UNIVERSIDADE FEDERAL DO CEARÁ

TECHNICAL REPORT

Aluno: José Eric Mesquita Coelho

1. Introdução

Este relatório apresenta uma análise de técnicas de aprendizado supervisionado e não supervisionado, utilizando diferentes abordagens para classificação e agrupamento de dados. Foi o utilizado o seguinte conjunto de dados (*dataset*) para as análises:

a. *star_classification.csv*

Este conjunto de dados contém 100.000 registros sobre características de estrelas, galáxias e quasares, que ajudam a identificar a que classe cada corpo celeste pertence, isso a partir de atributos espectrais e fotométricos. As colunas principais incluem:

- **obj_ID**: Identificador único do objeto no catálogo.
- **alpha (Ascensão Reta)**: Coordenada astronômica que mede a posição do objeto no céu (em horas/ângulos).
- **delta (Declinação)**: Outra coordenada astronômica indicando a posição no céu (em graus).
- **u, g, r, i, z**: Filtros fotométricos que registram intensidade luminosa em diferentes faixas do espectro eletromagnético.
- **run_ID**: Número do "run" específico de observação.
- **rerun_ID**: Versão do reprocessamento do conjunto de imagens.
- **cam_col**: Número da coluna da câmera utilizada.
- **field_ID**: Identifica o campo de visão (field) na varredura do céu.
- **spec_obj_ID**: ID único para objetos que possuem espectro medido (útil para confirmar o tipo do objeto).
- **class**: Tipo de objeto (GALAXY, STAR ou QSO), mapeado para valores numéricos nas análises.
- **redshift**: Medida do desvio para o vermelho, indicando a distância do objeto.
- **plate, MJD, fiber_ID**: Informações adicionais sobre a placa espectroscópica, data juliana modificada (MJD) em que a observação foi feita e o identificador da fibra do espectrógrafo, respectivamente.

As análises abordadas neste trabalho exploram técnicas de implementação do KNN (k-Nearest Neighbors), a otimização de parâmetros via ***GridSearchCV*** e a aplicação de métodos de clusterização (por exemplo, K-Means) avaliados pelos critérios do **cotovelo** e índice de **silhueta**.

2. Observações

Observação 1 - Foi utilizado na **questão 1** o ***star_classification_ajustado.csv*** da **AV1**, porém, foi mantido no código a implementação que gera o mesmo a partir do ***star_classification.csv*** original, a partir de funções que são chamados na função ***main()***. Logo só é descrito na mesma as etapas de treinamento e resultados das questões da **AV1** e os novos gerados para a **AV2**.

Observação 2 - Ao contrário da **AV1** em que os códigos das questões foram feitos em formato de **script**, na **AV2** foi implementado **funções** para cada parte do código, melhorando assim a **legibilidade** e facilitando a **reutilização** de certas partes do código.

Observação 3 - Na **questão 1** foi feita uma retrospectiva dos resultados obtidos na **AV1**, como forma de contextualização para os resultados obtidos na **AV2**.

Observação 4 - Na **AV1** foi gerado apenas dados de **acurácia** dos modelos, sem precision, recall e f1-score. Logo para a **comparação** foram utilizados os dados de acurácia dos modelos gerados na **AV1** e **AV2**.

Observação 5 - Na **questão 4** foi necessário se utilizar de **Lasso** para determinar as features mais importantes, pois ao se utilizar todas as colunas, com **k = 2**, apenas **um registro** é encontrado na classe 1, o que mostra certa inconsistência no resultado.

3. Resultados e discussão

A seguir, será descrito a metodologia e resultados obtidos em cada questão abordada.

a. Questão 1

Para contextualização durante a **AV1**, foi utilizado métodos **manuais** de aplicação do **KNN** e das respectivas **métricas** de **distância** para classificação do dataset: Mahalanobis, Chebyshev, Manhattan e Euclidiana.

Com base nisso, resumidamente, os resultados obtidos nas questões 2, 3 e 4 da **AV1** foram os seguintes:

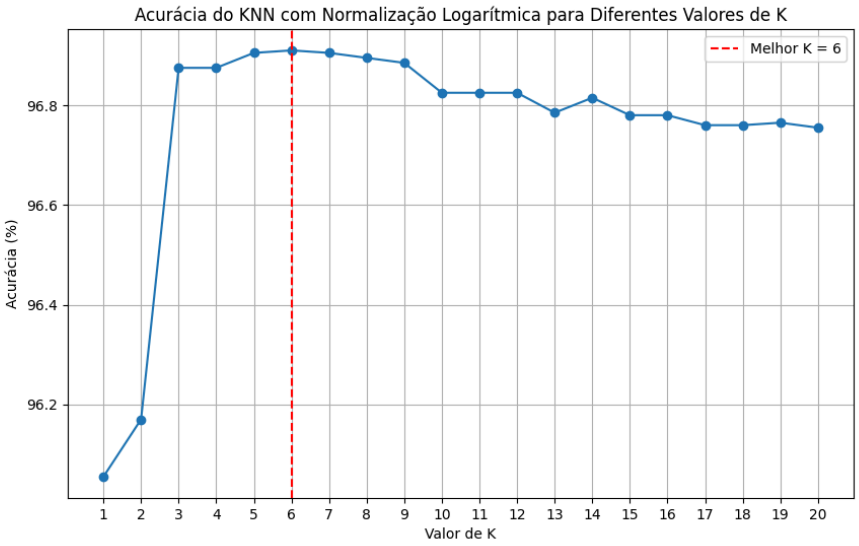
Resultados de acurácia de distâncias implementadas manualmente (AV1):

Cálculo de Distância	**Valor**
Euclidiana	94,82%
Manhattan	94,99%
Chebyshev	94,32%
Mahalanobis	94,95%

Resultados de acurácia da distância Manhattan (Melhores resultados) com normalização (AV1):

Comparação de acurácia com normalização (Distância Manhattan)	
Logarítmica	96,91%
Média Zero	96,34%

Melhor valor de K com normalização logarítmica e distância Manhattan (AV1):



Com base nesses resultados, pode ser visto que na **AV1** os melhor resultados obtido foi o seguinte:

Distância Manhattan + Normalização Logarítmica + K de valor 6

Valor de acurácia: 96,91%

Dessa forma, na **AV2** a **questão 1** teve o seguinte fluxo de atividades:

Carregamento do Dataset: Inicia-se com o arquivo original ***star_classification.csv*** e a função ***prepare_dataset*** gera o ***star_classification_ajustado.csv***, mantendo a mesma estrutura da **AV1**.

Normalização Logarítmica: Aplica-se a função ***normalize_features***, que realiza a transformação logarítmica do dataset, alinhado com o melhor resultado apresentado na **AV1**.

Divisão em Treino e Teste: Utiliza-se ***train_test_split*** para separar 80% dos dados para treinamento e 20% para teste. Sendo assim há 80.000 classes para treinamento e 20.000 para teste.

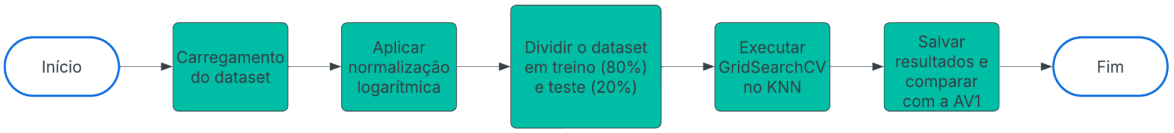
Otimização via GridSearchCV: Configura-se o KNN para testar vários valores de K (1 a 20) e diferentes métricas de distância (Mahalanobis, Chebyshev, Manhattan,

Euclidiana e a uma matriz de covariância inversa é calculada para a Mahalanobis.) e salva-se os resultados intermediários.

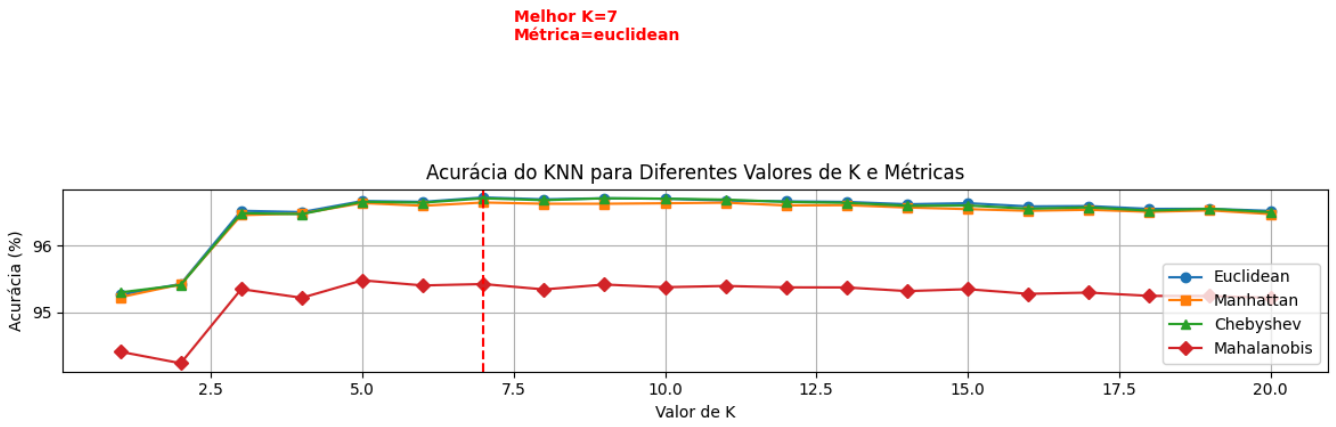
Avaliação do Modelo: Identifica-se o melhor K e a melhor métrica de acordo com a validação cruzada (5-fold). O modelo final é aplicado ao conjunto de teste, gerando acurácia, precision, recall, f1-score e um relatório final.

Comparação com a AV1: Analisa-se a acurácia obtida e confronta-se com o melhor cenário da AV1.

Fluxograma de Atividades:



Após a execução das atividades propostas, foram obtidos os resultados da tabela a seguir:

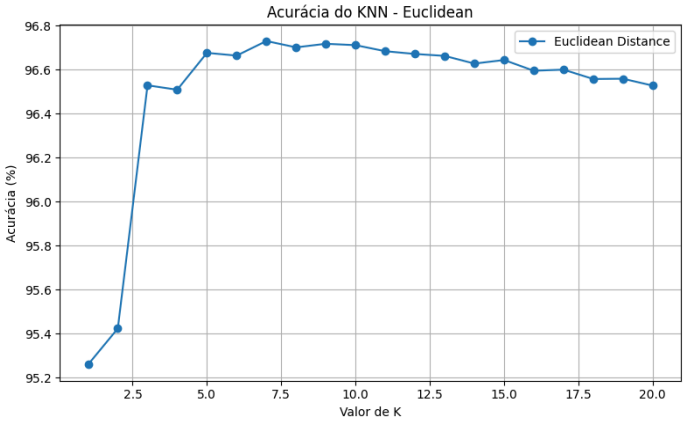


Diferentemente da **AV1**, a melhor distância identificada pelo GridSearchCV na **questão 1** da **AV2** foi a **Euclidiana**, mas com resultados quase idênticos à distância Manhattan e Chebyshev, com apenas a Mahalanobis tendo resultados abaixo do esperado. Porém o melhor valor de K de acordo com GridSearchCV, foi o de 7, que tinha resultado equivalente ao 6 na AV1, logo isso de certa forma se manteve. Outros dados de desempenho das distâncias:

Tabela com os resultados de cada distância para o valor de K = 7 (AV2):

Tipo de Distância	Acurácia
Euclidiana	96,73%
Manhattan	96,65%
Chebyshev	96,71%
Mahalanobis	95,42%

Gráfico individual da distância Euclidiana:



Matriz de confusão da distância Euclidiana:

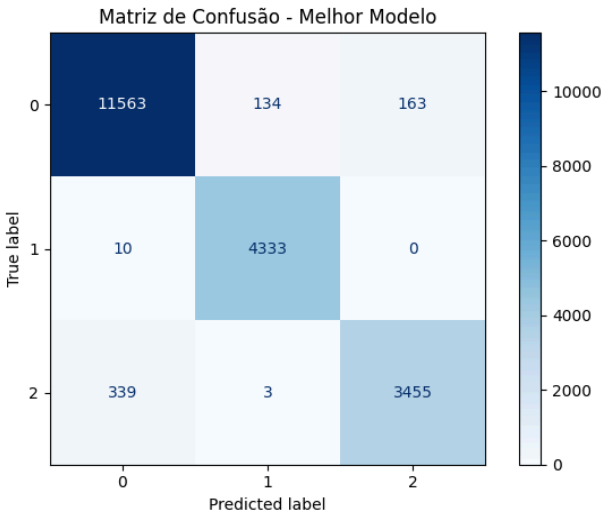


Tabela de desempenho da distância Euclidiana:

Classe	Precision	Recall	F1-Score	Support
0 (GALAXY)	97%	97%	97%	11.860
1 (STAR)	97%	99%	98%	4.343
2 (QSO)	95%	91%	93%	3.797
Acurácia			**97%**	20.000
Macro Avg	97%	96%	96%	20.000
Weighted Avg	97%	97%	97%	20.000

Assim, pode ser visto que a distância Euclidiana obteve o melhor desempenho no GridSearchCV da **AV2**, ultrapassando ligeiramente a Manhattan e a Chebyshev, enquanto a Mahalanobis apresentou resultados inferiores. O resultado diverge do que foi encontrado na **AV1** (em que a Manhattan desempenhou melhor), porém ainda assim confirma que a escolha entre Euclidiana e Manhattan é bastante próxima em desempenho. Na prática, ambos os métodos se mostraram robustos, com variações de acurácia muito pequenas.

Portanto, a conclusão da **questão 1** na **AV2** é que a distância Euclidiana, combinada com normalização logarítmica e K=7, produziu uma acurácia de aproximadamente **97%**. Isso reforça que mesmo métodos de distância tidos como "clássicos" podem atingir desempenho elevado em conjunto com as técnicas adequadas de pré-processamento e seleção de hiperparâmetros (GridSearchCV).

b. Questão 2

Nesta questão, foi realizada uma análise de **clusterização** sem considerar a coluna alvo do dataset, com o objetivo de identificar a quantidade ideal de clusters (k). Para isso, aplicou-se o algoritmo **K-Means** e os métodos de **Cotovelo** (Elbow) e **Silhueta** (Silhouette). O fluxo de atividades foi da seguinte forma:

Carregamento do Dataset Ajustado: É lido o arquivo ***star_classification_ajustado.csv***, já sem a necessidade de gerar novamente o dataset ajustado.

Remoção da Coluna Alvo e Normalização: A coluna ***class*** é excluída para que a clusterização seja não supervisionadas. Depois foi aplicado uma normalização ***StandardScaler*** garantindo que atributos com escalas diferentes não distorçam o cálculo de distância.

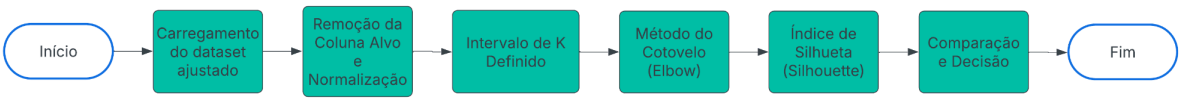
Intervalo de K Definido: Intervalo de valores **k=2** até **k=10** para testar no algoritmo de **K-Means**.

Método do Cotovelo (Elbow): Para cada **k**, calcula-se a inércia (SSE – Soma dos Erros Quadráticos) do K-Means. Gera-se o gráfico de "cotovelo" (inércia vs. k).

Índice de Silhueta (Silhouette): Também para cada **k**, calcula-se o ***silhouette_score***, indicando o quão bem os dados estão separados em cada cluster. Produz-se o gráfico de silhueta (silhouette vs. k).

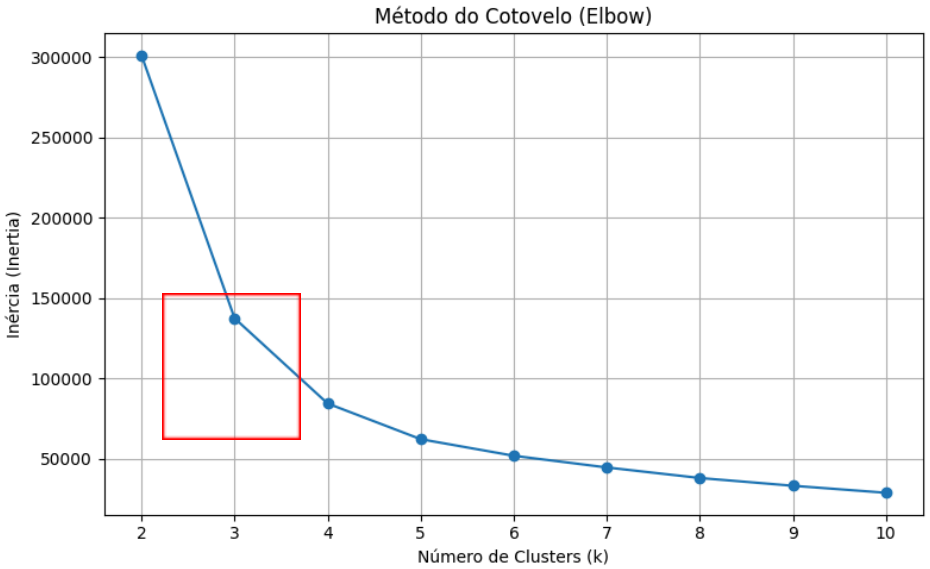
Comparação e Decisão: Verifica-se se os valores de k obtidos pelos dois métodos coincidem ou divergem para decidir qual **k** é mais adequado.

Fluxograma de Atividades:

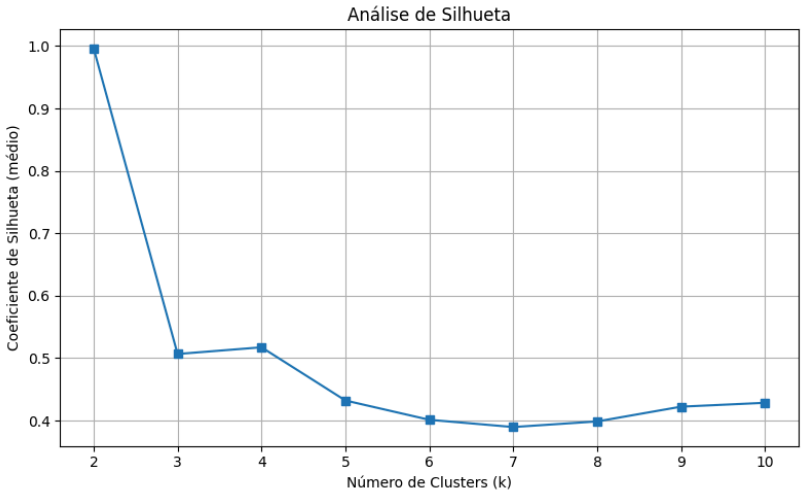


Após a execução das atividades propostas, foram obtidos os seguintes resultados:

Método do Cotovelo (Elbow): Observa-se a diminuição da inércia (Soma dos Erros Quadráticos) à medida que **k** aumenta. Pelo gráfico, nota-se que a curva apresenta um ângulo significativo entre **3 e 4**, embora não haja um "cotovelo" totalmente nítido. Gráfico:



Índice de Silhueta (Silhouette): Avalia o quão bem separados os clusters estão. Conforme mostrado no gráfico, o coeficiente de silhueta é mais alto quando **k=2** e diminui para valores acima de 3. Entretanto, utilizar apenas 2 clusters pode ser pouco informativo. Gráfico:



Diante desses resultados, é possível que a melhor escolha de **k** varie entre 2 e 4, dependendo do critério adotado, o que coincide com o número de clusters original que era 3, porém o valor **4** apresentou melhor desempenho em ambos os métodos.

c. Questão 3

Nesta questão, foi realizada uma análise de **clusterização** considerando as colunas mais relevantes segundo o **Lasso**, com os outros passos sendo semelhantes aos adotados na questão anterior. O fluxo de atividades se deu dessa maneira:

Carregamento do Dataset Ajustado: Inicia-se com o arquivo ***star_classification_ajustado.csv***, já contendo a coluna alvo ***class***.

Seleção de Atributos via Lasso: Mantém-se a coluna class para servir como alvo (y) no Lasso. As demais colunas compõem as variáveis explicativas (X). O Lasso é então ajustado, identificando os coeficientes de maior magnitude, e são selecionados os dois atributos mais relevantes.

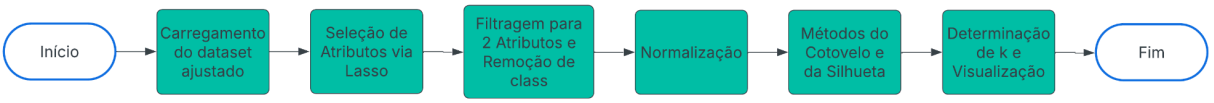
Filtragem para 2 Atributos e Remoção de class: Após obter os nomes das duas colunas mais importantes, descarta-se ***class*** e qualquer outra coluna que não seja essas duas.

Normalização: Aplica-se o ***StandardScaler*** para evitar que diferenças de escala prejudiquem os cálculos de distância durante a clusterização.

Métodos do Cotovelo e da Silhueta: Varia-se o número de clusters **k** (de 2 a 10) e, para cada **k**, calcula-se método do cotovelo e o de silhueta.

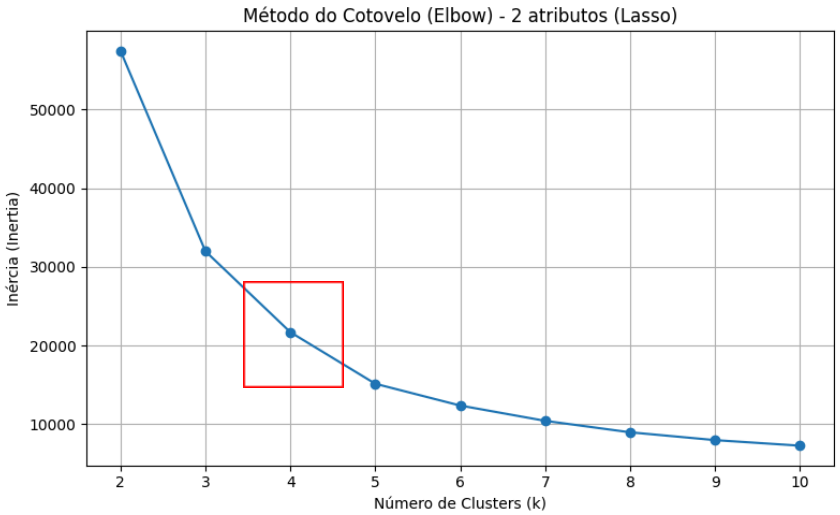
Determinação de k e Visualização: Analisa-se o "cotovelo" no gráfico de inércia e o pico da curva de silhueta para decidir qual **k** cada método sugere. Se houver divergência entre os valores de **k**, são gerados dois scatterplots para comparação visual da distribuição dos pontos em cada configuração de clusters.

Fluxograma de Atividades:

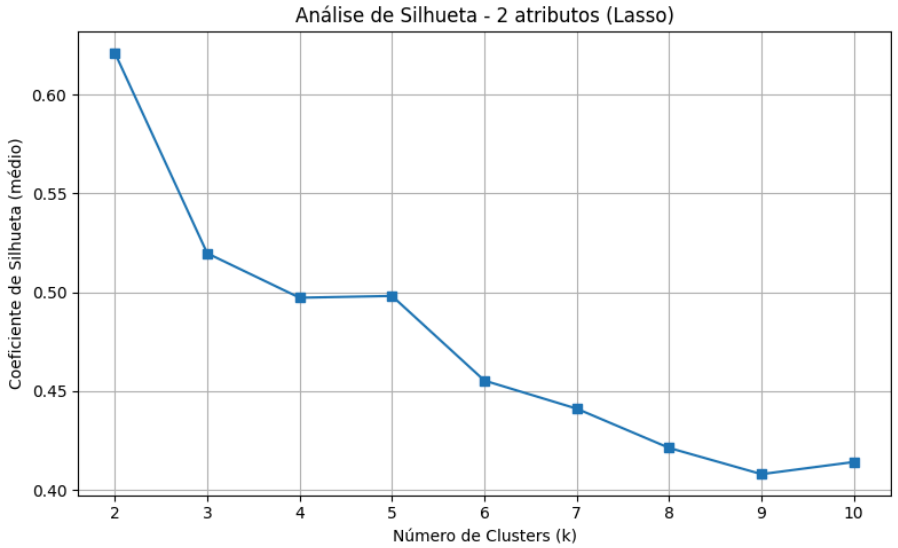


Após a execução das atividades propostas, foram obtidos os seguintes resultados utilizando apenas dois atributos selecionados via **Lasso**, que foram as colunas **i** e **r**:

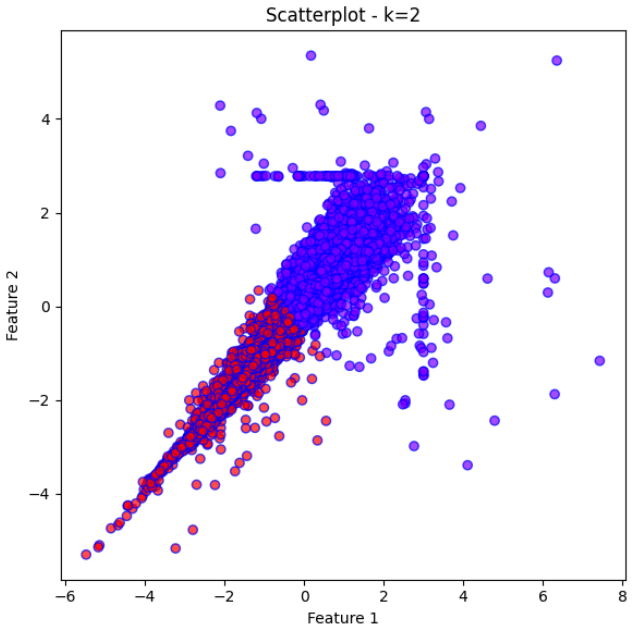
Método do Cotovelo (Elbow): Observa-se que a inércia decresce substancialmente à medida que **k** aumenta de 2 a 3 ou 4, mas novamente não há um ponto de "cotovelo" estritamente definido, mas de forma subjetiva escolheria o **4**. Gráfico:



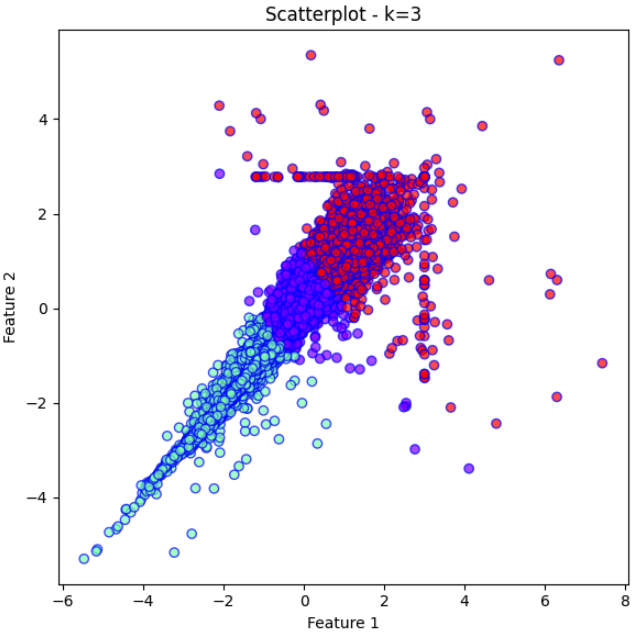
Índice de Silhueta (Silhouette): O coeficiente de silhueta atinge seu pico em **k=2**, porém cai de forma constante para valores superiores, porém usar apenas dois clusters pode ser pouco informativo, logo, semelhante a questão 2, poderia seria melhor optar por outro valor, nesse caso o **3**. Gráfico:



Scatterplots: Em **k=2**, a divisão dos dados mostra dois grandes grupos (muitas vezes um com maior densidade de pontos e outro relativamente menor). Gráfico:



Em **k=3**, há uma separação adicional de um subgrupo, indicando uma possível estratificação mais refinada, como visto no segundo scatterplot. Gráfico:



Diante desses resultados, pode-se concluir que, segundo o método de silhueta, **k=2** seria o número de clusters mais coeso, porém o método do cotovelo sugere que **3 ou 4** clusters podem ser mais adequados para capturar maiores nuances nos dados. Assim, os resultados dessa questão não diferem da **questão 2**.

d. Questão 4

Nesta questão, foi realizada uma análise de **clusterização** com o objetivo de gerar um **crosstab** entre as **classes originais** e os clusters identificados pelo método de **silhueta**, com os outros passos sendo semelhantes aos adotados na questão anterior. O fluxo de atividades se deu dessa maneira:

Carregamento do Dataset Ajustado: Inicia-se com o arquivo *star_classification_ajustado.csv*, já contendo a coluna alvo *class*.

Seleção de Atributos via Lasso: Mantém-se a coluna ***class*** como variável-alvo no **Lasso**. As demais colunas (features) compõem a matriz de entrada.

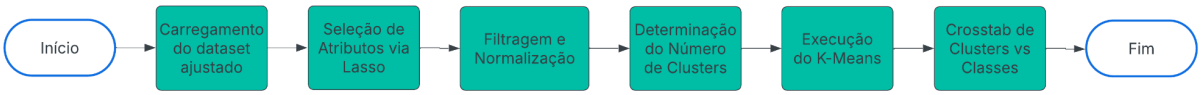
Filtragem e Normalização: Após identificar as duas colunas principais (***top2_features***), descarta-se as outras colunas e mantém-se somente as duas selecionadas. Aplica-se a técnica de normalização ***StandardScaler***.

Determinação do Número de Clusters: Realiza-se uma busca do melhor **k** (número de clusters) com base no índice de **silhueta** (Silhouette). Testam-se valores de **k** de 2 até 10 e é selecionado então o **k** que maximiza a silhueta média.

Execução do K-Means: Com o melhor **k** escolhido, roda-se o K-Means novamente sobre os dois atributos selecionados, atribuindo cada objeto a um cluster específico.

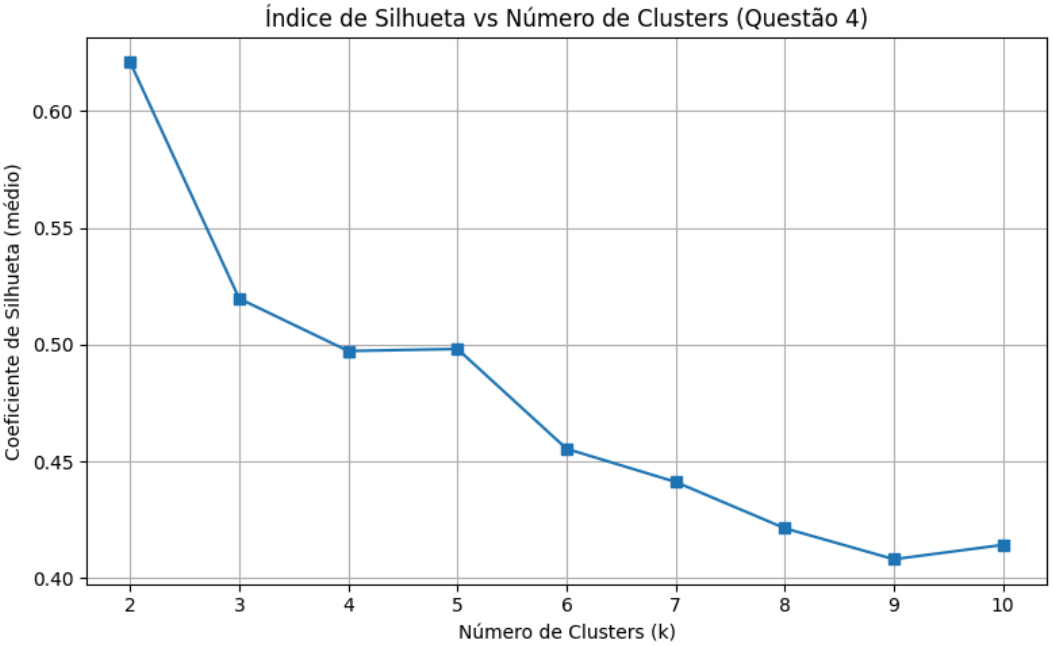
Crosstab de Clusters vs Classes: Para verificar como cada **cluster** se relaciona com as **classes** originais, gera-se uma tabela de contingência (**crosstab**) comparando os rótulos de **cluster** com a coluna ***class***.

Fluxograma de Atividades:



Após a execução das atividades propostas, foram obtidos os seguintes resultados utilizando apenas dois atributos selecionados via **Lasso**, que foram as colunas **i** e **r** (mesmo da **questão 3**):

Gŕafico de índice de Silhueta:



Crosstab entre classes originais e clusters gerados com k = 2:

Classe(y) & Clusters(x)	**0**	**1**
0 (GALAXY)	36757	22688
1 (STAR)	10380	11214
2 (QSO)	17499	1462

Os resultados de **silhueta** indicaram que a maior pontuação ocorreu para **k = 2**, assim como na **questão 3**, só que com um índice abaixo de 1 (**~0,60%)**.

Foi então gerada uma tabela **crosstab** cruzando as classes originais (coluna class) com os clusters formados. Na maioria das linhas, observou-se que cada cluster agrupa principalmente dois subconjuntos de classes, ainda que com alguma sobreposição. Esse resultado sinaliza que, na divisão em 2 clusters, os dados separam-se em dois grandes grupos coerentes, embora não tenham tanta semelhança com as classes originais.

Dessa forma, a análise revelou que a configuração **k = 2** maximiza o índice de silhueta ao usar apenas os atributos **i** e **r**, porém cada cluster resultante acaba mesclando instâncias de classes diferentes. Se o objetivo for capturar subgrupos mais específicos, pode ser necessário escolher um valor de **k** um pouco maior para aproximação das classes originais.

4. Conclusões

Em síntese, o uso de métodos de **classificação** (**KNN**) e **clusterização** (**K-Means**) em conjunto com a **normalização** de dados, seleção de atributos por **Lasso** e validações via **GridSearchCV** ou métodos de **silhueta/cotovelo** demonstrou a capacidade de extrair informações expressivas do dataset. Foi possível observar que, mesmo métodos comuns como **KNN** e **K-Means**, quando aliados a estratégias de pré-processamento (logarítmica, standard scaling, etc.) e seleção cuidadosa de hiperparâmetros, podem resultar em taxas de acerto e separação de grupos satisfatórias. As divergências entre **AV1** e **AV2** (por exemplo, **Manhattan** vs. **Euclidiana**) reforçam que diferentes métricas de distância apresentam desempenhos muito próximos. No cenário de **clusterização**, apesar de a silhueta sugerir **k = 2** em alguns casos, a decisão final sobre o número de clusters deve levar em conta a aplicação prática e a relação com as **classes originais**.

5. Próximos passos

Explorar valores adicionais de normalização (como MinMaxScaler ou técnicas robustas a outliers) para verificar eventuais melhorias.

Investigar se há outras formas de clusterização para que os clusters criados possam se assemelhar mais às classes originais ou até mesmo gerar melhores classificações.