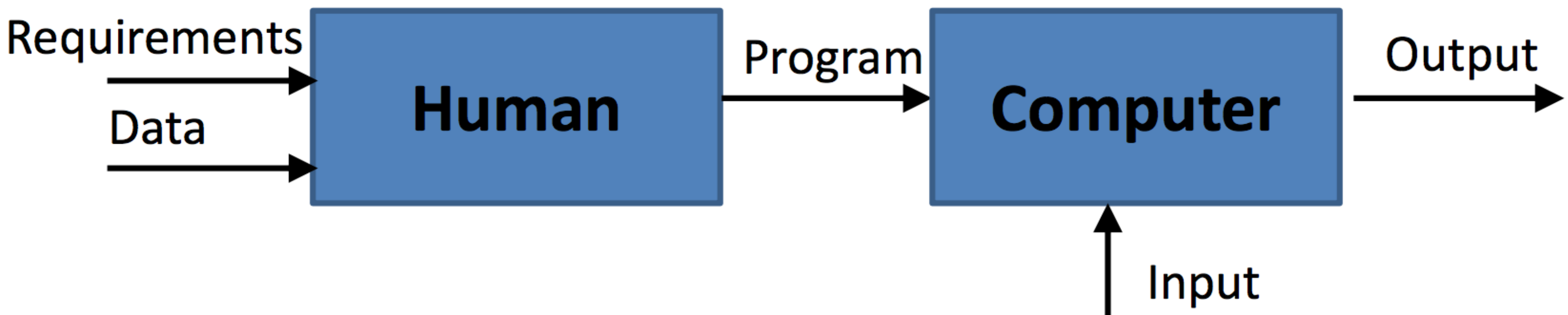
- About this class
- Introduction to machine learning (definitions, basic concepts, challenges)



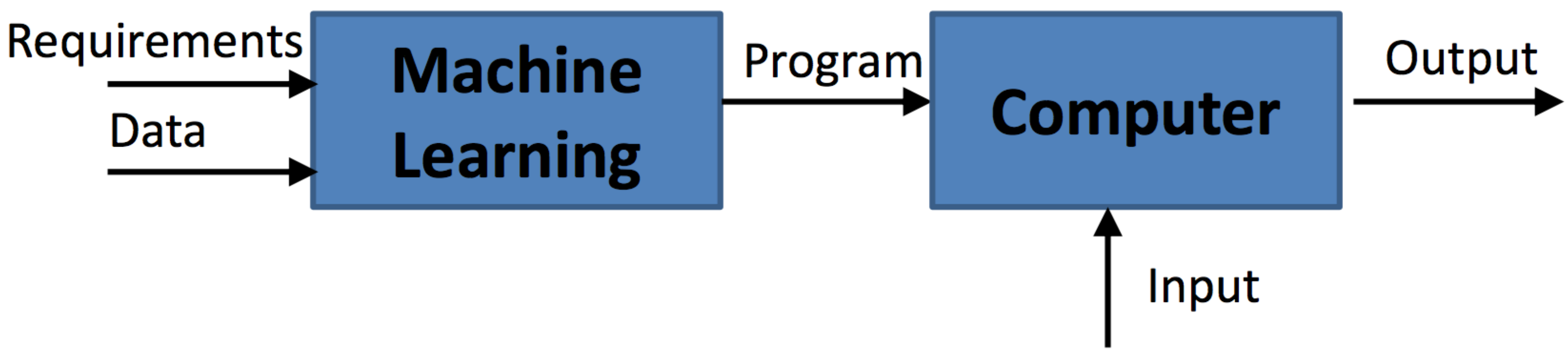
Machine learning is everywhere

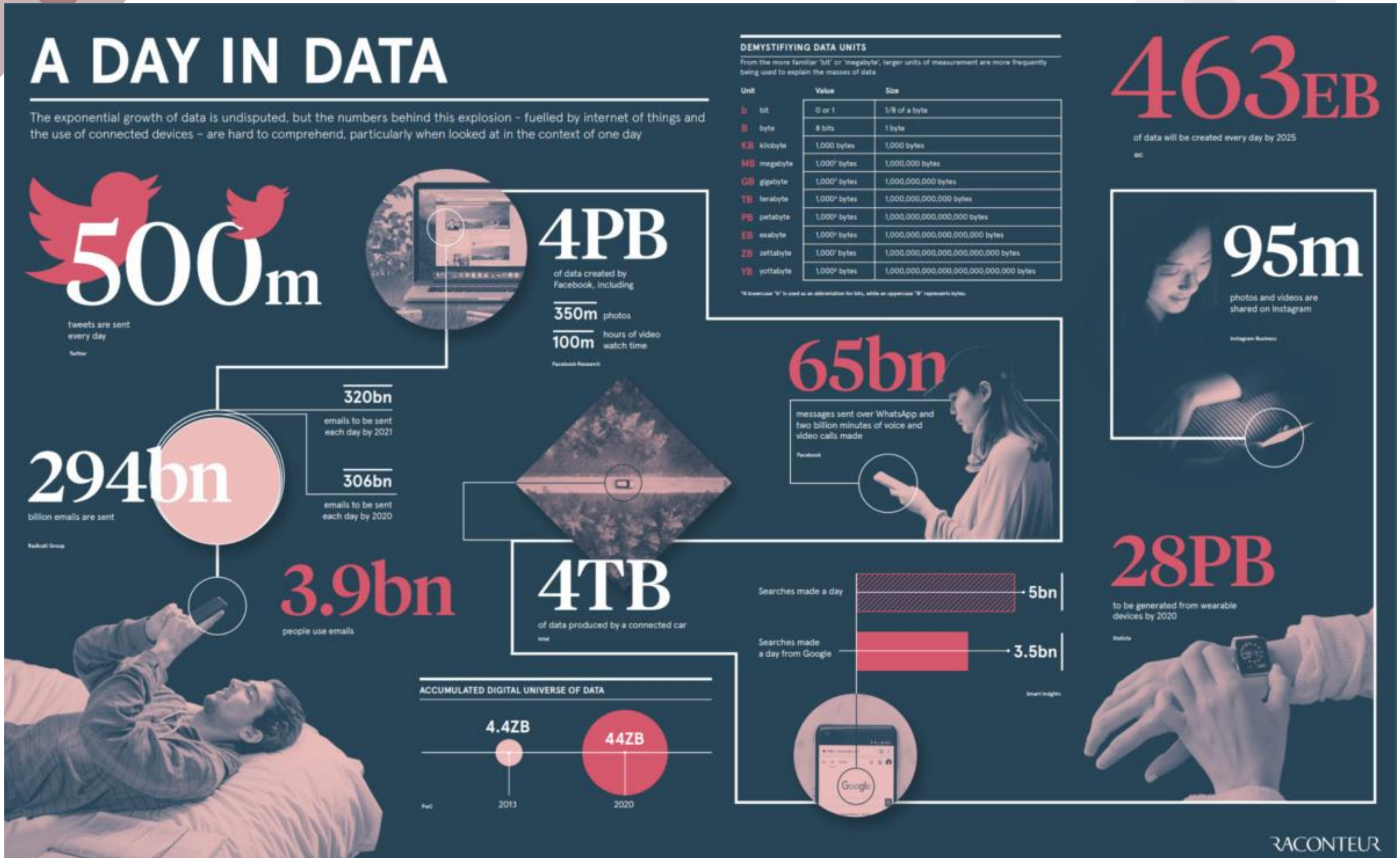


What is machine learning?



Requirements and data change often





A DAY IN DATA
The exponential growth of data is undisputed, but the numbers behind this explosion - fuelled by internet of things and the use of connected devices - are hard to comprehend, particularly when looked at in the context of one day
500m
tweets are sent every day
Twitter
4PB
of data created by Facebook, including
350m photos
100m hours of video watch time
Facebook Research
320bn
emails to be sent each day by 2021
294bn
billion emails are sent
306bn
emails to be sent each day by 2020
Radicati Group
3.9bn
people use emails
4TB
of data produced by a connected car
Intel
ACCUMULATED DIGITAL UNIVERSE OF DATA
4.4ZB
44ZB
2013
2020
PwC
DEMYSTIFIYING DATA UNITS
From the more familiar 'bit' or 'megabyte', larger units of measurement are more frequently being used to explain the masses of data
Unit | Value | Size
b bit | 0 or 1 | 1/8 of a byte
B byte | 8 bits | 1 byte
KB kilobyte | 1,000 bytes | 1,000 bytes
MB megabyte | $1{,}000^2$ bytes | 1,000,000 bytes
GB gigabyte | $1{,}000^3$ bytes | 1,000,000,000 bytes
TB terabyte | $1{,}000^4$ bytes | 1,000,000,000,000 bytes
PB petabyte | $1{,}000^5$ bytes | 1,000,000,000,000,000 bytes
EB exabyte | $1{,}000^6$ bytes | 1,000,000,000,000,000,000 bytes
ZB zettabyte | $1{,}000^7$ bytes | 1,000,000,000,000,000,000,000 bytes
YB yottabyte | $1{,}000^8$ bytes | 1,000,000,000,000,000,000,000,000 bytes
*A lowercase "b" is used as an abbreviation for bits, while an uppercase "B" represents bytes.
65bn
messages sent over WhatsApp and two billion minutes of voice and video calls made
Facebook
Searches made a day
5bn
Searches made a day from Google
3.5bn
Smart Insights
463EB
of data will be created every day by 2025
IDC
95m
photos and videos are shared on Instagram
Instagram Business
28PB
to be generated from wearable devices by 2020
Statista
RACONTEUR

What is machine learning?

A possible definition[1]
A set of methods that can automatically detect patterns in data, and then use those to predict future data or perform other kinds of decision making under uncertainty.

A more formal definition[2]
A computer program is said to learn from experience E with respect to some task T and some performance measure P, if its performance on T as measured by P improves with experience E

[1] From K.P. Murphy
[2] From T. Mitchell

What is machine learning?

Definition: A computer program learns from experience E with respect to some task T and some performance measure P, if its performance on T as measured by P improves with experience E

Question: Let's consider a medical application where a computer program is designed to diagnose whether a patient is pre-diabetic based on a set of records. What is experience E in this setting?

A. Classifying a patient as pre-diabetic or not pre-diabetic.
B. Learning from a dataset containing medical records of patients.
C. The accuracy of the program in correctly diagnosing patients.
D. All of the above

What is machine learning?

Definition: A computer program learns from experience E with respect to some task T and some performance measure P, if its performance on T as measured by P improves with experience E

Question: Let's consider a medical application where a computer program is designed to diagnose whether a patient is pre-diabetic based on a set of records. What is experience E in this setting?

A. Classifying a patient as pre-diabetic or not pre-diabetic (task T).
B. Learning from a dataset containing medical records of patients (experience E).
C. The accuracy of the program in correctly diagnosing patients (performance P).
D. All of the above

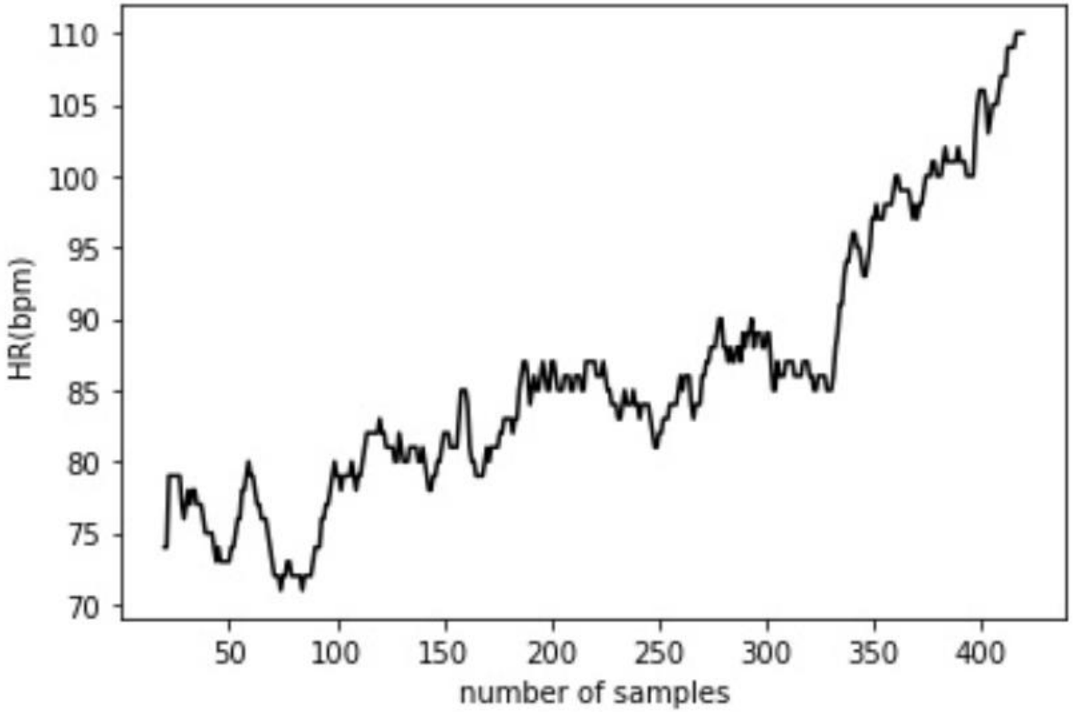
Key ingredients for a machine learning task

- Data
  - Collected from past observations (training data)
- Model
  - Captures/quantifies patterns in data
  - Doesn't have to be absolutely true, as long as it is close enough
- Prediction
  - Apply the model to
    - Forecast what is going to happen in the future
    - Automatically make a decision for unknown data (testing data)

Example: Detecting Patterns

Below is the heart rate of an individual during incremental exercise

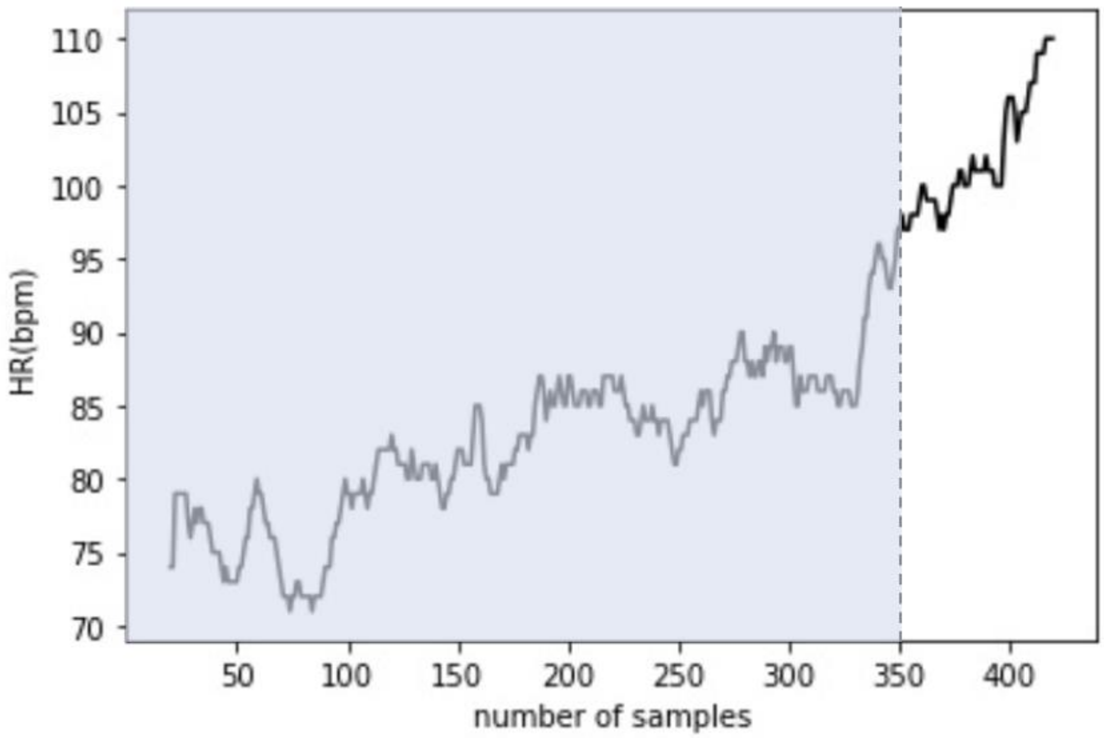
How has the heart rate been changing over the course of the exercise?



- Generally increasing patterns
- Local oscillations

Example: Describing Patterns

Learn a linear (or non-linear) line using part of the data

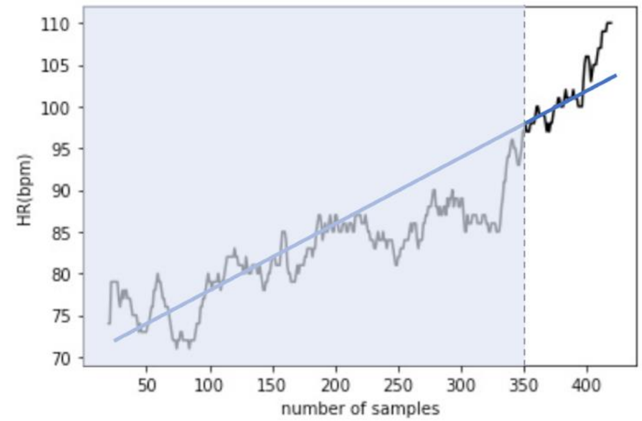


- Training data: Samples 1-350

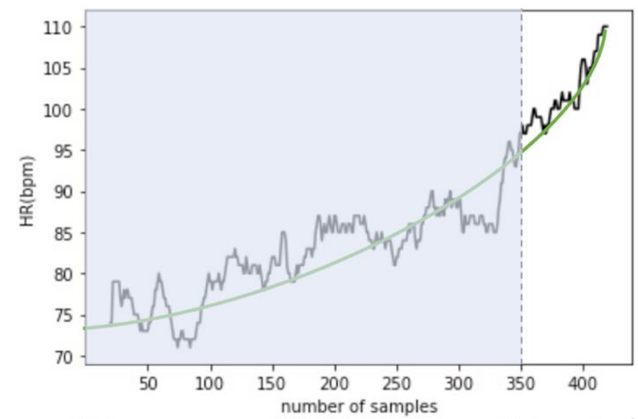
Example: Describing Patterns

Learn a linear (or non-linear) line using part of the data

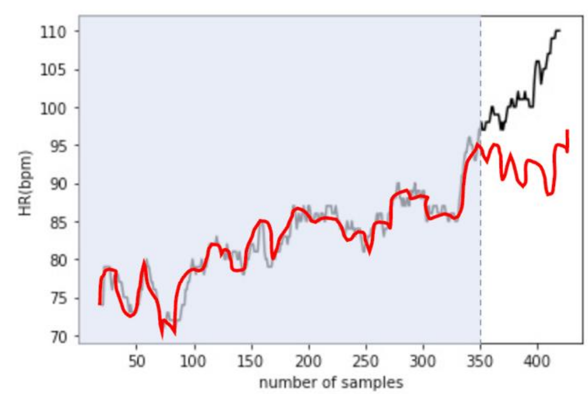
- Linear regression (blue line) is too simple
- 2nd degree non-linear regression (green line) captures the general trend
- 9th degree regression (red line) is complex (more than needed?)



$$HR = c_0 + c_1 \times NSamples$$



$$HR = c_0 + c_1 \times NSamples + c_2 \times NSamples^2$$

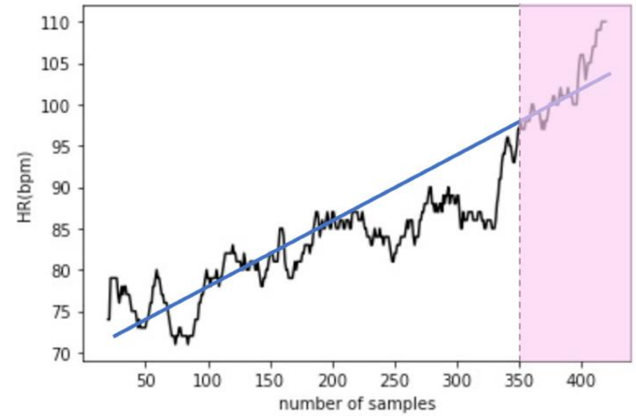


$$HR = c_0 + c_1 \times NSamples + c_2 \times NSamples^2 + c_3 \times NSamples^3 + \ldots + +c_9 \times NSamples^9$$

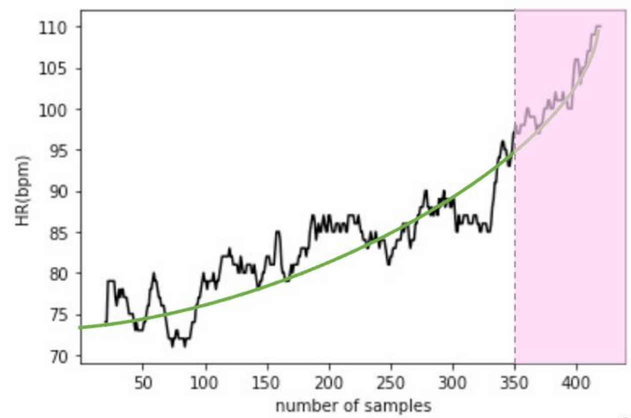
Example: Predicting Future Values

What is the heart rate for future time points based on each model

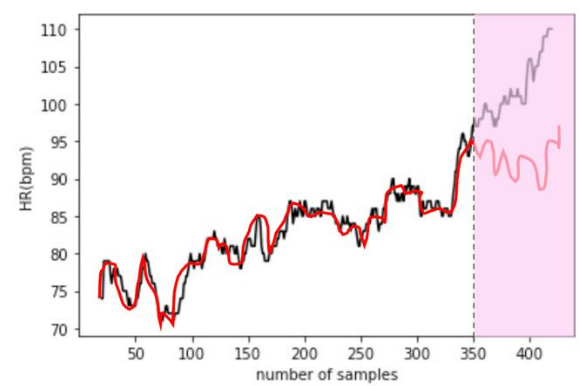
- Testing data: Samples 350-420
- Linear regression (blue line) is okay, but fails to accurately predict values after sample 400
- 2nd degree non-linear regression (green line) is not fully accurate, but is close enough!
- 9th degree regression (red line) completely fails to accurately predict future values



$$HR = c_0 + c_1 \times NSamples$$



$$HR = c_0 + c_1 \times NSamples + c_2 \times NSamples^2$$



$$HR = c_0 + c_1 \times NSamples + c_2 \times NSamples^2 + c_3 \times NSamples^3 + \ldots + +c_9 \times NSamples^9$$

The three components of learning

Representation	Evaluation	Optimization
Instances	Accuracy/Error rate	Combinatorial optimization
K-nearest neighbor	Precision and recall	Greedy search
Support vector machines	Squared error	Beam search
Hyperplanes	Likelihood	Branch-and-bound
Naive Bayes	Posterior probability	Continuous optimization
Logistic regression	Information gain	Unconstrained
Decision trees	K-L divergence	Gradient descent
Sets of rules	Cost/Utility	Conjugate gradient
Propositional rules	Margin	Quasi-Newton methods
Logic programs		Constrained
Neural networks		Linear programming
Graphical models		Quadratic programming
Bayesian networks		
Conditional random fields		

a learner must be represented in some formal language

a loss function assessing the performance of a learner

the process of finding the highest-scoring learner based on the loss function

Source: P. Domingos, 2014

Types of Learning

- Supervised (or predictive) learning
  - Learns associations between inputs and outputs
  - Requires labelled data, i.e., set of (input, output) pairs
  - Evaluated via obvious error metrics, e.g., accuracy
- Unsupervised (or descriptive) learning
  - Finds hidden/interesting structure in data (“knowledge discovery”)
  - Training data is not labelled, i.e., does not include desired outputs
  - Less well-defined problem with less obvious error metrics, e.g., cluster coherence
- Reinforcement learning
  - The learner interacts with the world via actions
  - Finds the optimal policy of behavior based on “rewards” it receives
  - Labels obtained as the training progresses

Supervised Learning

- Learning a mapping from inputs $\mathbf{x}_i$ to outputs y_i given a labelled set of input-output pairs (N samples)

$$\mathcal{D} = \{(\mathbf{x_i}, y_i)\}_{i=1}^{N}$$

- Data Matrix (N samples, D features)

$$\mathbf{X} = [\mathbf{x_1}^T \ldots \mathbf{x_N}^T] \in \Re^{D \times N} \qquad \mathbf{x_i} \in \Re^{1 \times D}$$

- Function approximation, function f is unknown and we approximate it

$$y = f(\mathbf{x})$$

- Classification
  - y_i is categorical or nominal (C classes): $y_i \in \{1, \ldots, C\}$
- Regression
  - y_i is real-valued, usually scalar: $y_i \in \mathbb{R}$

Supervised Learning: Classification

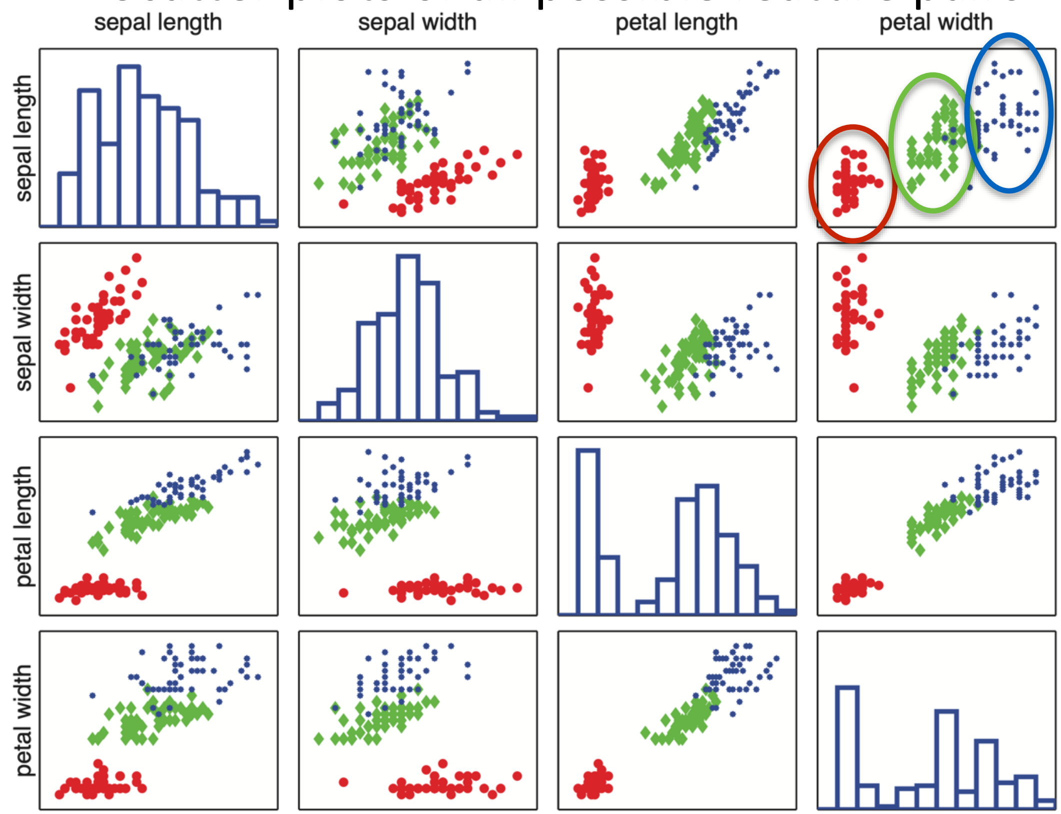
Recognizing types of Iris flowers (by R. Fisher)

setosa "●"

versicolor "◆"

virginica "✶"

Scatter plots of all possible feature pairs

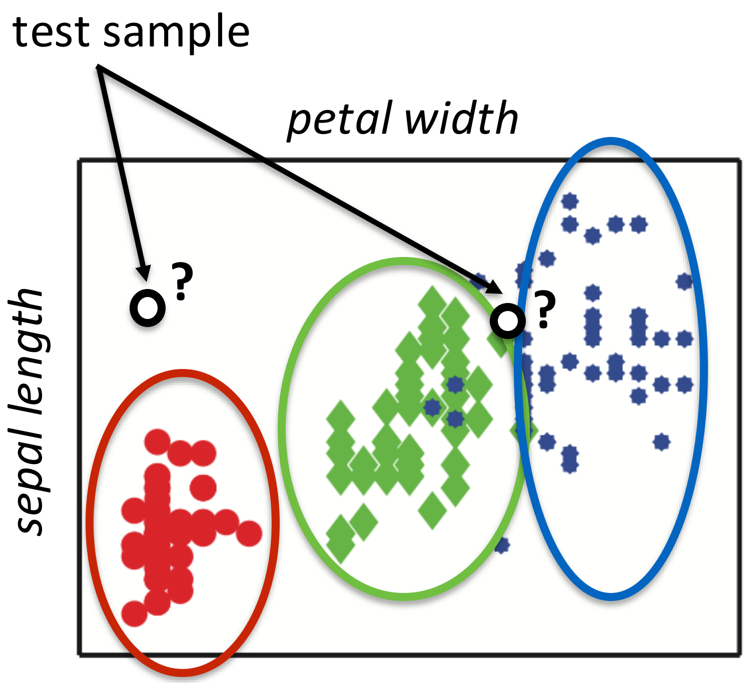


Exploratory data analysis (intuition)

Supervised Learning: Classification

Recognizing types of Iris flowers (by R. Fisher)

setosa "●", versicolor "◆", virginica "✶"



K-Nearest Neighbor (K-NN) classifier

- Test sample **x** is assigned to the most common class among its neighbors [N]

$$y = f(\mathbf{x}) = arg \max_{c=1,\ldots,C} v_c$$

most common class

number of votes from class c

Brief probability review

Probability

- P(A): probability that event A is true
  - A: “it will rain tomorrow”
  - p(A)=0.2: “there is 20% chance of rain tomorrow”

Conditional probability

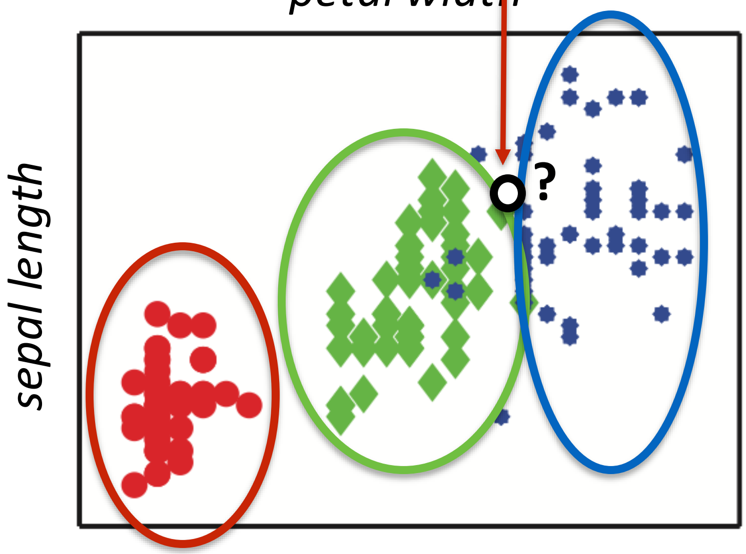
- P(A|B): probability of event A, given that event B is true
  - A: “it will rain tomorrow”
  - B: “today is humid”, C: “today is windy”
  - p(A|B): “chance of rain tomorrow, given that today is humid”, e.g. p(A|B)=0.6
  - p(A|B⋀C): “chance of rain tomorrow, given that today is humid and windy”, e.g. p(A|B⋀C)=0.7

Supervised Learning: Classification

Recognizing types of Iris flowers (by R. Fisher)

setosa "●", versicolor "◆", virginica "✶"

ambiguous test sample



The need of probabilistic predictions

- The right class of testing samples is unclear
- Return probabilities to handle ambiguity

$$y = f(\mathbf{x}) = arg \max_{c=1,\ldots,C} p(y = c|\mathbf{x}, \mathcal{D})$$

most **likely** class

posterior probability: probability of test sample belonging to class c given input vector **x and** training set D

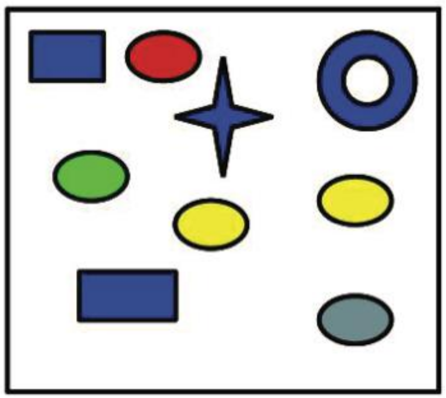
MAP estimate (maximum a posteriori)

Why is it important to model uncertainty?

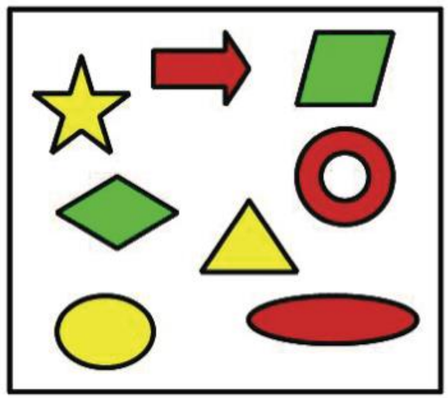
Question: Given the training data below, what would be a reasonable probability that a classifier would assign to the following test samples?

Training Set D

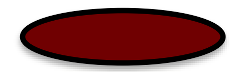
Class A



Class B



Test Set

Test sample s1: 

Test sample s2: 

A. P(s1∈A|D)=0.9, P(s2∈A|D)=1

B. P(s1∈B|D)=0.9, P(s2∈B|D)=0.1

C. P(s1∈B|D)=0.9, P(s2∈A|D)=0.5

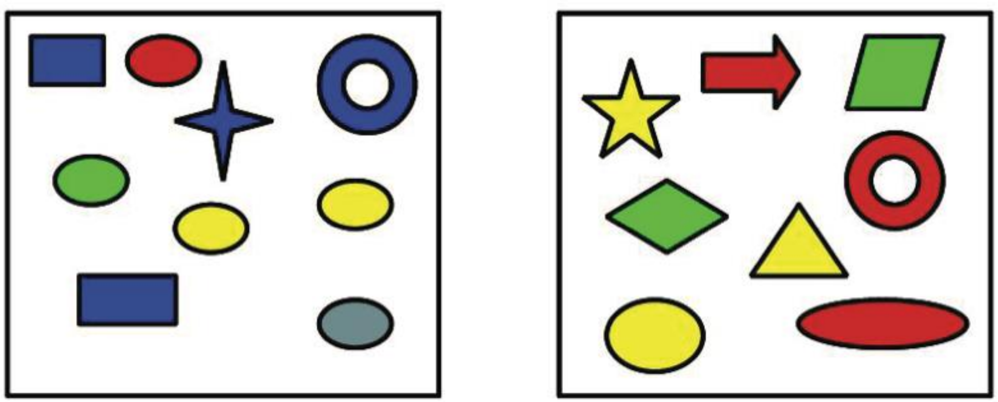
D. None of the above

Why is it important to model uncertainty?

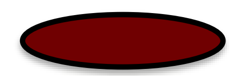
Question: Given the training data below, what would be a reasonable probability that a classifier would assign to the following test samples?

Training Set D

Class A **Class B**



Test Set

Test sample s1: 

Test sample s2: 

A. $P(s1 \in A | D) = 0.9$, $P(s2 \in A | D) = 1$

B. $P(s1 \in B | D) = 0.9$, $P(s2 \in B | D) = 0.1$

C. $P(s1 \in B | D) = 0.9$, $P(s2 \in A | D) = 0.5$

D. None of the above

Correct is C

Supervised Learning: Regression

Predict the price of a used car

- Input: $\mathbf{x} = [x_1, \ldots, x_D]^T$, car attributes (e.g., brand, year, mileage)
- Output y: price of car
- Model parameters: $\mathbf{w} = [w_1, \ldots, w_D]^T$

- Deterministic linear model

$$y = f(\mathbf{x}|\mathbf{w}) = \mathbf{w}^T \mathbf{x}$$

- Deterministic non-linear model (ϕ: non-linear function)

$$y = f(\mathbf{x}|\mathbf{w}) = \mathbf{w}^T \phi(\mathbf{x})$$

- Non-linear model - Probabilistic interpretation

$$y = f(\mathbf{x}|\mathbf{w}) = \mathbf{w}^T \phi(\mathbf{x}) + \epsilon, \quad \epsilon \sim \mathcal{N}(\mu, \sigma^2)$$

Unsupervised Learning

- Discovering structure (patterns, regularities, etc.) in "unlabelled" data
- Density estimation: we want to see what generally happens and what not

$$p(\mathbf{x_i}|\boldsymbol{\theta})$$

instead of $p(y_i|\mathbf{x_i};\boldsymbol{\theta})$ (supervised learning)

- Clustering
  - identifying sub-populations in the data
- Dimensionality reduction
  - project data to a lower dimensional subspace capturing its essence
- Matrix completion
  - data imputation to infer values of non-existing entries

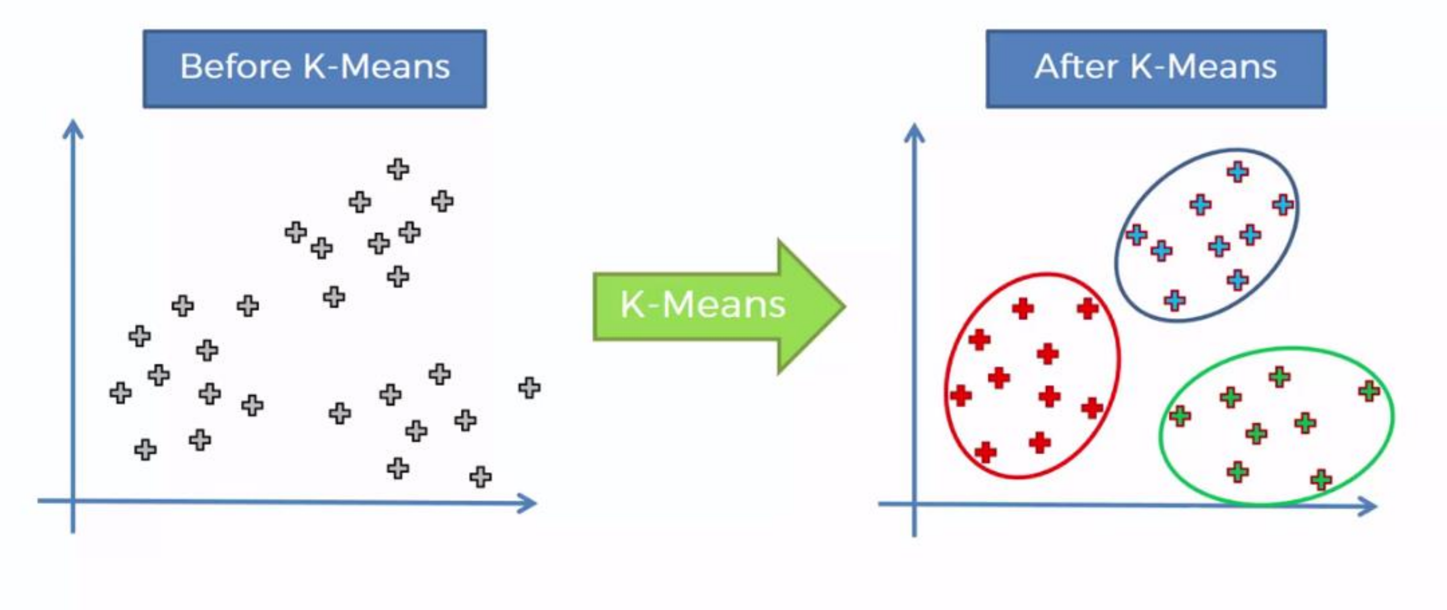
Unsupervised Learning: Clustering

- Step 1: Estimate the distribution over the number of clusters

$$p(K|\mathcal{D})$$

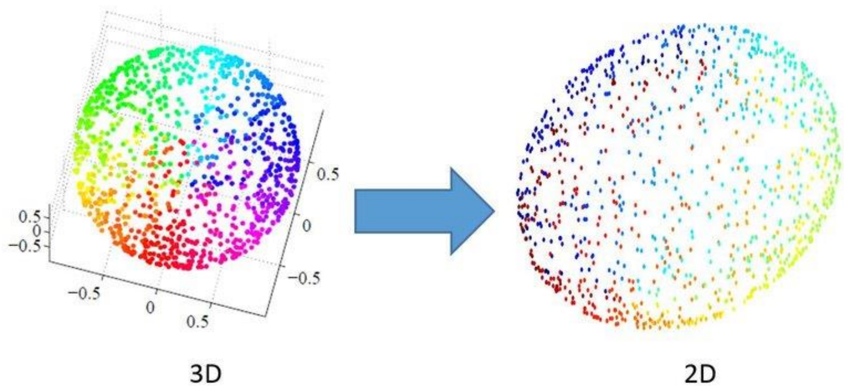
- Step 2: Estimate which cluster each point belongs to

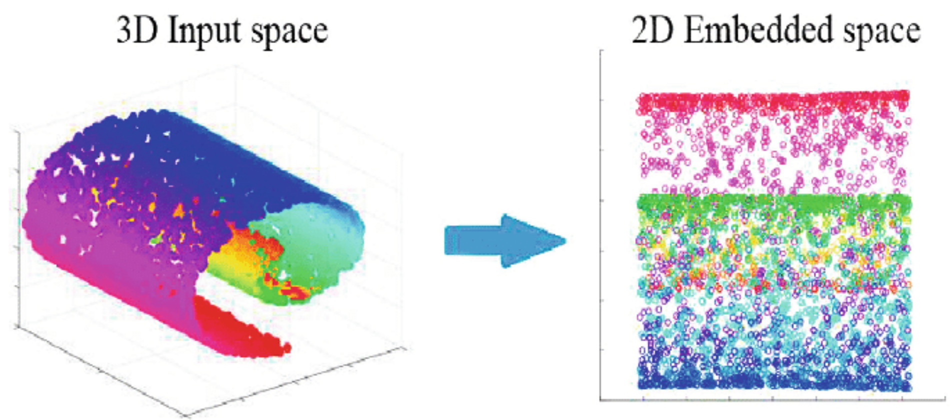
$$z_i^* = arg \max_{k=1,\ldots,K} p(z_i = k|\mathbf{x_i}, \mathcal{D})$$



Unsupervised Learning: Dimensionality Reduction

- Lower dimensional representations can have better predictive power
  - minimized data redundancies
  - avoiding "curse of dimensionality"





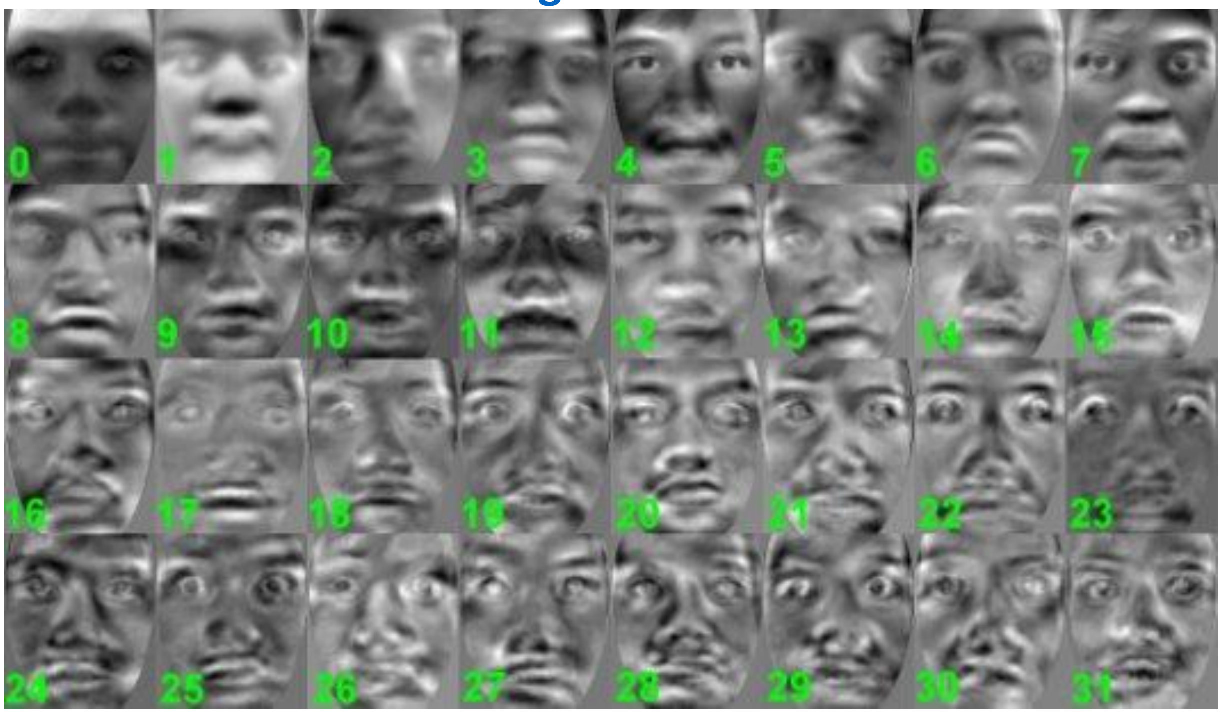
Principal component analysis (PCA)

identifies a set of uncorrelated axes that maximize the variance of the data

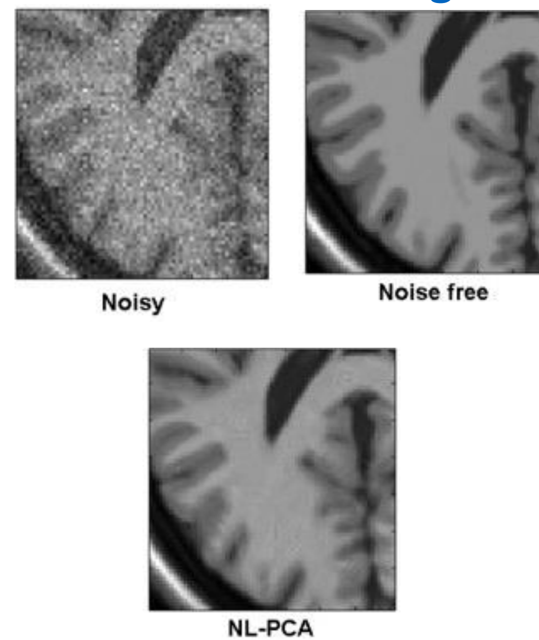
Unsupervised Learning: Dimensionality Reduction

Example applications of PCA

Eigenfaces



MRI denoising



Unsupervised Learning: Matrix completion

Recommender systems

users

movies

1		?	3	5	?
?	1				2
	4		4	5	?

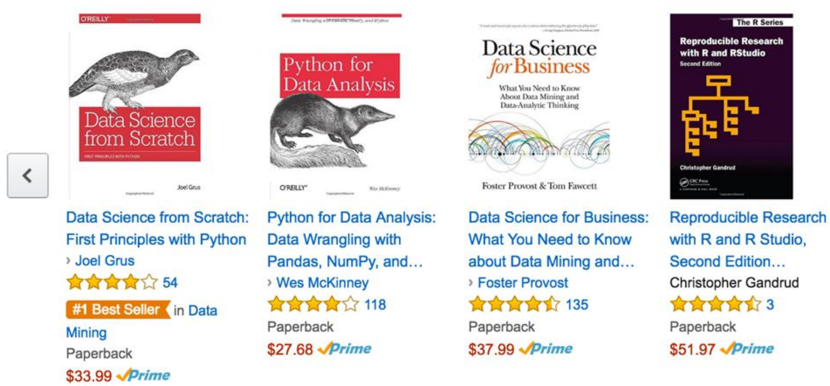
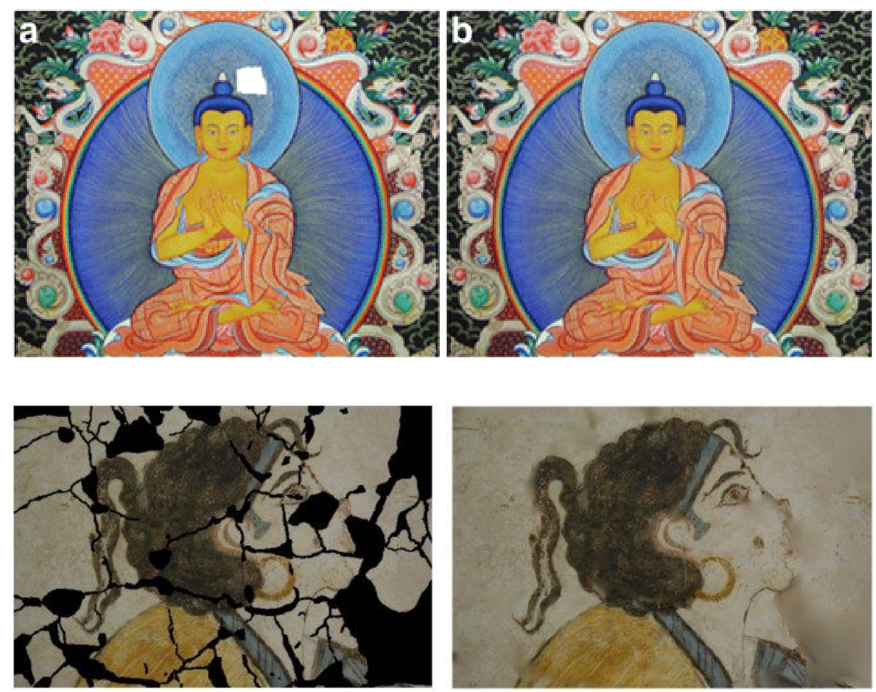


Image restoration



Sources: Wang & Jia, 2017;

Papandreou, Maragos, & Kokaram, 2008

To sum up

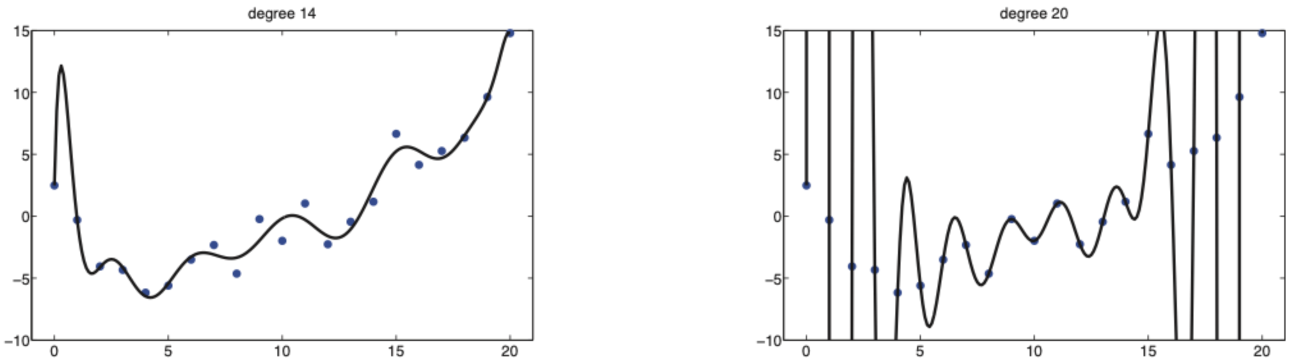
- Machine learning definition
- Key components of learning: representation, evaluation, optimization
- Types of learning systems: supervised & unsupervised

Key Machine Learning Challenges
Generalization

- Biggest ML challenge is to generalize beyond the training set
- **Never evaluate your ML system on the train data only**
  - Use development set for hyper-parameter tuning
  - Use test data for final evaluation
- Contamination of the ML system from the test data can occur when:
  - use test through excessive parameter tuning
    - Avoid this with (**cross-)validation** and **development set**
- On the positive side 🙂
  - We may not need to fully optimize it, since the objective function is only a proxy of the true one

Key Machine Learning Challenges
Overfitting

- The risk of using highly flexible (complicated) models without having enough data
- Ways to avoid overfitting
  - (cross-)validation
  - regularization

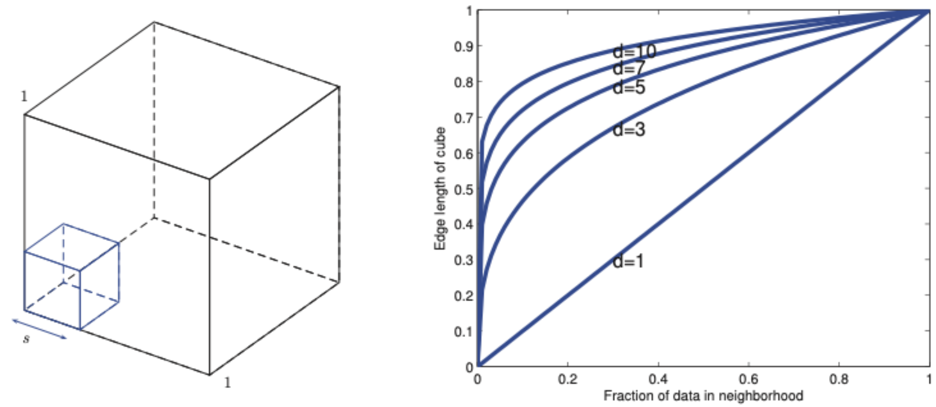


(a) (b)

Example of polynomial fit

Key Machine Learning Challenges
Curse of dimensionality

- All intuition fails in higher dimensions
- For a fixed training set, generalization gets harder in larger dimensions
  - harder to systematically search a high-dimensional grid-space
  - harder to accurately approximate a high-dimensional function
- On the positive side 🙂
  - "blessing of non-uniformity": examples aren't usually spread uniformly



(a) (b)

Key Machine Learning Challenges
Feature Engineering

- Learning is easy if you have informative features for the problem
- Automating the feature engineering process
  - Deep learning systems producing output from raw input

Key Machine Learning Challenges
"No-free-lunch" theorem

- "All models are wrong but some models are useful", G. Box, 1987
- There is no single best ML system that works optimally for all kinds of problems
- On the positive side 🙂
  - General assumptions can actually work pretty well, e.g.
    - Similar examples belong to similar classes
    - Independence and smoothness assumptions
- We might need to try lots of different ML systems and learning algorithms to cover the wide variety of real-world data.
- Machine learning is not magic: it can't get something out of nothing, but it can get more from less!

To sum up

- Machine learning definition
- Key components of learning: representation, evaluation, optimization
- Types of learning systems: supervised & unsupervised
- Challenges in machine learning

Readings:

- Alpaydin Ch1, Abu-Mostafa Ch 1
- Syllabus; check conflicts with exams dates and notify course staff ASAP