食品營養成分資料庫網路擷取、資料處理及視覺化

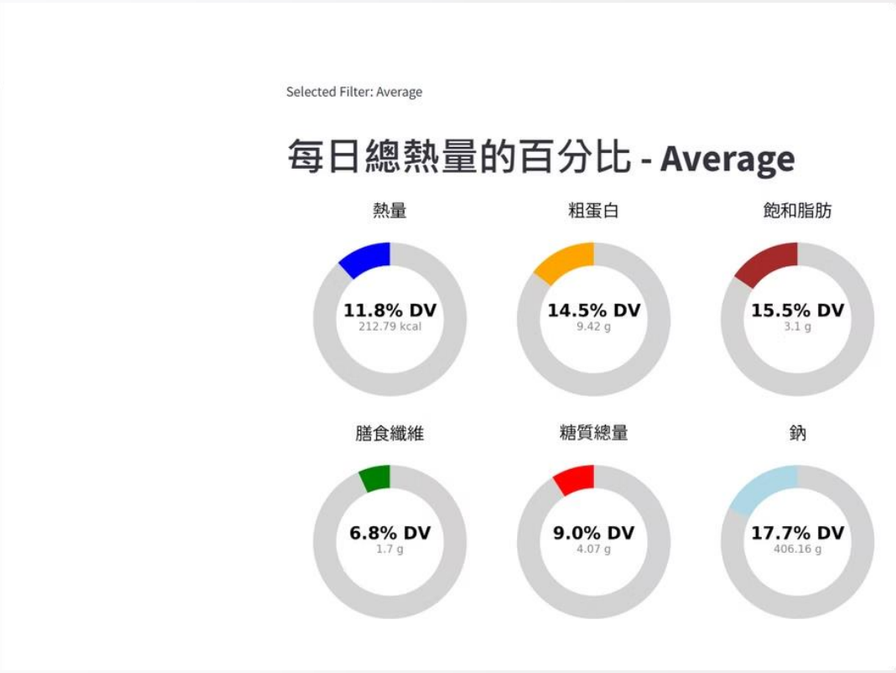
本項目旨在自動化擷取台灣衛服部食藥署的食品營養成分資料庫。我們將運用先進的網路爬蟲技術，高效地收集並處理大量食品營養數據，為後續分析奠定基礎。

E **投稿人：Eve Lee**

Table of Contents

- 專案目標與使用技術
- 程式設計: 網路擷取、資料處理、資料視覺化
- 環境配置與工具準備
- 網站結構分析與擷取策略
- 核心函數與代碼實現
- 優化與效能提升
- 挑戰與解決方案
- 結論與未來展望

Nutrient	Average	Unit per 100 g or ml
熱量	212.79	kcal
粗蛋白	9.42	g
粗脂肪	8.44	g
飽和脂肪	3.10	g
總碳水化合物	24.81	g
膳食纖維	1.70	g
糖質總量	4.07	g
鈉	406.16	g
膽固醇	36.00	g
反式脂肪	43.71	mg
HSR	3.60	Score

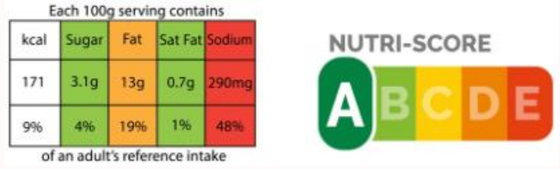


專案目標與使用技術

專案目標

- 透過 Selenium、Webdriver 和 Fake User Agent 從台灣食品藥物管理局 (TFDA) 取得營養資訊
- 分析「加工調理食品類」中「冷凍食品」的營養資料並將其數據處理製作營養分數標籤
- 從新加坡、澳洲和其他國家的營養視覺標籤中取得靈感，使用了台灣的營養資訊，創建一個類似於Health Star Rating/Nutri-Score的Dashboard。

Australia: Health Star Rating (Source: Wikipedia)



Singapore: Nutri-Score

食品營養成分資料庫(

食品分類	穀物類
資料類別	樣品基本資料
整合編號	A0100101
樣品名稱	大麥仁
俗名	小薏仁,洋薏仁,珍珠薏仁
樣品英文名稱	Barley
內容物描述	樣品狀態:生,已去殼; 前處理描述:混合均勻磨碎
廢棄率	0.0%
單位重(可食部分)：	1 x 杯 212.0克 = 212.0克
計算每	100 克成分值

重置 匯出EXCEL

分析項分類	分析項	單位	每100克含量	樣本數	
一般成分	熱量	kcal	365		
一般成分	修正熱量	kcal	347		
一般成分	水分	g	11.7	4	1.
一般成分	粗蛋白	g	8.6	4	1.

使用模組

1 自動化擷取

利用Selenium實現對動態內容的高效處理，確保數據的準確性和完整性。

2 多標籤頁功能

通過多標籤頁同時操作，大幅提升數據收集的效率和速度。

3 數據擷取及處理

將擷取的原始數據轉化為結構化格式，便於後續分析和應用。

4 Dashboard

使用Pandas、Matplotlib、Plotly製作圖表並呈現於Streamlit

- 網路擷取: Selenium (WebDriver: Switch Tabs, etc.)
- 數據處理&圖表: Pandas、 Matplotlib、Plotly
- 數據呈現: Streamlit

程式解說

程式設計1: 網路擷取

瀏覽目標網站 - visit()

首先，程式會自動瀏覽指定的食品營養成分資料庫網站。此步驟包含確認網站結構，識別目標數據（例如，食品名稱、營養成分表等），以及規劃數據擷取策略。

選單篩選與關鍵字搜尋

多標籤頁開啟與切換

搜尋結果頁面會包含多個食品項目的連結。程式利用Selenium的多標籤頁功能，同時開啟所有搜尋結果頁面。 Selenium的WebDriver將負責在不同標籤頁間進行高效切換。

食品名稱與其連結擷取

程式會從每個開啟的標籤頁中擷取目標食品的營養成分數據，以及對應的連結，並將其儲存到Python list中。這些數據將在後續的步驟中進行整理和處理。

程式設計2: 網路擷取、資料處理及視覺化

設定擷取目標食品資訊和營養成分並 Iterate list_data的Links以利讀取所有搜尋結果的頁面

將資訊儲存為Pandas DataFrame和.csv檔案

透過csv檔案處理數據並計算健康評分

使用matplotlib和plotly在Streamlit dashboard上將數據視覺化為潛在營養標籤

程式解說

網路擷取部分

網站結構分析與自動化擷取策略

選單選取食品分類與關鍵字搜尋

依照欲搜尋食物類別，選取選單中一類食品分類，進行關鍵字搜尋。此專案查詢"加工調理食品類"分類中的"冷凍"食品

多分頁處理

依照搜尋結果的頁數，開啟其頁數並透過切換多個分頁，實現並同步執行全部搜尋結果的資料擷取。

共有 81 筆搜尋結果

項次	整合編號	樣品名稱	俗名	樣品英文名稱	內容物描述
81	R7500501	毛鱗魚(柳葉魚)(裹粉未炸)	柳葉魚		樣品狀態:冷凍包裝; 前處理描述:混合均勻打碎

« ◀ 5 6 7 8 9 共有9頁

食品名稱與其連結擷取

在此步驟中，透過程式自動化操作瀏覽器，從食品營養資料網站的搜尋結果中擷取每個食品名稱及其對應的詳細資訊連結。這些資料會以字典格式存儲於 `list_data`，方便後續分析與處理使用。

```
列表
in range(0, len(driver.window_handles)):

hange tabs
iver.switch_to.window(driver.window_handles[index])

取得列表連結與內文
ems = driver.find_elements(By.CSS_SELECTOR, 'td[data-th="
r a in items:
  list_data.append({
      'Food': a.get_attribute('innerText'),
      'Link': a.get_attribute('href')
  })

Exception as e:
int(f"Error extracting data at tab {index}: {e}")
```

```
凍火腿炒飯',
ps://consumer.fda.gov.tw/Food/tfndDetail.aspx?nodeID=1788
凍蝦仁炒飯',
ps://consumer.fda.gov.tw/Food/tfndDetail.aspx?nodeID=1788
凍筒仔米糕',
ps://consumer.fda.gov.tw/Food/tfndDetail.aspx?nodeID=1788
凍芝麻湯圓',
ps://consumer.fda.gov.tw/Food/tfndDetail.aspx?nodeID=1788
凍花生湯圓',
ps://consumer.fda.gov.tw/Food/tfndDetail.aspx?nodeID=1788
```

核心函數: `extract_food_nutr()`

擷取特定營養素的資料(`selected_items`)

從此函數負責從HTML表格中提取特定營養素數據，處理可能的異常情況。

錯誤處理

使用try-except塊捕獲和處理可能出現的異常。

數據存儲

將擷取的數據存儲到food_nutr的list中，以利後續處理。

```python
# Loop through each row and extract '分析項' and '每100克含量'
def extract_food_nutr():
    """Extract food name and nutrition data for selected items from the TFDA website."""
    try:
        # Define the specific '分析項' you're interested in
        selected_items = ["熱量", "粗蛋白", "粗脂肪", "飽和脂肪", "總碳水化合物", "膳食纖維",
                          "糖質總量", "鈉", "膽固醇", "反式脂肪"]

        # Locate rows in a table
        trs = driver.find_elements(By.CSS_SELECTOR, 'table.rwd-table > tbody > tr')

        # Append food name
        nutr = []
        food_name = driver.find_element(
            By.CSS_SELECTOR, '#ctl00_content_lbFoodName'
            ).get_attribute('innerText')
        nutr.append(food_name)
        for tr in trs:
            tds = tr.find_elements(By.CSS_SELECTOR, 'td.txt_C')
            if len(tds) >= 2:  # Check if there are at least 2 <td> elements
                analysis_item = tds[1].text.strip()
                # Skip the '修正熱量' column
                if analysis_item == "修正熱量":
                    continue  # Skip this iteration if it's "修正熱量"
                if analysis_item in selected_items:
                    # Extract based on selected items
                    nutr.append(tds[3].text.strip())

        # Append the extracted info
        food_nutr.append(nutr)

    except Exception as e:
        print(f"Error extracting data: {e}")
        return []
```

資料處理及存取

1

Iterate list_data中的Links：讀取所有搜尋結果的頁面

2

營養數據轉成Pandas Dataframe

提取的營養數據會先轉換成Pandas DataFrame，方便後續的數據分析和視覺化。

3

存取為.csv檔案

DataFrame會儲存為一個.csv檔案，以便日後進行更進階的數據處理或分享。

```python
max_tabs = len(driver.window_handles)

#iterate over list_data (all of frozen food links)
for index, item in enumerate(list_data):
    url_link = item['Link']
    food_name = item['Food']

    # Switch tabs
    tab_index = (index % max_tabs)  # Cycle through tabs
    driver.switch_to.window(
            driver.window_handles[tab_index]
    )

    # 使分頁自動連結到指定網址 (此時的 drive 變數指向切後的分頁)
    driver.get(url_link)

    extract_food_info()
    extract_food_nutr()

    # You can also process or store the food_name if needed
    print(f"Processing food: {food_name} at {url_link}")

    sleep(3)

#check output
pprint.pprint(food_info, sort_dicts=False)
pprint.pprint(food_nutr)
```

✓ 7m 48.4s Python

```
Processing food: 冷凍火腿炒飯 at https://consumer.fda.gov.tw/Food/tfndDetail.aspx?nodeID=178&f=0&id=1753
Processing food: 冷凍蝦仁炒飯 at https://consumer.fda.gov.tw/Food/tfndDetail.aspx?nodeID=178&f=0&id=1776
Processing food: 冷凍筒仔米糕 at https://consumer.fda.gov.tw/Food/tfndDetail.aspx?nodeID=178&f=0&id=1797
Processing food: 冷凍芝麻湯圓 at https://consumer.fda.gov.tw/Food/tfndDetail.aspx?nodeID=178&f=0&id=1755
Processing food: 冷凍花生湯圓 at https://consumer.fda.gov.tw/Food/tfndDetail.aspx?nodeID=178&f=0&id=1756
```

```python
df = pd.DataFrame(food_nutr, columns=selected_items)
df
```

✓ 0.5s Python

	樣品名稱	熱量	粗蛋白	粗脂肪	飽和脂肪	總碳水化合物	膳食纖維	糖質總量	鈉	膽固醇	反式脂肪
0	冷凍火腿炒飯	189	5.0	5.6	1.1	29.7	2.0		220	37	
1	冷凍蝦仁炒飯	148	4.7	3.4	0.5	24.6	2.7	1.3	222	11	13.41
2	冷凍筒仔米糕	212	6.5	5.8	2.2	33.6	0.7		331	13	
3	冷凍芝麻湯圓	352	4.8	16.3	6.6	46.7	0.9	9.7	3	0	69.29
4	冷凍花生湯圓	350	5.2	15.6	7.3	47.1	0.6	9.8	6	0	66.28
...	...	...	...	...	...	...	...	...	...	...	...
76	冷凍魚卵卷	115	12.9	0.2	0.1	15.5		3.7	830	28	6.32
77	冷凍花枝漿	227	11.2	12.6	4.5	17.1	2.0	1.8	637	188	
78	冷凍花枝羹	126	11.4	4.7	1.8	9.4			448	80	
79	冷凍烤雞翅	217	18.8	14.2	3.9	3.6			509	109	
80	毛鱗魚(柳葉魚)(裹粉未炸)	171	12.9	4.9	1.6	19.0	0.6	1.6	493	112	39.19

81 rows × 11 columns

```python
df.to_csv('FrozenFood_ExtractedNutrInfo.csv', index=False)
```

優化與效能提升

限制標籤頁數量

通過控制同時打開的標籤頁數量，優化內存使用和提高運行效率。

數據過濾

實施數據過濾機制，專注於關鍵營養素，減少不必要的數據處理。

顯式等待

引入WebDriverWait，確保元素加載完成後再進行操作，提高穩定性。

並行處理

探索多進程或異步技術，進一步提升大規模數據擷取的效率。

挑戰與解決方案

- 重複的"熱量" 及
  重複的"每100克含量"
- 因每個row都是個別的tr > td.txttext_ca無法偵測CSS選擇器
- 後續須分析資料

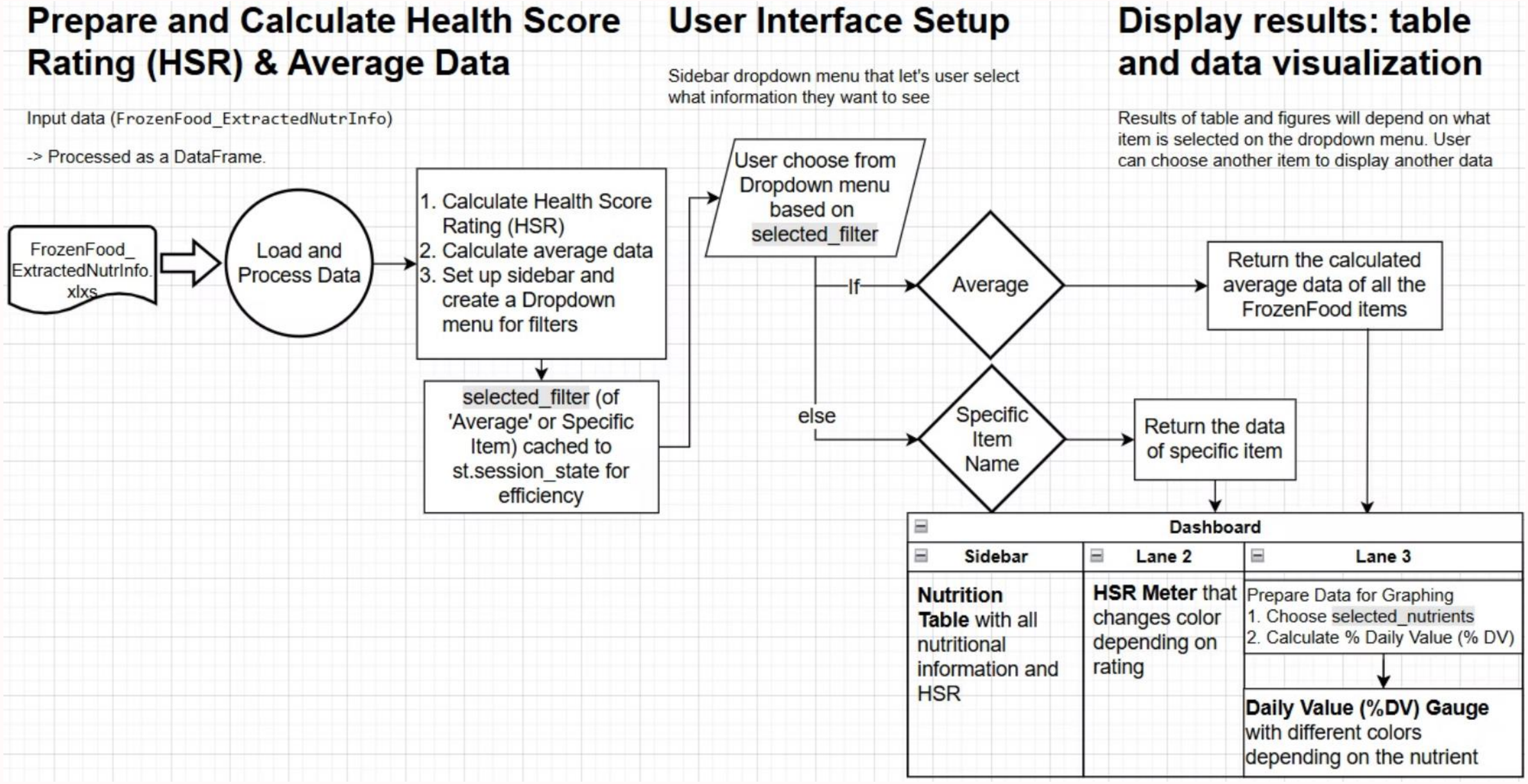
- 用continue跳過"修正熱量" 及用index擷取column 數據
- 先擷取tr全部，在用for loop & if statement 篩選需要的營養素資料
- 以list儲存value，以利後續使用pandas處理

分析項分類	分析項	單位	每100克含量	樣本數	標準差	每單位量(27.0克)含量x1	每100克含量
一般成分	熱量	kcal	171			46	171
一般成分	修正熱量	kcal	170			46	170
一般成分	水分	g	61.3	1		16.6	61.3
一般成分	粗蛋白	g	12.9	1		3.5	12.9
一般成分	粗脂肪	g	4.9	1		1.3	4.9
一般成分	飽和脂肪	g	1.6			0.4	1.6

程式解說

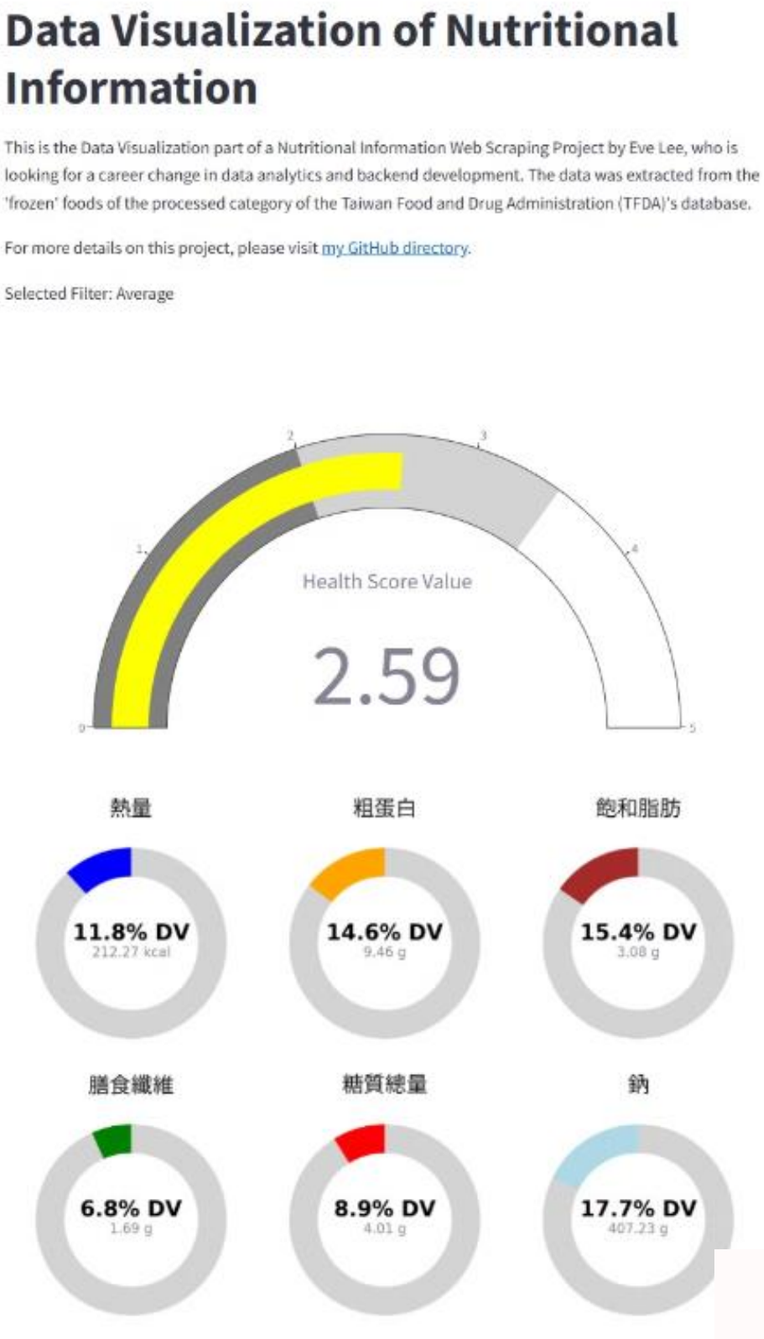
資料視覺化

Data Processing and Streamlit Dashboard



Sample Screenshot

Nutrient	Average	Unit per 100 g or ml
熱量	212.27	kcal
粗蛋白	9.46	g
粗脂肪	8.40	g
飽和脂肪	3.08	g
總碳水化合物	24.74	g
膳食纖維	1.69	g
糖質總量	4.01	g
鈉	407.23	g
膽固醇	37.10	g
反式脂肪	43.59	mg
HSR	2.59	Score



結論

結論與未來展望

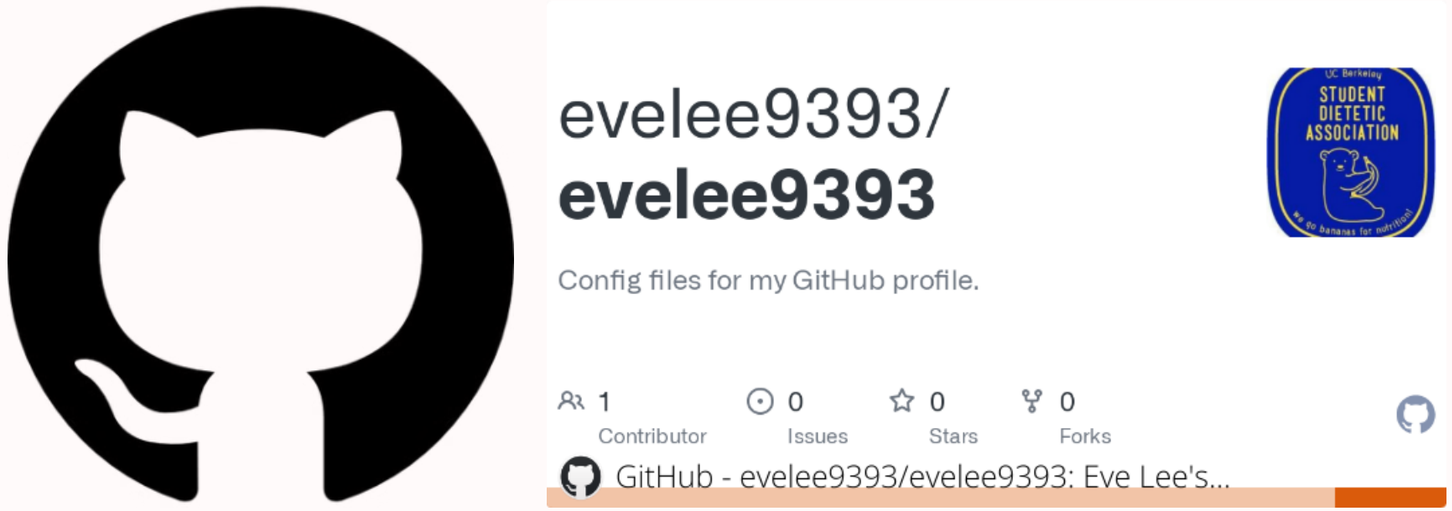
1. 成果總結：成功實現食品營養數據的自動化擷取和處理，為後續研究奠定基礎
2. 分析其他資料：此專案重點在資料視覺化，但可根據關鍵字和選擇的食品類別，使用者可以分析營養資訊資料以供其使用。
3. 整合 Nutri-Score API：為了更準確地呈現標籤，使用 nutri-score API 會更具說服力，本專案使用簡單的計算。
4. 擴展範圍：考慮擴展到其他食品類別或營養數據源，建立更全面的數據庫。
5. 更成熟的儀表板：加入比較、其他資料視覺效果和圖表。

References

- https://consumer.fda.gov.tw/Food/TFND.aspx?nodeID=178
- https://en.wikipedia.org/wiki/Health_Star_Rating_System
- https://www.researchgate.net/publication/335867875_A_Randomized_Controlled_Trial_Evaluating_the_Relative_Effectiveness_of_the_Multiple_Traffic_Light_and_Nutri-Score_Front_of_Package_Nutrition_Labels
- https://github.com/food-nutrients/nutri-score

李敬怡的 GitHub 頁面

歡迎到我的 GitHub 頁面，查看這個專案的原始碼和更多有趣的作品。



https://github.com/evelee9393/evelee9393

Thank you!