UNIVERSIDADE FEDERAL DO CEARÁ

TECHNICAL REPORT

Aluno: Esdras Souza dos Santos

1. Introdução

Classificação: Gender Classification Dataset

Gender Classification Dataset é um dataset simples com 7 colunas com dados gerados.

As colunas deste dataset são:

long_hair - Esta coluna contém 0's e 1's onde 1 é "cabelo longo" e 0 é "cabelo não longo".

cheek_width_cm - Esta coluna está em centímetros, e é a largura da testa.

cheek_height_cm - Esta é a altura da testa e está em centímetros.

nose_wide - Esta coluna contém 0's e 1's onde 1 é "nariz largo" e 0 é "nariz não largo".

nose_long - Esta coluna contém 0's e 1's onde 1 é "nariz longo" e 0 é "nariz não longo".

lips_thin - Esta coluna contém 0's e 1's onde 1 representa os "lábios finos" enquanto 0 é "lábios não finos".

distance_nose_to_lip_long - Esta coluna contém 0's e 1's onde 1 representa a "longa distância entre o nariz e os lábios" enquanto 0 é a "curta distância entre o nariz e os lábios".

gender - Pode ser "Masculino" ou "Feminino".

Nesse dataset o objetivo é descobrir se o gênero é masculino ou feminino com base nos dados fictícios que estão no dataset.

2. Observações

Alguns dados podem variar se o código for rodado novamente.

3. Resultados e discussão

3.1. Questão 1

Na primeira questão tem que descobrir o número de k utilizando o Grid Search CV do scikit-learn.

Primeiro comecei carregando o dataset importando ele do src.utils, pois este dataset foi utilizado anteriormente e os importe já estava no utils.

Logo em seguida fiz o procedimento padrão para o knn, iniciei 'X' com o dataset sem a variável alvo e iniciei 'y' com os valores da variável alvo 'gender'. Logo em seguida o dataset foi dividido em treino e teste usando o train test split e iniciado o modelo knn.

Depois foi definidos os parâmetros a serem testados, que seria:

- **n_neighbors:** 3,5,7,9,11;
- **metric:** euclidean, manhattan, minkowski.

Também foi iniciado o KFold com n-splits igual a 5, o embaralhar ativado e com o random state igual a 42.

Em seguida o Grid Search CV foi iniciado e utilizado o fit com o 'x' e 'y' de treino.

Os resultados foram:

Melhores parâmetros: metric	Manhattan
Melhores parâmetros: n neighbors	9
Melhor acurácia média:	0.9730000000000001
Acurácia no conjunto de teste:	0.968031968031968

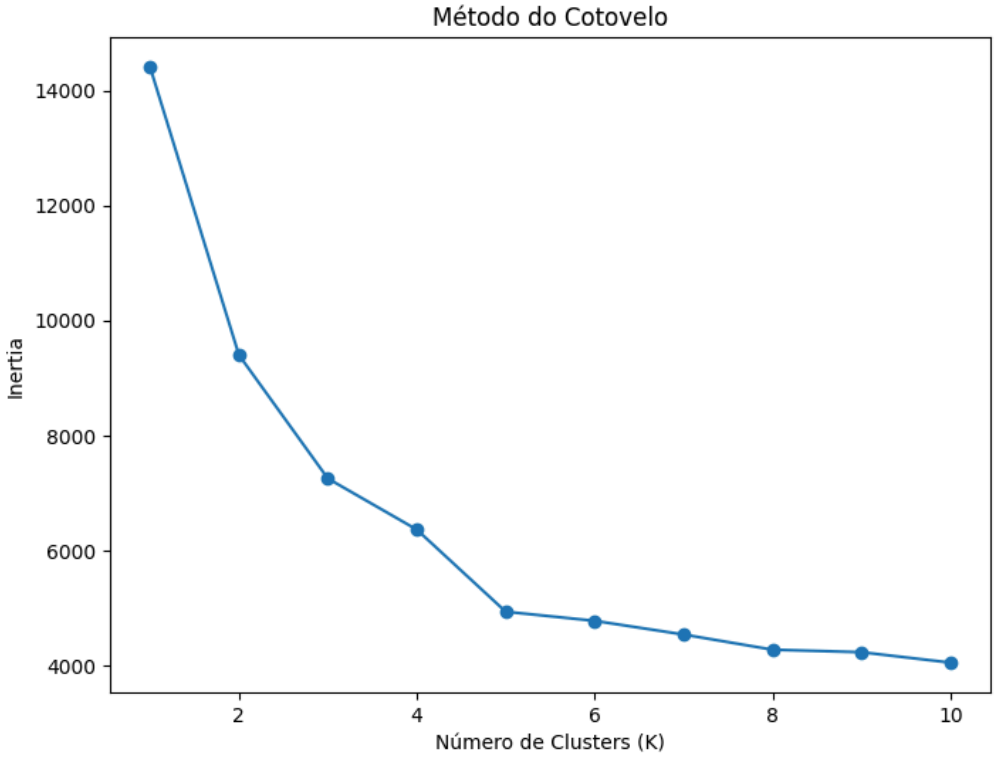
Analisando os resultados adquiridos pelo Grid Search CV verifiquei se o resultado foi igual ao da Prova1. Os olhando resultados da Prova1, que foram: métrica Euclidiana e mais ou menos 8 em número de vizinhos. Ou seja, os resultados foram diferentes.

3.2 Questão 2

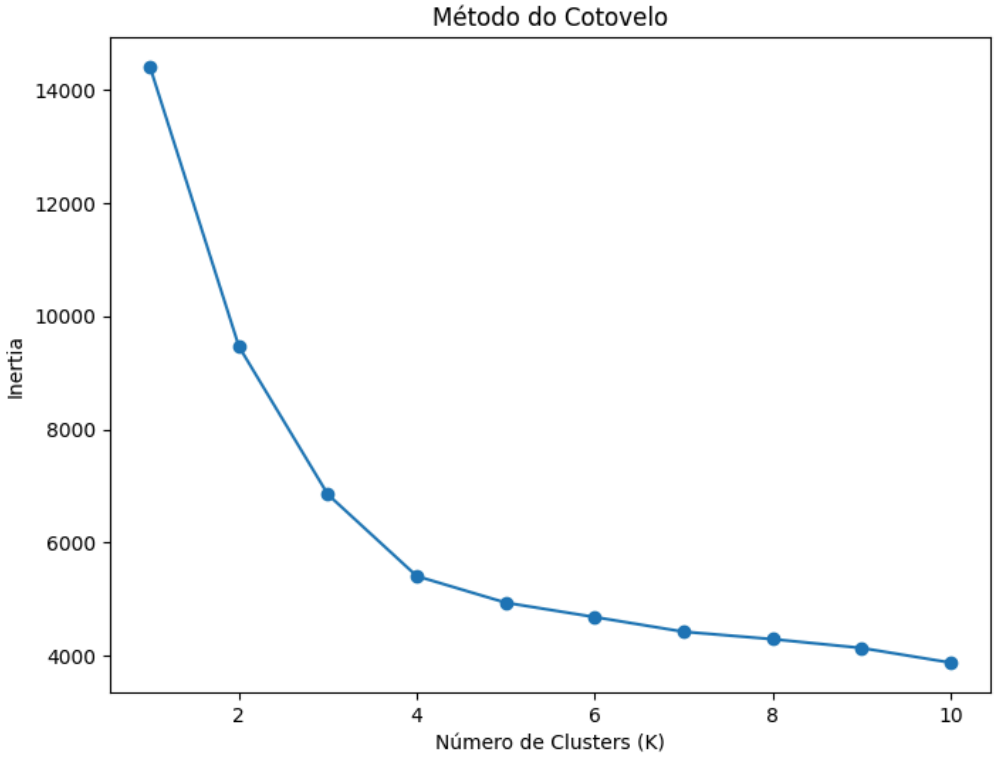
Na questão 2 temos que descobrir o número ideal de clusters, utilizando o método do cotovelo e da silhueta.

Comecei carregando o dataset e depois iniciando duas variáveis para armazenar os valores para os gráficos de inércia e o de silhueta.

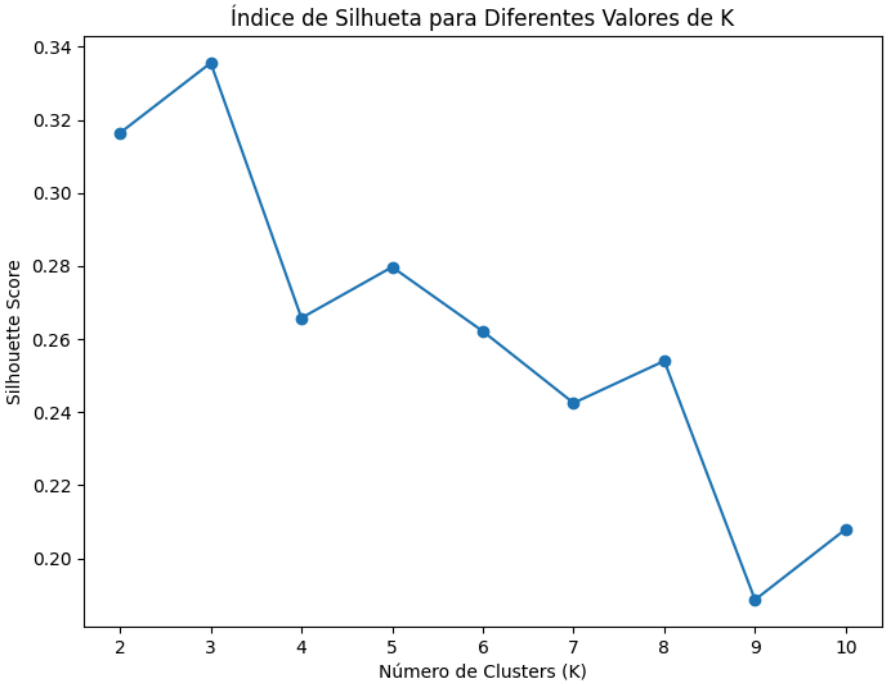
Para o grafico de inercia(cotovelo) começei iniciando o modelo do K Means com k de 1 a 11 dando fit e salvando na variável do grafico. Depois plotei o grafico.

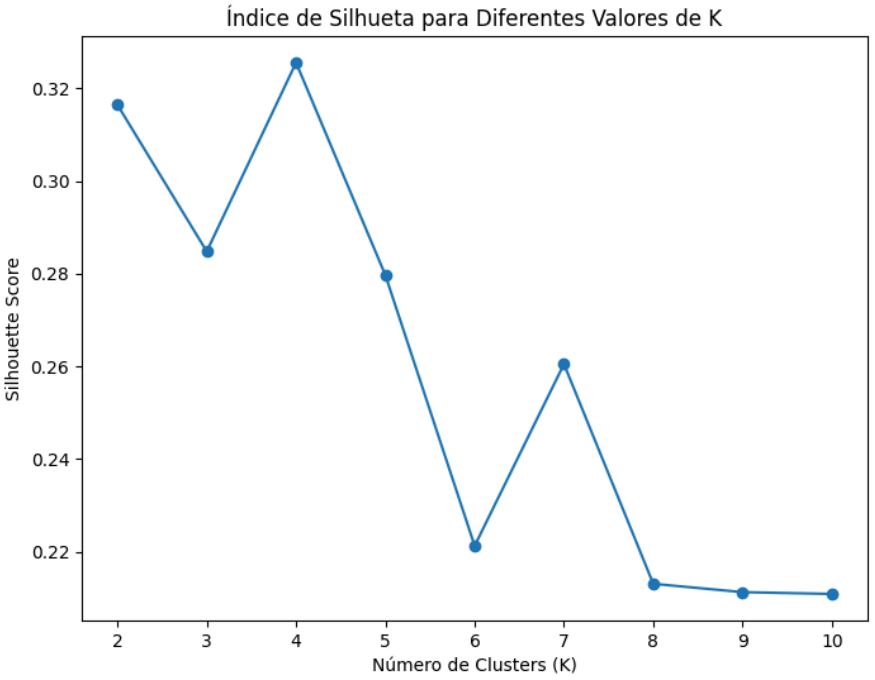


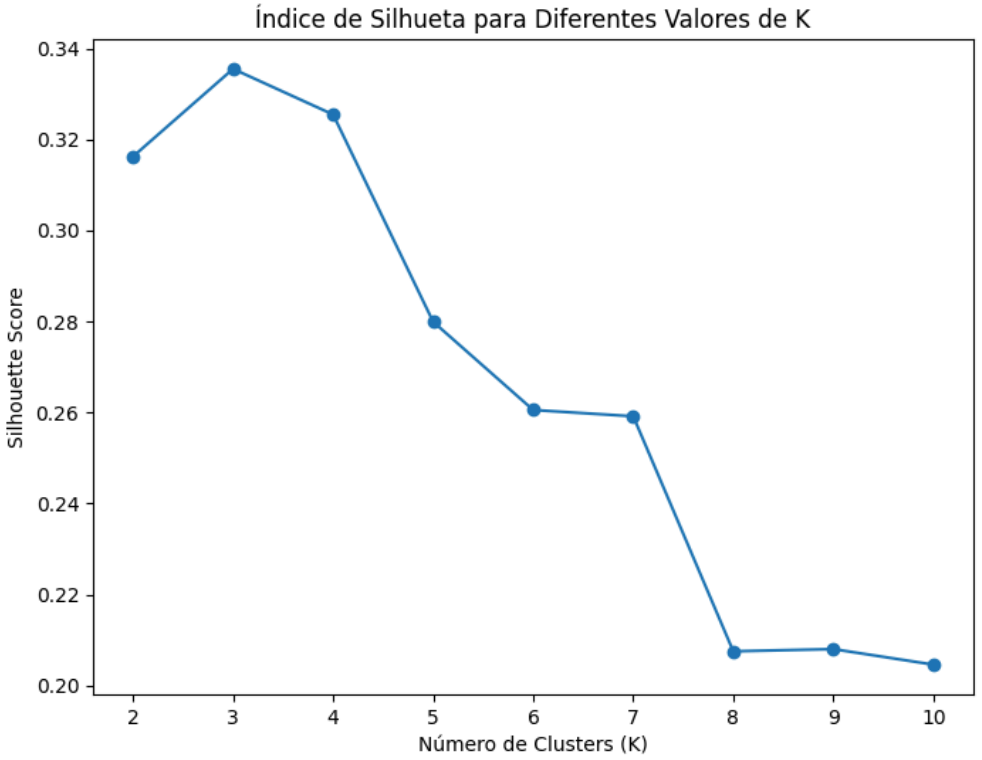
Bem, como era um dataset de classificação, ele possuía muitos 1 e 0 em várias colunas então os resultados variam a cada vez que o código era rodado, mesmo não tendo nada no código que fosse randomizado(Não que eu tenha visto). A variação está a seguir:



Para o gráfico de silhueta eu fiz do mesmo jeito do gráfico de cotovelo, mudando apenas o fato de o valor do ser de 2 a 11 e ter uma linha de código a mais. Os resultados retornados foram esse:


Índice de Silhueta para Diferentes Valores de K
Silhouette Score
0.34
0.32
0.30
0.28
0.26
0.24
0.22
0.20
2
3
4
5
6
7
8
9
10
Número de Clusters (K)


Índice de Silhueta para Diferentes Valores de K
Silhouette Score
0.32
0.30
0.28
0.26
0.24
0.22
2
3
4
5
6
7
8
9
10
Número de Clusters (K)



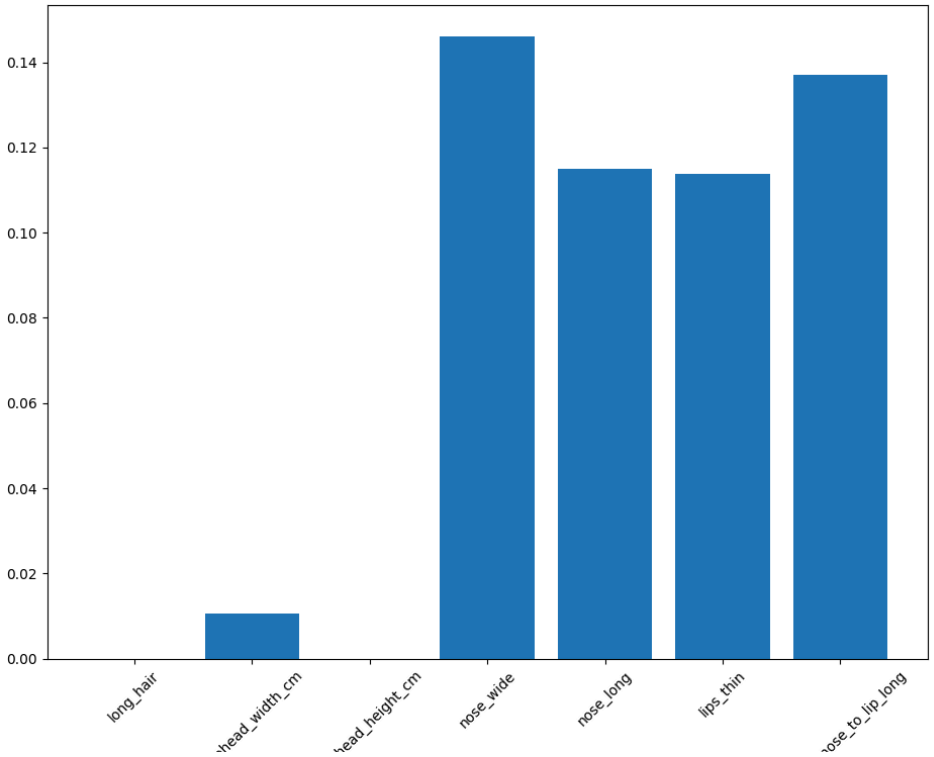
Bom verificando os resultados dos dois gráficos. O método do cotovelo indica o número de k de 4 ou 5 enquanto o método da silhueta indica um número entre 3 e 4.

3.3 Questão 3

Na questão 3 temos que usar o método de Lasso para escolher as duas variáveis mais relevantes e recalcular a quantidade de cluster com os métodos de cotovelo e de silhueta.

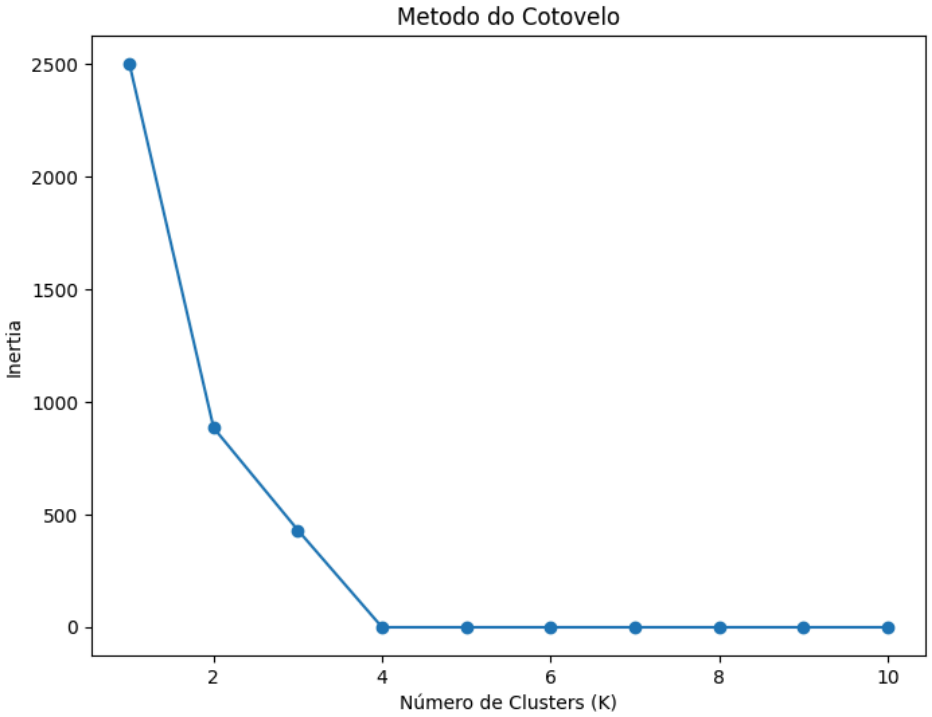
Foi Iniciado o dataset e separado em ‘X’ e ‘y’. além de ser separado as colunas do dataset para ser usado no método Lasso.

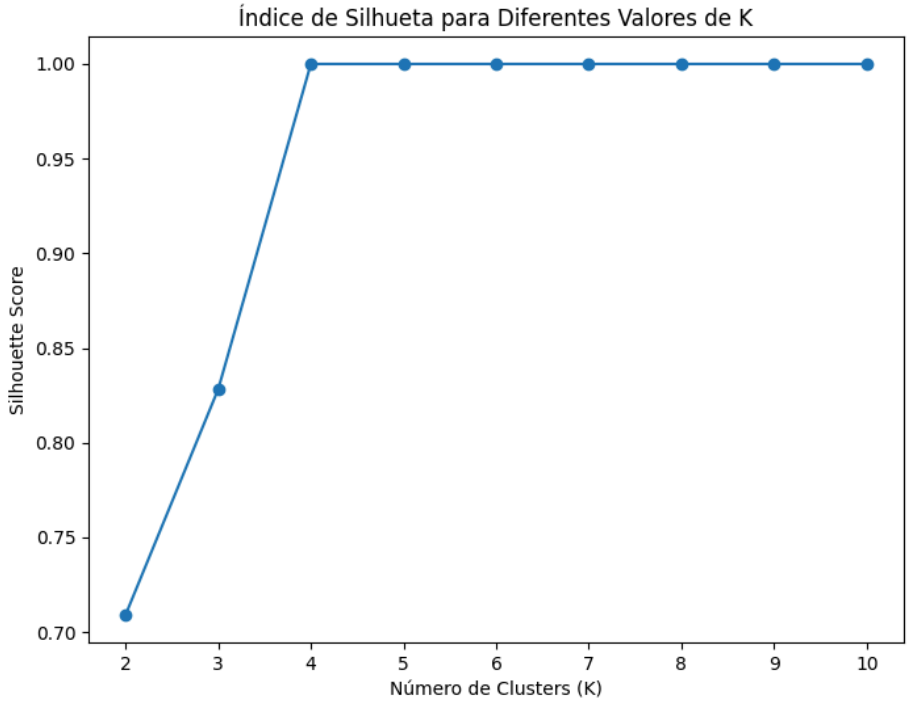
O método Lasso é iniciado e computado o coeficiente dele e logo em seguida plotado, esse é o resultado:



As duas colunas mais relevantes foram: ‘nose_wide’ e ‘distance_nose_to_lip_long’.

Adicionando apenas as duas colunas em uma variável nova, iniciei o mesmo processo usado na questão anterior para criar o gráfico de cotovelo e de silhueta e ficou assim:

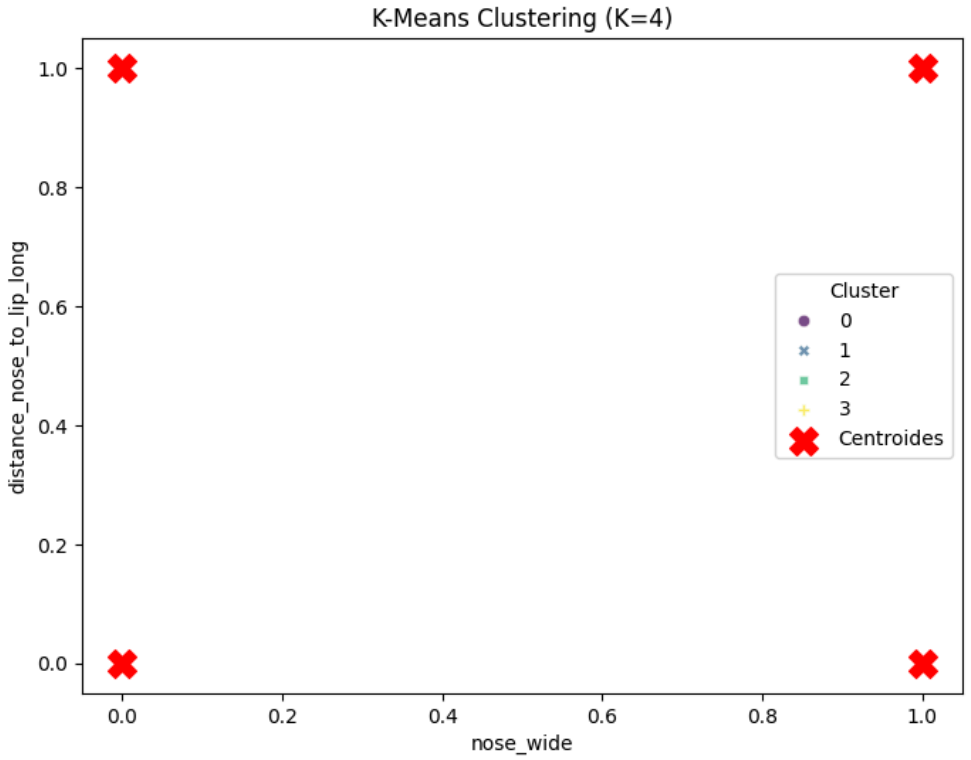

Metodo do Cotovelo
Inertia
2500
2000
1500
1000
500
0
2
4
6
8
10
Número de Clusters (K)


Índice de Silhueta para Diferentes Valores de K
Silhouette Score
1.00
0.95
0.90
0.85
0.80
0.75
0.70
2
3
4
5
6
7
8
9
10
Número de Clusters (K)

Como podemos ver alguma coisa não parece estar certo. Essa linha reta nos dois gráficos aconteceu porque os valores de ‘nose_wide’ e ‘distance_nose_to_lip_long’ são valores binarios, ou seja, 1 's e 0' s. Por serem valores binários, o código que cria os gráficos retorna um erro dizendo que os valores depois do ponto 4 são possivelmente duplicados.

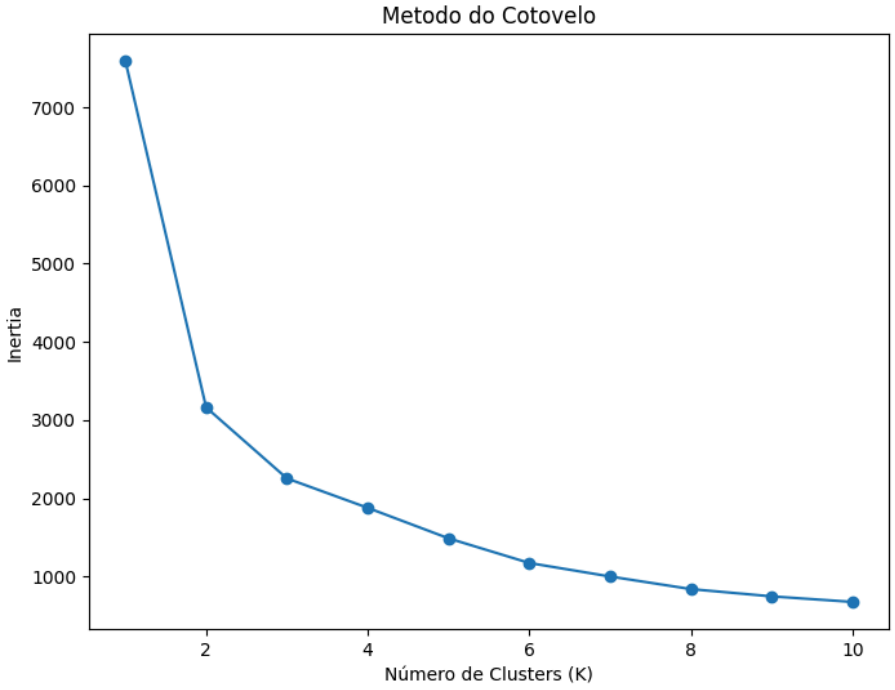
Mas o resultado comparado com os resultados da questão anterior foi parecido, pois os dois resultam dizendo que o melhor é 4 clusters.

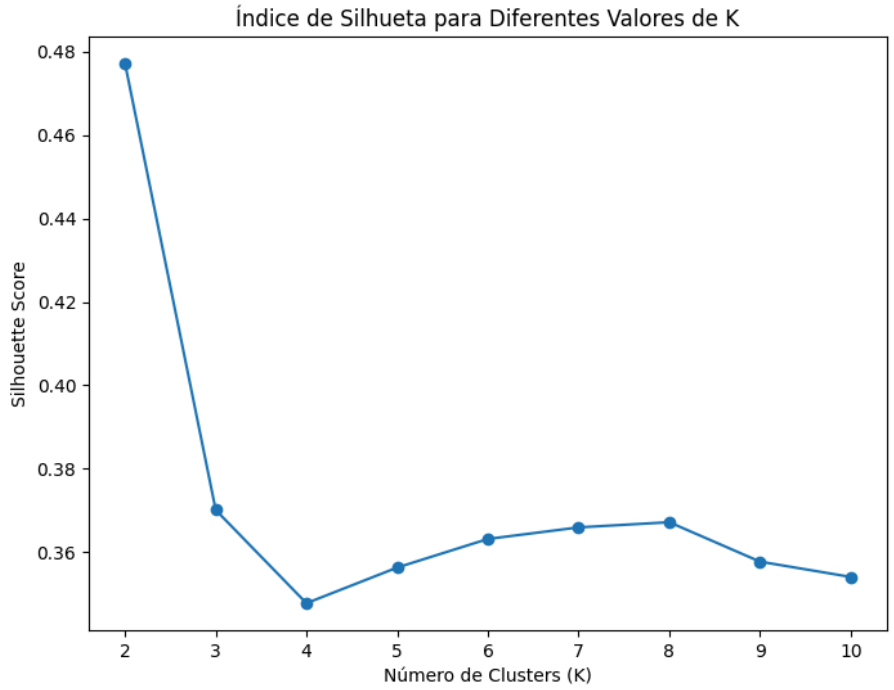
Não era necessário mais fiz o Scatter desse resultado para apenas verificar como tava, e assim ele está:

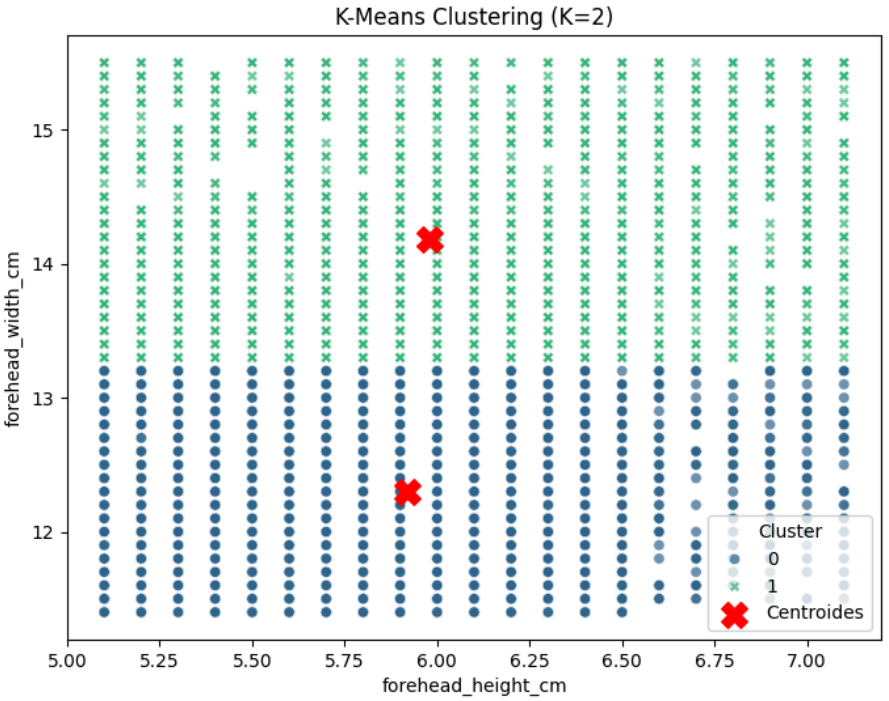


3.3.1 Colunas Diferentes

Estudando sobre o erro que deu no gráfico de cotovelo e de silhueta, eu vi que para criar esse gráficos sem nenhum problema é bom usar dados numéricos. Olhando o meu dataset eu vi que ele tem apenas duas colunas com dados numéricos então decidi adicionar eles e ver como seria os três gráficos apresentados anteriormente.


Metodo do Cotovelo
Inertia
7000
6000
5000
4000
3000
2000
1000
2
4
6
8
10
Número de Clusters (K)


Índice de Silhueta para Diferentes Valores de K
Silhouette Score
0.48
0.46
0.44
0.42
0.40
0.38
0.36
2
3
4
5
6
7
8
9
10
Número de Clusters (K)



3.4 Questão 4

Na questão 4 temos que fazer um crosstab e verificar como ficou a distribuição de cluster.

Comecei carregando o dataset e iniciando uma variável com as colunas mais relevantes. Adicionei o valor da coluna alvo em uma variável e iniciei o KMeans com 4 clusters como decidido na questão anterior. Os resultados do crosstab foi:

	Female	Male
0	1941	39
1	31	1913
2	273	278
3	256	270

Bom, como a coluna alvo só tem dois valores 0 e 1, usar quatro clusters dá um resultado estranho, mas levando em consideração apenas as duas primeiras linhas eu diria que o resultado foi satisfatório. Obs: quando se roda o código novamente os

valores da linha 0 e da linha 1 trocam de lugar, mas os da linha 2 e 3 permanecem idênticos.

3.4.1 Dois clusters

Usando apenas dois cluster para combinar com a coluna alvo o resultado fica assim:

	Female	Male
0	2470	587
1	31	1913

Tendo 2470 e 1913 Positivos, 587 Falsos Negativos e 31 Falsos Positivos. Obs: os dados também invertem quando o código é rodado novamente.

4. Conclusões

Com todos os resultados, eu diria que os resultados não foram satisfatórios.

Na questão 1 os resultados foram muito altos para o número de vizinhos, além de que o resultado da métrica é diferente do da Prova1.

Na questão 2 os resultados variados que ele apresentava em cada execução do código podem ser confusos e não dão um valor exato.

A questão 3 foi a melhor, mas os erros que as colunas causam no código podem facilmente enganar e parecer que está errado.

E na questão 4 como o número de cluster escolhido pela questão 3 é maior do que a variância da coluna alvo, não consigo interpretar os dados do crosstab.
E depois de colocar o número de resultados da coluna alvos no número de clusters, os dados ficam variando entre os resultados certo e os resultados errados, ou seja, toda vez que executar o código não dá para saber se os dados vão vir certo ou não.

Por causa de todos esses incômodos e erros que acontecem quando usamos o dataset, eu considero esses resultados Não Satisfatórios.

5. Próximos passos

Uma escolha de Dataset de regressão por não ter ou ter poucas colunas binárias.