In [1]:

```
import pandas as pd
import numpy as np
```

In [2]:

```
data=pd.read_csv(r"C:\Users\prasa\OneDrive\Desktop\Wholesale customers data.csv")
data
```

Out[2]:

	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
...	...	...	...	...	...	...	...	...
435	1	3	29703	12051	16027	13135	182	2204
436	1	3	39228	1431	764	4510	93	2346
437	2	3	14531	15488	30243	437	14841	1867
438	1	3	10290	1981	2232	1038	168	2125
439	1	3	2787	1698	2510	65	477	52

440 rows × 8 columns

In [3]:

```
data.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]:

```
data.tail()
```

Out[4]:

	Channel	Region	Fresh	Milk	Grocery	Frozen	Detergents_Paper	Delicassen
435	1	3	29703	12051	16027	13135	182	2204
436	1	3	39228	1431	764	4510	93	2346
437	2	3	14531	15488	30243	437	14841	1867
438	1	3	10290	1981	2232	1038	168	2125
439	1	3	2787	1698	2510	65	477	52

In [5]:

```
# Normalize the data  i.e scales each variable to range 0 to 1
# why we need to scale?
# if data is not scaled, then the model is biased towards high magnitudes values which
```

In [6]:

```
#from sklearn.preprocessing import MinMaxScaler
#scaler=MinMaxScaler()
#new_data=scaler.fit_transform(data)
#new_data
```

In [7]:

```
from sklearn.preprocessing import normalize
new_data=normalize(data)
new_data
```

Out[7]:

```
array([[1.11821406e-04, 1.67732109e-04, 7.08332695e-01, ...,
        1.19648904e-02, 1.49505220e-01, 7.48085205e-02],
       [1.25321880e-04, 1.87982820e-04, 4.42198253e-01, ...,
        1.10408576e-01, 2.06342475e-01, 1.11285829e-01],
       [1.24839188e-04, 1.87258782e-04, 3.96551681e-01, ...,
        1.50119124e-01, 2.19467293e-01, 4.89619296e-01],
       ...,
       [5.01633106e-05, 7.52449659e-05, 3.64461533e-01, ...,
        1.09606834e-02, 3.72236846e-01, 4.68274505e-02],
       [9.11309417e-05, 2.73392825e-04, 9.37737390e-01, ...,
        9.45939175e-02, 1.53099982e-02, 1.93653251e-01],
       [2.41225630e-04, 7.23676891e-04, 6.72295832e-01, ...,
        1.56796660e-02, 1.15064626e-01, 1.25437328e-02]])
```

In [8]:

```
new_data=pd.DataFrame(new_data,columns=data.columns)
new_data
```

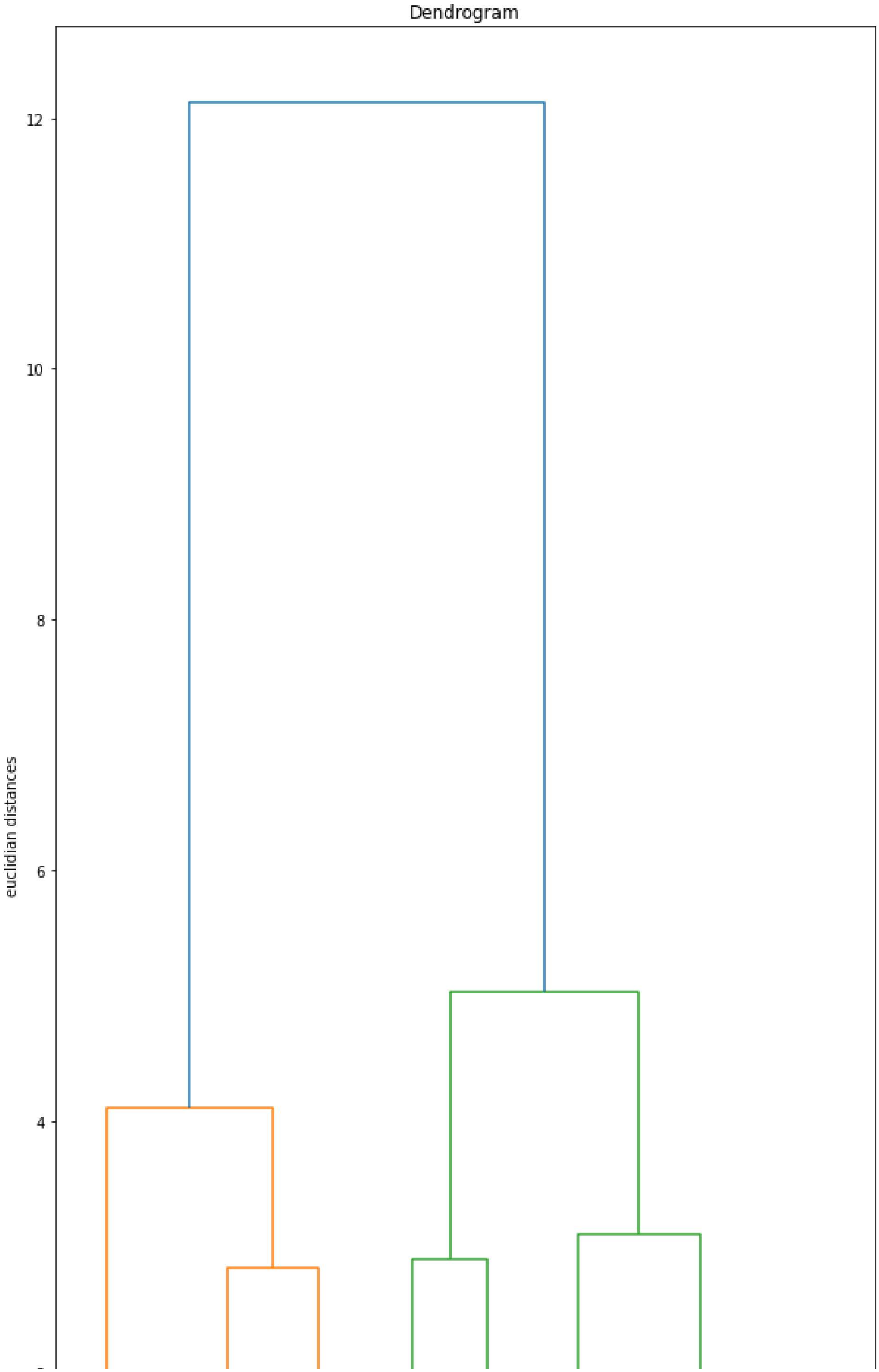
Out[8]:

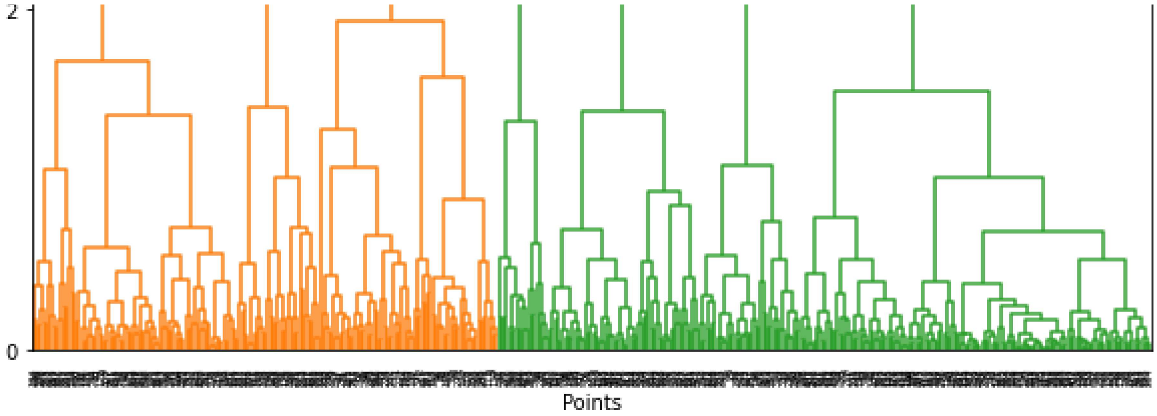
	Channel	Region	Fresh	Milk	Grocery	Frozen	Detergents_Paper	Delicassen
0	0.000112	0.000168	0.708333	0.539874	0.422741	0.011965	0.149505	0.074809
1	0.000125	0.000188	0.442198	0.614704	0.599540	0.110409	0.206342	0.111286
2	0.000125	0.000187	0.396552	0.549792	0.479632	0.150119	0.219467	0.489619
3	0.000065	0.000194	0.856837	0.077254	0.272650	0.413659	0.032749	0.115494
4	0.000079	0.000119	0.895416	0.214203	0.284997	0.155010	0.070358	0.205294
...	...	...	...	...	...	...	...	...
435	0.000026	0.000078	0.776890	0.315197	0.419191	0.343549	0.004760	0.057646
436	0.000025	0.000076	0.990872	0.036146	0.019298	0.113919	0.002349	0.059258
437	0.000050	0.000075	0.364462	0.388465	0.758545	0.010961	0.372237	0.046827
438	0.000091	0.000273	0.937737	0.180530	0.203404	0.094594	0.015310	0.193653
439	0.000241	0.000724	0.672296	0.409601	0.605476	0.015680	0.115065	0.012544

440 rows × 8 columns

In [9]:

```
#determine no.of clusters by using the Dendrogram
import scipy.cluster.hierarchy as shc
import matplotlib.pyplot as plt
plt.figure(figsize=(10,20))
plt.title("Dendrogram")
plt.xlabel("Points")
plt.ylabel("euclidian distances")
dend=shc.dendrogram(shc.linkage(new_data,method="ward"))
```



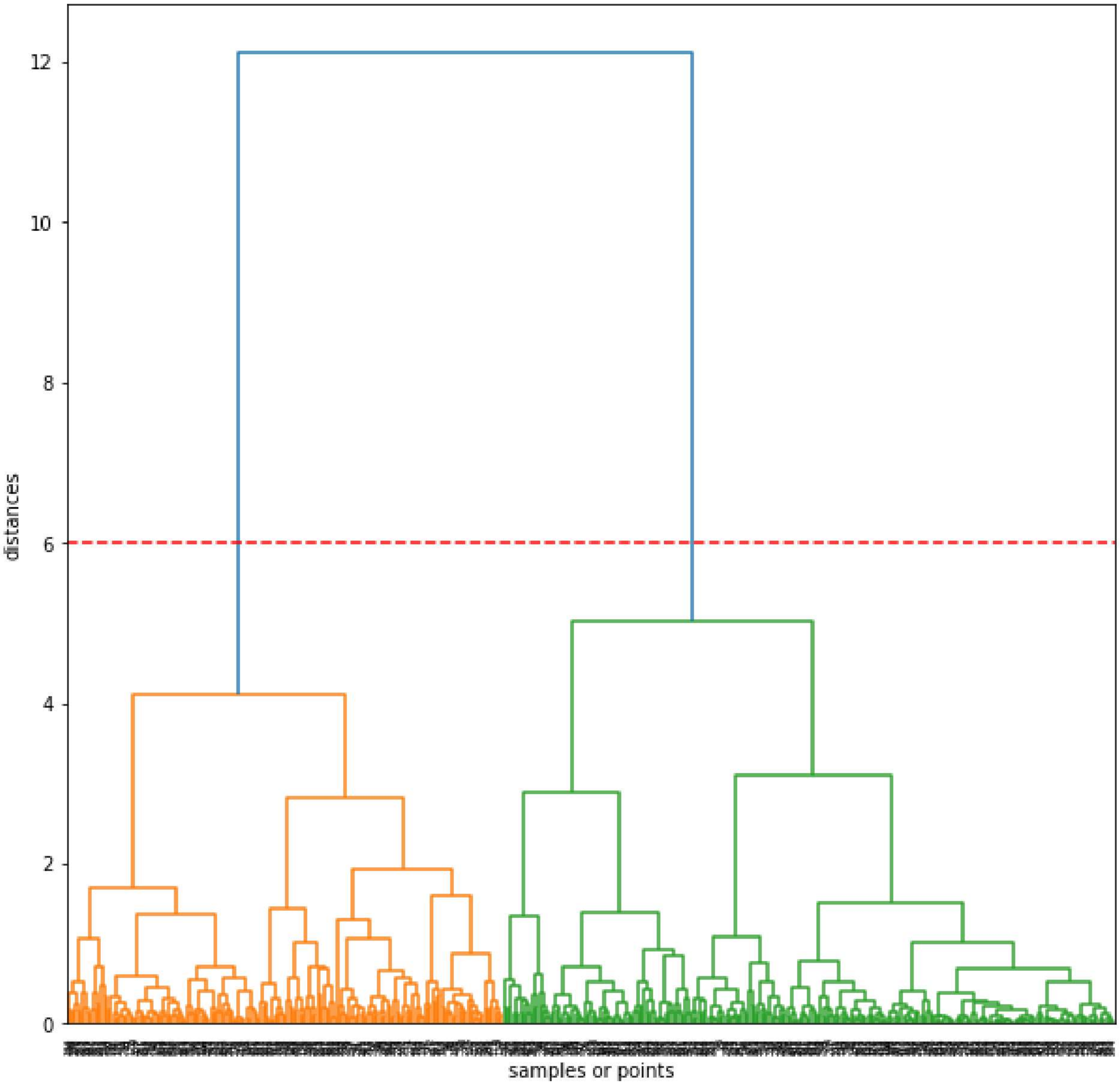


In [10]:

```
# how do u decide the threshold to cut the dendrogram to know the no.of clusters.
plt.figure(figsize=(10,10))
plt.ylabel("distances")
plt.xlabel("samples or points")
dend=shc.dendrogram(shc.linkage(new_data,method="ward"))
plt.axhline(y=6,color="red",linestyle="--")  # draw a horizontal line
```

Out[10]:

```
<matplotlib.lines.Line2D at 0x2af5cf43040>
```



since the horizontal line crosses the vertical axis at 2 different points. the no.of clusters is 2

In [11]:

```
from sklearn.cluster import AgglomerativeClustering
cluster=AgglomerativeClustering(n_clusters=2,affinity="euclidean",memory=None,linkage="
cluster
```

Out[11]:

```
AgglomerativeClustering()
```

In [12]:

```
cluster.fit_predict(new_data)
```

Out[12]:

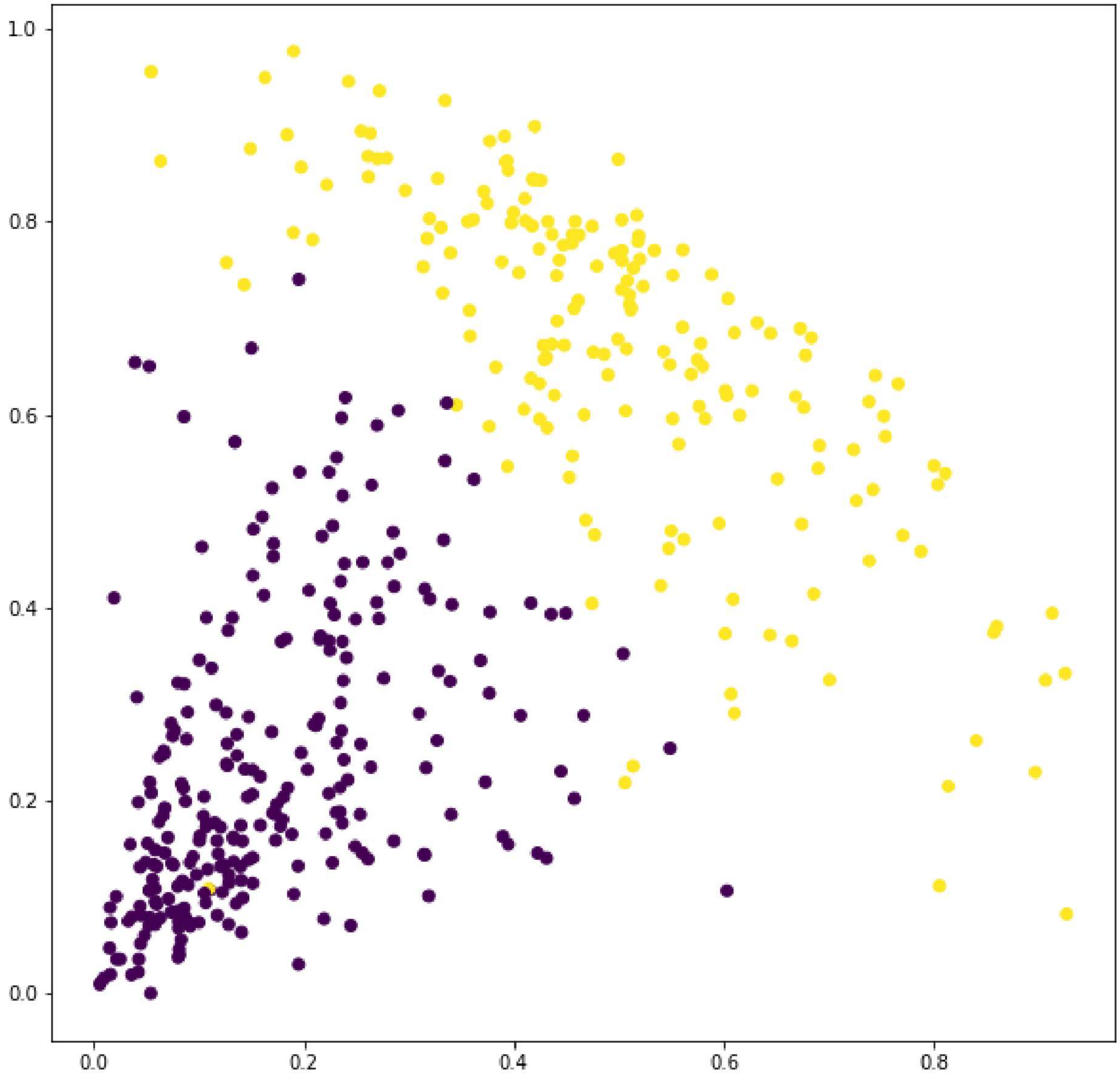
```
array([1, 1, 1, 0, 0, 1, 0, 1, 1, 1, 1, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0,
       0, 1, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 1, 1, 0, 1, 1, 0, 0, 0, 1, 1,
       1, 1, 1, 1, 1, 1, 0, 1, 0, 1, 0, 1, 1, 1, 0, 1, 0, 1, 1, 1, 0, 1,
       1, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 1, 0, 1, 1, 0, 0, 1, 1, 0,
       0, 0, 0, 0, 1, 0, 1, 1, 1, 0, 0, 0, 1, 1, 1, 0, 0, 0, 1, 1, 1, 1,
       0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0,
       0, 0, 0, 0, 1, 1, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 1,
       0, 1, 1, 0, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1, 1,
       0, 0, 0, 1, 0, 0, 1, 1, 1, 0, 0, 1, 1, 1, 0, 0, 0, 1, 0, 0, 0, 1,
       0, 0, 1, 1, 0, 1, 1, 1, 0, 1, 1, 1, 0, 1, 0, 1, 1, 1, 1, 0, 1, 0,
       0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0,
       0, 0, 1, 1, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1,
       1, 1, 1, 0, 1, 0, 0, 1, 1, 0, 1, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 1, 0, 1, 1, 0, 0, 0, 0, 1, 1, 0, 1, 1, 1, 1, 1, 1, 0,
       0, 1, 0, 0, 1, 0, 0, 1, 0, 1, 0, 1, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0,
       0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 0, 1, 1, 0, 1, 1, 0, 1,
       1, 1, 0, 1, 0, 1, 1, 1, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0,
       0, 0, 1, 0, 1, 1, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 1, 1, 0, 0, 0,
       1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 1, 0, 0, 1, 1, 1,
       1, 0, 1, 0, 0, 0, 0, 0, 1, 0, 1, 0, 1, 0, 0, 1, 0, 0, 0, 1, 0, 1],
      dtype=int64)
```

In [13]:

```
plt.figure(figsize=(10,10))
plt.scatter(new_data["Milk"],new_data["Grocery"],c=cluster.labels_)
```

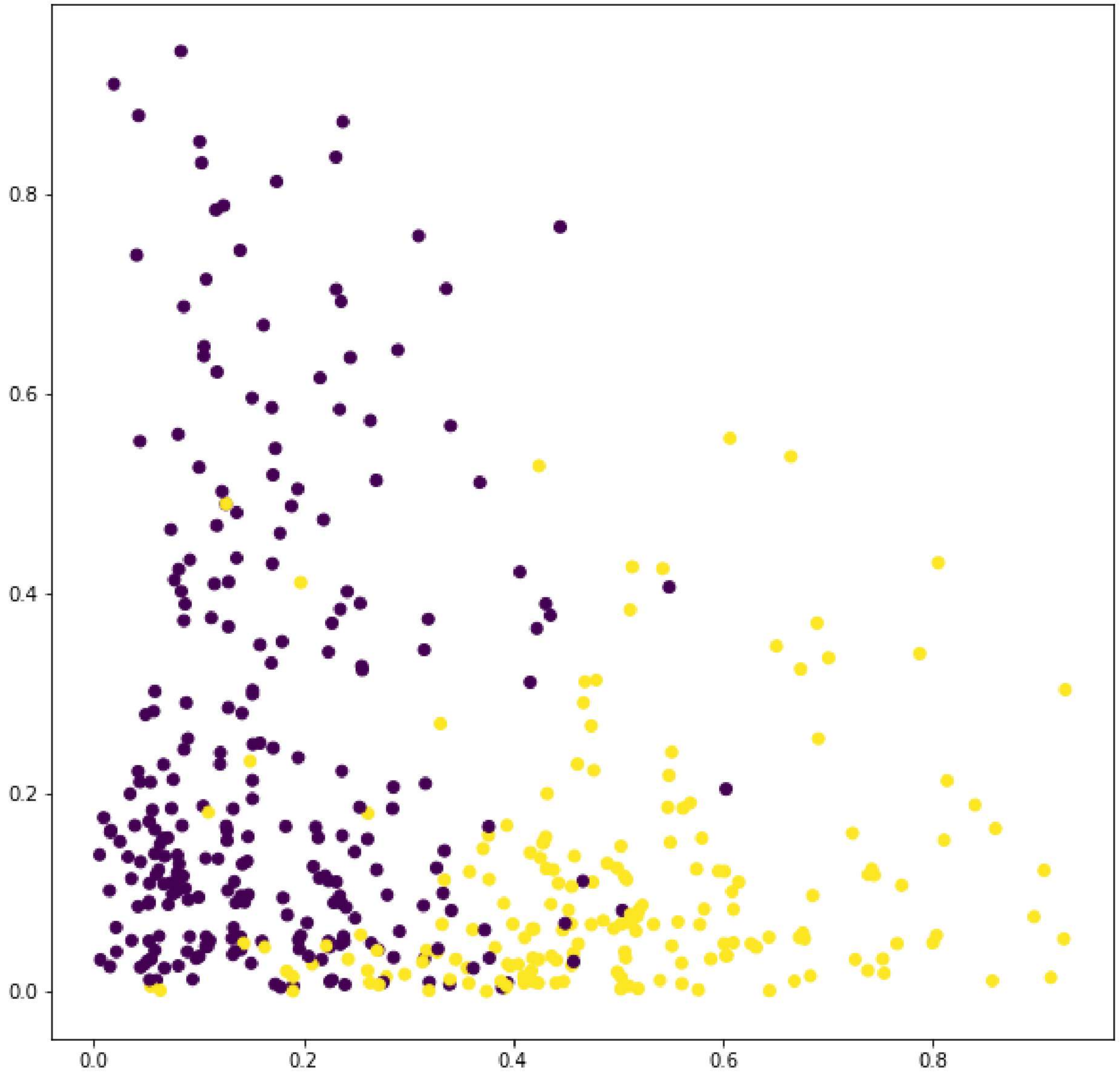
Out[13]:

```
<matplotlib.collections.PathCollection at 0x2af5ef46130>
```



```
In [14]: plt.figure(figsize=(10,10))
         plt.scatter(new_data["Milk"],new_data["Frozen"],c=cluster.labels_)
```

```
Out[14]: <matplotlib.collections.PathCollection at 0x2af5d065f40>
```



```
In [ ]:
```