LESSON 1: THE BUILDING BLOCKS

Savannah Thais

Intro to Machine Learning

Princeton Wintersession 2021

01/25/2021

iris hep

Princeton Research Computing

Welcome!

About me:

- Particle physics PhD
- Now working in physics-informed ML
- Excited about scientific ML + AI ethics!

What We'll Cover:

- Conceptual foundation
- Mathematical intuition
- Model building fundamentals
- Application examples

More to Learn:

- Not a deep dive into the mathematics
- Not an exhaustive run-down of models/methods
- Find out what to learn more about

Course Guidelines:

- First time teaching this course
- Please interrupt!
- Course website
- Setup a local environment or work in Google Collab

Outline for the Course

- **Monday**
  - Conceptual overview
  - Mathematical foundations
- **Tuesday**
  - Neural Networks
- **Wednesday**
  - Convolutional Neural Networks
  - Recurrent Neural Networks
- **Thursday**
  - Generative Models
  - Unsupervised Models
- **Friday**
  - Designing an ML experiment
  - Examples of research applications
  - AI Ethics and Algorithmic Fairness
  - Other topics

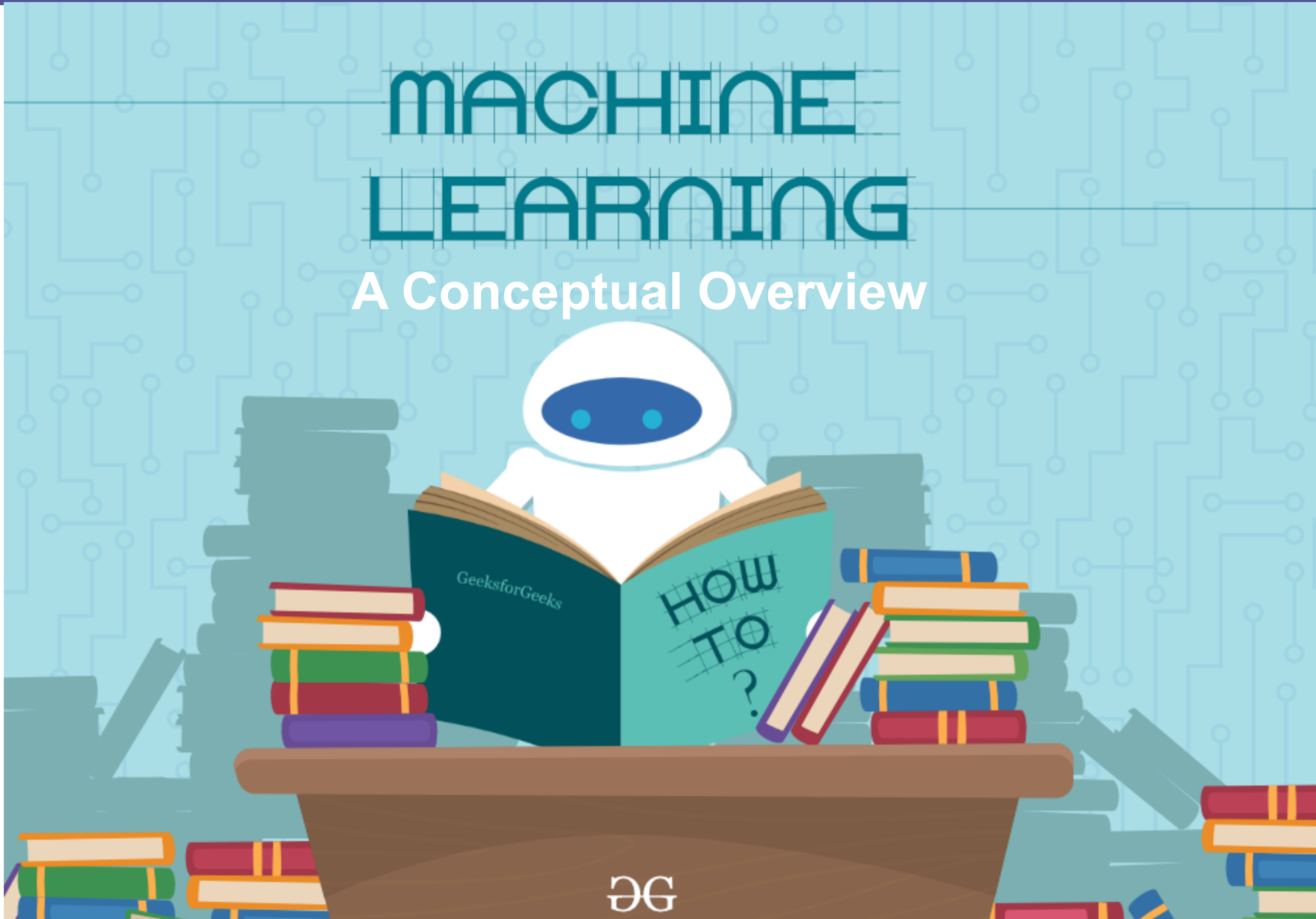
Outline for Today

- **What is ML and What Can It Do?**
- **Supervised Learning: Training**
- **Mathematical Foundations**
- **Introduction to Libraries**
- **Coding Exercises**

Introduction Time!

Why are you interested in ML?

- Name + Affiliation
- What made you want to take this course?
- What do you most hope to learn?


MACHINE
LEARNING
A Conceptual Overview
GeeksforGeeks
HOW
TO
?

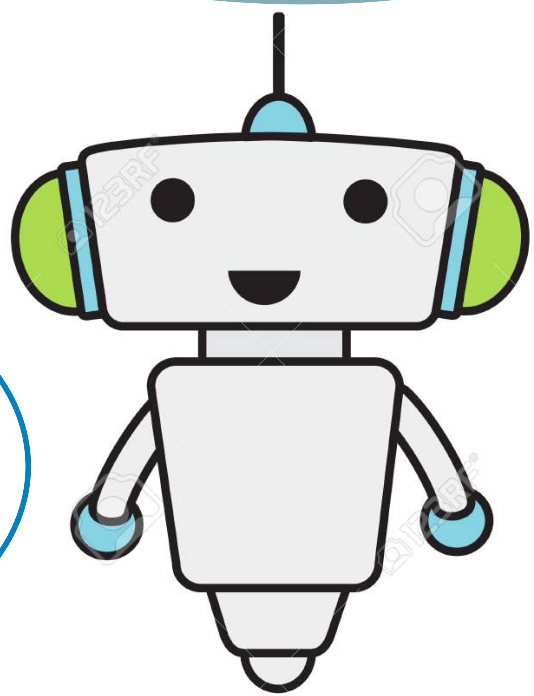
Killer Robots! Or, What is Machine Learning?

A field of study that gives the ability to the computer to learn without being explicitly programed

Algorithms that improve automatically through experience

Computer programs that can access data and use it to learn for themselves

Using statistics to find patterns in large datasets without the patterns being explicitly stated



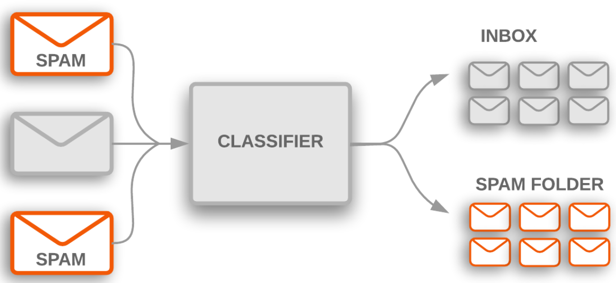
Teaching a computer system how to make accurate predictions when fed data

ML *can* be a great tool to enable scientific (and other) research, but it doesn't know what what it doesn't know

What are Problems for ML?

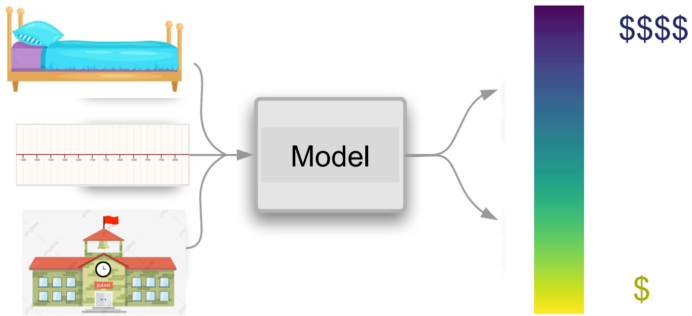
Classification:

Predict a class label for an input



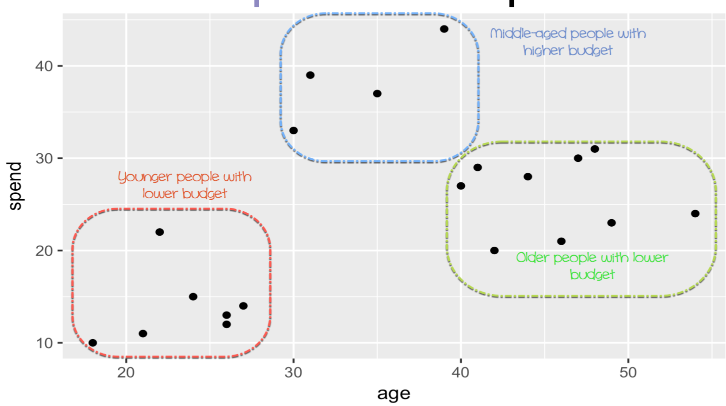
Regression:

Predict a continuous variable



Clustering:

Group similar inputs



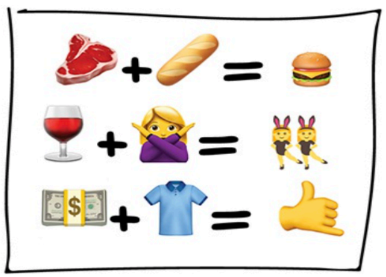
Generation:

Construct new data within pattern

What are Problems for ML?

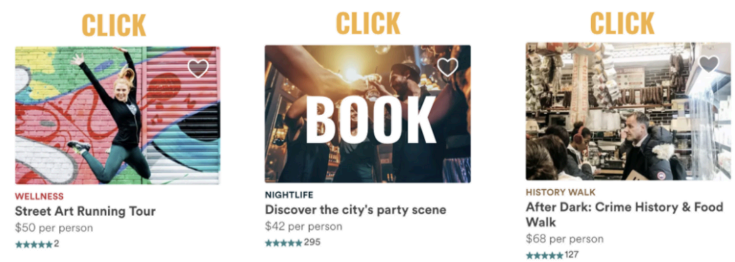
Association Rules:

Identify common patterns in data



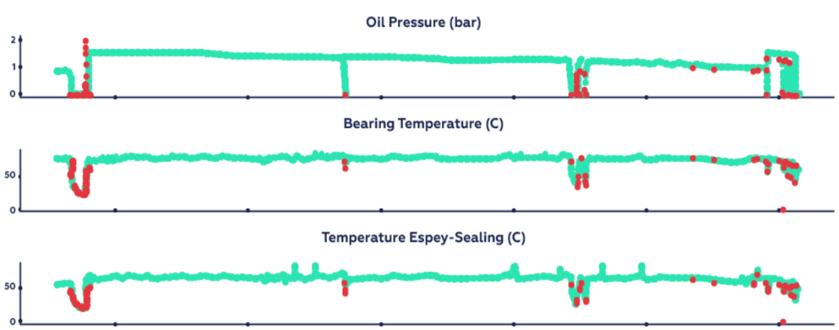
Ranking:

Generate optimal orderings



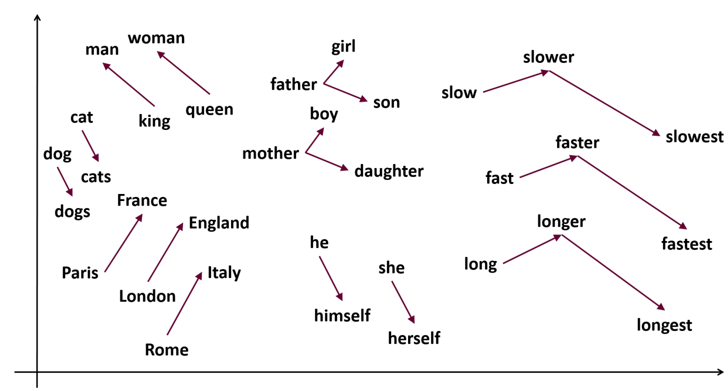
Anomaly Detection:

Identify statistical outliers



Restructuring:

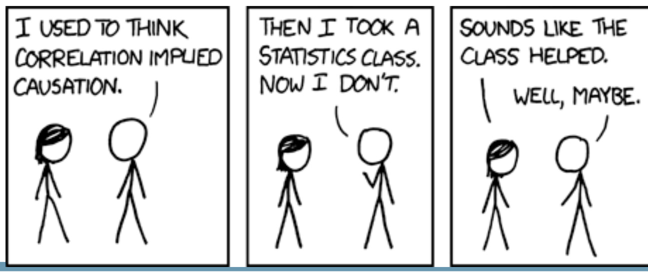
Transform data representations



What are NOT Problems for ML?*

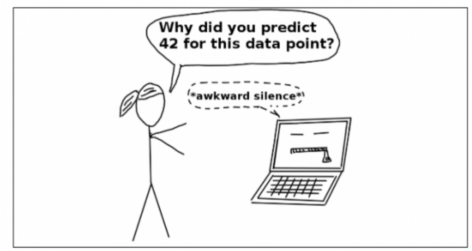
Causation:

Models learn correlations, but can't infer causality or intent



Precise Interpretability:

It's often difficult to understand what a model is learning

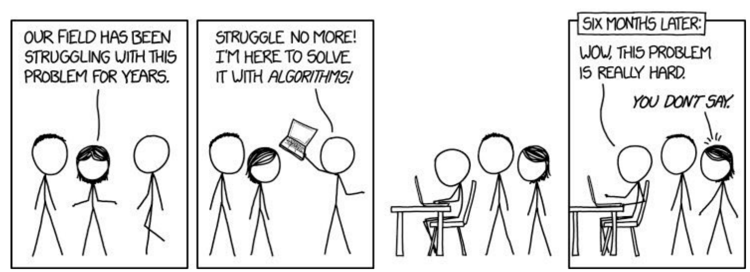


Context:

Models are incapable of non-mathematical reasoning

Data Limitations:

Models can't fix problems in data or learn without examples

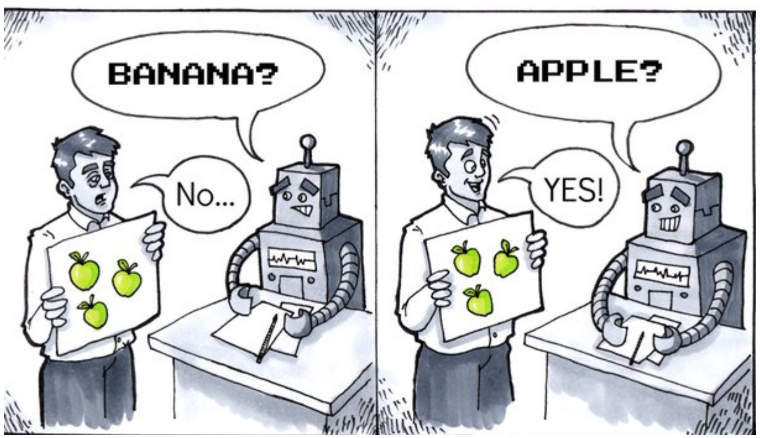


Supervision Required?

Supervised Learning

- Model is provided with labels
- Algorithm learns relationship* between features and labels during training

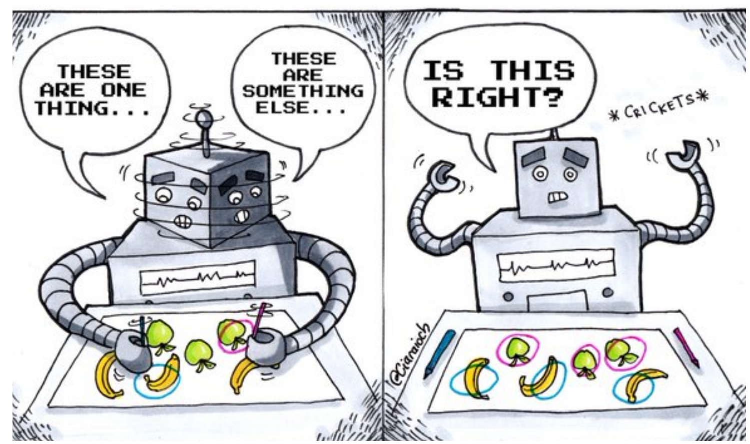
*if that relationship can be expressed as a mathematical function



Un-supervised Learning

- Model takes unlabeled or unclassified data
- Algorithm learns patterns* or groupings in the data during training

*but no promises that those patterns are useful!

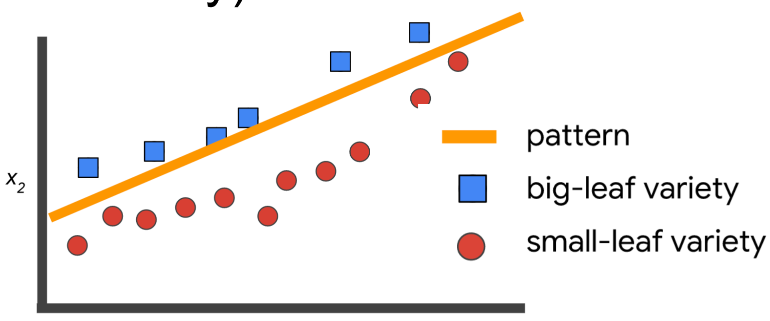


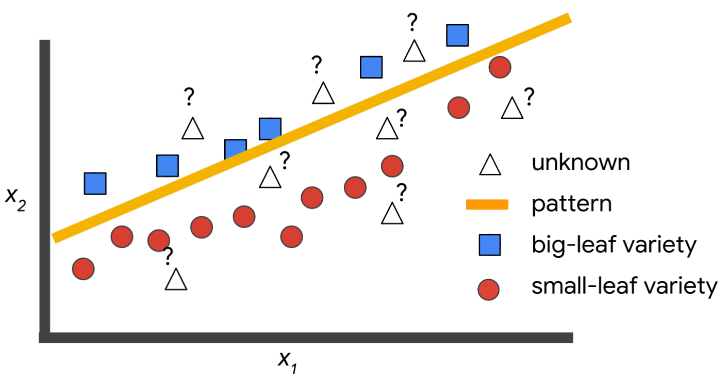
Bonus: Reinforcement Learning – models are rewarded for meeting goals

Supervision Required?

Supervised Learning

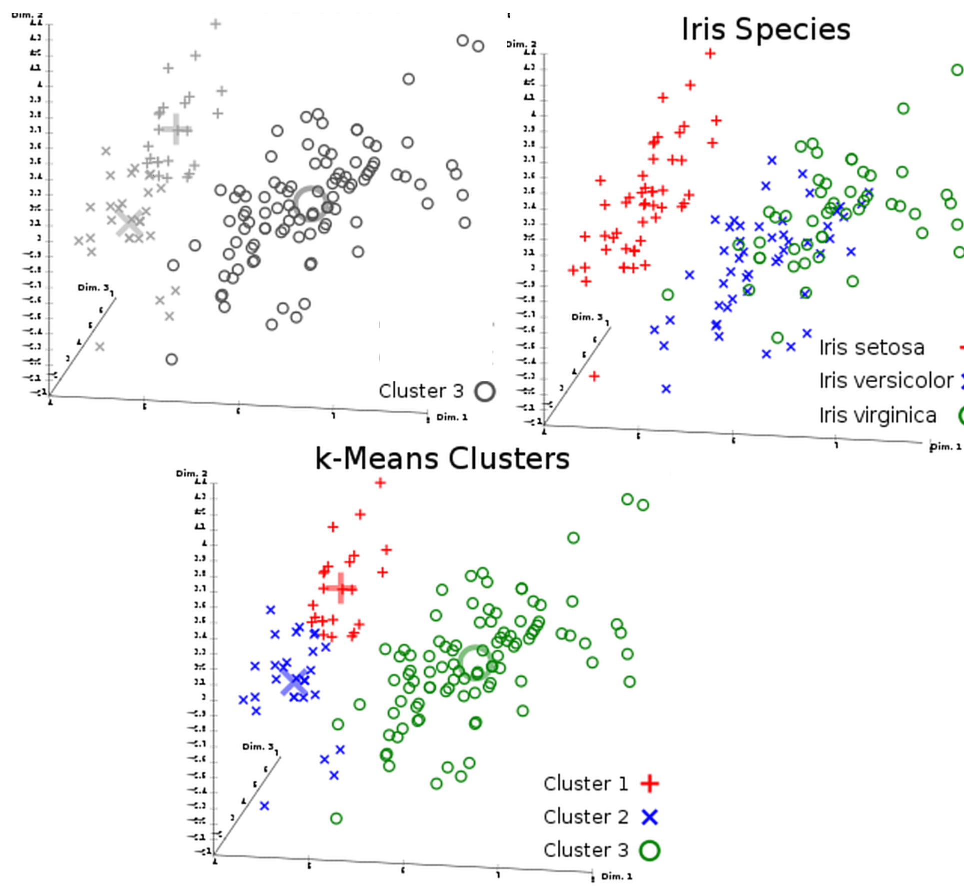
Example: distinguishing between two known classes (learn a decision boundary)



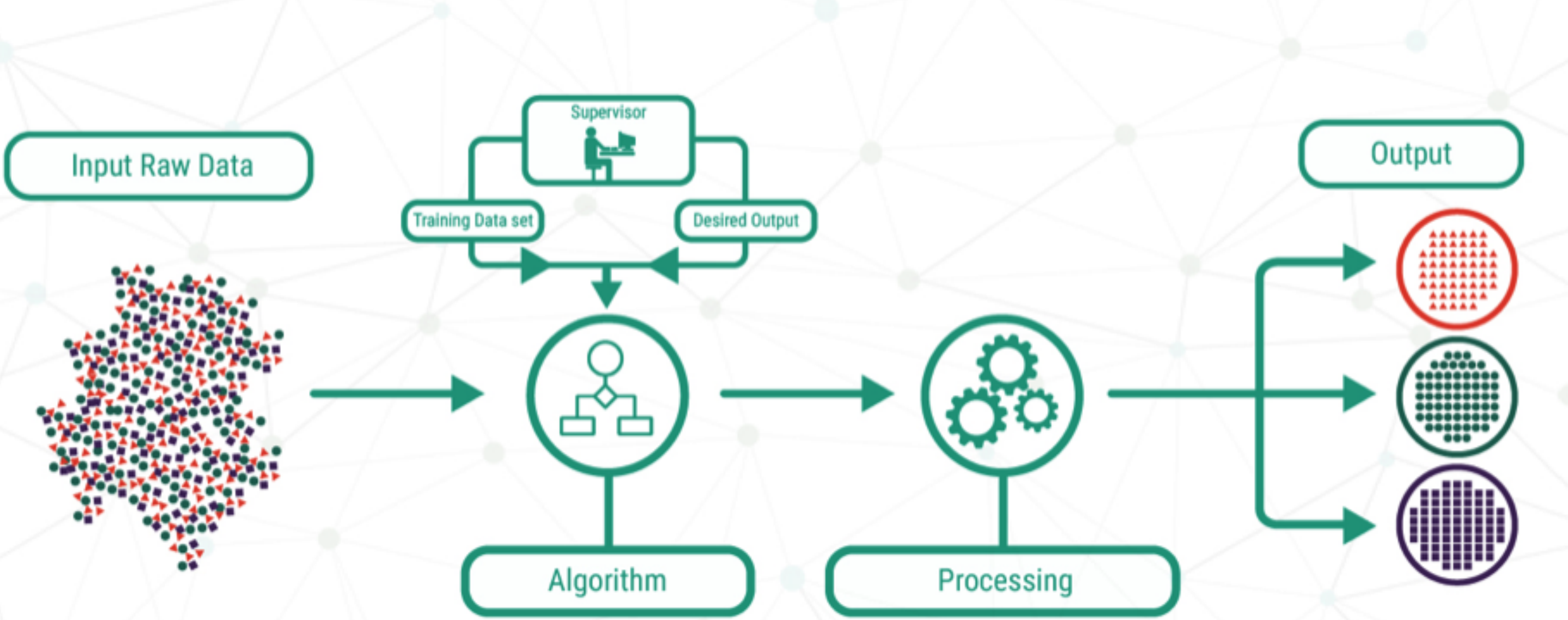


Un-supervised Learning

Example: grouping data into 3 classes (learn relational structure)

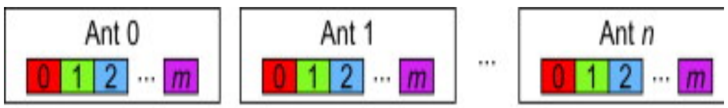


Supervised Algorithms



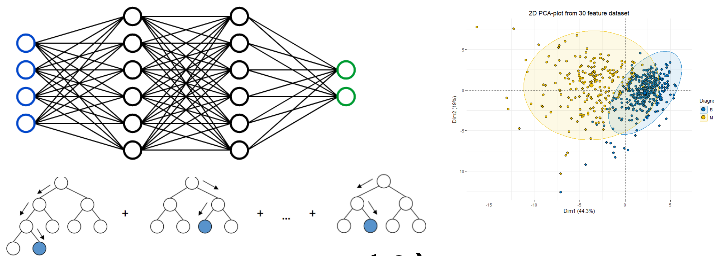
How Do You Actually Do This?

Training data



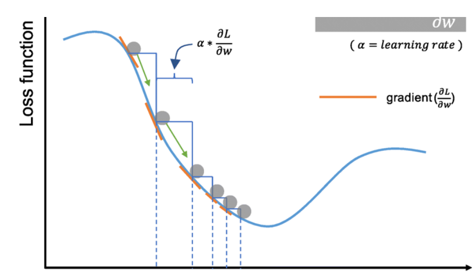
$D_{train} = (\vec{x}, y)$

Choose a Model



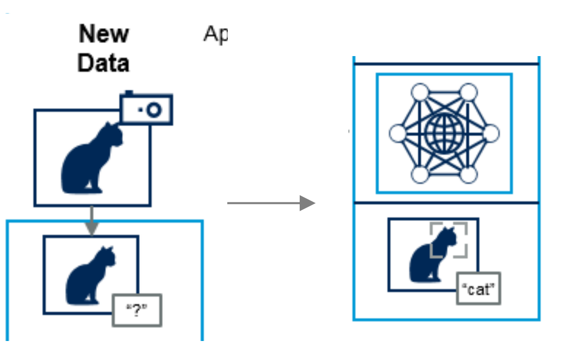
$M(\theta)$

Optimize Parameters



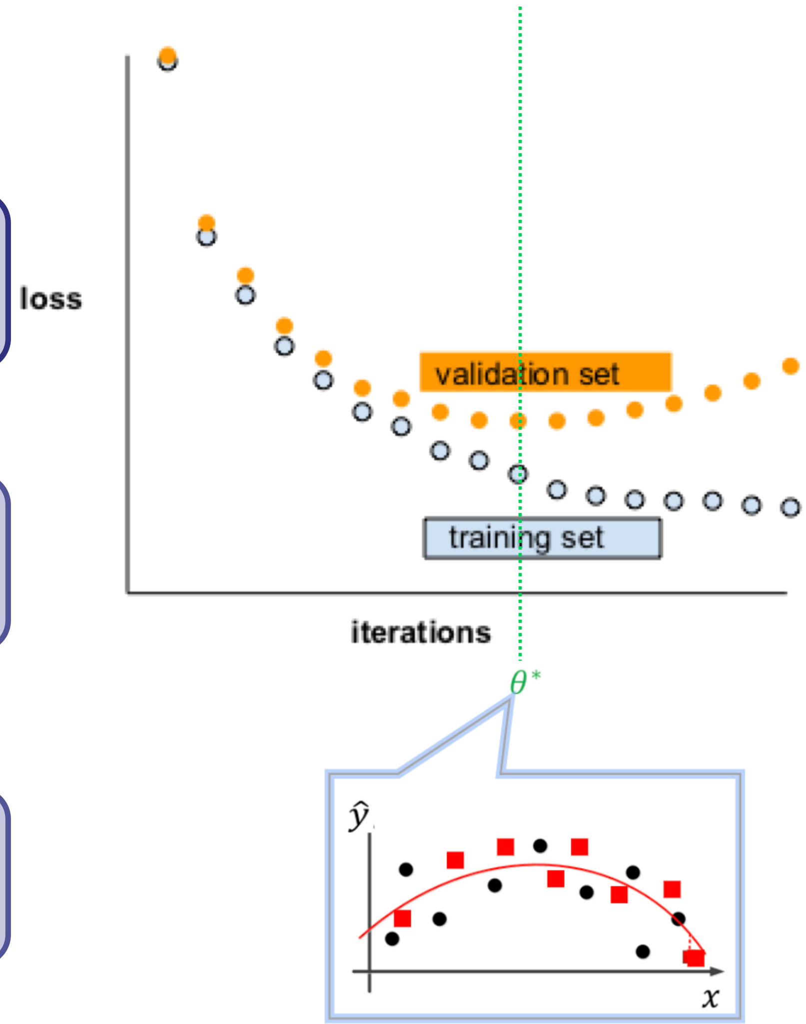
Loss function: L

Make an Inference!



Training a Model

Data Split	Purpose
Training Set	Train Model
Validation Set	Evaluate Model During Training
Test Set	Evaluate Final Model

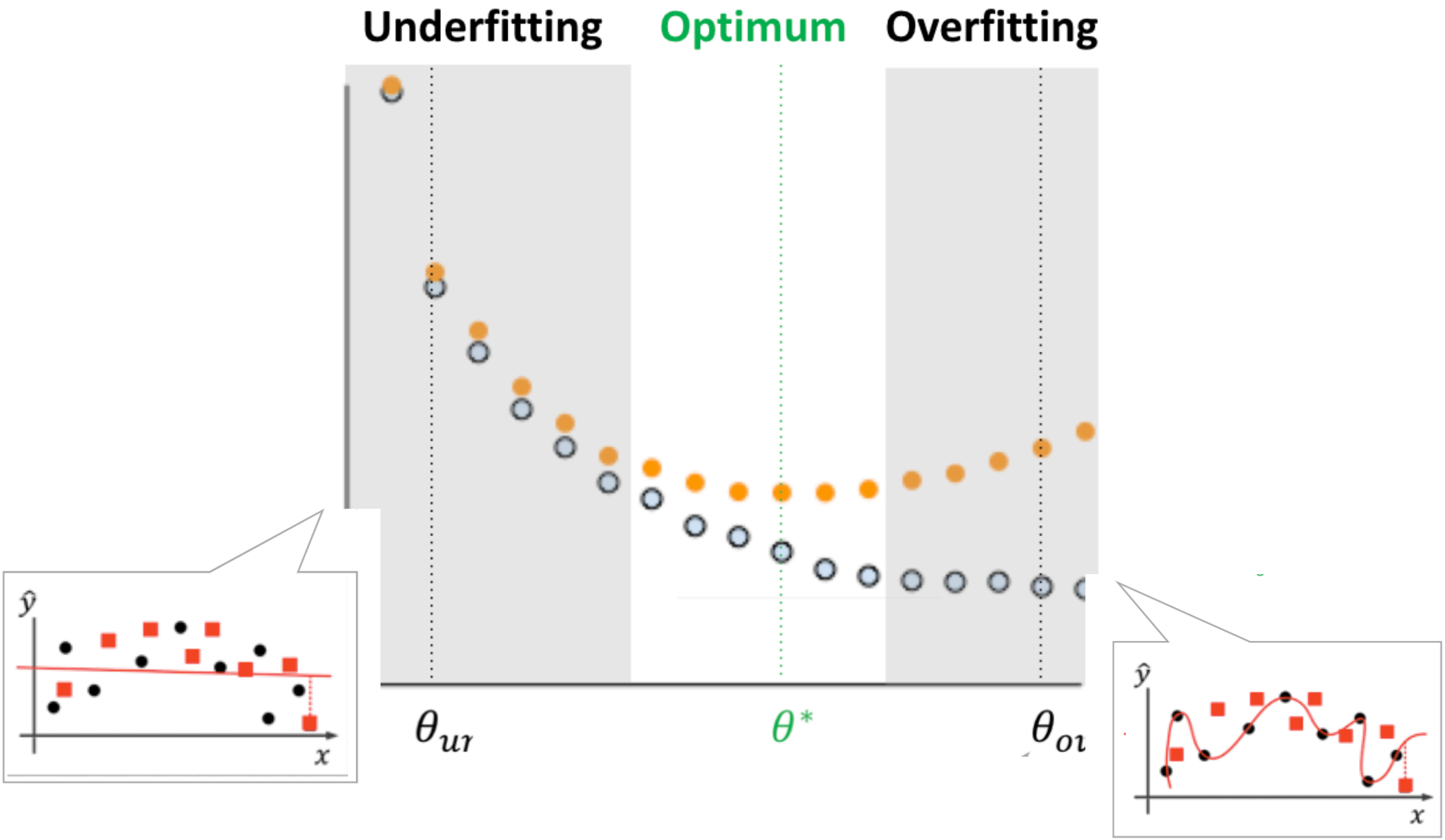


Over and Under Fitting

- Didn't learn general patterns in data
- High bias
- Low variance
- Inaccurate inference

- Learned specific details of training set
- Low bias
- High variance
- Inaccurate inference

Over and Under Fitting



Choosing a Cost Function + Evaluation

Loss Functions

- Common choices:
  - Mean Squared Error

$$MSE = \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{n}$$

  - Hinge Loss

$$SVMLoss = \sum_{j \neq y_i} max(0, s_j - s_{y_i} + 1)$$

  - Cross Entropy

$$CrossEntropyLoss = -(y_i log(\hat{y}_i) + (1 - y_i) log(1 - \hat{y}_i))$$

- Considerations:
  - Type of model
  - Presence of outliers in dataset
  - Sensitive outcomes
  - Training behavior

further reading

Evaluation Metrics

- Common choices:
  - Accuracy $\frac{Number\ of\ Correct\ predictions}{Total\ number\ of\ predictions\ made}$
  - Confusion matrix
  - Sensitivity $\frac{TruePositive}{FalseNegative + TruePositive}$
  - Specificity $\frac{TrueNegative}{TrueNegative + FalsePositive}$
  - False positive rate $\frac{FalsePositive}{TrueNegative + FalsePositive}$
  - Precision $\frac{TruePositives}{TruePositives + FalsePositives}$
  - Recall $\frac{TruePositives}{TruePositives + FalseNegatives}$
  - F1 Score $F1 = 2 * \frac{1}{\frac{1}{precision} + \frac{1}{recall}}$
- Considerations:
  - What type of outcomes matter?
  - Typically should report multiple

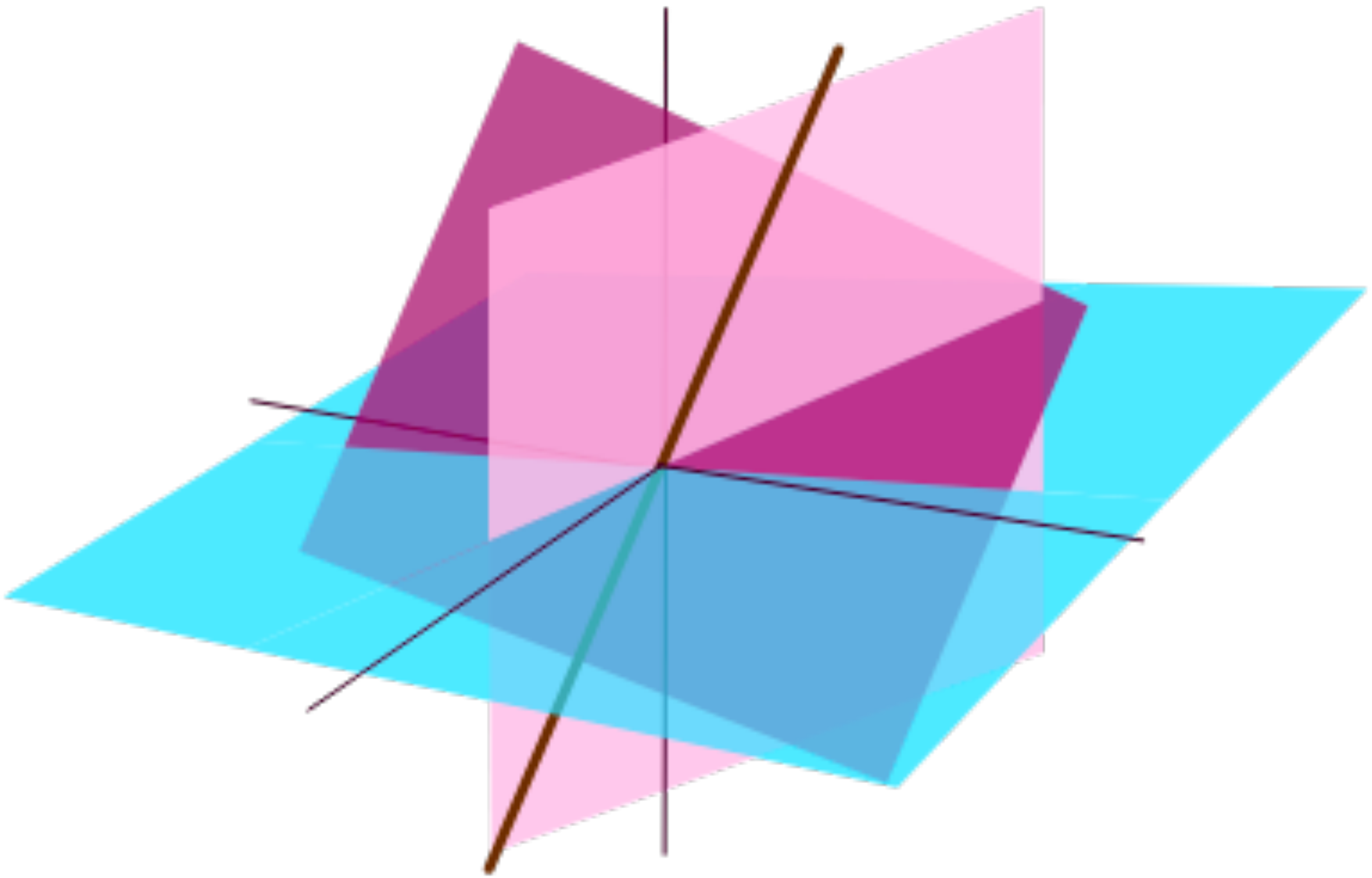
further reading

Question Time!

Can you think of an example of ML in your daily life?:

- Do you think it's a supervised or un-supervised model?
- What data could have been used to train it?
- What might have been the learning objective?
- Can you imagine some important training considerations?

A Quick Linear Algebra Refresher



Matrix Basics

$$\begin{bmatrix} 460 \\ 430 \\ 480 \\ \cdots \\ \cdots \end{bmatrix}$$

Vectors

- A single column of data
- Can represent a single variable, training labels, model parameters

$$\begin{pmatrix} a_1 & a_2 \\ a_3 & a_4 \end{pmatrix}$$

Where $a_{ij} \in R$

Matrices

- m*n matrix: m columns of n elements
- Often represents training data, model parameters, convolutions

Matrix Multiplication

- Dot product of rows and columns
- Rows of first matrix multiplied by columns of second matrix
- Allows dimension transformations

$$\begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{pmatrix} * \begin{pmatrix} 7 & 8 \\ 9 & 10 \\ 9 & 12 \end{pmatrix} = \begin{pmatrix} 52 & 64 \\ 127 & 154 \end{pmatrix}$$

Where,

[1, 2, 3] * [8, 10, 12] = 1 * 8 + 2 * 10 + 3 * 12 = 64

source

Matrix Types + Operations

$$A = \begin{pmatrix} 1 & 4 \\ -2 & 3 \end{pmatrix} \qquad A^T = \begin{pmatrix} 1 & -2 \\ 4 & 3 \end{pmatrix}$$

Matrix Transpose

B is transpose of A if $a_{ij} = b_{ji}$

Matrix Inverse

B is inverse of A if AB=BA

$$A = \begin{pmatrix} 1 & 2 & 1 \\ 4 & 4 & 5 \\ 6 & 7 & 7 \end{pmatrix} \qquad B = \begin{pmatrix} -7 & -7 & 6 \\ 2 & 1 & -1 \\ 4 & 5 & -4 \end{pmatrix}$$

$$AA^T = I = A^T A$$

Where, $A^{-1} = A^T$

Orthogonal Matrix

Square matrix with columns of unit length

Diagonal Matrix

$a_{ij} = 0$ for i!=j, $a_{ij}! = 0$ for i=j

$$A = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Matrices + Models

Matrix multiplication for model inference

No. of Rooms (x0)	Size (Square feet) (x1)	Year Built (x2)	No. of Floors (x3)	Price (y)
2	1434	2010	1	8500
3	1534	2019	2	9600
2	962	1996	3	25880
...	...	...	...	...

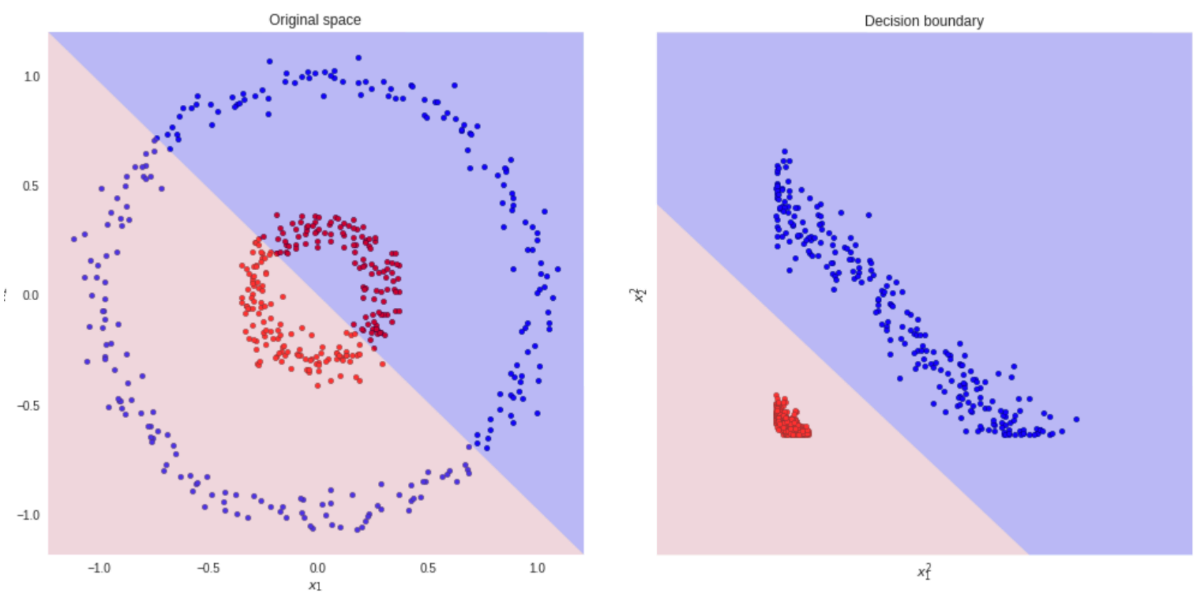
x_i = features of a house

y = target variable

$$h_\theta(x) = \theta_0 \, x0 + \theta_1 \, x1 + \; \cdots \; + \theta_n \, xn = y'$$

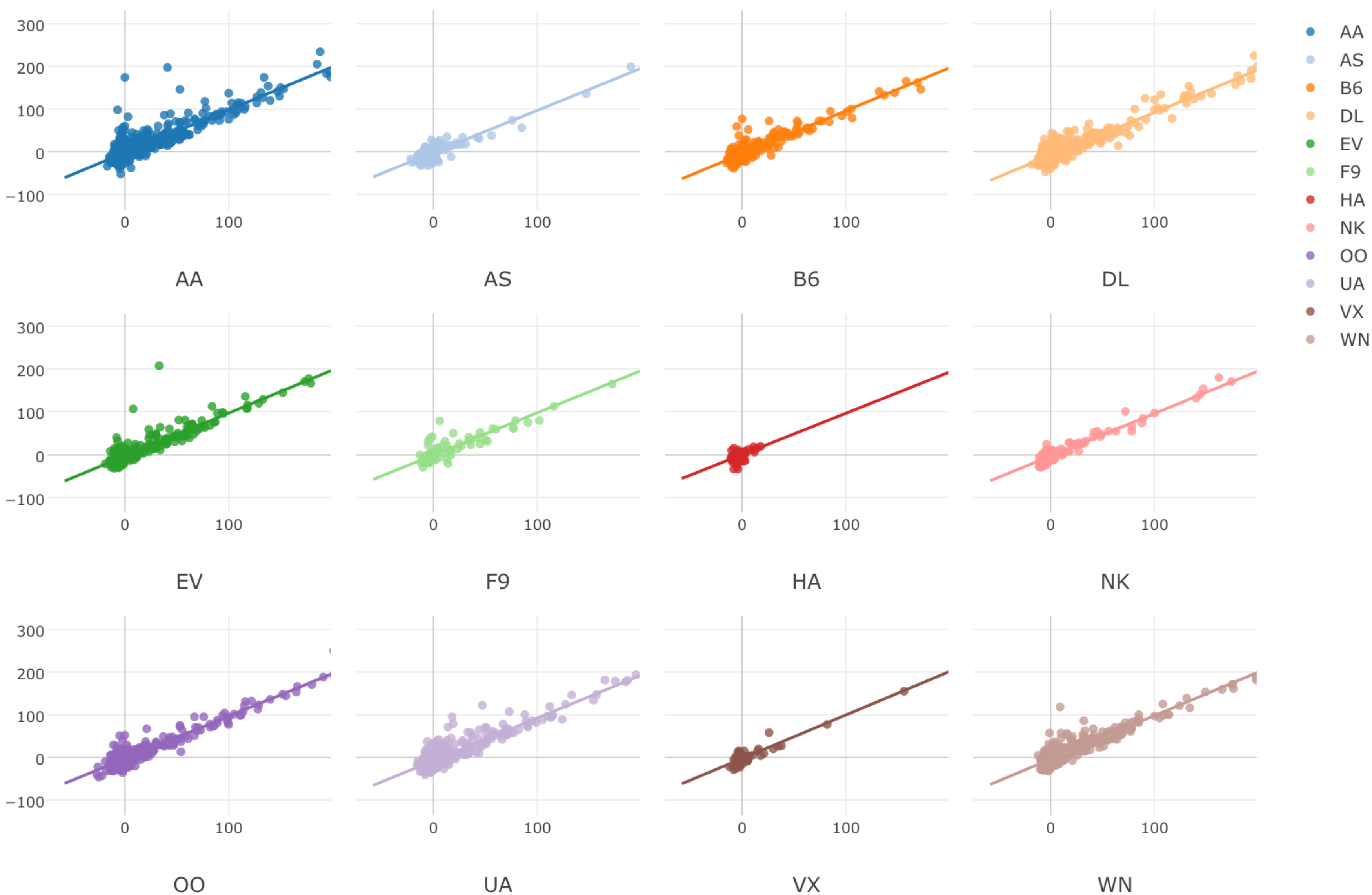
$$x = \begin{pmatrix} x0 \\ x1 \\ x2 \\ x3 \end{pmatrix} \quad \& \quad \theta = \begin{pmatrix} \theta 0 \\ \theta 1 \\ \theta 2 \\ \theta 3 \end{pmatrix}$$

Matrix multiplication for data transformation



$$y_n = \mathbf{W}^T x_n$$

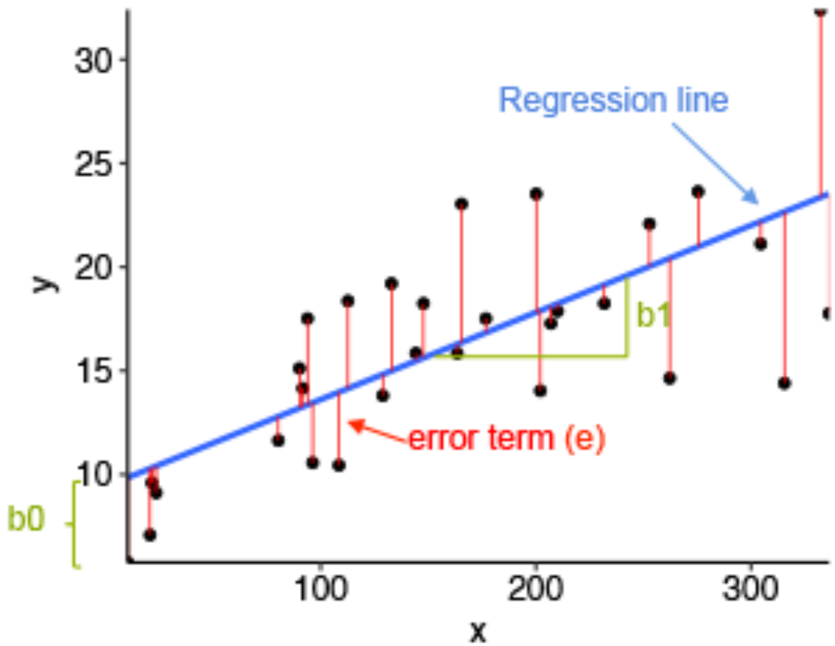
The Simplest 'ML' Models



Linear Regression as ML

Fit the set of dependent and independent data of a dataset to a linear line function

Allows prediction of continuous variables



A specific case of linear models

$$y = y(x) \to y(x_i) = \tilde{y}_i + \epsilon_i = \sum_{j=0}^{n-1} \beta_j x_i^j + \epsilon_i,$$

$$\begin{aligned}
y_0 &= \beta_0 + \beta_1 x_0^1 + \beta_2 x_0^2 + \cdots + \beta_{n-1} x_0^{n-1} + \epsilon_0 \\
y_1 &= \beta_0 + \beta_1 x_1^1 + \beta_2 x_1^2 + \cdots + \beta_{n-1} x_1^{n-1} + \epsilon_1 \\
y_2 &= \beta_0 + \beta_1 x_2^1 + \beta_2 x_2^2 + \cdots + \beta_{n-1} x_2^{n-1} + \epsilon_2 \\
&\dots\dots \\
y_{n-1} &= \beta_0 + \beta_1 x_{n-1}^1 + \beta_2 x_{n-1}^2 + \cdots + \beta_{n-1} x_{n-1}^{n-1} + \epsilon_{n-1}.
\end{aligned}$$

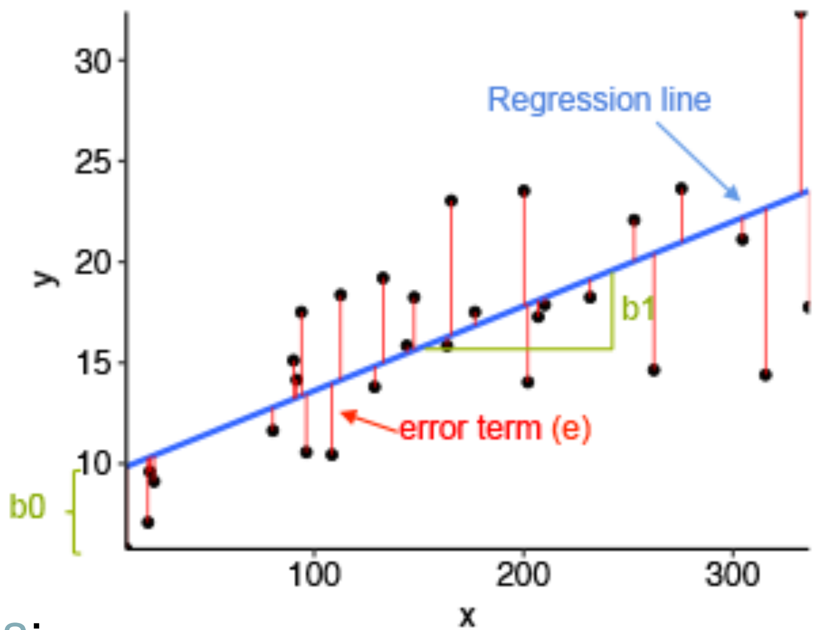
Learn parameters of β *and* ε

$$\boldsymbol{y} = \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}.$$

Linear Regression

Fit the set of dependent and independent data of a dataset to a linear line function

Allows prediction of continuous variables



- Training: Often Ordinary Least Squares
- Inference: use parameters of new data to calculate regression value
- Loss Function: $\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \tilde{y}_i)^2$

further reading

Pros:

- Easy to implement and train
- Fast inference
- Interpretable

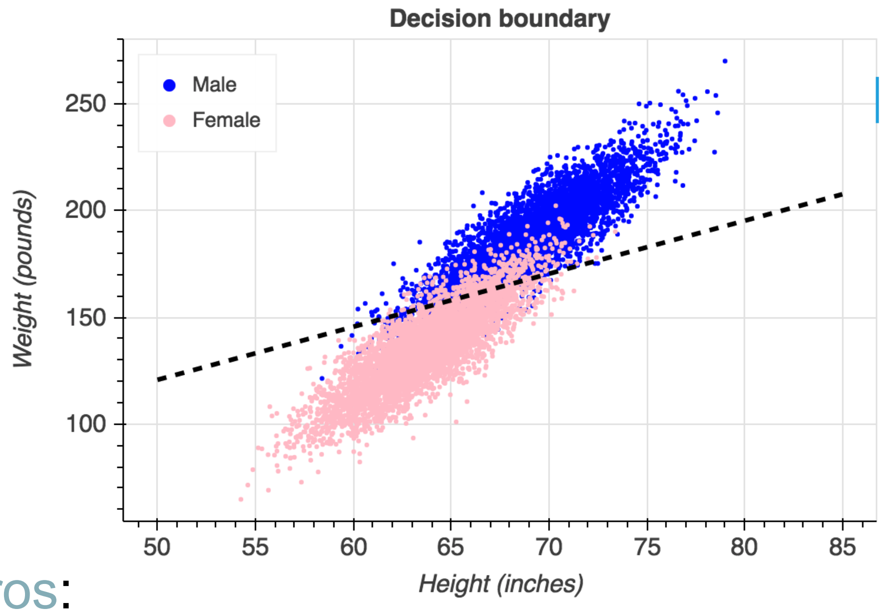
Cons:

- Assumes linear relationship
- Overfits or doesn't converge with correlated features

Logistic Regression

Estimate the logarithmic probability of a class from linear combination of inputs

Assume probability depends linearly on features: $h\theta(X) = \frac{1}{1+e^{-(\beta_0+\beta_1 X)}}$



- Training: GD to adjust slope parameters
- Inference: select a classification threshold
- Loss Function:

$$Cost(h_\theta(x), y) = \begin{cases} -log(h_\theta(x)) & \text{if } y = 1 \\ -log(1-h_\theta(x)) & \text{if } y = 0 \end{cases}$$

further reading

Pros:

- Easy to implement and train
- Fast inference
- interpretable

Cons:

- Assumes linear relationship
- Overfits or doesn't converge with correlated features

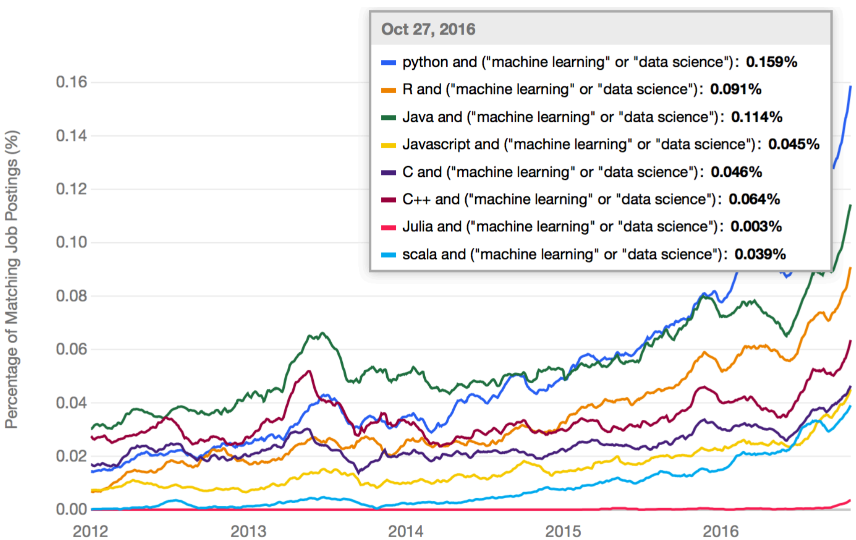
Machine Learning Libraries

Python

Interpreted, high-level, general-purpose programming language

- Most popular language for machine learning
- An extensive ecosystem of open-source libraries
- Easily readable, simple syntax
- Generally no need to compile

```
test_number = 407 # our example is not a prime number

# prime numbers are greater than 1

if test_number > 1:

# check for factors

number_list = range(2, test_number)

for number in number_list:

number_of_parts = test_number // number

print(f"{test_number} is not a prime number")

print(f"{number} times {number_of_parts} is {test_number}")

break

else:

print(f"{test_number} is a prime number")

else:

print(f"{test_number} is not a prime number")
```



further reading

Fundamental Libraries

NumPy

Provides an easy way to handle arrays (matrices) in python

pandas

Provides high-performance data structures and analysis tools (matrix manipulation)

matplotlib

Provides a range of customizable data visualization methods

SciPy

Provides modules for optimization, linear algebra, and mathematical/scientific data processing

Keras

An open-source library providing intuitive support for neural networks and other ML models

- A user-friendly interface for TensorFlow (Google's ML library)
  - Other common ML library is Pytorch (Facebook)
- Contains building blocks of models, optimization and training methods, and data loading functionality
  - However, limited customization

```
model = Sequential()
model.add(Dense(32, activation='relu', input_dim=100))
model.add(Dense(1, activation='sigmoid'))
model.compile(optimizer='rmsprop',
              loss='binary_crossentropy',
              metrics=['accuracy'])

# Generate dummy data
import numpy as np
data = np.random.random((1000, 100))
labels = np.random.randint(2, size=(1000, 1))

# Train the model, iterating on the data in batches of 32 samples
model.fit(data, labels, epochs=10, batch_size=32)
```

Coding Time!

Happy to answer any questions!

link to Day 1 notebook

sthais@princeton.edu @basicsciencesav