Python Program

CHAPTER 6: CLUSTER ANALYSIS

Chapter Objectives

In this chapter, we will:

- Explore Cluster Analysis
- Compare two algorithms
  - K-Means
  - Hierarchical

Chapter Concepts

Cluster Analysis

- Analysis tool to help make sense of the data before feeding it into other models
- Unsupervised
  - More about discovering patterns in data
  - Not about predicting values for unknown values
- Looks for natural groupings among the data
  - Voter groups (is it just left vs. right, or left, right, center, or more)
  - Species identification (are two groups of organisms different enough to be considered a different species or not)
  - Identify different types of customers we may have
- Often helpful as a preparatory step before classification to determine how many categories we may want to predict

Types of Cluster Analysis

- There are two main approaches to solve this
  - Top down (K-Means)
  - Bottom up (Hierarchical clustering)
- Both rely on the notion of similarity
  - Objects are similar if they share common attributes to others
  - The more similar they are, the closer they are to one another
  - If something is far away in similarity to one thing, it may be closer to something else
- Ultimately the goal is to take a large sample of data and break it up into a small number of meaningful groupings that shed insight as to what the data means

Dataset

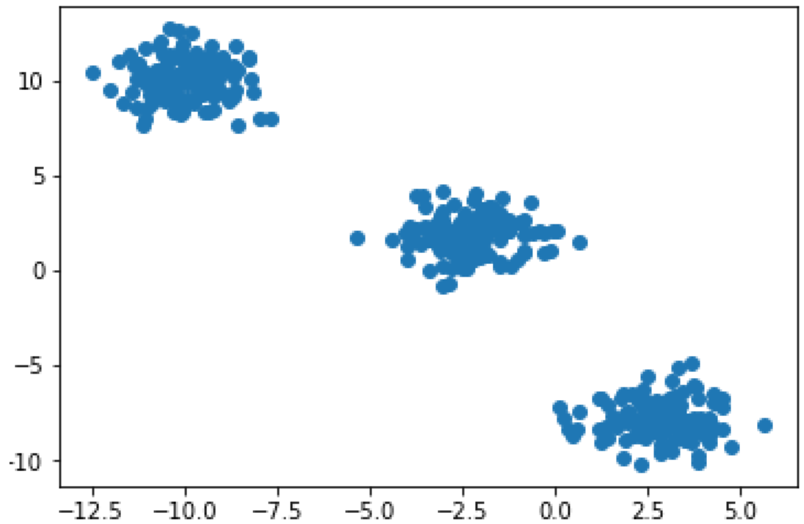
- For these examples let's generate some random datasets just because it's easier to analyze

```
import numpy as np
from sklearn.cluster import KMeans
from sklearn.datasets import make_blobs
# Creating a sample dataset with 3 clusters
x, y = make_blobs(n_samples=400, n_features=2, centers=3)
print (x[:5]) # shape location
print (y[:5]) # cluster member

[[-6.10513999 -3.58316594] [-7.6168443   5.40841142] [-
2.06235753 -3.92038777] [-1.8104498  -4.1218467 ] [-5.32915489
-6.17092626]]

[2 1 0 0 2]
```

Visualize the Data

- It is often helpful to visualize the data by plotting it
  - There are only two features in this set so it's easy to plot
  - You can also plot a 3D graph for three features
  - Beyond that, it's hard to visualize more features

```
import matplotlib.pyplot as plt
plt.rcParams['figure.figsize'] = (16, 9)
plt.plot(x[:,0],x[:,1],'o')
plt.show()
```



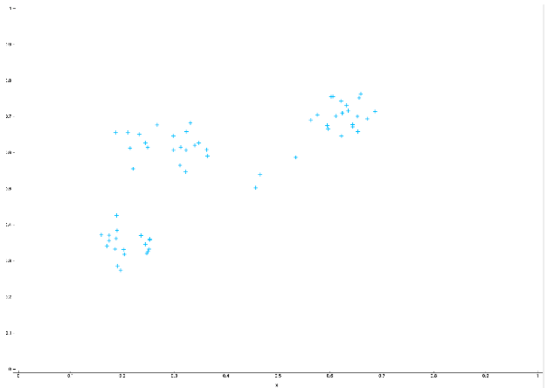
Chapter Concepts

Cluster Analysis

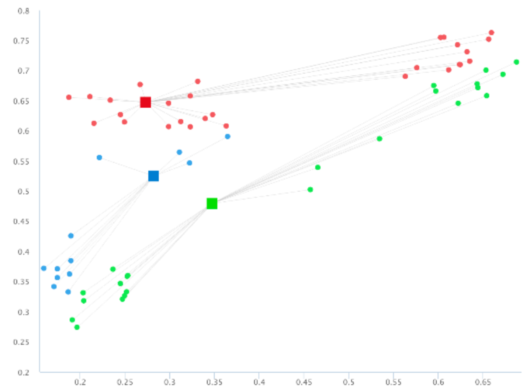
Algorithms

Chapter Summary

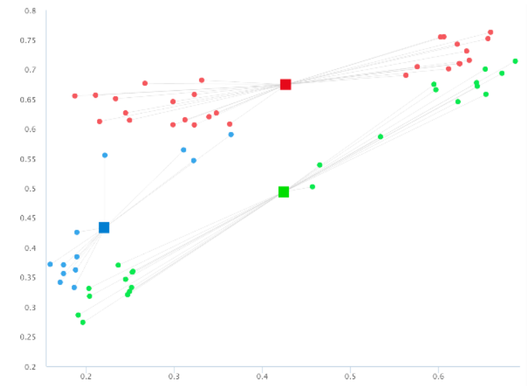
K-Means in Actions



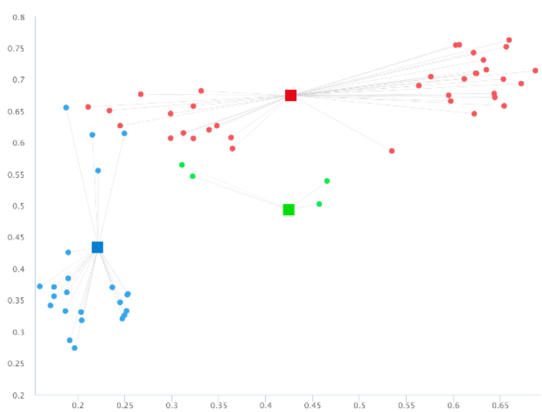
Random Data



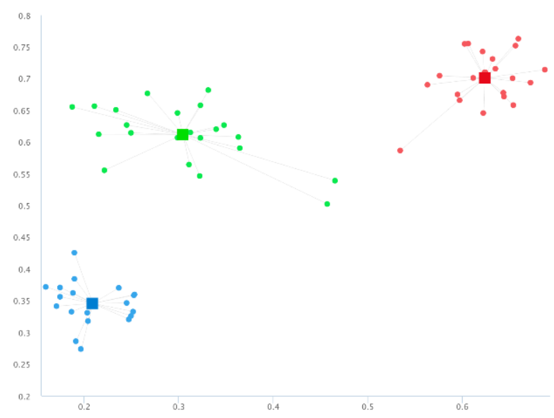
Random Centroids



Adjust Centroids



Reassign Membership

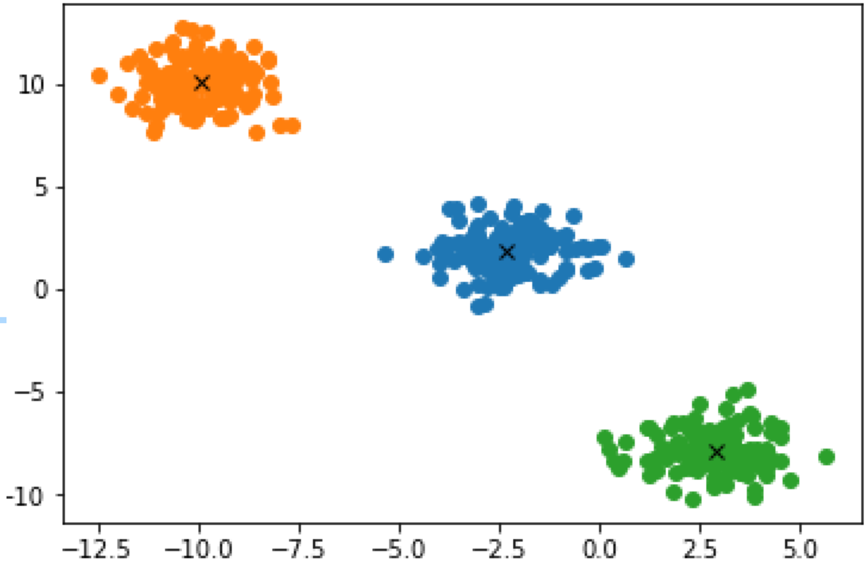


Keep Doing Until Stops Changing

Run K-Means

- Just eyeballing it, let's try out three clusters and plot the results

```
from sklearn import cluster
CLUSTERS = 3
k_means = cluster.KMeans(n_clusters=CLUSTERS)
k_means.fit(x)
labels = k_means.labels_
centroids = k_means.cluster_centers_
for i in range(CLUSTERS):
   ds = x[np.where(labels==i)]
   plt.plot(ds[:,0],ds[:,1],'o')
   lines = plt.plot(centroids[i,0]
            centroids[i,1],'kx')
plt.show()
```



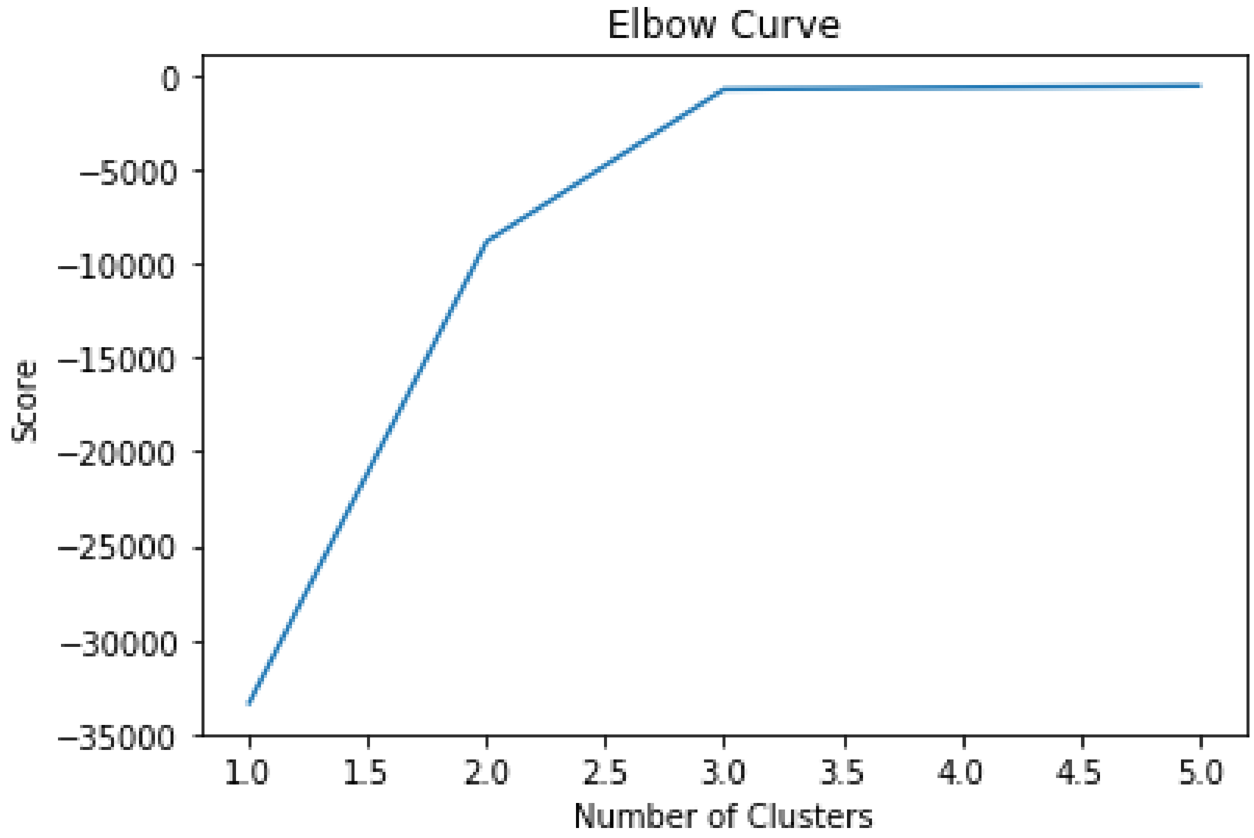
Elbow Chart

- Here the results are very clear cut, but sometimes the data overlap and don't fit nicely into a particular cluster
- It is often helpful to run a chart that helps figure out how many clusters is ideal
  - Too few and the items are too dissimilar
  - Too many and the additional distinctions become trivial
    - Is there much difference between a brown poodle and a chocolate poodle?

```
def plot_elbow(data, cluster_cnt = 6):
   CLUSTERS = range(1, cluster_cnt)
   kmeans = [cluster.KMeans(n_clusters=i) for i in CLUSTERS]
   score = [kmeans[i].fit(data).score(data) \
            for i in range(len(kmeans))]
   plt.plot(CLUSTERS ,score)
   plt.xlabel('Number of Clusters')
   plt.ylabel('Score')
   plt.title('Elbow Curve')
   plt.show()
plot_elbow(x)
```

Elbow Chart (continued)

- In the chart, we can see there is a bend between two to four clusters
- Three feels like the right number to start with in this case

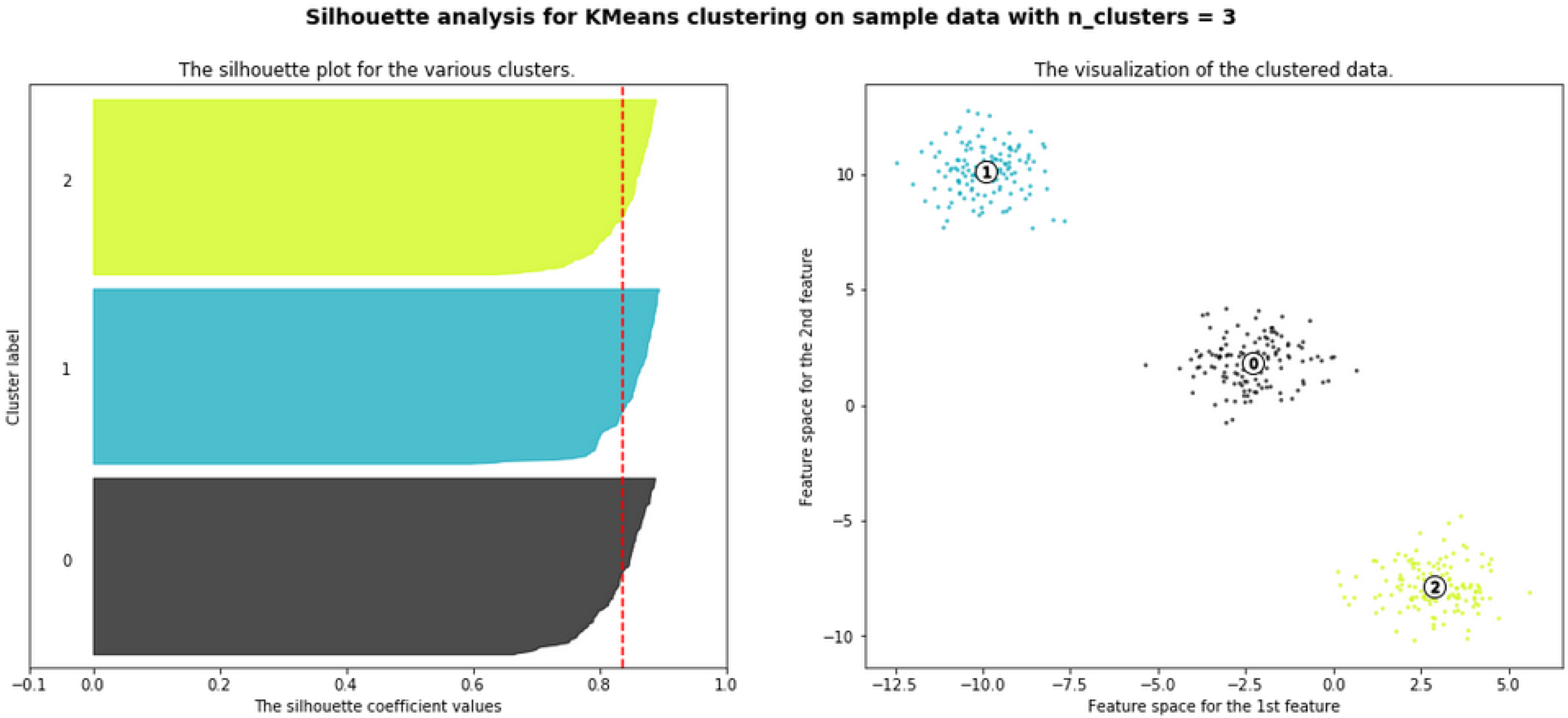


Silhouette Charts

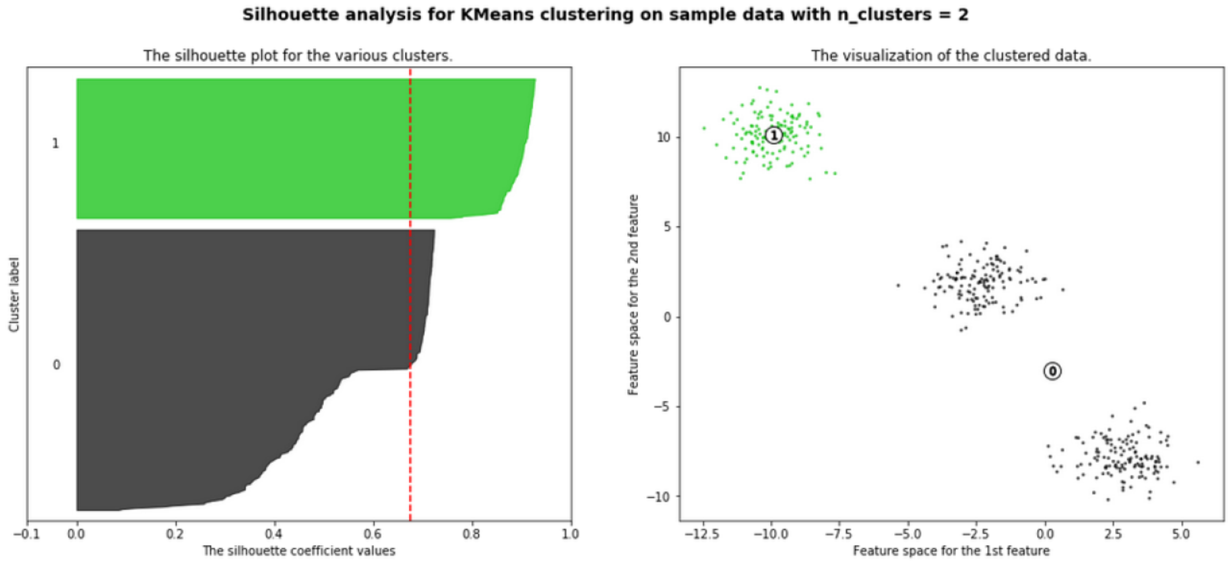
- Once you have figured out approximately how many clusters you have, you should run the analysis a few times with different cluster numbers
- A silhouette chart helps to visualize how well the clusters are at grouping similar items together
- Higher silhouette score (i.e., closer to 1) means in general the cluster does a good job at grouping similar items together
- Graphing how similar each item is to its neighbors helps to visualize how good the cluster is also
- Ideally, you want to settle upon a number of clusters that has a good mix of:
  - A high silhouette value
  - Few members that are far off from the average silhouette value
  - A number of clusters that are reasonably similar in size
  - A number of clusters that makes business sense of what you're trying to describe

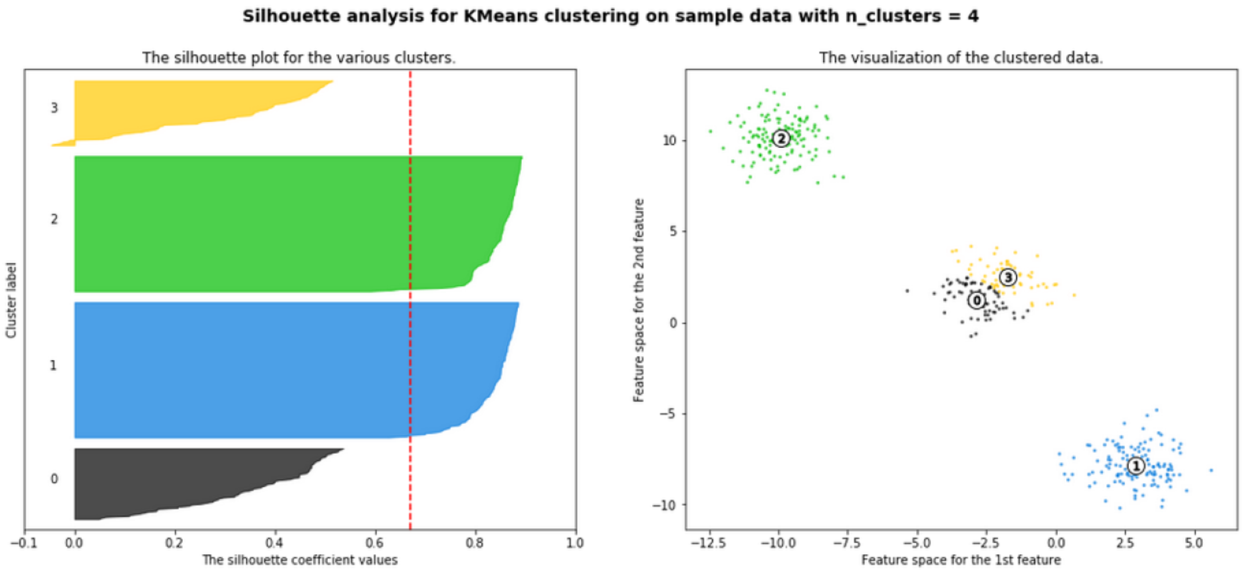
Silhouette Charts (continued)

```
For n_clusters = 2 The average silhouette_score is : 0.6756049213871368
For n_clusters = 3 The average silhouette_score is : 0.8378250424949772
For n_clusters = 4 The average silhouette_score is : 0.6699001879846088
For n_clusters = 5 The average silhouette_score is : 0.5071441264659202
For n_clusters = 6 The average silhouette_score is : 0.3347353201539845
```



Silhouette Charts (continued)



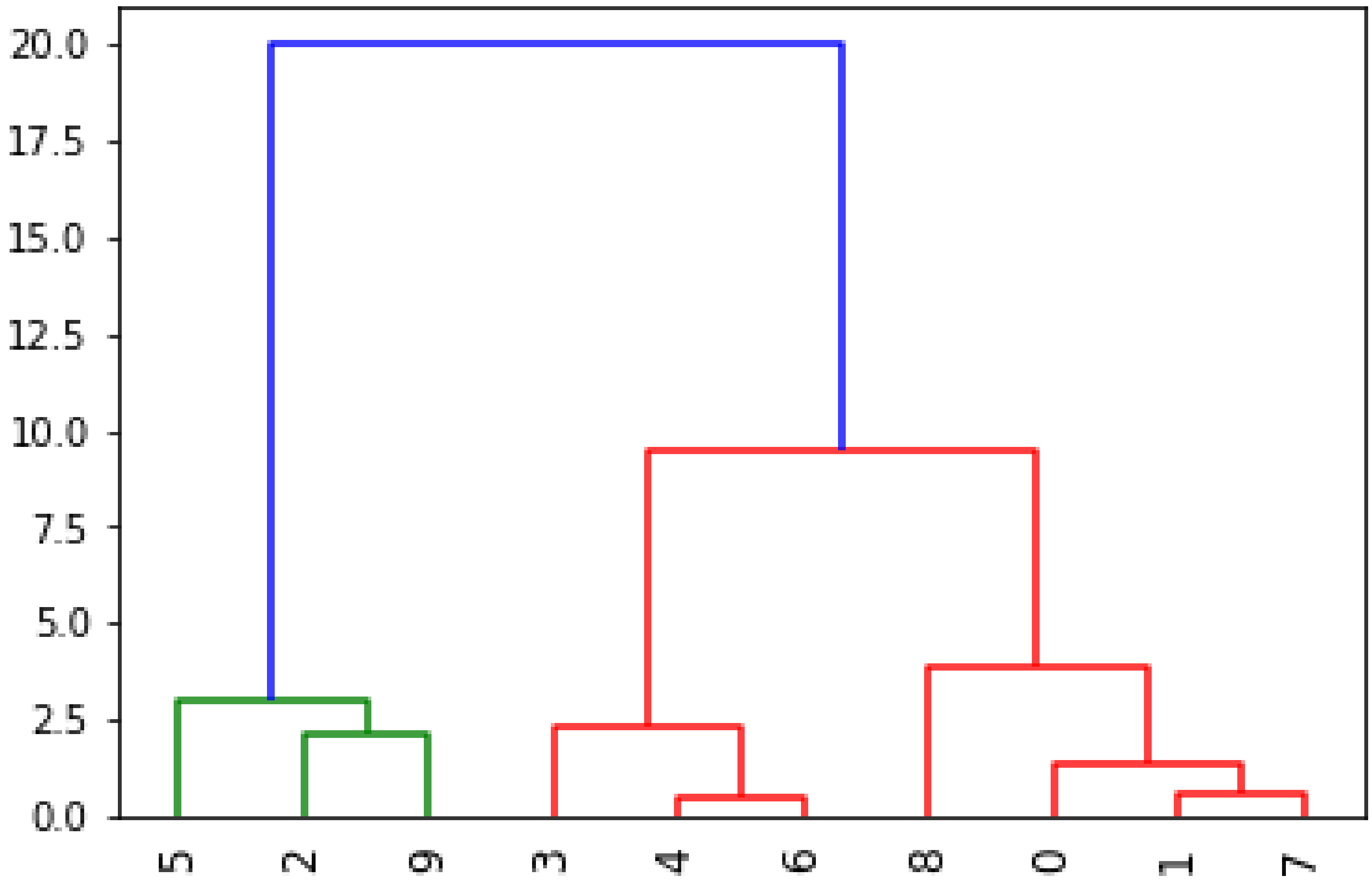


Hierarchical Clustering

- Often called bottom-up
- Finds two clusters closest to one another and merges them and keeps doing it until there is one big cluster
  - Uses distance of the features to determine closeness
- Creates a graph called a dendrogram which helps visualize the clusters and how similar they are
- Usually a good first step to take before K-Means to get a feel for how many clusters you should start with

```
x, y = make_blobs(n_samples=10, n_features=2, centers=3)
print (x)
print (y)
from scipy.cluster.hierarchy import dendrogram, linkage
z = linkage(x, 'ward')
dendrogram(z, leaf_rotation = 90, leaf_font_size=12)
```

Hierarchical Clustering (continued)



Chapter Concepts

Cluster Analysis

Algorithms

Chapter Summary

Next Steps

- The unsupervised model of clustering doesn’t make predictions so much as it helps understand the data
- Another unsupervised model to explore is association rules
  - Used to describe patterns like "people who like X also like Y"

Chapter Summary

In this chapter, we have:

- Explored Cluster Analysis
- Compared two algorithms
  - K-Means
  - Hierarchical