



# Introdução ao Machine Learning Aplicado com Scikit-learn e NumPy

Júlia Zoffoli, Luiz Nazareth, Robert Gonçalves

GET Engenharia Computacional

# Estrutura do Minicurso

- ① Introdução
- ② Pré-processamento
- ③ Supervisionado
- ④ Não Supervisionado
- ⑤ Validação de Modelos

# Introdução à Machine Learning

Processo de ensinar um computador a fazer previsões ou tomar decisões a partir de dados, identificando padrões ou regularidades.

## Exemplos:

- ① Recomendação de Produtos (ex.: Netflix, Amazon)
- ② Reconhecimento de Imagens (ex.: diagnóstico médico, reconhecimento facial)
- ③ Previsão de Séries Temporais (ex.: previsão de demanda, previsão de vendas)

# Introdução à Machine Learning

## Fluxo de Trabalho

- Pré Processamento
- Treinamento
- Validação



# Desenvolvimento no Colab

Link para o acesso ao minicurso no colab  
está no site do GET-EngComp:

[www2.ufjf.br/getengcomp/](http://www2.ufjf.br/getengcomp/)

# Pré-processamento de Dados

- Dados reais podem conter valores ausentes, variáveis desbalanceadas ou ruídos.
- Preparação, organização e estruturação dos dados.
- Modelos de ML requerem dados bem formatados.

	cargo	idade	salario	bonus	sócio
0	Diretor	45	24000.0	10000.0	sim
1	Analista	22	8000.0	2000.0	não
2	Programador	30	NaN	1000.0	não
3	Gerente	24	15100.0	NaN	não
4	Gerente	30	35000.0	6000.0	sim

# Técnicas

- Codificação de variáveis categóricas
- Padronização
- Normalização
- Divisão dos dados

# Codificação de variáveis categóricas

Utilizadas para converter variáveis categóricas em uma forma numérica que possa ser utilizada por modelos de Machine Learning.

- Label Encoder.
- One Hot Encoder.

# Label Encoder

- Converte variáveis categóricas em valores inteiros.
- Ideal para categorias com uma ordem (variáveis ordinais).

Cor	Label Encoder
Vermelho	0
Verde	1
Azul	2

Tamanho	Label Encoder
Baixo	0
Médio	1
Alto	2

Cor	Tamanho
Vermelho	Baixo
Verde	Médio
Azul	Alto
Vermelho	Alto
Verde	Baixo

Cor Codificada	Tamanho Codificado
0	0
1	1
2	2
0	2
1	0

# One Hot Encoder

- Cria uma nova variável binária (0 ou 1) para cada categoria única da variável original.
- Ideal para um número reduzido de categorias e variáveis nominais.

Cor	Tamanho
Vermelho	Baixo
Verde	Médio
Azul	Alto
Vermelho	Alto
Verde	Baixo

Cor_Vermelho	Cor_Verde	Cor_Azul	Tamanho_Baixo	Tamanho_Medio	Tamanho_Alto
1	0	0	1	0	0
0	1	0	0	1	0
0	0	1	0	0	1
1	0	0	0	0	1
0	1	0	1	0	0

## Padronização (Z-Score)

- Os dados são transformados de acordo com a distribuição estatística.
- Utiliza a Média e o Desvio Padrão.
- Menos sensível a outliers (valores extremos).
- Pode resultar em valores negativos.

$$X' = \frac{X - \mu}{\sigma}$$

# Padronização

Padronização para o exemplo anterior

- Média ( $\mu$ ): 41.0
- Desvio padrão ( $\sigma$ ): 19.97

Idade	Padronizados
15	-1,30
20	-1,05
35	-0,30
40	-0,05
47	0,30
50	0,45
80	1,95

# Normalização

- Usado para reescalar os valores dos dados para que fiquem dentro de um intervalo específico, geralmente entre 0 e 1.
- Garante que todas as variáveis tenham a mesma escala.
- Evita que variáveis com valores maiores influenciem de forma desproporcional o modelo de Machine Learning.

## Normalização Min-Max

- Um exemplo de normalização min-max é a conversão de idades de um grupo de pessoas para um intervalo entre 0 e 1.
- Se as idades são 15, 20, 35, 40 e 50 anos, o valor mínimo é 15 e o máximo é 50.

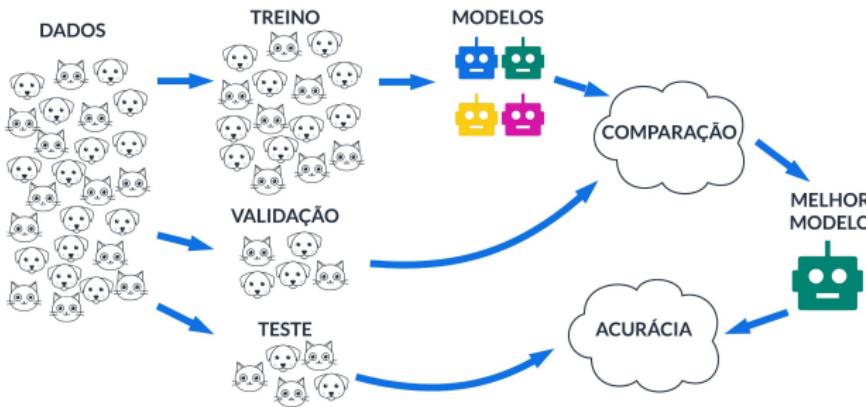
$$x' = \frac{x - \min(x)}{\max(x) - \min(x)}$$

Normalized Value      Original Value  
x' =  $\frac{x - \min(x)}{\max(x) - \min(x)}$   
Maximum Value of x      Minimum Value of x

Idade	Normalizados
15	0,0000
20	0,0769
35	0,3077
40	0,3846
47	0,4923
50	0,5385
80	1,0000

## Divisão dos dados

- **Treinamento:** Usado para ajustar os parâmetros do modelo.
- **Validação:** Usado para avaliar o desempenho do modelo e dos hiperparâmetros durante o treinamento.
- **Teste:** Usado para avaliar a performance final do modelo em dados não vistos durante o treinamento.



# Divisão dos dados

Divisão mais comum dos dados que pode variar com base no desempenho do modelo:

- **Treinamento:** 70% - 80% dos dados.
- **Validação:** 10% - 15% dos dados.
- **Teste:** 10% - 15% dos dados.

# Tipos de Aprendizado

## Aprendizado Supervisionado x Não Supervisionado

- Supervisionado:

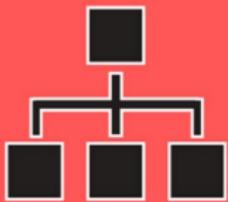
O modelo é treinado com dados rotulados, ou seja, dados em que já se sabe a resposta correta. (Regressão, Classificação).

- Não Supervisionado:

O modelo encontra padrões ou grupos sem rótulos, ou seja, dados em que não se sabe a resposta correta. (Clustering/Agrupamento).

# Tipos de Aprendizado

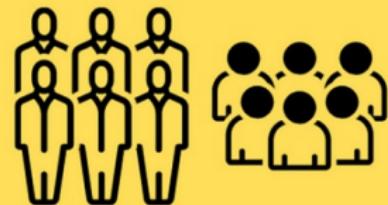
## Classificação



## Regressão



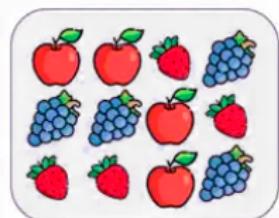
## Agrupamento



# Aprendizado Supervisionado

## Experiência adquirida:

Dados de entrada



Rótulos



Dados de teste  
(não rotulados)



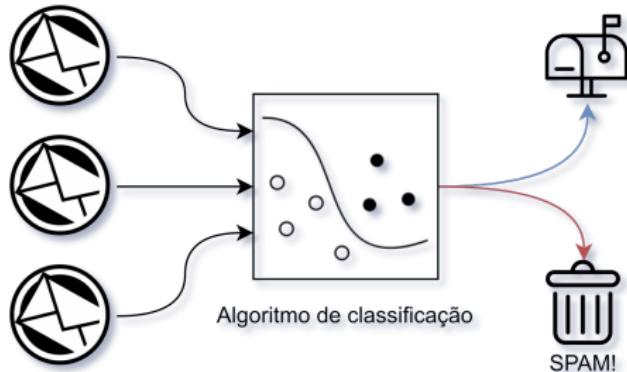
Predição



- **Predição: tolerância à estresses abióticos e bióticos;**
- **Produtividade;**
- **Potencial: futuras cultivares.**

# Classificação

- Entender, reconhecer padrões e agrupar o conjunto de dados em categorias.
- Prever a categoria



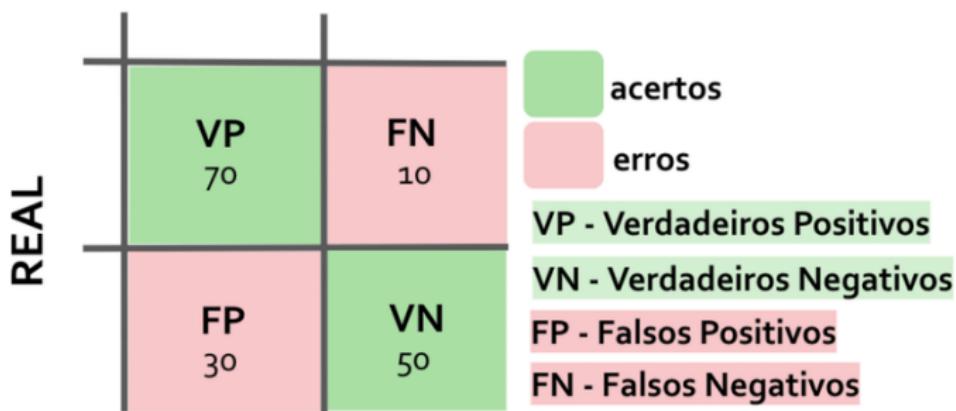
# Classificação

## Exemplos:

- **Classificação de Gêneros Musicais:** Um modelo pode classificar uma música em múltiplos gêneros (ex.: rock e jazz).
- **Classificação de Rótulos de Produtos:** Atribuir um rótulo (como eletrônicos, vestuário, alimentos) a um produto com base em sua descrição.
- **Diagnóstico Médico:** Um classificador que utilize dados não observados de um paciente e classifique-o como doente ou não-doente.

# Matriz de Confusão

## CLASSIFICAÇÃO DO MODELO



Usada para avaliar o desempenho de modelos de classificação.

# Acurácia e Precisão

## Acurácia

- Mede a proporção de **previsões corretas** (positivas e negativas) em relação ao total de previsões feitas.

$$\text{Acuracia} = \frac{VP + VN}{VP + VN + FP + FN}$$

## Precisão

- Mede a proporção de **previsões positivas** que estão realmente corretas.
- "Das vezes que o modelo previu positivo, quantas vezes ele estava certo?"

$$\text{Precisao} = \frac{VP}{VP + FP}$$

# Recall e F1 Score

## Recall

- "Dos exemplos realmente positivos, quantos o modelo conseguiu prever corretamente?"

$$\text{Recall} = \frac{VP}{VP + FN}$$

## F1 Score

- Média harmônica entre Precisão e Recall.
- Equilibra as duas métricas.

$$F1Score = 2 * \frac{\text{Precisao} * \text{Recall}}{\text{Precisao} + \text{Recall}}$$

# Não devemos olhar só a acurácia!

	Real Positiva	Real Negativa
Prevista Positiva	Verdadeiro Pos.	Falso Pos.
Prevista Negativa	Falso Neg.	Verdadeiro Neg.

	Real "Bom"	Real "Ruim"
Prevista "Bom"	0	0
Prevista "Ruim"	10	990

## Dataset desbalanceado

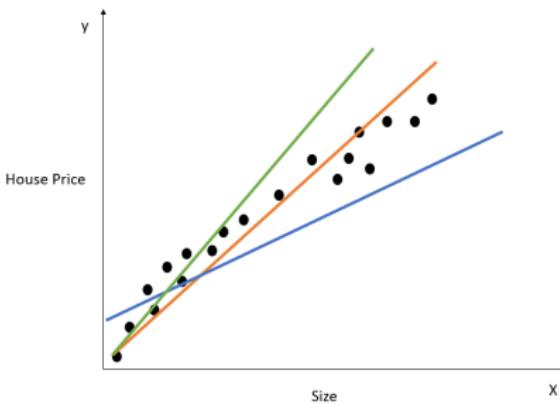
- 990 "Ruim"
- 10 "Bom"

## Modelo classifica tudo como "Ruim"

- Acurácia: 99%
- Recall: 0%

# Regressão

- Encontrar uma **função** que relate as variáveis de entrada com a variável de saída.



- A linha **laranja** é a mais próxima de todos os pontos de dados mostrados.
- "Linha de Melhor Ajuste".

# Mean Absolut Error - MAE

- Média da diferença entre o real e o previsto.
- valores maiores ou iguais a zero (módulo).

Previsto	Real	Diferença
3,34	3	0,34
4,18	4	0,18
3	3	0
2,99	3	0,01
4,51	4,5	0,01
5,18	4	1,18
8,18	4,5	3,68
		5,4

$$\text{MAE} = \sum_{i=1}^N \frac{|p_i - t_i|}{n}$$

$$\text{MAE} = \frac{5,4}{7} = 0,77$$

# Root Mean Squared Error - RMSE

- Penaliza erros maiores de forma mais severa (quadrado).
- **Baixo RMSE:** O modelo está prevendo os valores reais com maior precisão.
- **Alto RMSE:** O modelo está cometendo erros significativos nas previsões.

# Root Mean Squared Error - RMSE

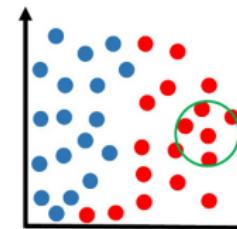
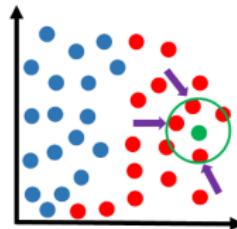
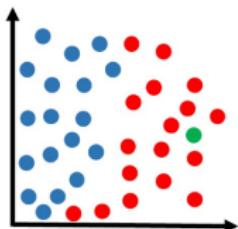
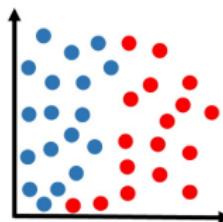
Previsto	Real	Diferença
3,34	3	0,1156
4,18	4	0,0324
3	3	0
2,99	3	1E-04
4,51	4,5	1E-04
5,18	4	1,3924
8,18	4,5	13,5424
		15,083

$$\text{RMSE} = \sqrt{\frac{\sum_{i=1}^N (p_i - t_i)^2}{N}}$$

$$\text{RMSE} = \sqrt{\frac{15,083}{7}}$$

# KNN

- K-nearest neighbors, ou “K-vizinhos mais próximos”
- Se os vizinhos mais próximos forem majoritariamente de uma classe, a amostra em questão será classificada nesta categoria.

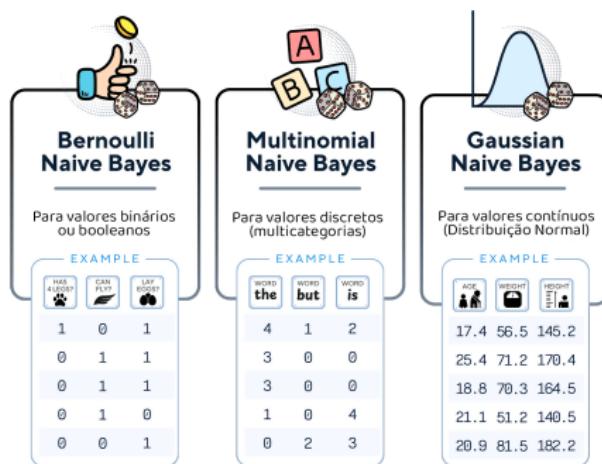


# Naive Bayes

- Baseado no Teorema de Bayes de probabilidades condicionais.

$$P(C|X) = \frac{P(X|C) * P(C)}{P(X)}$$

X: Data, C: Classe



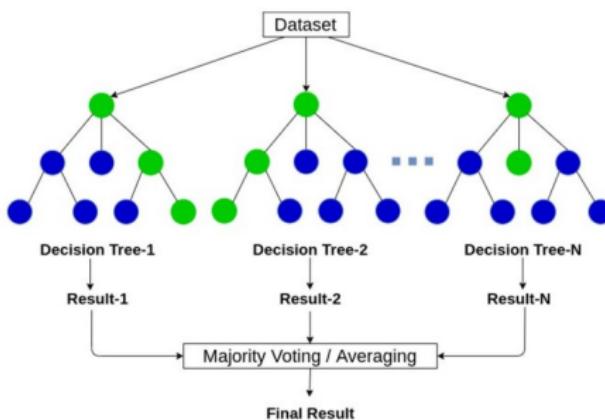
# Árvore de Decisão (Decision Trees)

- Vários pontos de decisão serão criados (nós).
- Em cada nó, o resultado da decisão será seguir por um caminho, ou por outro.
- Uma pergunta será feita e teremos duas opções de resposta: sim ou não. A opção “sim” levará a uma próxima pergunta, e a opção “não” a outra.



## Floresta Aleatória (Random Forest)

- Técnica de ensemble, ou seja, combina vários modelos para produzir um modelo final mais robusto.
- Constrói múltiplas Árvores de Decisão durante o treinamento e faz a média das previsões para melhorar a precisão e controlar o overfitting.

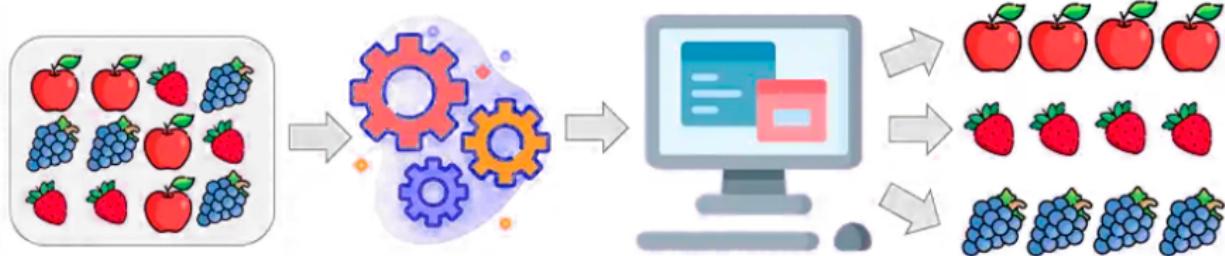


# Aprendizado Não Supervisionado

- Experiência adquirida: Apenas dados de entrada

Não existe supervisor

Dados de entrada

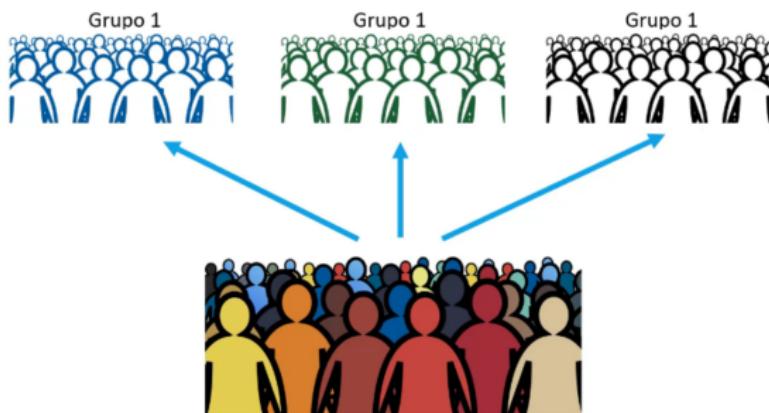


- Análise de agrupamentos;
- Estudos de divergência genética em BAG's.

Visualizar e compreender padrões e tendências dentro de um conjunto de dados

# Agrupamento

- Identificar e agrupar dados semelhantes com base em características.
- O modelo organiza os dados em grupos (clusters) sem rótulos predefinidos.



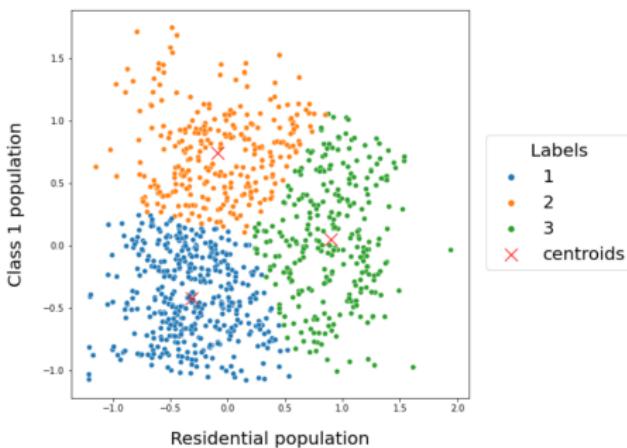
# Agrupamento

## Exemplos:

- **Segmentação de Clientes:** Agrupar clientes com comportamentos similares para personalização de campanhas (ex.: consumidores frequentes e esporádicos).
- **Agrupamento de Imagens:** Organizar imagens com base em semelhanças visuais (ex.: agrupar fotos de paisagens, animais ou objetos).
- **Agrupamento de Dados Geoespaciais:** Agrupar cidades, bairros e regiões com base em características socioeconômicas (renda, mobilidade, infraestrutura) ou ambientais (tipo de solo, cobertura florestal, disponibilidade de água)

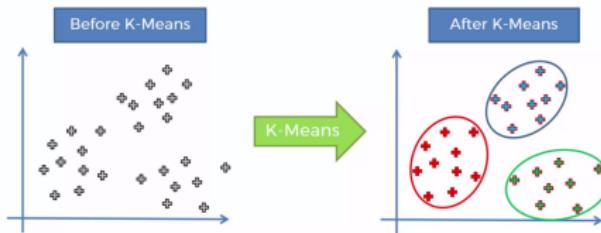
# Algoritmos K-Means

- "K" = Número de centroides (clusters) definidos previamente
- "Means" = Média dos pontos em cada cluster. Determina a posição de seu centroide.



# Algoritmos K-Means

- Os centroides são posicionados na média dos pontos pertencentes ao seu cluster.
- O algoritmo encontra o centroide mais próximo a um novo ponto.
- O ponto é então classificado como **pertencente ao cluster** daquele centroide.



# Métricas de Avaliação do Agrupamento

## Inércia (Coesão):

- Mede **o quão próximos os pontos estão** dentro de um mesmo cluster.
- Menores valores = maior densidade.
- Menor distância entre os pontos dentro de um cluster = **maior a inércia** e mais **similares** são entre si.

## Separação:

- Avalia a **distância** entre clusters.
- Valores maiores = os clusters estão bem separados.
- Um **bom** agrupamento deve ter **uma alta separação**, significando que os clusters não estão se sobrepondo e são bem distintos entre si.

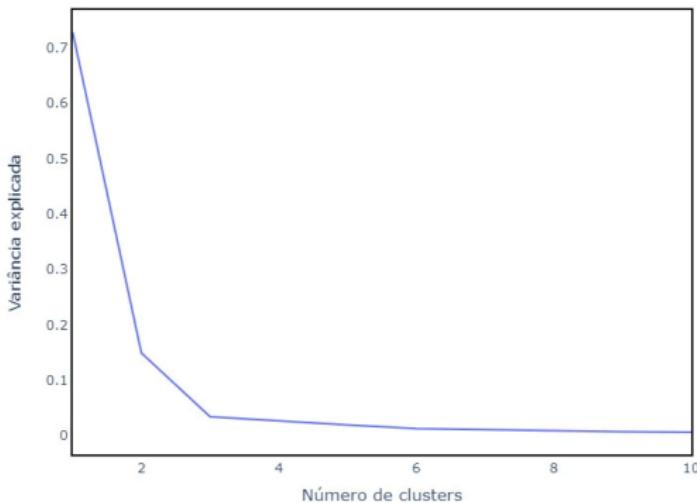
# Coeficiente de Silhueta (Silhouette Score)

- Combina a inércia e a separação para fornecer uma **visão balanceada** da clusterização;
- O valor dessa métrica varia de -1 a 1,
- Valor negativo indica que o ponto pode ter sido atribuído ao cluster errado.

$$\text{Silhueta}_{\text{ponto } i} = \frac{\text{Inercia}_i - \text{Separação}_i}{\max(\text{Inércia}_i, \text{Separação}_i)}$$

## Método do Cotovelo

- Determina o número ideal de clusters identificando o ponto onde a redução na inércia se torna marginal.



# Overfitting (Sobreajuste)

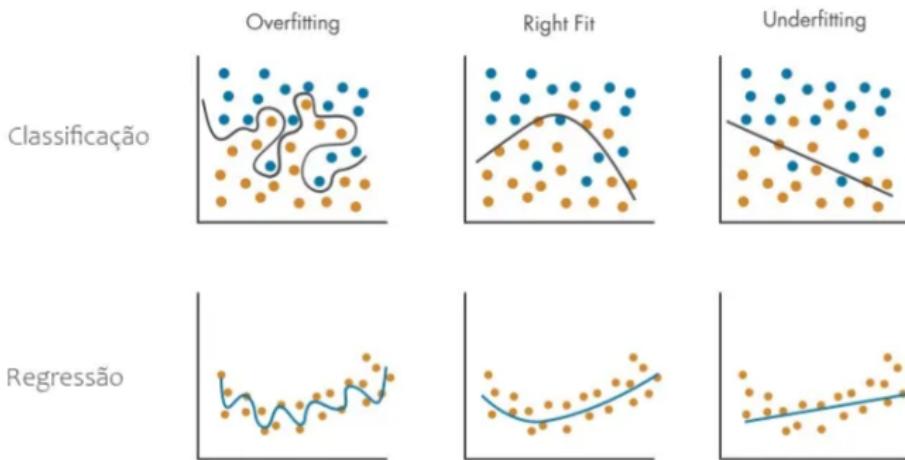
- Previsões precisas para dados de **treinamento**, mas não para **novos dados**.

## Alguns motivos:

- O tamanho dos dados de treinamento é muito pequeno.
- Dados de treinamento com grandes quantidades de informações irrelevantes.
- O modelo treina por muito tempo em um único conjunto de dados.
- A **complexidade** do modelo é **alta**, então ele aprende o ruído nos dados de treinamento.

# Underfitting (Subajuste)

- O modelo não pode determinar uma relação significativa entre os dados de entrada e saída.
- Fornecem resultados imprecisos tanto para os **dados de treinamento** quanto para o **conjunto de testes**.



# Validação de Modelos

A fase de testar o nosso modelo!

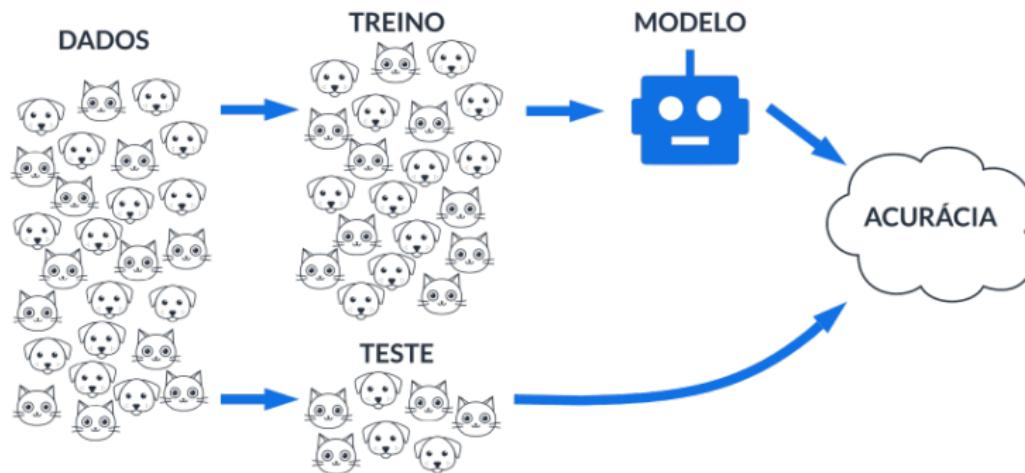
- A validação de modelos avalia o desempenho de um modelo em dados não vistos durante o treinamento.
- Garante que o modelo generalize bem para dados do mundo real.
- Evita problemas como overfitting (superajuste) e underfitting (subajuste).

## Técnicas comuns:

- Divisão Treino/Validação/Teste (Hold-out)
- Validação Cruzada (Cross-Validation)

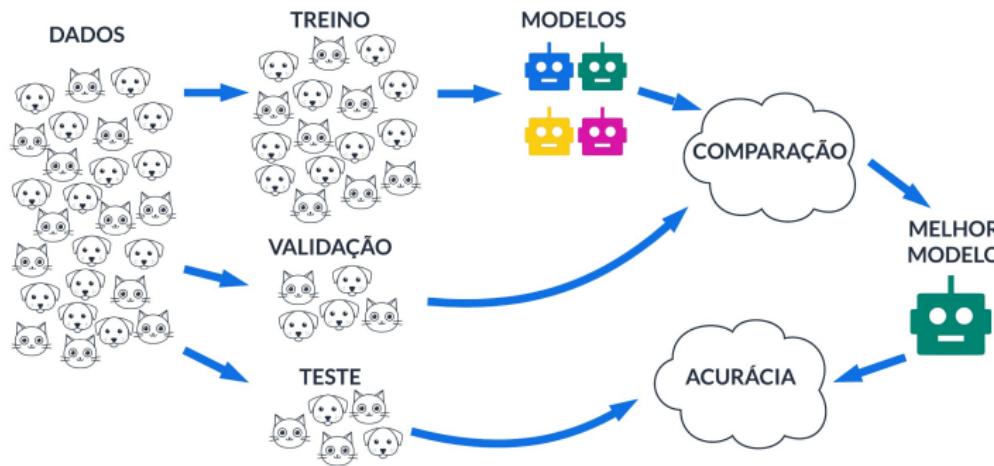
# Hold-Out

- Divide o dataset, separando os dados para o Treinamento e para o Teste do modelo.



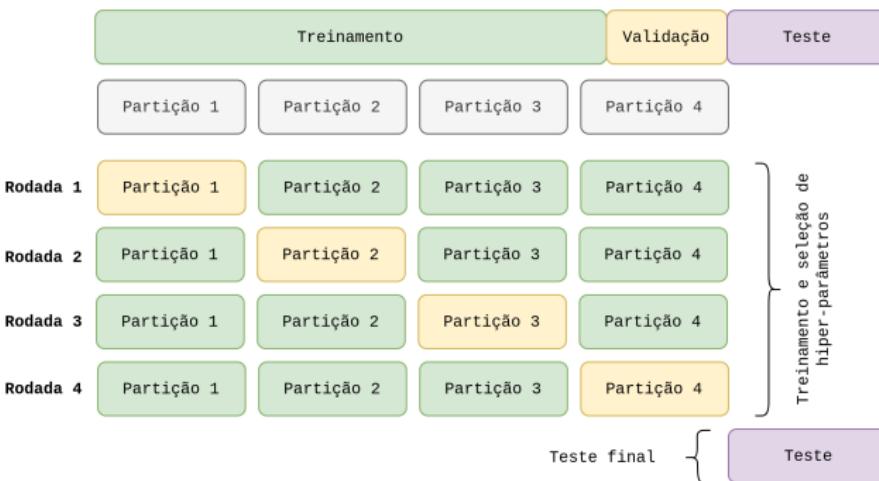
# Hold-Out

- Para modelos mais complexos os dados de treinamento são novamente divididos entre Treino e Validação, para ser possível fazer ajustes no modelo sem utilizar dos dados de Teste.



# Validação Cruzada (K-fold)

- Divide os dados em múltiplos subconjuntos (folds).
- Treina o modelo em  $k - 1$  folds e testa no fold restante.
- Processo repetido  $k$  vezes, garantindo que todos os dados sejam usados para treino e teste.





Obrigado pela atenção!