

IESB

Pós Graduação em Inteligência Artificial

Disciplina: Aprendizado não Supervisionado

Docente: Mateus Mendelson

Discente: Henrique Brandão

Trabalho Final

```
In [1]: import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
```

```
%matplotlib notebook
```

Dataset: [Wholesale customers Data Set \(https://archive.ics.uci.edu/n/wholesale+customers\)](https://archive.ics.uci.edu/n/wholesale+customers)

Attribute	Information
FRESH	annual spending on fresh products
MILK	annual spending milk products
GROCERY	annual spending on grocery products
FROZEN	annual spending on frozen products
DETERGENTS_PAPER	annual spending on detergents and paper products
DELICATESSEN	annual spending on and delicatessen products
CHANNEL	customers Channel - Horeca (Hotel/Restaurant/Caffe) or Retail channel
REGION	customers Region - Lisbon, Porto or Other

```
In [2]: df = pd.read_csv('wholesale-customers-data.csv')
df.shape

(440, 8)
```

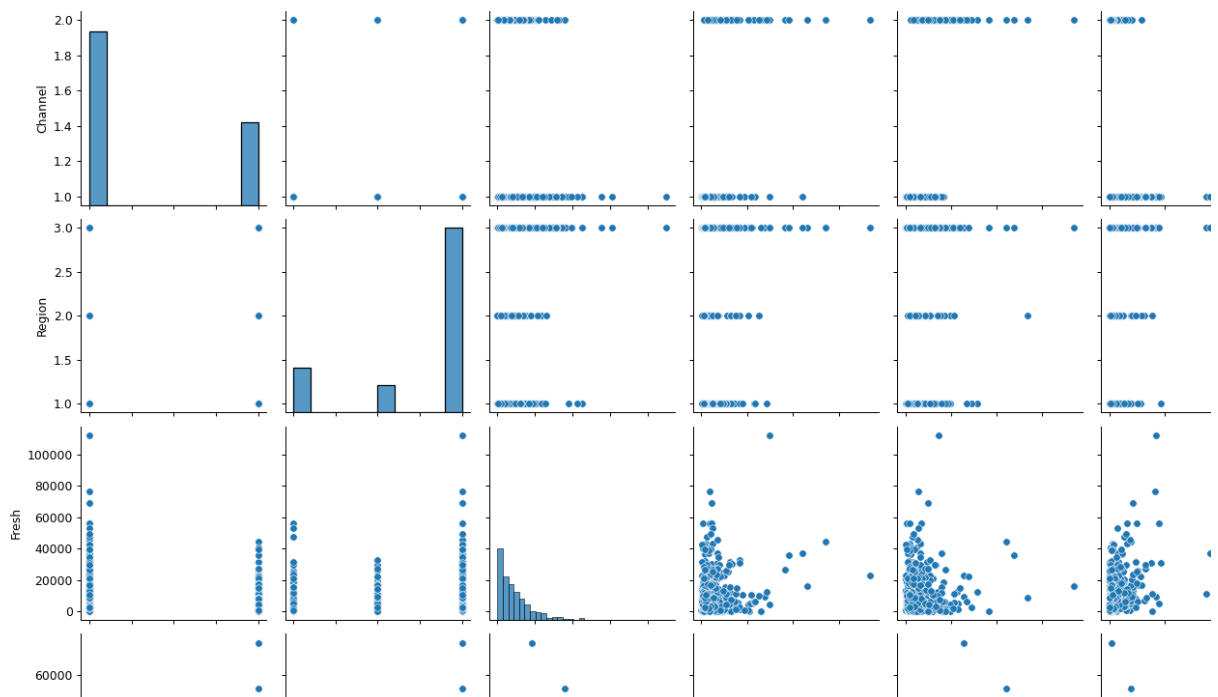
```
In [3]: df.head()
```

	Channel	Region	Fresh	Milk	Grocery	Frozen	Detergents_Paper	Delicas
0	2	3	12669	9656	7561	214	2674	1338
1	2	3	7057	9810	9568	1762	3293	1776

	Channel	Region	Fresh	Milk	Grocery	Frozen	Detergents_Paper	Delicas
2	2	3	6353	8808	7684	2405	3516	7844
3	1	3	13265	1196	4221	6404	507	1788

```
In [4]: pairplot = sns.pairplot(df)
pairplot.savefig('pairplot.png')
plt.title('Pairplot')
plt.show()
```

<IPython.core.display.Javascript object>



```
In [4]: df.describe()
```

	Channel	Region	Fresh	Milk	Grocery	Froz
count	440.000000	440.000000	440.000000	440.000000	440.000000	440.000000
mean	1.322727	2.543182	12000.297727	5796.265909	7951.277273	3071.9318
std	0.468052	0.774272	12647.328865	7380.377175	9503.162829	4854.6733
min	1.000000	1.000000	3.000000	55.000000	3.000000	25.000000
25%	1.000000	2.000000	3127.750000	1533.000000	2153.000000	742.250000
50%	1.000000	3.000000	8504.000000	3627.000000	4755.500000	1526.000000
75%	2.000000	3.000000	16933.750000	7190.250000	10655.750000	3554.250000
max	2.000000	3.000000	112151.000000	73498.000000	92780.000000	60869.000000

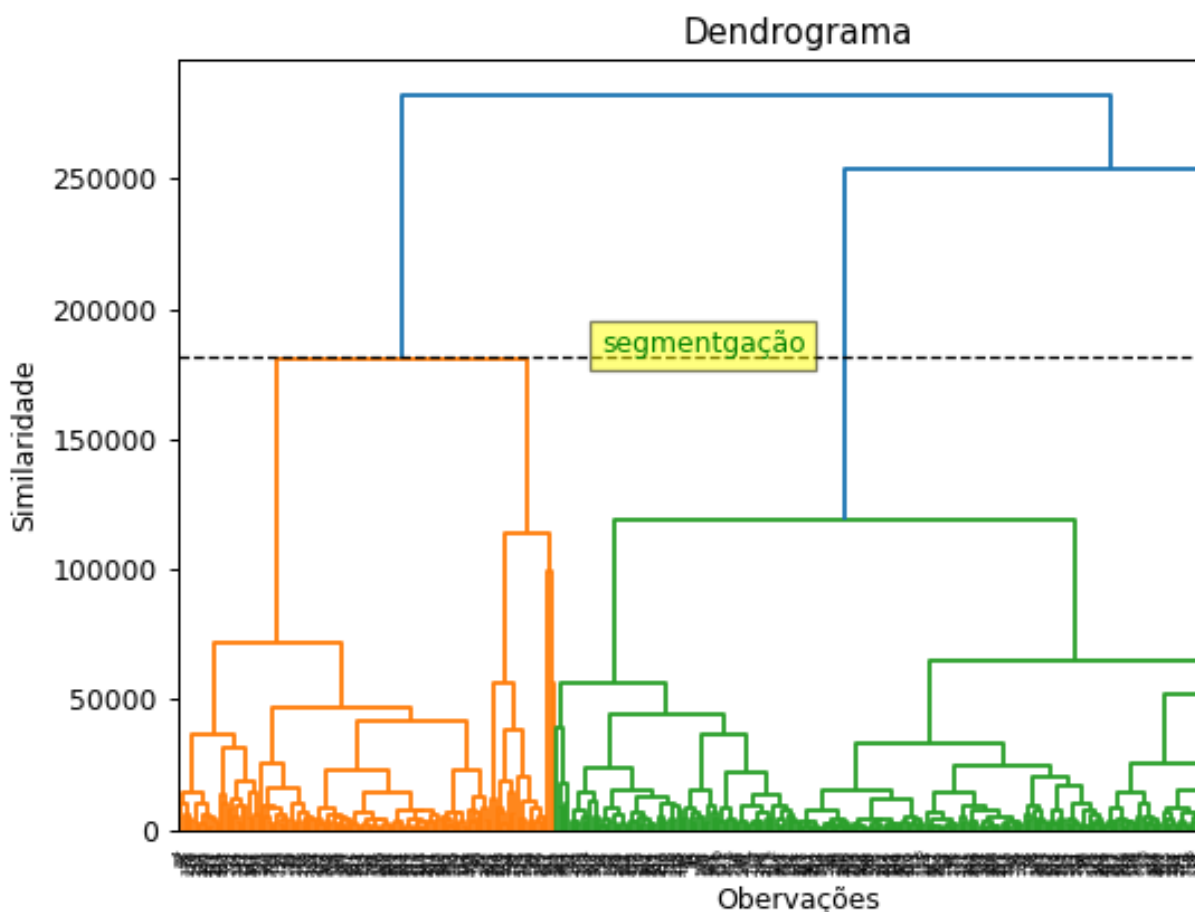
Dendrograma

Sobre Dendrograma :

Diagrama representativo de árvore para representação de dados, ilu
Exibe os grupos formados por agrupamento de observações em cad
similaridade. O nível de similaridade é medido ao longo do eixo verti
pode exibir o nível de distância) e as diferentes observações são list
horizontal.

```
In [5]: from scipy.cluster.hierarchy import dendrogram, linkage
```

```
In [6]: plt.figure(figsize=(8, 5))
plt.title("Dendrograma")
dendro = dendrogram(linkage(df, method='ward'))
plt.axhline(182000, linewidth=1, color='black', linestyle='--')
plt.text(x=1500, y=183000, s='segmentação',
         fontdict=dict(color='green',size=10),
         bbox=dict(facecolor='yellow',alpha=0.5))
plt.xlabel('Observações')
plt.ylabel('Similaridade')
plt.show()
```



Agglomerative Clustering

Sobre Agglomerative Clustering :

Emprega uma abordagem *bottom-up*, onde cada ponto começa em seu próprio cluster e estes são combinados de forma *greedy*, combinando dois clusters a cada passo até que todos os pontos pertencem a um único cluster. O ponto inicial começa como sendo seu próprio cluster, e estes são combinados de forma *greedy* até que todos os pontos pertencem a um único cluster.

```
In [7]: from sklearn.cluster import AgglomerativeClustering
```

```
In [9]: agg = AgglomerativeClustering(n_clusters=3)
```

```
In [10]: agg.fit(df)

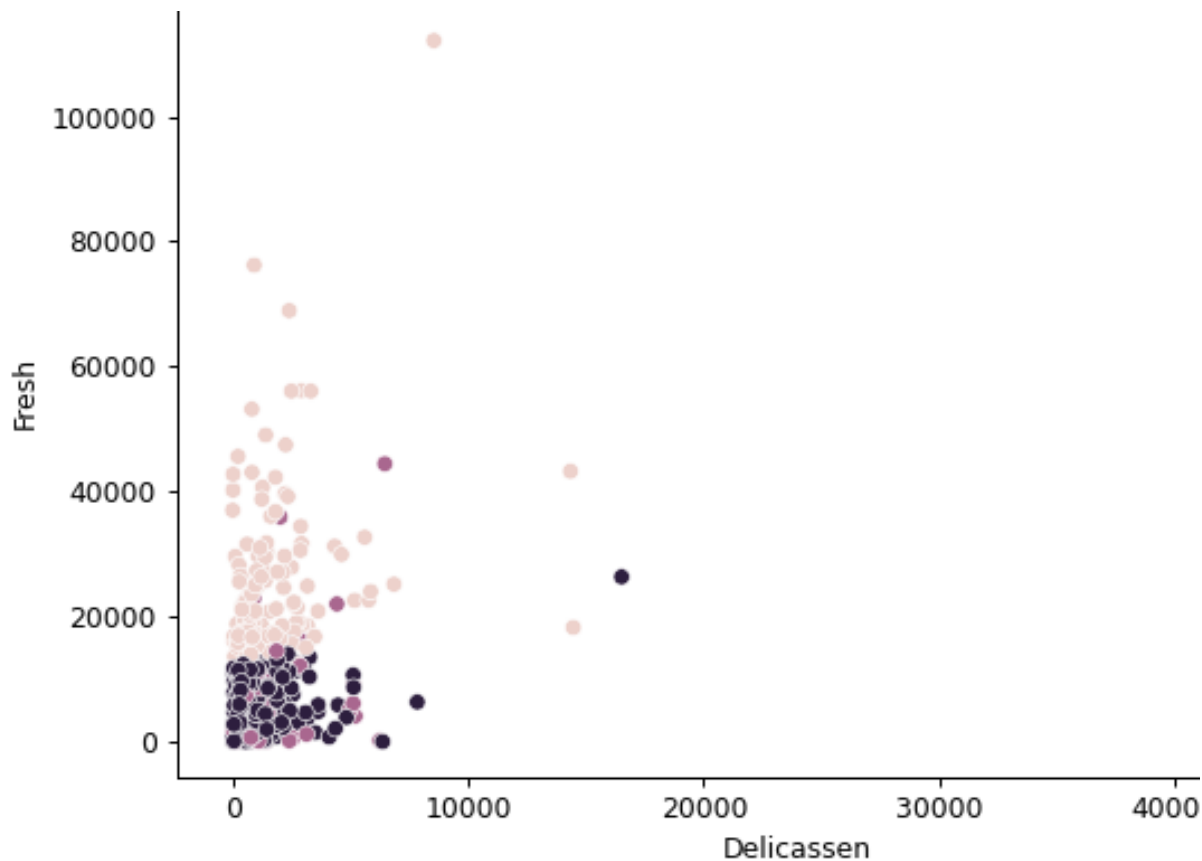
AgglomerativeClustering(n_clusters=3)
```

```
In [27]: len(agg.labels_), agg.labels_

(440,
 array([2, 2, 2, 2, 0, 2, 2, 2, 2, 1, 2, 2, 0, 0, 0, 2, 2, 2, 0, 2, 0, 2,
        0, 2, 0, 0, 2, 0, 1, 0, 0, 2, 0, 0, 2, 2, 0, 2, 1, 0, 0, 0, 2, 1,
        2, 1, 1, 1, 2, 1, 2, 2, 0, 2, 0, 2, 1, 2, 0, 2, 2, 1, 2, 2, 2, 1,
        2, 0, 2, 2, 0, 0, 2, 0, 2, 0, 2, 1, 2, 2, 2, 2, 2, 0, 2, 1, 1, 0,
        2, 0, 2, 2, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 0, 0, 0, 2, 2, 2, 1,
        2, 2, 0, 0, 0, 2, 2, 2, 0, 2, 0, 2, 0, 2, 0, 0, 0, 0, 2, 0, 2, 2,
        0, 2, 2, 2, 2, 2, 2, 2, 0, 0, 0, 2, 0, 1, 2, 2, 2, 0, 0, 2, 0, 2,
        2, 1, 2, 0, 2, 2, 2, 0, 0, 1, 2, 2, 2, 2, 2, 2, 2, 1, 2, 1, 2, 2,
        0, 2, 2, 2, 0, 0, 2, 0, 2, 2, 2, 2, 2, 2, 2, 0, 0, 2, 1, 2, 0, 0, 2,
        2, 2, 1, 1, 0, 2, 2, 1, 2, 2, 2, 1, 0, 1, 2, 2, 2, 2, 1, 0, 2, 2,
        0, 2, 2, 2, 2, 2, 0, 2, 2, 2, 2, 2, 0, 2, 0, 2, 2, 0, 0, 0, 0,
        0, 2, 2, 2, 2, 0, 0, 2, 2, 1, 2, 0, 2, 0, 2, 2, 0, 0, 2, 2, 0, 2,
        2, 2, 1, 0, 2, 0, 2, 2, 2, 0, 2, 2, 0, 2, 2, 0, 2, 2, 0, 0, 0, 0,
        2, 0, 0, 0, 2, 2, 2, 2, 0, 2, 0, 2, 2, 2, 0, 1, 2, 2, 1, 2, 1, 0,
        2, 2, 2, 0, 1, 2, 2, 2, 2, 2, 2, 1, 2, 2, 0, 0, 0, 0, 2, 2, 0, 2,
        2, 1, 0, 1, 0, 0, 0, 2, 2, 2, 2, 2, 2, 1, 2, 2, 2, 0, 2, 1, 2, 1,
        2, 1, 2, 2, 0, 1, 2, 2, 0, 2, 2, 2, 2, 2, 2, 2, 0, 2, 0, 0, 2, 0,
        2, 2, 2, 0, 2, 2, 0, 0, 0, 2, 2, 2, 2, 0, 2, 2, 2, 2, 2, 0, 2, 2,
        2, 2, 2, 2, 2, 0, 0, 0, 0, 2, 0, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2,
        1, 2, 2, 0, 0, 0, 0, 2, 2, 0, 2, 2, 2, 2, 0, 2, 0, 0, 0, 1, 2, 2]))
```

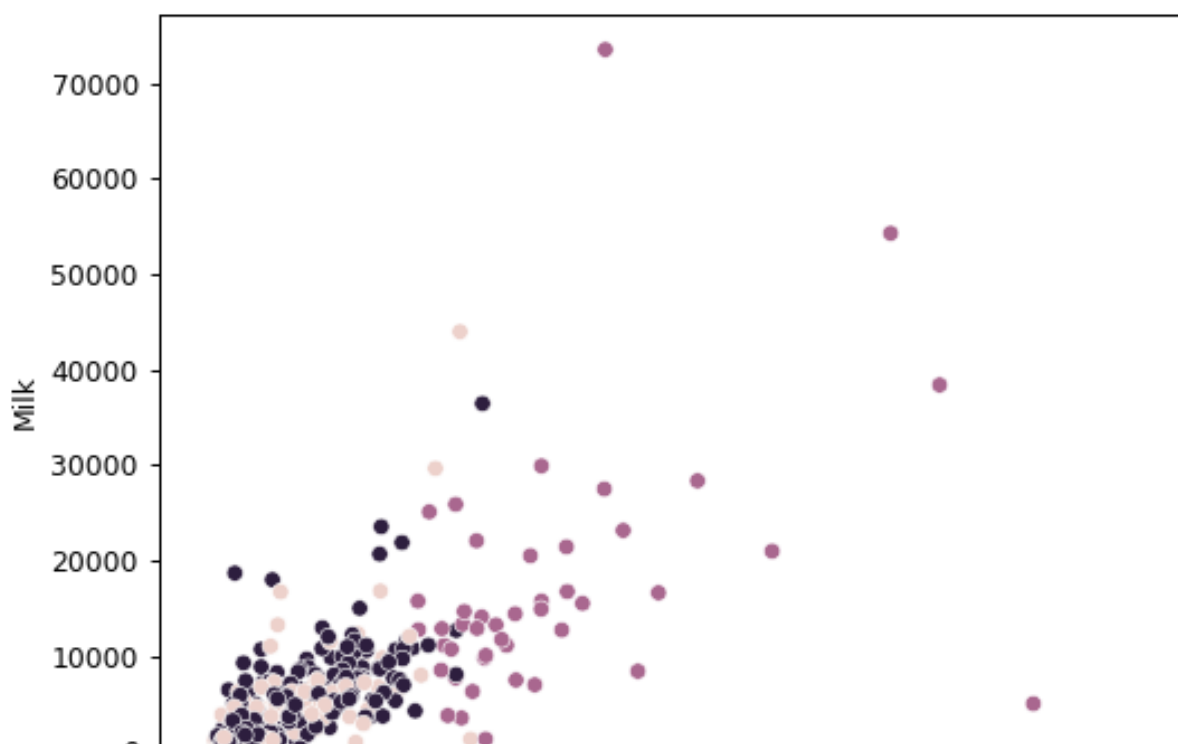
```
In [31]: plt.figure(figsize=(8, 5))
sns.scatterplot(x=df['Delicassen'], y=df['Fresh'], hue=agg.labels_)
plt.show()
```

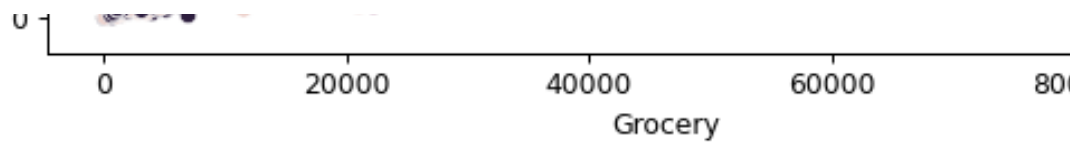
<IPython.core.display.Javascript object>



```
In [30]: plt.figure(figsize=(8, 5))
sns.scatterplot(x=df['Grocery'], y=df['Milk'], hue=agg.labels_)
plt.show()

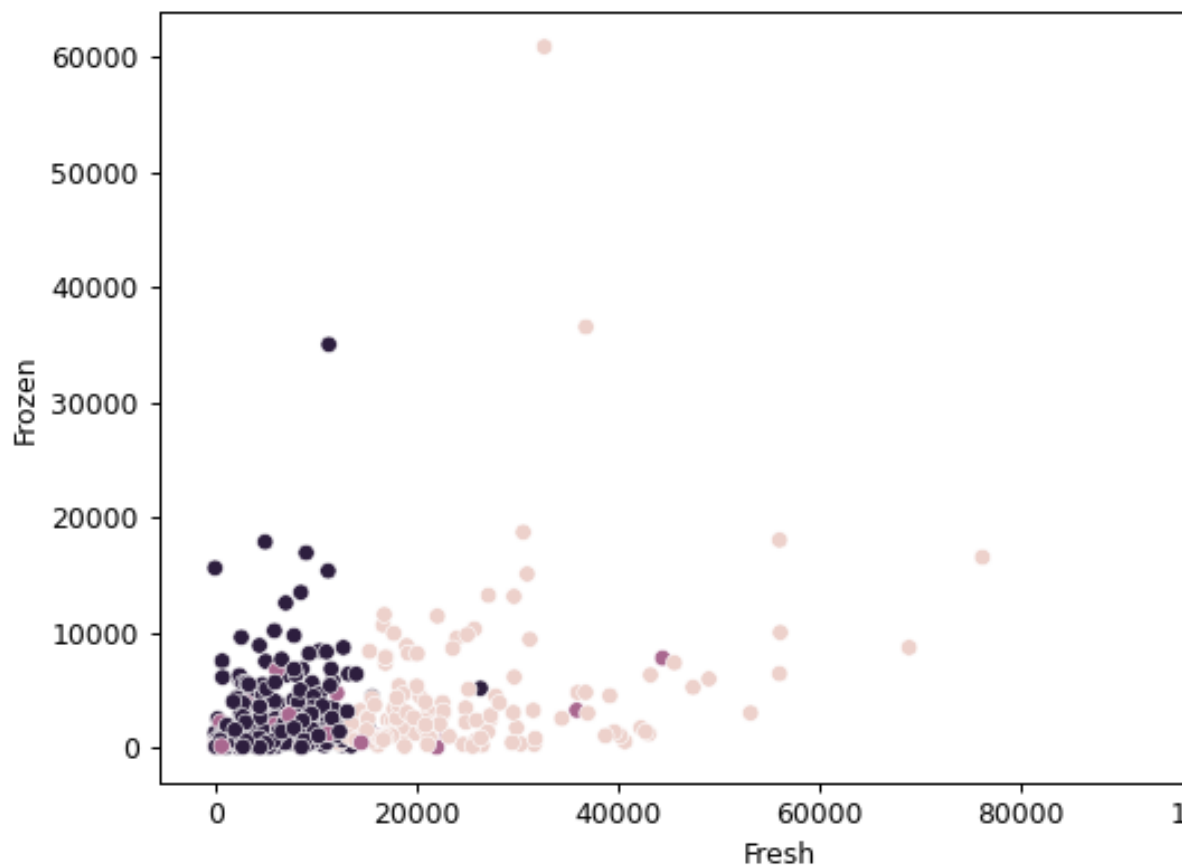
<IPython.core.display.Javascript object>
```





```
In [29]: plt.figure(figsize=(8, 5))
sns.scatterplot(x=df['Fresh'], y=df['Frozen'], hue=agg.labels_)
plt.show()

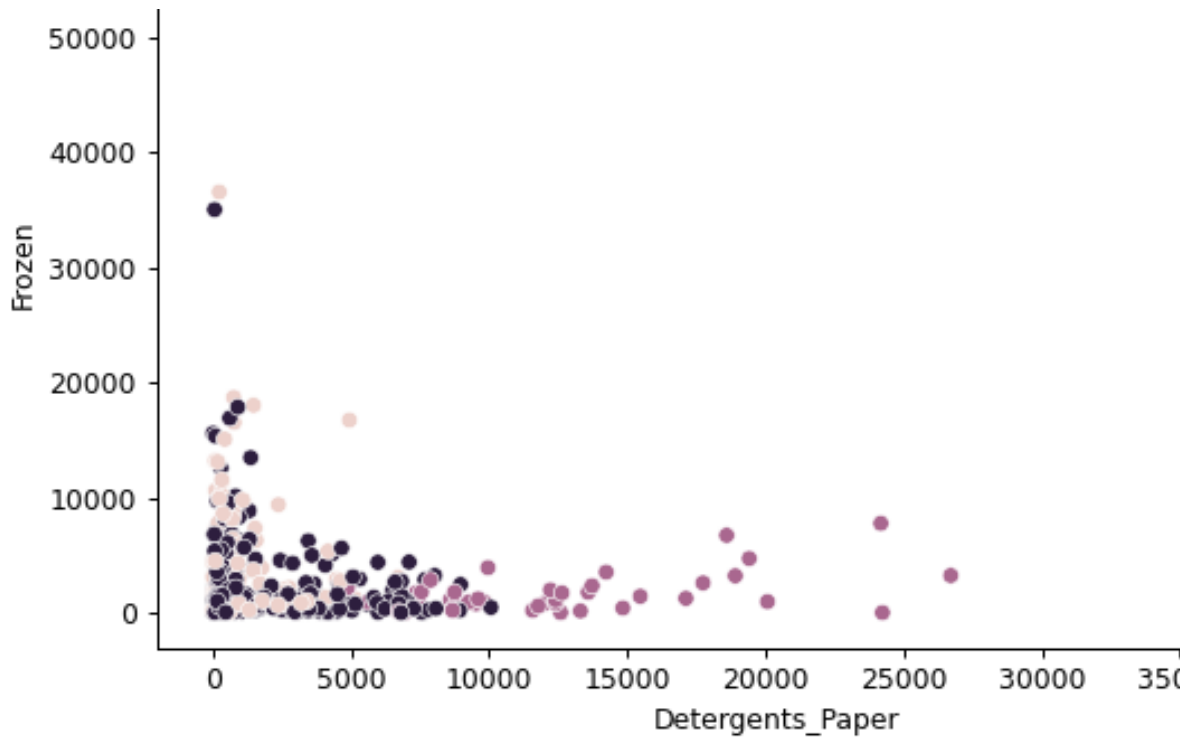
<IPython.core.display.Javascript object>
```



```
In [28]: plt.figure(figsize=(8, 5))
sns.scatterplot(x=df['Detergents_Paper'], y=df['Frozen'], hue=agg.labels_)
plt.show()

<IPython.core.display.Javascript object>
```





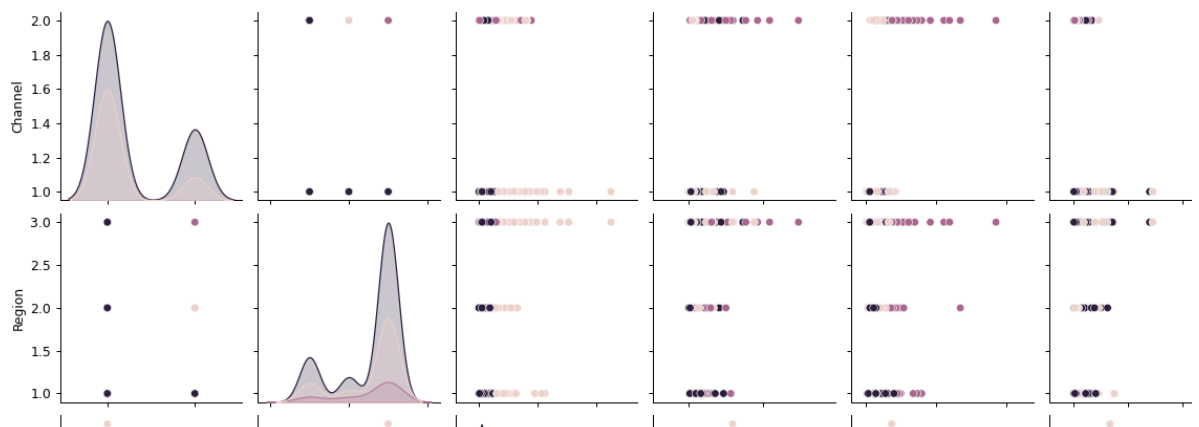
```
In [39]: df['cluster'] = agg.labels_
```

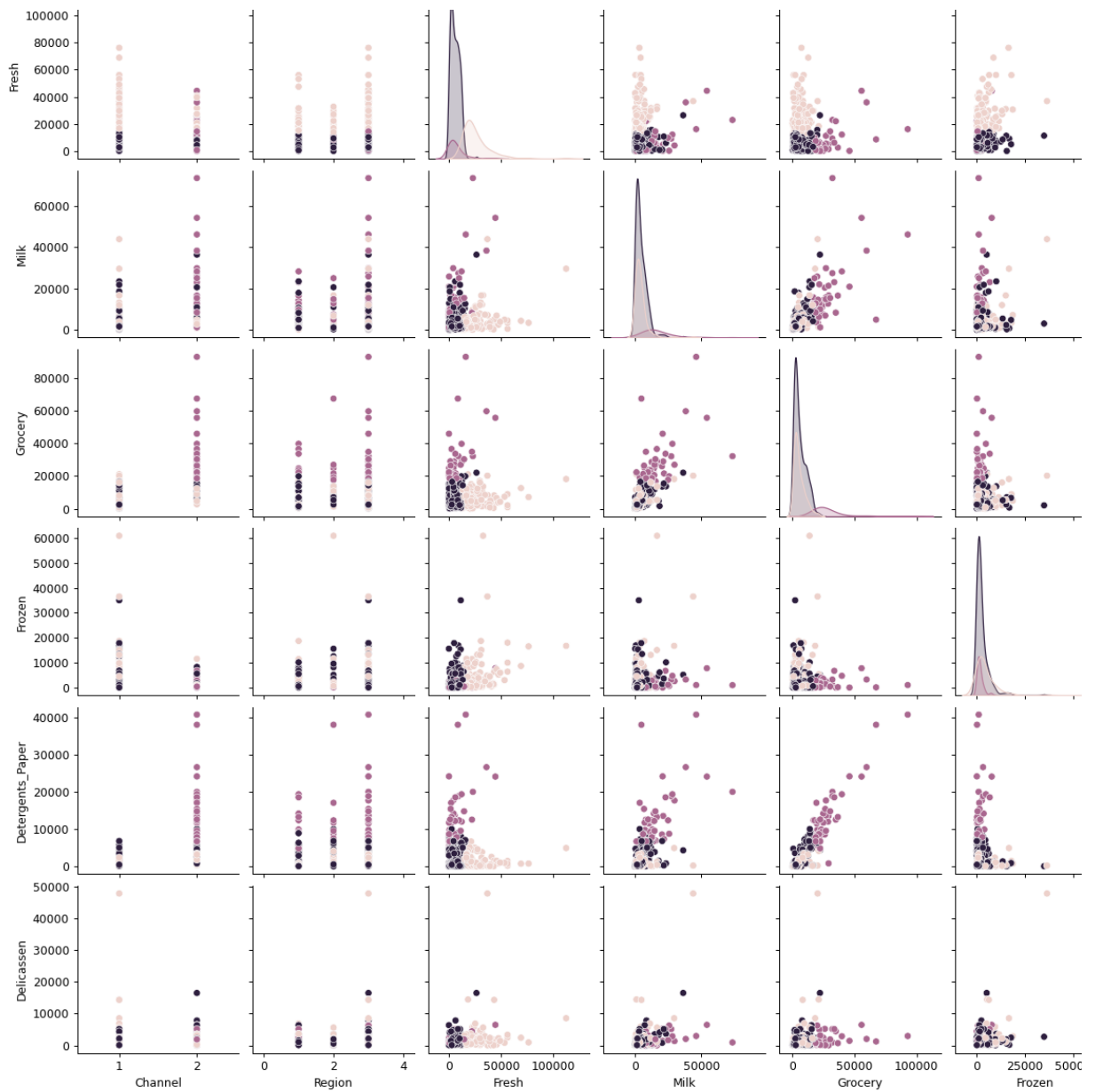
```
In [40]: df.head()
```

	Channel	Region	Fresh	Milk	Grocery	Frozen	Detergents_Paper	Delicas
0	2	3	12669	9656	7561	214	2674	1338
1	2	3	7057	9810	9568	1762	3293	1776
2	2	3	6353	8808	7684	2405	3516	7844
3	1	3	13265	1196	4221	6404	507	1788
4	2	3	22615	5410	7198	3915	1777	5185

```
In [42]: pairplot = sns.pairplot(df, hue='cluster')
pairplot.savefig('pairplot_clusters.png')
plt.show()
```

<IPython.core.display.Javascript object>





```
/home/hbrandao/Projects/post-grad-iesb-ai/venv/lib/python3.8/site-packages/seaborn/distr
has 0 variance; skipping density estimate.
warnings.warn(msg, UserWarning)
```

In []: