

# Lecture 8:

# Introduction to Applied ML

# in Practical Situations

CMSC 25910

Spring 2023

The University of Chicago



**(But First)  
Revisiting Ethical  
Research**



## Linux incident

A collection of resources regarding the incident involving University of Minnesota researchers and the Linux community

- [Initial Computer Science & Engineering statement](#)
- [Statement of apology from the involved researchers](#)
- [Computer Science & Engineering response to the Linux Foundation](#)

# Goals and Intuition

# Relationship Between Task & Methods

- Task: explain/describe data
  - Descriptive statistics (e.g., what percentage of people are late?)
- Task: use observed data to infer information about a population
  - Inferential statistics (e.g., what's the level of support for this candidate?)
- Task: draw a causal connection
  - Experiments (including on human subjects)
- Task: **predict** characteristics of **out-of-sample data**
  - Machine learning (prediction, forecasting, classification, etc.)

# High-Level Intuition



# High-Level Intuition



Fox



Wolf



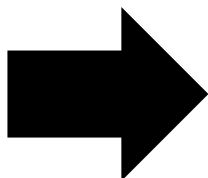
Fox



Wolf

Training

1



ML Model

# High-Level Intuition



Fox



Wolf



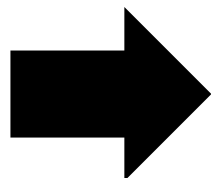
Fox



Wolf

Training

1



ML Model

2

Inference



**Fox : 77%**  
Wolf: 23%



# Why we build models...

- To understand data
- To make predictions about *out-of-sample* data

# Regression Example

# Let's Build a Model To Understand Data

- Running example: a regression problem
- Example:

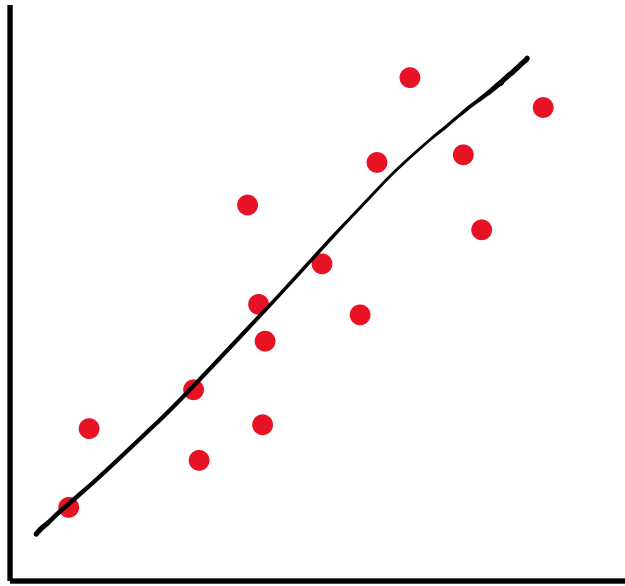
Name	Age	Department	Gender	Title	Salary
Jack	55	CS	M	Professor	??
Jane	27	Stats	F	Assistant Professor	??



Given these input vectors...

...predict this variable

# Building Intuition: Fitting a Line



# Given Input Vector $x$ , Predict $y$

- We need to choose a model to do that

$$\hat{y} = 0.3x$$

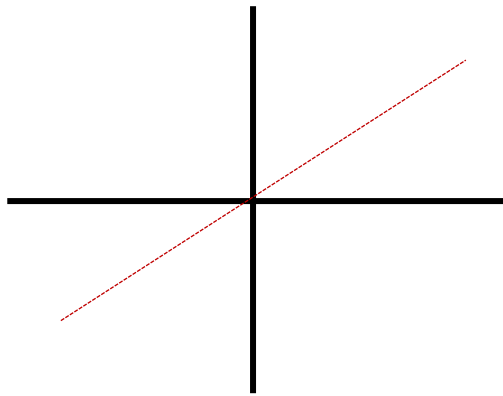


Diagram illustrating the linear model equation  $\hat{y} = w^T x$  and the associated variables:

- Output value / Explanatory**:  $\hat{y}$
- Parameters / weights**:  $w$
- Input vector / predictor**:  $x$

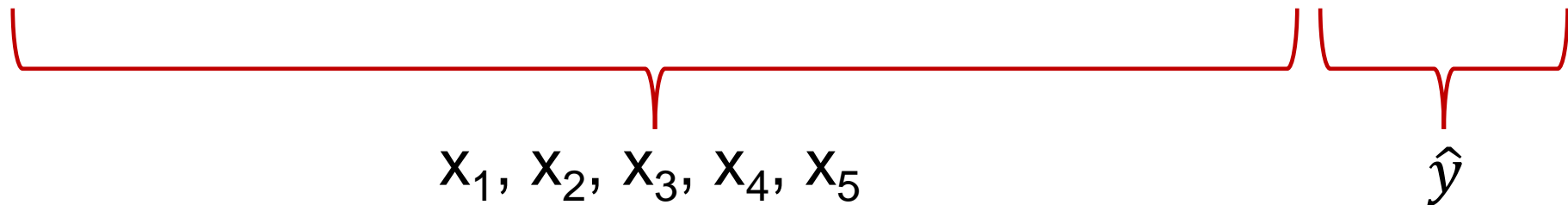
Mathematical definitions:

$$\begin{aligned}x &\in \mathbb{R}^n \\ y &\in \mathbb{R} \\ w &\in \mathbb{R}^n\end{aligned}$$

# Let's Build a Model To Understand Data

- Running example: a regression problem
- Example:

Name	Age	Department	Gender	Title	Salary
Jack	55	CS	M	Professor	??
Jane	27	Stats	F	Assistant Professor	??



Variables/Attributes/Columns become 'features' of the input vector

# Linear Regression Model

- 'Linear' because of the relationship between  $x$  and  $y$

$$\hat{y} = w^T x + b$$

# Linear Regression Model

- ‘Linear’ because of the relationship between  $x$  and  $y$
- A model is an assumption...
  - ...of what function represents data *well*
- Once we’ve fixed a model...
  - ...we find the parameters/weights  $w$  that make the model perform well

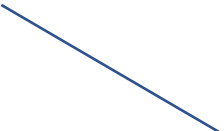
$$\hat{y} = w^T x + b$$




# Linear Regression Model

- ‘Linear’ because of the relationship between  $x$  and  $y$
- A model is an assumption...
  - ...of what function represents data *well*
- Once we’ve fixed a model...
  - ...we find the parameters/weights  $w$  that make the model perform well

$$\hat{y} = w^T x + b$$



We need a  
method to  
find those  
parameters



This suggests  
we need a  
performance  
metric

# Our Data

- A dataset becomes a matrix
  - Each row is an input vector

Name	Age	Department	Gender	Title	Salary
Jack	55	CS	M	Professor	33000
Jill	23	Econ	F	Professor	32000
Josh	32	Bio	M	Staff	28000
Jenn	44	Bio	F	Associate Professor	24000
Jane	27	Stats	F	Assistant Professor	25000

# Train-Test Split

- A dataset becomes a matrix
  - Each row is an input vector

Dataset  
contains the  
target  
variable /  
label

Training dataset	Name	Age	Department	Gender	Title	Salary
	Jack	55	CS	M	Professor	33000
	Jill	23	Econ	F	Professor	32000
Test dataset	Josh	32	Bio	M	Staff	28000
	Jenn	44	Bio	F	Associate Professor	24000
	Jane	27	Stats	F	Assistant Professor	25000

# Performance Metric

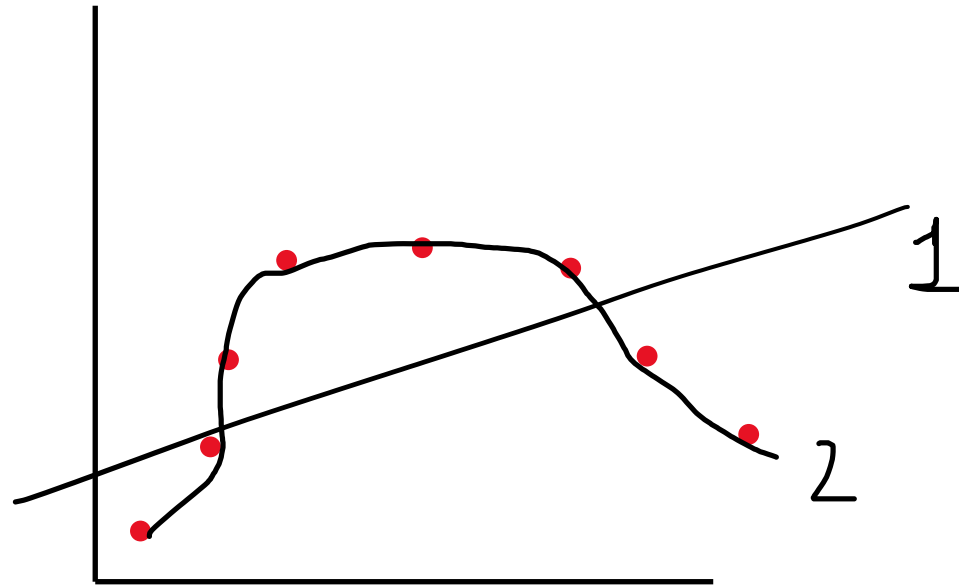
- Mean Squared Error (MSE)
  - Error decreases to 0 when *predicted y = ground-truth y*

$$\text{MSE}_{\text{test}} = \frac{1}{m} \sum_i (\hat{\mathbf{y}}^{(\text{test})} - \mathbf{y}^{(\text{test})})^2_i.$$

m test examples

- Goal: We want the model to perform well on the test data, which has “never been seen before” (out-of-sample data)

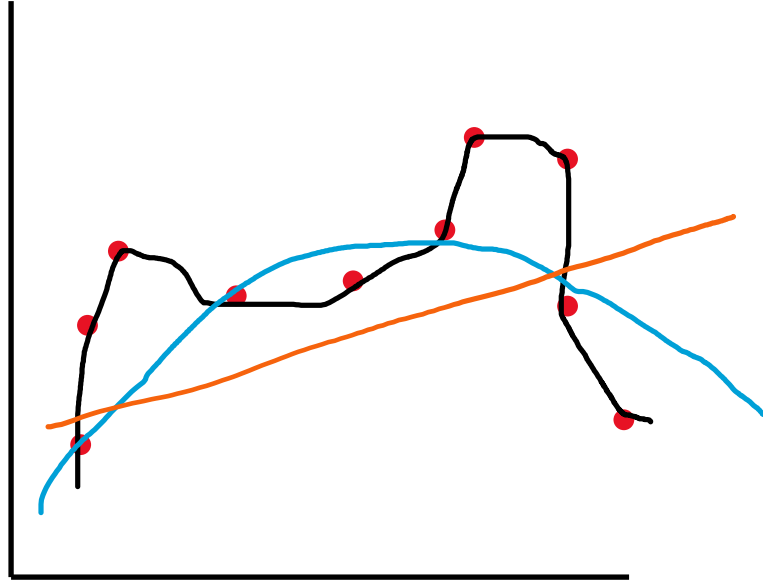
# Building Intuition...



- “*With four parameters I can fit an elephant, and with five I can make him wiggle his trunk.*”
  - John von Neumann (born Neumann János Lajos)

# Higher Capacity Models

- We can increase the capacity of the model by adding more parameters; this will help with obtaining a 'better' fit



$$\hat{y} = w^T x$$

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

$$y \in \mathbb{R}$$

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

# Goal

- We want to find parameters  $w$  using the training dataset

*We want to achieve  
a low training error*

# Optimization

- We want to find parameters  $w$  using the training dataset

$$\nabla_w \text{MSE}_{\text{train}} = 0$$

- This is an optimization problem that we know how to solve well; we can find the minimum MSE
- Consider that we run this optimization with the training data. What will happen when we run it on the test data?



# Challenges

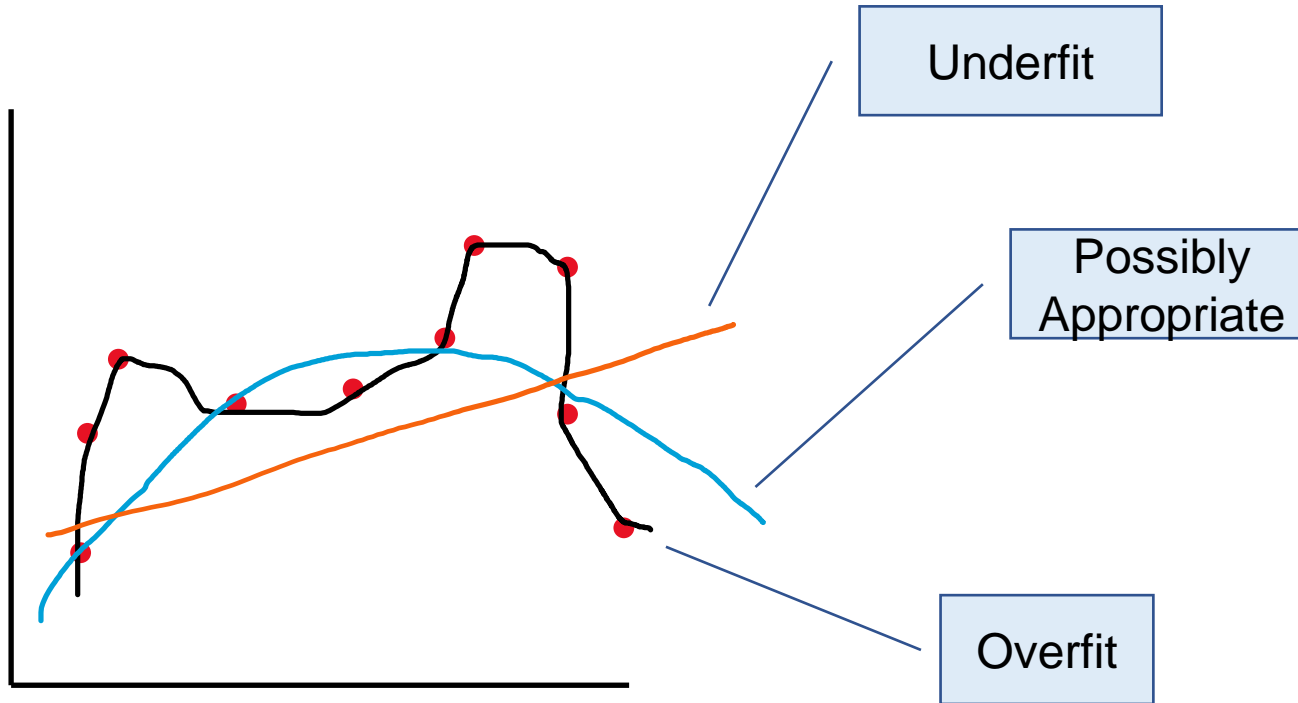
# Challenges For Machine Learning

- Learn parameters so the model performs well on unseen data
  - *Generalize* to **unseen data**
  - As opposed to the optimization problem of doing well on **training data**
- Remember why we build models:
  - To understand the process that generated the data
  - To make predictions about out-of-sample data
- Do you think minimizing the MSE on the training data helps us achieve any of those two goals?

# Underfitting, Overfitting

- Underfitting
  - When a model cannot reduce the *training error*
- Overfitting
  - A model achieves low *training error*, but high *test error*
- Ideally, we want low training error and small gap between training and test error
  - That's a model that explains the data generation process
  - That's a model that helps us predict out-of-sample data

# Underfitting, Overfitting...



# So, What Is Machine Learning?

- A model
  - Linear regression, logistic regression, ...
- Parameters
- A performance metric
  - MSE
- A training objective
  - Loss function
- A strategy to learn/fit the model parameters

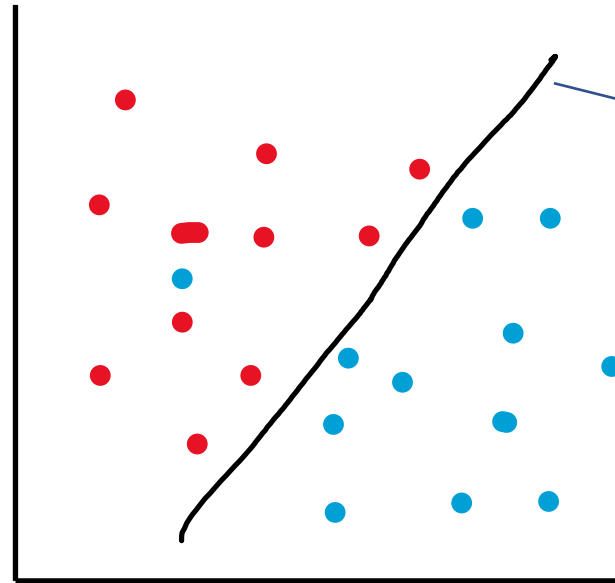
# One Common Task Formulation: Classification

# Classification Problem

- Given an input vector  $x$ , predict a class  $c$

- Binary classification problems

- Spam vs. not spam
- Give loan vs. don't
- Admit student vs. don't
- Will reoffend vs. won't



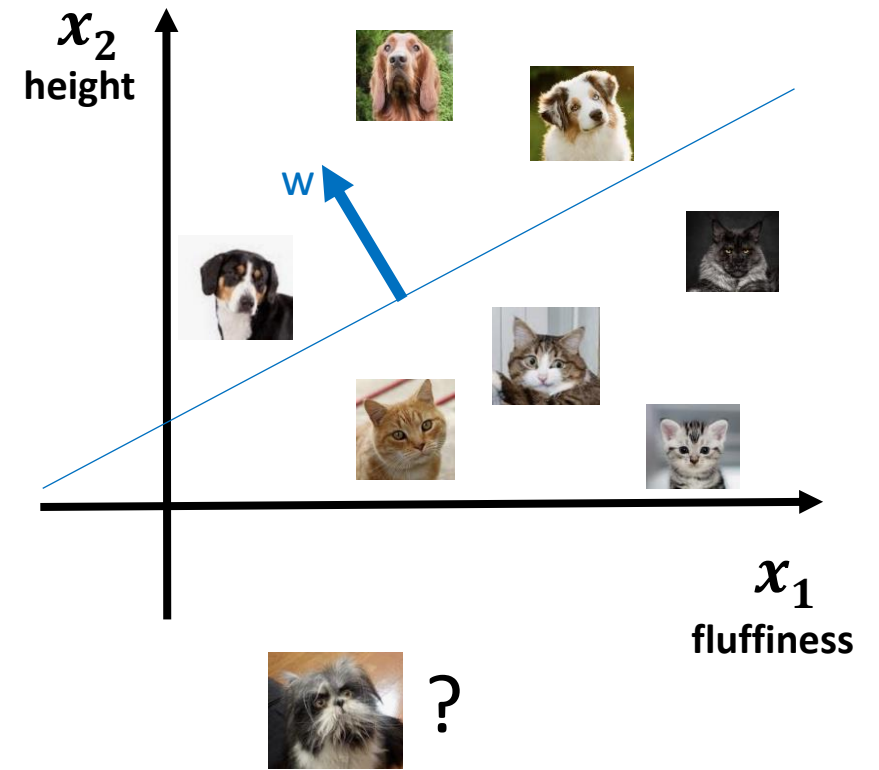
Find a  
hyperplane  
that separates  
the space of  
positive and  
negative  
samples

- How do you evaluate this?

- Accuracy, false positives/negatives, ...

# Classification Problem

- Build a model that can predict the categorical value of an unseen object
- Problem setting
  - $\mathbf{X}$  – set of possible instances with features  $x_i$
  - $Y$  – target class
  - Unknown target function  $f: \mathbf{X} \rightarrow Y$
  - Set of function hypotheses  $H = \{h | h: \mathbf{X} \rightarrow Y\}$
- Input
  - Training examples  $\{(\mathbf{x}^{(1)}, y^{(1)}), \dots, (\mathbf{x}^{(N)}, y^{(N)})\}$  of unknown distribution
- Output
  - Hypothesis  $h \in H$  that best approximates target function  $f$





# Logistic Regression

- Widely used models for **binary classification**:

$x$  = "Get a FREE sample ..."

➔  $y = 1$

1 = "Spam"  
0 = "Not spam"

$\phi(x) = [2.0, 0.0, \dots, 1.0, 0.5]$

- Models  $P(y=1|x)$ , the probability of  $y=1$  given  $x$

$$\hat{\mathbf{P}}_{\theta} (y = 1 \mid x) = \sigma(\phi(x)^T \theta) = \frac{1}{1 + \exp(-\phi(x)^T \theta)}$$

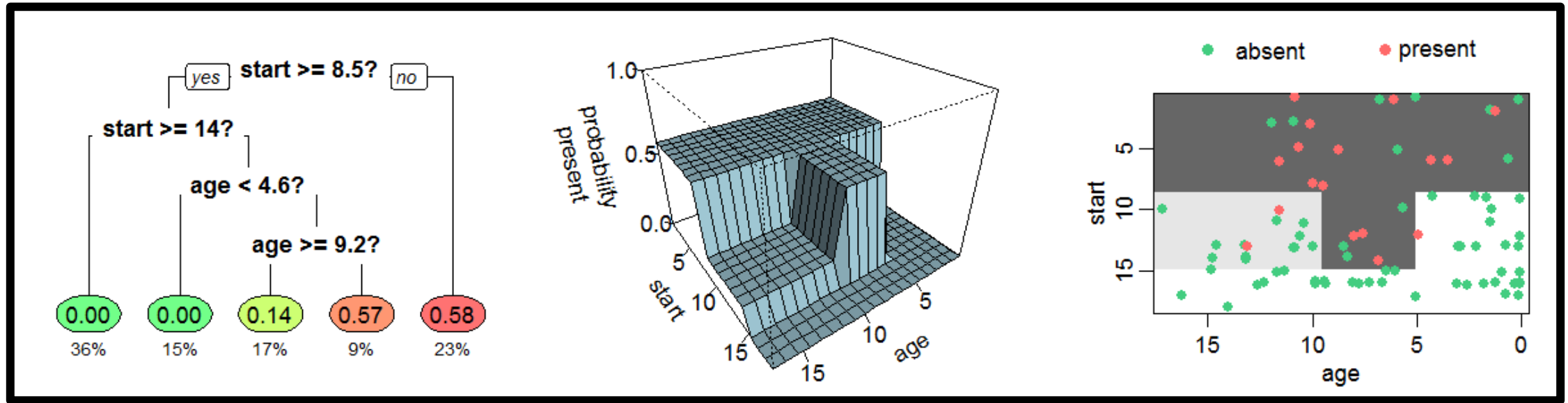
# Model Architectures

[Some of the slides in this section were cannibalized from Elena Zheleva at UIC, and by the transitive property from the Berkeley DS 100 team, Marine Carpuat, Lise Getoor, Brian Ziebart. Please do not further distribute. Mistakes are my own.]

# Some ML Model Architectures

- Regression models
- Decision trees
- Support Vector Machines (SVMs)
- Deep neural networks
- Many, many others:
  - PGM, genetic algorithms...

# Example Decision Tree



# Some Other Model Architectures

- **Support vector machine** (Boser et al. 1992)
  - Learning is **convex** (globally optimal weights)
  - Research shifted away from neural networks to SVMs / Kernel Methods
- SVMs are good for medium-large data.
- What about **REALLY BIG** data?

# Ensemble Methods

[Some of the slides in this section were cannibalized from Elena Zheleva at UIC, and by the transitive property from the Berkeley DS 100 team, Marine Carpuat, Lise Getoor, Brian Ziebart. Please do not further distribute. Mistakes are my own.]

# Ensemble Methods

- Simplest approach:
  1. Generate **multiple classifiers**
  2. Each votes on test instance
  3. Take majority as classification
- Classifiers can be different due to
  - different sampling of training data
  - randomized parameters within the classification algorithm
  - inductive bias (e.g, decision tree + perceptron + kNN)



# Random Forests

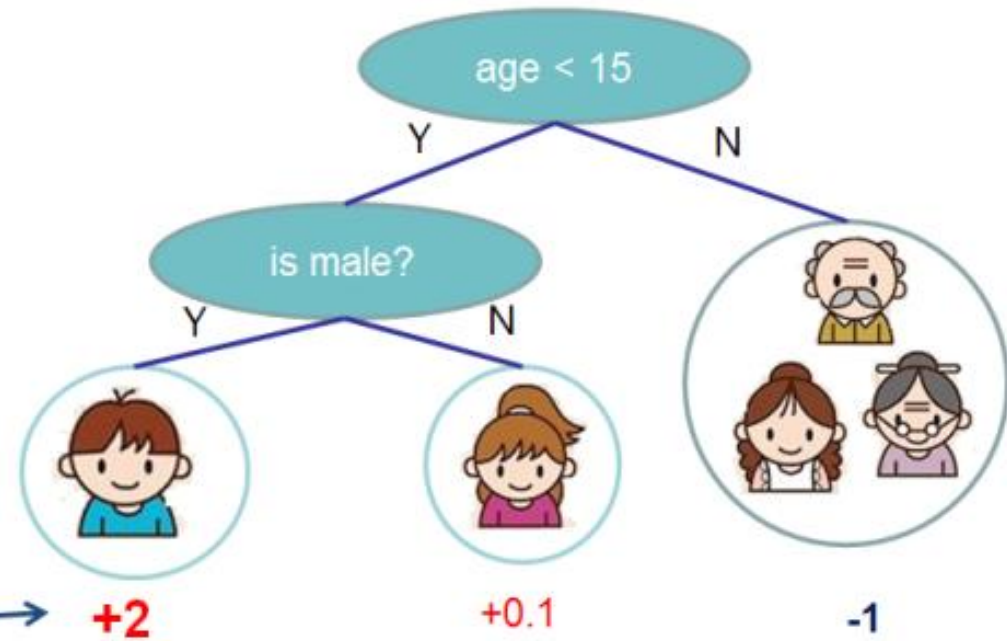
- Definition: Ensemble of decision trees
- Algorithm:
  - Divide training examples into multiple training sets (bagging)
  - Train a decision tree on each set
    - randomly select subset of variables to consider
  - Aggregate the predictions of each tree to make classification decision
    - e.g., can choose mode (most often) vote



# Regression Tree Ensemble

Input: age, gender, occupation, ...

Does the person like computer games



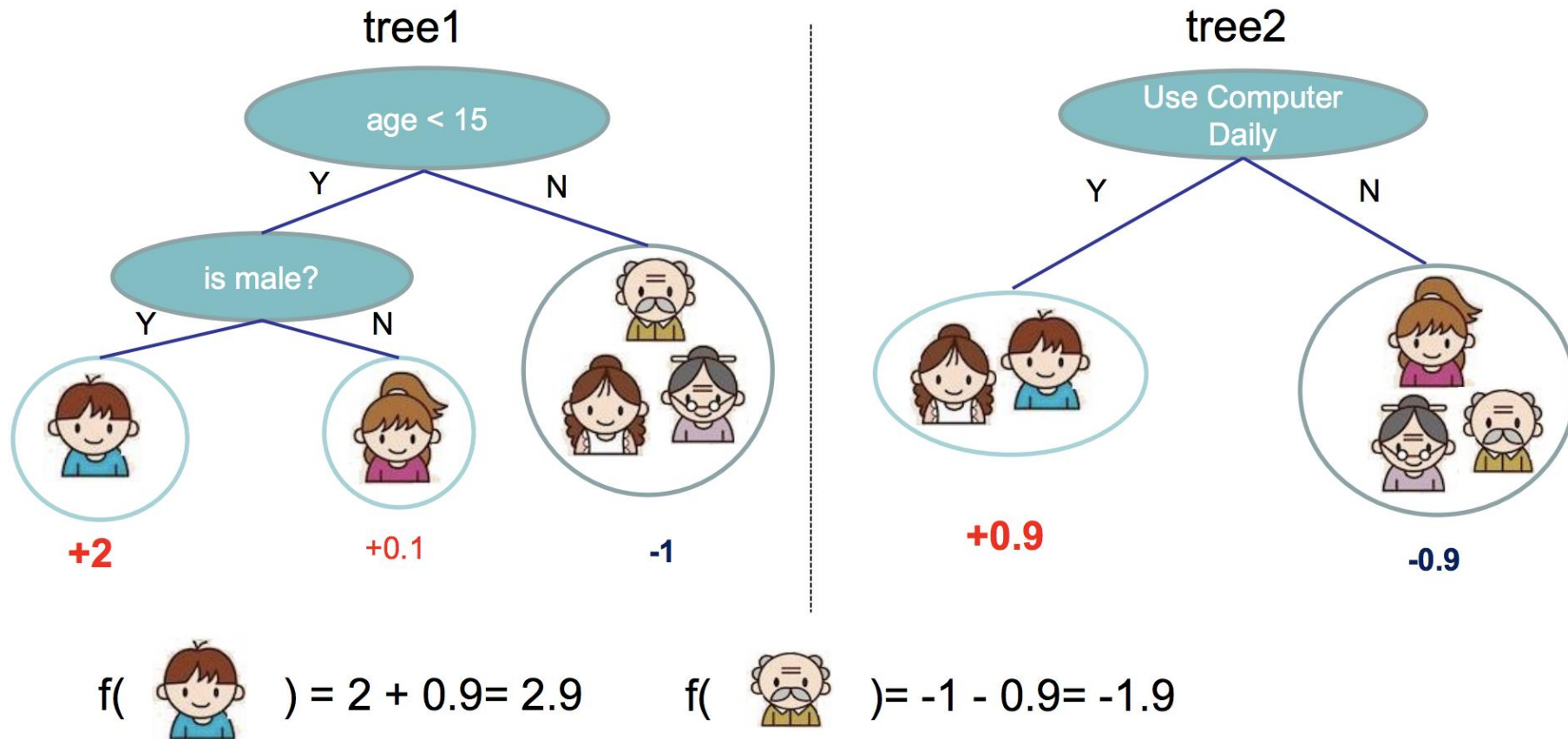
prediction score in each leaf →

**+2**

**+0.1**

**-1**

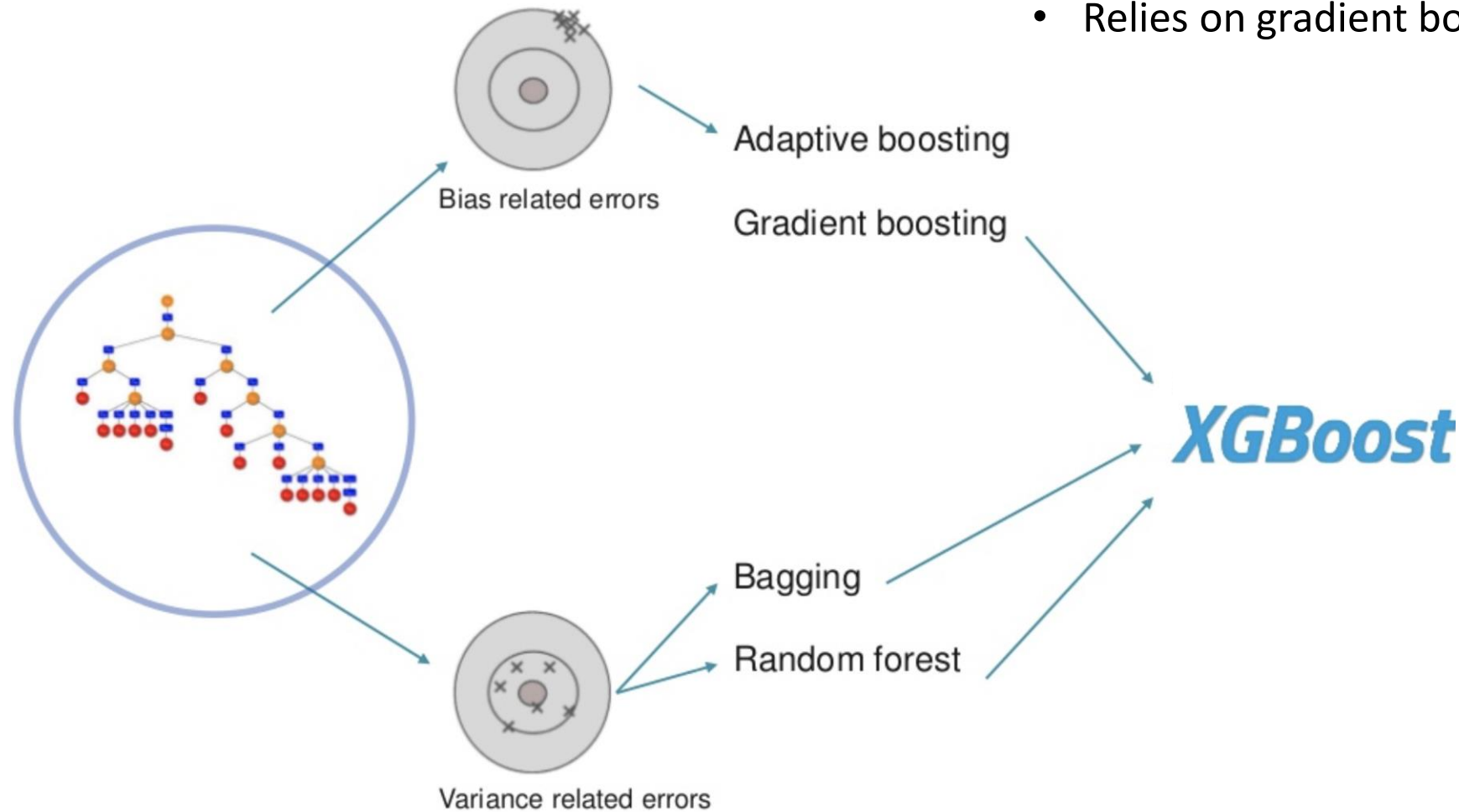
# Regression Tree Ensemble



Prediction of is sum of scores predicted by each of the tree

# XGBoost

- Developed by Chen and Guestrin (2016)
- Relies on gradient boosting

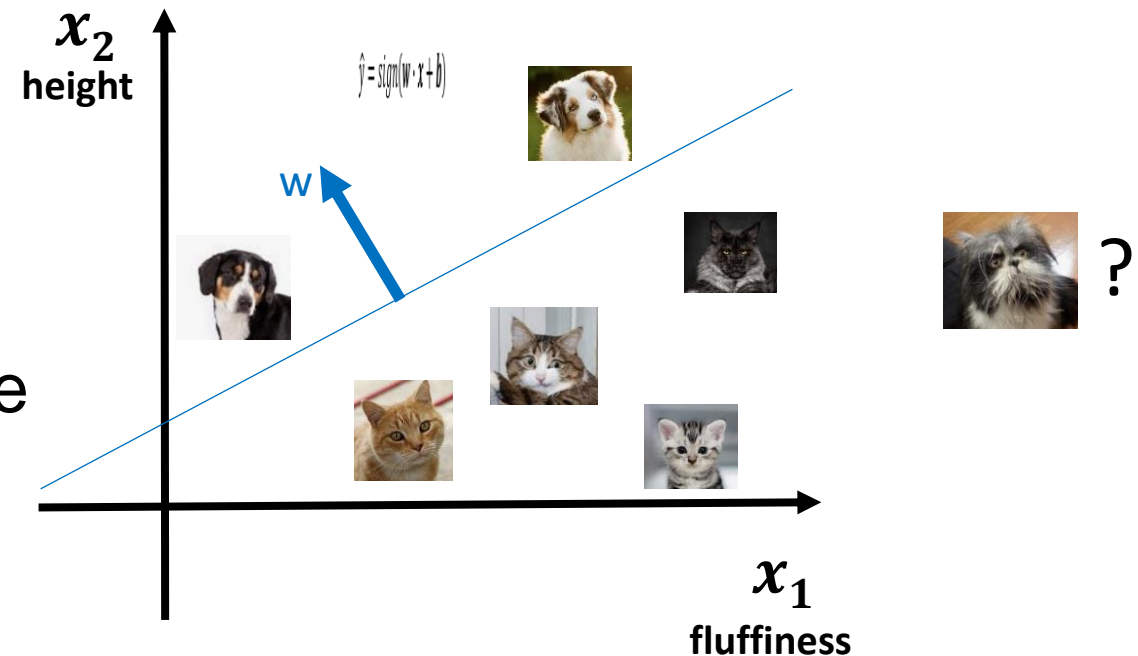


# Neural Networks

[Some of the slides in this section were cannibalized from Elena Zheleva at UIC, and by the transitive property from the Berkeley DS 100 team, Marine Carpuat, Lise Getoor, Brian Ziebart. Please do not further distribute. Mistakes are my own.]

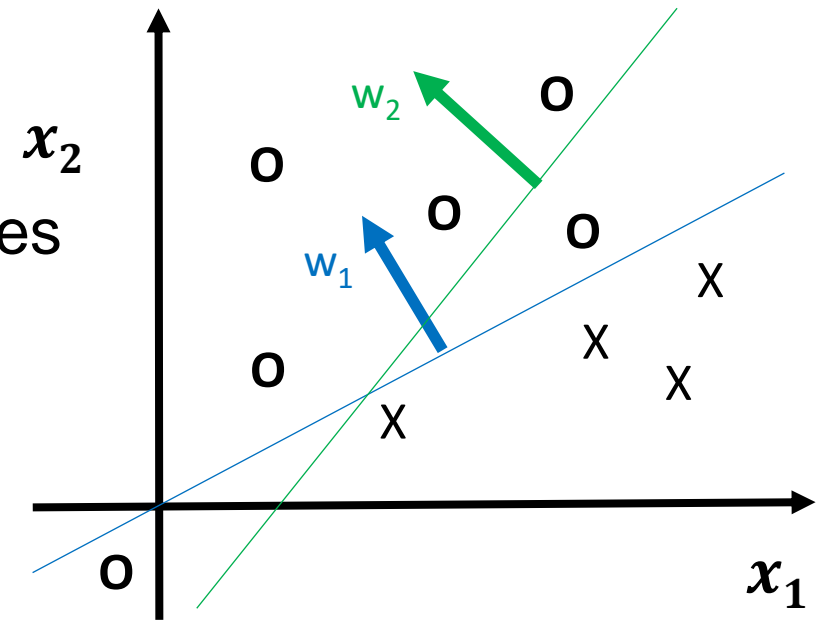
# Predecessor: Perceptron (1958)

- Assume decision boundary is a hyperplane
- Training = find a hyperplane  $w$  that separates positive from negative examples
- Testing = check on which side of the hyperplane examples fall
- Classifier = hyperplane that separates positive from negative examples
- See <https://en.wikipedia.org/wiki/Perceptron>



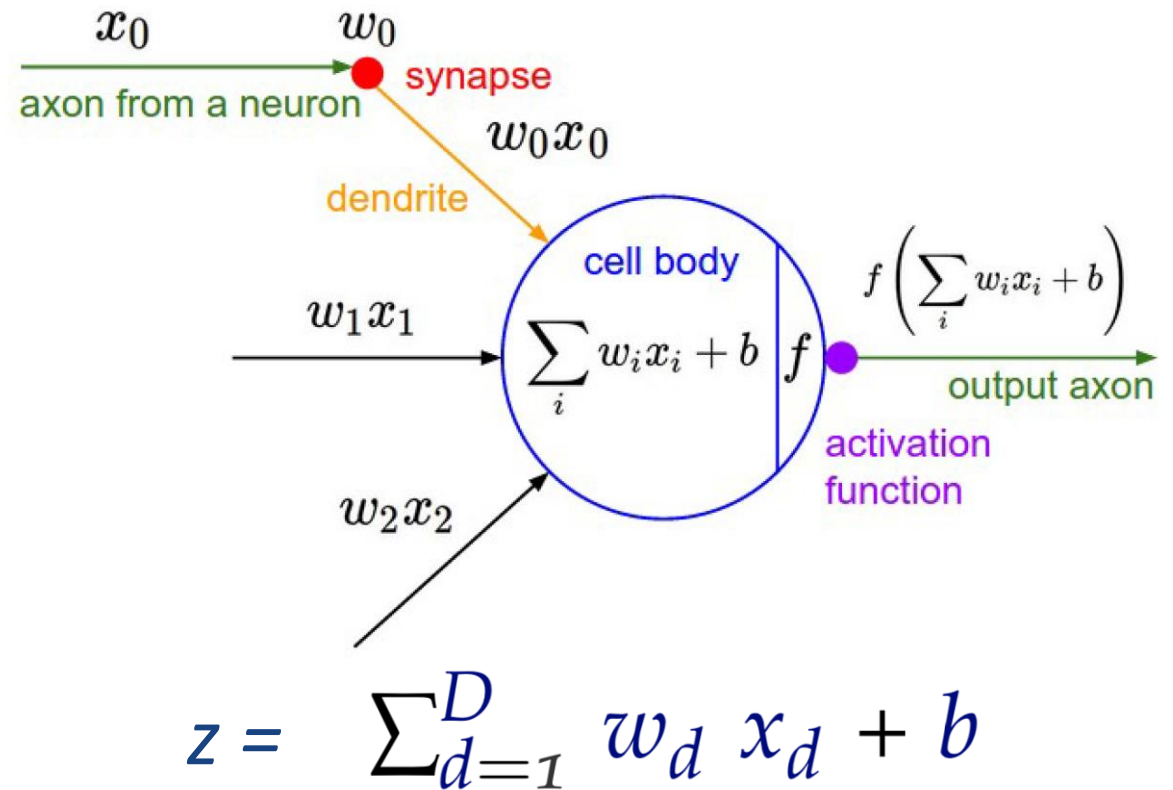
# Neural Networks

- We can think of neural networks as combination of multiple linear models (perceptrons)
  - Multilayer perceptron
- Why would we want to do that?
  - Discover more complex decision boundaries
  - Learn combinations of features



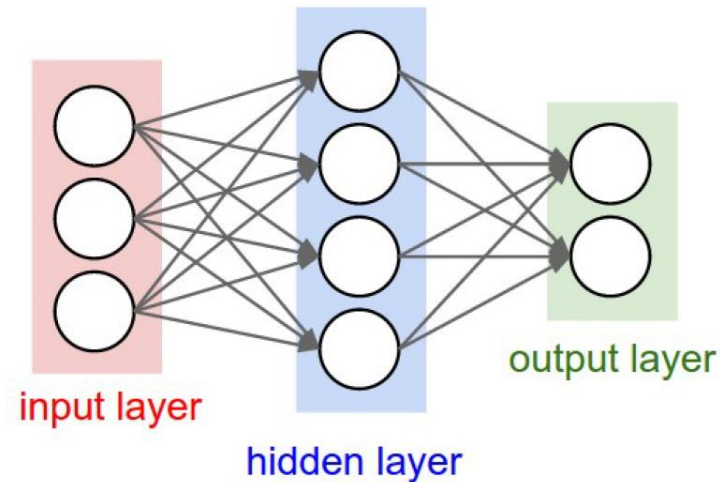
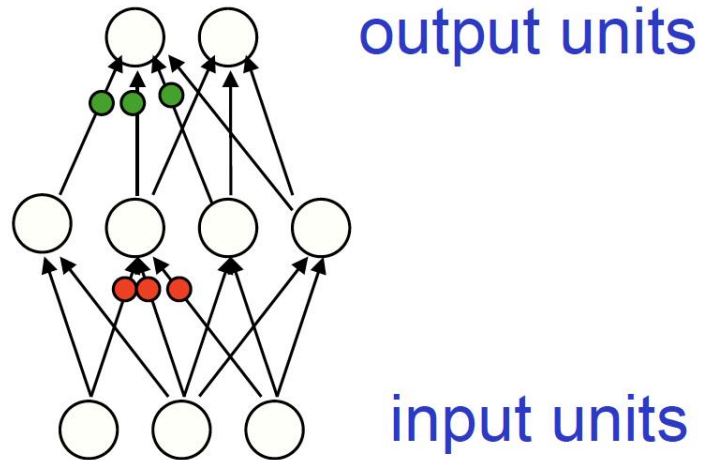
# Mathematical Model of a Neuron

- We can think of neural networks as combination of multiple perceptrons
  - **Hidden features** define functions of the inputs, computed by neurons
  - Artificial neurons are called **units**
  - Vanilla perceptron: activation function is  $\text{sign}(z)$



# Neural Network Architecture

- Neural network with one layer of four hidden units:



- Figure: Two different visualizations of a 2-layer neural network. In this example: 3 input units, 4 hidden units (layer 1) and 2 output units (layer 2)
- Each unit computes its value based on linear combination of values of units that point into it, and an activation function



# Neural Network Architecture

- Going deeper: a 3-layer neural network with two layers of hidden units
- N-layer neural network:
  - N-1 layers of hidden units
  - One output layer

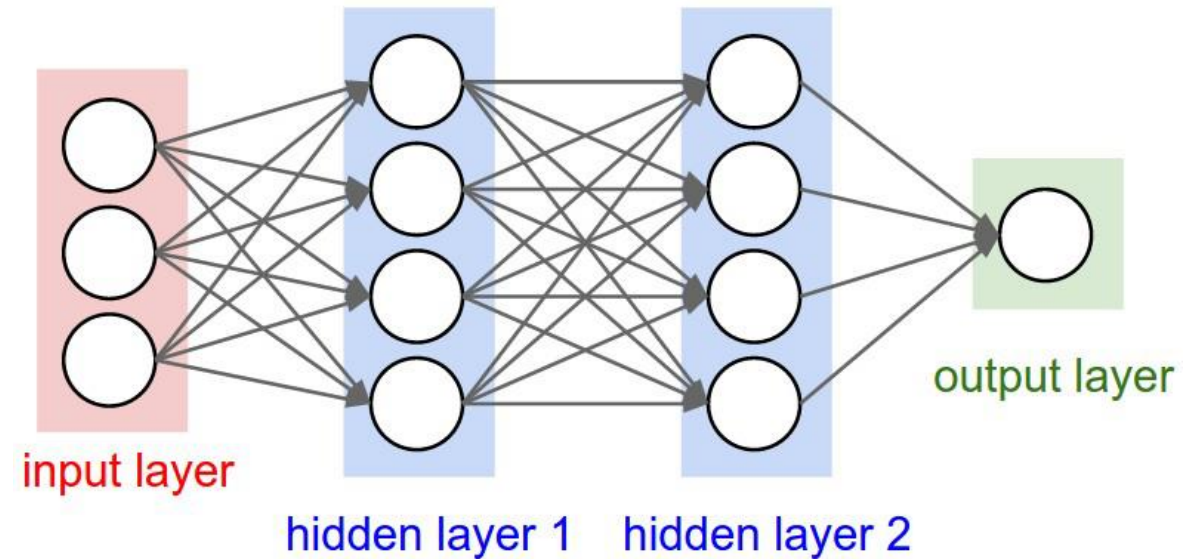
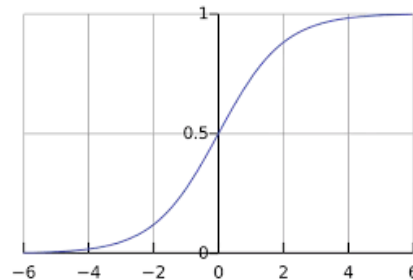
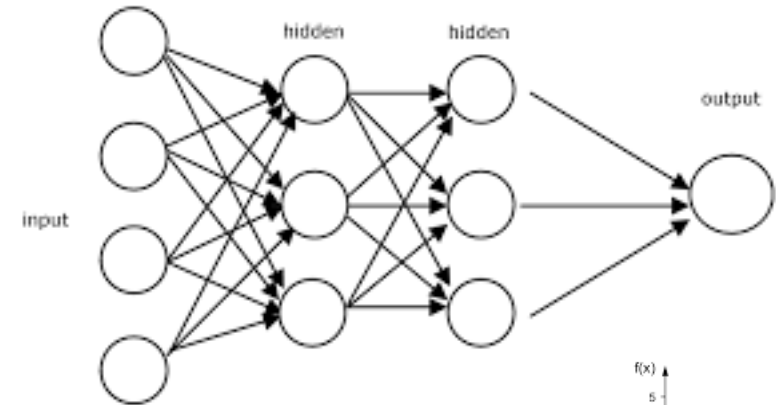


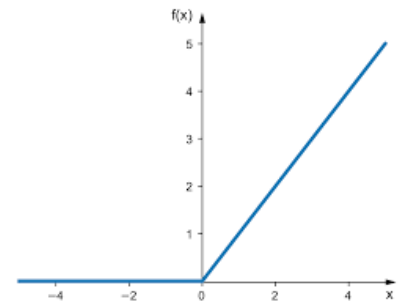
Figure : A 3-layer neural net with 3 input units, 4 hidden units in the first and second hidden layer and 1 output unit

# Neural Networks at 10,000 Feet

- $Y = f(X)$ 
  - F may be constructed by combining different functions
    - $\mathbf{h}^1 = g^1 (W^1 \mathbf{x} + b^1)$
    - $\mathbf{h}^2 = g^2 (W^2 \mathbf{h}^1 + b^2)$
    - ...
- Activation functions
  - Softmax
  - Relu
  - And many many more...
- Optimizers



Softmax



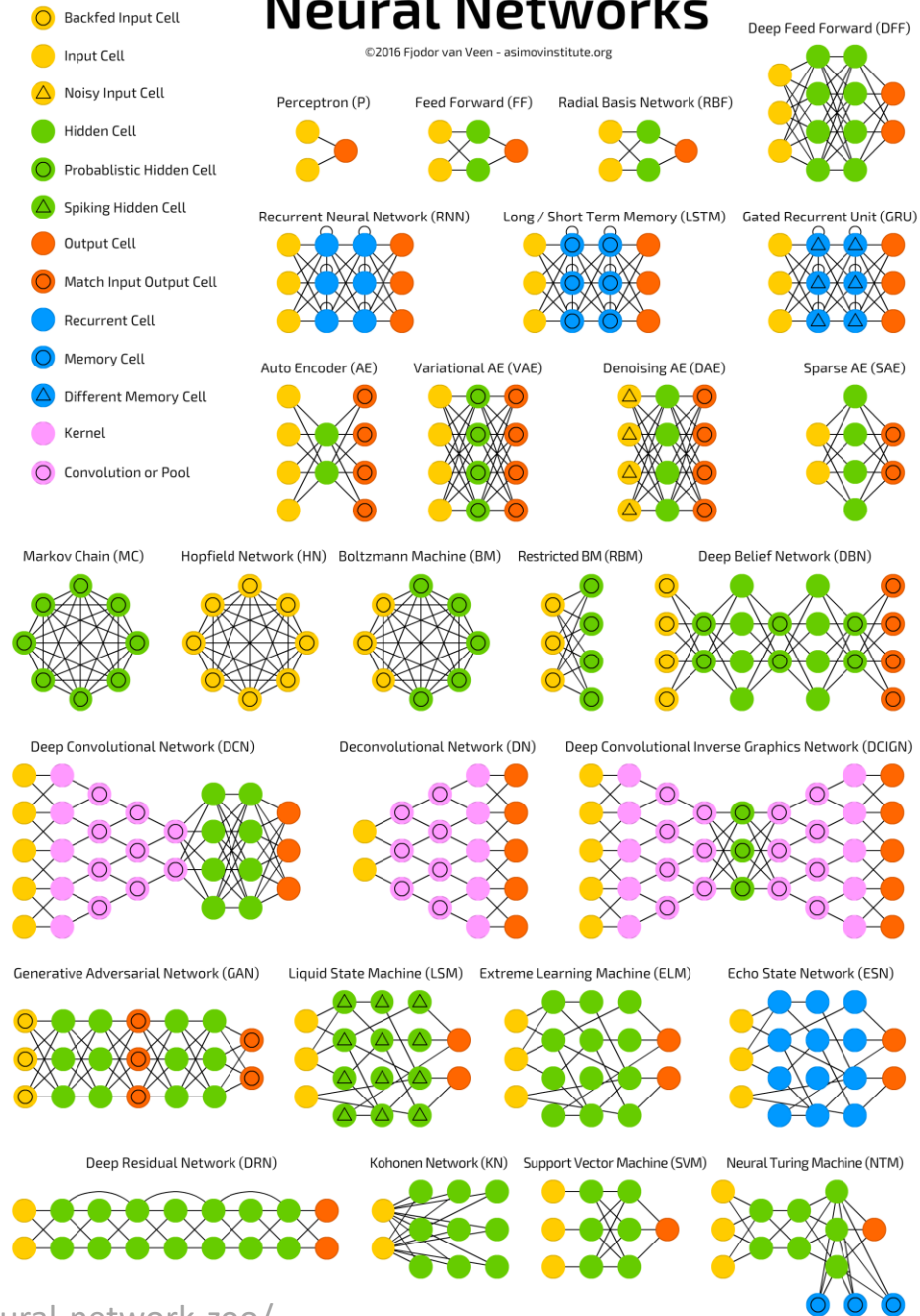
Relu

# Neural Networks: Backpropagation

- Goal: learn the weights of each layer
- Using backpropagation algorithm
  - Forward pass = prediction/inference
  - Backward pass = learning
    - Convert discrepancy between each output and its target value into an error derivative
    - Compute error derivatives in each hidden layer from error derivatives in layer above
- The optimization function is non-convex

# A mostly complete chart of Neural Networks

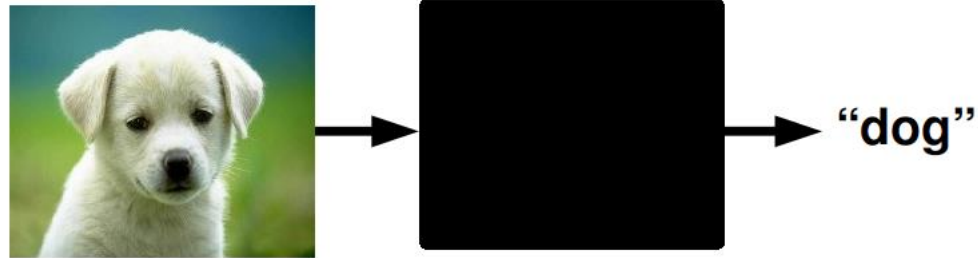
©2016 Fjodor van Veen - asimovinstitute.org



# “DEEP” Learning?

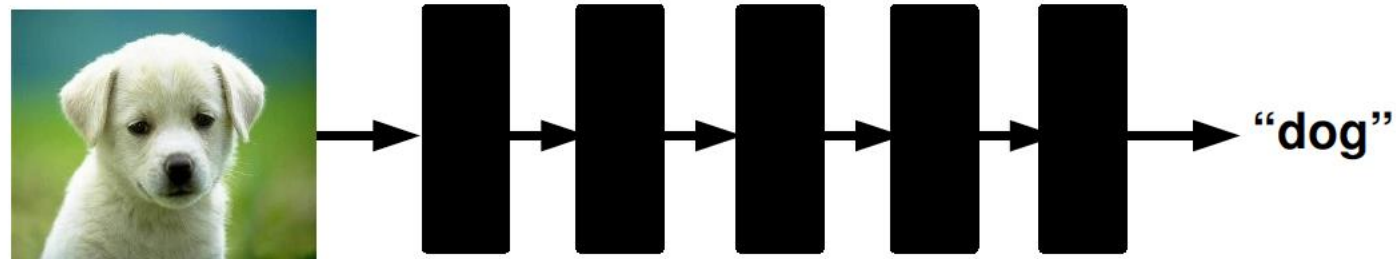
- Supervised learning

**Classification**



- Supervised deep learning

**Classification**



# Feature Engineering

# The Ingredients of an ML application

- (Possibly labeled) Training dataset:  $[X_i, y_i]$
- Model
- Task/Metric, Optimizer

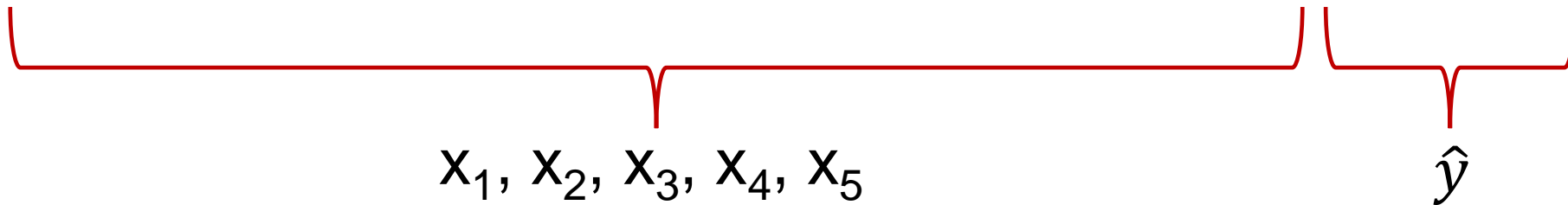
# Feature Representation

- Table data -> matrix
- Transform categorical variables into a numerical representation
  - Dummy coding
- Normalization
- Standardization
- Binning
- Other transformations



# Same Running Example As Last Time

Name	Age	Department	Gender	Title	Salary
Jack	55	CS	M	Professor	??
Jane	27	Stats	F	Assistant Professor	??



Variables/attributes/columns become 'features' of the input vector

# Feature Engineering

- Feature engineering
- Goal: Select the variables to feed into the model
  - More variables do not always lead to better models!

# Augmenting Features

- Augment initial data with more features
  - By joining with other datasets

Name	Age	Department
Jack	55	CS
Jane	27	Stats

JOIN

Gender	Title	Salary
M	Professor	??
F	Assistant Professor	??

# What Features Are You Selecting?

- How can you be sure that sensitive information is not represented in the model?
  - Is removing protected classes enough?
  - Think about information leakage!
- Anonymizing PII
  - Does this solve the problem?
  - Can you anonymize data?

Name	Age	Department	Gender	Title	Salary
Jack	55	CS	M	Professor	??
Jane	27	Stats	F	Assistant Professor	??

# Feature Engineering

- Preprocessing data
- What aspects of data matter?
  - What aspects **should** matter?

# Example of Feature Selection

Category	Collection Method	List of Features
Metadata	Google Drive/Dropbox API	account size, used space, file size, file type (img, doc, etc.), extension (jpg, txt, etc.), last modified date, last modifying user, access type (owner, editor, etc.), sensitive filename, sharing status
Documents	Local text processing	bag of words for top 100 content keywords, LDA topic models, TF-IDF vectors, word2vec representations, table schemas for spreadsheets
Images	Google Vision API [20]	image object labels, adult, racy, medical, violent, logos, dominant RGB values, average RGB value
Sensitive Identifiers	Google DLP API [18]	<i>counts</i> of the following identifiers in a file: name, gender, ethnic group, address, email, date of birth, drivers license #, passport #, credit card, SSN, bank account #, VIN

Table 3: A list of the features we automatically collected for each file using multiple APIs and custom code.

# Example of (Globally) Important Features

Task		Features
Sensitivity	Documents	gender; fraction of ethnic/VIN/location files; credit card; date of birth; email
	Images	fraction of gender/SSN/ethnic/location files; adult; credit card; racy; passport
Usefulness	Documents	access type; last modifying user; finance keywords; report & journal keywords
	Images	file size; finance keywords; access type; last modifying user; medical keywords
File Management	All Files	usefulness; sensitivity; spoof; account size; used space; finance keywords; medical keywords

Table 8: Top features for prediction tasks. Italicized *keywords* were top terms identified via the bag of words collections.

# Pitfalls of Feature Engineering

- ML model performance depends on the input data
  - Is the training data representative of the population?
  - Are the transformations applied to the data correct?
  - Is there enough training data to learn a good model?
- Many potential pitfalls throughout the process
  - Even careful humans will make mistakes!
- AutoML and automatic augmentation techniques
  - Opportunity or threat?



# Model Selection

- Backward elimination
  - Start with all variables and eliminate one by one
- Forward selection
  - Start with no variables and add one by one

# Training Data

# Training Datasets and Benchmarks

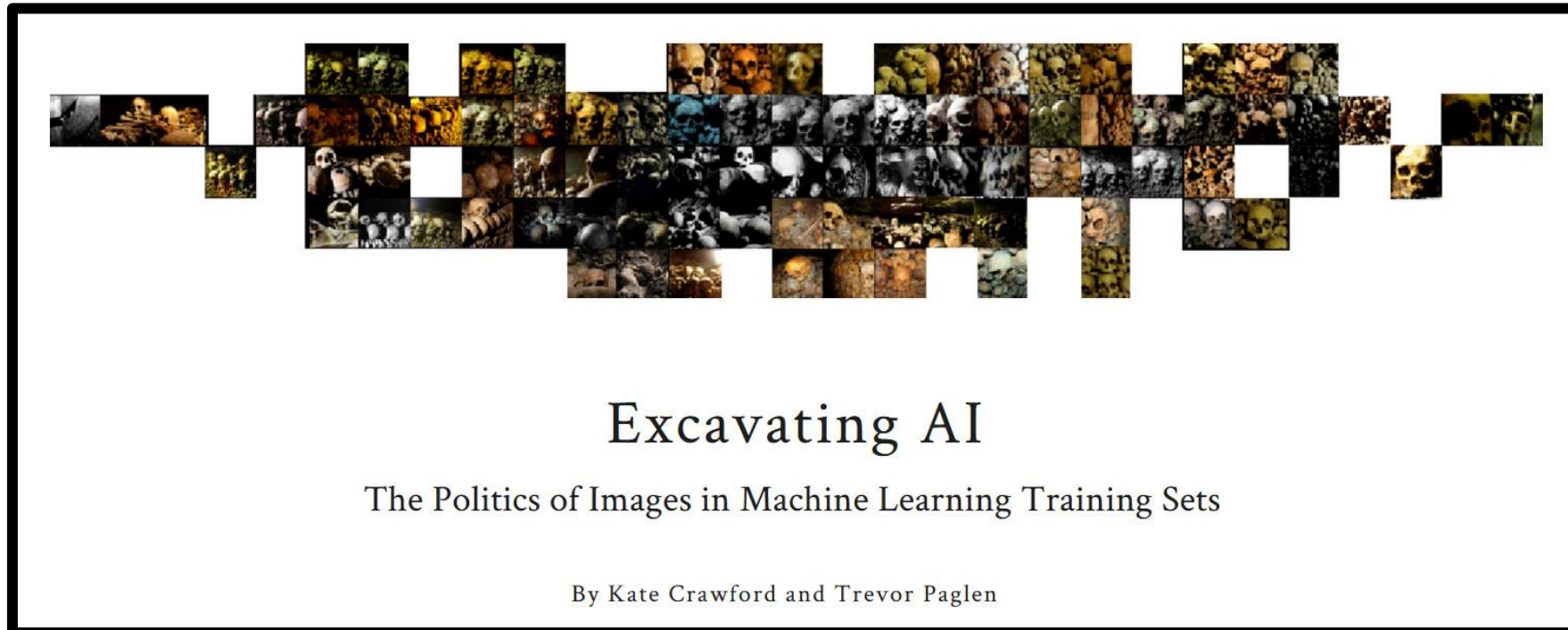
- Standardization of training datasets and benchmarks have arguably pushed the field of ML forward
  - Not without pitfalls
- If everyone is testing against the same datasets, what does that say about the ML model's generalizability?
  - Are results practically significant?
  - Do we notice errors that occur for data **excluded** from reference sets?
- There are more serious problems than a lack of progress!

# Imagenet: Computer Vision dataset

- 15 million images
  - Each image is annotated with a noun from Wordnet
    - Wordnet -> hierarchy of concepts
- Instrumental dataset to advance computer vision
- Where did these images come from?

# What Datasets Include/Exclude

- *Kate Crawford and Trevor Paglen, “Excavating AI: The Politics of Training Sets for Machine Learning (September 19, 2019)*
- <https://excavating.ai>



# What Datasets Include/Exclude




- “The automated interpretation of images is an inherently social and political project, rather than a purely technical one”
- “What work do images do in AI systems? What are computers meant to recognize in an image and what is misrecognized or even completely invisible?”
- “How do humans tell computers which words will relate to a given image? And what is at stake in the way AI systems use these labels to classify humans, including by race, gender, emotions, ability, sexuality, and personality?”
- “As the fields of information science and science and technology studies have long shown, all taxonomies or classificatory systems are political.”

# What Datasets Include/Exclude

*“There is much at stake in the architecture and contents of the training sets used in AI. They can promote or discriminate, approve or reject, render visible or invisible, judge or enforce. And so we need to examine them—because they are already used to examine us—and to have a wider public discussion about their consequences, rather than keeping it within academic corridors. As training sets are increasingly part of our urban, legal, logistical, and commercial infrastructures, they have an important but underexamined role: the power to shape the world in their own images.”*



# Where Did the Labels Come From?

We want to know if the main theme of the items below are "Cats". Label "Cat" if you think the main theme of the item is Cats, otherwise label "Not Cat". Label "Maybe/Not Sure" for items that you are uncertain about or if you think other workers might pick different labels.

	<input type="radio"/> Cat <input checked="" type="radio"/> Not Cat <input type="radio"/> Maybe/NotSure
	<input checked="" type="radio"/> Cat <input type="radio"/> Not Cat <input type="radio"/> Maybe/NotSure
	<input type="radio"/> Cat <input type="radio"/> Not Cat <input checked="" type="radio"/> Maybe/NotSure

**Figure 3.** Human Intelligence Task (HIT) interface for the Vote Stage. In addition to the predefined labels, crowdworkers can also select *Maybe/NotSure* when they were uncertain about the item.

The other workers have also finished labeling the same items you just labeled. The following items received different labels. Please provide an explanation for each of your labels below.

	You labeled "Not Cat". Please focus on describing things about the item that could have made it difficult or ambiguous for others. <input type="text" value="This is a tiger."/> <input type="button" value="Save"/>
	You labeled "Maybe/NotSure". Please focus on describing things about the item that could have made it difficult or ambiguous for others. <input type="text" value="This is a cartoon drawing of a cat."/> <input type="button" value="Save"/>

**Figure 4.** Human Intelligence Task (HIT) interface for the Explain Stage. Crowdworkers enter a short description for each item that was labeled differently in the Vote Stage. They were informed that disagreement occurred, but not the distribution of different labels used.



# Metrics

# Some Possible Metrics (Classes)

- Accuracy:  $\# \text{ correct} / \# \text{ total}$
- Confusion matrix (TP/FP/TN/FN)
- Area under the ROC curve (AUC)
  - True Positive Rate (TPR) =  $TP / P = TP / (TP + FN)$
  - False Positive Rate (FPR) =  $FP / N = FP / (FP + TN)$
  - ROC curve plots TPR vs. FPR at various thresholds
- Precision:  $TP / (TP + FP)$
- Recall:  $TP / (TP + FN)$
- Precision-Recall Curve

See <https://medium.com/analytics-vidhya/performance-metrics-for-machine-learning-models-80d7666b432e>  
[https://www.tutorialspoint.com/machine\\_learning\\_with\\_python/machine\\_learning\\_algorithms\\_performance\\_metrics.htm](https://www.tutorialspoint.com/machine_learning_with_python/machine_learning_algorithms_performance_metrics.htm)  
<https://www.justintodata.com/machine-learning-model-evaluation-metrics/> or many more!

# Some Possible Metrics (Numbers)

- Mean Squared Error
- Mean Absolute Error

See <https://medium.com/analytics-vidhya/performance-metrics-for-machine-learning-models-80d7666b432e>  
[https://www.tutorialspoint.com/machine\\_learning\\_with\\_python/machine\\_learning\\_algorithms\\_performance\\_metrics.htm](https://www.tutorialspoint.com/machine_learning_with_python/machine_learning_algorithms_performance_metrics.htm)  
<https://www.justintodata.com/machine-learning-model-evaluation-metrics/> or many more!

# Some Possible Metrics Revisited

- Do these metrics capture the **relationship** between **errors**?
- Do these metrics capture the **impact of errors**?
- Do these metrics capture the **differential** impact of **particular types of errors**?
- (Setting the stage for our next lecture!)

# Some Possible Metrics (Performance)

- Model training time
- Frequency of model re-training
- Model size
- Classification time
- Privacy issues of the model
- “Security” (future lecture)

# **Commoditization of ML**

# ML Models as a Commodity

- We've talked about ML as:
  - Find a training dataset, goal, metric
  - Train the model
  - Use it for the task at hand
- Many models take many weeks to train in data-center scale computers. They are made available to everyone publicly:
  - Download off-the-shelf models
  - They have been trained with data that may not be available to you

# The Trend Continues

- Models used as part of services packaged in the cloud vendors
  - Example: parts of AutoML offerings
- It's easy to lose sight of the models you are using
- Many models are an amalgam of others: e.g., consider NLP
  - Input data is *featurized* using a ML model
    - GPT-3, BERT, Transformer-like models
  - Parameters may have been *pre-trained* with some dataset
  - Then you *fine-tune* to your data
- Allen NLP examples
- Kaggle



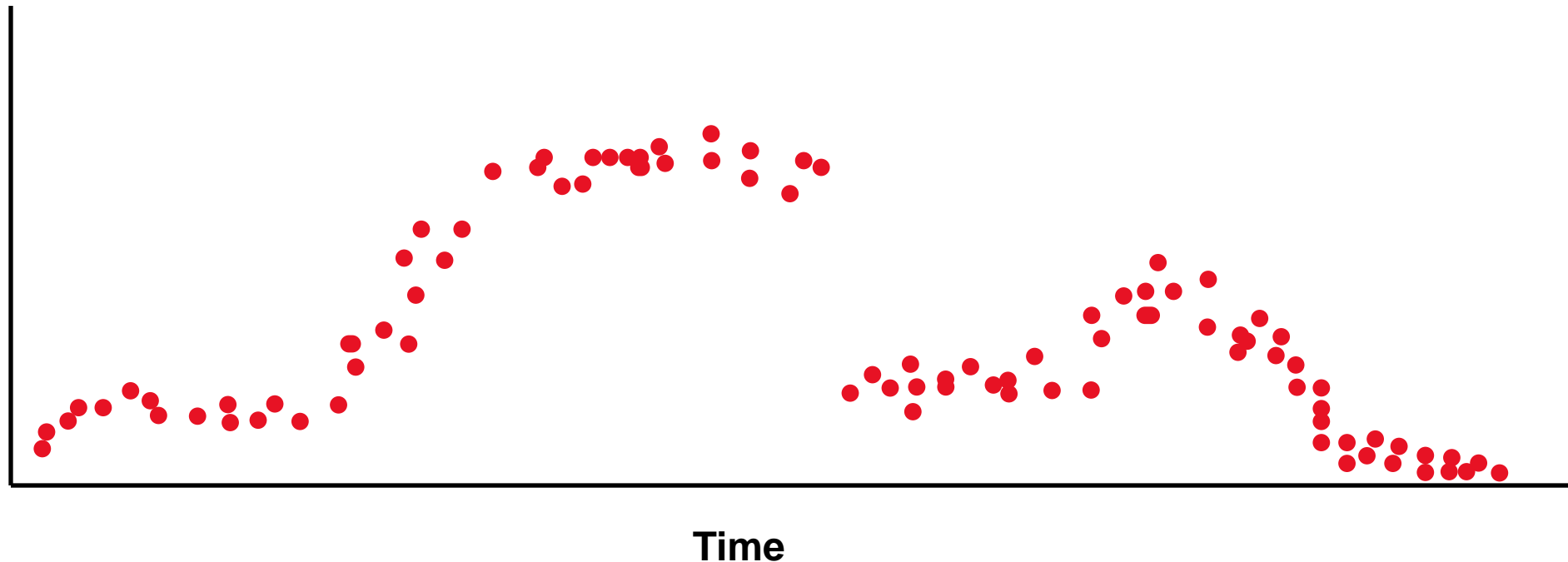
# ML in the Wild

Concept Drift

ML in Pipelines

# Concept Drift – The Passage of Time

- Extrapolation and Generalization
  - What population does the training data represent? At what point?
  - What claim can we make about the result?



# Real systems use multiple models

Example: An information extraction system

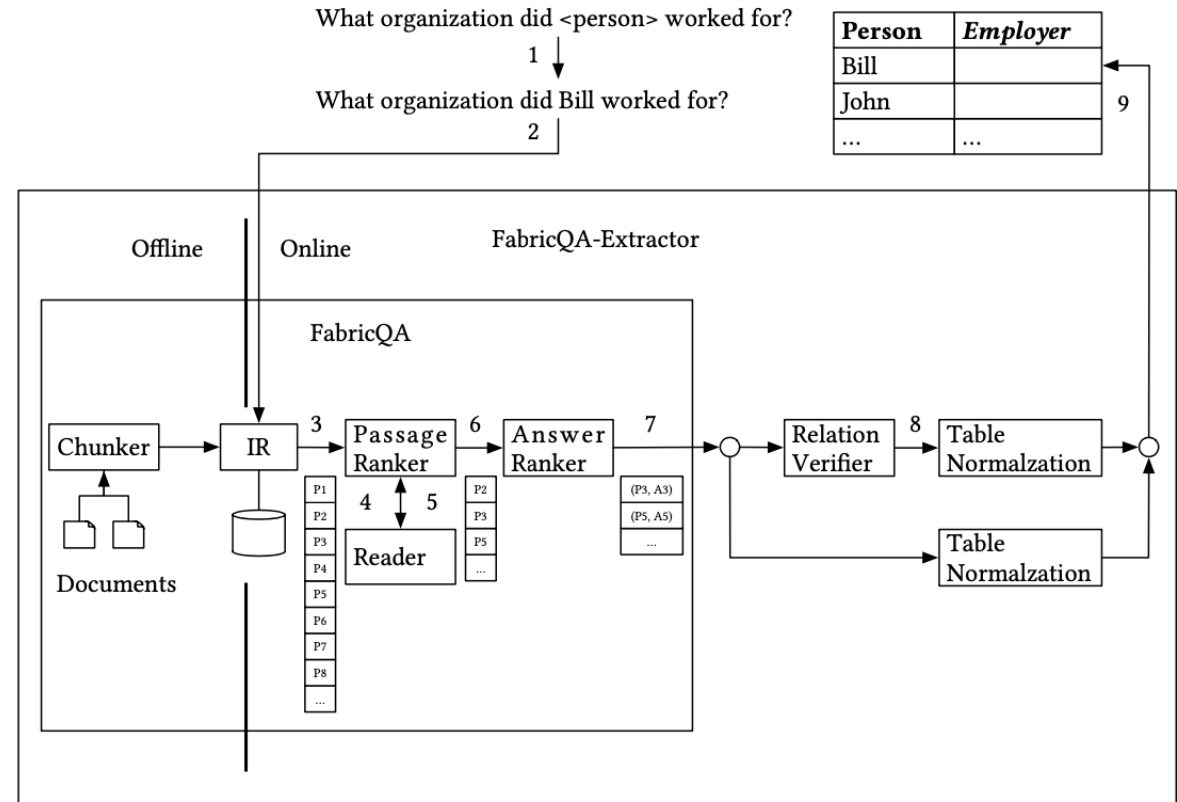


Figure 5: System Architecture

# **Algorithmic Decision Making**

# The Application Context Matters Greatly

Hiring

Online Advertising

Student Admissions

Criminal Justice

Health Insurance Markets

Creditworthiness

# Selbst et al.'s Five Pitfalls

- Framing Trap
  - “Failure to model the entire system over which a social criterion, such as fairness, will be enforced”
- Portability Trap
  - “Failure to understand how repurposing algorithmic solutions designed for one social context may be misleading, inaccurate, or otherwise do harm when applied to a different context”
- Formalism Trap
  - “Failure to account for the full meaning of social concepts such as fairness, which can be procedural, contextual, and contestable, and cannot be resolved through mathematical formalisms”
- Ripple Effect Trap
  - “Failure to understand how the insertion of technology into an existing social system changes the behaviors and embedded values of the pre-existing system”
- Solutionism Trap
  - “Failure to recognize the possibility that the best solution to a problem may not involve technology”

# What Does Accountability Mean Here?

- Who's accountable for the consequences of an ML model?
  - Those who deployed it?
  - Those who built it and trained it?
  - The owners of the training data?
  - Those who listened to the algorithm?