K-means 分群

CHAPTER 8

K-means 屬於非監督式學習，用於分群問題。隨機挑選 K 個點，將所有點劃分於這 K 個點中最靠近的點來進行分群。接著重新計算各群中各點的平均為新的 K 個中心點，所有點再依照新的中心點分群，不斷重複直到 K 個中心點不再移動。利用物以類聚的概念進行分群，表示同一群的資料點彼此之間應該越靠近。

8-1 K-means 分群的運作過程

K-means 分群的運作過程如下：

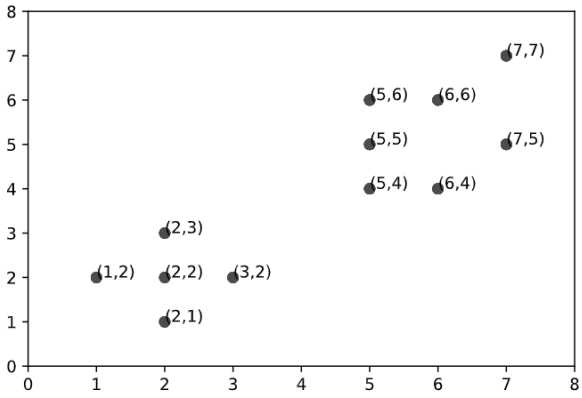
step01 隨機挑選 K 個點為中心點。

step02 找到所有點到這 K 個中心點的距離，每一個點被分到最接近的中心點。

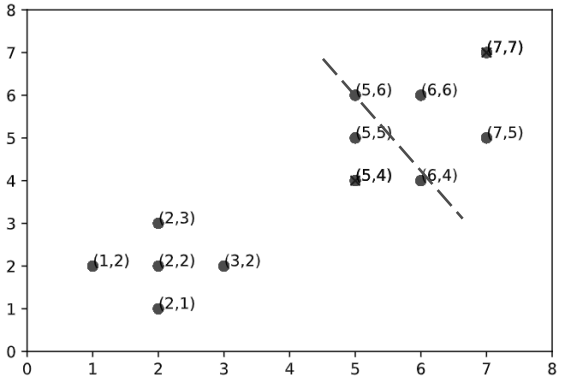
step03 重新計算每一群內各點的平均值為新的中心點。

step04 重複 step02 與 step03，直到中心點不再移動，或移動距離很短、或重複找中心點達到指定次數。

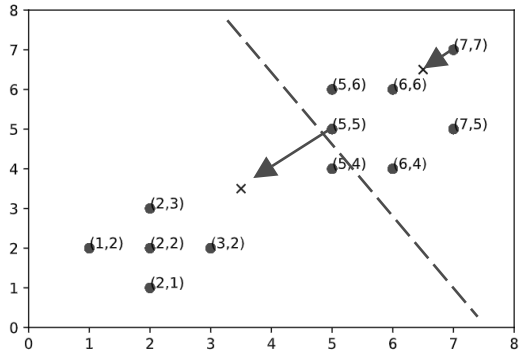
將以下 12 個點使用 K-means 分成兩群：



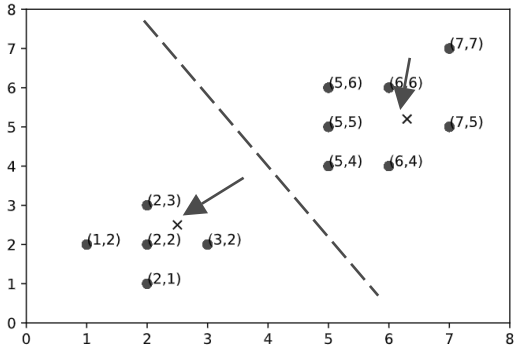
隨機挑選(5, 4)與(7, 7)為中心點，分成兩群，距離中心點越近者分成同一群，以斜線表示兩群的範圍。



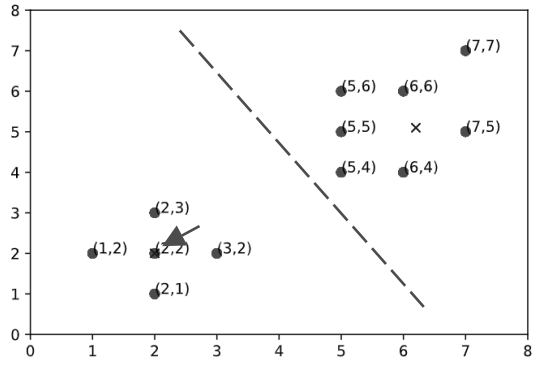
重新計算中心點，假設以(3.5, 3.5)與(6.4, 6.2)為中心點分成兩群，以斜線表示兩群的範圍。



重新計算中心點，假設以(2.5, 2.5)與(6.2, 5.1)為中心點分成兩群，以斜線表示兩群的範圍。



重新計算中心點，假設以(2, 2)與(6.2, 5.1)為中心點分成兩群，以斜線表示兩群的範圍。



重新計算中心點，可以發現還是以(2, 2)與(6.2, 5.1)為中心點，不再改變，到此完成使用 K-means 演算法將 12 個點分成兩群。

8-2 使用 sklearn 實作 K-means 分群

使用 sklearn 實作 K-means 分群，只要將每筆資料輸入 K-means 分群模型，就可計算出每筆資料的所在群組。使用 sklearn 實作 K-means 分群的步驟如下。

(1) 輸入資料

```
X = data
```

(2) 建立模型

```
model = KMeans(init = "k-means++", n_clusters = 3)
```

參數 init 表示初始化中心點的方式，參數 n_cluster 表示要分成幾群。

(3) 訓練模型與模型預測

```
model.fit(X)
model.predict(X)
```

8-3 使用 K-means 分群實作範例

8-3-1 使用 K-means 對消費者進行分群

【8-3-1 使用 K-means 對消費者進行分群.ipynb】使用性別、居住地、收入與花費等對消費者分群，本範例測資來自於 Kaggle 網站，可以從以下網址下載 ClusteringHSS.csv。

```
https://www.kaggle.com/harrimansaragih/clustering-data-id-gender-income-spending
```

step01 匯入資料

匯入 ClusteringHSS.csv 到 DataFrame。

行數	程式碼
1	`import pandas as pd`
2	`import matplotlib.pyplot as plt`
3	`from sklearn.cluster import KMeans`
4	`from sklearn.preprocessing import LabelEncoder`
5	`df = pd.read_csv("E:/data/ClusteringHSS.csv")`
6	`print(df.head())`

程式說明

- 第 1 到 4 行：匯入函式庫。
- 第 5 行：讀取 ClusteringHSS.csv 轉換成 DataFrame，變數 df 參考到此結果。
- 第 6 行：使用函式 head 顯示變數 df 的前五筆資料。

執行結果

```
   ID Gender_Code Region  Income  Spending
0   1      Female  Rural    20.0      15.0
1   2        Male  Rural     5.0      12.0
2   3      Female  Urban    28.0      18.0
3   4        Male  Urban    40.0      10.0
4   5        Male  Urban    42.0       9.0
```

step02 檢查資料

檢查資料是否有空值、資料維度大小、欄位名稱、第一筆資料內容與資料型別。

行數	程式碼
1	`print(df.isnull().values.sum())`
2	`print("資料筆數:", df.shape)`
3	`print("資料的欄位名稱，分別是:", df.keys())`
4	`print("第一筆的資料內容:", df.iloc[0,::])`
5	`print(df.dtypes)`

程式說明

- 第 1 行：使用函式 isnull 檢查資料是否有空值（NaN），如果欄位有空值就會回傳 True，轉換成數值 1，加總結果就會是空值個數。
- 第 2 到 4 行：使用 shape 顯示資料的筆數，函式 keys 顯示欄位名稱，iloc 顯示指定範圍的資料內容。
- 第 5 行：顯示資料集 df 每個欄位的資料型別。

執行結果

發現有 23 個 NaN。

```
23
資料筆數: (1113, 5)
資料的欄位名稱，分別是: Index(['ID', 'Gender_Code', 'Region', 'Income', 'Spending'], dtype='object')
第一筆的資料內容: ID                1
Gender_Code    Female
Region          Rural
Income             20
Spending           15
Name: 0, dtype: object
ID              int64
Gender_Code    object
Region         object
Income        float64
Spending      float64
dtype: object
```

step03 整理資料

刪除不需要的欄位。

行數	程式碼
1	`df = df.dropna()`
2	`LE = LabelEncoder()`
3	`df['Gender_Code'] = LE.fit_transform(df['Gender_Code'])`

行數	程式碼
4	`LE = LabelEncoder()`
5	`df['Region'] = LE.fit_transform(df['Region'])`
6	`X = df.drop(['ID'],axis=1)`
7	`print(X.head())`

程式說明

- 第 1 行：使用函式 dropna 刪除資料集 df 所有空值。
- 第 2 到 5 行：使用 LabelEncoder 將欄位 Gender_Code 與 Region 由文字轉換成數值。
- 第 6 行：對於分群沒有幫助，刪除欄位 ID。
- 第 7 行：使用函式 head 顯示變數 X 的前五筆資料。

執行結果

	Gender_Code	Region	Income	Spending
0	0	0	20.0	15.0
1	1	0	5.0	12.0
2	0	1	28.0	18.0
3	1	1	40.0	10.0
4	1	1	42.0	9.0

step04 建立與訓練模型

使用 KMeans 建立模型與訓練模型。

行數	程式碼
1	`model = KMeans(init = "k-means++", n_clusters = 5)`
2	`model.fit(X)`
3	`print(model.cluster_centers_)`

程式說明

- 第 1 行：使用 KMeans 建立 K-means 分群，設定 init 為 k-means++，表示選擇 k-means++演算法，設定 n_clusters 為 5，表示分成 5 群，指定給變數 model。

- 第 2 到 3 行：輸入 X 到 KMeans 模型進行訓練，顯示這 5 群的中心點。

執行結果

```
[[ 5.01930502e-01  2.89575290e-01  2.31544402e+01  1.08301158e+01]
 [ 5.61702128e-01  1.00000000e+00  4.51617021e+01  1.14595745e+01]
 [ 5.14893617e-01 -1.66533454e-16  1.04723404e+01  8.25957447e+00]
 [ 5.50660793e-01  1.00000000e+00  3.35418502e+01  1.13348018e+01]
 [ 4.17910448e-01 -1.11022302e-16  1.25149254e+01  1.66865672e+01]]
```

step05 模型預測

輸入資料進行模型預測。

行數	程式碼
1	`model.predict(X)`

程式說明

- 第 1 行：以 X 為輸入，使用函式 predict 進行模型預測。

執行結果

```
array([0, 2, 3, ..., 3, 1, 0])
```

step06 資料表新增分群結果

在資料表新增分群結果欄位，並且進行分析。

行數	程式碼
1	`df["群"] = model.labels_`
2	`print(df.groupby('群').mean())`

程式說明

- 第 1 行：使用模型預估分群結果到資料集 df 的欄位「群」。
- 第 2 行：使用欄位「群」來群組化輸入資料，並求每一群的平均值。

執行結果

群	Gender_Code	Region	Income	Spending
0	0.503846	0.288462	23.134615	10.842308
1	0.561702	1.000000	45.161702	11.459574
2	0.514894	0.000000	10.472340	8.259574
3	0.550661	1.000000	33.541850	11.334802
4	0.413534	0.000000	12.473684	16.706767

step07 繪製散布圖

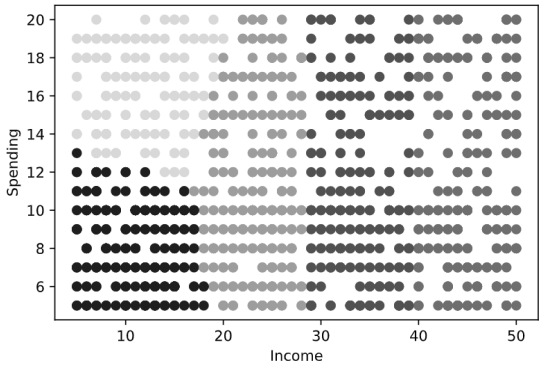
繪製收入、花費、所屬群組的散布圖。

行數	程式碼
1	`plt.scatter(X['Income'], X['Spending'], c=model.labels_)`
2	`plt.xlabel('Income')`
3	`plt.ylabel('Spending')`
4	`plt.show()`

程式說明

- 第 1 行：使用函式 scatter 繪製散佈圖，以 X['Income']為 X 軸座標值，X['Spending']為 Y 軸座標值，設定顏色 c 為 model.labels。
- 第 2 到 4 行：設定 X 座標軸標籤為 Income，設定 Y 座標軸標籤為 Spending，顯示散佈圖到螢幕上。

執行結果



step08 使用模組的屬性 inertia_決定分群個數

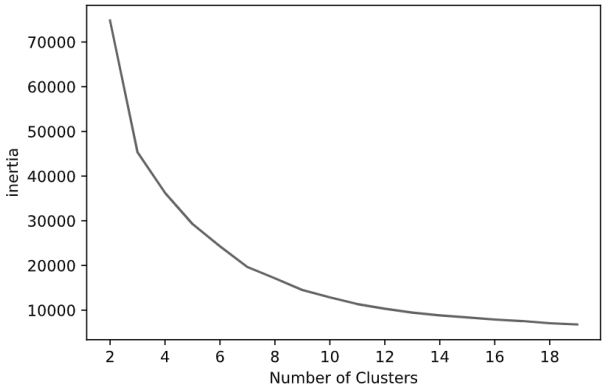
模組的屬性 inertia_表示每個樣本點距離該群中心點距離的平方和，當 inertia_的值越小，表示每個點距離該群中心點越近。當分群個數（n_clusers）越大時，則 inertia_數值一定會越小，當分群個數大到某個值以後，inertia_數值下降幅度會越來越不明顯，就可以選用此臨界值當成分群個數。

```
行數  程式碼
1     iner = []
2     for i in range(2,20):
3         model = KMeans(init = "k-means++", n_clusters = i)
4         model.fit(X)
5         iner.append(model.inertia_)
6     plt.plot(range(2, 20), iner)
7     plt.xlabel('Number of Clusters')
8     plt.ylabel('inertia')
9     plt.xticks(range(2,20,2))
10    plt.show()
```

程式說明

- 第 1 行：宣告變數 iner 為串列。
- 第 2 到 5 行：宣告迴圈變數 i，由 2 到 19 每次遞增 1，使用 KMeans 進行分群，迴圈變數 i 設定給參數 n_clusters，將訓練資料 X 輸入模組進行訓練，取得模組屬性 inertia_附加到變數 iner。
- 第 6 行：設定 X 軸為數值 2 到 19 每次遞增 1 的整數值，Y 軸為變數 iner。
- 第 7 到 10 行：設定 X 軸標題為 Number of Clusters，設定 Y 軸標題為 inertia，設定 X 軸座標點為 2 到 19 每次遞增 2，顯示繪圖結果到螢幕上。

執行結果



step09 重新分群

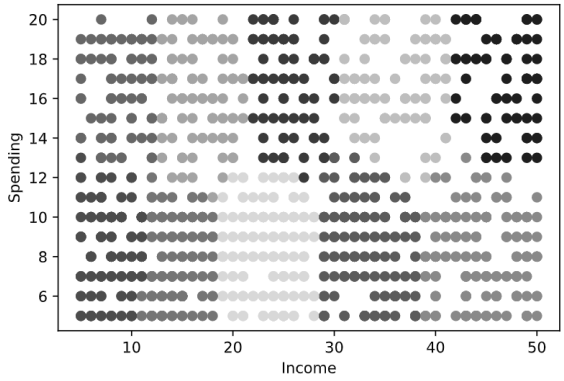
重新分成 10 個群，並繪製收入、花費、所屬群組的散布圖。

行數	程式碼
1	`model = KMeans(init = "k-means++", n_clusters = 10)`
2	`model.fit(X)`
3	`plt.scatter(X['Income'], X['Spending'], c=model.labels_)`
4	`plt.xlabel('Income')`
5	`plt.ylabel('Spending')`
6	`plt.show()`

程式說明

- 第 1 行：使用 KMeans 建立 K-means 分群，設定 init 為 k-means++，表示選擇 k-means++演算法，設定 n_clusters 為 10，表示分成 10 群，指定給變數 model。
- 第 2 行：輸入 X 到 KMeans 模型進行訓練。
- 第 3 行：使用函式 scatter 繪製散佈圖，以 X['Income']為 X 軸座標值，X['Spending']為 Y 軸座標值，設定顏色 c 為 model.labels。
- 第 4 到 6 行：設定 X 座標軸標籤為 Income，設定 Y 座標軸標籤為 Spending，顯示圖片到螢幕上。

執行結果



8-4 習題

一. 問答題

1. 舉例說明 K-means 模型運作過程。
2. 使用文字與程式寫出 K-means 模型的操作步驟。

二. 實作題

對 Kaggle 資料集進行分群

從以下網址下載資料檔 voted-kaggle-dataset.csv。

```
https://www.kaggle.com/canggih/voted-kaggle-dataset
```

以下為 Kaggle 資料集的前五筆資料：

```
                                      Tags Data Type    Size License  \
0                            crime\nfinance       CSV  144 MB    ODbL
1                association football\neurope    SQLite  299 MB    ODbL
2                                        film       CSV   44 MB   Other
3  crime\nterrorism\ninternational relations       CSV  144 MB   Other
4                            history\nfinance       CSV  119 MB     CC4

           Views          Download        Kernels     Topics  \
0  442,136 views  53,128 downloads  1,782 kernels  26 topics
1  396,214 views  46,367 downloads  1,459 kernels  75 topics
2  446,255 views  62,002 downloads  1,394 kernels  46 topics
3  187,877 views  26,309 downloads    608 kernels  11 topics
4  146,734 views  16,868 downloads     68 kernels  13 topics
```

建立一個 K-means 模型，對 Kaggle 資料集進行分群，撰寫程式完成以下功能。

1. 匯入資料檔 voted-kaggle-dataset.csv 到一個 DataFrame。
2. 檢查與統計資料
   (1) 檢查是否有空值、資料筆數、欄位名稱、第一筆資料內容。
   (2) 檢查每個欄位的資料型別與空值個數，請寫出非數值的欄位名稱。
3. 產生訓練資料集與測試資料集
   (1) 資料集只保留欄位 Votes、Data Type、License、Views 與 Download，並刪除所有空值。
   (2) 因為欄位 Views 與 Download 的內容為「442,136 views」與「53,128 downloads」，使用以下程式碼轉換成數值。

```
a = X['Views'].str.split(' ')
X['Views'] = pd.Series([int(a[i][0].replace(',','')) for i in range(len(a))])
```

   (3) 非數值的欄位 Data Type 與 License，請使用 LabelEncoder 轉換成數值。
4. 請使用 KMeans 建立 K-means 模型，並輸入訓練資料 X 進行訓練。

5. 分群結果

   (1) 在原始資料集新增一個欄位為「群」，將每一筆輸入資料的分群結果記錄在此欄位。

   (2) 繪製「Votes」、「Views」與所屬群組的分布圖。

6. 找出適合的分群個數

   (1) 使用迴圈設定 n_clusters 的數值由 2 到 19，每次遞增 1，每次代入輸入資料 X 到 K-means 模型，計算模型的 inertia_，繪製成折線圖。

   (2) 找出最適合的 n_clusters 值，使用此 n_clusters 值重新建立模型與訓練模型，重新繪製「Votes」、「Views」與所屬群組的散佈圖。