Disciplina: Inteligência Artificial

Professora: Cristiane Neri Nobre

Data de entrega: 20/04

Valor: 1,5 pontos

Aluno: Lucas Henrique Rocha Hauck

Github: https://github.com/o-hauck/IA

Para fazer as questões abaixo, sugiro que estude o material sobre **Agrupamento** que está no CANVAS. Assista também os vídeos disponibilizados sobre este assunto. Está junto com os slides.

Além disso, acesso o notebook "**Kmeans.ipynb"**, disponibilizado no CANVAS.

Questão 01

Rode o algoritmo Kmeans na base de dados a seguir da Iris, que está disponível no CANVAS.

1. Realize o pré-processamento necessário para esta base de dados: identificação de outlier e normalização (veja a etapa de normalização no arquivo **Parte 6 - Processamento - transformação de atributos numéricos.pdf).** Para identificação de outlier, investigue como eliminar outlier no livro texto da disciplina.
2. Encontre os agrupamentos, discuta a qualidade destes agrupamentos (usando Silhouette e Elbow) e caracterize os agrupamentos obtidos.
3. Investigue os hiperparâmetros do algoritmo, como a escolha do centróide e métricas de distâncias (euclidiana e etc).
4. Explique como se obtém estas duas métricas, ou seja, explique as equações matemáticas.
5. Investigue, explique e implemente, pelo menos, mais 1 métrica de avaliação dos agrupamentos, diferentes das 2 anteriores
6. Utilizando mais dois algoritmos de agrupamento, por exemplo o DBSCAN e o SOM, verifique se estes métodos encontraram a mesma quantidade de grupos que o Kmeans. Faça uma análise dos grupos encontrados pelos 3 algoritmos
7. Uma vez que a base é classificada (setosa, virgínica e versicolor), mostre **visualmente** que instâncias foram agrupadas incorretamente pelo kmeans. Discuta os resultados.
8. Faça um pequeno relatório explicando todas as etapas de pré-processamento realizadas e explicando todos os resultados obtidos.

Coloque os links para os códigos produzidos ao final de cada questão

1. Pré-processamento da base de dados (outliers e normalização)

Primeiramente, carreguei a base de dados Iris e realizei uma análise para detecção de *outliers*. Para isso, utilizei o método do IQR (Intervalo Interquartílico), conforme apresentado no livro texto da disciplina. Esse método considera valores abaixo de Q1 - 1.5×IQR ou acima de Q3 + 1.5×IQR como outliers. Após a remoção desses dados extremos, prossegui com a normalização dos atributos numéricos utilizando a normalização *Min-Max*, que transforma os dados para um intervalo entre 0 e 1. Isso foi fundamental para garantir que todos os atributos tivessem o mesmo peso nos algoritmos de agrupamento.

---

2. Agrupamentos com KMeans, qualidade dos clusters (Silhouette e Elbow) e caracterização

Apliquei o algoritmo KMeans para realizar o agrupamento. Para determinar a quantidade ideal de clusters, utilizei dois métodos: **Elbow** e **Silhouette**.

- Pelo método do **cotovelo (Elbow)**, o ponto de inflexão no gráfico ocorreu em **K = 4**, indicando que esse é um bom número de clusters.
- No entanto, o **índice de Silhouette** foi maior para **K = 2 (0.618)**, caindo à medida que K aumenta. Isso mostra que os agrupamentos ficam menos coesos com mais clusters.

Com base nisso, optei por analisar com **K = 3** e **K = 4**. O agrupamento com **K = 3** foi o mais próximo das três classes originais da base (Setosa, Versicolor e Virginica). Já com K = 4, houve uma divisão mais detalhada, o que pode indicar uma subestrutura nos dados.

---

3. Hiperparâmetros do KMeans: centróides e métricas de distância

O principal hiperparâmetro do KMeans é o número de clusters (K), definido com base nas análises anteriores. Além disso, o algoritmo permite escolher:

- **Inicialização dos centróides**: Utilizei o método k-means++, que escolhe centróides iniciais mais distantes entre si, o que tende a melhorar a performance e evitar mínimos locais.
- **Métrica de distância**: A distância **euclidiana** foi utilizada por padrão, mas também testei com a **distância de Manhattan** para comparação.

---

4. Explicação das métricas (Silhouette e WCSS)

As duas principais métricas usadas foram:

a) Silhouette Score:

Essa métrica mede o quão bem uma instância está posicionada dentro do seu cluster comparado com os clusters vizinhos. A fórmula é:

$$s = \frac{b - a}{\max(a, b)}$$

Onde:

- a = distância média da instância para os outros pontos no mesmo cluster
- b = menor distância média da instância para os pontos de outros clusters

Valores próximos de 1 indicam uma boa separação entre clusters.

b) WCSS (Within-Cluster Sum of Squares):

É a soma das distâncias quadradas dos pontos aos seus centróides. Fórmula:

$$WCSS = \sum_{i=1}^{k} \sum_{x \in C_i} ||x - \mu_i||^2$$

Onde Ci é o conjunto de pontos do cluster i e μi é o centróide do cluster.

---

5. Outra métrica de avaliação: Calinski-Harabasz

Implementei também o **índice de Calinski-Harabasz**, que considera a dispersão entre os clusters e dentro dos clusters. Quanto maior, melhor. A fórmula é:

$$CH = \frac{Tr(B_k)}{Tr(W_k)} \cdot \frac{N - k}{k - 1}$$

Onde:

- Tr(Bk) = soma entre-clusters

- Tr(Wk) = soma intra-cluster
- N = número total de pontos
- k = número de clusters

---

6. Comparação com DBSCAN e SOM

Utilizei os algoritmos **DBSCAN** e **SOM** (Self-Organizing Maps) para comparar com o KMeans:

- **DBSCAN**: Detectou 2 clusters e considerou alguns pontos como *ruído*. Não identificou bem a separação entre as três espécies, pois é mais sensível à densidade.
- **SOM**: Conseguiu separar bem as instâncias em 3 grupos distintos, bastante próximo do resultado do KMeans com K=3.

Conclusão: O KMeans com K=3 e o SOM foram os que melhor representaram a estrutura natural da base. DBSCAN teve mais dificuldade por causa da distribuição dos dados da Iris.

---

7. Visualização de erros de agrupamento

Utilizando a base rotulada, comparei os clusters obtidos pelo KMeans com as classes reais (Setosa, Versicolor e Virginica). Os maiores erros ocorreram entre as classes **Versicolor** e **Virginica**, que são mais próximas entre si. Já a **Setosa** foi quase sempre agrupada corretamente, o que faz sentido, pois é visualmente mais distinta no espaço de atributos.

A visualização em 2D com PCA (ou t-SNE) mostrou claramente quais pontos foram mal agrupados, possibilitando identificar os erros de forma visual.

---

8. Relatório final das etapas e resultados

Durante a análise da base Iris, segui as seguintes etapas:

- **Pré-processamento**: Remoção de outliers com IQR e normalização Min-Max.
- **Agrupamento com KMeans**: Testes de K variando de 2 a 10. K = 4 indicado pelo Elbow, mas K = 3 com melhor equilíbrio entre Silhouette e fidelidade à base real.

- **Avaliação**: Silhouette, WCSS e Calinski-Harabasz aplicados para avaliação dos agrupamentos.
- **Comparação com outros algoritmos**: DBSCAN e SOM utilizados. KMeans e SOM se saíram melhor.
- **Análise dos erros**: Comparação com rótulos reais mostrou que a classe Setosa é facilmente separável, enquanto Versicolor e Virginica se confundem.
- **Conclusão**: O KMeans com K=3 é o melhor modelo para essa base, e os resultados demonstram que mesmo métodos não supervisionados podem capturar bem estruturas em dados rotulados.