6b-Hierarchical.Clustering

November 13, 2024

1 Hierachical Clustering.

```
[1]: import pandas as pd
     import numpy as np
     import matplotlib.pyplot as plt
     import seaborn as sns
     from sklearn.preprocessing import MinMaxScaler
     import scipy.cluster.hierarchy as shc
```

Load the data.

```
[2]: url = "https://ddc-datascience.s3.amazonaws.com/Wholesale_Data.csv"
     data = pd.read_csv( url )
     data.head()
```

```
[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.shape
```

```
[3]: (440, 8)
```

```
[4]: data.describe().transpose()
```

```
[4]:                   count          mean           std   min      25%     50%  \
     Channel           440.0      1.322727      0.468052   1.0     1.00     1.0
     Region            440.0      2.543182      0.774272   1.0     2.00     3.0
     Fresh             440.0  12000.297727  12647.328865   3.0  3127.75  8504.0
     Milk              440.0   5796.265909   7380.377175  55.0  1533.00  3627.0
     Grocery           440.0   7951.277273   9503.162829   3.0  2153.00  4755.5
     Frozen            440.0   3071.931818   4854.673333  25.0   742.25  1526.0
     Detergents_Paper  440.0   2881.493182   4767.854448   3.0   256.75   816.5
     Delicassen        440.0   1524.870455   2820.105937   3.0   408.25   965.5
```

```
                       75%       max
Channel               2.00       2.0
Region                3.00       3.0
Fresh             16933.75  112151.0
Milk               7190.25   73498.0
Grocery           10655.75   92780.0
Frozen             3554.25   60869.0
Detergents_Paper   3922.00   40827.0
Delicassen         1820.25   47943.0
```

[5]:

```
data.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
```

[6]:

```
data[["Channel", "Region"]].nunique()
```

[6]:

```
Channel    2
Region     3
dtype: int64
```

[8]:

```
data[["Channel", "Region"]].value_counts( )
```

[8]:

```
Channel  Region
1        3         211
2        3         105
1        1          59
         2          28
2        2          19
         1          18
Name: count, dtype: int64
```

Scale the data.

[9]:

```
# Scale data
scaler = MinMaxScaler()
scaler.fit(data)
data_scaled = scaler.transform(data)
# Convert back to data frame
data_scaled = pd.DataFrame(data_scaled, columns = data.columns)
data_scaled.head()
```

[9]:

```
   Channel  Region     Fresh      Milk   Grocery    Frozen  Detergents_Paper  \
0      1.0     1.0  0.112940  0.130727  0.081464  0.003106          0.065427
1      1.0     1.0  0.062899  0.132824  0.103097  0.028548          0.080590
2      1.0     1.0  0.056622  0.119181  0.082790  0.039116          0.086052
3      0.0     1.0  0.118254  0.015536  0.045464  0.104842          0.012346
4      1.0     1.0  0.201626  0.072914  0.077552  0.063934          0.043455

   Delicassen
0    0.027847
1    0.036984
2    0.163559
3    0.037234
4    0.108093
```

[10]:

```
data_scaled.describe().transpose()
```

[10]:

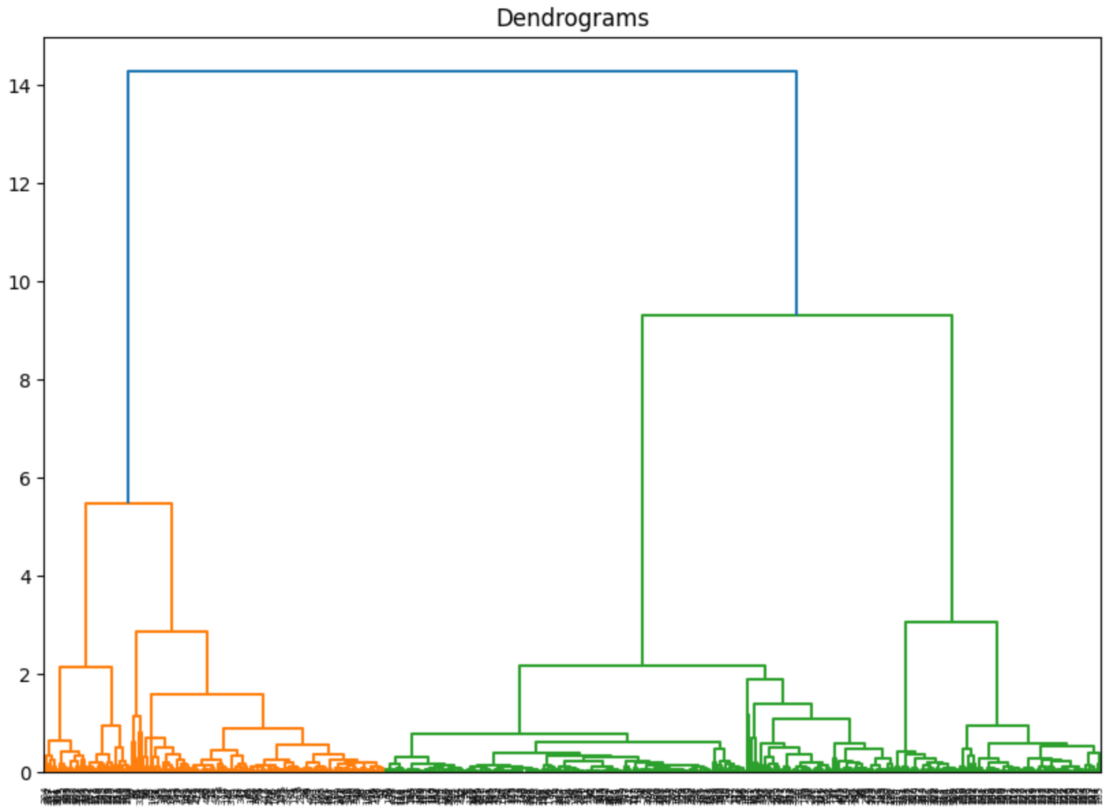
```
                  count      mean       std  min       25%       50%  \
Channel           440.0  0.322727  0.468052  0.0  0.000000  0.000000
Region            440.0  0.771591  0.387136  0.0  0.500000  1.000000
Fresh             440.0  0.106977  0.112774  0.0  0.027863  0.075802
Milk              440.0  0.078173  0.100491  0.0  0.020124  0.048636
Grocery           440.0  0.085671  0.102430  0.0  0.023174  0.051225
Frozen            440.0  0.050078  0.079789  0.0  0.011788  0.024670
Detergents_Paper  440.0  0.070510  0.116790  0.0  0.006216  0.019927
Delicassen        440.0  0.031745  0.058826  0.0  0.008453  0.020077

                       75%  max
Channel           1.000000  1.0
Region            1.000000  1.0
Fresh             0.150968  1.0
Milk              0.097154  1.0
Grocery           0.114821  1.0
Frozen            0.058005  1.0
Detergents_Paper  0.095997  1.0
Delicassen        0.037907  1.0
```

Create dendogram of the clustering.

[11]:

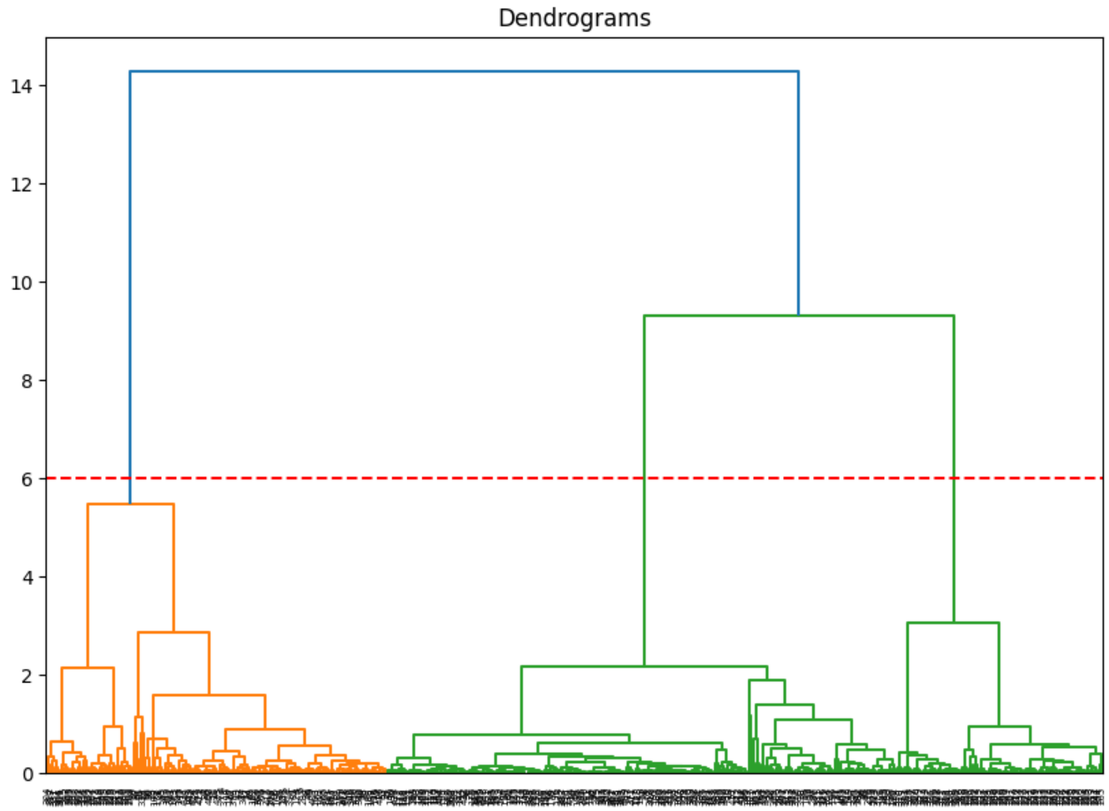
```
plt.figure(figsize=(10, 7))
plt.title("Dendrograms")
```

```
dend = shc.dendrogram(shc.linkage(data_scaled, method='ward'))
```



Choose an appropriate threshold for the clusters.

```
[12]: plt.figure(figsize=(10, 7))
      plt.title("Dendrograms")
      dend = shc.dendrogram(shc.linkage(data_scaled, method='ward'))
      plt.axhline(y=6, color='r', linestyle='--') ;
```



Classify all points based on the number of clusters at the threshold you chose.

```
[13]: from sklearn.cluster import AgglomerativeClustering
      cluster = AgglomerativeClustering(n_clusters=3, metric='euclidean',␣
       ↪linkage='ward')
      clusterNums = pd.Series(cluster.fit_predict(data_scaled))
```

```
[14]: clusterNums
```

```
[14]: 0      0
      1      0
      2      0
      3      2
      4      0
            ..
      435    2
      436    2
      437    0
      438    2
      439    2
      Length: 440, dtype: int64
```

```
[15]: clusterNums.value_counts()
```

```
[15]: 2    212
      0    142
      1     86
      Name: count, dtype: int64
```

Visualize the clusters.

```
[16]: data.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
```

```
[17]: cluster.labels_
```

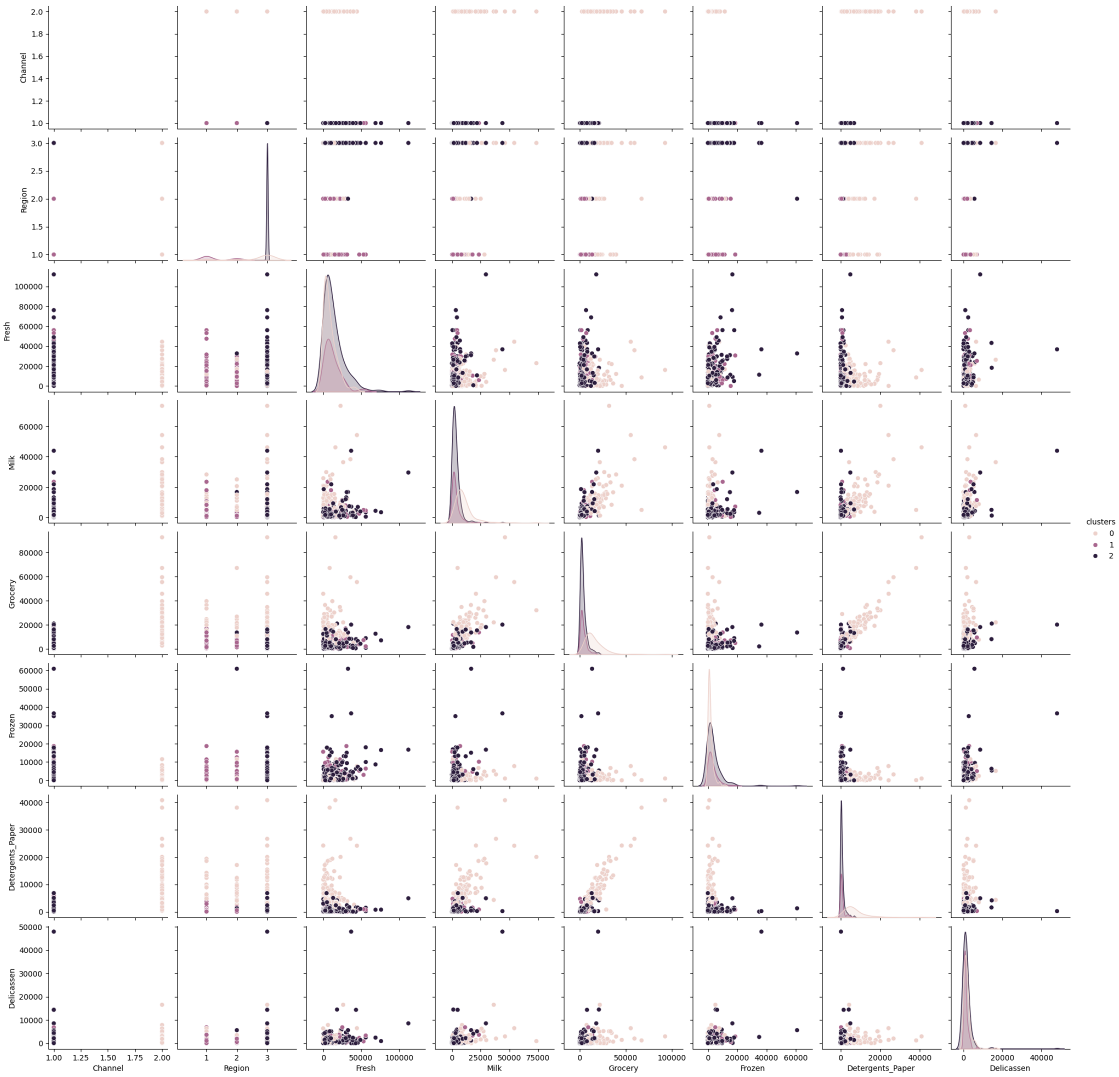
```
[17]: array([0, 0, 0, 2, 0, 0, 0, 0, 2, 0, 0, 0, 0, 0, 0, 2, 0, 2, 0, 2, 0, 2,
             2, 0, 0, 0, 2, 2, 0, 2, 2, 2, 2, 2, 2, 0, 2, 0, 0, 2, 2, 2, 0, 0,
             0, 0, 0, 0, 0, 0, 2, 2, 0, 0, 2, 2, 0, 0, 2, 2, 0, 0, 0, 0, 2, 0,
             2, 0, 2, 2, 2, 2, 2, 0, 0, 2, 2, 0, 2, 2, 2, 0, 0, 2, 0, 0, 0, 2,
             2, 2, 2, 2, 0, 2, 0, 2, 0, 2, 2, 2, 0, 0, 0, 2, 2, 2, 0, 0, 0, 0,
             2, 0, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 0, 2, 2, 2, 0, 2, 2, 2, 2,
             2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 0, 2, 2, 2, 2, 2, 2, 2, 2,
             2, 0, 0, 2, 0, 0, 0, 2, 2, 0, 0, 0, 0, 2, 2, 2, 0, 0, 2, 0, 2, 0,
             2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 0, 0, 2, 2, 2, 0, 2, 2, 1, 0,
             1, 1, 0, 0, 1, 1, 1, 0, 1, 0, 1, 0, 1, 0, 1, 1, 0, 1, 0, 1, 0, 1,
             1, 1, 1, 0, 1, 1, 0, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
             1, 1, 1, 0, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
             0, 1, 0, 1, 0, 1, 1, 1, 1, 2, 2, 2, 2, 2, 2, 0, 2, 0, 2, 2, 2, 2,
             2, 2, 2, 2, 2, 2, 2, 0, 1, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1,
             1, 0, 1, 1, 0, 1, 1, 0, 1, 1, 1, 0, 1, 1, 1, 1, 1, 2, 1, 1, 1, 1,
             1, 0, 1, 0, 0, 0, 1, 1, 1, 1, 0, 0, 2, 0, 2, 2, 0, 0, 2, 0, 2, 0,
             2, 0, 2, 2, 2, 0, 2, 2, 2, 2, 2, 2, 2, 0, 2, 2, 2, 2, 0, 2, 2, 0,
             2, 2, 0, 2, 2, 0, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
             0, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 0, 0, 2, 2, 2, 2, 2, 2, 0, 0, 2,
       0, 2, 2, 0, 2, 0, 0, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 0, 2, 2])
```

```
[18]: data['clusters'] = cluster.labels_
```

```
[19]: data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 440 entries, 0 to 439
Data columns (total 9 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
 8   clusters          440 non-null    int64
dtypes: int64(9)
memory usage: 31.1 KB
```

```
[20]: sns.pairplot(data, hue = 'clusters') ;
```


Channel
Region
Fresh
Milk
Grocery
Frozen
Detergents_Paper
Delicassen
1.0
1.2
1.4
1.6
1.8
2.0
1.5
2.5
3.0
0
20000
40000
60000
80000
100000
1.00
1.25
1.50
1.75
2.00
1
2
3
50000
100000
25000
50000
75000
30000
10000
clusters
0
1
2

[ ]: