

1.Import Libraries

```
In [1]: import pandas as pd
from sklearn.preprocessing import StandardScaler
from sklearn.cluster import DBSCAN
```

2.Import Datasets

```
In [2]: data=pd.read_csv('Wholesale customers data.csv')
data.head()
```

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

3.Data Understanding

```
In [3]: data.shape
```

Out[3]: (440, 8)

```
In [4]: data.isnull().sum()
```

Out[4]: Channel 0
Region 0
Fresh 0
Milk 0
Grocery 0
Frozen 0
Detergents_Paper 0
Delicassen 0
dtype: int64

```
In [5]: data.dtypes
```

Out[5]: Channel int64
Region int64
Fresh int64
Milk int64
Grocery int64
Frozen int64
Detergents_Paper int64
Delicassen int64
dtype: object

```
In [6]: data=data.drop(['Channel',"Region"],axis=1)
```

```
In [7]: array=data.values
array
```

Out[7]: array([[12669, 9656, 7561, 214, 2674, 1338],
[7057, 9810, 9568, 1762, 3293, 1776],
[6353, 8808, 7684, 2405, 3516, 7844],
...,
[14531, 15488, 30243, 437, 14841, 1867],
[10290, 1981, 2232, 1038, 168, 2125],
[2787, 1698, 2510, 65, 477, 52]], dtype=int64)

4.Data Processing

```
In [8]: sc=StandardScaler()  
s_data=sc.fit_transform(array)  
s_data=pd.DataFrame(s_data,columns=data.columns)  
s_data
```

Out[8]:

	Fresh	Milk	Grocery	Frozen	Detergents_Paper	Delicassen
0	0.052933	0.523568	-0.041115	-0.589367	-0.043569	-0.066339
1	-0.391302	0.544458	0.170318	-0.270136	0.086407	0.089151
2	-0.447029	0.408538	-0.028157	-0.137536	0.133232	2.243293
3	0.100111	-0.624020	-0.392977	0.687144	-0.498588	0.093411
4	0.840239	-0.052396	-0.079356	0.173859	-0.231918	1.299347
...
435	1.401312	0.848446	0.850760	2.075222	-0.566831	0.241091
436	2.155293	-0.592142	-0.757165	0.296561	-0.585519	0.291501
437	0.200326	1.314671	2.348386	-0.543380	2.511218	0.121456
438	-0.135384	-0.517536	-0.602514	-0.419441	-0.569770	0.213046
439	-0.729307	-0.555924	-0.573227	-0.620094	-0.504888	-0.522869

440 rows × 6 columns

```
In [9]: dbscan=DBSCAN(eps=0.8,min_samples=6)  
dbscan.fit(s_data)
```

Out[9]: DBSCAN(eps=0.8, min_samples=6)

```
In [10]: dbscan.labels_
```

Out[10]: array([0, 0, -1, 0, -1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
-1, 0, 0, 0, 0, -1, -1, -1, 0, 0, 0, -1, 0, 0, 0, 0, -1,
 0, 0, 0, 0, 0, -1, -1, 0, 0, 0, 0, -1, 0, -1, 0, -1, 0,
 0, 0, 0, 0, 0, -1, 0, 0, 0, 0, -1, 0, 0, 0, -1, 0, 0,
 0, 0, 0, -1, 0, 0, 0, 0, 0, -1, 0, 0, 0, 0, 0, 0, 0,
-1, -1, -1, 0, 0, 0, 0, -1, -1, 0, 0, 0, 0, 0, 0, 0, -1, 0,
 0, -1, 0, 0, 0, 0, 0, -1, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, -1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, -1, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, -1, 0, -1, 0, 0, 0,
 0, -1, 0, 0, 0, 0, -1, 0, 0, 0, 0, 0, -1, 0, -1, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, -1, 0, 0, 0, 0, -1, -1, 0,
 0, 0, 0, 0, 0, -1, 0, -1, 0, 0, 0, 0, -1, 0, -1, 0, 0,
 0, 0, 0, 0, 0, 0, 0, -1, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, -1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, -1, 0, -1, -1,
 0, 0, 0, -1, -1, 0, 0, 0, 0, 0, -1, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, -1, 0, 0, 0, 0, -1, 0, -1, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, -1, 0,
 0, 0, 0, -1, 0, 0, -1, 0, 0, 0, 0, 0, 0, 0, -1, 0, 0, 0,
 0, 0, -1, 0, 0, 0, 0, 0, -1, 0, -1, 0, 0, 0, 0, -1, 0,
 0, 0, 0, -1, 0, 0, 0, 0, 0, 0, 0, 0, -1, 0, 0, 0, 0,
-1, -1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, -1, 0,
 0, 0, -1, 0, 0, 0, 0, 0, -1, 0, -1, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, -1, 0, 0, 0,
 0, -1, 0, -1, 0, -1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
-1, 0, -1, 0, 0, -1, -1, 0, 0, 0, 0, -1, 0, -1, 0, 0, 0],
dtype=int64)

```
In [11]: cl=pd.DataFrame(dbscan.labels_,columns=["cluster"])
cl
```

Out[11]:

	cluster
0	0
1	0
2	-1
3	0
4	-1
...	...
435	-1
436	0
437	-1
438	0
439	0

440 rows × 1 columns

```
In [12]: pd.concat([data,cl],axis=1)
```

Out[12]:

	Fresh	Milk	Grocery	Frozen	Detergents_Paper	Delicassen	cluster
0	12669	9656	7561	214	2674	1338	0
1	7057	9810	9568	1762	3293	1776	0
2	6353	8808	7684	2405	3516	7844	-1
3	13265	1196	4221	6404	507	1788	0
4	22615	5410	7198	3915	1777	5185	-1
...
435	29703	12051	16027	13135	182	2204	-1
436	39228	1431	764	4510	93	2346	0
437	14531	15488	30243	437	14841	1867	-1
438	10290	1981	2232	1038	168	2125	0
439	2787	1698	2510	65	477	52	0

440 rows × 7 columns