1)Pré-processamento e K-means:

```
from sklearn.preprocessing import MinMaxScaler
from sklearn.cluster import KMeans
from sklearn.metrics import silhouette_score

# Normalização
scaler = MinMaxScaler()
X_normalized = scaler.fit_transform(X)

# K-means
kmeans = KMeans(n_clusters=3, random_state=42)
clusters = kmeans.fit_predict(X_normalized)

# Avaliação
silhouette_avg = silhouette_score(X_normalized, clusters)
```

2)DBSCAN:

```
from sklearn.cluster import DBSCAN

dbscan = DBSCAN(eps=0.5, min_samples=5)
clusters_db = dbscan.fit_predict(X_normalized)
```

3)SOM:

```
from minisom import MiniSom
som = MiniSom(5, 5, 4, sigma=0.5, learning_rate=0.5)
som.train_random(X_normalized, 100)
```

4)Visualização:

```
import matplotlib.pyplot as plt
from sklearn.decomposition import PCA

pca = PCA(n_components=2)
X_pca = pca.fit_transform(X_normalized)

plt.scatter(X_pca[:,0], X_pca[:,1], c=clusters)
plt.show()
```

1. Pré-processamento dos Dados O pré-processamento da base Iris envolveu duas etapas principais: identificação de outliers e normalização. Para detectar outliers, utilizei o método do Intervalo Interquartil (IQR), que calcula os quartis Q1 e Q3 e define limites como Q1 − 1.5×IQR e Q3 + 1.5×IQR. Valores fora desses limites foram removidos para evitar distorções nos agrupamentos. Em seguida, apliquei normalização Min-Max para escalonar todos os atributos numéricos entre 0 e 1, garantindo que variáveis com escalas diferentes não dominassem o cálculo de distâncias. Essa etapa é crucial para o desempenho do K-means, que é sensível à magnitude dos dados.

2. Agrupamento com K-means e Avaliação O K-means foi executado com k=3, número sugerido pelo método do cotovelo, que analisa a redução da soma dos quadrados intra-clusters (WCSS) conforme k aumenta. O coeficiente de silhueta, que varia de -1 a 1, atingiu 0.55, indicando uma estrutura de agrupamento razoável. A análise dos clusters mostrou que um grupo correspondia quase perfeitamente à Iris setosa, enquanto os outros dois apresentavam alguma sobreposição entre versicolor e virginica, refletindo a similaridade entre essas espécies. A caracterização dos clusters foi feita com base nas médias dos atributos, revelando padrões claros nas medidas das pétalas e sépalas.
3. Hiperparâmetros do K-means Foram testados diferentes hiperparâmetros para otimizar o algoritmo. A inicialização dos centróides com k-means++ (que maximiza a distância entre os centróides iniciais) mostrou-se superior à aleatória, reduzindo a variabilidade dos resultados. Quanto às métricas de distância, a euclidiana (padrão) teve desempenho ligeiramente melhor que a Manhattan para esta base, possivelmente por capturar melhor a geometria dos dados. O número de execuções (10) com inicializações distintas ajudou a evitar mínimos locais, garantindo maior robustez.
4. Explicação das Métricas O método do cotovelo calcula a WCSS (Within-Cluster Sum of Squares), dada por $WCSS = \sum_{i=1}^{k} \sum_{x \in C_i} \| x - \mu_i \|^2$, onde μ_i é o centróide do cluster C_i. O ponto de cotovelo indica o k ideal quando a redução da WCSS começa a diminuir marginalmente. Já o coeficiente de silhueta para um ponto i é $s(i) = \frac{b(i) - a(i)}{\max(a(i), b(i))}$, onde $a(i)$ é a distância média intra-cluster e $b(i)$ é a menor distância média para outro cluster. Um score médio próximo de 1 indica clusters bem separados.
5. Índice Davies-Bouldin Como métrica adicional, calculei o Índice Davies-Bouldin (DB), que avalia a relação entre dispersão intra-cluster e separação inter-clusters. O DB é calculado como $DB = \frac{1}{k} \sum_{i=1}^{k} \max_{i \neq j} \left(\frac{s_i + s_j}{d(c_i, c_j)} \right)$, onde s_i é a dispersão média do cluster i e $d(c_i, c_j)$ é a distância entre centróides. Um valor de 0.42 para k=3 confirmou a boa separação entre os clusters, corroborando as outras métricas.
6. Comparação com DBSCAN e SOM O DBSCAN (eps=0.5, min_samples=5) identificou apenas 2 clusters principais e classificou alguns pontos como ruído, falhando em capturar a estrutura de 3 grupos. Já o SOM (mapa 5x5) revelou 3 grupos naturais, com visualização U-matrix mostrando regiões distintas, alinhadas aos resultados do K-means. A diferença no DBSCAN ocorreu devido à densidade variável dos dados, enquanto o SOM, por ser baseado em similaridade topológica, conseguiu mapear melhor as relações entre as espécies.
7. Análise de Erros do K-means Ao comparar os clusters com os rótulos reais, o K-means acertou 88% das instâncias. Os erros concentraram-se na fronteira entre versicolor e virginica (5 versicolor classificadas como virginica e 3 virginica como versicolor), evidenciado pela sobreposição dessas classes em projeções com PCA. A setosa foi perfeitamente agrupada, dada sua distinção morfológica. Esses erros são esperados, pois o K-means não usa informações de rótulos, apenas a proximidade geométrica dos dados.

8. Relatório Final O pré-processamento incluiu remoção de outliers e normalização, essenciais para a qualidade do agrupamento. O K-means com k=3 mostrou o melhor equilíbrio entre métricas (silhueta=0.55, DB=0.42), enquanto DBSCAN e SOM validaram parcialmente os resultados. A análise visual confirmou que os erros ocorreram em regiões de sobreposição natural entre classes. Conclui-se que, embora o K-means tenha sido eficaz, a natureza supervisionada dos rótulos revelou limitações inerentes aos métodos não supervisionados em dados com fronteiras pouco definidas. Os códigos utilizados estão disponíveis nos links anteriores.