

# **INTELIGÊNCIA ARTIFICIAL & BIG DATA**

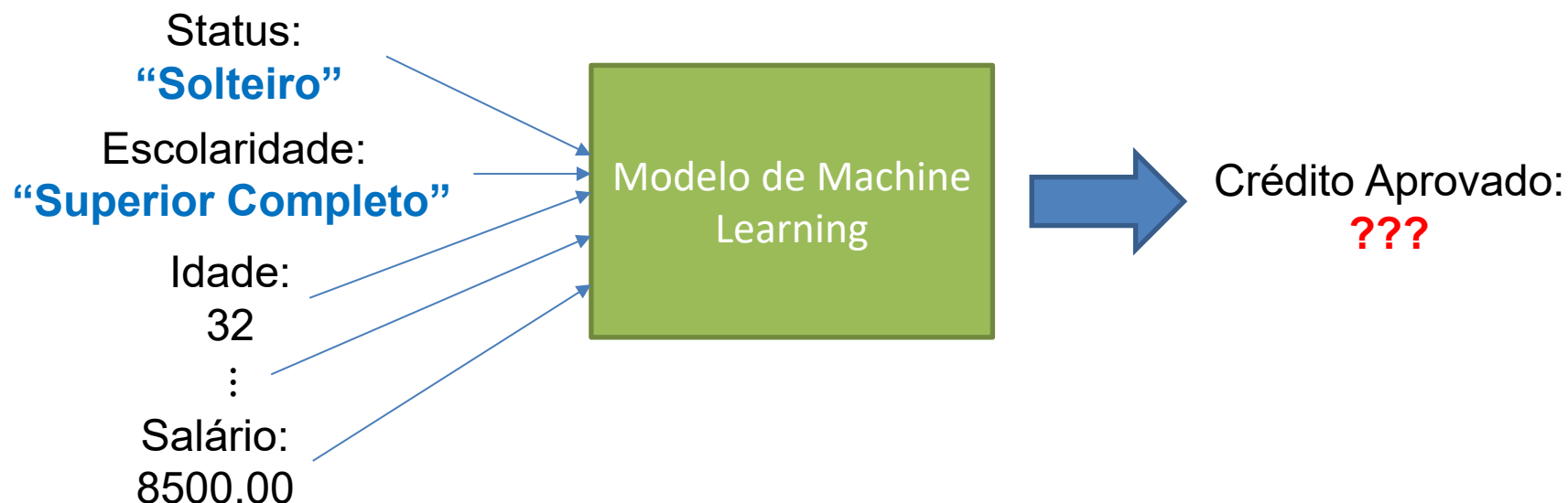
**Profª . Miguel Bozer da Silva**

Prof. Miguel Bozer da Silva

# **PREPARANDO DADOS COM O PANDAS: FOCO EM IA!**

## Preparando Dados com o Pandas

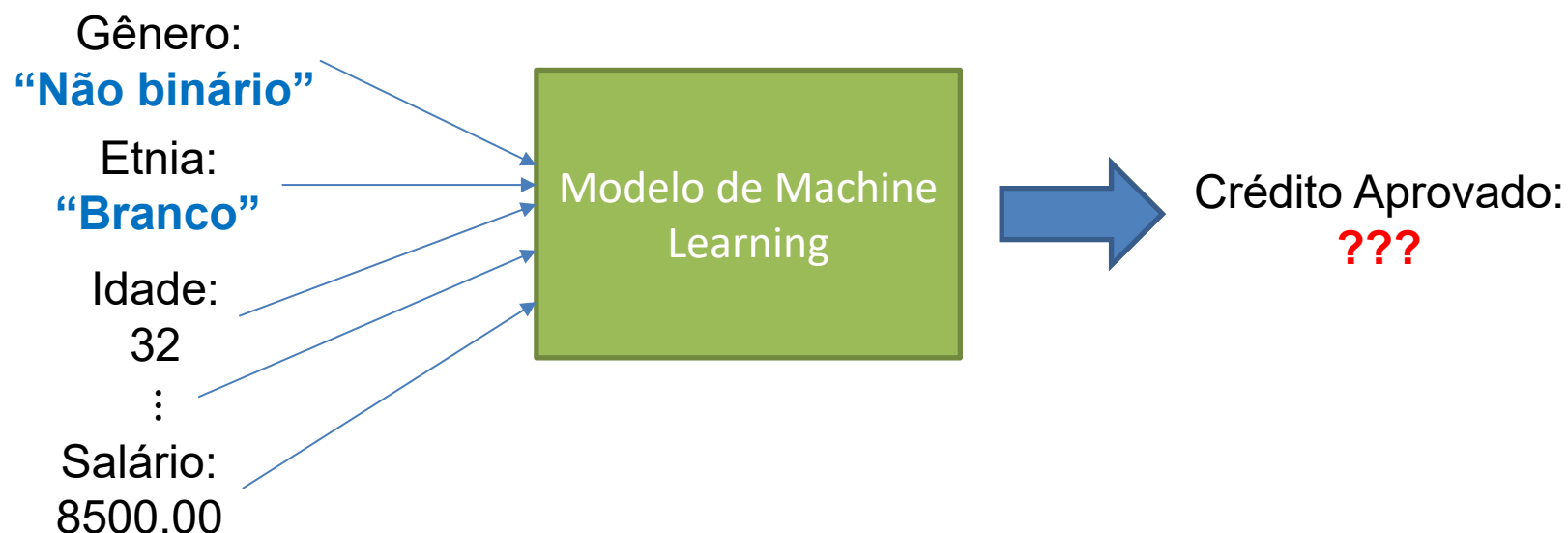
- Possíveis problemas para modelos de *Machine Learning*:
  - Dados armazenados como **strings**:



- Nesses casos, onde temos colunas com dados armazenados como **strings**, temos que ajustá-los para valores numéricos.
  - Modelos de Machine Learning irão realizar **operações matemáticas** com os dados de entrada, logo os valores devem ser numéricos

# Preparando Dados com o Pandas

- Possíveis problemas para modelos de *Machine Learning*:
  - Dados armazenados como *strings*:



- Nesses casos, onde temos colunas com dados armazenados como *strings*, temos que ajustá-los para valores numéricos.
  - Temos duas principais abordagens para quando **conseguimos** estabelecer uma ordenação dos dados e quando **não conseguimos** fazer isso.

## One Hot Enconding

- Quando não conseguimos ordenar os dados usamos o One Hot Enconding

**Imagine que uma das colunas do seu conjunto de dados tem o tipo *string***

FRUTA
MAÇA
BANANA
BANANA
MAÇA
MANGA



**Não há como criar uma ordem entre os elementos, por exemplo:**

**A média entre banana e manga é uma maçã?**

**Se a pergunta de média entre elementos não faz sentido, os mesmos não podem ser ordenados.  
Logo usamos o One Hot Encoding**

# One Hot Enconding

- Quando não conseguimos ordenar os dados usamos o One Hot Enconding

**O One Hot Encoding irá transformar uma coluna catagórica em colunas binárias (contendo 0 ou 1):**

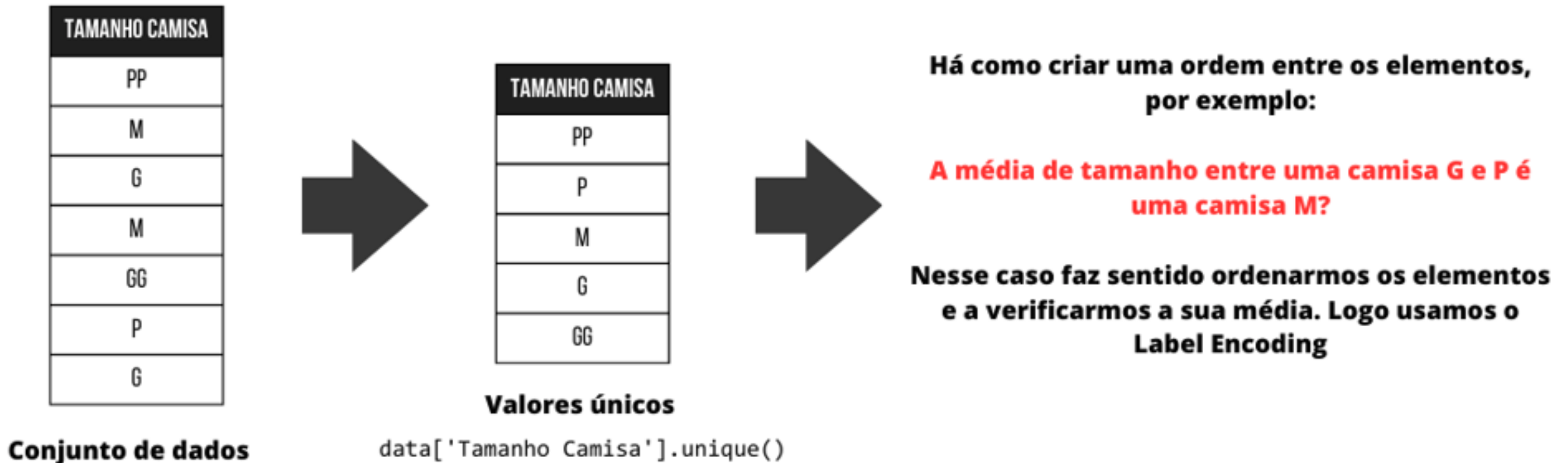
FRUTA		MAÇA	BANANA	MANGA
MAÇA		1	0	0
BANANA		0	1	0
BANANA		0	1	0
MAÇA		1	0	0
MANGA		0	0	1

```
one_hot_encoded_data = pd.get_dummies(data, columns = ['Fruta'])
```

# Label Enconding

- Quando conseguimos ordenar os dados usamos o Label Enconding

Imagine um outro cenário onde temos uma das colunas do seu conjunto de dados com o tipo *string*



# Label Encoding

O Label Encoding irá transformar uma coluna catagórica em uma coluna numérica

TAMANHO CAMISA
PP
M
G
M
GG
P
G

**Conjunto de dados**



TAMANHO CAMISA
PP
P
M
G
GG

**Valores únicos**

```
data['Tamanho Camisa'].unique()
```



TAMANHO CAMISA (NUMÉRICO)
0
1
2
3
4

**Valores únicos, forma numérica**

```
data['Tamanho Camisa'].replace({'PP':0,  
                                'P':1,  
                                'M':2,  
                                'G':3,  
                                'GG':4},  
                                inplace = True)
```

TAMANHO CAMISA
0
2
3
2
4
1
3

**Coluna com dados ajustados**



## Exercício



- No jupyter Notebook: “2. Label Encoding e One Hot Encoding.ipynb” vamos ver como usar os conceitos de Label Encoding e One Hot Encoding

# Preparando Dados com o Pandas



- Possíveis problemas para modelos de *Machine Learning*:
  - Dados nulos.
    - Pelo mesmo motivo dos dados armazenados como texto, os valores nulos não podem ser processados durante o treinamento ou o teste.
    - A exclusão das linhas com valores nulos ou a substituição dos valores já foram vistos no semestre passado e serão apresentados posteriormente em outros projetos que iremos trabalhar!

Prof. Miguel Bozer da Silva

# **APRENDIZADO SUPERVISIONADO**

- O que é o aprendizado supervisionado?
- No aprendizado supervisionado temos os dados de entrada do nosso modelo e também conhecemos os *labels* deles, isto é o valor esperado da saída do modelo para cada entrada:

$\mathbf{x}^{(i)}$  { i-ésima entrada do nosso modelo. Aqui temos todas as características diferentes que iremos utilizar para fazermos uma predição da saída ( $\hat{\mathbf{y}}^{(i)}$ )

$\mathbf{y}^{(i)}$  { Label com a saída esperada pelo nosso modelo. A diferença

# Aprendizado Supervisionado

- **O que é o aprendizado supervisionado?**
- No aprendizado supervisionado temos os dados de entrada do nosso modelo e também conhecemos os *labels* deles, isto é o valor esperado da saída do modelo para cada entrada:

Salario mensal	Nível de educação	Moradia	Aprovação do cartão de crédito
8500,00	Mestrado	Casa/Apartamento Próprio quitado	Aprovado
1950,00	Ensino médio / técnico	Aluguel	Reprovado
⋮	⋮	⋮	⋮
3500,00	Graduação	Casa/Apartamento Próprio financiado	Reprovado

$\mathbf{x}^{(i)}$  Entrada de dados

$y^{(i)}$  Saída de dados

# Aprendizado Supervisionado

- Para o caso dos **classificadores**, conhecemos os nossos dados de entrada ( $x$ ) e conhecemos os labels dele ( $y$ ) que são **categóricos**

altura	peso	Classe
1,83	95	adulto
1,65	77	adulto
1,25	50	criança
⋮	⋮	⋮
1,77	69	adulto

$x^{(i)}$  Entrada de dados

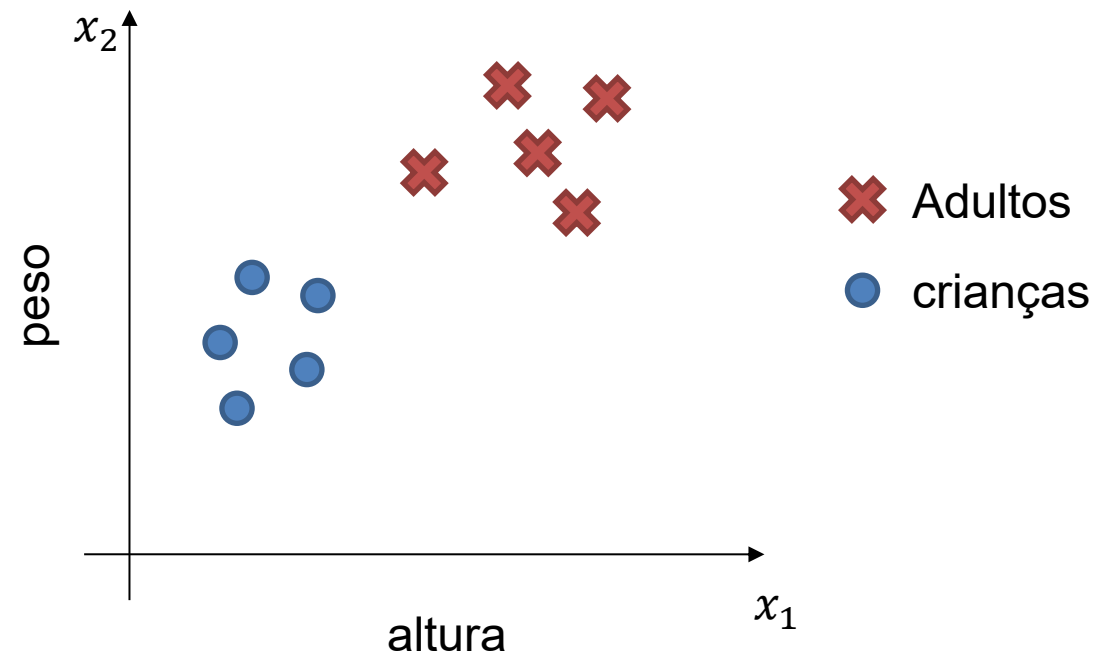
$y^{(i)}$  Saída de dados

# Aprendizado Supervisionado

- Os modelos **classificadores** irão estimar parâmetros ( $\theta$ ) que nos indicam a relação entre as nossas entradas ( $\mathbf{x}$ ) e a nossa saída – *label* ( $y$ )

$\mathbf{x} \rightarrow \theta \rightarrow y$

altura	peso	Classe
1,83	95	adulto
1,65	77	adulto
1,25	50	criança
⋮	⋮	⋮
1,77	69	adulto



$$\hat{y} = f(\theta) = \begin{cases} 0 & \text{se criança} \\ 1 & \text{se adulto} \end{cases}$$

**Observação: Podemos ter mais de duas classes nos nossos problemas!**

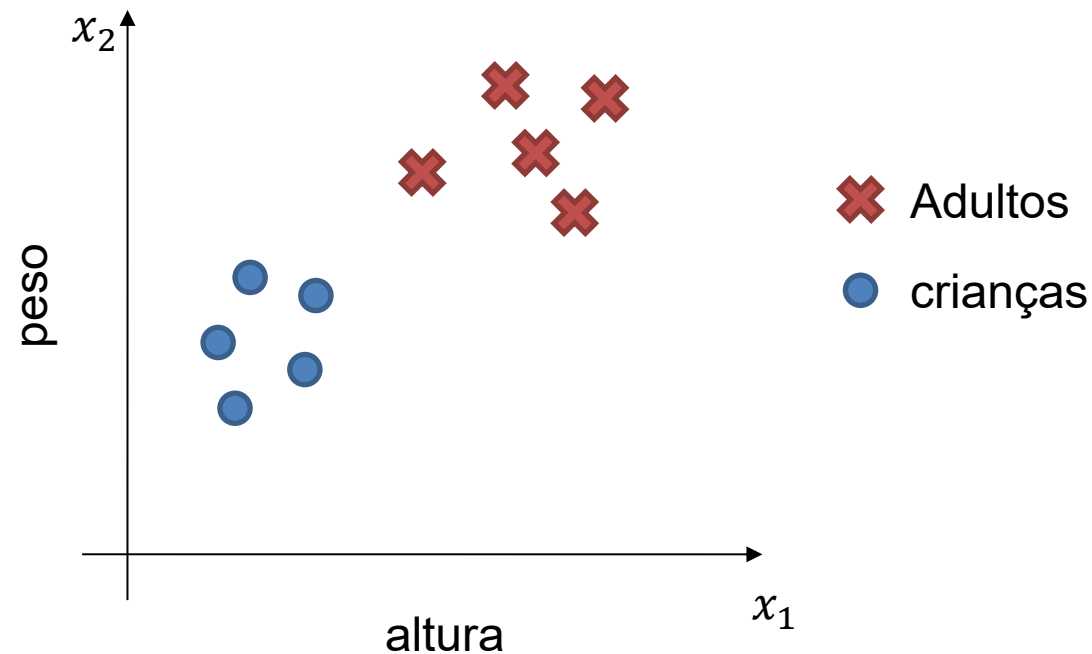
# Aprendizado Supervisionado

- A etapa de aprendizado no nosso modelo  $f(\theta)$  é chamada de **treinamento**. Nela o modelo aprenderá a relação das entradas com as saídas.

$x \rightarrow \theta \rightarrow y$

altura	peso	Classe
1,83	95	adulto
1,65	77	adulto
1,25	50	criança
⋮	⋮	⋮
1,77	69	adulto

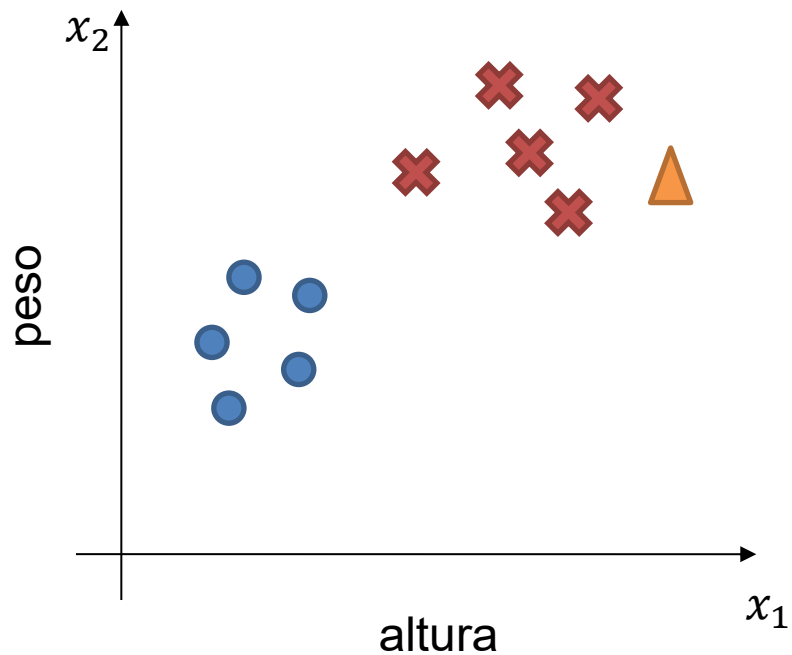
$$\hat{y} = f(\theta) = \begin{cases} 0 & \text{se criança} \\ 1 & \text{se adulto} \end{cases}$$



**Observação: Podemos ter mais de duas classes nos nossos problemas!**



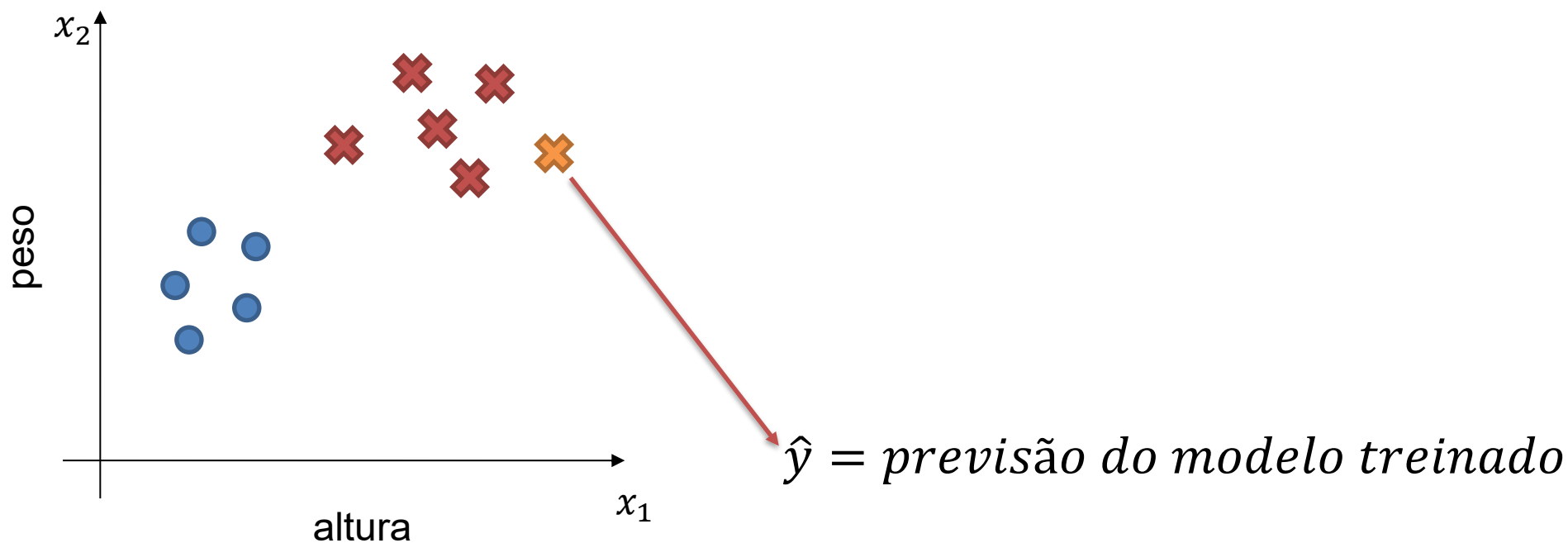
- Após o treinamento, podemos usar o nosso modelo para estimar dados desconhecidos: **Caso um novo dado** cuja classe é desconhecida for apresentado ao modelo, podemos classifica-lo!



**Observação: Podemos ter mais de duas classes nos nossos problemas!**

# Aprendizado Supervisionado

- Após o treinamento, podemos usar o nosso modelo para estimar dados desconhecidos: **Caso um novo dado** cuja classe é desconhecida for apresentado ao modelo, podemos classifica-lo!



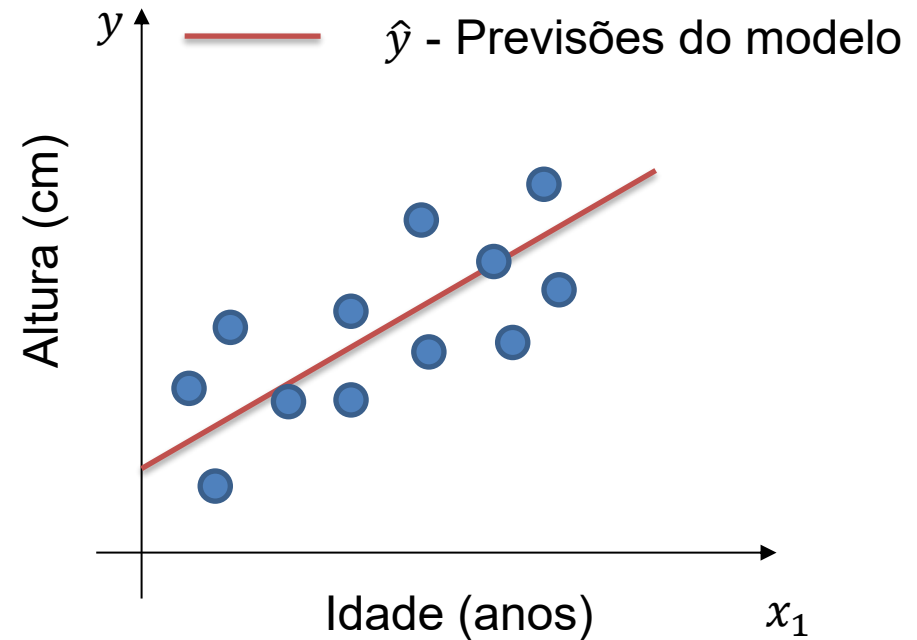
**Observação: Podemos ter mais de duas classes nos nossos problemas!**

# Aprendizado Supervisionado

- Podemos também estimar uma saída de valores **numéricos contínuos** ( $y$ ) a partir de um conjunto de dados de entrada ( $\mathbf{x}$ )

idade	altura
5	1,00
11	1,43
7	1,19
$\vdots$	$\vdots$
16	1,73

$\mathbf{x}^{(i)}$        $y^{(i)}$

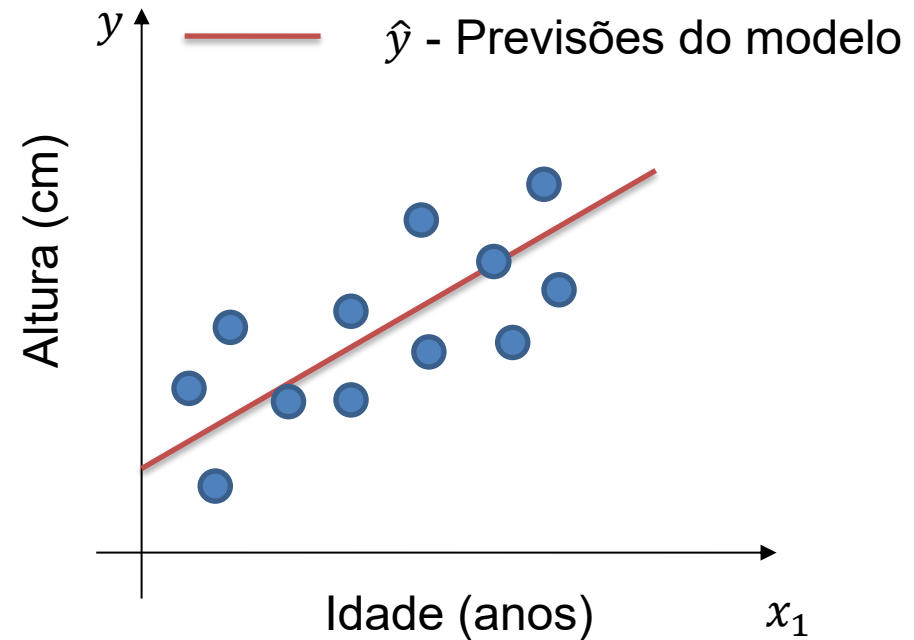


# Aprendizado Supervisionado

- Podemos também estimar uma saída de valores **numéricos contínuos** ( $y$ ) a partir de um conjunto de dados de entrada ( $\mathbf{x}$ )

idade	altura
5	1,00
11	1,43
7	1,19
$\vdots$	$\vdots$
16	1,73

$\mathbf{x}^{(i)}$        $y^{(i)}$

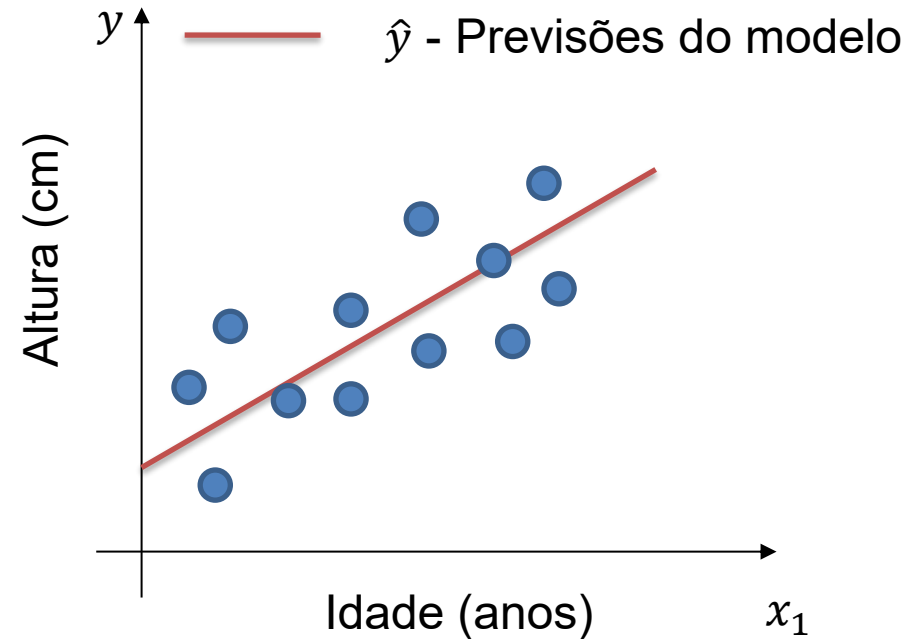
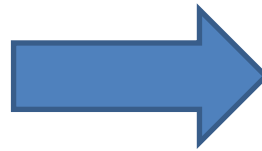


# Aprendizado Supervisionado

- Nesses casos temos a necessidade de utilizar modelos **regressores** para resolver o problema que estamos trabalhando

idade	altura
5	1,00
11	1,43
7	1,19
⋮	⋮
16	1,73

$\mathbf{x}^{(i)}$        $y^{(i)}$

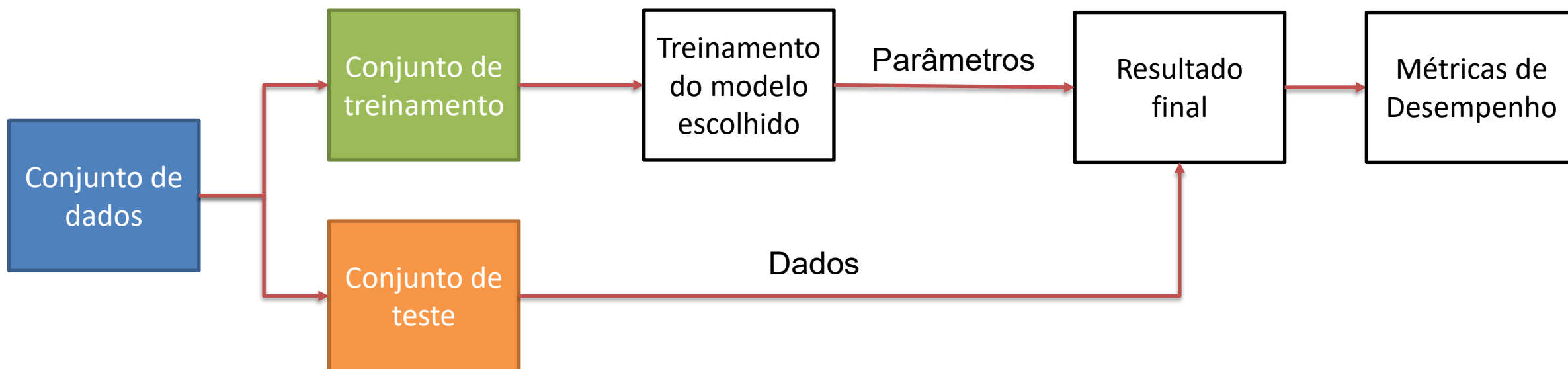


Prof. Miguel Bozer da Silva

# **DIVISÃO DOS CONJUNTOS DE DADOS**

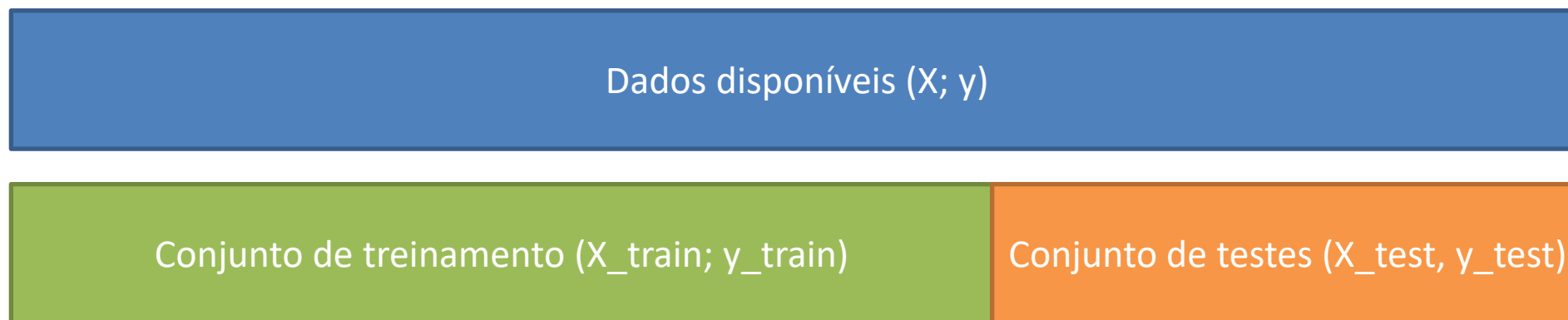
## Divisão dos Conjuntos de Dados

- Para usarmos modelos de Machine Learning temos que criar os conjuntos de dados para treinamento e teste



## Divisão dos Conjuntos de Dados

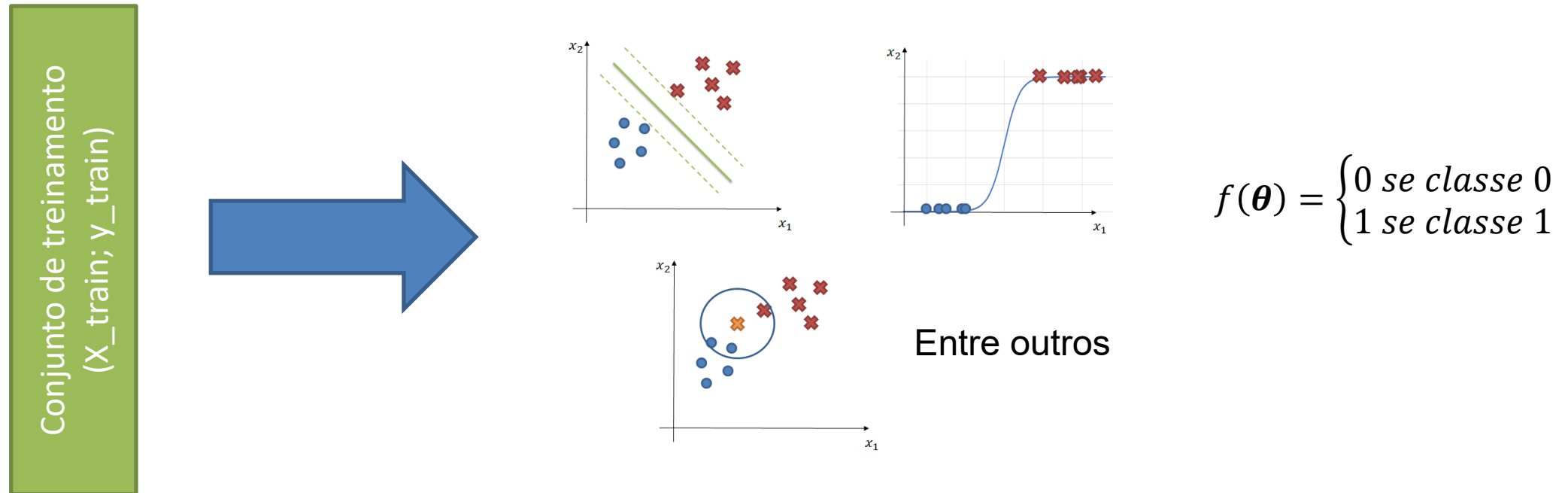
- **Conjunto de treinamento:** Utilizado para o modelo aprender as relações entre as entradas e saídas dos meus dados
- **Conjunto de teste:** Utilizado para verificar se o nosso modelo foi devidamente treinado e checamos com métricas de desempenho se o nosso





# Projeto de Aprendizado Supervisionado

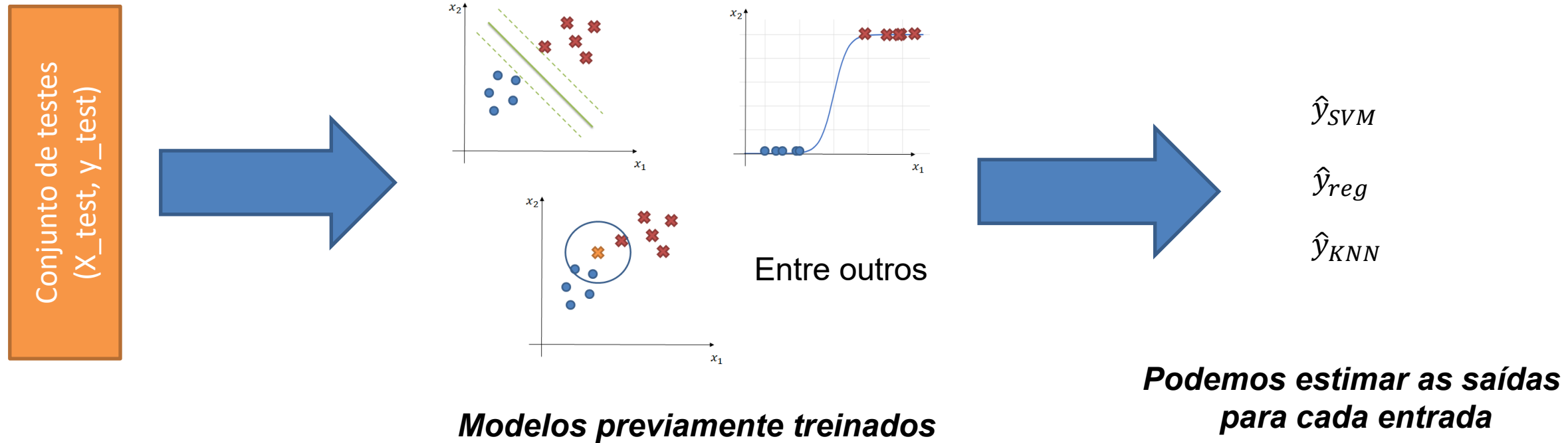
- Com a divisão dos dados podemos atuar da seguinte forma:



***Modelos treinados : Parâmetros  $\theta$  estimados para cada modelo!***

# Projeto de Aprendizado Supervisionado

- Com a divisão dos dados podemos atuar da seguinte forma:



# Projeto de Aprendizado Supervisionado



- Agora podemos comparar o que os modelos estavam ( $\hat{y}$ ) prevendo com o que eles deveriam estar prevendo ( $y_{test}$ )

$y_{test}$	$\hat{y}_{SVM}$	$\hat{y}_{reg}$	$\hat{y}_{KNN}$
Adulto	Criança	Adulto	Criança
Adulto	Adulto	Criança	Adulto
Criança	Criança	Criança	Adulto
Adulto	Adulto	Adulto	Criança
Criança	Criança	Adulto	Adulto
Adulto	Criança	Adulto	Adulto
Adulto	Criança	Adulto	Criança
⋮	⋮	⋮	⋮
Criança	Adulto	Criança	Adulto

Valores verdadeiros

Valores Estimados por diferentes modelos

# Projeto de Aprendizado Supervisionado



- Podemos comparar os modelos e tentar ver qual deles consegue chegar o mais próximo possível de  $y_{test}$ . Para isso, usamos as métricas de desempenho

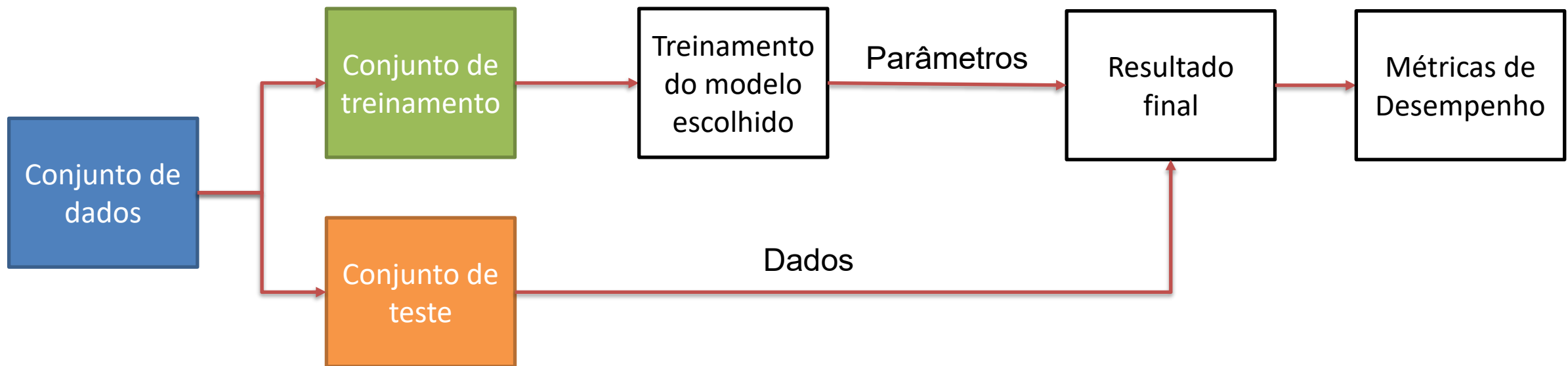
$y_{test}$	$\hat{y}_{SVM}$	$\hat{y}_{reg}$	$\hat{y}_{KNN}$
Adulto	Criança	Adulto	Criança
Adulto	Adulto	Criança	Adulto
Criança	Criança	Criança	Adulto
Adulto	Adulto	Adulto	Criança
Criança	Criança	Adulto	Adulto
Adulto	Criança	Adulto	Adulto
Adulto	Criança	Adulto	Criança
⋮	⋮	⋮	⋮
Criança	Adulto	Criança	Adulto

Valores verdadeiros

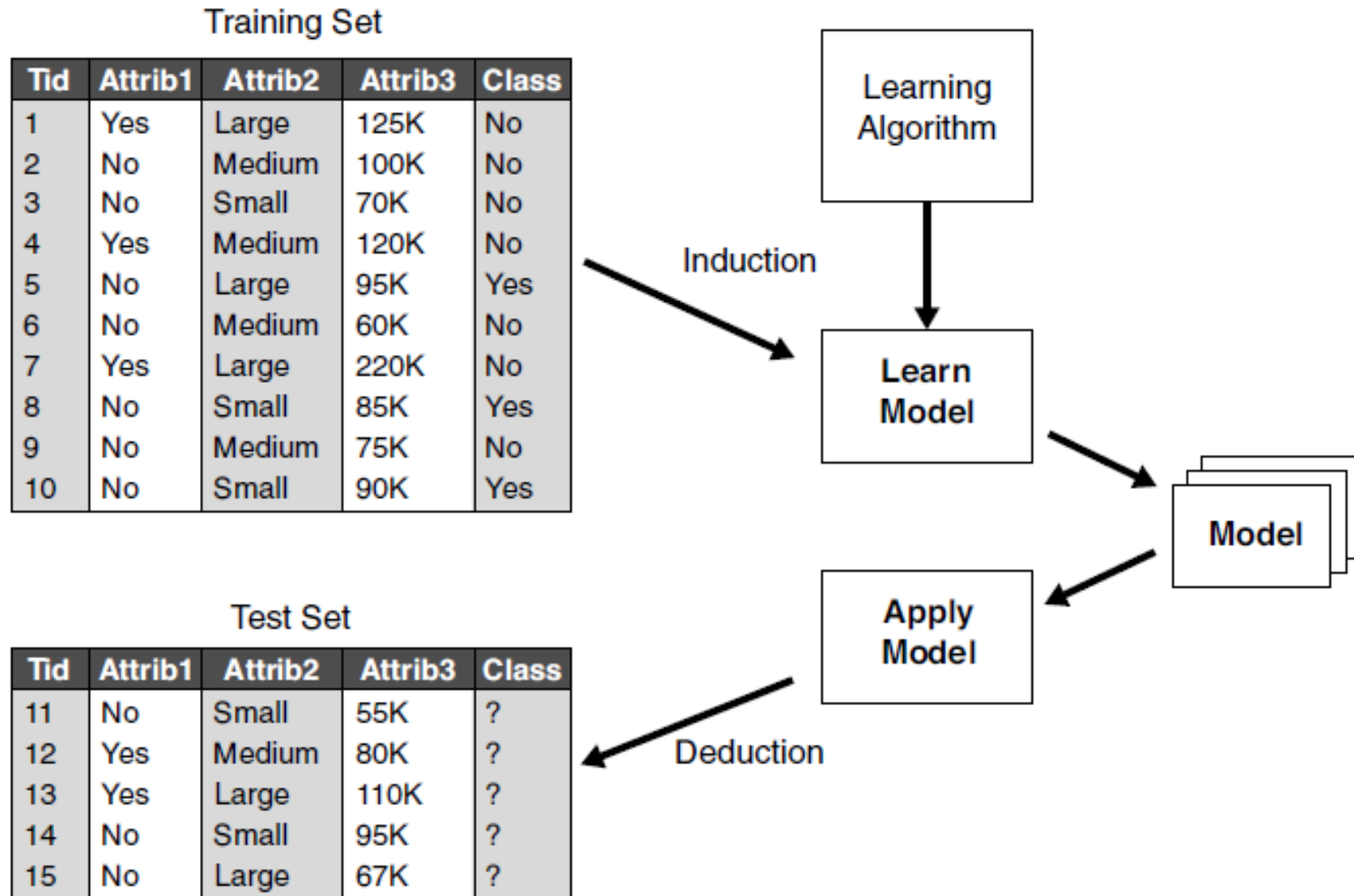
Valores Estimados por diferentes modelos

# Treinamento e Teste - Resumo

- Para usarmos modelos de Machine Learning temos que criar os conjuntos de dados para treinamento e teste



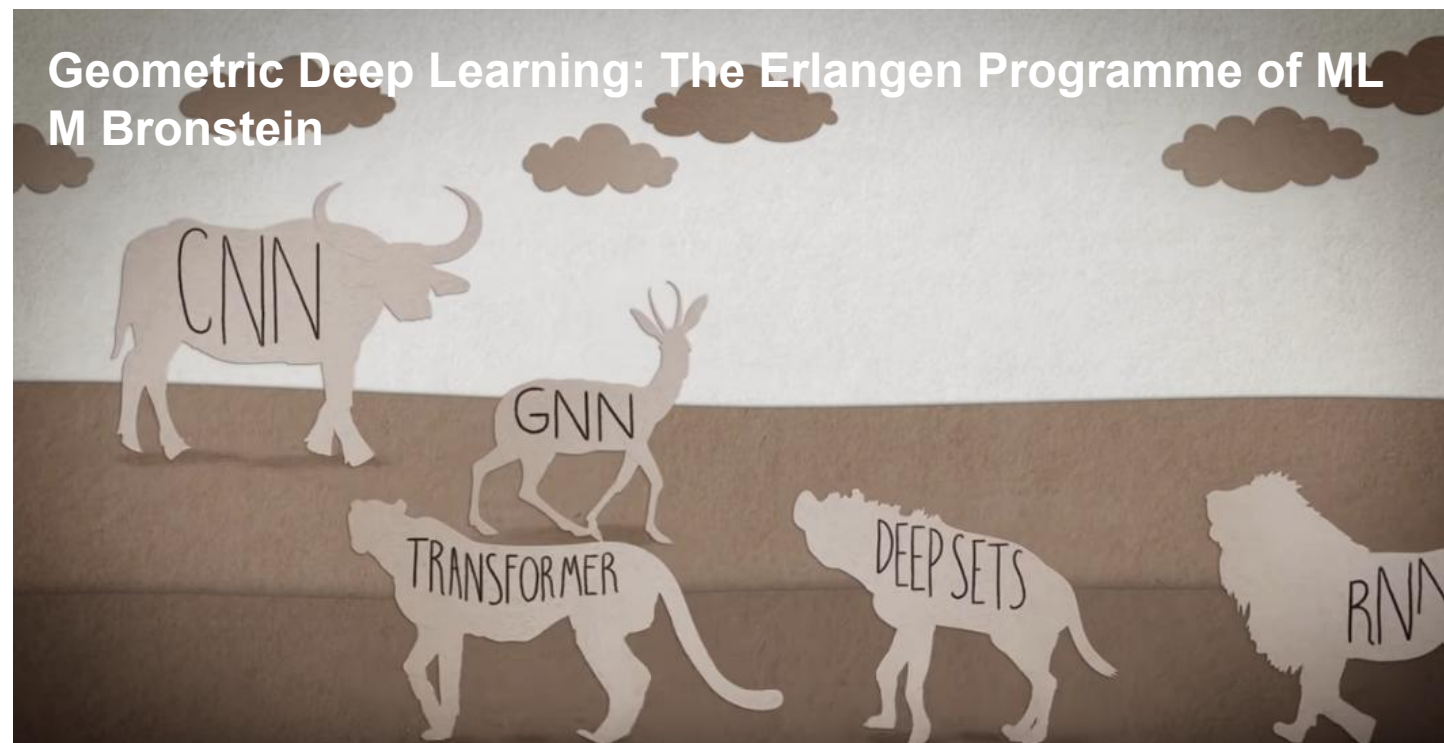
# Treinamento e Teste - Resumo



# Treinamento e Teste - Resumo

Como pudemos ver ao longo da aula, existem muitas formas de agrupar os diferentes algoritmos de Machine Learning. Muito desse processo de classificação ainda está sendo feito agora!

É realmente um Zoológico de Algoritmos!



# Treinamento e Teste - Resumo



- Infelizmente, o Zoológico é muito grande para nosso tour: não vamos conseguir conhecer todos os algoritmos que existem esse ano;
- Além disso, muitos outros algoritmos estão sendo propostos todos os meses!
- Vamos estudar alguns dos mais importantes para realizar tarefas básicas de Inteligência Artificial e Ciência de Dados, entre eles:
  - ❖ **Aprendizado supervisionado:**
    - Regressão: regressão linear, SVR (SVM), Árvore de Decisão e KNR;
    - Classificação: KNN, Árvore de Decisão, RandomForest, SVM, Naive Bayes, Regressão Lógica;
  - ❖ **Aprendizado não supervisionado:**
    - Agrupamento: k-means, hierárquico, DBSCAN, mistura gaussiana;
    - Redução de dimensionalidade: t-SNE, PCA, kPCA, Isomap;
- No segundo semestre iremos ver outra parte do zoológico que realiza essas mesmas tarefas de maneira diferente: as Redes Neurais Artificiais (Deep Learning).



# Avaliação do modelo



- Porque devemos avaliar o modelo?
- IA PODE ERRAR!

# Erros acontecem... overfitting e underfitting

## Overfitting

- Modelo que se ajusta aos dados de treinamento muito bem, incluindo outliers
- Impacto negativo na capacidade do modelo em generalizar

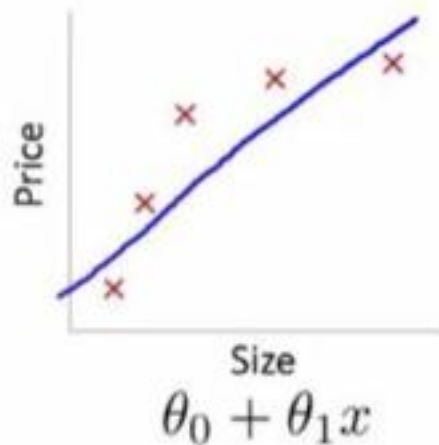
## Underfitting

- Um modelos que nem se ajusta bem aos dados de treino, nem generaliza para novos dados

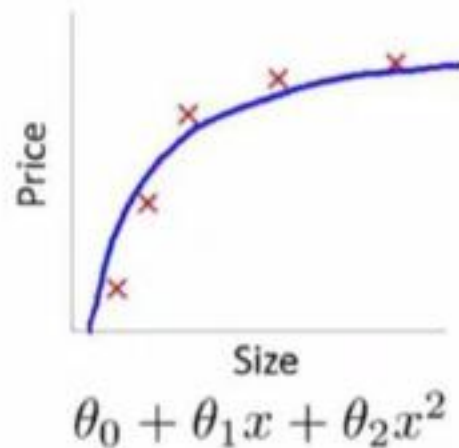
# Erros acontecem... overfitting e underfitting

Um modelo com **overfitting** tem mais coeficientes do que o necessário. É um modelo com **pouca capacidade de generalização**: ele terá alta acurácia para os dados de treinamento e acurácia extremamente baixa para os dados de teste.

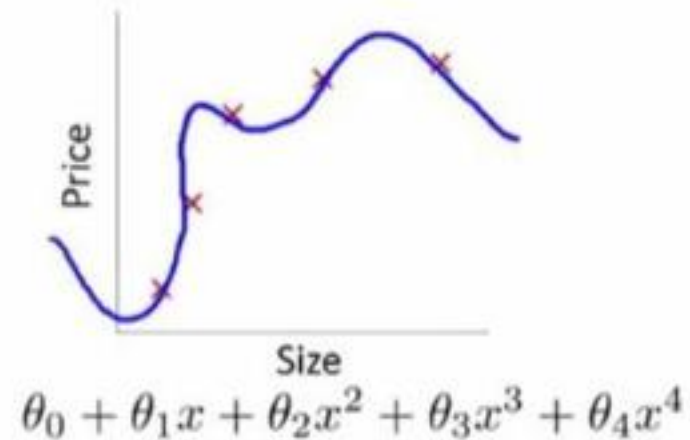
•



Viés alto  
(subajuste)



Ajuste de boa  
qualidade



Variância alta  
(superajuste)

# Erros acontecem...



**Bias (enviesamento):** Precisamos ser éticos na escolha das colunas que iremos usar e no dados que iremos fornecer aos algoritmos para que injustiças e preconceitos prévios não sejam ensinados aos algoritmos!

**Overffiting (sobreajuste):** Precisamos fazer separação treino/teste para atestar a generalidade de nosso modelo, levando em consideração o tipo de dados e o propósito para escolher tipo de metodologia de separação (80/20, cross validation, data leakage);

**Acurácia e Precisão:** Precisamos comparar com resultados de sistemas tradicionais (normalmente denominados de Modelo Base);

Prof. Miguel Bozer da Silva

# **NORMALIZAÇÃO E PADRONIZAÇÃO DOS DADOS**

# Normalização e Padronização dos Dados

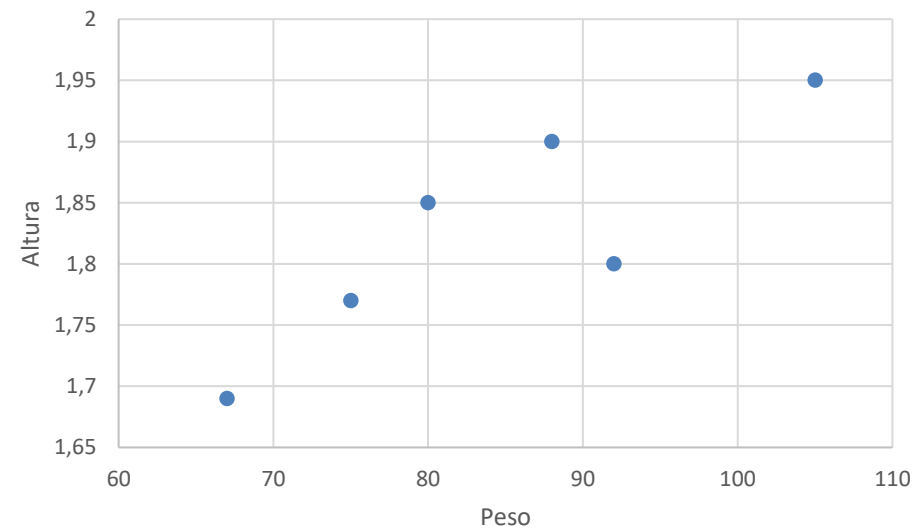


- Alguns modelos de Machine Learning exigem que os valores estejam em escalas similares para que eles não se tornem tendenciosos. Por exemplo:
- Se temos o peso, altura e o tamanho da camisa que uma pessoa usa. Podemos tentar usar esses dados para estimar qual o tipo de camisa uma pessoa pode comprar
- Para isso, o nosso modelo recebe os valores do peso e da altura e estima a saída de tamanho da camisa.

# Normalização e Padronização dos Dados



id	Peso	Altura(m)	Camisa
1	75	1,77	G
2	80	1,85	G
3	92	1,8	G
4	67	1,69	M
5	88	1,9	GG
6	105	1,95	GG



A escala do peso é muito maior que a altura.

# Normalização e Padronização dos Dados



- Caso as grandezas dos dados envolvidos forem muito diferentes, podemos padronizar ou normalizar os nossos dados:

- Padronização:

$$z_i = \frac{x_i - \mu}{\sigma}$$

Onde:

$z_i$  é o i-ésimo valor padronizado;

$x_i$  é o i-ésimo valor original dos nossos dados

$\sigma$  é o desvio padrão dos dados.

$\mu$  é a média dos dados

Sklearn: `StandardScaler()`

- Normalização:  $X_{changed} = \frac{X - X_{min}}{X_{max} - X_{min}}$

Onde:

$X_{changed}$  é o valor normalizado

$X$  é o valor antes da normalização

$X_{min}$  é o menor valor do conjunto de dados

$X_{max}$  é o maior valor do de dados

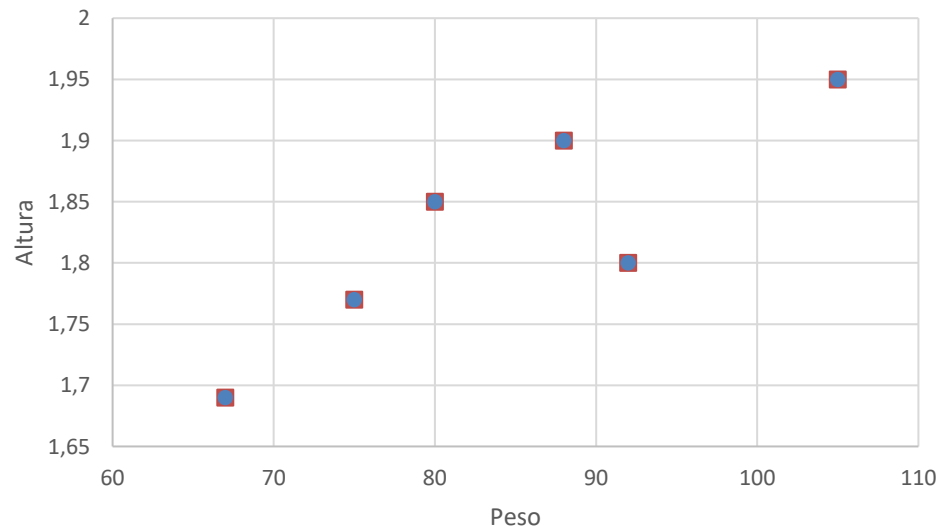
Sklearn: `MinMaxScaler()`



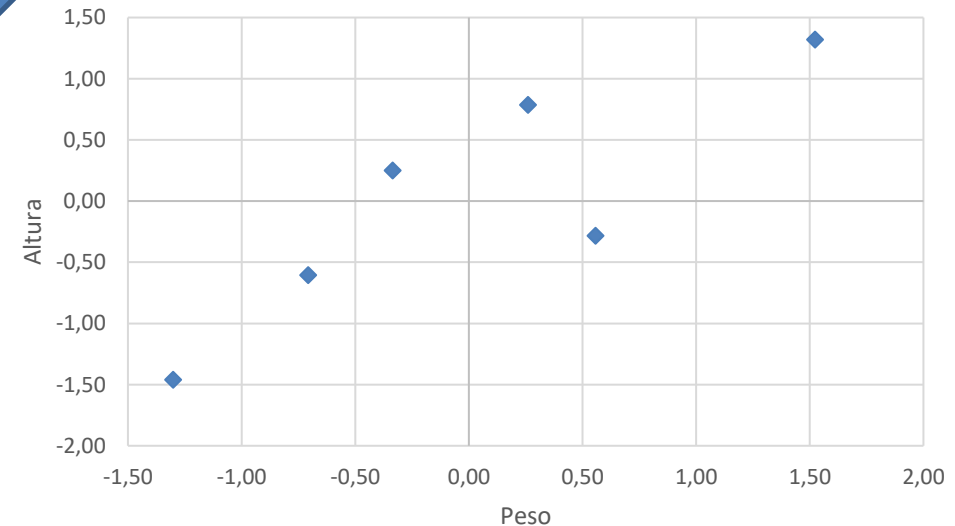
# Normalização e Padronização dos Dados

- Após a padronização dos dados:

id	Peso	Altura(m)	Camisa
1	75	1,77	G
2	80	1,85	G
3	92	1,8	G
4	67	1,69	M
5	88	1,9	GG
6	105	1,95	GG

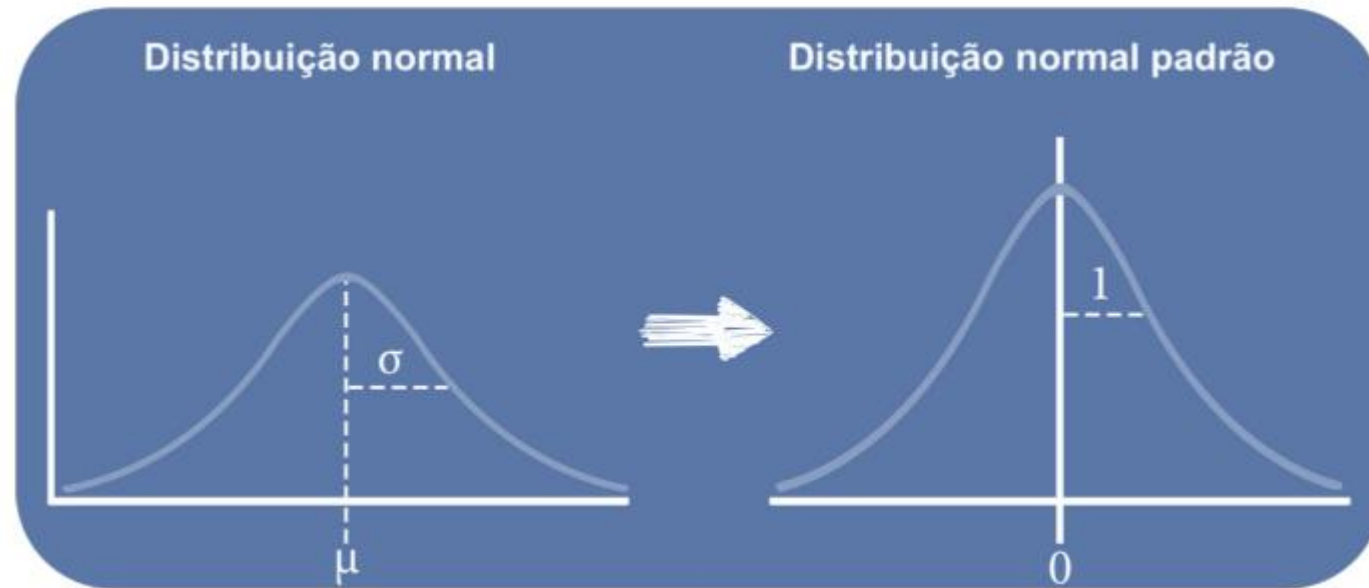


id	Peso	Altura(m)	Camisa
1	-0,71	-0,61	G
2	-0,33	0,25	G
3	0,56	-0,29	G
4	-1,30	-1,46	M
5	0,26	0,78	GG
6	1,52	1,32	GG



# Normalização e Padronização dos Dados

- A Padronização dos dados transforma os mesmos em uma distribuição normal padrão

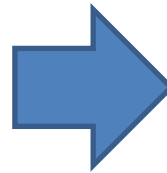
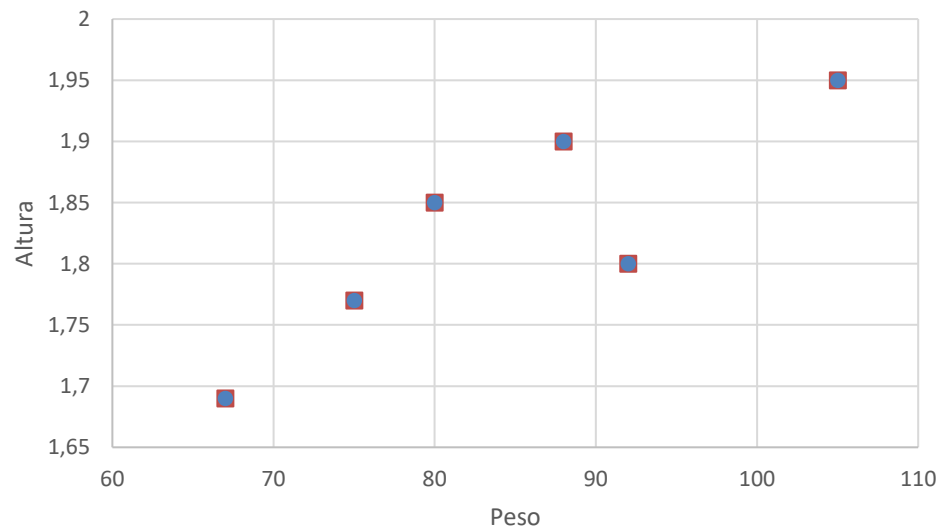


- Recomendado quando os dados **estão em uma distribuição normal**

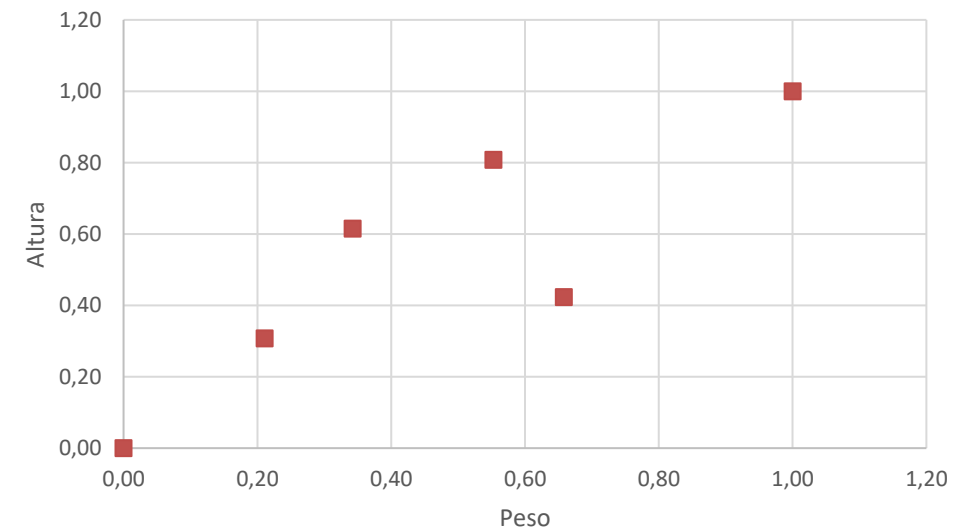
# Normalização e Padronização dos Dados

- Após a normalização dos dados:

id	Peso	Altura(m)	Camisa
1	75	1,77	G
2	80	1,85	G
3	92	1,8	G
4	67	1,69	M
5	88	1,9	GG
6	105	1,95	GG



id	Peso	Altura(m)	Camisa
1	0,21	0,31	G
2	0,34	0,62	G
3	0,66	0,42	G
4	0,00	0,00	M
5	0,55	0,81	GG
6	1,00	1,00	GG



# Normalização e Padronização dos Dados



- A normalização pode ser aplicada quando a distribuição dos dados não é normal ou se o desvio padrão dos mesmos for muito pequeno.

## Exercício



- No jupyter Notebook: “2.3 Normalização e Padronização dos Dados.ipynb” vamos ver como aplicar esses conceitos.