



CentraleSupélec

# Machine Learning

## M.Sc. in DSBA & M.Sc. in AI

### Lecture 1

#### Introduction

Fragkiskos Malliaros

Thursday, October 4, 2018

# About Me

- Undergrad at the University of Patras, Greece
- Ph.D. in CS at Ecole Polytechnique, Paris
- Postdoc researcher at UC San Diego
- Assistant Professor at CentraleSupélec (since Oct. 2017)

**Research interests:** Data science, ML, graph mining, text mining and NLP

# The Team



Instructor: **Fragkiskos Malliaros**

**Office:** Bâtiment Bouygues, CVN Lab, Room SC.217

**Office hours:** I will be available right after the lecture

Or, send me an email and we will find a good time to meet

**Email:** fragkiskos.me@gmail.com



TA: **Kimya Dhade**

**Email:** b00718577@essec.edu

# Acknowledgements

- The lecture is partially based on material by
  - Richard Zemel, Raquel Urtasun and Sanja Fidler (University of Toronto)
  - Chloé-Agathe Azencott (Mines ParisTech)
  - Julian McAuley (UC San Diego)
  - Dimitris Papailiopoulos (UW-Madison)

Thank you!

# Slides of Today's Lecture

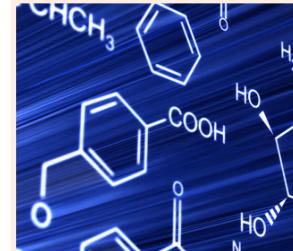
<http://fragkiskos.me/introduction.pdf>

# Why Machine Learning?

# Automation and Robotics



# Drug Discovery and Healthcare



# Intelligent Personal Assistants



# Recommender Systems

#### **Recommendations for You, Dimt**



# Automation and Robotics



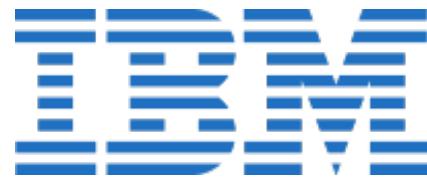
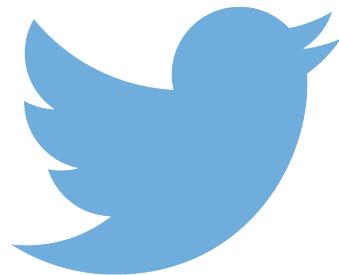
# Real and important problems



## Recommender Systems

Recommendations for You, Dimitris





LinkedIn®

What is LinkedIn? Join Today Sign In

Machine Learning



United States



Find jobs

8,184 Machine Learning jobs in United States

Get alerts for this search  
We'll email you new jobs as they  
become available

LinkedIn®

What is LinkedIn? Join Today Sign In

Data Scientist



United States



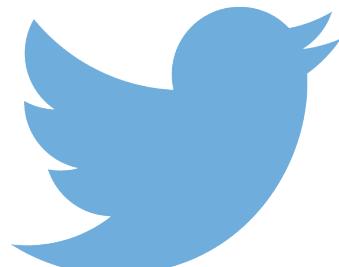
Find jobs

30,669 Data Scientist jobs in United States

Get alerts for this search  
We'll email you new jobs as they  
become available



facebook



# Job market

8,184 Machine Learning jobs in United States

Get alerts for this search  
We'll email you new jobs as they  
become available

LinkedIn®

What is LinkedIn? Join Today Sign In

Data Scientist



United States



Find jobs

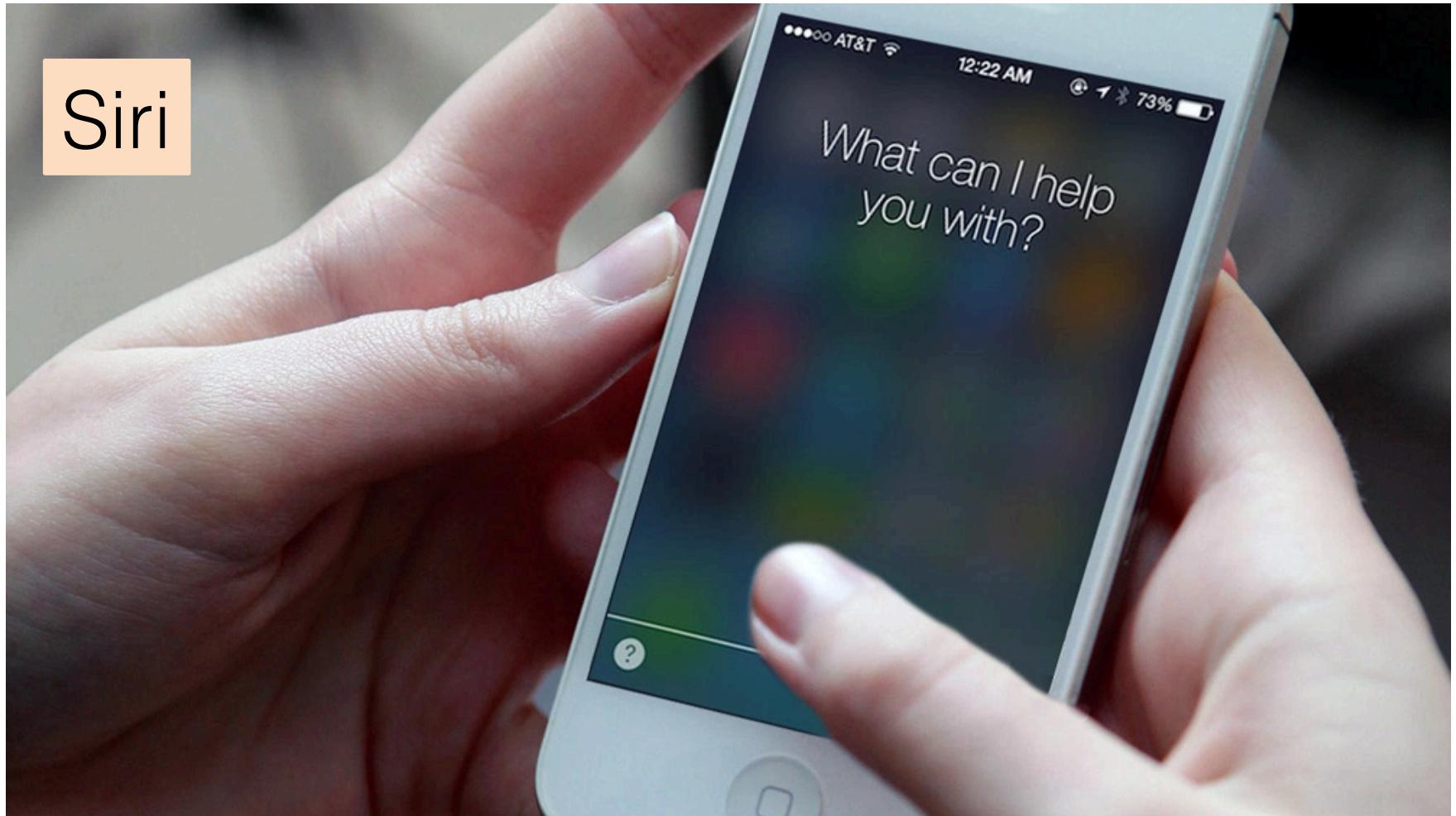
30,669 Data Scientist jobs in United States

Get alerts for this search  
We'll email you new jobs as they  
become available

# Machine Learning is Almost Everywhere

- Recognizing patterns: Speech Recognition, facial identity, etc

# Speech Recognition



Source: <http://bgr.com/2017/01/24/iphone-8-enhanced-siri-upgrade/>

# Machine Learning is Almost Everywhere

- Recognizing patterns: Speech Recognition, facial identity, etc
- Recommender Systems: Noisy data, commercial pay-off  
(e.g., Amazon, Netflix)

# Example of a Recommendation System

NETFLIX | Your Account & Help

Movies, TV shows, actors, directors, genres

Watch Instantly | Browse DVDs | Your Queue | **Movies You'll ❤**

**Congratulations!** Movies we think **You** will ❤

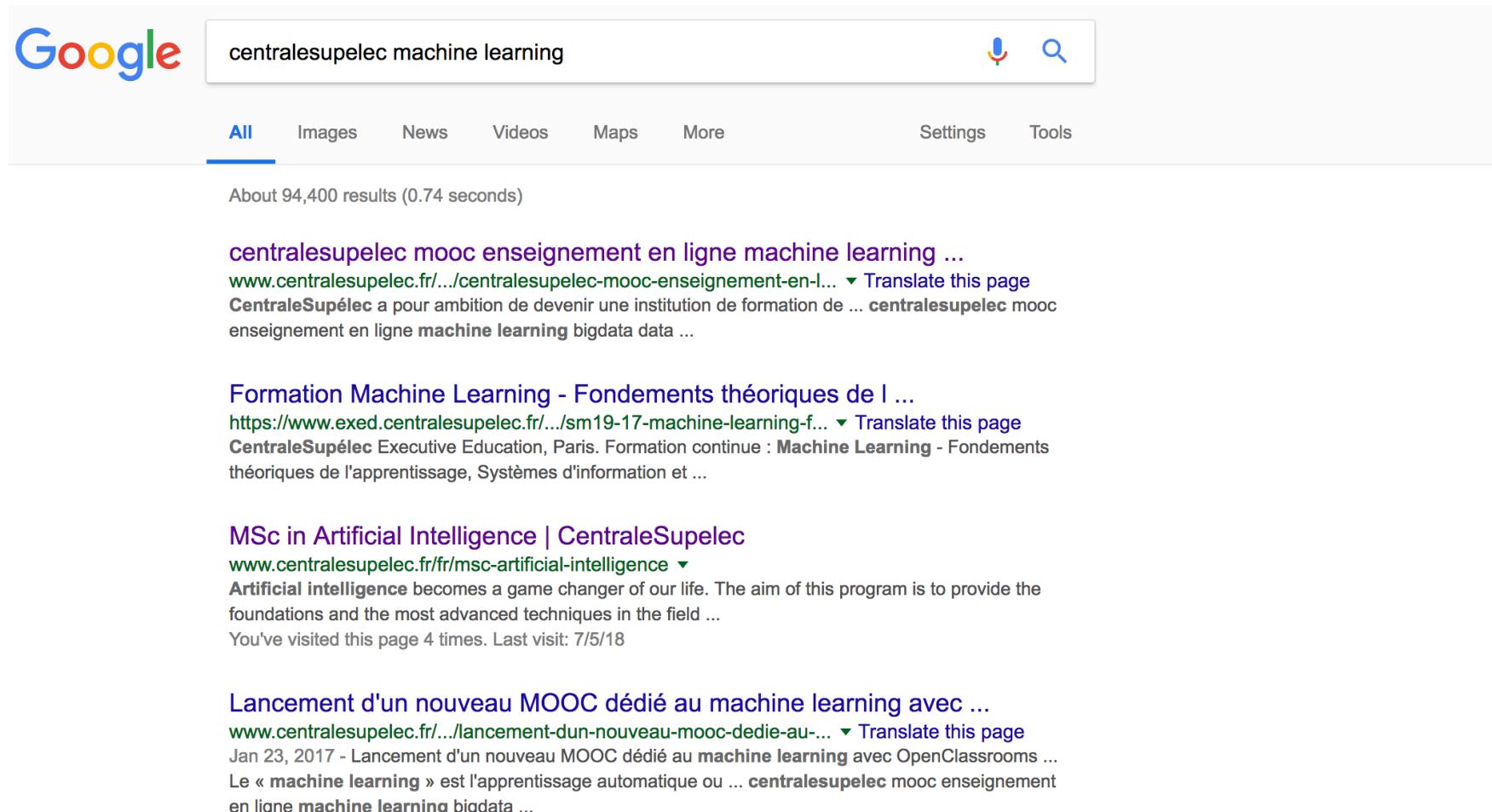
Add movies to your Queue, or Rate ones you've seen for even better suggestions.

Spider-Man 3 <input type="button" value="Add"/> 	300 <input type="button" value="Add"/> 	The Rundown <input type="button" value="Add"/> 	Bad Boys II <input type="button" value="Add"/> 
Las Vegas: Season 2 (6-Disc Series) 	The Last Samurai 	Star Wars: Episode III 	Robot Chicken: Season 3 (2-Disc Series) 

# Machine Learning is Almost Everywhere

- Recognizing patterns: Speech Recognition, facial identity, etc
- Recommender Systems: Noisy data, commercial pay-off  
(e.g., Amazon, Netflix)
- Information retrieval: Find documents or images with similar content

# Information Retrieval (1/2)



A screenshot of a Google search results page. The search query is "centralesupelec machine learning". The results are filtered by "All" and show approximately 94,400 results in 0.74 seconds. The first result is a link to a Mooc page titled "centralesupelec mooc enseignement en ligne machine learning ...". The second result is a link to a formation page titled "Formation Machine Learning - Fondements théoriques de l ...". The third result is a link to an MSc program page titled "MSc in Artificial Intelligence | CentraleSupélec". The fourth result is a news article titled "Lancement d'un nouveau MOOC dédié au machine learning avec ...".

Google

centralesupelec machine learning

All Images News Videos Maps More Settings Tools

About 94,400 results (0.74 seconds)

**centralesupelec mooc enseignement en ligne machine learning ...**  
[www.centralesupelec.fr/.../centralesupelec-mooc-enseignement-en-l...](http://www.centralesupelec.fr/.../centralesupelec-mooc-enseignement-en-l...) ▾ Translate this page  
CentraleSupélec a pour ambition de devenir une institution de formation de ... centralesupelec mooc enseignement en ligne machine learning bigdata data ...

**Formation Machine Learning - Fondements théoriques de l ...**  
<https://www.exed.centralesupelec.fr/.../sm19-17-machine-learning-f...> ▾ Translate this page  
CentraleSupélec Executive Education, Paris. Formation continue : Machine Learning - Fondements théoriques de l'apprentissage, Systèmes d'information et ...

**MSc in Artificial Intelligence | CentraleSupélec**  
[www.centralesupelec.fr/fr/msc-artificial-intelligence](http://www.centralesupelec.fr/fr/msc-artificial-intelligence) ▾  
Artificial intelligence becomes a game changer of our life. The aim of this program is to provide the foundations and the most advanced techniques in the field ...  
You've visited this page 4 times. Last visit: 7/5/18

**Lancement d'un nouveau MOOC dédié au machine learning avec ...**  
[www.centralesupelec.fr/.../lancement-dun-nouveau-mooc-dedie-au-...](http://www.centralesupelec.fr/.../lancement-dun-nouveau-mooc-dedie-au-...) ▾ Translate this page  
Jan 23, 2017 - Lancement d'un nouveau MOOC dédié au machine learning avec OpenClassrooms ...  
Le « machine learning » est l'apprentissage automatique ou ... centralesupelec mooc enseignement en ligne machine learning bigdata ...

# Information Retrieval (2/2)

Google Machine Learning

All News Videos **Images** Books More Settings Tools View saved SafeSearch ▾

data science artificial intelligence big data iot analytic scalable cloud self training statistical deep learning supervised neural bayes

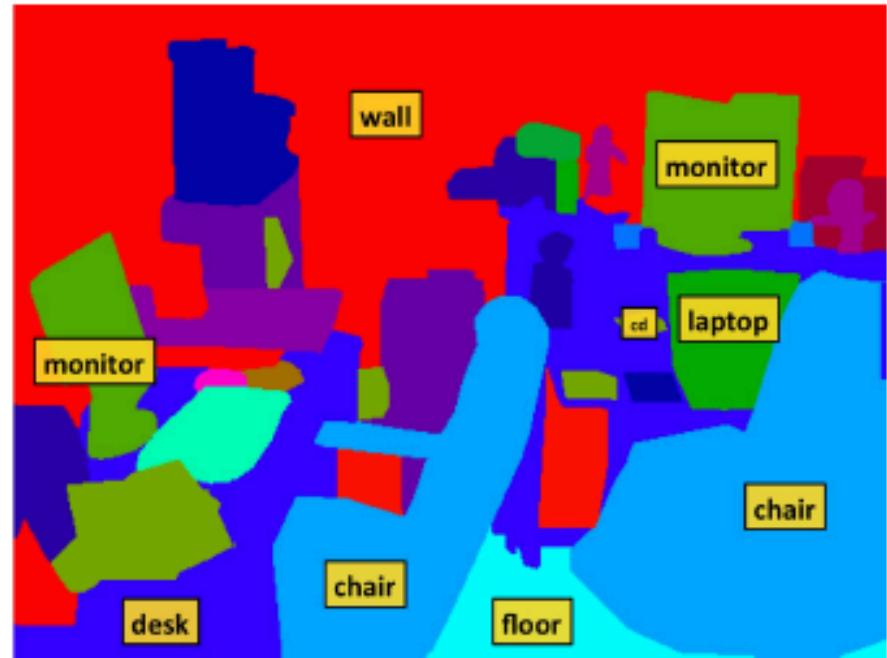
MACHINE LEARNING

The search results show eight images arranged in two rows of four. The first row includes: a brain composed of nodes and connections; a complex network diagram with various icons like clouds, databases, and code snippets; a white humanoid robot holding a tablet; and a hand interacting with a glowing neural network. The second row includes: a circular diagram showing the machine learning process (Preprocess data, Train model, Apply model, Capture feedback); a network graph with nodes labeled 'MACHINE LEARNING'; a glowing brain circuit board against a black background; and a profile of a head containing various icons representing different fields of knowledge.

# Machine Learning is Almost Everywhere

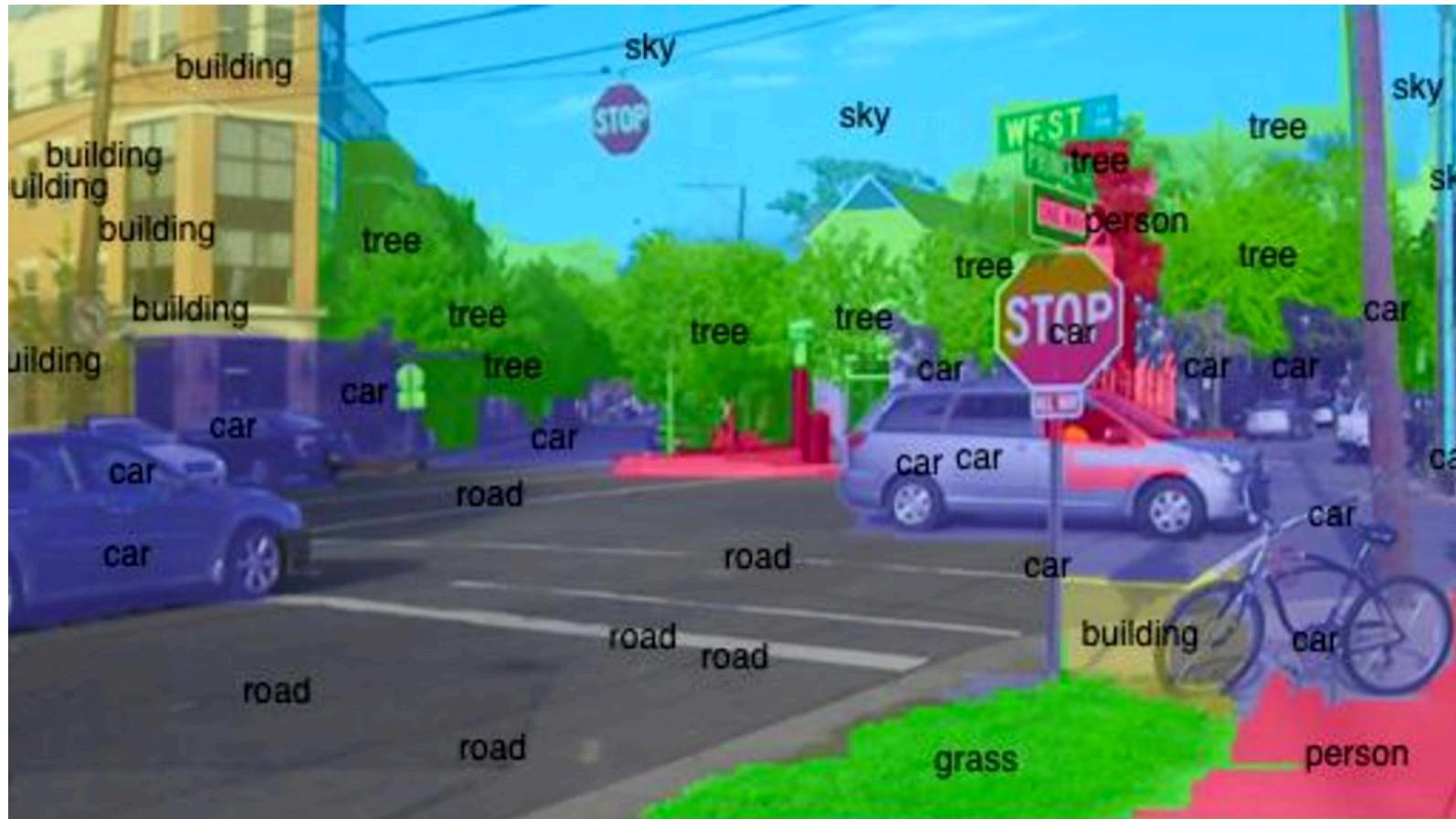
- Recognizing patterns: Speech Recognition, facial identity, etc
- Recommender Systems: Noisy data, commercial pay-off  
(e.g., Amazon, Netflix)
- Information retrieval: Find documents or images with similar content
- Computer vision: detection, segmentation, depth estimation, optical flow, etc.

# Computer Vision (1/3)



Source: R. Urtusan, U. of Toronto

# Computer Vision (2/3)



Source: <https://techcrunch.com/2016/11/13/wtf-is-computer-vision/>

# Computer Vision (3/3)



[Gatys, Ecker, Bethge. A Neural Algorithm of Artistic Style. Arxiv'15.]

# Machine Learning is Almost Everywhere

- Recognizing patterns: Speech Recognition, facial identity, etc
- Recommender Systems: Noisy data, commercial pay-off (e.g., Amazon, Netflix)
- Information retrieval: Find documents or images with similar content
- Computer vision: detection, segmentation, depth estimation, optical flow, etc.
- **Robotics:** perception, planning, autonomous driving, etc.

# Autonomous Driving



Source: <https://www.tesla.com/videos/autopilot-self-driving-hardware-neighborhood-long>  
<https://www.youtube.com/watch?v=hLaEV72elj0>

# Machine Learning is Almost Everywhere

- Recognizing patterns: Speech Recognition, facial identity, etc
- Recommender Systems: Noisy data, commercial pay-off (e.g., Amazon, Netflix)
- Information retrieval: Find documents or images with similar content
- Computer vision: detection, segmentation, depth estimation, optical flow, etc.
- Robotics: perception, planning, autonomous driving etc
- **Learning to play games**

# AlphaGo



Source: <https://www.newscientist.com>

Also, large-scale. Why?

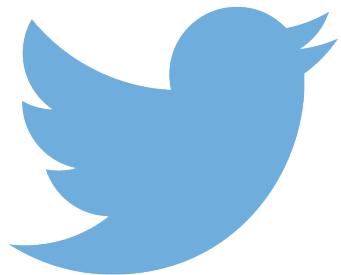
# Content Generation



More than 700 photos uploaded / second



More than 55K google searches / second



More than 7k tweets / second



More than 2.5 million emails sent / second

# Content Generation



More than 700 photos uploaded / second



We have to handle large data sets

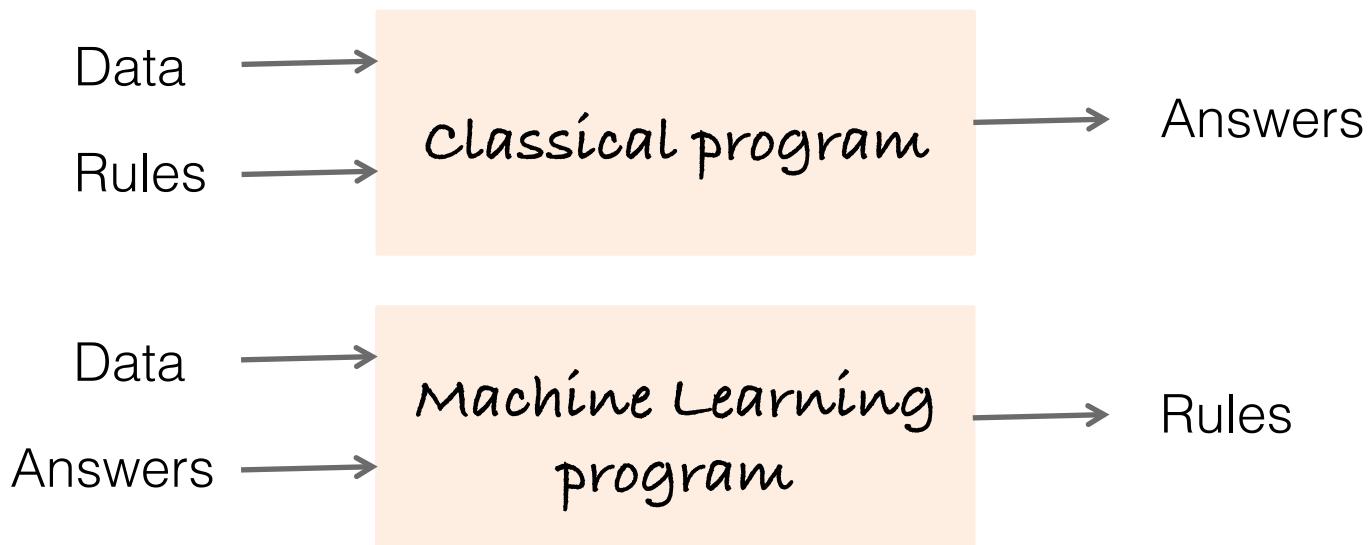


More than 2.5 million emails sent / second



# Why Learning?

- There is no need to “learn” to calculate our taxes
- Learning is used when
  - Human expertise does not exist (e.g., bioinformatics)
  - Humans are unable to explain their expertise (speech recognition, computer vision)
  - Complex solutions change in time (routing computer networks)



# What is Machine Learning?

- Learning systems are not directly programmed to solve a problem, instead **develop their own program** based on:
  - **Examples** of how they should behave
  - From **trial-and-error** experience trying to solve the problem
- Different than standard CS programs
  - Want to implement unknown function, only having access to sample input-output pairs (**training examples**)
- Learning simply means incorporating information from the training examples into the system

# Task that Requires ML

What makes a '2'?

0 0 0 1 1 ( 1 1 1, 2

2 2 2 2 2 2 2 3 > 3

3 4 4 4 4 5 5 5 5

6 6 7 7 7 7 7 8 8 8

8 8 9 7 9 4 9 9 7

# Why Use Learning?

- It is very hard to write programs that solve problems like recognizing a handwritten digit
  - What distinguishes a ‘2’ from a ‘7’?
  - How does our brain do it?
- Instead of writing a program by hand, we **collect examples** that specify the correct output for a given input
- A machine learning algorithm then takes these examples and produces a program that does the job
  - The program produced by the learning algorithm may look very different from a typical hand-written program
  - If we do it right, the program works for **new cases** as well as for the ones we trained it on

# Learning Objectives of Today's Class

- Given a problem
  - Decide weather it can be solved with machine learning
  - Decide as what type of machine learning problem you can formalize it (unsupervised – clustering, dimension reduction, supervised – classification, regression?)
  - Describe it formally in terms of design matrix (observations x features), features, samples
- Define a loss function
- Basic concepts on (convex) optimization

More on that will follow soon

# Example of ML Pipeline

- Running example: image classification
- Goal: “train a computer to recognize a cat from a dog”
- How?

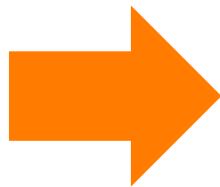


# Example of ML Pipeline

- Running example: image classification
- Goal: “train a computer to recognize a cat from a dog”
- How?

Simple idea, inspired by inductive human learning

Show it a lot of  
labeled examples



“predicts” the right  
label on unseen data

From This

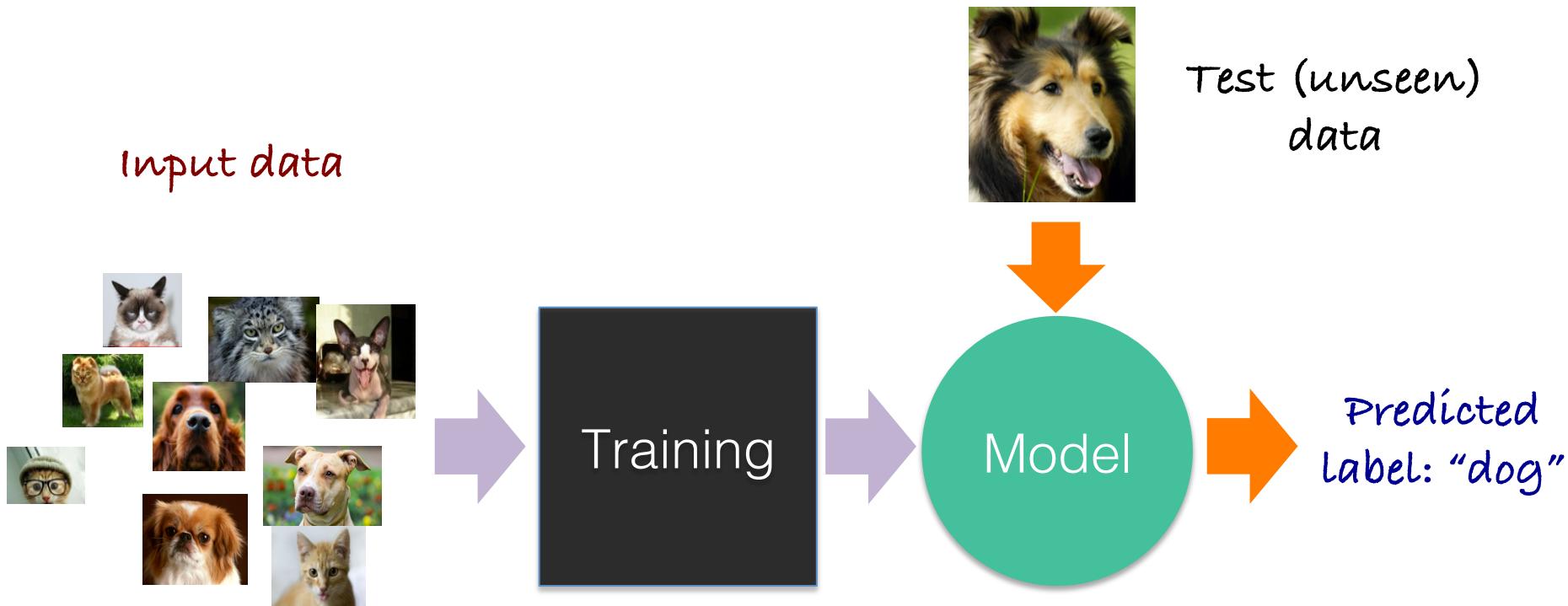


To this

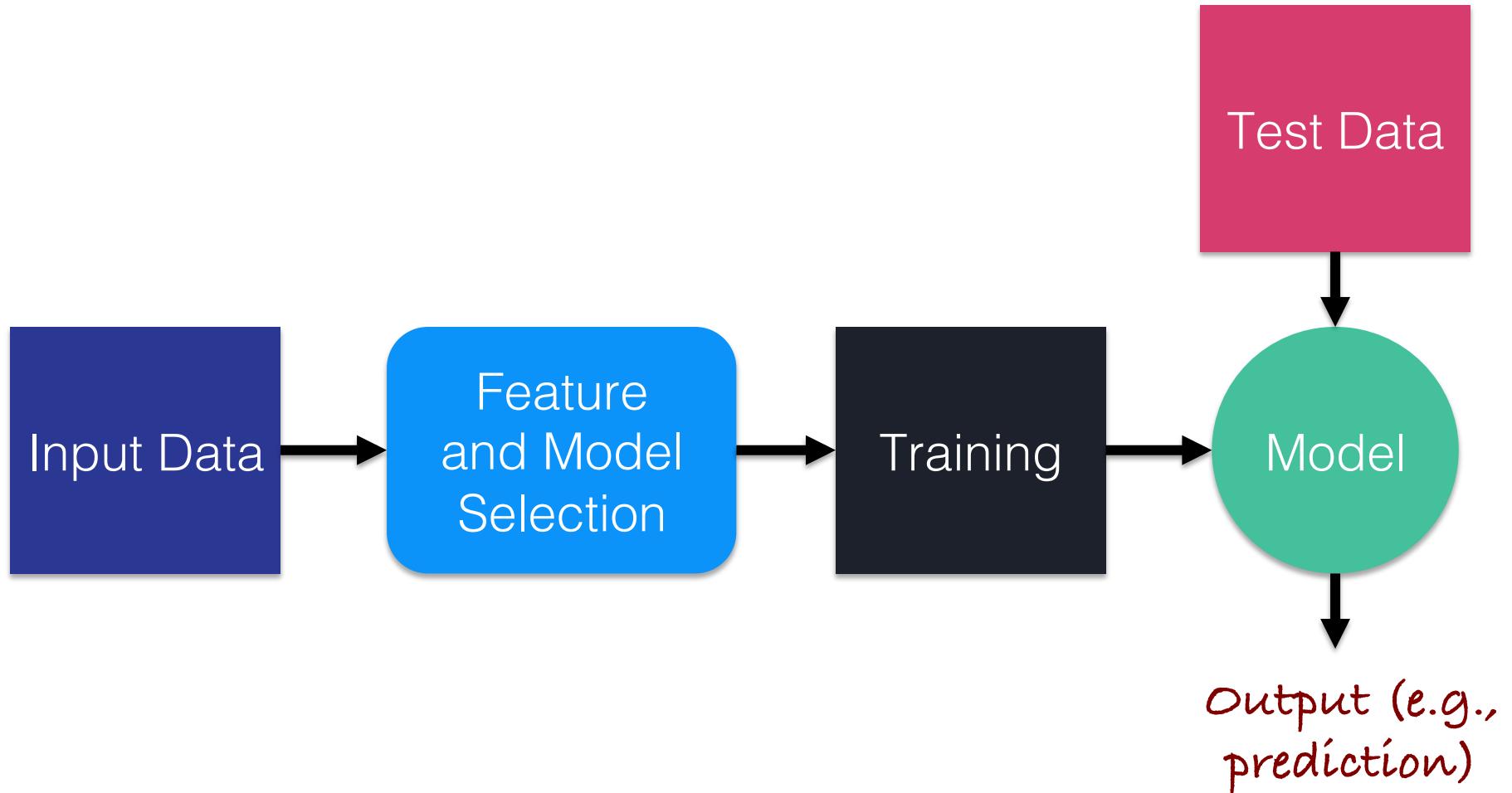


# Example of ML Pipeline

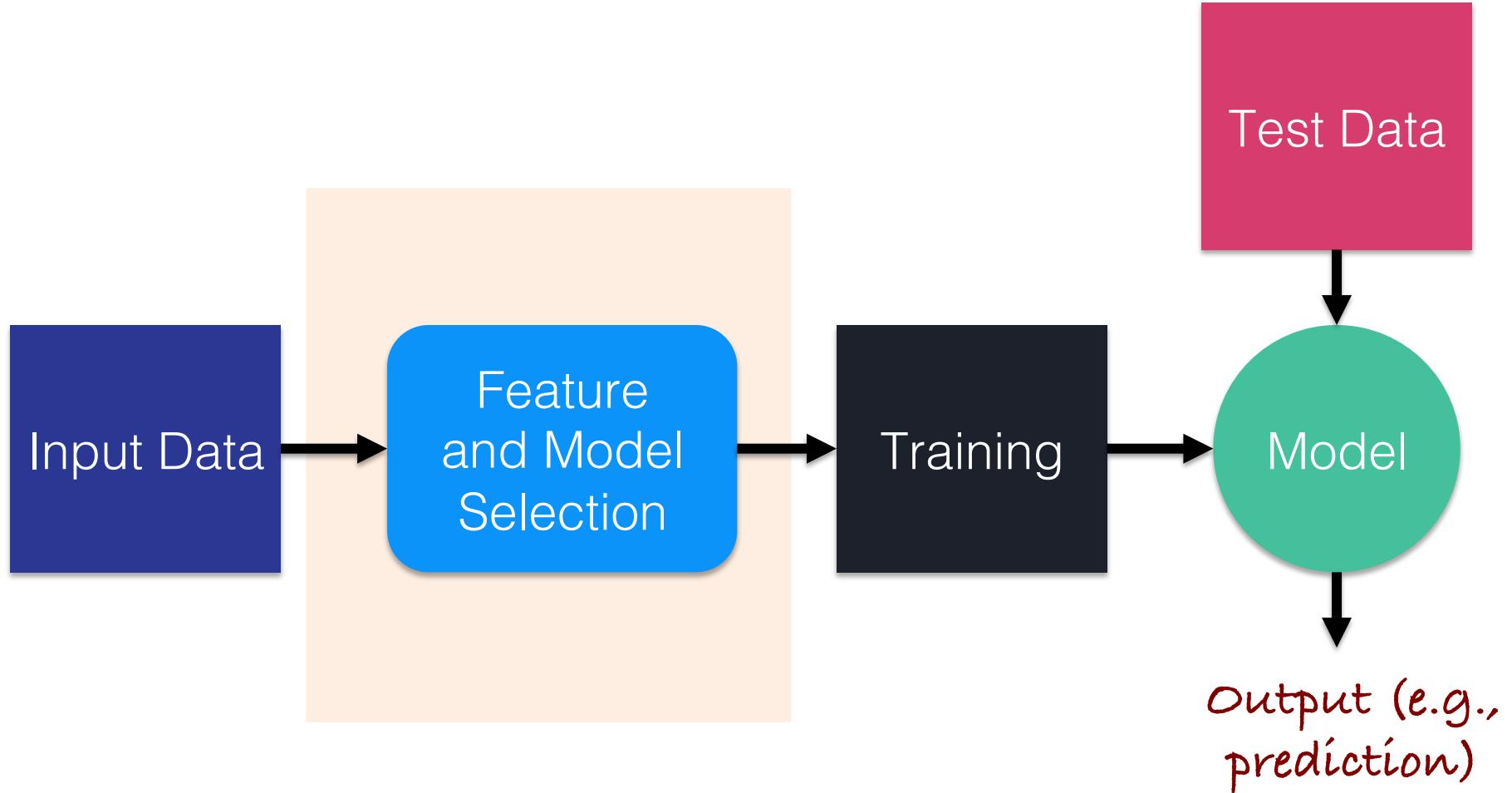
- Running example: image classification
- Goal: “train a computer to recognize a cat from a dog”
- How?



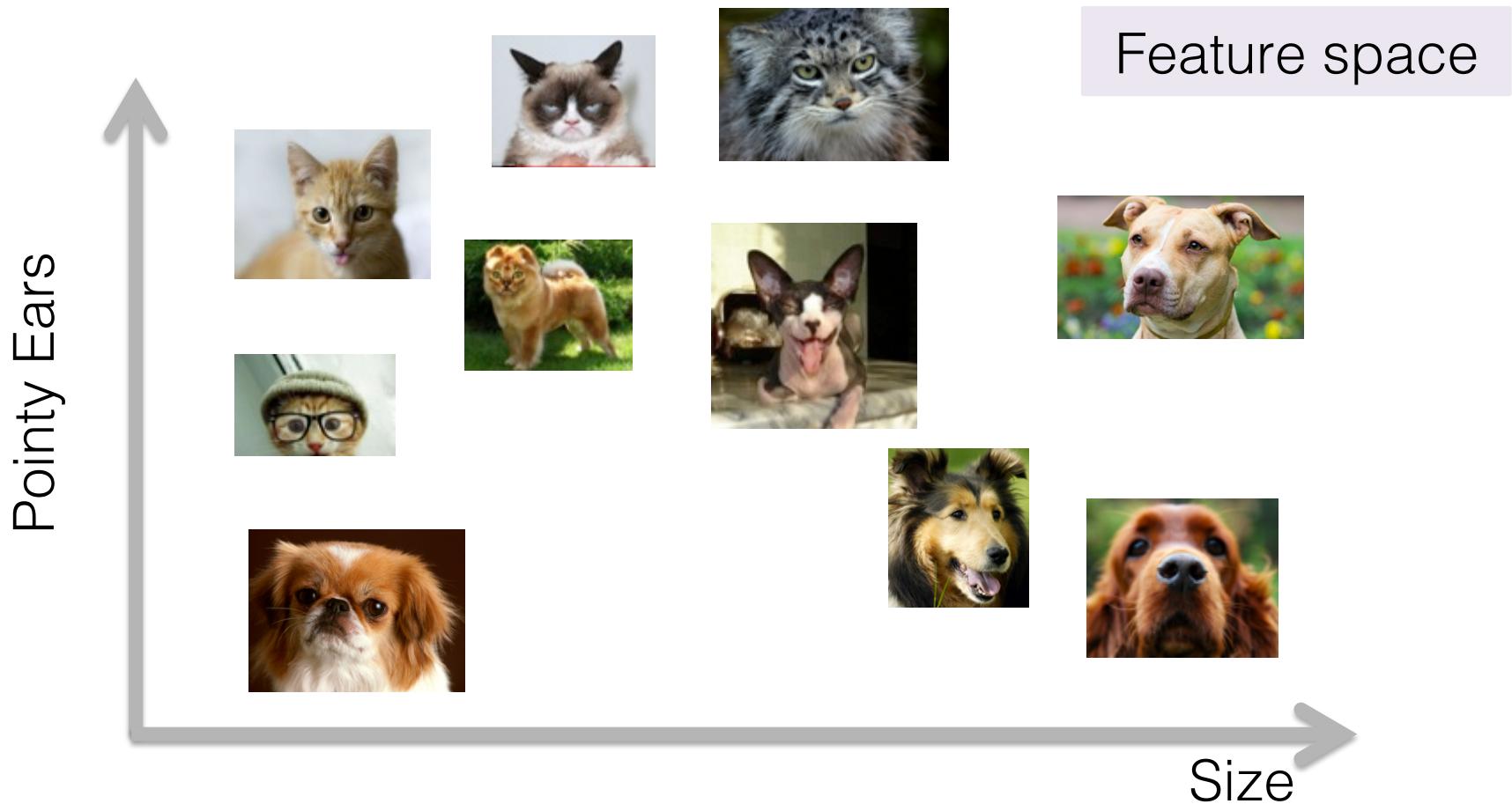
# ML Pipeline



# ML Pipeline

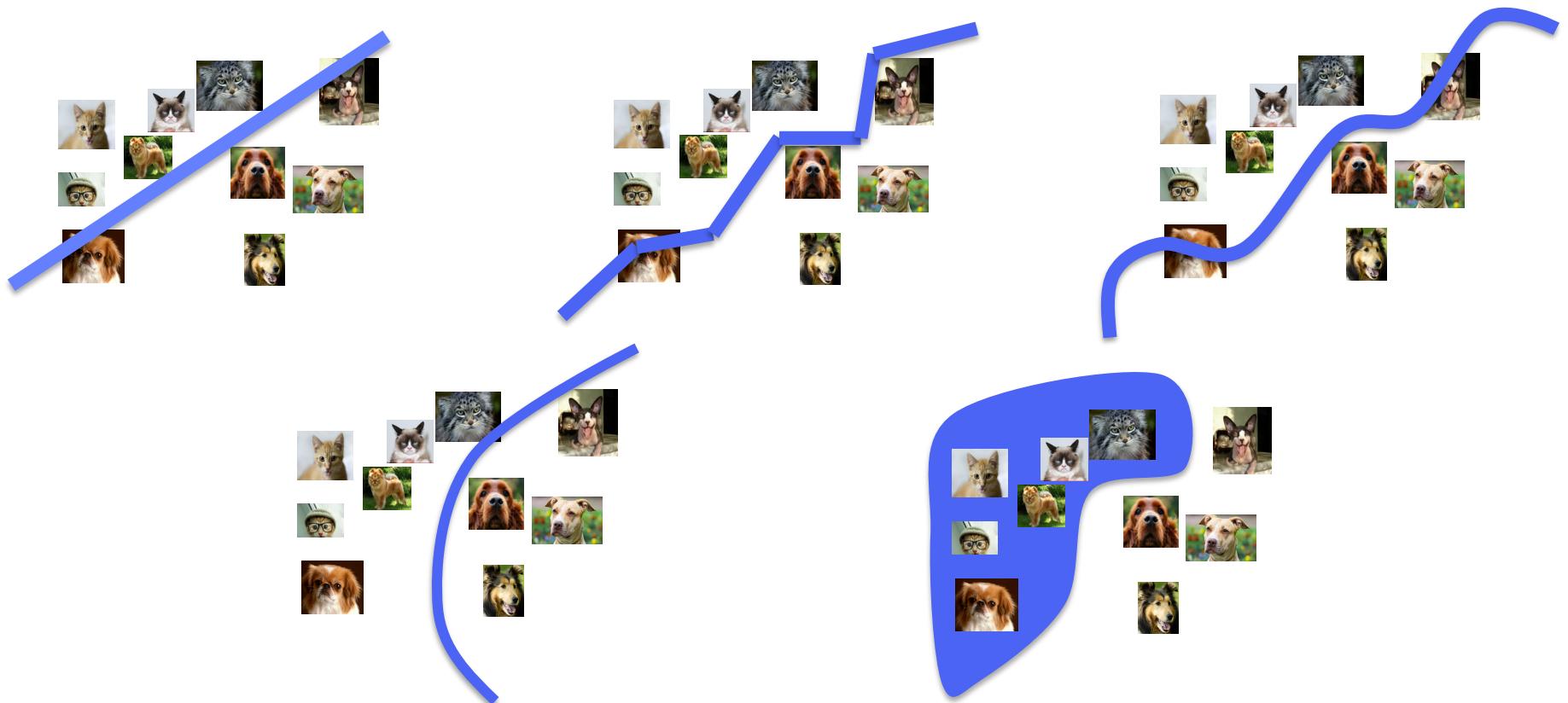


# Feature Selection



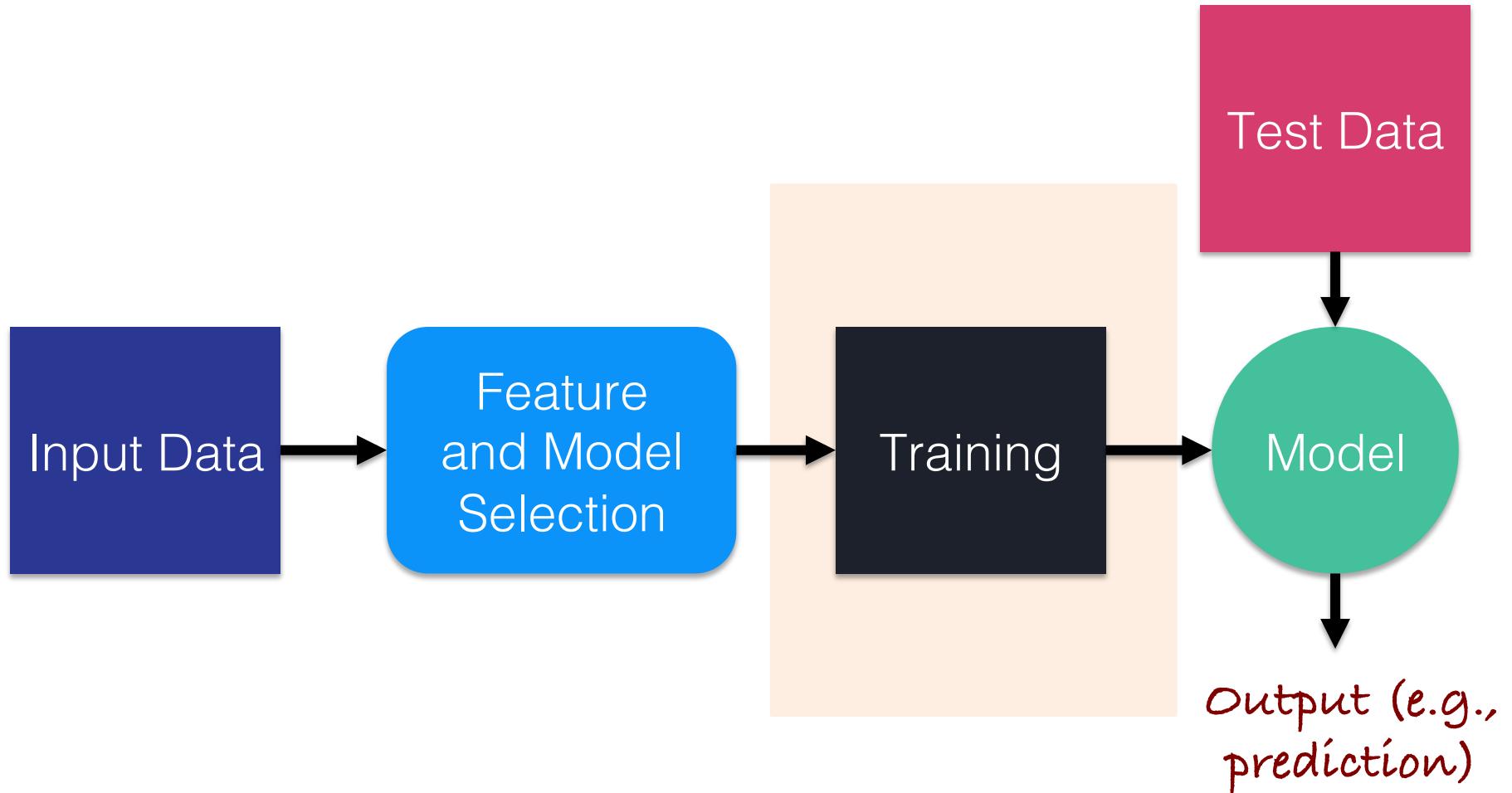
Goal: Use “informative” features

# Choose your Predictor (Hypothesis Class)



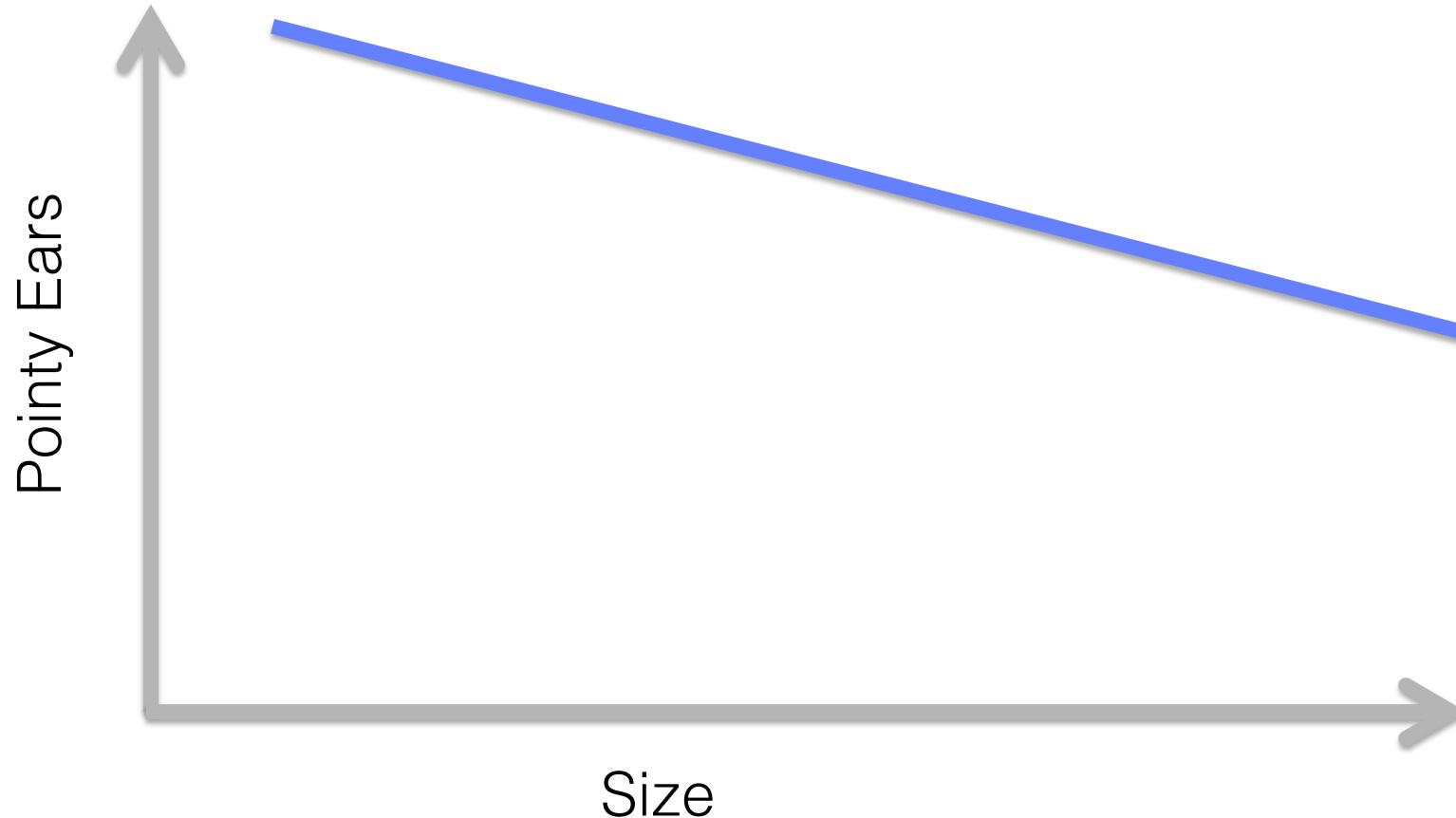
Goal: Pick a predictor that is  
1) expressive 2) easy to train 3) doesn't overfit

# ML Pipeline



# Then, Train the Model

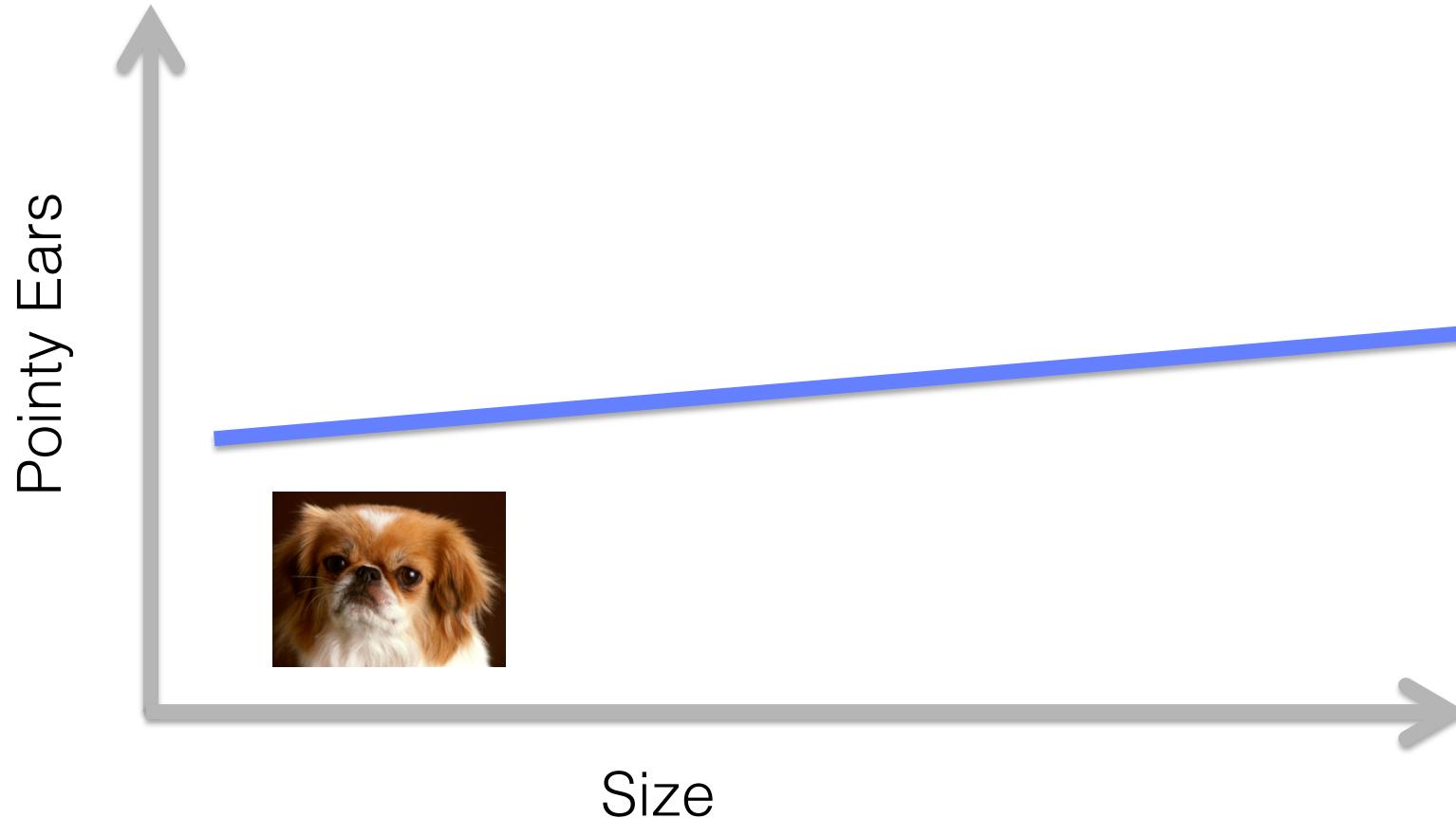
Feature space



Goal: Train a model to minimize training error

# Then, Train the Model

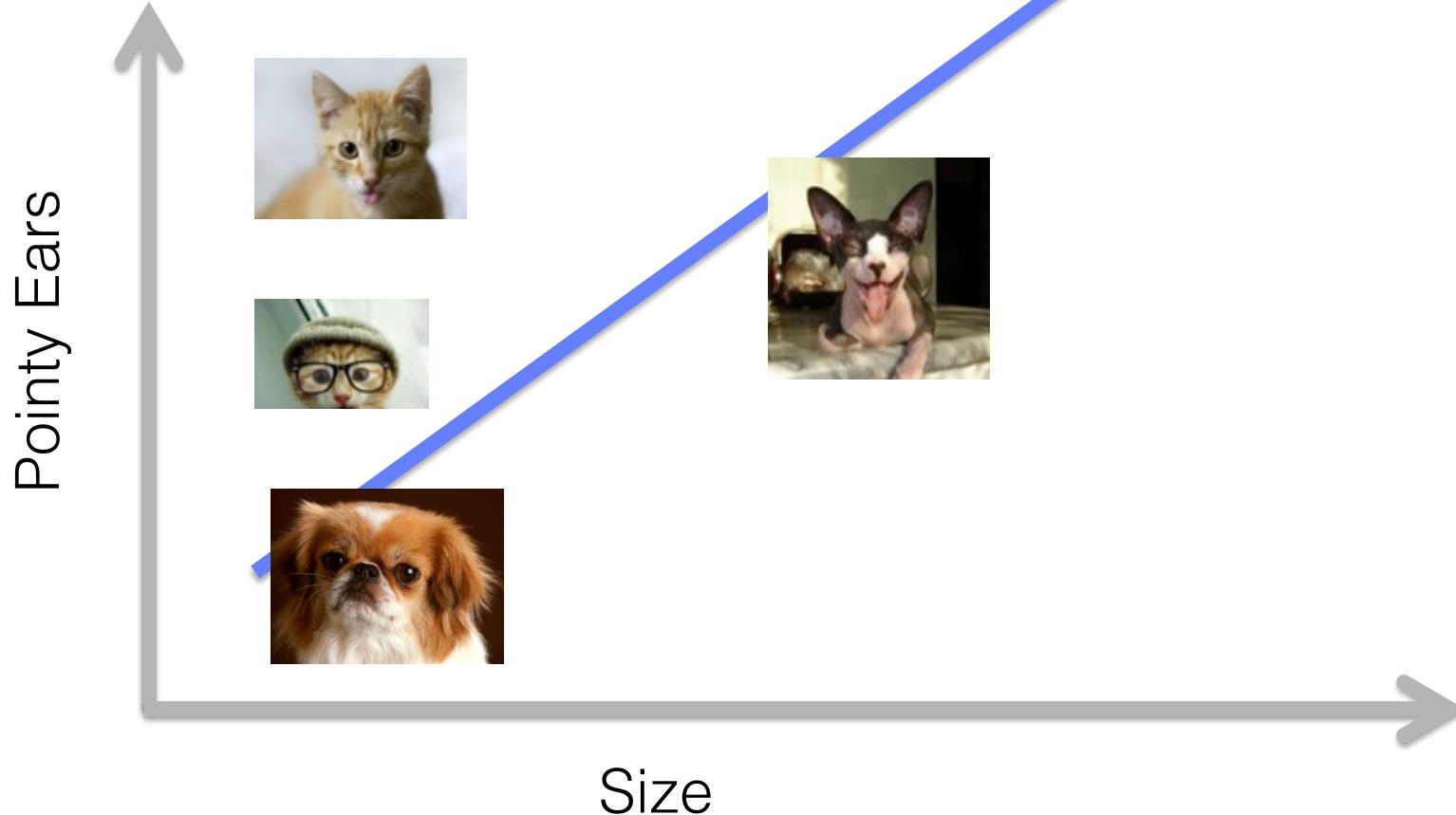
Feature space



Goal: Train a model to minimize training error

# Then, Train the Model

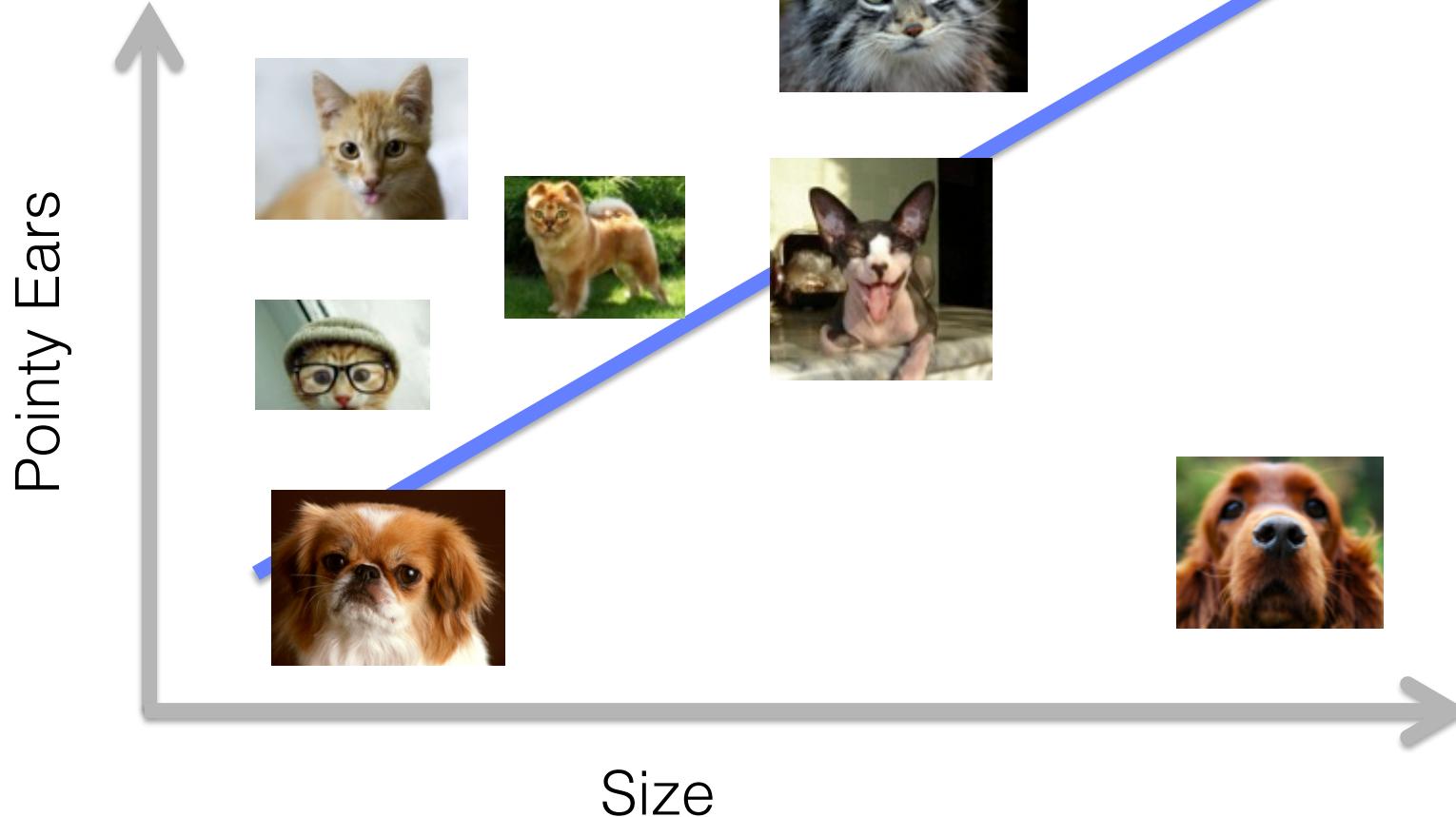
Feature space



Goal: Train a model to minimize training error

# Then, Train the Model

Feature space

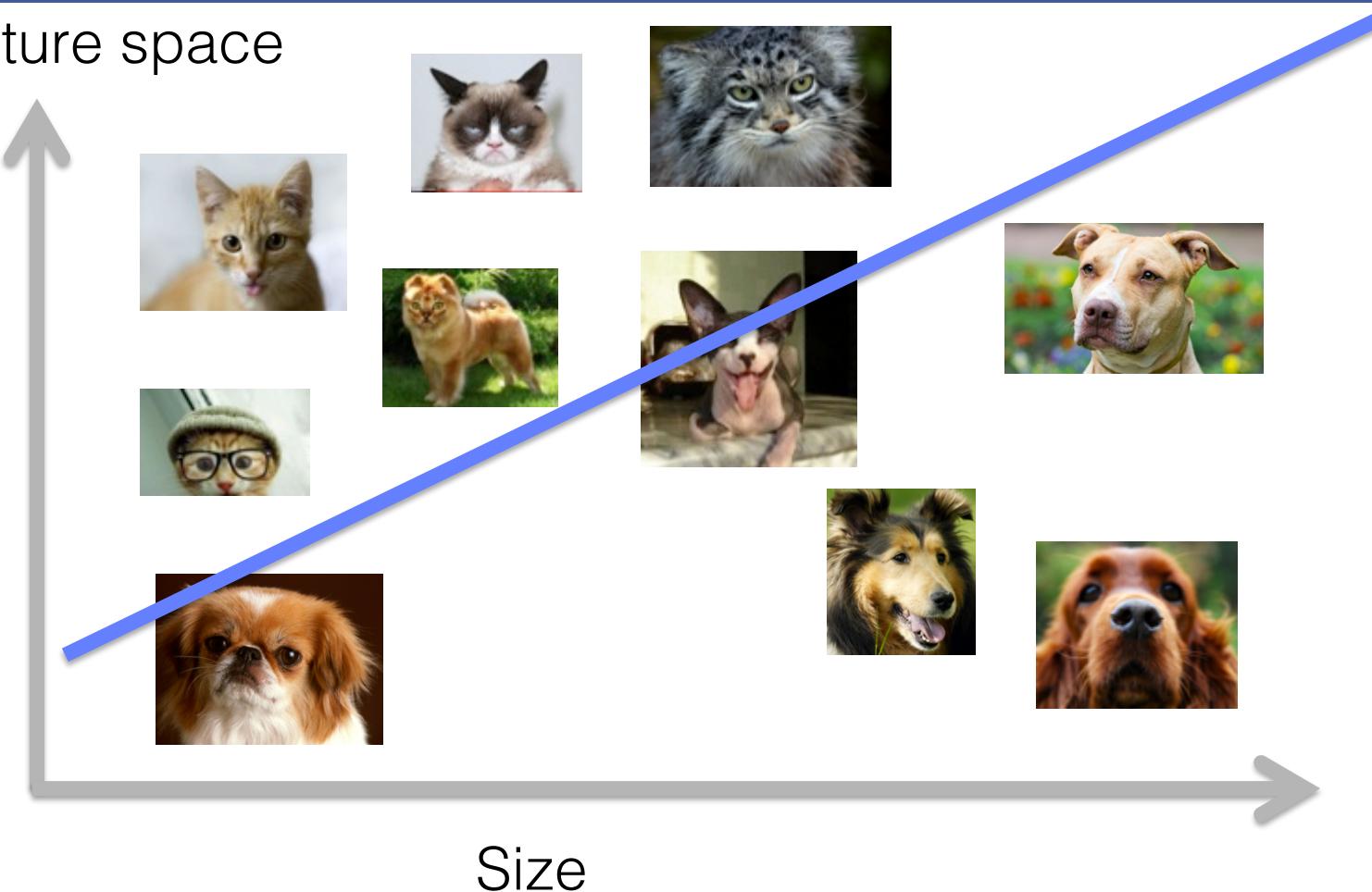


Goal: Train a model to minimize training error

# Then, Train the Model

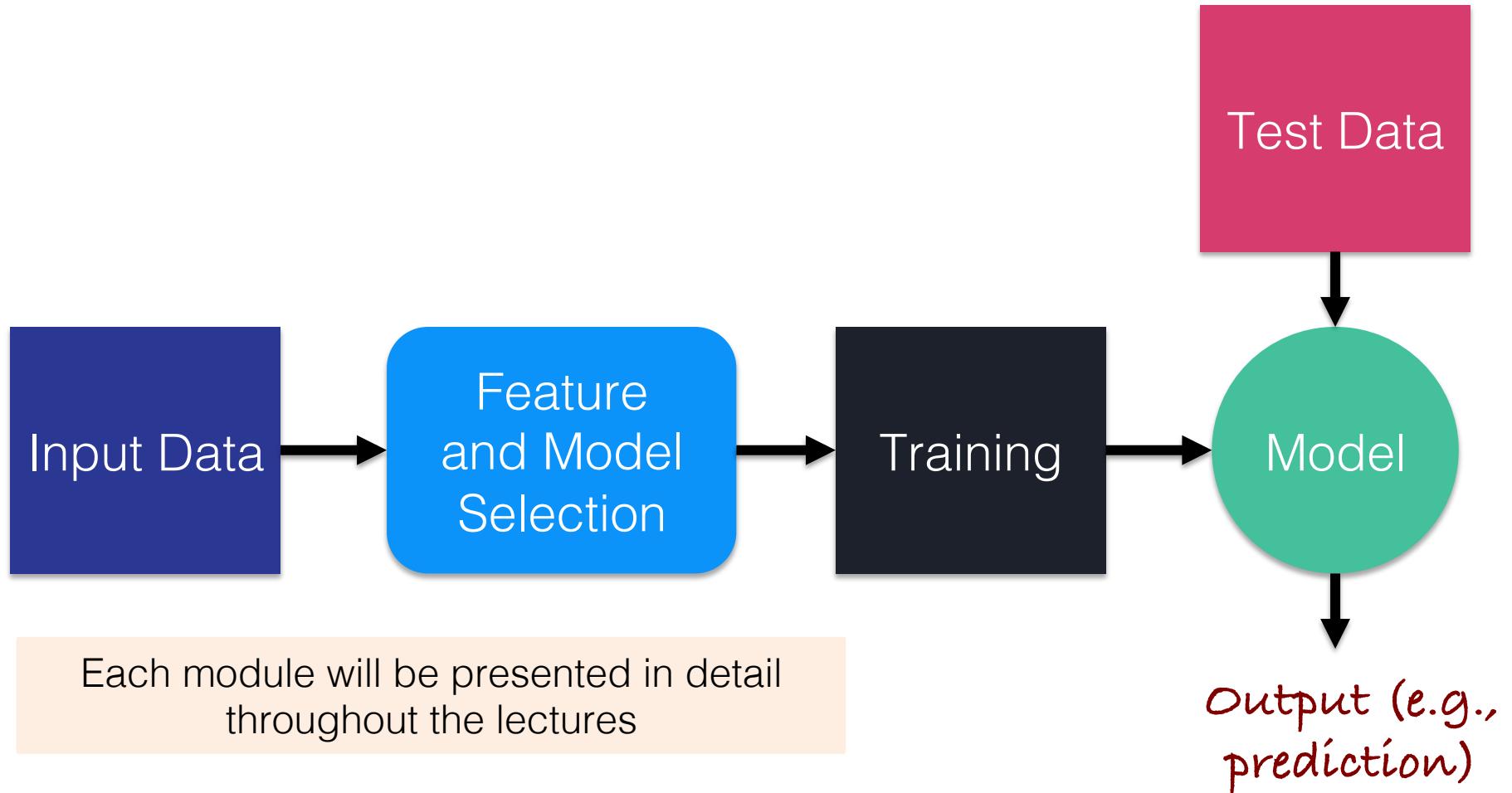
Feature space

Pointy Ears



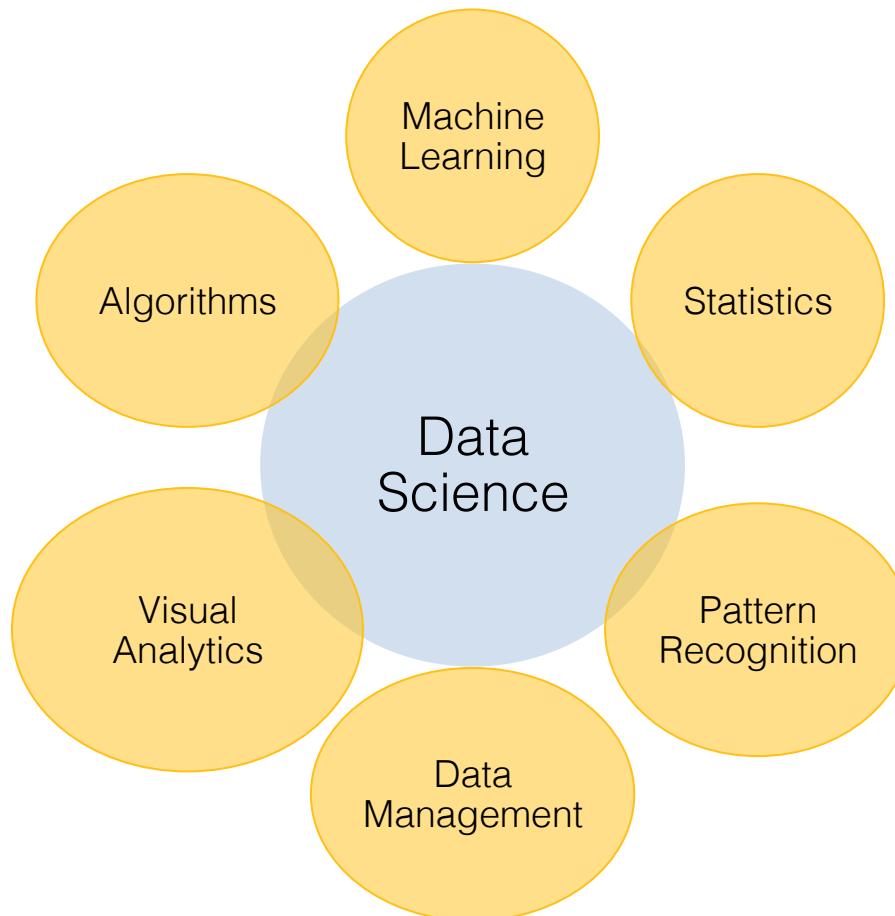
Goal: Train a model to minimize training error

# ML Pipeline



Questions?

# Data Science: The Big Picture



# About this course

# Learning Objectives

- By the end of the course, you will be able to
  - Identify problems that can be solved by machine learning
  - Formulate your problem in machine learning terms
  - Given such a problem, identify and apply the most appropriate classical algorithm(s)
  - Implement some of those algorithms by yourself
  - Evaluate and compare machine learning algorithms for a particular task
  - Deal with real-world data challenges

# Prerequisites

- Basic knowledge of
  - Probability theory and statistics
  - Linear algebra
  - Algorithms
- We will review the main background concepts
- Programming is necessary
  - Python (or any other language of your preference)
  - We will deal with real-world ML tasks

# Structure of the Course

Three components:

1. **Lectures**
  - First half of each session
2. **Lab sessions**
  - Need to install software and to experiment in class
  - Labs will not be graded, but will help you to further understand the material presented in the lectures
  - Hands-on experience on ML algorithms
    - Some of the algorithms will be implemented from scratch
3. **Assignments and project**

[We might slightly deviate in the first three 2h lectures]

# Coursework and Grading

	Weight	Details
Assignment 1 (individual)	15%	<ul style="list-style-type: none"><li>• Theoretical questions</li><li>• Some of them may also require some programming in order to perform some tasks</li><li>• <i>Lecture 3 (out) – Lecture 4 (due)</i></li></ul>
Assignment 2 (teams of 3-4 students)	35%	<ul style="list-style-type: none"><li>• Deal with a real machine learning task</li><li>• Kaggle competition</li><li>• Deliver short report and code</li><li>• <i>Lecture 4 (out) – Lecture 9 (due)</i></li></ul>
Project (teams of 3-4 students)	50%	<ul style="list-style-type: none"><li>• Project proposal (2-pages, mandatory)</li><li>• Final report and code</li><li>• Presentation in class</li><li>• <i>Proposal due: Lecture 4; Final report due: Lecture 10</i></li></ul>

- Small adjustments may be done in the weights of the coursework
- A detailed description of the project will be provided soon
- The exact dates will be posted on the website of the course (subject to change)

# Goals of the Different Course Components

- Understand the basics behind ML algorithms and get comfortable working with data and tools [Lab sessions]
- Comprehend the foundational material and the motivation behind different techniques [Assignment 1]
- Build something that actually works [Assignment 2]
- Apply your knowledge creatively [Project]

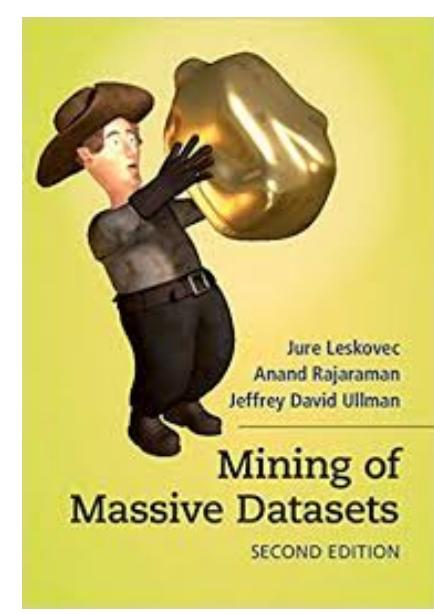
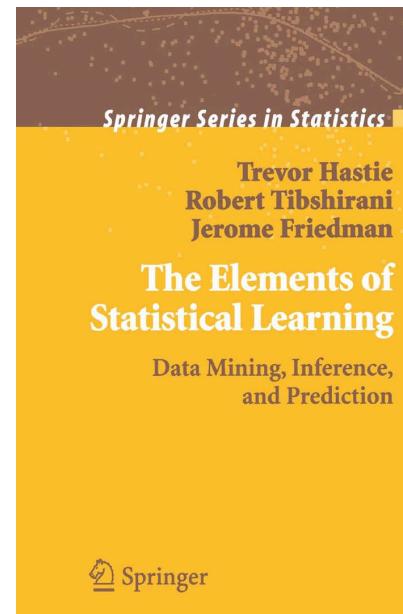
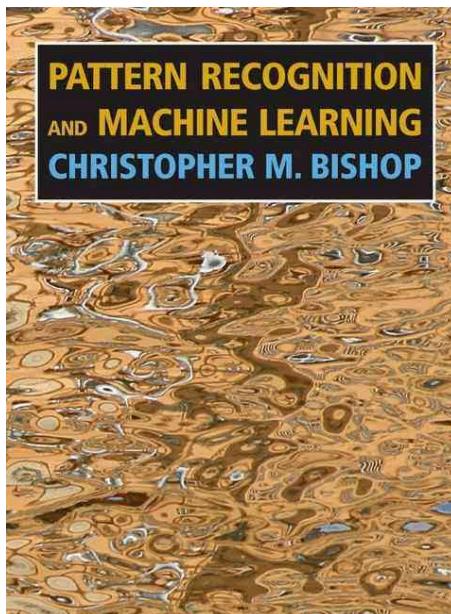
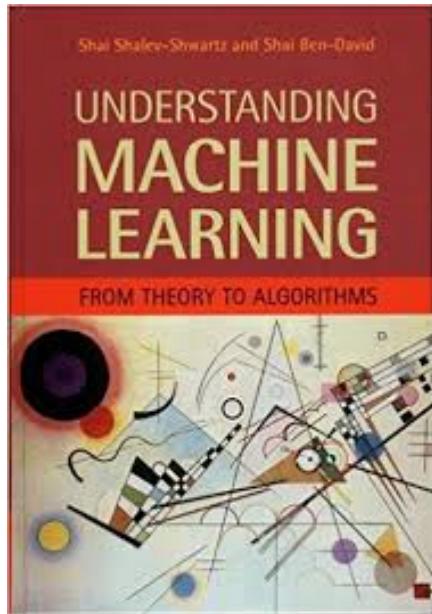
# Software Tools

- We strongly advise to use **Python**
  - numpy
  - scipy
  - scikit-learn
  - pandas
  - ...
  - **anaconda** includes almost all packages that will be needed
- Python will be used in the lab sessions
- See the **Resources** section of the website

# Course Logistics

- Website
  - <http://fragkiskos.me/teaching/ML-F18/>
  - Information about the course, schedule, reading material
  - Resources (helpful for the assignment and project)
- Piazza for Q&A and material
  - <https://piazza.com/centralesupelec/fall2018/mlsbaai/>
  - Please, participate and help each other!
  - All announcements will be posted there
  - Also, lecture slides and assignments
  - Use key to enroll: **ml-2018**

# Reading Material



- The books are publicly available in electronic form
  - Pointers to chapters for every lecture (see the website)
- Additional resources for every lecture will be given in the website of the course

# Some Personal Notes 😊

- Please ask questions, participate in discussions on piazza
- Check out the additional suggested material on the website
  - Search the web, google is your friend!
  - For every topic covered in the class, you can find material in textbooks or even in the web
  - Typically, the suggested reading material is overlapping – read selectively
- Play with software tools. Apply what you've learnt in theory
  - This is the actual goal of the lab sessions, assignments and the project
- Give us your feedback!

# Topics that will be covered

# Schedule (Subject to Change)

<http://fragkiskos.me/teaching/ML-F18/>

	Date	Topic	Material	Assignments/Project
1	October 4	Introduction	Lecture 1	
2	October 11	Dimensionality reduction		
3	October 18	Model selection and evaluation		<b>Assignment 1 out</b> <b>Project proposal due on October 21</b>
4	November 5	Linear and logistic regression		<b>Assignment 1 due on November 4</b> <b>Assignment 2 out</b>
5	November 12	Probabilistic classifiers and linear discriminant analysis		
6	November 14	Non-parametric learning and nearest neighbor methods		
7	November 19	Tree-based methods and ensemble learning		
8	November 21	Support Vector Machines		
9	November 26	Unsupervised learning: clustering		<b>Assignment 2 due on December 2</b>
10	December 12	Project presentations		<b>Project final report due</b>

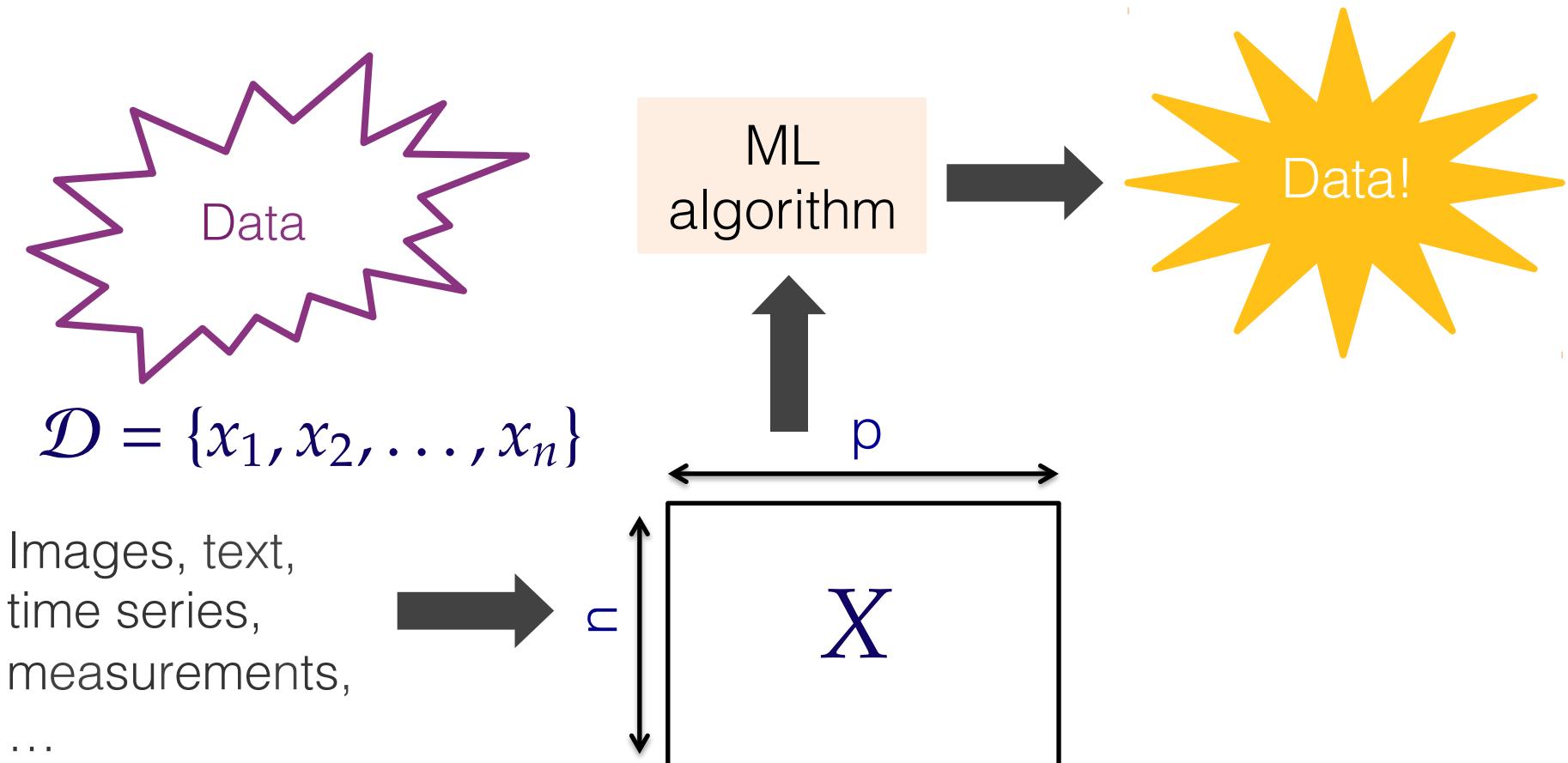
# Schedule (Subject to Change)

1. Introduction. Overview of ML problems. Basic optimization concepts
2. Dimensionality reduction. Feature selection. Principal Component Analysis (PCA).
3. Model evaluation and selection. Overfitting and regularization
4. Linear and logistic regression
5. Probabilistic classifiers. Linear Discriminant Analysis (LDA)
6. Non-parametric learning. K-Nearest Neighbors
7. Tree-based methods. Ensemble methods. Boosting. Random forests
8. Support Vector Machines (SVMs)
9. Unsupervised learning. Data Clustering

Over 9 lectures

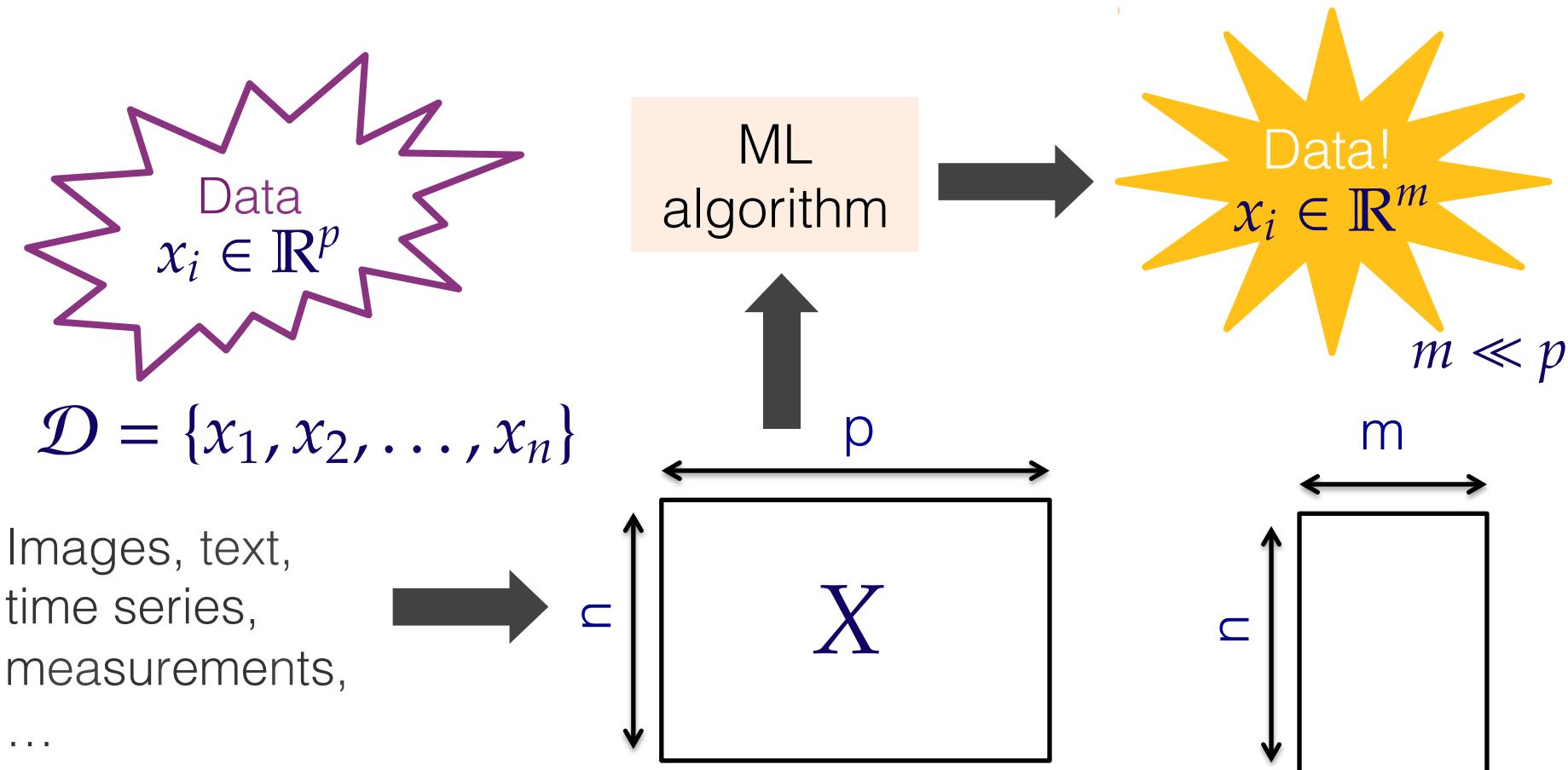
# Unsupervised Learning

Learn a new representation of the data



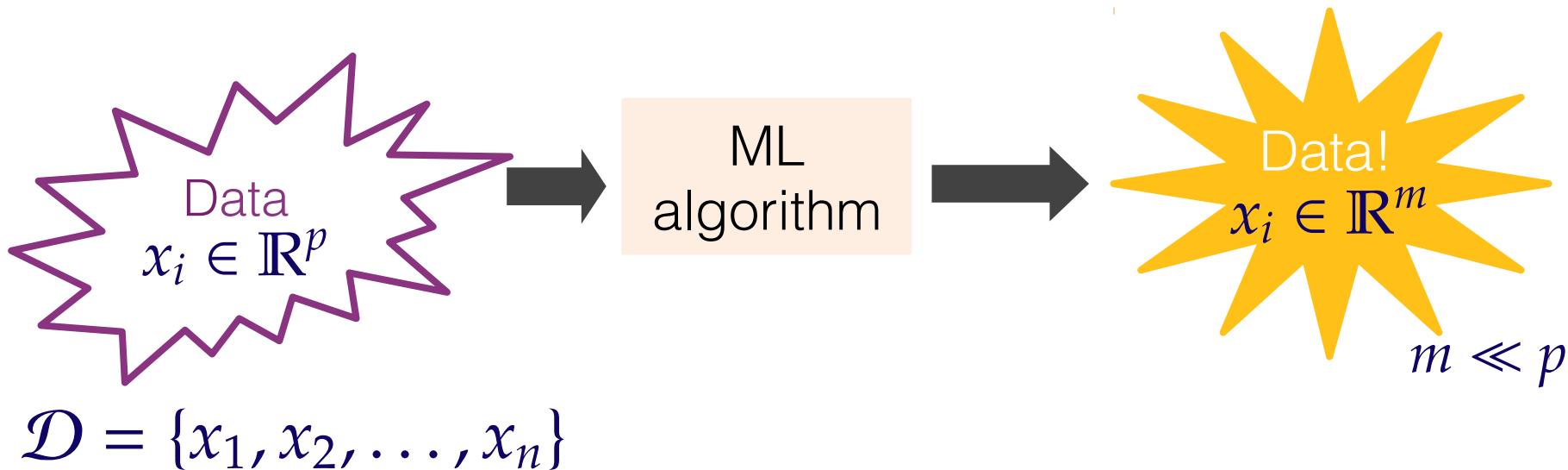
# Dimensionality Reduction (1/2)

Find a lower-dimensional representation



# Dimensionality Reduction (2/2)

Find a lower-dimensional representation

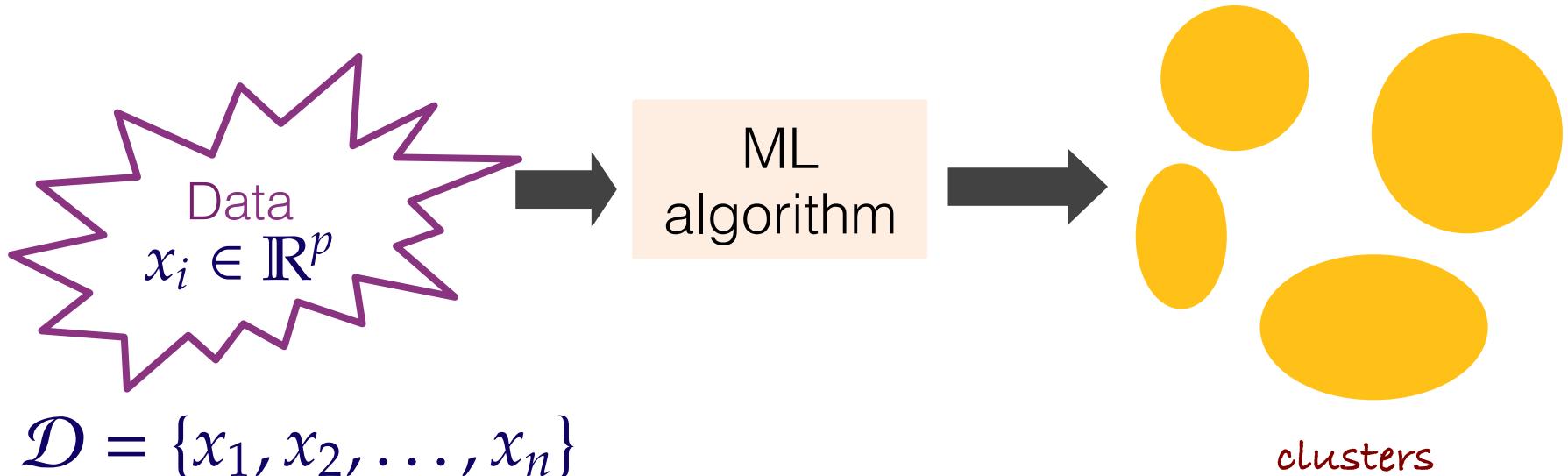


*why dimensionality reduction is useful?*

- Reduce storage space and computational time
- Remove redundancies
- Curse of dimensionality
- Visualization (in 2 or 3 dimensions) and interpretability

# Data Clustering

Group **similar** data points together



- Understand **general characteristics** of the data
- Infer some properties of an object based on how it relates to other objects
- Problems that can be solved by clustering?

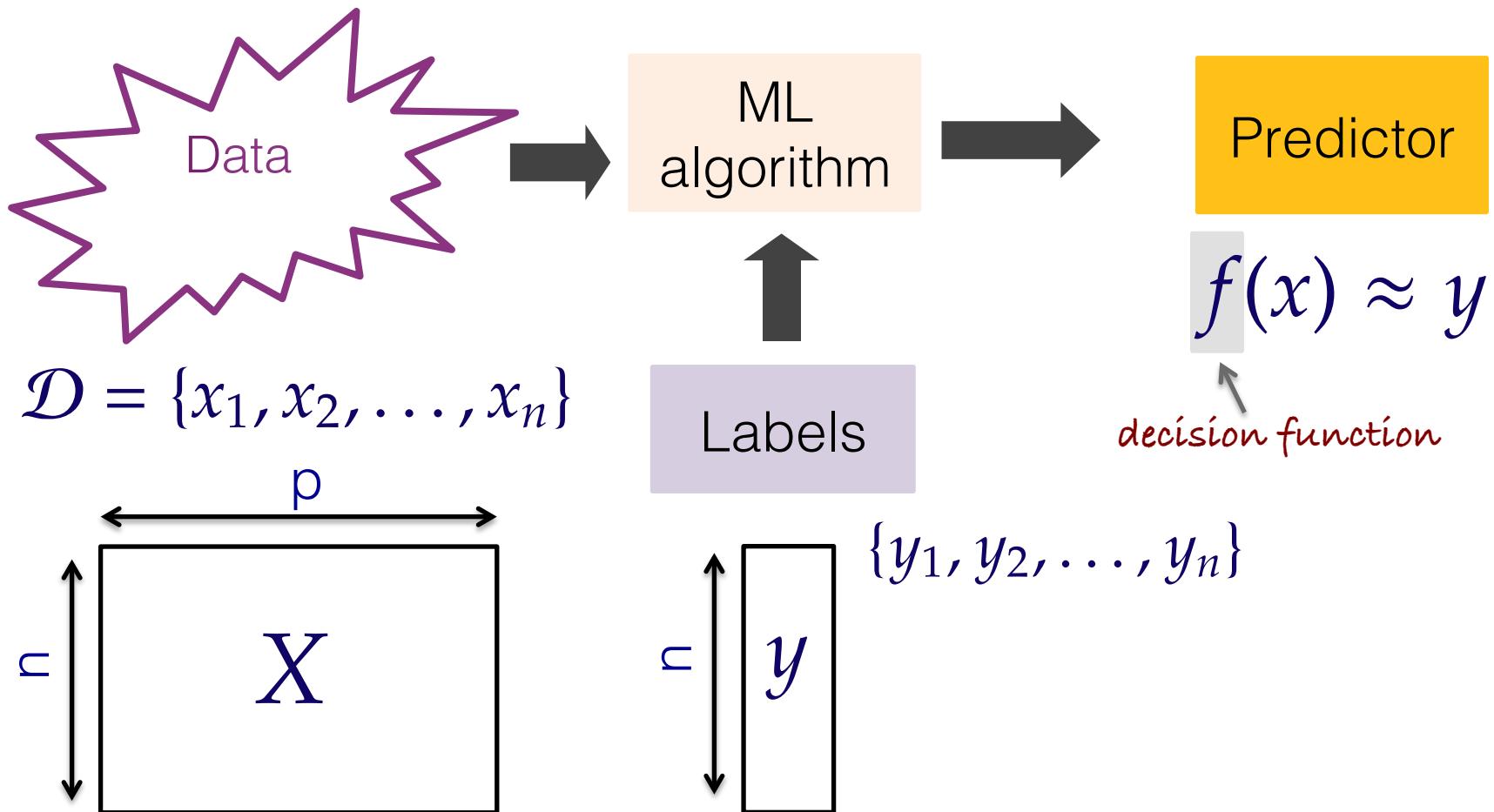
# Data Clustering – Applications

- Customer segmentation
  - Find groups of customers with similar buying behavior
- Topic modeling
  - Group documents based on the words they contain to identify common topics
- Image compression and segmentation
  - Find groups of similar pixels that can easily be summarized
- Disease subtyping (e.g., cancer, mental health)
  - Find groups of patients with similar pathologies (at the molecular or symptoms level)
- Community detection in networks
  - Communities of similar users in social networks
- ...

Can you think any inherent difficulty in the clustering task?

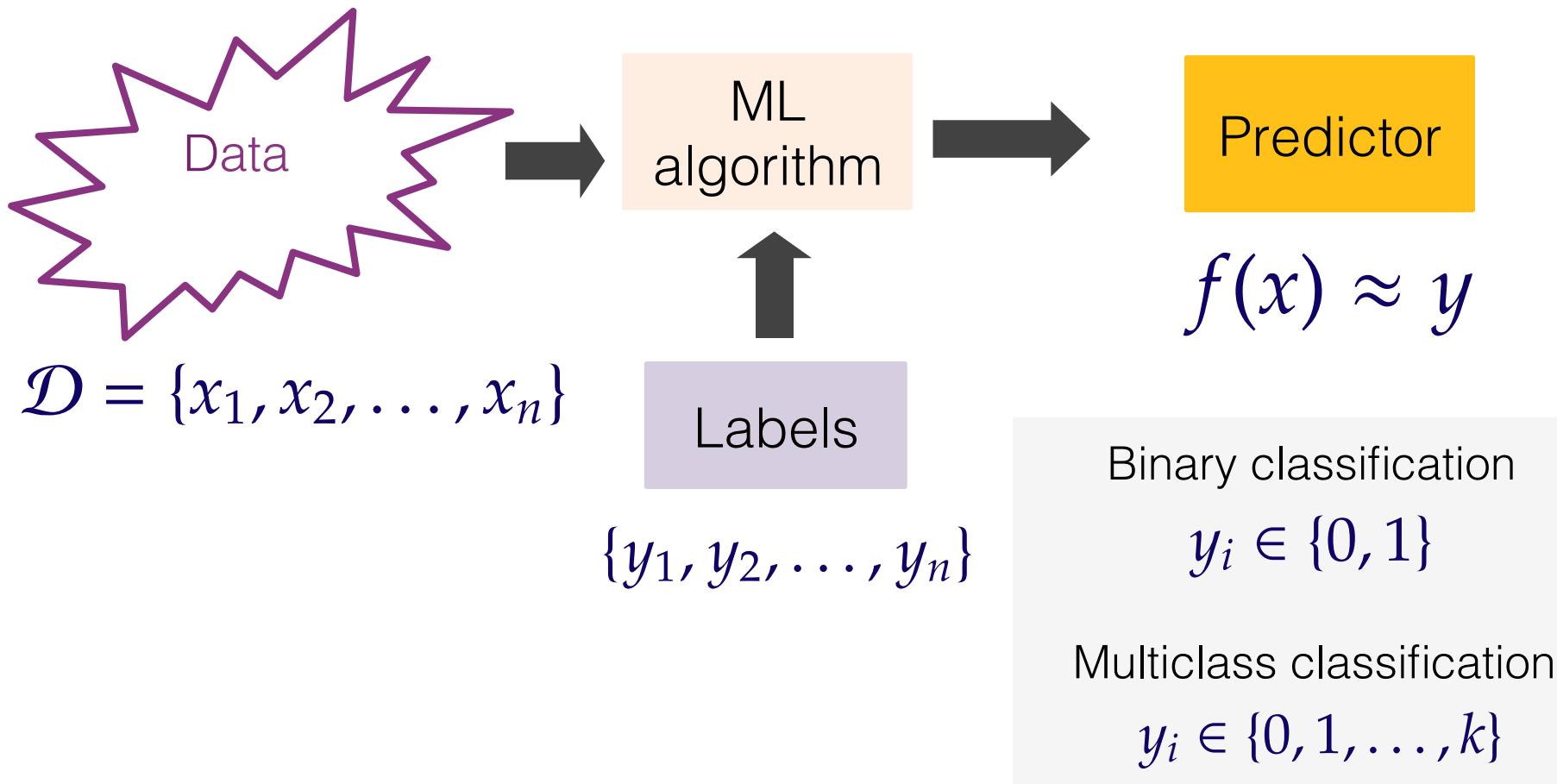
# Supervised Learning

Make predictions

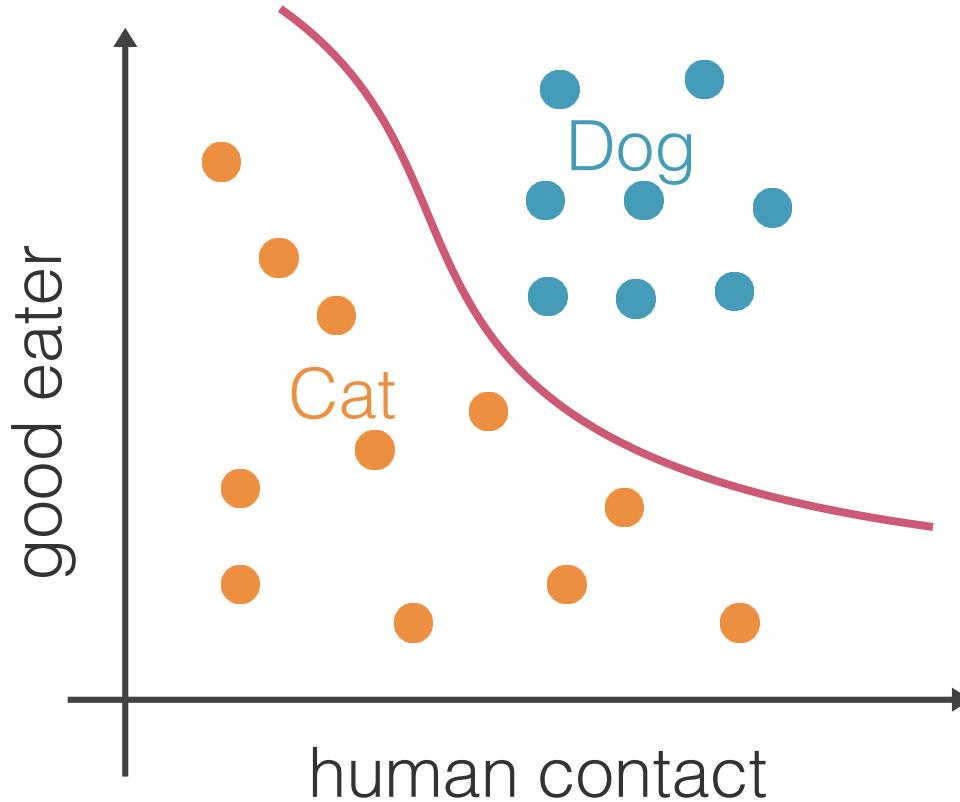


# Classification (1/2)

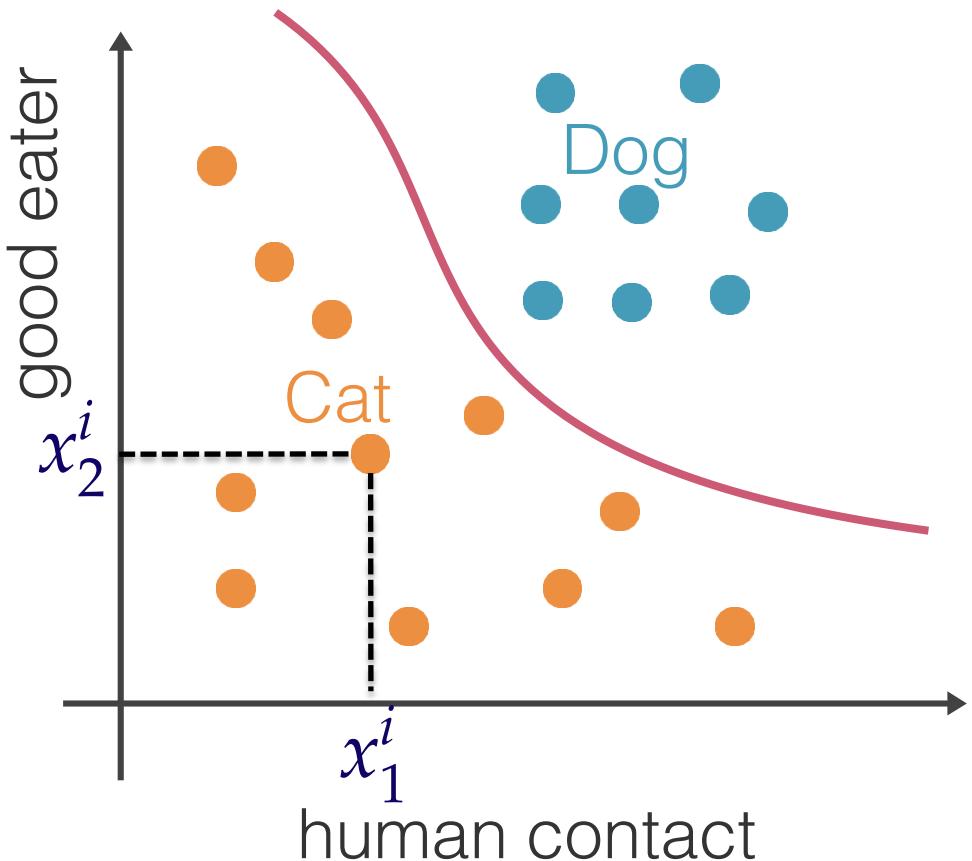
Make discrete predictions



# Classification (2/2)



# Training Set $\mathcal{D}$



$$\mathcal{D} = \{x^i, y^i\}_{i=1,\dots,n}$$

$$y_i = \begin{cases} 1 & \text{if } x^i \in \mathcal{P} \\ -1 & \text{if } x^i \in \mathcal{N} \end{cases}$$

$$x^i = \begin{pmatrix} x_1^i \\ x_2^i \end{pmatrix}$$

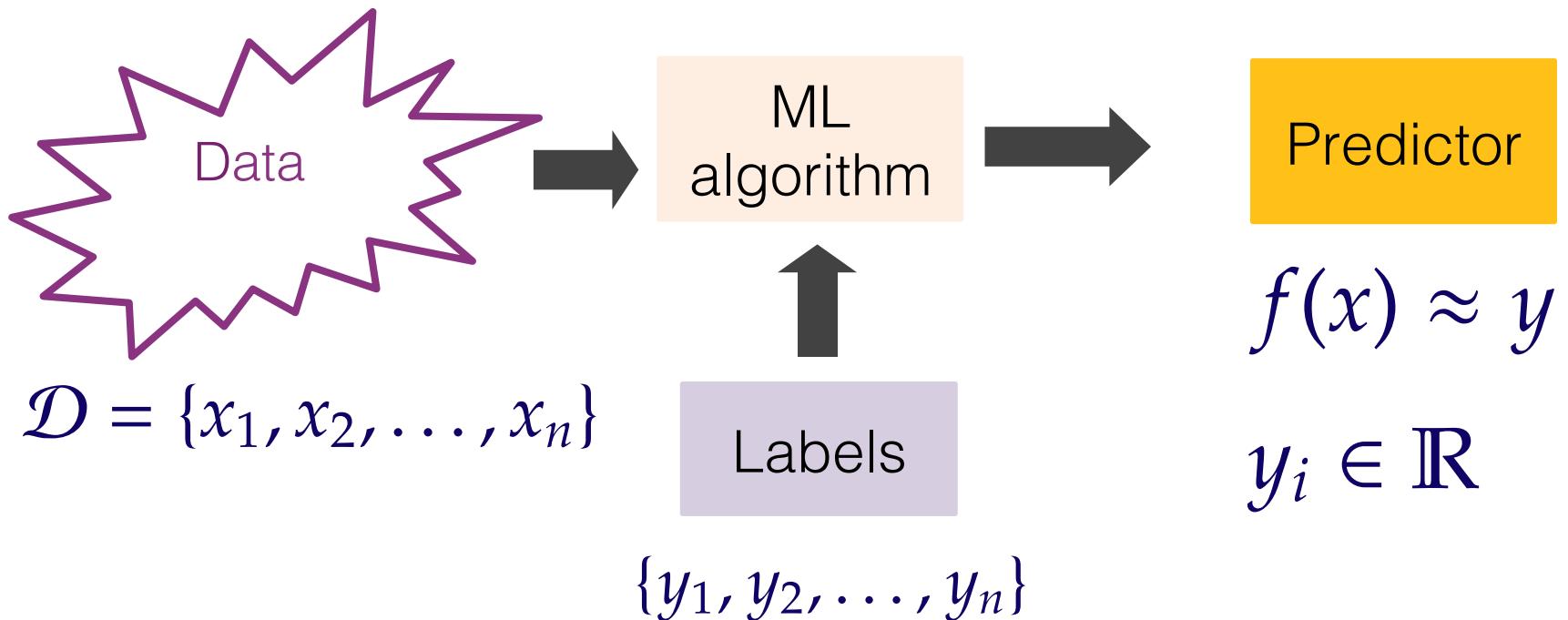
Given  $\mathcal{D} = \{x^i, y^i\}_{i=1,\dots,n}$  find  $\mathbf{f}$  such that  $f(x) \approx y$

# Classification – Applications

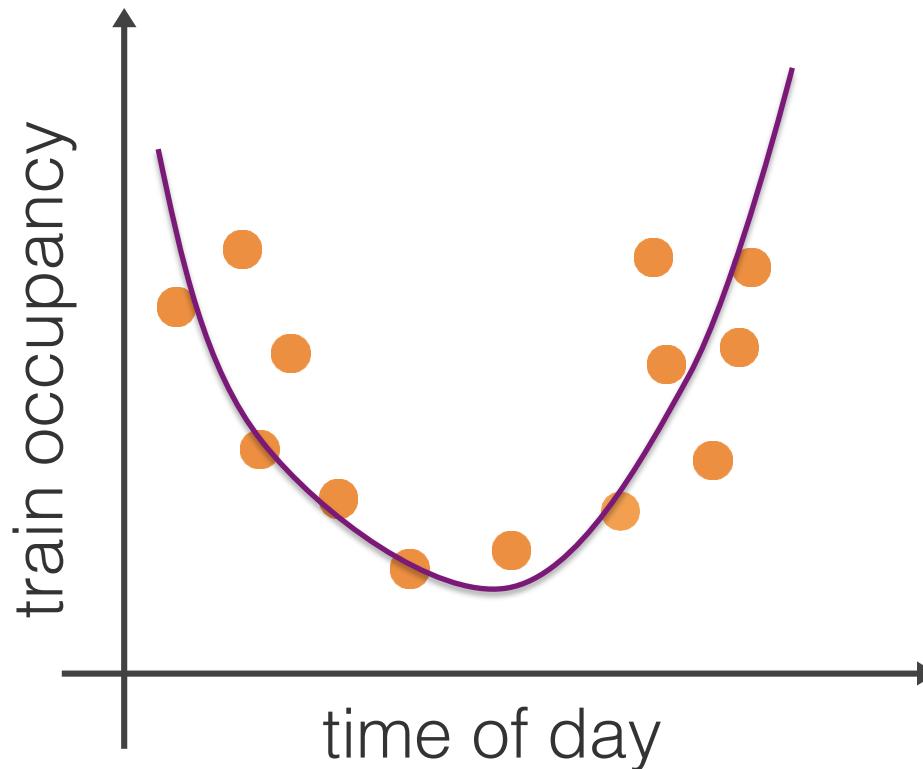
- Face recognition
  - Identify faces independently of pose, lighting, occlusion (glasses, beard), make-up, hair style
- Self-driving cars. How?
- Character recognition
  - Read letters or digits independently of different handwriting styles
- Sound recognition
  - Which language is spoken? Who wrote this music? What type of bird is this?
- Spam detection. Any spam application that you may know?
- Precision medicine
  - Does this sample come from a sick or healthy person? Will this drug work on this patient?

# Regression (1/3)

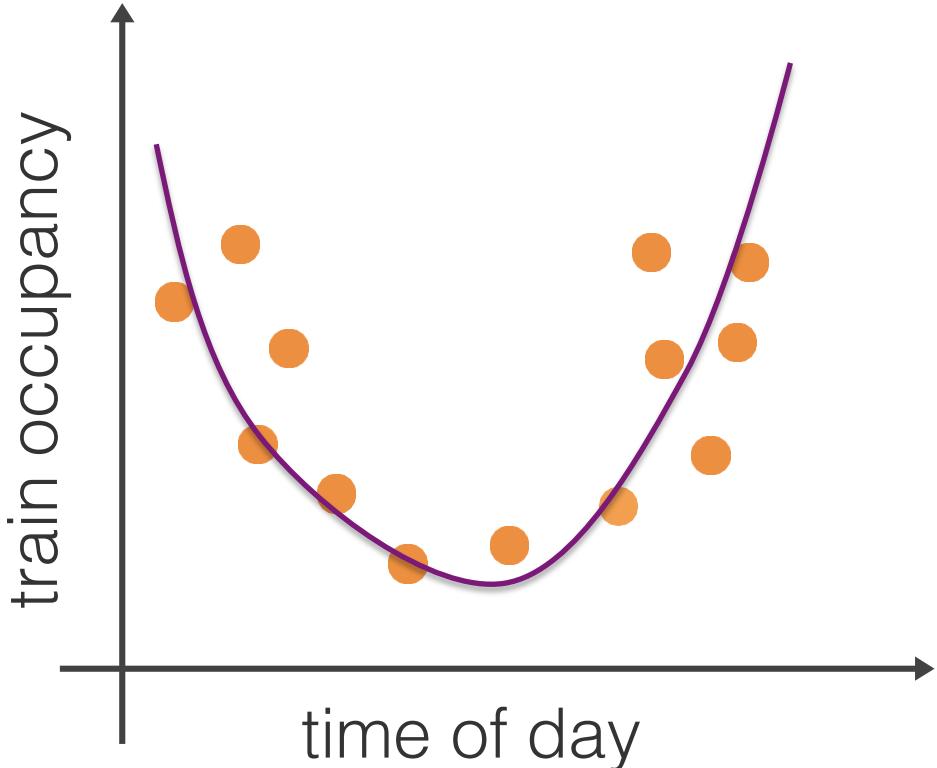
Make continuous predictions



# Regression (2/3)



# Regression (3/3)



$$\mathcal{D} = \{x^i, y^i\}_{i=1,\dots,n}$$
$$y^i \in \mathbb{R}$$

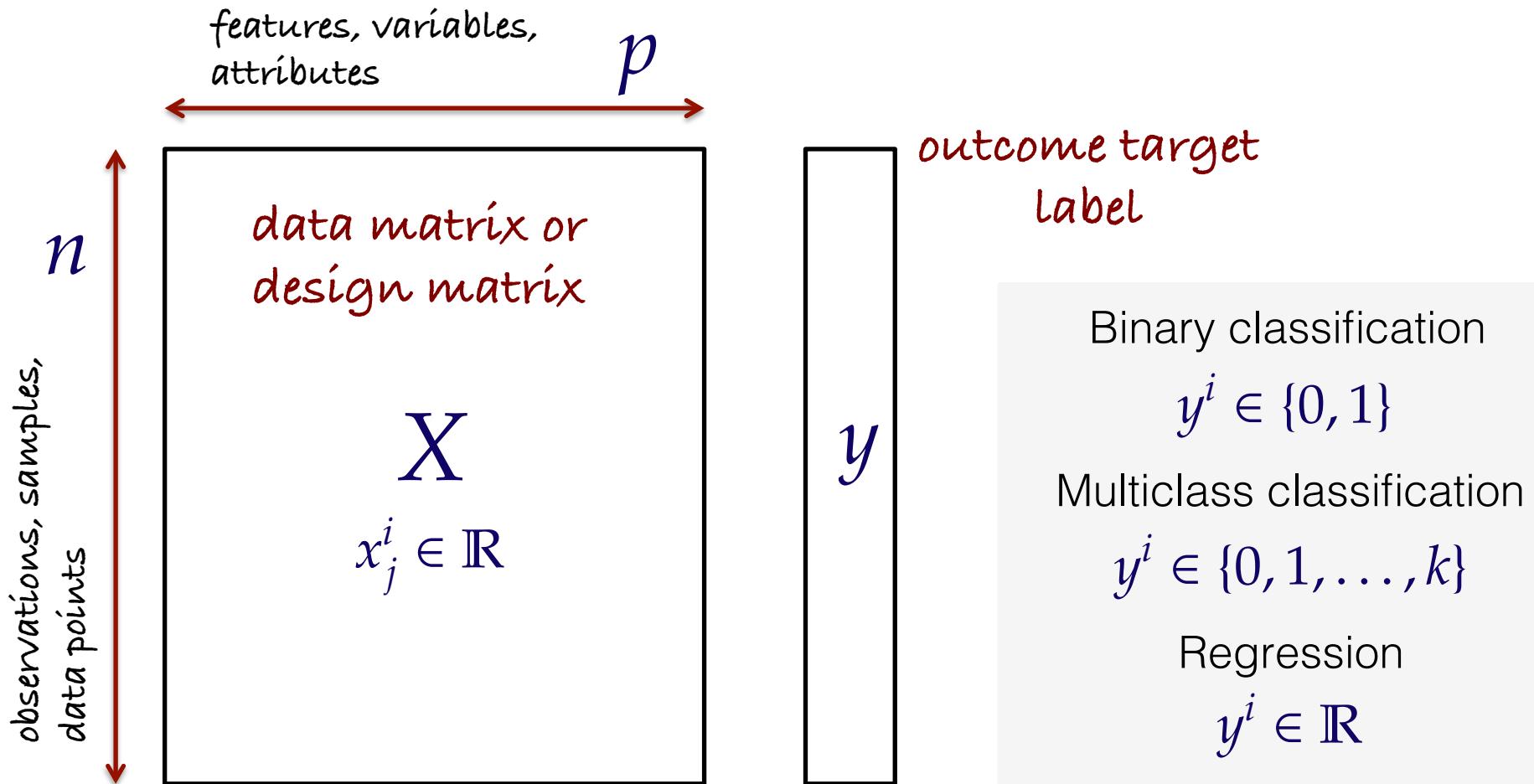
Given  $\mathcal{D} = \{x^i, y^i\}_{i=1,\dots,n}$  find  $f$  such that  $f(x) \approx y$

# Regression – Applications

- Click prediction
  - How many people will click on this ad? ... comment on this post? ... share this article on social media?
- Load prediction
  - How many users will my service have at a given time?
- Algorithmic trading
  - What will the price of this share be?

# Supervised Learning Setting – Summary

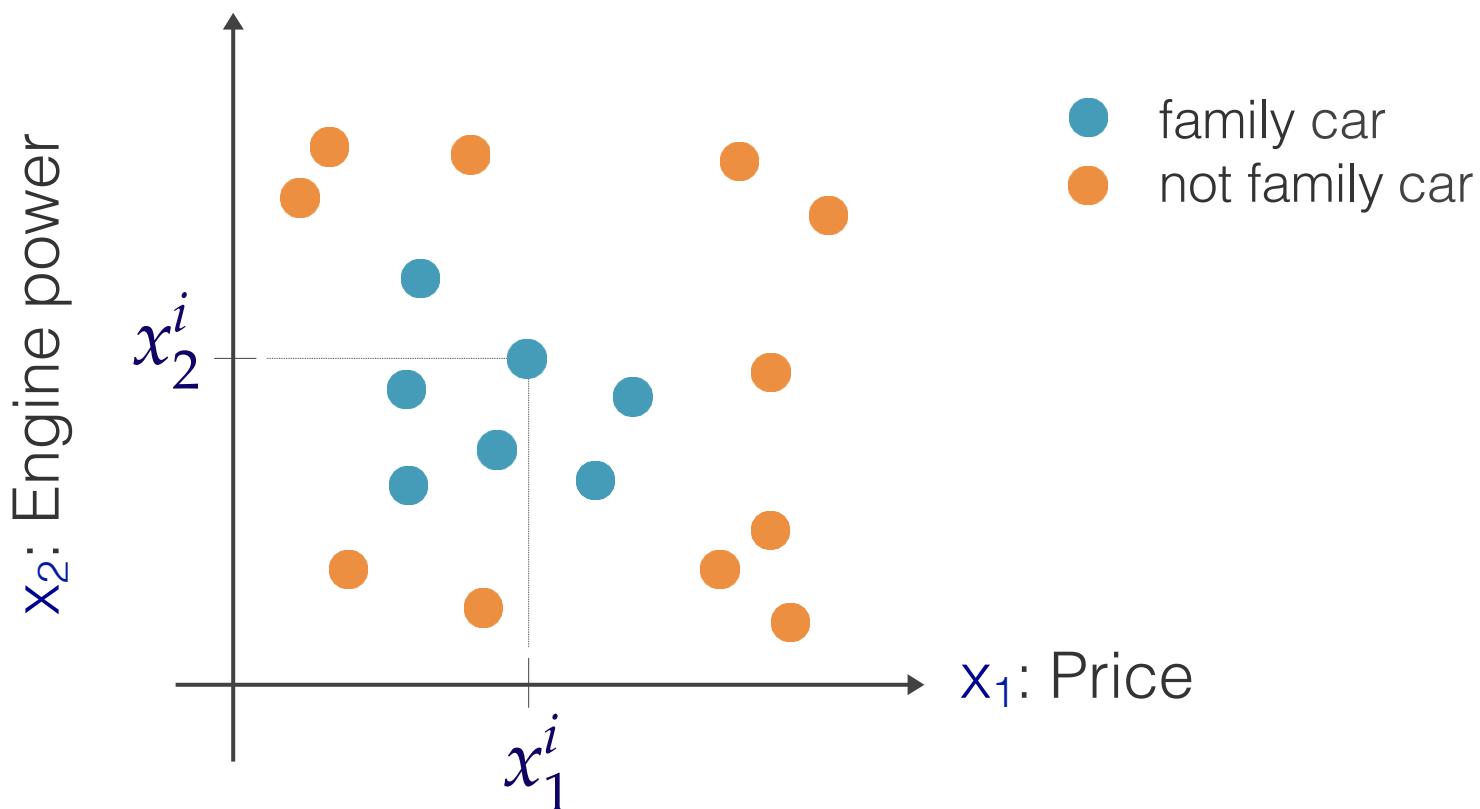
Given  $\mathcal{D} = \{x^i, y^i\}_{i=1,\dots,n}$  find  $f$  such that  $f(x) \approx y$



# Hypothesis class, loss function, and risk minimization

# Hypothesis Class

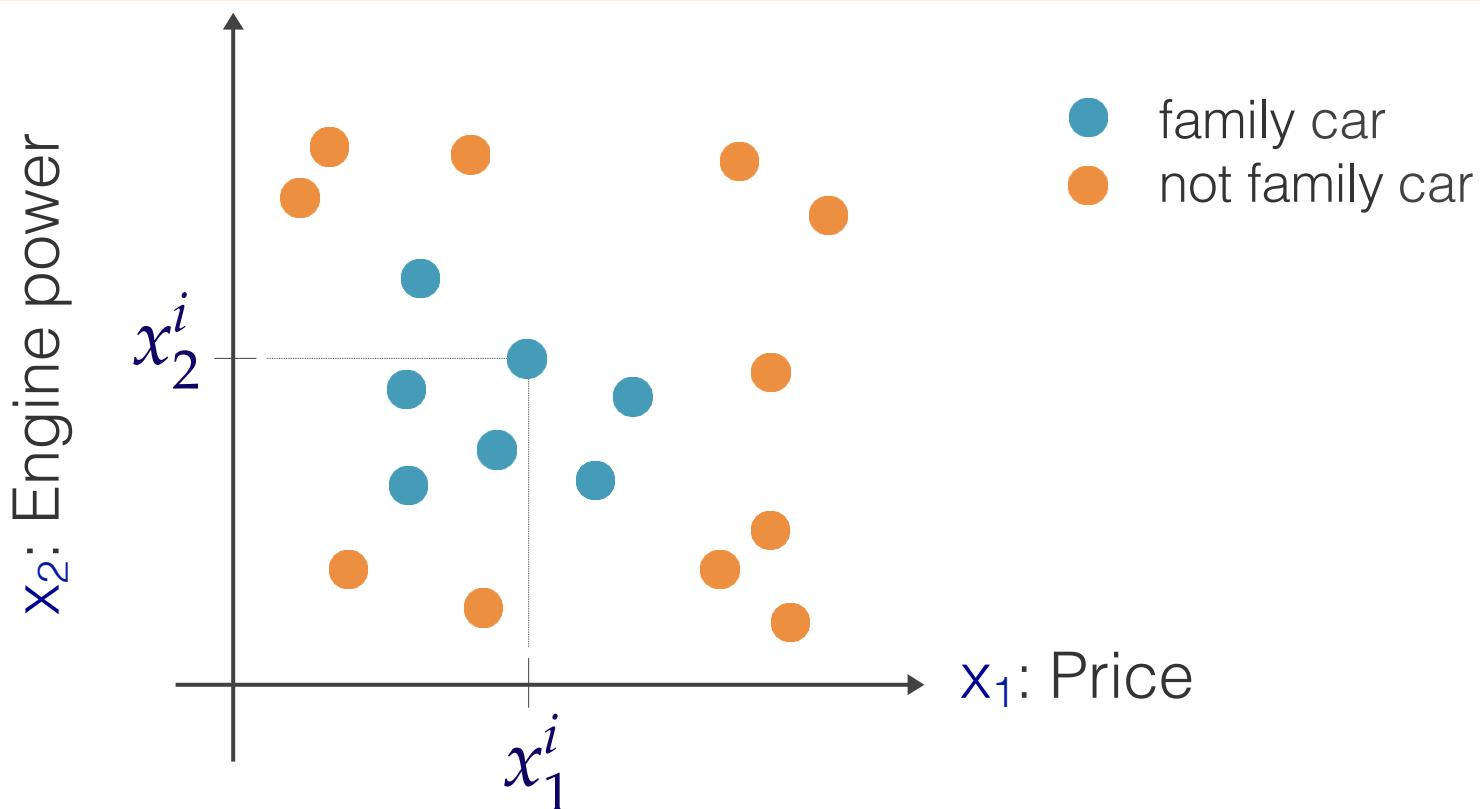
- Hypothesis class  $\mathcal{F}$ 
  - The **space of possible decision functions** we are considering
  - Chosen based on our **beliefs** about the problem



# Hypothesis Class

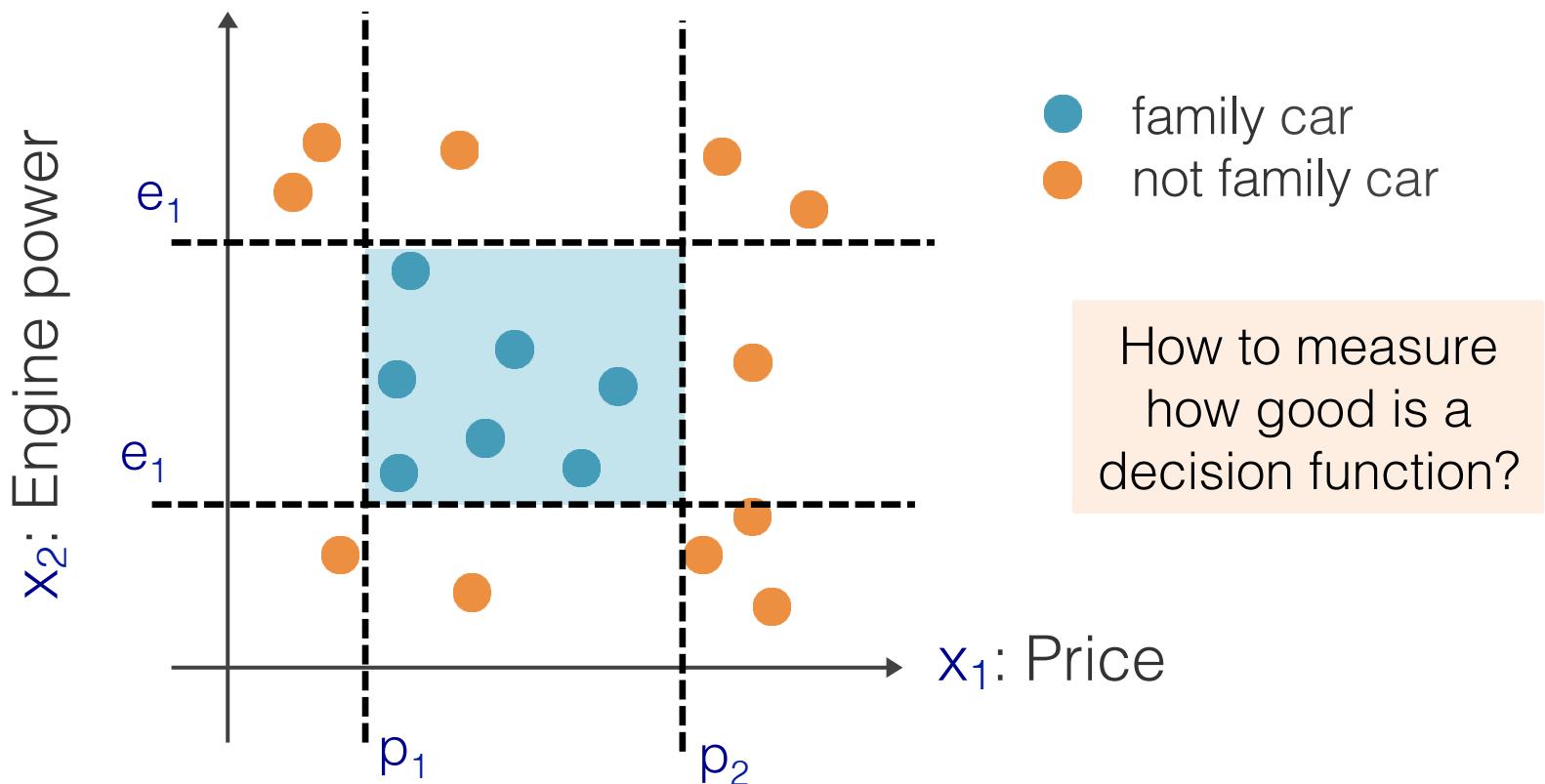
- Hypothesis class  $\mathcal{F}$

What shape do you think the discriminant should take?



# Hypothesis Class

- Hypothesis class  $\mathcal{F}$ 
  - Belief: the decision function is a rectangle
$$(p_1 \leq x_1 \leq p_2) \text{ AND } (e_1 \leq x_2 \leq e_2)$$



# Loss Function

- Loss function (or cost function, or risk):

$$\mathcal{L} : \mathcal{Y} \times \mathcal{Y} \rightarrow \mathbb{R}$$

$$y, f(x) \rightarrow \mathcal{L}(y, f(x))$$

Quantifies how far  
the decision function  
is from the truth

Example of loss  
functions:

$$\mathcal{Y} = \{0, 1\} \quad \mathcal{L}(y, f(x)) = \begin{cases} 0 & \text{if } y = f(x) \\ 1 & \text{otherwise} \end{cases}$$

$$\mathcal{Y} = \mathbb{R} \quad \mathcal{L}(y, f(x)) = \|y - f(x)\|^2$$

Training via optimization: find that  $f$  among the hypothesis class  $\mathcal{F}$  that minimizes the total loss

# The Goal of Training (1/4)

- What we have: labeled examples presented as (observations, label)

$$\mathbf{d}_i = (\mathbf{x}^i, y^i)$$

- E.g., observation = image, label = “cat” (-1), or “dog” (+1)



- **Assumption:** all labeled (train/test) examples come from unknown distribution, say  $\mathbf{D}$

# The Goal of Training (2/4)

- What we have:  $n$  labeled examples drawn *i.i.d.* from  $D$

$$(\mathbf{x}^i, y^i), \dots, (\mathbf{x}^n, y^n)$$

- What we want: train a model (a predictor function  $f$ )

$$f : \text{feature vector} \rightarrow \text{label}$$

That performs well on unseen data

- For example:

$$f\left( \begin{array}{c} \text{[Image of a dog]} \end{array} \right) = +1$$

# The Goal of Training (3/4)

- What we have:  $n$  labeled examples drawn *i.i.d.* from  $D$

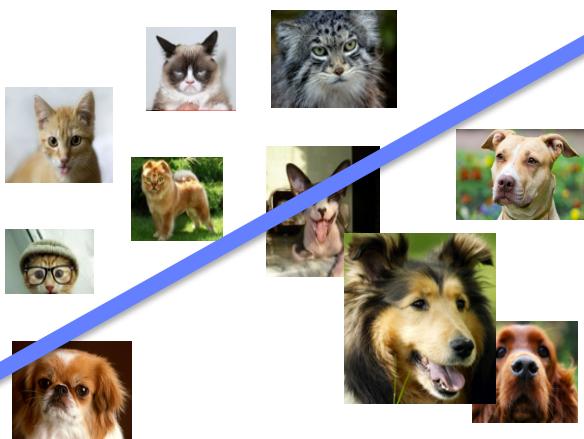
$$(\mathbf{x}^i, y^i), \dots, (\mathbf{x}^n, y^n)$$

- What we want: train a model (a predictor function  $f$ )

$f$  : feature vector  $\rightarrow$  label

That performs well on unseen data

- For example:



Our trained predictor

# The Goal of Training (4/4)

- How to measure performance? As we said before, we need to define the **loss**:

$$\mathcal{L}(y, f(\mathbf{x}))$$

Measures disagreement  
between  
predicted and true label

- **Goal:** we want a predictor  $f$  with small loss on **unseen data** (e.g., on a test set)

$$\sum_{(\mathbf{x}, y) \text{ is an unseen example}} \mathcal{L}(y, f(\mathbf{x}))$$

But, we haven't seen unseen examples  
(the test set is not known to the learning algorithm)

# Empirical Risk

- The loss on the “unseen” examples converges to the expected loss

$$\sum_{(\mathbf{x}, y) \text{ is an unseen example}} \mathcal{L}(y, f(\mathbf{x})) \rightarrow \mathbb{E}_{\mathbf{x}, y} \{ \mathcal{L}(y, f(\mathbf{x})) \}$$

Expected/True risk

- What else converges to the expected loss?

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

Empirical risk

By averaging the loss function on  
the training set

# Empirical Risk Minimization (ERM)

- Training via optimization: we want to solve:

$$\arg \min_{f \in \mathcal{F}} \mathbb{E}\{\mathcal{L}(y, f(x))\}$$

$f$  can be:

- separating hyperplanes
- NN's of depth-t
- ...

- We instead solve the ERM:

$$\arg \min_{f \in \mathcal{F}} \frac{1}{n} \sum_{i=1}^n \mathcal{L}(y^i, f(\mathbf{x}^i))$$

# Supervised Learning: 3 Ingredients

Given  $\mathcal{D} = \{x^i, y^i\}_{i=1,\dots,n}$  find  $\mathbf{f}$  such that  $f(x) \approx y$

- Chose a hypothesis class  $\mathcal{F}$ 
  - Parametric methods – e.g.,  $f(x) = \sum_{j=1}^p \beta_j x_j$
  - Non-parametric methods – e.g.,  $\mathbf{f}(x)$  is the label of the point closest to  $\mathbf{x}$  (Nearest Neighbors is such a method)
- Chose a loss function  $\mathcal{L}$ 
  - Empirical error:  $\frac{1}{n} \sum_{i=1}^n \mathcal{L}(y^i, f(\mathbf{x}^i))$
- Chose an optimization procedure

$$f^* = \arg \min_{f \in \mathcal{F}} \frac{1}{n} \sum_{i=1}^n \mathcal{L}(y^i, f(\mathbf{x}^i))$$

# Optimization

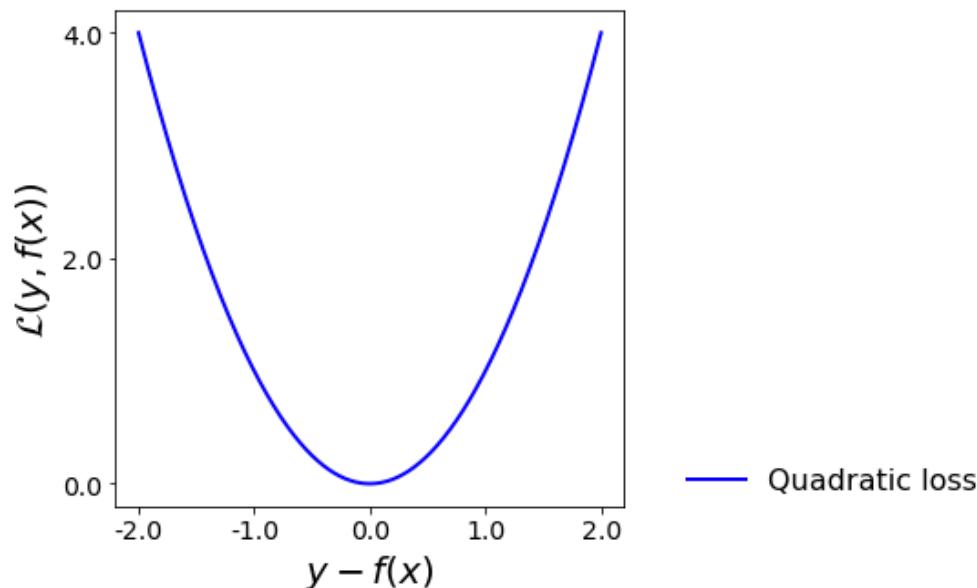
# Why Optimization? (1/4)

- Empirical risk minimization:

$$f^* = \arg \min_{f \in \mathcal{F}} \frac{1}{n} \sum_{i=1}^n \mathcal{L}(y^i, f(\mathbf{x}^i))$$

Which loss functions  
are mainly used?

- Quadratic loss:  $\mathcal{L}(y, f(\mathbf{x})) = (y - f(\mathbf{x}))^2$



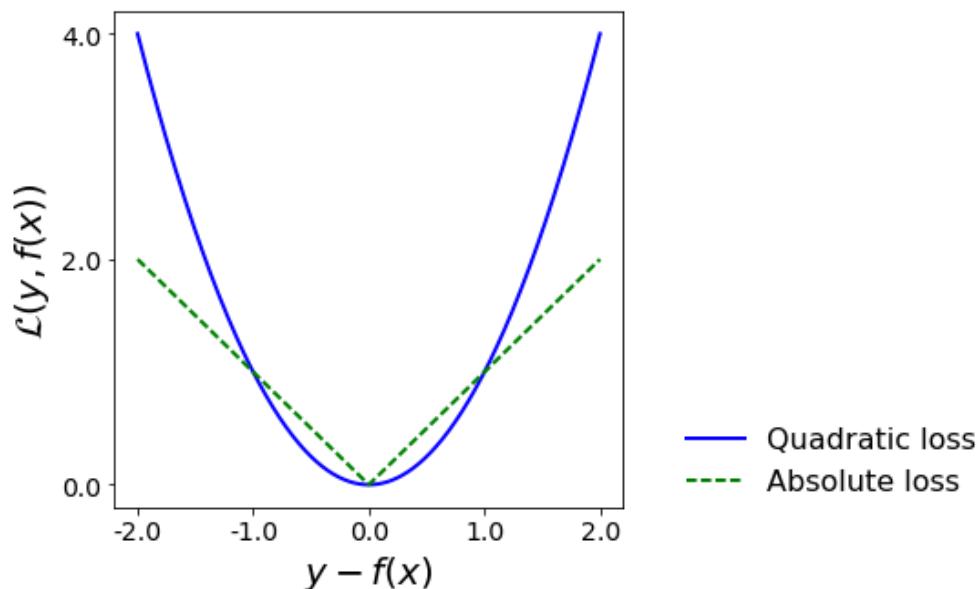
# Why Optimization? (2/4)

- Empirical risk minimization:

$$f^* = \arg \min_{f \in \mathcal{F}} \frac{1}{n} \sum_{i=1}^n \mathcal{L}(y^i, f(\mathbf{x}^i))$$

Which loss functions  
are mainly used?

- Absolute loss:  $\mathcal{L}(y, f(\mathbf{x})) = |y - f(\mathbf{x})|$



# Why Optimization? (3/4)

- Empirical risk minimization:

$$f^* = \arg \min_{f \in \mathcal{F}} \frac{1}{n} \sum_{i=1}^n \mathcal{L}(y^i, f(\mathbf{x}^i))$$

Which loss functions  
are mainly used?

- 0/1 loss:

$$\mathcal{L}(y, f(\mathbf{x})) = \begin{cases} 0 & \text{if } y = f(\mathbf{x}) \\ 1 & \text{otherwise} \end{cases}$$

# Why Optimization? (4/4)

- Unsupervised machine learning also involves optimization algorithms
- Some examples
  - Dimensionality reduction: find a set of  $n$  features,  $n < p$ , such that the data projected on these  $n$  features retains maximal information (e.g., PCA maximizes the total variance of the data)
  - Clustering: find  $k$  groups of samples such that the between-groups variance is high and the within-group variance is small

Both topics will be covered in next lecture

# Convex Set

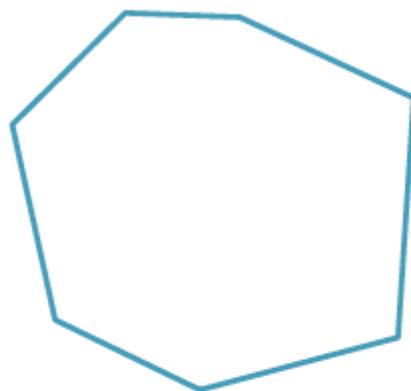
- $S \subseteq \mathbb{R}^n$  is a convex set iff:

$$t\mathbf{u} + (1 - t)\mathbf{v} \in S$$

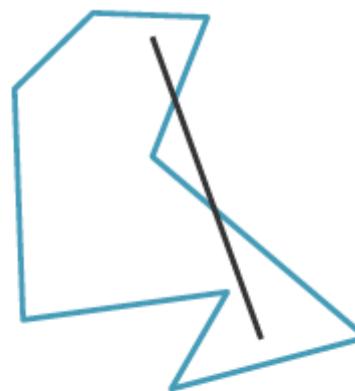
convex combination

for all  $\mathbf{u}, \mathbf{v}$  in  $S$  and  $0 \leq t \leq 1$

- Line segments between two points of  $S$  lie entirely in  $S$



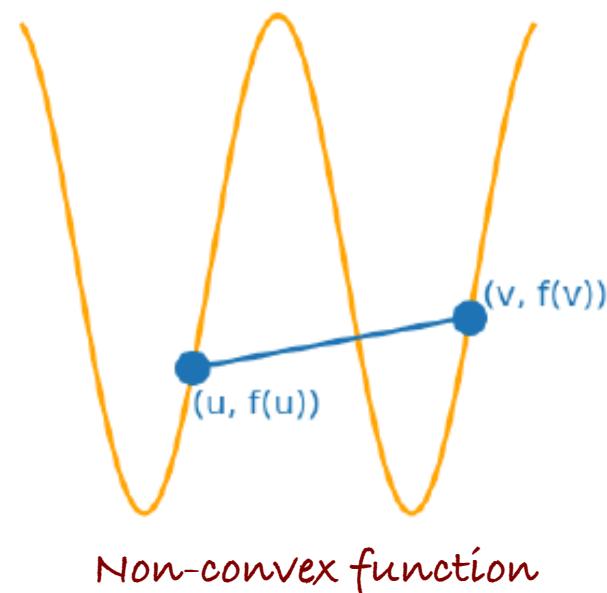
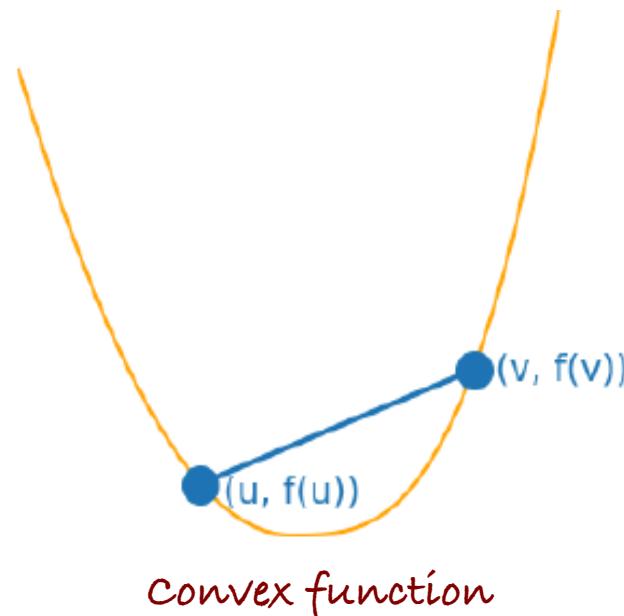
Convex set



Non-convex set

# Convex Function

- $f : \mathbb{R}^n \rightarrow \mathbb{R}$  is convex iff:
  1. Its domain is a convex set
  2.  $f(t\mathbf{u} + (1 - t)\mathbf{v}) \leq tf(\mathbf{u}) + (1 - t)f(\mathbf{v})$  for all  $\mathbf{u}, \mathbf{v}$  in  $\text{dom}(f)$  and  $0 \leq t \leq 1$
- $f$  lies below the line segment joining  $f(\mathbf{u})$  and  $f(\mathbf{v})$



# Concave Function

- $f : \mathbb{R}^n \rightarrow \mathbb{R}$  is concave iff:
  1. Its domain is a convex set
  2.  $f(t\mathbf{u} + (1 - t)\mathbf{v}) \geq tf(\mathbf{u}) + (1 - t)f(\mathbf{v})$  for all  $\mathbf{u}, \mathbf{v}$  in  $\text{dom}(f)$  and  $0 \leq t \leq 1$

$f$  concave  $\Leftrightarrow -f$  convex

# First-order Characterization of Convex Functions

- If  $f$  is differentiable, then  $f$  is convex if and only if:
  - Its domain is a convex set
  - For all  $\mathbf{u}, \mathbf{v} \in \text{dom}(f)$

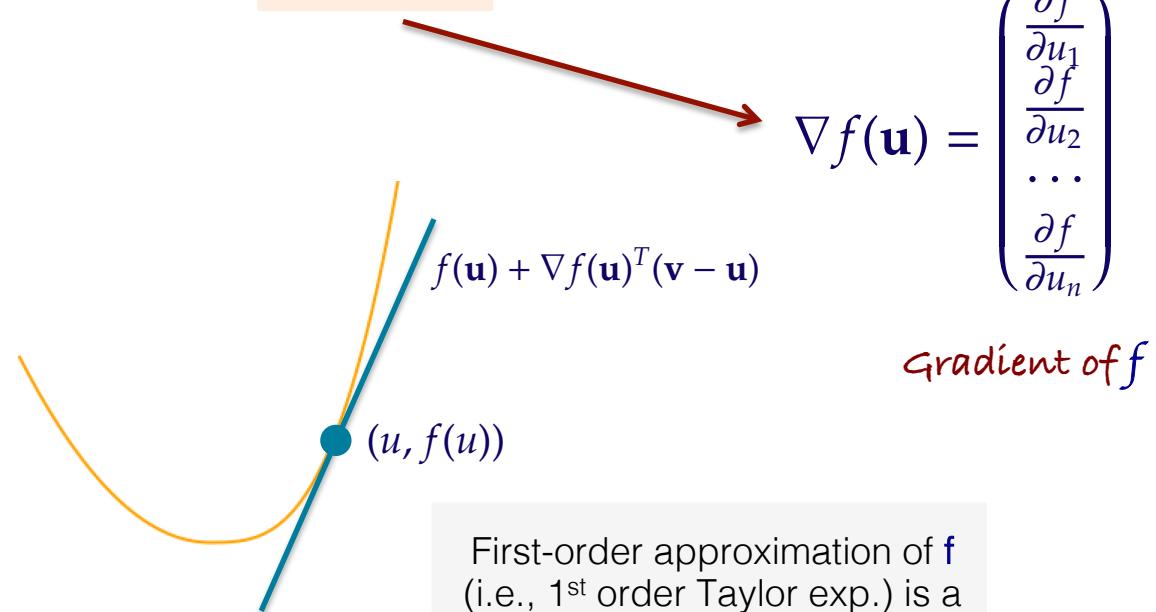
First-order Taylor expansion

$$f(\mathbf{v}) \geq f(\mathbf{u}) + \nabla f(\mathbf{u})^T (\mathbf{v} - \mathbf{u})$$

What does it mean if:

$$\nabla f(\mathbf{u}) = 0$$

$\mathbf{u}$  minimizes  $f$



# Second-order Characterization

- If  $f$  is twice-differentiable, then  $f$  is convex if and only if:
    - Its domain is a convex set
    - For all  $\mathbf{u}, \mathbf{v} \in \text{dom}(f)$
- Hessian of  $f$ ,  $H_f$
- $$\nabla^2 f(\mathbf{u}) \geq 0$$
- $$\nabla^2 f = \begin{pmatrix} \frac{\partial^2 f}{\partial u_1^2} & \frac{\partial^2 f}{\partial u_1 \partial u_2} & \cdots & \frac{\partial^2 f}{\partial u_1 \partial u_n} \\ \frac{\partial^2 f}{\partial u_2 \partial u_1} & \frac{\partial^2 f}{\partial u_2^2} & \cdots & \frac{\partial^2 f}{\partial u_2 \partial u_n} \\ \vdots & & \ddots & \\ \frac{\partial^2 f}{\partial u_n \partial u_1} & \frac{\partial^2 f}{\partial u_n \partial u_2} & \cdots & \frac{\partial^2 f}{\partial u_n^2} \end{pmatrix}$$
- Let  $f$  be a twice-differentiable function at  $\mathbf{u}$ 
    - If  $\nabla^2 f(\mathbf{u}) = H_f(\mathbf{u})$  is positive definite, then  $\mathbf{u}$  is a local minimum
    - If  $\nabla^2 f(\mathbf{u}) = H_f(\mathbf{u})$  is negative definite, then  $\mathbf{u}$  is a local maximum
  - Let  $S \subseteq \mathbb{R}^n$  be convex, and  $f$  be twice-differentiable on  $S$ 
    - If  $H_f(\mathbf{u})$  is positive semi-definite for any  $\mathbf{u} \in S$ , then  $f$  is convex on  $S$

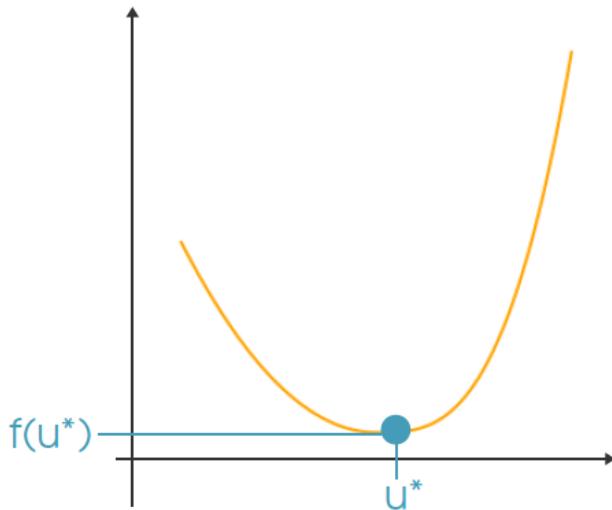
# Operations Preserving Convexity

- Non-negative linear combination
  - If  $f_1, f_2, \dots, f_m$  convex and  $\alpha_1, \alpha_2, \dots, \alpha_m \geq 0$   
then  $\alpha_1 f_1 + \alpha_2 f_2 + \dots + \alpha_m f_m$  is also convex
- Pointwise maximization
  - If  $f_1, f_2, \dots, f_m$  are convex  
then  $f(\mathbf{u}) = \max\{f_1(\mathbf{u}), \dots, f_m(\mathbf{u})\}$  is also convex
- Composition with scalar functions
  - Composition of  $g : \mathbb{R}^n \rightarrow \mathbb{R}$  and  $h : \mathbb{R} \rightarrow \mathbb{R}$   
$$f(\mathbf{u}) = h(g(\mathbf{u}))$$
  
 $f$  is convex if  $g$  is convex and  $h$  is convex

# Unconstrained Convex Optimization

$$\min_{\mathbf{u} \in \text{dom}(f)} f(\mathbf{u})$$

where  $f$  is convex



Any point  $\mathbf{u}^*$  that satisfies  $\nabla f(\mathbf{u}^*) = 0$  is a global minimum

- From the first-order characterization:

$$f(\mathbf{v}) \geq f(\mathbf{u}^*) + \nabla f^T(\mathbf{u}^*)(\mathbf{v} - \mathbf{u}^*), \forall \mathbf{v}$$

- Since  $\nabla f(\mathbf{u}^*) = 0$ , we get:

$$f(\mathbf{v}) \geq f(\mathbf{u}^*), \forall \mathbf{v}$$

# Constrained Convex Optimization (1/4)

$$\min_{\mathbf{u} \in D} f(\mathbf{u})$$

subject to  $g_i(\mathbf{u}) \leq 0, i = 1, \dots, m$

$$h_i(\mathbf{u}) = 0, i = 1, \dots, r$$

- $f$  is convex
- $g_i, i = 1, \dots, m$  are convex [inequality constraints]
- $h_i, i = 1, \dots, r$  are affine:  $h_j : \mathbf{u} \rightarrow \mathbf{a}_j^T \mathbf{u} + b_j$  [equality constraints]
- $D$  is the common domain of all functions

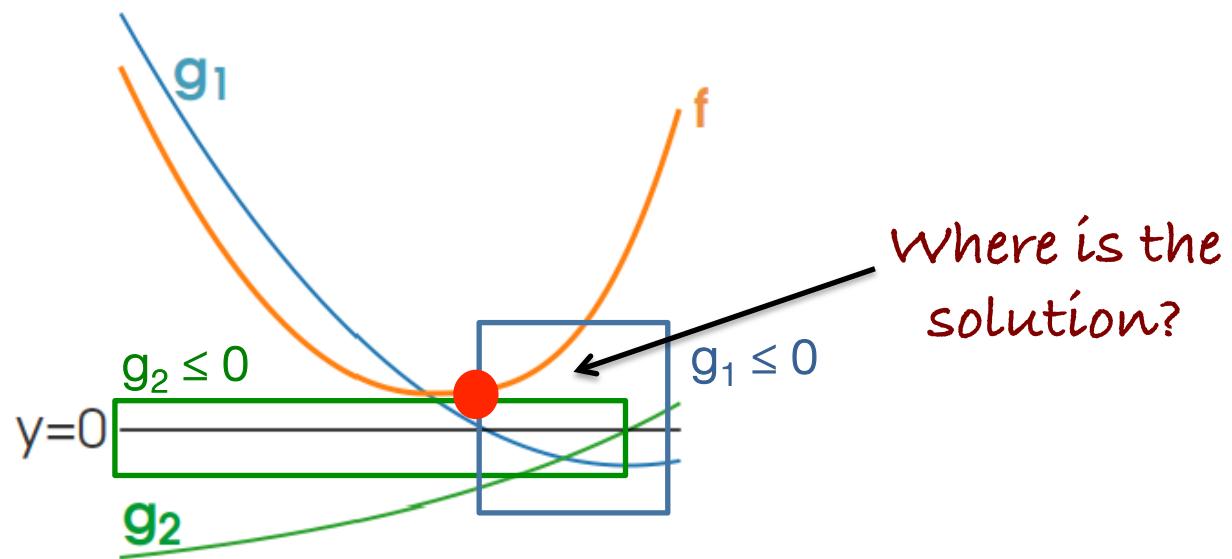
$$D = \text{dom}(f) \cap \bigcap_{i=1}^m \text{dom}(g_i) \cap \bigcap_{i=1}^r \text{dom}(h_i)$$

# Constrained Convex Optimization (2/4)

$$\min_{\mathbf{u} \in D} f(\mathbf{u})$$

subject to  $g_i(\mathbf{u}) \leq 0, i = 1, \dots, m$

$h_i(\mathbf{u}) = 0, i = 1, \dots, r$



# Constrained Convex Optimization (3/4)

$$\min_{\mathbf{u} \in D} f(\mathbf{u})$$

subject to  $g_i(\mathbf{u}) \leq 0, i = 1, \dots, m$

$$h_i(\mathbf{u}) = 0, i = 1, \dots, r$$

- $f$  is the **objective function**
- $g_i, i = 1, \dots, m$  are the **inequality constraints**
- $h_i, i = 1, \dots, r$  are the **equality constraints**
- $\mathbf{u} \in D$  that verifies all constraints is a **feasible point**  
 $g_i(\mathbf{u}) \leq 0, i = 1, \dots, m$  and  $h_j(\mathbf{u}) = 0, j = 1, \dots, r$
- The set of all feasible points is the **feasible region**  
 $\{\mathbf{u} : \mathbf{u} \in D; g_i(\mathbf{u}) \leq 0, i = 1, \dots, m; h_j(\mathbf{u}) = 0, j = 1, \dots, r\}$

# Constrained Convex Optimization (4/4)

$$\min_{\mathbf{u} \in D} f(\mathbf{u})$$

subject to  $g_i(\mathbf{u}) \leq 0, i = 1, \dots, m$

$$h_i(\mathbf{u}) = 0, i = 1, \dots, r$$

- Assuming it exists, the solution  $\mathbf{f}^*$ , that is to say the minimum value of  $\mathbf{f}$  over all feasible points, is the **optimal value (optimum)**
- A  $\mathbf{u}$  feasible such that  $f(\mathbf{u}) = f^*$  is called **optimal** or a **minimizer** (it doesn't need to be unique)

# Local and Global Minima

For convex optimization problems

$$\begin{aligned} & \min_{\mathbf{u} \in D} f(\mathbf{u}) \\ \text{subject to } & g_i(\mathbf{u}) \leq 0, i = 1, \dots, m \\ & h_i(\mathbf{u}) = 0, i = 1, \dots, r \end{aligned}$$

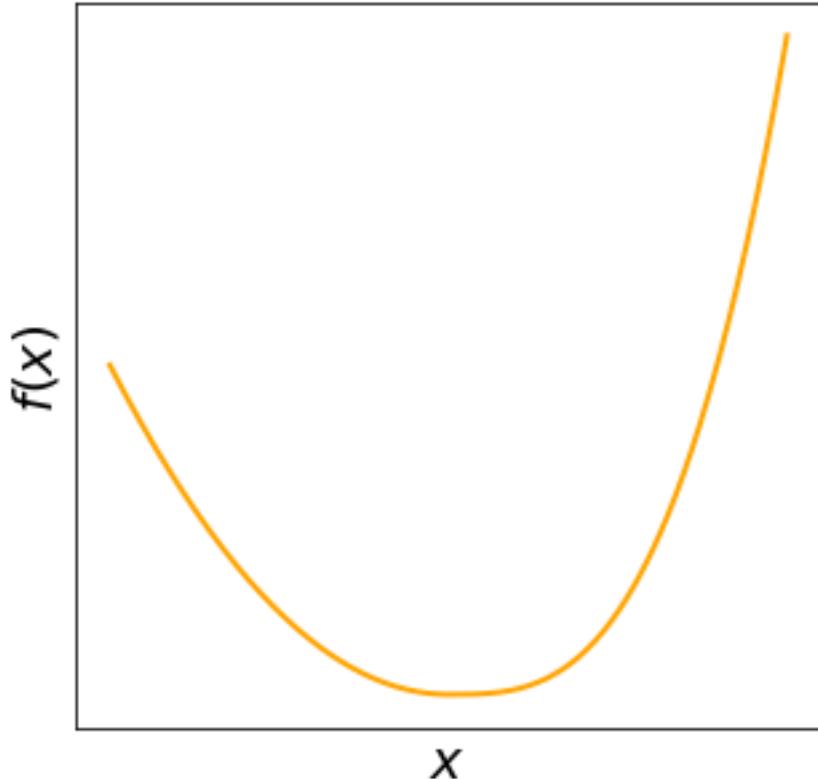
Local minima are global minima

If  $\mathbf{u}$  is a feasible solution and minimizes  $f$  in a local neighborhood:

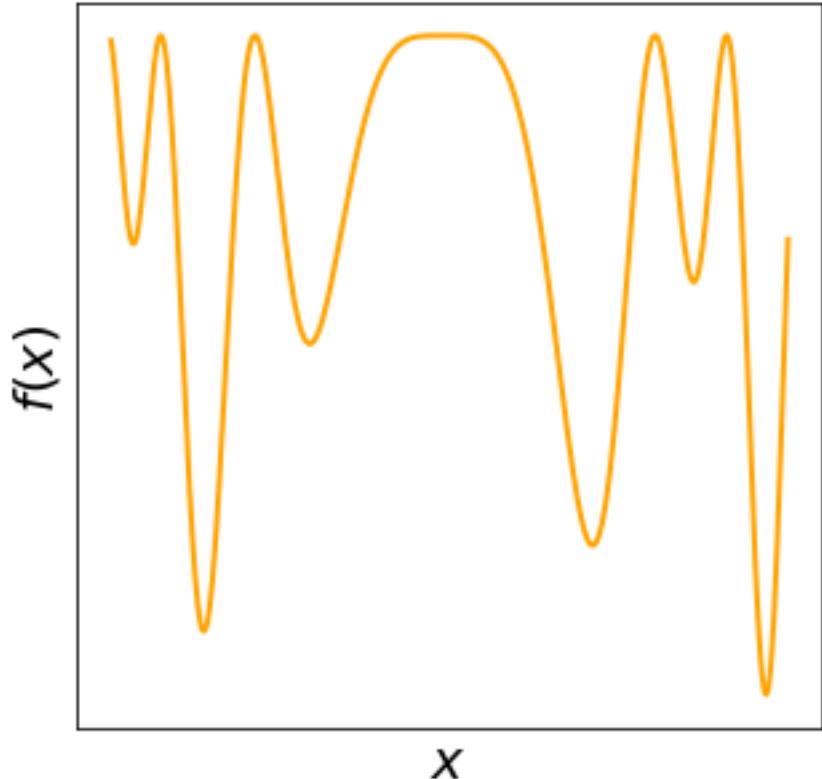
$f(\mathbf{u}) \leq f(\mathbf{v})$  for all feasible  $\mathbf{v}$ ,

then  $\mathbf{u}$  minimizes  $f$  globally

# Why Talk about Convex Optimization?



Convex function



Non-convex function

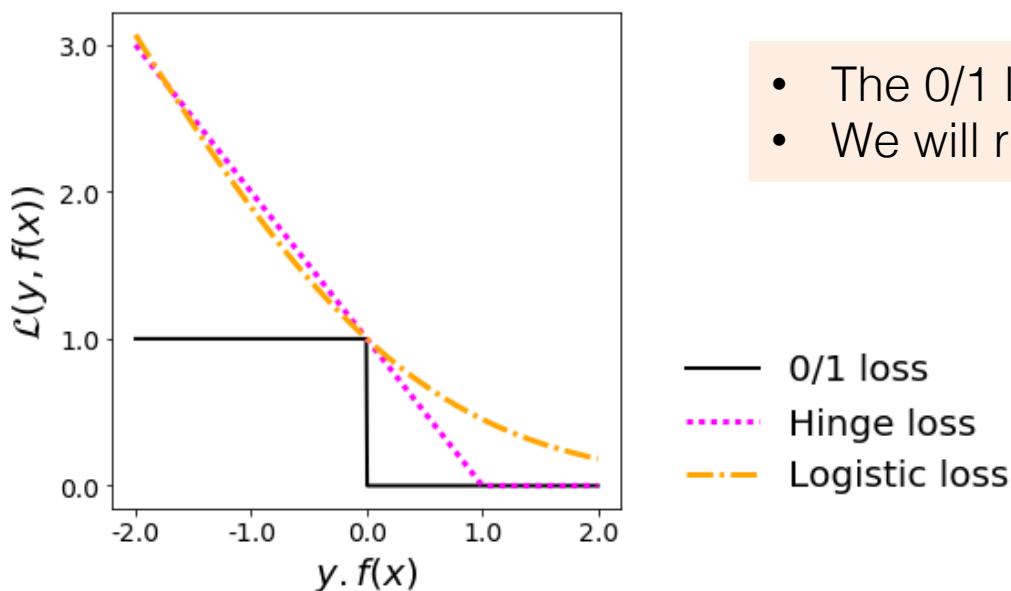
- Convex optimization is easy
- We will often try to formulate ML problems as convex optimization problems

# Back to the ERM Problem

- Empirical risk minimization:

$$f^* = \arg \min_{f \in \mathcal{F}} \frac{1}{n} \sum_{i=1}^n \mathcal{L}(y^i, f(\mathbf{x}^i))$$

- Loss functions for classification



- The 0/1 loss function is non-convex
- We will replace it with other losses

# Unconstrained Convex Optimization

- Suppose that  $f$  is differentiable
- Given the first-order characterization of convex functions, how can we solve  $\min_{\mathbf{u} \in \text{dom}(f)} f(\mathbf{u})$ ?

Recall that:

$$\nabla f(\mathbf{u}) = 0$$



$\mathbf{u}$  minimizes  $f$

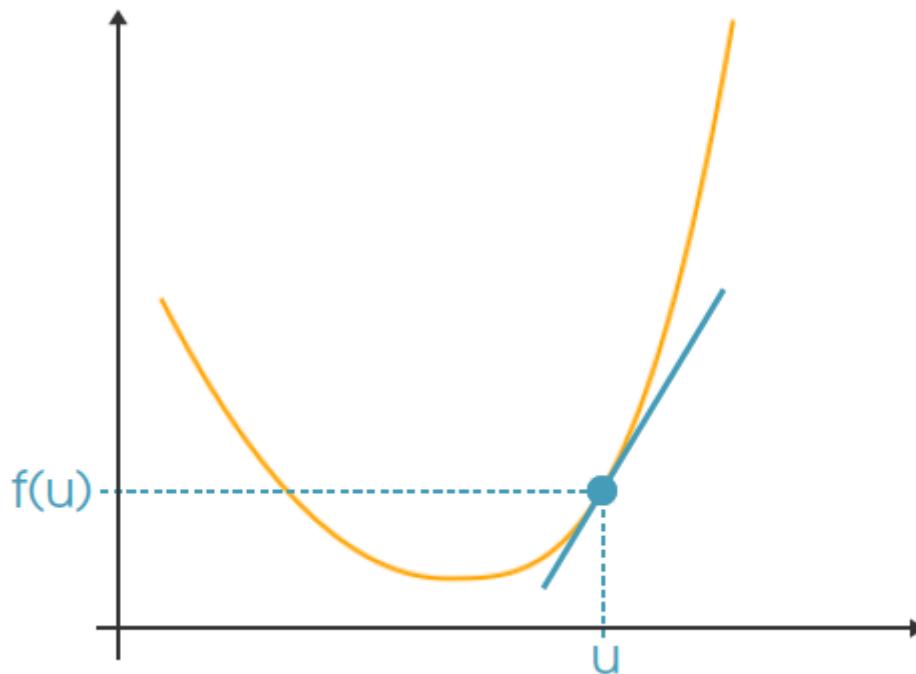


Set the gradient of  $f$  to 0

But, what if  $\nabla f(\mathbf{u}) = 0$  cannot be solved?

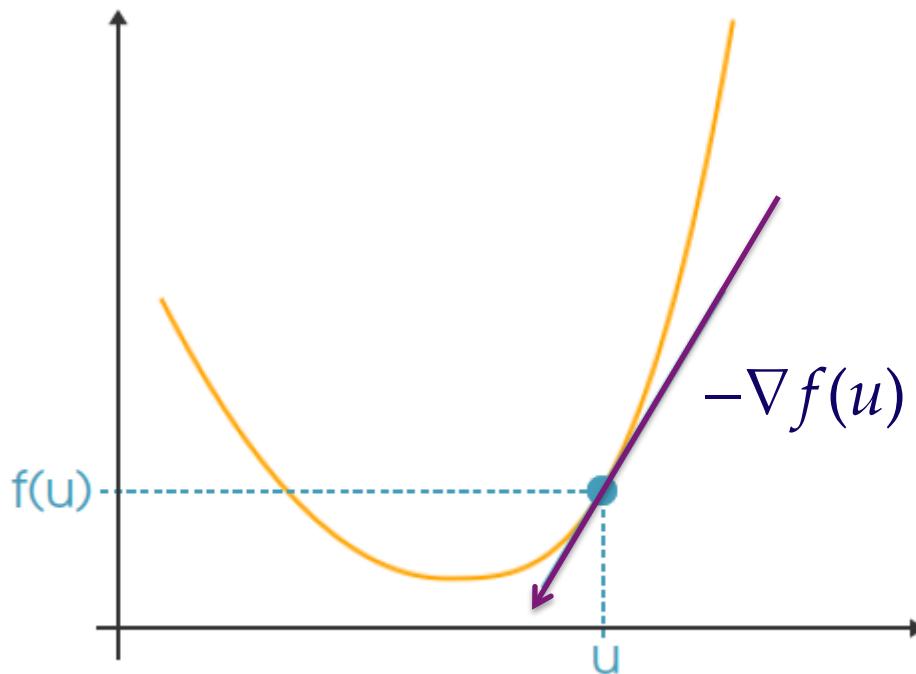
# Gradient Descent – The Idea (1/3)

- Start from a random point  $u$
- How do I get closer to the solution?



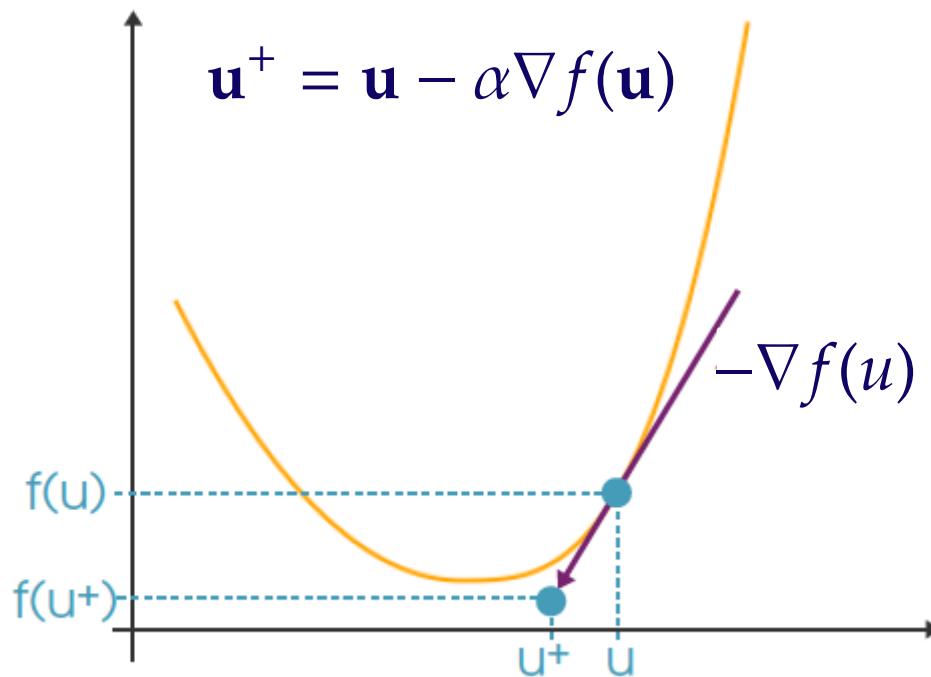
# Gradient Descent – The Idea (2/3)

- Start from a random point  $u$
- How do I get closer to the solution?
- Follow the opposite of the gradient
  - The gradient indicates the direction of the steepest descent



# Gradient Descent – The Idea (3/3)

- Start from a random point  $\mathbf{u}$
- How do I get closer to the solution?
- Follow the opposite of the gradient
  - The gradient indicates the direction of the steepest descent



# Gradient Descent Algorithm

- Choose an initial point  $\mathbf{u}^{(0)} \in \mathbb{R}^n$
- Repeat for  $k=1, 2, 3, \dots$

$$\mathbf{u}^{(k)} = \mathbf{u}^{(k-1)} - \boxed{\alpha_k} \nabla f(\mathbf{u}^{(k-1)})$$

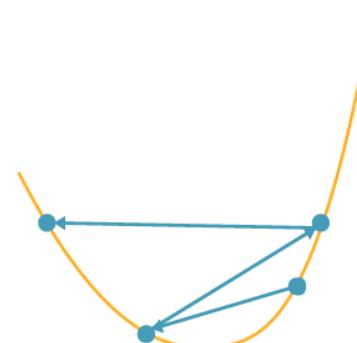
step size

- Stop at some point

Stopping criterion: Usually when  $\|\nabla f(\mathbf{u}^{(k)})\|_2 < \epsilon$ ,  $\epsilon$  = small constant

If the step is too big, the search might diverge

If the step is too small, the search might take very long time



# About the project

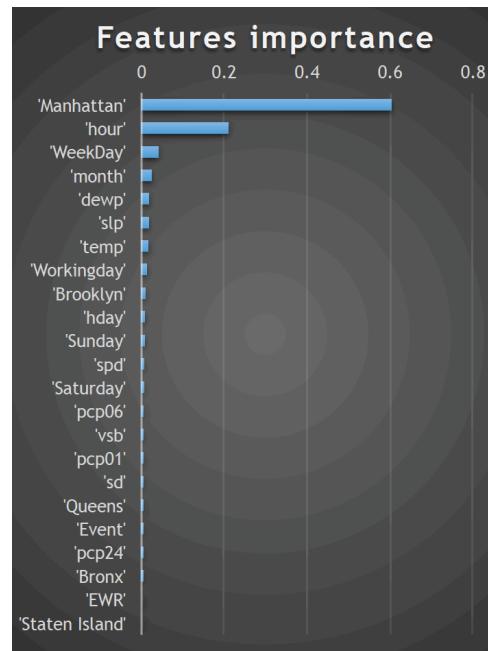
# Examples of Previous Projects (1/5)

How can Uber predict the demand for pickups?

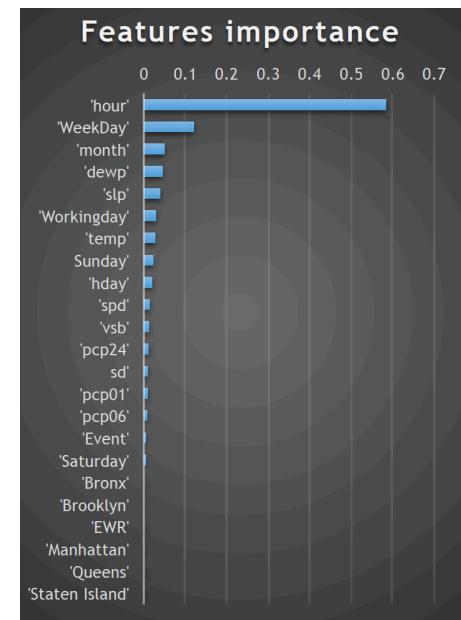
- Dataset: 34,000 rides per day in NYC
- Prediction problem

WEATHER	EVENTS	DAYS	BOROUGHS
TEMPERATURE	CONCERTS	TIME	BRONX
WIND SPEED	MUSIC FESTIVALS	DAY / MONTH	BROOKLYN
VISIBILITY		HOLIDAYS	EWR
PRECIPITATION			MANHATTAN
SEA LEVEL			QUEENS
DEW POINT			STATEN ISLAND
SNOW DEPTH			

Dataset description



For the whole dataset

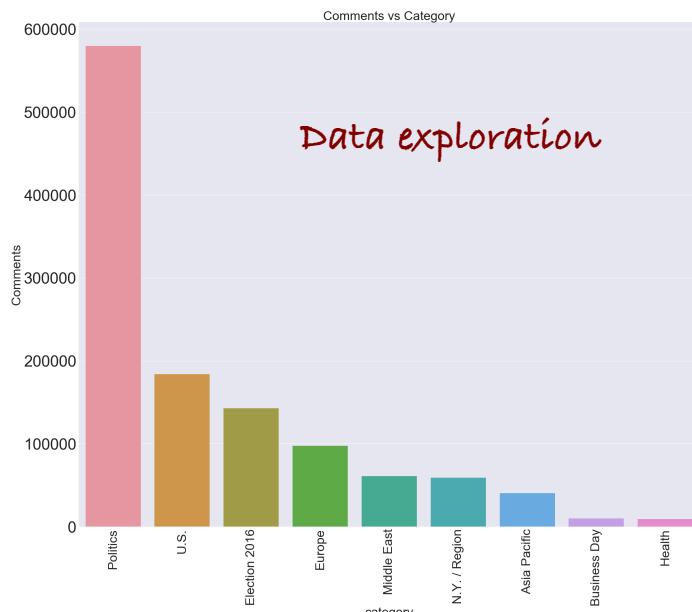


For a particular district  
in NYC (Borough)

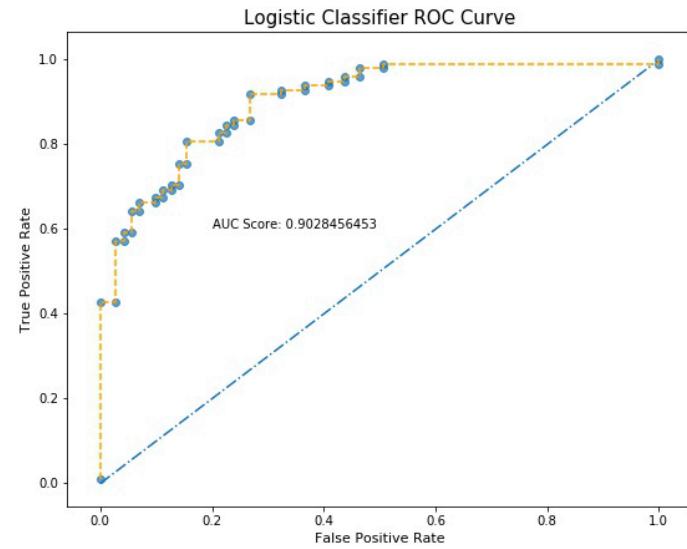
# Examples of Previous Projects (2/5)

## Analyzing New York Times' readers engagement

- Problem: predict if an article will be commented or not
- Data: 1,280 articles and 280 authors' data
- Features: article length, images, videos, content, authors' information, etc.



Top-10 most commented articles

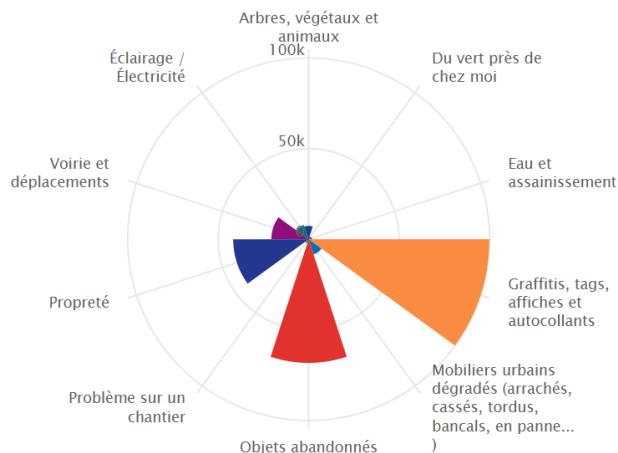
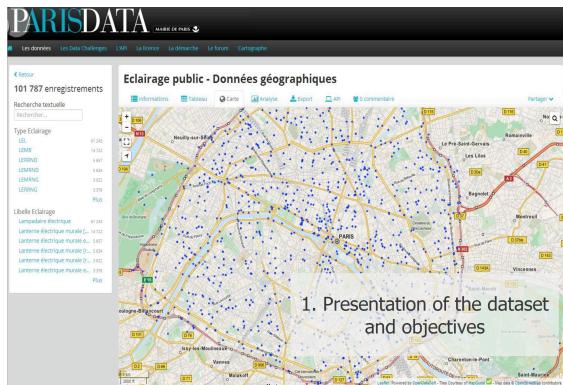


Prediction results (logistic regression)

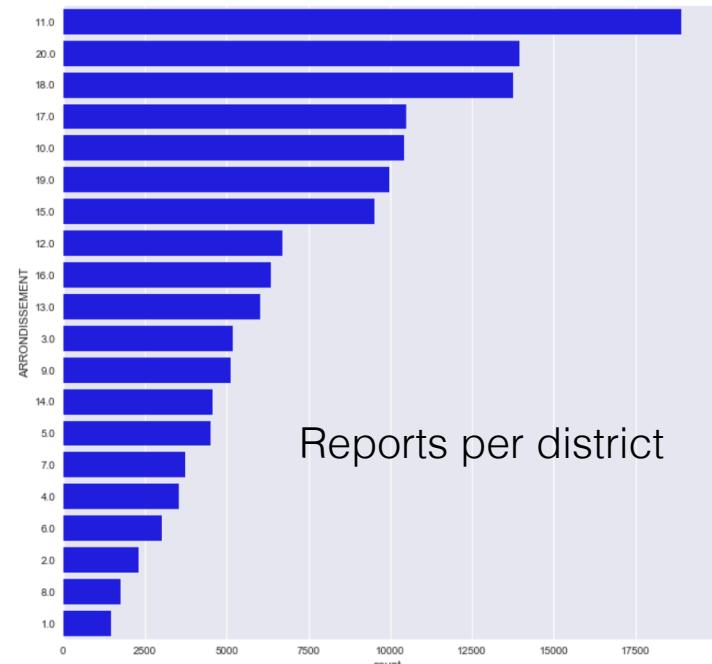
# Examples of Previous Projects (3/5)

## Predict reports made by citizens in Paris

- Data: open Paris data



Distribution of reports by type



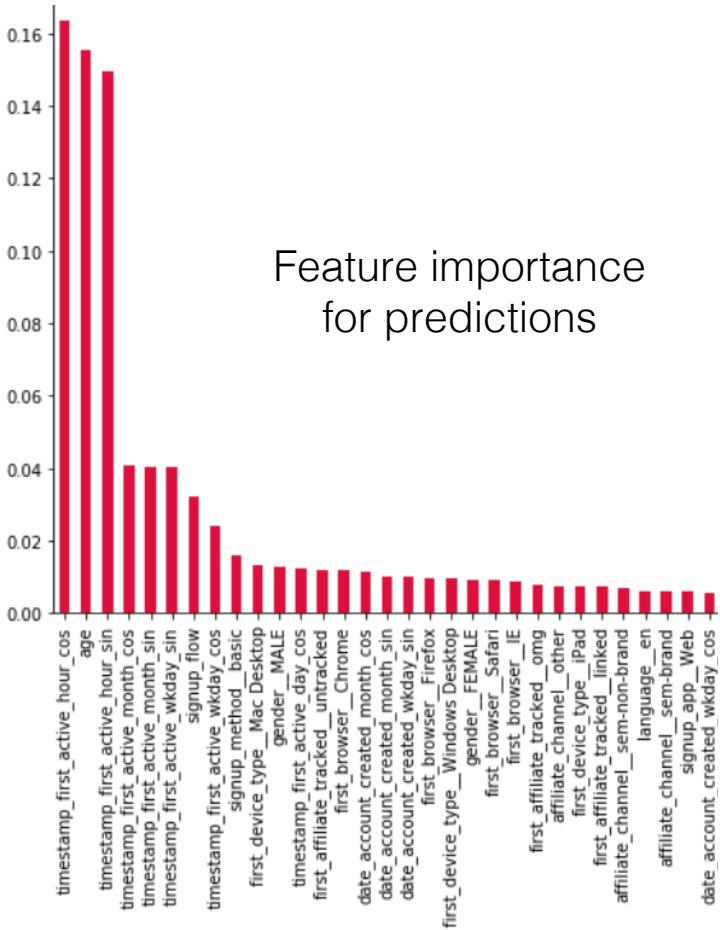
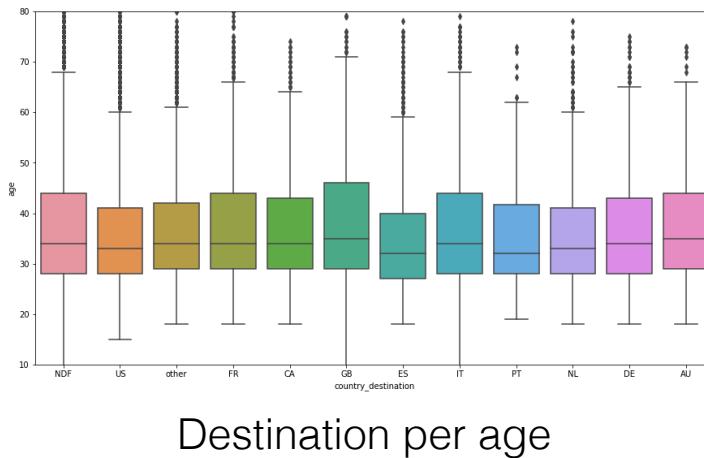
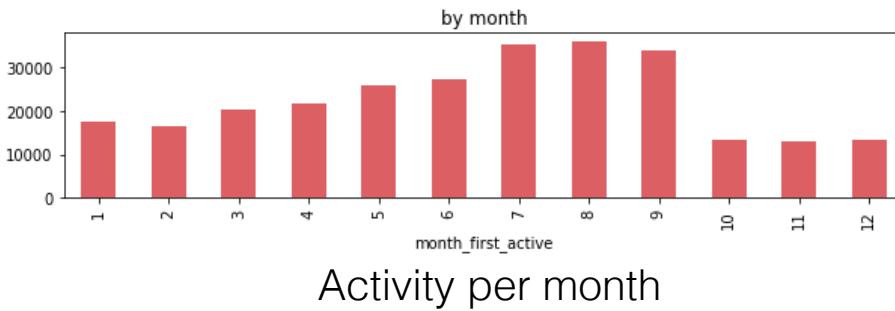
	precision	recall	f1-score	support
Arbres, végétaux et animaux	0.48	0.02	0.03	634
Du vert près de chez moi	0.47	0.15	0.23	277
Eau et assainissement	1.00	0.01	0.02	124
Graffitis, tags, affiches et autocollants	0.54	0.86	0.66	11314
Mobiliers urbains dégradés	0.52	0.05	0.10	1067
Objets abandonnés	0.57	0.58	0.58	7286
Problème sur un chantier	0.00	0.00	0.00	229
Propreté	0.42	0.20	0.27	4120
Voirie et déplacements	0.40	0.08	0.13	2301
Éclairage / Électricité	0.42	0.04	0.07	909
avg / total	0.51	0.54	0.47	28261

Prediction results  
(random forest)

# Examples of Previous Projects (4/5)

## Airbnb destination prediction

- Data: 214,000 users (USA), 11 countries
- Features: age, gender, language, important dates



# Examples of Previous Projects (5/5)

## Pokemon battle prediction

- Problem: predict the outcome of a pokemon fight

	First_pokemon	Second_pokemon	Winner
0	266	298	298
1	702	701	701
2	191	668	668
3	237	683	683
4	151	231	151

- Data:

#	Name	Type 1	Type 2	HP	Attack	Defense	Sp. Atk	Sp. Def	Speed	Generation	Legendary
1	Bulbasaur	Grass	Poison	45	49	49	65	65	45	1	False
2	Ivysaur	Grass	Poison	60	62	63	80	80	60	1	False
3	Venusaur	Grass	Poison	80	82	83	100	100	80	1	False
4	Mega Venusaur	Grass	Poison	80	100	123	122	120	80	1	False
5	Charmander	Fire	NaN	39	52	43	60	50	65	1	False

	Normal	Fire	Water	Electric	Grass	Ice	Fighting	Poison	Ground	Flying	Psychic	Bug	Rock	Ghost	Dragon	Dark	Steel	Fairy
Normal	1	1	1	1	1	1	2	1	1	1	1	1	1	1	0	1	1	1
Fire	1	0.5	2	1	0.5	0.5	1	1	2	1	1	0.5	2	1	1	1	0.5	0.5
Water	1	0.5	0.5	2	2	0.5	1	1	1	1	1	1	1	1	1	1	0.5	1
Electric	1	1	1	0.5	1	1	1	1	2	0.5	1	1	1	1	1	1	0.5	1
Grass	1	2	0.5	0.5	0.5	2	1	2	0.5	2	1	2	1	1	1	1	1	1
Ice	1	2	1	1	1	0.5	2	1	1	1	1	1	2	1	1	1	2	1
Fighting	1	1	1	1	1	1	1	1	1	2	2	0.5	0.5	1	1	0.5	1	2
Poison	1	1	1	1	0.5	1	0.5	0.5	2	1	2	0.5	1	1	1	1	1	0.5
Ground	1	1	2	0	2	2	1	0.5	1	1	1	1	0.5	1	1	1	1	1
Flying	1	1	1	2	0.5	2	0.5	1	0	1	1	0.5	2	1	1	1	1	1
Psychic	1	1	1	1	1	0.5	1	1	1	0.5	2	1	2	1	2	1	1	1
Bug	1	2	1	1	0.5	1	0.5	1	0.5	2	1	1	2	1	1	1	1	1
Rock	0.5	0.5	2	1	2	1	2	0.5	2	0.5	1	1	1	1	1	1	2	1
Ghost	0	1	1	1	1	1	0	0.5	1	1	1	0.5	1	2	1	2	1	1
Dragon	1	0.5	0.5	0.5	0.5	2	1	1	1	1	1	1	1	2	1	1	2	1
Dark	1	1	1	1	1	1	2	1	1	1	0	2	1	0.5	1	0.5	1	0.5
Steel	0.5	2	1	1	0.5	0.5	2	0	2	0.5	0.5	0.5	0.5	1	0.5	1	0.5	0.5
Fairy	1	1	1	1	1	1	0.5	2	1	1	1	0.5	1	1	0	0.5	2	1



Feature engineering  
(generate features for 2 opponents)

# Summary and next lecture

# Next Lecture

- **Lecture:** dimensionality reduction
- **Lab:** implementation of dimensionality reduction techniques in Python
  - The description/data will be posted on piazza
- **About the lab:**
  - Bring your laptops
  - Please install Python v2.7
  - Main libraries: numpy, scipy, matplotlib, scikit-learn
  - Strongly recommended to use **anaconda**
    - <https://www.anaconda.com/download/>

# Thank You!

