PÓS-GRADUAÇÃO MIT EM INTELIGÊNCIA ARTIFICIAL, MACHINE LEARNING & DEEP LEARNING

Bloco: Clusterização de Dados [22E4-22E4]

Disciplina: Validação de modelos de clusterização [22E4_3]

Docente: Luiz Fernando Frias

Aluno: Winicius Botelho Faquieri

Infraestrutura

Para as questões a seguir, você deverá executar códigos em um notebook Jupyter, rodando em ambiente local, certifique-se que:

- ✓ Você está rodando em Python 3.9+
  Estou utilizando a versão 3.10.5.
- ✓ Você está usando um ambiente virtual: Virtualenv ou Anaconda
  Para o presente trabalho criei o ambiente virtual “infnet-project-clustering-validation” (vide print).
- ✓ Todas as bibliotecas usadas nesse exercícios estão instaladas em um ambiente virtual específico.
  Sim, todas as bibliotecas estão instaladas no virtualenv criado na pasta “Lib\site-packages”.
- ✓ Gere um arquivo de requerimentos (requirements.txt) com os pacotes necessários. É necessário se certificar que a versão do pacote está disponibilizada.

O arquivo requirements.txt está disponível em: https://github.com/wfaquieri/infnet-project-clustering-validation/requirements.txt

✓ Tire um printscreen do ambiente que será usado rodando em sua máquina.

Segue abaixo o print do ambiente virtual rodando na minha máquina:

```
C:\Users\Winicius
λ cd OneDrive - FGV\GitHub\MIT-INFNET\08_Validacao_modelos_clusterizacao\

C:\Users\Winicius\OneDrive - FGV\GitHub\MIT-INFNET\08_Validacao_modelos_clusterizacao
λ python -m venv infnet-project-clustering-validation

C:\Users\Winicius\OneDrive - FGV\GitHub\MIT-INFNET\08_Validacao_modelos_clusterizacao
λ infnet-project-clustering-validation\Scripts\activate

C:\Users\Winicius\OneDrive - FGV\GitHub\MIT-INFNET\08_Validacao_modelos_clusterizacao
(infnet-project-clustering-validation) λ ipython kernel install --user --name=infnet-project-clustering-validation
Installed kernelspec infnet-project-clustering-validation in C:\Users\Winicius\AppData\Roaming\jupyter\kernels\infnet-project-clustering-validation

C:\Users\Winicius\OneDrive - FGV\GitHub\MIT-INFNET\08_Validacao_modelos_clusterizacao
(infnet-project-clustering-validation) λ jupyter notebook
[I 2022-12-19 08:12:58.217 LabApp] JupyterLab extension loaded from C:\Users\Winicius\anaconda3\lib\site-packages\jupyterlab
[I 2022-12-19 08:12:58.218 LabApp] JupyterLab application directory is C:\Users\Winicius\anaconda3\share\jupyter\lab
[I 08:12:58.233 NotebookApp] The port 8888 is already in use, trying another port.
[I 08:12:58.235 NotebookApp] The port 8889 is already in use, trying another port.
[I 08:12:58.243 NotebookApp] Serving notebooks from local directory: C:\Users\Winicius\OneDrive - FGV\GitHub\MIT-INFNET\08_Validacao_modelos_clusterizacao
[I 08:12:58.244 NotebookApp] Jupyter Notebook 6.4.8 is running at:
[I 08:12:58.245 NotebookApp] http://localhost:8890/?token=91a18397859c43e5af3a95e5a4986a8cce522ed9de8525f2
[I 08:12:58.249 NotebookApp]  or http://127.0.0.1:8890/?token=91a18397859c43e5af3a95e5a4986a8cce522ed9de8525f2
[I 08:12:58.250 NotebookApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation).
[C 08:12:58.477 NotebookApp]

    To access the notebook, open this file in a browser:
        file:///C:/Users/Winicius/AppData/Roaming/jupyter/runtime/nbserver-2044-open.html
    Or copy and paste one of these URLs:
        http://localhost:8890/?token=91a18397859c43e5af3a95e5a4986a8cce522ed9de8525f2
     or http://127.0.0.1:8890/?token=91a18397859c43e5af3a95e5a4986a8cce522ed9de8525f2
[I 08:13:11.763 NotebookApp] Creating new notebook in /infnet-project-clustering-validation
[I 08:13:19.034 NotebookApp] Kernel started: b24bbb18-490d-45fd-a6fa-1508df2939e8, name: infnet-project-clustering-validation
```

- ✓ Disponibilize os códigos gerados, assim como os artefatos acessórios (requirements.txt) e instruções em um repositório GIT público. (se isso não for feito, o diretório com esses arquivos deverá ser enviado compactado no moodle).
  O notebook com os códigos gerados, assim como os artefatos acessórios estão disponíveis em: https://github.com/wfaquieri/infnet-project-clustering-validation

Escolha de base de dados

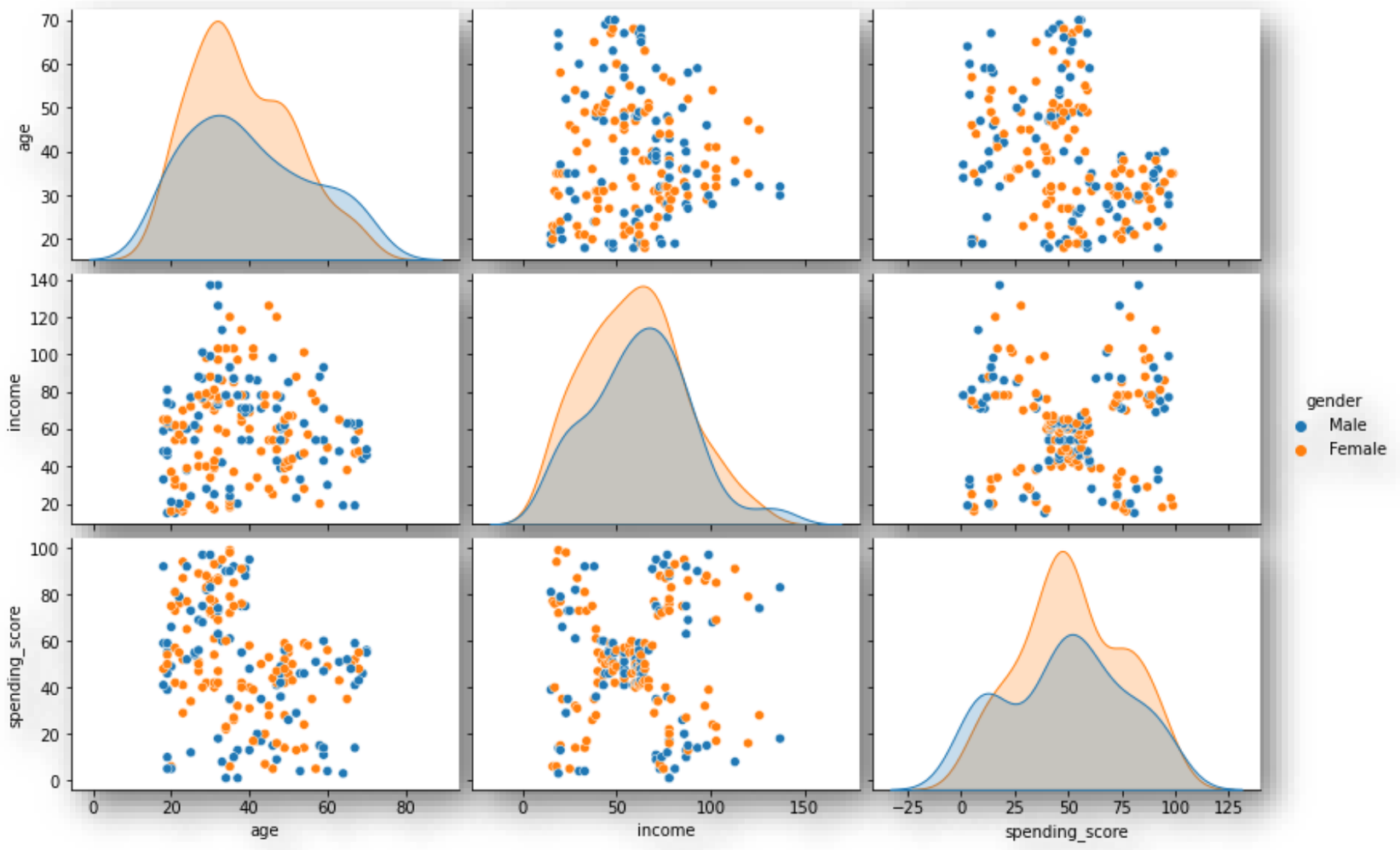
Para as questões a seguir, usaremos uma base de dados e faremos a análise exploratória dos dados, antes da clusterização.

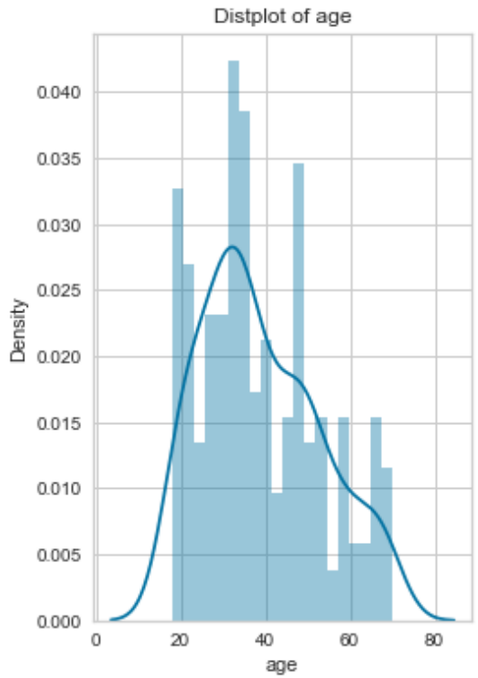
1. Escolha uma base de dados para realizar o trabalho. Essa base será usada em um problema de clusterização. ✔
   A base escolhida está disponível em: https://www.kaggle.com/datasets/vjchoudhary7/customer-segmentation-tutorial-in-python, bem como no repositório git na pasta 'data-raw'.

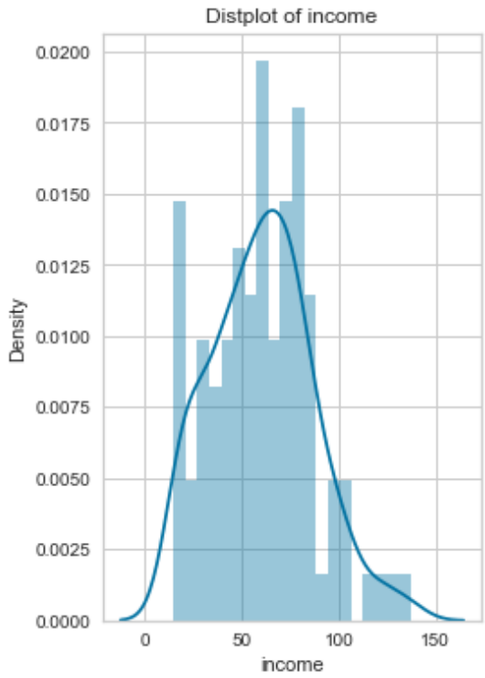
2. Escreva a justificativa para a escolha de dados, dando sua motivação e objetivos.
   A opção por esse conjunto de dados se deve a dois fatores principais. Primeiro, para obter alguma experiência com projetos de segmentação de clientes, pois há uma demanda do mercado para resolução desse tipo de problema; Além disso, procurei trabalhar, nesse momento, com um *dataset* simples, bem comportado, com baixa dimensionalidade, para fixar o conteúdo estudado ao longo do curso.

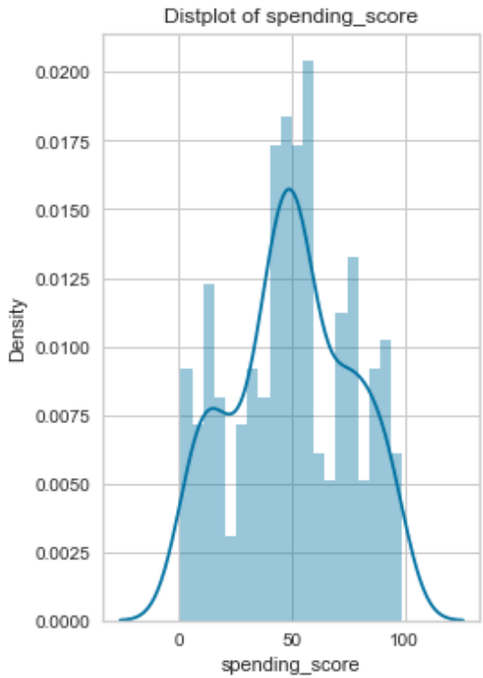
3. Mostre através de gráficos a faixa dinâmica das variáveis que serão usadas nas tarefas de clusterização. Analise os resultados mostrados. O que deve ser feito com os dados antes da etapa de clusterização?

A figura a seguir exibe a faixa dinâmica das variáveis que serão usadas nas tarefas de clusterização:









Antes da etapa de clusterização, é muito importante padronizar as features por causa da maneira como os clusters são calculados. Em geral, as variáveis em um conjunto de dados estão em unidades diferentes e variam em faixas específicas. Se houver uma variável com um intervalo de valores muito maior do que outra, essa diferença pode influenciar fortemente os resultados do agrupamento. Além disso, cada variável tem significado próprio em um dataset. Algumas vezes estamos lidando com renda (exemplo: R$), outras com altura (exemplo: cm), peso (exemplo: kg), etc, ou seja, variáveis que não são diretamente comparáveis.

4. Realize o pré-processamento adequado dos dados. Descreva os passos necessários.

   É possível transformar dados de várias *features* para a mesma escala normalizando os dados. Em particular, a normalização é adequada para processar a distribuição de dados mais comum, a distribuição gaussiana. Logo, o primeiro passo é **verificar a distribuição de cada *feature*** e, assim, definir o tratamento mais adequado. Além disso, é importante **observar se existem nos dados valores atípicos** e se para análise faz sentido manter como estão, remover (*truncation*) ou tratar.

   Como dito anteriormente, em casos onde a distribuição é gaussiana ou normal, você pode normalizar os dados, reescalando o centro para zero e variância igual a 1. Para isso, basta subtrair o valor pela média e dividir pelo desvio-padrão. Para distribuições não gaussianas, você pode utilizar outras técnicas:

- Rescaling: reescala para o intervalo 0 e 1;
- Min-Max Scaler: utiliza o range;
- Robust Scaler: utiliza o IQR;
- Entre outras.

Na presença de outliers, pode-se utilizar métodos robustos, como o Robust Scaler. Este Scaler remove a mediana e dimensiona os dados de acordo com o intervalo quantil (o padrão é IQR: intervalo interquartil). O IQR é o intervalo entre o 1º quartil (25º quantil) e o 3º quartil (75º quantil).

Clusterização

Para os dados pré-processados da etapa anterior você irá:

1. Realizar o agrupamento dos dados, escolhendo o número ótimo de clusters. Para tal, use o índice de silhueta e as técnicas:
    1. K-Means
    2. DBScan

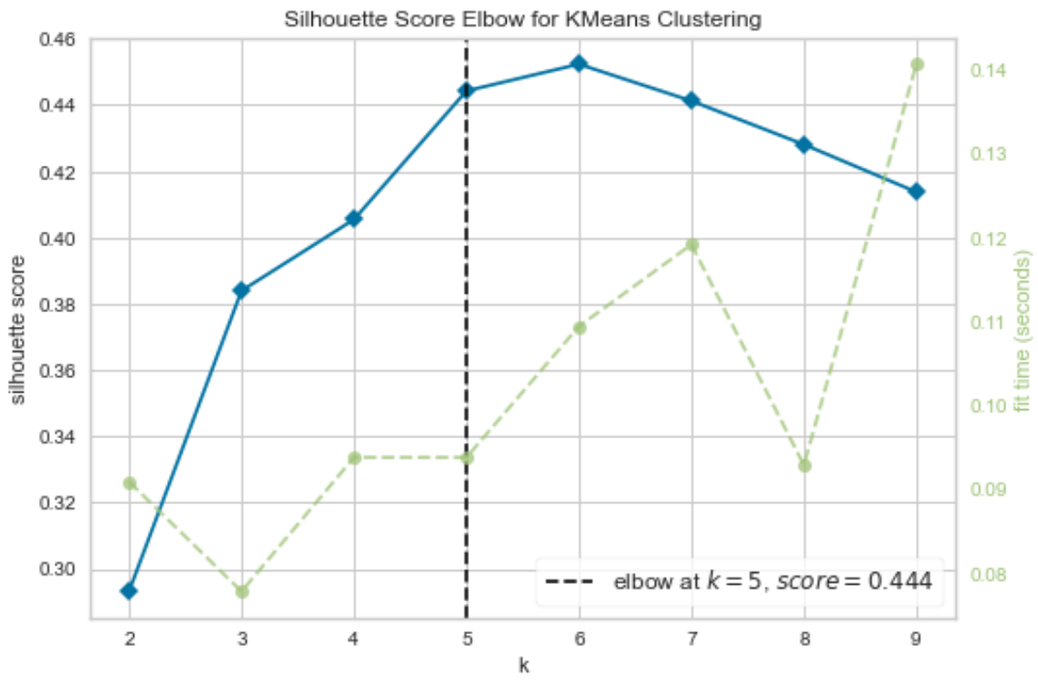
Com os resultados em mão, descreva o processo de mensuração do índice de silhueta. Mostre o gráfico e justifique o número de clusters escolhidos.

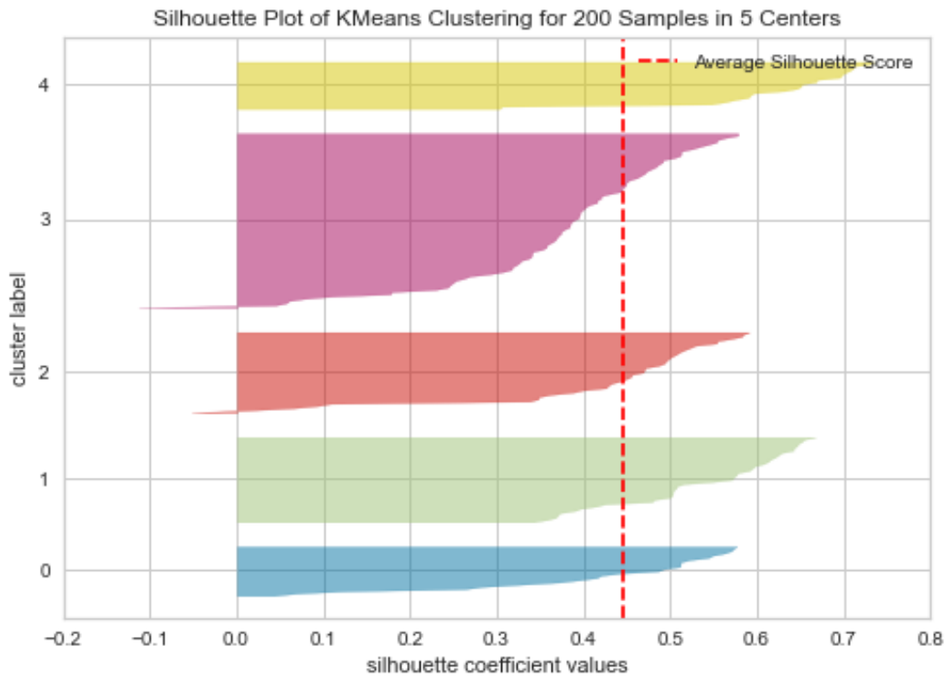
O índice de silhueta analisa a coesão – distância intra-cluster – e separação entre clusters. Responde a seguinte pergunta: a que distância estão os pontos no cluster mais próximo (b), em relação aos pontos no cluster (a)? Para isso, compara a distância média intra-cluster (a) com a distância média do cluster mais próximo (b) e calcula a seguinte pontuação s:

$$s = (b - a)/\max(a, b)$$

A pontuação pode variar entre -1 e 1. Caso a = b, significa que a amostra está em região de confusão, ou seja, não há uma separação apropriada entre os clusters e muitos pontos de fronteira. Com a << b, temos a situação ideal, onde o índice se aproxima de 1. Isto significa que há uma boa separação entre os clusters que me permite ter grupos bem definidos e coesos. Com a >> b, o índice se aproxima de -1 e a amostra provavelmente foi classificada errada.

Para implementar o método “silhueta” utilizou-se a biblioteca Yellowbrick. Segue abaixo o gráfico do índice de silhueta, que exibe o coeficiente de silhueta para cada amostra por cluster, avaliando visualmente a densidade e a separação entre os clusters. Para k=5, todos os clusters apresentaram pontuação acima da média do índice de silhueta, com espessura uniforme e sem grandes flutuações no tamanho.

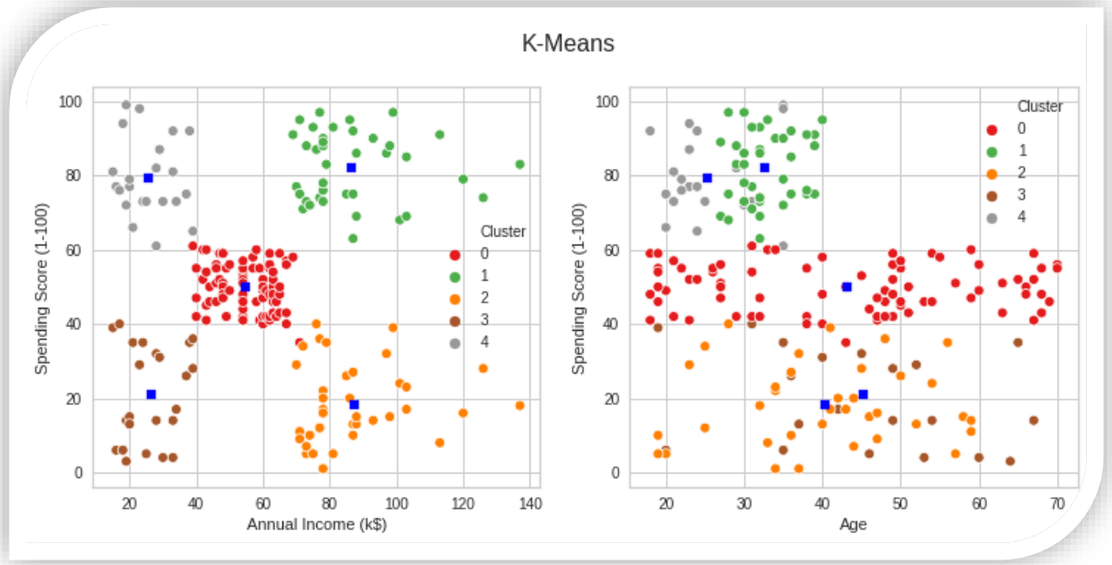


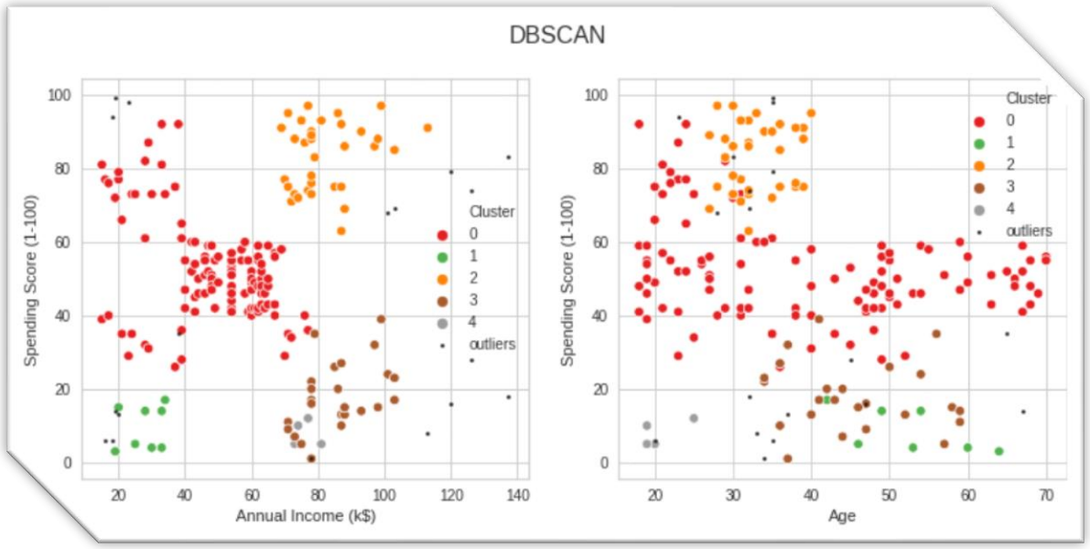


2. Compare os dois resultados, aponte as semelhanças e diferenças e interprete.

O algoritmo K-Means gerou os 5 clusters a seguir:

- Cluster 0: clientes com renda anual média e pontuação média de gastos;
- Cluster 1: clientes com alta renda anual e alta pontuação de gastos;
- Cluster 2: clientes com alta renda anual e baixa pontuação de gastos;
- Cluster 3: clientes com baixa renda anual e baixo índice de gastos;
- Cluster 4: clientes com baixa renda anual e alta pontuação de gastos;
- Não existem grupos distintos em função da idade dos clientes.


K-Means
Spending Score (1-100)
Annual Income (k$)
Age
Cluster
0
1
2
3
4


DBSCAN
Spending Score (1-100)
Annual Income (k$)
Age
Cluster
0
1
2
3
4
outliers

Já o DBSCAN não foi capaz de classificar tão bem cada agrupamento. O cluster zero, por exemplo, agrupa clientes com níveis muito diferentes de renda e pontuação. O cluster 4 tem baixa cardinalidade e se confunde com o cluster 3.

Comparando ambos os resultados a partir da cardinalidade - quantidade de elementos por cluster - pode-se observar:

- O DBSCAN criou 5 clusters mais o cluster de outliers (-1). Os tamanhos dos clusters variam de 0-4 - alguns têm apenas 4 ou 8 observações, enquanto outros clusters apresentam um número bem superior (112, por ex.). Existem 18 valores classificados como outliers;
- O KMEANS, por outro lado, gerou clusters mais uniformes, sem grandes flutuações no tamanho, em comparação com o DBSCAN.

cluster	KM_size	DBSCAN_size
0	23.0	112
1	39.0	8
2	37.0	34
3	79.0	24
4	22.0	4
-1	NaN	18

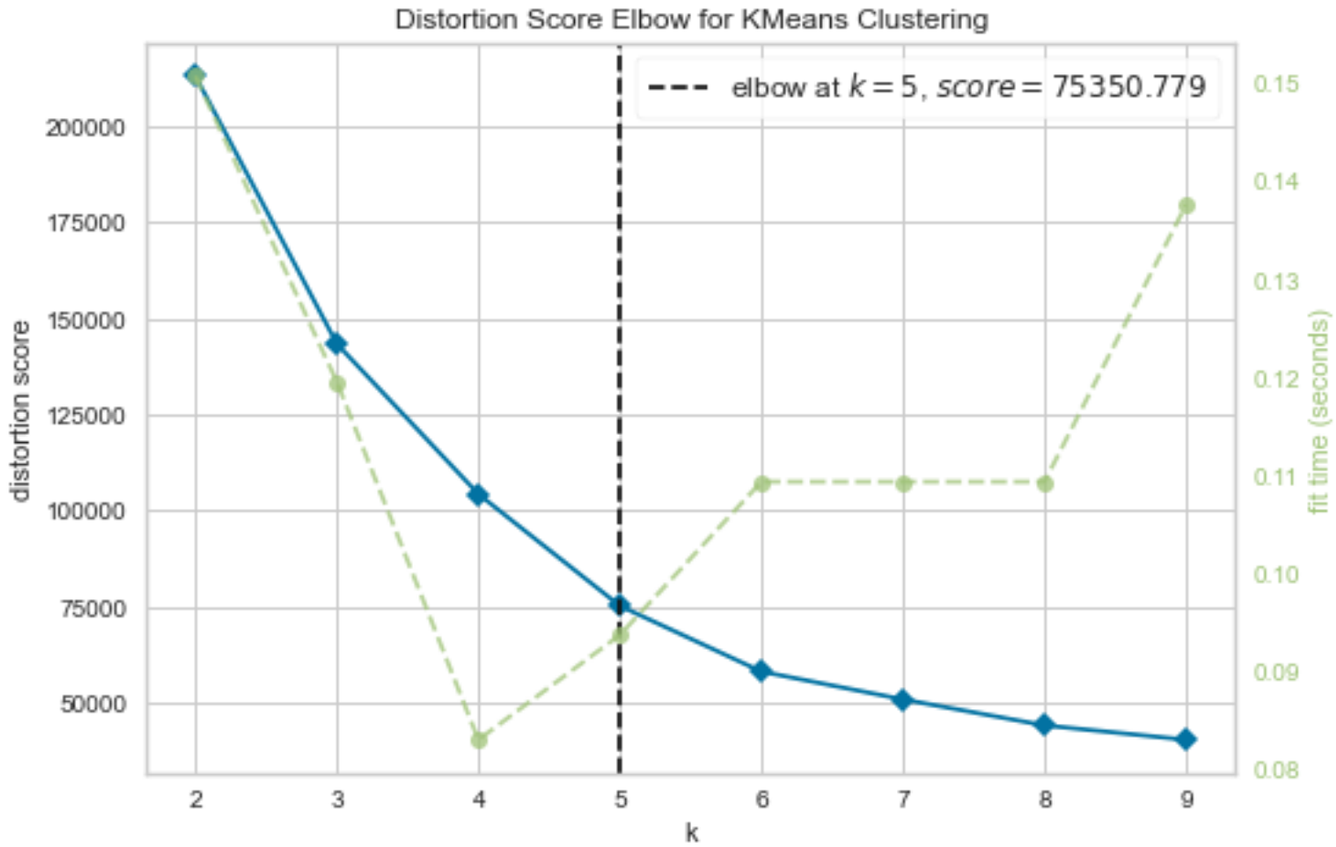
3. Escolha mais duas medidas de validação para comparar com o índice de silhueta e analise os resultados encontrados. Observe, para a escolha, medidas adequadas aos algoritmos.

   Além do índice de silhueta, foram definidas outras duas medidas de validação interna: o método do “cotovelo” - vide gráfico a seguir – e a cardinalidade, isto é, a quantidade de elementos por cluster. Podemos citar, ainda, o método DBCV. Este último foi desenvolvido para superar a limitação do índice de silhueta na medição de clusters não-convexos. Em essência, DBCV calcula dois valores:

   - A densidade dentro de um cluster;

- A densidade entre os clusters.

Alta densidade dentro de um cluster e baixa densidade entre clusters indica bons clusters.



4. Realizando a análise, responda: A silhueta é um o índice indicado para escolher o número de clusters para o algoritmo de DBScan?

   A silhueta não é um o índice indicado para escolher o número de clusters para o algoritmo de DBScan, pois possui como desvantagem gerar números maiores para clusters convexos do que para clusters baseados em densidade, o que impossibilita seu uso para comparação de diferentes modelos de clusters.

Medidas de similaridade

1. Um determinado problema, apresenta 10 séries temporais distintas. Gostaríamos de agrupá-las em 3 grupos, de acordo com um critério de similaridade, baseado no valor máximo de correlação cruzada entre elas. Descreva em tópicos todos os passos necessários.
   Uma ideia seria estudar a correlação cruzada de diferentes classes de ativos, no caso 10 séries temporais distintas, com um período de alta taxa de juros e, depois, um outro período de baixa taxa de juros. Em sua versão mais simples, ela pode ser descrita em termos de uma variável independente, X, e duas variáveis dependentes, Y e Z. Se a variável independente X influenciar a variável Y e as duas estiverem positivamente correlacionadas , então, à medida que o valor de X aumenta, também aumentará o valor de Y. Nesse caso, a variável independente X pode representar, por exemplo, um choque positivo na taxa de juros. E as duas variáveis Y e Z, a taxa de câmbio usd/brl e o par BTC/USD ou qualquer outro ativo de interesse, como ações da Amazon por ex. Se um aumento da taxa de juros (X), for seguido de movimentos de preços positivos dos ativos Y e Z, então as variáveis Y e Z podem ser consideradas correlacionadas porque seu comportamento é positivamente correlacionado como resultado de cada uma de suas relações individuais com a variável X. Em muitos casos, é necessário antes normalizar as séries, alterando o valor das colunas numéricas no conjunto de dados para uma escala comum sem distorcer as diferenças no intervalo de valores.

2. Para o problema da questão anterior, indique qual algoritmo de clusterização você usaria. Justifique.
   Utilizaria para este problema o algoritmo de clusterização Kmeans, uma vez que estamos lidando com dados não rotulados. A justificativa é que esse algoritmo além de ser amplamente utilizado é também simples de entender e fácil de interpretar, o que é sempre uma boa forma de se começar.

3. Indique um caso de uso para essa solução projetada.
   Um caso de uso é para gerenciamento de risco de um portfólio de ações, onde podemos responder a seguinte questão: qual classe de ativos seria um bom *hedge* para outra classe de ativos.

4. Sugira outra estratégia para medir a similaridade entre séries temporais. Descreva em tópicos os passos necessários.
   Uma outra abordagem é utilizar a correlação cruzada com defasagem de tempo - TLCC. Essa técnica permite identificar a direção entre dois sinais, como uma relação líder-seguidor na qual o líder inicia uma resposta que é repetida pelo seguidor. O TLCC é medido deslocando incrementalmente um vetor de série temporal e calculando repetidamente a correlação entre dois sinais. Se a correlação de pico estiver no centro (offset=0), isso indica que as duas séries temporais estão mais

sincronizadas naquele momento. É uma ótima maneira de visualizar a interação dinâmica refinada entre dois sinais, como a relação líder-seguidor e como eles mudam com o tempo, desde que os eventos sejam simultâneos e tenham em comprimentos semelhantes.