6b-Hierarchical.Clustering

November 8, 2024

1 Hierachical Clustering.

[ ]:
```
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.

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

[ ]:
```
data.shape
```

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

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

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

Scale the data.

[ ]:
```
# 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()
```

[ ]:
```
data_scaled.describe().transpose()
```

Create dendogram of the clustering.

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

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

Choose an appropriate threshold for the clusters.

[ ]:
```
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.

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

[ ]:
```
clusterNums
```

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

Visualize the clusters.

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

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

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

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

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

[ ]:
```

```