

PYTHON 實作

資料處理與分析




陳君毅



chunyi1999@cdc.gov.tw





目錄

大綱

資料處理

資料分析

案例分析

總結

參考資料





大綱

資料處理全流程概覽



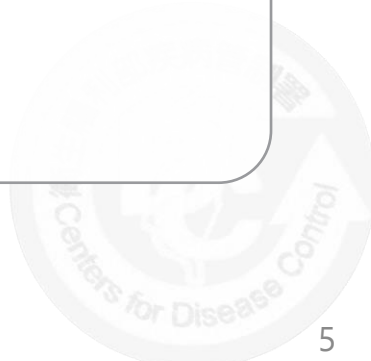
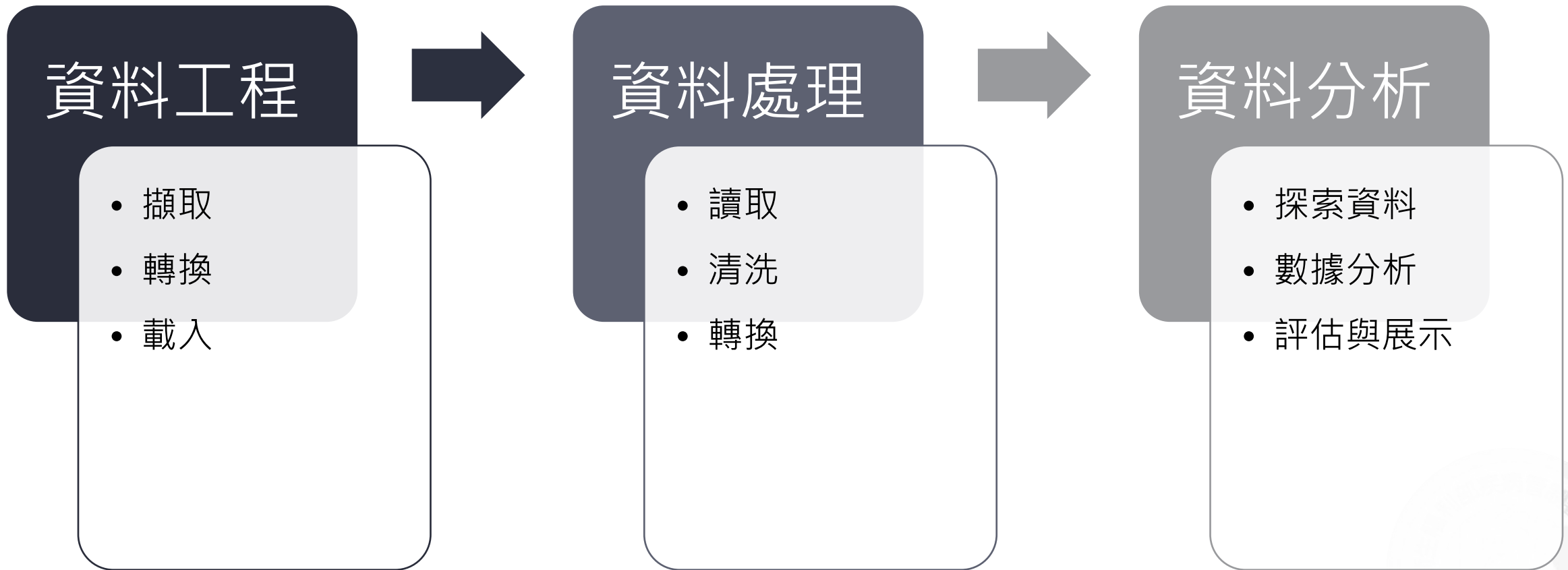


前言

資料分析的源頭

在當今數位時代，資料已成為各行各業的寶貴資產。Python作為一種強大而靈活的程式語言，在資料處理與分析領域中扮演著關鍵角色。本文將概述使用Python進行資料處理與分析的主要流程，提供系統性的理解。

資料分析流程圖



工具選擇



SQL

- 資料庫語言
- ETL

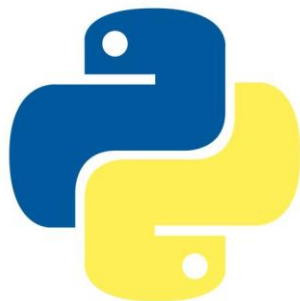


SAS

- 資料分析
- 授權費高、生物統計分析

Python

- 資料處理、資料分析
- 膠水語言



Excel

- 資料分析、視覺化
- 容易上手



R

- 資料分析、視覺化
- 統計分析



Looker Studio

Looker Studio

- 互動式資料視覺化面板
- 整合 GCP、Google 相關文件

PYTHON 常用套件



Pandas

- 數據表格處理的利器
- 表格處理、資料轉換



Scikit-learn

- 統計與機器學習的建模工具
- 建模的首選套件

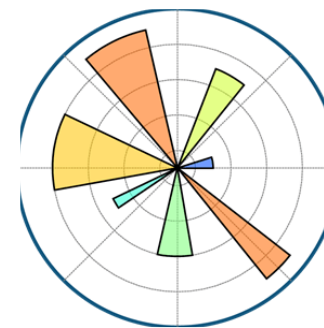
Numpy

- 大型數列運算的工具
- 高維度資料運算



Matplotlib

- 繪圖工具
- 簡易的數據視覺化



Scipy

- 科學、工程運算工具
- 統計、優化、整合、線性代數



Seaborn

- 繪圖工具
- 更精美的數據視覺化



資料處理

資料分析前的預處理

章節大綱

資料讀取

- 檔案格式
- 讀取方式
- 欄位選取

資料清洗

- 變數結構
- 遺失值
- 重複值
- 異常值

資料轉換

- 正規化/標準化
- 虛擬變數
- 表格合併

資料讀取 – 檔案格式與編碼

逗號分隔值、文字檔案

- CSV
- TXT

EXCEL相關檔案

- XLS, XLSX

SQL資料庫

- Oracle, MSSQL, Postgres, MySQL

API或爬蟲相關

- JSON



資料讀取

檔案格式

讀取方式

欄位選取



資料讀取 – 讀取方式

使用 Python 中的 pandas 套件讀取 .txt 與 .csv 檔案：

CODE

```
import pandas as pd
txt_data = pd.read_csv("data/demo1.txt", sep='\t')
csv_data = pd.read_csv("data/demo2.csv")
```

讀取後的資料會以 pandas 的 DataFrame 儲存，日後可以使用 pandas 相關模組對資料進行進一步的處理

CODE

```
# 資料檢視
txt_data.describe
txt_data.head(); txt_data.tail()
```

資料讀取

檔案格式

讀取方式

欄位選取



資料讀取 – 欄位選取

使用 Python 中 pandas 套件的 `.loc[row, column]` 與 `.iloc[row, column]`

CODE

```
import pandas as pd
txt_data = pd.read_csv("data/demo1.txt", sep='\t')
csv_data = pd.read_csv("data/demo2.csv")
```


資料清洗 – 變數結構

使用 Python 中的 pandas 查看資料的結構：

CODE

```
# 資料總數
txt_data.size

# 資料維度資訊
txt_data.ndim      # 維度數目
txt_data.shape     # 維度長度

# 變數型態
txt_data.dtypes

# 資料檢視
txt_data.describe
txt_data.info
```

資料清洗 – 尋找遺失值

遺失值通常是在問卷類型時因為拒訪、遺漏填寫或是調查員疏失所出現的空值，也有是資料工程轉換時出現的錯誤造成資料遺失或空值。進行數據分析要確保資料的完整性，因此要對資料進行遺失值處理。

CODE

```
# 尋找遺失值  
txt_data_missing.isnull()  
  
# 遺失值數量統計  
txt_data_missing.isnull().sum()
```

資料清洗 – 遺失值處理

當了解變數結構與遺失值的數量後，要將遺失值做相對應的處理，如：

- 填充：替換變數，如 0、平均值、中位數等。
- 刪除：刪除缺失值所在的樣本。

CODE

```
# 遺失值填補為 0
txt_data_clean = txt_data_missing.fillna(0)

# 遺失值填補平均值
hearing_maen = txt_data_missing['Hearing'].mean()
txt_data_clean = txt_data_missing.fillna(hearing_maen)

# 刪除 NA 所在之觀察值
txt_data_clean = txt_data_missing.dropna()
```

資料清洗 – 重複值處理

通常資料上以列作為個別的觀察值，因此當兩列以上有出現完全一樣的資料時，認為資料是有重複的情況，若確定其資料是在原始資料出現重複紀錄時要進行重複值處理，使用 Python 中的 pandas 處理重複值。

CODE

```
txt_data1 = txt_data.drop_duplicates()
```


資料清洗 – 異常值檢測

通過統計方法識別異常值，並對其進行修改或刪除。如標準差或四分位距（IQR）檢測異常值。

CODE

```
#####  
# 標準差  
#####  
txt_data.std()  
  
#####  
# 四分位距(IQR)  
#####  
# 自定義IQR程式碼  
def fetch_IQR(data:pd.DataFrame, col:str):  
    return data[col].quantile(.75)-data[col].quantile(.25)  
  
# txt_data 中 Hearing 的 IQR  
fetch_IQR(data = txt_data, col='Hearing')
```

資料轉換 – 資料正規/標準化

資料通常在進行模型建置時訓練時，為消除變數間不同單位對於分析結果的影響，因此資料分析時會因應模型假設或原始的資料結構做出對應的資料正規化或標準化，常見方法如下：

- Min-Max normalization
- Z-score standardization

CODE

```
from sklearn.preprocessing import MinMaxScaler, StandardScaler
txt_data_trans = txt_data.copy()

# 資料正規化
x_scale_norm = txt_data_trans.loc[:, ['Hearing']]
x_scale_norm = MinMaxScaler().fit_transform(x_scale_norm)

# 資料標準化
x_scale_std = txt_data_trans.loc[:, ['Hearing']]
x_scale_std = StandardScaler().fit_transform(x_scale_std)
```

資料轉換 – 虛擬變數

資料中若存在類別型變數，在進行資料分析之前需要類別變數轉換成虛擬變數(dummy variable)

ID	Var		ID	Var_A	Var_B	Var_C
1	A	➔	1	1	0	0
2	B		2	0	1	0
3	B		3	0	1	0
4	C		4	0	0	1

CODE

```
txt_data_dummy = txt_data.copy()

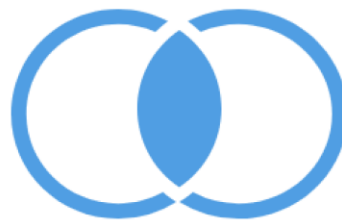
# 將 ListID 轉化成虛擬變數
pd.get_dummies(txt_data_dummy, prefix=['ListID'])
```

資料轉換 – 表格合併

同一專案中可能會有不同來源的資料，若兩資料表中有一定的關聯性主鍵(Key)，可以將資料適度的串連起來分析



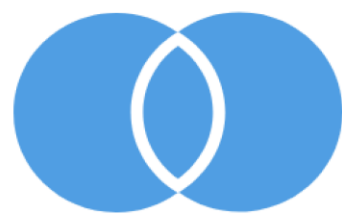
Left outer join



Inner join



Right outer join



Full outer join

資料來源：<https://ithelp.ithome.com.tw/articles/10305644?sc=rss.iron>

核心技術應用說明

章節大綱

資料探索

- 描述性統計
- 探索性分析
- 樞紐分析表

數據分析

- 分析模式
- 模型建立
- 模型評估

成果展示

- 數據解讀
- 視覺化呈現



資料探索 – 描述性統計

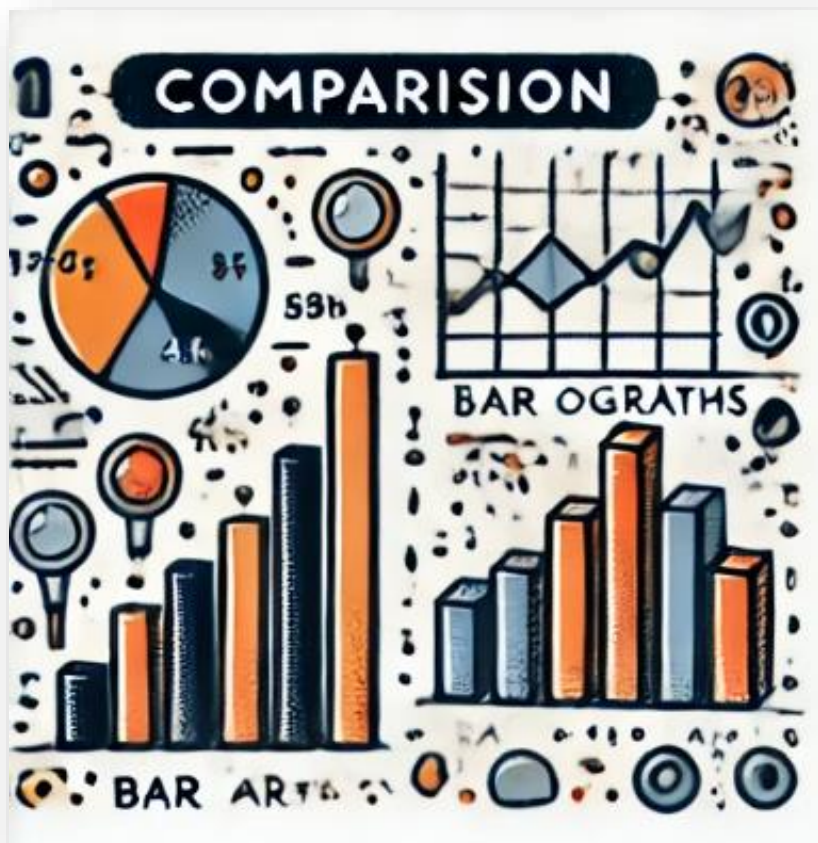
資料清洗完後，會對於資料進行初步的認識，如：資料維度、資料分布、各變數統計量(統計5M)。

CODE

```
# 建立描述性分析所需的資料
def fetch_data_basic_info(data:pd.DataFrame, colname:str):
    basic_info = {
        'Mean' : data[colname].mean(),
        'StdDev' : data[colname].std(),
        'Min' : data[colname].min(),
        'Q1' : data[colname].quantile(0.25),
        'Median' : data[colname].median(),
        'Q3' : data[colname].quantile(0.75),
        'Max' : data[colname].max()
    }
    return basic_info

# 呈現RSV的結果
fetch_data_basic_info(data=df_lars_data, colname='RSV')
```

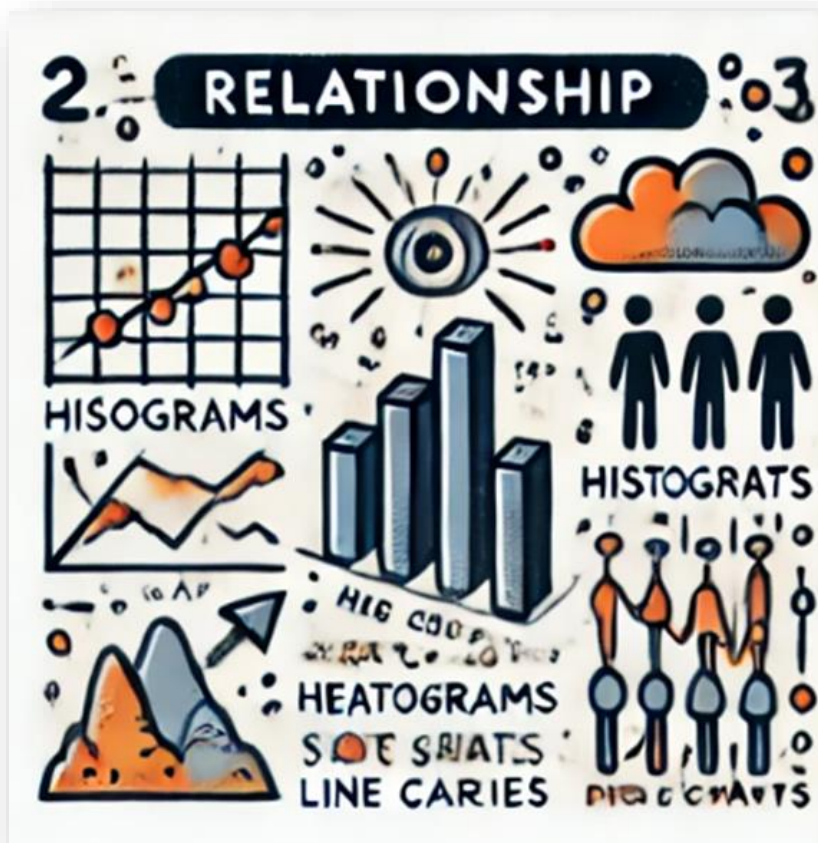
資料探索 – 探索性分析



Comparison (比較)

- 比較不同變量或分類之間的數據，包含，適合比較不同項目或時間段的資料
- 長條圖、直方圖、折線圖、時間序列圖

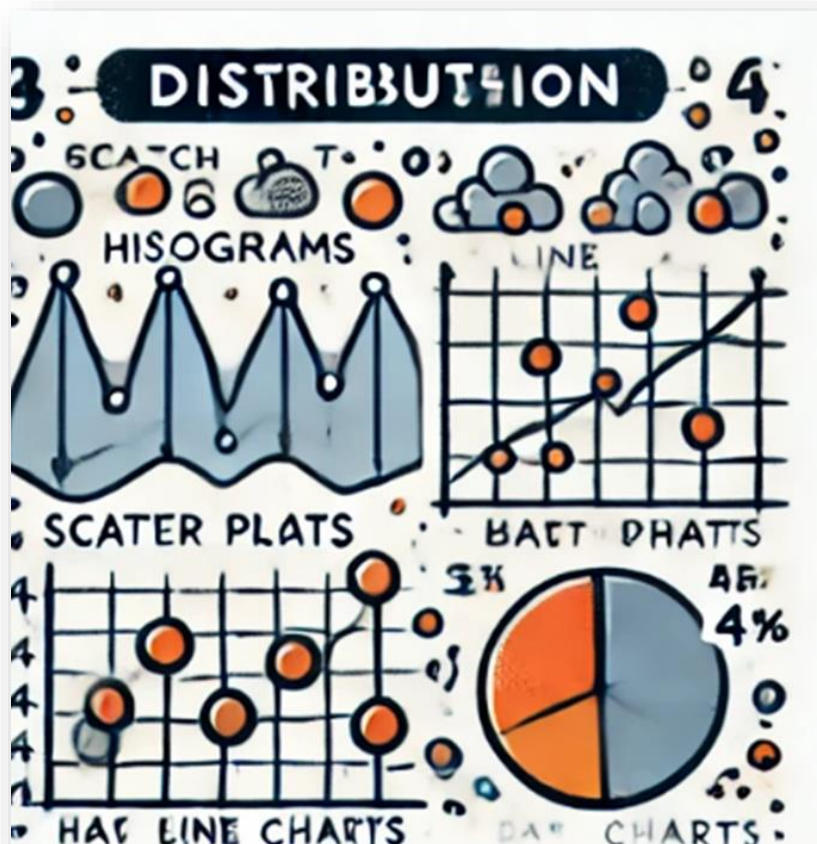
資料探索 – 探索性分析



Relationship (關係)

- 展示變量之間的關係，適合分析兩個或多個變量之間的關係
- 散佈圖、熱區圖和氣泡圖

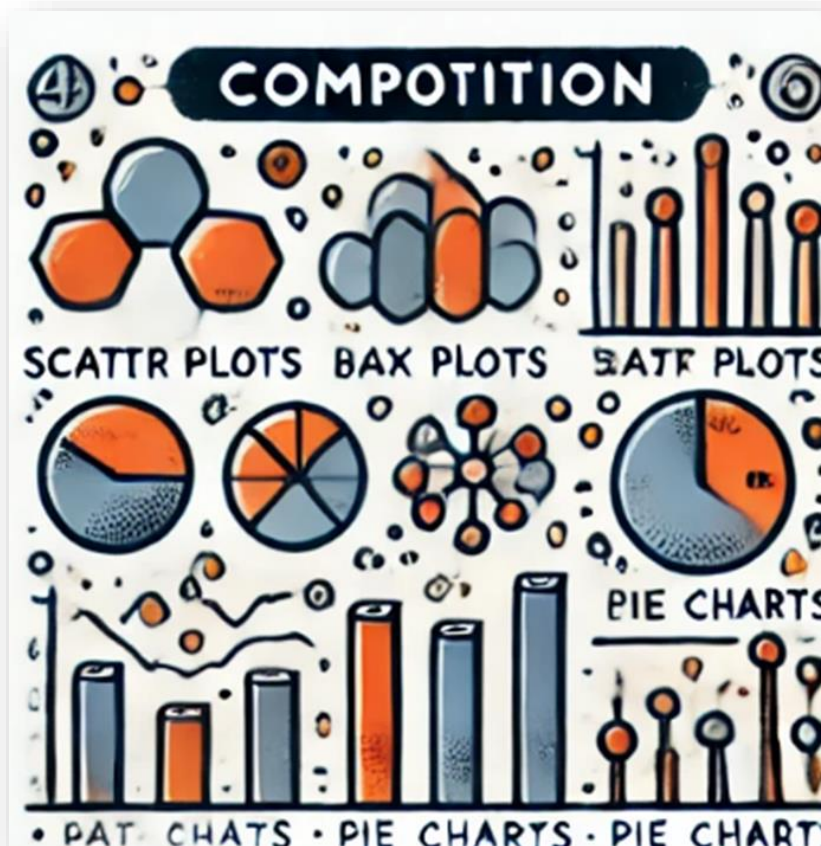
資料探索 – 探索性分析



Distribution (分佈)

- 顯示數據的分佈情況，適合展示單一變量或雙變量的數據分佈
- 直方圖、盒形圖、散佈圖

資料探索 – 探索性分析



Composition (組成)

- 展示數據構成的結構或比例，適合分析總體中的不同組成部分
- 堆疊柱狀圖、堆疊面積圖、圓餅圖、瀑布圖等

資料探索 – 樞紐分析表

探索性資料以數據呈現且處理的資料為原始觀察值時，可以使用樞紐分析表檢視資料，以 Python 的 pandas 中可以此用 pivot_table 或 groupby 並配合相對應的聚合函數 (aggregate function) 可以檢視資料的分組情況。

函數	分析情境	聚合函數
Pivot_table	雙維度	函數內建
groupby	單維度	另外連接

資料探索 – 樞紐分析表

Python pandas 中樞紐分析表(Pivot_table)函數：

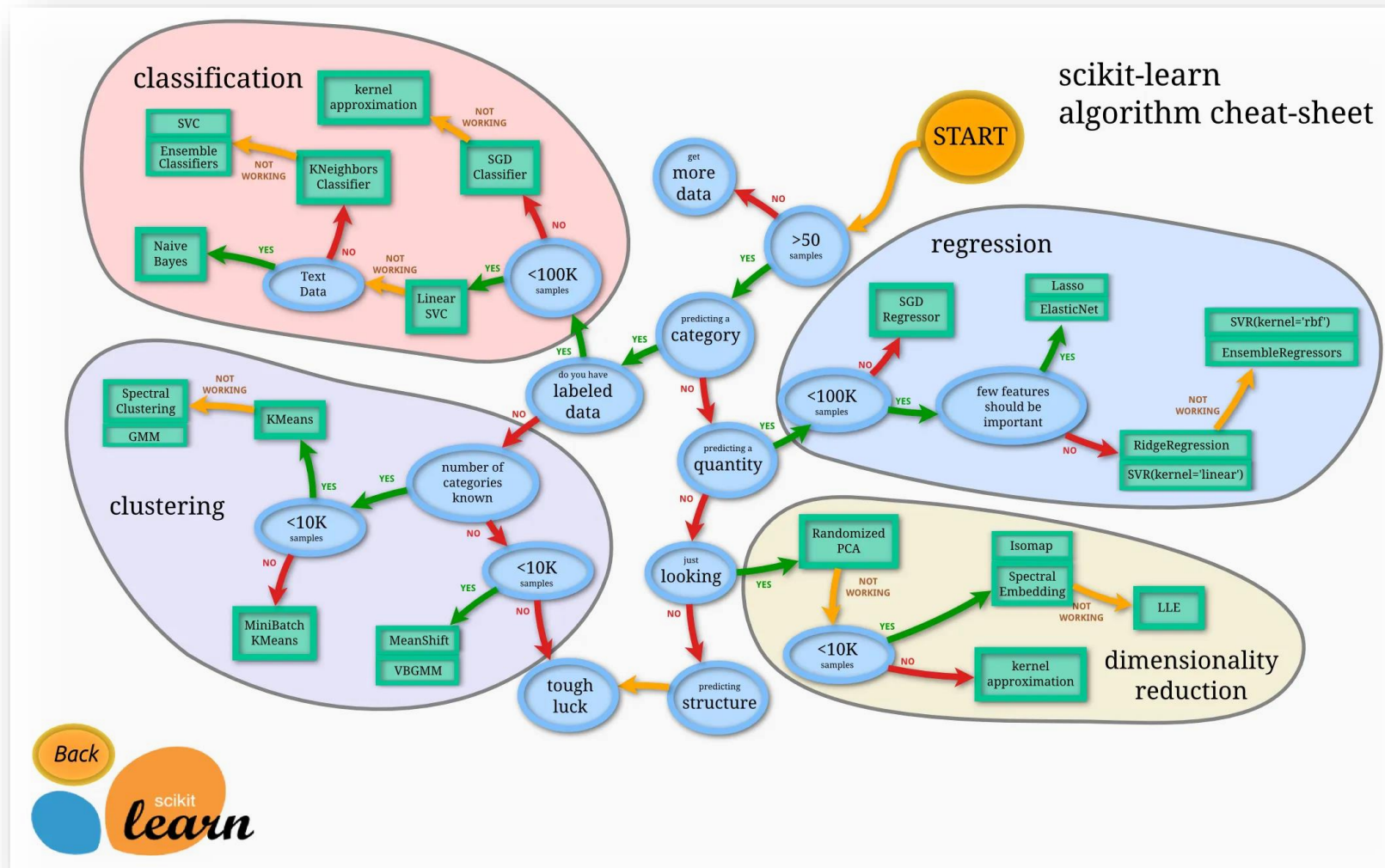
- data：資料名稱，需為 pd.DataFrame
- index：列(row)名稱
- column：欄(column)名稱
- aggfun：聚合函數

CODE

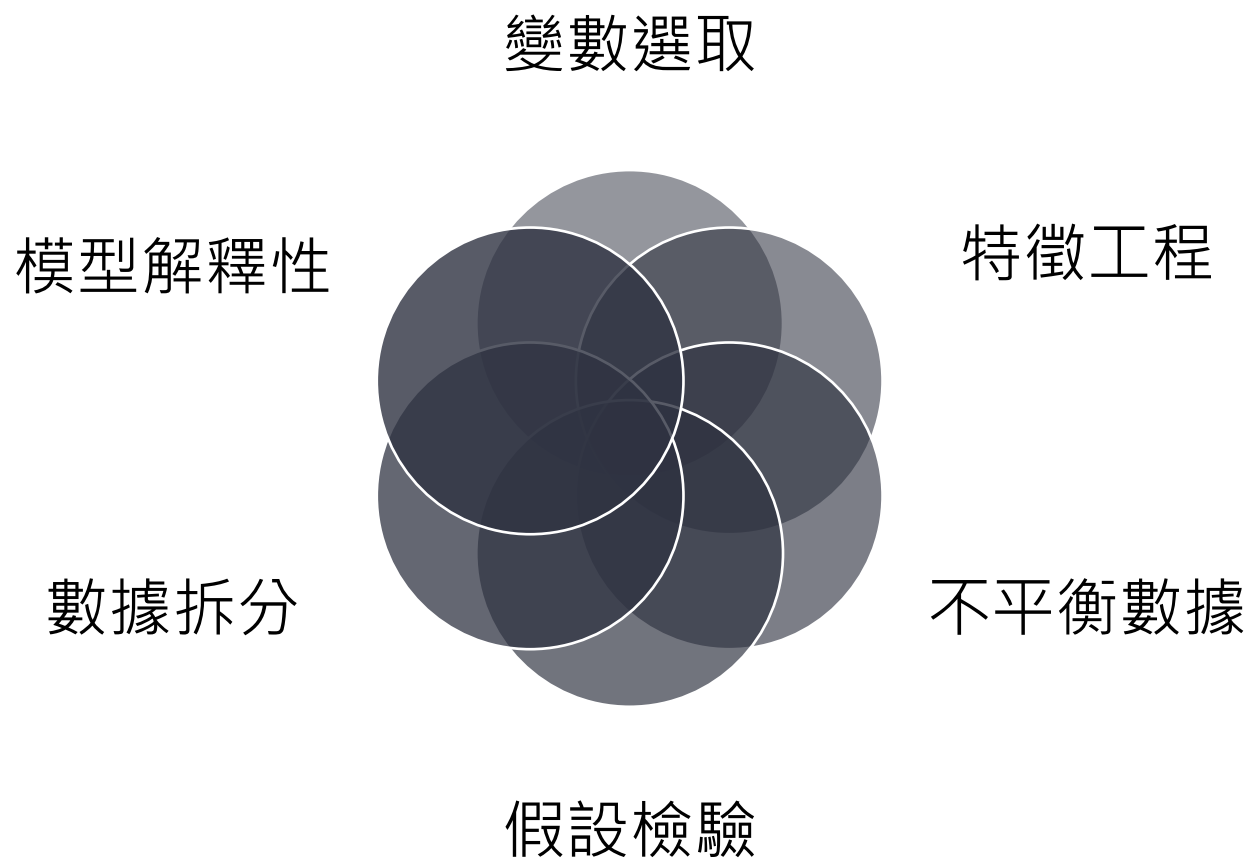
```
# 資料轉換成列資料
df_piv_lars_data = df_lars_data.melt(
    id_vars='Year-Week of Specimen Received'
)

# 建立樞紐分析表
piv_lars = pd.pivot_table(
    data = df_piv_lars_data,
    index = 'Year-Week of Specimen Received',
    columns = 'variable',
    aggfunc = 'mean'
)
```


資料分析 – 分析方法



資料分析 – 模型建立



資料分析

分析模式

模型建立

模型評估



資料分析 – 混淆矩陣

		Actual	
		True	False
Predicted	True	TP	FP
	False	FN	TN

$$Accuracy = \frac{TN + TP}{TN + FN + FP + TP}$$

$$Sensitivity(Recall) = \frac{TP}{TP + FN}$$

$$Specificity = \frac{TN}{TN + FP}$$

$$PPV(Precision) = \frac{TP}{TP + FP}$$

$$NPV = \frac{TN}{TN + FN}$$

$$F1 = \frac{2}{\frac{1}{Sensitivity} + \frac{1}{PPV}}$$

資料分析 – 選模指標

指標	優點	適用方向
MSE	對大誤差更敏感	異常值的影響需要被強調、誤差的分佈近似正態分佈
RMSE	單位與因變量相同，易於解釋	異常值的影響需要被強調、誤差的分佈近似正態分佈
MAE	對異常值較敏感、直觀表示平均誤差	不希望過度懲罰異常值
R-square	模型解釋變異的比例、範圍在0到1之間	線性迴歸適用
Adjust R-square	模型解釋變異的比例並引入懲罰項機制	多元迴歸分析平衡模型複雜度和擬合優度

資料分析 – 選模指標

方法	優點	適用方向
AIC	適合較多參數的模型($k \geq 8$)	預測角度選擇模型、需要在多個競爭模型中選擇時
BIC	適合較少參數的模型($k < 8$)	配飾角度選擇模型、傾向於選擇更簡單的模型
K-fold CV	提供穩健的模型性能估計	適用於小樣本量數據集 (尤其是LOOCV)
Residual Analysis	識別異常值和影響點	診斷迴歸模型的假設是否滿足
VIF	量化自變量間的相關程度、幫助識別可能導致模型不穩定的變量	多元迴歸分析中檢視自變量之間存在高度相關性時



案例分析

核心技術應用說明

章節大綱

案例1

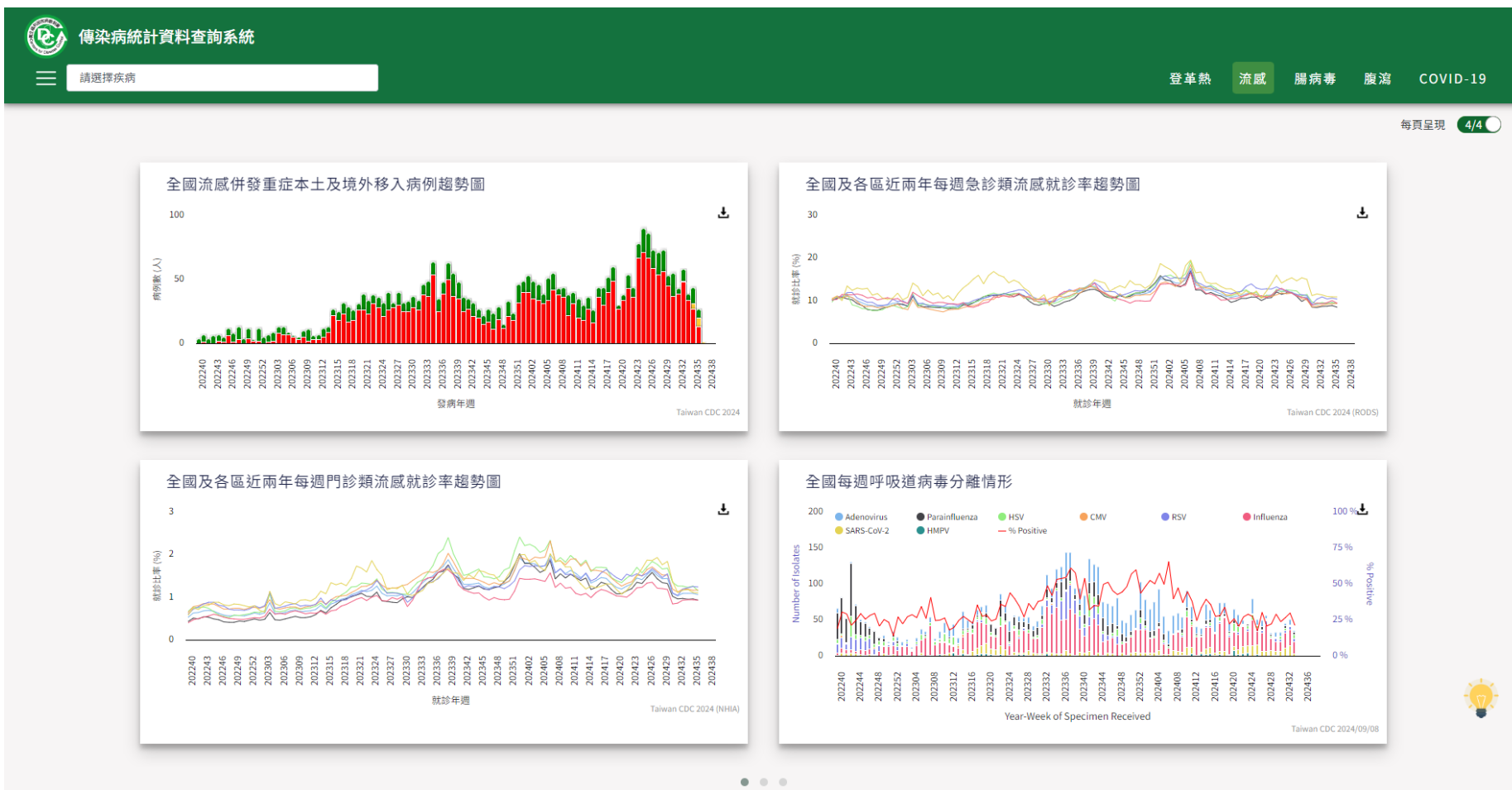
- 相關性分析
- 流感 LARS 病原體資料
- 數據呈現與視覺化

案例2

- 迴歸分析
- Spotify & YouTube 數據分析
- 模型選擇與套件介紹



LARS病原體-資料來源



案例分析

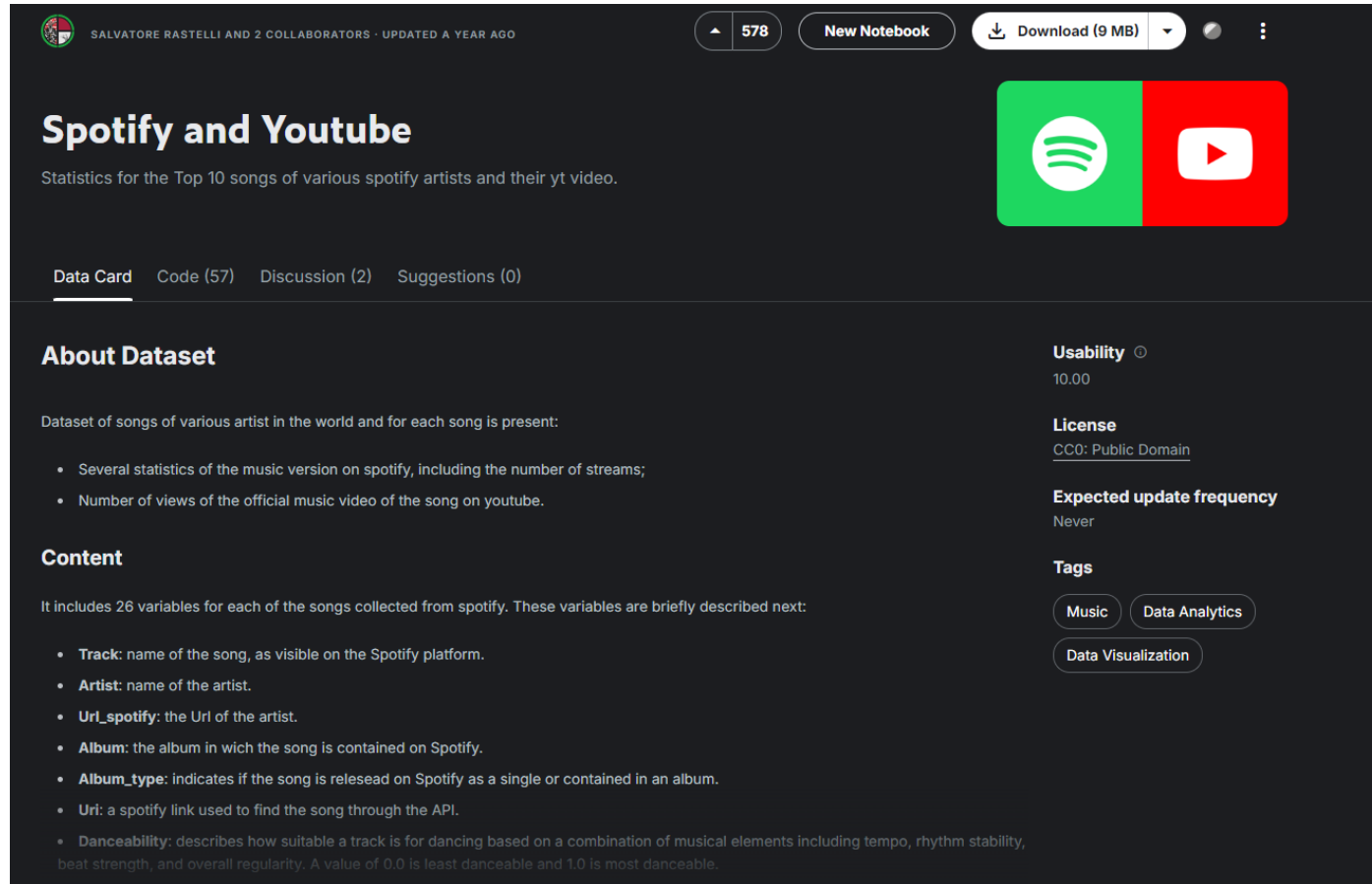
LARS病原體

串流平台數據

串流平台數據 – 變數介紹

Variable	
Year-Week of Specimen Received	Adenovirus
Parainfluenza	HSV
CMV	Influenza
SARS-CoV-2	HMPV
Positive (%)	

串流平台數據 – 資料來源



The screenshot shows the Kaggle dataset page for 'Spotify and Youtube' by Salvatore Rastelli and collaborators. The page is dark-themed and includes a header with the dataset name, author, and update date. Below the header, there are tabs for 'Data Card', 'Code (57)', 'Discussion (2)', and 'Suggestions (0)'. The 'Data Card' is selected, showing an 'About Dataset' section with a description and a list of statistics. To the right, there are sections for 'Usability' (10.00), 'License' (CC0: Public Domain), 'Expected update frequency' (Never), and 'Tags' (Music, Data Analytics, Data Visualization). The 'Content' section describes the 26 variables for each song collected from Spotify.

Spotify and Youtube
Statistics for the Top 10 songs of various spotify artists and their yt video.

About Dataset

Dataset of songs of various artist in the world and for each song is present:

- Several statistics of the music version on spotify, including the number of streams;
- Number of views of the official music video of the song on youtube.

Content

It includes 26 variables for each of the songs collected from spotify. These variables are briefly described next:

- **Track:** name of the song, as visible on the Spotify platform.
- **Artist:** name of the artist.
- **Url_spotify:** the Url of the artist.
- **Album:** the album in wich the song is contained on Spotify.
- **Album_type:** indicates if the song is relesead on Spotify as a single or contained in an album.
- **Url:** a spotify link used to find the song through the API.
- **Danceability:** describes how suitable a track is for dancing based on a combination of musical elements including tempo, rhythm stability, beat strength, and overall regularity. A value of 0.0 is least danceable and 1.0 is most danceable.

Usability
10.00

License
CC0: Public Domain

Expected update frequency
Never

Tags
Music Data Analytics Data Visualization

資料來源：[Kaggle-spotify_and_youtube](#)

串流平台數據 – 變數介紹

Spotify			Youtube	
Track	Danceability	Instrumentalness	Url_youtube	Description
Artist	Energy	Liveness	Title	Licensed
Url_spotify	Key	Valence	Channel	official_video
Album	Loudness	Tempo	Views	
Album_type	Speechiness	Duration_ms	Likes	
Uri	Acousticness	Stream	Comments	



總結

效益與應用





結論

資料清洗

探索性分析

分析方法

資料視覺化



相關連結

- <https://learningds.org/intro.html>
- https://scikit-learn.org/1.3/tutorial/machine_learning_map/index.html



THANK YOU FOR YOUR LISTENING

