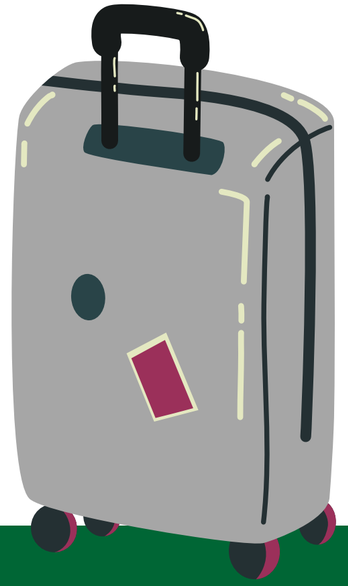
Python資料分析與機器學習應用

旅遊小幫手

生傳三 黃韻文 ｜ 財金三 黃敏瑄 ｜ 物理四 簡詩汶 ｜
社工三 王柏詒 ｜ 機械四 賴欣妤



大綱

1. 最終成果預期之重要性與貢獻性
2. 專案內容介紹（含差異）
3. 資料清理與模型
4. 專案 demo

1 最終成果預期之重要性與貢獻性

專案重要性

專案價值

觀光統計年報資訊過量

- 報告達 480 頁，**缺乏視覺化圖表**
- 趨勢分析僅依季度與區域，**無針對個別景點**
- 建議僅針對**大方向，無個別地區細緻化的發展建議**

少以數據為主的建議

- 少有以實證資料為主的研究建議
- 網路社群的旅遊評價多描述「主觀感受」

- **改善數據量過大、可讀性低、資料分散等問題**
- **以視覺化數據協助旅客作出判斷**

專案重要性

專案價值

疫情趨緩後，國旅商機恢復，旅遊變得越加重要，而我們的專案...

旅客

- 評估各行政區域的房價均值
- 旅遊選址時搭配房價作參考

旅宿業者

- 依據每月人流預測調整預算投入
- 依據旅遊型態分析擬定營運策略

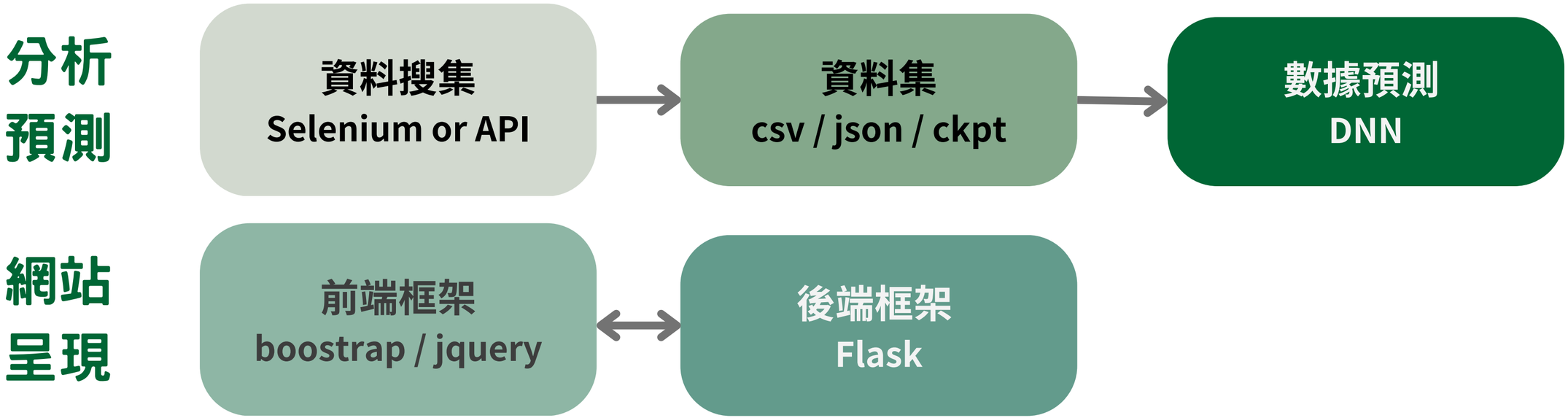
2 專案內容介紹

期中期末比較

專案介紹

專案主題	期中規劃	期末調整	完成度
預測	• 使用政府開放資料 • 爬取天氣、平假日、房價 • 預測未來每日人流量	• **景點每月人流量預測** • **旅館每日房價預測**	**90%**
分析	• 觀光局每月景點人流量 • 爬取天氣、平假日 • 景點分群、行銷建議	**無**	**100%**
呈現	自架網站	**無**	**100%**

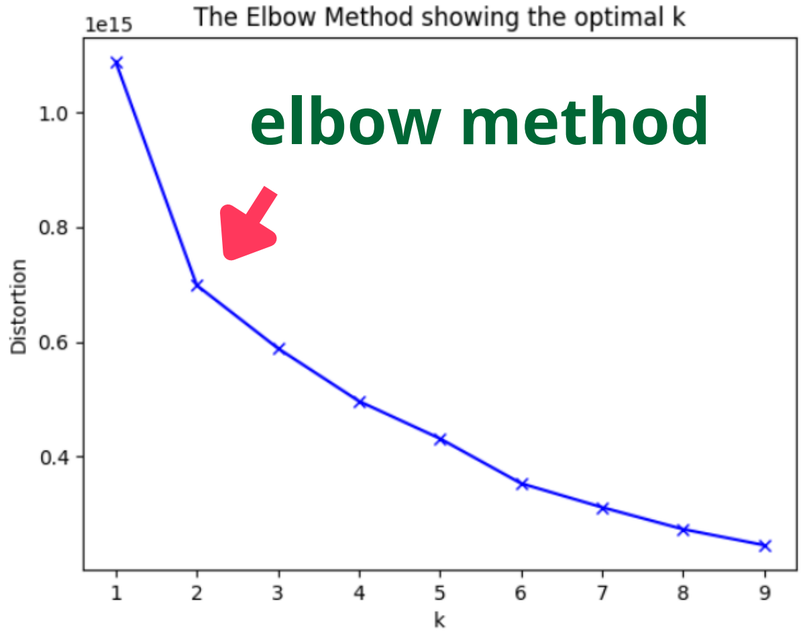
功能：房價查詢、景點分群說明

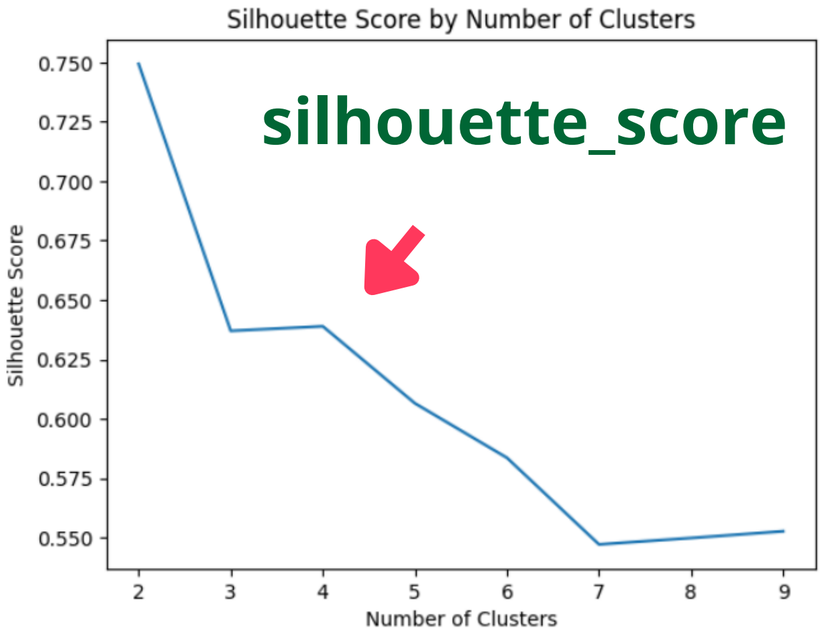


3 專案 Demo

4 資料前處理與建模

KMEANS-模型





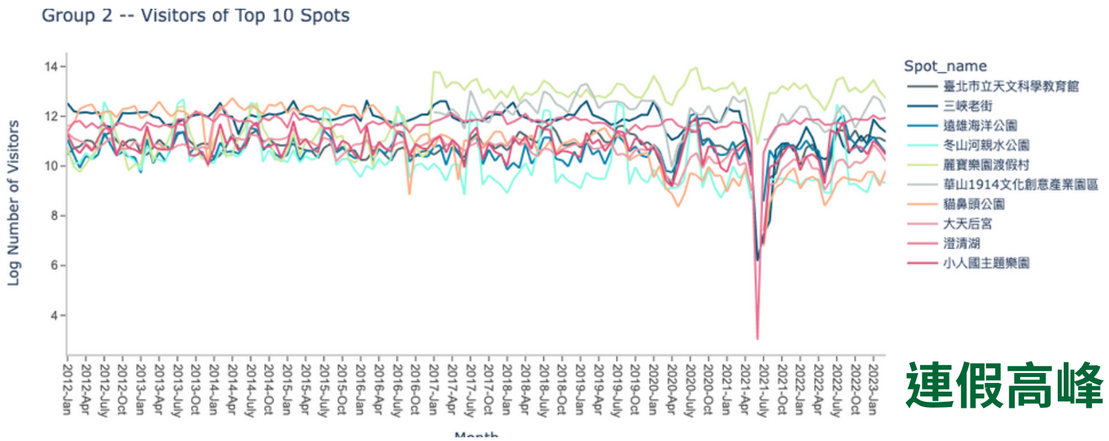

```
# cluster the data into n clusters using K-Means
clf_KMeans = KMeans(n_clusters= 2, random_state=0, n_init="auto")
y_pred = clf_KMeans.fit(X_true)  ## 分群
print(y_pred)
```

```
# cluster the data into n clusters using K-Means
clf_KMeans = KMeans(n_clusters= 4, random_state=0, n_init="auto")
y_pred = clf_KMeans.fit(X_true)  ## 分群
print(y_pred)
```

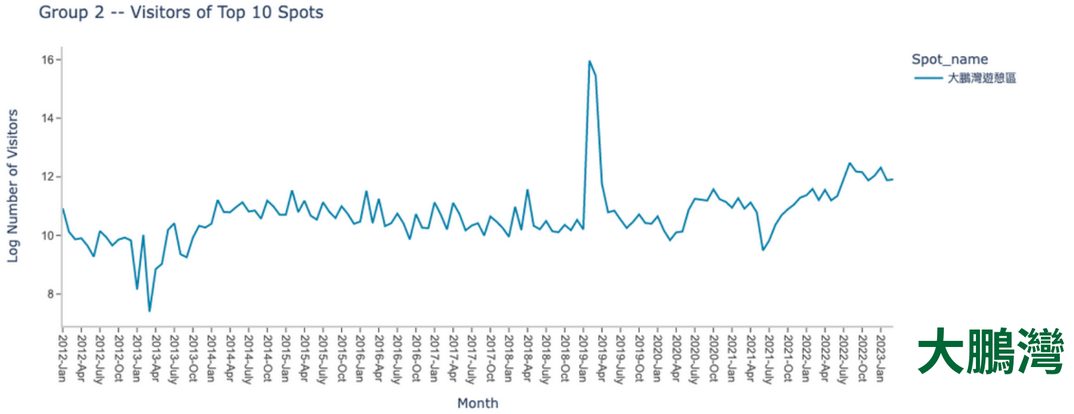
351 個景點資料

景點型態分析 | 房價預測 | 景點人流預測

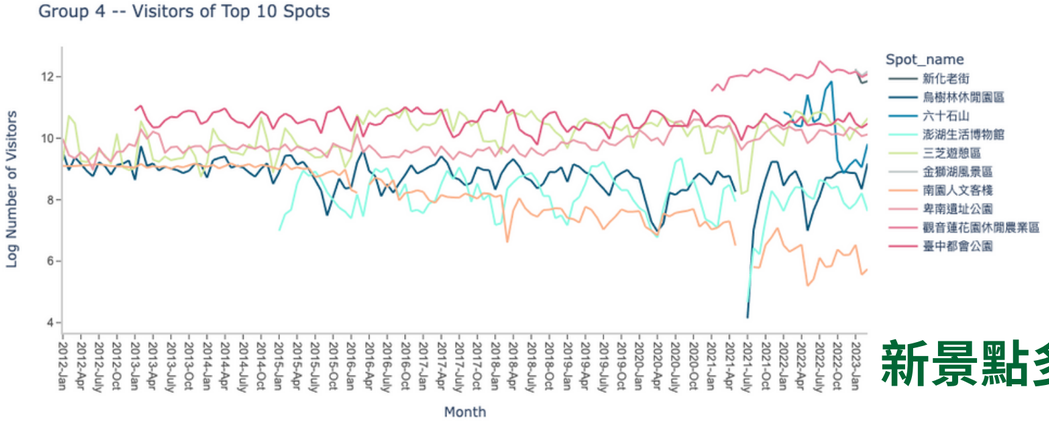
KMEANS-模型



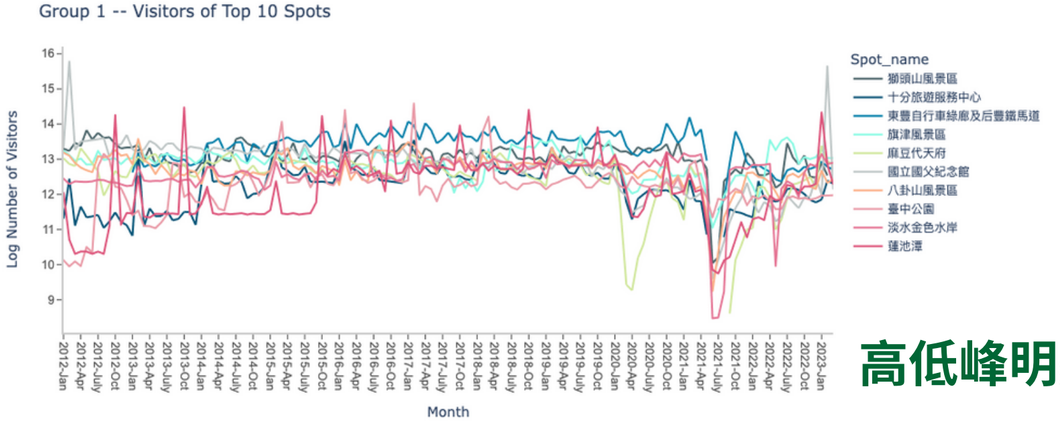
連假高峰



大鵬灣



新景點多



高低峰明顯

景點型態分析 | **景點人流預測** | 房價預測

觀光局資料集

Taiwan 交通部觀光局觀光統計資料庫
Tourism Statistics Database of the Taiwan Tourism Bureau

回首頁Home 網站導覽Sitemap

資料庫簡介 Introduction | 來臺資料 Inbound visitors | 出國資料 Outbound travelers | 觀光遊憩據點 Scenic Spots | 郵輪旅客 Visitor Arrivals by cruise | 統計報表 Statistics

首頁 Home / 觀光遊憩據點 Scenic Spots

觀光遊憩據點 Scenic Spots

單一類別查詢 Single Category Search | 交叉類別查詢 Cross Category Search | 遊憩區更名紀錄 Record of the changes in names of recreational areas

統計方式 (Year or Month) ○年表(Year) ◉月表(Month)

時間區間 (Time) 112(2023) 3 ~ 112(2023) 3

※觀光局每年選定之觀光遊憩據點非屬全面，且定期滾動增刪，本系統僅刊載部分據點資訊故不另予加總。

※The tourism and recreational areas selected by the Tourism Bureau every year are not comprehensive, and are added and deleted on a regular basis. This system only publishes information on some areas, and does not cover all the tourism sites in Taiwan.

遊憩據點 Scenic spots

請選擇區域(Select) 請選擇縣市(Select)

- 全部 (All)
- 鼻頭龍洞遊憩區 (Bitou Longdong Recreation Area)
- 大里外澳遊憩區 (Dali Waiao Recreation Area)
- 小野柳(加路蘭) (Jialulan)
- 八仙洞 (Basian Cave)
- 綠島 (Lyudao (Green Island))
- 石梯坪 (Shihtiping)
- 遠雄海洋公園 (Farglory Ocean Park)
- 鹽寮福隆遊憩區 (Yanliao Fujong Recreation Area)
- 宜蘭濱海遊憩區 (Yilan Ocean Recreation Area)
- 三仙臺 (Sansiantai)
- 秀姑巒溪遊客中心 (Siouguluan River)
- 都歷處本部 (East Coast National Scenic Area Administration Headquarters)
- 花蓮管理站遊客中心 (Hualien Ranger Station Visitor Center)
- 澎湖遊客中心 (Penghu Visitor Center)

資料前處理

label / one-hot encoding

pandas.melt

Spot_name	Spot_type	County	Year	Jan	Feb	Mar	Apr	May
陽明山遊客中心	國家公園	臺北市	2012	12187.0	18612.0	18580.0	13329.0	14179.0
陽明書屋	國家公園	臺北市	2012	2597.0	3617.0	5221.0	4317.0	3497.0
陽明公園	國家公園	臺北市	2012	161000.0	463000.0	705200.0	85600.0	38600.0
大屯遊憩區	國家公園	臺北市	2012	17047.0	29079.0	38558.0	34332.0	43906.0
龍鳳谷遊憩區	國家公園	臺北市	2012	21201.0	22134.0	62447.0	64691.0	16243.0

資料前處理

label / one-hot encoding

pandas.melt

Spot_code	County_encode	Spot_type_encode	population	Year	month
0	13	3	12187.0	2012	1
1	13	3	2597.0	2012	1
2	13	3	161000.0	2012	1
3	13	3	17047.0	2012	1
4	13	3	21201.0	2012	1

資料前處理

Label / OntHot Encoder

```
from sklearn.preprocessing import LabelEncoder
labelencoder = LabelEncoder()
data_le = labelencoder.fit_transform(df[['County']])
df['County_encode'] = data_le
data_le = labelencoder.fit_transform(df[['Spot_type']])
df['Spot_type_encode'] = data_le
```

```
from sklearn.preprocessing import OneHotEncoder
onehotencoder = OneHotEncoder()
data_str_ohe=onehotencoder.fit_transform(df[['Spot_type', 'County']]).toarray()
df_clean = pd.concat([df, pd.DataFrame(data_str_ohe,
                columns=onehotencoder.get_feature_names_out(['Spot_type', 'County']))]
                ,axis=1)
```

把前幾個月的人流量當成 feature

Year	Spot_code	County_encode	Spot_type_encode	month	population	group	last_month_ppl	last_2_month_ppl	last_3_month_ppl
2015	1	13	3	1	2811.0	0	2446.0	3010.0	2597.0
2016	1	13	3	1	2846.0	0	2811.0	2446.0	3010.0
2017	1	13	3	1	2834.0	0	2846.0	2811.0	2446.0
2018	1	13	3	1	1616.0	0	2834.0	2846.0	2811.0
2019	1	13	3	1	8761.0	0	1616.0	2834.0	2846.0

人流量預測 - DNN 模型

```
class My_spot_Model(nn.Module):
    def __init__(self, input_dim):
        super(My_spot_Model, self).__init__()
        self.layers = nn.Sequential(
            nn.Linear(input_dim, 128),
            nn.LeakyReLU(),
            nn.Linear(128, 48),
            nn.LeakyReLU(),
            nn.Linear(48, 16),
            nn.LeakyReLU(),
            nn.Linear(16, 24),
            nn.LeakyReLU(),
            nn.Linear(24, 8),
            nn.LeakyReLU(),
            nn.Linear(8, 1),
            nn.ReLU(),
        )
```

可調參數

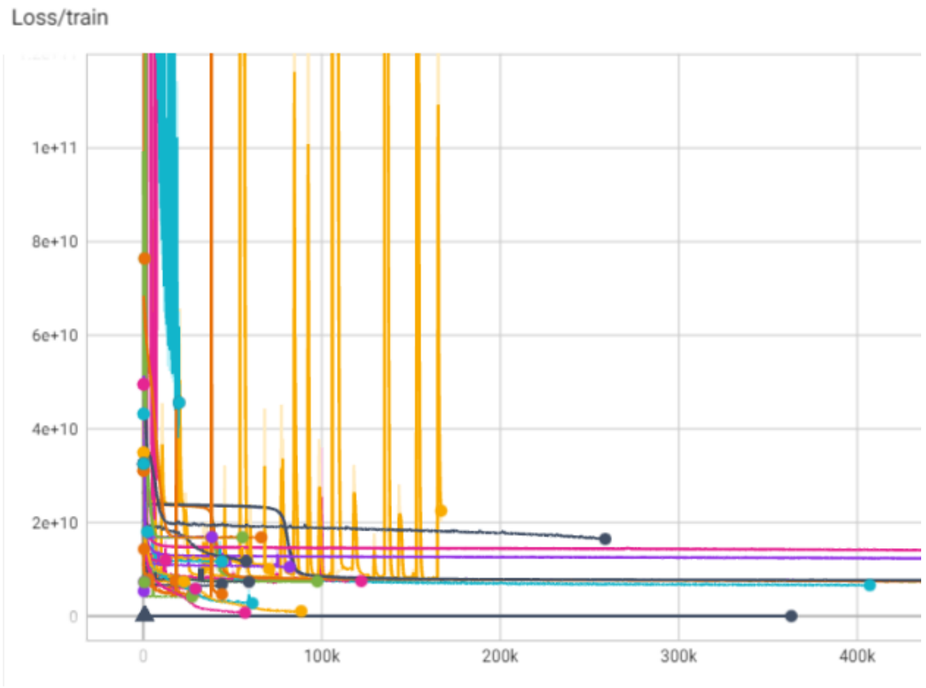
- valid_ratio: 0.2
- n_epochs: 3000
- batch_size: 256
- learning_rate: 1e-3
- early_stop: 300

不同資料集版

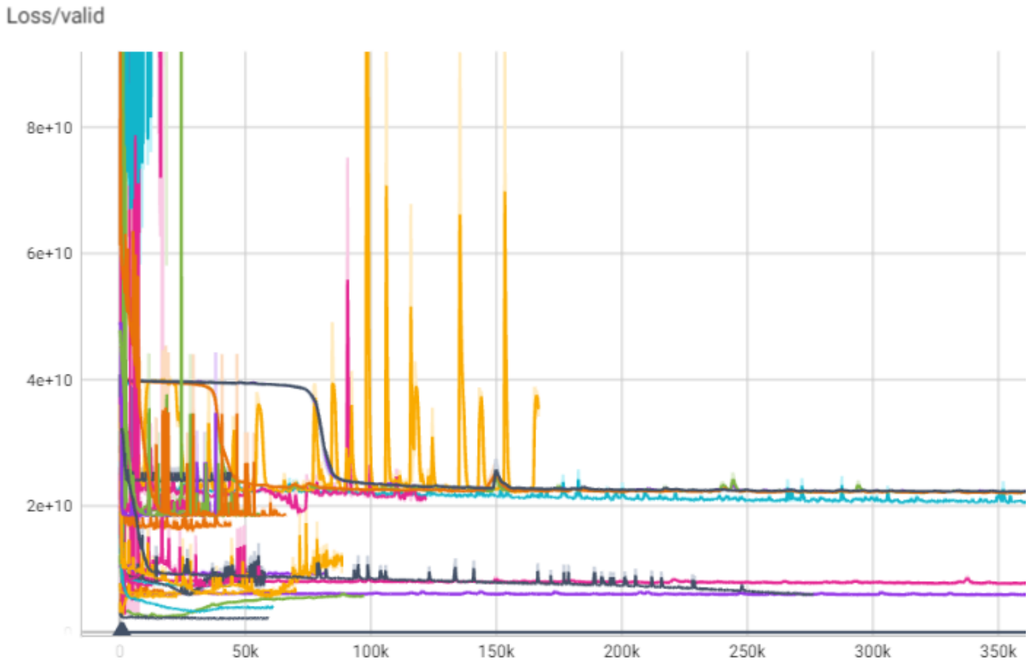
- one-hot encode / label encode
- MinMaxScaler / StandardScaler

模型訓練

training loss



valid loss



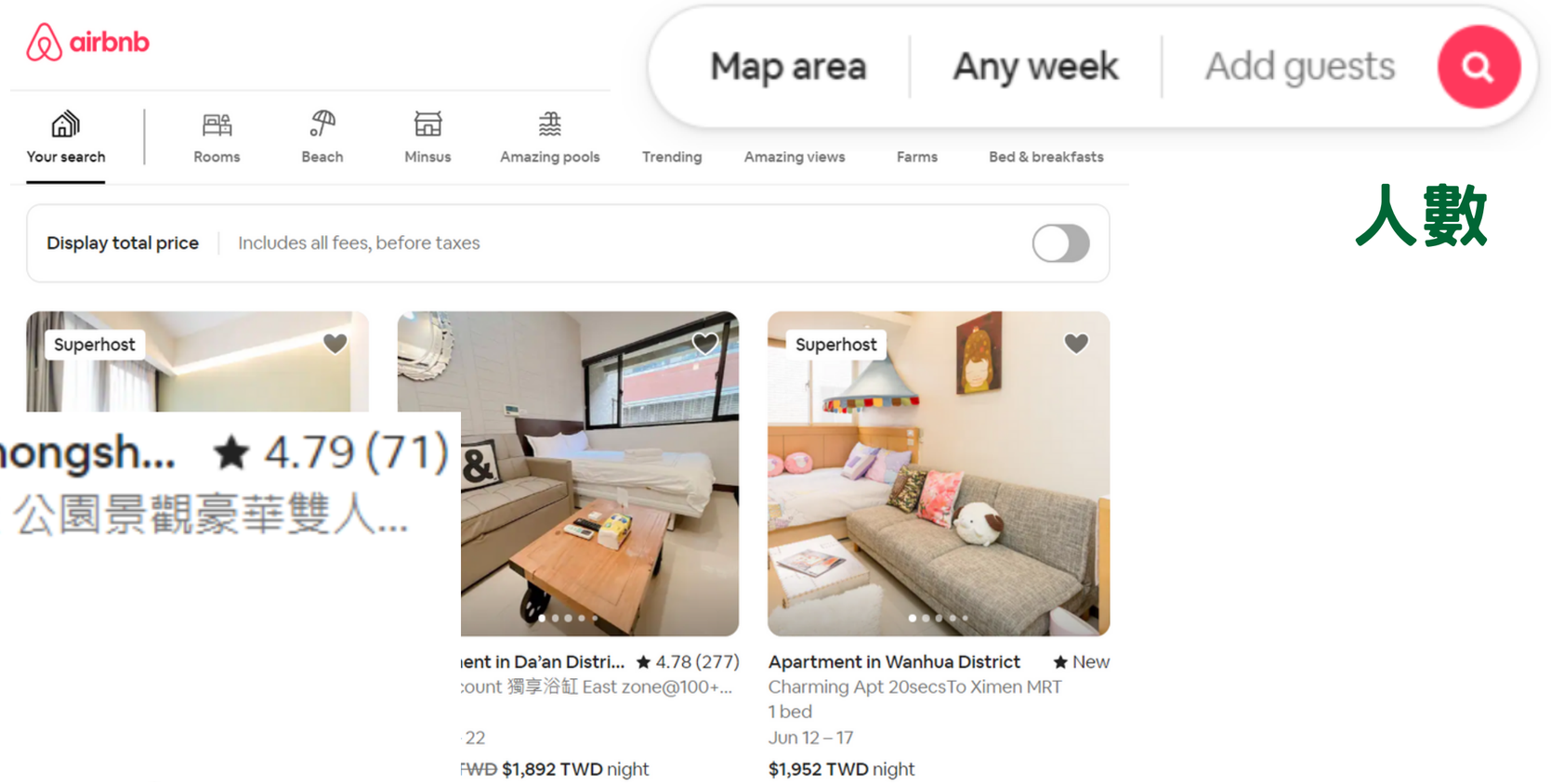
模型評估

MSE：61861.75

```
from sklearn.metrics import mean_squared_error
mean_squared_error(test_data[:, -1], preds, squared=False)
```

景點型態分析 | 景點人流預測 | **房價預測**

Airbnb網站爬蟲



地區、價錢

人數

Airbnb網站爬蟲

```python
def scrap_this_page(dataframe, url, checkin, checkout, adults, children, infants, pets):
    # Create selector
    html = requests.get(url).content
    sel  = Selector(text=html)

    # Get hotels
    hotels = sel.css('div.c4mnd7m')

    # Select the first announcement from the previous list of 20
    for i in range(len(hotels)):
        hotel = hotels[i]
        # Get main information
        title            = hotel.css('div[data-testid="listing-card-title"] ::text').extract_first()
        price            = hotel.css('span._tyxjp1 ::text').extract_first()
        if price == None:
            ori_price = hotel.css('span._1ks8cgb ::text').extract_first()
            dis_price = hotel.css('span._1y74zjx ::text').extract_first()
        else:
            ori_price = price
            dis_price = price
        rating           = hotel.css('span.r1dxllyb ::text').extract_first()
        url              = hotel.css('a.bn2bl2p ::attr(href)').extract_first()
        timestamp        = datetime.datetime.now().strftime("%Y-%m-%d %H:%M:%S")
        dataframe.loc[len(dataframe)] = [title, ori_price, dis_price, rating, checkin, checkout, adults, children, infants, pets, timestamp, main_url+url]
    return sel, dataframe
```

爬此頁

加入 dataframe

```python
def to_next_page(sel, page_i):
    next_page = sel.css('a.c1ackr0h ::attr(href)').extract()[page_i]
    return f'{main_url}{next_page}'
```

進入下一頁

天氣預報爬蟲

```python
def scrap_weather():
    import requests
    import pandas as pd
    from datetime import datetime

    df_pre = pd.DataFrame(columns=('date','region',
                                   'rain_0','rain_1','rain_2','rain_3','rain_4','rain_5','rain_6',
                                   'temp_0','temp_1','temp_2','temp_3','temp_4','temp_5','temp_6',
                                   'humi_0','humi_1','humi_2','humi_3','humi_4','humi_5','humi_6'))

    today = datetime.today().date()
    url = 'https://opendata.cwb.gov.tw/api/v1/rest/datastore/F-D0047-063?Authorization=CWB-8CB6
    data = requests.get(url)
    data_json = data.json()
    location = data_json['records']['locations'][0]['location']

    for i in range(len(location)):
        rain = location[i]['weatherElement'][0]
        temperature = location[i]['weatherElement'][1]
        humidity = location[i]['weatherElement'][2]
```

爬取當天-六天後的
雨量、氣溫、濕度

資料從氣象局
開放的 api 擷取

Replace string with regex

```
df = pd.read_csv('airbnb_taipei_spec_date_clean_copy.csv', encoding='unicode_escape')
df = df.replace({"Ã": "", "ƒ": "", "Â":"", ",":"", "TWD": "", ",":"", " ": "", r'\xa0':"", "\$": ""}, regex=
```

Normalization

```
from sklearn.preprocessing import MinMaxScaler
scaler = MinMaxScaler()
train.iloc[:, :-1] = scaler.fit_transform(train.iloc[:, :-1])
test.iloc[:, :-1] = scaler.transform(test.iloc[:, :-1])
```

原始數據

title	ori_price	dis_price	rating	checkin	checkout	adults	children	infants	pets	timestamp	
Hotel in Zhongzheng District	$3,000Â TWD	$3,000Â TWD	NaN	2023-05-28	2023-05-29	1	3	5	0	2023-05-28 16:35:00	https://www.airbnb.com/rooms/638252082882106842?adults=28&check_out=2023-05-29&previous_page_section_name=1000&fec
Apartment in Zhongshan District	$2,800Â TWD	$2,800Â TWD	NaN	2023-05-28	2023-05-29	1	3	5	0	2023-05-28 16:35:00	https://www.airbnb.com/rooms/577259487465118795?adults=28&check_out=2023-05-29&previous_page_section_name=1000&fec

Airbnb爬蟲資料 (merge) 捷運站數資料 (merge) 天氣爬蟲資料

整併數據

	adults	children	infants	pets	checkin_weekday	discount	title_code	mrt_count	rain_0	rain_1	temp_0	temp_1	humi_0	humi_1	dis_price
0	0.933333	0.733333	0.6	0.0	0.666667	0.000000	0.363636	1.000000	0.285714	0.000000	0.666667	0.428571	0.310345	0.190476	25106.000000
1	0.066667	0.066667	0.2	0.6	0.500000	0.000000	0.363636	1.000000	0.285714	0.000000	0.666667	0.428571	0.310345	0.190476	2538.000000
2	0.200000	0.066667	0.8	0.2	0.666667	0.000000	0.363636	1.000000	0.142857	0.000000	0.333333	0.571429	0.103448	0.095238	4770.000000
3	0.000000	0.000000	0.2	0.0	0.500000	0.004729	1.000000	0.888889	0.857143	1.000000	0.166667	0.285714	0.758621	0.714286	950.181818
4	0.200000	0.000000	0.2	0.6	0.166667	0.000000	1.000000	0.888889	0.571429	0.666667	0.166667	0.285714	0.793103	0.809524	1381.000000

建立模型

Airbnb - DNN 模型

```python
class My_airbnb_Model(nn.Module):
    def __init__(self, input_dim):
        super(My_airbnb_Model, self).__init__()
        self.layers = nn.Sequential(
            nn.Linear(input_dim, 128),
            nn.LeakyReLU(),
            nn.Linear(128, 48),
            nn.LeakyReLU(),
            nn.Linear(48, 16),
            nn.LeakyReLU(),
            nn.Linear(16, 24),
            nn.LeakyReLU(),
            nn.Linear(24, 8),
            nn.LeakyReLU(),
            nn.Linear(8, 1)
        )
```

可調參數

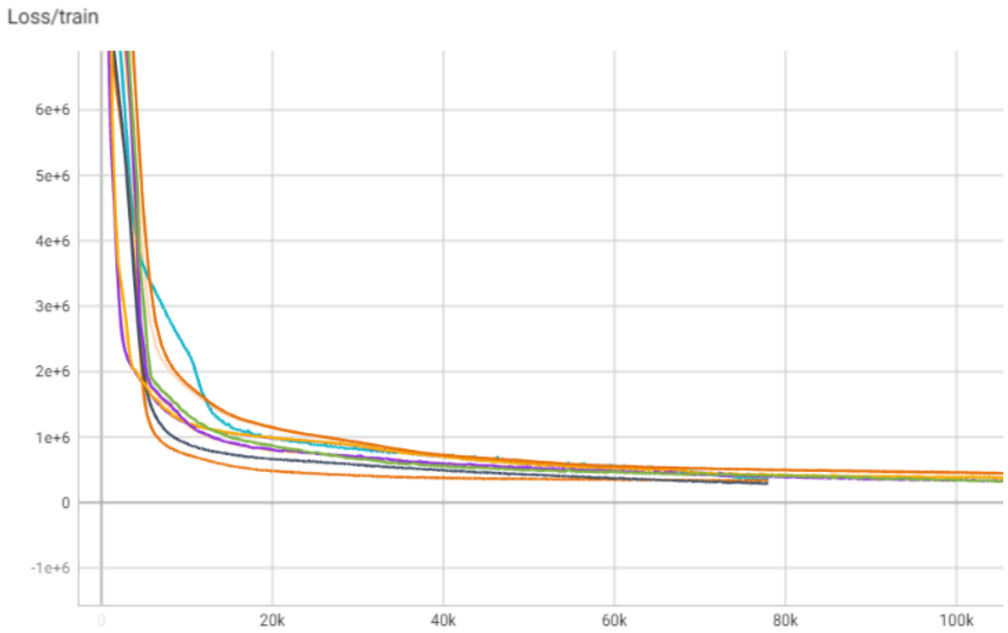
- valid_ratio: 0.2
- n_epochs: 3000
- batch_size: 256
- learning_rate: 1e-3
- early_stop: 300

不同資料集版

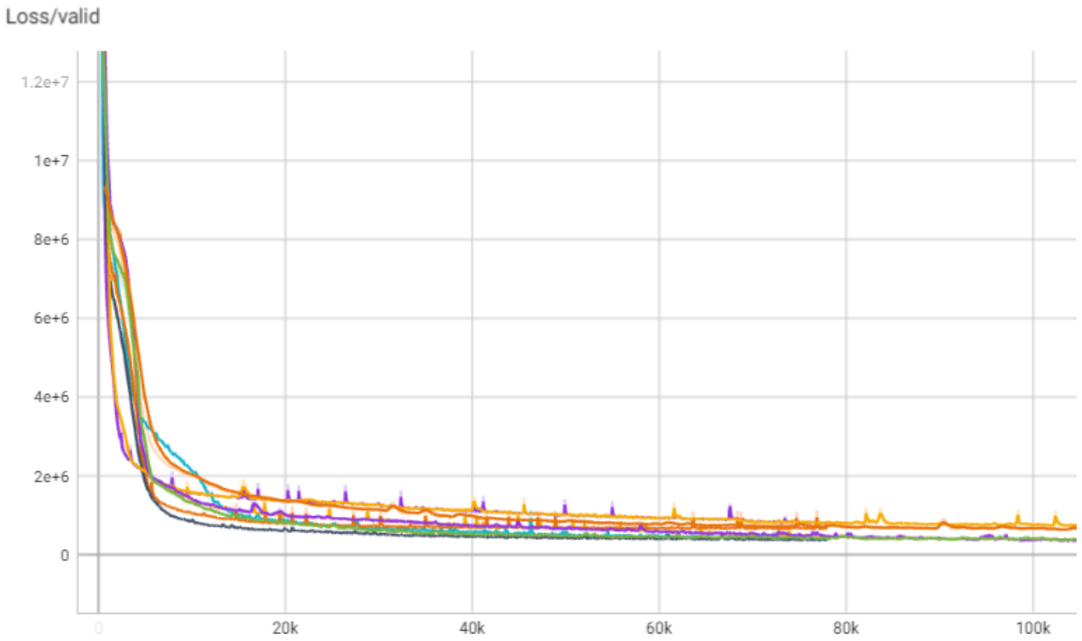
- one-hot encode / label encode
- MinMaxScaler / StandardScaler

模型訓練

training loss



valid loss



模型評估

MSE ： 549.04

```
from sklearn.metrics import mean_squared_error
mean_squared_error(test_data[:, -1], preds, squared=False)
```

Python資料分析與機器學習應用

旅遊小幫手

感謝大家