# Teste de Regressão Linear

## Parte A – Tarefas Aplicadas / Código

Utilize o conjunto de dados de Diabetes, fornecido pelo scikit-learn, para implementar um modelo de regressão linear, mas começando com uma etapa de exploração de dados (EDA) antes do ajuste do modelo.

### Contexto do Conjunto de Dados

Neste exame, utilizaremos o **conjunto de dados Diabetes**, disponível na biblioteca scikit-learn.

Esse conjunto foi construído a partir de um estudo médico real sobre pacientes com diabetes. Ele contém **442 observações** (linhas), cada uma representando um paciente, e **10 variáveis explicativas contínuas** (features), como idade, sexo, índice de massa corporal (IMC), pressão arterial e diversas medidas bioquímicas do sangue.

O objetivo é prever uma variável alvo (**target**), que é uma medida quantitativa da **progressão da doença um ano após a linha de base**.

Em termos práticos: **cada linha corresponde a um paciente, e o valor do target é um número que indica quão severa foi a progressão da diabetes desse paciente depois de 1 ano, baseado em exames médicos.**

Uma forma simples de entender: no clássico exemplo do **Boston Housing**, o target era o **preço médio de casas**. Já aqui, no dataset **Diabetes**, o target é a **pontuação de progressão da diabetes** — uma variável contínua, que permite aplicar e interpretar modelos de regressão linear de forma realista.

Esse dataset é muito útil porque:

* Possui variáveis em **escalas diferentes**, o que permite discutir normalização.
* Apresenta **multicolinearidade**, o que impacta a interpretação dos coeficientes.//Features corelacionadas com o resultado, porem pode haver mais de um marcadores diferentes possuem a mesma informação, as features precisam ser independentes entre si.
* É um problema biomédico **realista**, em que o modelo linear consegue capturar parte da variabilidade, mas não explica tudo, deixando espaço para reflexão crítica sobre as limitações do método.

1. Carregue o conjunto de dados:

- Use `from sklearn.datasets import load\_diabetes`.  
- Crie um DataFrame do pandas com todas as features e o alvo.

2. Exploração de Dados (EDA):

- Calcule a matriz de correlação (features vs alvo).  
- Identifique quais features são mais fortemente correlacionadas com o alvo.  
- Plote a distribuição de valores por feature (ex.: histogramas).  
- Pergunta: Os atributos estão em escalas comparáveis? Seria necessário normalizar/re-escalar as features? Por quê?

3. Modelo inicial (todas as features):

- Divida o dataset em treino e teste (80/20).  
- Ajuste um modelo LinearRegression com todas as features.  
- Reporte: coeficientes de cada feature, intercepto, R² no treino e teste.

4. Seleção de features com correlação:

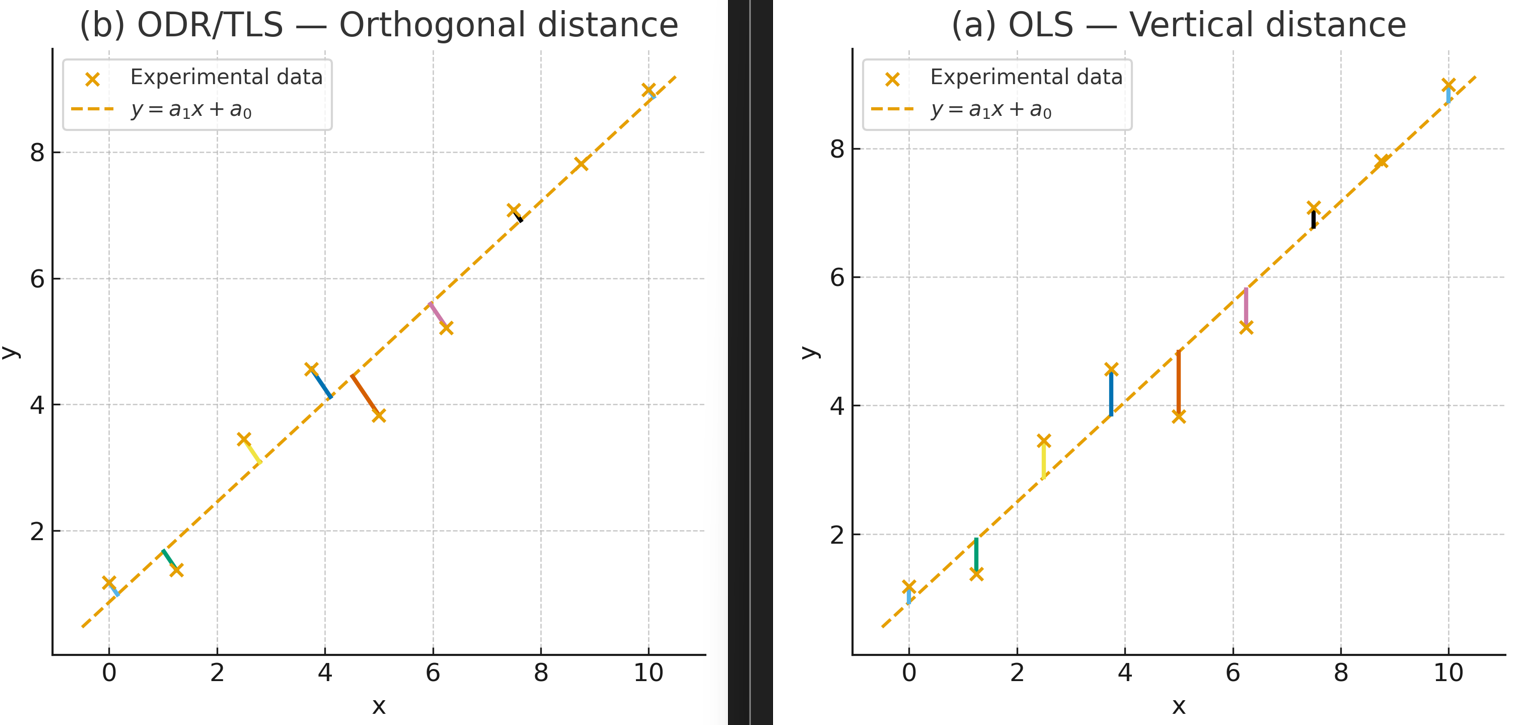
- Elimine pelo menos 2 features com menor correlação com o alvo.  
- Reajuste o modelo de regressão linear.  
- Reporte novamente os coeficientes, intercepto e R² no treino e teste.

5. Comparação e Reflexão:

- Compare o desempenho do modelo com todas as features versus o modelo reduzido.  
- Qual modelo teve melhor performance no conjunto de teste?

## Parte B – Regressão com Uma Feature e Função de Perda Alternativa

//Alterar o código do dia 22 de agosto – O do código do dia 22 é uma distancia vertial e da tarefa é uma distância ortogonal. Por isso no código eu preciso mudar o calculo da distância do erro.



Escolha uma única feature do dataset de Diabetes e compare dois ajustes de regressão linear:

1. Regressão Linear Ordinária (OLS):3

2. Regressão por Distância Ortogonal (ODR/TLS):

- OLS minimiza distâncias verticais. ODR minimiza distâncias perpendiculares à reta.  
- O scikit-learn não permite alterar a função de perda. Adapte o código 3ESPR\_28\_08.py que se encontra na aba arquivos do nosso canal no teams. A função erro deve ser alterada. A reta é:

y=mx+b

então sua forma geral é

mx−y+b=0

e a distância ortogonal de um ponto (x0,y0) até a reta é:

 --Mudar o código da distância do erro por esse código.

3. Discussão:

* - Em que situações OLS e ODR resultam em retas significativamente diferentes?  
  - Por que ODR é mais adequado quando tanto X quanto Y têm erro de medição?  
  - Como a escala/unidades das variáveis afetam a solução ODR (e por que padronizar é útil)?  
  - Qual modelo parece generalizar melhor e segundo qual critério?