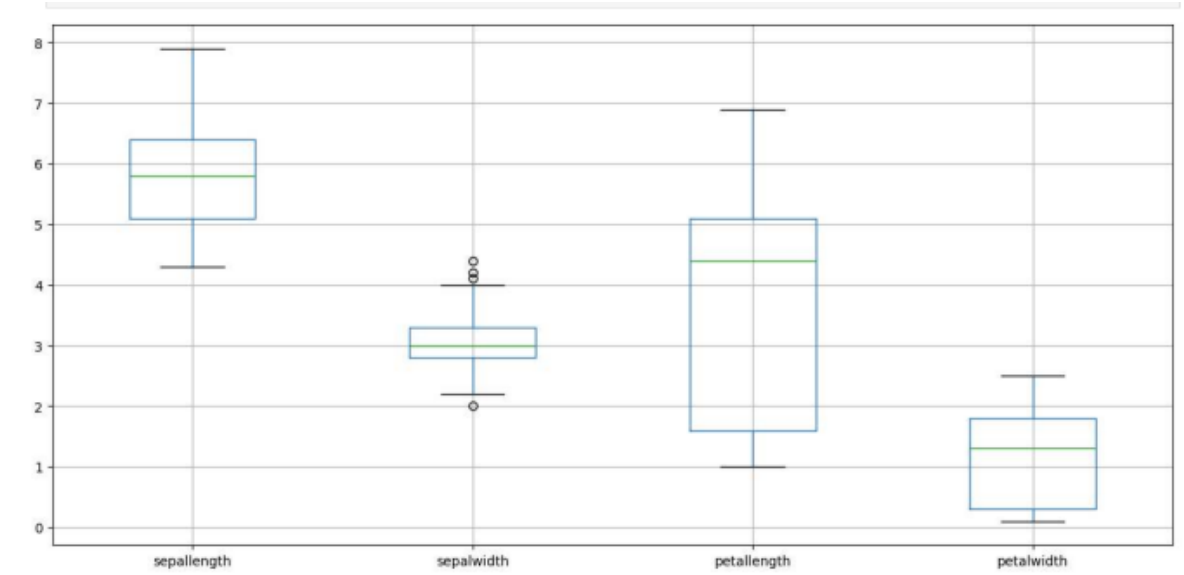
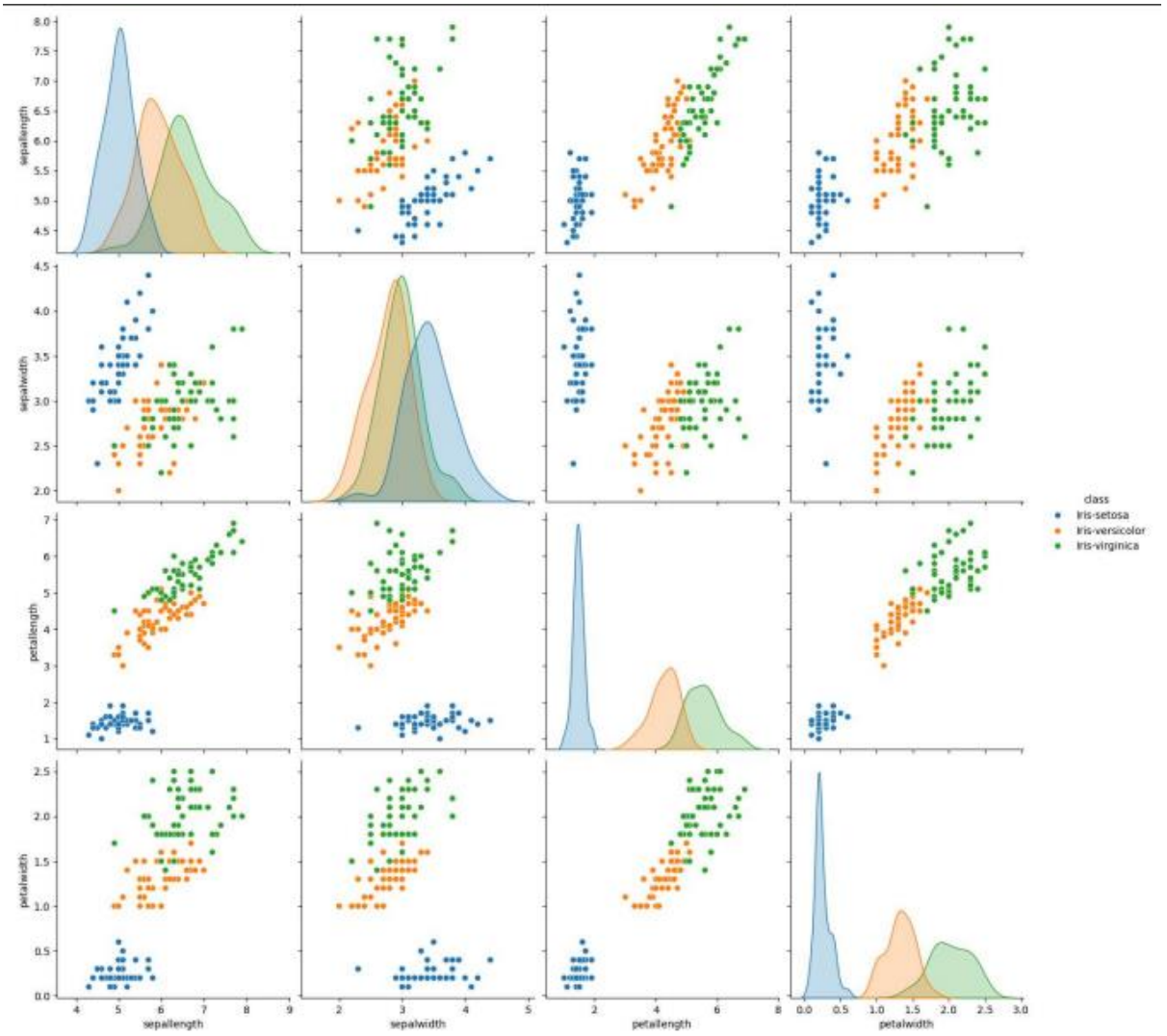
Lista IA

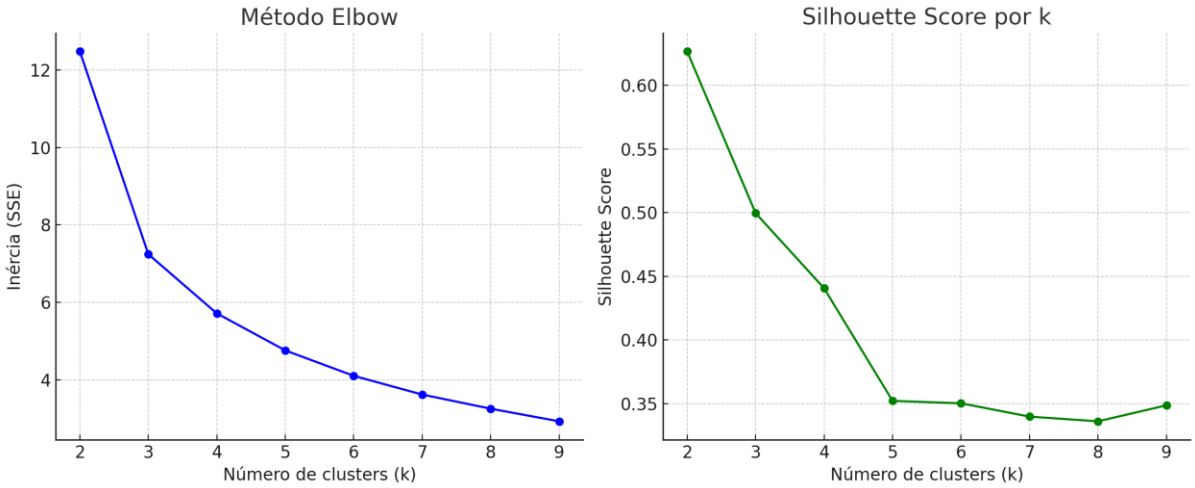
1.Boxplot e Scatter





2. Gráficos do Método Elbow (esquerda) e do Silhouette Score (direita):

- O Elbow mostra uma queda acentuada até k=3, sugerindo que 3 é um bom número de clusters (coerente com o dataset Iris).
- O Silhouette Score também atinge um valor alto em k=3, confirmando a escolha.



3. Hiperparâmetros do KMeans

Principais hiperparâmetros:

- n_clusters: número de grupos a ser encontrado (k).
- init: método de inicialização dos centróides. Pode ser:
  - `'k-means++'`: escolha inteligente dos centróides iniciais (padrão e mais eficaz).
  - `'random'`: escolha aleatória dos centróides iniciais.
- max_iter: número máximo de iterações.
- tol: tolerância para declarar convergência.
- random_state: fixa a semente aleatória para reprodutibilidade.

Métricas de distância utilizadas:

- Distância Euclidiana (mais comum).
- Outras possíveis, dependendo da implementação (como distância de Manhattan), mas o `KMeans` do scikit-learn usa a euclidiana por padrão.

4. Equações das métricas

Distância Euclidiana

A distância entre dois pontos $x=(x1,x2,...,xn)x = (x_1, x_2, ..., x_n)$x=(x1 ,x2 ,...,xn) e $y=(y1,y2,...,yn)y = (y_1, y_2, ..., y_n)$y=(y1 ,y2 ,...,yn) é:

$d(x,y)=\sum i=1n(xi-yi)2d(x, y) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}$d(x,y)=i=1∑n (xi −yi)2

Distância de Manhattan

(Se quiser testar outra métrica em outro algoritmo)

$d(x,y)=\sum i=1n|xi-yi|d(x, y) = \sum_{i=1}^{n} |x_i - y_i|$d(x,y)=i=1∑n |xi −yi |

5. Outra métrica de avaliação de agrupamentos

Diferente das métricas de distância, aqui falamos de métricas de avaliação, como:

- Silhouette Score (já conhecida)
- Davies-Bouldin Index
- Calinski-Harabasz Index → implementaremos essa

Calinski-Harabasz Index

Mede a razão entre a dispersão entre os clusters e a dispersão interna:

$CH=Tr(Bk)Tr(Wk)\cdot n-kk-1CH = \frac{\text{Tr}(B_k)}{\text{Tr}(W_k)} \cdot \frac{n - k}{k - 1}$CH=Tr(Wk)Tr(Bk) ·k−1n−k

- BkB_kBk : variância entre os centróides dos clusters
- WkW_kWk : variância dentro dos clusters
- nnn: número total de amostras
- kkk: número de clusters

Quanto maior o índice, melhor a separação dos clusters.

6. DBSCAN e SOM — comparação com KMeans

DBSCAN

- Não precisa definir o número de clusters.
- Parâmetros: `eps` (raio de vizinhança) e `min_samples`.
- Detecta outliers.

SOM (Self-Organizing Maps)

- Inspirado em redes neurais.
- Agrupa de forma topológica.

Após aplicar os três, você pode comparar o número de clusters encontrados. Se forem diferentes, analise:

- DBSCAN pode encontrar menos clusters (e detectar ruído).
- SOM pode ser sensível à configuração da grade (mapa 2D).

7. Visualização de erros do KMeans

Como o dataset possui rótulos verdadeiros, podemos visualizar:

- Quais amostras foram agrupadas incorretamente pelo KMeans
- Usando PCA ou t-SNE para reduzir a dimensionalidade para 2D
- Colorir os pontos por `rótulo verdadeiro` e por `cluster atribuído`

Isso permite visualizar quais pontos foram classificados errado.

8. Relatório

O relatório deve conter:

- Pré-processamento:
    - Normalização ou padronização dos dados
    - PCA (se aplicado)
- Execução do KMeans, DBSCAN e SOM
    - Parâmetros utilizados
    - Número de clusters encontrados
- Avaliação
    - Silhouette Score, Calinski-Harabasz, etc.
    - Visualização dos erros do KMeans
- Discussão
    - KMeans vs DBSCAN vs SOM
    - Quais agruparam melhor os dados reais
    - Erros e limitações

Github.com/N4lberth/IA