

Machine Learning

Dia 1 - Introdução

ImageU - Grupo de Pesquisa em Machine Learning e Visão Computacional

<https://imageu.github.io/>

Curso de Verão 2022

Instituto de Matemática e Estatística - IME USP



1. Apresentação e Informações Importantes
2. Machine Learning - Visão Inicial
3. Ponto de vista computacional
4. Como funciona?

Apresentação e Informações Importantes

- Este curso foi preparado pelos membros do ImageU, Grupo de Pesquisa em Machine Learning e Visão Computacional do IME USP:
 - prof. dra. Nina Hirata (coordenadora)
 - prof. dr. Roberto Hirata Jr.
 - Ana Martinazzo
 - Artur Oliveira
 - Mateus Espadoto
 - Tiago Bomventi

- Avaliação e Frequência:
 - Não haverá prova ou lista de exercícios
 - 85% de Frequência (significa “vir” em todas as aulas :-)
 - Certificados serão enviados por email no fim do mês de Abril

Principais Referências

- Teoria:
 - *Learning from Data*, Lin, Malik e **Mostafa**, 2012
 - <https://work.caltech.edu/telecourse>
- Prática:
 - *Hands-On Machine Learning with Scikit-Learn and TensorFlow*, Aurélien Geron, 2017 (1st), 2019 (2nd)
 - *Deep Learning with Python*, François Chollet, 2017

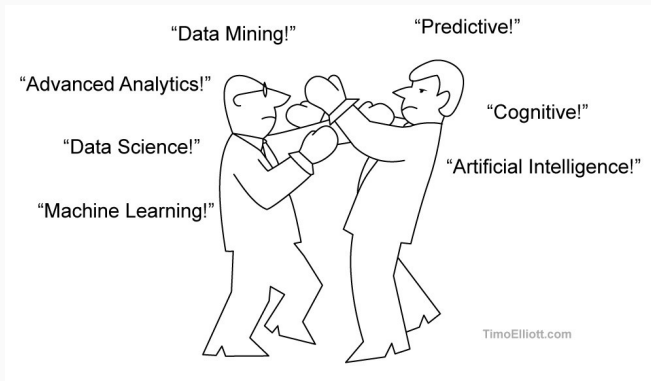
Programa do Curso

- Dia 1: Machine Learning: Introdução
- Dia 2: Regressão Linear
- Dia 3: Regressão Logística e Gradient Descent
- Dia 4: Stack Python, Regressão Linear e Logística do Zero
- Dia 5: Métricas e Seleção de Modelos, Regularização
- Dia 6: Pré-processamento, Regressão Logística Multi-classe
- Dia 7: Árvores e Ensembles
- Dia 8: Redes Neurais e Deep Learning
- Dia 9: Grid Search e Seleção de Variáveis
- Dia 10: Aprendizado não-supervisionado: Clustering

Machine Learning - Visão Inicial

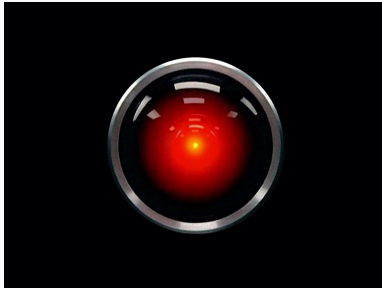
Machine Learning

- Machine Learning nos remete à inteligência computacional
- Inteligência computacional é uma área em desenvolvimento



Inteligência Artificial

- IA \rightsquigarrow máquinas inteligentes, capazes de raciocinar
- Ideia permeia a ficção desde sempre (robos, androides)



- O que mudou desde 1956? (quando John McCarthy cunhou o termo IA)

IA: Onde tem funcionado



Deep Blue vs. Kasparov



Deep Blue
IBM chess computer



Garry Kasparov
World Chess Champion

First match

- February 10, 1996: took place in [Philadelphia, Pennsylvania](#)
- Result: **Kasparov**-Deep Blue (4-2)
- Record set: First computer program to defeat a world champion in a *classical game* under tournament regulations

Second match (rematch)

- May 11, 1997: held in [New York City, New York](#)
- Result: **Deep Blue**-Kasparov (3½-2½)
- Record set: First computer program to defeat a world champion in a *match* under tournament regulations

(wikipedia)

IA: Onde tem funcionado

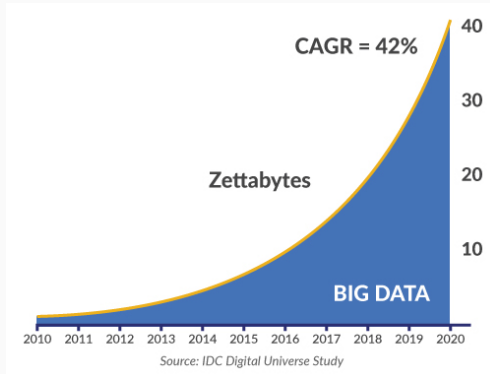


(AlphaGo - Google DeepMind)

IA: Onde tem funcionado

- Carros autônomos
- Assistentes digitais
- Biometria
- e muitas outras áreas

Razão 1: Volume de dados



zettabyte - one sextillion (10^{21}) ou 2^{70} bytes

Razão 2: Poder computacional

1 The accelerating pace of change ...



2 ... and exponential growth in computing power ...

Computer technology, shown here climbing dramatically by powers of 10, is now progressing more each hour than it did in its entire first 90 years

COMPUTER RANKINGS

By calculations per second per \$1,000



Analytical engine
Never fully built, Charles Babbage's invention was designed to solve computational and logical problems



Colossus
The electronic computer, with 1,500 vacuum tubes, helped the British crack German codes during WW II



UNIVAC I
The first commercially marketed computer, used to tabulate the U.S. Census, occupied 943 cu. ft.

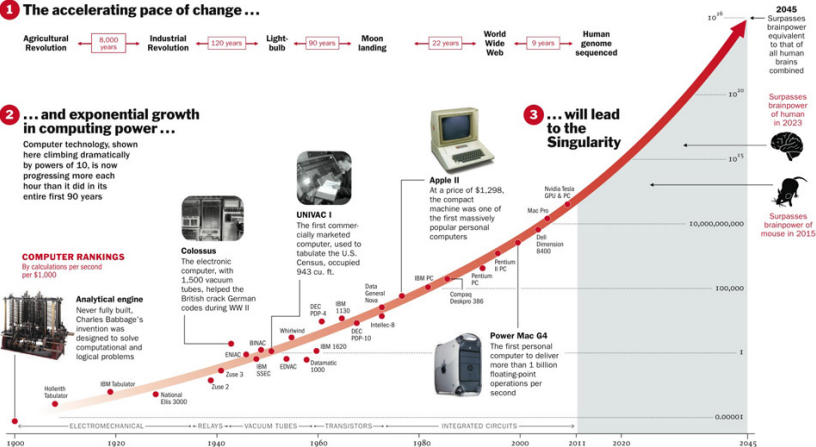


Apple II
At a price of \$1,298, the compact machine was one of the first massively popular personal computers



Power Mac G4
The first personal computer to deliver more than 1 billion floating-point operations per second

3 ... will lead to the Singularity



(Time)

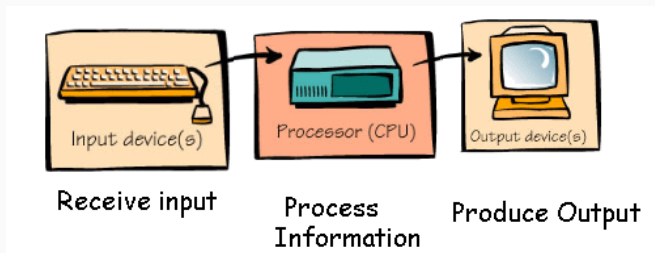
O que é Machine learning?

- Área ampla (teoria, algoritmos, métodos, aplicações)
- Esqueça paralelos com “como uma criança aprende”
- Fundamentos: álgebra, cálculo, probabilidade, estatística e computação

Ponto de vista computacional

ML: ponto de vista computacional

- Computadores processam dados segundo um algoritmo
- Algoritmos são soluções para problemas computacionais



<http://gebar.weebly.com/inputoutput-devices.html>

Problema: somar números

- Dada uma lista de números, calcular sua soma
- Os exemplos a seguir mostram a relação entrada-saída:

Entrada	Saída
3, 1, 7	11
0.25, 0.75, 0.5, 0.1	1.6
1, 3, 5, 7, 9	25

Problema: somar números

- Dada uma lista de números, calcular sua soma
- **Algoritmo:**

Entrada: uma lista de números a serem somados

Saída: a soma (total) dos números na lista

SOMA = 0

Enquanto lista de números não está vazia

 Coloque o próximo número da lista em NUM
 (remove o número da lista)

 SOMA = SOMA + NUM

Imprima SOMA

Problema: ordenação

- Em vez de definir o problema, dê-me exemplos!
- Problema de ordenação:

Entrada	Saída
9, 2, 0, -1, 7, 4	-1, 0, 2, 4, 7, 9
'x', 'a', 'm', 'b'	'a', 'b', 'm', 'x'

- Nós sabemos como resolver (computar) esse problema:
 - selection sort
 - bubble sort
 - merge sort
 - quick sort

- Desenhar algoritmos eficientes
- Garantir que o algoritmo está correto
- Fazer implementações eficientes
- Garantir que a implementação do algoritmo está correta

E quando não temos/sabemos um algoritmo?

- Como descrever formalmente a relação a seguir:



\Rightarrow 'a'



\Rightarrow 'X'



\Rightarrow 'A'



\Rightarrow 'd'



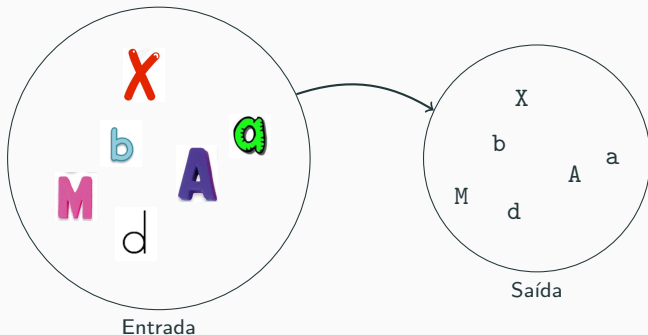
\Rightarrow 'b'



\Rightarrow 'M'

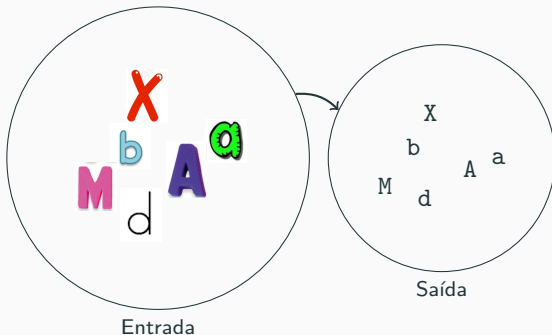
E quando não temos/sabemos um algoritmo?

- E se sabemos caracterizar a relação entrada-saída por meio de exemplos, ainda que a relação seja ambígua:



ML entra em cena

- Ideia central de ML: “aprender” a relação a partir de exemplos
- Produto final de ML: um programa que realiza o processamento entrada-saída



Aplicação: Classificação de dígitos manuscritos (MNIST)



The MNIST Database

(Yann LeCun *et. al*)

Melhor desempenho

convnet, erro 0.23%

(Ciresan *et al.* CVPR 2012)

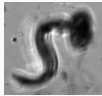
<http://yann.lecun.com/exdb/mnist/>

Aplicação: Classificação de imagens de galáxia



Espiral × elíptica

Aplicação: Classificação de imagens de plâncton



Appendicularia



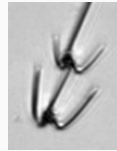
Bubble



Calanoida



Dinoflagellate



Dinoflagellate M



Chaetoceros



Cnidaria



Copepoda



Coscinodiscus



Cyclopoida



Detritus



Detritus ball



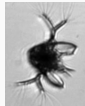
Filaments



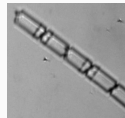
Nauplii



Noctiluca



Penilia

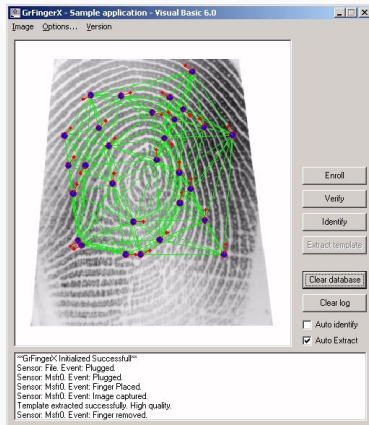
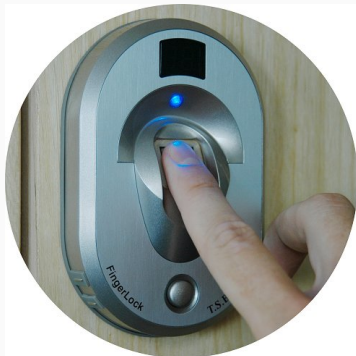


P. Dactyliosolen



Stalked c.

Aplicação: Controle de acesso com biometria



Aplicação: Sistemas de vigilância – detecção de eventos

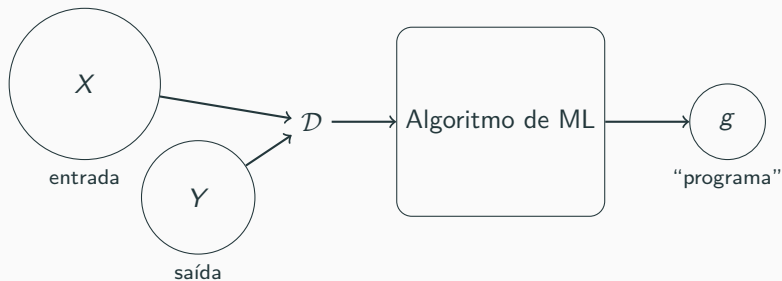


Como funciona?

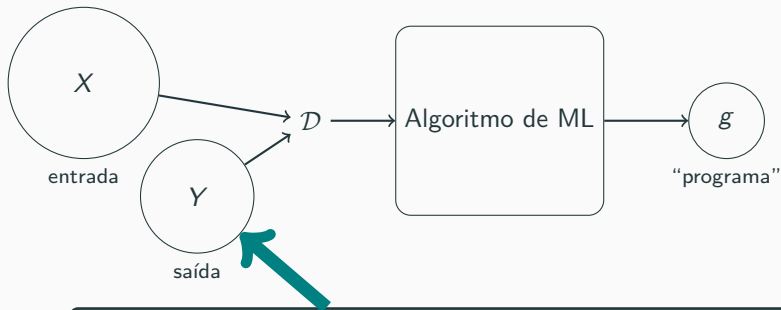
Como funciona a ML?

- Componentes do processo de ML
- Típica representação dos dados
- Treinamento / inferência
- Exemplo com árvore de decisão

Componentes do processo de ML

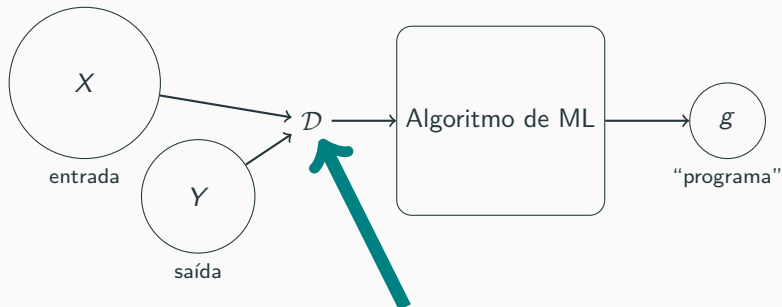


Componentes do processo de ML



Entrada-saída caracteriza o processamento desejado (**target**)

Componentes do processo de ML

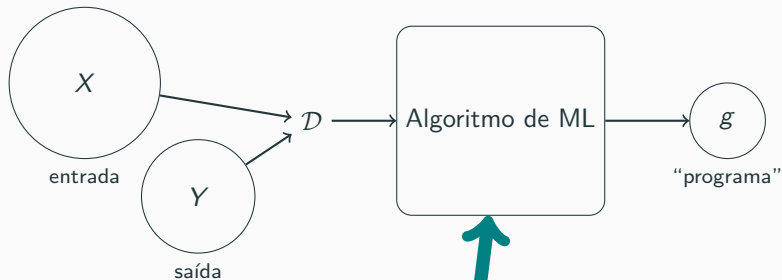


Dados de treinamento

$$\mathcal{D} = \left\{ (\mathbf{x}^{(i)}, y^{(i)}) \in X \times Y : i = 1, 2, \dots, N \right\}$$

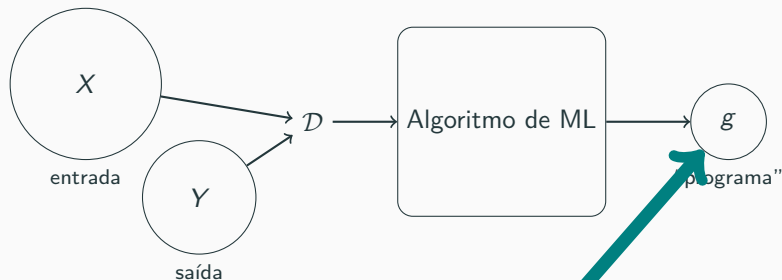
(exemplos de *relações input-output*)

Componentes do processo de ML



Algoritmo de aprendizado usa exemplos em \mathcal{D} para produzir um "programa" g

Componentes do processo de ML

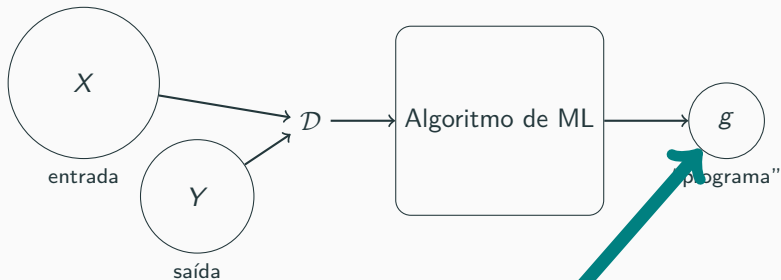


O “programa”, ou **hipótese** $g : X \rightarrow Y$, é o que queremos

Deve ser tal que $\hat{y} = g(\mathbf{x})$ seja

“o mais próximo possível” de y

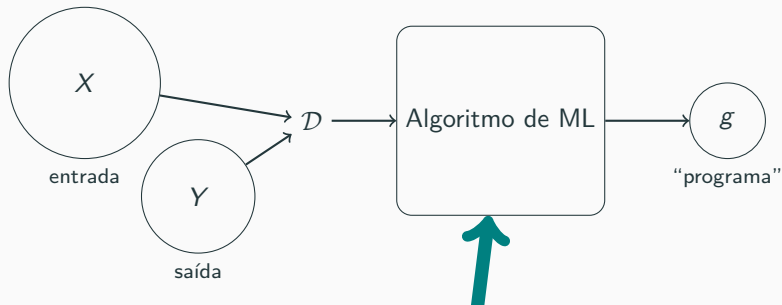
Componentes do processo de ML



Na prática, tenta-se encontrar $\hat{y} = g(\mathbf{x})$ que minimiza um **erro** empírico; ex.:

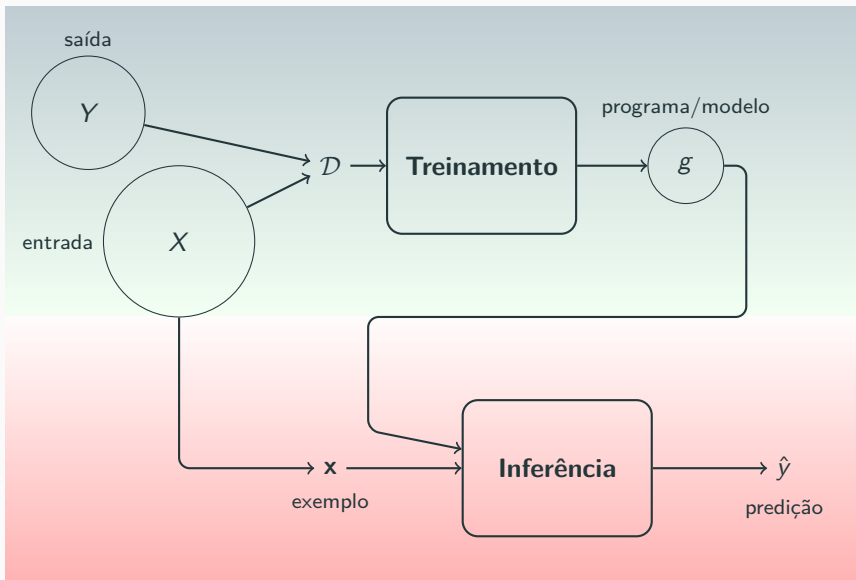
$$\hat{Err}(g) = \frac{1}{N} \sum_{i=1}^N \left(y^{(i)} - g(\mathbf{x}^{(i)}) \right)^2$$

Componentes do processo de ML

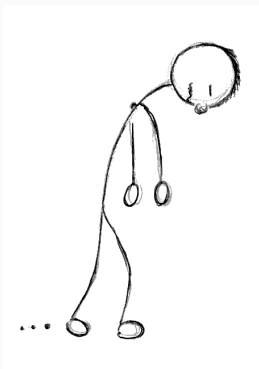


Treinamento: em geral os algoritmos de aprendizado usam alguma **técnica de otimização**

Processo de Machine Learning



O espaço X



- x_1 pressão arterial
- x_2 temperatura
- x_3 sente dor de cabeça
- x_4 peso
- x_5 altura
- x_6 idade
- x_7 sente fadiga
- x_8 tem tosse
- x_j ...
- x_d etc



$$\mathbf{x} = (x_1, x_2, \dots, x_d) \in \mathbb{R}^d$$



Processamento de imagens



Processamento de sinais



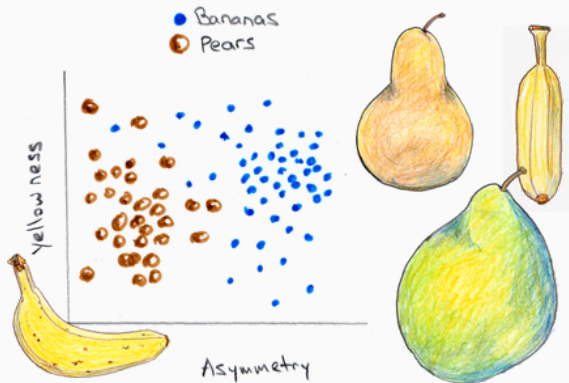
Extração de características diversas

- concentration
- asymmetry
- smoothness
- entropy
- spirality, etc



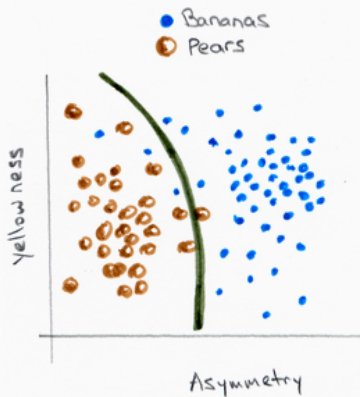
$$\mathbf{x} = (x_1, x_2, \dots, x_d)$$

Representação visual $X \times Y$ - classificação



<https://eightenthelephant.wordpress.com/2015/10/23/learning-about-machine-learning-part-i/>

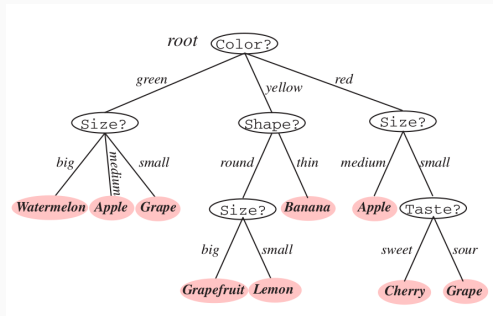
Representação visual das fronteiras de decisão - classificação



<https://eightenthelephant.wordpress.com/2015/10/23/learning-about-machine-learning-part-i/>

Exemplo de algoritmo (já treinado)

- Árvore de decisão para classificar frutas
- \mathbf{x} = (color, size, shape, taste)
- $y \in \{\text{watermelon, apple, grape, banana, grapefruit, lemon, cherry}\}$



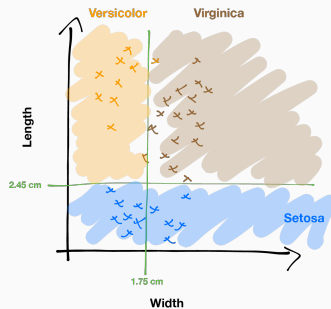
Exemplo de algoritmo (já treinado)

- Árvore de decisão para decidir sobre sair de casa
- x : informações sobre chuva e vento
- y : ação recomendada



Árvores de decisão

- Problema: classificação de flor de íris
- $\mathbf{x} = (\text{width}, \text{length})$
- $y \in \{\text{Setosa}, \text{Versicolor}, \text{Virginica}\}$
- Efeito: particionamento sucessivo do espaço X (um exemplo é um ponto no \mathbb{R}^2)



© Machine Learning @ Berkeley

- Aprendemos uma função $f : \mathbf{X} \rightarrow Y$
- Utilizando exemplos de treinamento

$$D = \{(\mathbf{x}^{(i)}, y^{(i)}) : i = 1, 2, \dots, N\}$$

- Em que cada exemplo pode ser descrito como $\mathbf{x} = (x_1, x_2, \dots, x_d) \in \mathbf{X} = \mathbb{R}^d$ (atributos)
- Se Y é contínuo, temos um problema de **regressão**
- Se Y é discreto, temos um problema de **classificação**:
 - **Binário** se Y possui 2 valores
 - **Multi-classe** se Y possui vários valores
 - **Multi-label** se Y é multidimensional (saída é uma tupla)

Próxima Aula: Regressão Linear

