

2024 Jul

# 比價小幫手

Feebee scraping

周匡文 張明暄 林子婷 陳亞柔

# Content

01 專案動機

02 功能與特色

03 執行流程

- 網路爬蟲
- 資料整合
- 分析呈現

04 未來展望



# 01 專題動機



網路普及、需求增加



翻轉資訊落差



創造更佳 UI/UX



大幅提升效率與滿意度

# ON 功能與特色



## 資料獲取:

可查詢網購商品最便宜的通路



## 視覺化圖表:

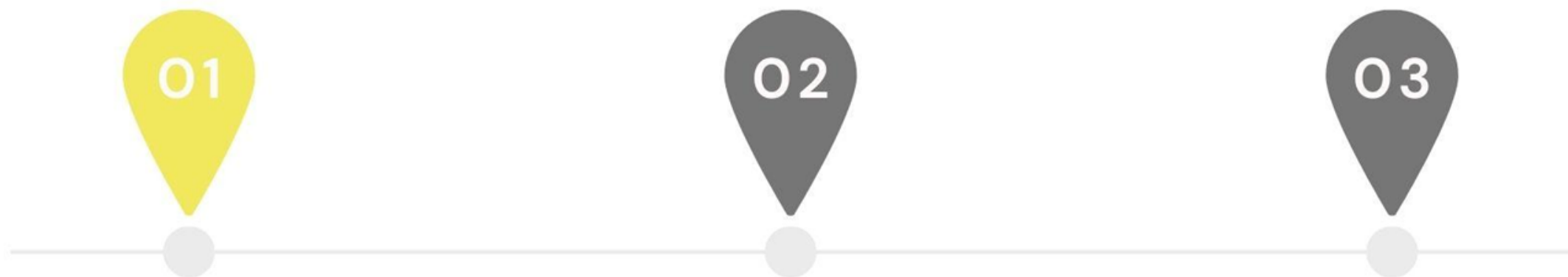
資料視覺化讓資訊一目了然



## Linebot:

提供介面讓使用者易於操作

## 03 執行流程



### 網路爬蟲：

輸入產品名稱後，程式可以  
抓取飛比價格比價網中大量  
的販售資訊

# #01 網路爬蟲 Web Crawler

- 解決搜尋準確度？

- 價格低到高？

- 相關度？

- 結論：

- 給予價格範圍

- 價格排序

比 飛比價格

首頁精選 分類瀏覽 今 | 商品情報 飛比點數 飛比

購物精選 拍賣商品 所有商品 熱搜 led水族燈3尺 小米 電

「iphone15 pro max」精選商品優惠價格低至高 價格由低到高 價格由高到低 相關度

**無卡分期** 優惠期間中  
手機、平板、筆電、遊戲機  
最高分期  
年滿18歲就可申辦 線上申辦  
(不須費過多等待時間)  
多家分期公司合作  
專人協助 提高過件率  
過件率高 免息分期  
(外島、非都會區皆可)

**iPhone 15 pro max / iphone 15 plus 無卡分期列車開跑, 全台皆可申辦**  
\$1  
樂天市場購物網 - 茗澄3C通訊 促銷 Summer Sale限定 5折起

**單顆鏡頭貼 適用於iphone 11 12 13 14 15 mini pro plus max 鏡頭圈**  
\$2  
蝦皮商城 - 上諭企業社 促銷 蝦皮免運馬拉松, 免運店家天天享蝦皮店到店\$99起免運無限次, 免領...

# 價格範圍：由低到高排序

feebee.com.tw/s/iphone%20pro%20max/?sort=p&pl=30000&ph=40000&ptab=1

飛比價格  登入/註冊

拍賣商品 所有商品 **熱搜** 葡眾 活逸康 pgo機車 遙控軍艦 哥吉拉公仔 明治巧喜糖 更多 >

one pro max 精選商品優惠價格低至高 **價格由低到高** 價格由高到低 相關度

**價格區間**

\$200 以下 \$200 ~ \$600

\$600 ~ \$4,400 \$4,400 ~ \$23,000

\$23,000 以上

補貨中的商品

☒ 顯示 ☐ 不顯示

**進階搜尋**

**價格範圍**

30000 ~ 40000

Apple iPhone 13 Pro Max 1024G 手機醫生認證二手機 保固12個月 K3數位

**\$30,228**

蝦皮商城 - K3數位福利機商城 | 手機醫生認證 促銷 蝦皮免運吃到飽, 購買免運店家商品, 蝦皮店到...

前往購買

Apple iPhone 14 Pro Max 256GB 太空黑

**\$30,290**

PChome24h購物

前往購買

# 例外處理

確保輸入的上下限(high/low price)  
讓程式可正確執行

```
def input_check(low_price, high_price):  
    #輸入檢查  
    try:  
        low_price = int(low_price)  
        high_price = int(high_price)  
    except ValueError:  
        print("價格輸入須為整數")  
        return False  
  
    if low_price > high_price:  
        print("最低價格不得大於最高價格")  
        return False  
  
    if low_price < 0 or high_price < 0:  
        print("價格輸入不得為負數")  
        return False  
  
    if high_price == 0:  
        print("最高價格不得為0")  
        return False  
  
    return True
```



## Package:

- Requests
- BeautifulSoup
- Pandas

```
async def scrape_feebee(product, min_price, max_price, pages=3):
    # 爬取 feebee 網站資訊
    if not input_check(min_price, max_price):
        return f"價格輸入錯誤或範圍不合理。請檢查最低價格和最高價格是否合理。"
    else:
        min_price, max_price = int(min_price), int(max_price)

    headers = {
        'User-Agent': 'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/102.0.0.0 Safari/537.36'
    }

    cheapest_products = {}

    for page in range(1, pages + 1):
        url = f'https://feebee.com.tw/s/{product}?sort=p&pl={min_price}&ph={max_price}&ptab=1&page={page}'
        html_source = await fetch_page(url, headers)

        if html_source is None:
            continue

        soup = BeautifulSoup(html_source, 'lxml')
```

## 最終結果

將前三頁中每個平台最低價的結果儲存成dataframe以繼續做視覺化分析

```
if not cheapest_products:
    return f'沒有找到符合條件的商品。\\n商品名稱: {product}\\n價格範圍: {min_price}-{max_price}'

data = [
    {
        '商品標題': product['title'],
        '通路': store,
        '價錢': product['price'],
        '連結': product['link'],
    }
    for store, product in cheapest_products.items()
]

df = pd.DataFrame(data)
sorted_df = df.sort_values(by='價錢').reset_index(drop=True)
print('商品資訊儲存成功')
```

## 03 執行流程

01

### 網路爬蟲：

輸入產品名稱後，程式可以  
抓取飛比價格比價網中大量的  
販售資訊

02

### 資料整合與視覺化：

只留下同通路最低價，將非  
目標的商品去除，視覺化整  
合好的資料

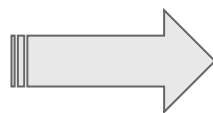
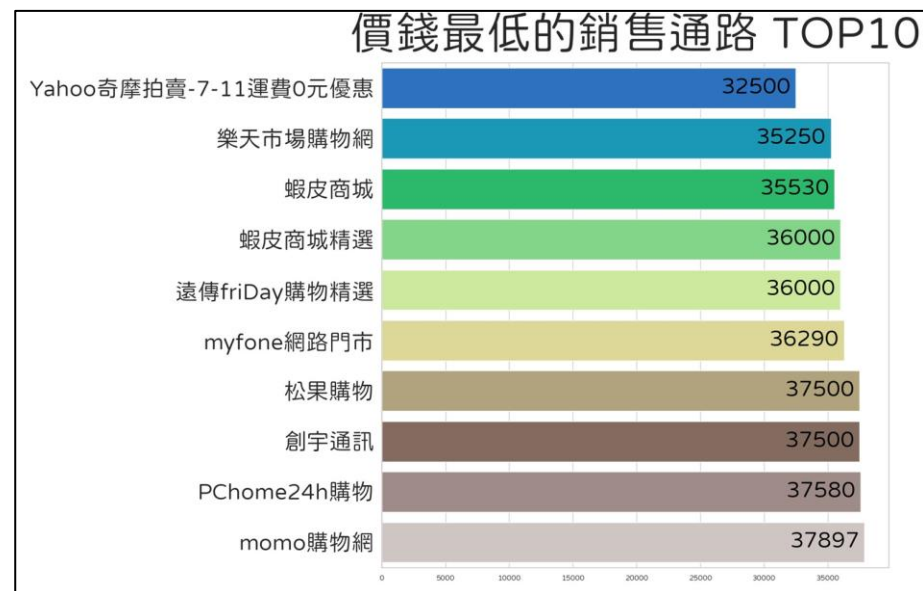
03

## #02 資料視覺化 Data Visualization

使用套件：

pandas、matplotlib、seaborn、ImgurClient

初版



優化版本



# Code

# Visualization

```
def price_chart(sorted_df, font_path, product, output_path='price_chart.png'):
    #繪製價格圖表

    # 加載中文字體：要先下載字體檔案
    font = FontProperties(fname=font_path)

    top_10_df = sorted_df.head(10)

    # 限制 Y 軸每筆資料名稱最多顯示前 10 個字
    top_10_df['通路'] = top_10_df['通路'].apply(lambda x: x[:10] + '...' if len(x) > 10 else x)

    # 設定繪圖風格
    sns.set(style="whitegrid")

    # 建立長條圖
    plt.figure(figsize=(12, 12))

    # 使用自定義的顏色漸層
    custom_palette = sns.color_palette("terrain", 10)
    bar_plot = sns.barplot(x='價錢', y='通路', data=top_10_df, palette=custom_palette, hue='通路', legend=False)

    # 添加資料標籤
    for index, row in top_10_df.iterrows():
        plt.text(row['價錢'], index, f'{row["價錢"]}', color = 'White', ha = "right", va = "center", fontproperties = font, fontsize = 40)

    # 設定標題和標籤，並設定中文字體
    plt.title(f'{product} \n價錢最低的銷售通路 TOP10', fontproperties=font, fontsize=60, y=1.05) # 設定
    plt.xlabel(" ", fontsize=5, labelpad=5)
    plt.ylabel(" ", fontsize=5, labelpad=5)

    # 隱藏 x 軸資料標籤
    plt.xticks([])

    # 設定資料標籤字體
    plt.yticks(fontproperties=font, fontsize=45) # 增大 y 軸資料標籤的字體大小
```

## 1. 加載中文字體

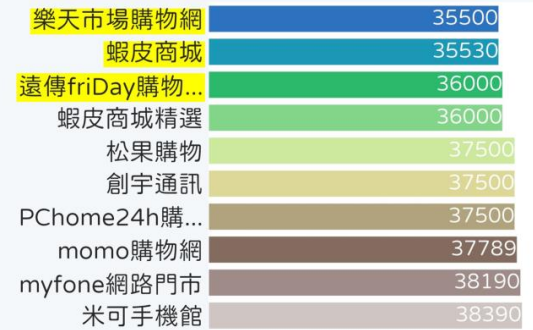
## 2. 限制 Y 軸字數

## 3. 增添資料標籤

## 4. 帶入商品名稱

## 5. 調整顯示參數

iphone15 pro max  
價錢最低的銷售通路 TOP10



## Code

## Visualization

```
# 為 Y 軸前三筆資料名稱加上底色
ytick_labels = bar_plot.get_yticklabels()
for i in range(3):
    ytick_labels[i].set_bbox(dict(facecolor='yellow', edgecolor='none'))
```

6. 增加醒目底色

```
# 移除邊框
ax = plt.gca()
for spine in ax.spines.values():
    spine.set_edgecolor('none') # 設定邊框顏色為透明
```

7. 移除圖表邊框

```
# 調整四周留白
plt.subplots_adjust(left = 0.15, right = 0.85, top = 0.85, bottom = 0.15)
```

8. 圖片四周留白

```
# 保存為 PNG 圖檔
plt.savefig(output_path, bbox_inches = 'tight', pad_inches = 1, facecolor = (0.9608, 0.9725, 0.9804)) # 設置背景顏色
# 保存為 PNG 圖檔
plt.savefig(output_path, bbox_inches='tight')
```

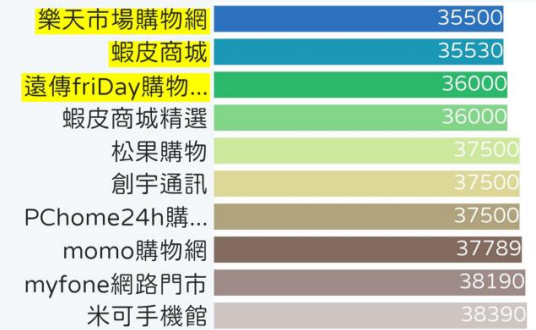
```
# 關閉圖表以釋放記憶體
plt.close()
```

9. 加上淺灰底色

```
# 重新打開圖像並添加彩色邊框
from PIL import Image, ImageOps
img = Image.open(output_path)
border_color = (211, 211, 211) # 邊框顏色
img_with_border = ImageOps.expand(img, border = 30, fill = border_color)
img_with_border.save(output_path)
```

10. 加上圖片邊框

iphone15 pro max  
價錢最低的銷售通路 TOP10



## 03 執行流程

01

### 網路爬蟲：

輸入產品名稱後，程式可以抓取飛比價格比價網中大量的販售資訊

02

### 資料整合與視覺化：

只留下同通路最低價，將非目標的商品去除，視覺化整合好的資料

03

### 價格分析與呈現：

運用Linebot，使用者輸入想查詢的商品，即可挑選出最佳的販售通路並提供分析圖

## #03 LineBot

- 主要套件：

- Flask
- LineBotApi

- 其他相關工具：

- LIFF
- ngrok



# 商品查詢

# 函式：處理表單提交

```
@app.route("/submit", methods=['POST'])
```

```
def submit():
```

```
    try:
```

```
        data = request.get_json()
```

```
        product_name = data.get('product_name')
```

```
        min_price = data.get('min_price')
```

```
        max_price = data.get('max_price')
```

```
        line_user_id = data.get('line_user_id')
```

```
        print(product_name, min_price, max_price, line_user_id)
```

```
    # 觸發爬蟲並取得用戶消息
```

```
    if user_text == '查詢':
```

```
        user_message = asyncio.run(scrape_feebee(product_name, min_price, max_price,))
```

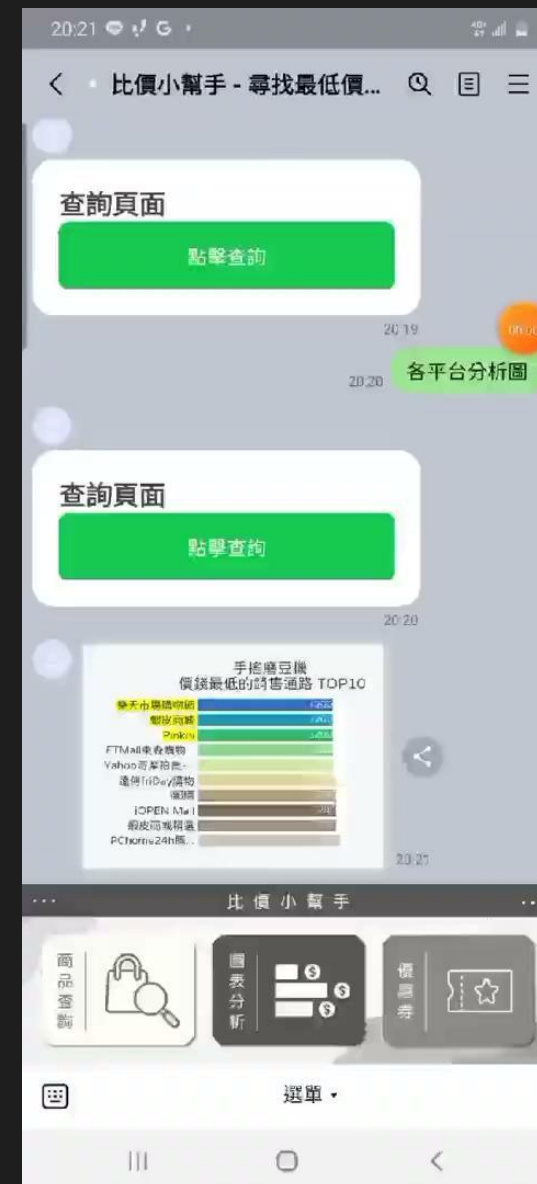


# 商品各平台分析圖

```
# 觸發爬蟲並取得用戶消息
if user_text == '查詢':
    user_message = asyncio.run(scrape_feebee(product_name, min_price, max_price,))
elif user_text == '各平台分析圖':
    chart_path = asyncio.run(scrape_feebee(product_name, min_price, max_price,))
    if chart_path:
        img_url = upload_image_to_imgur(imgur_client, chart_path)

        if img_url:
            line_user_id,
            ImageSendMessage(
                original_content_url=img_url,
                preview_image_url=img_url
            )

            # 刪除本地保存的圖像
            os.remove(chart_path)
            response_message = "圖片已成功發送到 LINE"
        else:
            response_message = "無法上傳圖片到 Imgur"
    else:
        response_message = "無法生成圖表"
```

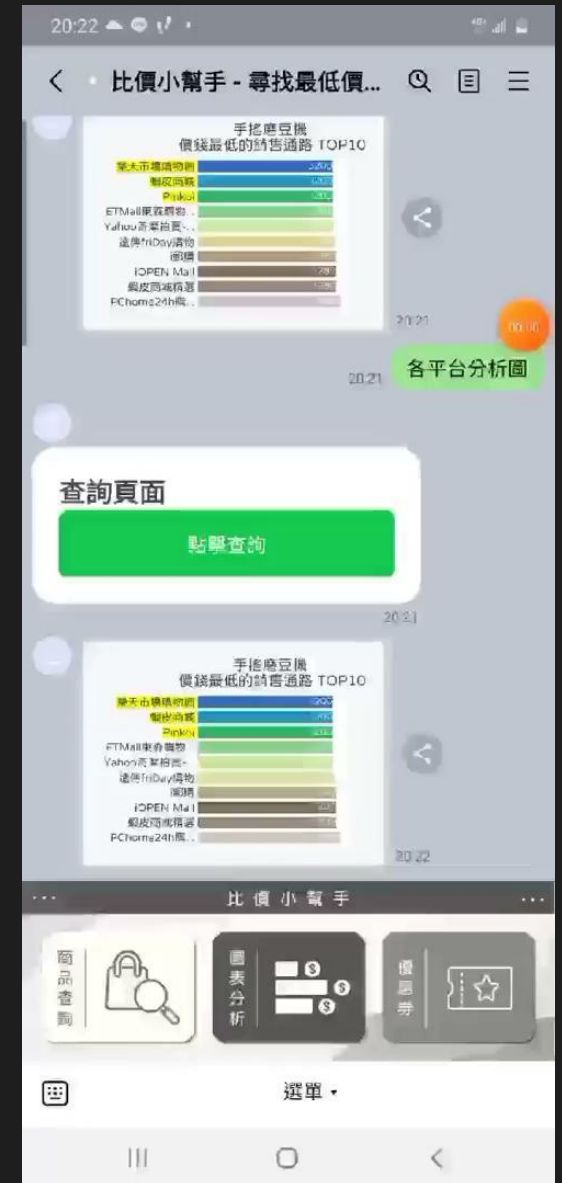


# 優惠券資訊

```
elif event.message.text == "優惠券查詢":
    try:
        message=TextSendMessage(
            text="請選擇平台",
            quick_reply=QuickReply(
                items=[
                    QuickReplyButton(
                        action=MessageAction(label="蝦皮",text="蝦皮")
                    ),
                    QuickReplyButton(
                        action=MessageAction(label="露天",text="露天")
                    ),
                    QuickReplyButton(
                        action=MessageAction(label="PChome",text="PChome")
                    ),
                    QuickReplyButton(
                        action=MessageAction(label="momo",text="momo")
                    ),
                    QuickReplyButton(
                        action=MessageAction(label="博客來",text="博客來")
                    ),
                ]
            )
        )
    except:
        pass

elif event.message.text == "蝦皮":
    message=TextSendMessage(text="蝦皮優惠券:https://liff.line.me/2005730520-aM2Le0R3")
    line_bot_api.reply_message(event.reply_token, message)

elif event.message.text == "露天":
    message=TextSendMessage(text="露天優惠券:https://liff.line.me/2005730526-qwbV6vyd")
    line_bot_api.reply_message(event.reply_token, message)
```



## 04 未來發展



# Thank you!

Feebee scraping