Relatório de Análise Comparativa: K-means Hardcore vs. Scikit-learn

Trabalho Prático 02**:** Clusterização K-means - Inteligência Artificial
Maria Eduarda Ferreira da Silva e Vitória Christie Amaral Santos

1. Introdução

Este relatório compara duas implementações do algoritmo K-means na base de dados Iris: uma desenvolvida "do zero" (hardcore) e outra da biblioteca Scikit-learn. O objetivo é validar a implementação manual e analisar as diferenças de performance, utilizando k=3, número ideal de clusters encontrado via Silhouette Score (0.5528 para k=3 vs. 0.4922 para k=5).

2. Análise Comparativa de Desempenho

2.1. Qualidade da Clusterização

Para avaliar a qualidade dos clusters, utilizamos as classes originais da base Iris (target) e comparamos com os grupos gerados por cada algoritmo, utilizando três métricas principais:

- **Adjusted Rand Index (ARI):** Mede a similaridade entre os clusters reais e os previstos.
- **Homogeneidade:** Verifica se cada cluster contém apenas membros de uma única classe real.
- **Completude:** Verifica se todos os membros de uma classe real são atribuídos ao mesmo cluster.

Os resultados demonstraram que a implementação hardcore produziu clusters com qualidade idêntica à da biblioteca Scikit-learn. Os scores iguais em todas as métricas validam a corretude lógica e matemática do nosso algoritmo, confirmando que ele agrupa os dados de forma tão eficaz quanto a implementação padrão da indústria.

2.2. Performance Computacional

Análise de Tempo: A implementação do Scikit-learn foi aproximadamente 4.9 vezes mais rápida. Essa diferença significativa é esperada, pois a biblioteca possui otimizações de baixo nível, com suas funções mais críticas (como os cálculos de distância) escritas em linguagens compiladas, que superam o desempenho do Python puro, que é uma linguagem interpretada.

Análise de Memória: Nossa implementação consumiu menos memória. Isso é explicado porque nosso código é mais simples e direto, alocando apenas as estruturas de dados essenciais para a execução. A biblioteca Scikit-learn, por ser mais robusta e com mais funcionalidades, resulta em um consumo de memória ligeiramente superior para este caso de uso específico.

3. Conclusões

A realização deste trabalho permitiu concluir que:

1. A implementação do algoritmo K-means do zero foi bem-sucedida, produzindo resultados consistentes e com a mesma qualidade da biblioteca Scikit-learn, o que valida a sua corretude.
2. A análise de desempenho ressaltou a vantagem de velocidade das bibliotecas otimizadas, justificando sua ampla utilização em projetos de larga escala.
3. A implementação manual, apesar de mais lenta, pode ser mais eficiente em termos de memória em cenários simples, devido à sua menor complexidade interna.
4. A técnica de PCA (Análise de Componentes Principais) mostrou-se indispensável para a visualização e interpretação dos clusters em dados multidimensionais, confirmando visualmente a coesão dos grupos encontrados.