Trabalho Prático 02- Clusterização K-means

Análise dos Resultados

Para k = 3

- **sklearn**: Score de Silhueta = **0.55**
- **hardcore (minha implementação):** Score de Silhueta = **0.55**

Neste caso, os resultados são idênticos. Isso sugere que a minha implementação do algoritmo de K-Means, quando usada para agrupar em 3 clusters, atingiu a mesma qualidade de agrupamento que a versão otimizada e robusta da biblioteca sklearn. Um score de 0.55 é considerado um resultado decente, indicando que os clusters estão moderadamente bem separados.

Para k = 5

- **sklearn**: Score de Silhueta = **0.72**
- **hardcore (minha implementação)**: Score de Silhueta = **0.37**

Aqui, há uma diferença significativa. O **sklearn** obteve um score de **0.72**, que é um resultado excelente, indicando clusters muito bem separados. Por outro lado, a minha implementação teve um score de **0.37**, que é consideravelmente mais baixo.

O que explica a diferença?

A principal razão para essa discrepância é a **inicialização dos centróides**.

- **Minha implementação (hardcore)**: Provavelmente, a inicialização dos centróides é feita de forma totalmente aleatória. Dependendo de onde os centróides são posicionados no início, o algoritmo pode convergir para um **mínimo local** que não é o ideal, especialmente quando o número de clusters (k) é maior.
- **sklearn**: A sklearn utiliza, por padrão, o algoritmo **k-means++** para a inicialização dos centróides. Esse método não é aleatório; ele seleciona os centróides iniciais de forma inteligente, garantindo que eles estejam o mais distantes possível uns dos outros. Isso ajuda o algoritmo a convergir mais rapidamente para a melhor solução global, evitando os mínimos locais.