Cristian César Martins – RA: 799714
Rodrigo Pavão Coffani Nunes – RA: 800345
Vinícius de Oliveira Guimarães – RA: 802431
Vitor Gabriel Orsin – RA: 801575

Trabalho Prático 2
Aprendizado de Máquina 1

São Carlos/SP, Brasil

27 de Junho, 2023

Cristian César Martins – RA: 799714
Rodrigo Pavão Coffani Nunes – RA: 800345
Vinícius de Oliveira Guimarães – RA: 802431
Vitor Gabriel Orsin – RA: 801575

Trabalho Prático 2
Aprendizado de Máquina 1

Relatório científico de pesquisa do segundo trabalho da disciplina de Aprendizado de Máquina 1, com o dataset sobre o tema "Qualidade da Água"

Universidade Federal de São Carlos – UFSCar

Departamento de Computação

Bacharelado em Ciência da Computação

São Carlos/SP, Brasil

27 de Junho, 2023

LISTA DE ILUSTRAÇÕES

LISTA DE TABELAS

SUMÁRIO

1 INTRODUÇÃO

Existem diversas maneiras de analisar grandes conjuntos de dados com a ajuda do aprendizado de máquina, e uma delas é o Aprendizado Não Supervisionado. Esse método foca em não prover rótulos (*labels*) para o algoritmo, assim, fazendo-o aprender de maneira "independente".

De certa forma, esse processo "acelera"o processo de análise uma vez que não é necessário rotular dados como no Aprendizado Supervisionado. No entanto, é necessário mais cuidado e um estudo bem estruturado sobre os dados e algoritmos a serem utilizados para que o resultado seja coerente e apresente novas informações interessantes sobre padrões presentes no *dataset*.

1.1 Enunciado do trabalho

Para a realização deste trabalho, é necessário escolher um conjunto de dados (*dataset*) que seja: relevante, desafiador e complexo. Após a escolha, deve-se explorá-lo e então solucionar eventuais problemas de atributos, criar visualizações, assim como aplicar seleção e redução destes atributos. É preciso deixar o conjunto de dados em condições ideias para aplicação dos métodos, sendo necessário uma explicação e justificativa de cada técnica aplicada.

Com o conjunto de dados devidamento preparado, devem-se ser aplicados diferentes métodos Aplicação de diferentes métodos de para realizar a tarefa não-supervisionada, justificando a escolha, o ajuste de parâmetros, e comparando de forma adequada os resultados. Ao final, apresentar gráficos que ilustrem os resultados encontrados.

1.2 *Dataset* escolhido

O conjunto de dados escolhidos para ser explorado foi encontrado no Kaggle e seu tema trata da emissão de CO2 por veículos automotores, exclusivamente de carros. [1]. É composto por dados sobre os veículos, assim como detalhes sobre seus motores e consumos.

O *dataset* possui 12 colunas. Com 7385 linhas, foi considerado um bom conjunto a ser escolhido pela quantidade satisfatória de amostras e boas notas na plataforma.

[1] Obtido em <https://www.kaggle.com/datasets/bhuviranga/co2-emissions>.

1.2.1 Colunas

As colunas presentes no *dataset* e suas descrições fornecidas estão presentes na Tabela 1. São informações como: fabricante, modelo, tipo de combustível, taxa de emissão de dióxido de carbono (CO_2), entre outras mais.

Tabela 1 – Colunas presentes no *dataset*

coluna	**descrição**
Make	fabricante
Model	modelo do veículo
Vehicle Class	tipo de veículo
Engine Size (L)	tamanho do motor, medido em litros
Cylinders	quantidade de cilindros no motor
Transmission	tipo de transmissão (manual, automática...)
Fuel Type	tipo de combustível utilizado
Fuel Consumption City (L/100 km)	consumo médio na cidade (litros em 100 Km)
Fuel Consumption Hwy (L/100 km)	consumo médio na estrada (litros em 100 Km)
Fuel Consumption Comb (L/100 km)	consumo médio geral (litros em 100 Km)
Fuel Consumption Comb (mpg)	consumo médio geral (*miles per gallon*)
CO_2 Emissions(g/km)	emissão de CO_2 (gramas por Km)

Fonte: os autores

Duas colunas que precisam ser mais bem explicadas, são as de *Transmission* e *Fuel Type*. O valores dessas colunas são siglas, com significados não tão claros.

A coluna *Transmission*, possui letras que indicam qual é o estilo de transmissão e quantas marchas ela possui. Os valores podem ser AX, AMX, ASX, AVX, e MX, com o X sendo um valor numérico presente na sigla. As descrições são as seguintes:

Tabela 2 – Tipo de Transmissões

Transmission	**Transmissão**
A	Automático
AM	Automático Manual
AS	Automático com *Selection Shift*
AV	Automático contínuo
M	Manual
3 - 10	Quantidade de marchas

Fonte: os autores

A coluna *Fuel Type*, por sua vez, também possui letras que indicam qual é o tipo de combustível utilizado pelo veículo, variando desde Gasolina Comum até mesmo Gás Natural (também conhecido como GNV - Gás Natural Veicular). Os valores e descrições são os seguintes:

Tabela 3 – Tipos de combustíveis

Fuel Type	**Tipo de combustível**
X	Gasolina
Z	Gasolina Premium
D	Diesel
E	Etanol
N	Gás Natural (GNV)

Fonte: os autores

1.3 Problemas a serem enfrentados

Os problemas a serem enfrentados neste trabalho são os mesmos que ocorrem com grandes conjuntos de dados: entender corretamente os atributos, suas correlações e possíveis informações a serem extraídas, assim como selecionar quais serão mais relevantes para a análise a ser realizada.

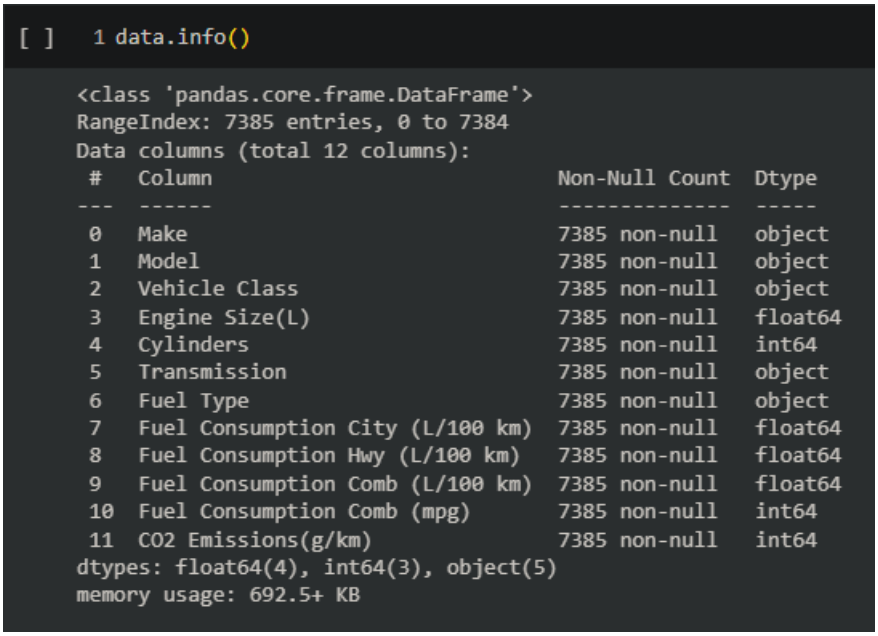
A partir de análises presentes em etapas seguintes deste relatório, será possível determinar se existem amostras com dados problemáticos, quais atributos posssuem grande influência nas informações extraídas ou quais podem ser considerados redundantes. Assim como, após a redução de atributos e aplicação de filtros, quais são os métodos mais adequados para analisar os dados uma vez já pré-processados.

2 DISCUSSÃO E RESULTADOS

2.1 Abordagens iniciais

Após carregar os dados, utilizando a biblioteca pandas[1], é importante realizar uma análise inicial superficial desses dados, entendendo melhor o que será tratado ao decorrer dessa pesquisa.

Figura 1 – *Output* de `data.info()`



```
[ ]  1 data.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 7385 entries, 0 to 7384
Data columns (total 12 columns):
 #   Column                          Non-Null Count  Dtype
---  ------                          --------------  -----
 0   Make                            7385 non-null   object
 1   Model                           7385 non-null   object
 2   Vehicle Class                   7385 non-null   object
 3   Engine Size(L)                  7385 non-null   float64
 4   Cylinders                       7385 non-null   int64
 5   Transmission                    7385 non-null   object
 6   Fuel Type                       7385 non-null   object
 7   Fuel Consumption City (L/100 km)  7385 non-null   float64
 8   Fuel Consumption Hwy (L/100 km)   7385 non-null   float64
 9   Fuel Consumption Comb (L/100 km)  7385 non-null   float64
 10  Fuel Consumption Comb (mpg)     7385 non-null   int64
 11  CO2 Emissions(g/km)             7385 non-null   int64
dtypes: float64(4), int64(3), object(5)
memory usage: 692.5+ KB
```

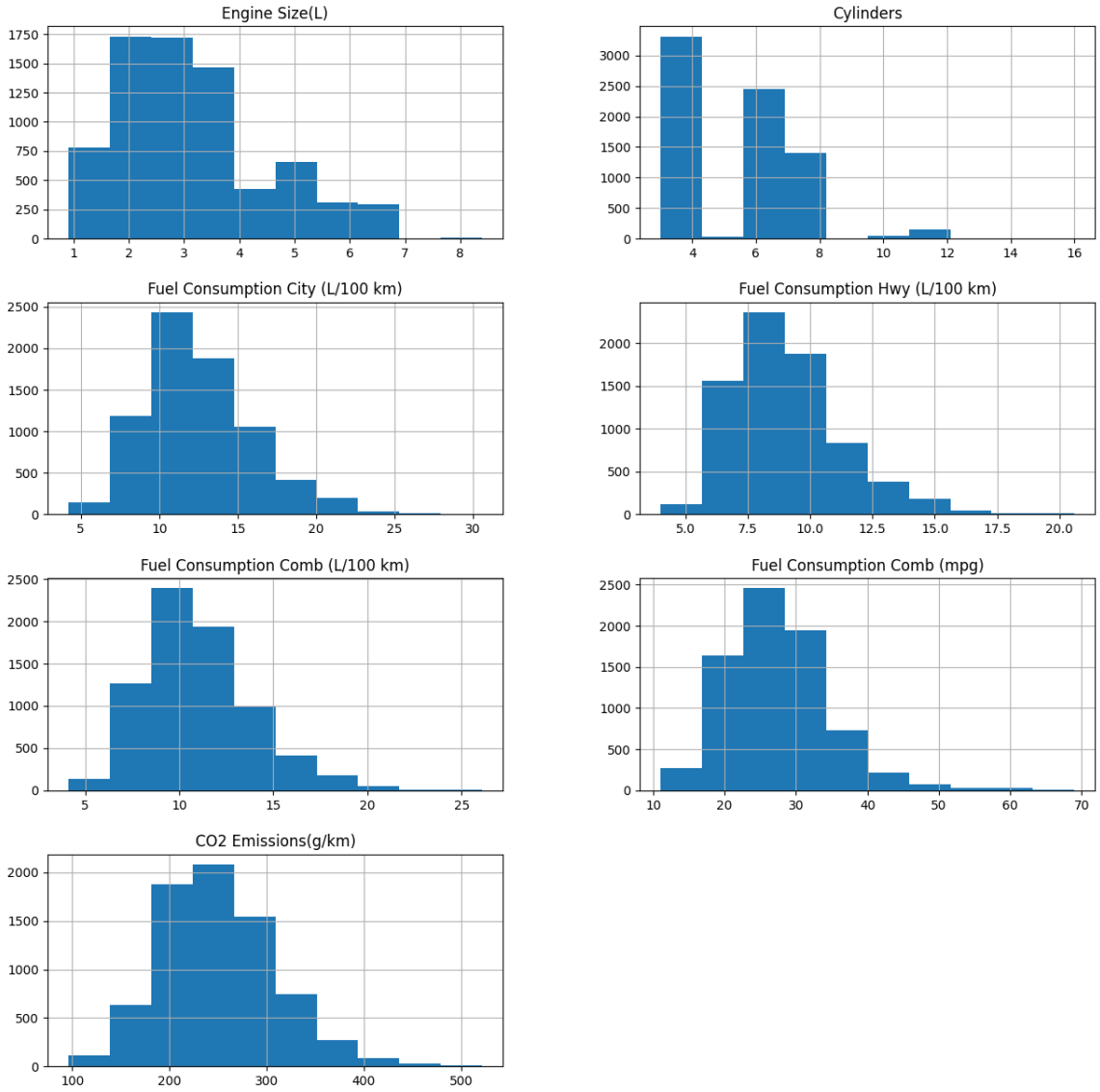
Fonte: os autores

2.2 Análise de atributos

2.2.1 Histogramas

[1] Disponível em: <https://pandas.pydata.org/>

Figura 2 – Histogramas de todas as colunas



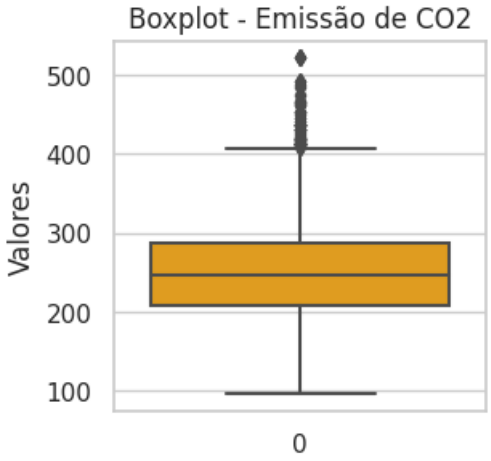
Fonte: os autores

2.2.2 Boxplots

A utilização do gráfico de caixas (*boxplot*) tem como objetivo facilitar a visualização de amostras. Entendendo melhor quais serão os valores que limitam os dados, quais são as regiões onde se encontram a maioria das amostras e a dispersão dos seus dados no geral.

2.2.2.1 Emissão de CO2

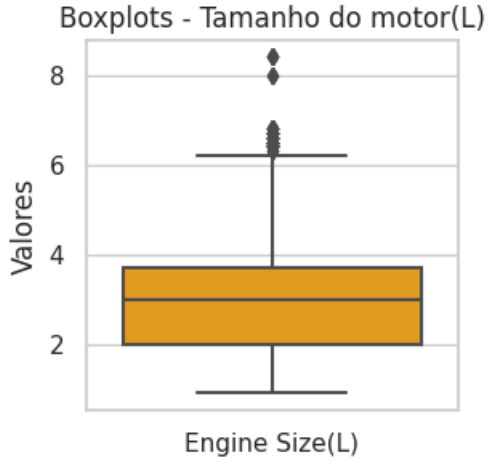
Figura 3 – Boxplot sobre Emissão de Gás Carbônico



Fonte: os autores

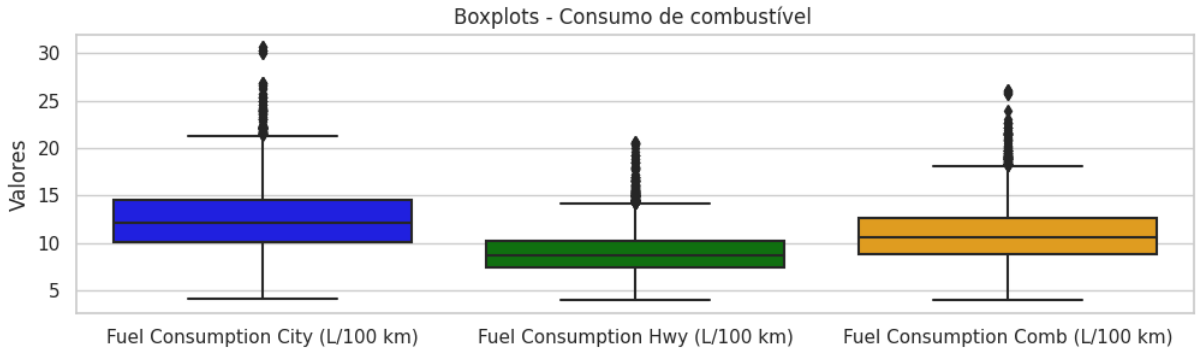
2.2.2.2 Volume do motor

Figura 4 – Boxplot sobre Tamanho dos Motores



Fonte: os autores

Figura 5 – Boxplot sobre Consumo de Combustível



Fonte: os autores

2.2.2.3 Consumo de combustível

Analisando os boxplots, é possível observar que existem diversos amostras que podem ser consideradas *outliers*. No entanto, são mantidos uma vez que queremos analisar as informações de todos os carros existentes dentro do dataset e os mesmos não serem erros de medição, mas sim tipos específicos de carros.

2.3 Pré-processamento

Dados problemáticos podem atrapalhar a análise de *datasets*, uma vez que valores nulos, em formatos diferentes do planejado, amostras duplicadas, entre diversos outros problemas, conseguem enviesar os resultados. Dessa forma, é um importante passo limpar o conjunto de dados, possibilitando uma análise futura mais precisa dessas informações.

2.3.1 Valores nulos

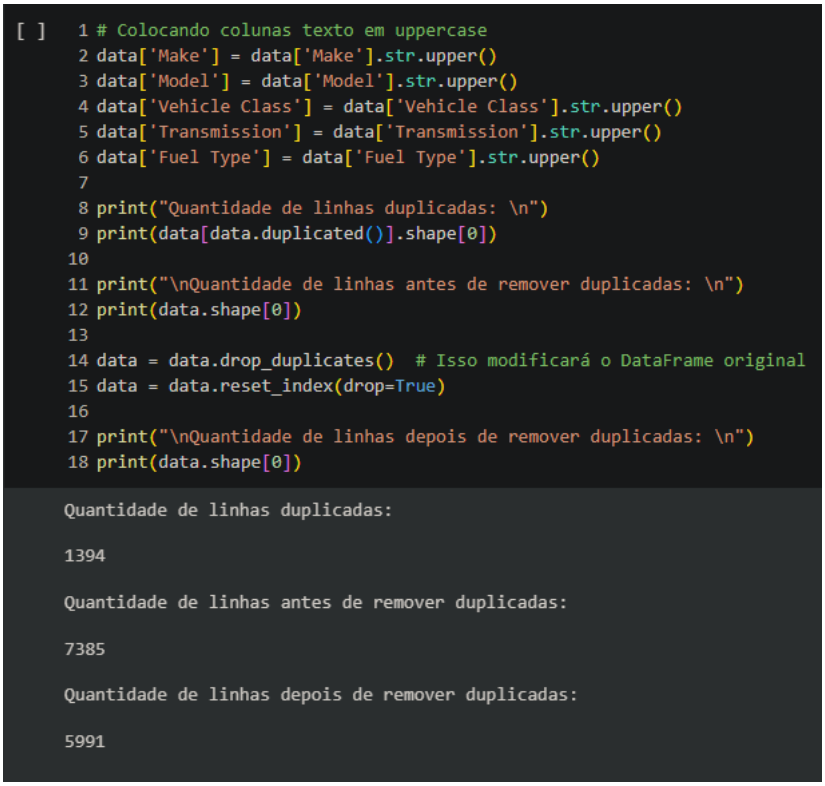
Dando sequência à análise dos dados, o próximo passo é buscar por estes dados no *dataset* e, se encontrados e tratá-los. E, como apresentado na Figura 1, vemos que todas as colunas possuem dados 7385 linhas, com valores não nulos, variando entre valores numéricos (inteiros ou com pontos flutuantes) e objects. Portanto, é possível afirmar que dados nulos não serão um problema neste conjunto.

2.3.2 Duplicatas

Contudo, com base na Figura 6, podemos afirmar que do total de amostras existentes, 1394 delas foram consideradas como sendo dados duplicados.

Apontada a existência de duplicatas nos dados, removemos essas amostras e então atualizamos o conjunto de dados, resultando em 5991 linhas restantes no *dataset*.

Figura 6 – Verificação e remoção de dados duplicados



```
[ ]  1 # Colocando colunas texto em uppercase
     2 data['Make'] = data['Make'].str.upper()
     3 data['Model'] = data['Model'].str.upper()
     4 data['Vehicle Class'] = data['Vehicle Class'].str.upper()
     5 data['Transmission'] = data['Transmission'].str.upper()
     6 data['Fuel Type'] = data['Fuel Type'].str.upper()
     7
     8 print("Quantidade de linhas duplicadas: \n")
     9 print(data[data.duplicated()].shape[0])
    10
    11 print("\nQuantidade de linhas antes de remover duplicadas: \n")
    12 print(data.shape[0])
    13
    14 data = data.drop_duplicates()  # Isso modificará o DataFrame original
    15 data = data.reset_index(drop=True)
    16
    17 print("\nQuantidade de linhas depois de remover duplicadas: \n")
    18 print(data.shape[0])

    Quantidade de linhas duplicadas:

    1394

    Quantidade de linhas antes de remover duplicadas:

    7385

    Quantidade de linhas depois de remover duplicadas:

    5991
```

Fonte: os autores

2.3.3 Seleção de atributos

É possível assumir que a coluna "*Fuel consumption Comb (mpg)*"é redundante para a base de dados, visto que se trata da conversão métrica de outro campo (*Fuel consumption Comb* (*L/100Km*). Dessa forma, podemos removê-la sem maiores problemas.

Código para remoção da coluna *Fuel Consumption Comb* (*mpg*):

```
# Removendo atributo Fuel Consumption Comb (mpg)
data = data.drop("Fuel Consumption Comb (mpg)", axis=1)
display(data)
```

2.3.4 Embaralhamento da amostra

Como visto durante a disciplina, caso os dados do conjunto estejam ordenados é possível que o algoritmo de aprendizado identifique padrões que não existem e que, portanto, não deveriam fazer parte desse aprendizado. Dessa maneira, como modo de evitar um viés indesejado durante a análise dos dados, todas as amostras são embaralhadas.

Isso evita qualquer ordenação prévia dos dados, preservando a possibilidade de um aprendizado melhor definido e mais confiável.

Instruções de embaralhamento:

```
# Embaralhando as linhas do dataframe
data = data.sample(frac=1).reset_index(drop=True)
```

2.3.5 Separação de colunas

Para possibilitar uma verificação da qualidade da análise dos dados, a coluna com os dados de emissão de dióxido de carbono (CO_2) foi copiada um conjunto separado de dados. Essa análise não será obrigatoriamente realizada neste estudo, mas esta separação a torna possível caso seja identificada sua necessidade.

```
# Separando os dados entre atributos comuns e o atributo de emissão de CO2
data_co2_emission = data[["CO2 Emissions(g/km)"]]

display(data_co2_emission)
```

2.3.6 Normalização

A normalização de dados coloca os dados dentro de um intervalo controlado, diminuindo a diferença de valores entre dados muito discrepantes. Esse valores normalmente ficam próximos do intervalo de -1 e 1.

Para alguns algoritmos, é essencial a realização da normalização. Estes são aquele onde é extremamente necessário que os dados estejam na mesma escala, como K-Means, Regressão Linear, entre outros. Sem a normalização, dados com escalas maiores acabam possuindo maior importância no cálculo da distância entre os pontos, levando a um agrupamento incorreto.

2.3.7 Transformação de atributos

Nesta seção, as colunas com texto (*strings*) são convertidas para um tipo de variável mais fácil de trabalhar, que possibilitam o uso dos algoritmos de aprendizado.

2.4 Análise não supervisionada

2.4.1 K-means

Para utilizarmos o K-means, foi necessário utilizar o algoritmo do cotovelo para observar a minimização do quadrado dos erros (J) nos clusters para encontrar a quantidade ideal de cluster para o nosso dataset, mostrado na Figura 9.

Figura 7 – Normalização das colunas com formato

```
[ ]  1 # Seleciona as colunas que são necessárias aplicar normalização
     2 cols_normalizacao = []
     3 for coluna in data.columns:
     4   if data[coluna].dtype == 'float64':
     5     cols_normalizacao.append(coluna)
     6
     7 # Normalização
     8 scaler = StandardScaler()
     9 df_zscore = pd.DataFrame(scaler.fit_transform(data[cols_normalizacao]), columns=cols_normalizacao)
    10
    11 display(df_zscore)
```

	Engine Size(L)	Fuel Consumption City (L/100 km)	Fuel Consumption Hwy (L/100 km)	Fuel Consumption Comb (L/100 km)
0	0.898451	0.940810	0.832342	0.930097
1	-0.492842	-0.154486	0.047393	-0.084268
2	-0.126712	-0.491500	-0.606730	-0.523826
3	1.118128	0.210613	-0.083431	0.118605
4	0.166191	0.154444	-0.170648	0.017169
...	...	...	...	...
5986	-0.858971	-0.856599	-0.868380	-0.861948
5987	-0.858971	-1.053191	-0.868380	-0.997197
5988	-0.346390	0.126359	0.483476	0.253854
5989	-0.566068	-0.407247	-0.039823	-0.287141
5990	2.582647	2.260783	1.660898	2.079711

5991 rows × 4 columns

Fonte: os autores

Figura 8 – Transformação das colunas textuais

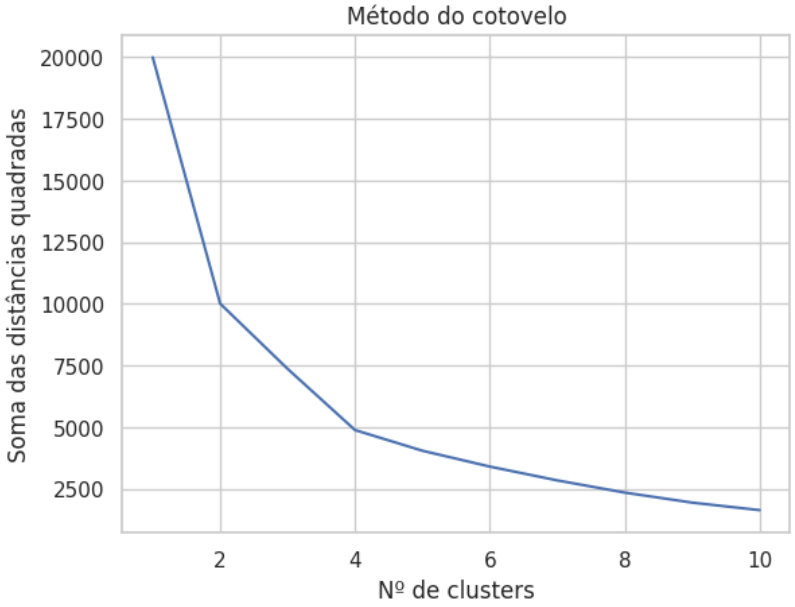
```
[ ]  1 def label_encode(df):
     2     le = LabelEncoder()
     3     for col in df.columns:
     4         if df[col].dtypes=='object':
     5             df[col]=le.fit_transform(df[col])
     6         else:
     7             pass
     8
     9     return df
```

Fonte: os autores

2.4.2 K-means. Gráfico dos clusters formados

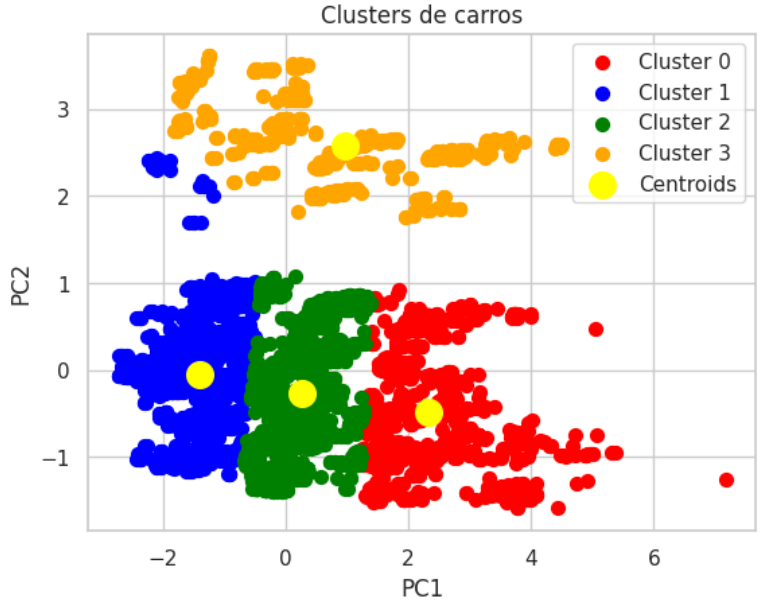
Com isso, ao minimizar o valor de J, ficamos com um total de 4 clusters para separar todos os nossos pontos, onde um dos clusters ficou distante dos outros 3 clusters, enquanto que os outros 3 clusters ficaram bem próximos, como mostrado em Figura 10:

Figura 9 – Minimização do quadrado dos erros



Fonte: os autores

Figura 10 – Clustersgerados pelo K-means



Fonte: os autores

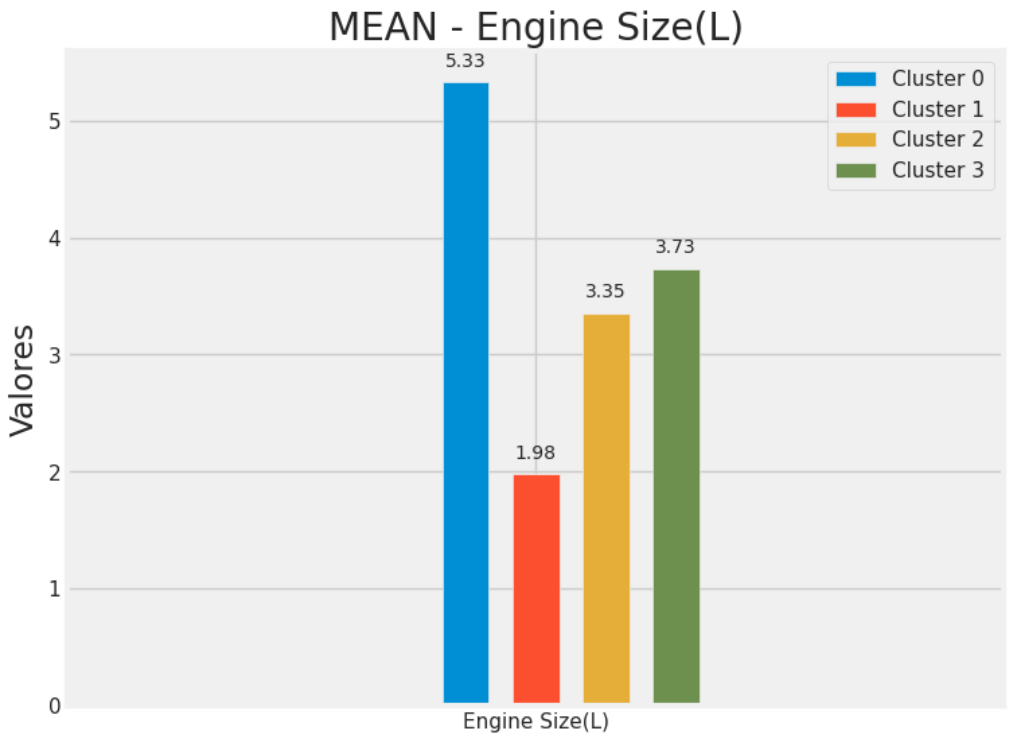
2.4.3 K-means. Analisando propriedades de cada cluster

Observando as imagens dos atributos de cadaum dos clusters formados, temos qeu os 4 clusters possuem algumas características bem evidentes, de forma que se torna algo importante para ser observado.

Na Figura 10 por exemplo, é possível visualizar que o cluster 0 possui carros com

o manior tamanho dos motores, enquanto que o cluster 1 possui carros com o menor tamanho dos motores. Os outros 2 clusters ficam com valores medianos.

Figura 11 – Tamanho do motor para cada cluster



Fonte: os autores

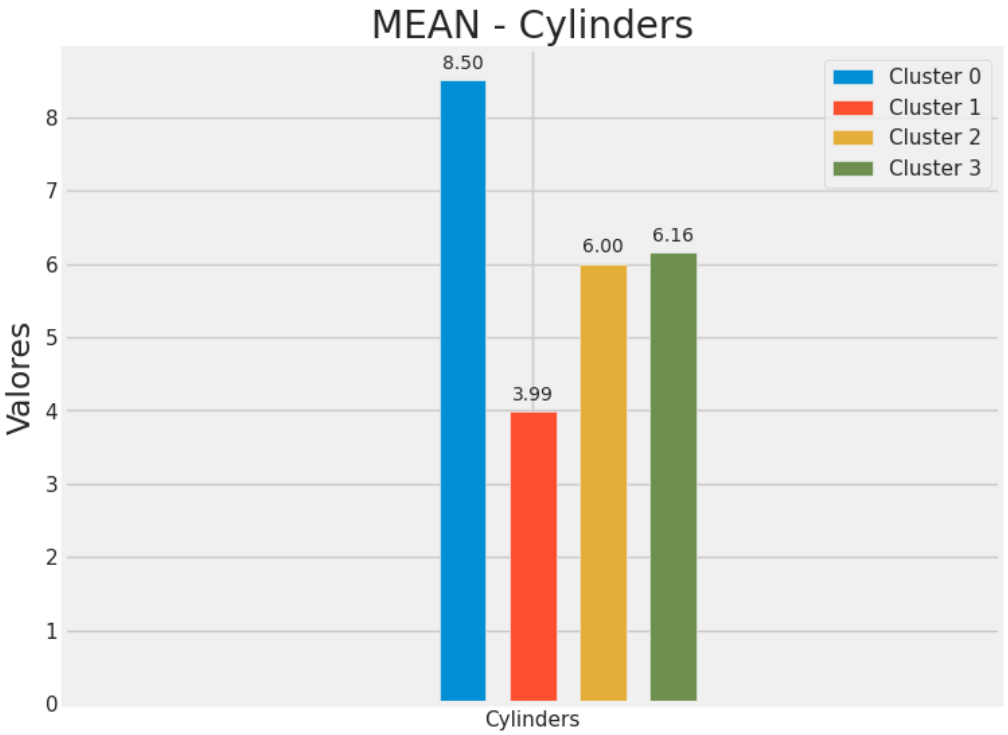
Da mesma forma, na Figura 12 também acontece a mesma coisa que o tamanho do motor, onde geralmente o tamanho do motor segue uma relação direta com a quantidade de cilindros. O cluster 1 possui o maior valor, enquanto que o cluster 1 possui a menor quantidade de cilindros.

Somente observando esses gráficos, podemos começar a perceber que o cluster 1 possui carros bem mais fracos cmoparativamente do que os outros clusters.

Agora, olhando a parte de consumo de combustível na Figura 13 temos que o cluster 3 se destaca contendo os carros que mais consomem combustível dentre os 4 clusters. Além disso, o cluster 1 se destava como sendo o cluster que menos consome combustível dentre os 4 cllusters existentes.

Observando a informação de emissão de CO2, temos que o cluster 0 fica muito distante dos outros clusters quando visto na posição desse atributo CO2 Emission, possuindo um valor muito maior do que o resto. O cluster 1 por sua vez, possui uma emissão de CO2 muito moenor do que os outros clusters existentes.

Figura 12 – Quantidade de cilindros do motor



Fonte: os autores

2.4.4 K-means. Resumo dos 4 clusters

Além dos gráficos anteriores, podemos comparar os clusters segundo atributos categóricos, resultando no seguinte resumo para cada um dos 4 clusters:

1. Cluster 0
    - Tamanho do motor: 5.33
    - Cilindros: 8.5
    - Consumo de combustível (L/100km): 14.41
    - Emissão de CO2 (g/km): 335.55
    - Tipo de combustível: Z (68.2%), X (31.8%)
    - Tipo de marchas: AS8 (21.7%), A6 (15.7%), A8 (14.2%)
    - Fabricantes: Chevrolet (13.6%), Mercedes-Benz (13.1%), GMC (10.4%)
    - Modelo de carro
        - $SUV Standard (21.7\%)$
        - $Pickup Truck (17.2\%)$
        - $Two Seater (14\%)$

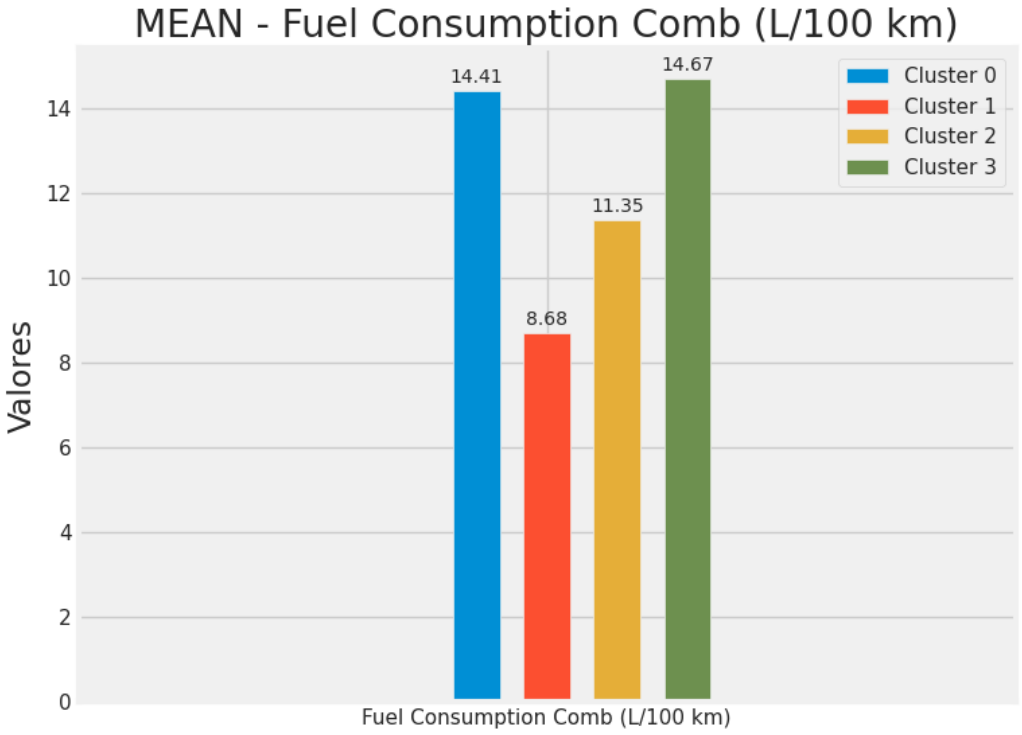
2. Cluster 1

    - Tamanho do motor: 1.98
    - Cilindros: 3.99
    - Consumo de combustível (L/100km): 8.68
    - Emissão de CO2 (g/km): 201.98
    - Tipo de combustível: X (65.6%), Z (33.6%)
    - Tipo de marchas: AS6 (24.4%), M6 (19.1%), AS8 (12.9%)
    - Fabricantes: Ford (8.2%), Mini (7.4%), Chevrolet (6.8%)
    - Modelo de carro
        - $SUVSmall$(23.6%)
        - $Compact$(23.5%)
        - $MidSize$(20.4%)

3. Cluster 2

    - Tamanho do motor: 3.35
    - Cilindros: 6

Figura 13 – Consumo de combustível



Fonte: os autores

- Consumo de combustível (L/100km): 11.35
- Emissão de CO2 (g/km): 264.36
- Tipo de combustível: Z (54.3%), X (45.6%)
- Tipo de marchas: AS8 (21%), AS6 (17.6%), AM7 (8.8%)
- Fabricantes: BMW (12.8%), Porsche (11.7%), Ford (8.9%)
- Modelo de carro
  - $SUVSmall$(16.4%)
  - $Mid-Size$(13.7%)
  - $SUVStandard$(11%)

4. Cluster 3

   - Tamanho do motor: 3.73
   - Cilindros: 6.16
   - Consumo de combustível (L/100km): 14.67
   - Emissão de CO2 (g/km): 267.67
   - Tipo de combustível: E (70%), D (30%)
   - Tipo de marchas: A6 (40.2%), AS6 (16.2%), AS8 (11.8%)
   - Fabricantes: Ford (24%), Chevrolet (20.1%), GMC (17.8%)
   - Modelo de carro
     - $Pickup-TruckStandard$(29.8%)
     - $SUVStandard$(21.8%)
     - $SUVSmall$(10.7%)

   Assim, após analisar todas as informações de todos os 4 clusters, podemos dizer então que:

   - Cluster 0 (CARROS MAIS POLUIDORES): é composto de carros mais rápidos, que consome uma grande quantidade de combustível, que poluem mais e em sua parte contendo carros esportivos.
   - Cluster 1 (CARROS ECONÔMICOS): é composto de carros mais simples e populares, com motores mais fracos, com baixo consumo de combustível e com isso, baixa emissão de CO2.
   - Cluster 2 (CARROS INTERMEDIÁRIOS - SUV's): é composto de carros intermediários entre esportivo e compactos, contendo uma grande quantidade de SUV's movidos a gasolina.
   - Cluster 3 (CARROS CONSUMIDORES): é composto por carros que consomem muito combustível, devido ao fato de que esses carros utilizam etanol e diesel.

2.4.5 Agrupamento hierárquico

Para o método hierárquico, não foi feito mais nenhuma alteração ao dataset além do pré-processamento. Ao executar, o algoritmo definiu 4 clusters, como visto a seguir: Figura 15

Note que os resultados obtidos tem semelhanças com o que foi obtido anteriormente por k-means. O cluster 0 tem um conjunto de dados mais bem separável dos outros clusters, enquanto os clusters 1 e 2 definem a parte com maior concentração de dados. Note também que devido às peculiaridades do algoritmo, foi criado um cluster com um único representante, que é um outlier isolado do gráfico.

Além disso, podemos exibir o dendograma da exececução Figura 16. Fazendo uma análise do gráfico, podemos notar novamente a grande concentração de dados semelhantes, que correspondem principalmente a carros populares, se agrupam (em amarelo). Também podemos ver o galho em azul se aglomerando ao resto em uma altura elevada, mostrando como outliers podem ser notados nessa representação.

Para análise métrica dos cluster, foi aplicado os score de silhueta e Davies-Bouldin. Silhueta: 0,533 Davies-Bouldin: 0,678 Com essas métricas, podemos dizer que os clusters estão bem concentrados e bem agrupados, mas sofrem com a proximidade entre clusters distintos, que podem causar problemas.

2.4.6 Agrupamento densidade

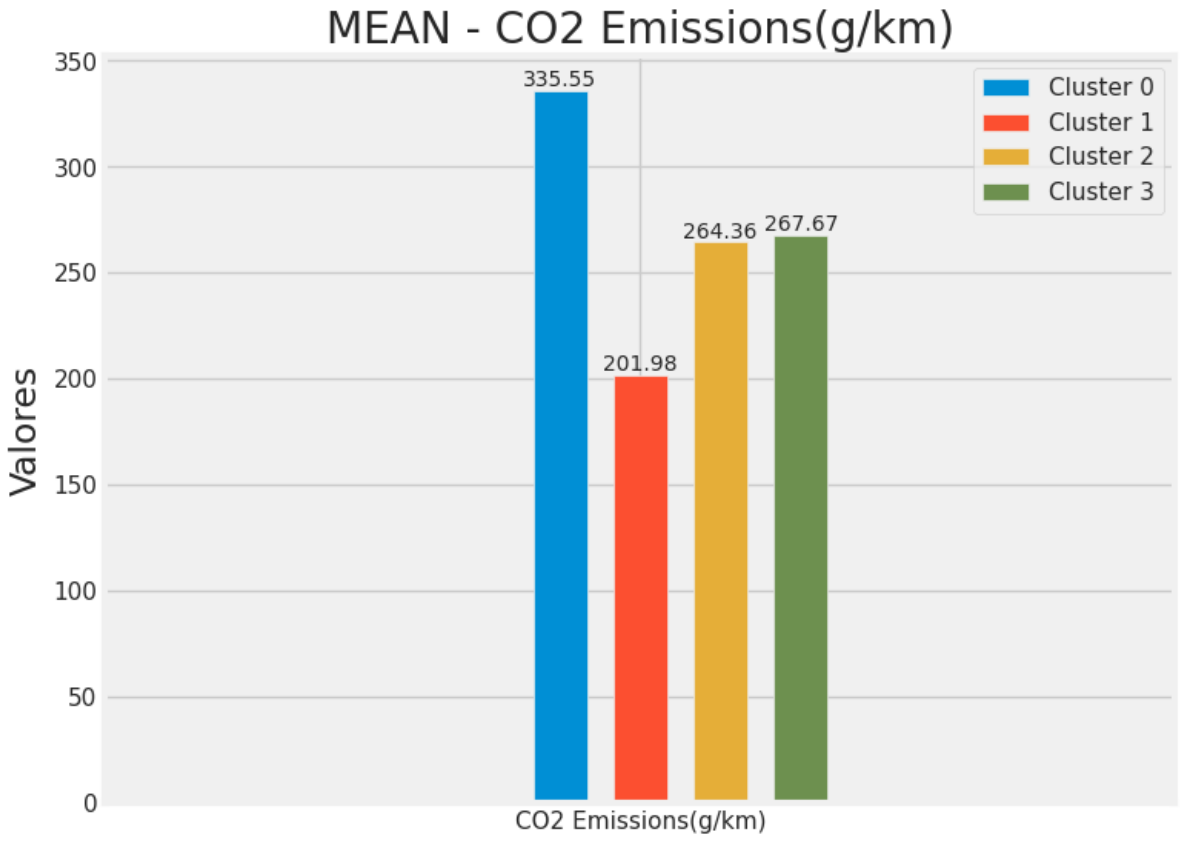
Para utilizar o método de densidade, utilizamos o algoritmo de DBSCAN. Após execução, os clusters foram definidos como a seguir: Figura 17.

Note que os resultados obtidos tem semelhanças com o que foi obtido anteriormente por k-means. O cluster 0 tem um conjunto de dados mais bem separável dos outros clusters, enquanto os clusters 1 e 2 definem a parte com maior concentração de dados. Note também que devido às peculiaridades do algoritmo, foi criado um cluster com um único representante, que é um outlier isolado do gráfico.

Para análise métrica dos cluster, foi aplicado os score de silhueta e Davies-Bouldin. Silhueta: 0,526 Davies-Bouldin: 0,664.

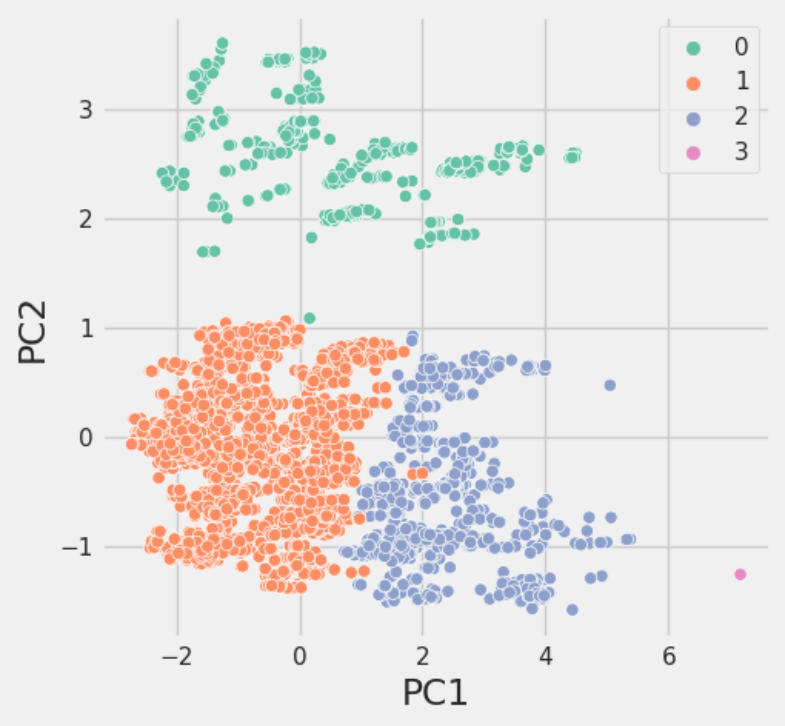
Com essas métricas, podemos dizer que os clusters estão bem concentrados e bem agrupados, mas sofrem com a proximidade entre clusters distintos, que podem causar problemas.

Figura 14 – Emissão de C02



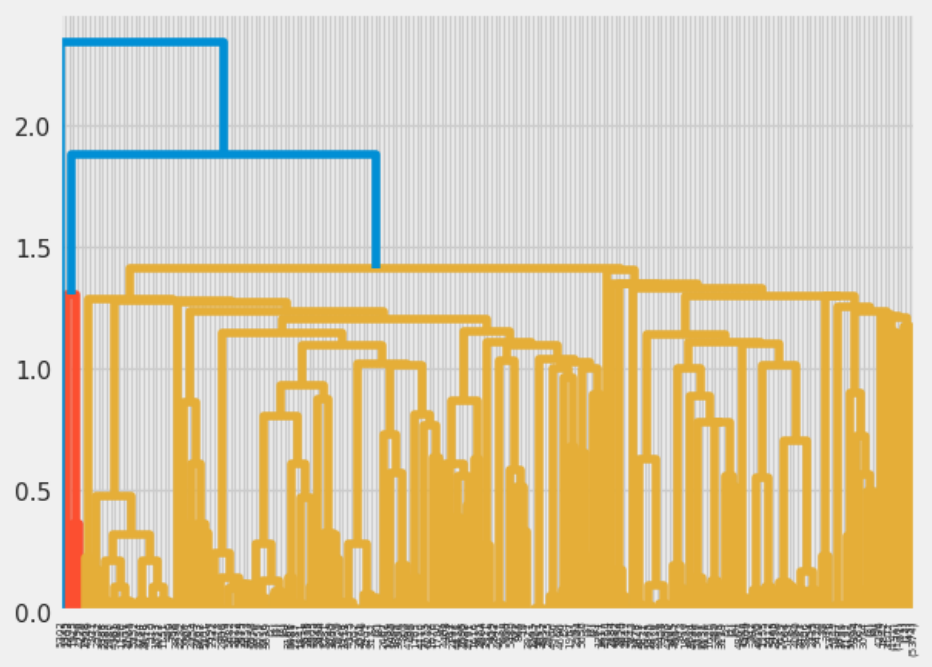
Fonte: os autores

Figura 15 – Gráfico - Hierarquico



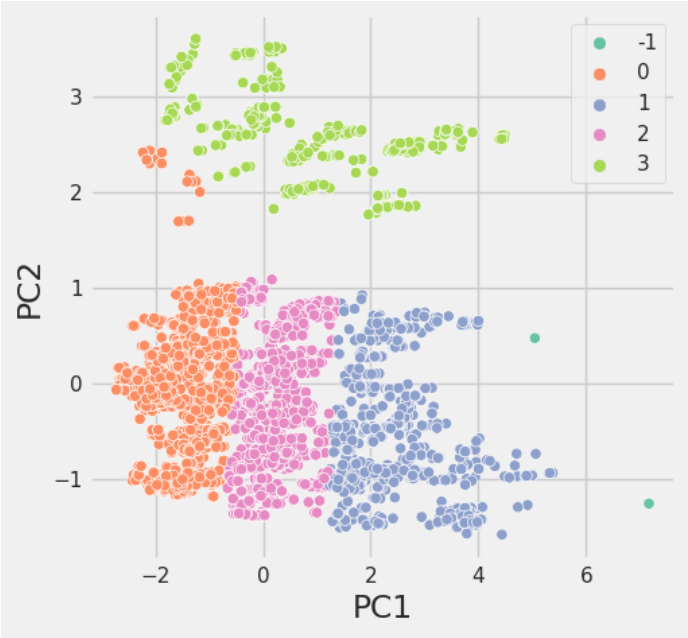
Fonte: os autores

Figura 16 – Dendograma



Fonte: os autores

Figura 17 – Consumo de combustível



Fonte: os autores

3 CONCLUSÃO

Após testar e comparar todos os métodos de aprendizado não supervisionado proposta, podemos concluir que os aglomeramentos são relativamente eficientes para a tarefa proposta. Foi possível definir grupos que definem carros principalmente pela emissão de CO2, mas também os outros fatores, mostrando quais são mais ou menos danosos à natureza, ou ainda quais são mais econômicos.