

# Machine Learning

ESPECIALIZAÇÃO EM CIÊNCIA DE DADOS



Vinícius Rodrigues Oviedo

Santa Maria, 2025

# O que é Machine Learning

## O que é Machine Learning

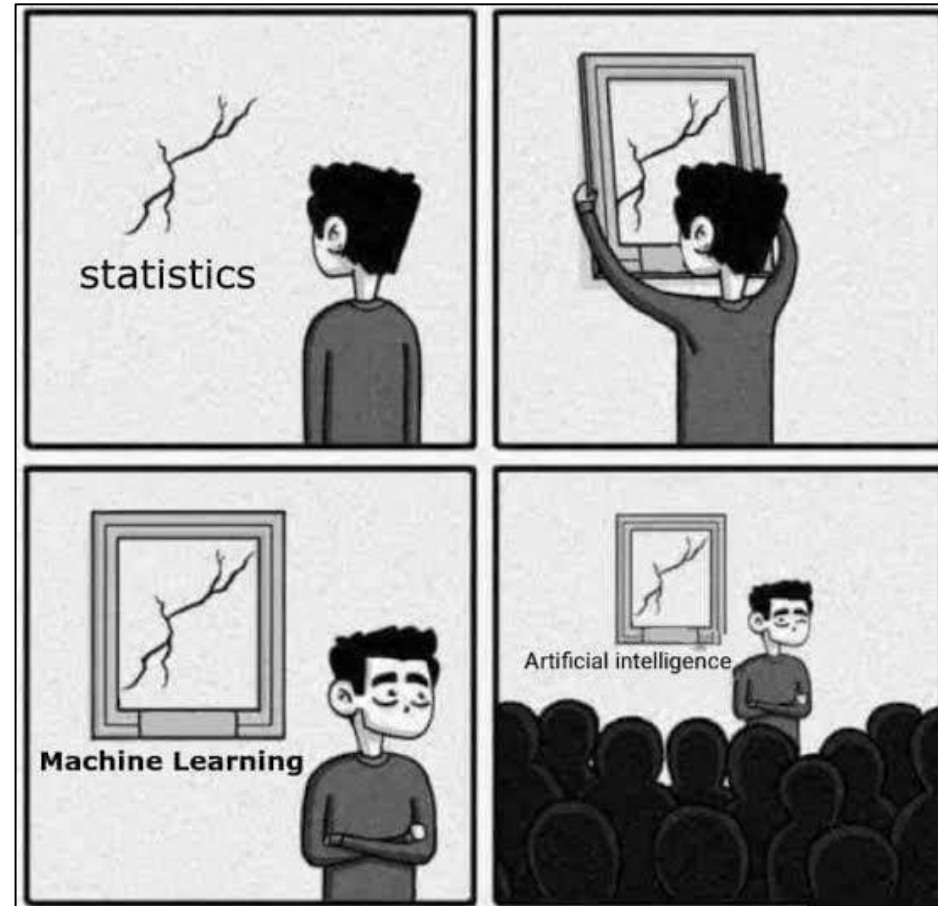
[Hands On Machine Learning with Scikit Learn and TensorFlow](#) (GÉRON, 2017):

Ciência de programar computadores de forma que eles aprendam com os dados.

[Some Studies in Machine Learning Using the Game of Checkers](#) (SAMUEL, 1959)

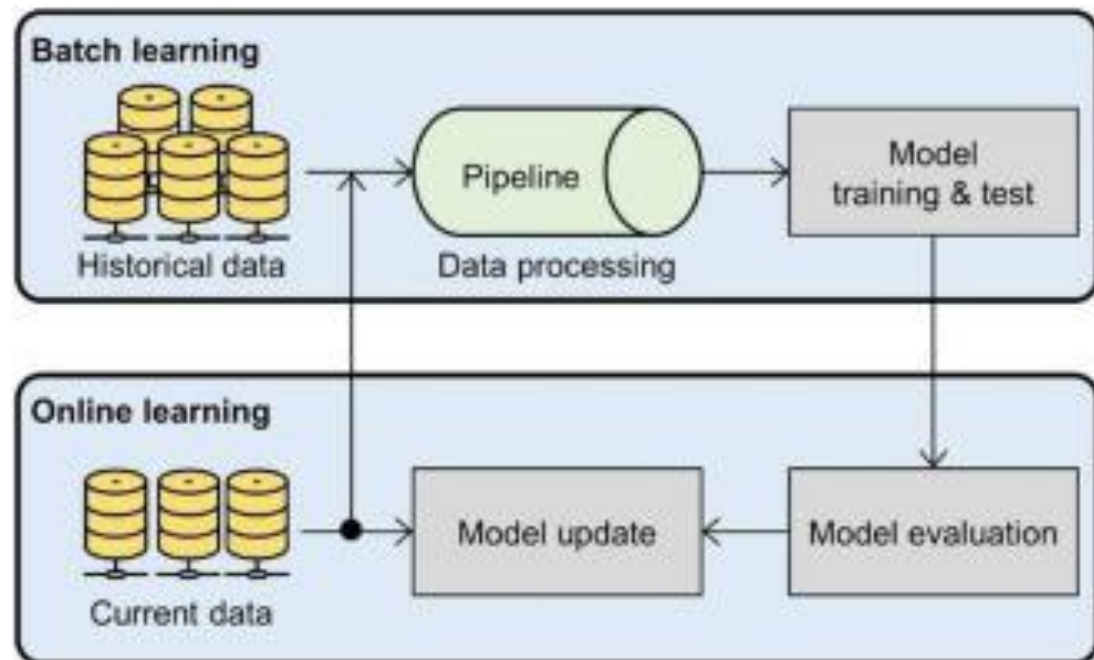
Machine Learning é o campo de estudo que dá aos computadores a capacidade de aprender sem ser explicitamente programado.

# O que é Machine Learning



# Tipos de aprendizado

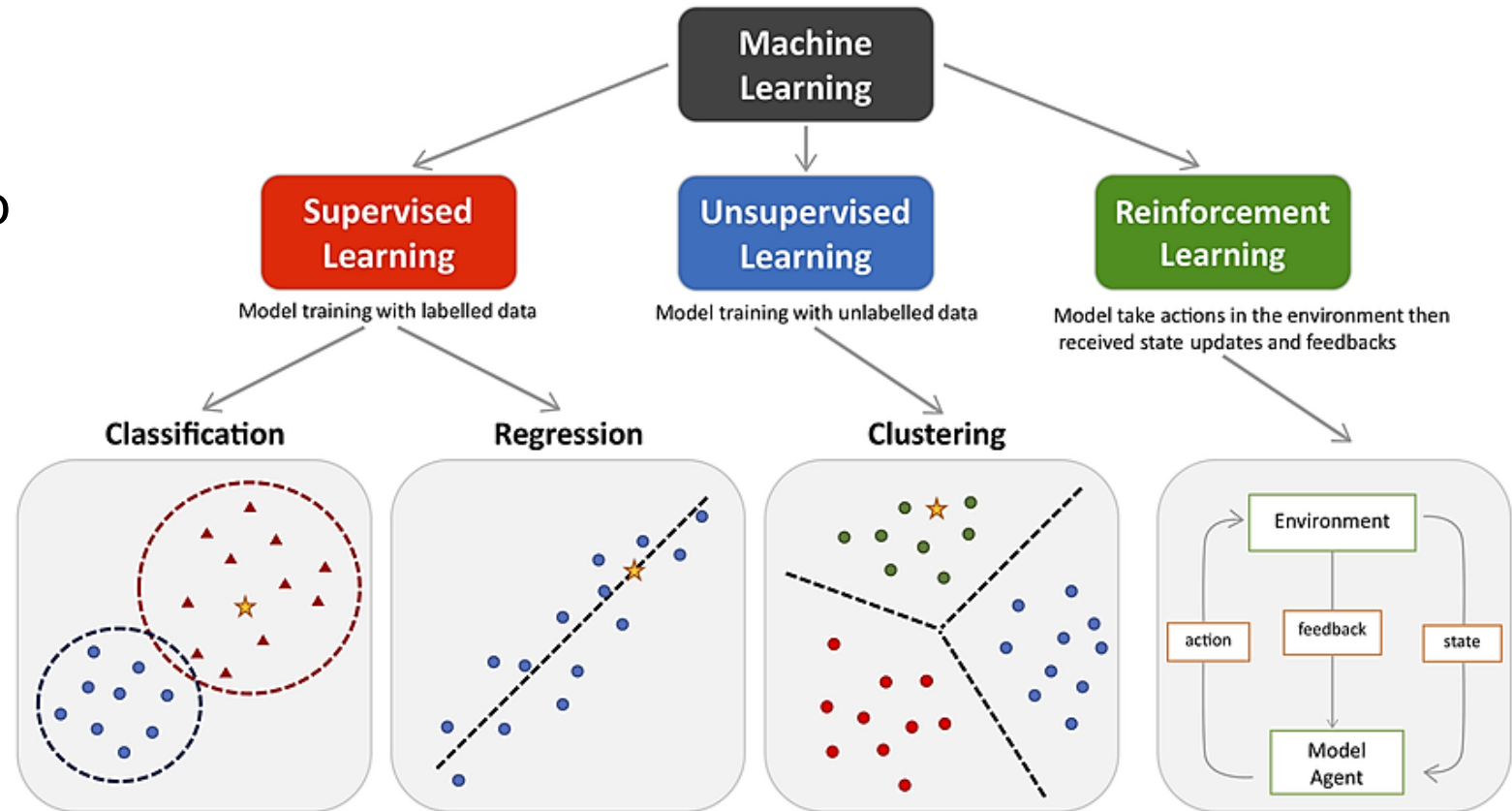
- *Batch* (em lote)
- Online/Real Time



DOI: [10.1016/j.mlwa.2023.100505](https://doi.org/10.1016/j.mlwa.2023.100505)

# Tipos de aprendizado

- Supervisionado
- Não-supervisionado
- Por reforço



DOI: [10.3389/fphar.2021.720694](https://doi.org/10.3389/fphar.2021.720694)

# Problemas

- Regressão
- Classificação
- Agrupamento

**Gradiente descendente (Gradient Descent):**

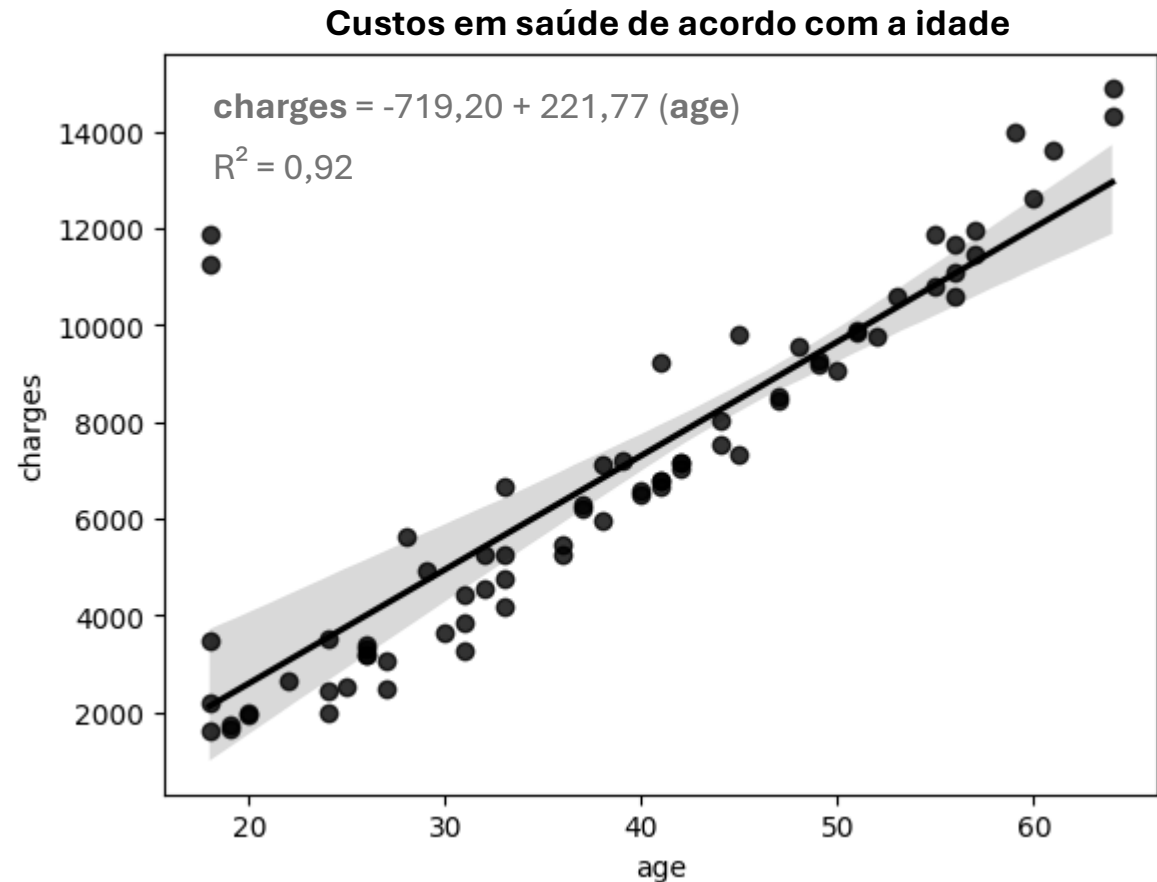
<https://www.ibm.com/br-pt/topics/gradient-descent>

## Aplicações

- Aplicações de crédito e seguros
- Imóveis (*real estate*)
- Detecção de fraudes
- Detecção de anomalias
- Segmentação de clientes
- Sistemas de recomendação
- Predição de *Turnover*
- Predição de *Churn*
- Previsão de demandas
- Precificação
- Diagnósticos
- Reconhecimento de imagem
- Séries temporais
- Experimentos

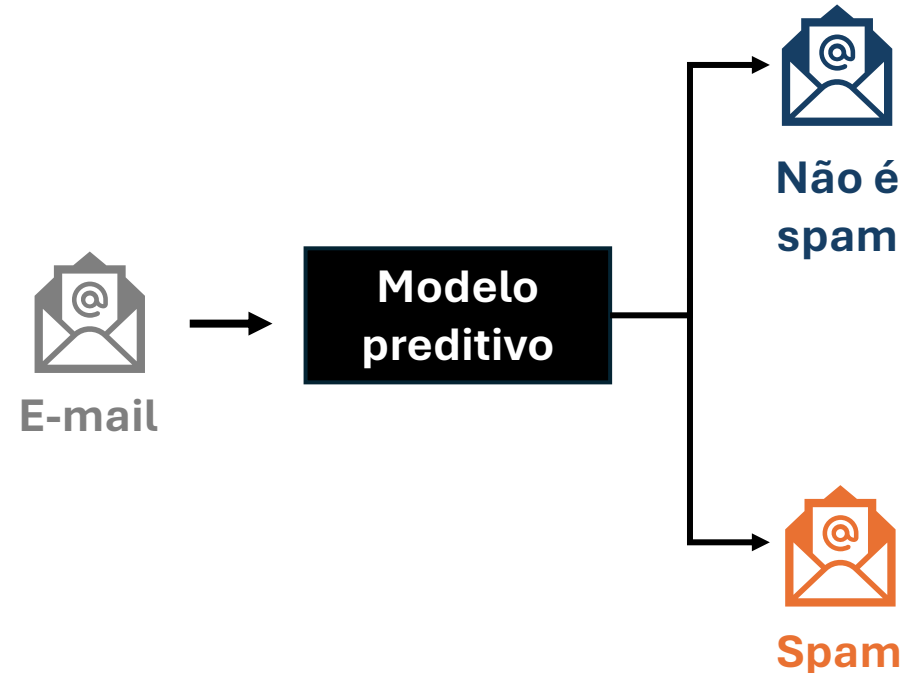
# Regressão

- Variáveis preditoras (*features*) e resposta (*target*)
- Resposta: *target* numérico
- Linear, múltipla, polinomial
- **Algoritmos:**
  - Regressão Linear Simples
  - Regressão Linear Múltipla
  - DT Regressor, RF regressor
  - Ridge, Lasso (Regularizações)



# Classificação

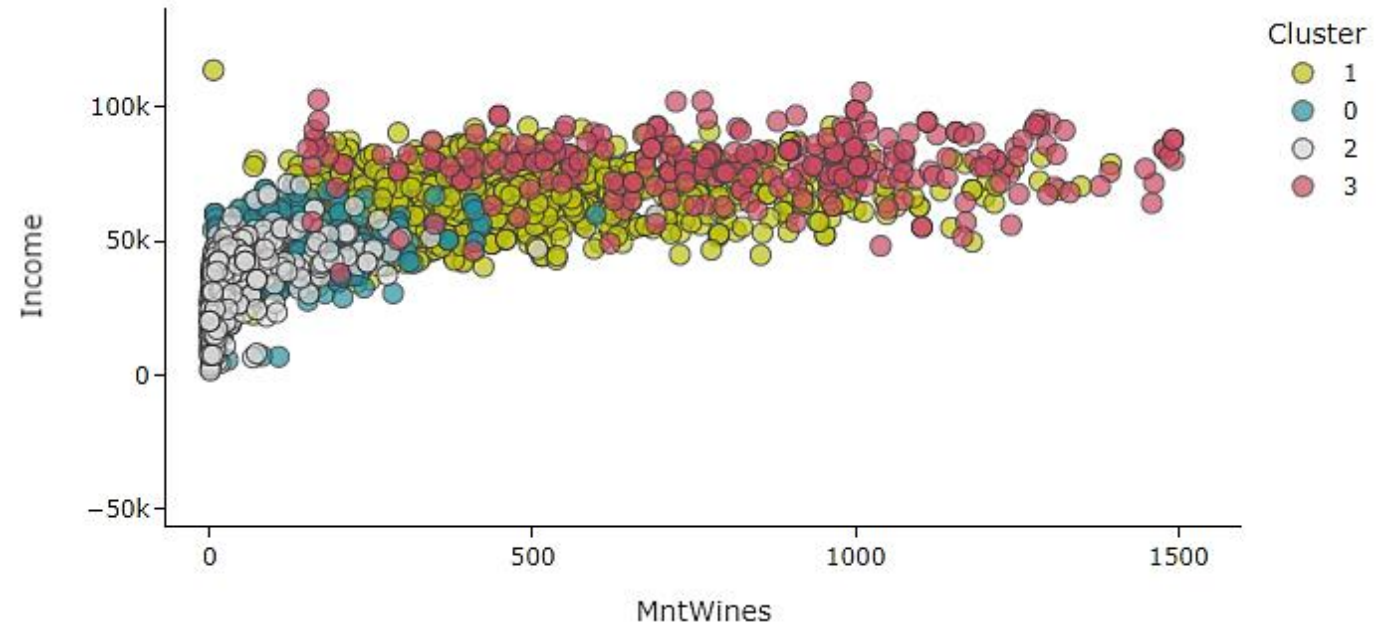
- Variáveis preditoras (*features*) e resposta (*target*)
- Resposta: classes (qualitativo)
- Tipos: binário, multi-classe
- **Algoritmos:**
  - Regressão Logística
  - Naïve-Bayes
  - Decision Tree
  - Random Forest
  - Redes Neurais Artificiais





# Agrupamento

- Não se tem uma variável resposta (*target*)
- Dados brutos → **grupamentos**
- **Algoritmos:**
  - K-means
  - K-modes
  - Clusterização hierárquica
  - DBSCAN
  - Modelos de Misturas Gaussianas (GMM)
  - Autoencoders (redes neurais)



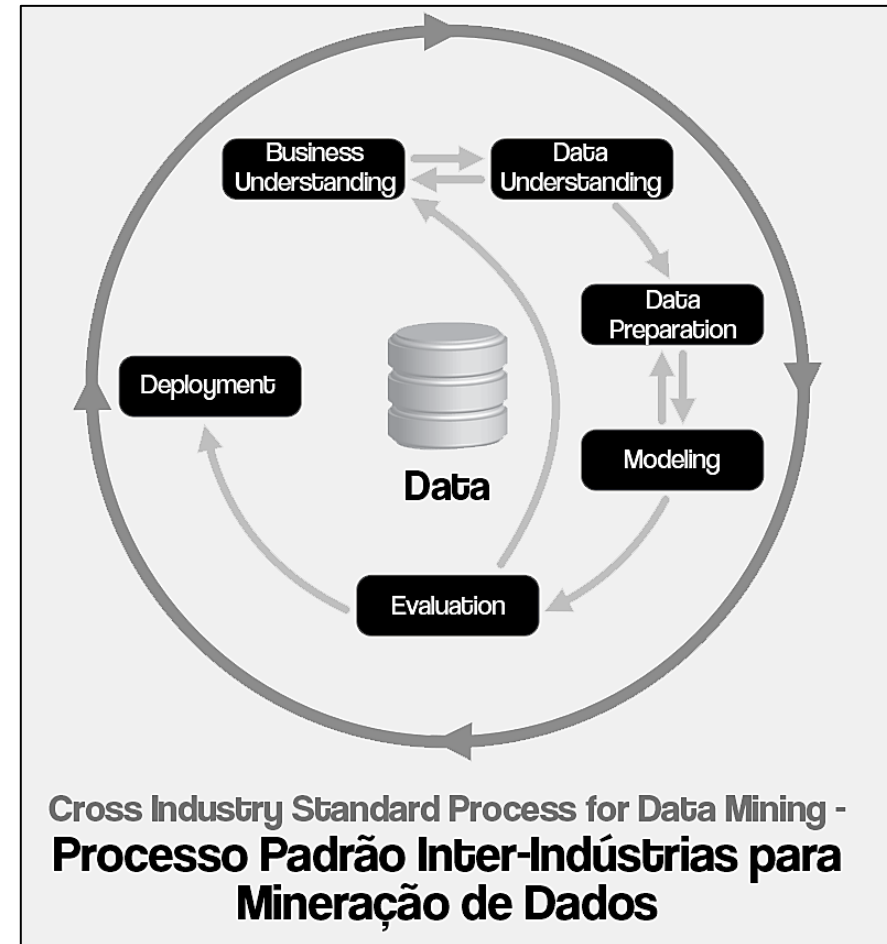
# METODOLOGIA

- **CRISP-DM**
- Metodologias ágeis



## Pré-processamento dos dados:

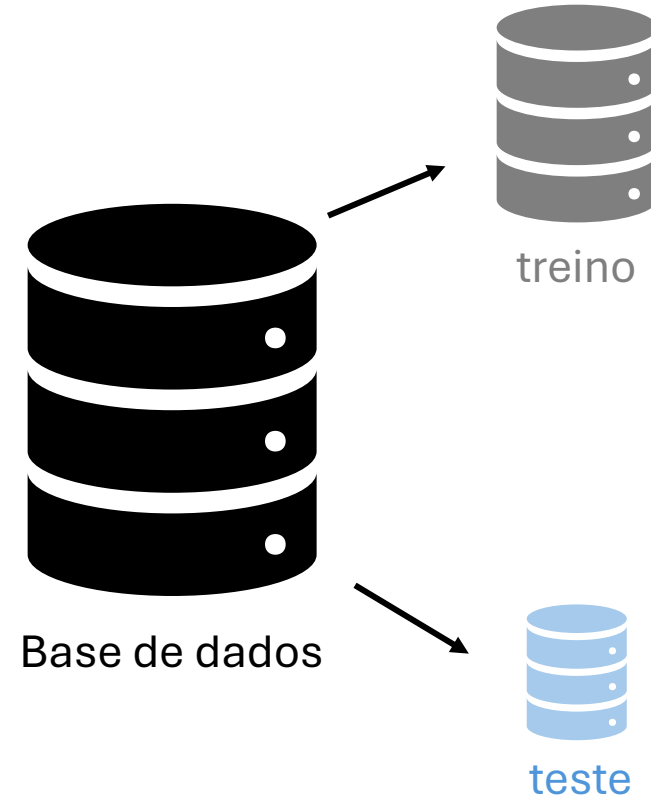
- Análise Exploratória de Dados (EDA)
- Tratamento de nulos (dados ausentes)
- Tratamento de *outliers*
- Normalização
- Codificação de variáveis categóricas
- Desbalanceamento (classificação)



Fonte: google imagens

# DADOS DE TREINO E TESTE

- Validação do modelo de Machine Learning (LM)
- **Exemplos:** 80% treino, 20% teste
- Amostragem estatística
- Scikit-Learn: `train_test_split`

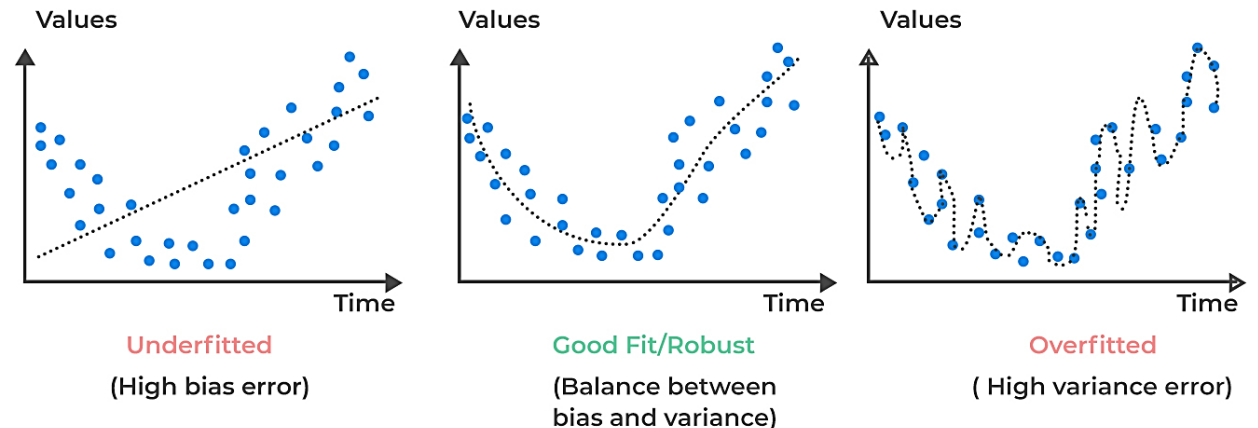


# UNDERFITTING E OVERFITTING

- Giram em torno da generalização do modelo
- Underfitting: modelo muito simples
- Overfitting: modelo performa muito bem no treino, mas que generaliza mal para novos dados (“**decore**” os dados de treino)



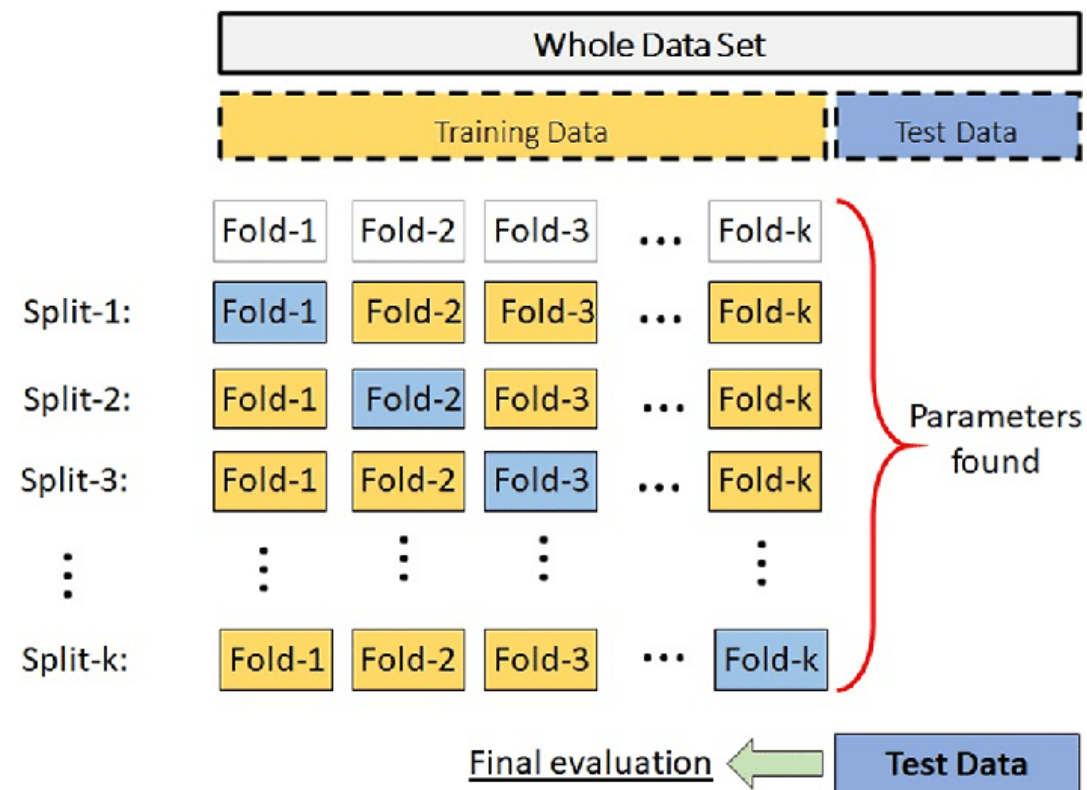
## Generalization and Overfitting



Fonte: <https://analystprep.com/study-notes/cfa-level-2/quantitative-method/overfitting-methods-addressing/>

# VALIDAÇÃO CRUZADA

- Reduzir *underfitting*
- Reduzir *overfitting*
- *Hold out*
- *Validação K-fold*
- *Stratified K-Fold*



DOI: <http://dx.doi.org/10.1016/j.cie.2021.107912>

# HIPER-PARÂMETROS

- **Foco:** melhorar desempenho do modelo de ML
- Evitar *under/overfitting*
- Performance computacional
- Manual
- Métodos automatizados:
  - Grid Search
  - Random Search
  - Bayesian Search
  - Optuna
- **AutoML:** [pyCaret](#)

- Exemplos:
  - Intercepto (regressão)
  - Número de árvores em RF
  - Clusters iniciais no K-means
  - Taxa de aprendizado, batch size, número de camadas ocultas, neurônios por camada em uma Rede Neural
  - Profundidade máxima em DT

# DEPLOY

- API
- Bots
- Plano de ação
- Dashboard (Looker, PBI, Tableau)
- Interface Web:
  - **Streamlit**
  - **Dash (plotly)**
  - **Flask**
  - **Django**
  - **Taipy**

Web App com Streamlit – **previsão de churn.**

The screenshot shows a Streamlit web application interface. On the left is a sidebar with navigation links: Home, Data, Dashboard, Predict (highlighted), and History. Below these is a Logout button. The main content area contains a form with several input fields and dropdown menus. The form fields include: Paperless Billing (Yes), Payment Method (Credit card (automatic)), Monthly Charges (\$) (0,00), Total Charges (\$) (50,00), Tech Support (Yes), Streaming TV (No), Streaming Movies (Yes), and Contract (Month-to-month). A Submit button is at the bottom of the form. Below the form, the text reads: "Customer will leave 😞 ." and "Probability: 63.63%".

Fonte: <https://medium.com/@briankimagut/building-streamlit-machine-learning-app-220249e573de>

# SUGESTÕES DE REFERÊNCIA

- *Introduction to Machine Learning with Python: A Guide for Data Scientists* ([link](#))
- *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow* ([link](#))
- Medium
- Linkedin
- Artigos científicos