DBSCAN Project

The Data

Source: https://archive.ics.uci.edu/ml/datasets/Wholesale+customers

Margarida G. M. S. Cardoso, margarida.cardoso '@' iscte.pt, ISCTE-IUL, Lisbon, Portugal

Data Set Information:

Provide all relevant information about your data set.

Attribute Information:

```
1) FRESH: annual spending (m.u.) on fresh products (Continuous);
2) MILK: annual spending (m.u.) on milk products (Continuous);
3) GROCERY: annual spending (m.u.)on grocery products (Continuous);
4) FROZEN: annual spending (m.u.)on frozen products (Continuous)
5) DETERGENTS_PAPER: annual spending (m.u.) on detergents and paper products (Cont
inuous)
6) DELICATESSEN: annual spending (m.u.)on and delicatessen products (Continuous);
7) CHANNEL: customers  Channel - Horeca (Hotel/Restaurant/CafÃ©) or Retail channel
(Nominal)
8) REGION: customers  Region Lisnon, Oporto or Other (Nominal)
```

Relevant Papers:

Cardoso, Margarida G.M.S. (2013). Logical discriminant models â€“ Chapter 8 in Quantitative Modeling in Marketing and Management Edited by Luiz Moutinho and Kun-Huang Huarng. World Scientific. p. 223-253. ISBN 978-9814407717

Jean-Patrick Baudry, Margarida Cardoso, Gilles Celeux, Maria JosÃ© Amorim, Ana Sousa Ferreira (2012). Enhancing the selection of a model-based clustering with external qualitative variables. RESEARCH REPORT NÂ° 8124, October 2012, Project-Team SELECT. INRIA Saclay - ÃŽle-de-France, Projet select, UniversitÃ© Paris-Sud 11

DBSCAN and Clustering Examples

COMPLETE THE TASKS IN BOLD BELOW:

TASK: Run the following cells to import the data and view the DataFrame.

In [1]:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

In [2]:

```
df = pd.read_csv("D:\\Study\\Programming\\python\\Python course from udemy\\Udemy - 2022
Python for Machine Learning & Data Science Masterclass\\24 - DBSCAN - Density-based spati
al clustering of applications with noise\\33643066-wholesome-customers-data.csv")
```

In [3]:

```
df.head()
```

Out[3]:

	Channel	Region	Fresh	Milk	Grocery	Frozen	Detergents_Paper	Delicassen
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 [4]:

```
df.info()
```

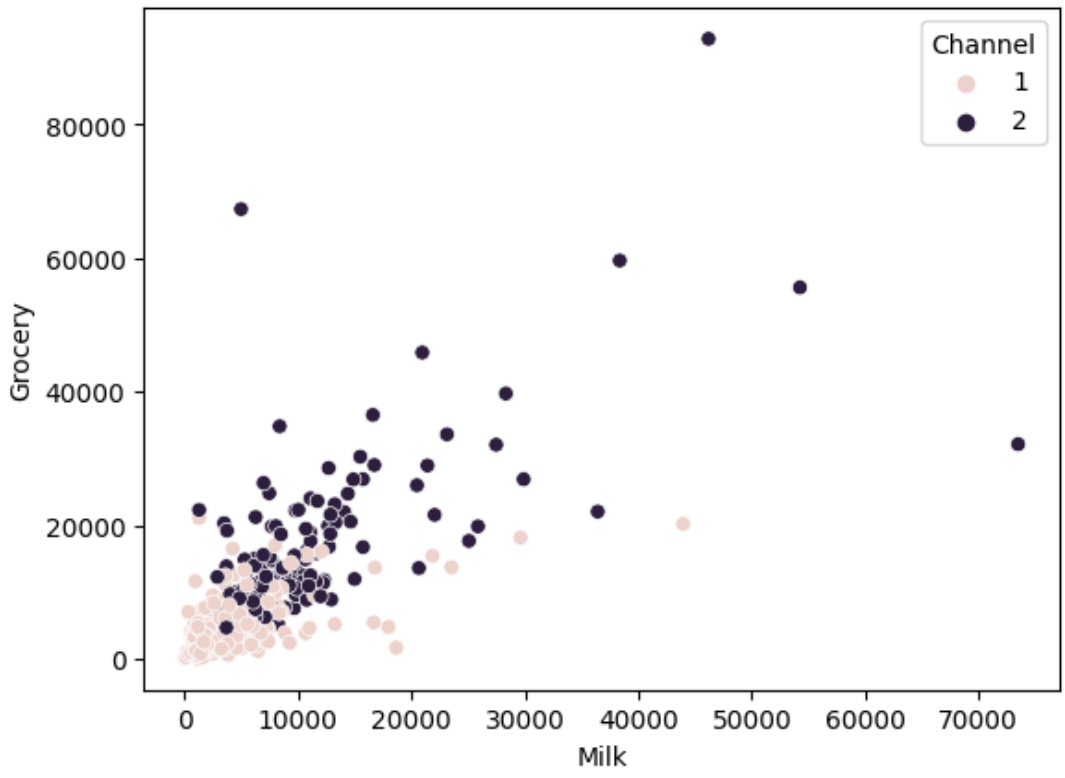
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 440 entries, 0 to 439
Data columns (total 8 columns):
 #   Column            Non-Null Count  Dtype
---  ------            --------------  -----
 0   Channel           440 non-null    int64
 1   Region            440 non-null    int64
 2   Fresh             440 non-null    int64
 3   Milk              440 non-null    int64
 4   Grocery           440 non-null    int64
 5   Frozen            440 non-null    int64
 6   Detergents_Paper  440 non-null    int64
 7   Delicassen        440 non-null    int64
dtypes: int64(8)
memory usage: 27.6 KB
```

EDA

TASK: Create a scatterplot showing the relation between MILK and GROCERY spending, colored by Channel column.

In [7]:

```
sns.scatterplot(data=df, x='Milk',y='Grocery',hue='Channel');
```



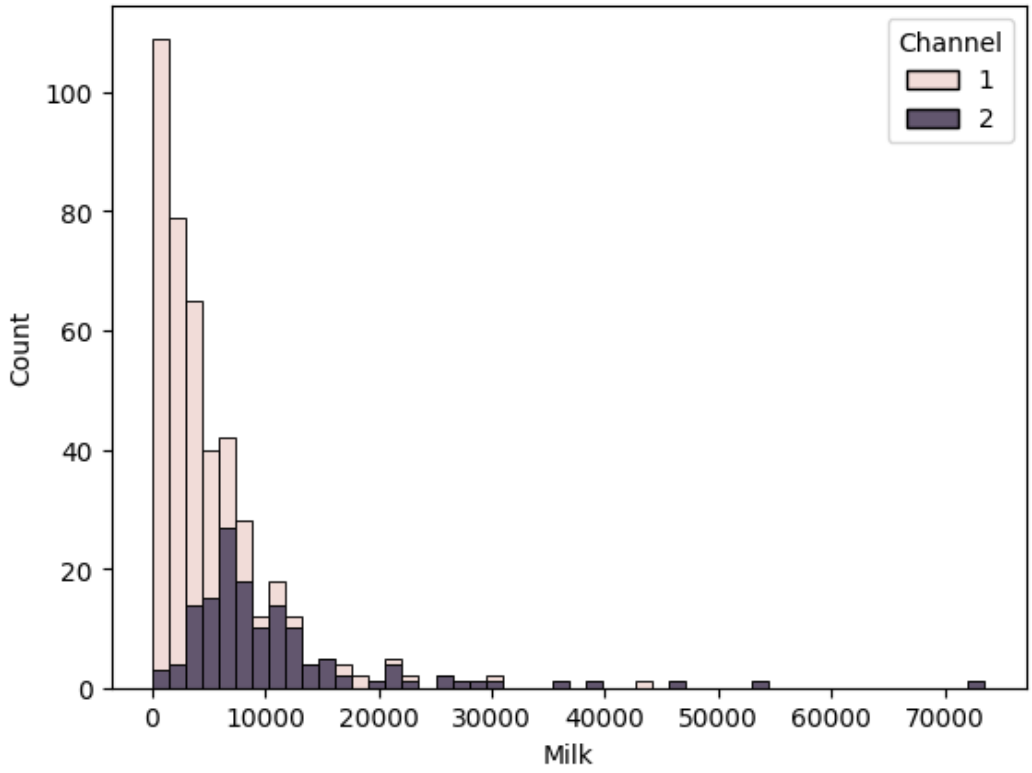
TASK: Use seaborn to create a histogram of MILK spending, colored by Channel. Can you figure out how to use seaborn to "stack" the channels, instead of have them overlap?

In [13]:

```
sns.histplot(data=df,x='Milk',hue='Channel',multiple="stack")
```

Out[13]:

```
<AxesSubplot: xlabel='Milk', ylabel='Count'>
```

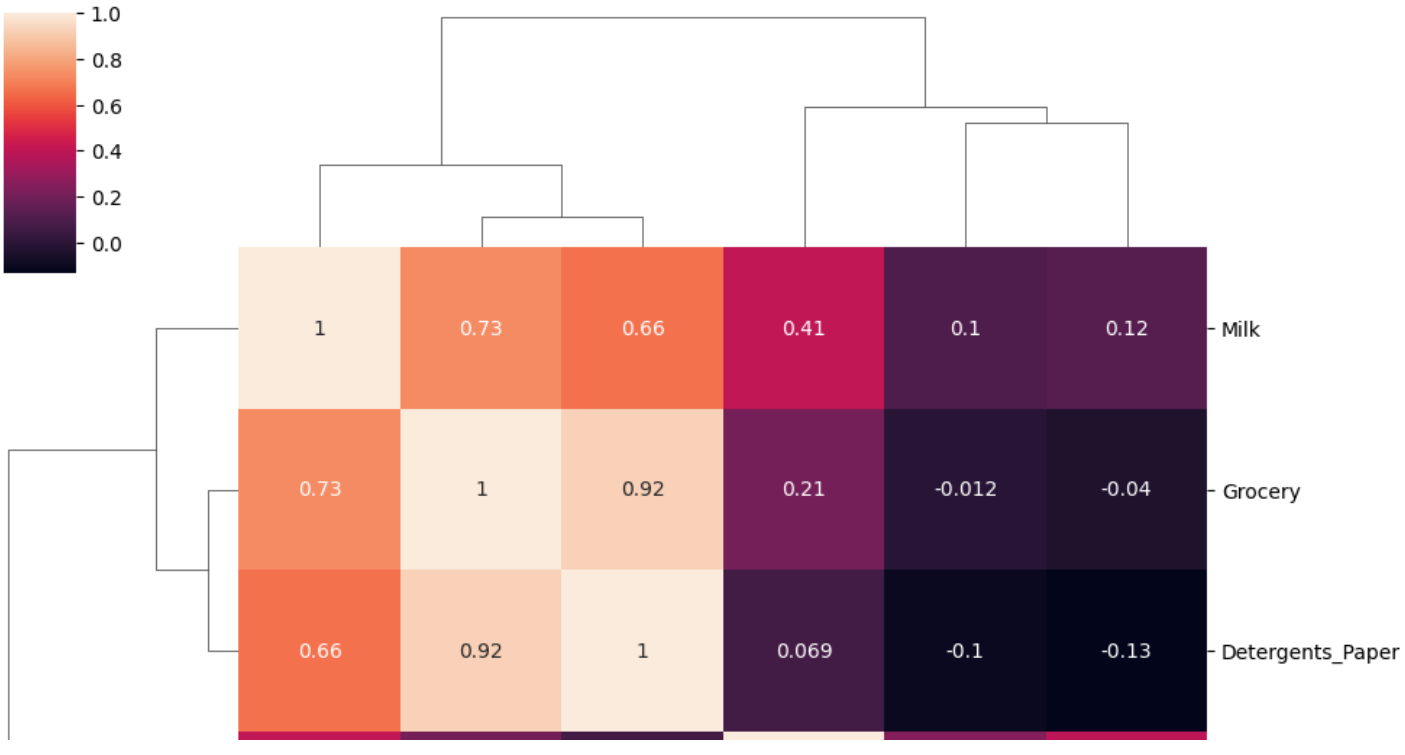


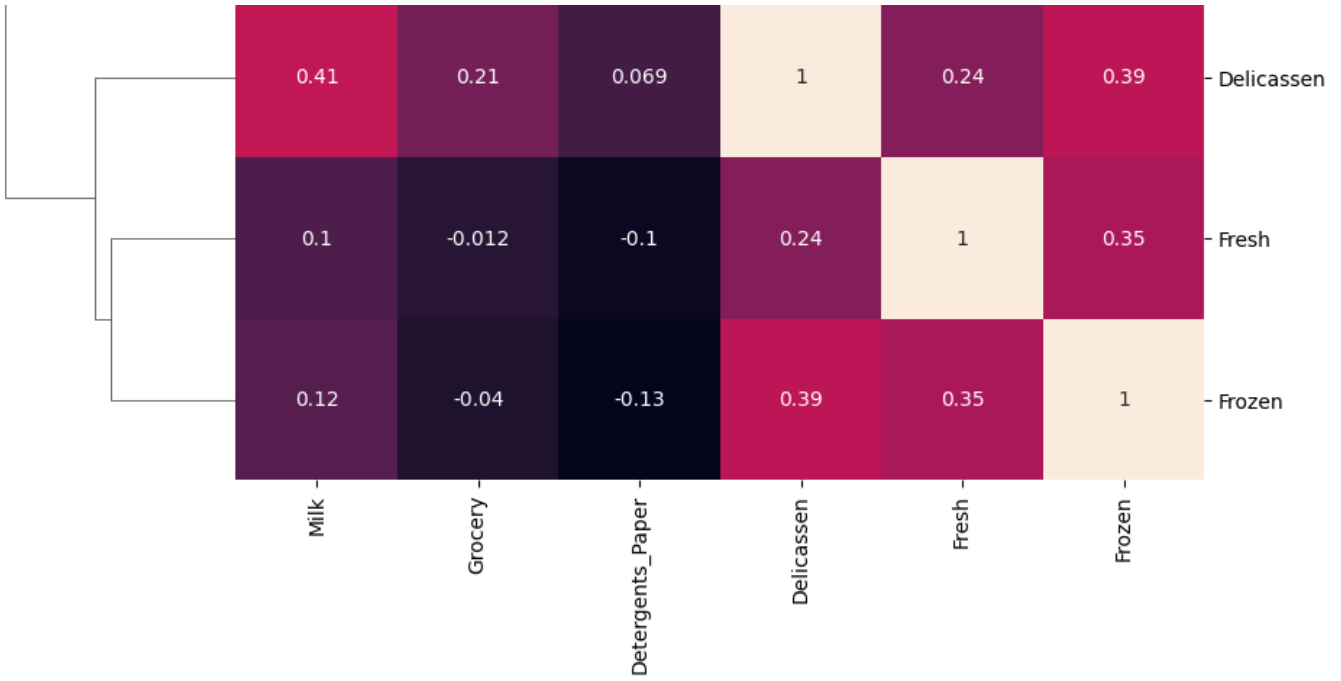
TASK: Create an annotated clustermap of the correlations between spending on different cateogires.

In [20]:

```
print('Correlation Between Spending Categories')
sns.clustermap(data=df.drop(['Region','Channel'],axis=1).corr(),annot=True);
```

```
Correlation Between Spending Categories
```

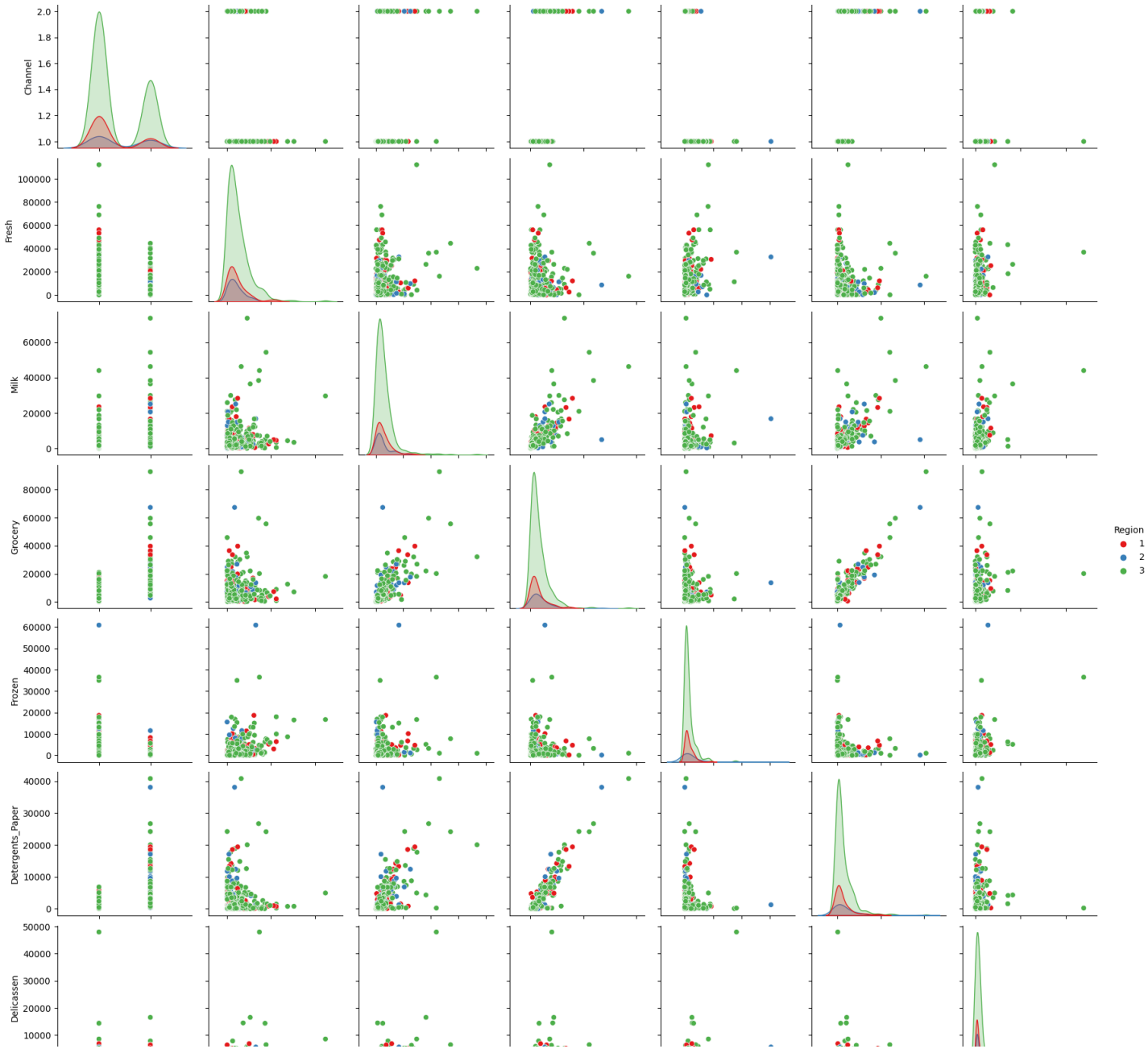


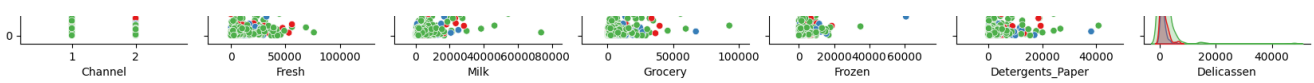


TASK: Create a PairPlot of the dataframe, colored by Region.

In [23]:

```
sns.pairplot(df,hue='Region',palette='Set1');
```





DBSCAN

TASK: Since the values of the features are in different orders of magnitude, let's scale the data. Use StandardScaler to scale the data.

In [24]:

```
from sklearn.preprocessing import StandardScaler
scaled_x = StandardScaler()
```

In [32]:

```
scaled_x = scaled_x.fit_transform(df)
```

TASK: Use DBSCAN and a for loop to create a variety of models testing different epsilon values. Set min_samples equal to 2 times the number of features. During the loop, keep track of and log the percentage of points that are outliers. For reference the solutions notebooks uses the following range of epsilon values for testing:

```
np.linspace(0.001,3,50)
```

In [27]:

```
from sklearn.cluster import DBSCAN
```

In [35]:

```
outlier_percent = []

for eps in np.linspace(0.001,3,50):
    dbscan = DBSCAN(eps=eps,min_samples=2*scaled_x.shape[1])
    dbscan.fit(scaled_x)

    # Log percentage of points that are outliers
    perc_outliers = 100 * np.sum(dbscan.labels_ == -1) / len(dbscan.labels_)

    outlier_percent.append(perc_outliers)
```

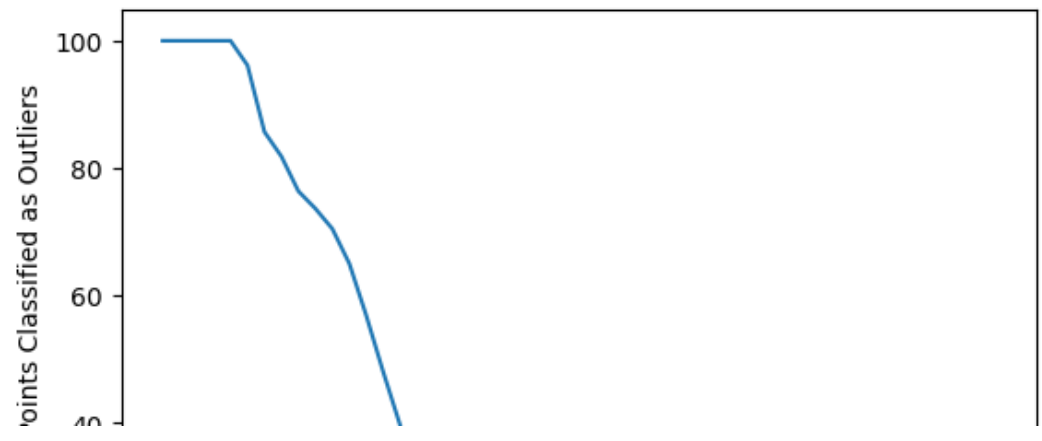
TASK: Create a line plot of the percentage of outlier points versus the epsilon value choice.

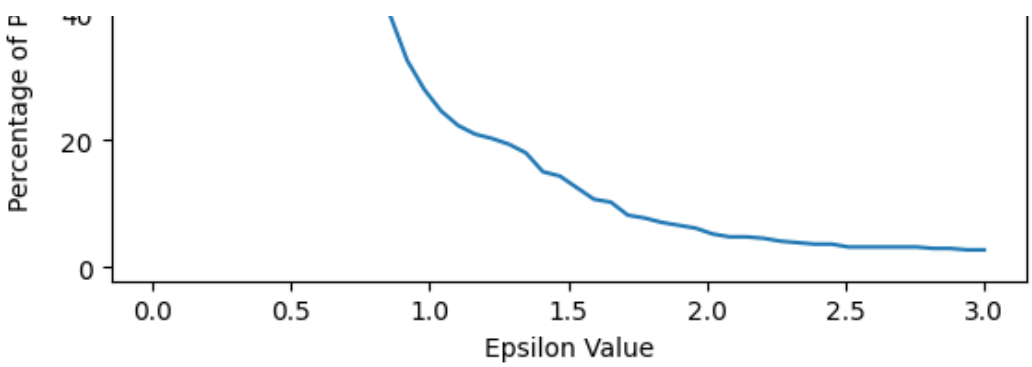
In [38]:

```
sns.lineplot(x=np.linspace(0.001,3,50),y=outlier_percent)
plt.ylabel("Percentage of Points Classified as Outliers")
plt.xlabel("Epsilon Value")
```

Out[38]:

```
Text(0.5, 0, 'Epsilon Value')
```





DBSCAN with Chosen Epsilon

TASK: Based on the plot created in the previous task, retrain a DBSCAN model with a reasonable epsilon value. Note: For reference, the solutions use eps=2.

In [39]:

```
dbscan = DBSCAN(eps=2)
dbscan.fit(scaled_x)
```

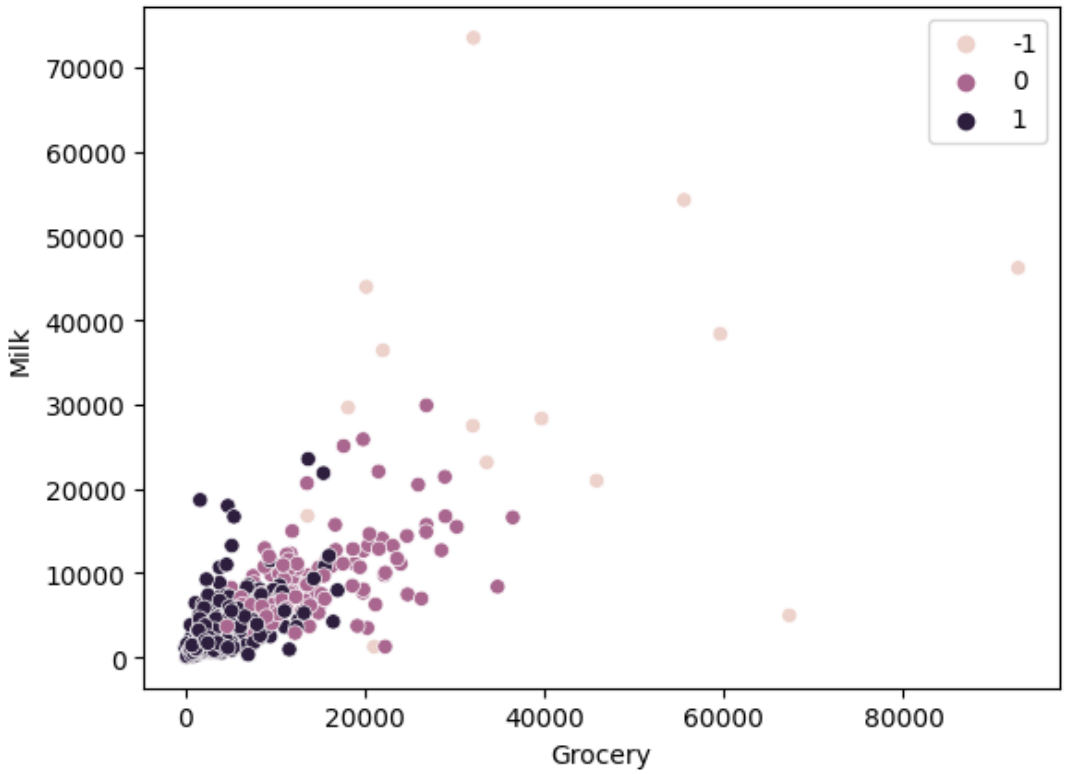
Out[39]:

```
DBSCAN
DBSCAN(eps=2)
```

TASK: Create a scatterplot of Milk vs Grocery, colored by the discovered labels of the DBSCAN model.

In [41]:

```
sns.scatterplot(data=df,x='Grocery',y='Milk',hue=dbscan.labels_);
```



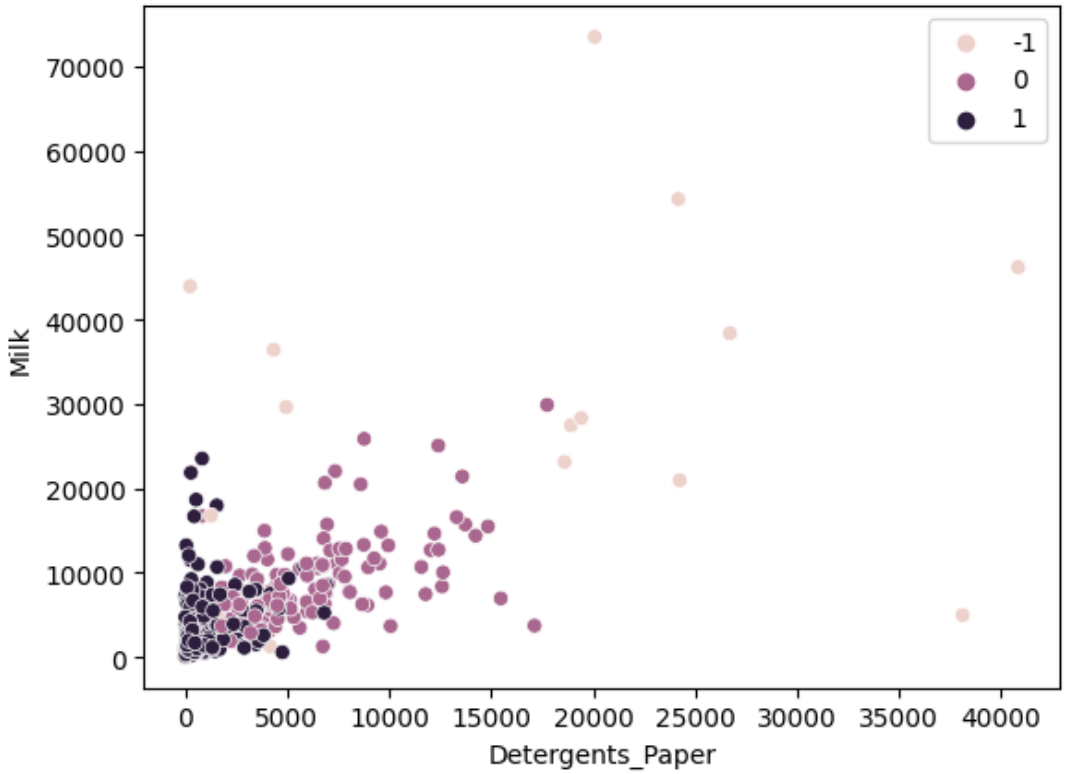
TASK: Create a scatterplot of Milk vs. Detergents Paper colored by the labels.

In [44]:

```
sns.scatterplot(data=df,x='Detergents_Paper',y='Milk',hue=dbscan.labels_)
```

Out[44]:

<AxesSubplot: xlabel='Detergents_Paper', ylabel='Milk'>



TASK: Create a new column on the original dataframe called "Labels" consisting of the DBSCAN labels.

In [45]:

```
df['Labels'] = dbscan.labels_
```

In [47]:

```
df.head()
```

Out[47]:

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

TASK: Compare the statistical mean of the clusters and outliers for the spending amounts on the categories.

In [49]:

```
cat = df.drop(['Channel','Region'],axis=1)
cat_mean = cat.groupby('Labels').mean()
```

In [50]:

```
cat_mean
```

Out[50]:

	Fresh	Milk	Grocery	Frozen	Detergents_Paper	Delicassen
Labels						
-1	30161.529412	26872.411765	33575.823529	12380.235294	14612.294118	8185.411765

	Fresh	Milk	Grocery	Frozen	Detergents_Paper	Delicassen
0	8200.681818	8849.446970	13919.113636	1527.174242	6037.280303	1548.310606
1	12662.869416	3180.065292	3747.250859	3228.862543	764.697595	1125.134021

Labels

TASK: Normalize the dataframe from the previous task using MinMaxScaler so the spending means go from 0-1 and create a heatmap of the values.

In [51]:

```
from sklearn.preprocessing import MinMaxScaler
```

In [55]:

```
scaler = MinMaxScaler()
data = scaler.fit_transform(cat_mean)
scaler_mean = pd.DataFrame(data,cat_mean.index,cat_mean.columns)
```

In [56]:

```
scaler_mean
```

Out[56]:

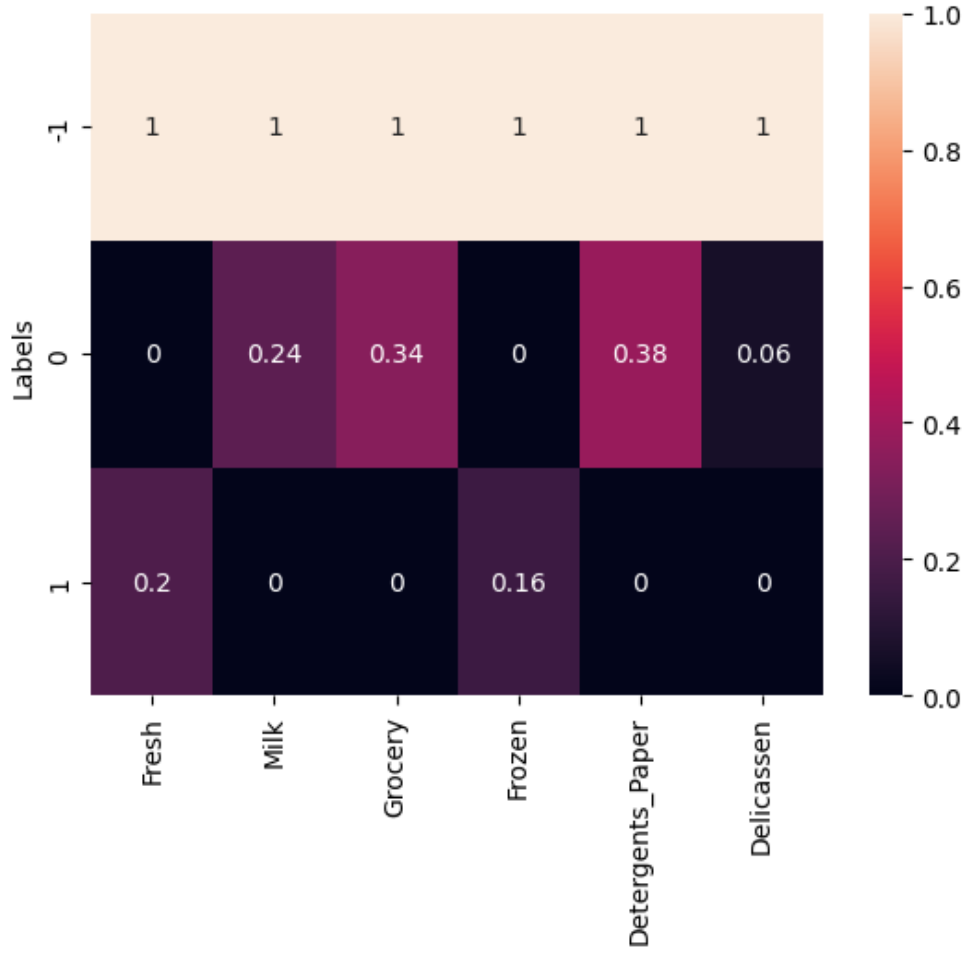
Labels	Fresh	Milk	Grocery	Frozen	Detergents_Paper	Delicassen
-1	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
0	0.000000	0.239292	0.341011	0.000000	0.380758	0.059938
1	0.203188	0.000000	0.000000	0.156793	0.000000	0.000000

In [63]:

```
sns.heatmap(scaler_mean,annot=True)
```

Out[63]:

```
<AxesSubplot: ylabel='Labels'>
```



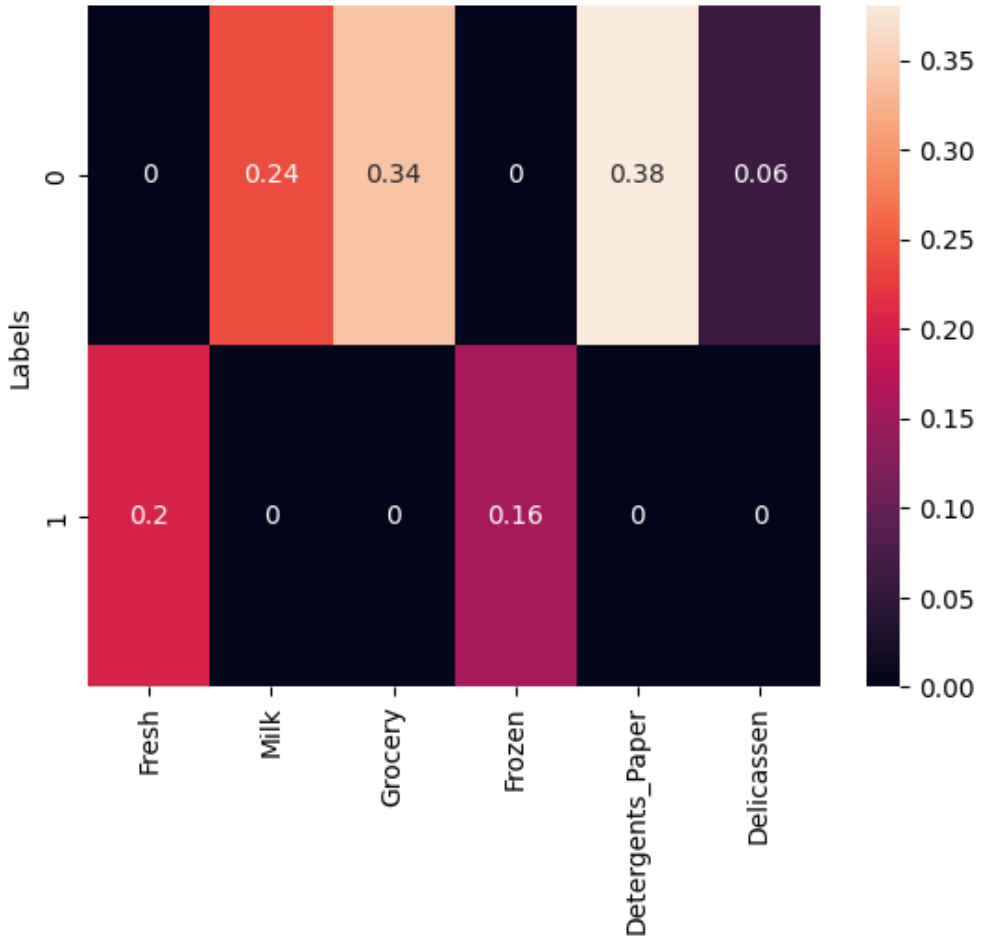
TASK: Create another heatmap similar to the one above, but with the outliers removed

In [62]:

```
sns.heatmap(scaler_mean.loc[[0,1]],annot=True)
```

Out[62]:

```
<AxesSubplot: ylabel='Labels'>
```



TASK: What spending category were the two clusters mode different in?

We can see that Detergents Paper was the most significant difference.