Astronomia ao meio-dia

Machine learning e alguns exemplos de aplicação em dados astronômicos

Nina S. T. Hirata

Departamento de Ciência da Computação
Instituto de Matemática e Estatística
Universidade of São Paulo (USP)

São Paulo, 03/05/2018

Máquinas inteligentes?

Existem máquinas inteligentes ?

O que as máquinas inteligentes são capazes de fazer ?

Máquinas aprendem ? (elas podem fazer mais ?)

Máquinas inteligentes?

Existem máquinas inteligentes ?

O que as máquinas inteligentes são capazes de fazer ?

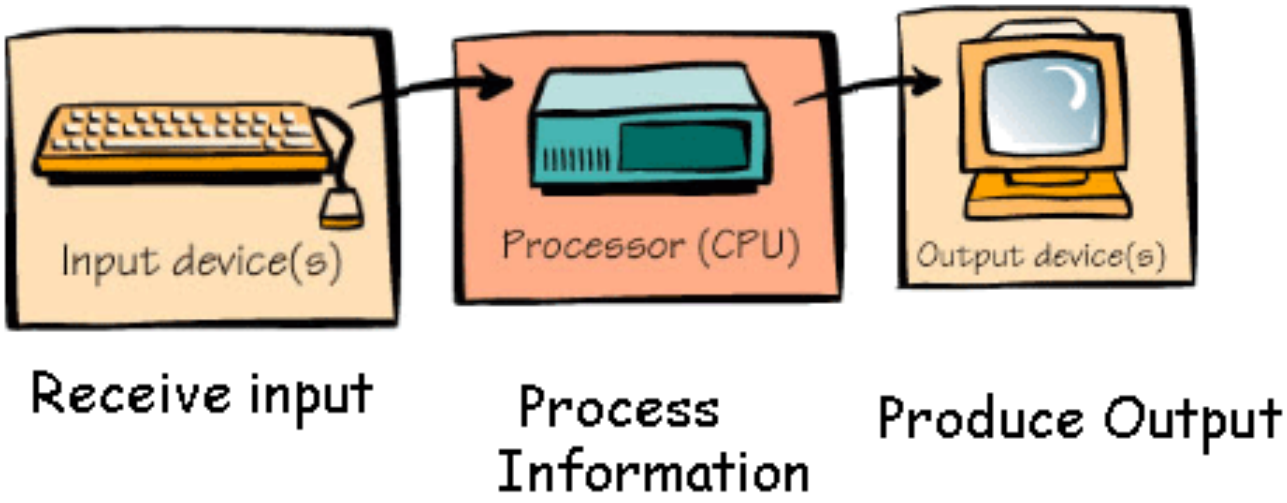
Máquinas aprendem ? (elas podem fazer mais ?)

Não vamos discutir este tema do ponto de vista filosófico / ético

Machine learning (ML)

Vamos discutir ML do ponto de vista computacional

O que os computadores fazem ?



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

Computadores processam dados segundo um **algoritmo**

Algoritmos são soluções para **problemas computacionais**

O que são **problemas computacionais** ?

Exemplo: somar números

- Dada uma lista de números, calcular a soma dos números na lista

Entrada	$\Longrightarrow$	Saída
`3, 1, 7`	$\Longrightarrow$	`11`
`0.25, 0.75, 0.5, 0.1`	$\Longrightarrow$	`1.6`
`1, 3, 5, 7, 9`	$\Longrightarrow$	`25`

Exemplo: somar números

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
   (remova o número da lista)
   SOMA = SOMA + NUM
Imprima SOMA
```

Exemplo: o que está sendo computado neste caso ?

Entrada	$\Longrightarrow$	Saída
`9, 2, 0, -1, 7, 4`	$\Longrightarrow$	`-1, 0, 2, 4, 7, 9`
`'x', 'a', 'm', 'b'`	$\Longrightarrow$	`'a', 'b', 'm', 'x'`

Exemplo: o que está sendo computado neste caso ?

Entrada	$\Longrightarrow$	Saída
`9, 2, 0, -1, 7, 4`	$\Longrightarrow$	`-1, 0, 2, 4, 7, 9`
`'x', 'a', 'm', 'b'`	$\Longrightarrow$	`'a', 'b', 'm', 'x'`

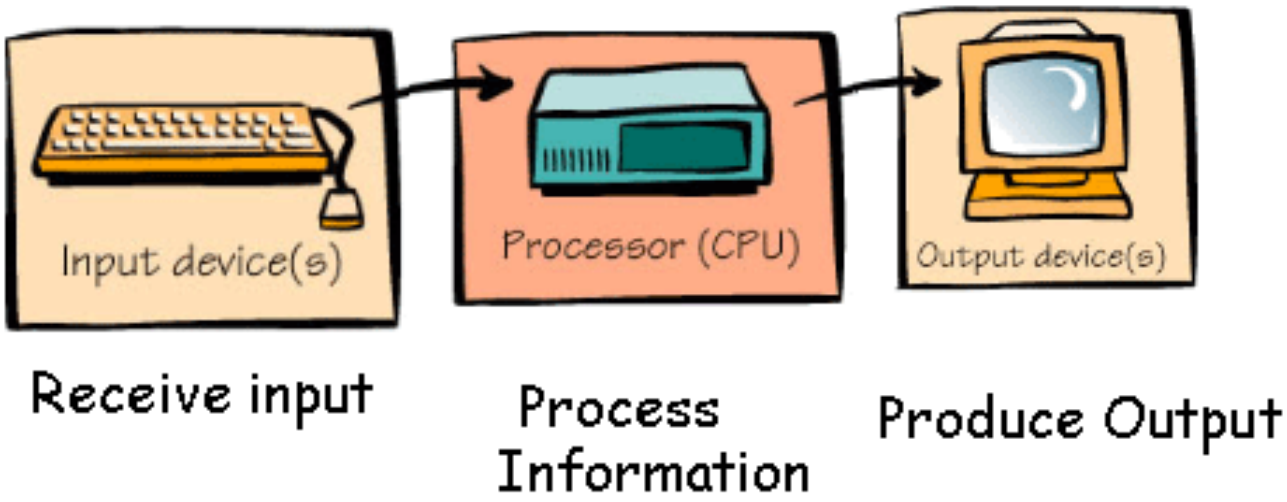
Problema de **ordenação**!

Nós sabemos como resolver (computar) esse problema!

Vários algoritmos de ordenação

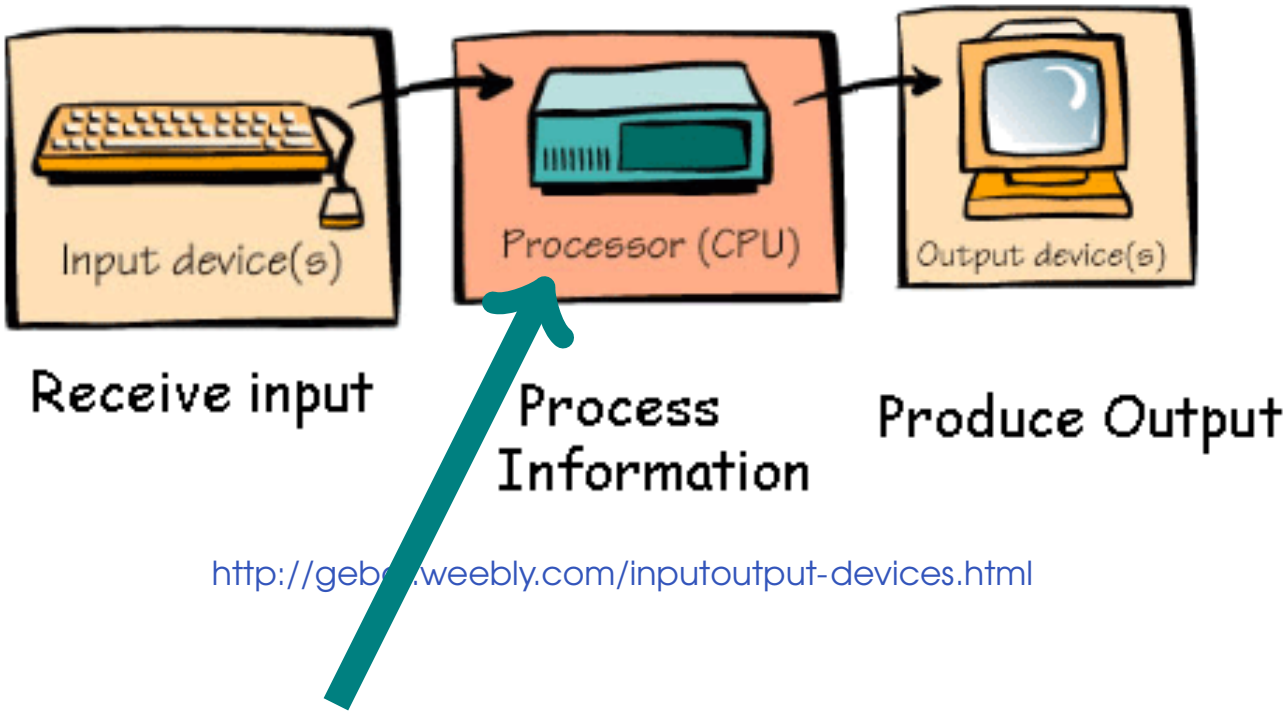
- selection sort
- insertion sort
- bubble sort
- merge sort
- quick sort
- etc

Como os dados são processados?



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

Como os dados são processados?



http://geb[illegible].weebly.com/inputoutput-devices.html

Programas (implementação de algoritmos)

Desafios computacionais

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

- O que fazer quando não há algoritmo eficiente ?
- O que fazer quando a relação entrada-saída não está bem definida?
- Ou quando não sabemos descrever formalmente a relação?

Quando não temos um algoritmo


⟹ `'a'`


⟹ `'X'`


⟹ `'A'`

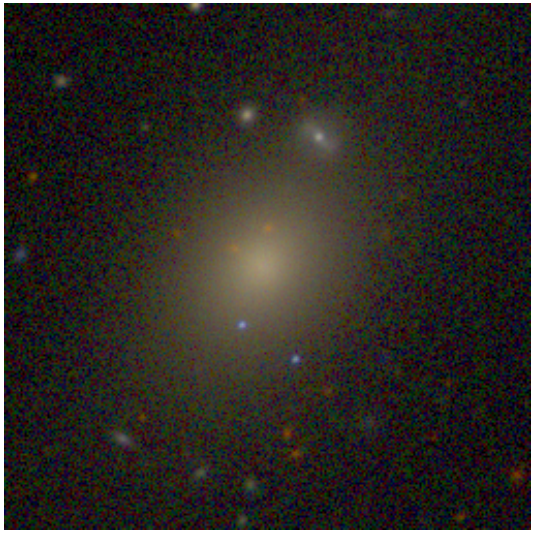
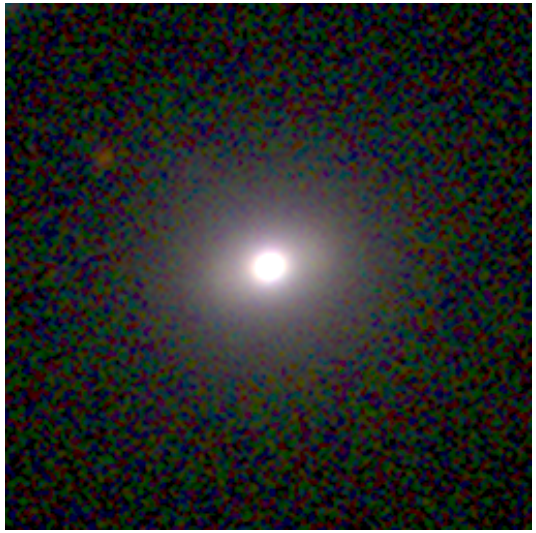
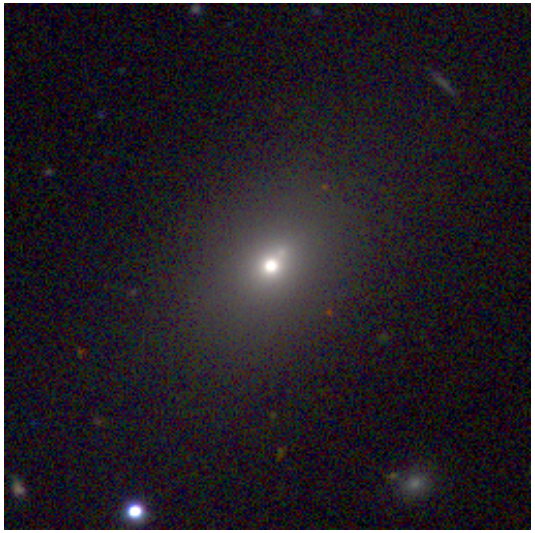

⟹ `'d'`


⟹ `'b'`

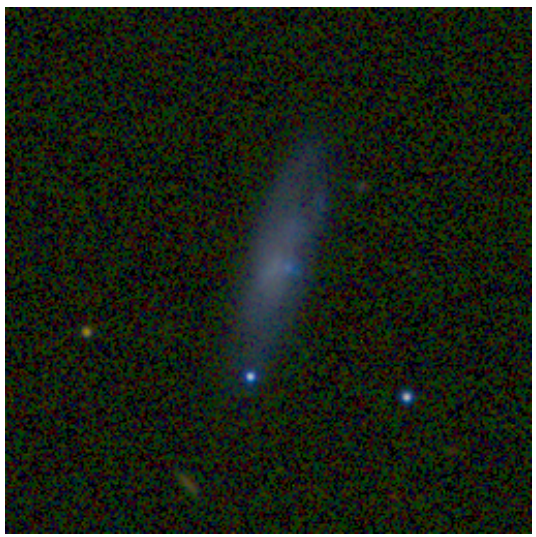
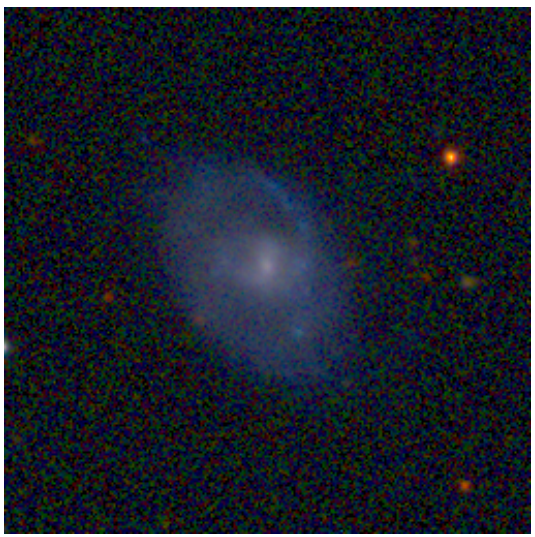

⟹ `'M'`

Quando não temos um algoritmo

Elíptica

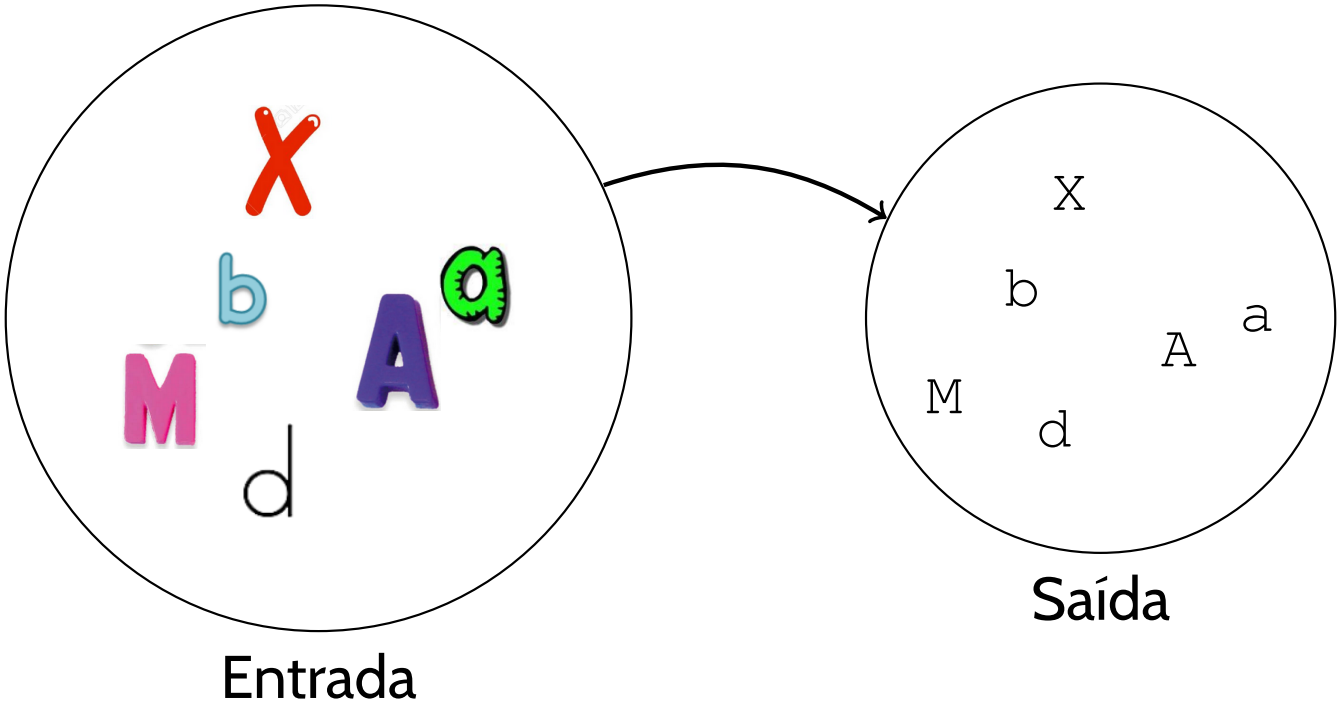




Espiral

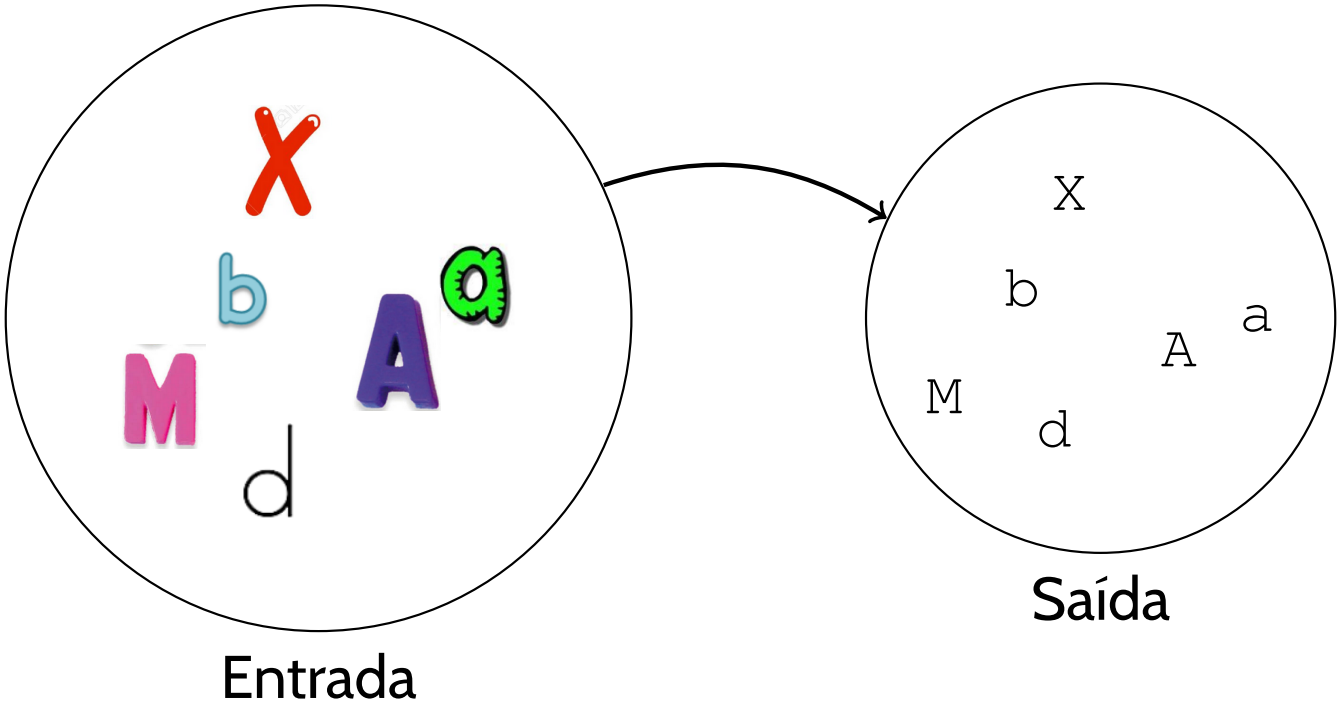
Imagens do EFIGI

Entra em cena ML



Há uma **relação** entre elementos no espaço de entrada e elementos no espaço de saída $\Longrightarrow$ que eu não sei descrever formalmente

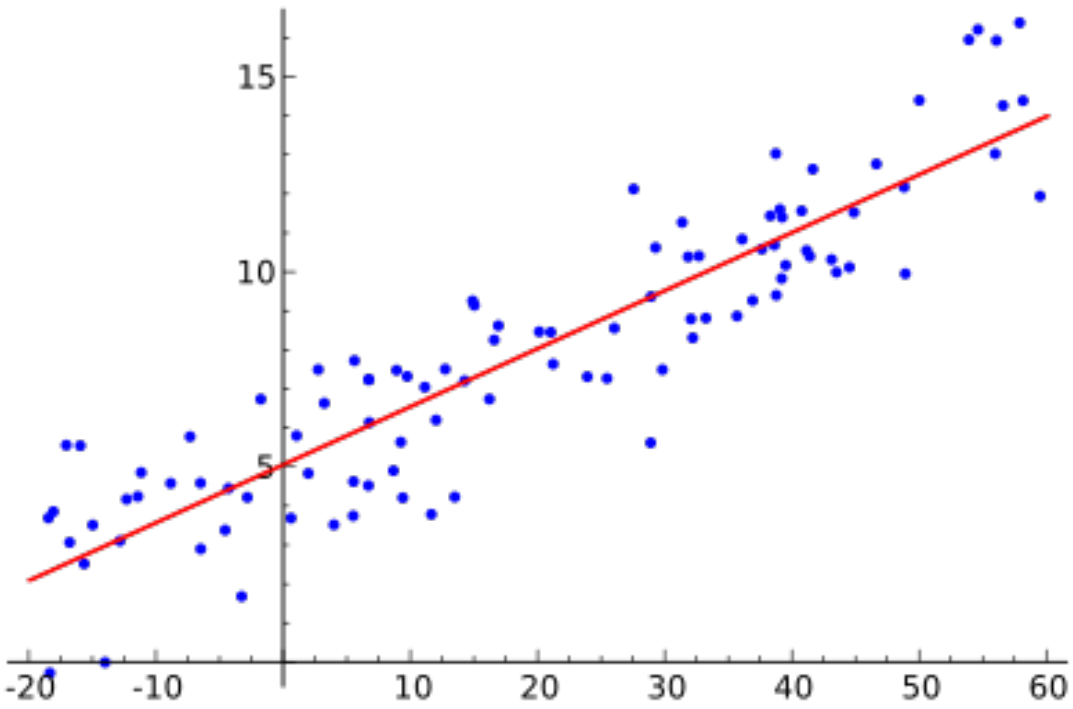
Entra em cena ML



Idéia central de ML: "aprender" a relação a partir de exemplos

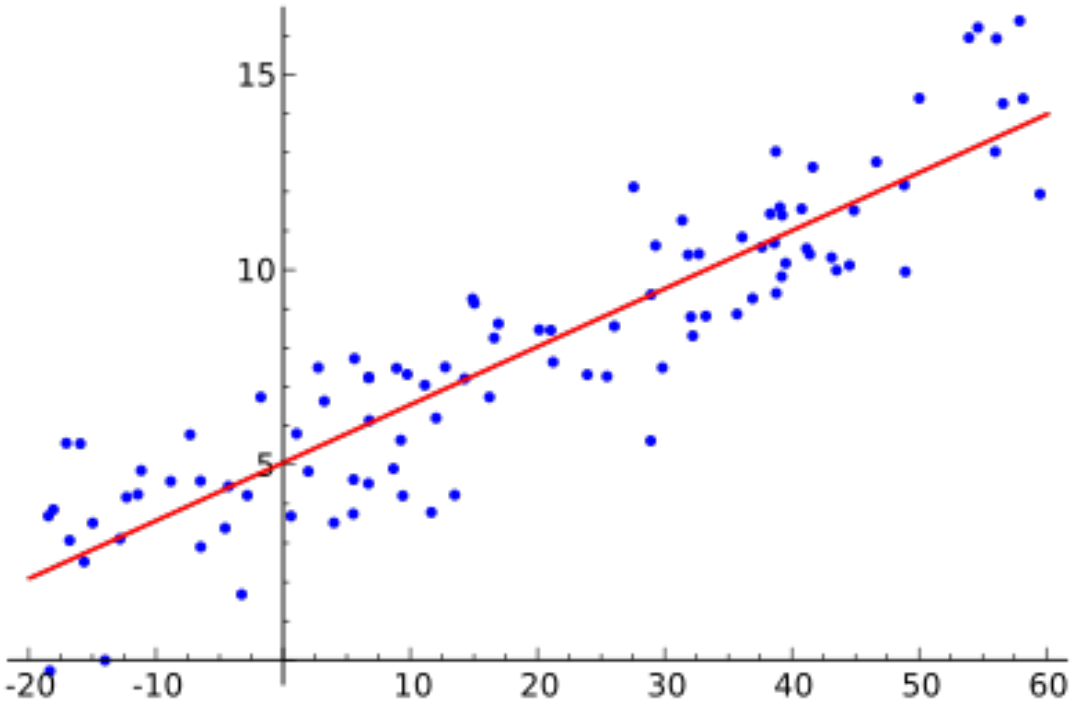
Produto final de ML: um programa que realiza o processamento entrada-saída

Um exemplo simples: regressão linear



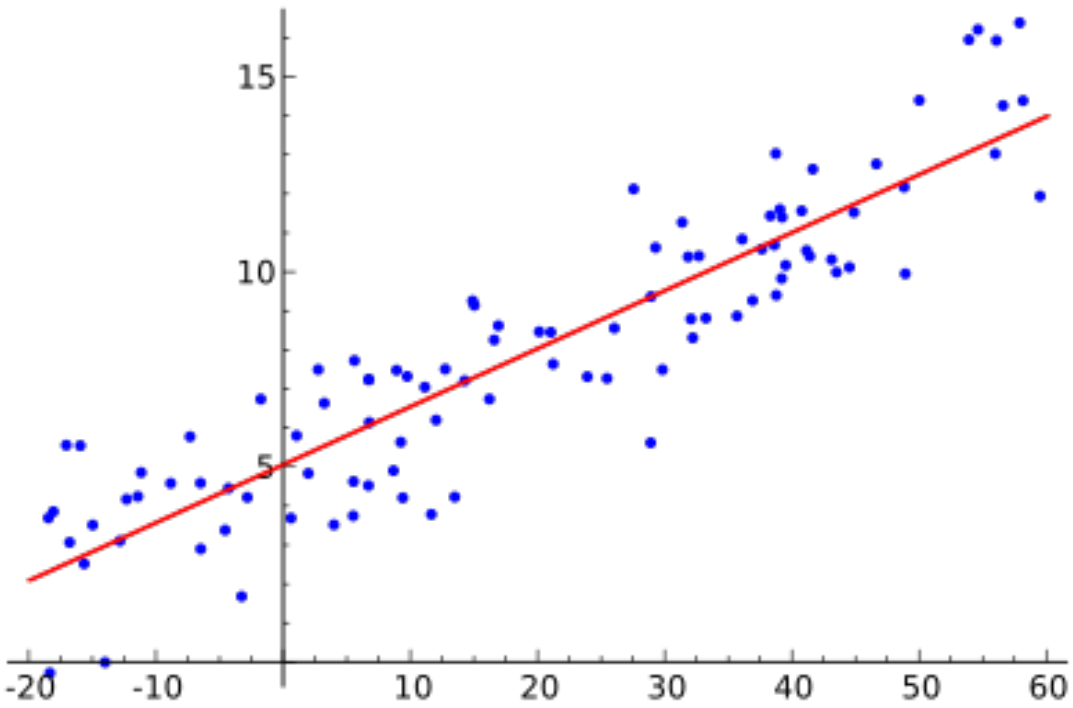
Os pontos azuis (x_i, y_i) são os exemplos de treinamento

Um exemplo simples: regressão linear



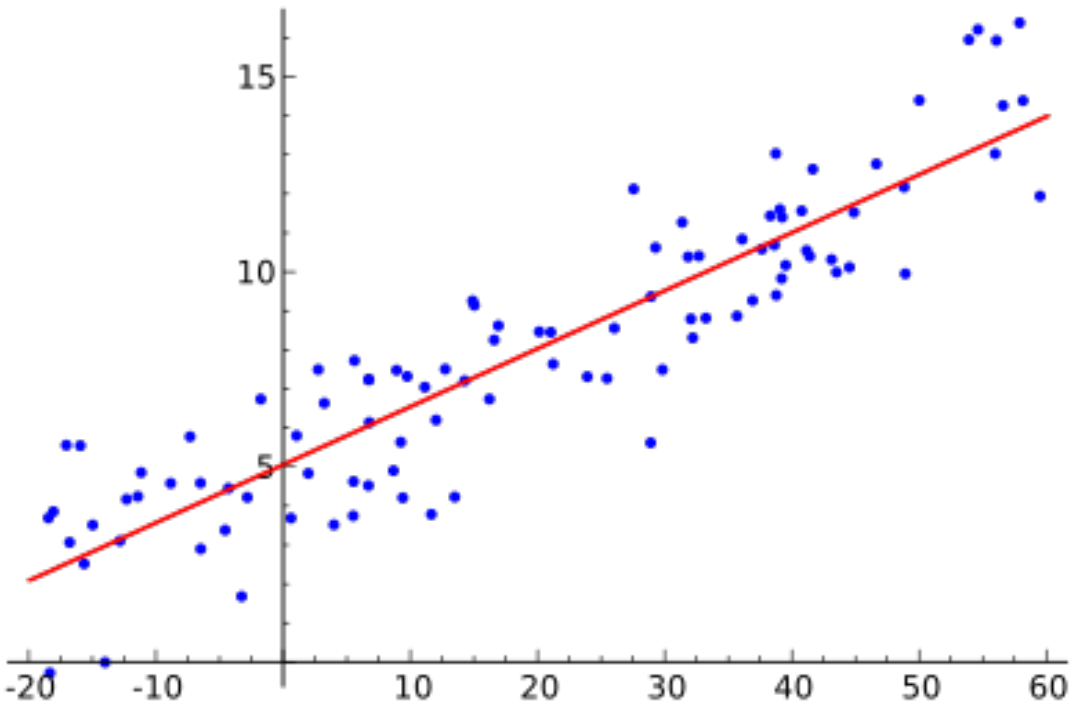
Há uma relação linear entre x e y

Um exemplo simples: regressão linear



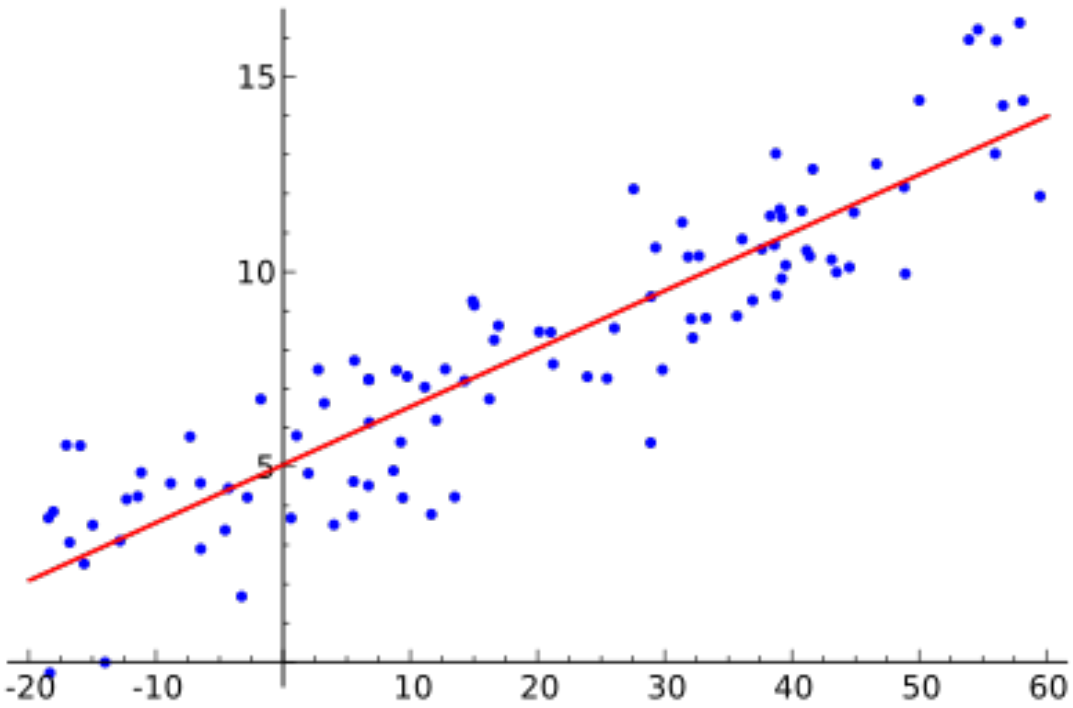
Família de hipóteses adequada: $g(x) = a + b\,x$

Um exemplo simples: regressão linear



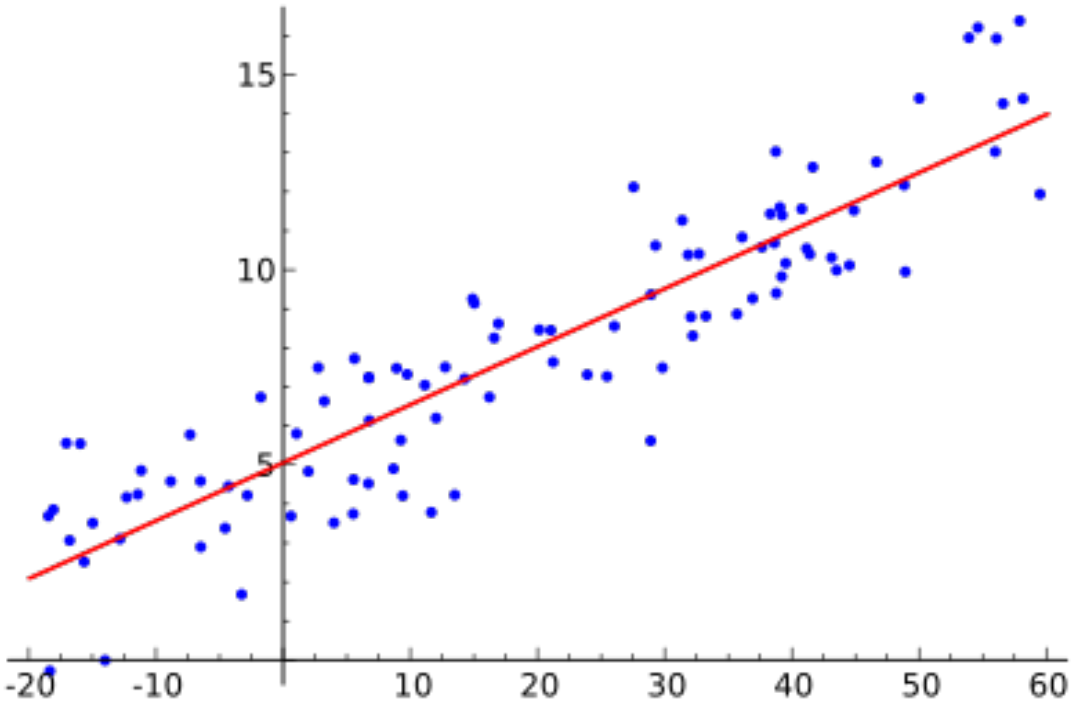
Função custo : $J(a,b) = \frac{1}{N}\sum_{i=1}^{N}\left(y_i - g(x_i)\right)^2$

Um exemplo simples: regressão linear



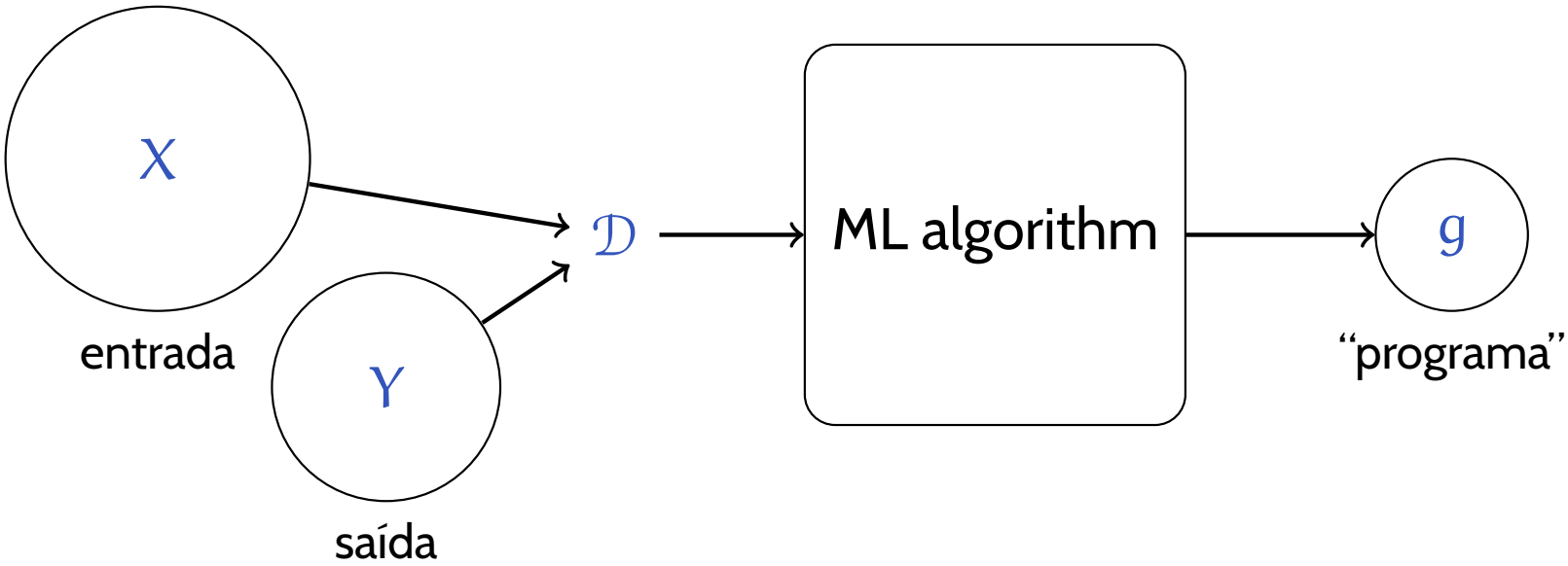
Otimização : encontrar a e b ($g(x) = a + b\,x$) que minimiza o custo $J(a, b)$

Um exemplo simples: regressão linear

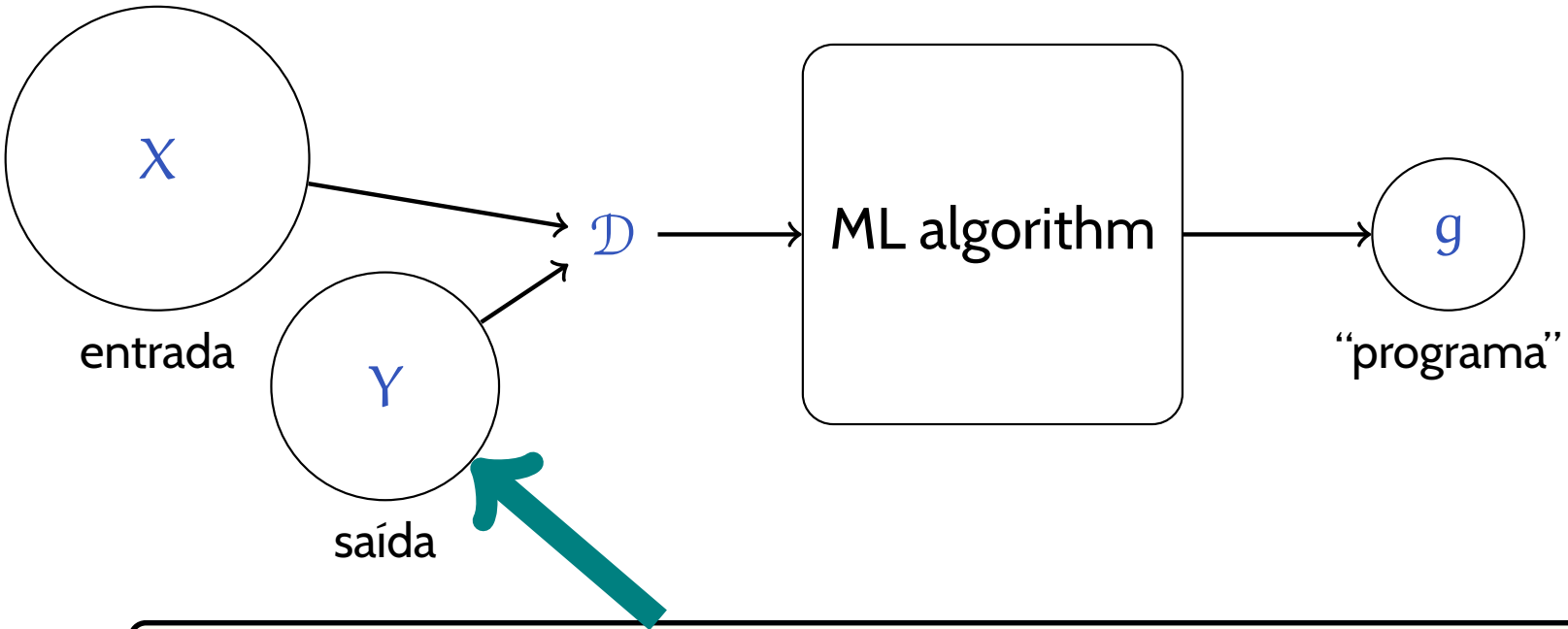


Aprendizado : há uma solução analítica para a regressão linear mas pode-se também usar uma abordagem de ajuste de parâmetros iterativa (ex.: *gradient descent*)

Componentes de um processo de ML

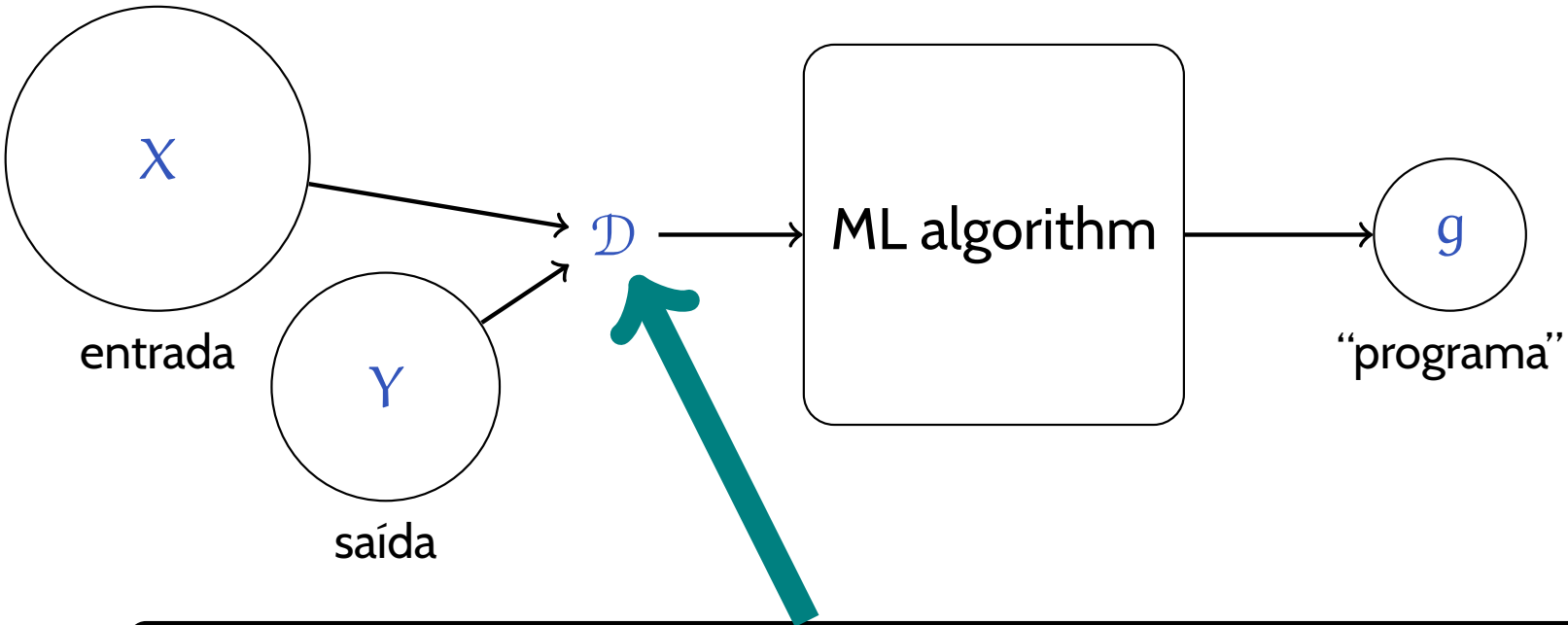


Componentes de um processo de ML



Entrada-saída caracteriza o processamento desejado

Componentes de um processo de ML

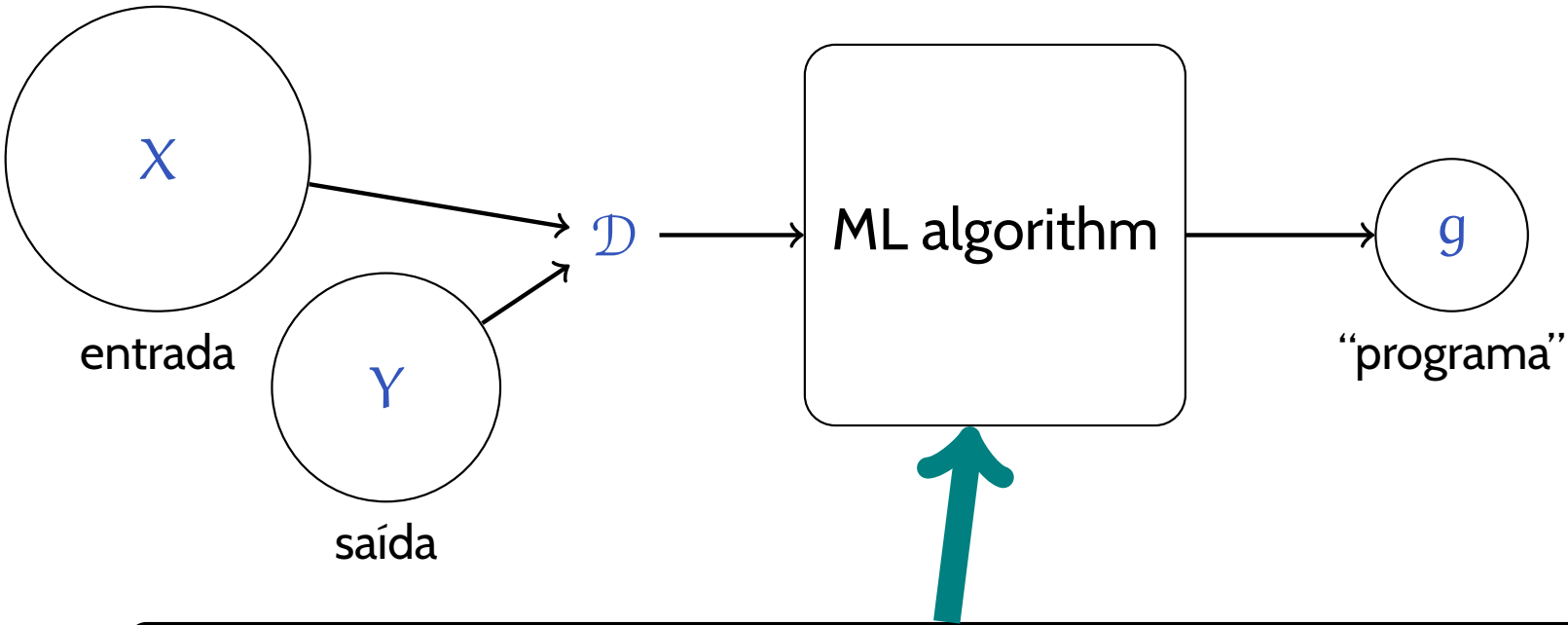


Dados de treinamento

$$\mathcal{D} = \left\{ (\mathbf{x}_i, y_i) \in X \times Y : i = 1, 2, \ldots, N \right\}$$

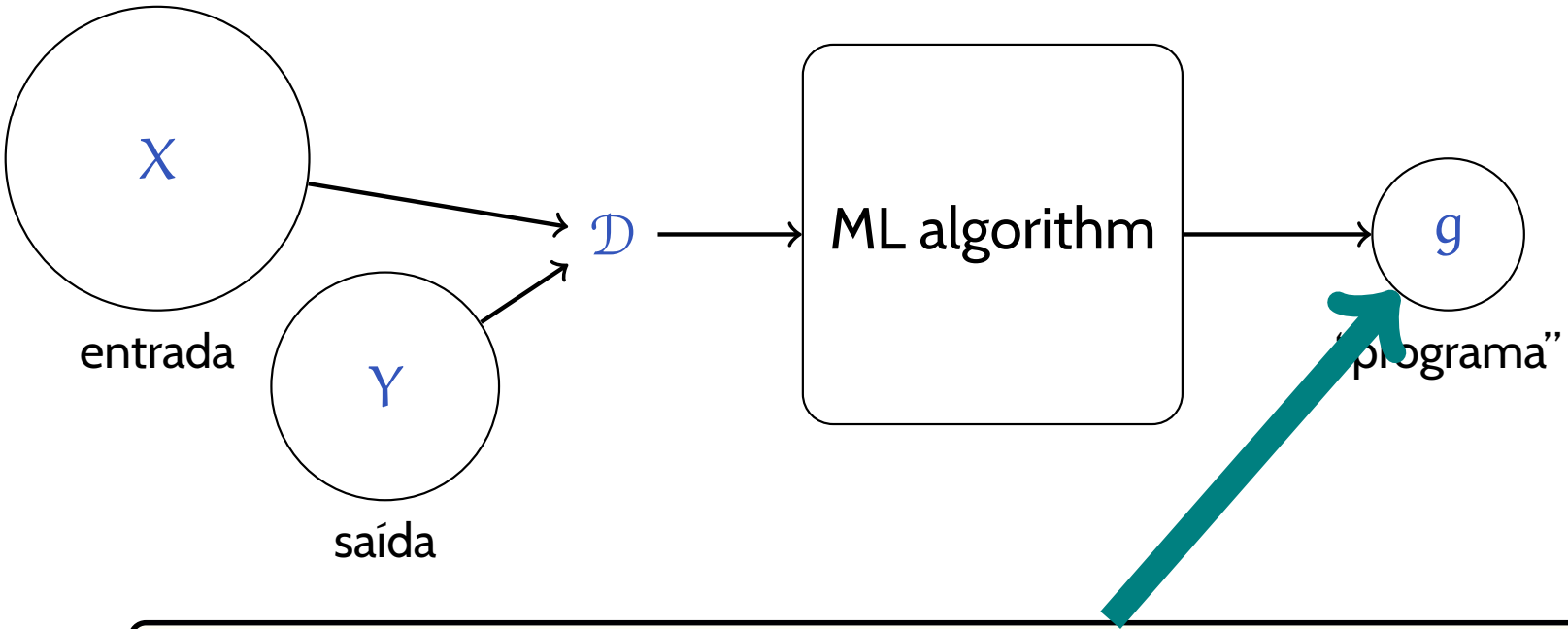
(exemplos de **relações *input-output***)

Componentes de um processo de ML



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

Componentes de um processo de ML

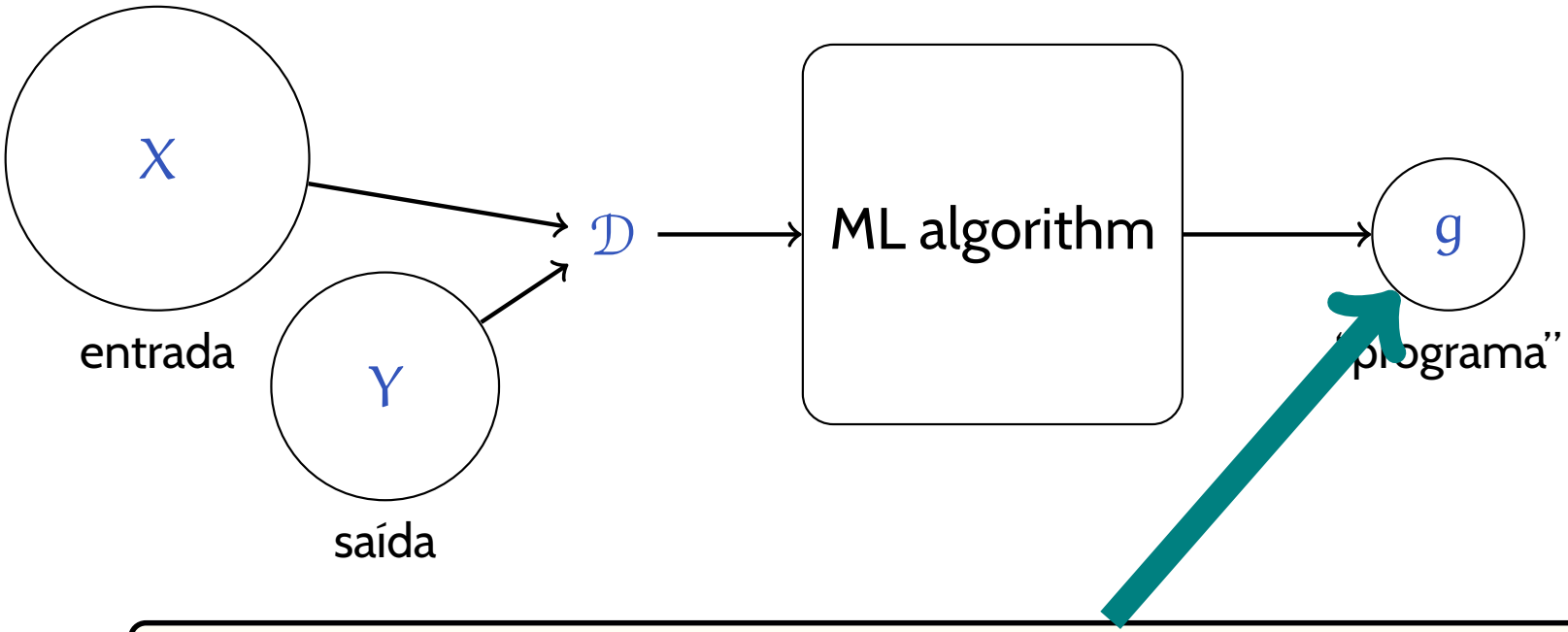


O "programa", ou hipótese $g : X \to Y$, é o que queremos

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

"o mais próximo possível" de y

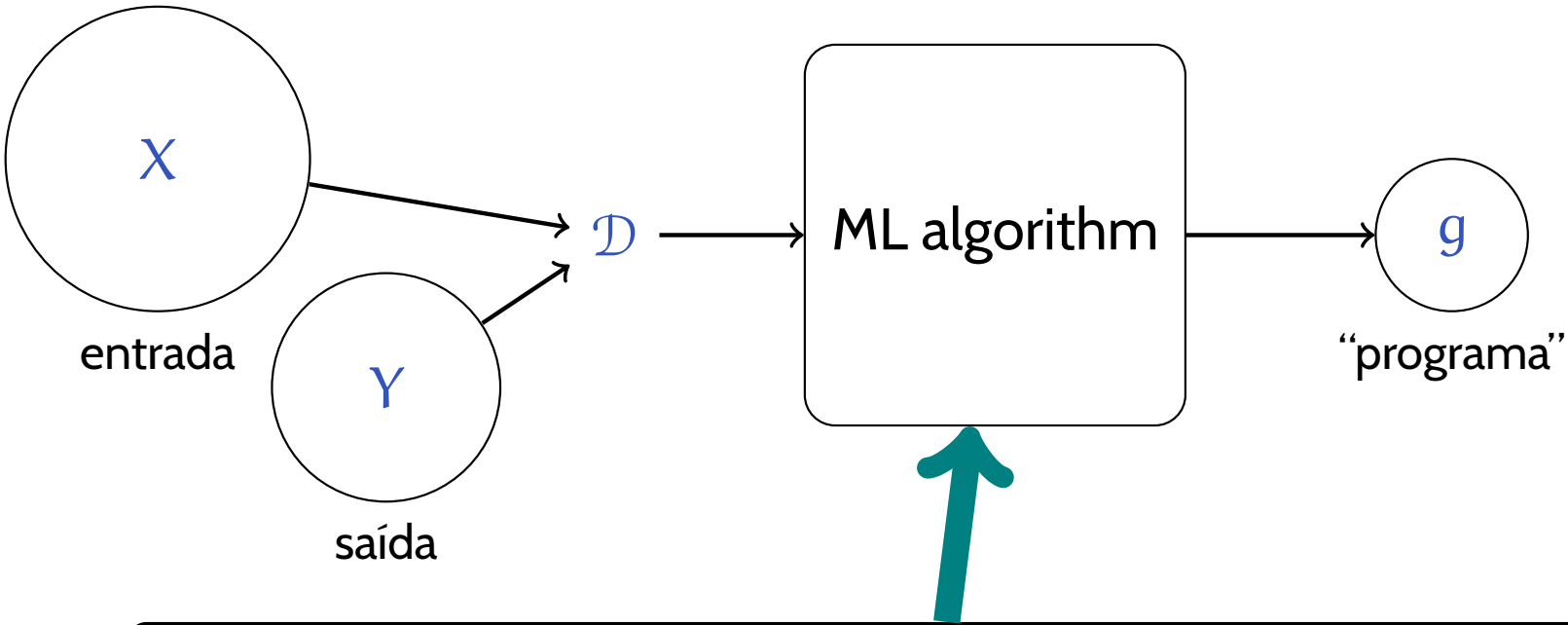
Componentes de um processo de ML



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

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

Componentes de um processo de ML



Portanto, em geral os algoritmos de aprendizado usam alguma **técnica de otimização**

Exemplo

Classificação de imagens de galáxia

O que precisamos considerar

Entrada / saída

X : todas as possíveis imagens de galáxias (espiral / elíptica)

Y : classe (espiral / elíptica)

Dados de treinamento

Quantos pares $(\mathbf{x}, y)$ precisamos? De onde eles vem ?

Treinamento

Custo a ser minimizado : erro de classificação

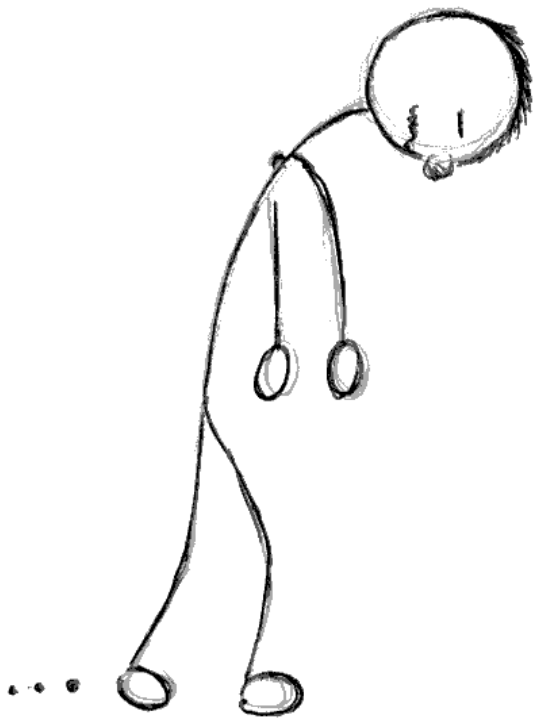
Algoritmo de aprendizado : SVM, redes neurais, random forests, etc

Representação dos dados

Como esses algoritmos irão "enxergar" as imagens ? $\Longrightarrow$ como representá-las?

Representação dos dados de entrada

Exemplo: No caso de diagnóstico médico, informações sobre o paciente



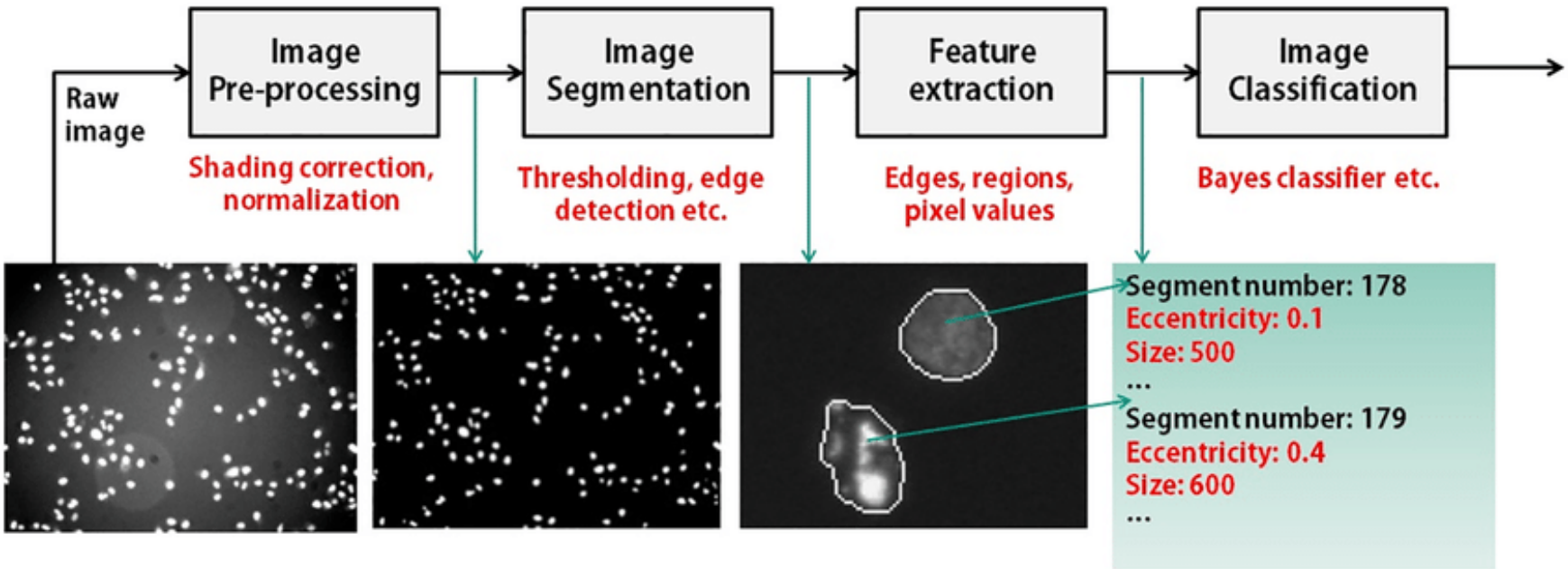
$\Longrightarrow$

- 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_n etc

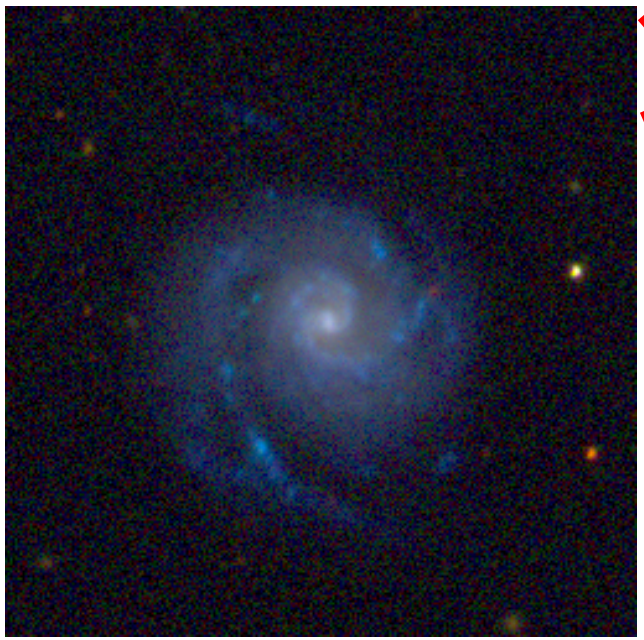
$\Downarrow$

$$\mathbf{x} = (x_1, x_2, \ldots, x_n) \in \mathbb{R}^n$$

Quando a entrada é uma imagem



Classificação de imagens de galáxias



Processamento de imagens

Processamento de sinais

Extração de características diversas

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

$$\mathbf{x} = (x_1, x_2, \ldots, x_n)$$

Classificação de imagens de galáxias

(com Fabrício Ferrari e Mateus Espadoto)

Dataset usado: EFIGI (https://www.astromatic.net/projects/efigi)

Procedimento executado:

1. Selecionar apenas as espirais e as elípticas
2. Extrair *features* para cada imagem
   - Usando Morfometryka (dados fornecidos por Fabrício Ferrari, FURG)
   - Usando uma rede convolucional pré-treinada com o ImageNet
3. Separar dados em treinamento e validação
4. Treinar SVM usando dados de treinamento
5. Avaliar desempenho (de predição) no conjunto de validação


3245: eli
439: spi
1038: spi
1838: spi
360: spi
Classificação errada
4 vizinhos mais próximos
2856: eli
1449: spi
1431: spi
1524: spi
1587: spi
Features convolucionais
3308: spi
3248: eli
395: eli
3190: eli
3276: eli
1855: eli
1049: spi
1616: spi
972: spi
2754: spi
2892: eli
2893: spi
2935: spi
472: spi
2623: spi


Classificação errada
4 vizinhos mais próximos
Features convolucionais
132: eli
3237: spi
816: spi
1906: spi
1515: spi
1935: spi
3235: eli
2135: eli
2458: eli
190: eli
3008: spi
3122: eli
2238: eli
63: eli
1508: eli
1035: spi
2482: eli
2181: eli
2040: eli
3220: eli

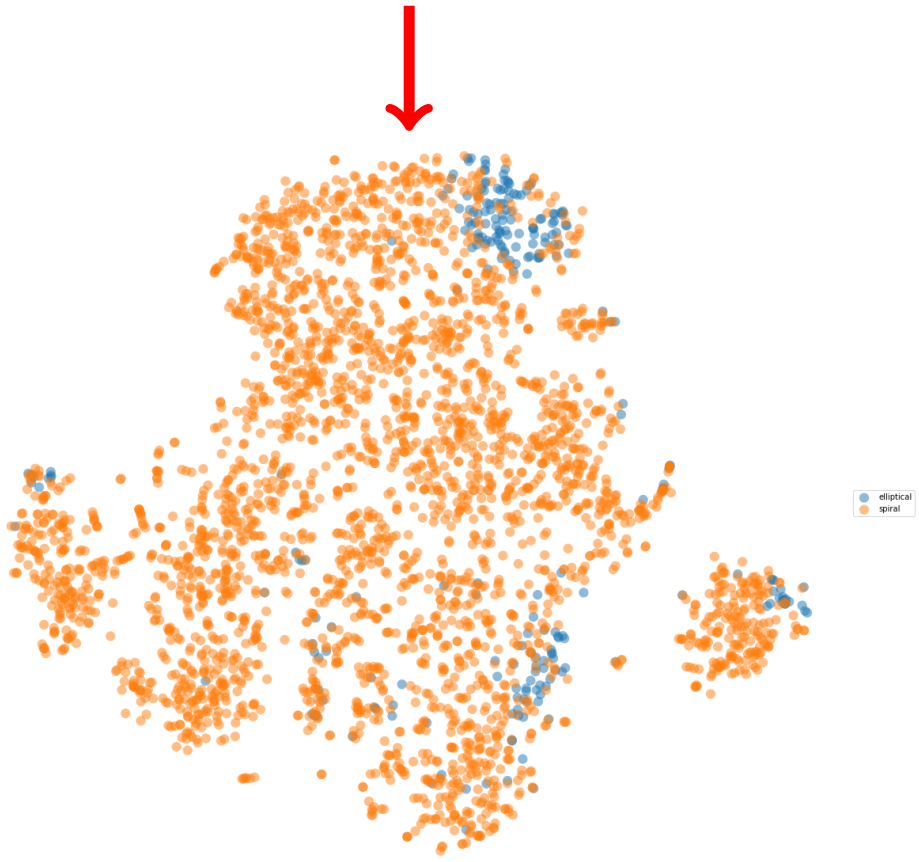

1023: eli
1087: spi
982: spi
1222: spi
974: spi
Classificação errada
4 vizinhos mais próximos
1773: eli
2084: spi
680: spi
2594: spi
3177: spi
Features morfométricos
987: eli
1040: spi
1229: spi
998: spi
1078: spi
1282: spi
975: eli
1007: eli
1083: eli
1116: eli
1876: eli
1615: spi
541: spi
2434: spi
2764: spi


Classificação errada
4 vizinhos mais próximos
Features morfométricos
1830: eli
139: spi
3192: spi
2927: spi
1989: spi
42: eli
1320: spi
1600: spi
2199: spi
3231: spi
2541: eli
2037: spi
1449: spi
73: spi
615: spi
1854: eli
94: spi
164: spi
57: spi
2944: spi

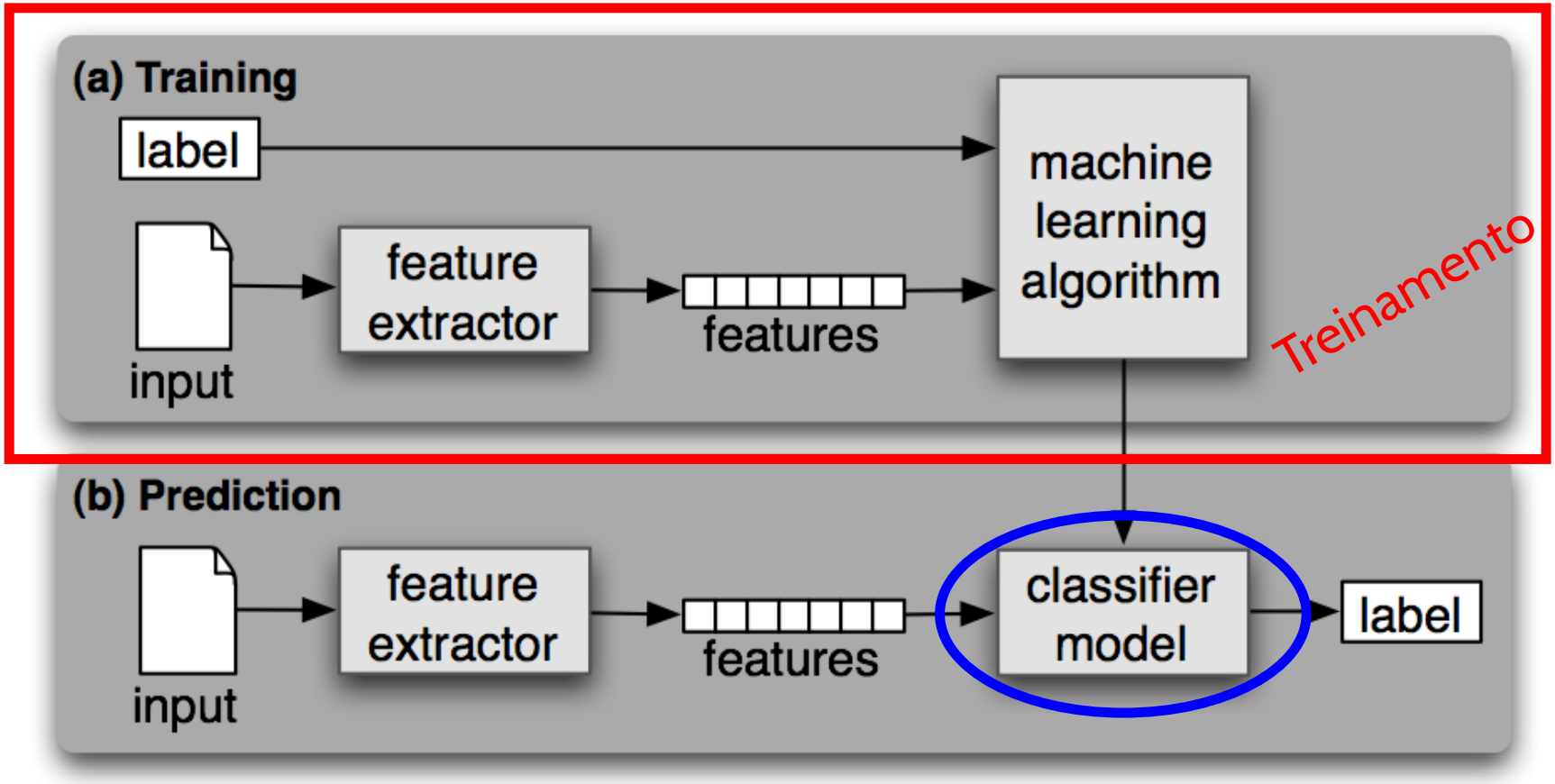
Projeções das *features*

Features convolucionais

Features morfométricos

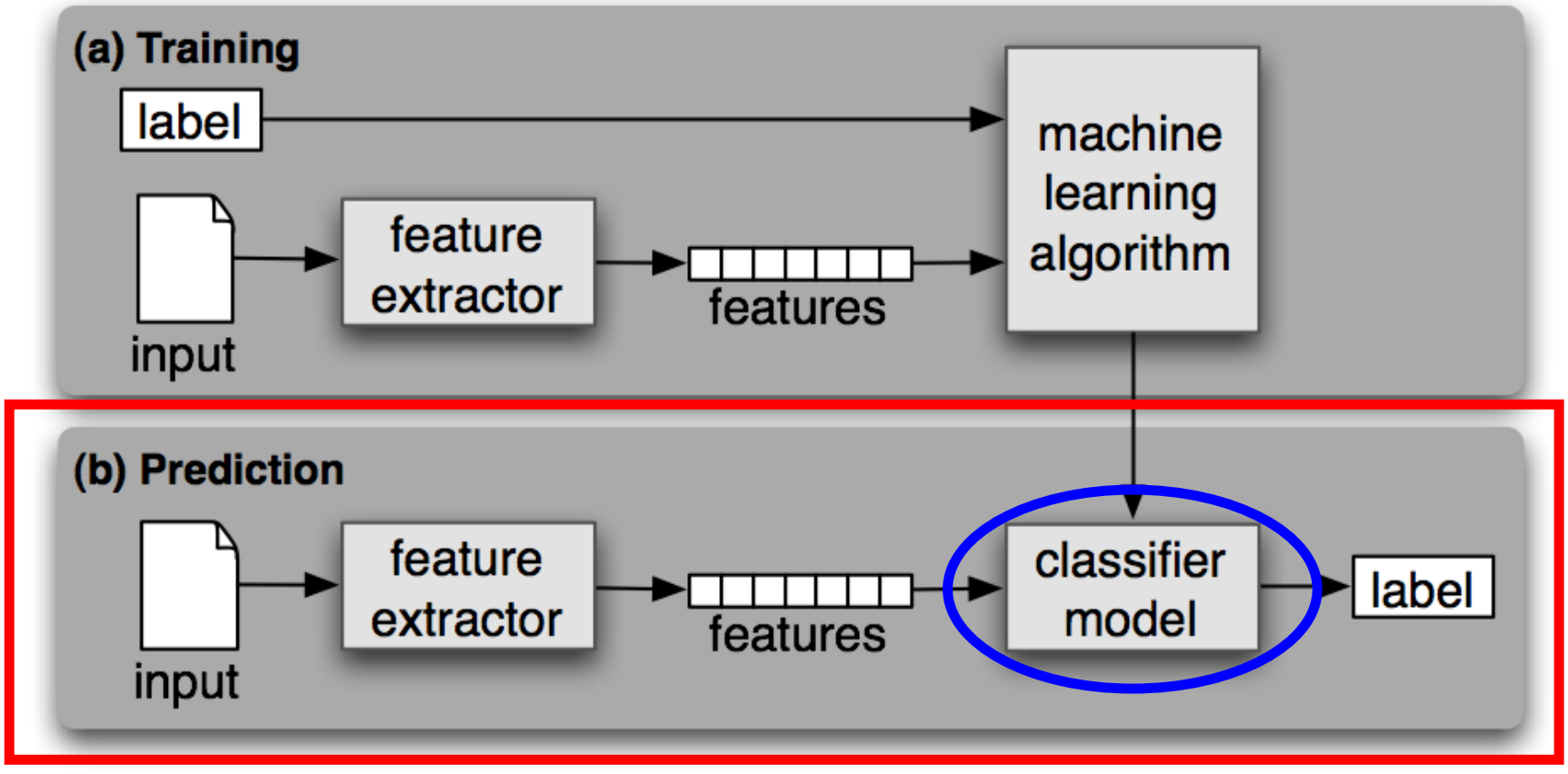


Processo de machine learning



http://www.nltk.org/book/ch06.html

Processo de machine learning



http://www.nltk.org/book/ch06.html

Predição

Alguns comentários

- ML é uma forma de **meta-programação**

  Programas que criam programas (hipóteses g) especializados em realizar a transformação *input-output*

- **Generalização**

  O programa criado deve funcionar para exemplos previamente não vistos

  A teoria de ML apresenta alguns resultados interessantes sobre isso

- **Complexidade da família de hipóteses $\times$ quantidade de dados**

  Quanto mais complexos forem as hipóteses, mais dados de treinamento são necessários

Tipos de aprendizado

Supervisionado: Rótulos numéricos, y, representando:

- um valor numérico ($y \in \mathbb{R}$)
  - problema de regressão
  - estimar *redshift*
- um valor categórico ($y \in \{1, 2, \ldots, c\}$)
  - problema de classificação
  - morfologia da galáxia ($y \in \{1, 2, \ldots, c\}$)

Não-supervisionado: Y "desconhecido"

- clustering, etc

Exemplo

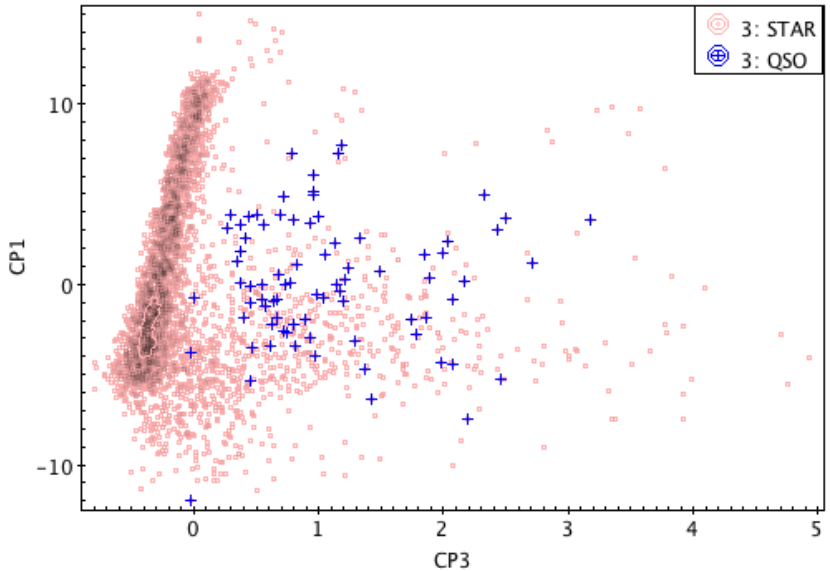
Deteção de quasares em catálogos

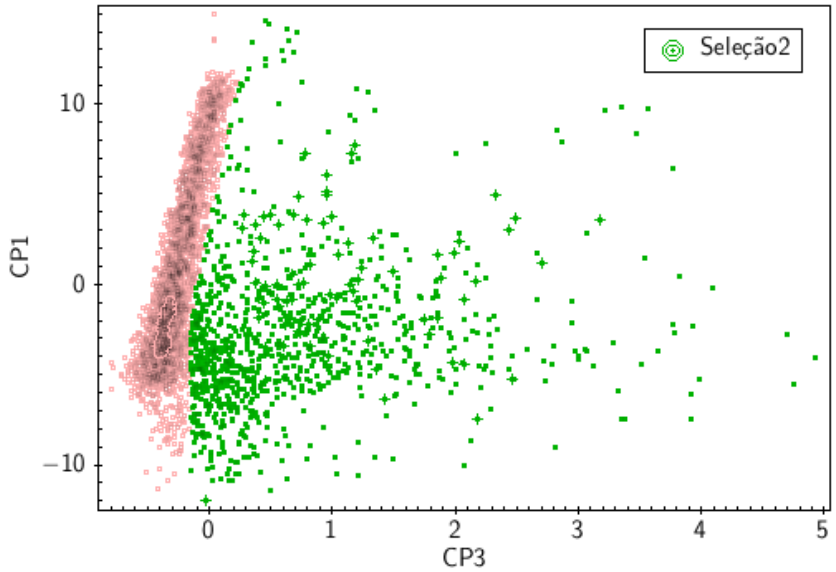
Deteção de quasares

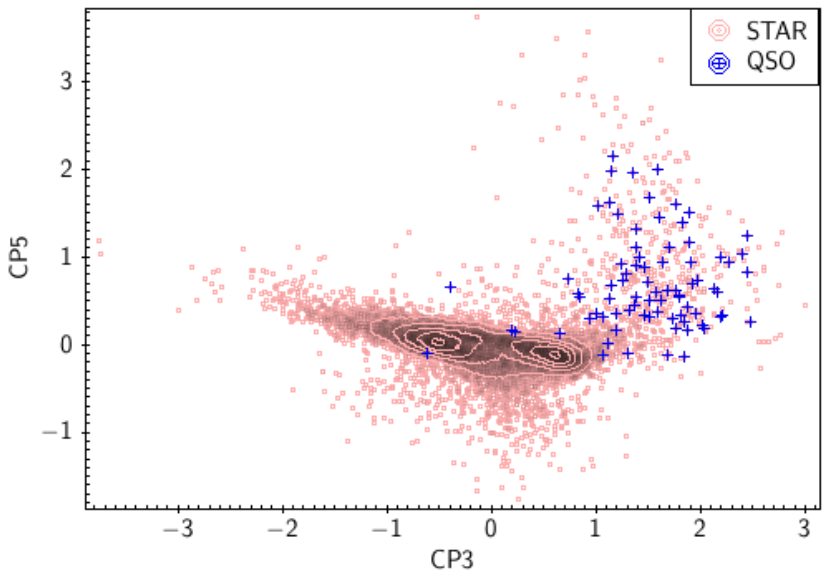
(com Cláudia Oliveira, João Steiner, Liliane Nakazono)

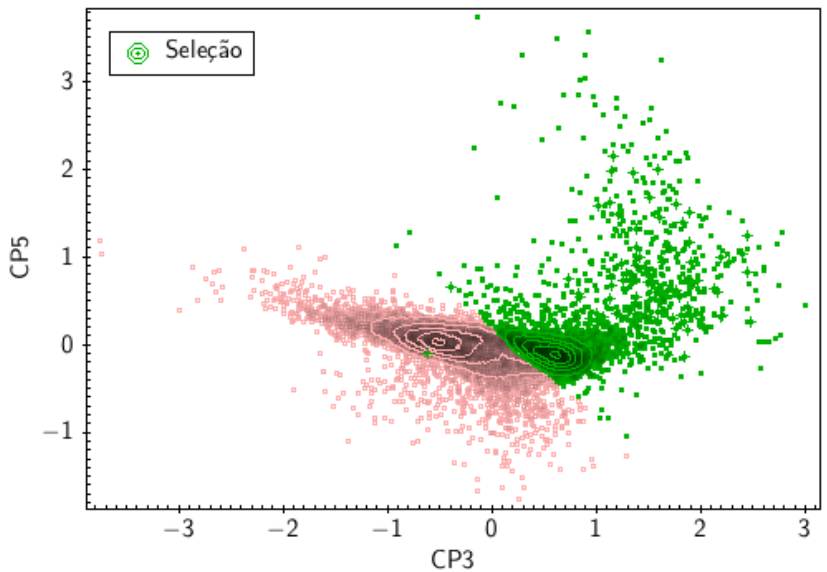
- Dados: todos os objetos correspondentes a fontes pontuais no catálogo (ALHAMBRA)
- Como detectar fonte pontual que é QSO (e não estrela) ?
- Selecionar do catálogo os QSOs conhecidos (buscar QSO em listas de objetos identificados e fazer *match* de coordenada com os objetos do catálogo em estudo)
- Representação adequada para cada objeto do catálogo: *features* podem ser calculadas a partir da magnitude dos filtros
- Aplicar PCA (análise de componentes principais)
- Projeção 2D dos objetos – pares de componentes do PCA
- Selecionar projeção na qual QSOs (inicialmente selecionados) ficam concentrados em uma região compacta, mesmo que misturados às supostas estrelas
- Para cada objeto na região, calcular os k-NN em um conjunto conhecido de QSOs (possivelmente externos ao catálogo) – *redshift* homogêneo entre k-vizinhos confere maior confiança de que de fato aquele é um QSO, e de quebra fornece uma estimativa do seu *redshift*

Deteção de quasares









Deteção de quasares

(com Cláudia Oliveira, João Steiner, Liliane Nakazono)

- Dados: todos os objetos correspondentes a fontes pontuais no catálogo (ALHAMBRA)
- Como detectar fonte pontual que é QSO (e não estrela) ?
- Selecionar do catálogo os QSOs conhecidos (buscar QSO em listas de objetos identificados e fazer *match* de coordenada com os objetos do catálogo em estudo)
- Representação adequada para cada objeto do catálogo: *features* podem ser calculadas a partir da magnitude dos filtros
- Aplicar PCA (análise de componentes principais)
- Projeção 2D dos objetos – pares de componentes do PCA
- Selecionar projeção na qual QSOs (inicialmente selecionados) ficam concentrados em uma região compacta, mesmo que misturados às supostas estrelas
- Para cada objeto na região, calcular os k-NN em um conjunto conhecido de QSOs (possivelmente externos ao catálogo) – *redshift* homogêneo entre k-vizinhos confere maior confiança de que de fato aquele é um QSO, e de quebra fornece uma estimativa do seu *redshift*

Machine learning ao longo dos anos × dados

zero dados

regras

```
if age > 40:
    if is_home_owner:
        print("give a credit")
    else:
        if income > 5000:
            print("give a credit")
        else:
            print("to refuse")
else:
    if education == "university":
        print("...")
    else:
        print("...")
```

Machine learning ao longo dos anos × dados

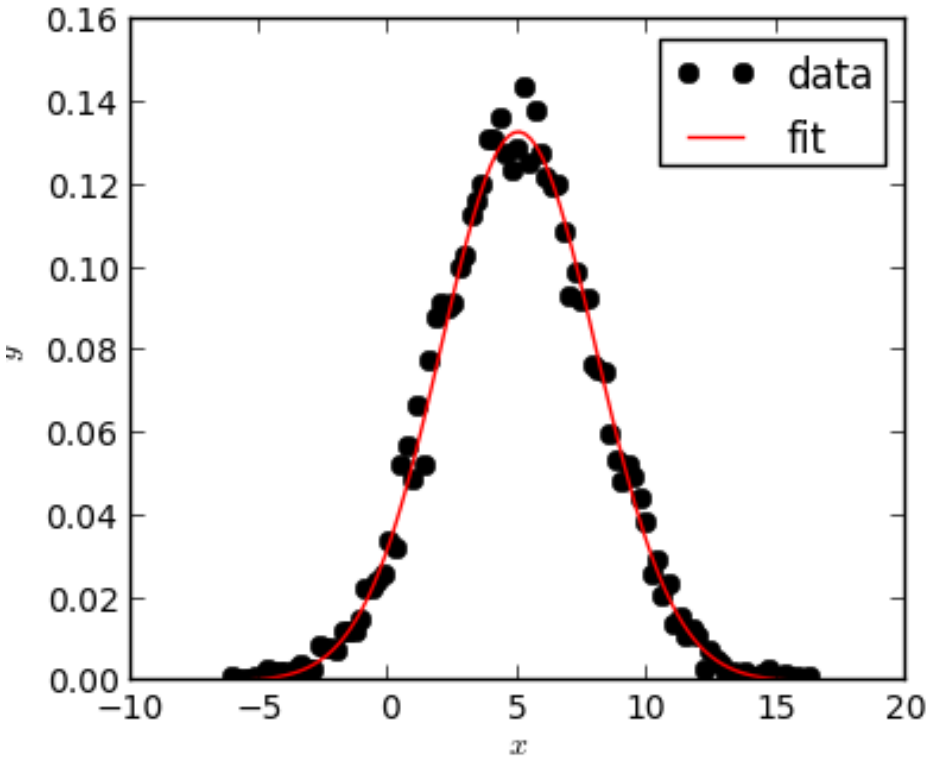
zero dados

regras

poucos dados

estimação paramétrica

template matching



Machine learning ao longo dos anos × dados

zero dados

regras

poucos dados

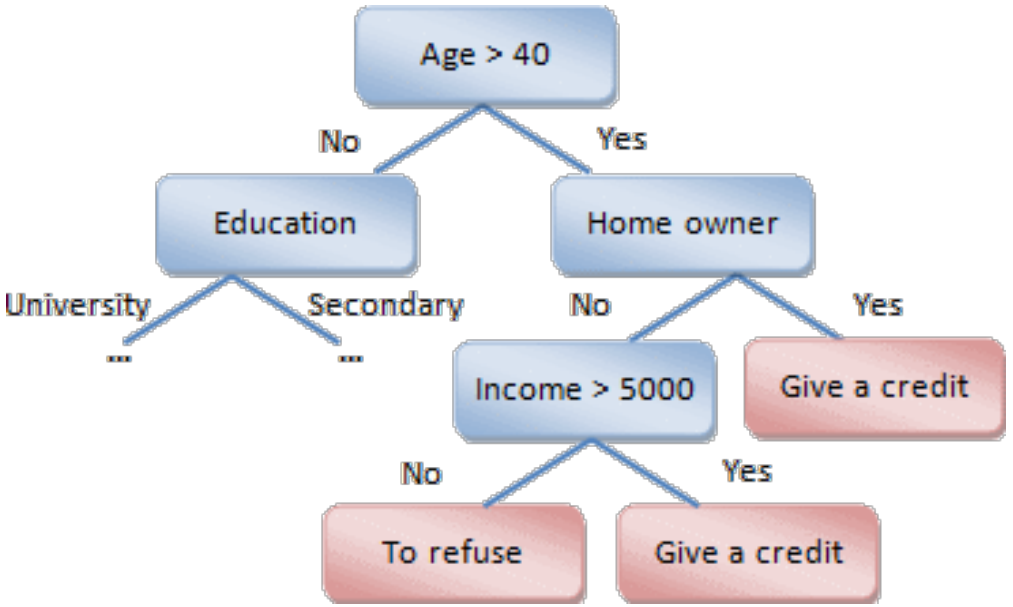
estimação paramétrica

template matching

mais dados

indução de modelos

(algoritmos de ML)



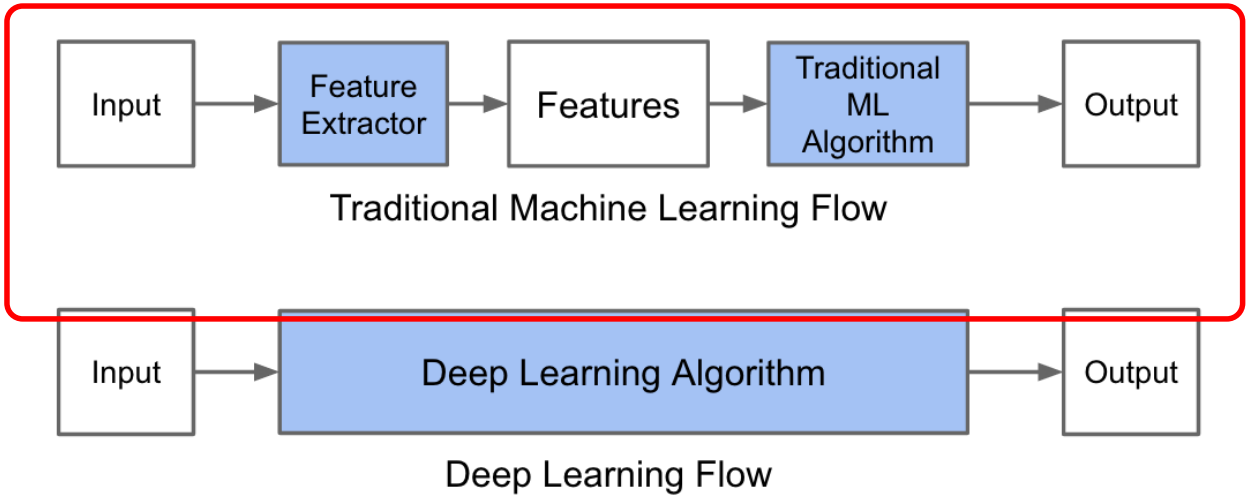
Machine learning ao longo dos anos × dados

zero dados
regras

poucos dados
estimação paramétrica
template matching

mais dados
indução de modelos
(algoritmos de ML)

muito mais dados
engenharia de *features*
(representação de dados)



Machine learning ao longo dos anos × dados

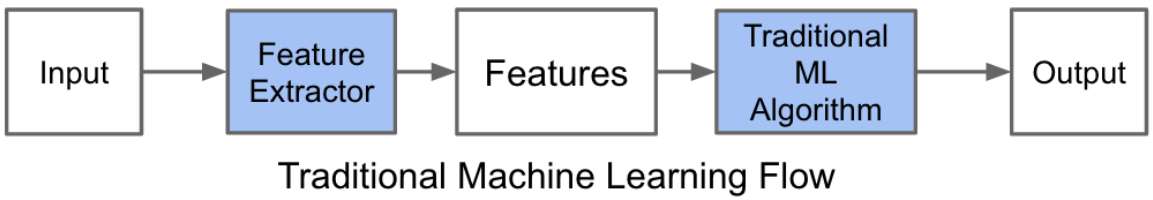
zero dados
regras

poucos dados
estimação paramétrica
template matching

mais dados
indução de modelos
(algoritmos de ML)

muito mais dados
engenharia de *features*
(representação de dados)

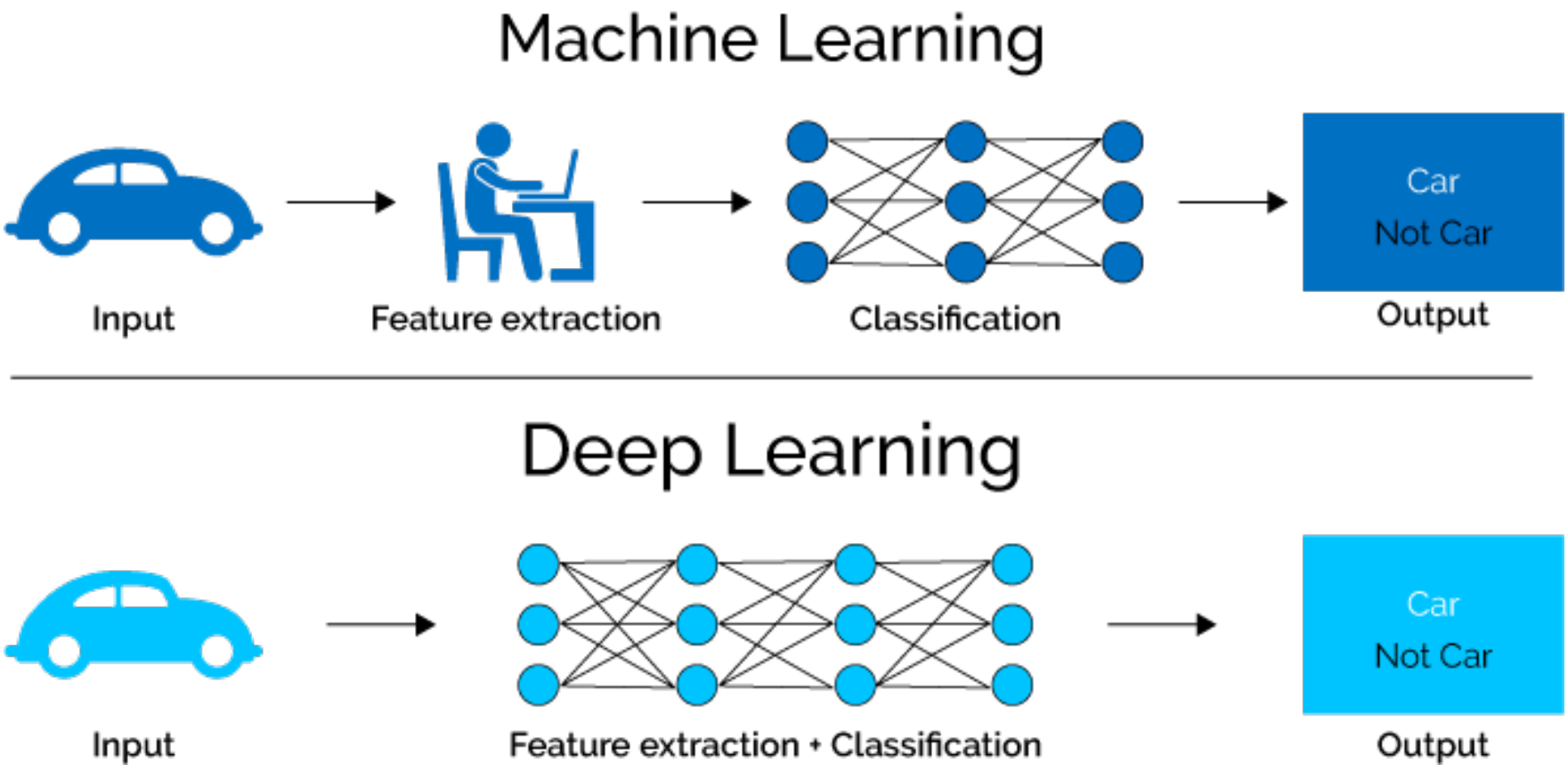
a era do Big Data
aprendizado de representação
(deep learning)



Traditional Machine Learning Flow

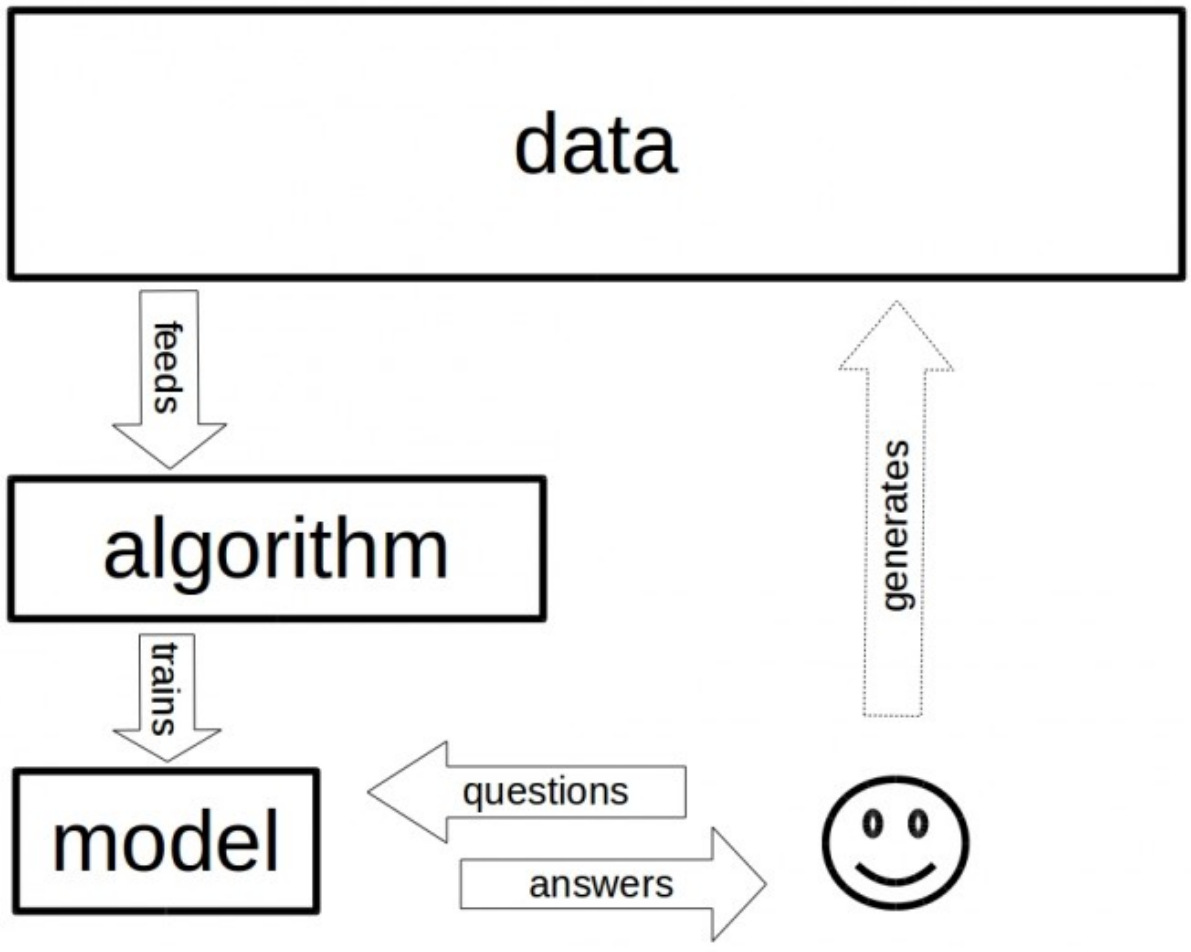
Deep Learning Flow

Traditional ML × Deep Learning



(https://hackernoon.com/log-analytics-with-deep-learning-and-machine-learning-20a1891ff70e)

Data Science



https://pieroit.org/portfolio/how-much-does-machine-learning-cost/

Referências diversas

Python, R : ambiente para prototipação

Scikit-learn : biblioteca de ferramentas úteis em ML
(http://scikit-learn.org)

TensorFlow, PyTorch : framework para desenvolvimento de *deep learning algorithms*

Intro a redes neurais: Livro do Nielsen
(http://neuralnetworksanddeeplearning.com/)

Intro a ML: curso da Caltech (https://work.caltech.edu/telecourse.html)

Comentários finais

- Fizemos uma breve apresentação sobre **noções básicas de ML** (ponto de vista computacional)
- **Quando aplicar ML** : quando há alguma relação entrada-saída, difícil de ser descrita formalmente
- **Falta** : trabalhar o *gap* entre diferentes áreas
- **Utilidade bem pé no chão** : automatizar tarefas de "baixo-nível", para dedicar mais tempo às tarefas de "alto-nível"