

IA aplicada

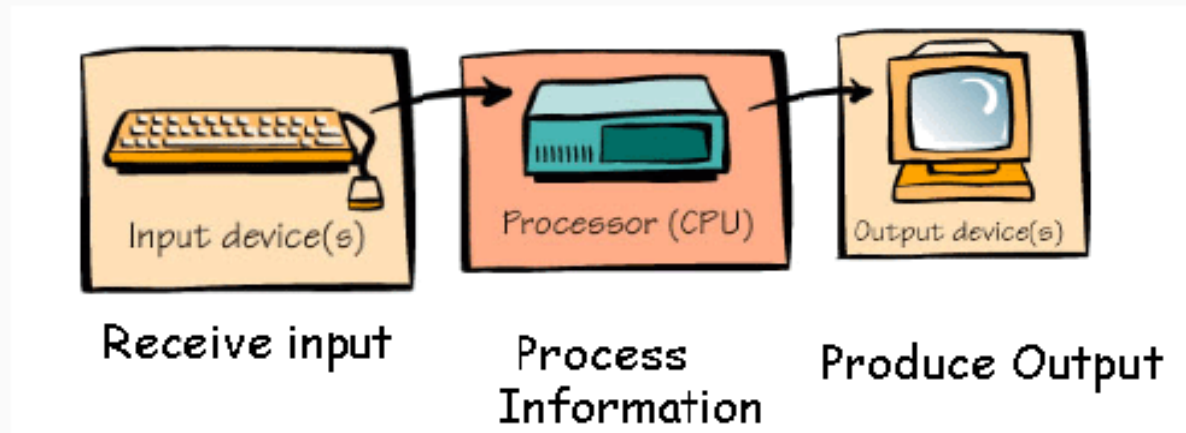
O que é Machine learning?

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

IA aplicada

ML: ponto de vista computacional

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



IA aplicada

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

IA aplicada

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

IA aplicada

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

IA aplicada

Resolução de problemas computacionais – Desafios

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

IA aplicada


E quando não temos/sabemos um algoritmo?

- Como descrever formalmente a relação a seguir:

 \Rightarrow 'a'

 \Rightarrow 'X'

 \Rightarrow 'A'

 \Rightarrow 'd'

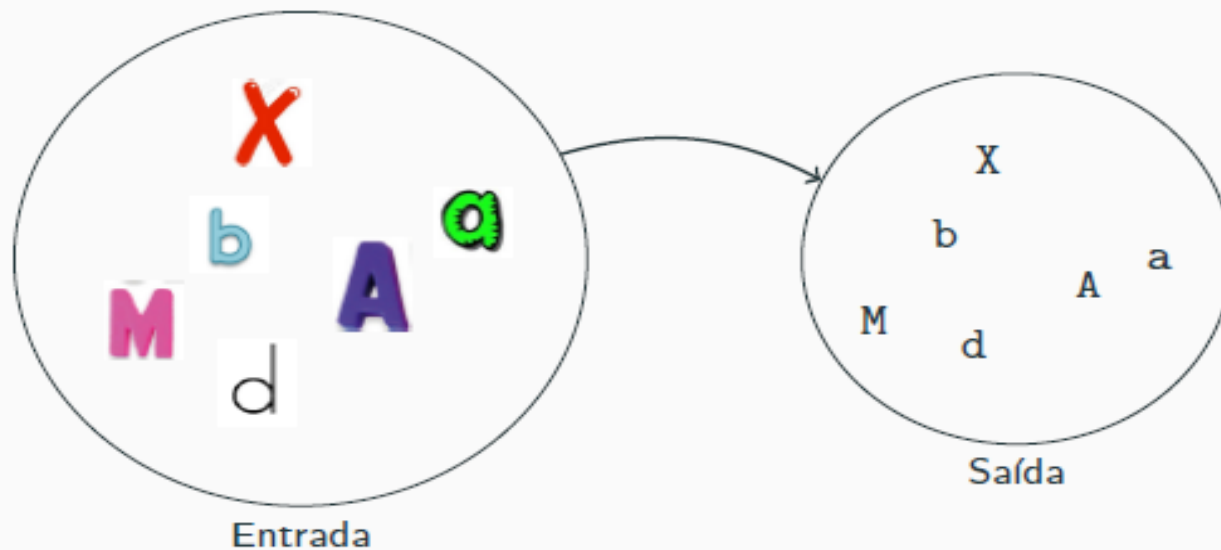
 \Rightarrow 'b'

 \Rightarrow 'M'

IA aplicada

E quando não temos/sabemos um algoritmo?

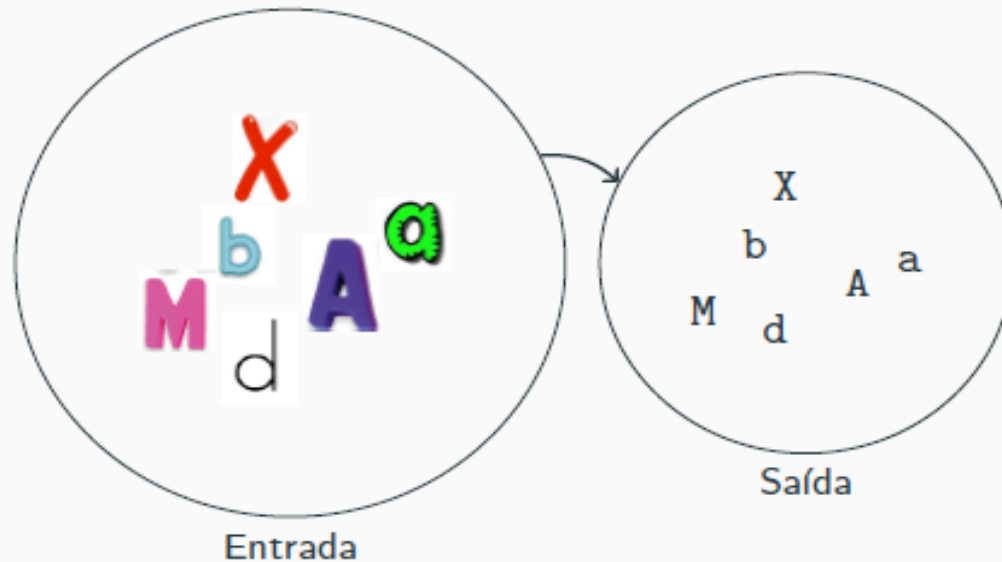
- E se sabemos caracterizar a relação entrada-saída por meio de exemplos, ainda que a relação seja ambígua:



IA aplicada

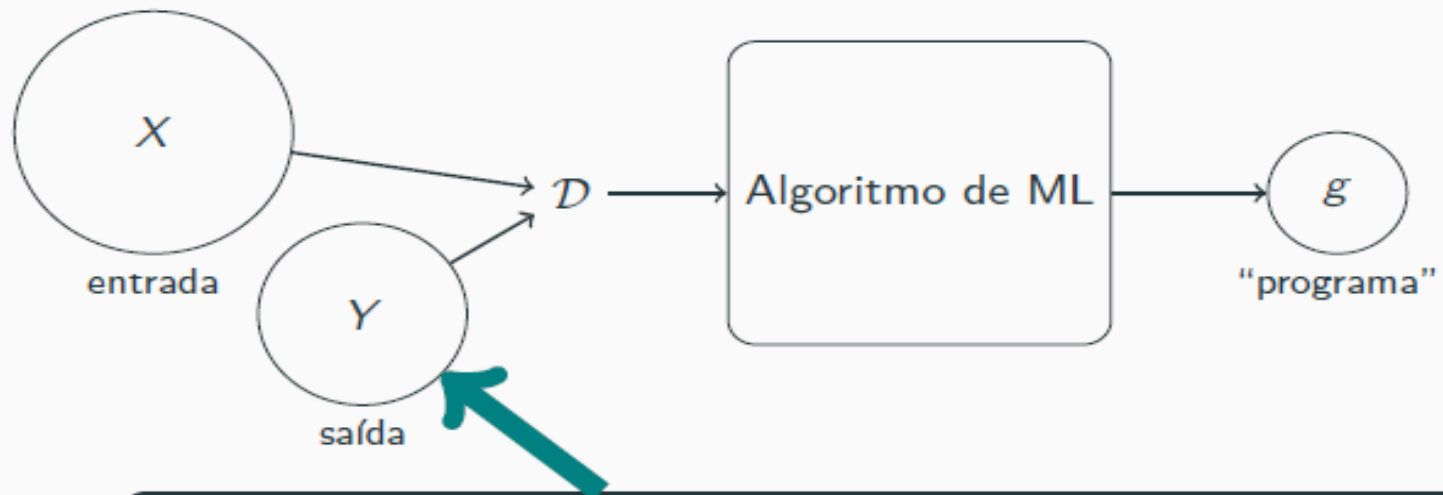
ML entra em cena

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



IA aplicada

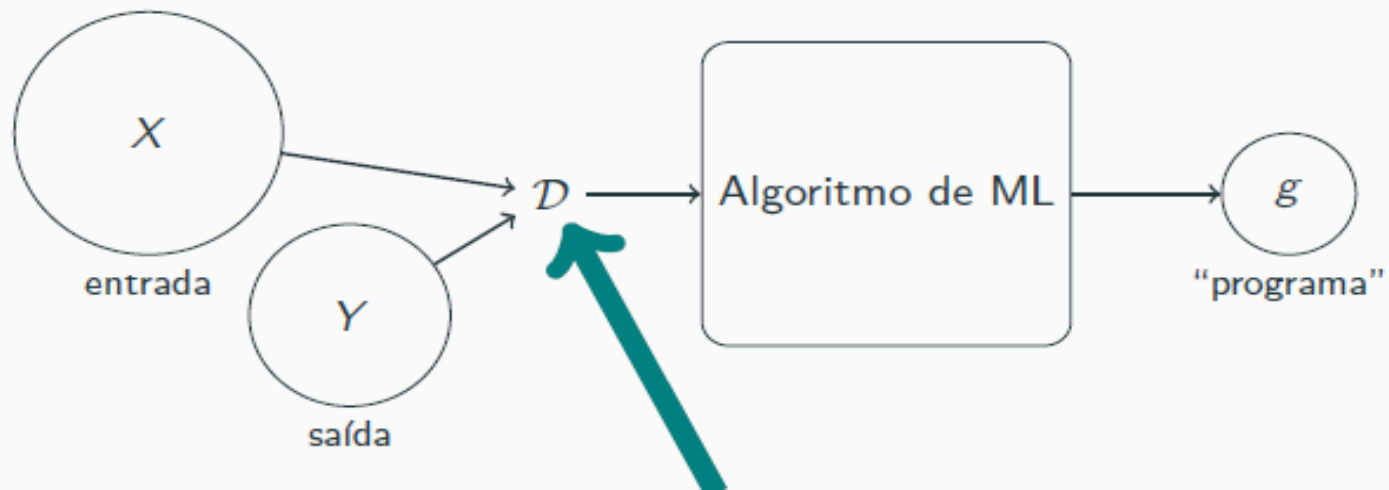
Componentes do processo de ML



Entrada-saída caracteriza o processamento desejado (target)

IA aplicada

Componentes do processo de ML



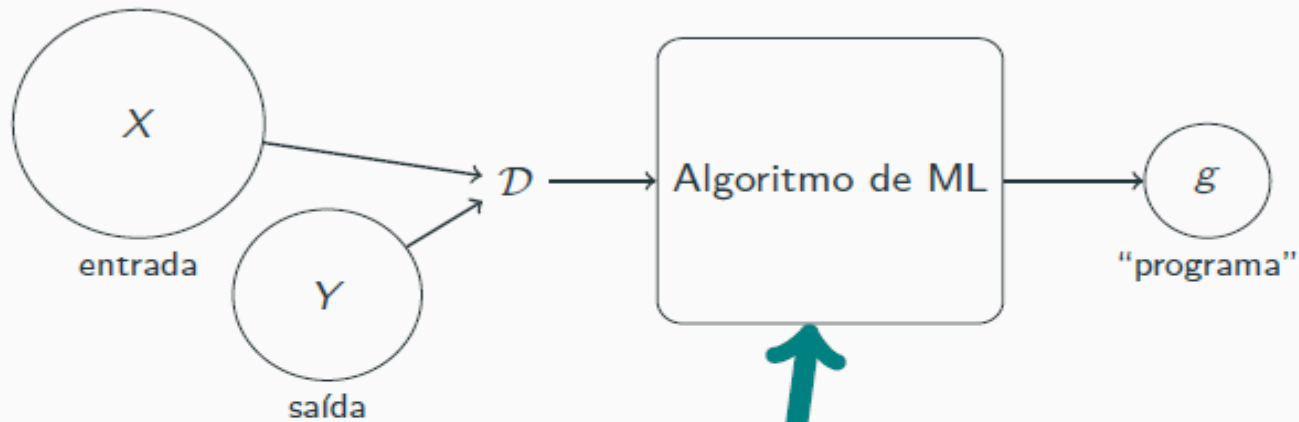
Dados de treinamento

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

(exemplos de relações *input-output*)

IA aplicada

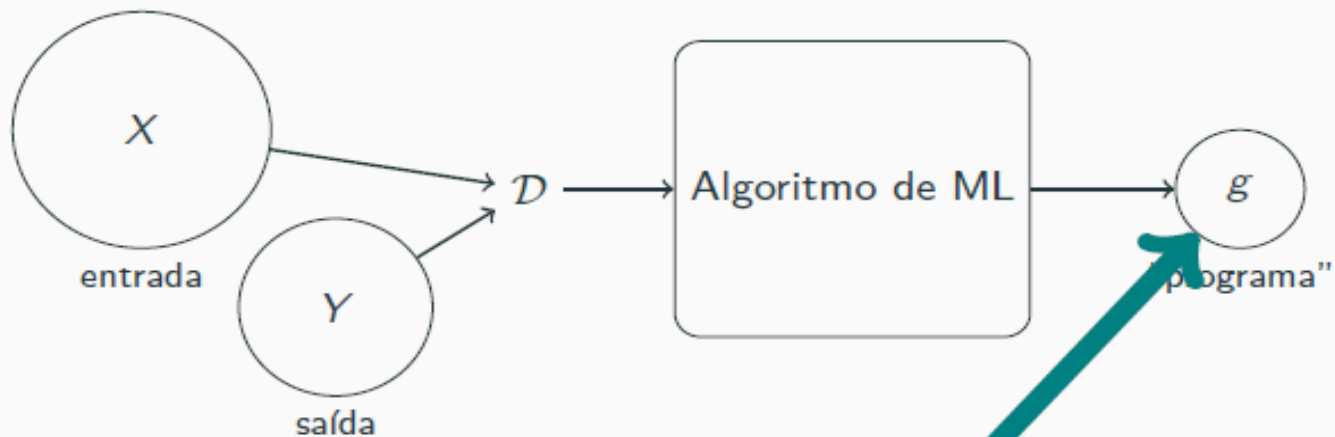
Componentes do processo de ML



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

IA aplicada

Componentes do processo de ML



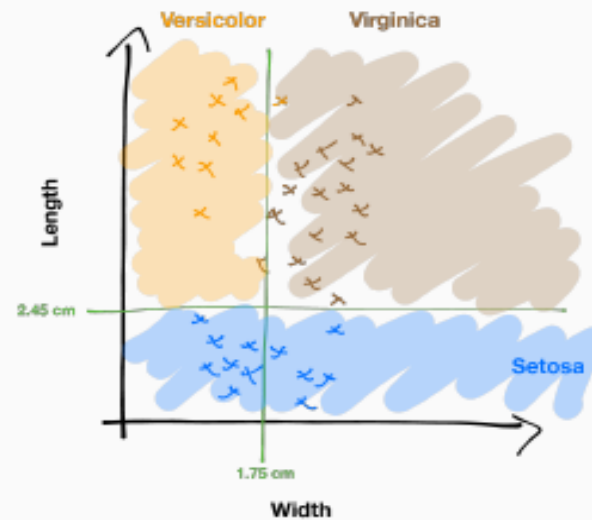
O “programa”, ou **hipótese** $g : X \rightarrow Y$, é o que queremos

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

“o mais próximo possível” de y

IA aplicada

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

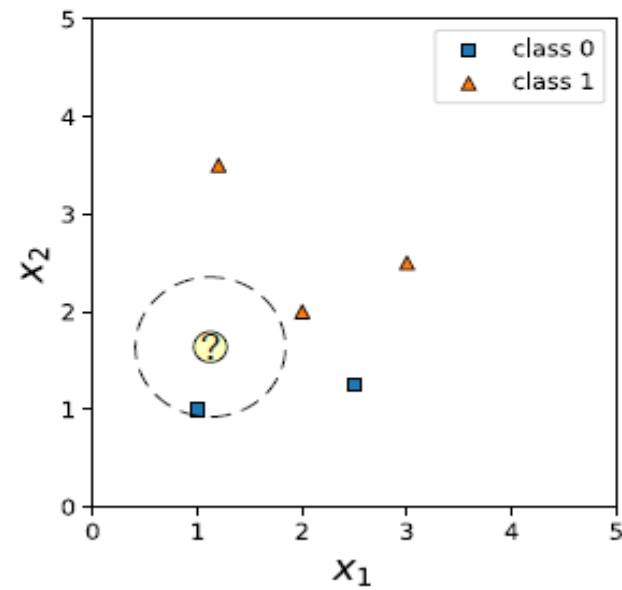
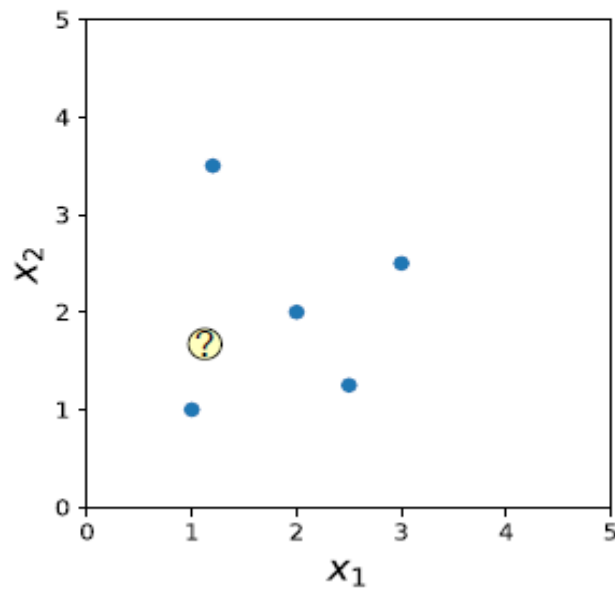


IA aplicada

```
iris.csv x
1 SepalLength[cm],SepalWidth[cm],PetalLength[cm],PetalWidth[cm],Species
2 5.1,3.5,1.4,0.2,Iris-setosa
3 4.9,3.0,1.4,0.2,Iris-setosa
4 4.7,3.2,1.3,0.2,Iris-setosa
5 4.6,3.1,1.5,0.2,Iris-setosa
6 5.0,3.6,1.4,0.2,Iris-setosa
7 5.4,3.9,1.7,0.4,Iris-setosa
8 4.6,3.4,1.4,0.3,Iris-setosa
9 5.0,3.4,1.5,0.2,Iris-setosa
10 4.4,2.9,1.4,0.2,Iris-setosa
11 4.9,3.1,1.5,0.1,Iris-setosa
12 5.4,3.7,1.5,0.2,Iris-setosa
13 4.8,3.4,1.6,0.2,Iris-setosa
14 4.8,3.0,1.4,0.1,Iris-setosa
15 4.3,3.0,1.1,0.1,Iris-setosa
16 5.8,4.0,1.2,0.2,Iris-setosa
17 5.7,4.4,1.5,0.4,Iris-setosa
```

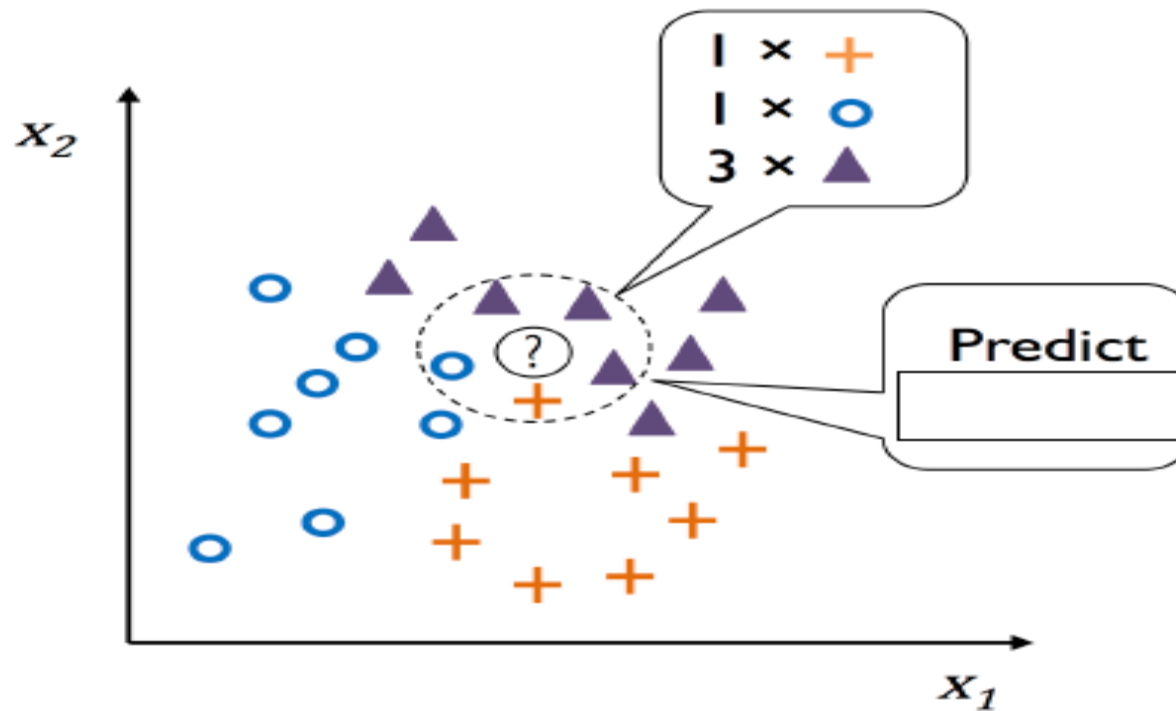
IA aplicada

1-Nearest Neighbor



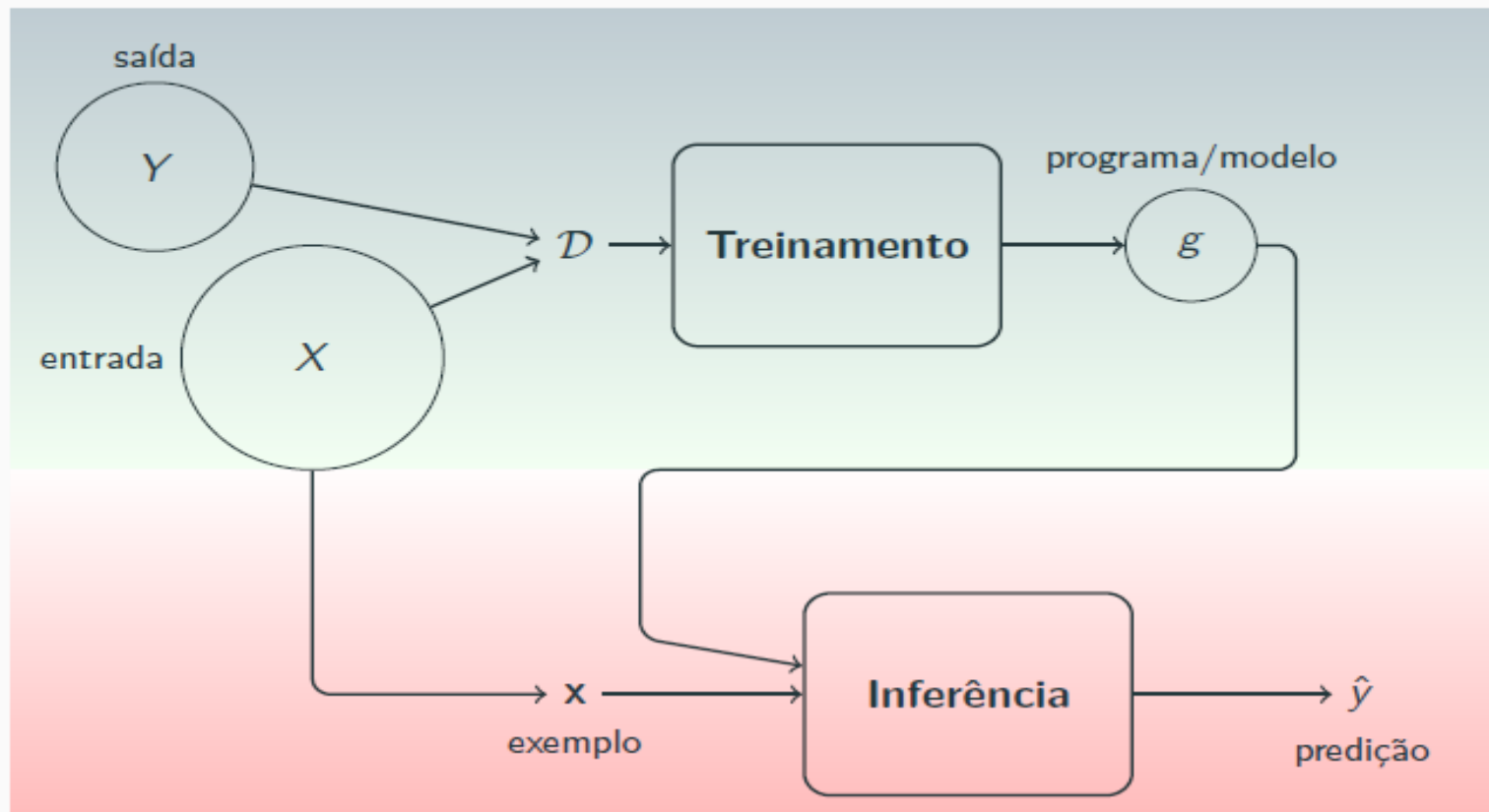
IA aplicada

k-Nearest Neighbors



IA aplicada

Processo de Machine Learning

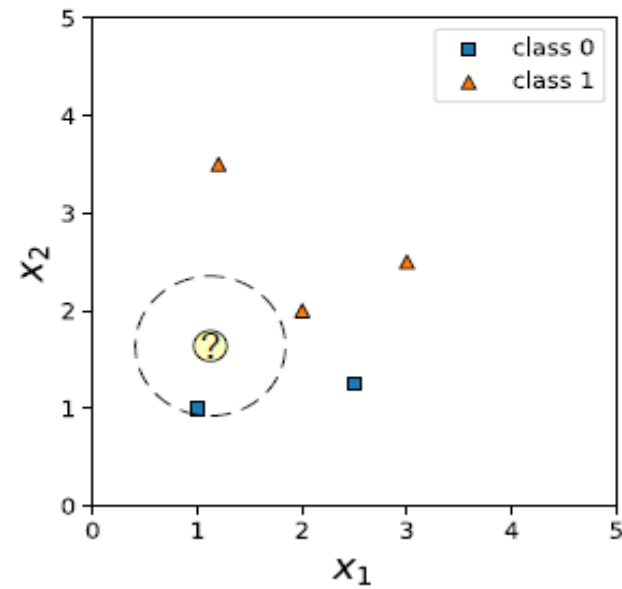
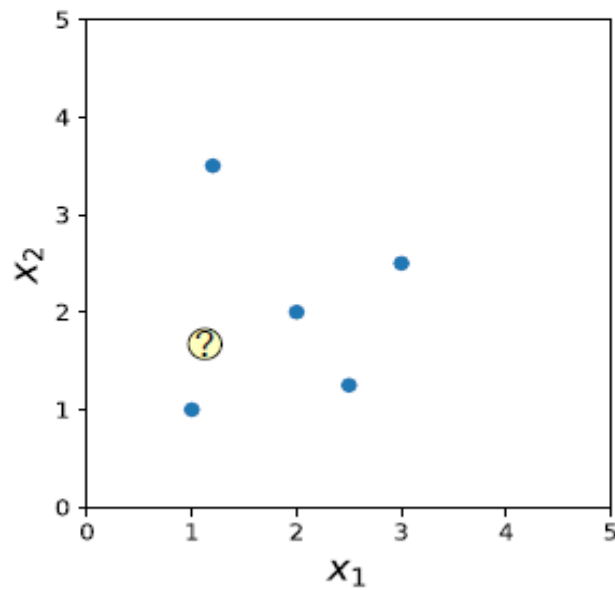


IA aplicada (Exemplo ML Iris)

```
# 5 Uso do KNN (no caso com k=3)
from sklearn.neighbors import KNeighborsClassifier
# Faz o ajuste do modelo na base de treinamento
knn_model = KNeighborsClassifier(n_neighbors=3)
knn_model.fit(X_train, Y_train)
```

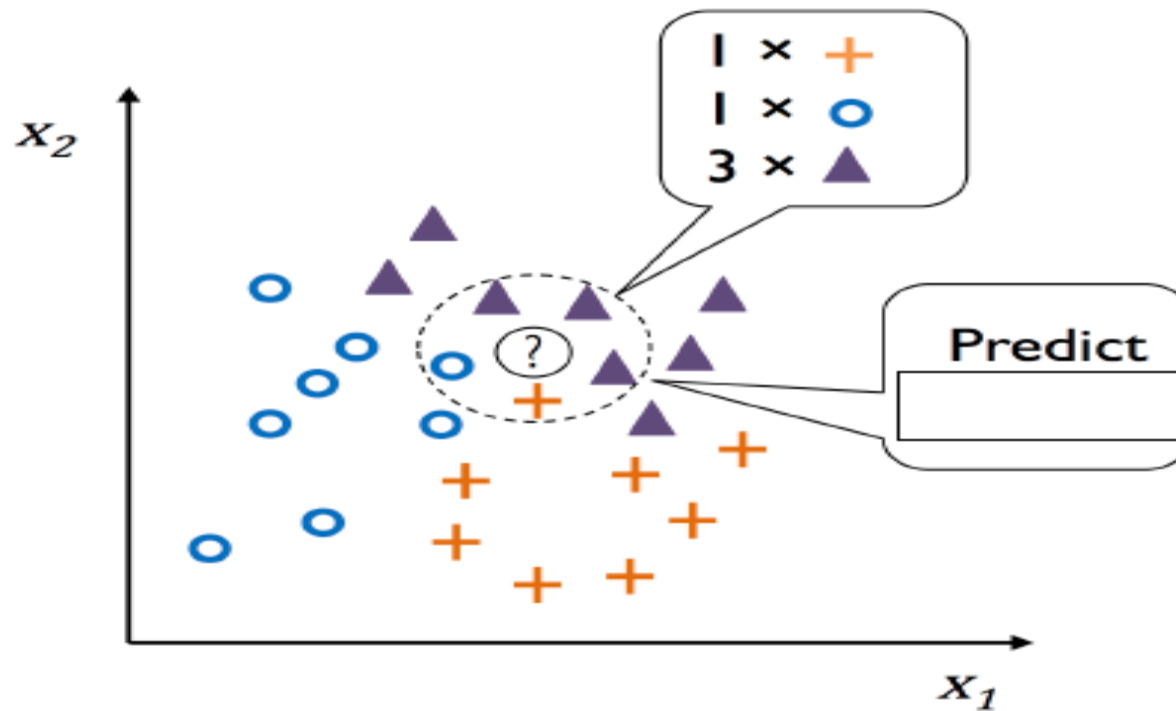
IA aplicada

1-Nearest Neighbor

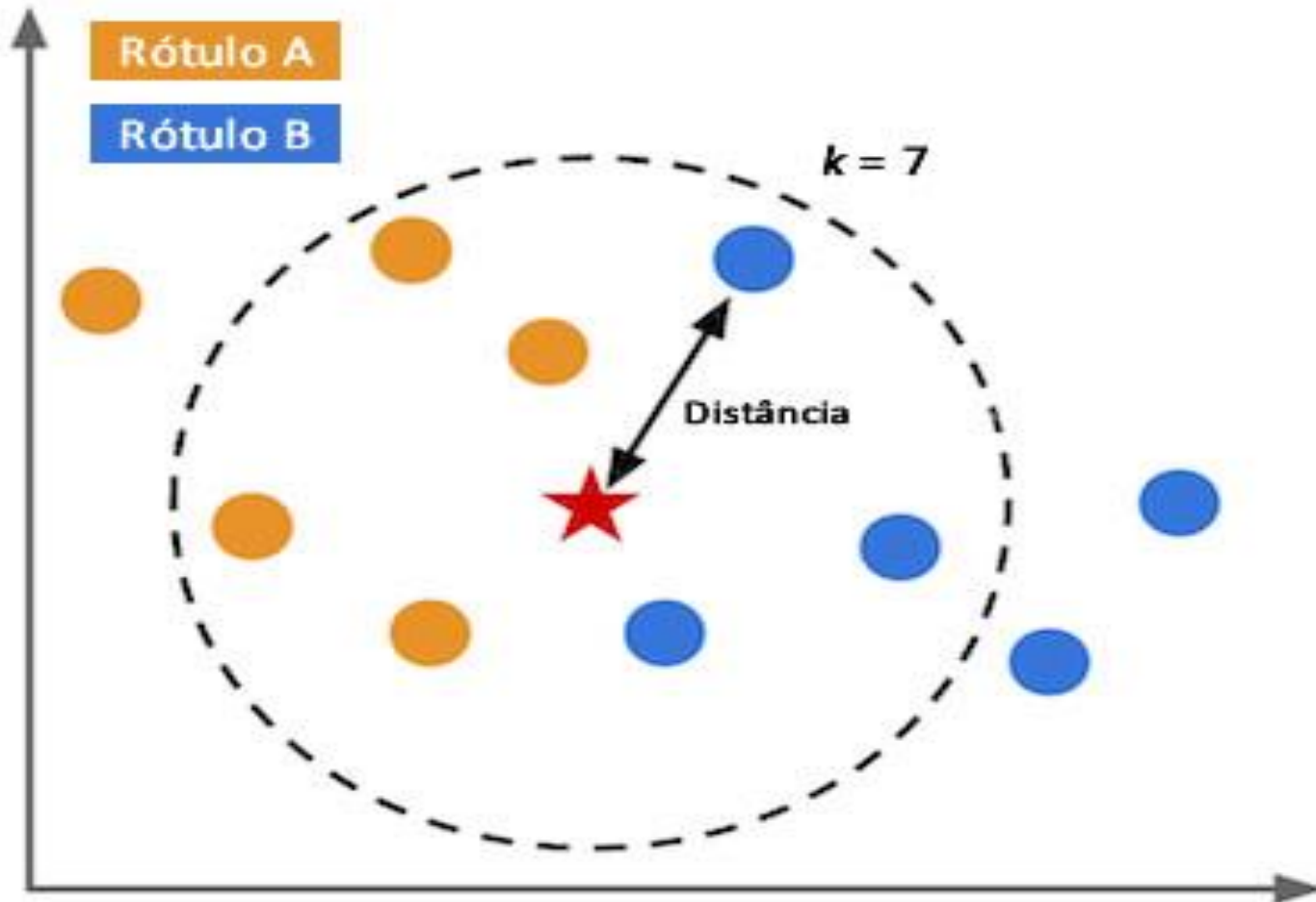


IA aplicada

k-Nearest Neighbors



IA aplicada



IA aplicada

“Machine learning is the hot new thing”

— John L. Hennessy, President of Stanford (2000–2016)

“A breakthrough in machine learning would be worth ten Microsofts”

— Bill Gates, Microsoft Co-Founder

“Machine learning is the field of study that gives computers the ability to learn without being explicitly programmed”

— Arthur L. Samuel, AI pioneer, 1959

IA aplicada

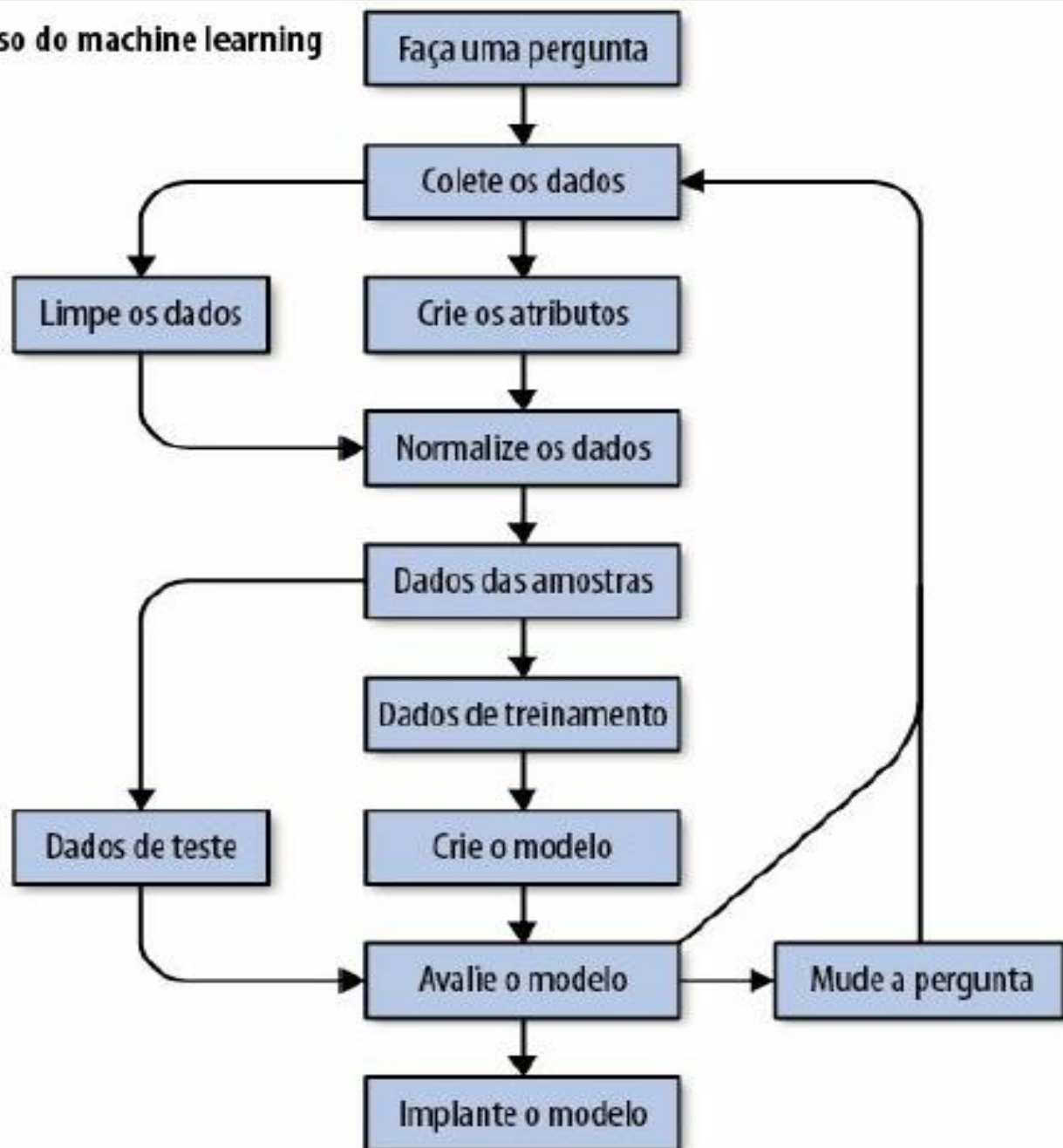
Machine learning is the field of study that gives computers the ability to learn without being explicitly programmed

– *Arthur Samuel (1959)*



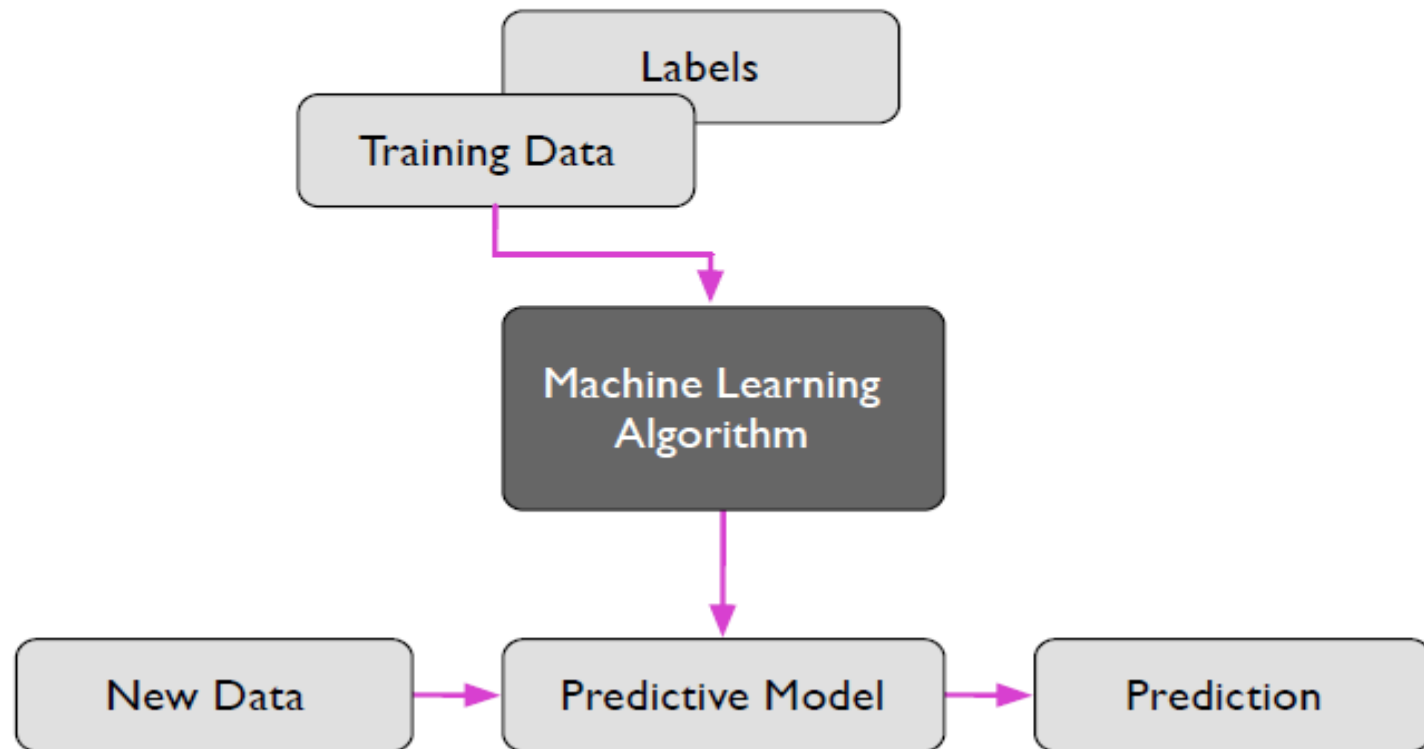
IA aplicada

Processo do machine learning



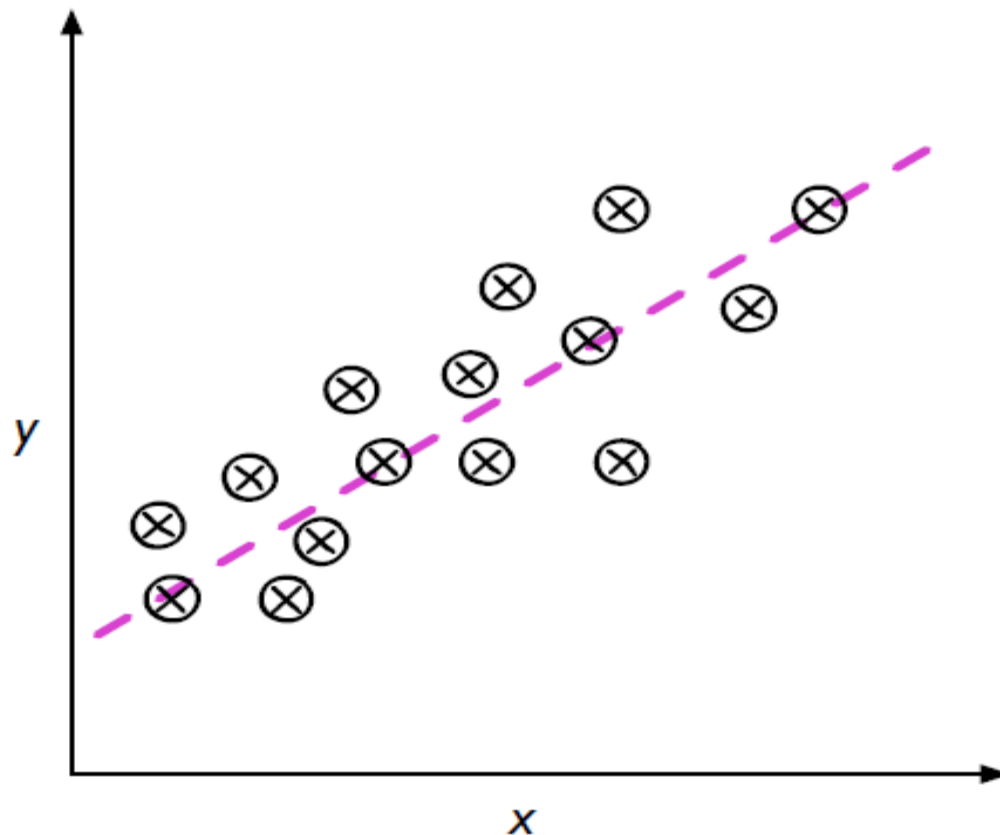
IA aplicada

Supervised Learning Workflow -- Overview



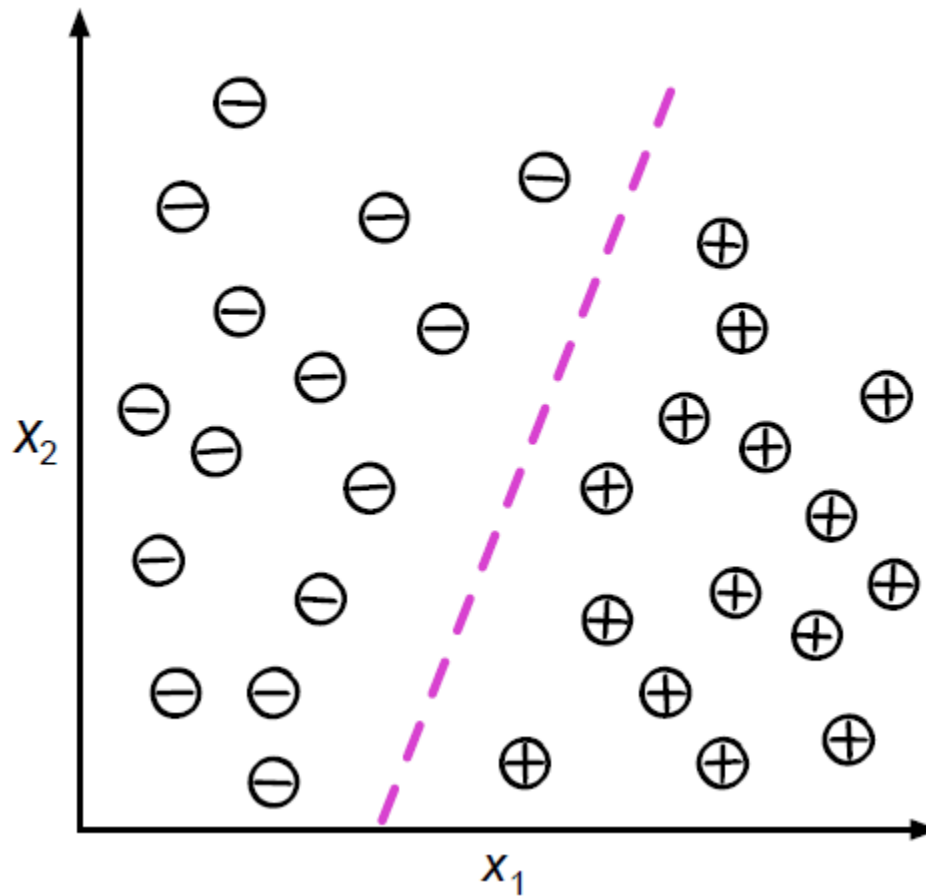
IA aplicada

Supervised Learning: Regression



IA aplicada

Supervised Learning: Classification



IA aplicada

The Iris Dataset



Iris-Setosa



Iris-Versicolor

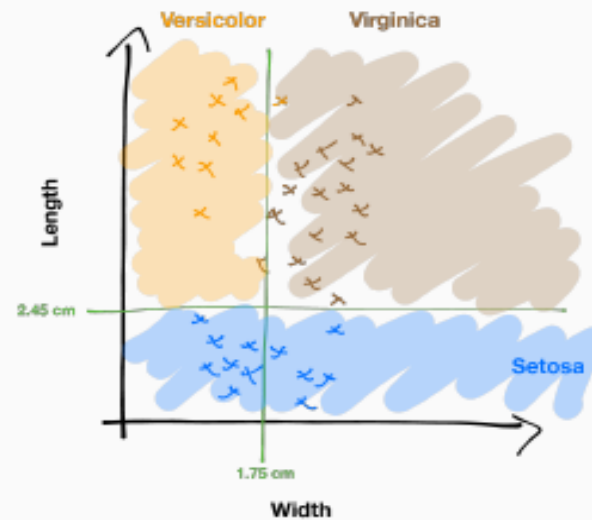


Iris-Virginica

Fisher, R.A. "The use of multiple measurements in taxonomic problems" Annual Eugenics, 7, Part II, 179-188 (1936); also in "Contributions to Mathematical Statistics" (John Wiley, NY, 1950).

IA aplicada

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



IA aplicada

```
# 1
# Biblioteca : Scikit-learn do Python
# Dados : Iris dataset (150 exemplos - 3 tipos de flor)
# Serão criados 2 conjuntos do iris dataset
# (30% test data and 70% training data):
# O iris é nativo : não precisa carregar arquivo

from sklearn.datasets import load_iris
from sklearn.model_selection import train_test_split
```

IA aplicada

```
iris.csv x
1 SepalLength[cm],SepalWidth[cm],PetalLength[cm],PetalWidth[cm],Species
2 5.1,3.5,1.4,0.2,Iris-setosa
3 4.9,3.0,1.4,0.2,Iris-setosa
4 4.7,3.2,1.3,0.2,Iris-setosa
5 4.6,3.1,1.5,0.2,Iris-setosa
6 5.0,3.6,1.4,0.2,Iris-setosa
7 5.4,3.9,1.7,0.4,Iris-setosa
8 4.6,3.4,1.4,0.3,Iris-setosa
9 5.0,3.4,1.5,0.2,Iris-setosa
10 4.4,2.9,1.4,0.2,Iris-setosa
11 4.9,3.1,1.5,0.1,Iris-setosa
12 5.4,3.7,1.5,0.2,Iris-setosa
13 4.8,3.4,1.6,0.2,Iris-setosa
14 4.8,3.0,1.4,0.1,Iris-setosa
15 4.3,3.0,1.1,0.1,Iris-setosa
16 5.8,4.0,1.2,0.2,Iris-setosa
17 5.7,4.4,1.5,0.4,Iris-setosa
```


IA aplicada

```
# 2 Carregar o iris
```

```
iris = load_iris()
```

```
# Carregou os dados do Iris da seguinte forma
```

```
# X ficou com os dados das características (features) escolhidas:
```

```
# PetalLength e PetalWidth)
```

```
# Y ficou com os dados da coluna target (resultados)
```

```
iris
```

```
X, Y = iris.data[:, 2:], iris.target
```

```
X
```

```
Y
```

IA aplicada

```
[6.7, 2.5, 5.8, 1.8],  
[7.2, 3.6, 6.1, 2.5],  
[6.5, 3.2, 5.1, 2. ],  
[6.4, 2.7, 5.3, 1.9],  
[6.8, 3. , 5.5, 2.1],  
[5.7, 2.5, 5. , 2. ],  
[5.8, 2.8, 5.1, 2.4],  
[6.4, 3.2, 5.3, 2.3],  
[6.5, 3. , 5.5, 1.8],  
[7.7, 3.8, 6.7, 2.2],  
[7.7, 2.6, 6.9, 2.3],  
[6. , 2.2, 5. , 1.5],  
[6.9, 3.2, 5.7, 2.3],  
[5.6, 2.8, 4.9, 2. ],  
[7.7, 2.8, 6.7, 2. ],  
[6.3, 2.7, 4.9, 1.8],  
[6.7, 3.3, 5.7, 2.1],
```

IA aplicada

```
[6.2, 3.4, 5.4, 2.3],  
[5.9, 3. , 5.1, 1.8]]),  
'data_module': 'sklearn.datasets.data',  
'feature_names': ['sepal length (cm)',  
 'sepal width (cm)',  
 'petal length (cm)',  
 'petal width (cm)'],  
'filename': 'iris.csv',  
'frame': None,  
'target': array([0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,  
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,  
 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,  
 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,  
 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2,  
 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,  
 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2]),  
'target_names': array(['setosa', 'versicolor', 'virginica'], dtype='<U10')}
```

IA aplicada

[illegible]

IA aplicada

```
# 4 Conferências
```

```
X_train
```

```
X_test
```

```
Y_train
```

```
Y_test
```

IA aplicada

```
# 5 Uso do KNN (no caso com k=3)
from sklearn.neighbors import KNeighborsClassifier
# Faz o ajuste do modelo na base de treinamento
knn_model = KNeighborsClassifier(n_neighbors=3)
knn_model.fit(X_train, Y_train)
```

IA aplicada

```
# 6 Realização das predições
X_valid = [[4.9,1.5]]
X_valid = [[4.9,3.5]]
X_valid = [[1,3]]
Y_pred = knn_model.predict(X_valid)
Y_pred
```

```
array([0])
```

Dados

Qualitativo ou quantitativo?

Quantitativo:

- Idade, preço, temperatura
- Mostrado como média

Dados

Qualitativo ou quantitativo?

Qualitativo (categoria):

- Bairro, sabor favorito, sim/não
- Mostrado como frequência, proporção ou %

Sabor	Frequência	Proporção	%
Chocolate	27	0,3375	33,75%
Baunilha	11	0,1375	13,75%
Flocos	24	0,3	30%
Coco	18	0,225	22,5%

Dados

Aplicações

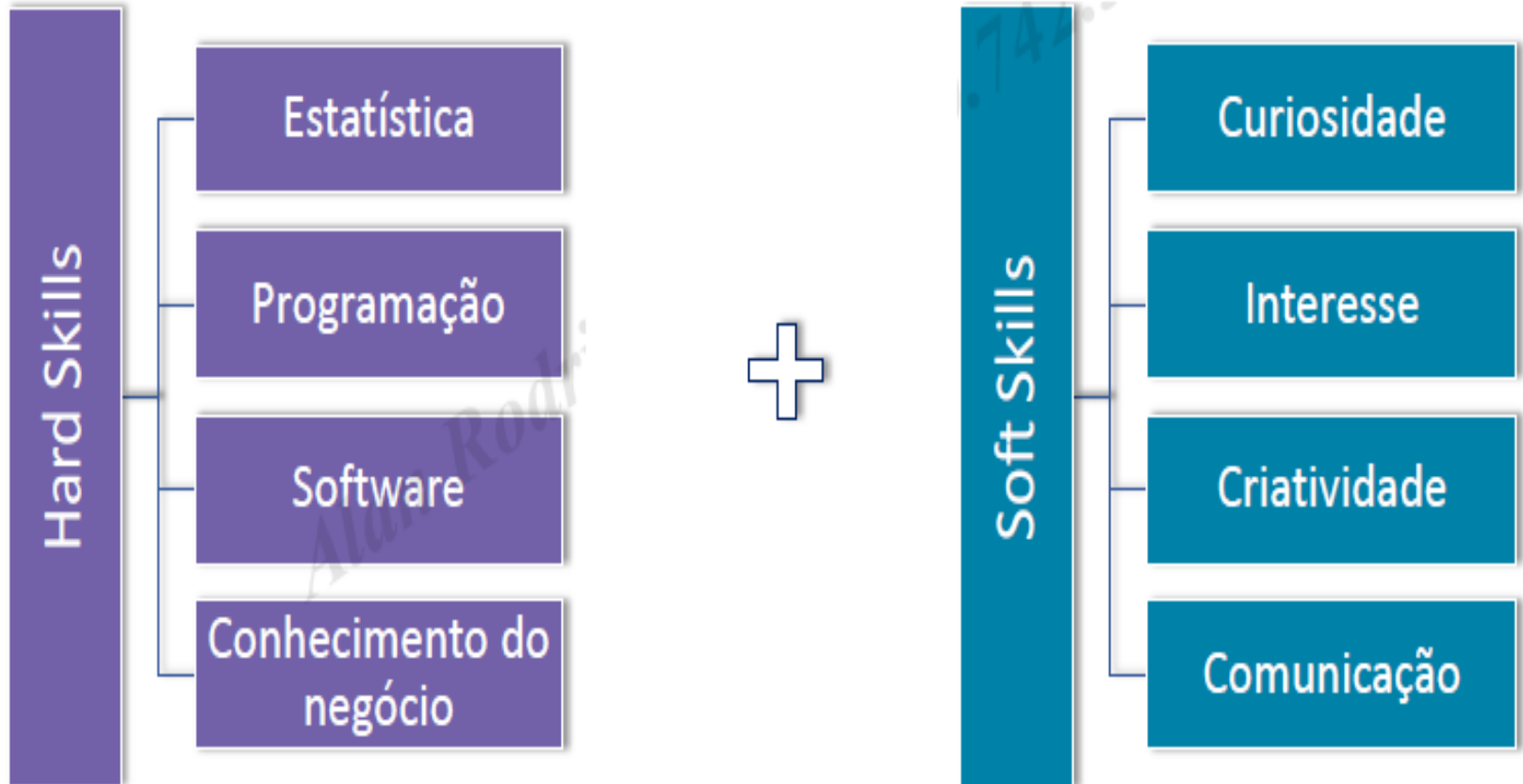
- Relacionamento com o consumidor
 - Entender as preferências do cliente
- Campanhas de marketing
 - Informações de qual tipo de cliente a empresa busca
- Gerenciamento de Supply Chain
 - Mais eficiente com o uso de dados

Dados

Aplicações

- Recursos Humanos
 - Número ideal de empregados e quantos precisam contratar
- Precificação
 - Otimização de preços de produtos
- Estratégia no esporte
 - Analisar a performance do jogador

Dados



Dados

Ferramentas

Excel

- Explorar e analisar dados

Tableau/ Power BI

- Visualizar dados em dashboard

Python/ R

- Construir modelos estatísticos

SQL

- Interação entre as bases de dados

Dados

Processo Analítico



IA aplicada

MACHINE LEARNING



IA aplicada

HOW TO CONFUSE MACHINE LEARNING



IA aplicada

