Introduction to Machine Learning

Day 1

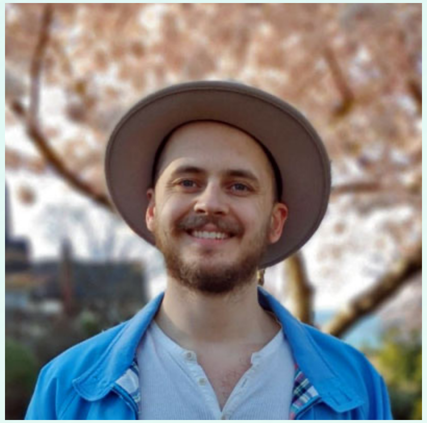
Labatt Impact Lab Bootcamp

Alex Olson

- Studied Artificial Intelligence at the University of Edinburgh
- Came to UofT in 2018 to research applications of machine learning
- Published research applying machine learning to a wide range of disciplines: urban studies, cardiovascular surgery, construction…
- Helped to develop the very first ML Bootcamp in 2019
- Now: research associate at CARTE

Teaching Assistants

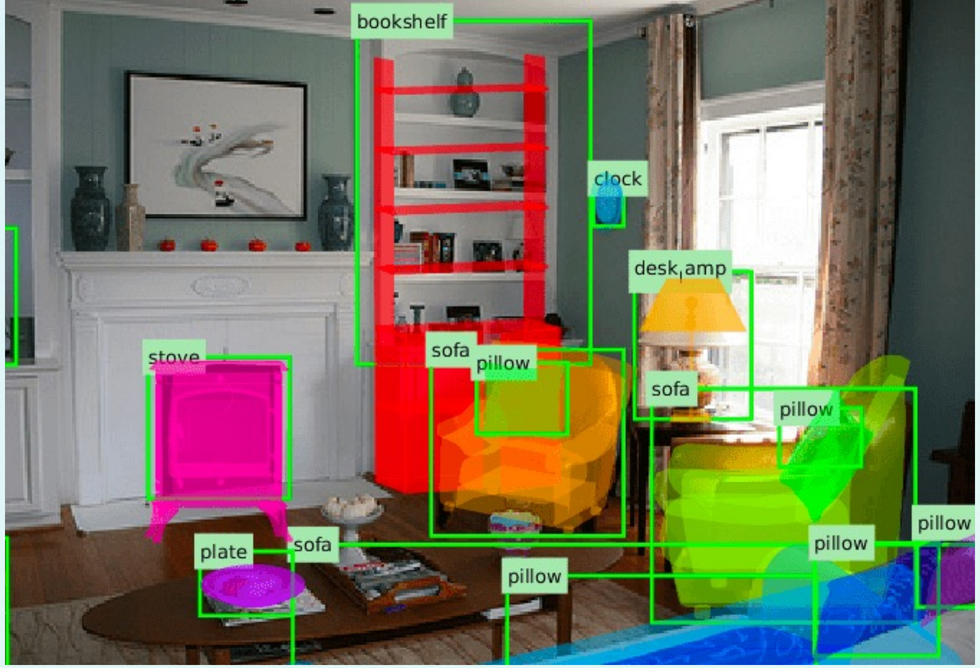
- Christina Seo and Jesse Ward-Bond
- Graduate students under Timothy Chan
- Researching Applied Optimization and Machine Learning



Artificial Intelligence

Getting computers to behave intelligently:

- Perform **non-trivial tasks** as well as humans do
- Perform **tasks that even humans struggle with**



Many sub-goals:

- Perception
- Reasoning
- Control
- Planning

My poker face: AI wins multiplayer game for first time

Pluribus wins 12-day session of Texas hold'em against some of the world's best human players

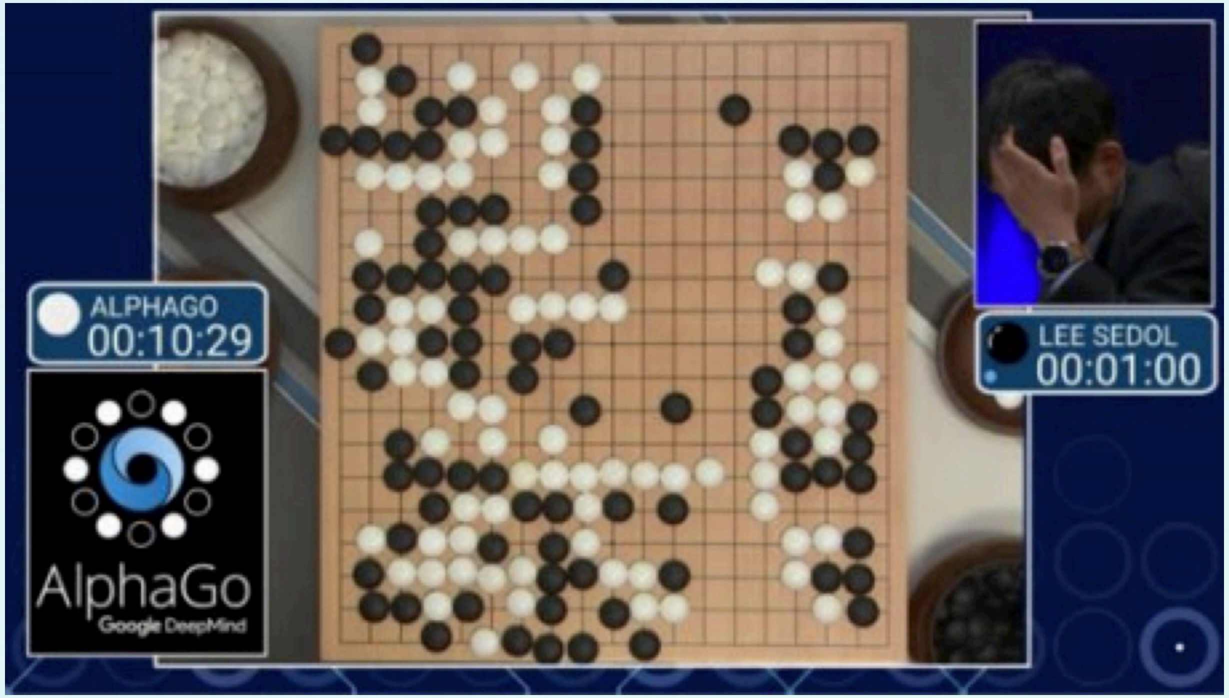
Speech Recognition   Perception + Reasoning

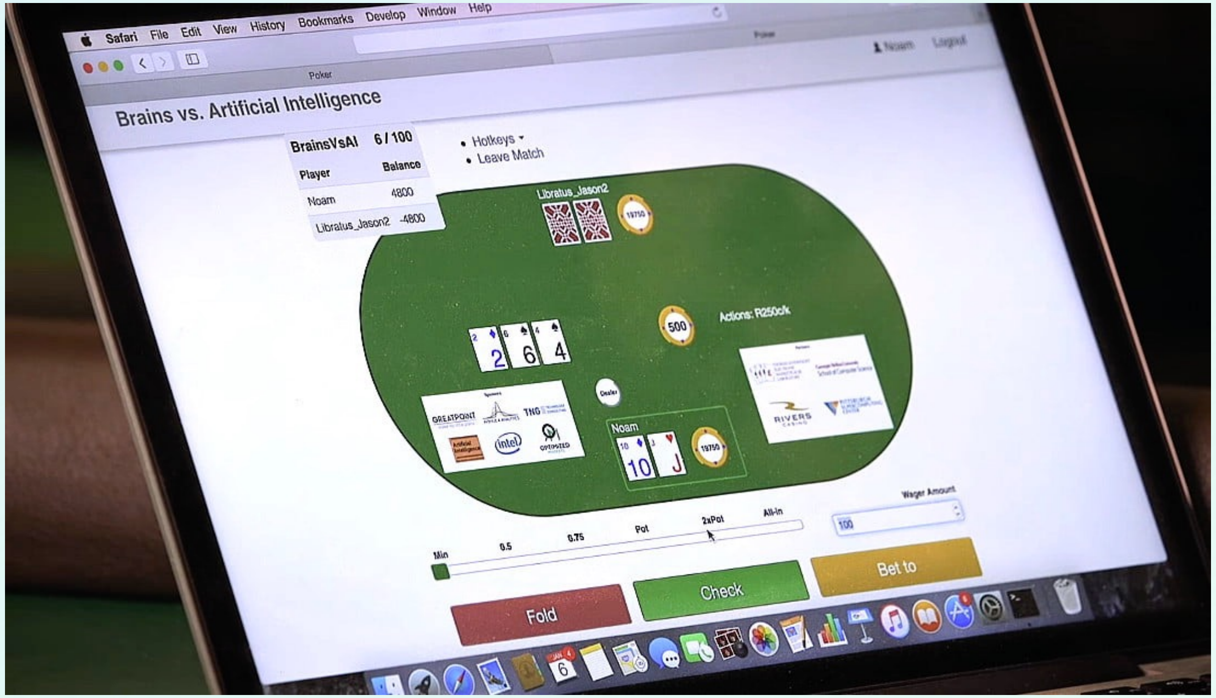
Autonomous Driving

Perception + Reasoning
Control + Planning

Game Playing

Reasoning + Planning



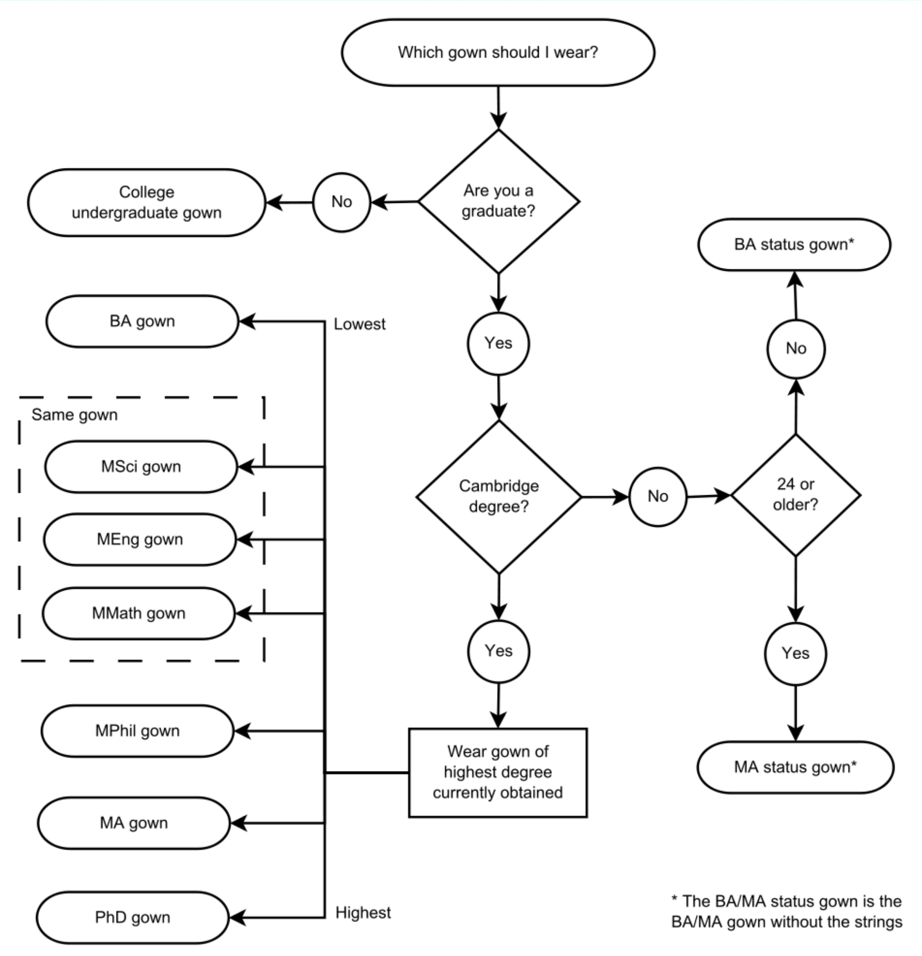


Knowledge-Based AI

Write programs that simulate how people solve the problem

Fundamental limitations:

- Will never get better than a person
- Requires deep domain knowledge
- We don't know how we do some things (e.g., riding a bicycle)



Data-Based AI = Machine Learning

Write **programs that learn** the task **from examples**

✅ No need to know how we do it as humans
✅ Performance should improve with more examples

❌ May need **many examples**!
❌ May not understand how the program works!

Machine Learning:

Study of algorithms that

- Improve their <u>performance</u> P
- At some <u>task</u> T
- With <u>experience</u> E

Well defined task: <P,T,E>

The Machine Learning Process

Experience

- Examples of the form

  (input, correct output)

Task

- Mapping from input to output

Performance

- ”Loss function” that measures error w.r.t. desired outcome

Choices in ML Problem Formulation

Experience

- Examples of the form
  (input, correct output)

Task

- Mapping from input to output

Performance

- "Loss function" that measures error w.r.t. desired outcome

Loan Applications

- What historical examples do I have? What is a correct output?
- Predict probability of default? Loan decision? Credit score?
- Do I care more about minimizing False Positives? False negatives?

Machine Learning:

Study of algorithms that

- Improve their performance P Optimization, Evaluation
- At some task T Classification, regression, clustering
- With experience E Tabular, image, sequence

Well defined task: <P,T,E>

How will I rate “Chopin’s 5th Symphony”?

Song	Rating
Some nights	⭐⭐⭐⭐⭐
Skyfall	⭐
Comfortably numb	⭐⭐⭐
We are young	⭐⭐⭐⭐
…	…
…	…
Chopin’s 5th	???

Classification: Three Elements

1. **Data**:
   - x: data example with *d* attributes
   - y: label of example (what you care about)

2. Classification **model**: a function $f_{(a,b,c,\ldots)}$
   - Maps from X to Y
   - (a,b,c,…) are the **parameters**

3. **Loss** function:
   - Penalizes the model's mistakes

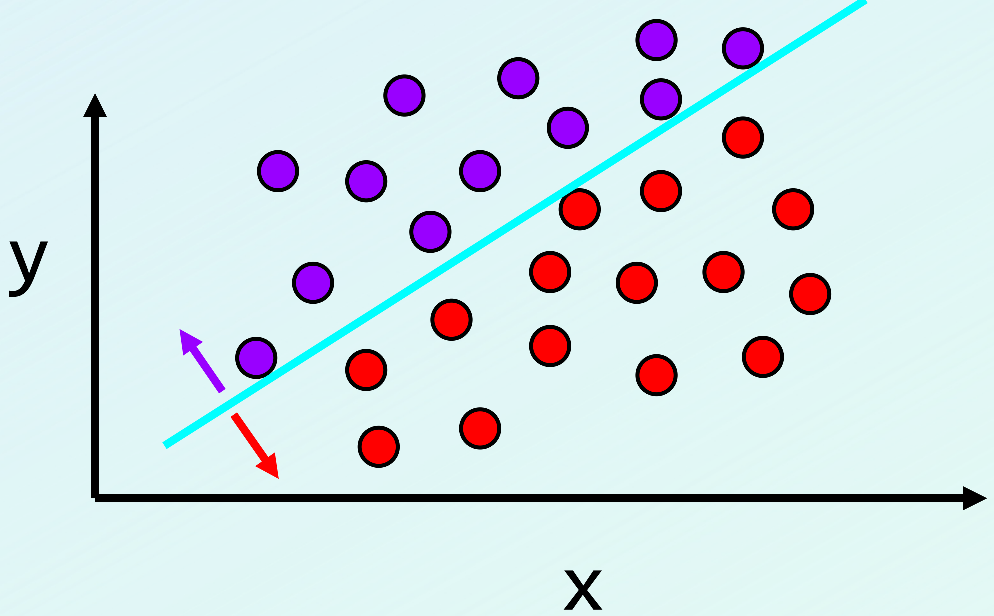
Song	Rating
Some nights	⭐⭐⭐⭐⭐
Skyfall	⭐
Comfortably numb	⭐⭐⭐
We are young	⭐⭐⭐⭐
…	…
…	…
Chopin's 5^{th}	???

Terminology Explanation

Song	Artist	Length	…	Rating
Some nights	Fun	4:23	…	⭐⭐⭐⭐⭐
Skyfall	Adele	4:00	…	⭐
Comf. Numb	Pink Floyd	6:13	…	⭐⭐⭐
We are young	Fun	3:50	…	⭐⭐⭐⭐
…	…	…	…	…
…	…	…	…	…
Chopin's 5th	Chopin	5:32	…	???

Data example = data instance
Attribute = feature = dimension
Label = target attribute

What is a "model"?



A **useful approximation** of the world

Typically, there are **many reasonable models** for the same data

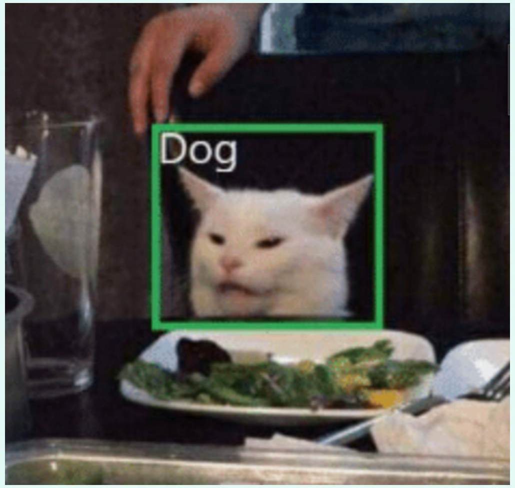
Training a model = finding appropriate values for (a,b,c,…)

- An **optimization** problem
- "appropriate" = **minimizes the Loss (cost) function**
- We will focus on a common training algorithm later on

Classification Loss Function

- How unhappy are you with the answer that the model gave?

- $\mathbf{L_{0\text{-}1}(y, f(x))} = \begin{cases} 1 & \text{if: } \mathbf{y \neq f(x)} \\ 0 & \text{otherwise} \end{cases}$



- **0-1 loss** function: intuitive but hard to optimize = train

- In practice, we use **approximations** of the 0-1 loss – getting warmer or getting colder

Why should this work at all?

The main theoretical basis of ML:

With a **sufficient amount of "similar" data**

+

an **expressive model class**:

Minimizing the loss function on the training data yields a highly accurate model on **unseen test data**, with high probability

1. **Data**: $S = \{(x_i, y_i)\}_{i = 1,\ldots,n}$
   - x_i: data example with d attributes
   - y_i: label of example (what you care about)

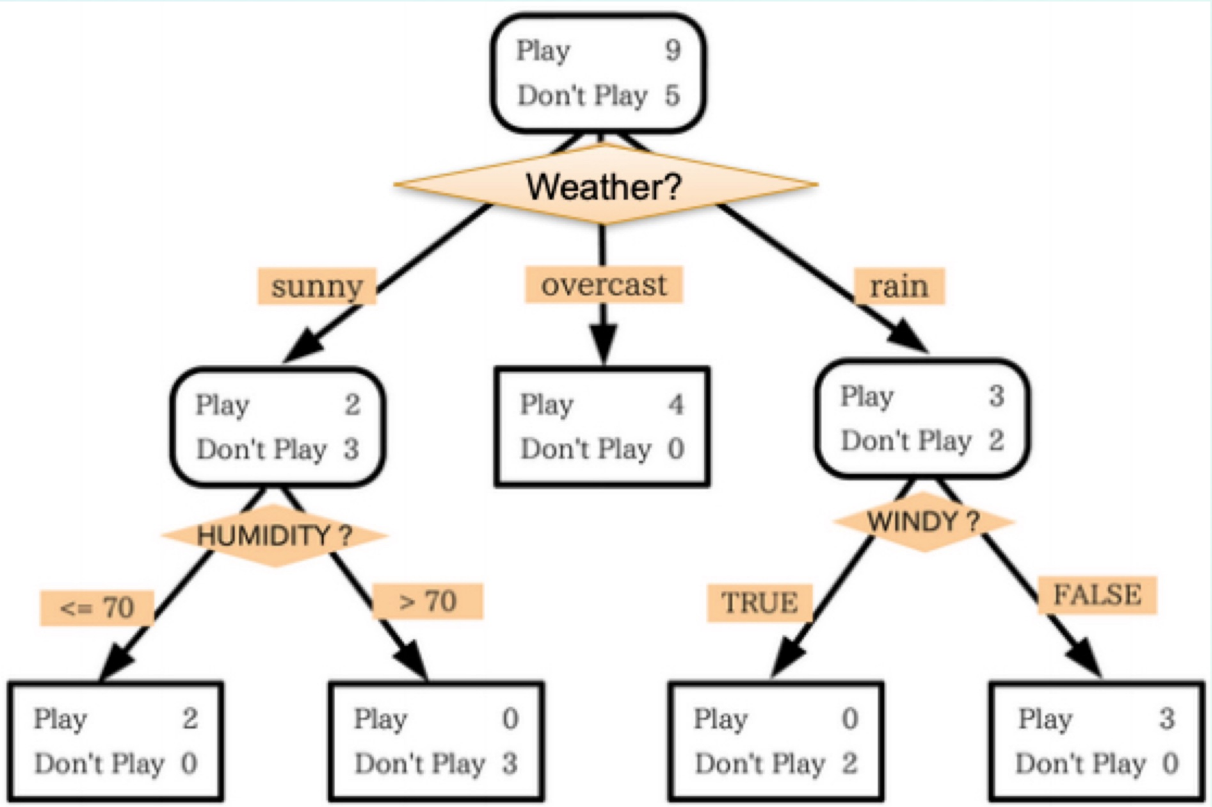
2. Classification **model**: a function $f_{(a,b,c,\ldots)}$
   - Maps from X to Y
   - (a,b,c,…) are the **parameters**

3. **Loss** function: **L(y, f(x))**
   - Penalizes the model's mistakes

Decision Trees: To play tennis or not to?

- **Data**: attributes describing the weather; (sunny? humidity level, …)
- **Target**: 1 if it's good to **Play**, 0 otherwise

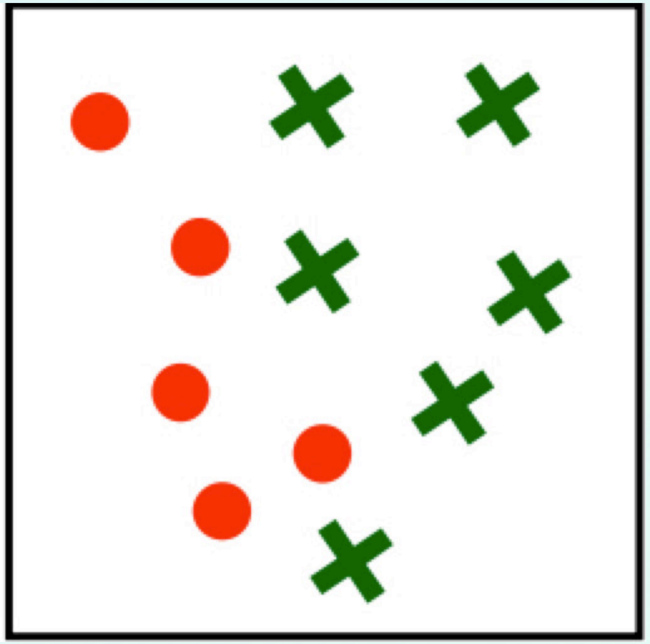
- **Model:** $f_T(x)$
- **Model parameters**: T, the tree structure (and size)



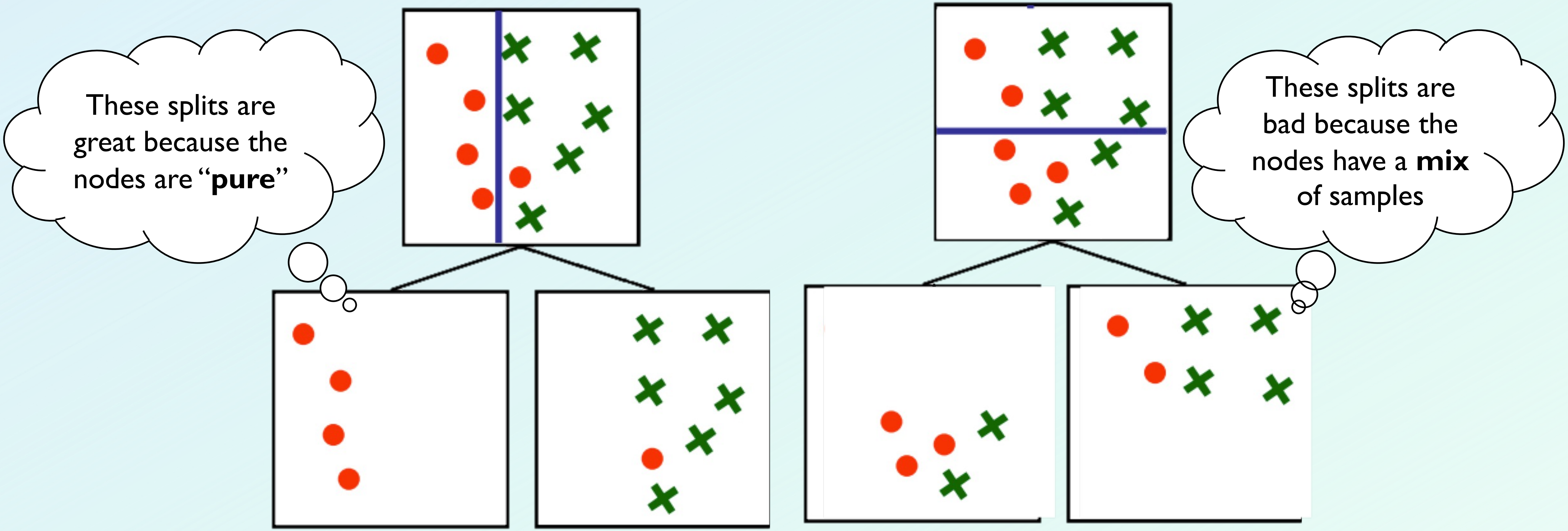
Training (fitting) a Decision Tree

How to choose the attribute/value to split on at each level of the tree?

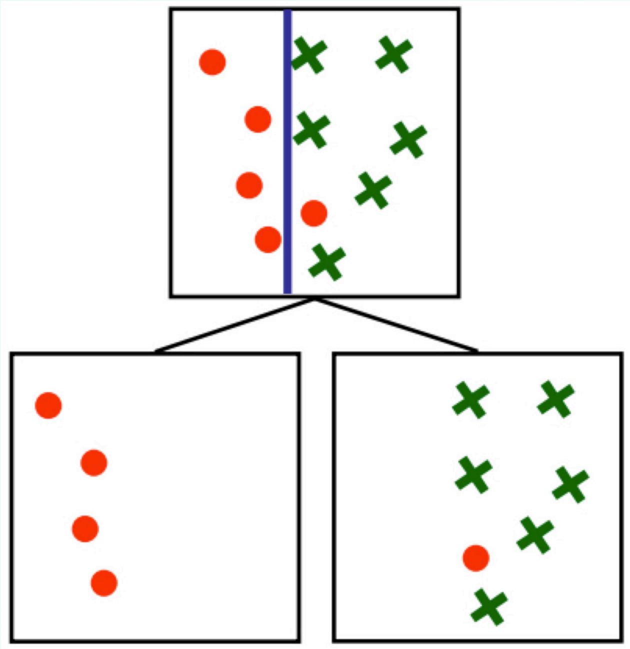
- Two classes (red circles / green crosses)
- Two attributes: X and Y
- 11 points in training data
- Idea: construct a decision tree such that the leaf nodes correctly predict the class for all the training examples



Training (fitting) a Decision Tree



Training (=fitting) a Decision Tree



1. Find the **best attribute** to split on
2. Find the **best split** on the chosen attribute
3. Repeat 1 & 2 until **stopping criterion** is met

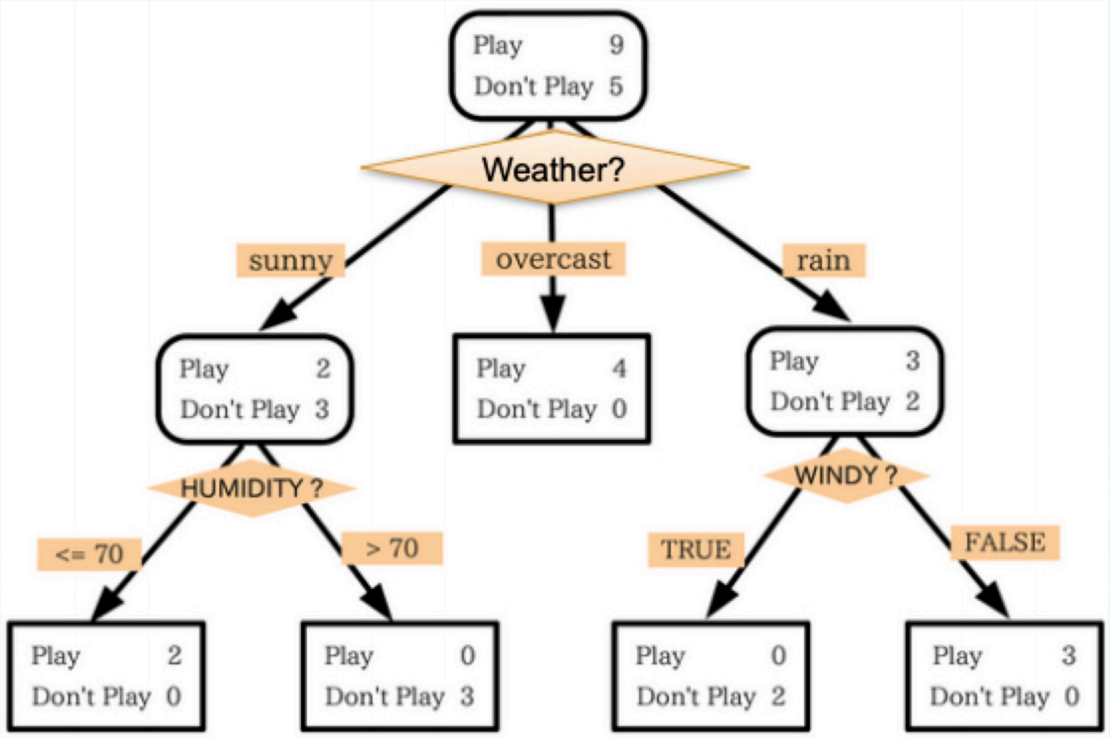
Common **stopping criteria**:

- Node contains very few data points
- Node is pure: most training data in node have same label

Final words on Decision Trees

Advantages

- Simple interpretation
- Fast predictions
- Handles mixed-type attributes



Caveats

- May be too simple for complex data
- Hard to figure out the right depth, stopping criterion, especially at the node level

Forecasting

Decision Trees predict **discrete** outcomes

- Given X features, what is y outcome?

What about something **continuous?**

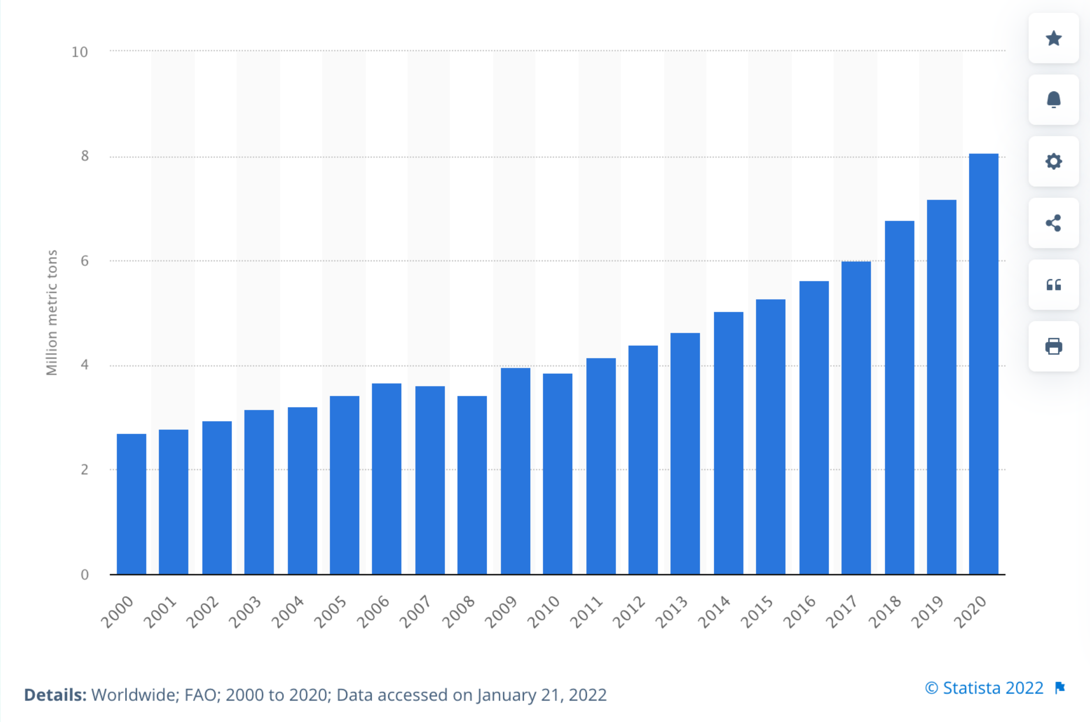
- How is something likely to change over time?

- Time series data tracks a value over time
  - e.g. temperature in one location, company sales data
- Often we want to predict what will happen *next*, i.e. in sequence

Time Series Data: Three Main Components

1. Trend: the long-term trajectory of the data

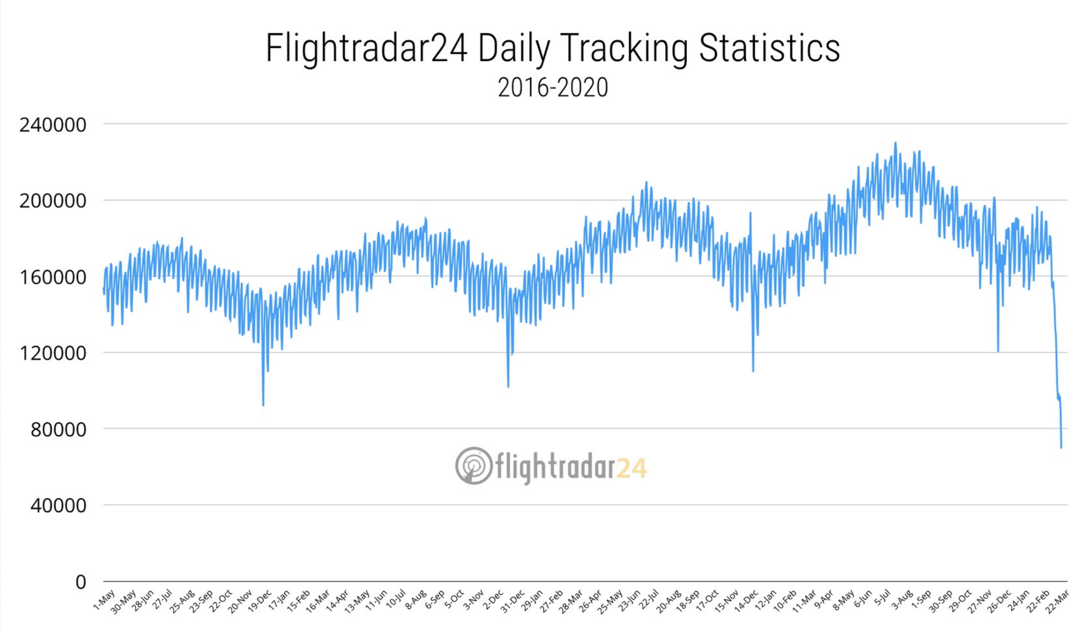
Avocado production is steadily increasing year-on-year



Time Series Data: Three Main Components

2. Seasonality: how the data changes according to the calendar

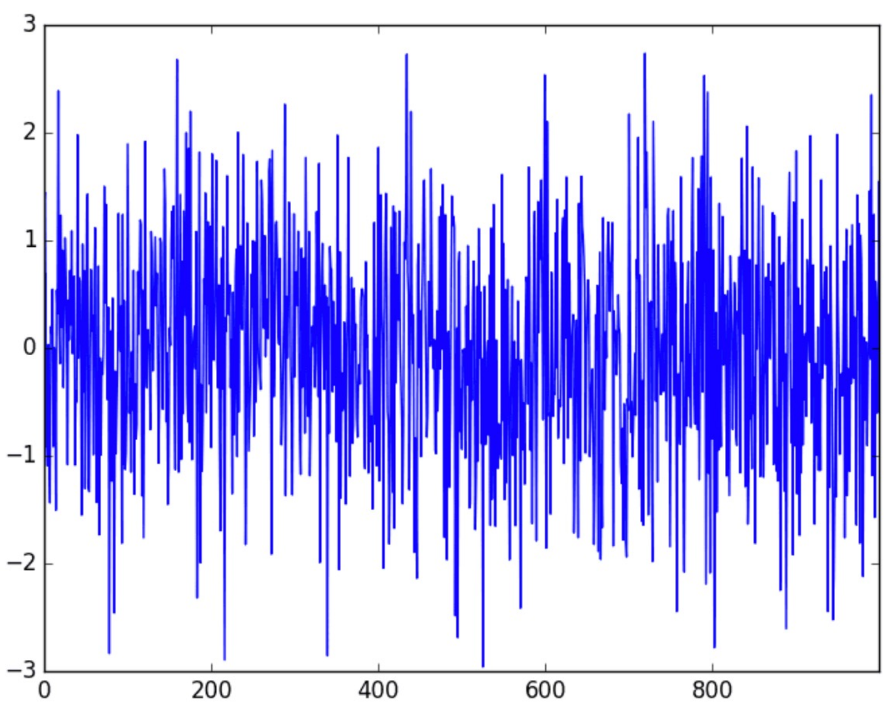
This is highly domain specific – e-commerce platforms experience the most sales around Christmas, but homes are sold most frequently in the summer



Time Series Data: Three Main Components

3. Noise: The random element left over

Noise comes from short-term changes that are unpredictable. More noise makes it harder to anticipate change



White noise (Image by Morn from Wiki Media Commons)

Goals of Time Series Analysis

1. Describe the data: represent the entire dataset compactly
2. Interpretation: understand the factors that drive change in the data
3. <u>Forecasting:</u> Predict what will come next!