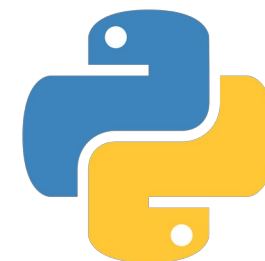




Uniube

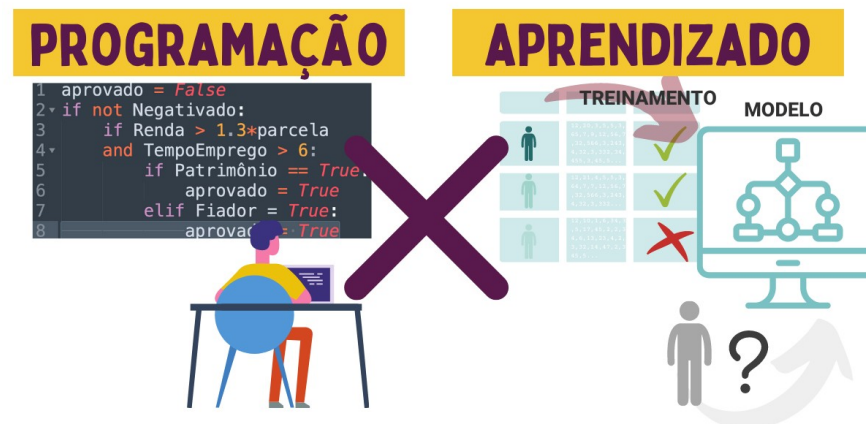
# Aprendizado de Máquina



Prof. Me. Clênio Silva  
clenio.silva@uniube.br

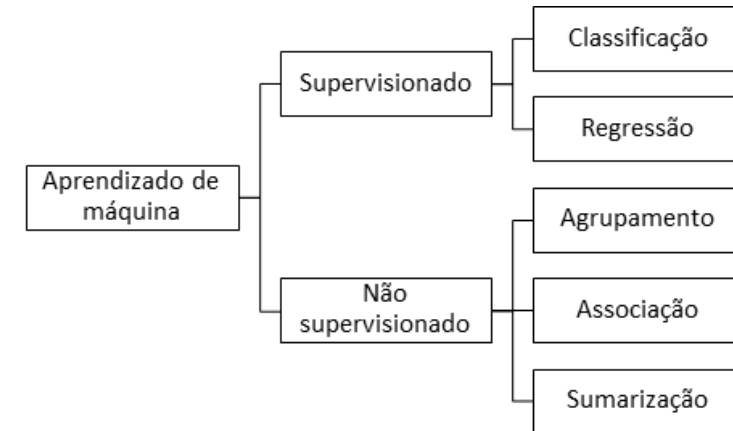
# O que é Aprendizado de Máquina

- É um ramo da inteligência artificial que permite que sistemas aprendam e façam previsões ou decisões em dados, sem serem explicitamente programados para cada tarefa.
  - Algoritmos identificam padrões a partir de conjuntos de dados



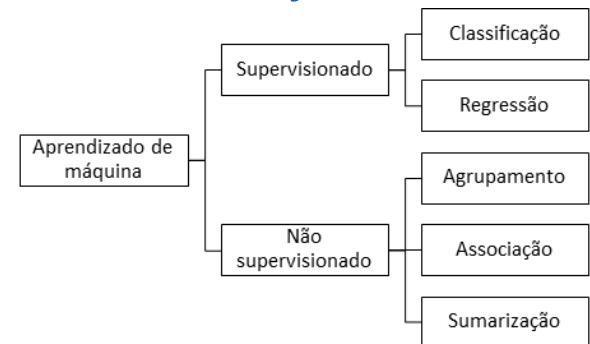
# Tipo de Aprendizado

- **Aprendizado Supervisionado:**
  - O modelo é treinado com um conjunto de dados rotulado, onde cada entrada é associada a uma saída conhecida.
  - **Exemplos:**
    - **Classificação:** Identificar se um e-mail é ou não spam com base em características como palavras-chave, remetente, etc.
    - **Regressão:** Prever o preço de uma casa com base em suas características, como número de quartos, localização e área.



# Tipo de Aprendizado

- **Aprendizado Não Supervisionado:**
  - O modelo é treinado com dados não rotulados. O objetivo é encontrar padrões ou grupos nos dados sem a orientação de saídas conhecidas.
  - **Exemplos:**
    - **Agrupamento (Clustering):** Agrupar clientes em segmentos com base em seu comportamento de compra, sem saber previamente quais são os grupos.
    - **Redução de Dimensionalidade:** Usar técnicas como PCA (Análise de Componentes Principais) para reduzir o número de variáveis em um conjunto de dados, preservando a maior parte da informação.



# Tipo de Aprendizado

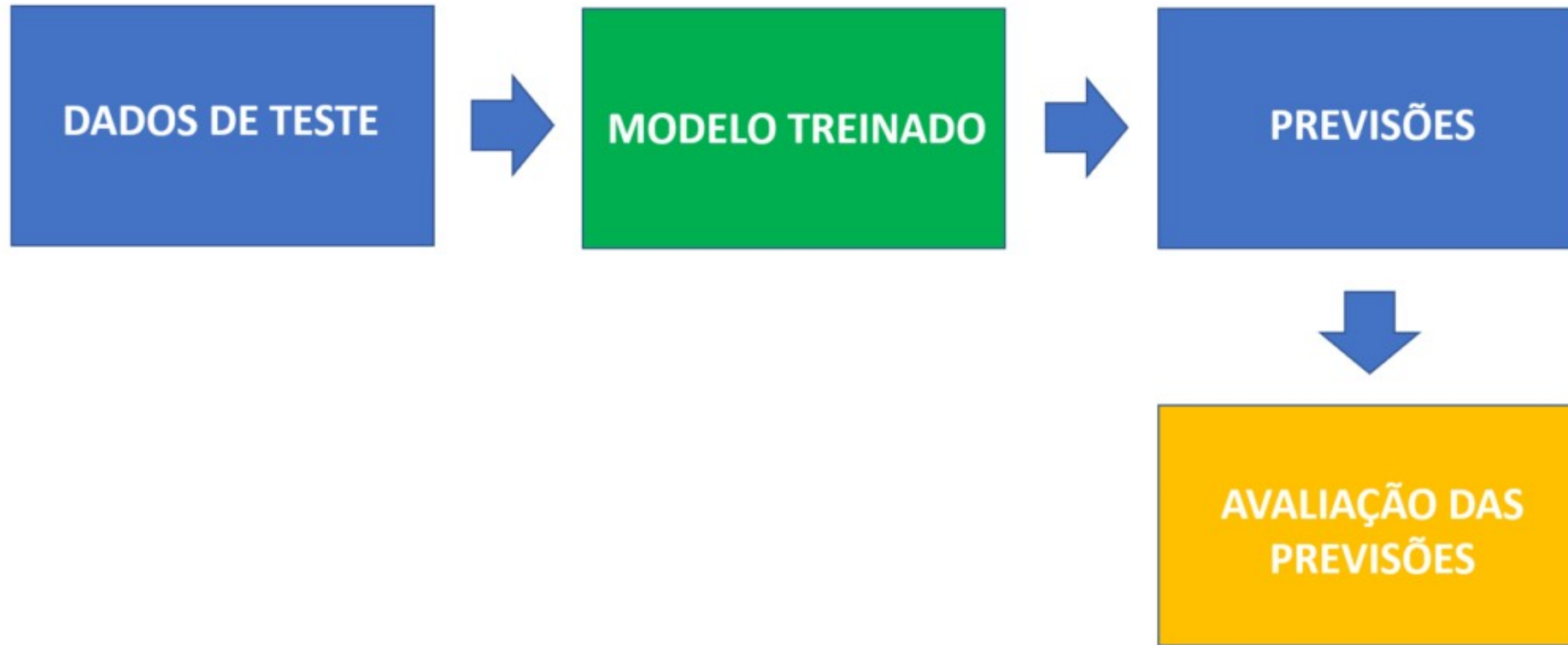
- **Aprendizado por Reforço:**
  - O agente aprende a tomar decisões através de interações com um ambiente. Ele recebe recompensas ou punições com base em suas ações, o que o ajuda a melhorar seu desempenho ao longo do tempo.
  - **Exemplos:**
    - **Jogos:** Um agente treinado para jogar xadrez ou Go, aprendendo a fazer movimentos que levam à vitória com base em recompensas.
    - **Robótica:** Um robô que aprende a navegar em um espaço, recebendo recompensas por alcançar certos pontos e punições por colisões.



# Algoritmos Clássicos

- Regressão Linear
- Regressão Logística
- Árvores de Decisão
- K-Nearest Neighbors (KNN)
- Máquinas de Vetores de Suporte (SVM)
- Redes Neurais

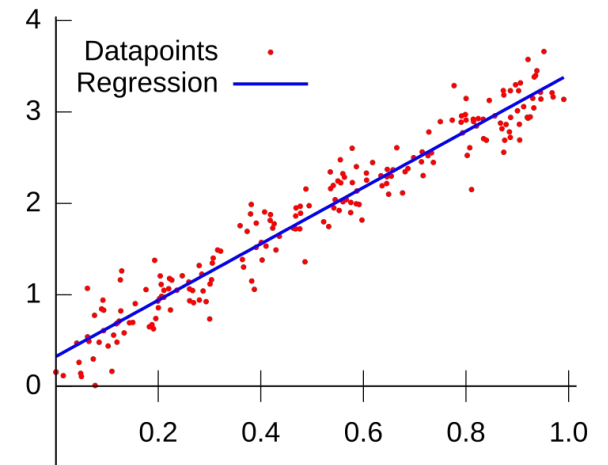
# Onde entra em Ciência de Dados



# Algoritmos de AM

- **Regressão Linear**

- é um método estatístico usado para modelar a relação entre uma variável dependente (ou resposta) e uma ou mais variáveis independentes (ou preditoras).
- A ideia principal é encontrar a melhor linha que se ajusta aos dados de modo que possamos prever a variável dependente com base nas variáveis independentes.





# Algoritmos de AM

- Modelo Matemático
  - Equação:

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \dots + \beta_n x_n + \epsilon$$

onde:

- $y$  é a variável dependente.
- $x_1, x_2, \dots, x_n$  são as variáveis independentes.
- $\beta_0$  é o intercepto (valor de  $y$  quando todas as  $x$  são zero).
- $\beta_1, \beta_2, \dots, \beta_n$  são os coeficientes que representam a variação em  $y$  para cada unidade de mudança em  $x$ .
- $\epsilon$  é o termo de erro, que representa a diferença entre os valores observados e os valores previstos.

Os coeficientes  $\beta$  são estimados usando o método dos mínimos quadrados, que busca minimizar a soma dos quadrados das diferenças entre os valores observados e os valores previstos.

Após ajustar o modelo, é importante avaliar sua performance:  
 $R^2$  (Coeficiente de Determinação): Mede a proporção da variação na variável dependente que é explicada pelo modelo.  $R^2$  varia de 0 a 1; quanto mais próximo de 1, melhor.

Erros Residuais: Analisar os resíduos (diferenças entre valores observados e previstos) ajuda a verificar a adequação do modelo.

Com os coeficientes estimados, é possível fazer previsões para novos dados substituindo os valores das variáveis independentes na equação da regressão.

# Algoritmos de AM

- Código Regressão Linear:

```
# Função para calcular os coeficientes da regressão linear
def calcular_coeficientes(x, y):
    n = len(x)

    # Cálculo das médias
    media_x = sum(x) / n
    media_y = sum(y) / n

    # Cálculo do numerador e denominador para beta_1
    numerador = sum((x[i] - media_x) * (y[i] - media_y) for i in range(n))
    denominador = sum((x[i] - media_x) ** 2 for i in range(n))

    # Coeficientes
    beta_1 = numerador / denominador
    beta_0 = media_y - (beta_1 * media_x)

    return beta_0, beta_1
```

- Calcula os coeficientes da regressão linear usando as fórmulas:

- $\beta_1 = \frac{\sum (x_i - \bar{x})(y_i - \bar{y})}{\sum (x_i - \bar{x})^2}$

- $\beta_0 = \bar{y} - \beta_1 \bar{x}$

# Algoritmos de AM

- Código Regressão Linear:

```
# Função para fazer previsões
def prever(x, beta_0, beta_1):
    return [beta_0 + beta_1 * xi for xi in x]
```

Usa os coeficientes calculados para fazer previsões com base em novas entradas.

```
# Dados de exemplo
areas = [50, 60, 70, 80, 90, 100, 110, 120, 130, 140]
precos = [150000, 180000, 210000, 240000, 270000, 300000, 330000, 360000, 390000, 420000]

# Calcular os coeficientes
beta_0, beta_1 = calcular_coeficientes(areas, precos)

# Fazer previsões
precos_previstos = prever(areas, beta_0, beta_1)

# Resultados
print("Coeficiente (beta_1):", beta_1)
print("Intercepto (beta_0):", beta_0)
print("Previsões:", precos_previstos)

# Visualizar resultados (apenas texto)
for area, preco_real, preco_previsto in zip(areas, precos, precos_previstos):
    print(f"Área: {area} m², Preço Real: R${preco_real}, Preço Previsto: R${preco_previsto:.2f}")
```