

# Aprendizagem de Máquina

a.k.a Machine Learning

# Objetivo e Expectativa

Meu nome é ...

Eu trabalho ...

Meu objetivo com o curso é ...

Minha expectativa com a matéria é ...

# Nasser Boan

- Cientista de Dador Sr. - Falconi
- IBM Data Science Professional Certified
- Especialista em Inteligência Artificial 🤖
- + 50 projetos de Ciência de Dados
- 3 anos de experiência com Gerência de Projetos de Data Science
- Terrabolista 🌍
- Vacinado 💉



**mesttra**  
discover. shape. connect.

# Ementa

- **Conceitos básicos**

- Como uma máquina aprende
- Paradigmas de Aprendizagem
  - Aprendizagem Supervisionada
  - Aprendizagem Não Supervisionada
- Métodos de Validação
- Underfitting e Overfitting

- **Classificação**

- Métricas de Classificação
- Algoritmos
- Bases Desbalanceadas

- **Regressão**

- Métricas de Regressão
- Algoritmos
- Séries Temporais

- **Ensemble**

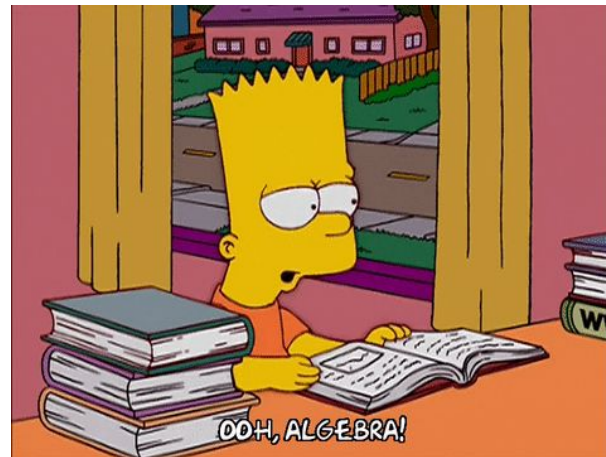
- RandomForest
- Bagging e Boosting
- Máquinas de Comitê
- Ensemble Regressors

- **Seleção e Tuning de modelos**

- Validação Cruzada
- GridSearchCV

# Avaliação

- Trabalhos
  - Trabalho 001 - Classificação
  - Trabalho 002 - Regressão
  - Trabalho 003 - Clusterização \*



\* Se der tempo

# Agenda

	04/10	05/10	06/10	07/10	08/10	09/10	11/10	12/10	13/10	14/10	15/10	16/10	18/10
Aula	3H	3H	3H	3H	3H	3H	3H	3H	3H	3H	3H	4H	-
Trab							Trab. 01						Trab. 02



# Ferramentas

## instalação do sklearn

[Installing scikit-learn — scikit-learn 0.24.2 documentation](#)

## instalação do imblearn

<https://imbalanced-learn.org/stable/>

## instalação do jupyter notebook

[Project Jupyter | Installing the Jupyter Software](#)

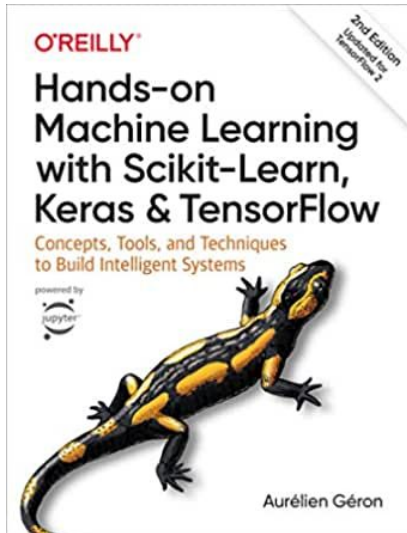
## google colab

[Welcome To Colaboratory - Colaboratory](#)

## tutorial rápido de git

[git - guia prático - sem complicação!](#)

# Referências



## youtube

[StatQuest with Josh Starmer](#)

[3Blue1Brown](#)

[PyData](#)

## Documentações

[User guide: contents — scikit-learn 0.24.2 documentation](#)

[imbalanced-learn documentation — Version 0.8.0](#)

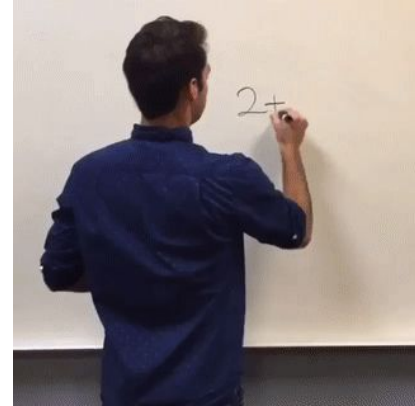
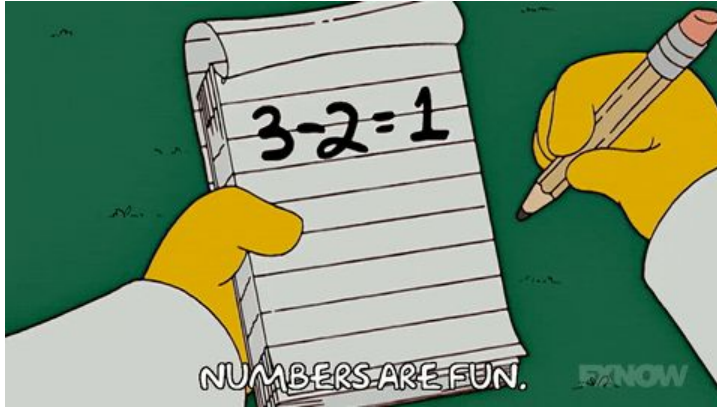


# Nosso repositório

<https://github.com/nasserboan/corujas-cultas>



# Expectativa vs realidad

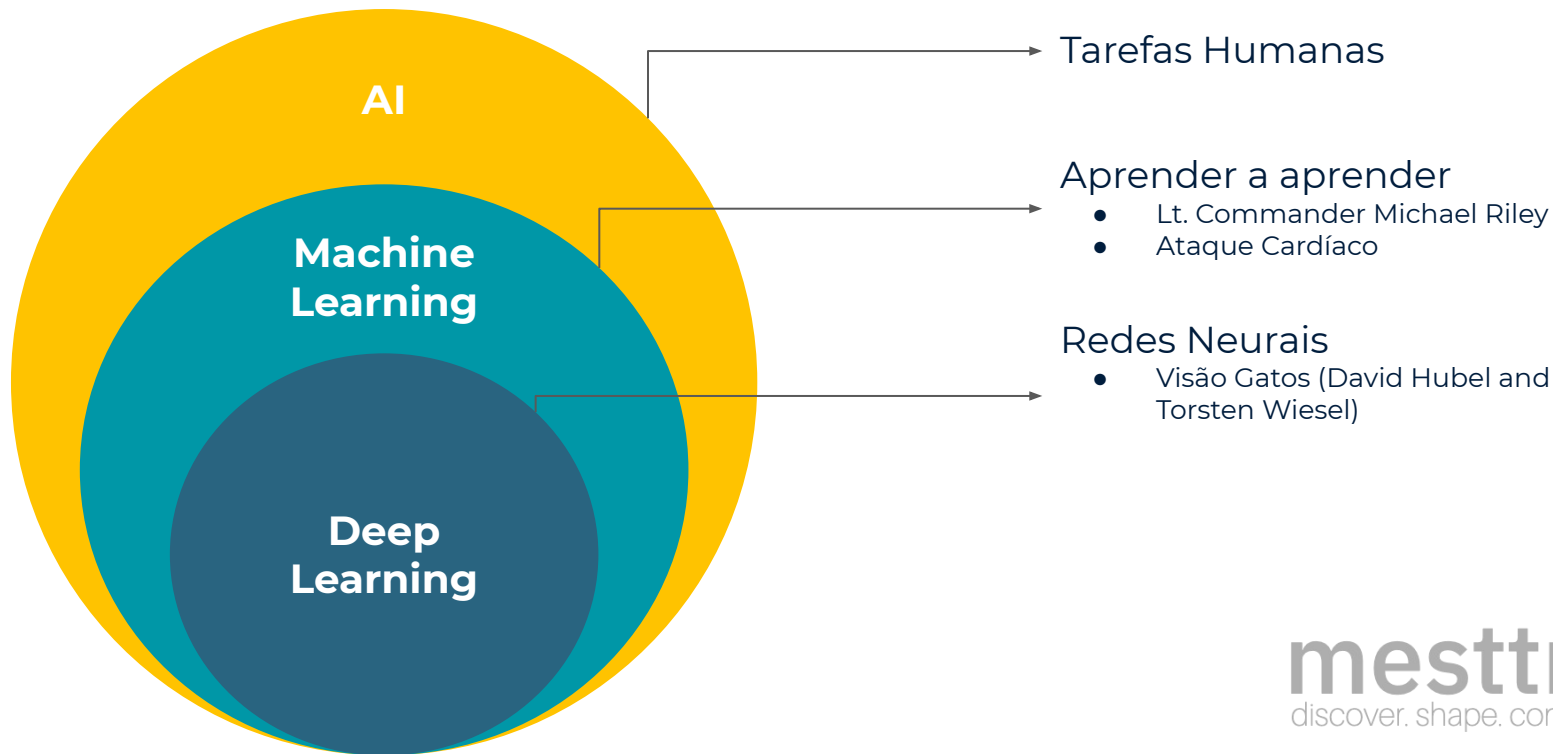




# Conceitos Básicos

**mesttra**  
discover. shape. connect.

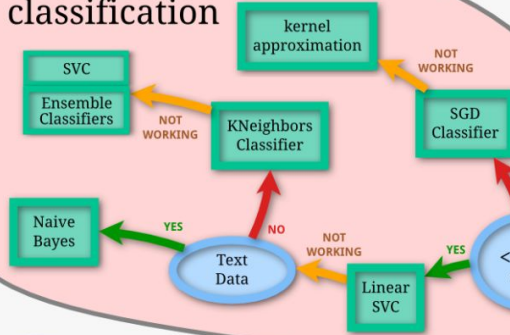
# Machine Learning



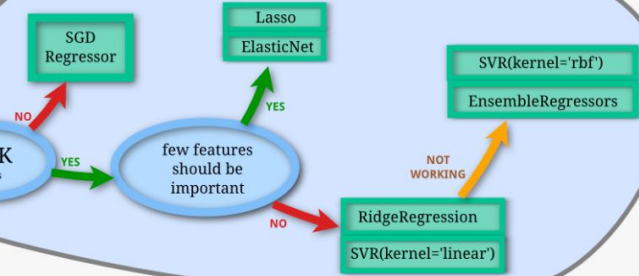
# scikit-learn algorithm cheat-sheet

START

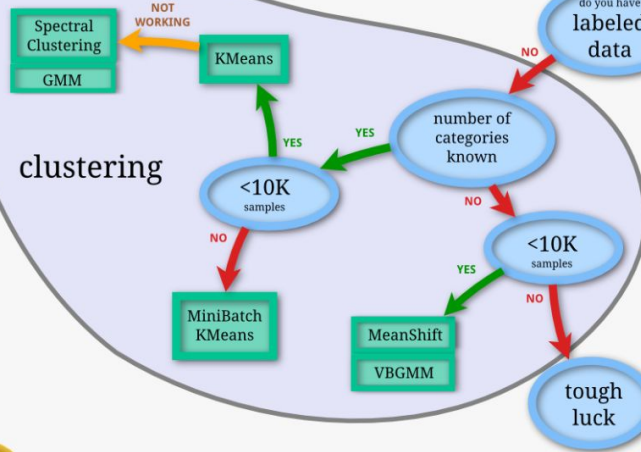
## classification



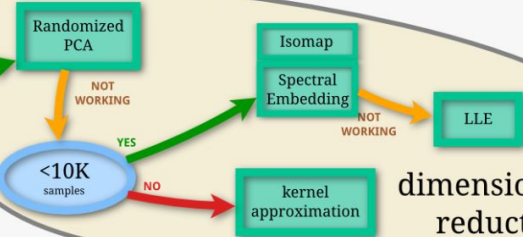
## regression



## clustering



## dimensionality reduction

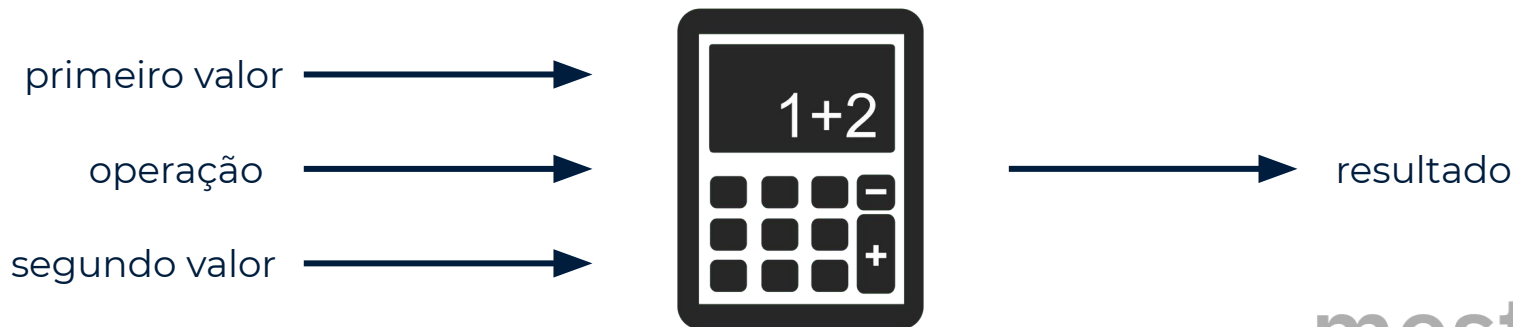
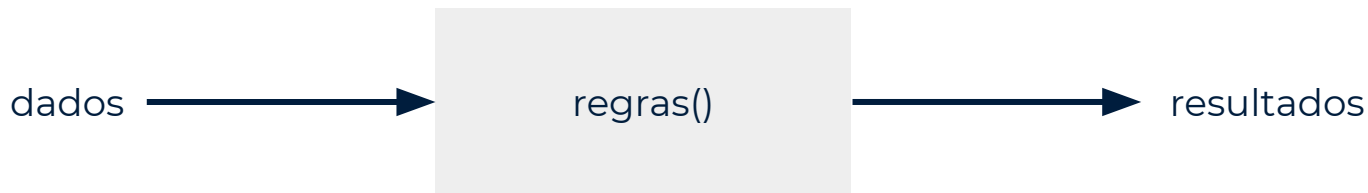


Back

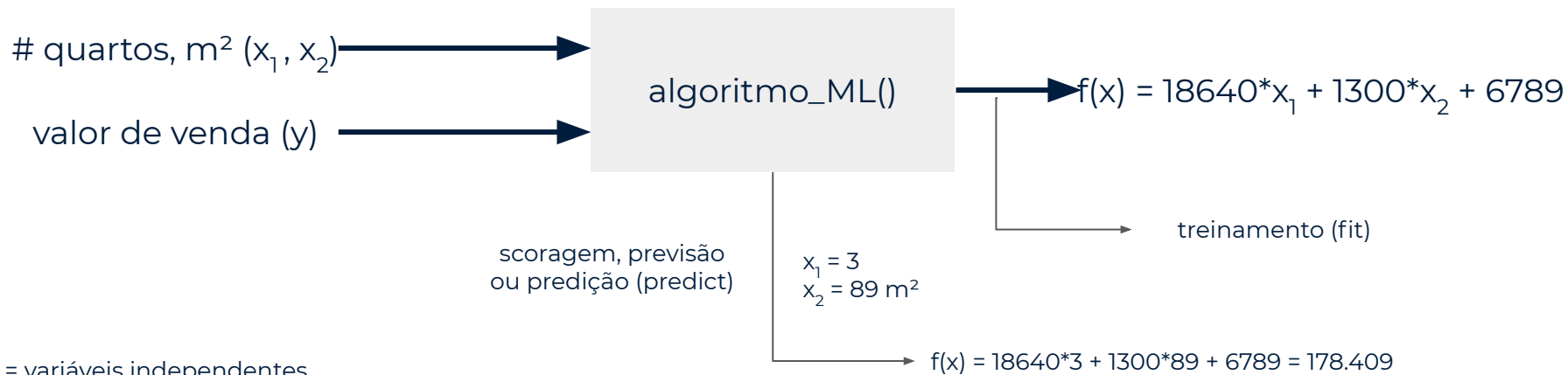
# Machine Learning



# Machine Learning



# Machine Learning



$x$  = variáveis independentes  
 $y$  = variável dependente



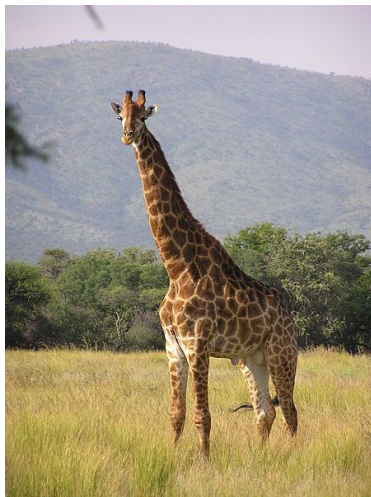
# Paradigmas de aprendizagem

O cavalo é um animal áceros?



# Como uma máquina aprende?

Não Áceros



Áceros



**mesttra**  
discover. shape. connect.

# Paradigmas de aprendizagem



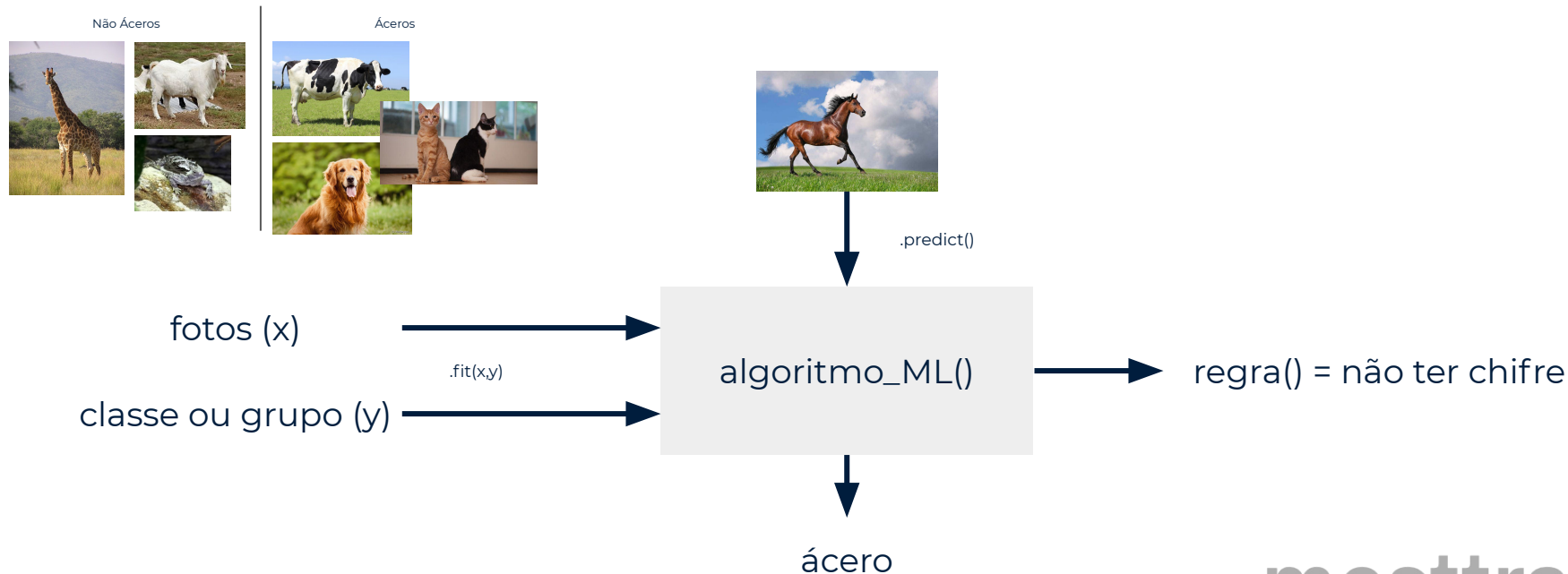
- ☐ Ácero
- ☐ Não Ácero

# Paradigmas de aprendizagem



- ☒ Ácero
- ☐ Não Ácero

# Paradigmas de aprendizagem

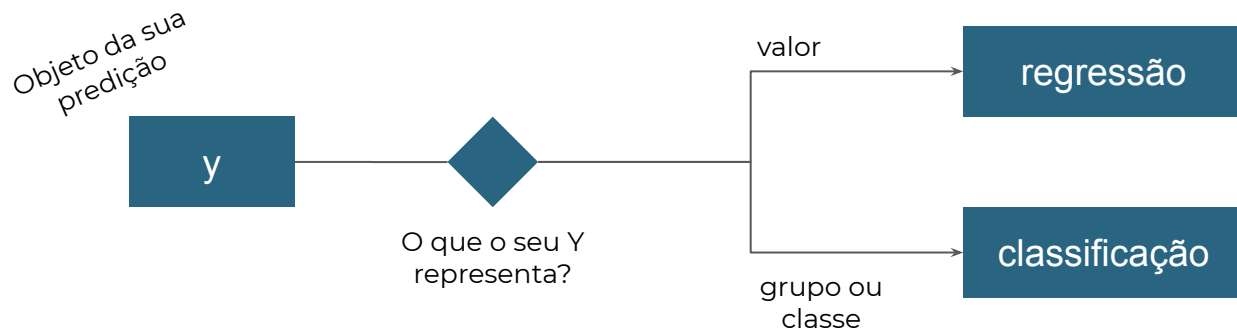


Para cada X (foto) eu tenho um Y (classe) associada. Nós já temos as respostas, só precisamos das regras.

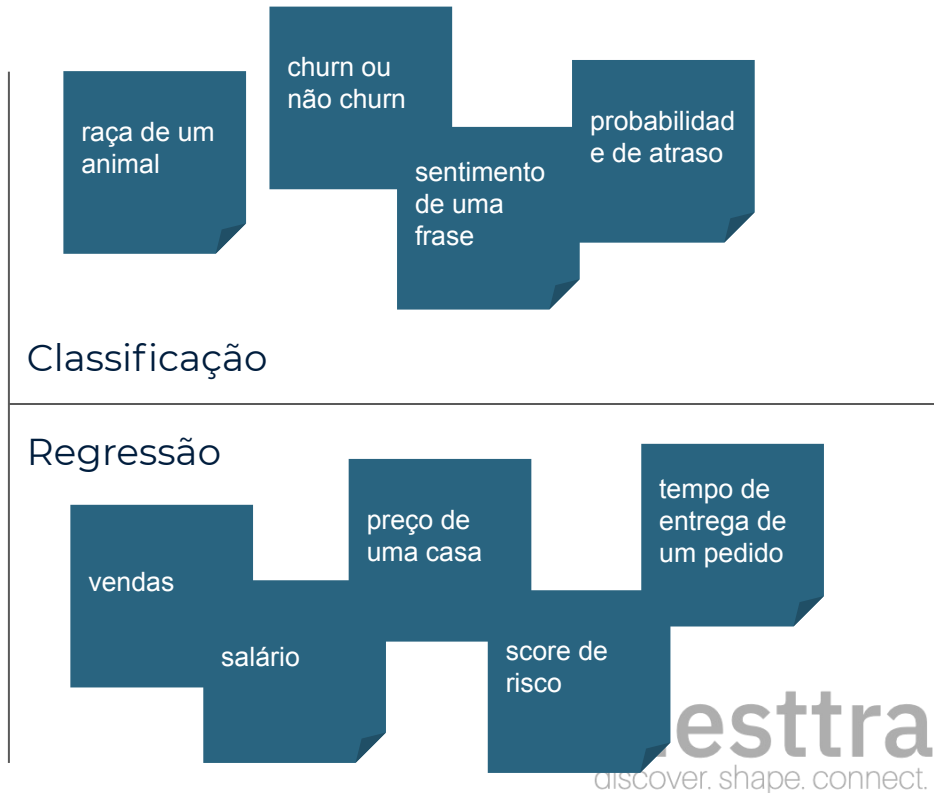
# Aprendizagem Supervisionada

Temos um problema de aprendizagem supervisionada quando temos vários exemplos de resposta ( $y$ ) para entregar ao nosso modelo.

A aprendizagem supervisionada possui 2 tarefas bem comuns:



# Aprendizagem Supervisionada



# Aprendizagem Supervisionada

LogisticRegression()

SVC()

KNeighborsClassifier()

DecisionTreeClassifier()

RandomForestClassifier()

Classificação

---

Regressão

LinearRegression()

SVR()

KNeighborsRegressor()

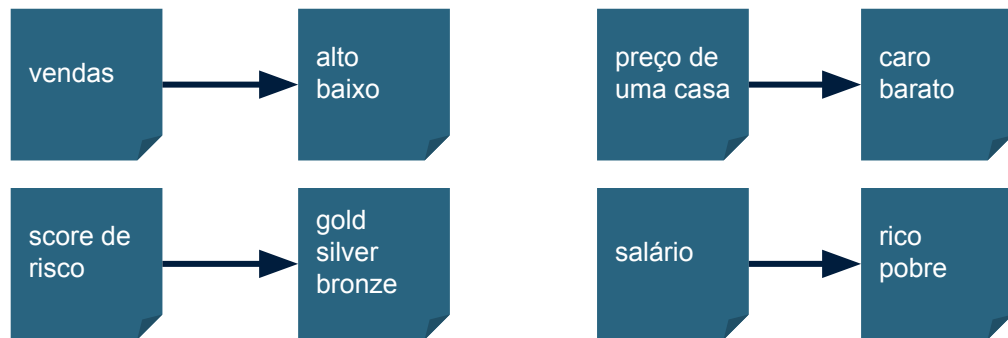
DecisionTreeRegressor()

RandomForestRegressor()



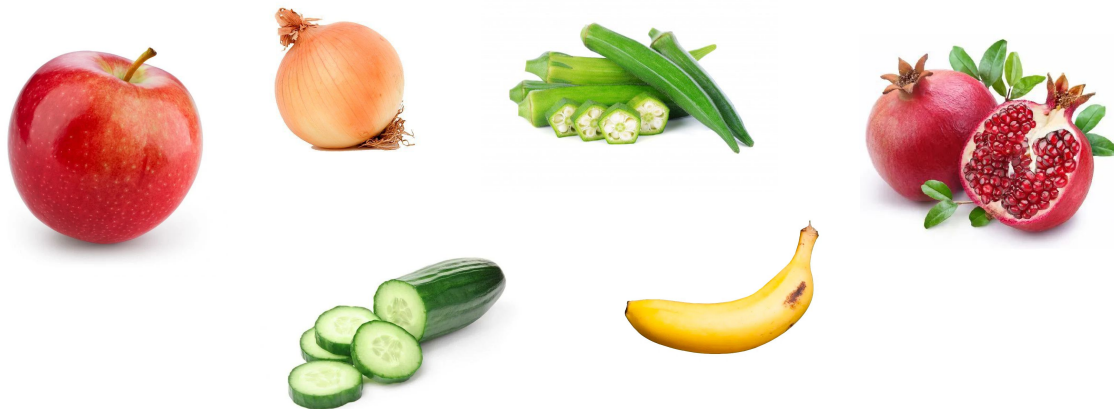
# Aprendizagem Supervisionada

É possível transformar um trabalho de regressão em um trabalho de classificação. Basta mudarmos o nosso Y no momento de treinamento, a técnica mais comum para isso é o binning (criação de grupos).



# Paradigmas de aprendizagem

Dividir em dois grupos

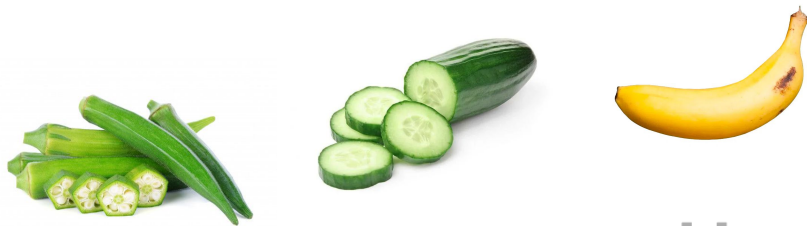


# Paradigmas de aprendizagem

Dividir em dois grupos

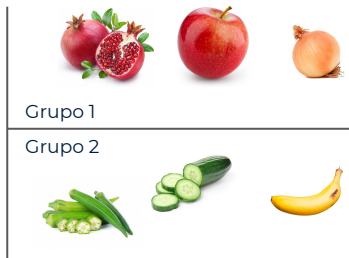


Grupo 1



Grupo 2

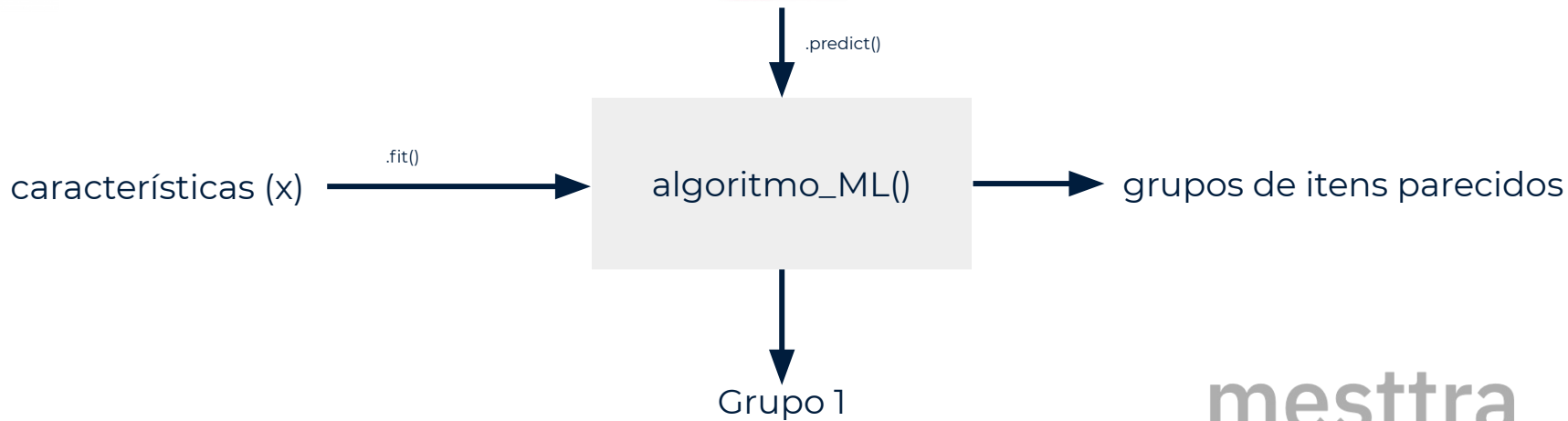
# Paradigmas de aprendizagem



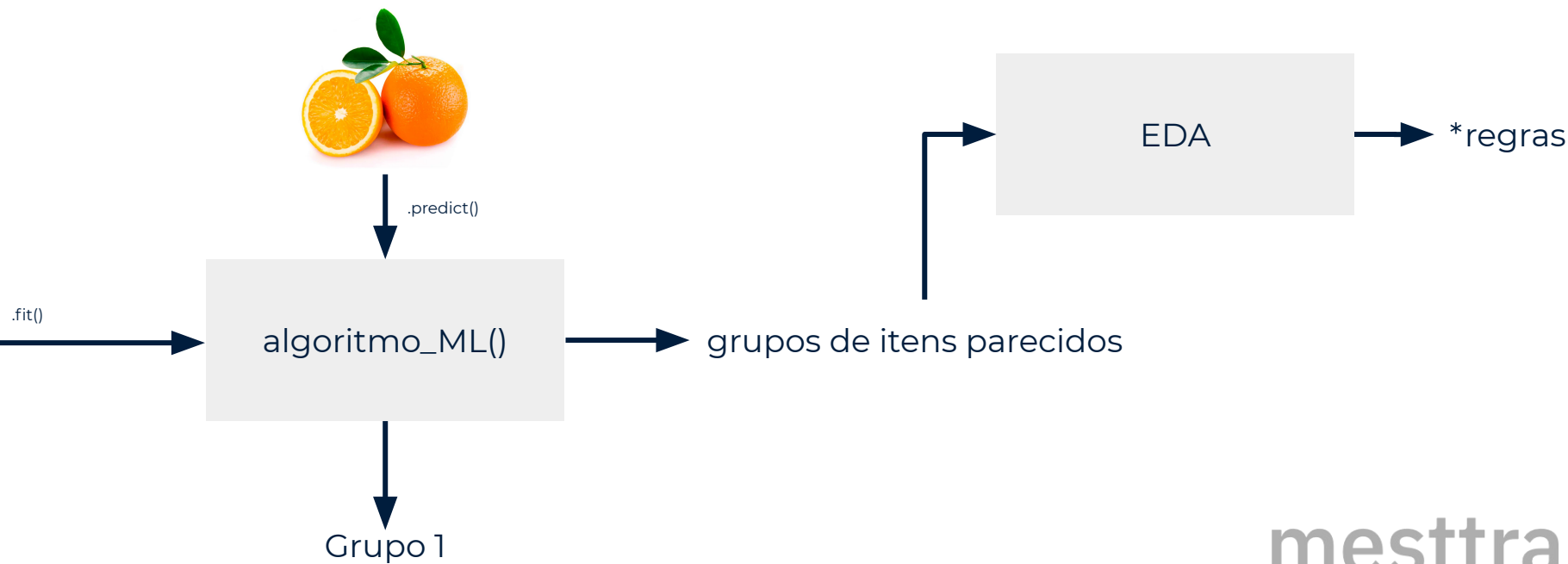
características (x) → algoritmo\_ML() → grupos de itens parecidos

Não existe um valor Y associado a cada X, procuramos então, grupos parecidos. Ou seja, modelamos os padrões e não as respostas.

# Paradigmas de aprendizagem



# Paradigmas de aprendizagem

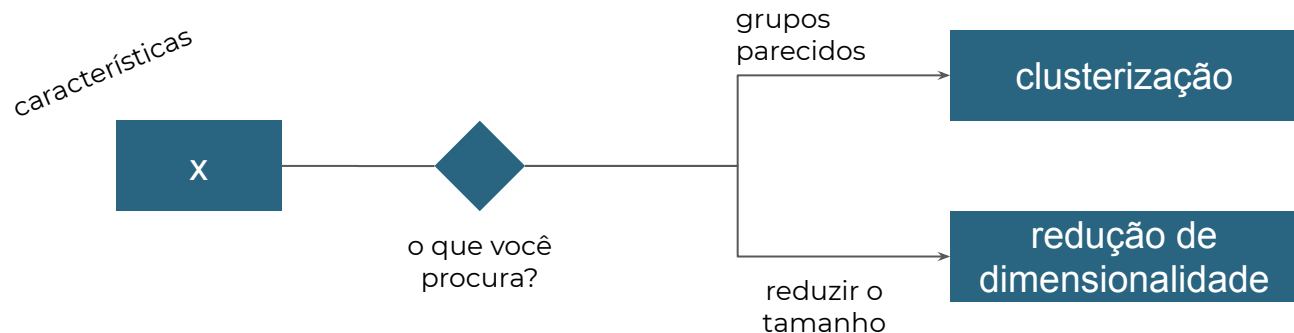


\* supostas

# Aprendizagem Não Supervisionada

Temos um problema de aprendizagem não supervisionada quando não temos as respostas para treinamento de antemão.

A aprendizagem não supervisionada possui 2 tarefas bem comuns:



# Aprendizagem Não Supervisionada





# Aprendizagem Não Supervisionada

KMeans()

DBSCAN()

MiniBatchKMeans()

AgglomerativeClustering()

Clusterização

---

Redução de  
Dimensionalidade

PCA()

TSNE()

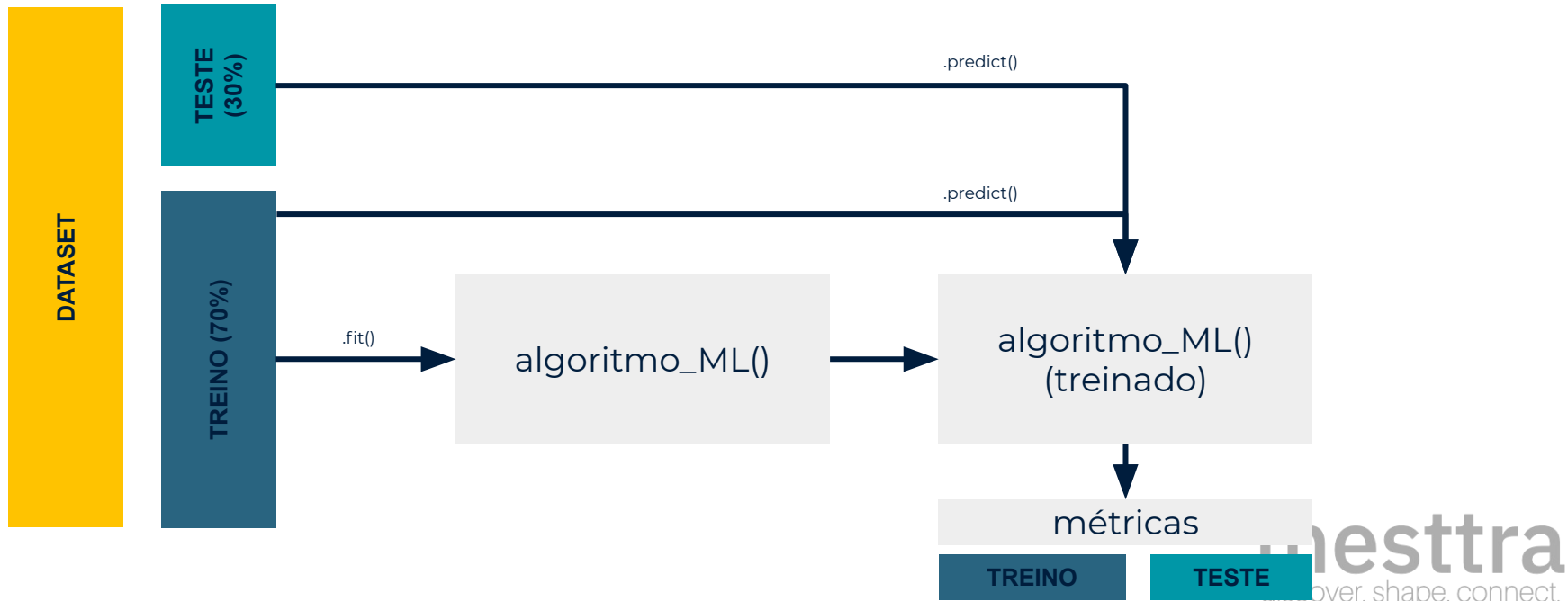
IncrementalPCA()

# Como uma máquina aprende?

Exemplos no Notebook

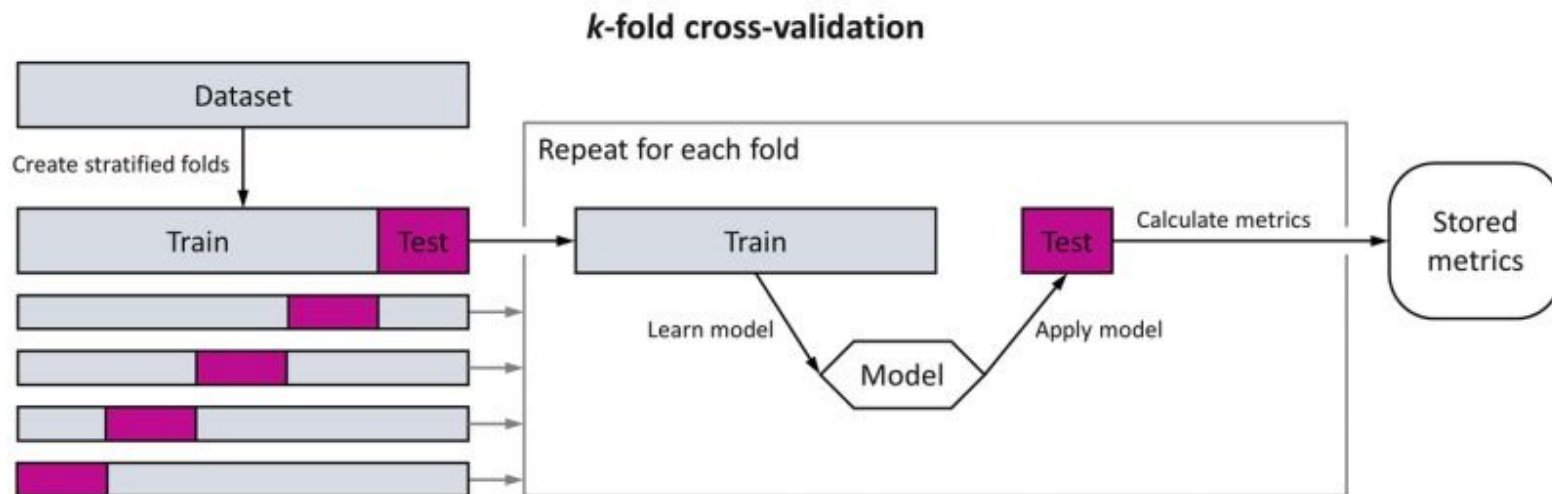
# Métodos de Validação

Hold-out

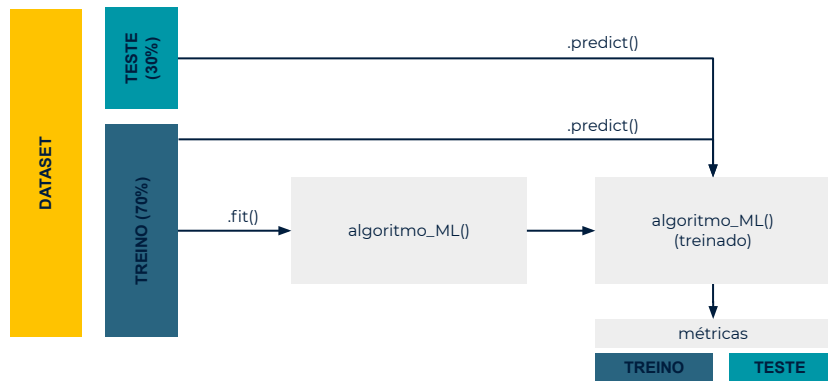


# Métodos de Validação

## KFold



# Underfitting e Overfitting



Underfitting

TREINO ↑

TESTE ↑

Good Fit

TREINO ↓

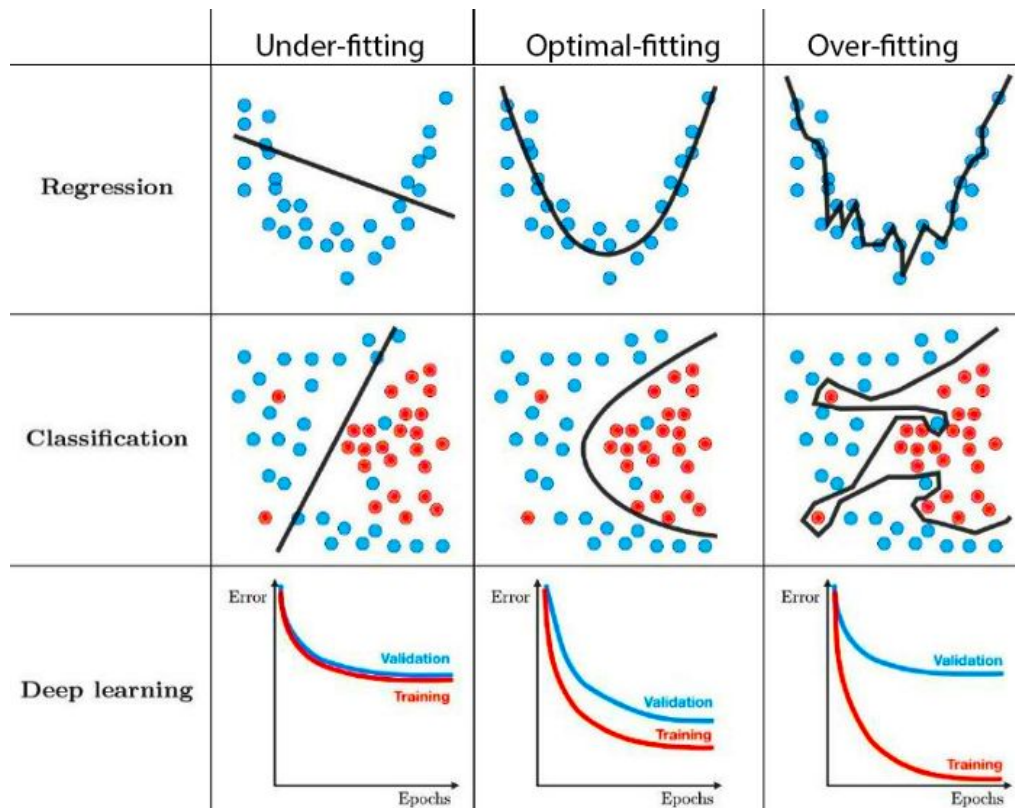
TESTE ↓

Overfitting

TREINO ↓

TESTE ↑

# Underfitting e Overfitting



Jä •

Nej

Vet ej

**Classificação**

**mesttra**  
discover. shape. connect.

# Tipos de Classificação

- Classificação binária
- Classificação multiclasse
- Classificação multilabel (ou multitarget)
- Classificação desbalanceada



# Tipos de Classificação

- **Classificação binária**
- Classificação multiclasse
- Classificação multilabel (ou multitarget)
- Classificação desbalanceada



# Tipos de Classificação

- Classificação binária
- **Classificação multiclasse**
- Classificação multilabel (ou multitarget)
- Classificação desbalanceada



# Tipos de Classificação

- Classificação binária
- Classificação multiclasse
- **Classificação multilabel (ou multitarget)**
- Classificação desbalanceada



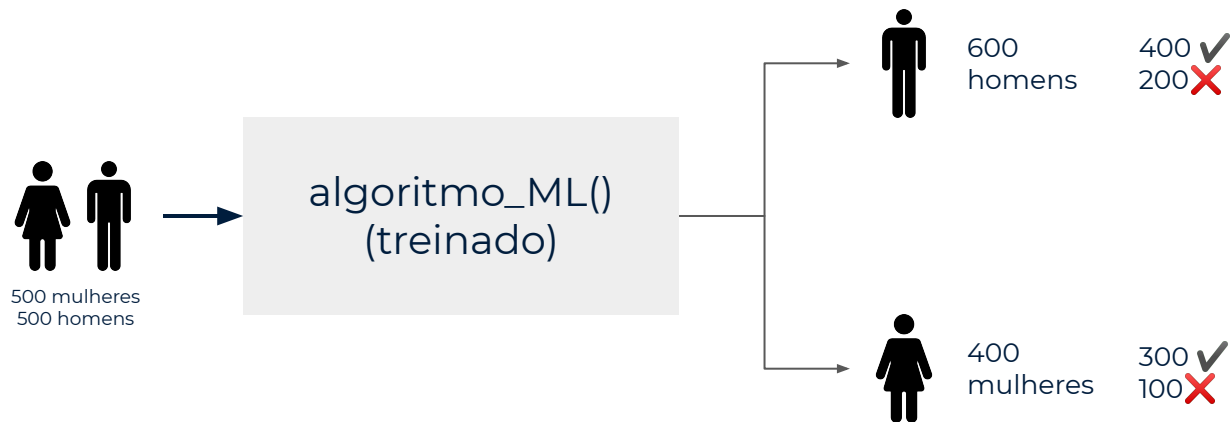
# Tipos de Classificação

- Classificação binária
- Classificação multiclasse
- Classificação multilabel (ou multitarget)
- **Classificação desbalanceada**



# Métricas de Classificação

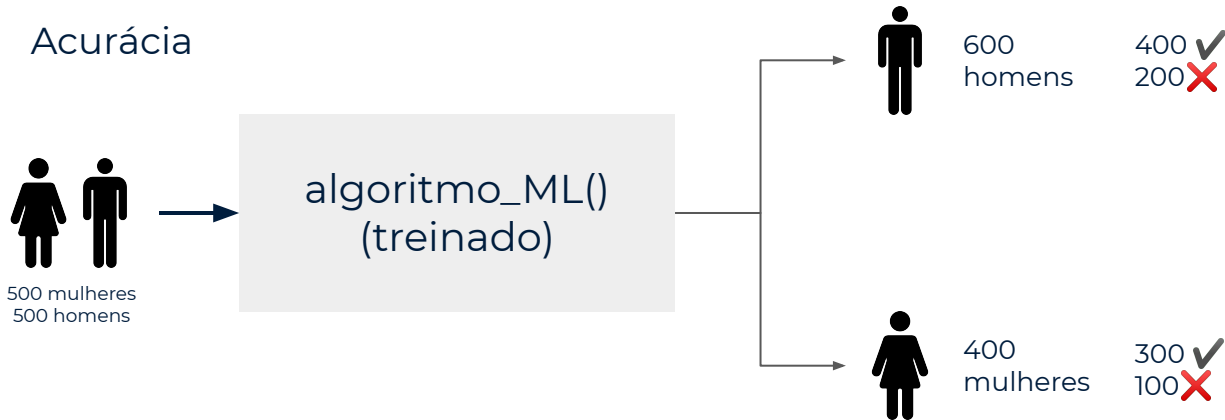
- Acurácia



$$\text{acc} = (400 + 300) / 1000 = 70\%$$

# Métricas de Classificação

- Acurácia



$$\text{acc} = (400 + 300) / 1000 = 70\%$$

	M_true	H_true
M_pred	300	100
H_pred	200	400

# Métricas de Classificação

- Matriz de Confusão

	M_true	H_true
M_pred	300	100
H_pred	200	400

$$\text{acc} = (400 + 300) / 1000 = 70\%$$

# Métricas de Classificação

- Precisão (“De tudo que eu previ, quanto eu acertei?”)

	M_true	H_true
M_pred	300	100
H_pred	200	400

$$\text{acc} = (400 + 300) / 1000 = 70\%$$

$$\text{precisão (M)} = 300 / (300+100) = 75\%$$

$$\text{precisão (H)} = 400 / (200+400) = 66\%$$



# Métricas de Classificação

- Revocação (“De tudo que existia, quanto eu acertei?”)

	cov+_true	cov-_true
cov+_pred	300	100
cov-_pred	200	400

$$\text{acc} = (400 + 300) / 1000 = 70\%$$

$$\text{precisão (M)} = 300 / (300+100) = 75\%$$

$$\text{precisão (H)} = 400 / (200+400) = 66\%$$

$$\text{revocação (M)} = 300 / (300+200) = 60\%$$

$$\text{revocação (H)} = 400 / (100+400) = 80\%$$

# Métricas de Classificação

	M_true	H_true
M_pred	300	100
H_pred	200	400
	recall	recall

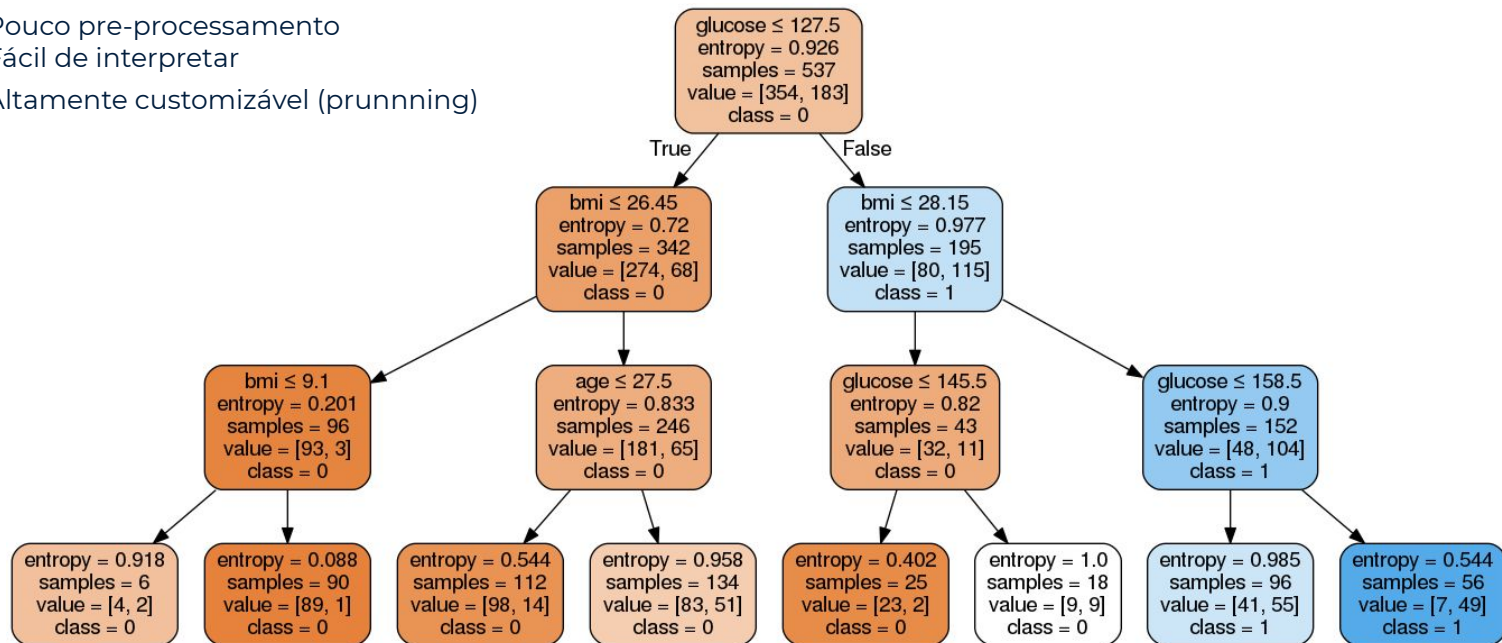
# Métricas de Classificação

	M_true	H_true	
M_pred	300	100	precision
H_pred	200	400	precision

# Algoritmos

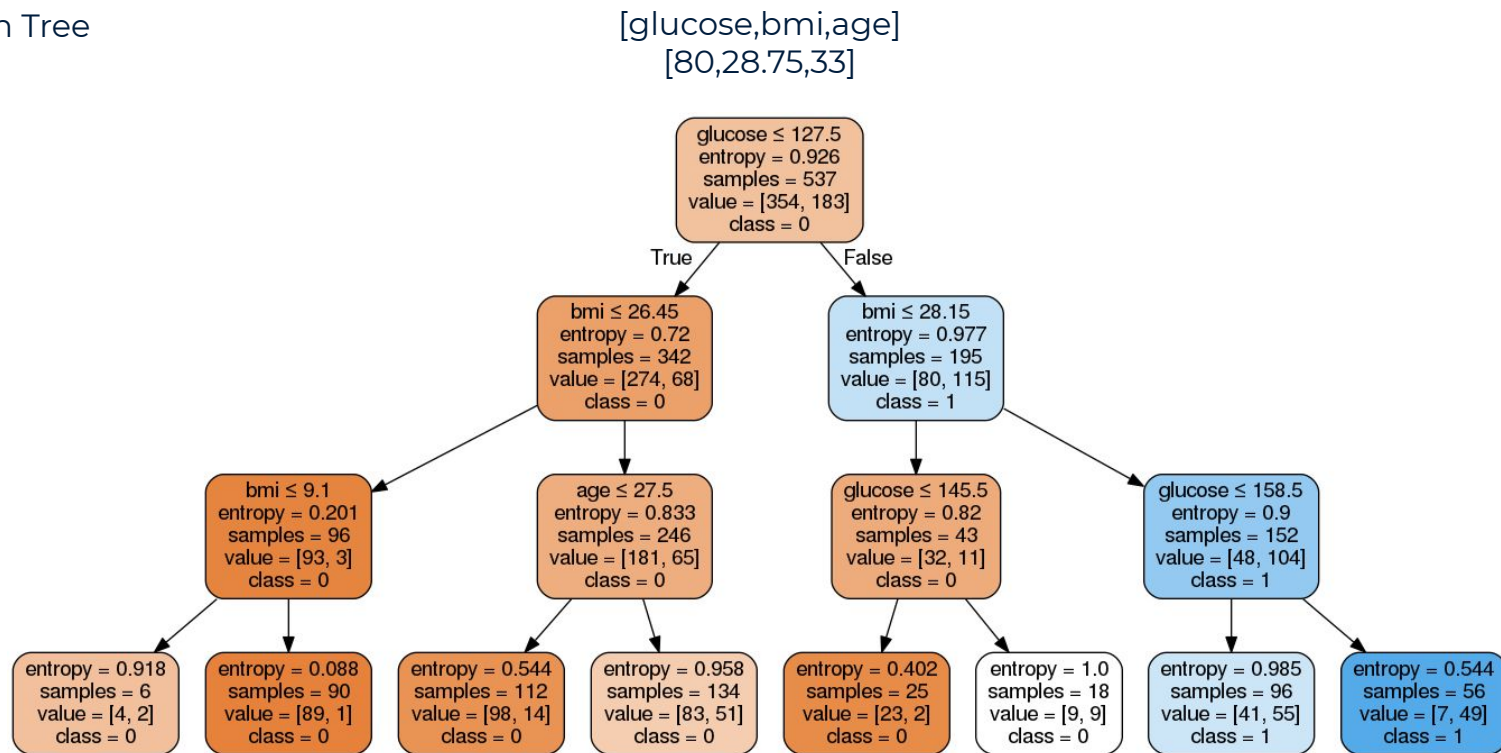
## Decision Tree

- Pouco pre-processamento
- Fácil de interpretar
- Altamente customizável (prunning)



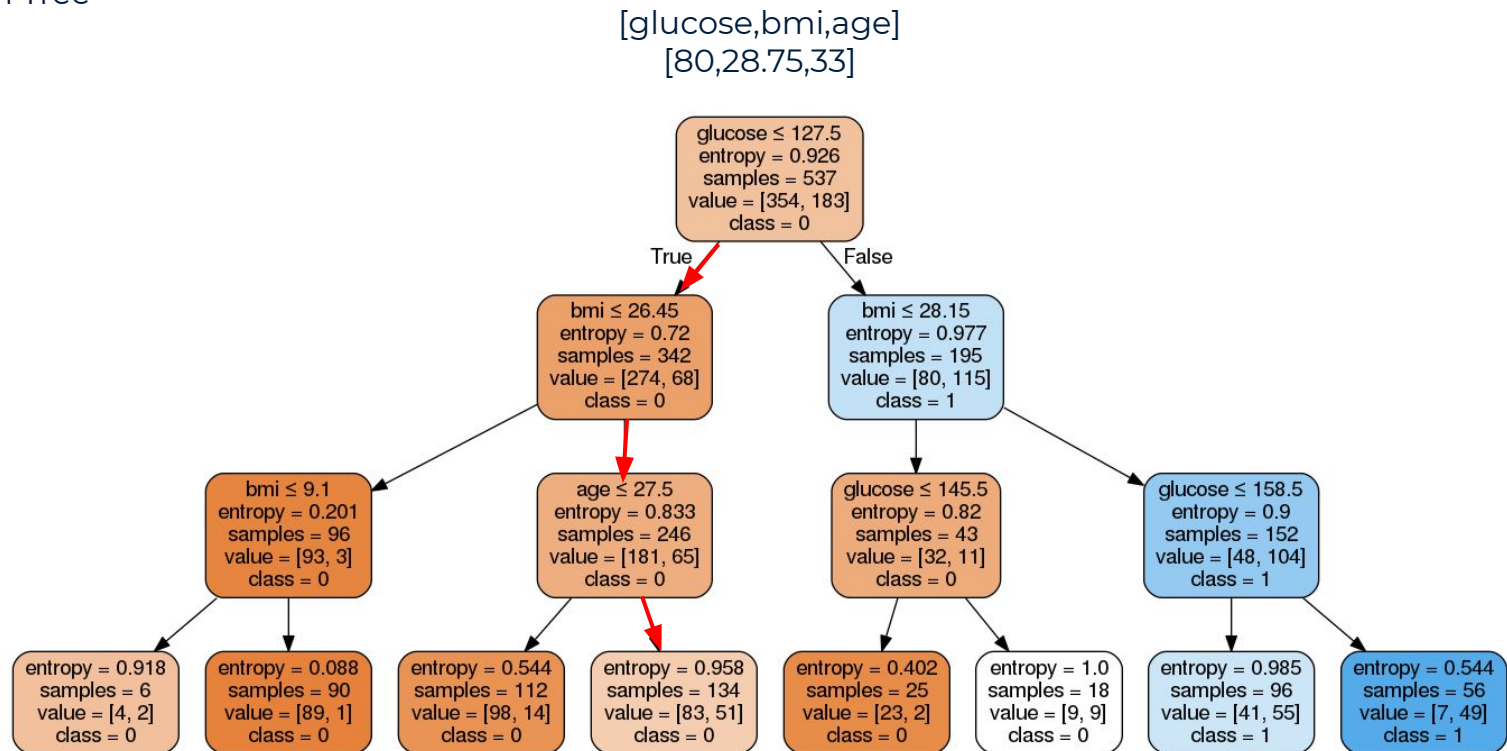
# Algoritmos

## Decision Tree



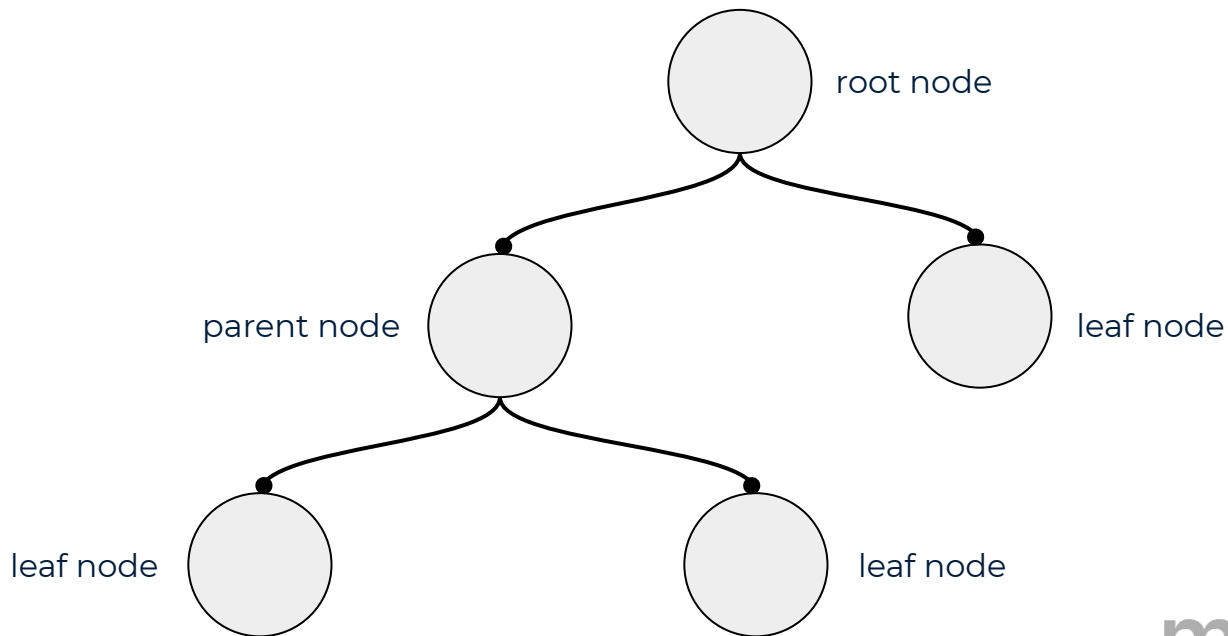
# Algoritmos

## Decision Tree



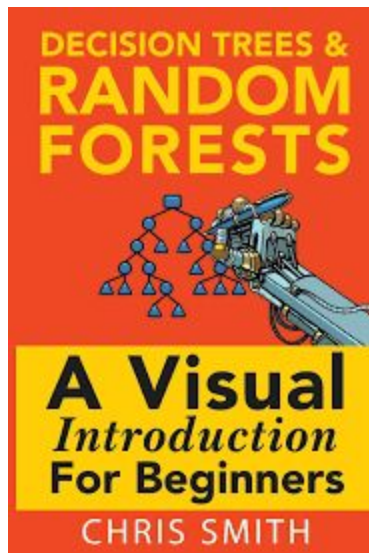
# Algoritmos

## Decision Tree



# Algoritmos

Decision Tree





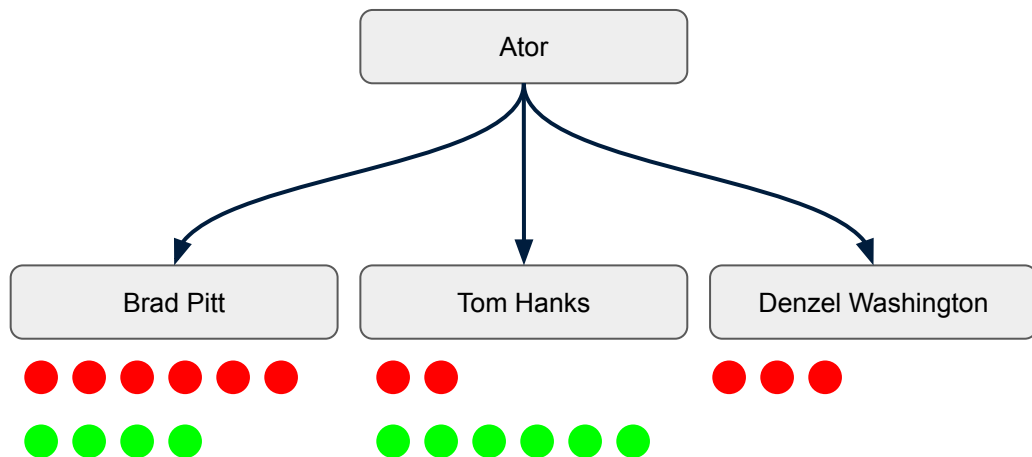
# Algoritmos

## Decision Tree

Filme	Ator	Categoria	Avaliação	Assistiu?
Bastador Inglórios	Brad Pitt	Ação	Mediana	Não
Porcos e Diamantes	Brad Pitt	Ação	Mediana	Não
Tróia	Brad Pitt	Ação	Baixa	Não
O curioso caso de Benjamin Button	Brad Pitt	Drama	Mediana	Não
Sr e Sra Smith	Brad Pitt	Romance	Baixa	Não
Era uma vez em Hollywood	Brad Pitt	Ação	Baixa	Não
Os sete magníficos	Denzel Washington	Ação	Mediana	Não
O livro de Eli	Denzel Washington	Ação	Baixa	Não
Decisão de Risco	Denzel Washington	Drama	Mediana	Não
Inferno	Tom Hanks	Drama	Baixa	Não
Mensagem para você	Tom Hanks	Romance	Baixa	Não
Clube da Luta	Brad Pitt	Ação	Mediana	Sim
Corações de Ferro	Brad Pitt	Drama	Mediana	Sim
O homem que mudou o jogo	Brad Pitt	Drama	Mediana	Sim
Aliados	Brad Pitt	Drama	Excepcional	Sim
Anjos e Demônios	Tom Hanks	Drama	Excepcional	Sim
Capitão Phillips	Tom Hanks	Drama	Mediana	Sim
Apollo 13	Tom Hanks	Drama	Excepcional	Sim
Forrest Gump	Tom Hanks	Drama	Excepcional	Sim
O Náufrago	Tom Hanks	Drama	Mediana	Sim
Sintonia de amor	Tom Hanks	Romance	Excepcional	Sim

# Algoritmos

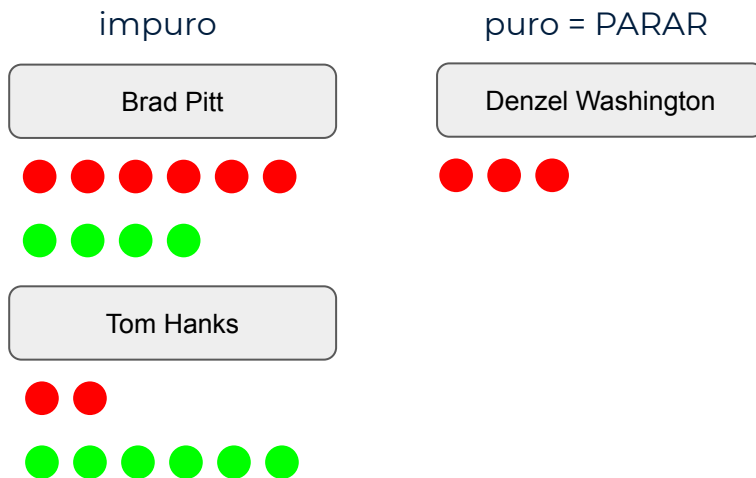
## Decision Tree



● sim  
● não

# Algoritmos

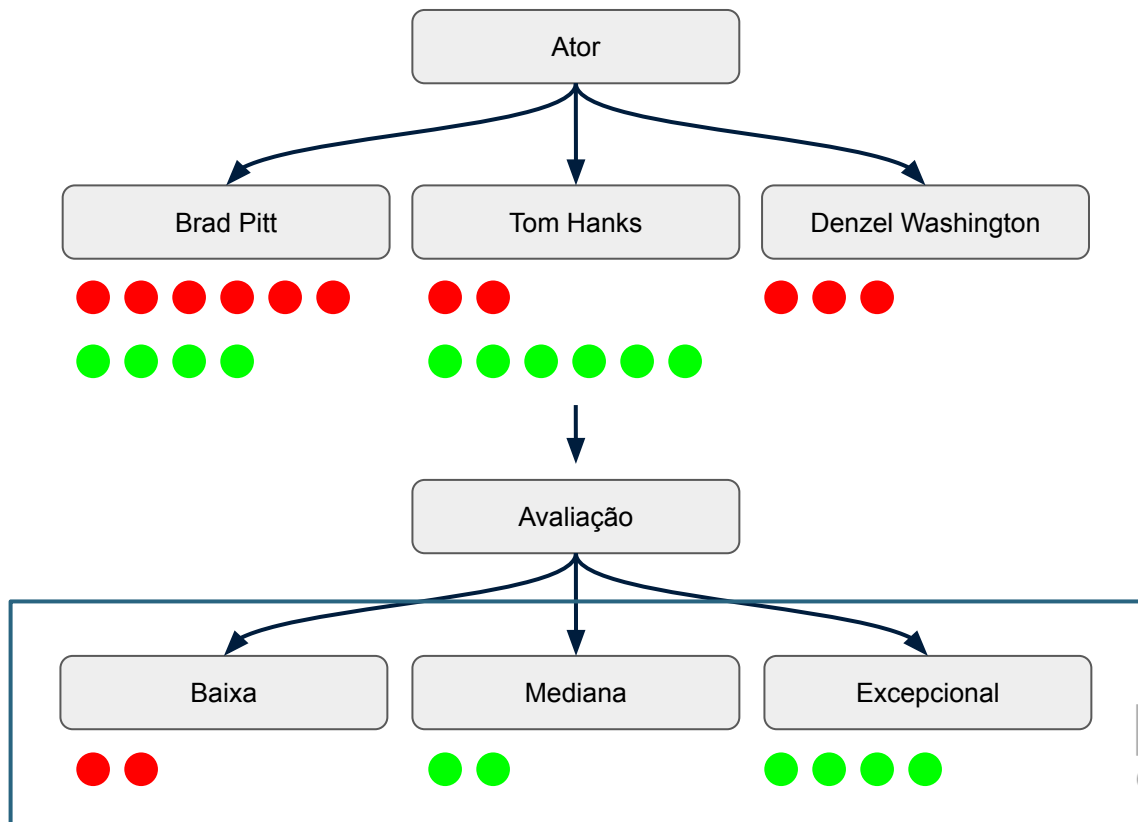
## Decision Tree



● sim  
● não

# Algoritmos

## Decision Tree



puro

mesttra  
discover. shape. connect.

# Algoritmos

## Decision Tree

$$I_H = - \sum_{j=1}^c p_j \log_2(p_j)$$

$p_j$ : proportion of the samples that belongs to class  $c$  for a particular node.

A entropia pode ser definida de várias formas, mas vamos considerar ela como uma medida de quanta informação está faltando!

# Algoritmos

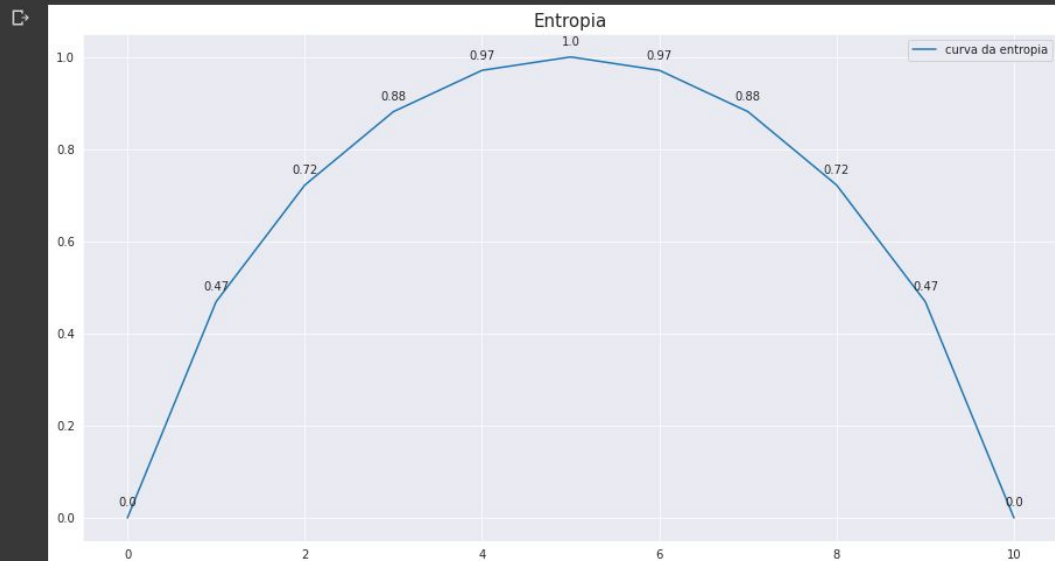
## Decision Tree

```
[320] 1  ## demostrando valores de entropia
      2
      3  from scipy.stats import entropy
      4
      5  data = []
      6
      7  for i in range(11):
      8  |  data.append(entropy([i/10, (10-i)/10], base=2))
```

# Algoritmos

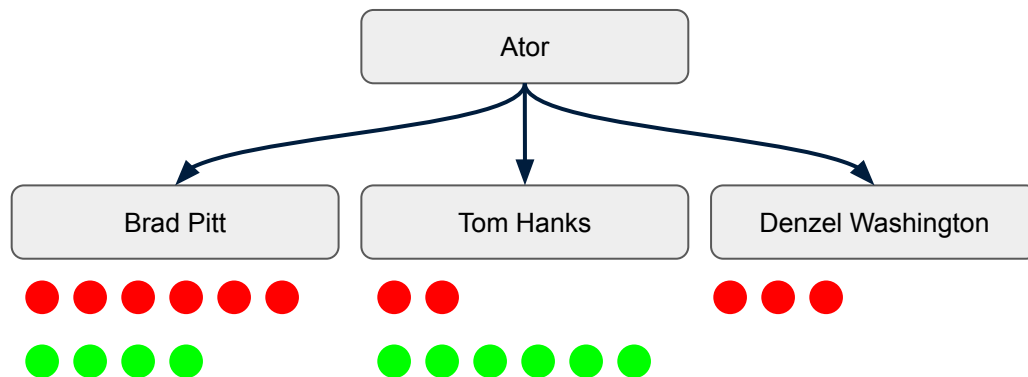
## Decision Tree

```
[330] 1  ## plotando entropia
2
3  plt.figure(figsize=(15,8))
4  sns.lineplot(x=list(range(11)),y=data,label='curva da entropia')
5
6  for i,value in enumerate(data):
7      plt.text(x = i , y = value + 0.02, s = round(value,2),horizontalalignment='center',verticalalignment='bottom')
8
9  plt.title('Entropia',size=15)
10 plt.show()
```



# Algoritmos

## Decision Tree



Brad Pitt

Tom Hanks

$$\text{Entropia}(\text{Brad Pitt}) = - (6/10 * \log_2(6/10)) - (4/10 * \log_2(4/10))$$

$$\text{Entropia}(\text{Brad Pitt}) = - (-0.44) - (4/10 * \log_2(4/10))$$

$$\text{Entropia}(\text{Brad Pitt}) = 0.44 - (-0.52)$$

$$\text{Entropia}(\text{Brad Pitt}) = 0.44 + 0.52 = 0.97$$

$$\text{Entropia}(\text{Tom Hanks}) = - (2/8 * \log_2(2/8)) - (6/8 * \log_2(6/8))$$

$$\text{Entropia}(\text{Tom Hanks}) = - (-0.5) - (6/8 * \log_2(6/8))$$

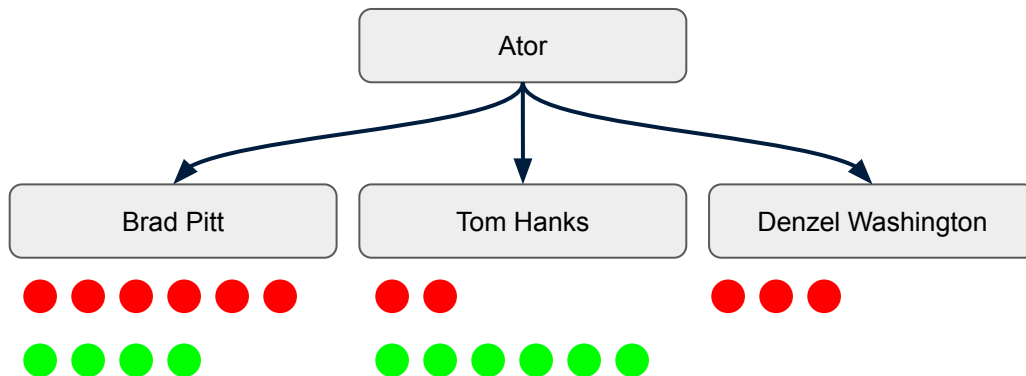
$$\text{Entropia}(\text{Tom Hanks}) = 0.5 - (-0.32)$$

$$\text{Entropia}(\text{Tom Hanks}) = 0.5 + 0.32 = 0.82$$



# Algoritmos

## Decision Tree



Denzel Washington

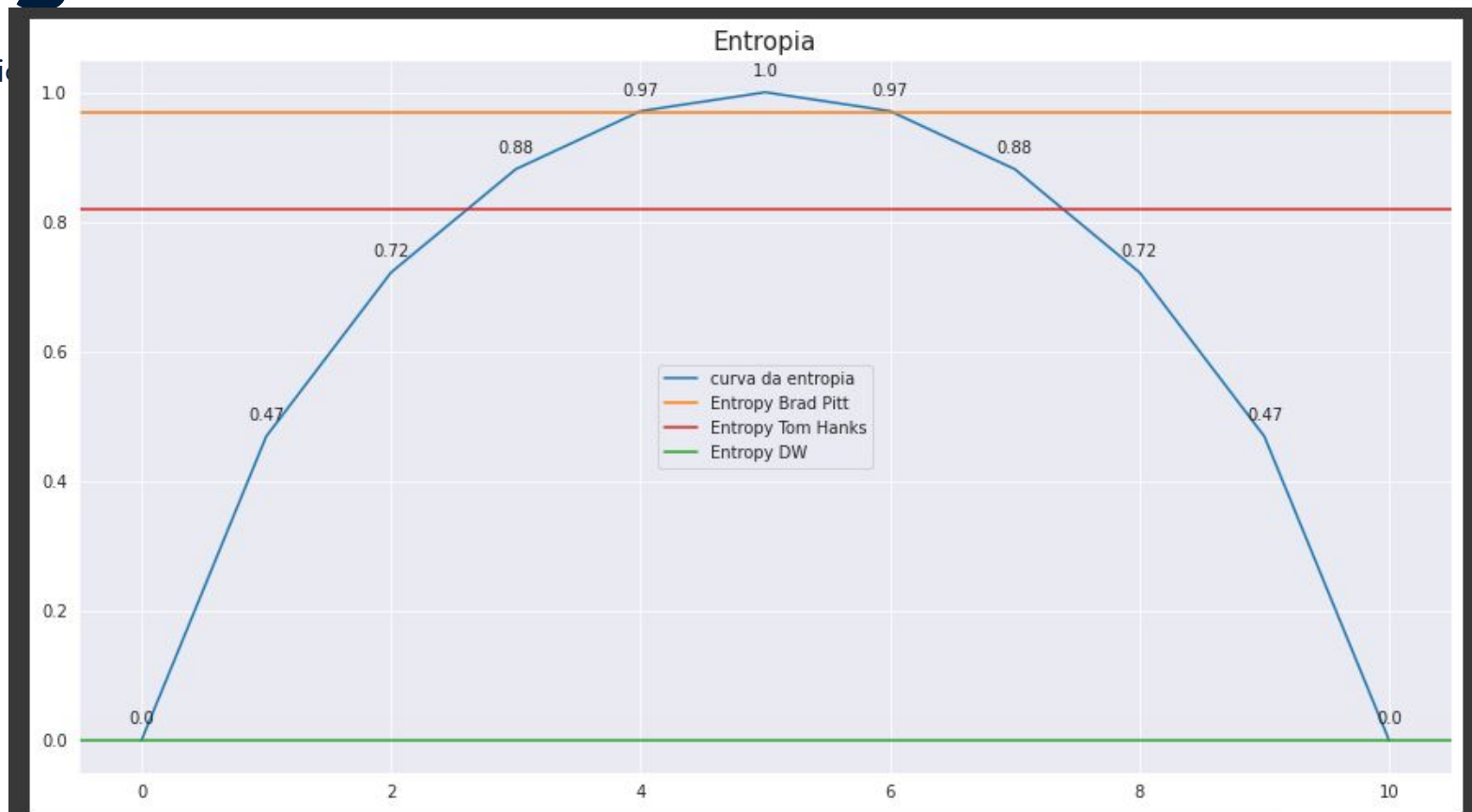
$$\text{Entropia(DW)} = - (3/3 * \log_2(3/3)) - (0/3 * \log_2(0/3))$$

$$\text{Entropia(DW)} = - (0) - (0)$$

$$\text{Entropia(DW)} = 0$$

# Algoritmos

Decisi



# Algoritmos

## Decision Tree

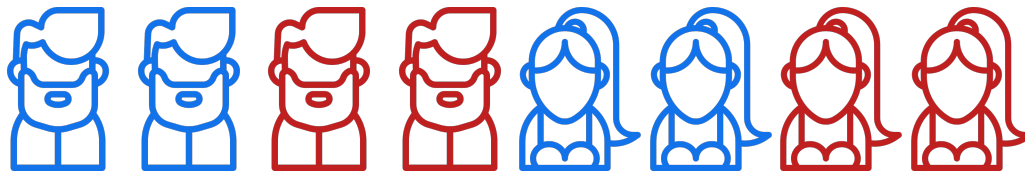
Information Gain é a medida do decrescimento da entropia para subsets dos dados.

Information Gain é usado para que o algoritmo decida qual feature vai ser usada para dividir (split) o dataset e tomar suas decisões.

<https://gaussian37.github.io/ml-concept-Information-Theory/>

# Algoritmos

Decision Tree

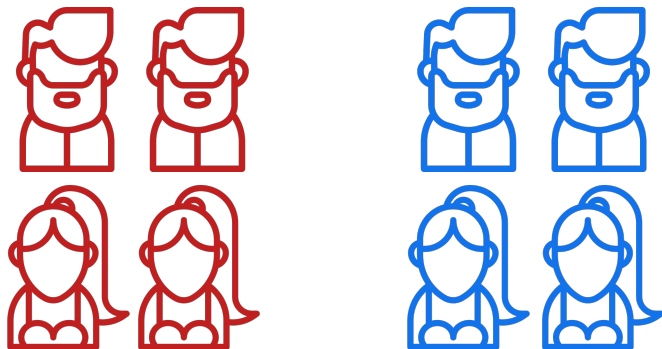


Cor da roupa

# Algoritmos

## Decision Tree

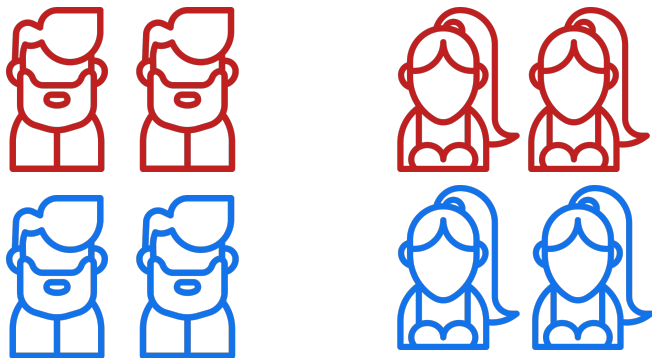
Cor da roupa  
(zero de ganho de informação)



# Algoritmos

## Decision Tree

Barba ?  
(máximo de ganho de informação)



# Algoritmos

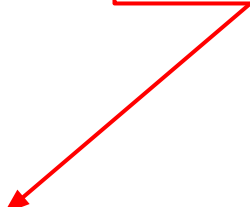
## Decision Tree

Information Gain = entropia(parent) - [média ponderada entropia(subsets)]

# Algoritmos

## Decision Tree

Information Gain =  $\text{entropia}(\text{parent}) - [\text{média ponderada entropia}(\text{subsets})]$



Entropia antes do split



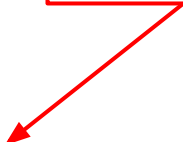
# Algoritmos

## Decision Tree

● não x 11

● sim x 10

Information Gain = entropia(parent) - [média ponderada entropia(subsets)]

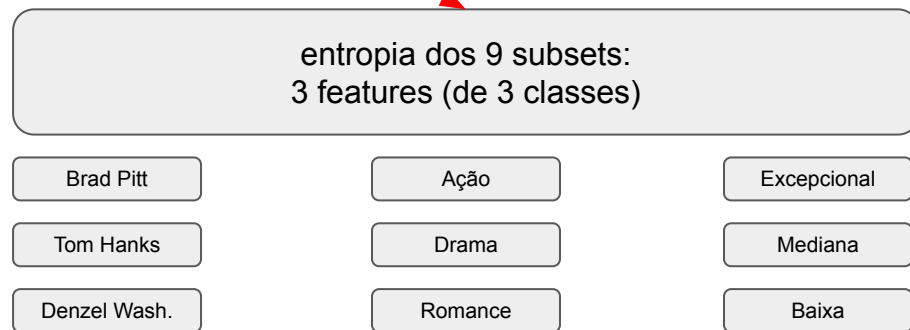

$$\text{ent}(\text{assistir}) = -(10/21 * \log_2(10/21)) - (11/21 * \log_2(11/21))$$

$$\text{ent}(\text{assistir}) = -(-0.51) - (-0.49) = 1$$

# Algoritmos

## Decision Tree

Information Gain = entropia(parent) - [média ponderada entropia(subsets)]



# Algoritmos

## Decision Tree

Information Gain = entropia(parent) - [média ponderada entropia(subsets)]

Brad Pitt

Tom Hanks

Denzel Wash.

$$\text{ent(ator)} = (10/21 * 0.97) + (8/21 * 0.82) + (3/21 * 0)$$

$$\text{ent(ator)} = 0.46 + 0.31 + 0$$

$$\text{ent(ator)} = 0.77$$

# Algoritmos

## Decision Tree

$$\text{Information Gain(ator)} = 1 - 0.77$$

$$\text{Information Gain(ator)} = 0.23$$

0.23 é bom ou ruim? Ator é uma feature boa para splitar nosso dataset de início?  
Para saber essas respostas temos que calcular o information gain para todas as features de um dataset. É exatamente isso que árvore de decisão faz.

Quanto maior for o information gain,  
menor vai ser a impureza do resultado do split.

# Algoritmos

## Decision Tree

Information Gain(ator) =  $1 - 0.77$

Information Gain(ator) = 0.23

Information Gain(categoria) =  $1 - 0.78$

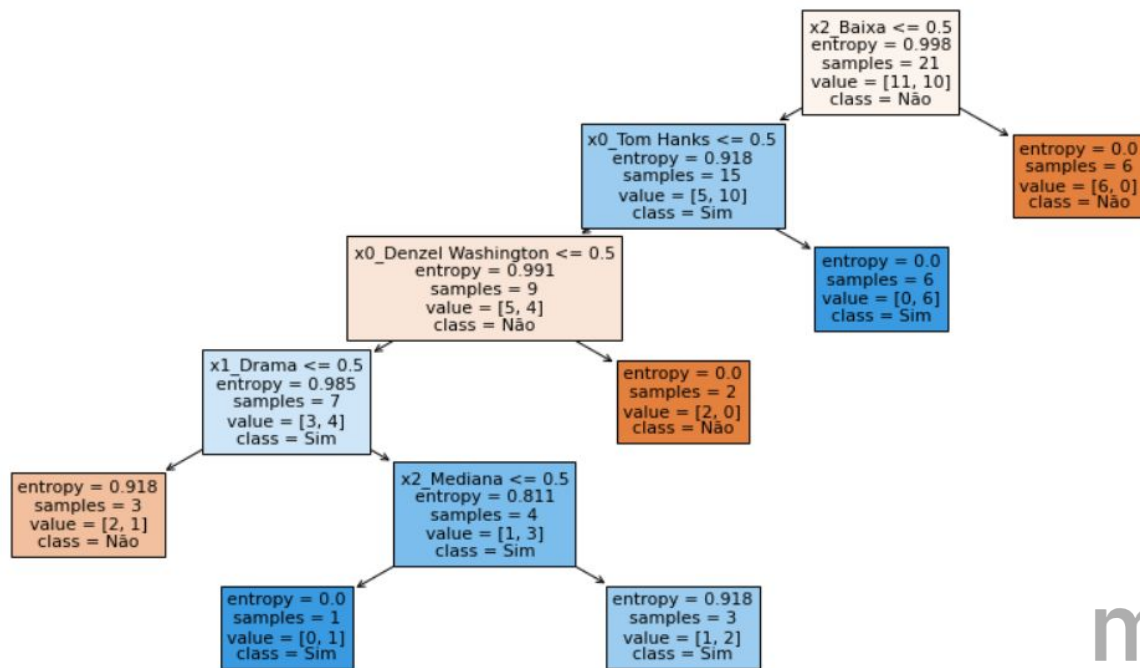
Information Gain(categoria) = 0.22

Information Gain(avaliação) =  $1 - 0.47$

Information Gain(avaliação) = 0.53

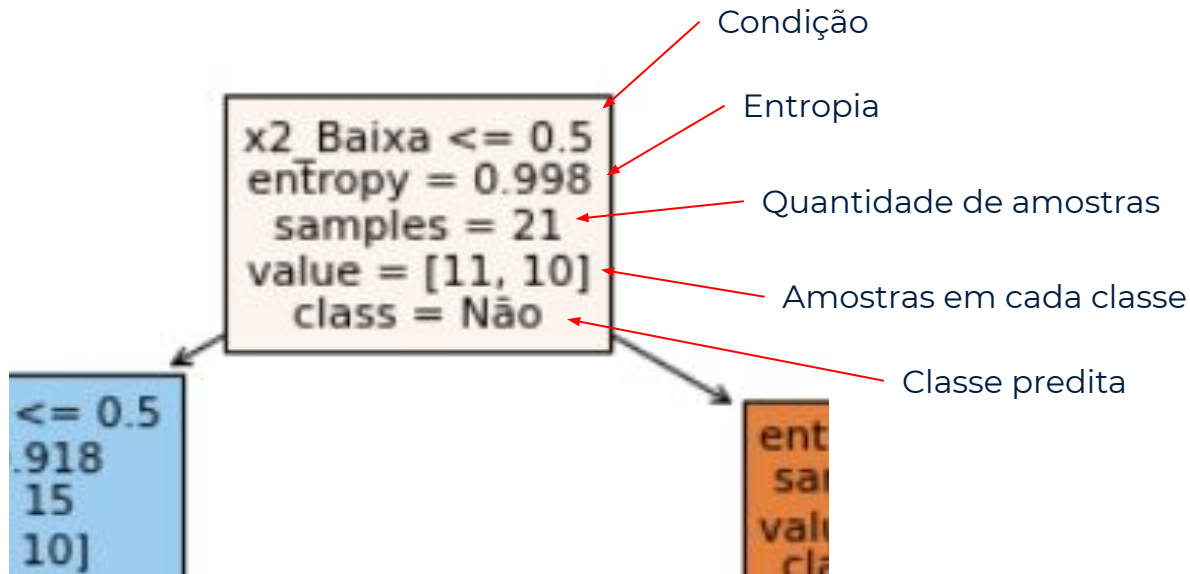
# Algoritmos

## Decision Tree



# Algoritmos

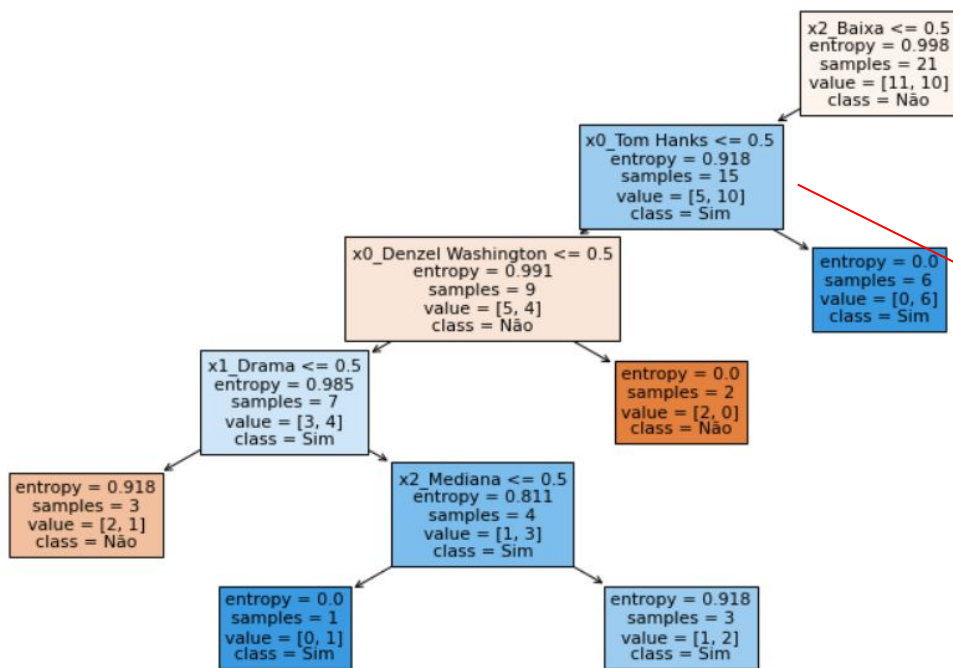
## Decision Tree



# Algoritmos

## Decision Tree

Avaliação baixa?



Avaliação	Assistiu?
Baixa	Não
Baixa	Não
Baixa	Não
Baixa	Não
Baixa	Não
Baixa	Não

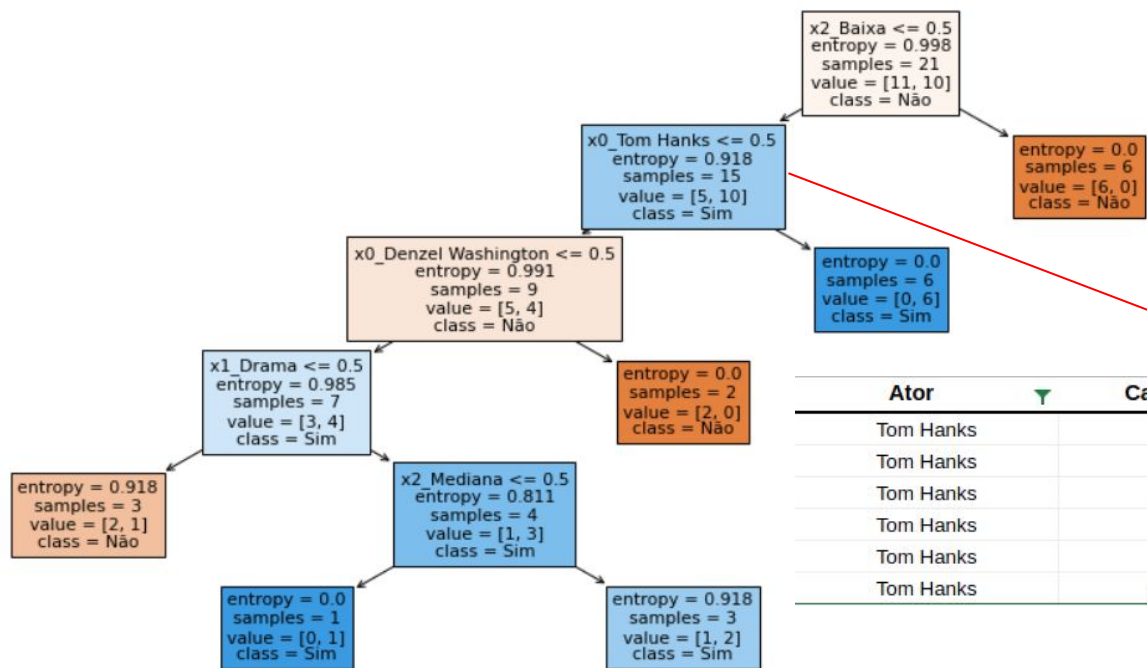
Avaliação	Assistiu?
Mediana	Não
Mediana	Não
Mediana	Não
Mediana	Não
Mediana	Sim
Mediana	Sim
Mediana	Sim
Excepcional	Sim
Excepcional	Sim
Excepcional	Sim
Excepcional	Sim
Mediana	Sim
Excepcional	Sim

sttra  
shape. connect.



# Algoritmos

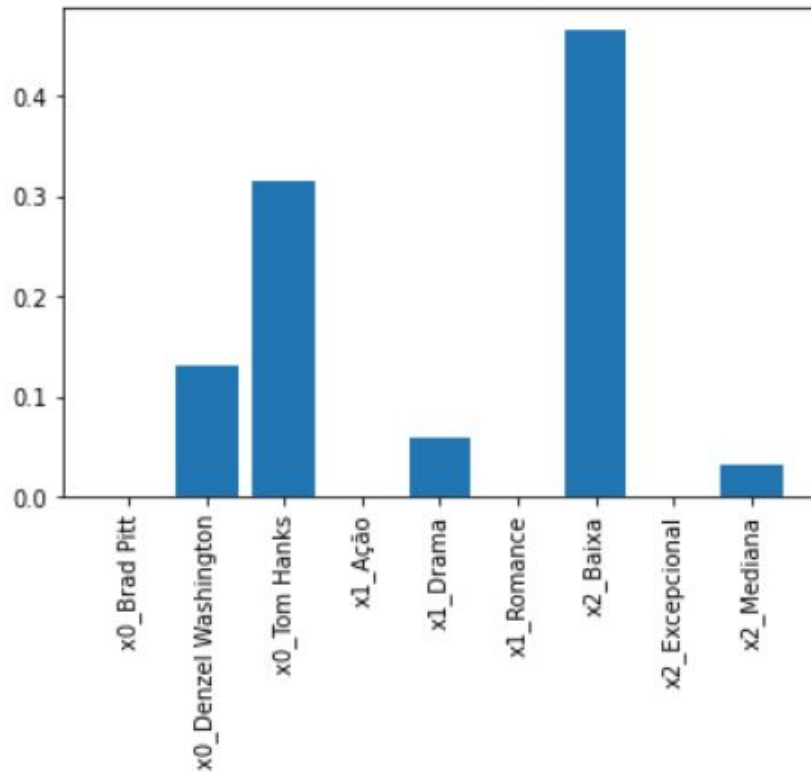
## Decision Tree



Ator	Categoria	Avaliação	Assistiu?
Tom Hanks	Drama	Excepcional	Sim
Tom Hanks	Drama	Mediana	Sim
Tom Hanks	Drama	Excepcional	Sim
Tom Hanks	Drama	Excepcional	Sim
Tom Hanks	Drama	Mediana	Sim
Tom Hanks	Romance	Excepcional	Sim

# Algoritmos

## Decision Tree



# Algoritmos

## Decision Tree

```
[92]: ohe.get_feature_names()
```

```
[92]: array(['x0_Brad Pitt', 'x0_Denzel Washington', 'x0_Tom Hanks', 'x1_Ação',  
          'x1_Drama', 'x1_Romance', 'x2_Baixa', 'x2_Excepcional',  
          'x2_Mediana'], dtype=object)
```

```
[94]: dt.predict([[0,0,1,1,0,0,0,1,0]])
```

```
[94]: array([1])
```

```
[95]: dt.predict_proba([[0,0,1,1,0,0,0,1,0]])
```

```
[95]: array([[0., 1.]])
```

# Algoritmos

Titanic

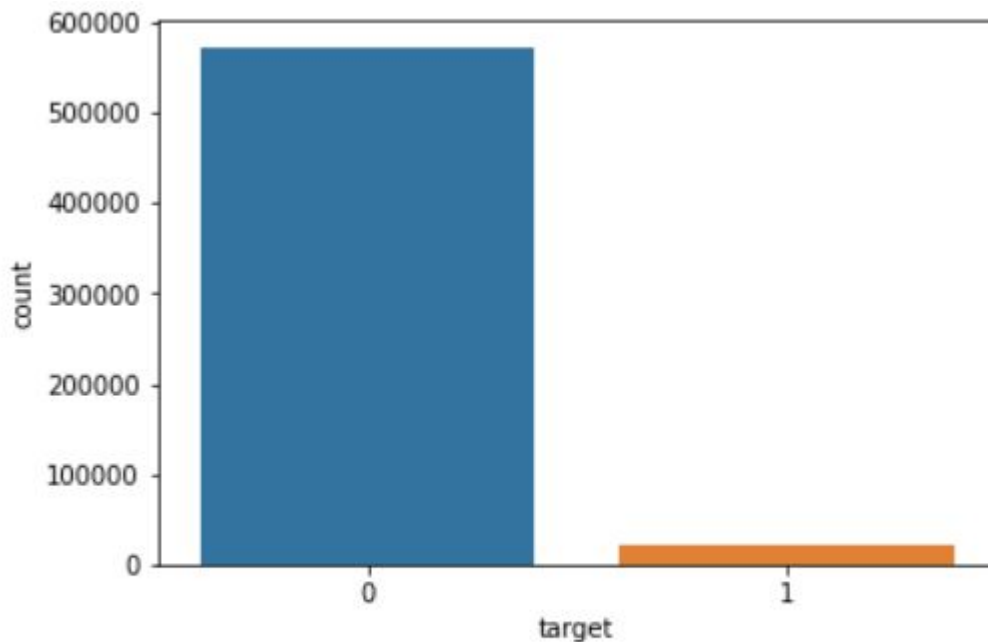
Me: \*uses machine learning\*

Machine: \*learns\*

Me:

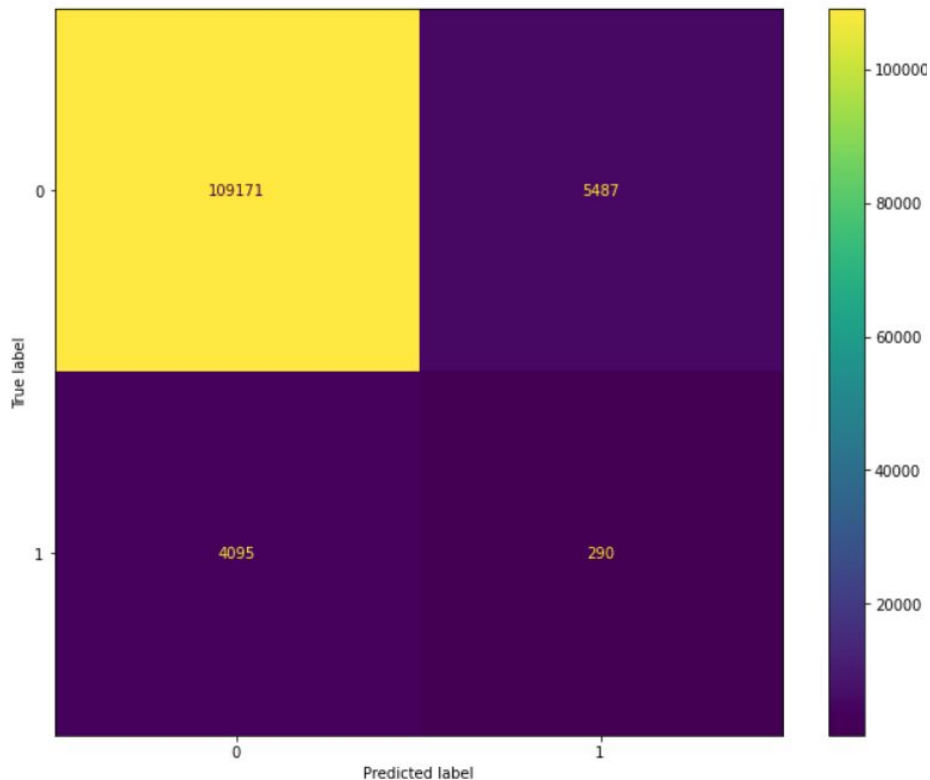


# Bases Desbalanceadas



```
0    0.963552
1    0.036448
Name: target, dtype: float64
```

# Bases Desbalanceadas



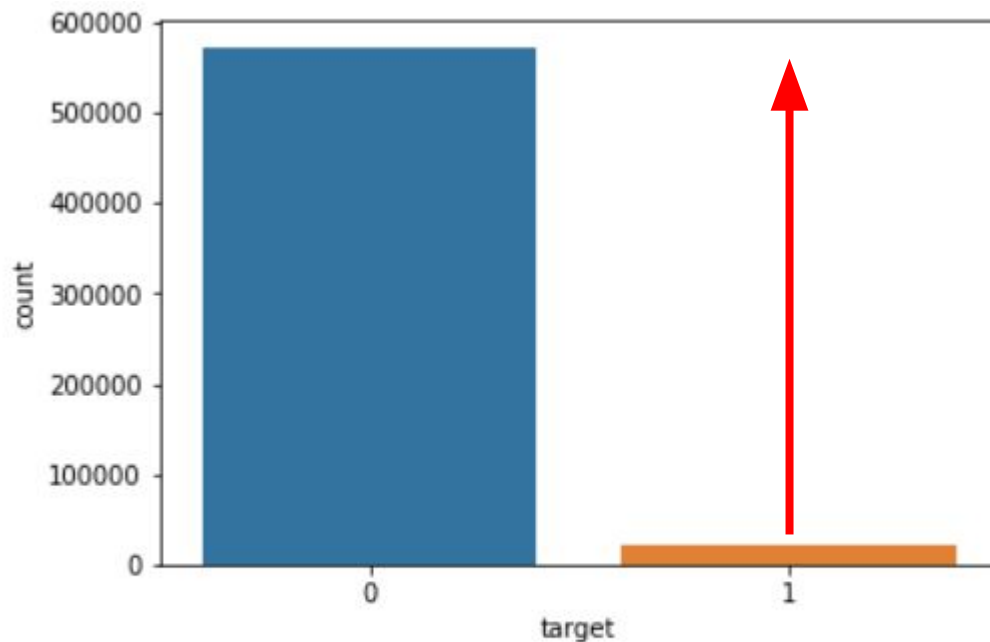
	precision	recall	f1-score	support
0	0.96	0.95	0.96	114658
1	0.05	0.07	0.06	4385
accuracy			0.92	119043
macro avg	0.51	0.51	0.51	119043
weighted avg	0.93	0.92	0.92	119043

# Bases Desbalanceadas



# Bases Desbalanceadas

Under e OverSampling

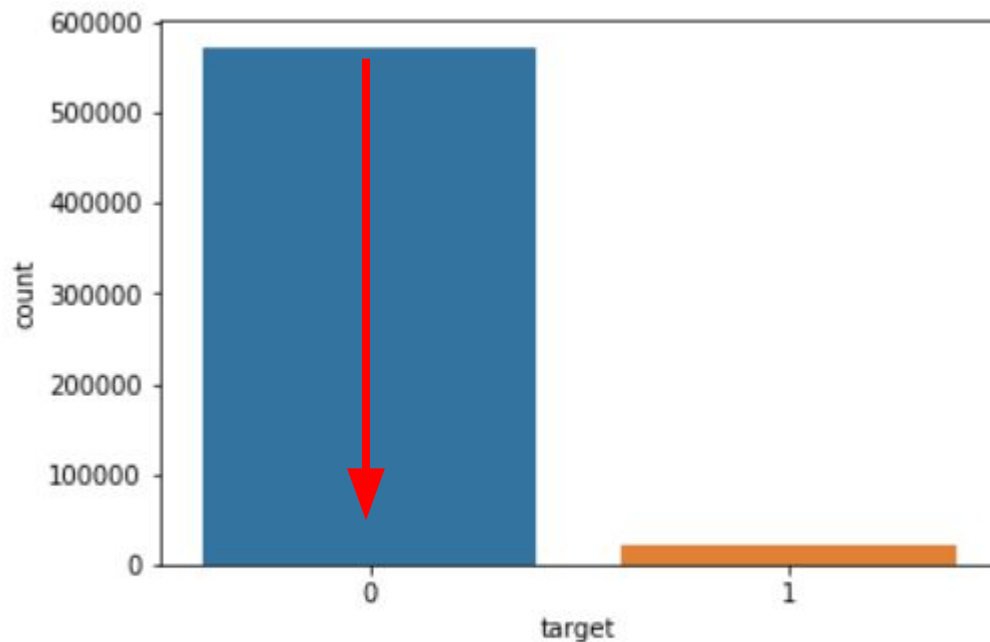


```
0    0.963552
1    0.036448
Name: target, dtype: float64
```



# Bases Desbalanceadas

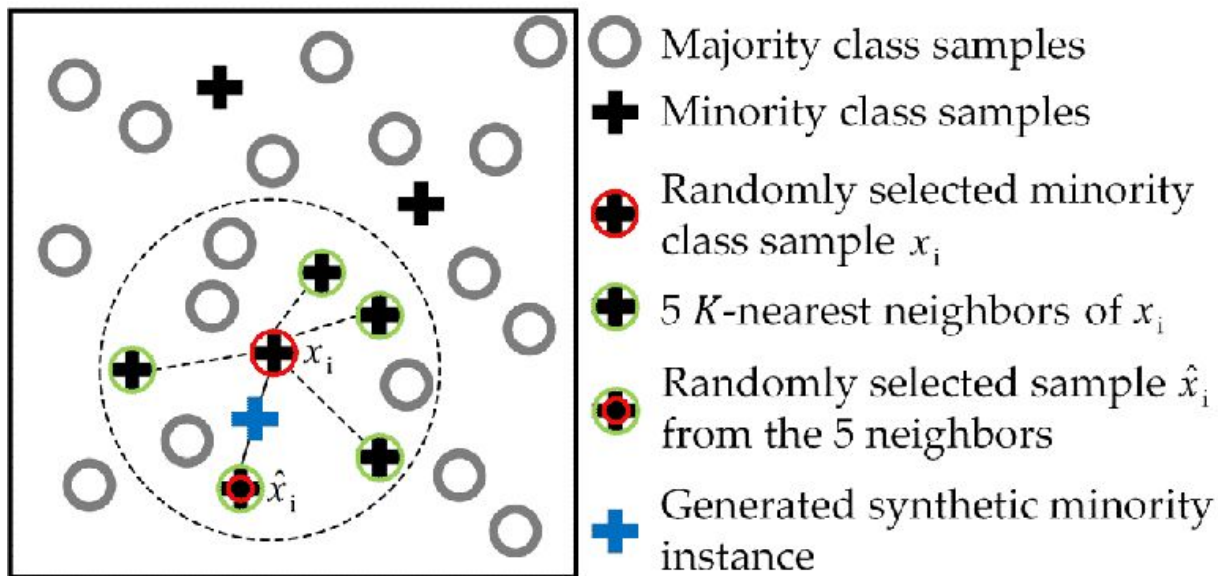
Under e OverSampling



```
0    0.963552
1    0.036448
Name: target, dtype: float64
```

# Bases Desbalanceadas

SMOTE



# Bases Desbalanceadas

SMOTE

```
[32]: from imblearn.over_sampling import SMOTE
```

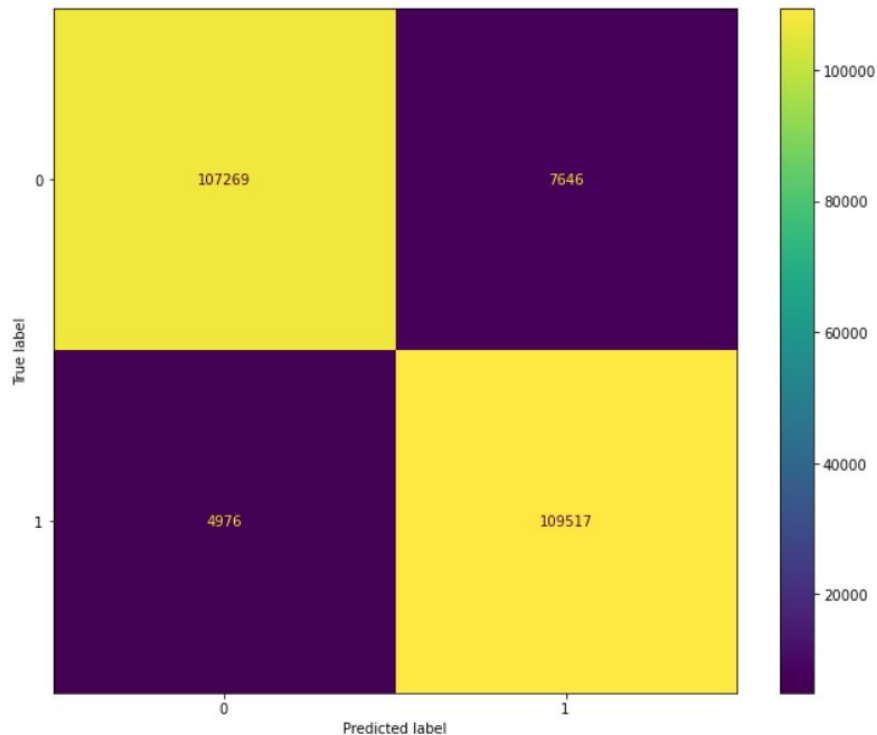
```
[35]: x_res, y_res = SMOTE(sampling_strategy='minority').fit_resample(x, y)
```

```
[37]: pd.Series(y_res).value_counts(normalize=True)
```

```
[37]: 0    0.5  
     1    0.5  
     Name: target, dtype: float64
```

# Bases Desbalanceadas

SMOTE



	precision	recall	f1-score	support
0	0.96	0.93	0.94	114915
1	0.93	0.96	0.95	114493
accuracy			0.94	229408
macro avg	0.95	0.95	0.94	229408
weighted avg	0.95	0.94	0.94	229408

# Bases Desbalanceadas

RandomUnderSampler

```
[39]: x_res, y_res = RandomUnderSampler().fit_resample(x, y)
```

```
[44]: pd.Series(y).value_counts()
```

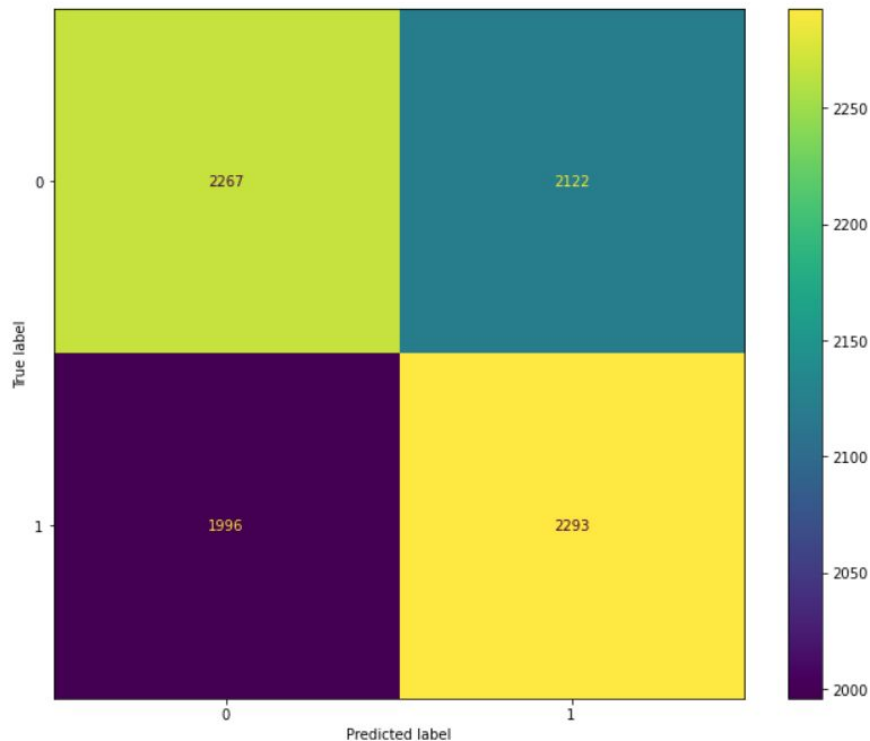
```
[44]: 0    573518  
      1     21694  
      Name: target, dtype: int64
```

```
[41]: pd.Series(y_res).value_counts()
```

```
[41]: 0     21694  
      1     21694  
      Name: target, dtype: int64
```

# Bases Desbalanceadas

RandomUnderSampler



	precision	recall	f1-score	support
0	0.53	0.52	0.52	4389
1	0.52	0.53	0.53	4289
accuracy			0.53	8678
macro avg	0.53	0.53	0.53	8678
weighted avg	0.53	0.53	0.53	8678

# Regressão

**mesttra**  
discover. shape. connect.

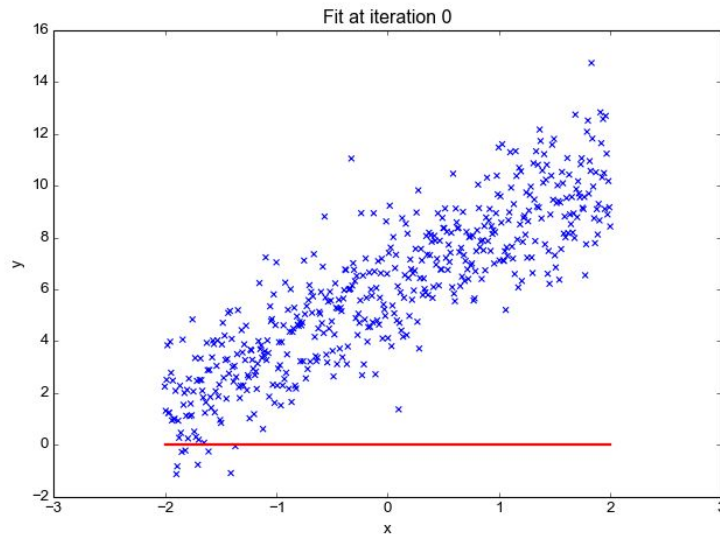
# Tipos de Regressão

- Regressão Linear Simples
- Regressão Linear Múltipla
- Regressão Polinomial



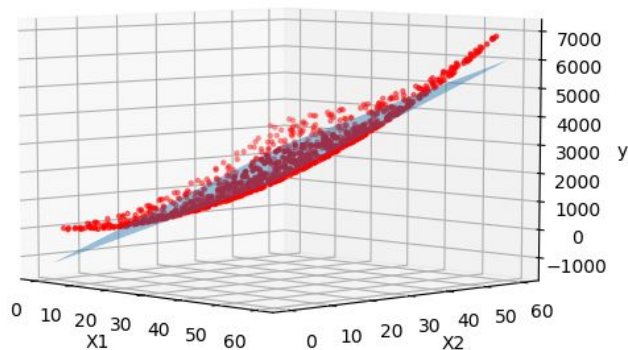
# Tipos de Regressão

- **Regressão Linear Simples**
- Regressão Linear Múltipla
- Regressão Polinomial



# Tipos de Regressão

- Regressão Linear Simples
- **Regressão Linear Múltipla**
- Regressão Polinomial



# Tipos de Regressão

- Regressão Linear Simples
- Regressão Linear Múltipla
- **Regressão Polinomial**



**mesttra**  
discover. shape. connect.

**mesttra**  
discover. shape. connect.

**mesttra**  
discover. shape. connect.

**mesttra**  
discover. shape. connect.