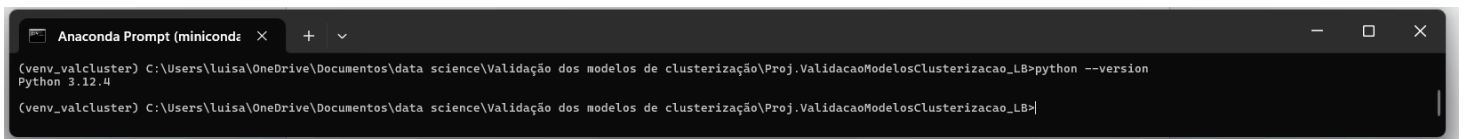


LUISA BRASIL DE MATOS - Respostas do projeto Validação dos modelos de clusterização

Infraestrutura

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

1. Você está rodando em Python 3.9+
2. Você está usando um ambiente virtual: Virtualenv ou Anaconda

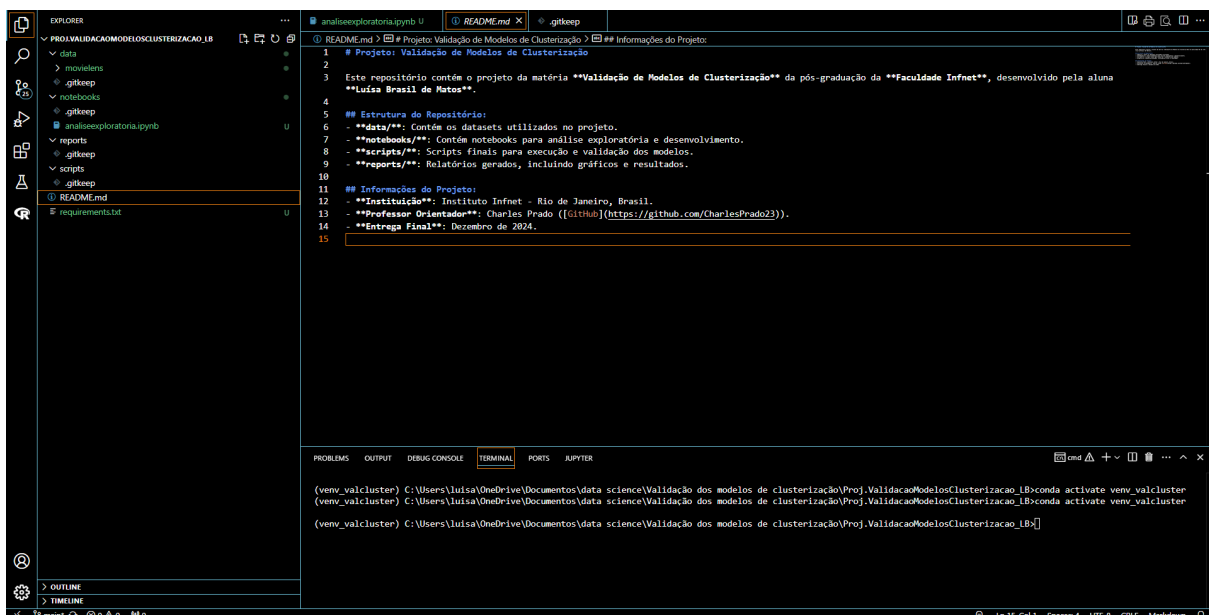


```
Anaconda Prompt (miniconda) x + v
(venv_valcluster) C:\Users\luisa\OneDrive\Documentos\data science\Validação dos modelos de clusterização\Proj.ValidacaoModelosClusterizacao_LB>python --version
Python 3.12.4
(venv_valcluster) C:\Users\luisa\OneDrive\Documentos\data science\Validação dos modelos de clusterização\Proj.ValidacaoModelosClusterizacao_LB>
```

3. Todas as bibliotecas usadas nesse exercícios estão instaladas em um ambiente virtual específico
4. Gere um arquivo de requerimentos (requirements.txt) com os pacotes necessários. É necessário se certificar que a versão do pacote está disponibilizada.

Arquivo Requirements.text disponível no Github.

5. Tire um printscreen do ambiente que será usado rodando em sua máquina.
6. 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).



LINK GITHUB:

https://github.com/luisabrasildematos/Proj.ValidacaoModelosClusterizacao_LB

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.

MovieLens

GroupLens Research has collected and made available rating data sets from the MovieLens web site (<https://movielens.org>). The data sets were collected over various periods of time, depending on the size of the set. Before using these data sets, please review their README files for the usage licenses and other details.

Seeking permission? If you are interested in obtaining permission to use MovieLens datasets, please first read the terms of use that are included in the README file. Then, please [fill out this form](#) to request use. We typically do not permit public redistribution (see [Kaggle](#) for an alternative download location if you are concerned about availability).

recommended for new research

MovieLens 32M

MovieLens 32M movie ratings. Stable benchmark dataset. 32 million ratings and two million tag applications applied to 87,585 movies by 200,948 users. Collected 10/2023 Released 05/2024

- [README.txt](#)
- [ml-32m.zip](#) (size: 239 MB, [checksum](#))

A base de dados escolhida para o projeto de clusterização é a “u.user” que faz parte do dataset MovieLens e contém informações sobre os usuários dos filmes em que a pesquisa foi feita.

user_id: Identificador único do usuário.

age: Idade do usuário.

gender: Gênero do usuário.

occupation: Ocupação do usuário.

zip_code: Código postal do usuário.

2. Escreva a justificativa para a escolha de dados, dando sua motivação e objetivos.

O dataset `u.user` foi escolhido por conter informações relevantes para a segmentação de usuários, como idade, ocupação e gênero. Essas características permitem agrupar os usuários com base em perfis comportamentais ou demográficos, o que pode ser aplicado em:

Recomendações de conteúdo: Identificar preferências baseadas em perfis semelhantes.

Estudo de tendências: Entender padrões de consumo de usuários de diferentes grupos.

Marketing direcionado: Criar campanhas personalizadas para grupos específicos.

Objetivo:

Realizar uma clusterização para identificar perfis de usuários e explorar como essas características demográficas podem impactar preferências ou comportamentos.

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

Visualização: Criar histogramas e gráficos de barras para as variáveis `idade`, `gênero` e `ocupação`.

Análise:

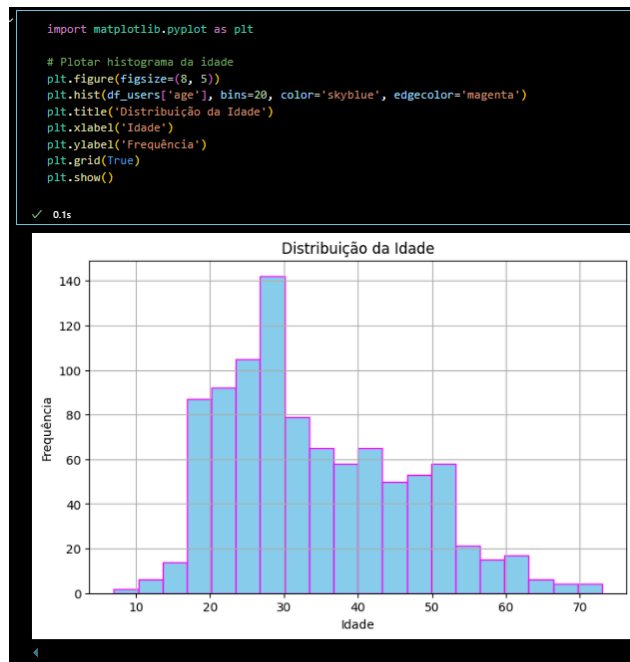
Verificar a distribuição das idades (faixa etária predominante).

Analisar a frequência de cada ocupação.

Conferir o balanceamento dos gêneros.

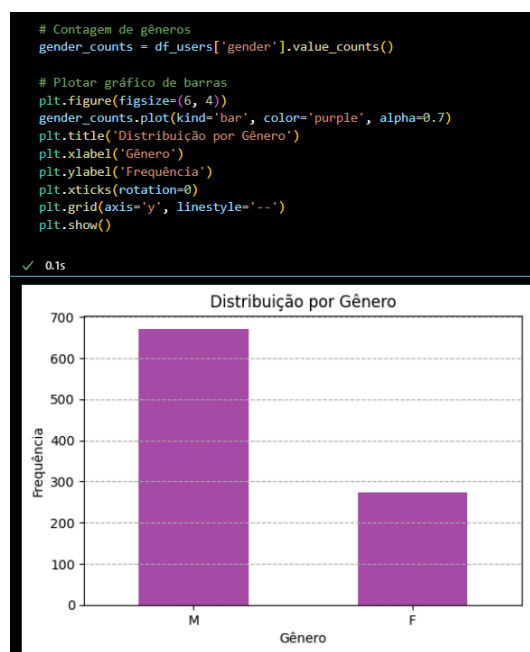
Análise do gráfico de distribuição de idade:

O dataset apresenta indivíduos com idades variando de aproximadamente 5 até 75 anos. A faixa etária predominante está concentrada entre 20 e 30 anos, representando a maior parte da amostra. Essa distribuição pode indicar que o público-alvo principal dos dados corresponde a jovens adultos, possivelmente em fase universitária ou início da vida profissional.



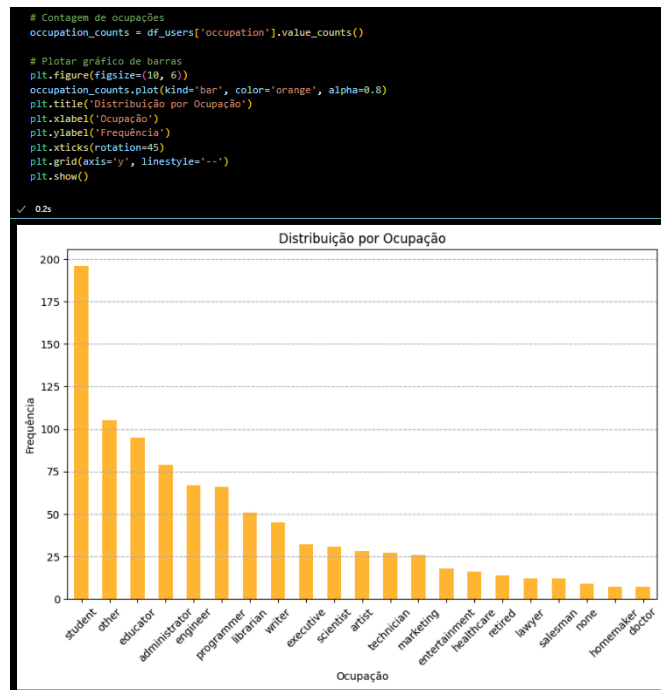
Análise do gráfico de distribuição de gênero:

O dataset é composto por dois gêneros, masculino e feminino, com predominância masculina. Homens representam mais que o dobro da quantidade de mulheres na amostra. Essa disparidade de gênero pode refletir características demográficas e comportamentais específicas da população analisada.



Análise do gráfico de distribuição de gênero:

A distribuição de ocupações revela uma ampla variedade de profissões, sendo os estudantes o maior grupo representado. Essa observação é coerente com a predominância da faixa etária jovem no dataset. Ocupações como "educador", "administrador" e "engenheiro" também possuem representação significativa, enquanto profissões como "médico" e "dona de casa" têm menor frequência.



Além da análise exploratória inicial, que incluiu a plotagem de gráficos de idade, gênero e ocupação, foi realizada a preparação do dataset para as etapas de clusterização. Seguindo os objetivos propostos, foram realizadas as seguintes etapas:

Remoção de Colunas Irrelevantes:

As colunas "user_id" e "zip_code" foram removidas por serem irrelevantes para a análise demográfica e para a criação de clusters significativos. Mantivemos apenas as colunas "age", "gender" e "occupation", que apresentam informações relevantes para a análise.

Verificação de Valores Ausentes:

Foi verificado que o dataset não apresenta valores ausentes em nenhuma das colunas, o que garante a integridade dos dados e evita a necessidade de imputação ou remoção de registros.

Normalização:

A coluna "age" foi normalizada utilizando a técnica de escala Min-Max. Essa técnica transforma os valores da idade para um intervalo entre 0 e 1, preservando as proporções relativas. Isso é fundamental para evitar que variáveis com escalas diferentes dominem a análise de clusterização.

Codificação de Variáveis Categóricas:

As colunas "gender" e "occupation" foram codificadas utilizando variáveis dummy. Esse processo transformou as categorias em valores binários (0 ou 1), permitindo que os algoritmos de clusterização processem os dados corretamente. A opção `drop_first=True` foi utilizada para evitar redundâncias.

—

Essas etapas garantem que os dados estão preparados para a aplicação de algoritmos de clusterização, com variáveis relevantes, escalas uniformes e representações numéricas adequadas para variáveis categóricas. Os gráficos e os resultados das transformações realizadas no dataset estão documentados no notebook do VSCode, permitindo análise e replicação. Essa abordagem garante que os dados estejam prontos para a próxima etapa de criação de clusters.

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:
 - a. K-Médias
 - b. DBScan

K-Médias: O processo de mensuração do índice de silhueta no algoritmo K-Means foi realizado após a definição do número ideal de clusters. Inicialmente, utilizei o método do cotovelo para identificar o ponto em que a diminuição da inércia começa a se estabilizar, indicando um equilíbrio entre a quantidade de clusters e a simplicidade do modelo. Esse ponto foi observado em **k = 4**, que foi definido como o número ideal de clusters para os dados.

Com os clusters definidos, apliquei o algoritmo K-Means aos dados já normalizados e pré-processados. Após a clusterização, utilizei a função `silhouette score` da biblioteca `sklearn` para calcular o índice médio de silhueta. Esse índice é uma métrica que mede a separação dos clusters: valores próximos de 1 indicam uma boa separação, enquanto valores próximos de 0 ou

negativos indicam sobreposição ou má separação. O índice médio de silhueta obtido foi **0.4197**, o que indica uma separação moderada entre os clusters.

Para complementar a análise, gerei um gráfico de silhueta, que permite uma visualização da qualidade dos clusters. O gráfico mostra a distribuição dos coeficientes de silhueta para cada ponto em relação ao cluster ao qual pertence, destacando que alguns clusters apresentaram melhores separações do que outros. Essa análise gráfica foi essencial para identificar potenciais problemas, como clusters que podem estar mal definidos ou sobrepostos.

Portanto, concluí que a escolha de 4 clusters foi adequada dentro do contexto da base de dados, levando em consideração tanto a análise quantitativa quanto visual. Apesar de o índice de silhueta indicar que a separação é apenas moderada, ele é suficiente para prosseguir com a análise, dado o tamanho do dataset e a complexidade das variáveis envolvidas.

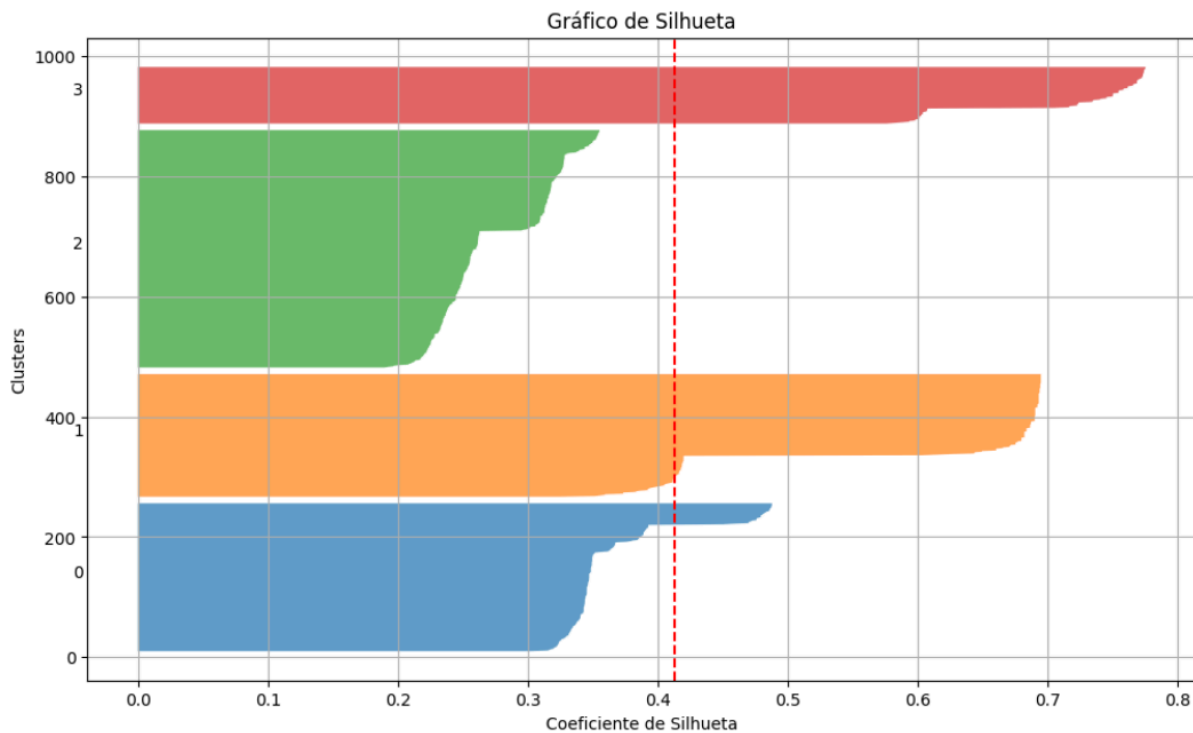
DBScan: O DBScan foi aplicado para realizar a clusterização dos dados utilizando os parâmetros eps (distância máxima entre pontos) e min samples (número mínimo de pontos para formar um cluster). Foram testados diferentes valores de eps entre 0.05 e 0.3, e de min_samples nos valores 3, 5 e 7. Os resultados mostraram que a combinação eps = 0.2 e min_samples = 3 apresentou o melhor desempenho, com um índice de Silhueta de 0.9020, a formação de 35 clusters e apenas 11 ruídos. Valores menores de eps produziram mais clusters e ruídos, enquanto valores maiores reduziram a quantidade de clusters. O parâmetro min_samples influenciou diretamente a formação de clusters ao exigir maior densidade de pontos para agrupamento. Comparado ao K-Means, o DBScan se destacou pela capacidade de lidar com outliers e identificar clusters de formas não lineares, oferecendo uma solução mais robusta para este conjunto de dados.

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

Para o K-Means, o número de clusters foi determinado utilizando o Método do Cotovelo, resultando em 4 clusters. O gráfico do índice de Silhueta revelou um valor médio de 0.4197, o que indica uma qualidade moderada da clusterização, com alguns pontos próximos às fronteiras entre clusters. Apesar de não ser um valor ideal, ele reflete o melhor equilíbrio encontrado entre a quantidade de clusters e a distribuição dos dados.

Já no caso do DBScan, o índice de Silhueta foi calculado para diferentes combinações dos parâmetros eps e min_samples. A combinação mais satisfatória foi eps = 0.2 e min_samples = 3, que resultou em um índice de Silhueta de 0.9020, com 35 clusters e apenas 11 ruídos. Este valor indica uma clusterização de alta qualidade, com os pontos bem agrupados em regiões de maior densidade e uma clara separação entre os clusters.

Em resumo, a avaliação do índice de Silhueta justificou a escolha de 4 clusters no K-Means devido ao equilíbrio observado, enquanto no DBScan, a combinação $\text{eps} = 0.2$ e $\text{min_samples} = 3$ foi selecionada por apresentar um desempenho superior em termos de qualidade de agrupamento. O gráfico do índice de Silhueta foi utilizado como suporte visual para validar as decisões tomadas em ambos os métodos.



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

Os resultados obtidos com o K-Means e o DBScan apresentaram diferenças e semelhanças significativas, tanto na forma como os clusters foram formados quanto na qualidade dos agrupamentos. Ambos os métodos possuem o objetivo de agrupar os dados com base em similaridades entre os pontos e utilizaram o índice de silhueta como métrica de avaliação da qualidade dos clusters. No entanto, suas abordagens e resultados diferiram.

No K-Means, o número de clusters foi definido manualmente a partir do método do cotovelo e do índice de silhueta, resultando em 4 agrupamentos. O índice de silhueta obtido foi de 0.4197, indicando uma qualidade moderada dos clusters. O gráfico de silhueta mostrou que alguns grupos apresentaram boa coesão, enquanto outros tiveram pontos que não se ajustaram bem aos limites dos clusters. O K-Means, por sua natureza, obriga todos os pontos a pertencerem a algum cluster, não identificando outliers ou ruídos nos dados.

Por outro lado, o DBScan formou os clusters de maneira mais flexível, dependendo dos parâmetros ajustados, como o valor de eps e min_samples. Com a combinação de melhores parâmetros (eps=0.3 e min_samples=5), o DBScan identificou 31 clusters, além de separar pontos considerados ruídos. Isso representou uma análise mais robusta, especialmente em relação a dados com diferentes densidades. O índice de silhueta alcançado foi significativamente superior, com um valor de 0.9012, sugerindo que os clusters formados possuem alta coesão e boa separação entre si.

A principal diferença entre os dois métodos está na capacidade do DBScan de identificar pontos de ruído e detectar clusters em dados menos uniformes, enquanto o K-Means apresentou limitações por exigir a definição prévia do número de clusters e por forçar todos os pontos a se encaixarem em algum grupo. Assim, o DBScan apresentou uma visão mais detalhada e precisa dos dados, enquanto o K-Means foi eficiente na sua simplicidade, mas com uma qualidade de agrupamento mais limitada.

4. 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.

```
from sklearn.metrics import davies_bouldin_score, calinski_harabasz_score

# Davies-Bouldin e Calinski-Harabasz para K-Means
dbi_kmeans = davies_bouldin_score(df_users, df_users['Cluster'])
ch_kmeans = calinski_harabasz_score(df_users, df_users['Cluster'])

print(f"Índice Davies-Bouldin (K-Means): {dbi_kmeans:.4f}")
print(f"Índice Calinski-Harabasz (K-Means): {ch_kmeans:.4f}")

# Davies-Bouldin e Calinski-Harabasz para DBScan
# Filtra os ruídos do DBScan (-1)
dbscan_clusters = df_users[df_users['Cluster_DBScan'] != -1]

dbi_dbscan = davies_bouldin_score(dbscan_clusters, dbscan_clusters['Cluster_DBScan'])
ch_dbscan = calinski_harabasz_score(dbscan_clusters, dbscan_clusters['Cluster_DBScan'])

print(f"Índice Davies-Bouldin (DBScan): {dbi_dbscan:.4f}")
print(f"Índice Calinski-Harabasz (DBScan): {ch_dbscan:.4f}")

✓ 0.0s

Índice Davies-Bouldin (K-Means): 6.5722
Índice Calinski-Harabasz (K-Means): 33.9887
Índice Davies-Bouldin (DBScan): 0.1325
Índice Calinski-Harabasz (DBScan): 92271.6260
```

Os resultados mostram que o DBScan apresentou um desempenho superior em ambas as métricas em relação ao K-Means. O índice Davies-Bouldin do DBScan foi muito menor, confirmando clusters mais compactos e bem definidos. Além disso, o Calinski-Harabasz do DBScan foi significativamente maior, destacando a satisfatória separação e coesão dos clusters.

Isso evidencia que o DBScan, com os parâmetros configurados, conseguiu lidar melhor com os dados, especialmente ao tratar ruídos e identificar clusters de diferentes densidades. Já o

K-Means apresentou resultados consistentes, mas com menor eficiência na separação dos grupos e na compactação.

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

O índice de silhueta é amplamente utilizado para avaliar a qualidade dos agrupamentos em algoritmos como o K-Means, onde o número de clusters é pré-definido. No entanto, quando aplicado ao DBScan, a situação é mais complexa devido às características do algoritmo. O DBScan não exige a definição prévia do número de clusters, pois ele descobre os agrupamentos com base nos parâmetros epsilon (eps) e min_samples, que controlam a densidade dos clusters. Como resultado, o DBScan pode gerar um número variável de clusters, incluindo a identificação de ruídos (pontos que não pertencem a nenhum cluster e são atribuídos como -1). Essa flexibilidade faz com que a avaliação por silhueta nem sempre seja a mais adequada. Na análise realizada, o índice de silhueta foi calculado e apresentou valores variados para diferentes combinações de parâmetros eps e min_samples. Apesar de fornecer uma medida de qualidade dos clusters, ele possui limitações em relação ao DBScan, principalmente porque:

O DBScan pode produzir clusters de formas não esféricas, enquanto o índice de silhueta assume uma preferência por formas circulares ou esféricas.

A presença de ruídos identificados como -1 pode distorcer os valores da silhueta, impactando negativamente a interpretação do índice.

O DBScan não busca otimizar a separação ou coesão dos clusters como o K-Means, mas sim agrupar os pontos com base na densidade, o que faz com que outras métricas, como Davies-Bouldin e Calinski-Harabasz, sejam mais adequadas.

Portanto, embora o índice de silhueta forneça uma visão geral da qualidade dos clusters formados pelo DBScan, ele não é o índice mais indicado para determinar o número ideal de clusters nesse algoritmo. Em vez disso, métricas como Davies-Bouldin e Calinski-Harabasz, que avaliam a compactação e separação dos clusters sem depender de formas específicas, tendem a ser mais apropriadas para validar os resultados do DBScan.

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.

Preparação dos Dados:

- Garantir que todas as séries temporais estejam alinhadas no mesmo intervalo de tempo.
- Normalizar os valores das séries temporais, se necessário, para evitar vieses causados por escalas diferentes.

Cálculo da Similaridade:

- Calcular a correlação cruzada entre todas as combinações de pares de séries temporais.
- Para cada par de séries, identificar o valor máximo da correlação cruzada e armazenar esses valores em uma matriz de similaridade.

Definição do Número de Grupos:

- Decidir que as séries temporais serão agrupadas em 3 clusters, conforme solicitado.

Aplicação do Algoritmo de Agrupamento:

- Aplicar um algoritmo de agrupamento, como K-Médias ou DBScan, utilizando a matriz de similaridade como entrada.

Análise e Validação:

- Verificar a adequação dos clusters gerados em relação à similaridade entre as séries temporais.
- Validar os grupos formados utilizando métricas como índice de silhueta, índice Davies-Bouldin, ou outras métricas relevantes.

Interpretação dos Grupos:

- Analisar as características comuns dentro de cada cluster e interpretar o agrupamento com base no problema proposto.

2. Para o problema da questão anterior, indique qual algoritmo de clusterização você usaria. Justifique.

O algoritmo mais indicado seria o DBScan, pois ele é adequado para situações onde os dados podem apresentar ruídos e onde a forma dos clusters não é necessariamente esférica, como pode ser o caso ao agrupar séries temporais. O DBScan é capaz de identificar grupos com base

na densidade de pontos e também tratar séries que não se ajustam bem a nenhum cluster como ruído, garantindo maior robustez nos resultados. A escolha do DBScan é especialmente relevante quando a matriz de similaridade apresenta séries temporais com diferentes níveis de correlação cruzada, pois o algoritmo pode lidar bem com essa variabilidade sem depender de uma forma específica de distribuição dos dados.

3. Indique um caso de uso para essa solução projetada.

Um caso de uso para a solução projetada seria a segmentação de consumo de energia elétrica ao longo do tempo. As séries temporais representariam o consumo de diferentes usuários, e o agrupamento com base na correlação cruzada ajudaria a identificar perfis de consumo similares, como consumidores residenciais, comerciais e industriais. Isso permitiria a empresas de energia criar estratégias personalizadas para cada grupo, como ofertas específicas ou alertas de consumo.

4. Sugira outra estratégia para medir a similaridade entre séries temporais. Descreva em tópicos os passos necessários.

Transformação das Séries Temporais: Realizar a transformação das séries para o domínio da frequência usando a Transformada de Fourier.

Comparar as séries no domínio da frequência para capturar padrões cíclicos e sazonais.

Cálculo de Similaridade: Calcular a distância de DTW (Dynamic Time Warping) entre as séries temporais no domínio da frequência ou no domínio original. A DTW é uma técnica que mede a similaridade considerando deslocamentos temporais entre pontos das séries.

Construção da Matriz de Distância: Criar uma matriz de distâncias baseada nos valores de DTW para todas as combinações de pares de séries.

Agrupamento: Aplicar algoritmos como Hierarchical Clustering ou DBScan utilizando a matriz de distância como entrada.

Validação e Interpretação: Validar os resultados utilizando métricas como índice de silhueta ou visualizações de dendrogramas.

Interpretar os clusters com base nas características comuns entre as séries temporais agrupadas.