บทที่ 10 Clustering

หัวข้อหลัก

- การจัดกลุ่ม (Clustering) เป็นการเรียนรู้แบบไม่มีผู้สอน (unsupervised learning) ที่นิยมใช้สำหรับการวิเคราะห์ข้อมูลเชิงสำรวจ (exploratory data analysis)
- Hierarchical Cluster Analysis เป็นอัลกอริทึมการจัดกลุ่มแบบ bottom-up ที่เหมาะกับกรณีที่เราไม่ทราบจำนวนคลัสเตอร์ที่เหมาะสม

การจัดกลุ่มเป็นการเรียนรู้แบบไม่มีผู้สอน (unsupervised learning) ที่นิยมใช้สำหรับการวิเคราะห์ข้อมูลเชิงสำรวจ (exploratory analysis) เพื่อค้นหารูปแบบแฝงในชุดข้อมูลที่ไม่มีป้ายชื่อ (unlabeled data) อัลกอริทึมการจัดกลุ่ม (clustering algorthm) จะจัดจุดข้อมูลที่มีความคล้ายคลึงกันมากไว้ในกลุ่มเดียวกัน เพื่อให้จุดข้อมูลที่อยู่ในกลุ่มเดียวกันมีความเป็นเนื้อเดียวกันสูง (high intra-group homogeneity) และจุดข้อมูลที่อยู่คนละกลุ่มกันมีความต่างกันสูง (high inter-group heterogeneity) ตัวอย่างการประยุกต์ใช้งาน clustering algorithm เช่น การแบ่งกลุ่มลูกค้า (customer segmentation), การวิเคราะห์ข้อมูล (data analysis), การลดมิติของข้อมูล (dimensionality reduction), การตรวจจับข้อมูลผิดปกติ (outlier detection), การแบ่งส่วนรูปภาพ (image segmentation), การค้นหาข้อมูล (search engine) ในบทนี้ เราจะศึกษาเทคนิคการจัดกลุ่มข้อมูลที่เรียกว่า Hierachical Cluster Analysis (HCA)

10.1 Hierarchical Cluster Analysis (HCA)

Hierarchical Cluster Analysis (HCA) เป็นอัลกอริทึมการจัดกลุ่มข้อมูลที่ทำงานแบบ bottom-up, HCA เหมาะกับกรณีที่ผู้ใช้งานไม่ทราบหรือไม่สามารถประมาณค่าจำนวนคลัสเตอร์ที่ต้องการสร้างได้ หลักการทำงานของ HCA คือการผนวกรวมคลัสเตอร์ของจุดข้อมูลที่มีความเหมือนกันเป็นคลัสเตอร์เดียวกัน โดยเริ่มต้นจากคลัสเตอร์ที่ประกอบด้วยจุดข้อมูลเพียงจุดเดียวและดำเนินการรวมตัวคลัสเตอร์ที่เหมือนกันที่สุดเข้าด้วยกัน จนกระทั่งเหลือพียงคลัสเตอร์เดียว ความเหมือนหรือความคล้ายคลึงกันระหว่างจุดข้อมูลสามารถวัดค่าเป็นตัวเลขได้โดยใช้ Euclidean distance ดังสมการ

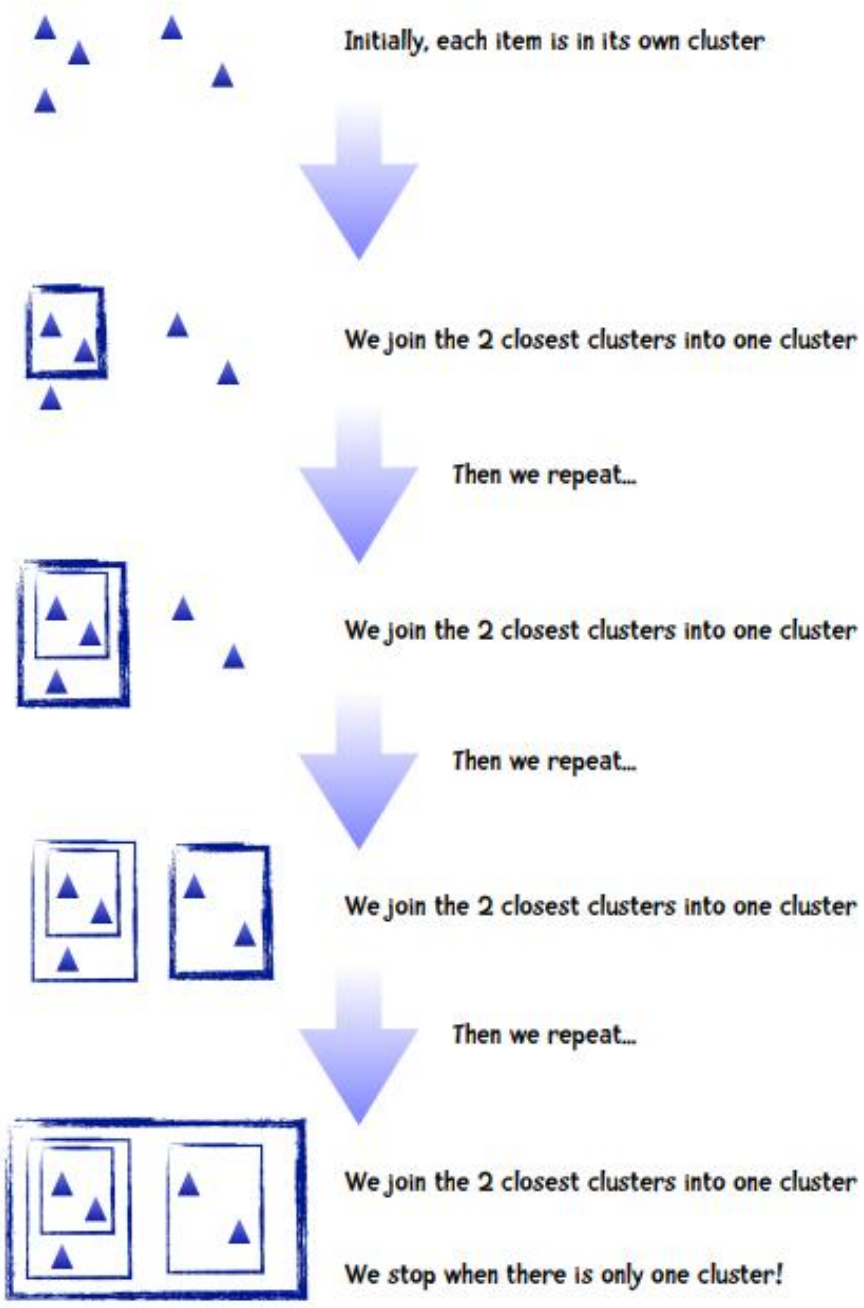
$$dist(x, y) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}$$

เมื่อ $x = (x_1, x_2, ..., x_n)$ และ $y = (y_1, y_2, ..., y_n)$ คือ ฟีเจอร์เวคเตอร์ของจุดข้อมูลสองจุด โดยจุดข้อมูลที่เหมือนกันจะมีค่า Euclidean distance น้อย ส่วนจุดข้อมูลที่ต่างกันมากจะมีค่า Euclidean distance มาก จากสูตรคำนวณความคล้ายคลึงกันระหว่างจุดข้อมูลเราสามารถคำนวณความคล้ายคลึงกันของคลัสเตอร์สองคลัสเตอร์ได้ 3 วิธีดังนี้คือ

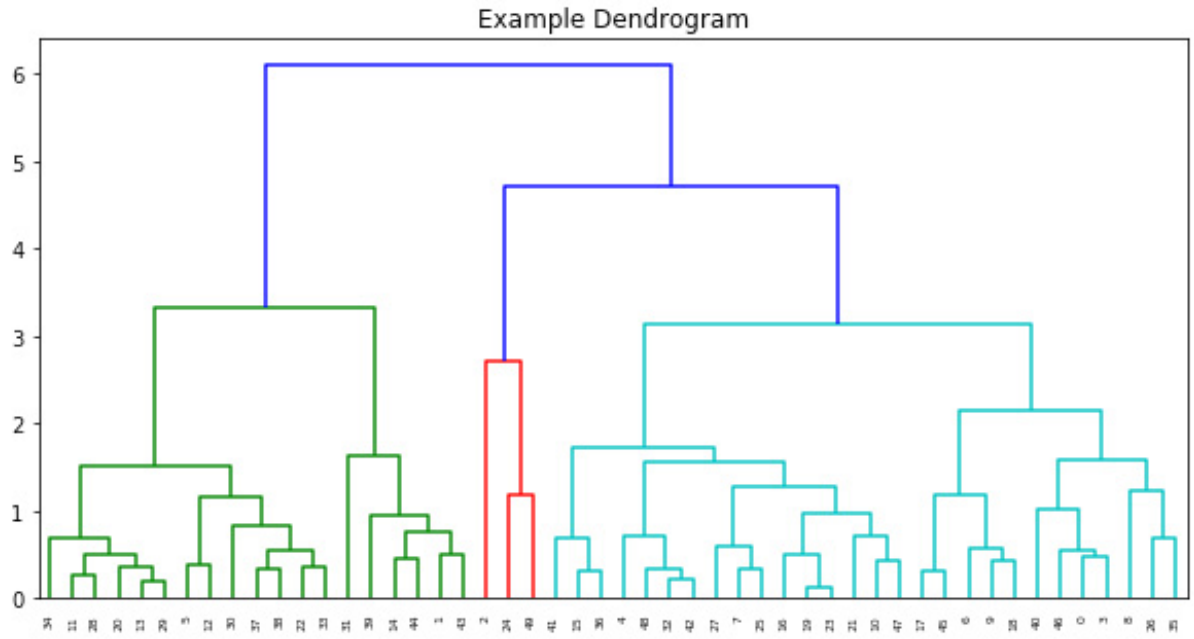
1. **Single-linkage Clustering** ความคล้ายคลึงระหว่างคลัสเตอร์คำนวณได้จากระยะทางที่ใกล้ที่สุดจากสมาชิกของคลัสเตอร์หนึ่งไปยังสมาชิกของอีกคลัสเตอร์หนึ่ง
2. **Complete-linkage Clustering** ความคล้ายคลึงระหว่างคลัสเตอร์คำนวณได้จากระยะทางที่ไกลที่สุดจากสมาชิกของคลัสเตอร์หนึ่งไปยังสมาชิกของอีกคลัสเตอร์หนึ่ง

3. **Average-linkage Clustering** ความคล้ายคลึงระหว่างคลัสเตอร์คำนวณได้จากค่าเฉลี่ยของระยะทางระหว่างสมาชิกของคลัสเตอร์หนึ่งไปยังสมาชิกของอีกคลัสเตอร์หนึ่ง

กระบวนการทำงานของ HCA แสดงดังในรูปที่ 1 และความสัมพันธ์ระหว่างคลัสเตอร์แต่ละคลัสเตอร์ที่สร้างขึ้นโดย HCA สามารถแสดงได้โดยใช้แผนภูมิต้นไม้ dendrogram ซึ่งมีลักษณะเป็นโครงสร้างเหมือนต้นไม้ ดังแสดงตัวอย่างในรูปที่ 2



รูปที่ 1: กระบวนการทำงานของ Hierarchical Clustering Analysis (HCA)



รูปที่ 2: ตัวอย่าง Dendrogram ที่ได้จากกระบวนการ Hierarchical Clustering Analysis
ตัวเลขบนแกนนอน คือ identifier ของจุดข้อมูล ส่วนตัวเลขบนแกนตั้ง คือค่าระดับความลึกของต้นไม้

10.2 ตัวอย่างการจัดกลุ่มข้อมูลโดยใช้ HCA โดยใช้ไลบารี่ scipy

กำหนดชุดข้อมูลพันธุ์สุนัข ดังตารางที่ 1 จงจัดกลุ่มสายพันธุ์สุนัขโดยใช้อัลกอริทึม HCA

ตารางที่ 1: ชุดข้อมูลพันธุ์สุนัข

สายพันธุ์	ความสูง (นิ้ว)	น้ำหนัก (ปอนด์)
Border Colle	20	45
Boston Terrier	16	20
Brittany Spaniel	18	35
Bullmastiff	27	120
Chihuahua	8	8
German Shepherd	25	78
Golden Retriever	23	70
Great Dane	32	160
Portuguese Water Dog	21	50
Standard Poodle	19	65
Yorkshire Terrier	6	7

1. สร้างดาต้าเฟรมเพื่อเก็บข้อมูลสายพันธุ์สุนัขและแปลงค่าน้ำหนักและความสูงให้อยู่ในช่วงค่ามาตรฐาน

```
import pandas as pd

dogs = pd.DataFrame({
            "Breed" : ["Border Colle", "Boston Terrier", "Brittany Spaniel",
                       "Bullmastiff", "Chihuahua", "German Shepherd",
                       "Golden Retriever", "Great Dane", "Portuguese Water Dog",
                       "Standard Poodle", "Yorkshire Terrier"],
            "Height": [20, 16, 18, 27, 8, 25, 23, 32, 21, 19, 6],
            "Weight": [45, 20, 35, 120, 8, 78, 70, 160, 50, 65, 7]
})

dogs.set_index("Breed", inplace=True)

dogs.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 11 entries, Border Colle to Yorkshire Terrier
Data columns (total 2 columns):
Height    11 non-null int64
Weight    11 non-null int64
dtypes: int64(2)
memory usage: 264.0+ bytes
```

```
from sklearn.preprocessing import StandardScaler

dogs_std = StandardScaler().fit_transform(dogs)

from scipy.stats import zscore

dogs_standardized = dogs.apply(zscore)

dogs_standardized
```

Breed	Height	Weight
Border Colle	0.062238	-0.330507
Boston Terrier	-0.485456	-0.888111
Brittany Spaniel	-0.211609	-0.553549
Bullmastiff	1.020703	1.342305
Chihuahua	-1.580845	-1.155761
German Shepherd	0.746856	0.405530
Golden Retriever	0.473009	0.227097
Great Dane	1.705321	2.234471
Portuguese Water Dog	0.199162	-0.218986
Standard Poodle	-0.074686	0.115576
Yorkshire Terrier	-1.854692	-1.178065

2. จัดกลุ่มโดยใช้ HCA แบบ Single-Linkage Clustering

```
from scipy.cluster.hierarchy import linkage

single_linkage_model = linkage(dogs_standardized, method='single')
```

3. แสดง dendrogram ของโมเดลที่ได้

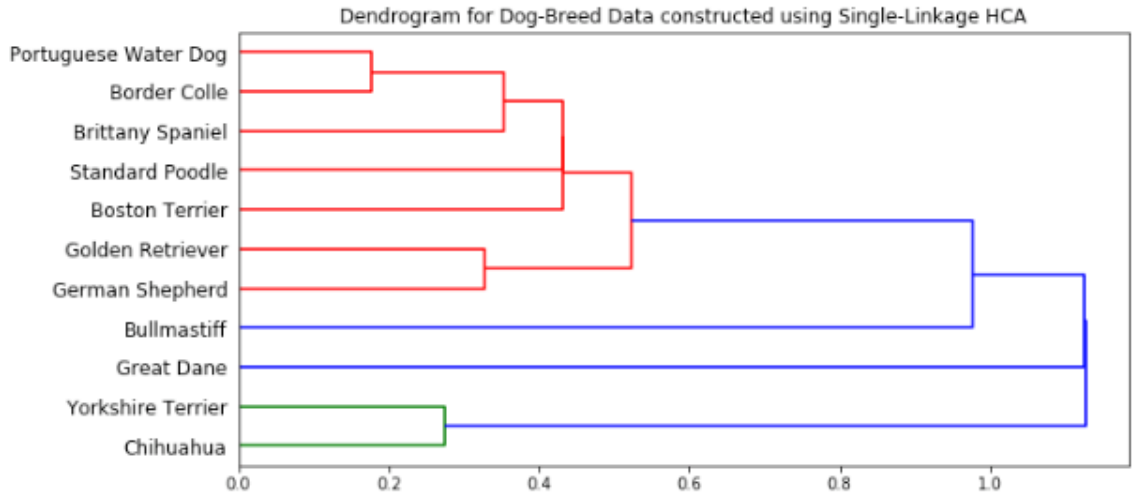
```
import matplotlib.pyplot as plt
%matplotlib inline

from scipy.cluster.hierarchy import dendrogram

plt.figure(figsize=(10,5))

plt.subplot(111)
plt.title("Dendrogram for Dog-Breed Data constructed using Single-Linkage HCA")
dendrogram(single_linkage_model, orientation='right', leaf_font_size=12,
           labels=dogs_standardized.index)

plt.show()
```



10.3 ตัวอย่างการจัดกลุ่มข้อมูลโดยใช้ HCA โดยไม่ใช้ไลบารี่ scipy

เพื่อให้เข้าใจกลไกการทำงานของ Single-linkage HCA และตรวจสอบผลลัพธ์การจัดกลุ่มที่ได้จากการใช้ไลบารี่ scipy ข้างต้น ในลำดับถัดไปเราจะทดลองสร้าง dendrogram โดยใช้ Single-linkage HCA โดยไม่ใช้ไลบารี่ scipy แต่จะทำการคำนวณหาค่า Euclidean distance และสร้างคลัสเตอร์เองทีละขั้นตอน

1. ขั้นตอนแรกเราจะคำนวณหาระยะทางระหว่างจุดข้อมูลแต่ละจุด โดยใช้เมธอด scipy.spatial.distance.euclidean

```
from scipy.spatial.distance import euclidean

distances = {}
M = {}
done = []
for breed in dogs_standardized.index:
    M[breed] = {}
    done.append(breed)
    for other_breed in dogs_standardized.index:
        if other_breed in done: continue
        M[breed][other_breed] = euclidean(dogs_standardized.loc[breed],
                                          dogs_standardized.loc[other_breed])
        distances[breed+":"+other_breed] = M[breed][other_breed]
```

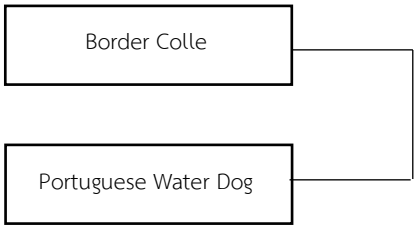
2. จากนั้นทำการเรียงลำดับข้อมูลตามระยะทาง Euclidean distance จากน้อยไปหามาก

```
sorted_distances = sorted(distances.items(), key=lambda kv: (kv[1], kv[0]))
i = 1
for dogs, dist in sorted_distances:
    d1, d2 = dogs.split(':')
    print("{}. {:20} => {:20} : {:.5}".format(i, d1, d2, dist))
    i = i + 1
```

ซึ่งจะได้ผลลัพธ์ดังรูปที่ 3 จากข้อมูลระยะทางที่ได้ เราสามารถสร้างคลัสเตอร์ได้ ดังนี้

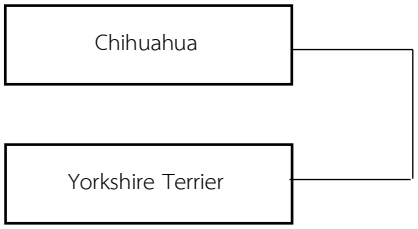
1) รวมจุดข้อมูล 'Border Colle' กับ 'Portuguese Water Dog' เป็นคลัสเตอร์ที่ 1

(1. Border Colle => Portuguese Water Dog : 0.17659)



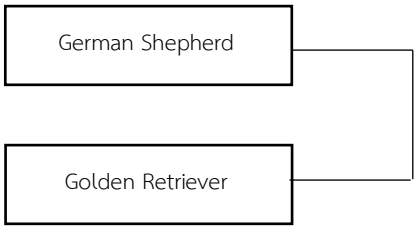
2) รวมจุดข้อมูล 'Chihuahua' กับ 'Yorkshire Terrier' เป็นคลัสเตอร์ที่ 2

(2. Chihuahua => Yorkshire Terrier : 0.27475)



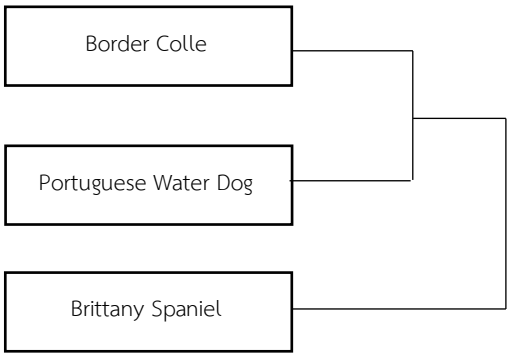
3) รวมจุดข้อมูล 'German Shepherd' กับ 'Golden Retriever' เป็นคลัสเตอร์ที่ 3

(3. German Shepherd => Golden Retriever : 0.32685)



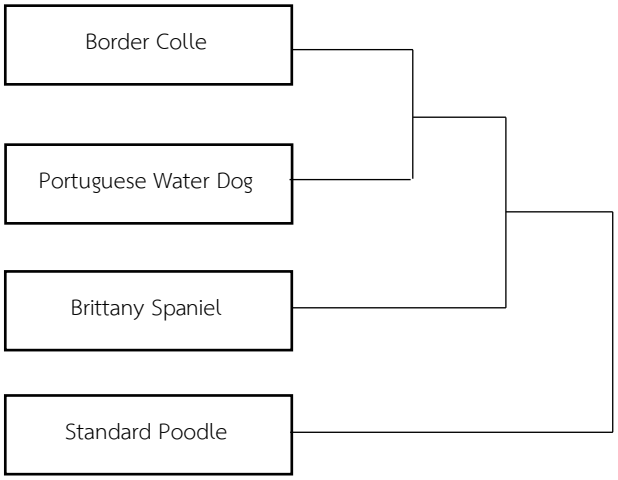
4) รวมจุดข้อมูล 'Brittany Spaniel' กับคลัสเตอร์ที่ 1 ได้เป็นคลัสเตอร์ที่ 4

(4. Border Colle => Brittany Spaniel : 0.35319)



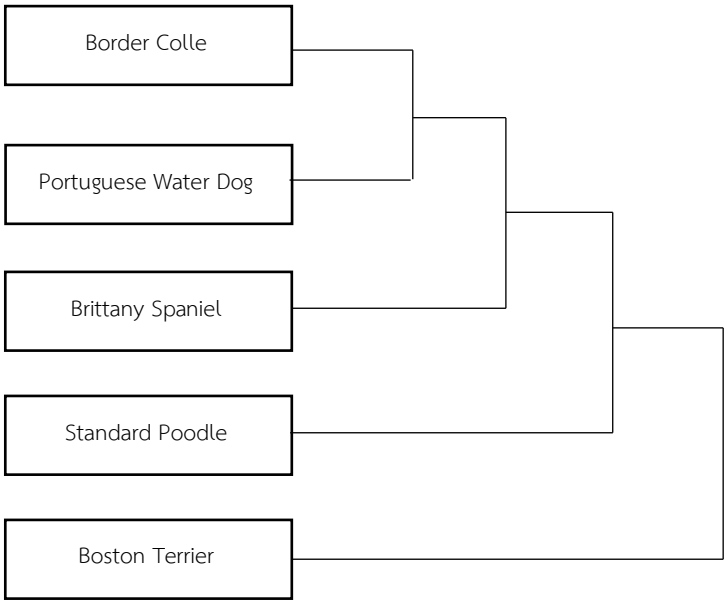
5) รวมจุดข้อมูล 'Standard Poodle' กับคลัสเตอร์ที่ 4 ได้เป็นคลัสเตอร์ที่ 5

(5. Portuguese Water Dog => Standard Poodle : 0.43235)



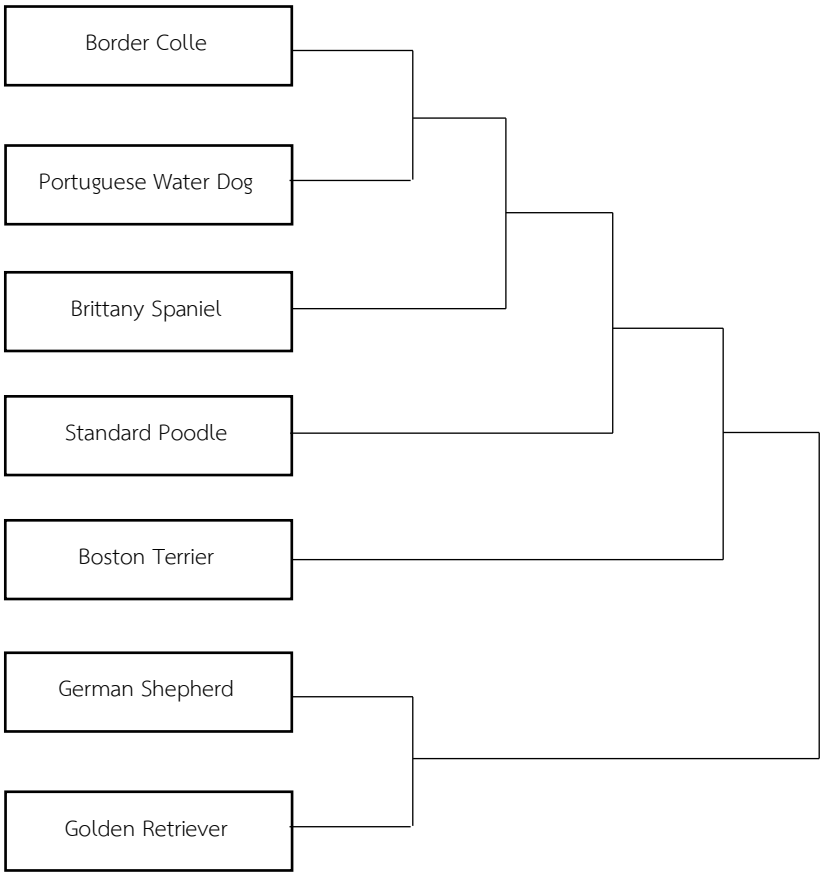
6) รวมจุดข้อมูล 'Boston Terrier' กับคลัสเตอร์ที่ 5 ได้เป็นคลัสเตอร์ที่ 6

(6. Boston Terrier => Brittany Spaniel : 0.43235)



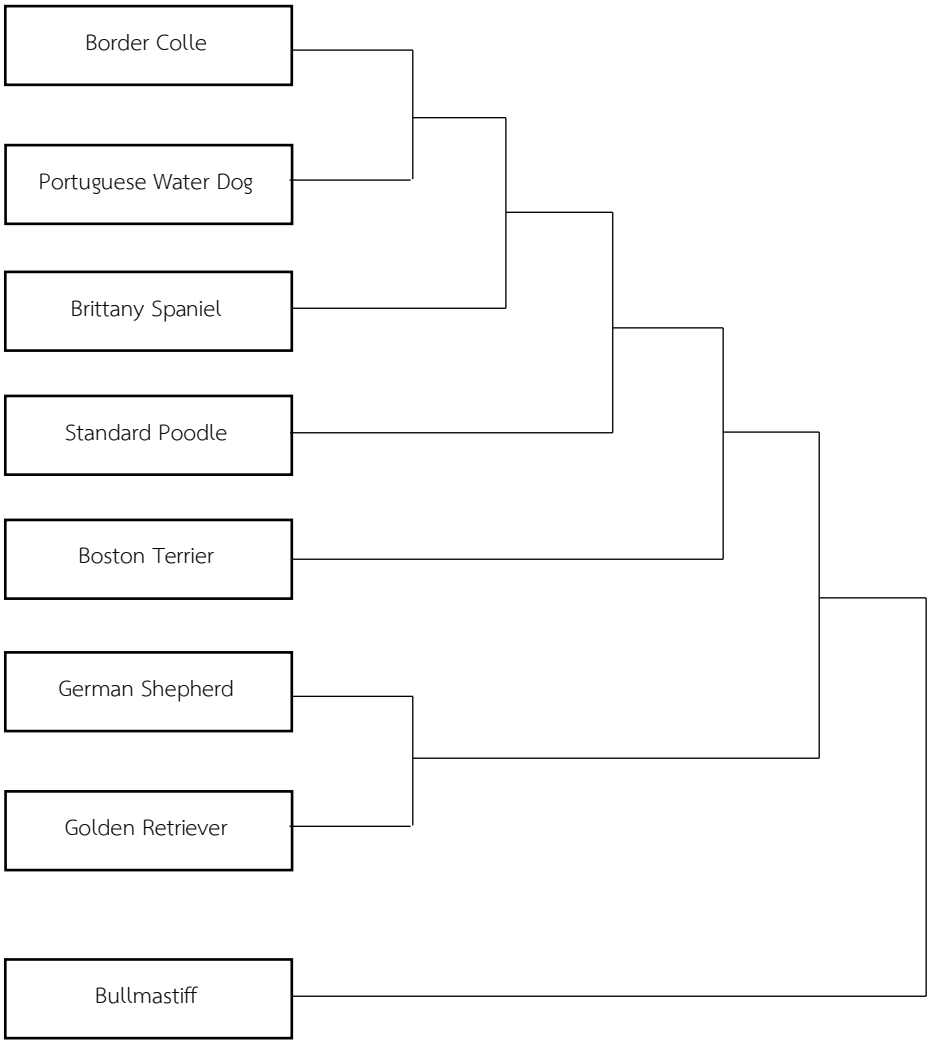
7) รวมคลัสเตอร์ที่ 3 กับคลัสเตอร์ที่ 6 ได้เป็นคลัสเตอร์ที่ 7

(8. Golden Retriever => Portuguese Water Dog : 0.52343)



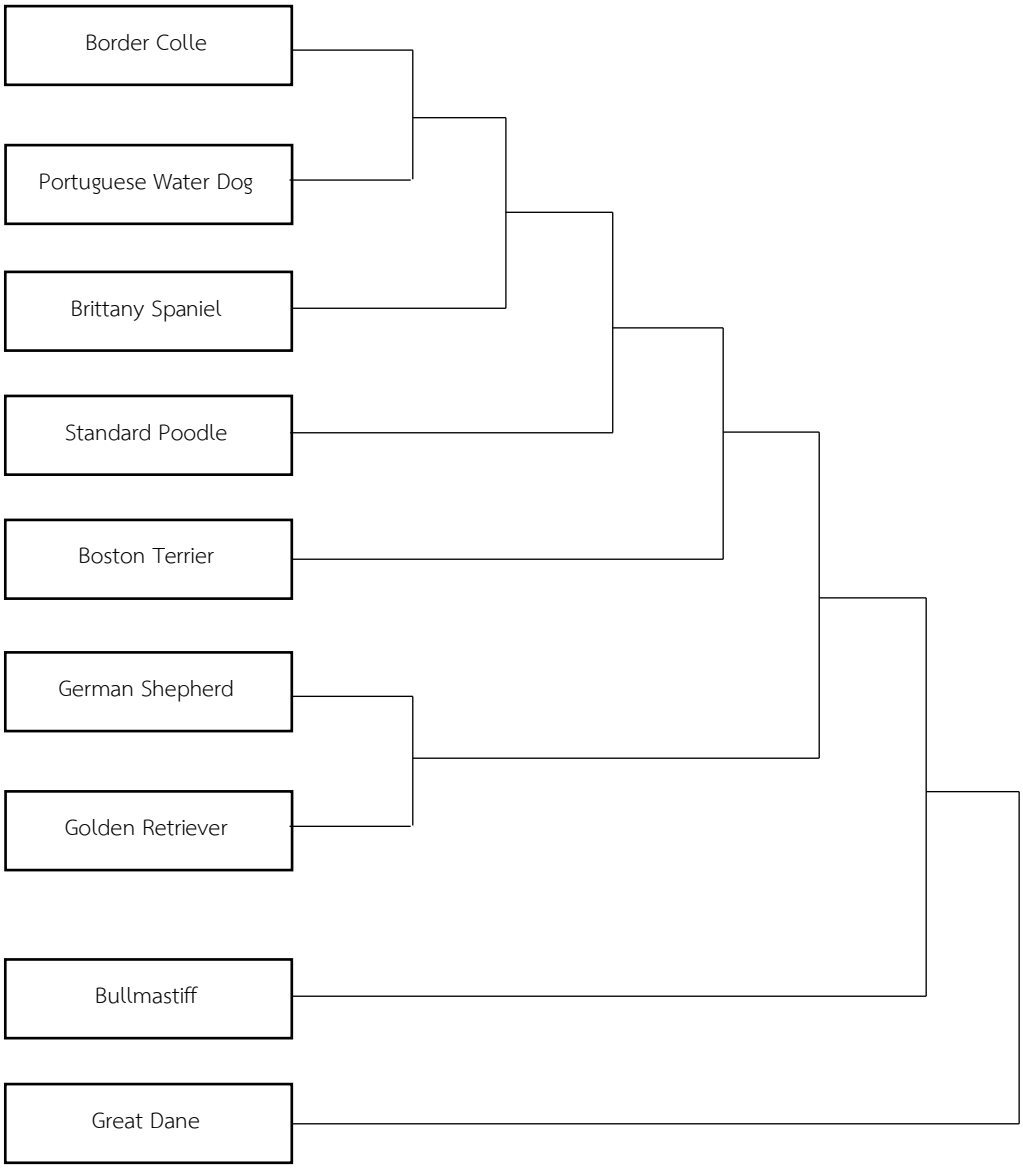
8) รวมจุดข้อมูล 'Bullmastiff' กับคลัสเตอร์ที่ 7 ได้เป็นคลัสเตอร์ที่ 8

(17. Bullmastiff => German Shepherd : 0.97598)



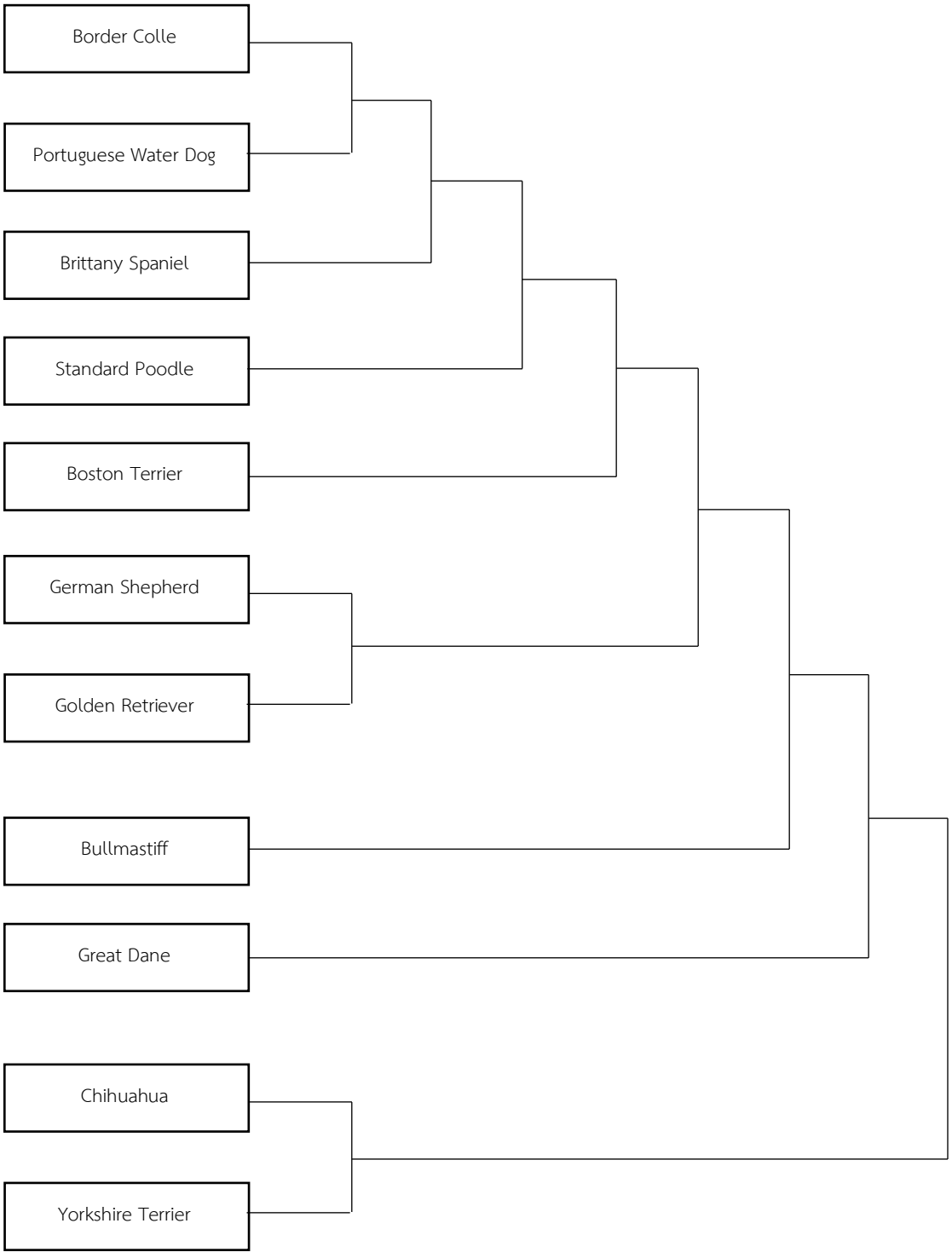
9) รวมจุดข้อมูล 'Great Dane' กับคลัสเตอร์ที่ 8 ได้เป็นคลัสเตอร์ที่ 9

(21. Bullmastiff => Great Dane : 1.1246)



10) รวมคลัสเตอร์ที่ 2 กับคลัสเตอร์ที่ 9 ได้เป็นคลัสเตอร์สุดท้ายคือคลัสเตอร์ที่ 10

(22. Boston Terrier => Chihuahua : 1.1276)



```
1. Border Colle          => Portuguese Water Dog : 0.17659
2. Chihuahua             => Yorkshire Terrier    : 0.27475
3. German Shepherd       => Golden Retriever     : 0.32685
4. Border Colle          => Brittany Spaniel     : 0.35319
5. Portuguese Water Dog  => Standard Poodle      : 0.43235
6. Boston Terrier        => Brittany Spaniel     : 0.43235
7. Border Colle          => Standard Poodle      : 0.46662
8. Golden Retriever      => Portuguese Water Dog : 0.52343
9. Brittany Spaniel      => Portuguese Water Dog : 0.52978
10. Golden Retriever      => Standard Poodle      : 0.55893
11. Brittany Spaniel      => Standard Poodle      : 0.68299
12. Border Colle          => Golden Retriever     : 0.69257
13. Border Colle          => Boston Terrier       : 0.7816
14. German Shepherd       => Portuguese Water Dog : 0.83066
15. German Shepherd       => Standard Poodle      : 0.87121
16. Boston Terrier        => Portuguese Water Dog : 0.9573
17. Bullmastiff           => German Shepherd      : 0.97598
18. Border Colle          => German Shepherd      : 1.0052
19. Brittany Spaniel      => Golden Retriever     : 1.0383
20. Boston Terrier        => Standard Poodle      : 1.0845
21. Bullmastiff           => Great Dane           : 1.1246
22. Boston Terrier        => Chihuahua            : 1.1276
23. Bullmastiff           => Golden Retriever     : 1.2424
24. Brittany Spaniel      => German Shepherd      : 1.3559
25. Boston Terrier        => Yorkshire Terrier    : 1.3996
26. Boston Terrier        => Golden Retriever     : 1.4705
27. Brittany Spaniel      => Chihuahua            : 1.4958
28. Bullmastiff           => Standard Poodle      : 1.6446
29. Brittany Spaniel      => Yorkshire Terrier    : 1.7578
30. Bullmastiff           => Portuguese Water Dog : 1.7642
31. Boston Terrier        => German Shepherd      : 1.7866
32. Border Colle          => Chihuahua            : 1.8387
33. Border Colle          => Bullmastiff          : 1.9279
34. Chihuahua             => Standard Poodle      : 1.971
35. Chihuahua             => Portuguese Water Dog : 2.0115
36. German Shepherd       => Great Dane           : 2.0649
37. Border Colle          => Yorkshire Terrier    : 2.0959
38. Standard Poodle       => Yorkshire Terrier    : 2.2004
39. Brittany Spaniel      => Bullmastiff          : 2.2612
40. Portuguese Water Dog  => Yorkshire Terrier    : 2.2667
41. Golden Retriever      => Great Dane           : 2.3555
42. Chihuahua             => Golden Retriever     : 2.476
43. Boston Terrier        => Bullmastiff          : 2.6913
44. Golden Retriever      => Yorkshire Terrier    : 2.7189
45. Great Dane            => Standard Poodle      : 2.7673
46. Chihuahua             => German Shepherd      : 2.8028
47. Great Dane            => Portuguese Water Dog : 2.8789
48. German Shepherd       => Yorkshire Terrier    : 3.0456
49. Border Colle          => Great Dane           : 3.0461
50. Brittany Spaniel      => Great Dane           : 3.3834
51. Bullmastiff           => Chihuahua            : 3.6067
52. Boston Terrier        => Great Dane           : 3.8144
53. Bullmastiff           => Yorkshire Terrier    : 3.8236
54. Chihuahua             => Great Dane           : 4.7215
55. Great Dane            => Yorkshire Terrier    : 4.9314
```

รูปที่ 3: Euclidean Distance ของจุดข้อมูลแต่ละคู่ในชุดข้อมูลพันธุ์สุนัข

แบบฝึกหัด

1. จงสร้าง scatterplot เพื่อแสดงความสัมพันธ์ระหว่างความสูงและน้ำหนักในชุดข้อมูลพันธุ์สุนัข รูป scatterplot ที่ได้มีความสอดคล้องกับ dendrogram ที่ได้จาก HCA หรือไม่อย่างไร
2. จงจัดกลุ่มข้อมูลในชุดข้อมูลพันธุ์สุนัข โดยใช้วิธีการดังต่อไปนี้
   ก. Complete-Linkage Clustering
   ข. Average-Linkage Clustering

3. จงอธิบายอัลกอริทึมการจัดกลุ่มข้อมูล k-means และแสดงตัวอย่างโปรแกรมการจัดกลุ่มข้อมูลด้วย k-means

เอกสารอ้างอิง

[1] Aurelien Geron. Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow, 2ed, O'Reilly Media, Inc. 2019.

[2] Mohamed Noordeen Alaudeen; Rohan Chopra; Aaron England. Data Science with Python, Packt Publishing, 2019.

[3] Ron Zacharski. A Programmer's Guide to Data Mining: http://guidetodatamining.com (retrieved: October 29, 2019)