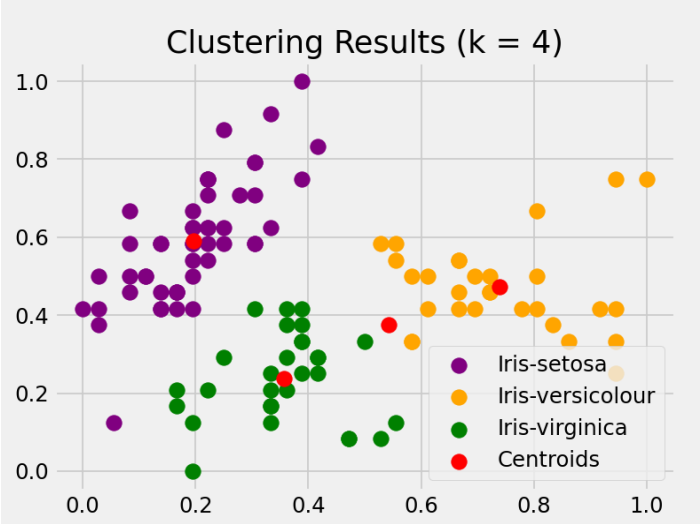
Pedro Olyntho Carvalho Alves

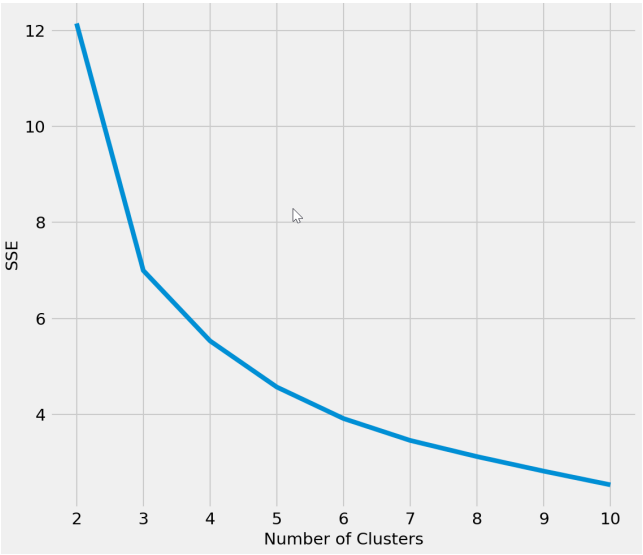
Lista 11: K Means

1.

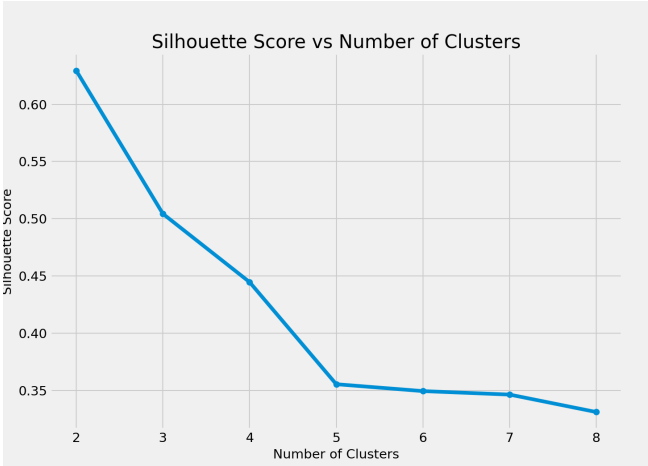
Agrupamentos encontrados (utilizando o número de clusters definido pela métrica elbow):



Valores para a métrica elbow:
- O número de clusters definido pelo método foi 4, o que representa o ponto de inflexão da curva (elbow), indicando um bom número de centróides.



Valores para a métrica silhouette:



Já pela métrica silhouette, o número de clusters ideal encontrado é 2, pois apresenta o valor mais próximo de 1. Valores próximos de 1 indicam um bom agrupamento, enquanto aqueles próximos de -1 indicam que as instâncias estão atribuídas a clusters incorretos.
https://github.com/pedroolynth0/LISTA11/blob/main/questao1.py

2.

- **Métrica Silhouette:** É uma medida para medir o quanto os objetos se encaixam em seu próprio cluster em relação aos demais A fórmula da Silhouette para um objeto individual é calculada da seguinte maneira:

    - Para um objeto 'i' em seu próprio cluster:

  a(i) = média da distância entre 'i' e todos os outros objetos dentro do mesmo cluster (distâncias intra-cluster).

    - Para um objeto 'i' em relação a outros clusters:

  b(i) = média da distância entre 'i' e todos os objetos de um cluster diferente mais próximo (distâncias inter-cluster).

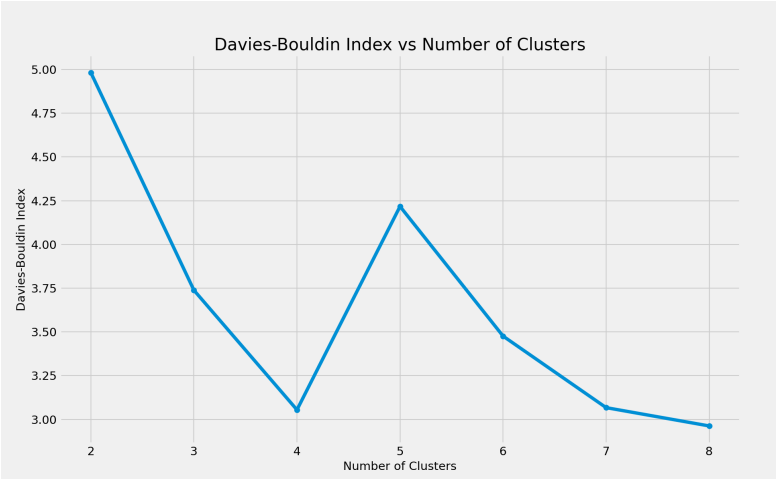
$$a(i) = \frac{1}{|C_I| - 1} \sum_{j \in C_I, i \neq j} d(i, j)$$

- **Elbow:** É utilizado para determinar um número ideal de clusters avaliando a variação explicada pelos clusters em relação ao número de clusters .O mesmo método pode ser usado para escolher o número de parâmetros em outros modelos baseados em dados, como o número de componentes principais para descrever um conjunto de dados.
  É importante observar que o método do cotovelo é uma heurística e pode não fornecer uma resposta definitiva em todos os casos. É necessário usar o conhecimento do domínio e fazer análises adicionais para tomar uma decisão final sobre o número de clusters.

3.

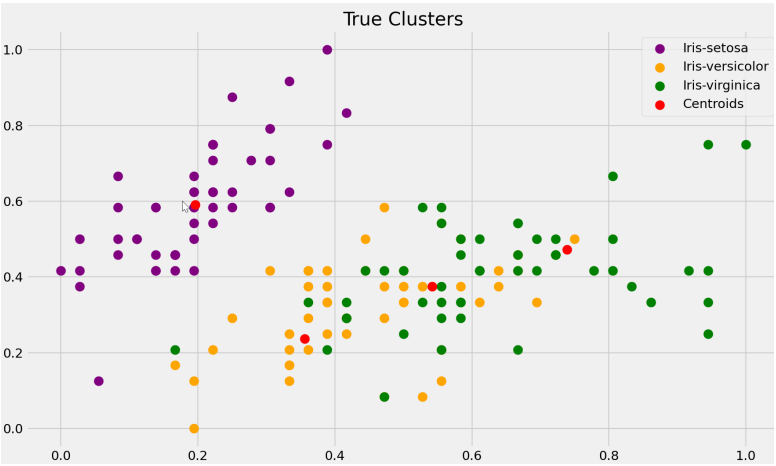
Métrica Davies-Bouldin - avalia a qualidade dos agrupamentos gerados

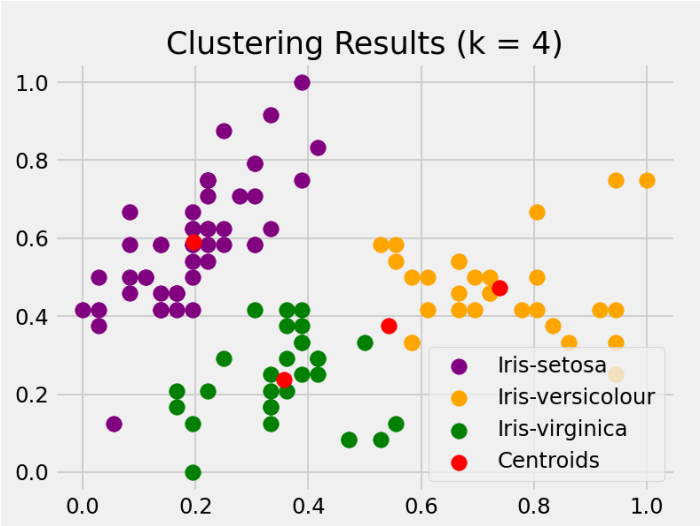
- Valor 0: clusters bem definidos e separados, sem sobreposição ou interseção.
- Valores maiores: clusters menos bem definidos, mais sobrepostos.
- A fórmula do índice é dada por DB igual ao quociente com numerador sendo o somatório dos máximos das medidas de similaridades (R_i_j) de cada cluster e denominador sendo a quantidade de clusters definido por k



Os resultados encontrados indicam que o melhores agrupamentos ocorreriam com o número de centróides igual a 8, pois apresenta o valor mais próximo de 0 encontrado (menor valor de DB).
https://github.com/pedroolynth0/LISTA11/blob/main/questao3.py

4.





Ao analisar os resultados obtidos ao aplicar o algoritmo K-means e observar os dois gráficos gerados, pude identificar a ocorrência de erros na classificação de alguns pontos no primeiro gráfico. Especificamente, percebi que os pontos das classes iris-virginica e iris-versicolor foram os mais afetados por essa classificação incorreta. Acredito que isso tenha ocorrido devido à grande proximidade e similaridade existente entre essas duas classes de flores.

https://github.com/pedroolynth0/LISTA11/blob/main/questao4.py

5.

O código apresentado nas questões anteriores funciona da seguinte maneira:

- Inicialmente, realiza a importação das bibliotecas necessárias para seu funcionamento
- Leitura dos dados da base íris, que terá seus dados agrupados. Os dados são armazenados na variável “base”
- Pré processamento: normalização de dados usando o min max scaler, dimensionando os valores no intervalo [0, 1]
- Cálculo da métrica silhouette: de k até o limite calculado (raiz quadrada da metade do número de linhas dos dados normalizados) o K-Means é executado e o score é calculado usando a função silhouette_score
- Cálculo da métrica elbow: cálculo do SSE (Sum of Squared Errors), utilizando o mesmo intervalo de k da métrica anterior, e armazenando os valores em wcss - valor encontrado é 4.
- Cálculo da métrica Davies Bouldin
- Plotagem das métricas elbow, silhouette e Davies-Bouldin usando matplotlib.pyplot
- Execução final do algoritmo K-Means utilizando o número de clusters encontrado pela métrica elbow (4): execução e posterior plotagem do algoritmo -instâncias plotadas em um gráfico de dispersão usando cores diferentes para cada classe.