วิชาปัญญาประดิษฐ์
(Artificial Intelligence)

อ. พิชัย จอดพิมาย

Email : pichaiku@gmail.com

แผนการสอน

- สัปดาห์ที่ 1 : นิยาม ความสำคัญ และเทคโนโลยีของปัญญาประดิษฐ์
- สัปดาห์ที่ 2-3 : ปริภูมิสถานะและการค้นหา
- สัปดาห์ที่ 4-5 : ตรรกศาสตร์ประพจน์และตรรกศาสตร์พรีดิเคต
- สัปดาห์ที่ 6-8 : การประยุกต์ใช้ในเกม หุ่นยนต์ และระบบผู้เชี่ยวชาญ
- สัปดาห์ที่ 9 : สอบกลางภาค
- สัปดาห์ที่ 10-11 : **การเรียนรู้ของเครื่องจักร**
- สัปดาห์ที่ 12 : การประมวลผลภาพ
- สัปดาห์ที่ 13 :การประมวลผลภาษาธรรมชาติ
- สัปดาห์ที่ 14-15 : การประยุกต์ใช้ในเหมืองข้อมูลและการวิเคราะห์ข้อมูลขนาดใหญ่
- สัปดาห์ที่ 16 : สอบปลายภาค

เนื้อหาการเรียนรู้ของเครื่องจักร (ส่วนที่ 2)

1. เครื่องมือสำหรับการวิเคราะห์ข้อมูลเบื้องต้นและการเรียนรู้ของเครื่องจักร
2. การวิเคราะห์ข้อมูลเบื้องต้น
3. การเรียนรู้ของเครื่องจักรสำหรับงานจัดกลุ่มข้อมูล
4. การเรียนรู้ของเครื่องจักรสำหรับงานแบ่งกลุ่มข้อมูล
5. การเรียนรู้ของเครื่องจักรสำหรับงานรีเกรสชัน
6. แบบฝึกหัด

เครื่องมือสำหรับการวิเคราะห์ข้อมูลเบื้องต้น และ ML

pandas

$$y_{it} = \beta' x_{it} + \mu_i + \epsilon_{it}$$

https://pandas.pydata.org/

เป็น Open-source Library สำหรับการจัดการข้อมูลและการวิเคราะห์ข้อมูล (Data Analysis) ด้วยภาษา Python นอกจาก pandas ยังมี library อื่น ๆ ที่จำเป็น เช่น numpy/scipy/matplotlib

https://scikit-learn.org/stable/

เป็น Open-source Library ซึ่งพัฒนาโดย David Cournapeau ในปี 2007 ภายใต้โครงการ Google Summer of Code project มีวัตถุประสงค์เพื่อให้นักพัฒนาสามารถพัฒนา Machine Learning ได้ง่ายด้วยภาษา Python

https://www.tensorflow.org/

เป็น Open-source Library ซึ่งพัฒนาโดยทีม Google Brain มีการเผยแพร่ในปี 2015 มีวัตถุประสงค์เพื่อให้นักพัฒนาสามารถพัฒนา Machine Learning ได้ง่ายด้วยภาษา Python เป็นหลัก และภาษาอื่น ๆ เช่น Java, C++, Go, etc.

https://rapidminer.com/

เป็น Commercial/Community Software สำหรับการจัดเตรียมข้อมูล (Data Preparation) การวิเคราะห์ข้อมูล และการเรียนรู้ของเครื่องจักร

https://www.cs.waikato.ac.nz/ml/weka/

เป็น Open-source Software ซึ่งพัฒนาจากภาษา Java โดย University of Waikato, New Zealand สำหรับการจัดเตรียมข้อมูล การวิเคราะห์ข้อมูล และการเรียนรู้ของเครื่องจักร

Machine Learning service

https://azure.microsoft.com/

เป็น Commercial/Trial Software ซึ่งติดตั้งในสภาพแวดล้อมแบบ Cloud Computing สำหรับการจัดเตรียมข้อมูลและการเรียนรู้ของเครื่องจักร แม้ว่า Azure จะเตรียม ML ไว้ให้ มันยังอนุญาตให้นักพัฒนาสามารถใช้งาน TensorFlow และ Scikit-learn ใน Azure ได้

การวิเคราะห์ข้อมูลเบื้องต้น

การวิเคราะห์ข้อมูล (Data Analysis) คือ กระบวนการในการตรวจสอบข้อมูล เพื่อค้นหาประโยชน์หรือข้อสรุปจากข้อมูล สำหรับใช้ในการตัดสินใจ การวิเคราะห์ข้อมูลเบื้องต้น สามารถแบ่งออกเป็น สถิติเชิงบรรยาย (Descriptive Statistics) การวิคราะห์ข้อมูลด้วยภาพ (Data Visualization) และการทดสอบสมมติฐานทางสถิติ (Statistical Hypothesis Test)

ตัวอย่างการวิคราะห์ข้อมูลเบื้องต้น

- การเชื่อมต่อฐานข้อมูล
- สถิติเชิงบรรยาย (Descriptive Statistics)
- การวิเคราะห์ข้อมูลด้วยภาพ (Data Visualization)
  - การสร้างกราฟแท่ง (Bar plot)
  - การสร้างกราฟแท่ง (Multiple bar plot)
  - การสร้างกราฟเส้น (Line Chart)
  - การสร้างกราฟวงกลม (Pie Chart)
  - การสร้างฮิสโทแกรม (Histogram)
  - การสร้างฮิสโทแกรมย่อย (Sub histogram)
  - การสร้างบ๊อกพล็อต (Boxplot)
  - การสร้างสแคท'เทอะ (Scatter)

การเชื่อมต่อฐานข้อมูล

การเชื่อมต่อไฟล์ CSV

```
import pandas as pd #library for data analysis
x=pd.read_csv('data.csv')
```

การเชื่อมต่อ MySQL Server

ติดตั้ง MySQL Server Connector (ติดตั้งครั้งเดียว)

```
!pip install mysql-connector-python==8.0.11
```

เชื่อมต่อ MySQL Server Connector

```
import pandas as pd #library for data analysis
import mysql.connector as sql #library for mysql connection
db_connection = sql.connect(host='localhost',
database='ai_db',user='root', password='')
x=pd.read_sql('select * from cocomo_traindata', con=db_connection)
```

Index	KSLOC	PREC	FLEX	RESL	TEAM	PMAT
0	0.27	3.72	1.01	5.65	3.29	4.68
1	1.02	4.96	1.01	5.65	2.19	4.68
2	2.52	3.72	1.01	5.65	3.29	4.68
3	4.02	3.72	3.04	2.83	0	3.12
4	4.28	4.96	1.01	5.65	3.29	4.68
5	4.48	4.96	1.01	5.65	1.1	4.68
6	5	4.96	1.01	5.65	1.1	4.68
7	5.11	2.48	1.01	5.65	1.1	4.68
8	5.29	1.24	1.01	5.65	0	4.68
9	7.44	4.96	1.01	5.65	1.1	4.68
10	7.7	4.96	1.01	5.65	0	4.68
11	8.41	4.96	1.01	5.65	1.1	4.68

รูปแสดงข้อมูลที่ได้จากการเชื่อมต่อ MySQL Server

สถิติเชิงบรรยาย (Descriptive Statistics)

```
import pandas as pd #library for data analysis
import mysql.connector as sql #library for mysql connection
import numpy as np #library for scientific computation
import scipy as sp #library for mathematics, science, and engineering
computation

db_connection = sql.connect(host='localhost', database='ai_db',
                            user='root', password='')
x=pd.read_sql('select * from cocomo_traindata', con=db_connection)

stat=pd.DataFrame({'MIN':np.min(x), 'MAX':np.max(x), 'MEAN':np.mean(x),
                   'MEDIAN':np.median(x), 'SD':np.std(x), 'SKEW':sp.stats.skew(x),
                   'KUR':sp.stats.kurtosis(x)})
```

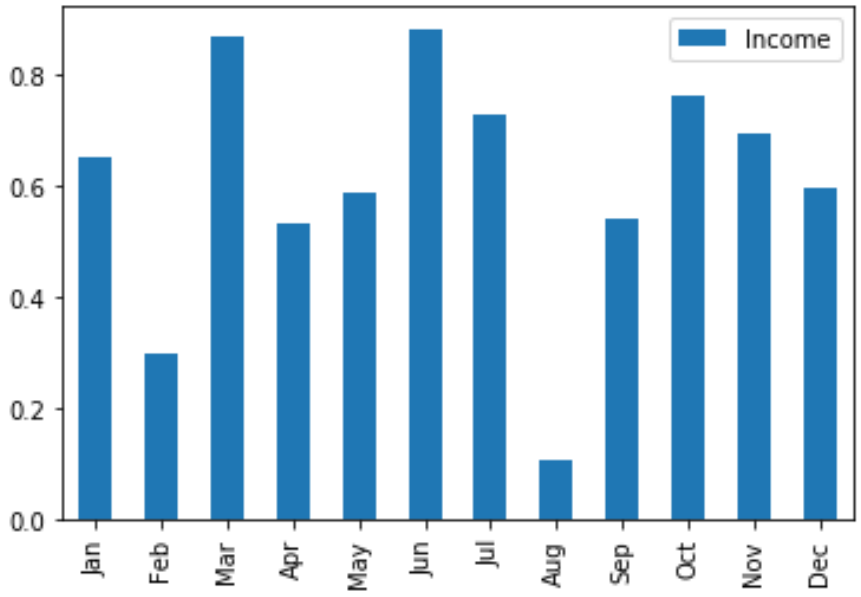
stat - DataFrame

Index	MIN	MAX	MEAN	MEDIAN	SD	SKEW	KUR
KSLOC	0.27	112.28	29.657	1	27.8977	0.989402	0.490078
PREC	1.24	6.2	4.092	1	1.32589	-0.945581	-0.249845
FLEX	1.01	3.04	1.31467	1	0.700936	1.95548	1.9605
RESL	1.41	5.65	5.27367	1	0.959957	-2.79781	7.30108
TEAM	0	3.29	1.39	1	1.01735	0.444869	-0.601404
PMAT	3.12	4.68	4.524	1	0.468	-2.66667	5.11111
RELY	1	1.1	1.01333	1	0.0339935	2.15728	2.65385
DATA	0.9	1	0.91	1	0.03	2.66667	5.11111
CPLX	0.87	1.17	1.00267	1	0.0551926	0.931892	4.85822
RUSE	0.95	1.24	1.05967	1	0.102648	1.06615	-0.62192
DOCU	0.91	1.11	0.936	1	0.0621611	2.20505	3.23977
TIME	1	1.29	1.00967	1	0.0520566	5.19947	25.0345
STOR	1	1	1	1	0	0	-3
PVOL	0.87	0.87	0.87	1	2.22045e-16	-1	-2
ACAP	0.85	0.85	0.85	1	3.33067e-16	-1	-2
PCAP	0.88	1	0.888	1	0.0299333	3.4744	10.0714
PCON	0.81	0.81	0.81	1	2.22045e-16	1	-2
APEX	0.81	1.1	0.994	1	0.109502	-0.465157	-1.32158
PLEX	0.85	1.19	0.952333	1	0.0794432	1.16885	0.920406

รูปแสดงการค่าสถิติเชิงบรรยาย

การสร้างกราฟแท่ง (Bar plot)

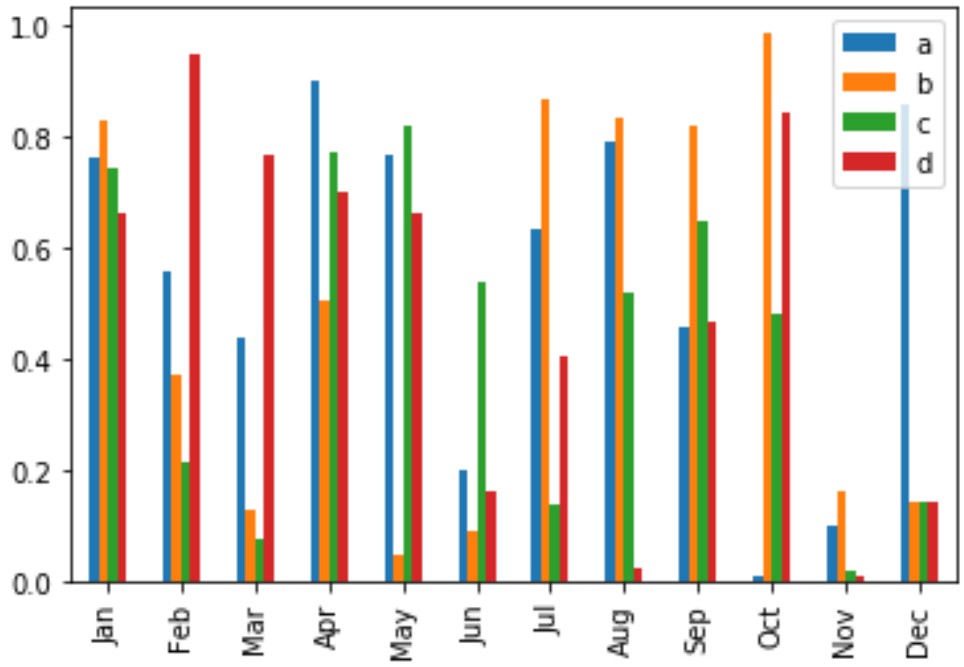
```
import pandas as pd #library for data analysis
import numpy as np #library for scientific computing

x = pd.DataFrame(np.random.rand(12, 1),
      index=['Jan','Feb','Mar','Apr','May','Jun','Jul','Aug','Sep','Oct','Nov','Dec'],
      columns=['Income'])
x.plot.bar()
```



รูปแสดงกราฟแท่ง

การสร้างกราฟแท่ง (Multiple bar plot)

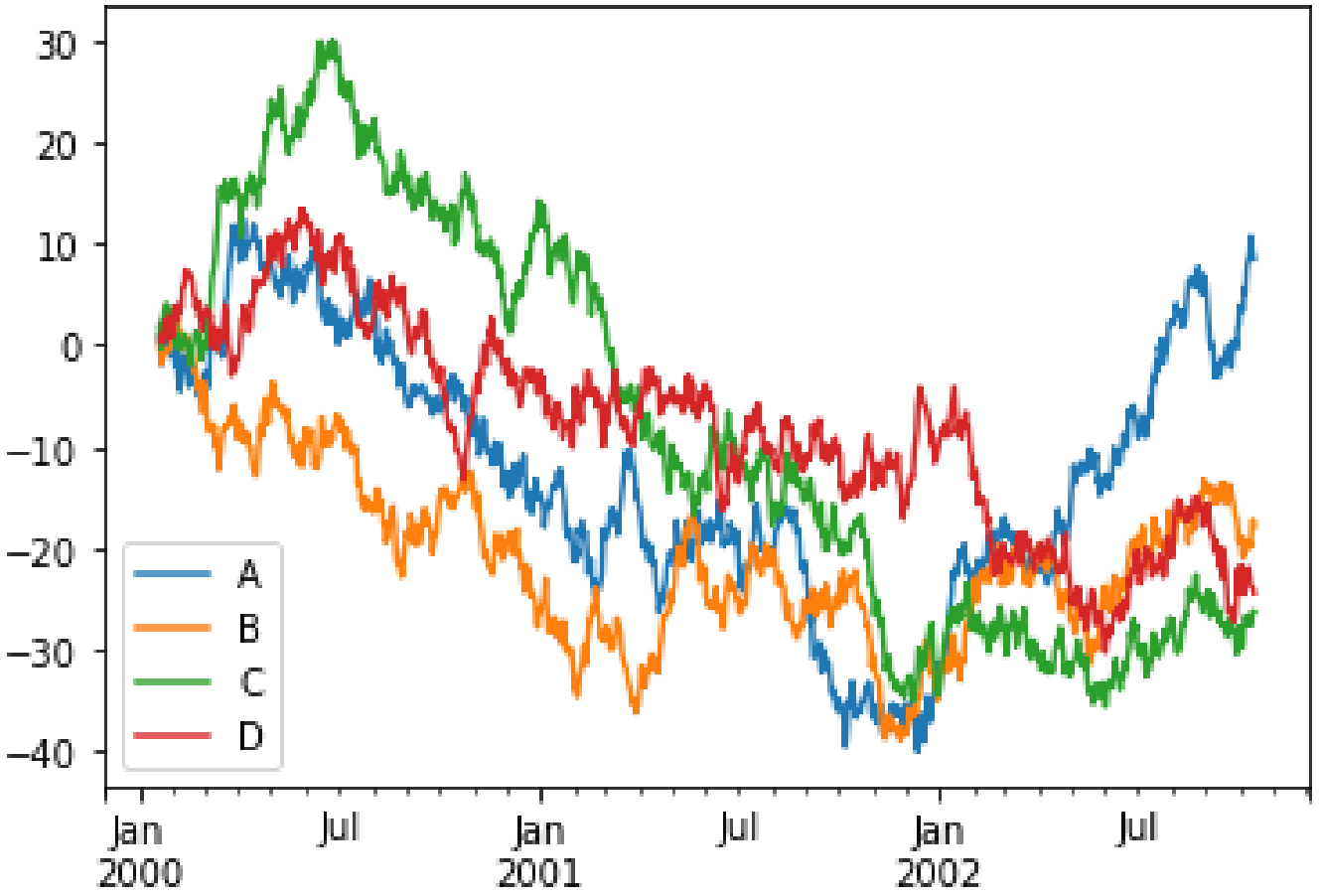
```
import pandas as pd #library for data analysis
import numpy as np #library for scientific computing

x = pd.DataFrame(np.random.rand(12, 4),
      index=['Jan','Feb','Mar','Apr','May','Jun','Jul','Aug','Sep','Oct','Nov','Dec'],
      columns=['a', 'b', 'c', 'd’])
x.plot.bar()
```



รูปแสดงกราฟหลายแท่ง

การสร้างกราฟเส้น (Line Chart)

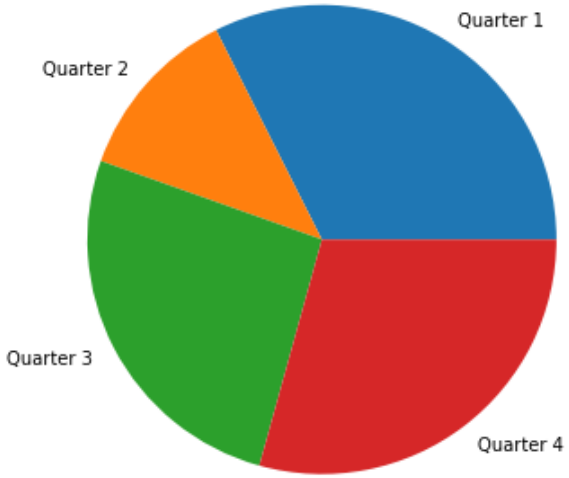
```
import pandas as pd #library for data analysis
import numpy as np #library for scientific computing

x = pd.DataFrame(np.random.randn(1000, 4),
                 index=pd.date_range('20/1/2000',
                 periods=1000),
                 columns=list('ABCD'))#create data frame
x = x.cumsum()#sum data in each column
x.plot()#plot
```



รูปแสดงกราฟเส้น

การสร้างกราฟวงกลม (Pie Chart)

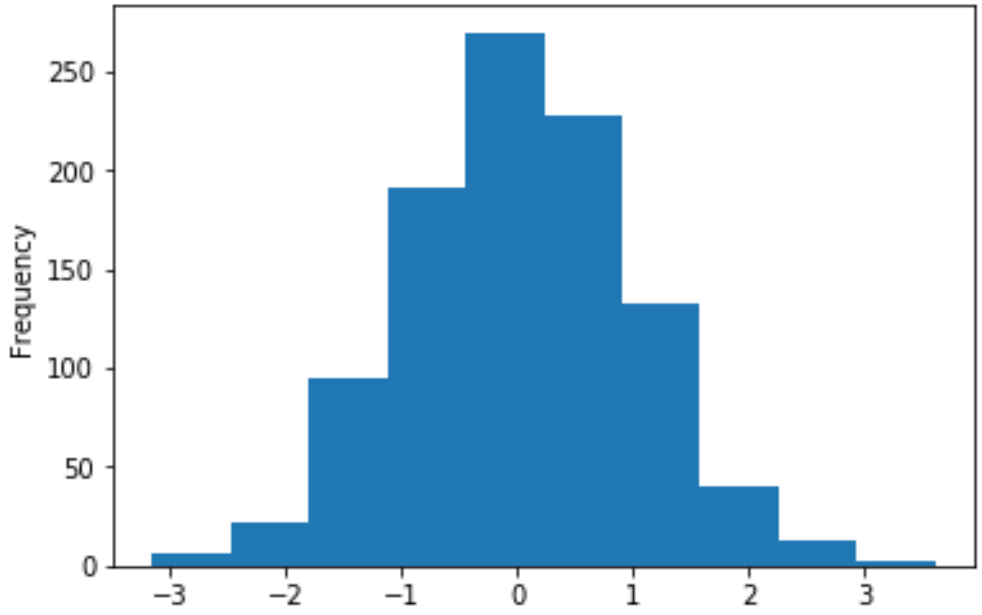
```
import pandas as pd #library for data analysis
import numpy as np #library for scientific computing

x = pd.Series(100*np.random.rand(4),
      index=['Quarter 1','Quarter 2','Quarter 3','Quarter 4'],
      name='')
x.plot.pie(figsize=(6, 6))
```



รูปแสดงกราฟวงกลม

การสร้างฮิสโทแกรม (Histogram)

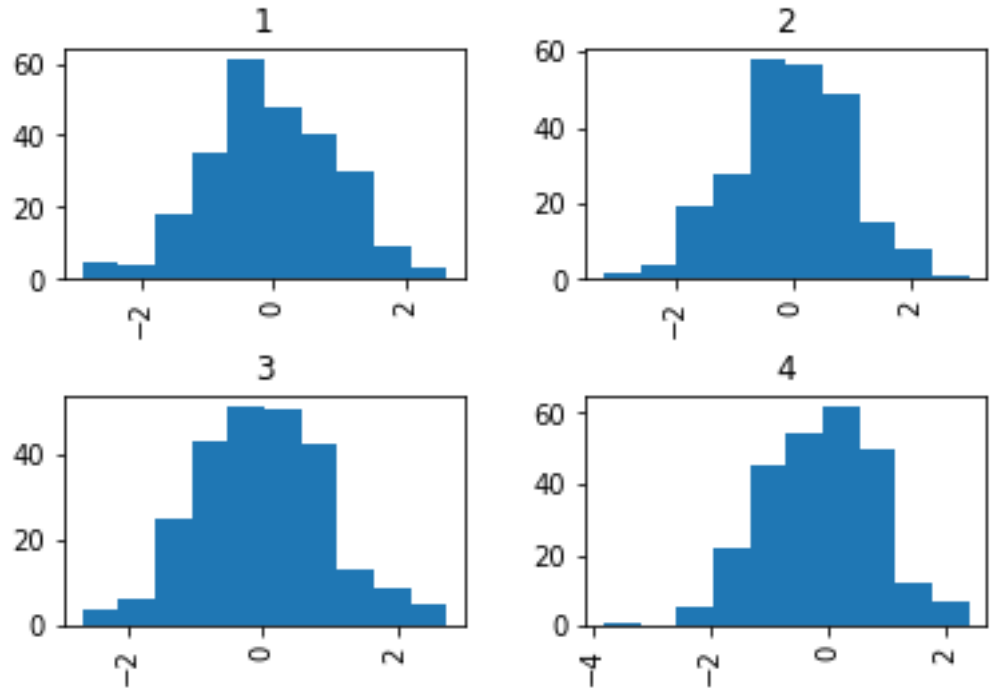
```
import pandas as pd #library for data analysis
import numpy as np #library for scientific computing

x = pd.Series(np.random.randn(1000))
x.plot.hist()
```



รูปแสดงฮิสโทแกรม

การสร้างฮิสโทแกรมย่อย (Sub histogram)

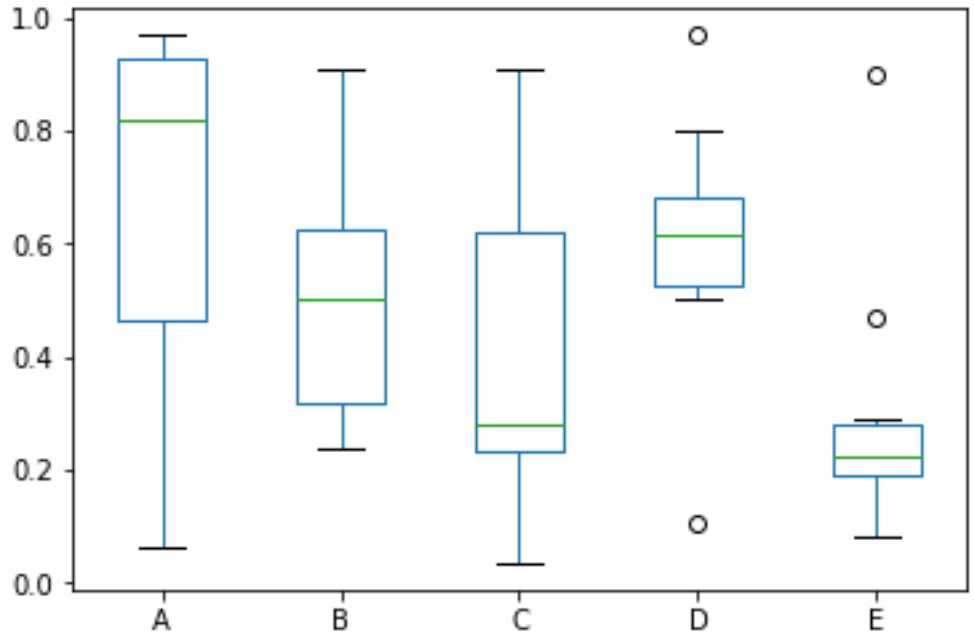
```
import pandas as pd #library for data analysis
import numpy as np #library for scientific computing

x = pd.Series(np.random.randn(1000))
x.hist(by=np.random.randint(1, 5, 1000), figsize=(6, 4))
```



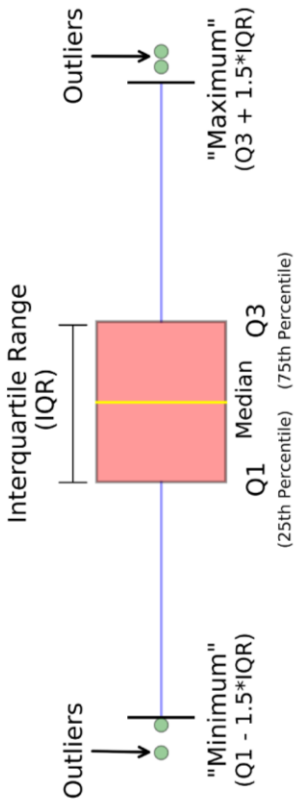
รูปแสดงฮิสโทแกรมย่อย 4 ฮิสโทแกรม

การสร้างบ๊อกพล็อต (Boxplot)

```
import pandas as pd #library for data analysis
import numpy as np #library for scientific computing

x = pd.DataFrame(np.random.rand(10, 5), columns=['A', 'B', 'C', 'D', 'E'])
x.plot.box()
```

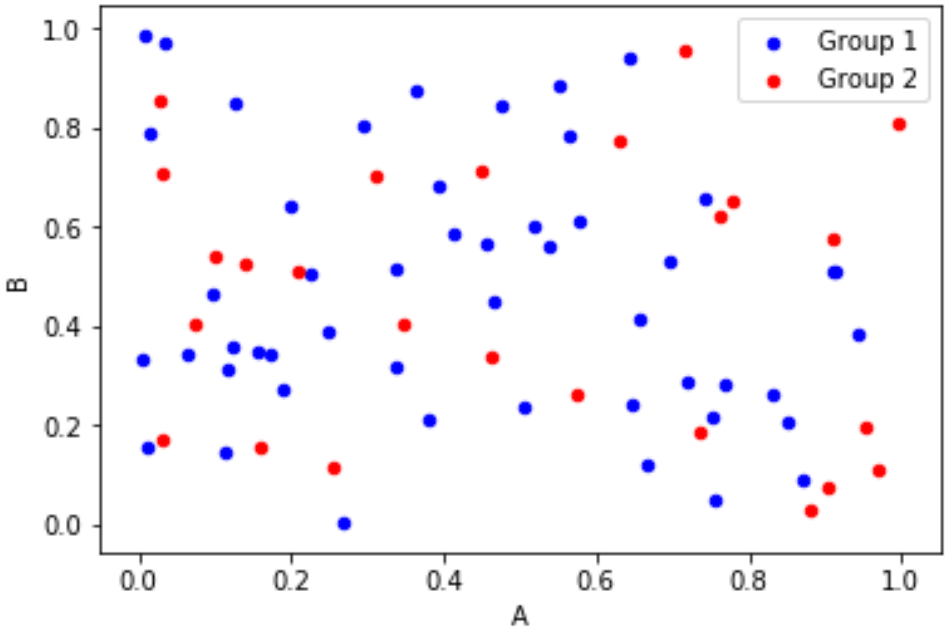


รูปแสดงบ๊อกพล็อต



การสร้างสแคท'เทอะ (Scatter)

```
import pandas as pd #library for data analysis
import numpy as np #library for scientific computing

x1 = pd.DataFrame(np.random.rand(50, 2), columns=['A', 'B'])
x2 = pd.DataFrame(np.random.rand(25, 2), columns=['A', 'B'])
ax = x1.plot.scatter(x='A', y='B', color='Blue', label='Group 1');
x2.plot.scatter(x='A', y='B', color='Red', label='Group 2', ax=ax);
```



รูปแสดงสแคท'เทอะ

การเรียนรู้ของเครื่องจักรสำหรับงานจัดกลุ่มข้อมูล

เทคนิคการเรียนรู้ของเครื่องจักรสำหรับงานจัดกลุ่มข้อมูล (Clustering Task) เป็นรูปแบบหนึ่งของเทคนิคการเรียนรู้ของเครื่องจักร ที่ไม่มีการสอนหรือไม่มีเฉลย

ตัวอย่างการนำไปประยุกต์ใช้ ดังนี้

- การจัดกลุ่มลูกค้าเพื่อเสนอประกันหรือโปรโมชัน
- การจัดกลุ่มร้านค้าปลีกเพื่อสร้างกลยุทธ์
- การจัดกลุ่มทางชีววิทยาของสิ่งมีชีวิต
- การจัด Zone อันตรายของแผ่นดินไหว
- นำไปจัดกลุ่มข้อมูลเพื่อเตรียมข้อมูล (Pre-processing) ก่อนดำเนินงานแบ่งกลุ่มข้อมูล (Classification Task) และงานรีเกรสชัน (Regression Task)

ตัวอย่างเทคนิค ดังนี้

- K-mean
- Competitive Learning
- Hierarchical Clustering
- Fuzzy C-Mean
- DBSCAN
- Self-Organizing Map (SOM)

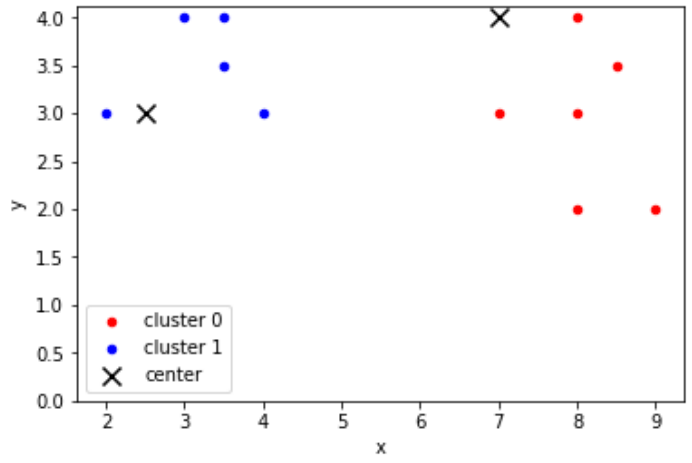
K-mean Technique

K-mean เป็นเทคนิคสำหรับการจัดกลุ่มข้อมูลที่ง่ายและได้รับความนิยมสูง ซึ่งใช้ Euclidean Distance Method ในการวัดระยะทางหรือความคล้ายกันระหว่างศูนย์กลางของกลุ่มข้อมูล (Center) กับรายการข้อมูล (Data Point) แล้วหาค่าเฉลี่ย (Mean) ของข้อมูลที่เป็นสมาชิกของกลุ่มข้อมูลนั้น ๆ เพื่อทำการปรับตำแหน่งของ Center

K-mean Algorithm

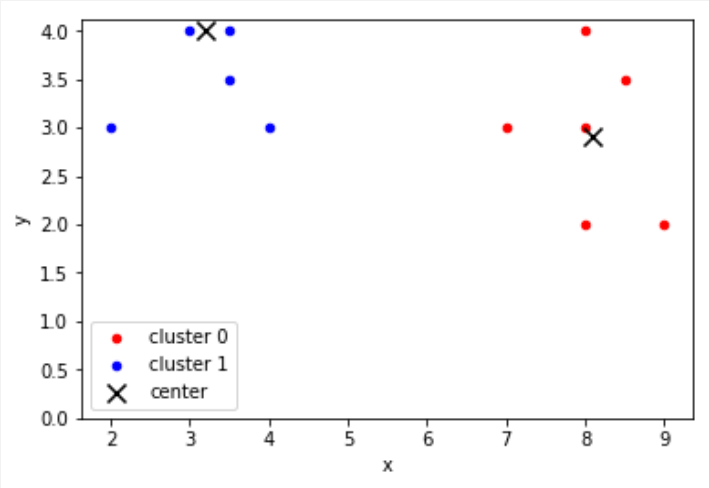
1: Initialise the centres $\mathbf{m}_i$, $i = 1, \ldots, K$.
2: **until** no center is changed
3: For each centre i, find all the $\mathbf{x}^n$ for which i is the nearest (in Euclidean sense) centre.
4: Call this set of points $\mathcal{N}_i$. Let N_i be the number of datapoints in set $\mathcal{N}_i$.
5: Update the means

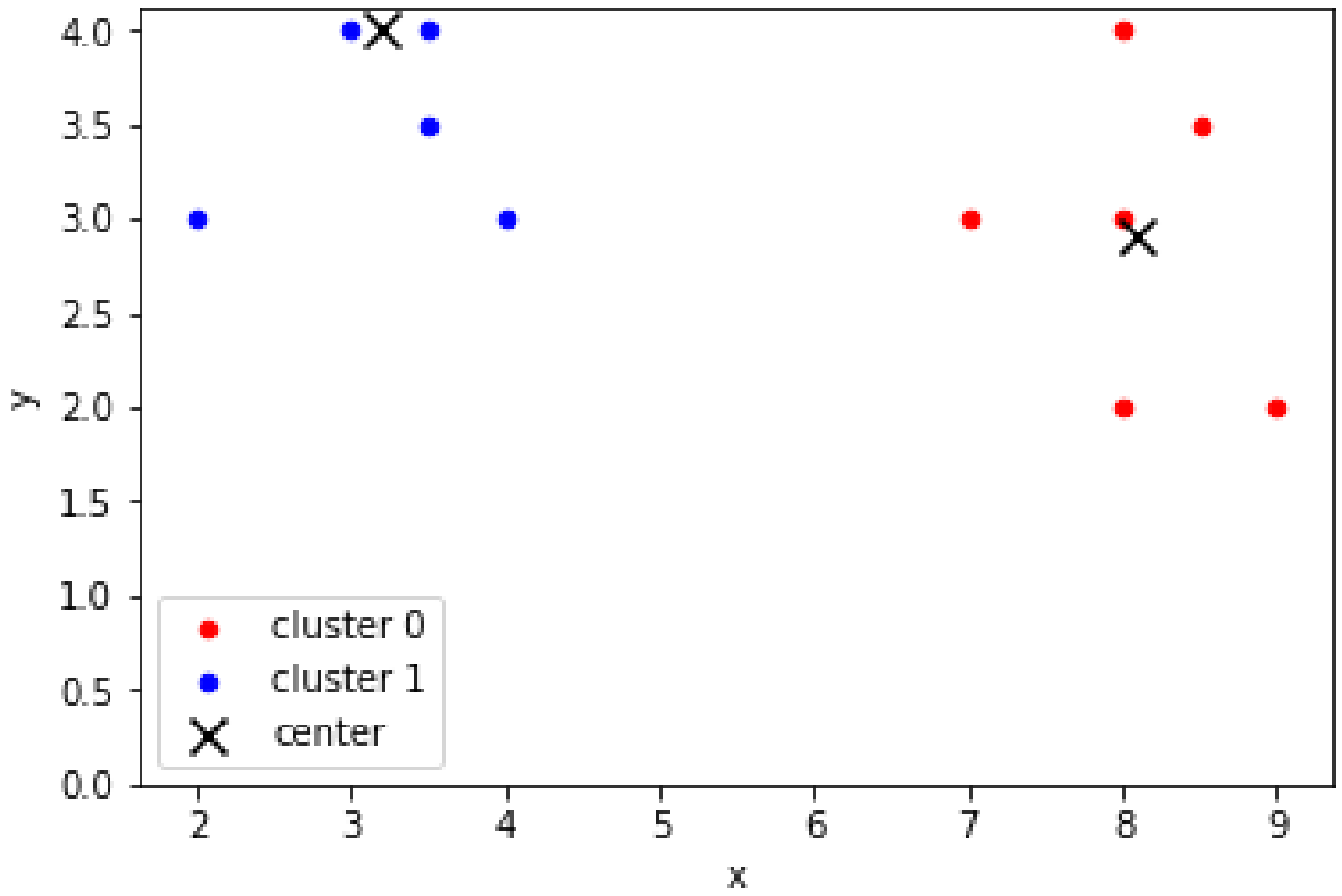
$$\mathbf{m}_i^{new} = \frac{1}{N_i} \sum_{n \in \mathcal{N}_i} \mathbf{x}^n$$



การเรียกใช้งาน k-mean จาก Scikit-learn

```
from sklearn.cluster import KMeans #import k-mean ligrary
import pandas as pd #import pandas library for data analysis (plot)

data=pd.DataFrame([[2, 3],[3, 4], [4, 3],[3.5, 3.5],[3.5, 4],[4, 5],
                   [2.5, 4.5],[3, 5],[8, 2],[7, 3],[8, 4],[8.5, 3.5],
                   [9, 2], [8, 3]],
                  columns=['x','y'])

kmeans = KMeans(n_clusters=2).fit(data) #find clusters

kmeans.predict([[0, 0], [12, 3]])#predict a cluster

#visualize data
data['cluster']=kmeans.labels_ #given clusters
cluster0=data.loc[data['cluster'] == 0] #find cluster0 data
cluster1=data.loc[data['cluster'] == 1] #find cluster1 data
center=pd.DataFrame(kmeans.cluster_centers_,
                    columns=['x','y'])#given centers of clusters #create center data
ax=cluster0.plot.scatter(x='x', y='y',
                         ylim=0, color='Red', label='cluster 0')#plot cluster 0
ax=cluster1.plot.scatter(x='x', y='y',
                         ylim=0, color='Blue', label='cluster 1', ax=ax)#plot cluster 1
center.plot.scatter(x='x', y='y',
                    color={'Black'},
                    marker='x', s=100, label='center', ax=ax);#plot center
```





รูปแสดงการจัดกลุ่มข้อมูลด้วย K-mean Technique

Competitive Learning Technique

Competitive Learning เป็นหนึ่งในเทคนิคสำหรับการจัดกลุ่มข้อมูล ซึ่งจัดอยู่ในกลุ่มของ Artificial Neural Network หลักการทำงานคือ เริ่มต้นโดยการกำหนด Weights หรือ Centers (สามารถสุ่มค่าหรือเลือกจาก Train data) ทำการวัดระยะทางระหว่าง Sample ใน Train set กับ Weights ถ้า Weight ใดมีระยะทางใกล้กว่าถือเป็นผู้ชนะ และ Weight นั้นจะถูกปรับค่า Algorithm จะดำเนินไปเรื่อย ๆ จนกว่า Weight จะไม่เปลี่ยนแปลง

1: Initialise the weights (or centers) W_i, i=1,...,K

2: **until** no weights is changed

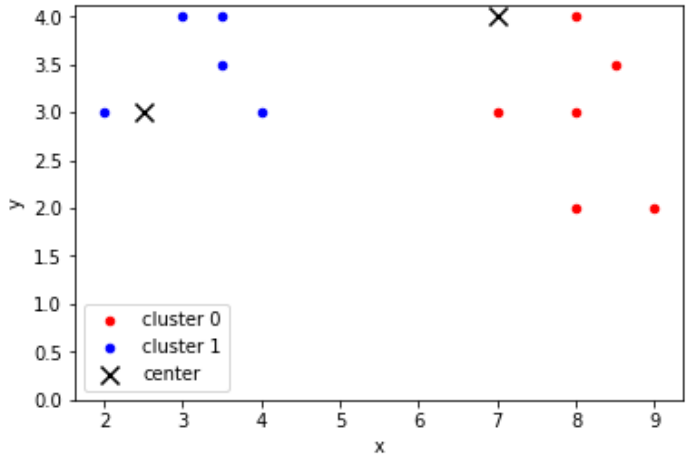
3: choose sample vector X_j from train set

4: compute distance between sample and weight vectors

$$|| X_j - W_i ||$$

5: find the winner weight (center), where i* is the property that

$$|| X_j - W_{i*} || < || X_j - W_i ||$$

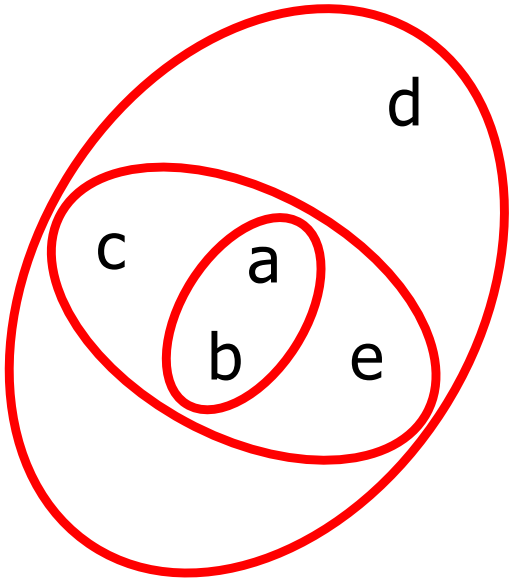


6: update the weight vector of winning unit only with

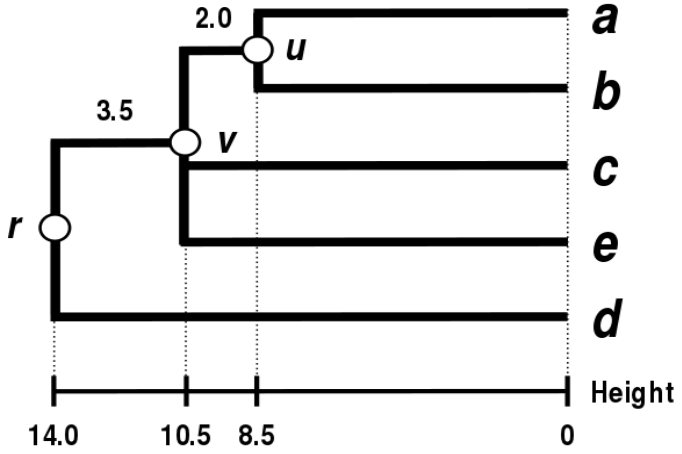
$$W_{i*} = W_{i*} + \eta(X_j - W_{i*}) \quad \# \ \eta \text{ is learning rate having a small value}$$

Hierarchical Clustering Technique

Hierarchical Clustering เป็นเทคนิคสำหรับการจัดกลุ่มข้อมูล ซึ่งสร้างกลุ่มย่อย (Nested Cluster) โดยการควบรวม (Merge) หรือการแยกกลุ่มข้อมูล (Split) ไปเรื่อย ๆ โดยที่ชั้นของกลุ่มข้อมูล(Hierarchy) ที่เกิดขึ้นจะถูกแสดงในรูปแบบของต้นไม้ (Tree หรือ Dendrogram) โดย Root Node จะครอบคุมทุกข้อมูล (All Data Points) ขณะที่ Leave Node เป็นกลุ่มข้อมูลที่ประกอบด้วยสมาชิกเพียงแค่ 1 ข้อมูล (Data Point)



รูปแสดงการจัดกลุ่มโดยใช้เทคนิค Hierarchical Clustering



รูปแสดง Tree หรือ Dendrogram ที่ได้จาก Hierarchical Clustering

Hierarchical Clustering Algorithm

1. Begin with n observations and a measure (such as Euclidean distance) of all the $\binom{n}{2} = n(n-1)/2$ pairwise dissimilarities. Treat each observation as its own cluster.

2. For $i = n, n-1, \ldots, 2$:

   (a) Examine all pairwise inter-cluster dissimilarities among the i clusters and identify the pair of clusters that are least dissimilar (that is, most similar). Fuse these two clusters. The dissimilarity between these two clusters indicates the height in the dendrogram at which the fusion should be placed.

   (b) Compute the new pairwise inter-cluster dissimilarities among the $i-1$ remaining clusters.

ตัวอย่างการคำนวณ

กำหนดให้ข้อมูลประกอบด้วยสมาชิก (a, b, c, d, e) และมี Euclidian Distance ดังแสดงในเมตริก D1

	a	b	c	d	e
a	0	17	21	31	23
b	17	0	30	34	21
c	21	30	0	28	39
d	31	34	28	0	43
e	23	21	39	43	0

จาก D1 พบว่า (a,b) อยู่ใกล้กันมากที่สุด คือ 17 จากนั้นทำการรวมกลุ่ม (a,b) เข้าด้วยกันได้ดังนี้

หมายเหตุ : ความสูงของ Dendrogram จะใช้ค่า 17/2=8.5 แทน 17

ทำการคำนวณระยะทางระหว่างกลุ่ม (a,b) กับกลุ่มอื่น ๆ ที่เหลือ ดังนี้

$$D_2((a,b),c) = min(D_1(a,c), D_1(b,c)) = min(21,30) = 21$$

$$D_2((a,b),d) = min(D_1(a,d), D_1(b,d)) = min(31,34) = 31$$

$$D_2((a,b),e) = min(D_1(a,e), D_1(b,e)) = min(23,21) = 21$$

	(a,b)	c	d	e
(a,b)	0	**21**	**31**	**21**
c	**21**	0	*28*	*39*
d	**31**	*28*	0	*43*
e	**21**	*39*	*43*	0

จากเมตริก D2 พบว่า ((a,b),c) และ ((a,b),e) อยู่ใกล้กันมากที่สุด คือ 21 จากนั้นทำการรวมกลุ่มเข้าด้วยกันได้ดังนี้ ((a,b),c,e)

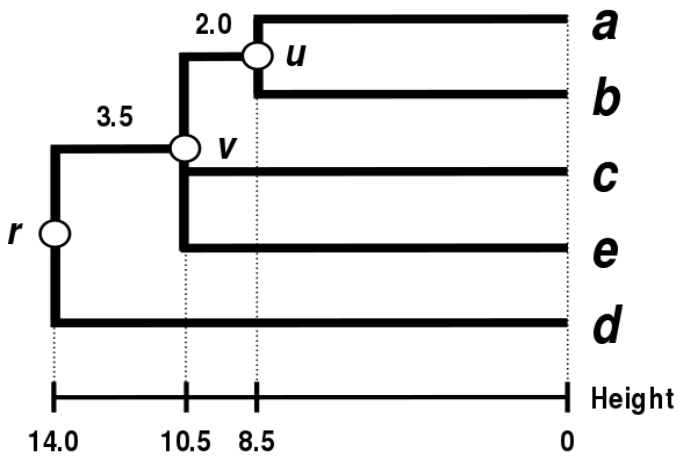
หมายเหตุ : ความสูงของ Dendrogram จะใช้ค่า 21/2=10.5 แทน 21

ทำการคำนวณระยะทางระหว่างกลุ่ม ((a,b),c,e) กับ d ดังนี้

$$D_3(((a,b),c,e),d) = min(D_2((a,b),d), D_2(c,d), D_2(e,d)) = min(31,28,43) = 28$$

	((a,b),c,e)	d
((a,b),c,e)	0	**28**
d	**28**	0

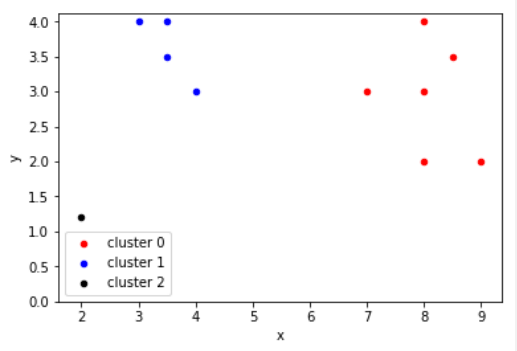
หมายเหตุ : ความสูงของ Dendrogram จะใช้ค่า 28/2=14 แทน 28

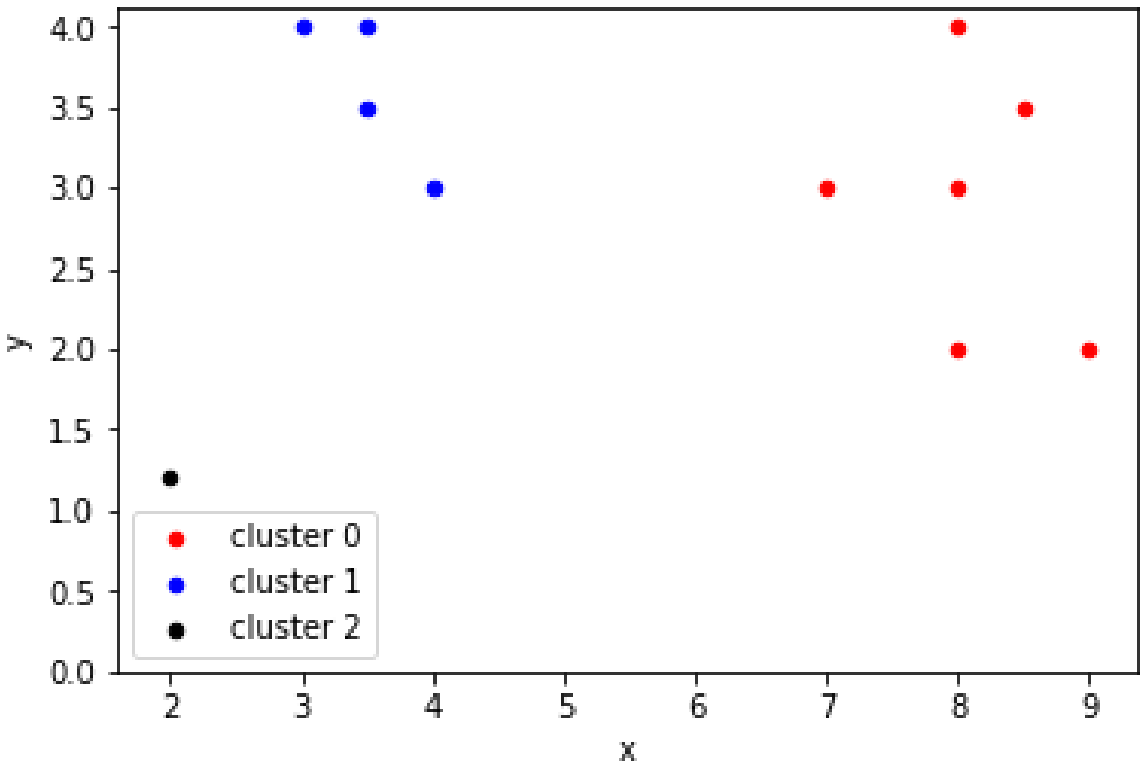


รูปแสดง Tree หรือ Dendrogram ที่ได้จาก Hierarchical Clustering

การเรียกใช้งาน Hierarchical Clustering จาก Scikit-learn

```
from sklearn.cluster import AgglomerativeClustering #import hierarchical clustering ligrary
import pandas as pd #import pandas library for data analysis (plot)

data=pd.DataFrame([[2, 1.2],[3, 4], [4, 3],[3.5, 3.5],[3.5, 4],[4, 5],
                   [2.5, 4.5],[3, 5],[8, 2],[7, 3],[8, 4],[8.5, 3.5],
                   [9, 2], [8, 3]],
                  columns=['x','y'])

clustering = AgglomerativeClustering(linkage='single',n_clusters=3).fit(data) #find clusters

#visualize data
data['cluster']=clustering.labels_ #given clusters
cluster0=data.loc[data['cluster'] == 0]
cluster1=data.loc[data['cluster'] == 1]
cluster2=data.loc[data['cluster'] == 2]

ax=cluster0.plot.scatter(x='x', y='y',
                         ylim=0, color='Red', label='cluster 0')#plot cluster 0
ax=cluster1.plot.scatter(x='x', y='y',
                         ylim=0, color='Blue', label='cluster 1', ax=ax)#plot cluster 1
ax=cluster2.plot.scatter(x='x', y='y',
                         ylim=0, color='Black', label='cluster 2', ax=ax)#plot cluster 2
```





รูปแสดงการจัดกลุ่มข้อมูลด้วย Hierarchical Clustering Technique

การเรียนรู้ของเครื่องจักรสำหรับงานแบ่งกลุ่มข้อมูล

เทคนิคการเรียนรู้ของเครื่องจักรสำหรับงานแบ่งกลุ่มข้อมูล (Classification Task) เป็นรูปแบบหนึ่งของเทคนิคการเรียนรู้ของเครื่องจักร ที่มีการสอนหรือมีเฉลย และค่าเป้าหมายหรือเฉลยเป็นค่าไม่ต่อเนื่อง (Discrete Value) การแบ่งกลุ่มข้อมูลมีทั้งการแบ่งข้อมูลออกเป็น 2 กลุ่ม (Binary Classification) และการแบ่งข้อมูลหลายกลุ่ม (Multi-Classification)

ตัวอย่างการนำไปประยุกต์ใช้ ดังนี้

- การแยก Spam Mail กับ Non-spam Mail ของ Mail Server
- การแยก คน สัตว์ สิ่งของ หรือรถ ใน Self-Car Driving
- การยืนยันตัวตนโดยการสแกนใบหน้า เช่น Face ID ของ iPhone
- การอนุมัติเงินกู้หรือบัตรเครดิต
- การทำนายอารมณ์ของพนักงานหรือลูกค้า (Sad/Neutral/Happy)
- การทำนายผลฟุตบอล (Win/Tie/Loss)
- การทำนายการเกิดโรค (เป็น/ไม่เป็น)

ตัวอย่างเทคนิค ดังนี้

- KNN
- Neural Network (MLP/RBFN/CNN/RNN)
- Naïve Bays/SVM/CART/Logit

K Nearest Neighbours (KNN)

KNN ทำการทำนายโดยอาศัยหลักการวัดระยะทาง (โดยปกติใช้ Euclidian Distance Method) หรือ ความคล้ายกัน หรือความต่างกัน ระหว่าง Test data และ Train data แล้วใช้ค่า Target หรือ Dependent Variable จาก Train data ที่มีความต่างกันน้อยที่สุดจำนวน *k* Data Points เพื่อคำนวณผลลัพธ์ของการทำนาย

KNN Algorithm

Nearest neighbour algorithm to classify a vector $\mathbf{x}$, given train data $\mathcal{D} = \{(\mathbf{x}^n, c^n), n = 1, \ldots, N\}$

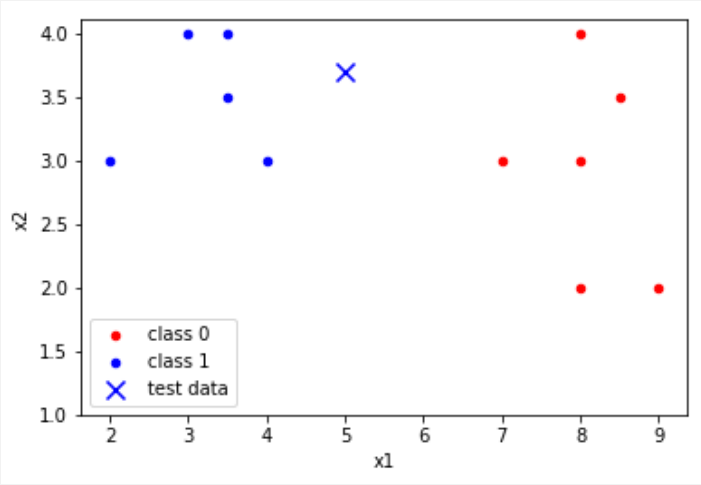
1: Calculate the dissimilarity of the test point $\mathbf{x}$ to each of the train points, $d^n = d(\mathbf{x}, \mathbf{x}^n)$, $n = 1, \ldots, N$.
2: Find the train point $\mathbf{x}^{n^*}$ which is nearest to $\mathbf{x}$:

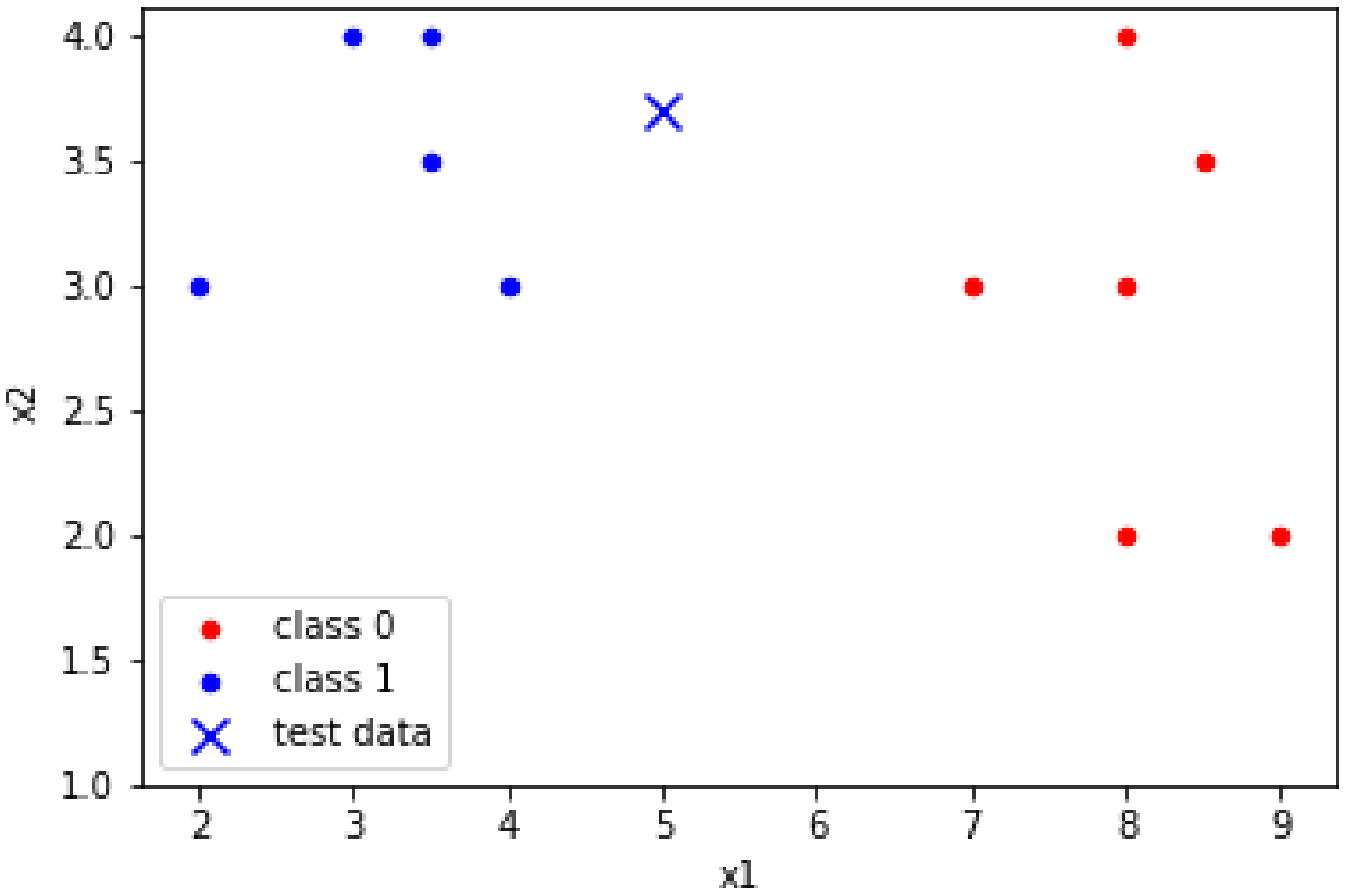
$$n^* = \underset{n}{\operatorname{argmin}}\, d(\mathbf{x}, \mathbf{x}^n)$$

3: Assign the class label $c(\mathbf{x}) = c^{n^*}$.
4: In the case that there are two or more nearest neighbours with different class labels, the most numerous class is chosen. If there is no one single most numerous class, we use the K-nearest-neighbours.

การเรียกใช้งาน KNN จาก Scikit-learn

```
from sklearn.neighbors import KNeighborsClassifier #import knn ligrary
import pandas as pd #import pandas library for data analysis (plot)

train_data=pd.DataFrame([[2.0,3.0,1],[3.0,4.0,1],[4.0,3.0,1],[3.5,3.5,1],[3.5,4.0,1],[4.0,5.0,1],[2.5,4.
5,1],[3.0,5.0,1],[8.0,2.0,0],[7.0,3.0,0],[8.0,4.0,0],[8.5,3.5,0], [9.0,2.0,0],[8.0,3.0,0]],
            columns=['x1','x2','class'])
test_data=pd.DataFrame([[5,3.7]],columns=['x1','x2'])
neigh = KNeighborsClassifier(n_neighbors=2) #setup parameters
neigh.fit(train_data.loc[:,['x1','x2']],train_data['class']) #train data
y=neigh.predict(test_data) #predict a class

#visualize data
if y[0]==0:
    test_color='Red'
else:
    test_color='Blue'

cluster0=train_data.loc[train_data['class'] == 0]
cluster1=train_data.loc[train_data['class'] == 1]
ax=cluster0.plot.scatter(x='x1', y='x2',
                         ylim=1, color='Red', label='class 0')#plot cluster 0
ax=cluster1.plot.scatter(x='x1', y='x2',
                         ylim=1, color='Blue', label='class 1', ax=ax)#plot cluster 0
test_data.plot.scatter(x='x1', y='x2',
                  color=test_color, marker='x', s=100, label='test data', ax=ax)#plot test data
```





รูปแสดงการจัดกลุ่มข้อมูลด้วย KNN technique

แบบฝึกหัด

1. ให้นักศึกษาค้นหาตัวอย่างชุดข้อมูล (Dataset) สำหรับงาน Clustering งาน Classification และงาน Regression จากฐานข้อมูล UCI (https://archive.ics.uci.edu/ml/datasets.php) มางานละ 1 ชุดข้อมูล

2. ให้นักศึกษาพัฒนา K-mean และ Competitive Learning ด้วยภาษา Python โดยใช้ข้อมูลสำหรับงาน Clustering ที่ได้จากข้อ 1

3. ให้นักศึกษาพัฒนา KNN ด้วยภาษา Python โดยใช้ข้อมูลสำหรับงาน Classification ที่ได้จากข้อ 1

4. ให้นักศึกษาใช้ RapidMiner หรือ Weka เพื่อ Train Model โดยใช้ Multilayer Perceptron แล้วนำ Model ที่ได้ไปพัฒนา ด้วยภาษา Python เพื่อใช้สำหรับการทดสอบ โดยใช้ข้อมูลสำหรับงาน Regression ที่ได้จากข้อ 1