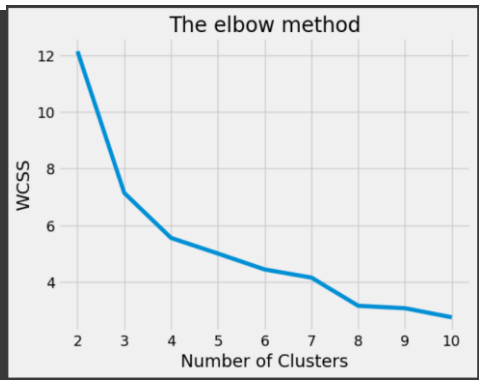
Lista 6 - Inteligência Artificial

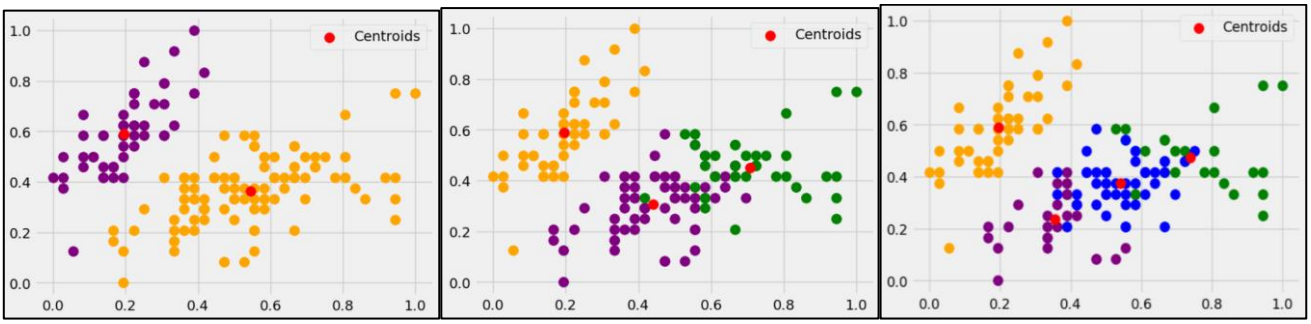
Arthur de Sá Braz de Matos

1- Ao usar o algoritmo KMeans na base Iris, verificamos que os resultados do coeficiente de Silhouette e do método Elbow diferiram na indicação do número ideal de agrupamentos. O coeficiente de Silhouette alcançou sua melhor qualidade com 2 clusters (foram realizados testes de k=2 até k=8, e o valor mais próximo do ideal, que seria 1, foi de 2 clusters), sugerindo que, com essa quantidade, os pontos estão mais próximos do centroide de seu cluster e mais afastados de outros clusters. Já o método Elbow indicou 4 clusters como o ideal, por ser o valor que desacelera a diminuição do valor de WCSS (um índice usado nesta avaliação). No entanto, a solução ideal, que já era conhecida para esta base, é de 3 agrupamentos, que agrupam corretamente os três tipos de flores representadas (setosa, versicolor e virginica).

Abaixo, seguem os valores de Silhouette e o gráfico associando o valor de WCSS e clusters para encontrar o Elbow, e três gráficos representando os agrupamentos encontrados com 2, 3 e 4 clusters, sendo respectivamente, o número sugerido pela métrica Silhouette, o número real de grupos e o sugerido pelo Elbow.

```
Silhouette Score k = 2: 0.629
Silhouette Score k = 3: 0.504
Silhouette Score k = 4: 0.445
Silhouette Score k = 5: 0.417
Silhouette Score k = 6: 0.325
Silhouette Score k = 7: 0.340
Silhouette Score k = 8: 0.326
```





Agora, uma caracterização de cada agrupamento contendo a média aproximada de cada atributo, para os três valores de clusters mencionados acima.

cluster	sepallength mean	sepalwidth mean	petallength mean	petalwidth mean
0	0.196111	0.590833	0.078644	0.060000
1	0.545000	0.363333	0.662034	0.656667

cluster	sepallength mean	sepalwidth mean	petallength mean	petalwidth mean
0	0.441257	0.307377	0.575715	0.549180
1	0.196111	0.590833	0.078644	0.060000
2	0.707265	0.450855	0.797045	0.824786

cluster	sepallength mean	sepalwidth mean	petallength mean	petalwidth mean
0	0.356322	0.237069	0.509059	0.471264
1	0.196111	0.590833	0.078644	0.060000
2	0.738506	0.472701	0.822911	0.863506
3	0.541667	0.375000	0.656578	0.641865

Link para acessar os códigos da questão 1:

https://colab.research.google.com/drive/1RtySCqqnPevREvV5bX9YAfrYXtn_CQKo?usp=sharing

2- Silhouette: é uma métrica que avalia a qualidade de um agrupamento, considerando tanto a coesão quanto a separação dos clusters. Ele fornece uma pontuação para cada ponto em um cluster, refletindo o quão bem o ponto está dentro de seu cluster e quão distante ele está de outros clusters. Esta fórmula para o Silhouette Score (Si) mede a qualidade do agrupamento de um ponto Xi considerando a coesão (quão próximo está dos pontos de seu próprio cluster) e a separação (quão distante está do cluster vizinho mais próximo). μ_{in} (Xi) representa a distância média de Xi para os pontos de seu próprio cluster, enquanto $\mu^{min}{}_{out}$ (Xi) é a distância média de Xi para o cluster mais próximo. A fórmula calcula a diferença entre essas distâncias e a normaliza, resultando em valores entre -1 e 1. Quanto mais próximo de 1, melhor o ponto está agrupado, valores próximos a -1 indicam que o ponto pode estar no cluster errado.

$$S_i = \frac{\mu_{out}^{min}(x_i) - \mu_{in}(x_i)}{max\{\mu_{out}^{min}(x_i), \mu_{in}(x_i)}$$

Elbow: A ideia principal é avaliar a soma das distâncias quadráticas dentro dos clusters (WCSS) para diferentes valores de K e identificar um ponto em que a diminuição do WCSS começa a desacelerar. Esse ponto é chamado de elbow, e o número de clusters correspondente a esse ponto é considerado o ideal. Na fórmula, J(K) é o valor da função objetivo (WCSS), K é o número de clusters, Xi é um ponto de dados, Cj é o conjunto de pontos atribuídos ao cluster j, e μ_j é o centroide do cluster j. A fórmula calcula a soma das distâncias quadradas entre cada ponto Xi e o centroide μ_j ao qual ele está atribuído, sendo essa a medida usada para ajustar os centroides e melhorar a compactação dos clusters. O objetivo do K-means é minimizar essa soma para criar clusters mais compactos e bem definidos.

$$J(k) = \sum_{j=1}^{k} \sum_{\mathbf{x}_i \in C_j} ||\mathbf{x}_i - \mu_j||^2$$

Ambos os métodos foram usados no código da questão 1:
https://colab.research.google.com/drive/1RtySCqqnPevREvV5bX9YAfrYXtn_CQKo?usp=sharing

3- Uma outra métrica de avaliação de agrupamento é o Índice de Dunn. Ele é utilizado para avaliar a compactação e a separação entre os clusters, ou seja, ele mede a distância mínima entre os pontos de diferentes clusters e a maior distância dentro de um mesmo cluster. Ele tende a valorizar agrupamentos bem separados, onde os clusters são compactos e distantes uns dos outros. Quanto maior o valor do Índice de Dunn, melhor a separação entre os clusters. Na fórmula, diâmetro de um cluster é a maior distância entre qualquer par de pontos dentro do cluster.

$$D = \frac{\min_{i \neq j} \text{distância mínima entre os clusters } i \text{ e } j}{\max_k \text{diâmetro do cluster } k}$$

Aplicando esta métrica ao KMeans, obtive o seguinte resultado: `Índice de Dunn: 0.06939133310888188`

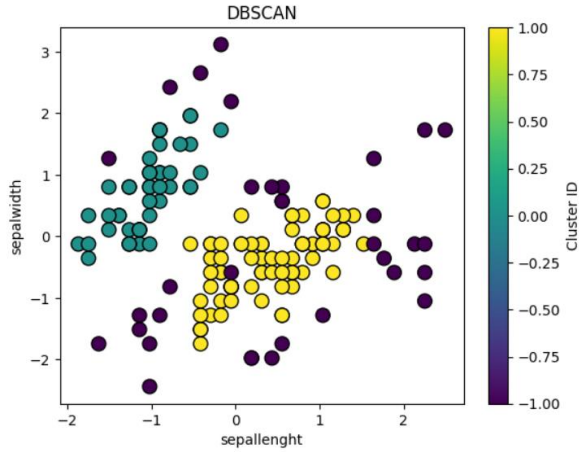
O valor obtido sugere que os clusters estão mal separados ou que há sobreposição significativa entre eles. Isso significa que os grupos formados pelo KMeans não são muito distintos, o que pode indicar que a quantidade de clusters não é ideal ou que a distribuição dos dados não é bem dividida.

Link para acessar o código da questão 3:

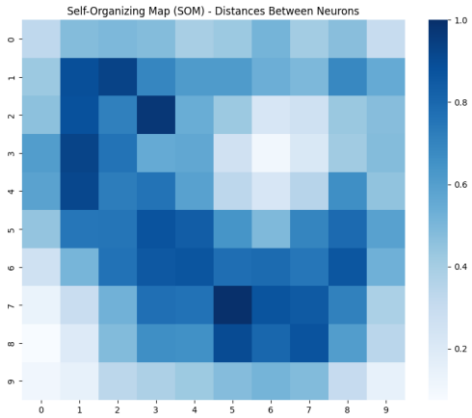
https://colab.research.google.com/drive/1dxkCgwyYGUu7YL1S_FkvKGodzq-qMfZh?usp=sharing

4- DBSCAN: É um algoritmo de clustering baseado em densidade. Ele agrupa pontos próximos entre si e marca pontos em regiões de baixa densidade como ruído (outliers). Este algoritmo permite ajustar alguns parâmetros, como “eps”, que é a distância máxima entre dois pontos para que eles sejam considerados no mesmo cluster e “min_samples”, o número mínimo de pontos em um cluster. Ao utilizar tais valores, como 0.5 e 5, respectivamente, obtive o mesmo resultado que a Silhouette no KMeans, ou seja, 2 grupos. Porém, foram encontrados alguns outliers, mas quando alteramos os parâmetros mencionados o número de grupos e outliers altera. A seguir, o trecho de código da definição dos parâmetros e chamada da função e o gráfico gerado.

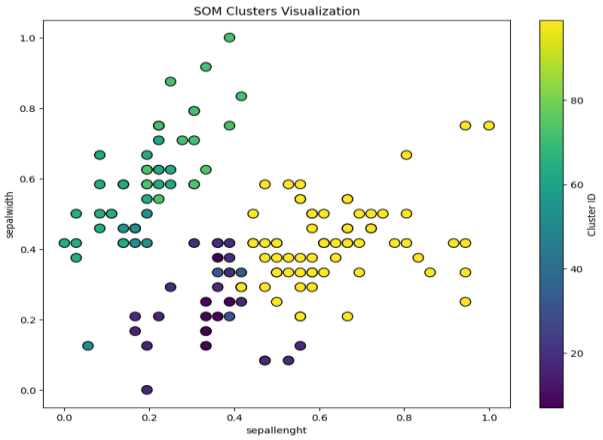
```
dbscan = DBSCAN(eps=0.5, min_samples=5)
```



SOM: Self-Organizing Map é um tipo de rede neural não supervisionada. Ele é utilizado principalmente para redução de dimensionalidade e visualização de dados de alta dimensão, mapeando esses dados em uma grade bidimensional de neurônios, onde as relações de proximidade entre os neurônios na grade refletem as semelhanças entre os dados originais. Quando visualizamos o resultado de um SOM, as cores podem ser usadas para diferenciar diferentes agrupamentos ou clusters. Cada cor geralmente corresponde a um agrupamento específico de dados que o SOM conseguiu identificar. Isso permite que ver como os dados foram agrupados, sem precisar de rótulos explícitos. Ele tenta preservar a topologia dos dados, ou seja, dados semelhantes são mapeados para regiões próximas no mapa. Como resultado, áreas do mapa que possuem a mesma cor indicam que os dados que caem nessas regiões são semelhantes entre si. Segue abaixo, o mapa gerado usando a base Iris.



O SOM não precisa de um número fixo de clusters como o KMeans. A estrutura do mapa é construída de maneira contínua, em vez de definir explicitamente K clusters. Apesar disso, podemos associar áreas ou regiões do mapa SOM a agrupamentos, e isso pode ser feito visualmente usando diferentes cores. Ao comparar os dois, cada região colorida no SOM pode ser associada a um cluster identificado pelo KMeans. Mesmo que o KMeans atribua um único rótulo para cada instância de dado (1, 2, 3, etc.), a cor no SOM pode ajudar a visualizar quais dados caem dentro de cada cluster.

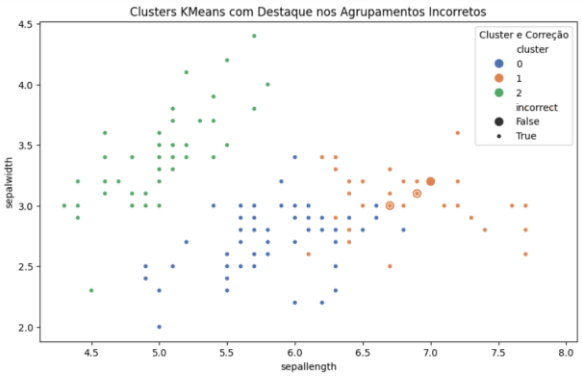


Links para acessar os códigos da questão 4 (DBSCAN e SOM, respectivamente):

https://colab.research.google.com/drive/1kHn-DR9T4x6jGt6ZkGcOskMnhSW7HvS8?usp=sharing

https://colab.research.google.com/drive/1JMYVUDttQR6CPu9OsrCNe89AtJ1PCZSm?usp=sharing

5- Ao aplicar o algoritmo KMeans com 3 clusters na base de dados Iris, as instâncias foram agrupadas de acordo com suas características numéricas. O gráfico gerado revelou que, apesar de algumas instâncias terem sido agrupadas corretamente (indicadas por círculos menores), outras foram classificadas incorretamente (representadas por círculos maiores). Isso ocorreu devido à semelhança entre algumas classes, como “versicolor” e “virginica”, que possuem características muito próximas, dificultando a distinção para o KMeans. Como o algoritmo é baseado em distâncias e não utiliza informações de classe, ele pode gerar agrupamentos que não correspondem exatamente às classes reais de flores. Esse comportamento ilustra a limitação do KMeans, que pode não conseguir fazer uma separação perfeita. A análise visual dos agrupamentos mostra que, apesar de o número de clusters ser 3, o KMeans teve dificuldades em distinguir algumas classes corretamente.



Link para acessar o código da questão 5:

https://colab.research.google.com/drive/15Bv4z-fqFcaKKQSUPuonZo-NfmgkGFv9?usp=sharing

6- Foi necessário realizar algumas técnicas de pré-processamento dos dados para garantir que os algoritmos de agrupamento funcionassem de maneira eficiente e eficaz. A normalização dos dados foi uma das primeiras etapas essenciais, pois as variáveis tinham escalas diferentes e a normalização garantiu que todos os atributos estivessem na mesma faixa, evitando que atributos com maior magnitude influenciassem desproporcionalmente os resultados. Além disso, a escolha de atributos foi feita (apenas removi o da classe), para não influenciar no resultado, mas em outras bases esta escolha ajuda a melhorar a qualidade do agrupamento e a reduzir o tempo computacional. Também foi feita a escolha fixa do número de clusters(com K=número real de grupos) em alguns testes, para ver a diferença do gráfico real com o sugerido pelo Silhouette e Elbow. Apesar de não ter usado muitas etapas de pré-processamento, para outras bases de dados o pré-processamento pode exigir uma abordagem mais abrangente. Por exemplo, a codificação de atributos nominais pode ser necessária, já que a maioria dos algoritmos de agrupamento, incluindo KMeans e DBSCAN, requer que as variáveis categóricas sejam convertidas em um formato numérico. A remoção de outliers também é importante em bases com dados extremos ou ruidosos, já que esses pontos podem distorcer os resultados do agrupamento e gerar clusters artificiais. Em bases de dados com muitas variáveis, a redução de dimensionalidade, é uma prática comum, pois reduz o número de atributos sem perder informações essenciais, facilitando a visualização e aumentando a performance dos algoritmos de clustering.