Machine Learning:
Basic Concepts

Week 1

Machine Learning for Problem Solving

www.andrew.cmu.edu/user/lakoglu/courses/95828/index.htm

Leman Akoglu
Asst. Prof. of Information Systems

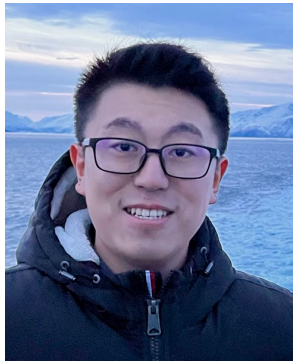
Heinz College

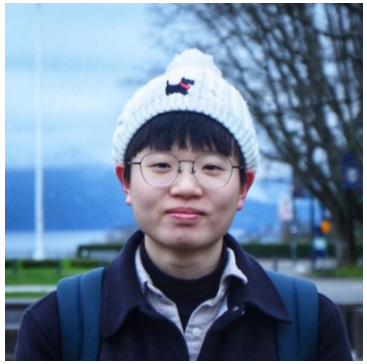
Carnegie Mellon University

Course staff

- Instructor:
  Dr. Leman Akoglu

- **TA**s:
  - Zijun Ding
  - Shahriar Noroozizadeh
  - Xiaobin Shen





All Office Hours (OH) will be in person.
Locations will be found on Course website.

Course Logistics

What this course is about

- Covers a wide range of ML techniques, from basic to state-of-the-art
  - There exist many models and algorithms to fit them;
  - The set of assumptions that works well in one domain may work poorly in another;
  - Therefore, we will learn different types of models and algorithms, with varying accuracy-speed-complexity-interpretability trade-offs.
  - Will cover most often used models & algorithms:
    - regression, Lasso, nearest neighbors, decision trees, Naïve Bayes, ensemble methods, boosting, bagging, overfitting, regularization, model selection, dimensionality reduction, PCA, SVMs, kernels, k-means clustering, EM, …

What this course is about

- Covers a wide range of ML techniques
  - with varying accuracy-speed-complexity-interpretability
- Helps you gain hands-on experience applying ML algorithms on real world datasets
- Emphasis will be on which ML methods exist, how they work, and how to use them on real data
  - with less emphasis on theory (i.e., why and when they work)
- It is going to be fun and hard work! ☺

Learning objectives

By the end of the semester, you should be able to:

- *Approach problems data-analytically*
  - Look at a real-world problem and decide if ML is an appropriate approach
  - If so, identify which type of ML problem it is and what types of models and algorithms might be applicable

- *Be able to implement ML solutions*
  - Apply ML algorithms on the real-world data using best practices
  - Evaluate your solution and technically communicate performance results

What you will gain

Skills

- Regression
- Classification
- Clustering
- ML best practices for model selection and evaluation
- …

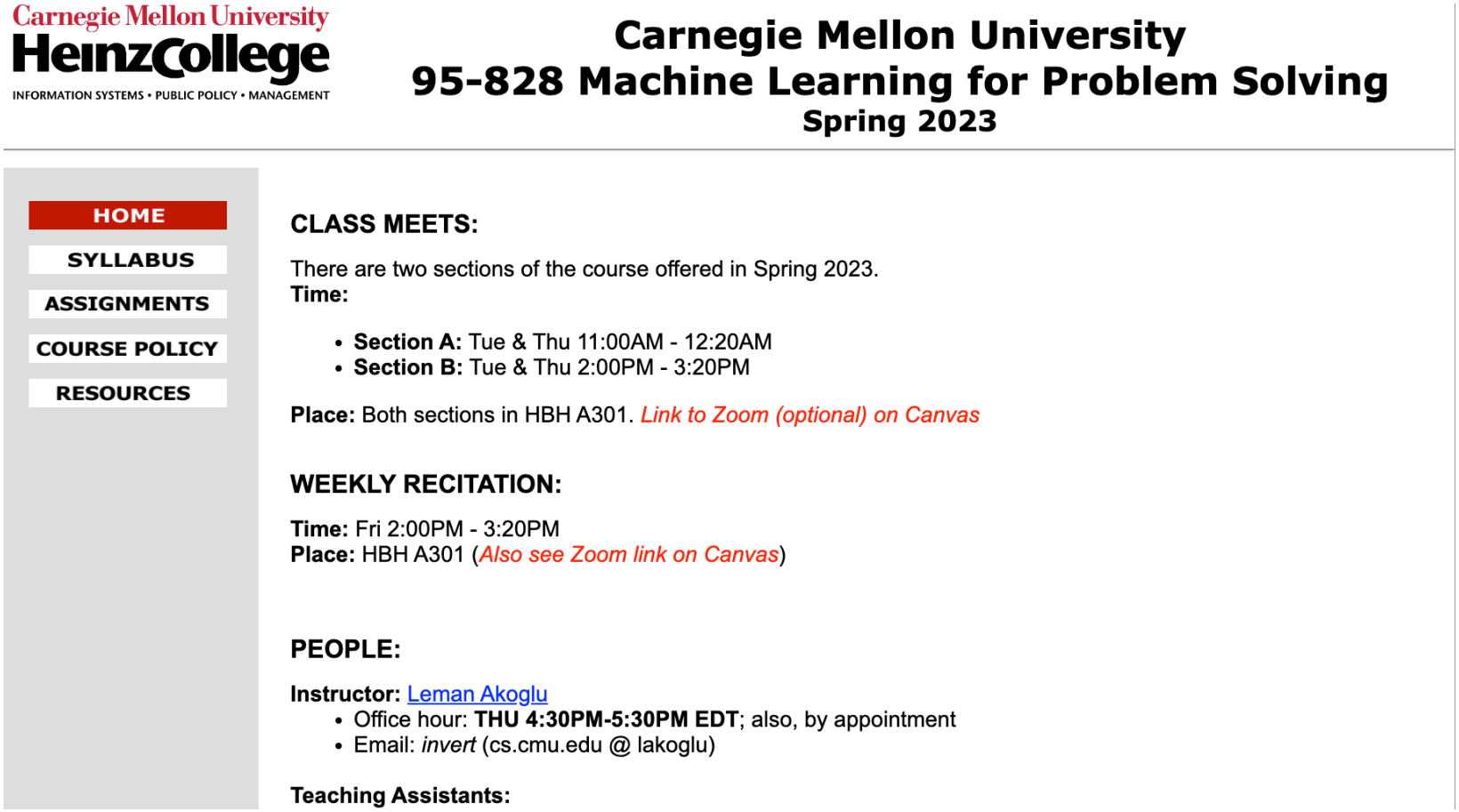
Practices

- Hands-on experience on ML problems on real-world datasets
- Scikit-Learn
- Scipy
- …

Course website(s)

http://www.andrew.cmu.edu/user/lakoglu/courses/95828/index.htm → Main **course website**

- People (contact, office hours), Syllabus (week by week), Assignment information (grading, due dates, etc.), Course policy, Resources



Carnegie Mellon University
HeinzCollege
INFORMATION SYSTEMS • PUBLIC POLICY • MANAGEMENT

Carnegie Mellon University
95-828 Machine Learning for Problem Solving
Spring 2023

HOME
SYLLABUS
ASSIGNMENTS
COURSE POLICY
RESOURCES

CLASS MEETS:

There are two sections of the course offered in Spring 2023.
Time:

- **Section A:** Tue & Thu 11:00AM - 12:20AM
- **Section B:** Tue & Thu 2:00PM - 3:20PM

Place: Both sections in HBH A301. *Link to Zoom (optional) on Canvas*

WEEKLY RECITATION:

Time: Fri 2:00PM - 3:20PM
Place: HBH A301 (*Also see Zoom link on Canvas*)

PEOPLE:

Instructor: Leman Akoglu
- Office hour: **THU 4:30PM-5:30PM EDT**; also, by appointment
- Email: *invert* (cs.cmu.edu @ lakoglu)

Teaching Assistants:

Course website(s)

- **Canvas** https://www.cmu.edu/canvas/
  - **Lecture Notes & Lecture Slides**
  - Assignments & Solutions
  - Email announcements
  - Uploading and Grading HW
  - Quizzes (to-be-done before each week's recitation)

- **Piazza** https://piazza.com/cmu
  - Questions and discussion (on HW, course material, etc. – anything on machine learning)
  - Sign up @ http://piazza.com/cmu/spring2023/95828

Lectures

- **A: TUE/THU 11:00AM – 12:20PM EDT**
  B: TUE/THU 2:00PM – 3:20PM EDT
  - Zoom link: https://bit.ly/2MinJgY (also on Canvas)
  - Recorded live, posted on Canvas after lecture

- How you are expected to use Zoom:
  - "Raise hand" : for immediate clarification questions
  - Post questions on "chat" : to be answered at break-points (see next slide)

- Audio/video
  - Mute unless talking
  - Video optional, but photo strongly recommended

Lectures

- Break-points (every ~15-20 minutes)
  - I will pause for Q&A : to answer “chat” questions as well as live questions
  - We will also have one “bio-break” somewhere in the middle
    - Stretch, relax, grab a drink/snack, …

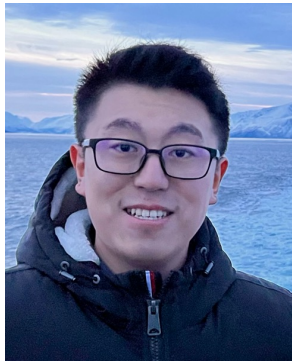
Lectures

- Break-points (every ~15-20 minutes)
  - I will pause for Q&A : to answer "chat" questions as well as live questions
  - We will also have one "bio-break" in the middle
    - Stretch, relax, grab a drink/snack, …

- "Prompt/poll questions" during class
  - Type your (short) answers on "chat" privately to the instructor (Leman)
  - Not graded – only for me to monitor general understanding & make clarifications

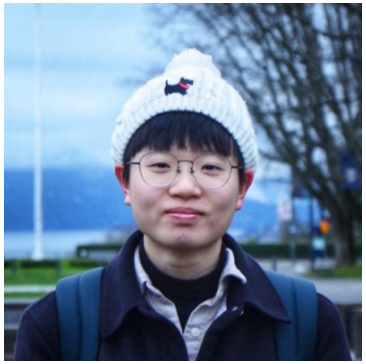
Course staff & Communication

- Instructor: Dr. Leman Akoglu
  - Office hour (OH): **THU 4:30-5:30PM EDT**
  - Also by appointment
- **TA** Ohs (all in EDT):
  - Zijun Ding
    OH: **TUE 7:30-8:30PM**



  - Shahriar Noroozizadeh
    OH: **FRI 5-6PM**

  - Xiaobin Shen
    OH: **FRI 10-11AM**



  - All office hours to be held in person (locations on website)
- Please also use Piazza for questions

Recitations

- TAs will hold weekly recitations to
  - Review material
  - Give you demos in Python
  - Answer clarification questions about homework and week's quiz questions
  - Walk you through assignments

- When: **Every FRI 2:00-3:20 PM EDT**
  Where: HBH A301

  Zoom link: https://bit.ly/3fuYgvV (may be updated, use the link on Canvas)

Graders

Graders:

Yanjun Chen
- Email: *invert* (andrew.cmu.edu @ yanjunch)
- Office hours: by appointment

Rui Pan
- Email: *invert* (andrew.cmu.edu @ ruipan)
- Office hours: by appointment

Yilin Cao
- Email: *invert* (andrew.cmu.edu @ yilincao)
- Office hours: by appointment

- Contact for re-grading requests (See Course Policy)

Syllabus

- “Machine Learning is magic!” *
  - * “mathemagic” – involves statistics and math

- Syllabus covers core concepts and topics that are frequently used in practice

Syllabus

Week	Lectures	Notes
Week 1	**INTRO TO MACHINE LEARNING** [+]	**HW 0 out** • Python and Jupyter setup
	DATA PREPARATION [+]	**Recitation 1** • Python setup • Data prep demos • Linear Algebra review
Week 2	**LINEAR REGRESSION (LR)** [+]	**Recitation 2** • Linear Regression review and demos • Convex optimization basics
Week 3	**MODEL SELECTION** [+]	HW 1 out • EDA • LR • Model selection • LogR
	LOGISTIC REGRESSION (LogR) [+]	**Recitation 3** • Cross-validation • LogR • Gradient descent
Week 4 • Week 5	**NON-PARAMETRIC LEARNING** [+]	**Recitation 4** • Non-parametric learning • kNN • Local regression
	MODEL EVALUATION [+]	HW 2 out • Non-parametric learning • Model evaluation • DT
		Recitation 5 • Model evaluation
Week 6 • Week 7	**DECISION TREES (DT)** [+]	**Recitation 6** • DT • Random Forest
	ENSEMBLE METHODS [+]	
	NAIVE BAYES (NB) [+]	**Recitation 7** • Boosting • NB

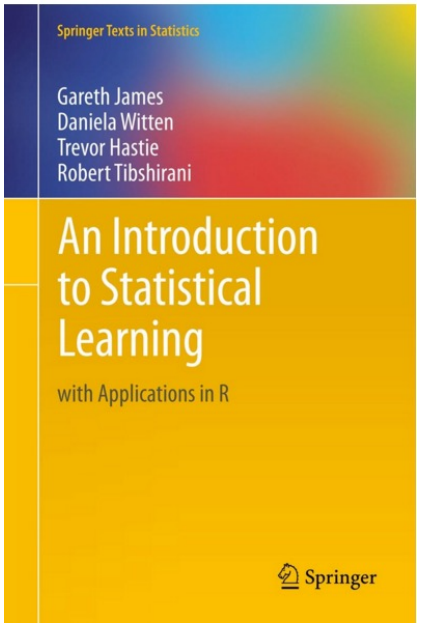
Syllabus (continued)

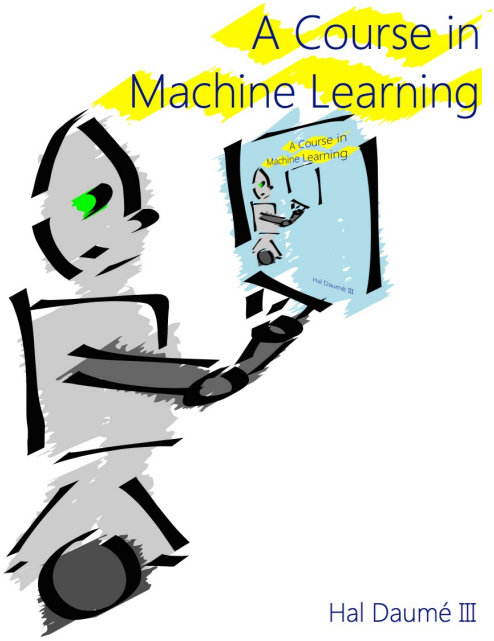
Week	Lectures	Notes
Week 10	**SUPPORT VECTOR MACHINES (SVM)** [+]	HW 3 out • Ensembles • NB • SVM **Recitation 8** • SVM and Kernels
Week 11	**NEURAL NETWORKS (NN)** [+]	**Recitation 9** • NNs • Back-propagation
Week 12 • Week 13	**DEEP LEARNING** [+] PART II: UNSUPERVISED LEARNING **DENSITY ESTIMATION** [+]	HW 4 out • Kernels • Neural Nets • Density estimation **Recitation 10** • Deep learning • Density estimation
	Thur NO CLASS: Spring carnival	Friday NO RECITATION
Week 14 • Week 15	**CLUSTERING** [+] **DIMENSIONALITY REDUCTION** [+]	HW 5 out • Clustering • EM • Dimensionality reduction **Recitation 11** • Clustering • k-means • EM **Recitation 12** • Dim. reduction
Week 16	**RECOMMENDER SYSTEMS** [+] **Case Study & Final Review**	Case Study out • Dataset provided, Tasks recommended **Recitation 13** • Recommender systems • Case Study review • Final Q&A

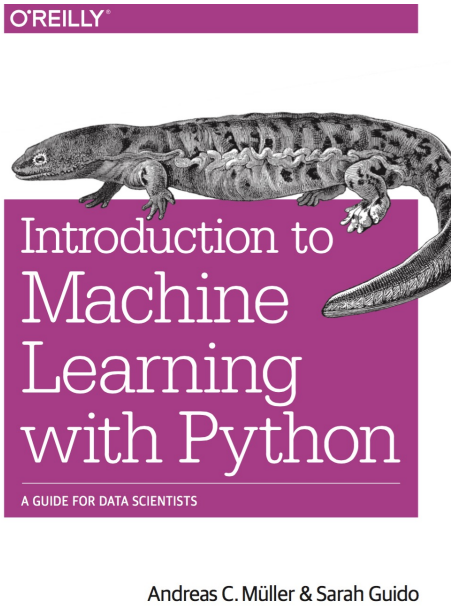
Textbook and Lecture notes

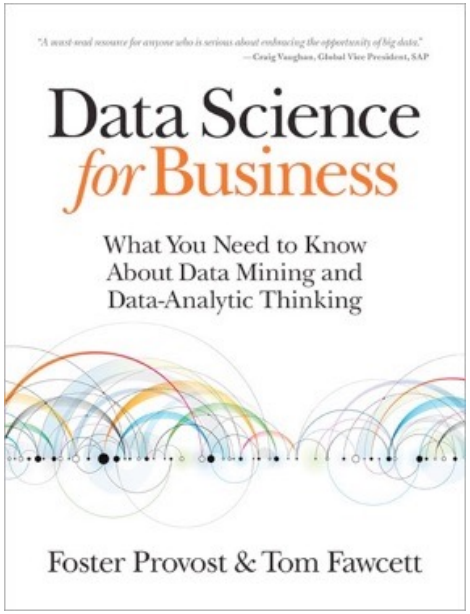
- I will post Lecture notes (in pdf) (along with slides) *before* class on Canvas
  - I strongly suggest you read them before class, or right after class (or both!)
- No required textbook. Recommended textbooks:
  - (ISL) An Introduction to Statistical Learning, [FREE pdf]
    Gareth James, Daniela Witten, Trevor Hastie, Robert Tibshirani
  - (CML) A Course in Machine Learning, [FREE pdf]
    Hal Daumé III
  - (MLP) Introduction to Machine Learning with Python, O'Reilly
    Sarah Guido and Andreas Muller
  - (DSB) Data Science for Business, O'Reilly
    Foster Provost and Tom Fawcett

Recommended textbooks









ISL CML MLP DSB

- Lecture notes partly but mainly from various sections in these books (see Acknowledgements)

Course Workload

- Coursework consists of (grading in parentheses):
  - 5 Homework (9% each)
  - 1 Midterm exam (15%)
  - 1 Final exam (25%)
  - 1 Case Study (15%)

Course Workload

- 5 HW (~2 weeks each), 2 Exams, 1 Case Study

IMPORTANT DATES:

Assignment	Note	Out	Due	Weight
Homework 0	Setting up Python and Jupyter	Jan 17	n/a	0%
Homework 1	EDA, LR, Model selection	Jan 31	Feb 14	9%
Homework 2	LogR, Model eval., Non-parametric, DT	Feb 14	Feb 28	9%
Midterm Exam	(in class)	Mar 2	--	15%
Homework 3	Ensemble models, NB, SVM	Mar 14	Mar 28	9%
Homework 4	Kernels, Neural nets, Density estimation	Mar 28	Apr 11	9%
Homework 5	Clustering, EM, Dimensionality reduction	Apr 11	Apr 25	9%
Case Study	Mini 4	Mar 16	Apr 28	15%
Final Exam		Check out univ. calendar	--	25%

HW assignments

- HW (as well as solutions) will be posted on Canvas
- Each will have two parts:
  - Conceptual: short questions answered in words
  - Programming/Applied: we will provide partial code in iPython/Jupyter notebook

- Submitting: Upload to Gradescope, 2 files per HW:
  1. A pdf file with answers to conceptual questions
  2. A Jupyter notebook with filled-in code and text
  - Can upload multiple times, BUT last submission marks the "submission time".

HW assignments - help

- Getting help:
  - Post questions on Piazza
  - Visit the instructor and the TAs during office hours
  - Work with the others in the class
- Collaboration
  - Each student can discuss the questions with others, but writes their own answers.
- **Plagiarism** (penalized heavily, no exceptions!)
  - **Please do *not* search for answers on the Web**
  - Ask us when not sure if you can use a reference
- Also see The Carnegie Mellon Code at http://www.cmu.edu/academic-integrity/

Case Study

- Performed in **groups of 4**
  - You can use Piazza to find team member(s)

- In 2nd half of course (after Midterm exam), we will provide you with
  - A large dataset
  - A list of potential ML problems using the dataset
- You will be expected to
  - carefully choose to apply the techniques and tools you have learned throughout the course
  - to address problems of your interest using machine learning on the provided dataset

Case Study

- Evaluation: We will assess your case study outcomes in terms of
  - how methodical you were in your analysis in terms of the tools you chose to apply to solve the problem at hand,
  - in the way you draw conclusions from your own results,
  - the sequence of steps you took based on your analyses and intermediate results
  - if you used the best practices in building your solutions, including proper model selection, model comparisons to appropriate baselines, choice of evaluation metrics, etc.
- Submitting: (due last Friday of sem., no slip days)
  - submit a single Jupyter notebook on Canvas, composed of all your code and results

Assignments – slip days

- **4 late/slip days** granted to each student in total (no questions asked) to accommodate coinciding deadlines/interviews/etc.
  - You can use the extension on any assignment
    - e.g., you can hand in one assignment 4 days late, or four different assignments 1 day late each.
  - Late hours are rounded up to the nearest integer.
    - e.g., a submission that is 4 hours late will count as 1 day late, 26 hours late as 2 days late, etc.
  - After you have used up all your slip days, any late assignment will be marked off 25% per day of delay.
  - Late days apply to all study group members.

Next Steps for You…

- Start HW0 – setting up Python & Jupyter
  - download from Canvas
- Review Python and linear algebra basics
  - http://www.mypythonquiz.com/
  - http://ai.berkeley.edu/tutorial.html#PythonBasics
- Visit course web page
  - http://www.andrew.cmu.edu/user/lakoglu/courses/95828/index.htm
  - See syllabus, assignments, course policy
- Signup @ Piazza for asking questions
  - http://piazza.com/cmu/spring2023/95828
- Next lecture: Slides & Notes will be on Canvas

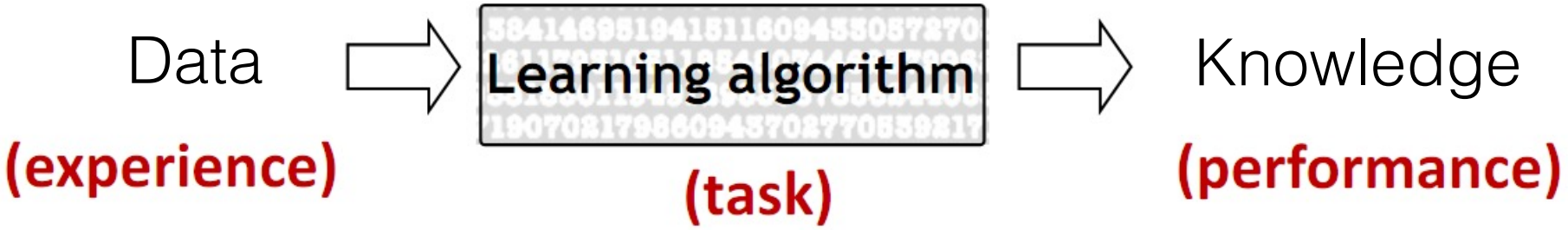
Enjoy!

- ML is ubiquitous in science, finance, engineering, and beyond
- This class should give you the foundation for applying ML techniques to real-world problems
- **Let the fun begin…**

Basic Concepts

What is Machine Learning?

- Study of algorithms that
  - improve their performance
  - at some task
  - with experience

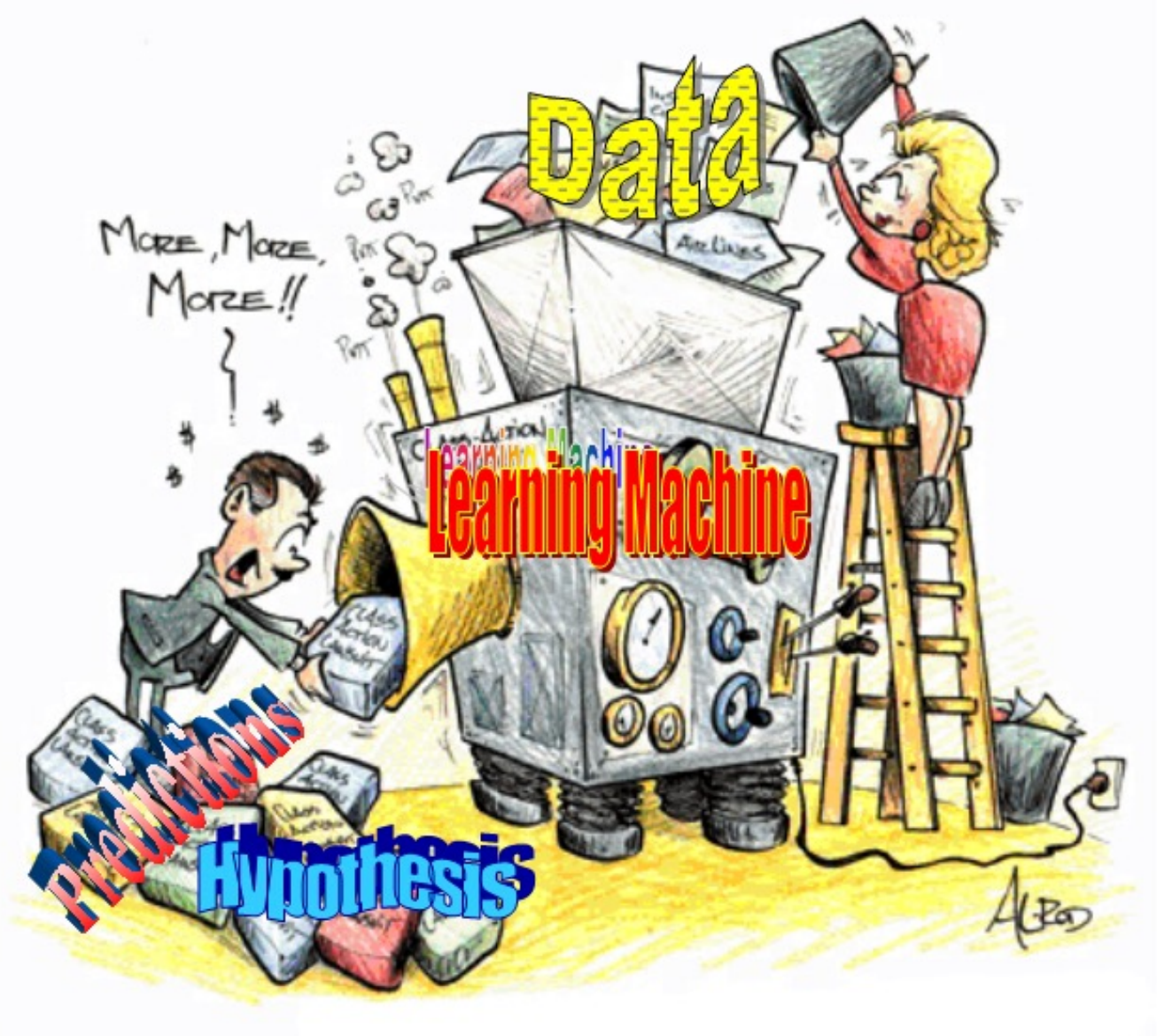


Learning problem

What is Machine Learning?

- The essence of machine learning:
  - Data contains patterns / structure.
  - We cannot pin it down mathematically.
  - But we have lots of data (observations) on it.
- Machine learning system: find and exploit patterns to figure out what humans want based on observations

What is Machine Learning?



Agenda

- Components of learning
  - Formalization
- Types of learning
- ML applications in the real world
- What does learning mean?
  - Overfitting and Generalization

Components of Learning

- Example application: loan application approval
- Application information:

age	23 years
annual salary	$30,000
years in residence	1 year
years in job	1 year
current debt	$15,000
...	...

- Approve loan? (yes/no)

Components of Learning I

Formalization:

- Input **example** : i (a.k.a. **observation, instance**)
  (loan application)

- **Features** : $\mathbf{x}_i$ (a.k.a. **attributes, variables, predictors**)
  (properties like age, salary, etc. derived from i)

- **Output** : y_i (a.k.a. **target, response, label**)
  (good/bad customer?)

- (Unknown) Target function $f : \mathcal{X} \rightarrow \mathcal{Y}$
  (ideal credit approval formula)

Components of Learning II (cont.)

- **Training data** : $(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_i, y_i), \ldots, (\mathbf{x}_n, y_n)$
  (historical examples of loan applications)

- Hypothesis set : $\mathcal{H} = \{h\}$
  (family of candidate formulas)

- Learning algorithm A
  (optimization / search algorithm)

- Hypothesis function $g \in \mathcal{H} : \mathcal{X} \to \mathcal{Y}$
  (a.k.a. **classifier** or **regressor**)
  (loan approval formula to be deployed)

- **Test data** : $(\mathbf{x}_{n+1}, ?), (\mathbf{x}_{n+2}, ?) \ldots$

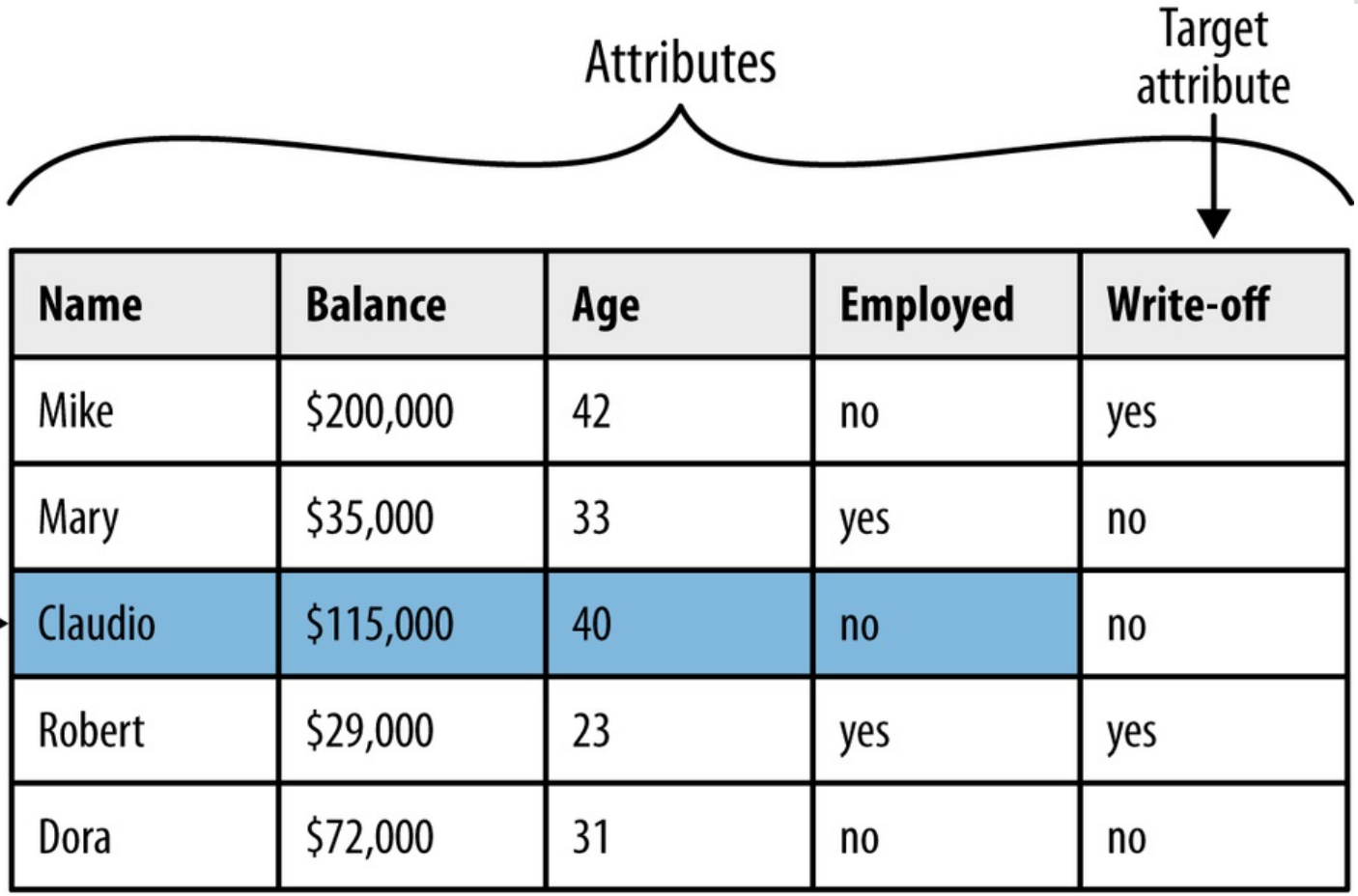
Data table

- We often assume data is **tabular**, i.e., provided in the form of a **data table** (or data matrix)

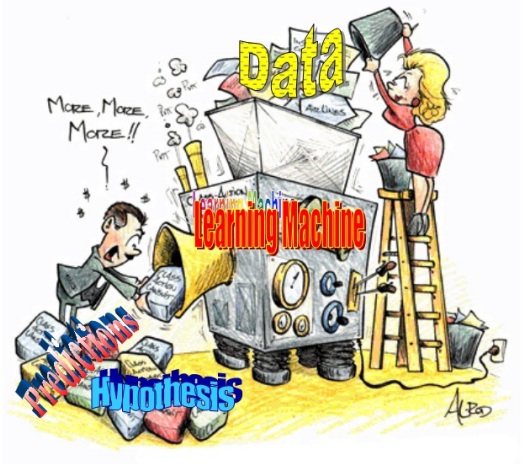
	feature 1	feature 2	…	…	feature d	target
example 1						
example 2						
…						
example n						

- Rows called **examples, observations**, tuples, instances, records, objects
- Columns called **features**, attributes, variables, covariates, predictors
- Output (if any) called target, response, label, outcome

Data table


Attributes
Target attribute
Name
Balance
Age
Employed
Write-off
Mike
$200,000
42
no
yes
Mary
$35,000
33
yes
no
Claudio
$115,000
40
no
no
Robert
$29,000
23
yes
yes
Dora
$72,000
31
no
no
This is one row (example).
Feature vector is: <Claudio,115000,40,no>
Class label (value of Target attribute) is no
i
$\mathbf{x}_i$
y_i

The Learning Problem



UNKNOWN TARGET FUNCTION
$f : \mathcal{X} \to \mathcal{Y}$
(ideal credit approval formula)

TRAINING EXAMPLES
$(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_i, y_i), \ldots, (\mathbf{x}_n, y_n)$
(historical records of loan applications)

TEST EXAMPLES
$(\mathbf{x}_{n+1}, ?), (\mathbf{x}_{n+2}, ?) \ldots$
(new incoming loan applications)

LEARNING ALGORITHM A
(search and optimization)

FINAL HYPOTHESIS
$g \in \mathcal{H} \approx f$
(learned credit approval formula)

HYPOTHESIS SET
$\mathcal{H}$
(set of candidate formulas)

PREDICTIONS
(predicted approval: yes/no labels)

Agenda

- Components of learning
  - Formalization
- Types of learning
- ML applications in the real world
- What does learning mean?
  - Overfitting and Generalization

”Bio” break

Song Playing:
Luciano - I Can You Can

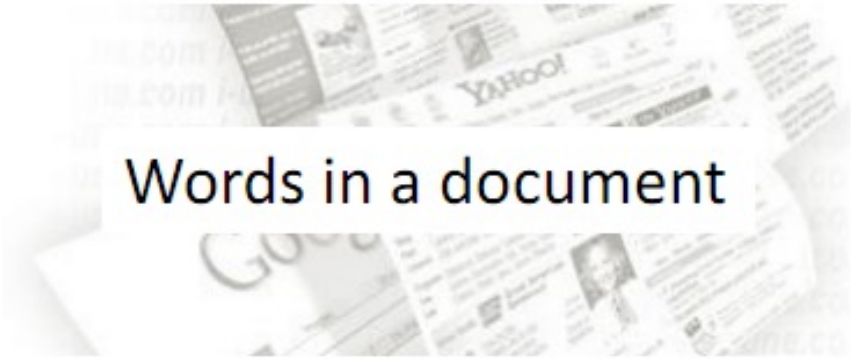
Canonical Learning Tasks

We will focus on two fundamental and popular learning paradigms:

- Supervised Learning (learning with teacher)
  Classification, Regression
- Unsupervised Learning (learning w/out teacher)
  Clustering, Density estimation, Dimensionality reduction
- Many other setups exist: (not in the scope of this course)
  - Reinforcement learning
  - Semi-supervised learning
  - Active learning
  - Imitation learning
  - Meta-learning (learning to learn)
  - …

Supervised Learning

Feature Space $\mathcal{X}$



Label Space $\mathcal{Y}$

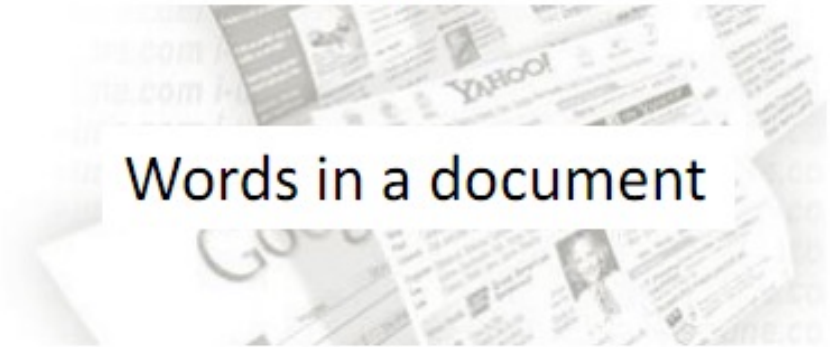
"Sports"
"News"
"Science"
...

Share Price
"$ 24.50"

Task: Given $X \in \mathcal{X}$, predict $Y \in \mathcal{Y}$.

Supervised Learning - Classification

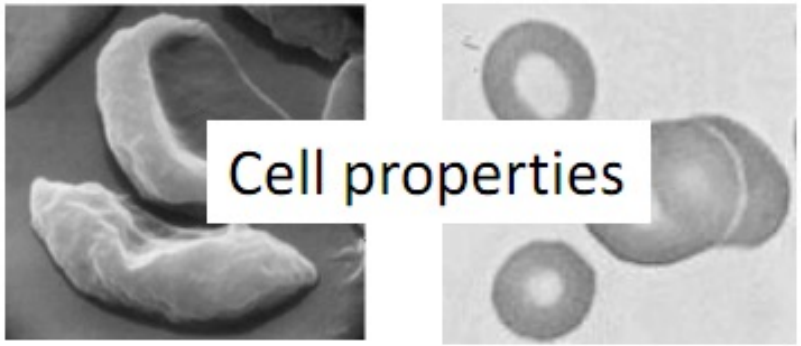
Feature Space $\mathcal{X}$



Label Space $\mathcal{Y}$

"Sports"
"News"
"Science"
...

Multi-class classification
(>2 classes)



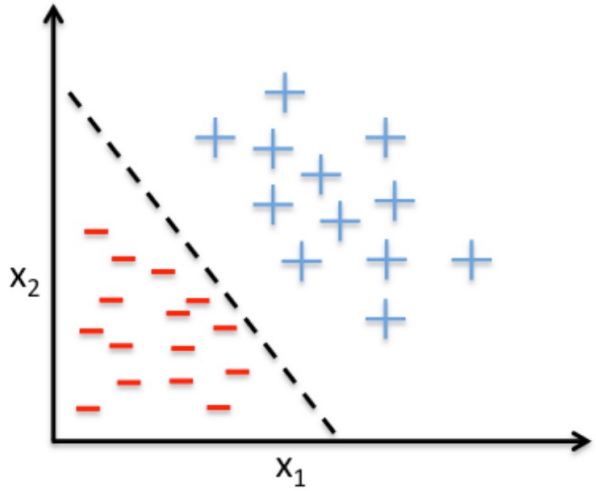
"Anemic cell"
"Healthy cell"

Binary classification
(yes/no output)

Discrete Labels

Supervised Learning - Classification

ID	Clump	UnifSize	UnifShape	MargAdh	SingEpiSize	BareNuc	BlandChrom	NormNucl	Mit	Class
1000025	5	1	1	1	2	1	3	1	1	benign
1002945	5	4	4	5	7	10	3	2	1	benign
1015425	3	1	1	1	2	2	3	1	1	malignant
1016277	6	8	8	1	3	4	3	7	1	benign
1017023	4	1	1	3	2	1	3	1	1	benign
1017122	8	10	10	8	7	10		7	1	malignant
1018099	1	1	1	1	2	10	3	1	1	benign
1018561	2	1	2	H	2	1	3	1	1	benign
1033078	2	1	1	1	2	1	1	1	5	benign
1033078	4	2	1	1	2	1	2	1	1	benign



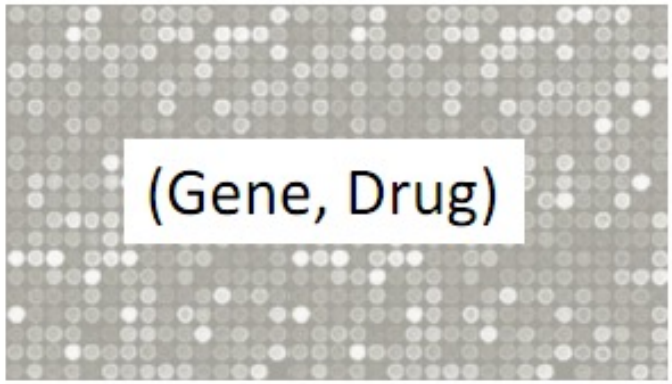
Supervised Learning - Regression

Feature Space $\mathcal{X}$

Label Space $\mathcal{Y}$

Share Price
"$ 24.50"

Real-valued output



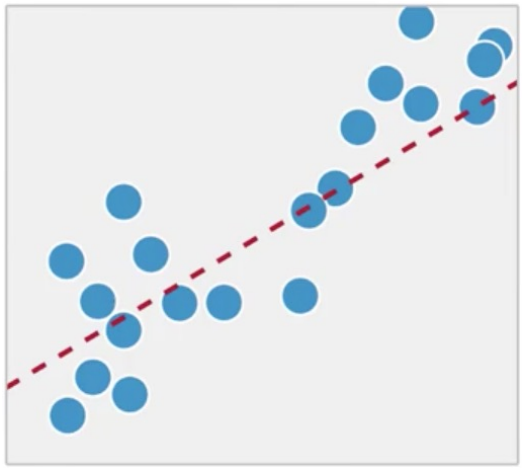
Expression level
"0.01"

Real-valued output

Continuous Labels

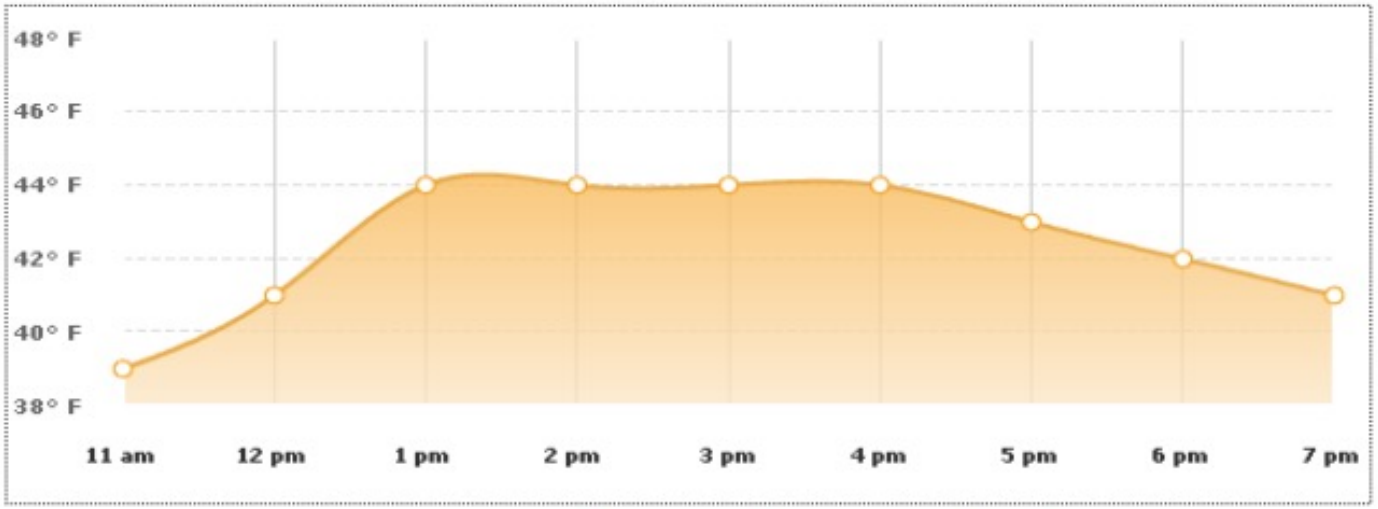
Supervised Learning - Regression

	ENGINESIZE	CYLINDERS	FUELCONSUMPTION_COMB	CO2EMISSIONS
0	2.0	4	8.5	196
1	2.4	4	9.6	221
2	1.5	4	5.9	136
3	3.5	6	11.1	255
4	3.5	6	10.6	244
5	3.5	6	10.0	230
6	3.5	6	10.1	232
7	3.7	6	11.1	255
8	3.7	6	11.6	267
9	2.4	4	9.2	?



Supervised Learning problems

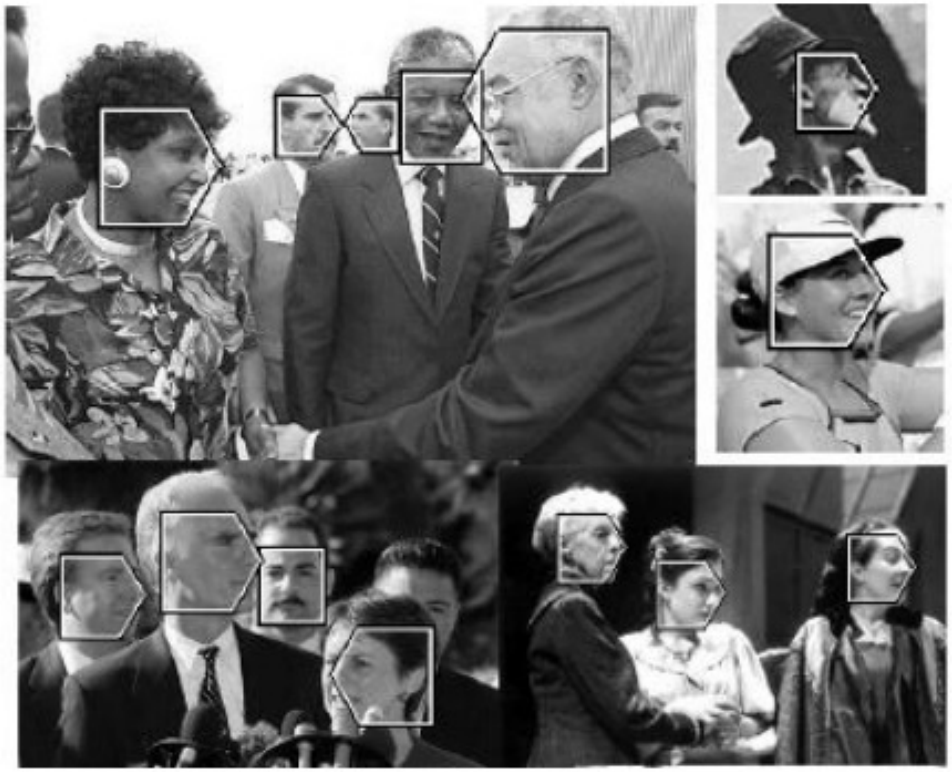
Features? **Labels?** **Classification/Regression?**



Temperature/Weather prediction

Supervised Learning problems

Features? **Labels?** **Classification/Regression?**



Face Detection

Supervised Learning problems

Features? **Labels?** **Classification/Regression?**

Robotic Control

Canonical Learning Tasks

We will focus on two fundamental and popular learning paradigms:

- Supervised Learning $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^n$
  1. Classification (Binary or Multi-class) : **Discrete** output
  2. Regression : **Real-valued** output

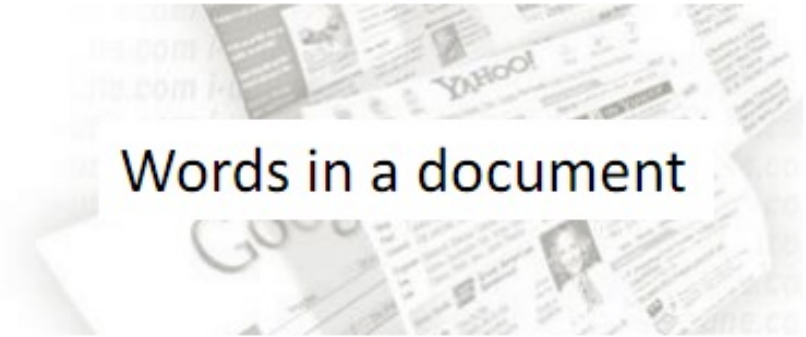
- Unsupervised Learning $\mathcal{D} = \{\mathbf{x}_i\}_{i=1}^n$
  1. Density estimation,
  2. Clustering,
  3. Dimensionality reduction

- Many other setups exist (not in the scope of this course)

Unsupervised Learning

Aka "learning without a teacher"

Feature Space $\mathcal{X}$



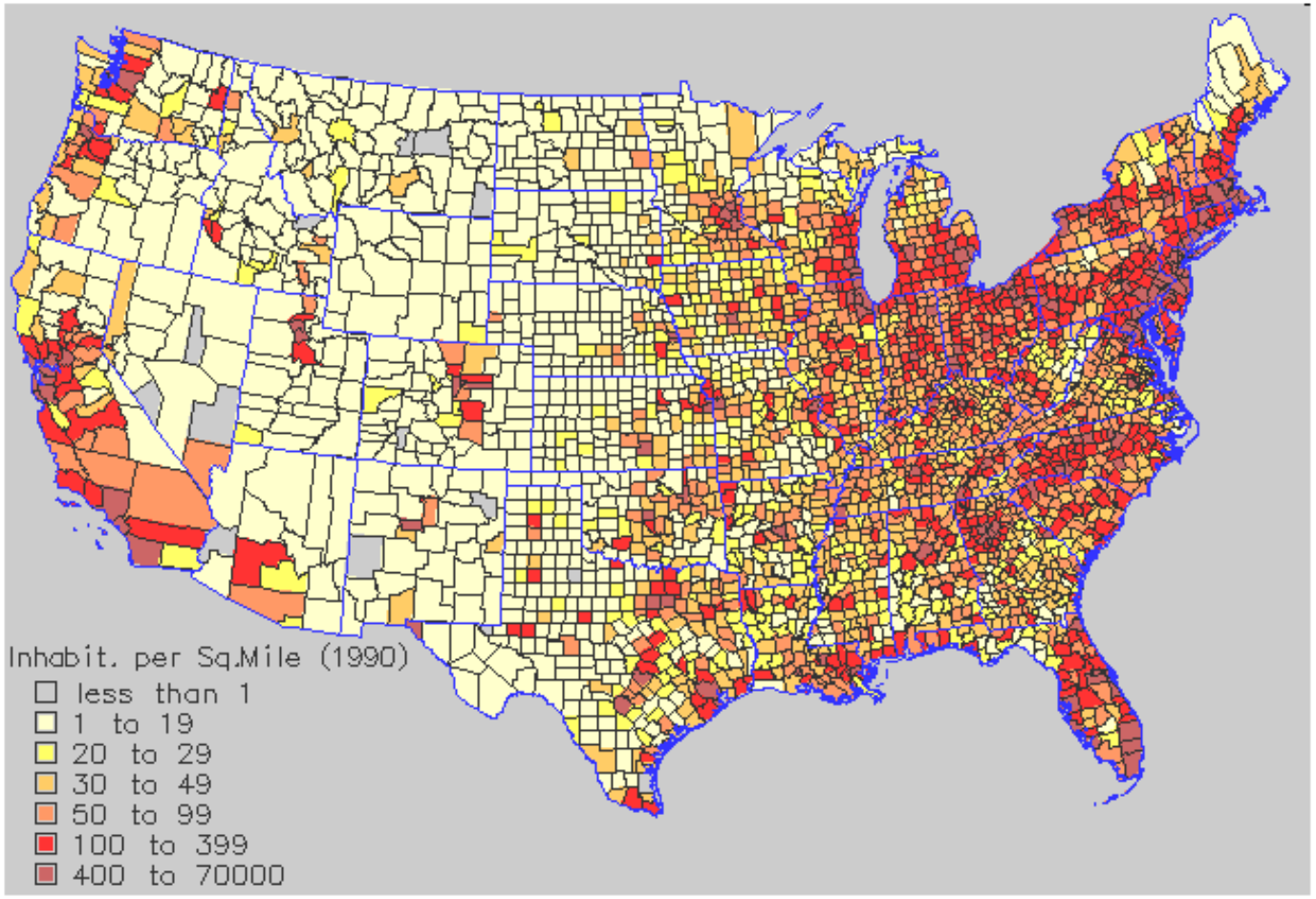
Word distribution
(Probability of a word)

Density estimation

Task: Given $X \in \mathcal{X}$, learn $f(X)$.

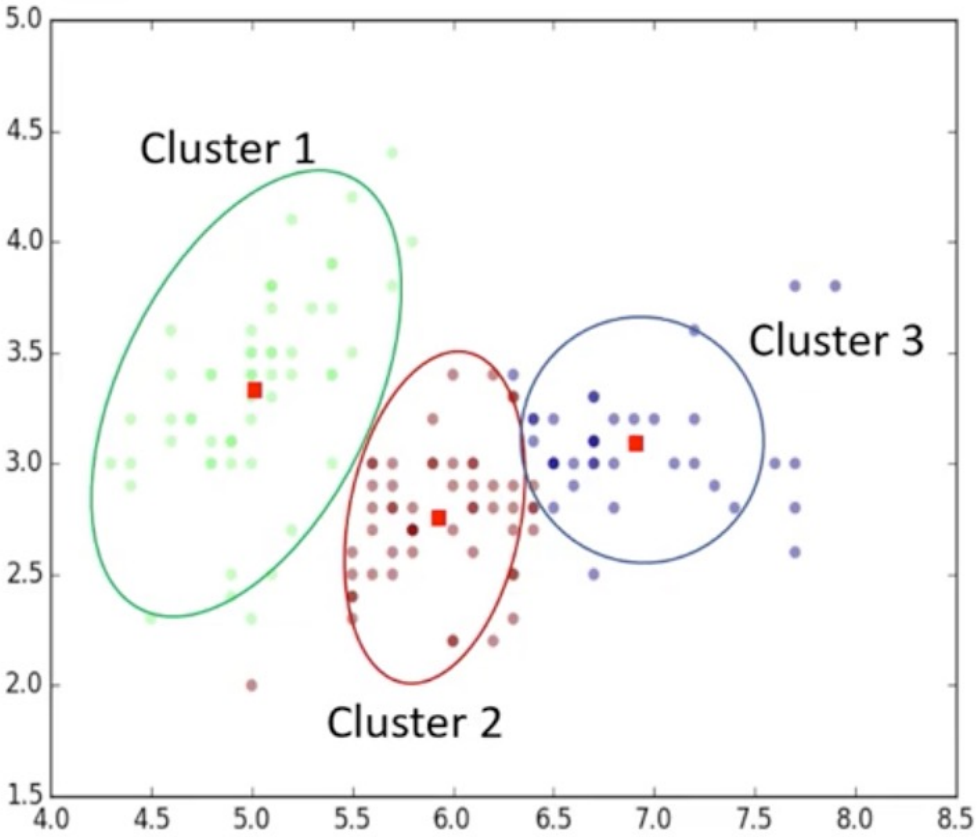
Unsupervised Learning- Density Estimation

Population density



Unsupervised Learning- Clustering

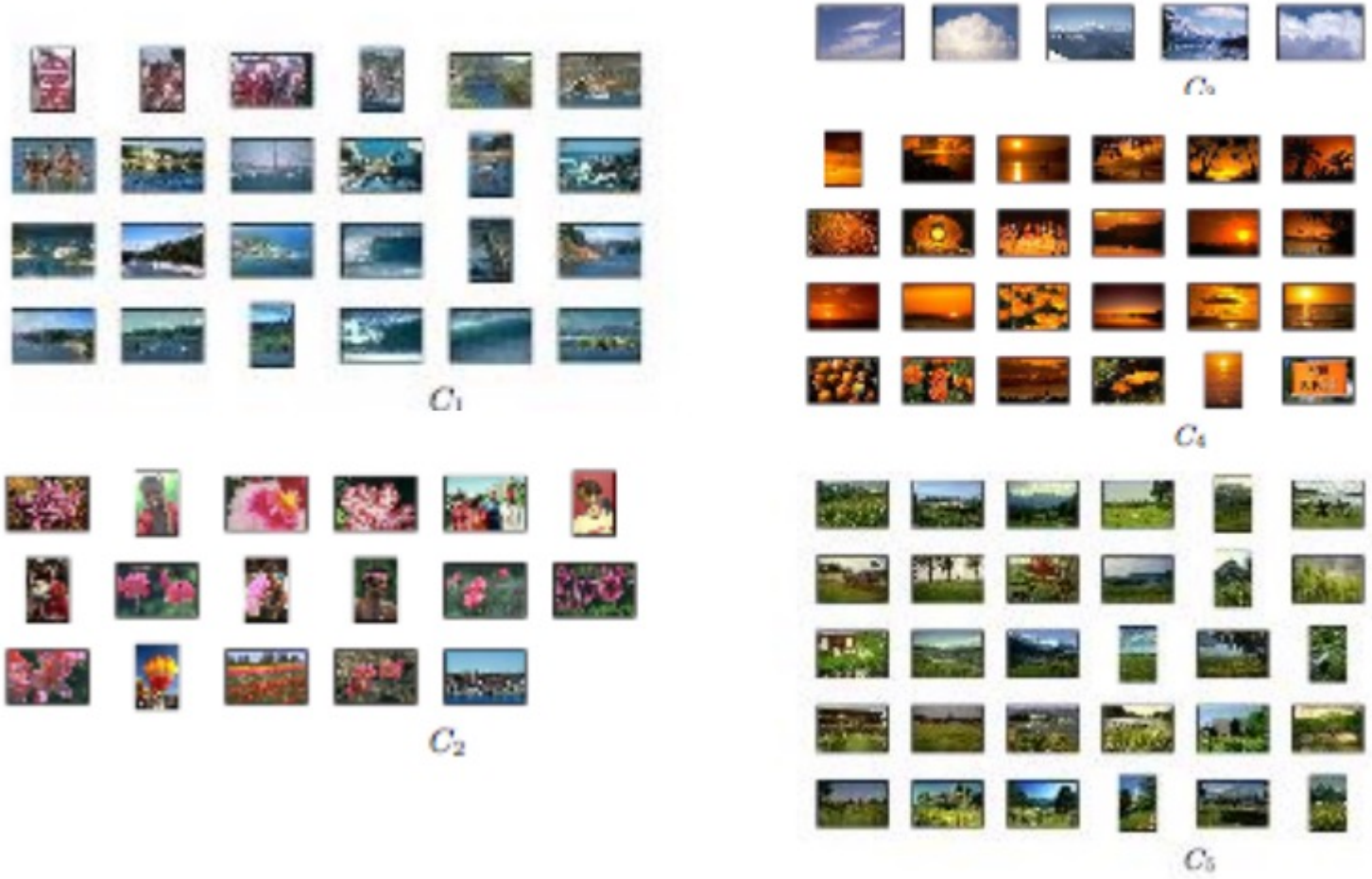
- Grouping data instances that are similar, for
  - Discovering structure
  - Data summarization
  - Anomaly detection



Unsupervised Learning- Clustering

Group similar things e.g. images

[Goldberger et al.]

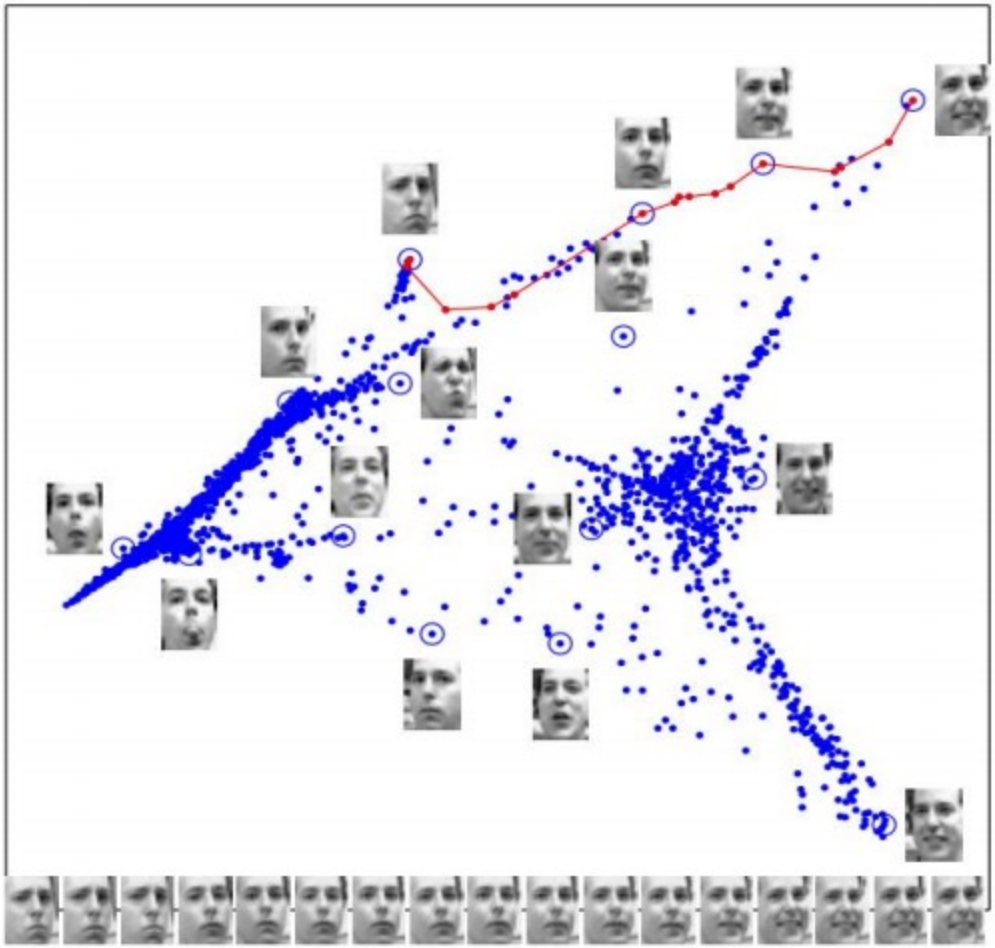


Unsupervised Learning - Embedding

- Dimensionality reduction: Finding a small number of dimensions that capture latent structure

Images have thousands or millions of pixels.

Can we give each image a coordinate,
such that similar images are near each other?



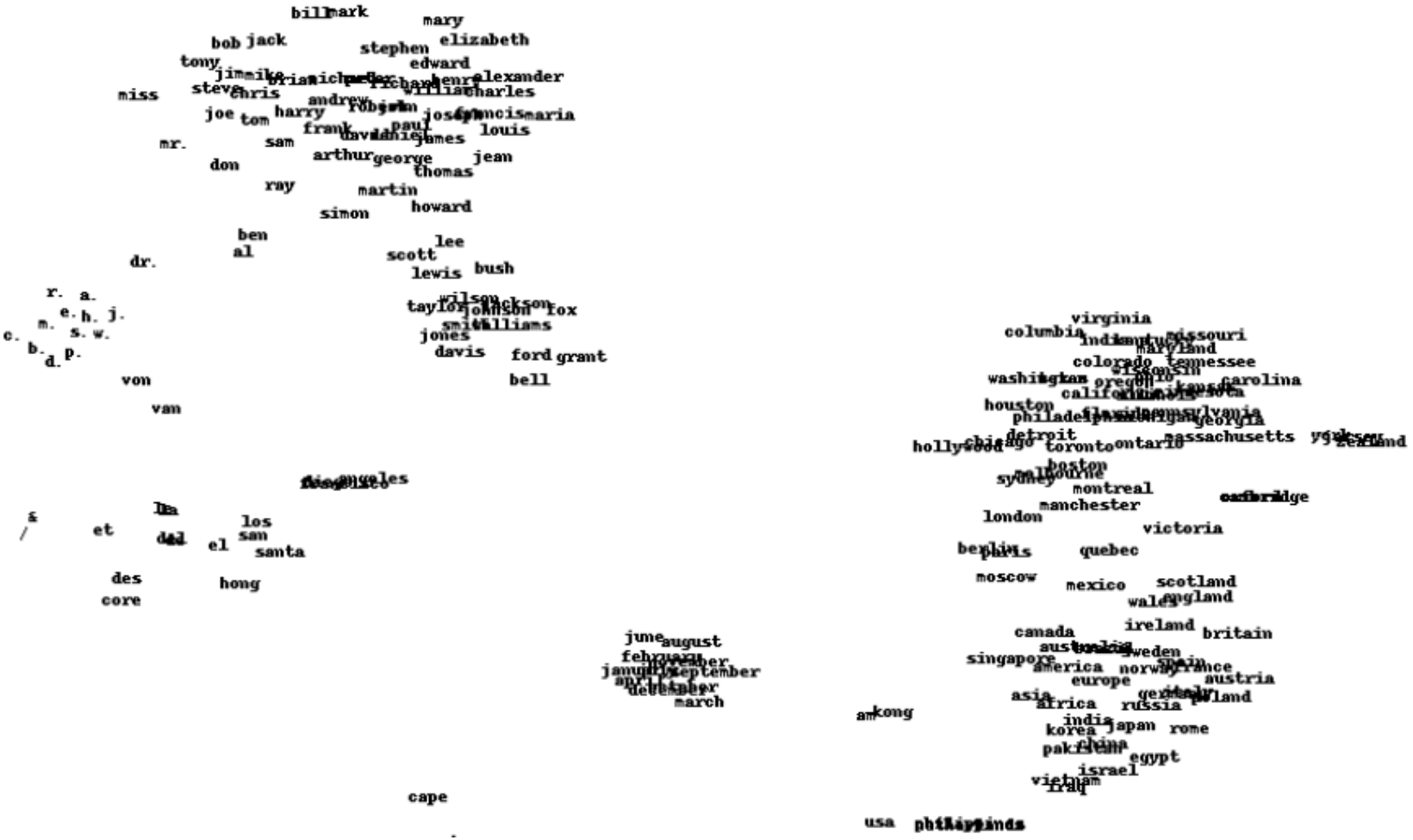
Unsupervised Learning - Embedding

Dimensionality Reduction - words [Joseph Turian]

Unsupervised Learning - Embedding

Dimensionality Reduction - words

[Joseph Turian]



ML tasks: enterprise example

- Classification and Class-Probability Estimation
  - How likely is this consumer to respond to campaign?
    Prob(respond=yes | customer and campaign features)
- Regression
  - How much will she use the service?
- Clustering
  - Do my customers form natural groups?
- Dimensionality Reduction
  - Which latent dimensions describe consumer taste/ preferences?

Exercises

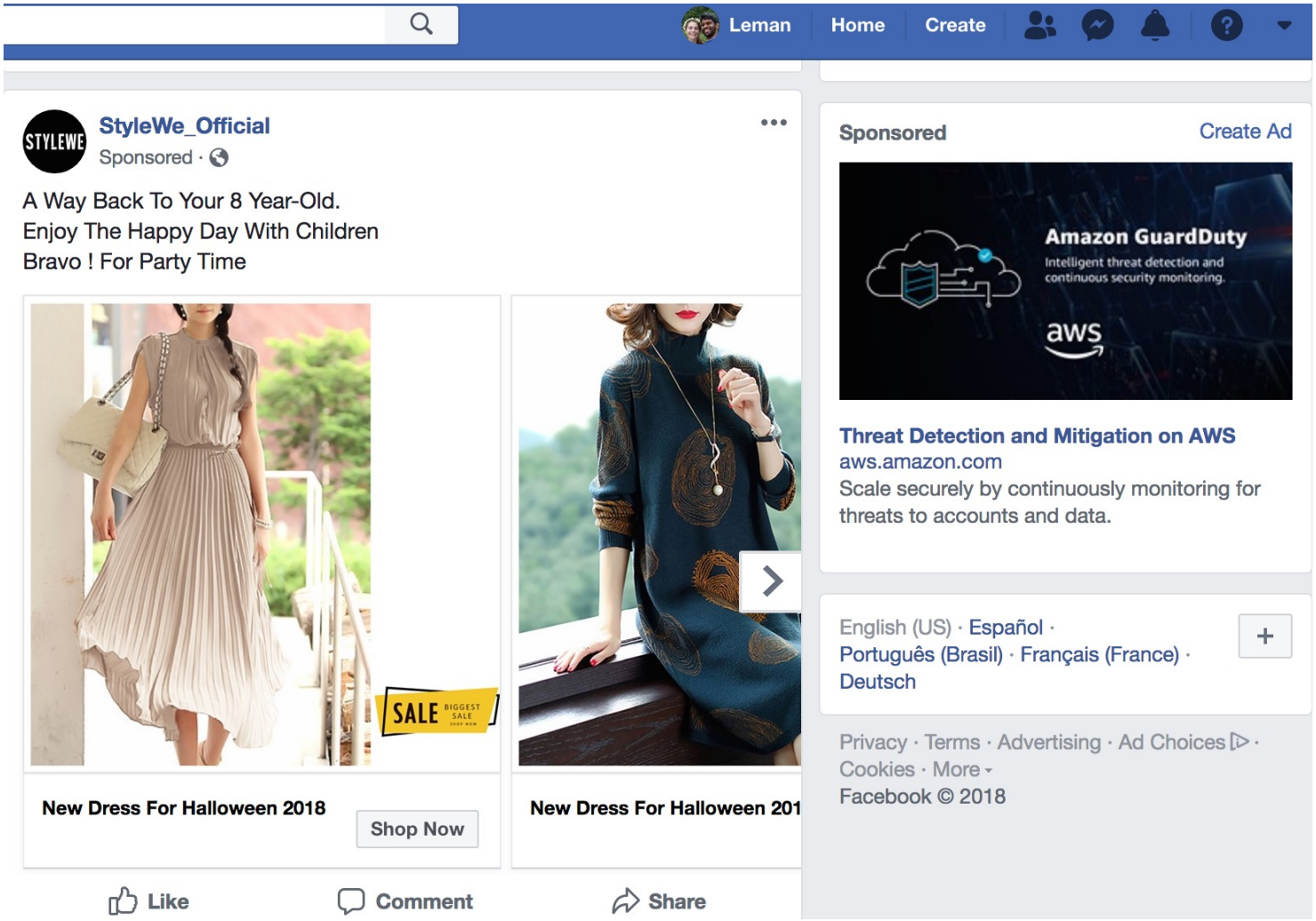
- Let's express each of the following tasks within the formalization of learning, specifying the **type** of learning task, **feature** space $\mathcal{X}$, and **output** space $\mathcal{Y}$ (if any):
  - Organizing books on shelves based on subject
  - Identifying topics in a set of blog posts
  - Segmenting grocery store customers by purchase behavior
  - Identifying zipcode from handwritten digits on an envelope
  - Detecting fraudulent credit card transactions
  - Predicting the electric usage at a household
  - Diagnosing a patient that walks in with certain symptoms
  - Recognizing facial expressions
  - Determining the credit limit for a bank customer
  - Categorizing courses at a university into different types
  - Recommending a song to a user in an online music store
  - Estimating the probability of word 'lottery' in spam emails

Agenda

- Components of learning
  - Formalization
- Types of learning
- ML applications in the real world
- What does learning mean?
  - Overfitting and Generalization

ML Applications

- Ad placement



ML Applications

- Face recognition and tagging

ML Applications

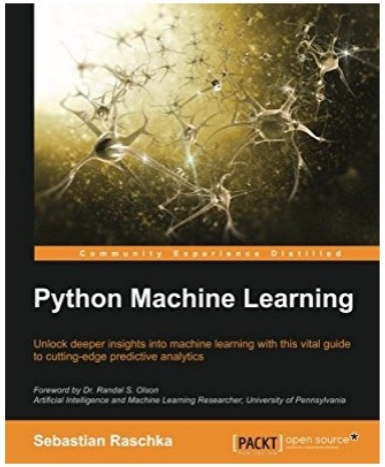
- Pose recognition

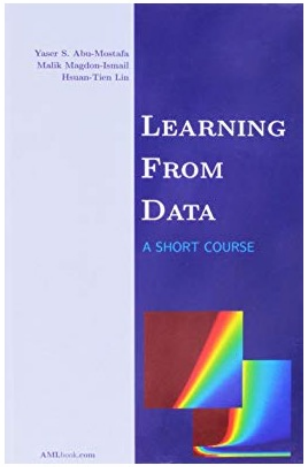
ML Applications

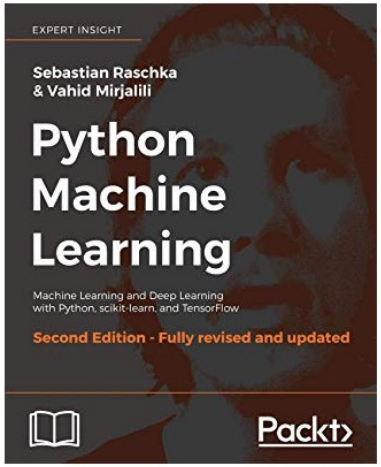
- Product recommendation

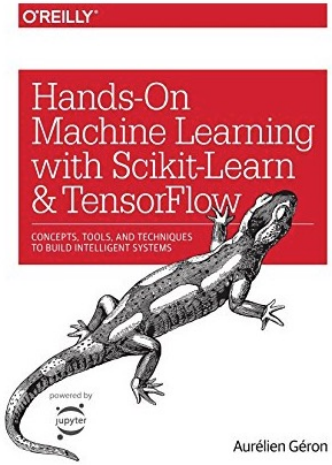
amazon Try Prime

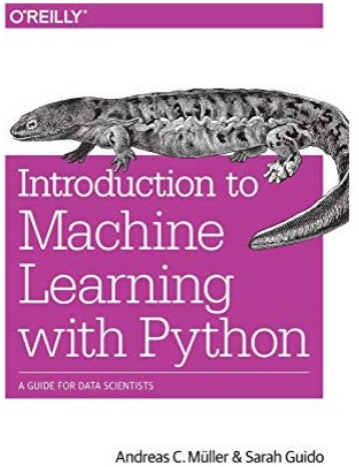
Related to items you've viewed See more





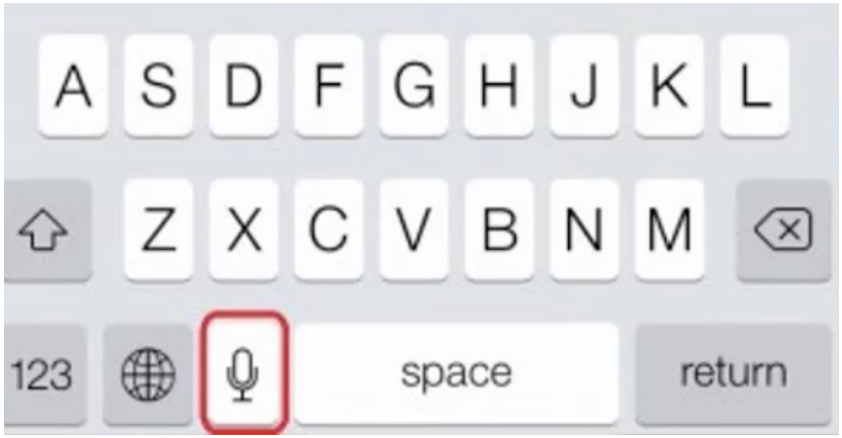






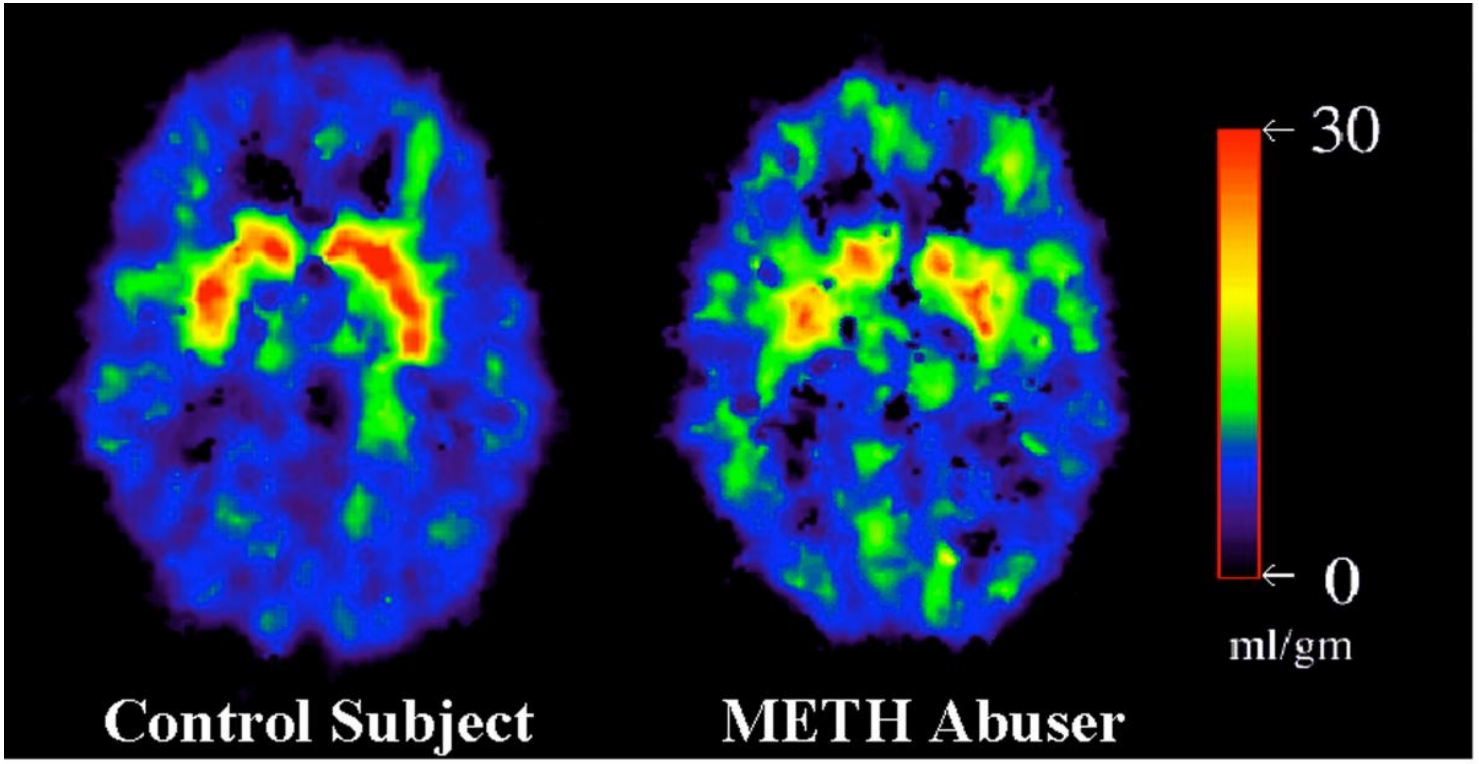
ML Applications

- Speech to text



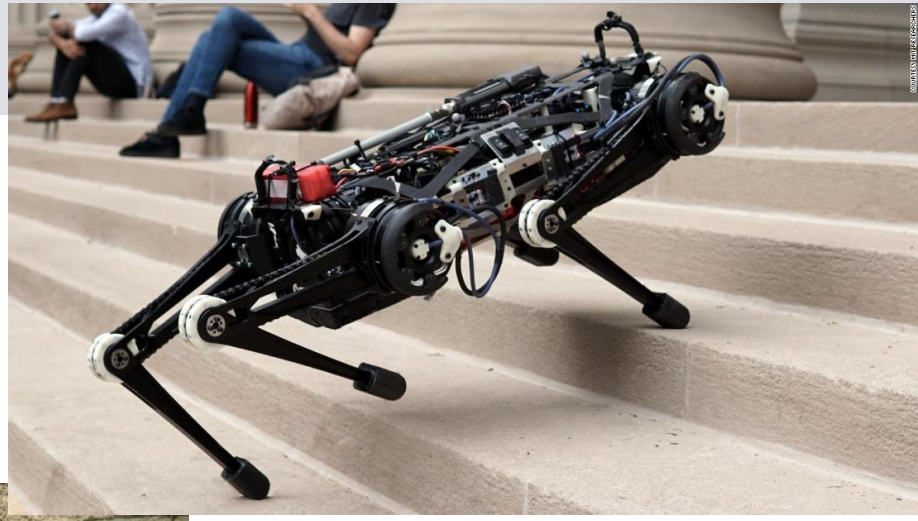
ML Applications

- Recognizing addiction, tumor, etc.



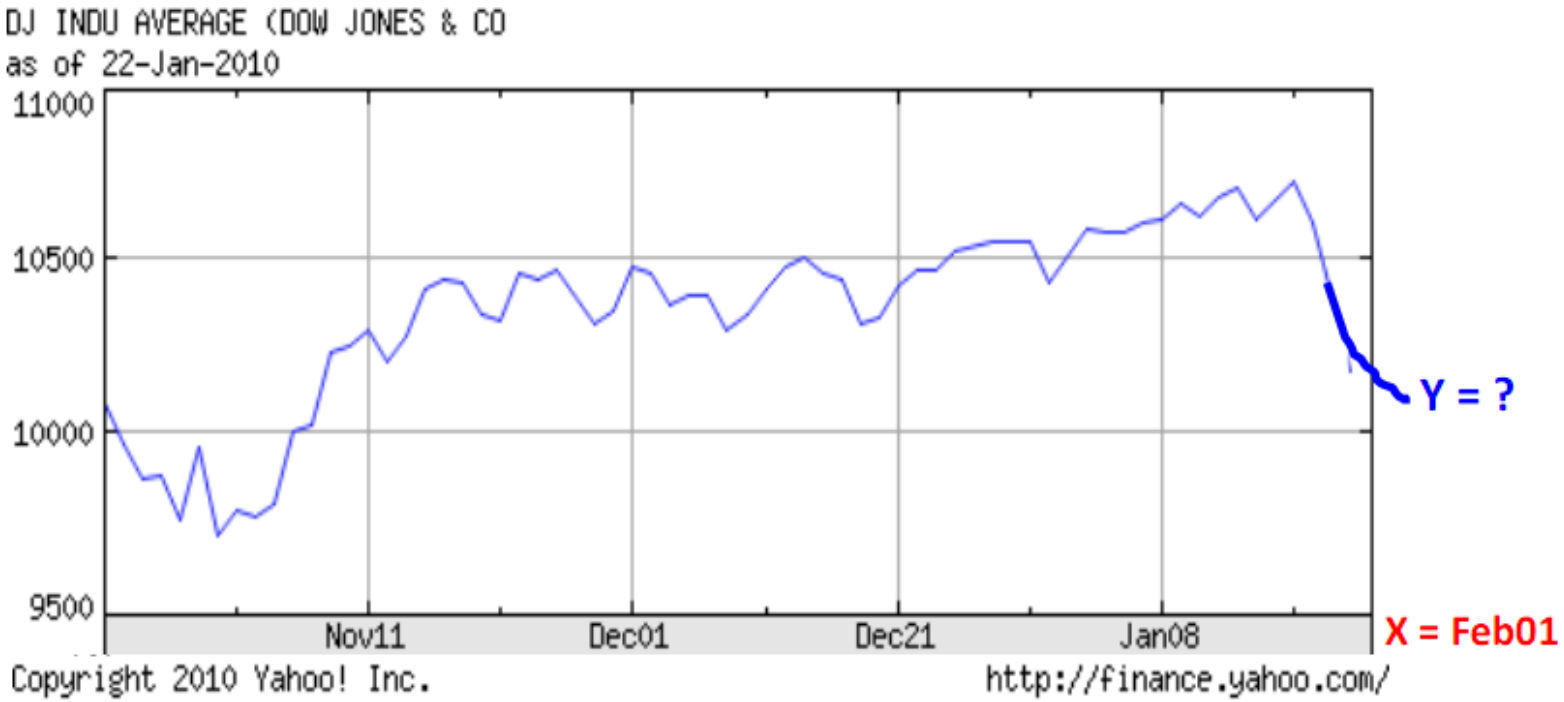
ML Applications

- Autonomous robots →
- Self driving cars:



ML Applications

- Stock Market Prediction



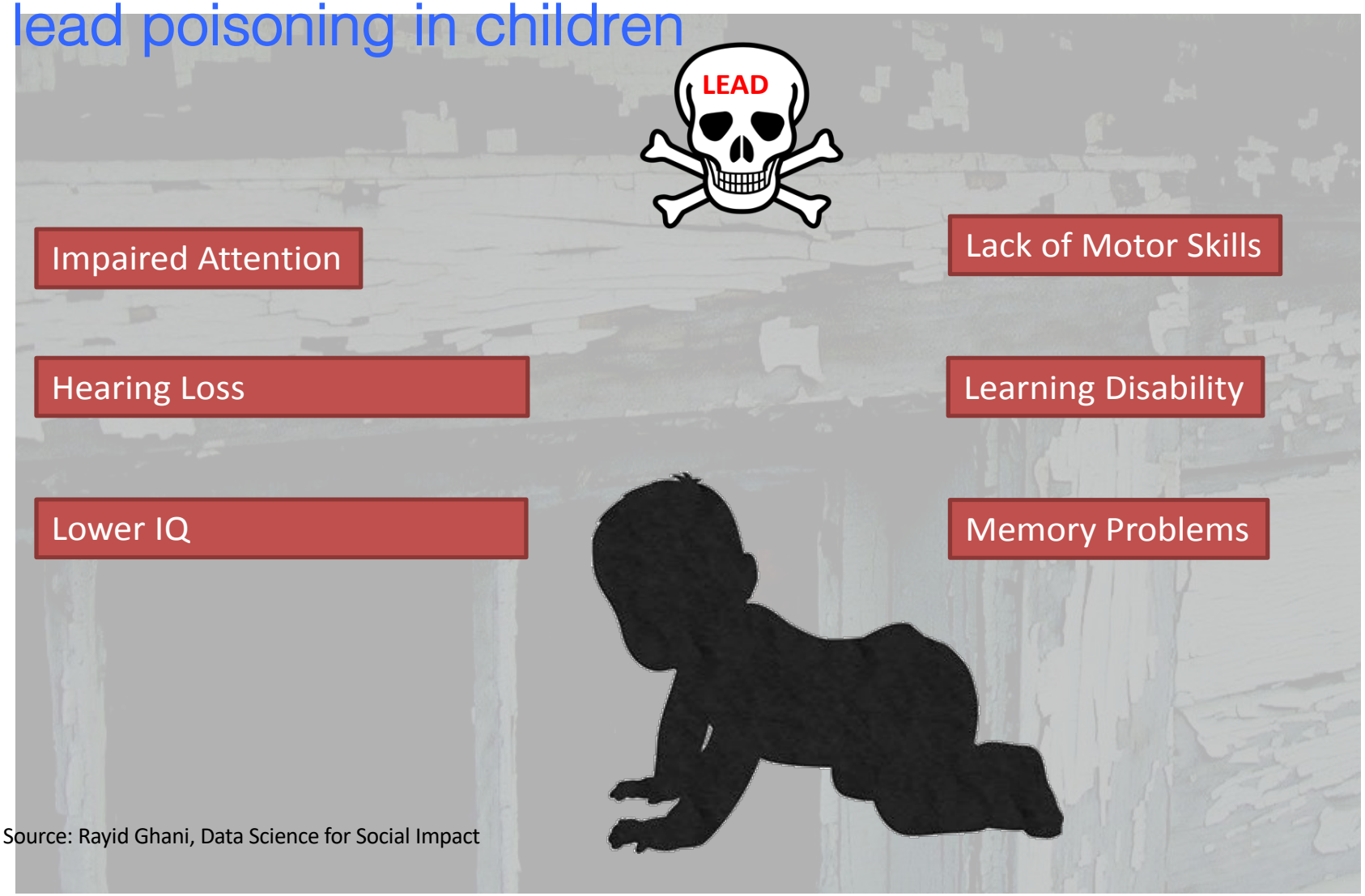
ML Applications

- Predicting Police Conflict
  - identify officers (and dispatches) at risk of adverse interactions with the public

Source: Rayid Ghani, Data Science for Social Impact

ML Applications

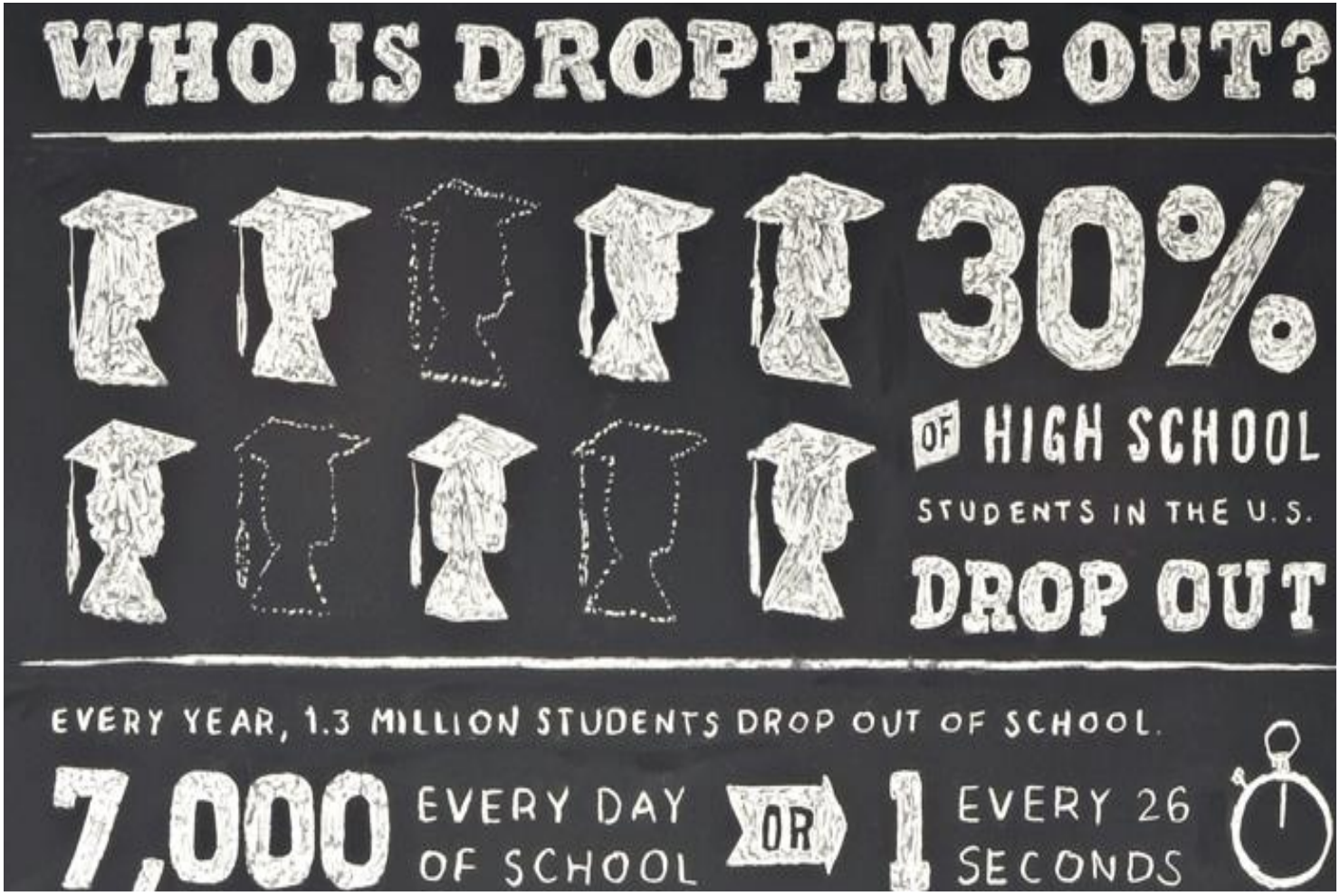
- Preventative lead inspections and reducing lead poisoning in children



Source: Rayid Ghani, Data Science for Social Impact

ML Applications

- Identifying students in need of extra support to achieve educational outcomes



Source: Rayid Ghani, Data Science for Social Impact

ML Applications

- Blight prediction for improving home inspection processes and early intervention

Source: Rayid Ghani, Data Science for Social Impact

Agenda

- Components of learning
  - Formalization
- Types of learning
- ML applications in the real world
- What does learning mean?
  - Overfitting and Generalization

What does it mean to learn?

- To start, we must know what learning means, and how to determine success (or failure).

- Example scenario: Bob just started taking MLPS.
  - At the end, we expect him to have "learned" the topic
  - How to gauge whether or not he did?
  - Alice, the teacher, can give him a test: if Bob answers questions correctly, he has done well on learning ML.

What does it mean to learn?

Example scenario (continued): We need to test for learning, but what makes a reasonable test?

- unfair to give a test on "History of Pottery" ☺
  - ➢ No prior experience with it, not be representative of his learning
  - → There should be a strong relationship between the data that learning system sees at **training** time and the data it sees at **test** time. Formally, they should come from same **data generating distribution**

What does it mean to learn?

Example scenario (continued): We need to test for learning, but what makes a reasonable test?

- unfair to give a test on "History of Pottery" ☺
  - ➢ No prior experience with it, not be representative of his learning
  - → There should be a strong relationship between the data that learning system sees at **training** time and the data it sees at **test** time. Formally, they should come from same **data generating distribution**
- bad if asks exact questions Alice answered in class
  - ➢ we would expect Bob give correct answers, but this would not demonstrate that he has learned—he would simply be recalling his past experience
  - → performance of the learning system should be measured on **unseen test** data

What does it mean to learn?

- Rules of thumb:
  - Training and test data should come from same data generating distribution
    - Unfair to give an unrelated test on non-ML subject
  - Learning should be measured on unseen test data
    - Memorizing the training data not OK.

What does it mean to learn?

- Rules of thumb:
  - Training and test data should come from same data generating distribution
    - Unfair to give an unrelated test on non-ML subject
  - Learning should be measured on unseen test data
    - Memorizing the training data not OK.

- What is desired:
  - Bob (the ML system) observes specific examples, and
  - Then answers related, but new questions correctly

  → This tests whether Bob (the system) has the ability to **generalize**. Generalization is perhaps the most central concept in machine learning.

Memorization vs. Generalization

- Consider the following regression prediction function g:

$$g(\mathbf{x}) = \begin{cases} y_i, & \text{if } \mathbf{x} = \mathbf{x}_i \text{ for } i = 1, \ldots, n \\ \text{any random value}, & \text{otherwise} \end{cases}$$

  - It would do perfect on training examples
  - But have very large **generalization error (!)** on unseen examples
  - Because *g* has not learned to generalize, but rather memorized (overfit to) the training dataset

Memorization vs. Generalization

- Consider the following classification scenario:
  - 8 training examples:

Class	Outlook	Temperature	Windy?
Play	Sunny	Low	Yes
No play	Sunny	High	Yes
No play	Sunny	High	No
Play	Overcast	Low	Yes
Play	Overcast	High	No
Play	Overcast	Low	No
No play	Rainy	Low	Yes
Play	Rainy	Low	No

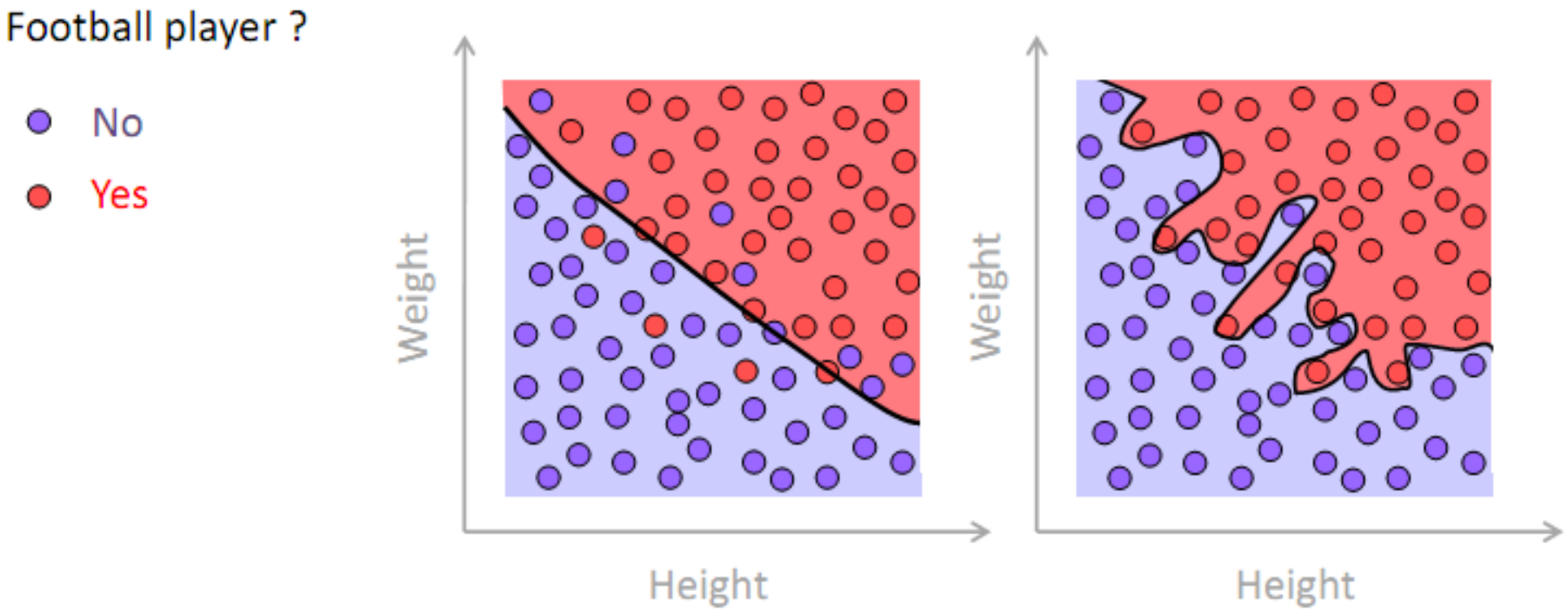
  - Test example:

Class	Outlook	Temperature	Windy?
???	Sunny	Low	No

  - Requires us to **generalize** beyond training data

Overfitting vs. Generalization

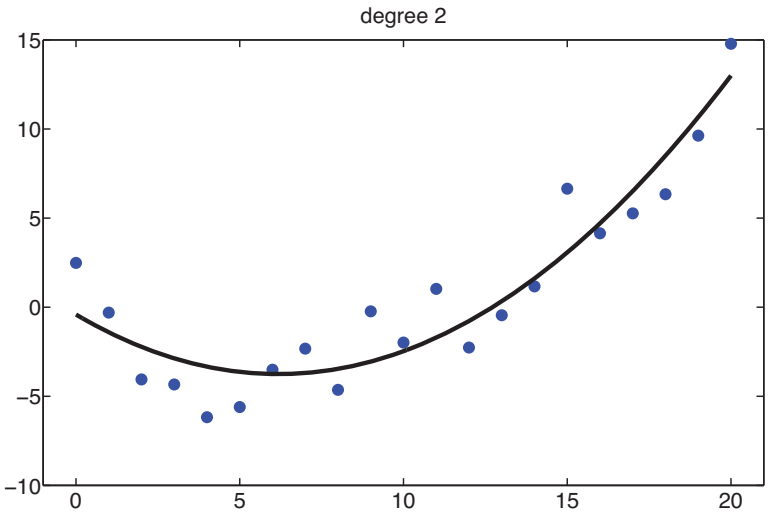
- In general, highly flexible models tend to overfit
- Classification example: (right) model overfit,
  (left) model "just right"

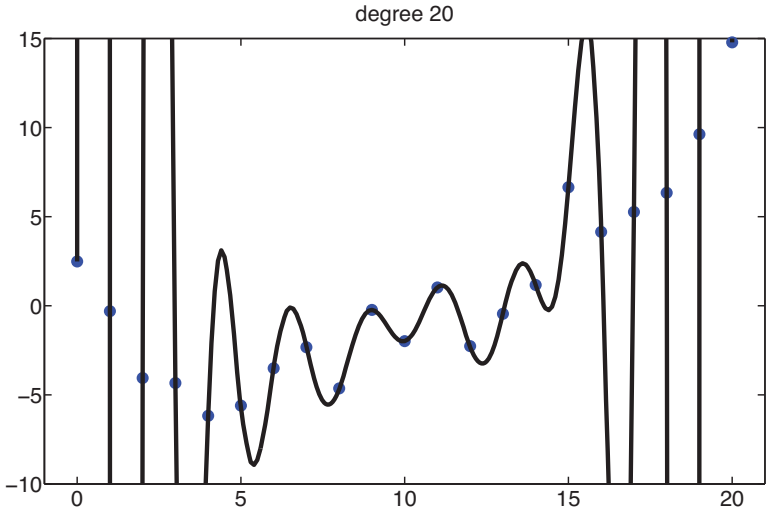


(a) Binary classification

Overfitting vs. Generalization

- In general, highly flexible models tend to overfit
- Regression example: (right) model overfit,
  (left) model "just right"





(b) Polynomial regression: poly-fits of degrees (left) 2 and (right) 20 onto 21 training points. (Based on Figure 1.18 from Kevin Murphy, Machine Learning, 2012)

Regularization and Model Selection to the rescue against Overfitting

- Trying to model every minor variation in the input, that is more likely to be noise than true signal, yields **overfitting**.
- **Model selection** is the task of carefully choosing among models with varying degrees of flexibility.
- We will learn the best practices including **regularization** and **cross validation** for selecting the models that are "just right".