



MAPA – Material de Avaliação Prática da Aprendizagem

Acadêmico: Lucas Antonio Rocha Santos	R.A. 22134899-5
Curso: Ciências de Dados e Análise de Comportamento	
Disciplina: Reconhecimento de padrão	
Valor da atividade: 3,00	Prazo: 01/05/2023

Faça uma breve discussão sobre qual é o tipo de técnica que seria possível aplicar nessa base de dados para o reconhecimento de padrões dos Produtos.

Uma técnica que pode ser usada é o K-means, o K-means é uma técnica de clusterização que pode ser útil para segmentar produtos de acordo com suas características.

Aplique a técnica, descrita por você no item anterior, aos dados e mostre uma evidência dos resultados — pode ser um *print* com as informações.

É possível aplicar um modelo de classificação para os Produtos? Se sim, aplique aos dados e mostre os principais resultados — pode ser um *print* com as informações.

*Realizei a atividade de acordo com a explicação do professor.

Durante toda a disciplina não encontrei um conteúdo prático para um melhor entendimento da atividade MAPA*

Atividade realizada no Google Colaboratory



27/04/2023, 20:03

MAPA.ipynb - Colaboratory

```

from google.colab import files

uploaded = files.upload()

import pandas as pd
import io

df = pd.read_csv('/content/products.csv')
print(df)

import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline

df.shape

(32951, 9)

df.columns

Index(['product_id', 'product_category_name', 'product_name_lenght',
       'product_description_lenght', 'product_photos_qty', 'product_weight_g',
       'product_length_cm', 'product_height_cm', 'product_width_cm',
       'product_vol_cm3', 'density_g/cm3', 'cluster'],
      dtype='object')

df.duplicated().sum()

0

df.describe()


```

	product_name_lenght	product_description_lenght	product_photos_qty
count	32340.000000	32340.000000	32340.000000
mean	48.476592	771.492393	1.492393
std	10.245699	635.124831	1.124831
min	5.000000	4.000000	0.000000
25%	42.000000	339.000000	1.000000
50%	48.000000	771.000000	1.000000
75%	55.000000	1111.000000	2.000000
max	255.000000	4499.000000	10.000000

```

df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 32951 entries, 0 to 32950
Data columns (total 9 columns):
#   Column                                Non-Null Count  Dtype  
---  --
0   product_id                            32951 non-null  object  
1   product_category_name                 32341 non-null  object  
2   product_name_lenght                   32341 non-null  float64  
3   product_description_lenght            32341 non-null  float64  
4   product_photos_qty                    32341 non-null  float64  
5   product_weight_g                      32949 non-null  float64  
6   product_length_cm                     32949 non-null  float64  
7   product_height_cm                     32949 non-null  float64  
8   product_width_cm                      32949 non-null  float64  
dtypes: float64(7), object(2)
memory usage: 2.3+ MB

df.isna().sum()[:20]

product_id                0
product_category_name      610
product_name_lenght        610
product_description_lenght  610
product_photos_qty         610
product_weight_g            2
product_length_cm           2

```

27/04/2023, 20:03

```
product_height_cm      2
product_width_cm       2
dtype: int64
```

```
df['product_vol_cm3'] = df.product_length_cm * \
    df.product_width_cm * df.product_height_cm
df['density_g/cm3'] = df.product_weight_g / \
    df.product_vol_cm3
display(df)
```

	product_id	product_cate
0	1e9e8ef04dbcf4541ed26657ea517e5	
1	3aa071139cb16b67ca9e5dea641aaa2f	
2	96bd76ec8810374ed1b65e291975717f	es
3	ce657bcfe19066a932b7673e239eb23d	
4	9dc1a7de274444849c219cff195d0b71	utilidades_
...
32946	a0b7d5a992coda846f2d34e418ff5a0	moveis
32947	b4538d88321d0fd4412a93c974510e6	construcao_ferramentas
32948	9a7c8041fa9592d9d9ef6cfe62a7118c	cama_m
32949	83808703fc0706a22e264b9d75f04a2e	informatica

```
df.dropna(subset= ['product_category_name', 'product_name_lenght', 'product_description_lenght', 'product_photos_qty'], inplace=True)
```

```
print(df.isna().sum())
```

```
df = df.dropna()
```

```
print(df.isna().sum())
```

```
product_id      0
product_category_name  0
product_name_lenght  0
product_description_lenght  0
product_photos_qty  0
product_weight_g  1
product_length_cm  1
product_height_cm  1
product_width_cm  1
product_vol_cm3  1
density_g/cm3    1
dtype: int64
product_id      0
product_category_name  0
product_name_lenght  0
product_description_lenght  0
product_photos_qty  0
product_weight_g  0
product_length_cm  0
product_height_cm  0
product_width_cm  0
product_vol_cm3  0
density_g/cm3    0
dtype: int64
```

```
x = df[['product_vol_cm3', 'density_g/cm3']]
```

```
n_clusters = 5
```

```
kmeans = KMeans(n_clusters=n_clusters)
```

```
kmeans.fit(x)
```

```
df["cluster"] = kmeans.labels_
```

```
print(df.groupby("cluster").mean())
```

27/04/2023, 20:03

```
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning: The default value of 'n_init' will change fr
warnings.warn(
product_name_length product_description_length product_photos_qty \
cluster
0 49.464288 733.075297 2.315383
1 48.752415 965.105072 2.283816
2 48.138769 774.398474 2.132084
3 48.623188 844.512077 2.531401
4 48.821598 775.028481 2.301828

product_weight_g product_length_cm product_height_cm \
cluster
0 2857.867150 40.198426 23.354421
1 15254.258454 56.748792 49.981884
2 813.003902 24.534177 11.488787
3 24759.106280 67.512077 62.908213
4 7607.809731 51.160206 34.227453

product_width_cm product_vol_cm3 density_g/cm3
cluster
0 31.283753 23572.892146 0.121314
1 45.425121 109744.096618 0.139890
2 17.952861 4978.678463 0.237615
3 55.268870 213799.922705 0.118159
4 38.958861 54281.453718 0.138924

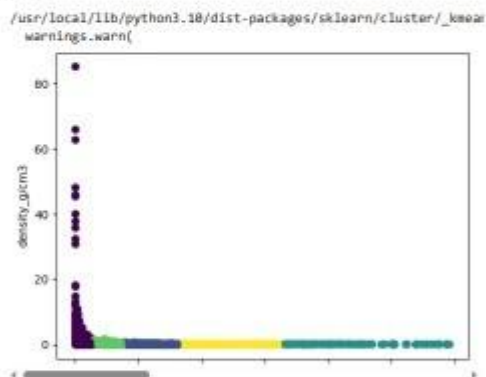
<ipython-input-123-22f401148fe2>:11: FutureWarning: The default value of numeric_only in DataFrameGroupBy.mean is deprecated. In a
print(df.groupby("cluster").mean())
```

```
x = df[["product_vol_cm3", "density_g/cm3"]]

kmeans = KMeans(n_clusters=5, random_state=42).fit(x)

df["cluster"] = kmeans.labels_

plt.scatter(x=df["product_vol_cm3"], y=df["density_g/cm3"], c=df["cluster"])
plt.xlabel("product_vol_cm3")
plt.ylabel("density_g/cm3")
plt.show()
```



```
from sklearn.preprocessing import StandardScaler

scaler = StandardScaler()

x = df[["product_vol_cm3", "density_g/cm3"]]

scaler.fit(x)

dados_escalados = scaler.transform(x)

kmeans = KMeans(n_clusters=5, random_state=0)
kmeans.fit(dados_escalados)

df["cluster"] = kmeans.labels_

plt.scatter(x=df["product_vol_cm3"], y=df["density_g/cm3"], c=df["cluster"])
```



27/04/2023, 20:03

MAPA.ipynb - Colaboratory

```
plt.xlabel("product_vol_cm3")  
plt.ylabel("density_g/cm3")  
plt.show()
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans  
warnings.warn(  

```

