資料科學的旋風之旅 http://bit.ly/2xQzGji (http://bit.ly/2xQzGji)

A Whirlwind tour of data science

郭耀仁 tony@kyosei.ai (mailto:tony@kyosei.ai)

大綱

- 關於我
- 資料科學的前世今生
- 資料科學的旋風之旅
- 從 Google Colaboratory 與 Kaggle 啟程
- Q&A

關於我

Could that data be any tidier? It is always nice to meet a data enthusiast / 2:43 marathon runner.

Python, R, 資料科學講師

- 台大資工系統訓練班資深講師（授課時數 2,000+ 小時）
- 資策會資料工程師養成班（Python 網路爬蟲、R）
- 中華電信學院（Python 資料科學、Python 機器學習）
- 華南銀行 Python 資料科學講師
- 2017 資料科學年會講者 http://datasci.tw/tony/ (http://datasci.tw/tony/)
- 玉山銀行 Python 資料科學講師

Blogging and Writing

- 進擊的資料科學 (https://www.books.com.tw/products/0010827812)
- 2,500+ Likes at https://www.facebook.com/datainpoint/ (https://www.facebook.com/datainpoint/)
- 1,700+ Followers at https://medium.com/datainpoint (https://medium.com/datainpoint)
- https://www.datainpoint.com/ (https://www.datainpoint.com/)
- 2017 iT 邦幫忙 Big Data 組冠軍 https://ithelp.ithome.com.tw/ironman/articles/1077 (https://ithelp.ithome.com.tw/ironman/articles/1077)

工作經歷

- Senior Data Analyst, Coupang (https://www.coupang.com/)
- Senior Analytical Consultant, SAS (https://www.sas.com)
- Management Associate, CTBC (https://www.ctbcbank.com/)
- Research Intern, McKinsey & Company (https://www.mckinsey.com/)

學歷

- MBA@台大商研所
- BA@台大工商管理系

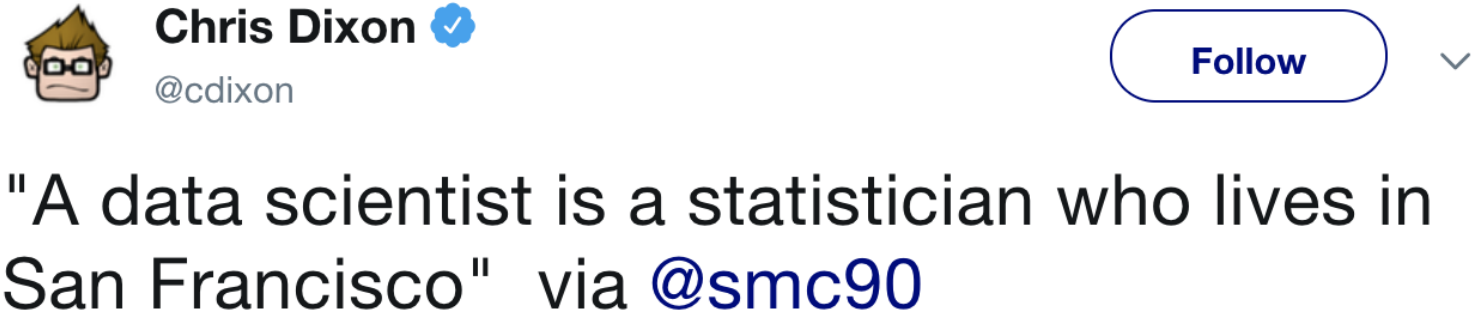
資料科學的前世今生

這要從 2012 年談起...

> *The sexiest job in the 21st century is data scientist.*
>
> *Harvard Business Review*

Source: Data Scientist: The Sexiest Job of the 21st Century (https://hbr.org/2012/10/data-scientist-the-sexiest-job-of-the-21st-century)

什麼是資料科學，給我們一個定義吧！？



Source: https://twitter.com/cdixon/status/428914681911070720 (https://twitter.com/cdixon/status/428914681911070720)

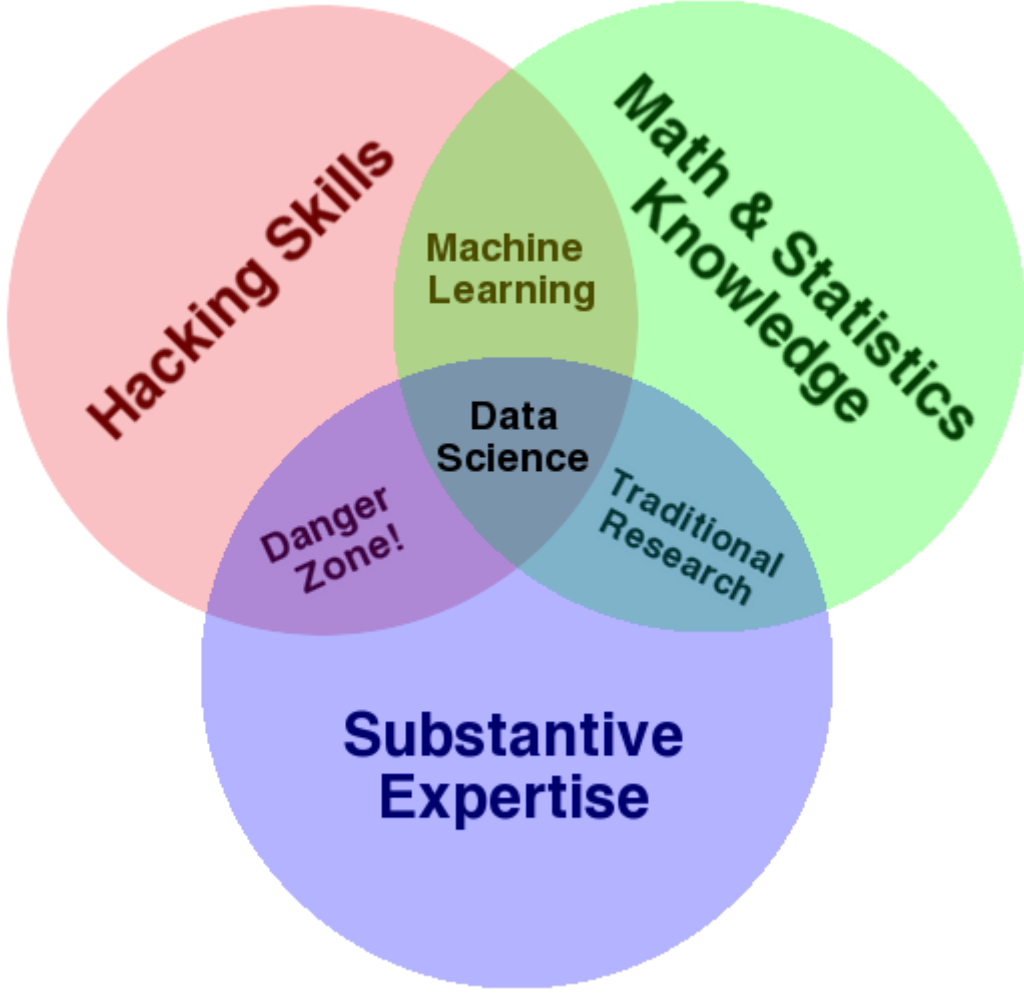
Josh Wills
@josh_wills

Following

Data Scientist (n.): Person who is better at statistics than any software engineer and better at software engineering than any statistician.

9:55 AM - 3 May 2012

Source: https://twitter.com/josh_wills/status/198093512149958656 (https://twitter.com/josh_wills/status/198093512149958656)



Source: http://drewconway.com/zia/2013/3/26/the-data-science-venn-diagram (http://drewconway.com/zia/2013/3/26/the-data-science-venn-diagram)

如果講得很簡略...

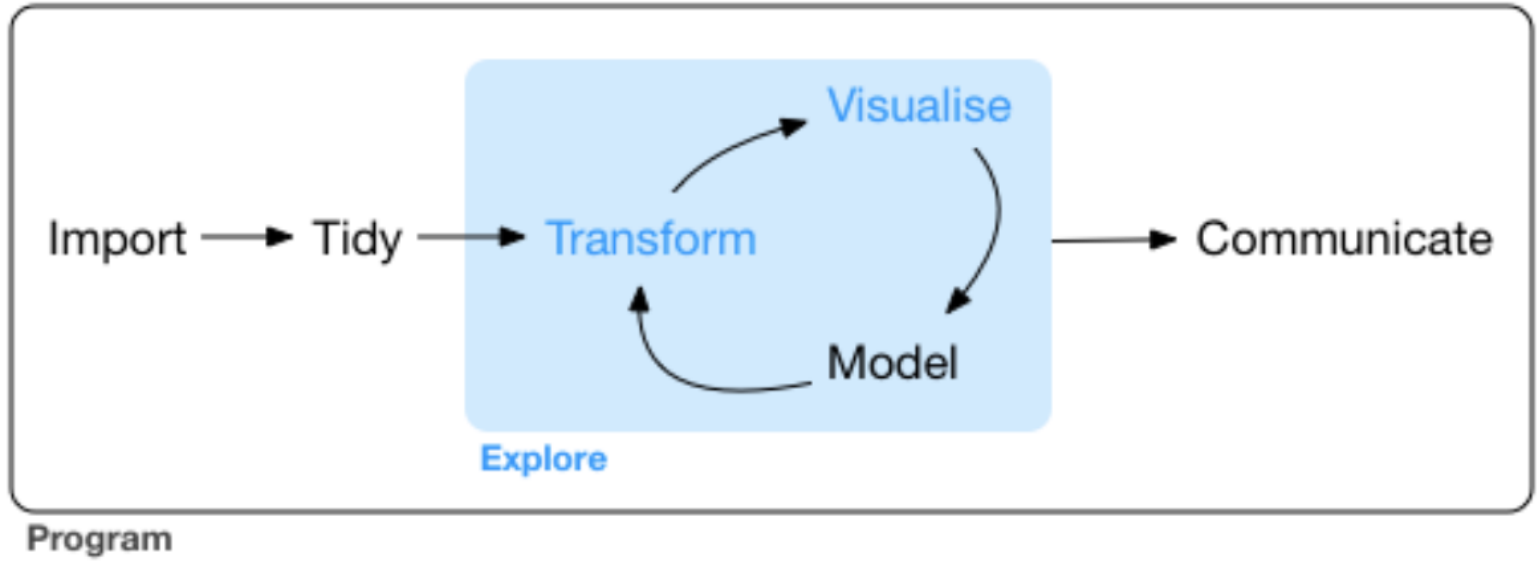
$$\text{Statistics} \cap \text{Computer Science} \cap \text{Applications} = \text{Data Science}$$

The earliest writing on statistics was found in a 9th-century book entitled Manuscript on Deciphering Cryptographic Messages.

Source: https://en.wikipedia.org/wiki/Statistics#History (https://en.wikipedia.org/wiki/Statistics#History)

Wilhelm Schickard designed and constructed the first working mechanical calculator in 1623.

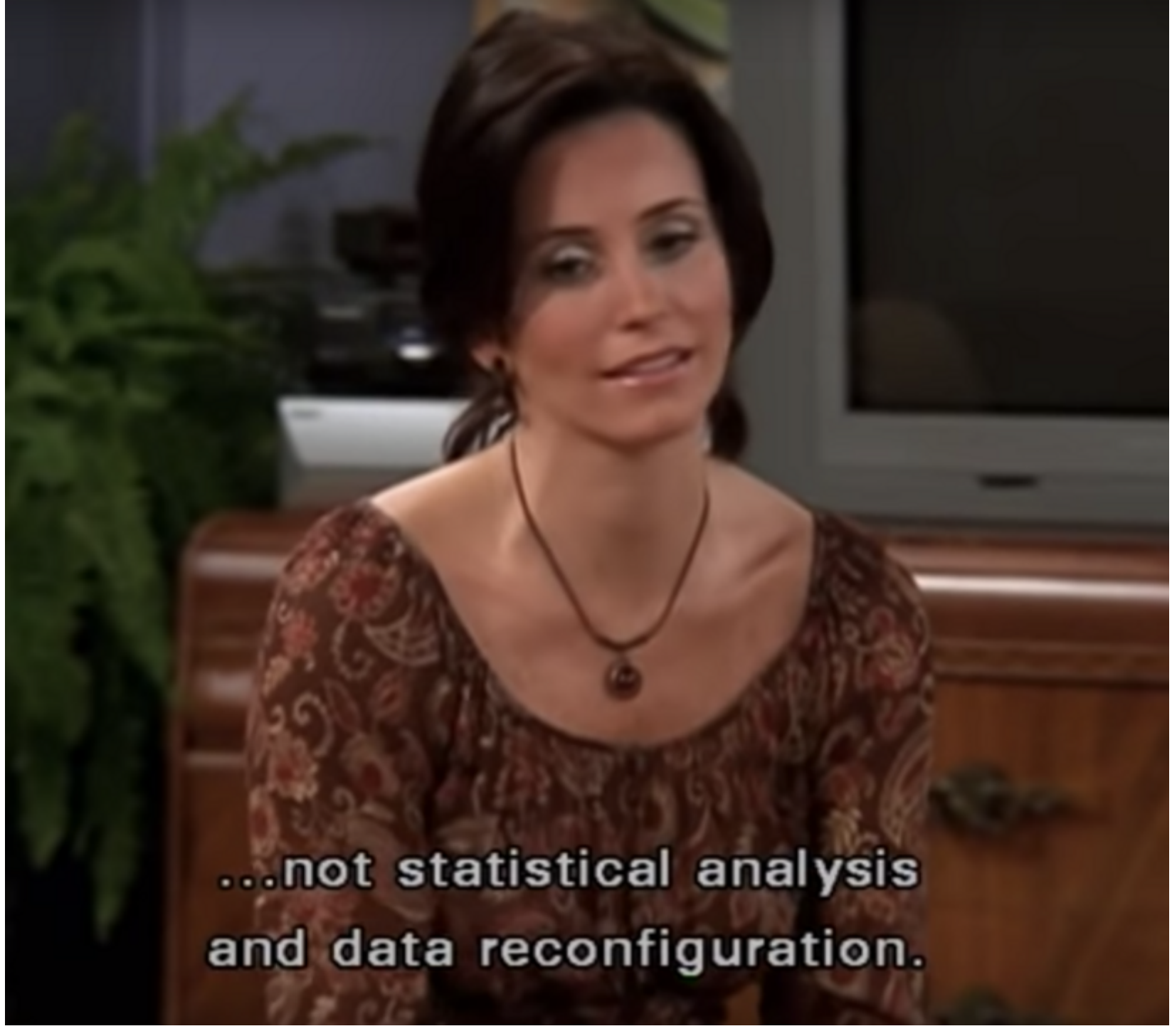
Source: https://en.wikipedia.org/wiki/Computer_science#History (https://en.wikipedia.org/wiki/Computer_science#History)



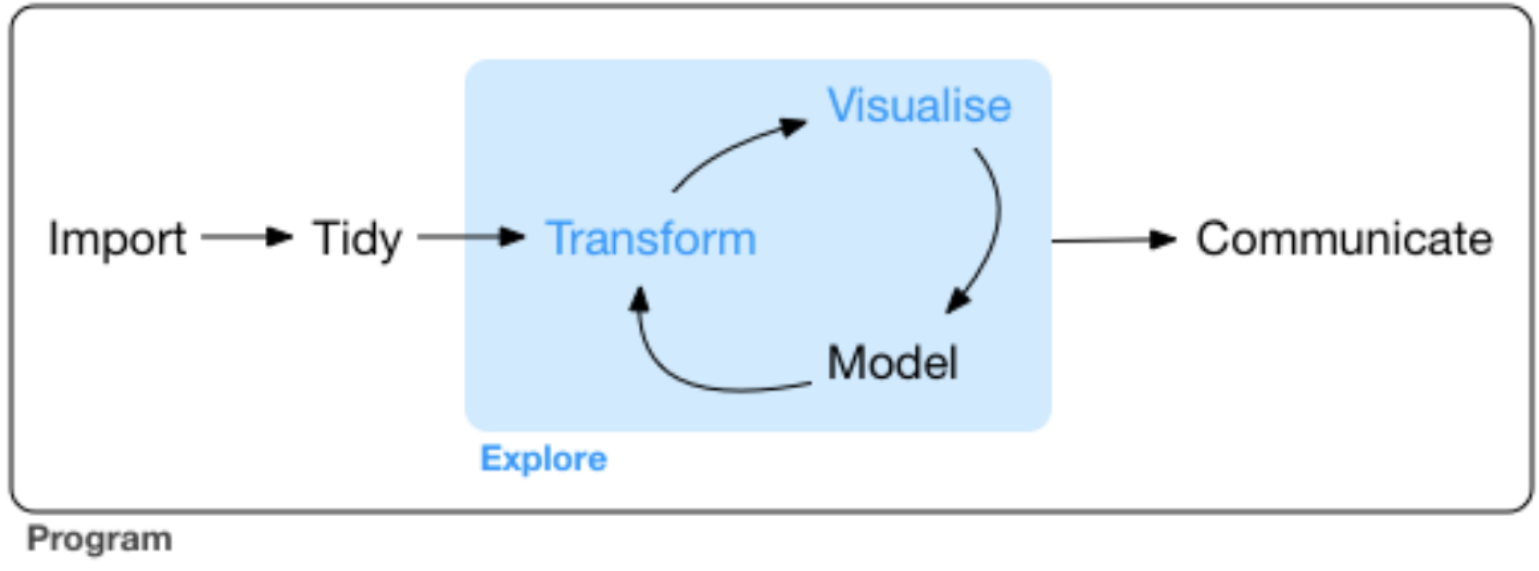
Source: [https://r4ds.had.co.nz/explore-intro.html (https://r4ds.had.co.nz/explore-intro.html)](https://r4ds.had.co.nz/explore-intro.html)

資料科學不是一個橫空出世的學科

Source: https://youtu.be/Z4gjQ8_Gvu4 (https://youtu.be/Z4gjQ8_Gvu4)


...not statistical analysis
and data reconfiguration.

資料科學的旋風之旅



Source: [https://r4ds.had.co.nz/explore-intro.html (https://r4ds.had.co.nz/explore-intro.html)](https://r4ds.had.co.nz/explore-intro.html)

獲取資料

- 檔案
- 資料庫
- Web API
- 非 Web API 的網頁資料

掌控資料

- 基礎資料型態
- 常見資料結構
- 資料框處理
- 文字處理

探索資料

- 敘述性統計
- 視覺化

資料分析與預測

- 商業分析
- 統計與機率
- 機器學習

溝通資料

- 資料分享
- 簡報
- 互動式圖表

在工作上以知識、程式或解決方案面對這些應用，稱為從事資料科學相關的職業

多采多姿的職稱

- 資料分析師 Data Analyst
- 資料工程師 Data Engineer
- 資料科學家 Data Scientist
- 機器學習工程師 Machine Learning Engineer
- ...etc.

對這些職稱在美國的薪資水平有興趣，可以利用 Glassdoor (https://www.glassdoor.com) 查詢

那麼如何能夠從事資料科學相關的職業，成為一個資料分析師/工程師/科學家/…?

是不是想辦法成為一個電腦科學家 / 統計學家?

是，也不是。

The Science of Deduction

Source: https://www.bbc.co.uk/programmes/b018ttws (https://www.bbc.co.uk/programmes/b018ttws)

從需求端：研究資料科學相關職位的工作描述（Job Descriptions）

- Glassdoor (https://www.glassdoor.com/)
- LinkedIn (https://www.linkedin.com/)
- Indeed (https://www.indeed.com/)
- Stackoverflow Jobs (https://stackoverflow.com/jobs)
- 104 人力銀行 (https://www.104.com.tw/)
- CakeResume (https://www.cakeresume.com/)
- ...etc.

從供給端：研究任職資料科學相關工作者的技能（Skill Sets）

探索 2017 Kaggle 資料科學調查資料
(https://medium.com/pyradise/%E6%8E%A2%E7%B4%A2-2017-kaggle-%E8%B3%87%E6%96%99%E7%A7%91%E5%AD%B8%E8%AA%BF%E6%9F%A5%E8%B3%
f03f1617ae5e)

勾勒出從業人員的輪廓

MODERN DATA SCIENTIST

Data Scientist, the sexiest job of 21th century requires a mixture of multidisciplinary skills ranging from an intersection of mathematics, statistics, computer science, communication and business. Finding a data scientist is hard. Finding people who understand who a data scientist is, is equally hard. So here is a little cheat sheet on who the modern data scientist really is.

MATH & STATISTICS

- Machine learning
- Statistical modeling
- Experiment design
- Bayesian inference
- Supervised learning: decision trees, random forests, logistic regression
- Unsupervised learning: clustering, dimensionality reduction
- Optimization: gradient descent and variants

PROGRAMMING & DATABASE

- Computer science fundamentals
- Scripting language e.g. Python
- Statistical computing package e.g. R
- Databases SQL and NoSQL
- Relational algebra
- Parallel databases and parallel query processing
- MapReduce concepts
- Hadoop and Hive/Pig
- Custom reducers
- Experience with xaaS like AWS

DOMAIN KNOWLEDGE & SOFT SKILLS

- Passionate about the business
- Curious about data
- Influence without authority
- Hacker mindset
- Problem solver
- Strategic, proactive, creative, innovative and collaborative

COMMUNICATION & VISUALIZATION

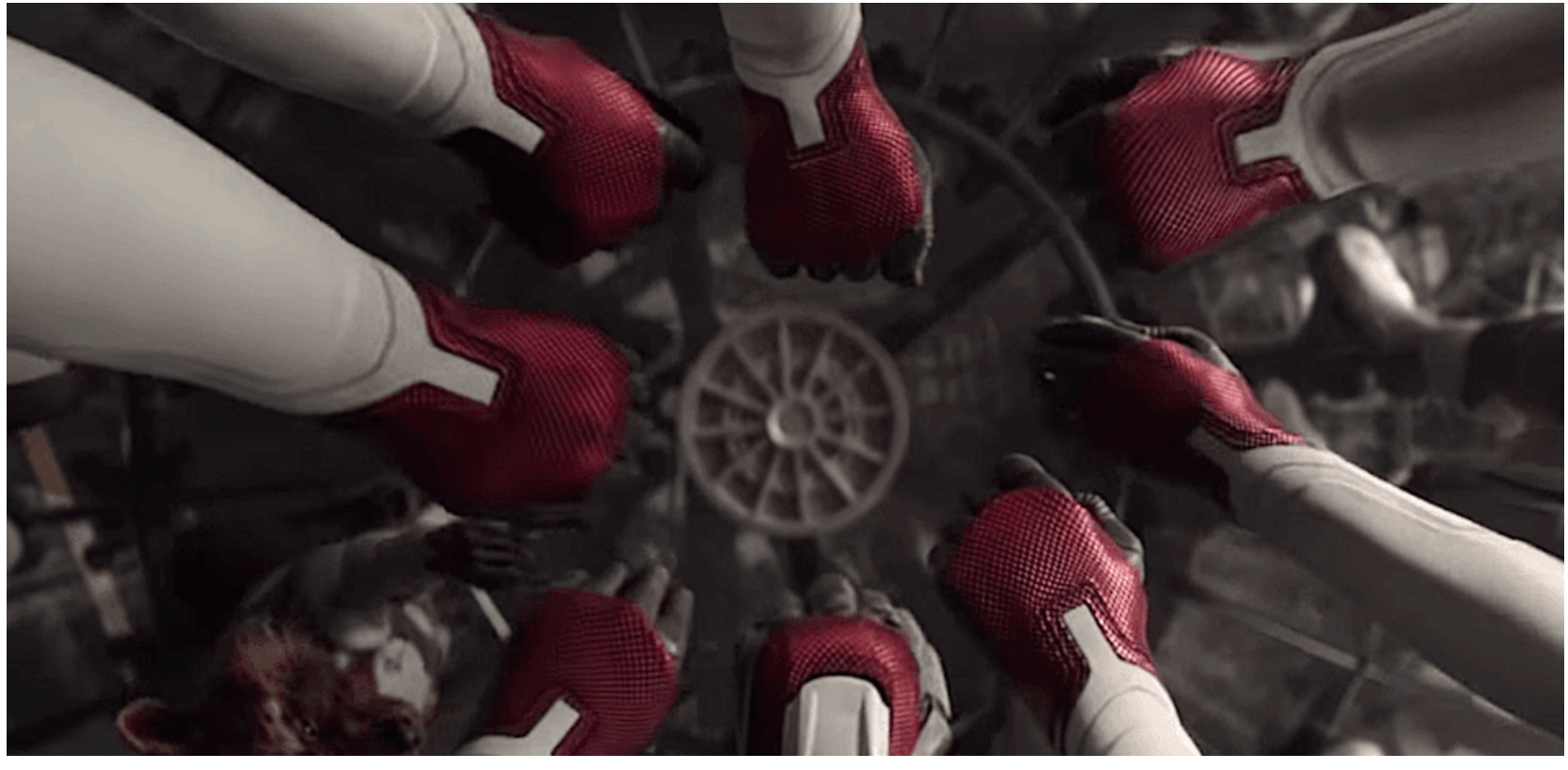
- Able to engage with senior management
- Story telling skills
- Translate data-driven insights into decisions and actions
- Visual art design
- R packages like ggplot or lattice
- Knowledge of any of visualization tools e.g. Flare, D3.js, Tableau

其實就像規劃旅遊行程一樣，決定景點、決定路線然後出發

Source: https://unsplash.com/photos/TrhLCn1abMU (https://unsplash.com/photos/TrhLCn1abMU)

這趟旅程需要持之以恆，因為它跟軟體工程、科技息息相關

如果現在回到 2014 年，我會怎麼規劃這趟旅程?



Source: https://www.screengeek.net/2019/05/13/avengers-endgame-directors-time-travel/ (https://www.screengeek.net/2019/05/13/avengers-endgame-directors-time-travel/)

技術基礎

- 文字編輯器
- 命令列工具與 Shell Script
- Git/GitHub
- 標記語言 Markdown/HTML
- 環境設定
- SQL

程式設計

- 程式語言 Python/R/JavaScript
- 物件導向
- 函數型編程
- 演算法與資料結構

為什麼是這三個?

StackOverflow Trends (https://insights.stackoverflow.com/trends?tags=java%2Cc%2Cc%2B%2B%2Cpython%2Cjavascript%2Cr)

透過廣大使用者社群所支撐的標準或第三方框架套件能支援幾乎所有的資料科學應用

- 獲取資料
- 掌控資料
- 探索資料
- 資料分析與預測
- 溝通資料

Python

`requests`

> *Requests allows you to send organic, grass-fed HTTP/1.1 requests,*
> *without the need for manual labor.*

https://2.python-requests.org/en/master/ (https://2.python-requests.org/en/master/)

`BeautifulSoup4`

> *Beautiful Soup is a Python library for pulling data out of HTML and XML files.*

https://www.crummy.com/software/BeautifulSoup/bs4/doc/ (https://www.crummy.com/software/BeautifulSoup/bs4/doc/)

`selenium`

Selenium Python bindings provides a simple API to write functional/acceptance tests using Selenium WebDriver.

https://selenium-python.readthedocs.io/ (https://selenium-python.readthedocs.io/)

`numpy`

> *NumPy is the fundamental package for scientific computing in Python.*

https://www.numpy.org/devdocs/user/index.html (https://www.numpy.org/devdocs/user/index.html)

`pandas`

> *Flexible and powerful data analysis / manipulation library for Python, providing labeled data structures similar to R data.frame objects, statistical functions, and much more.*

https://pandas.pydata.org/ (https://pandas.pydata.org/)

`matplotlib.pyplot`

> *matplotlib.pyplot is a collection of command style functions that make matplotlib work like MATLAB.*

https://matplotlib.org/3.1.0/tutorials/introductory/pyplot.html (https://matplotlib.org/3.1.0/tutorials/introductory/pyplot.html)

scikit-learn

Simple and efficient tools for data mining and data analysis.

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

`flask`

Flask is a lightweight WSGI web application framework.

https://flask.palletsprojects.com/en/1.1.x/ (https://flask.palletsprojects.com/en/1.1.x/)

R

`rvest`

rvest helps you scrape information from web pages.

https://rvest.tidyverse.org/ (https://rvest.tidyverse.org/)

jsonlite

A Robust, High Performance JSON Parser and Generator for R.

https://github.com/jeroen/jsonlite (https://github.com/jeroen/jsonlite)

`ggplot2`

> *ggplot2 is a system for declaratively creating graphics, based on The Grammar of Graphics.*

https://ggplot2.tidyverse.org/ (https://ggplot2.tidyverse.org/)

`shiny`

> *Shiny is an R package that makes it easy to build interactive web apps straight from R.*

https://shiny.rstudio.com/ (https://shiny.rstudio.com/)

JavaScript

Node.js

> *As an asynchronous event driven JavaScript runtime, Node is designed to build scalable network applications.*

https://nodejs.org/en/ (https://nodejs.org/en/)

`d3.js`

> *D3.js is a JavaScript library for manipulating documents based on data.*

https://d3js.org/ (https://d3js.org/)

如何有系統地自學程式語言：以 Python 為例

- 起步走
- 資料型態
- 流程控制
- 資料結構
- 程式封裝
- 特別的部分

從 Google Colaboratory 與 Kaggle 啟程

A taste of Python：使用 Google Colaboratory (https://colab.research.google.com/)

Colaboratory is a free Jupyter notebook environment that requires no setup and runs entirely in the cloud.

在瀏覽器使用 Google Colaboratory

1. 登入 Google 帳號，開啟雲端硬碟
2. 點選「新增」
3. 點選「連結更多應用程式」
4. 搜尋「Colaboratory」
5. 點選「連結」
6. 新增 Google Colaboratory
7. 完成

透過 Colaboratory Notebooks 瞭解 Colaboratory

- Overview of Colaboratory (https://colab.research.google.com/notebooks/basic_features_overview.ipynb)
- Importing libraries and installing dependencies (https://colab.research.google.com/notebooks/snippets/importing_libraries.ipynb)
- Saving and loading notebooks in GitHub (https://colab.research.google.com/github/googlecolab/colabtools/blob/master/notebgithub-demo.ipynb)
- Interactive forms (https://colab.research.google.com/notebooks/forms.ipynb)

Python 禪學，Zen of Python

In [ ]:
```
import this
```


Imgur

Source: https://www.imdb.com/title/tt4154796/ (https://www.imdb.com/title/tt4154796/)

In [ ]:

```
movie_title = "Avengers: Endgame"
movie_release_year = 2019
movie_rating = 8.7
movie_is_good = movie_rating > 8
movie_is_perfect = movie_rating > 9.5
```

布林的應用場景

- 條件判斷
- `while` 迴圈
- 資料篩選

條件判斷

In [ ]:

```
movie_title = input("請輸入電影名稱: ")
movie_rating = input("請輸入電影評分: ")
movie_rating = float(movie_rating)

if movie_rating > 8:
    print("{}的評分為 {} 分要去電影院看! ".format(movie_title, movie_rating))
elif movie_rating > 7:
    print("{}的評分為 {} 分值得一看! ".format(movie_title, movie_rating))
else:
    print("{}的評分為 {} 分看了會後悔! ".format(movie_title, movie_rating))
```

迴圈是用來解決需要反覆執行、大量手動複製貼上程式碼的任務

- 重複印出（Print）
- 計數（Counter）
- 加總（Summation）
- 合併（Concatenation）

while 迴圈

印出介於 1 到 10 之間的偶數

In [ ]:
```
i = 2
while i <= 10:
    print(i)
    i += 2
```

for 迴圈

印出介於 1 到 10 之間的偶數

In [ ]:
```
for i in range(2, 11, 2):
    print(i)
```

`list` 是 Python 基礎的資料結構

In [ ]:
```
the_avengers = ["Iron Man", "Captain America", "Hulk", "Thor", "Black Widow", "Haw
keye"]
print(type(the_avengers))
```

dict 是 Python 將資料與標籤綁定的彈性資料結構

In [ ]:

```
the_avengers = {
    "Iron Man": "Tony Stark",
    "Captain America": "Steve Rogers",
    "Hulk": "Bruce Banner",
    "Thor": "Thor",
    "Black Widow": "Natasha Romanoff",
    "Hawkeye": "Clint Barton"
}
print(type(the_avengers))
```

程式封裝的三個層級

- 模組與套件（Modules and classes）
    - 類別（Classes）
        - 函數（Functions）

In [ ]:

```python
# Define
def get_bmi(height, weight):
    """
    依據身高、體重計算 BMI 身體質量指數
    身高：以公分（cm）為單位
    體重：以公斤（kg）為單位
    """
    bmi = weight / (height/100)**2
    return bmi
```

In [ ]:

```
class Movie:
    """
    Information about a certain movie.
    """
    def __init__(self, title, imdb_rating, release_date):
        self._title = title
        self._imdb_rating = imdb_rating
        self._release_date = release_date
    def get_title(self):
        return self._title
    def get_imdb_rating(self):
        return self._imdb_rating
    def get_release_date(self):
        return self._release_date
    def going_to_movie_theater(self):
        if self._imdb_rating >= 8:
            return "{} 的評等為 {}，值得去電影院看！".format(self._title, self._imdb_r
ating)
        else:
            return "{} 的評等為 {}，應該不用去電影院看...".format(self._title, self._i
mdb_rating)
```

In [ ]:

```
avengers_endgame = Movie("Avengers: Endgame", 8.7, "2019-04-24")
print(avengers_endgame.get_title())
print(avengers_endgame.get_imdb_rating())
print(avengers_endgame.get_release_date())
print(avengers_endgame.going_to_movie_theater())
```

使用 `import` 指令載入標準與外部模組套件

- 標準函式庫（Standard Libraries）
- 外部函式庫（Third-party Libraries）

In [ ]:

```
import numpy as np
import pandas as pd

print(np.__version__)
print(pd.__version__)
```

外部函式庫：使用 pip install 指令

```
pip install LIBRARY_NAME
```

`pip` 是 Python 套件管理工具

- 不需額外安裝
- 協助使用者從 Python Pacakge Index (https://pypi.org/) 下載、安裝或更新

模組

- 將函數或類別封裝在一個 `.py` 檔案中
- `.py` 的檔名就是模組名稱

自訂一個名為 `movie` 的模組

In [ ]:

```
# movie.py ------
from urllib.parse import quote_plus
import requests
from bs4 import BeautifulSoup

class Movie:
    def __init__(self, movie_title, movie_url=None, movie_cast=None):
        self._movie_title = movie_title
        self._movie_url = movie_url
        self._movie_cast = movie_cast

    def get_movie_url(self):
        query_str = quote_plus(self._movie_title)
        page_url = "https://www.imdb.com/find?q={}&s=tt&ttype=ft&ref_=fn_ft".forma
t(query_str)
        r = requests.get(page_url)
        soup = BeautifulSoup(r.text, 'lxml')
        first_link = soup.select(".result_text a")[0].get("href")
        movie_url = "https://www.imdb.com" + first_link
        self._movie_url = movie_url
        return self

    def print_movie_cast(self):
        movie_url = self._movie_url
        r = requests.get(movie_url)
        soup = BeautifulSoup(r.text, 'lxml')
        n_obs = 15
        cast = [soup.find_all("td")[i].find("img").get("alt") for i in range(1, 1+
n_obs*4, 4)]
        for actor in cast:
            print(actor)
```

載入 movie 模組中的 Movie 類別

In [ ]:
```
from movie import Movie

endgame = Movie("Avengers: Endgame")
endgame.get_movie_url().print_movie_cast()
```

套件

- 多個功能相關的模組（.py 檔案）可以組織成一個套件（資料夾）
- 在工作目錄建立一個資料夾，將這個資料夾的名稱更改為 movie_package
- 在這個資料夾中新增檔案：
    - movie.py

載入 movie_package 套件中 movie 模組的 Movie 類別

In [ ]:
```
from movie_package.movie import Movie

endgame = Movie("Avengers: Endgame")
endgame.get_movie_url().print_movie_cast()
```

NumPy

在科學計算使用者眼裡以純量作為運算單位還是太麻煩

- 哪些程式語言內建了 Vectorization（向量化）功能?
  - Matlab
  - R
  - Julia
  - ...etc.

NumPy as in: Numerical Python

創建一種稱為 ndarray 的類別，彌補了原生 list 缺少的向量化運算（vectorization）功能

1 公里是 0.62137 英里，將這幾個長跑距離（公里）轉換為英里

In [ ]:
```
distances = [1, 1.6, 3, 5, 10, 21.097, 42.195]
```

In [ ]:
```
dist_in_mile = []
for d in distances:
    dist_in_mile.append(d * 0.62137)
print(dist_in_mile)
```

計算 A 與 B 的內積 C

$$C_{i,j} = \sum A_{i,k} B_{k,j}$$

In [ ]:

```
def get_mat_dot(A, B):
    I = len(A)
    K_A = len(A[0])
    K_B = len(B)
    J = len(B[0])
    if K_A != K_B:
        raise ValueError("shapes ({},{}) and ({},{}) not aligned: {} (dim 1) != {}
(dim 0)".format(I, K_A, K_B, J, K_A, K_B))
    C = [[0 for j in range(J)] for i in range(I)]
    for i in range(I):
        for k in range(K_A):
            for j in range(J):
                C[i][j] += A[i][k] * B[k][j]
    return C
```

In [ ]:

```
A = [
    [1, 2],
    [4, 5]
]
B = [
    [4, 3],
    [2, 1]
]

get_mat_dot(A, B)
```

NumPy to the Rescue!

1 公里是 0.62137 英里，將這幾個長跑距離（公里）轉換為英里

In [ ]:
```
import numpy as np

distances = [1, 1.6, 3, 5, 10, 21.097, 42.195]
distances = np.array(distances)
dist_in_mile = distances * 0.62137
print(dist_in_mile)
```

計算 A 與 B 的內積 C

$$C_{i,j} = \sum A_{i,k} B_{k,j}$$

In [ ]:

```
import numpy as np

A = [
    [1, 2],
    [4, 5]
]
B = [
    [4, 3],
    [2, 1]
]
A = np.array(A)
B = np.array(B)
C = A.dot(B)
print(C)
```

Pandas

Pandas as in

- **Pan**el
- **Da**taFrame
- **S**eries

主要的應用場景

- 表格式資料的讀取
- 豐富的資料清理與分析函數
- 視覺化：包裝了常用的 matplotlib.pyplot 圖形

Python 一直以來都非常適合資料處理，但她的分析能力很薄弱，`pandas` 的開發有助於補足 Python 資料分析的需求，讓使用者能夠在 Python 中執行完整的資料分析流程，而無需切換到 data-centric 的特定語言，如 R。

在科學計算使用者眼裡表格資料處理很重要

- 哪些程式語言內建了表格資料處理功能?
  - R
  - Matlab
  - SAS
  - ...etc.

計算註冊於開曼群島的上市公司股價中位數

https://tw.stock.yahoo.com/d/i/rank.php?t=pri&e=tse&n=100
(https://tw.stock.yahoo.com/d/i/rank.php?t=pri&e=tse&n=100)

In [ ]:

```
import requests
from bs4 import BeautifulSoup

def get_price_rank():
    page_url = "https://tw.stock.yahoo.com/d/i/rank.php?t=pri&e=tse&n=100"
    r = requests.get(page_url)
    soup = BeautifulSoup(r.text)
    stock_tickers = []
    stock_names = []
    for i in soup.select(".name a"):
        stock_ticker = i.text.split()[0]
        stock_name = i.text.split()[1]
        stock_tickers.append(stock_ticker)
        stock_names.append(stock_name)
    prices = []
    for i in range(5, 5+10*100, 10):
        price = soup.find_all("table")[2].find_all("td")[0].find_all("td")[i].text
        prices.append(float(price))
    return stock_tickers, stock_names, prices
```

In [ ]:

```
stock_tickers, stock_names, prices = get_price_rank()
print(stock_tickers)
print(stock_names)
print(prices)
```

In [ ]:

```
from statistics import median

ky_prices = [price for stock_name, price in zip(stock_names, prices) if "KY" in st
ock_name]
print(median(ky_prices))
```

Pandas to the Rescue!

In [ ]:

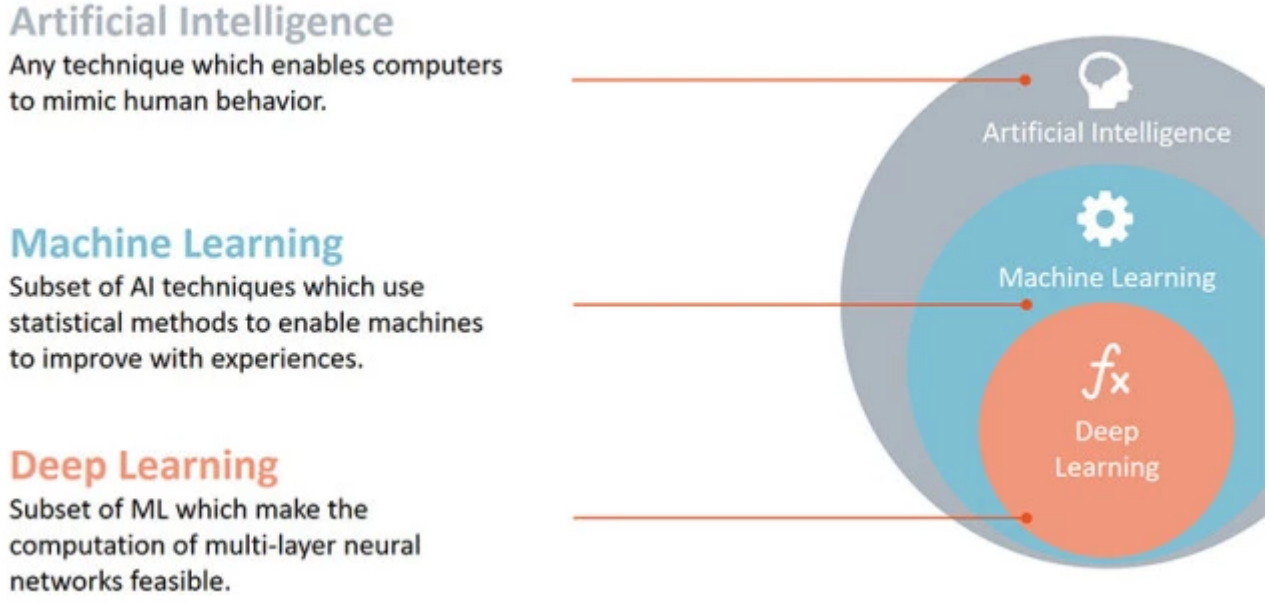
```
import pandas as pd

df = pd.DataFrame()
df["ticker"] = stock_tickers
df["stock_name"] = stock_names
df["price"] = prices
df.head()
```

In [ ]:
```
df[df["stock_name"].str.contains("KY")]["price"].median()
```

A taste of Python Machine Learning：使用 Google Colaboratory (https://colab.research.google.com/) 與 Kaggle (https://www.kaggle.com)

Inside Kaggle you'll find all the code & data you need to do your data science work.

機器學習是實踐人工智慧的手段，深度學習是機器學習中的一種方式



Source: [rapidminer (https://rapidminer.com/artificial-intelligence-machine-learning-deep-learning/)](https://rapidminer.com/artificial-intelligence-machine-learning-deep-learning/)

精準地用一句話說明機器學習

> *A computer program is said to learn from experience E with respect to some task T and some performance measure P, if its performance on T, as measured by P, improves with experience E.*
>
> *Tom Mitchell (http://www.cs.cmu.edu/~tom/), 1997*

勇者鬥惡龍：擊敗龍王的任務

Source: https://dragonquest.square-enix-games.com (https://dragonquest.square-enix-games.com)

機器學習：於可及範圍的 $\mathbf{H}$ 中尋找一個與 $\mathbf{f}$ 最相似 $\mathbf{h_i}$ 的任務

$$y = f(x)$$
$$\boldsymbol{H} = \{h_1, h_2, \ldots, h_n\}$$
$$\hat{y} = h_i(x)$$

機器學習的全景

- 監督式學習（Supervised Learning）
- 半監督式學習（Semi-supervised Learning）
- 遷移學習（Transfer Learning）
- 非監督式學習（Unsupervised Learning）
- 結構式學習（Structured Learning）
- 強化式學習（Reinforcement Learning）
- 未知的

常見學習途徑

- 從理論著手
- 從工具著手
- 從練習著手

從理論著手

- Andrew Ng: Machine Learning (https://www.coursera.org/learn/machine-learning)
- 林軒田：機器學習基石 (https://www.coursera.org/learn/ntumlone-mathematicalfoundations/)
- 李宏毅：機器學習 (https://www.youtube.com/playlist?list=PLJV_el3uVTsPy9oCRY30oBPNLCo89yu49)
- Deep Learning (https://www.deeplearningbook.org/)
- Introduction to Statistical Learning (http://www-bcf.usc.edu/~gareth/ISL/)
- The Elements of Statistical Learning (https://web.stanford.edu/~hastie/ElemStatLearn/)

從工具著手

- Scikit-Learn (https://scikit-learn.org/stable/)
- Google Machine Learning Crash Course (https://developers.google.com/machine-learning/crash-course/ml-intro)
- fast.ai (https://www.fast.ai/)
- Introduction to TensorFlow for Artificial Intelligence, Machine Learning, and Deep Learning (https://www.coursera.org/learn/introduction-tensorflow)
- PyTorch (https://pytorch.org/)

從練習著手

- Kaggle (https://www.kaggle.com/)
- DataCamp (https://www.datacamp.com?tap_a=5644-dce66f&tap_s=194899-1fb421)

Kaggle Datasets API

- 安裝 Kaggle 模組
- 註冊 Kaggle 帳號
- 建立新的 API 憑證
- 參與三個 Getting Started 競賽
- 設定 Kaggle Data API

安裝 Kaggle 模組

```
# Run in command line
pip install kaggle
```

註冊 Kaggle (https://www.kaggle.com/) 帳號

建立新的 API 憑證

- My Account
- Create New API Token

參與三個 Getting Started 競賽

- Titanic (https://www.kaggle.com/c/titanic)
- House Prices: Advanced Regression Techniques (https://www.kaggle.com/c/house-prices-advanced-regression-techniques)
- Digit Recognizer (https://www.kaggle.com/c/digit-recognizer)

設定 Kaggle Data API

```
!mkdir /root/.kaggle
import json
token = {"username": "YOUR-USERNAME", "key": "YOUR-KEY"}
with open('/root/.kaggle/kaggle.json', 'w') as file:
    json.dump(token, file)
!chmod 600 /root/.kaggle/kaggle.json
```

在 Google Colaboratory 中操作 Getting Started 競賽的 Kernels

Q&A