大台北房價你高攀不起，像極了愛情。

大台北地區房價預測

BDSE21 第二組

翁婉容、陳冠中、趙怡瑄、李惠萍

專題大綱

1 | **專案簡介**
團隊介紹與專題概要

2 | **環境處理**
叢集建置

3 | **資料探索**
資料蒐集與資料清洗

4 | **模型建置**
模型選用與預測

5 | **視覺化**
網頁呈現

6 | **總結**
結果分析

Chapter 1 專案簡介

講者：翁婉容

1 | 團隊成員介紹

2 | 研究動機

3 | 研究目的

4 | 商業價值

團隊成員介紹

翁婉容

組長

模型建置

資料清洗

Hadoop環境建置

陳冠中

組員

資料探索/清洗

視覺化

Hadoop環境建置

趙怡瑄

組員

Hadoop環境建置

資料清洗

模型建置

李惠萍

組員

視覺化

資料清洗

Hadoop環境建置

研究動機

翁婉容

2020年台北市房價

62.1萬/坪

相較2019年 ↑3.7%

2020年新北市房價

33.4萬/坪

相較2019年 ↑3.4%

研究目的

翁婉容

利用大數據

購屋議價空間

分析趨勢、預測房價

最完整的房價網站

商業價值

翁婉容

專業角度

1. 房市預測與趨勢分析。
2. 探勘雙北地區未來有升值潛力不動產，精準投資房產在未來的精華地區。
3. 讓買房者迅速了解複雜的房市交易資料並掌握房市走向。
4. 互動式網頁。

專題流程與工具

翁婉容

系統建置

ubuntu

APACHE Spark

hadoop

資料蒐集

python

Selenium

BeautifulSoup

Requests

資料清洗

python

pandas

NumPy

模型建置

scikit learn

dmlc XGBoost

RANDOM FOREST

PROPHET

APACHE Spark

視覺化

HTML JS CSS

Flask web development, one drop at a time

Leaflet

Power BI

HEROKU

Bootstrap

Chapter 2 環境處理

講者：趙怡瑄

1 | Hadoop 叢集系統

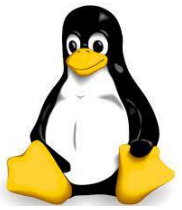
2 | Hadoop 叢集硬體配置

3 | Hadoop 叢集架構

4 | Hadoop 叢集效能實測

Hadoop 叢集系統

趙怡瑄

應用層	PySpark python jupyter
資源分配管理層	hadoop YARN
儲存層	hadoop HDFS
OS	vmware  ubuntu

Hadoop 叢集硬體配置

趙怡瑄

	實體主機	Hadoop VMs	Spark VMs
台數	6	12	6
CPU/Each	4 Cores 邏輯處理: 8	2 Cores	2 Cores
RAM/Each	32 GB	8 GB	4 GB
HDD/Each	1 TB	500 GB	500 GB

Total 虛擬主機

- 18 台
- CPU : 24 Cores
- RAM : 96 GB

Hadoop 叢集架構

趙怡瑄

Spark Nodes

bdse159.example.org

bdse31.example.org

bdse143.example.org

bdse95.example.org

bdse151.example.org

bdse183.example.org

Namenode / Secondary namenode
bdse153.example.org

DataNode
處理分散式檔案系統的資料

Resource Manager
bdse25.example.org

Node Manager
負責監控Hadoop叢集內每台機器的資源使用情況

Job History Server
bdse137.example.org

Workers

bdse154.example.org

bdse26.example.org

bdse138.example.org

bdse89.example.org

bdse90.example.org

bdse177.example.org

bdse178.example.org

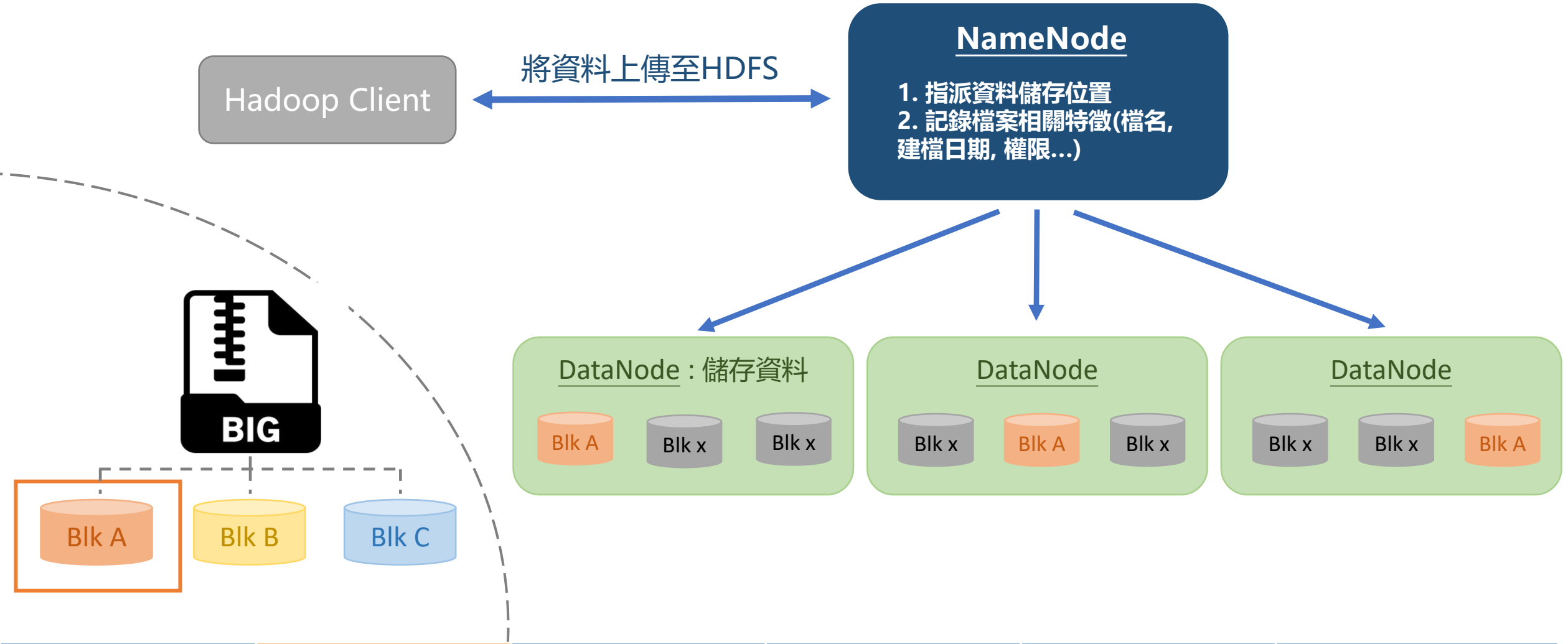
bdse145.example.org

bdse146.example.org

bdse137.example.org

Hadoop 叢集-HDFS

趙怡瑄



Hadoop 叢集-YARN

趙怡瑄

Edge node
(Spark node : bdse183)

Hi RM!! 我想要跑一支 python 程式，請給我 executor 執行程式

```
hadoop@bdse183:~$ spark-submit     --master yarn     --driver-memory 2G
   --driver-cores 2     --executor-memory 6G     --executor-cores 2
-num-executors 99 /home/hadoop/apps/rfr.py
```

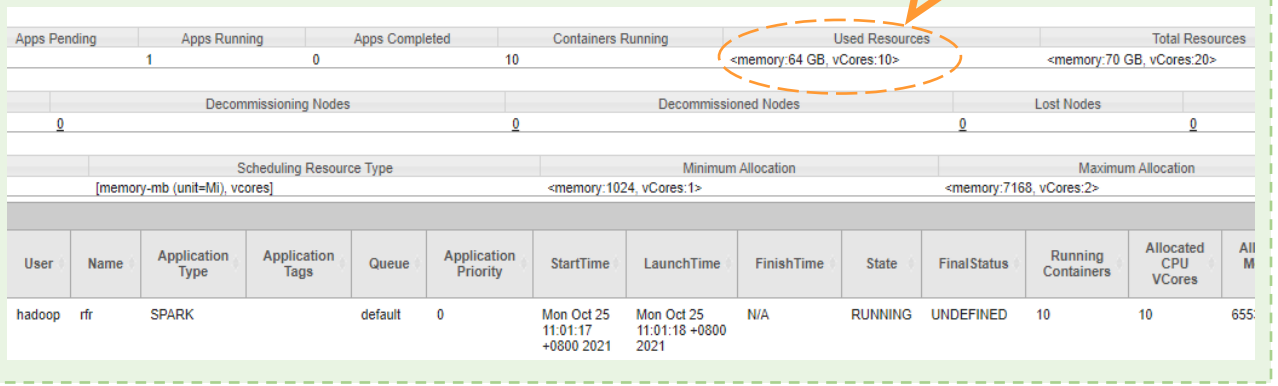
Resource Manager

負責整個系統的資源管理和分配

NM 接收 RM 的資源分配請求

RM Web UI 查看動用資源：

- 10 vCores 開 10 個 container
- 每個 container 相當於一個 executor



Used Resources
<memory:64 GB, vCores:10>

Apps Pending	Apps Running	Apps Completed	Containers Running	Used Resources	Total Resources
	1	0	10	<memory:64 GB, vCores:10>	<memory:70 GB, vCores:20>

Decommissioning Nodes	Decommissioned Nodes	Lost Nodes	
0	0	0	0

Scheduling Resource Type	Minimum Allocation	Maximum Allocation
[memory-mb (unit=Mi), vcores]	<memory:1024, vCores:1>	<memory:7168, vCores:2>

User	Name	Application Type	Application Tags	Queue	Application Priority	StartTime	LaunchTime	FinishTime	State	FinalStatus	Running Containers	Allocated CPU VCores	All M
hadoop	rfr	SPARK		default	0	Mon Oct 25 11:01:17 +0800 2021	Mon Oct 25 11:01:18 +0800 2021	N/A	RUNNING	UNDEFINED	10	10	655

Node Manager

分配具體的 Container 給程式，
同時負責監控並回報 Container 使用資訊（cpu和記憶體等資源）
給 ResourceManager

Container (executor) — rfr

Container (executor) — rfr

同時啟動一支程式，分散在各個點運算，平行處理不同的資料

Hadoop 叢集效能實測

趙怡瑄

資料量大小: 59.86 MB

- **Windows 本機**
- Scikit-learn
- 時間 : 7 分 55 秒

時間節省
近40倍

勝

- **Spark on Yarn**
- Pyspark.ml
- 時間 : 14.22 秒

```
In [14]: %%time

         rfr = RandomForestRegressor()
         rfr.fit(X_train, y_train.ravel())

         Wall time: 7min 55s

Out[14]: RandomForestRegressor()
```

```
[40]: start_time = time.time()
      rf = RandomForestRegressor(featuresCol='scaledFeatures',labelCol='unit price_ping', numTrees=100)
      rf_model = rf.fit(train_df)
      end_time=time.time()
      # predict on the test set
      model_predictions = rf_model.transform(test_df)
      evaluator=RegressionEvaluator(labelCol='unit price_ping',predictionCol='prediction')
      # test_data pred
      model_predictions = rf_model.transform(test_df)
      rmse=evaluator.evaluate(model_predictions,{evaluator.metricName: "rmse"})
      r2 = evaluator.evaluate(model_predictions,{evaluator.metricName: "r2"})

      print(f'rf_model training time: {end_time - start_time} seconds')
      print('-'*30)
      print('RandomForestRegressor evaluation numTrees=100:')
      print(f'rmse : {rmse}')
      print(f'r2 : {r2}')

      rf_model training time: 14.216609716415405 seconds
      ------------------------------
      RandomForestRegressor evaluation numTrees=100:
      rmse : 99312.08745976086
```

Chapter 3 資料探索

講者：陳冠中 & 趙怡瑄

1 | 資料來源

2 | 網路爬蟲

3 | 資料處理

4 | 相關矩陣

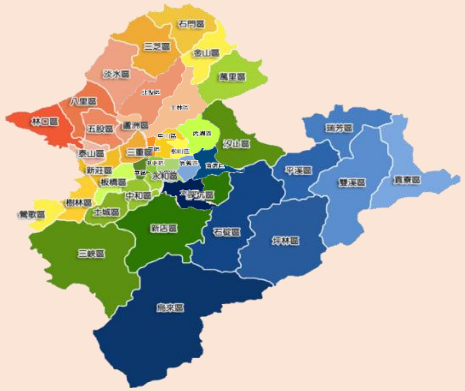
5 | 目標變量與特徵

資料來源與類型

陳冠中

內政部
不動產資訊平台

- 臺北市 & 新北市: 共41個行政區
- 民國102年~民國110年第2季
- 不含預售屋買賣、不動產租賃
- 原始資料：689,149筆



政府資料開放平臺
DATA.GOV.TW

- 捷運 ： 臺北捷運所有出入口之座標
- 學區 ： 大台北升學率前16名之明星國中
- 醫院 ： 區域醫院與醫療中心
- 超商 ： 7-11. 全家. OK. 萊爾富
- 公園
- 診所

網路爬蟲

目的: 學區資訊配對

陳冠中

Selenium

- **模擬使用者的動態操作**

  開啟瀏覽器/鍵盤輸入/滑鼠點擊。

- **HTML定位爬取文字**

  第一筆搜尋結果之鄰、里，若無則回傳NOT FOUND。

Demo

資料處理-流程

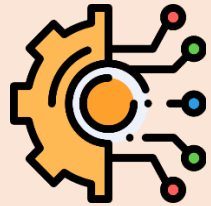
陳冠中

清洗

不動產資訊

資料篩選

空缺值處理



類別編碼

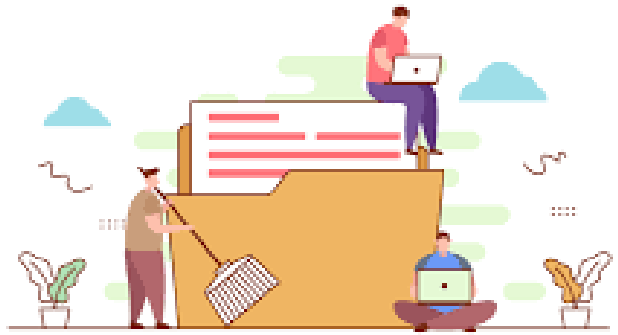
離群值處理

新增

生活機能

座標轉換

範圍密集度



資料處理-篩選

目的: 過濾資料使符合專題目標

陳冠中

- **移除建築完成年月為空值的資料**

  建築完成年月→屋齡

- **篩選建物型態**

  保留: 住宅大樓/華廈/套房/公寓/透天厝

  移除: 工廠/倉庫/農舍...等非居住建物型態

- **篩選交易標的**

  保留: 房地 / 房地+車位 / 建物

  移除: 單純車位、土地交易資料項

篩選建物型態

```
In [10]: df['建物型態']

Out[10]: 0          住宅大樓(11層含以上有電梯)
         1            華廈(10層含以下有電梯)
         2            華廈(10層含以下有電梯)
         3               套房(1房1廳1衛)
         4            華廈(10層含以下有電梯)
                        ...
         563408     住宅大樓(11層含以上有電梯)
         563409     住宅大樓(11層含以上有電梯)
         563410     住宅大樓(11層含以上有電梯)
         563411     住宅大樓(11層含以上有電梯)
         563412     住宅大樓(11層含以上有電梯)
         Name: 建物型態, Length: 563413, dtype: object

In [14]: df['建物型態'].unique()

Out[14]: array(['住宅大樓(11層含以上有電梯)', '華廈(10層含以下有電梯)', '套房(1房1廳1衛)', '公寓(5樓含以下無電梯)',
                '辦公商業大樓', '其他', '廠辦', '店面(店鋪)', '透天厝', '工廠', '倉庫', '農舍'],
               dtype=object)

In [15]: # 篩選建物型態
         filt = (df['建物型態']=="住宅大樓(11層含以上有電梯)") | \
                (df['建物型態']=="華廈(10層含以下有電梯)") | \
                (df['建物型態']=="套房(1房1廳1衛)") | \
                (df['建物型態']=="公寓(5樓含以下無電梯)") | \
                (df['建物型態']=="其他") | \
                (df['建物型態']=="透天厝")
         filt
```

資料處理-空缺值

目的: 提升數據完整性

陳冠中

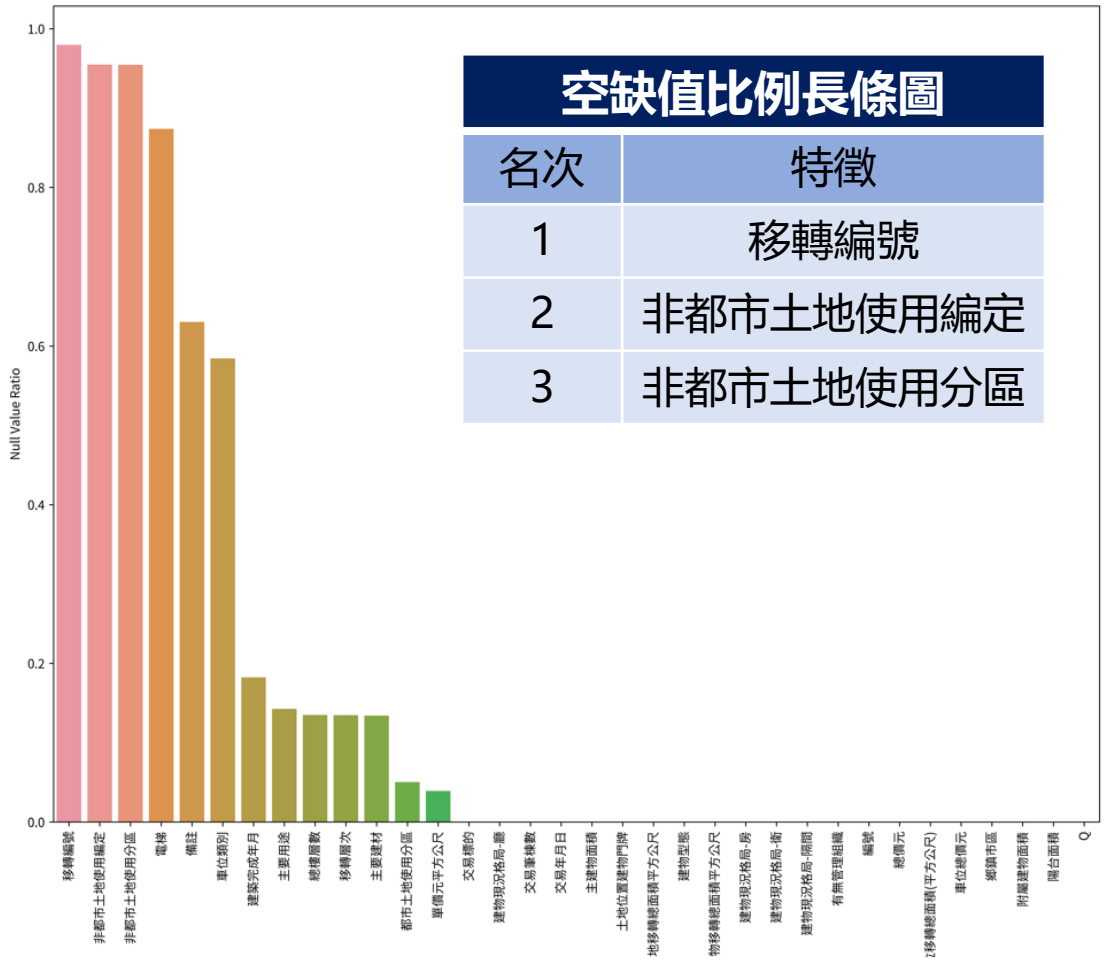
- **以其他相關欄位進行運算填補**

  總價元 ÷ 建物移轉總面積平方公尺 = 單價元平方公尺

- **移除空值過多且非專題探討之特徵欄位**

  如: 移轉編號/非都市土地使用編定

  /非都市土地使用分區...等



空缺值比例長條圖

名次	特徵
1	移轉編號
2	非都市土地使用編定
3	非都市土地使用分區

資料處理-類別編碼

目的: ML模型針對數值進行運算

陳冠中

LABEL ENCODER

```
[ ]  from sklearn.preprocessing import LabelEncoder
     construction_type_le = LabelEncoder()
     df['建物型態'] = construction_type_le.fit_transform(df['建物型態'].values)
     df['建物型態']
```

```
0         0
1         4
2         4
3         3
4         4
         ..
526441    0
526442    0
526443    0
526444    0
526445    0
Name: 建物型態, Length: 526446, dtype: int64
```

手動編碼 (MAP函數)

```
[5]  #  建立對應字典
     #  將類標籤從字符串轉換為整數
     class_mapping = {label: idx for idx, label in enumerate(np.unique(df['鄉鎮市區']))}
     df['鄉鎮市區'] = df['鄉鎮市區'].map(class_mapping)
     df['鄉鎮市區']
```

```
0         18
1         18
2         18
3         18
4         18
          ..
526441     2
526442     2
526443     2
526444     2
526445     2
Name: 鄉鎮市區, Length: 526446, dtype: int64
```

資料處理-離群值處理

目的: 提升機器學習模型成效

陳冠中

移除資料筆數過少的年份

→移除2012年以前資料

依每年、各行政區製作房價分布圖→右偏分布

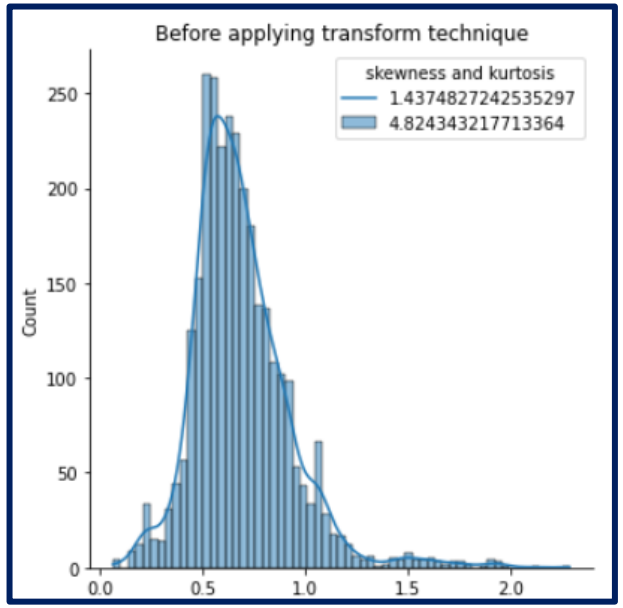
以IQR四分位距法移除離群值→常態分布

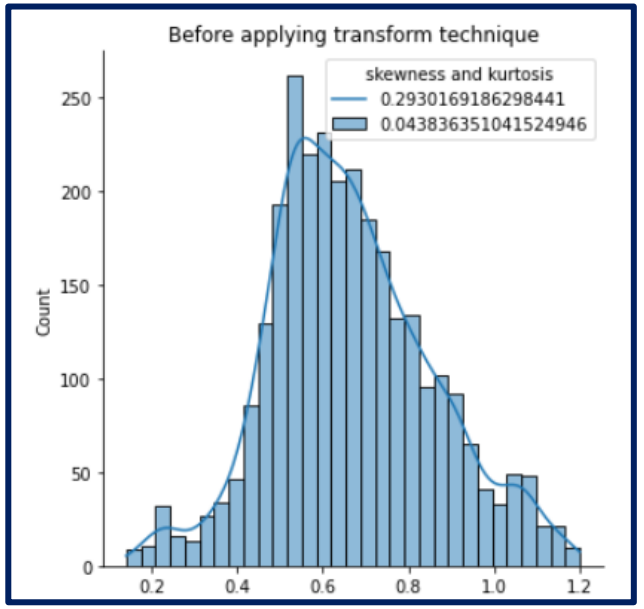
公式: Q1- (1.5*IQR) 及 Q3+(1.5*1QR)

```
In [6]: # 每年的交易筆數
        df['交易西元年'].value_counts(sort=True)

Out[6]: 2013    87489
        2020    64969
        2014    64167
        2019    58651
        2018    54975
        2017    53455
        2015    50644
        2016    45913
        2012    23017
        2021    19998
        2011     1442
        2010     1388
        2009      291
        2008       24
        2007       14
        2006        2
        1975        1
        2005        1
        2001        1
        1998        1
        1992        1
        1989        1
        1974        1
        Name: 交易西元年,
```

2011	1442
2010	1388
2009	291
2008	24
2007	14
2006	2
1975	1
2005	1
2001	1
1998	1
1992	1
1989	1
1974	1





資料處理-座標轉換

目的: 便於計算距離與密集度

陳冠中

Google Cloud Platform

Geocoding API

Google Enterprise API

- 藉由**地址**轉換取得**經緯度**
- **困難**: 使用額度/老舊地址

```
def get_latitude_longtitude(address):
    # decode url
    Output = []
    for i in range(len(address)):
        print(i)
        temp = urllib.parse.quote(address[i])
        #記得改key!!!
        url = "https://maps.googleapis.com/maps/api/geocode/json?address=" + temp + "&key=
        while True:
            res = requests.get(url)
            js = json.loads(res.text)
```

```
+ temp + "&key=  Google API Key  "
```

地址

資料處理-範圍密集度

目的: 分析生活機能對房價之影響

陳冠中

- **依生活機能特性調整特徵型態**

例如:

區域級以上醫院→密集度低
→距離範圍內是否存在(1,0)

便利商店→密集度高
→距離範圍內密集度(店家數)

```
#計算地址的經緯度到捷運的距離;  lat:緯度;  lng:經度
#lat1,lng1:  目標地址的經緯度
#lat2,lng2:  捷運座標
def  Distance(lat1,lng1,lat2,lng2):
        radlat1=radians(lat1)
        radlat2=radians(lat2)
        a=radlat1-radlat2
        b=radians(lng1)-radians(lng2)
        s=2*asin(sqrt(pow(sin(a/2),2)+cos(radlat1)*cos(radlat2)*pow(sin(b/2),2)))
        earth_radius=6378.137
        s=abs(s*earth_radius)
        #目標地址的經緯度  &  捷運座標  相差0.3km，則return  1
        if  s<=0.3:
                return  1
        else:
                return  0
```

資料處理-總結

陳冠中

- 資料總筆數: 689,149筆 → 508,431筆
- 特徵欄位: 33欄
- 資料型態: 數值型態 + 日期型態

- **其他資料轉換工程**

✓ 地址切割/校正
✓ 土地單位轉換
✓ 屋齡計算
✓ 日期資料格式轉換
✓ 生活機能資料格式轉換

```
In [13]: df_final.reset_index(drop=True)
```

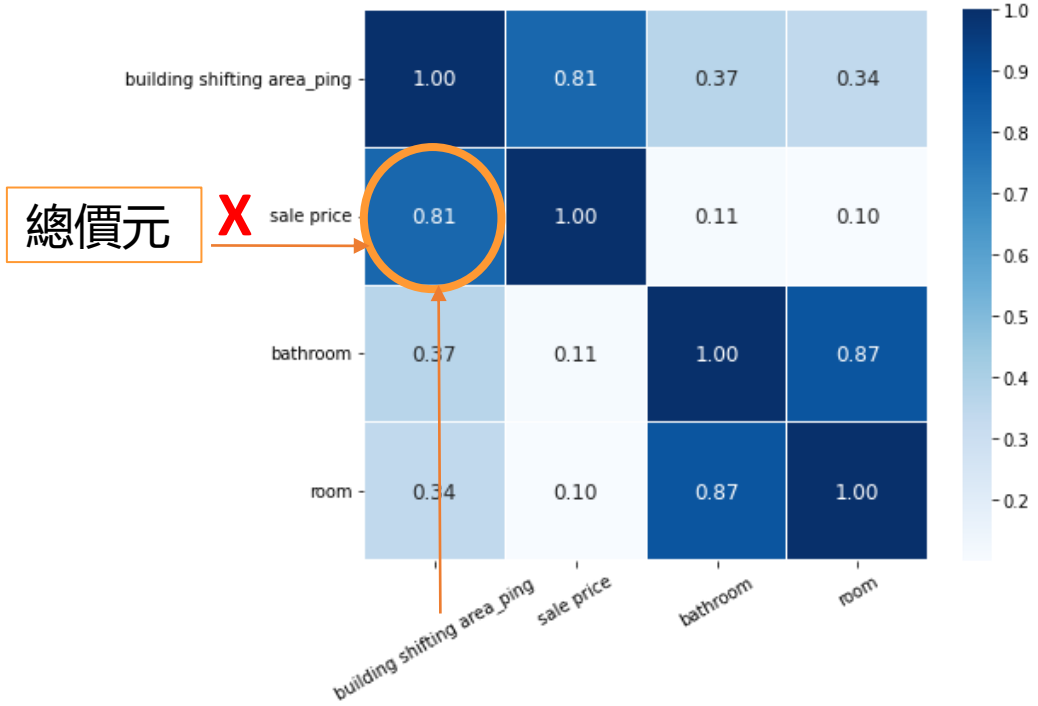
Out[13]:

	Unnamed: 0	quater	main building area	transaction date	transaction obj	building state	hall	room	bathroom	partition	...	MRT_1km	MRT_500m	MRT_300m	park_500m	park
0	0	1	49.09	2012-10-12	1	0	2	2	1	1	...	0	0	0	1	
1	1	1	113.67	2012-10-09	0	4	2	4	2	1	...	0	0	0	1	
2	2	1	34.02	2012-10-20	1	4	1	2	2	1	...	1	0	0	1	
3	3	1	27.68	2012-10-01	0	3	1	1	1	1	...	1	1	1	1	
4	4	1	68.17	2012-10-08	1	4	2	3	2	1	...	1	1	0	1	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
508426	503070	1	98.10	2021-01-08	0	1	0	0	0	0	...	0	0	0	0	
508427	520354	2	77.48	2021-04-20	0	1	1	2	0	1	...	0	0	0	1	
508428	500901	1	72.37	2021-01-15	0	4	2	3	2	1	...	0	0	0	1	
508429	517041	2	97.12	2021-04-04	0	1	2	3	2	1	...	0	0	0	1	
508430	519428	2	81.79	2021-04-17	0	1	1	3	2	1	...	0	0	0	1	

相關矩陣

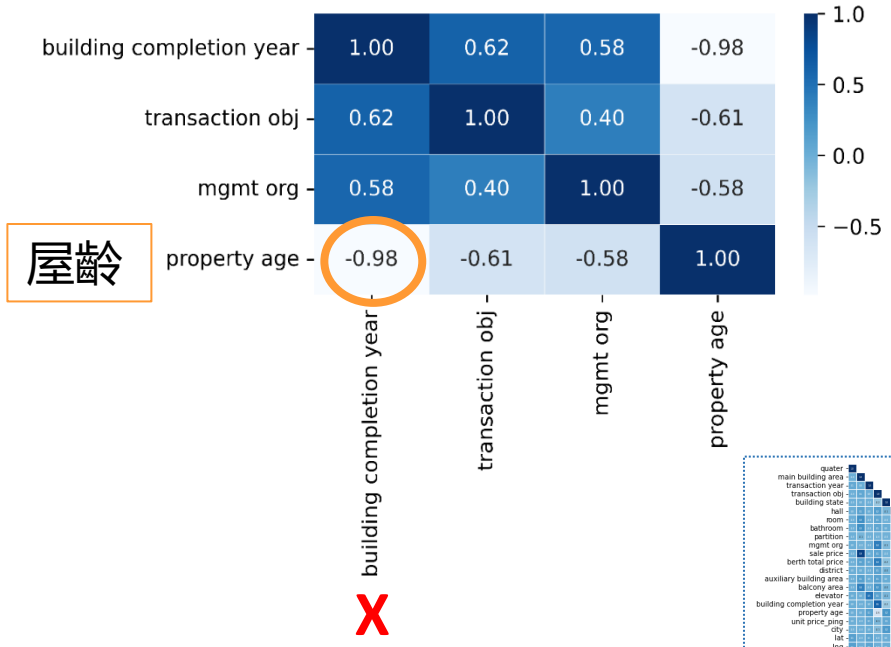
趙怡瑄

正相關

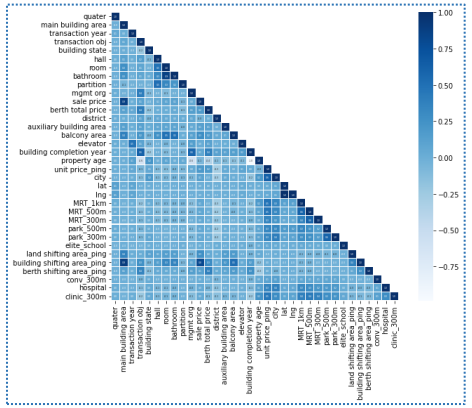


- 目標變數: unit price_ping = 總價元 / 建物移轉總面積坪

負相關



- 屋齡 = 交易年 – 建築完成西元年



Corr. Matrix for all Attr.

目標變量與特徵間之相關性

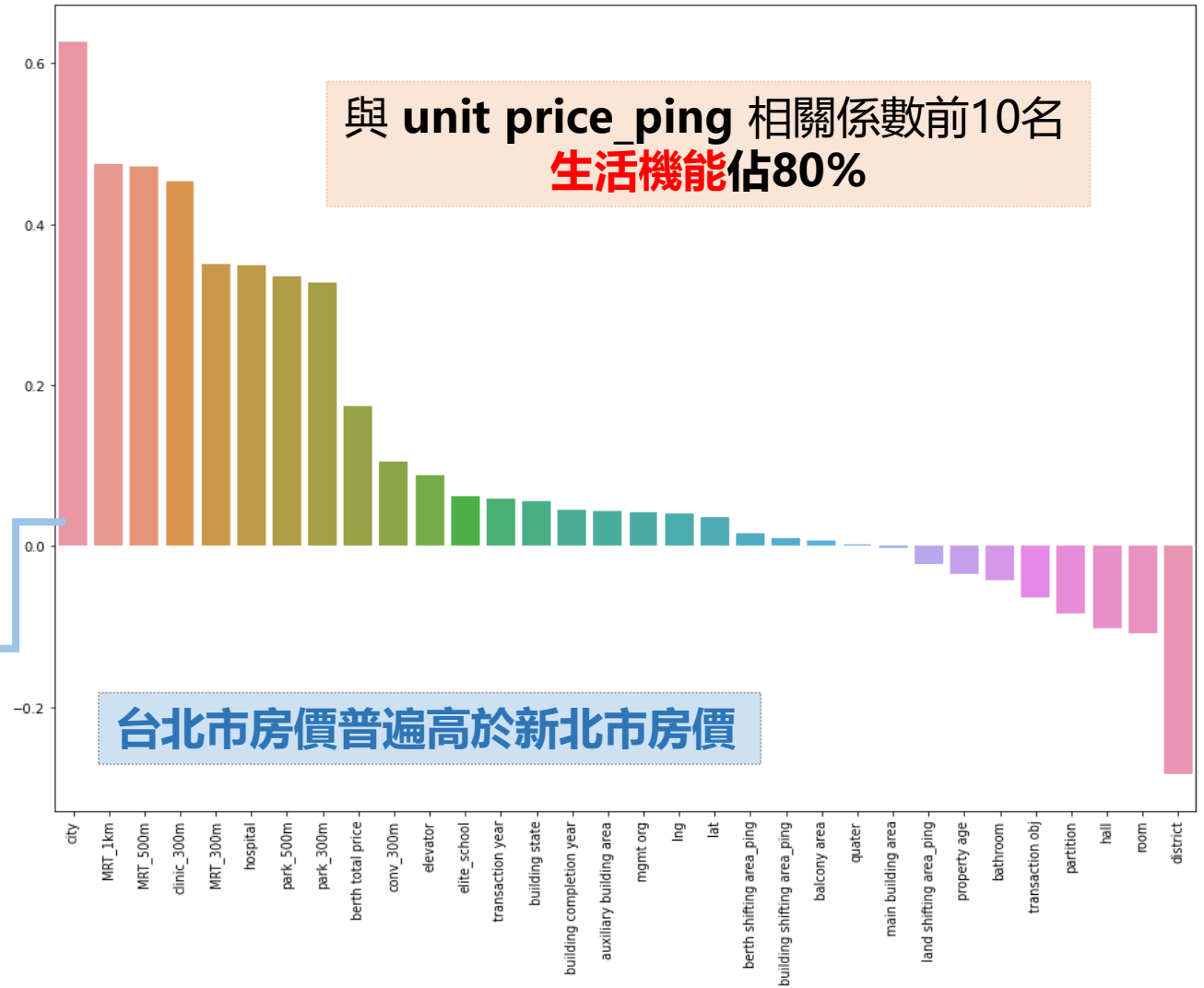
趙怡瑄

- **目標變量: unit price_ping**

名次	特徵	名次	特徵
1	City	6	Hospital
2	MRT_1km	7	Park_500m
3	MRT_500m	8	Park_300m
4	Clinic_300m	9	Berth total price
5	MRT_300m	10	Conv_300m

- 正相關 **city → 0.63**
  - city 與 房價 呈正向關係

	台北市	新北市
類別編碼	1	0
資料量比	28%	72%



Chapter 4 模型建置

講者：翁婉容

1 | 模型比較與選用

2 | 模型架構圖

3 | 關鍵因素

4 | 時間序列

模型選用

翁婉容

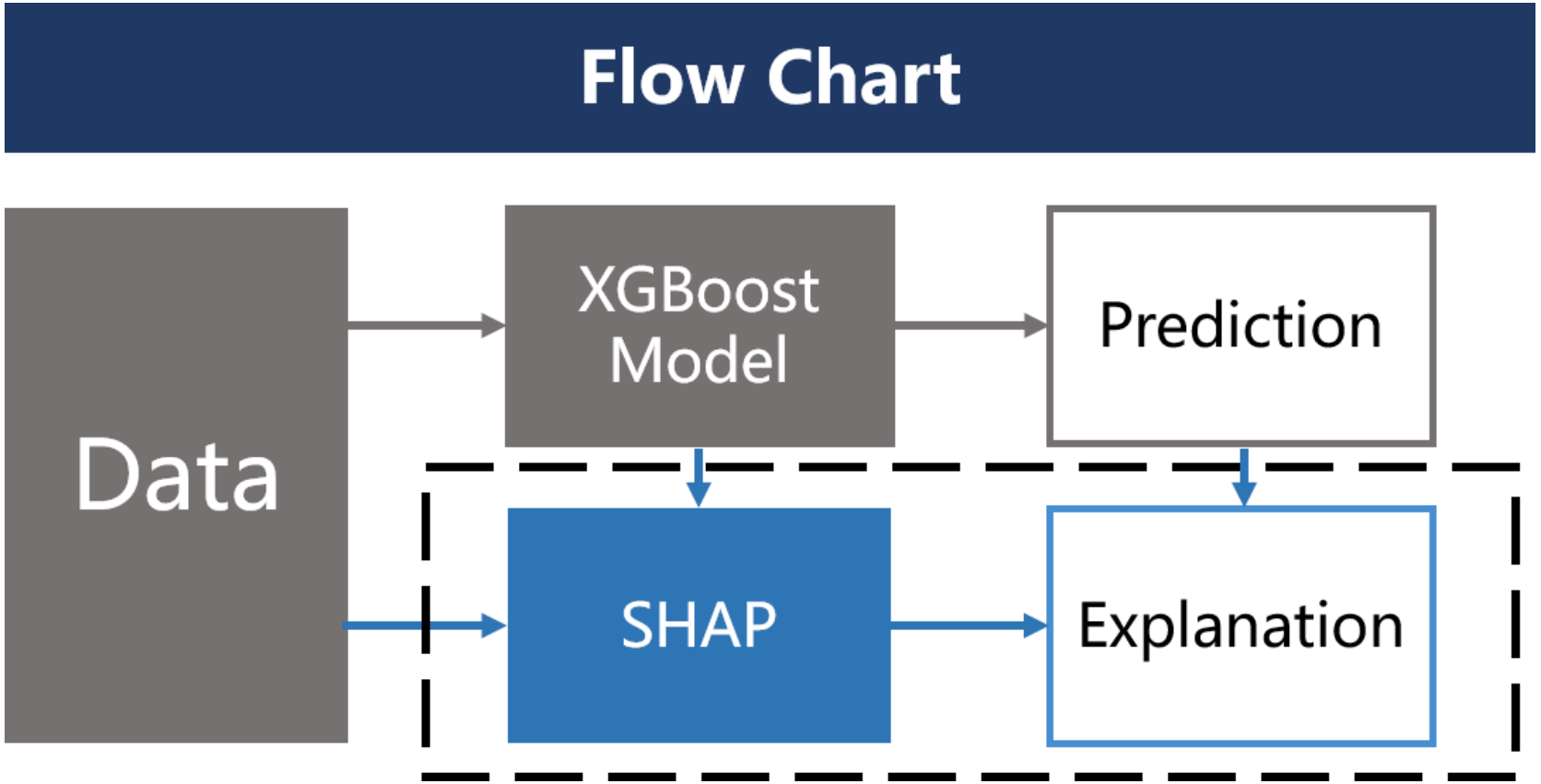
Our Model

dmlc XGBoost + SHAP

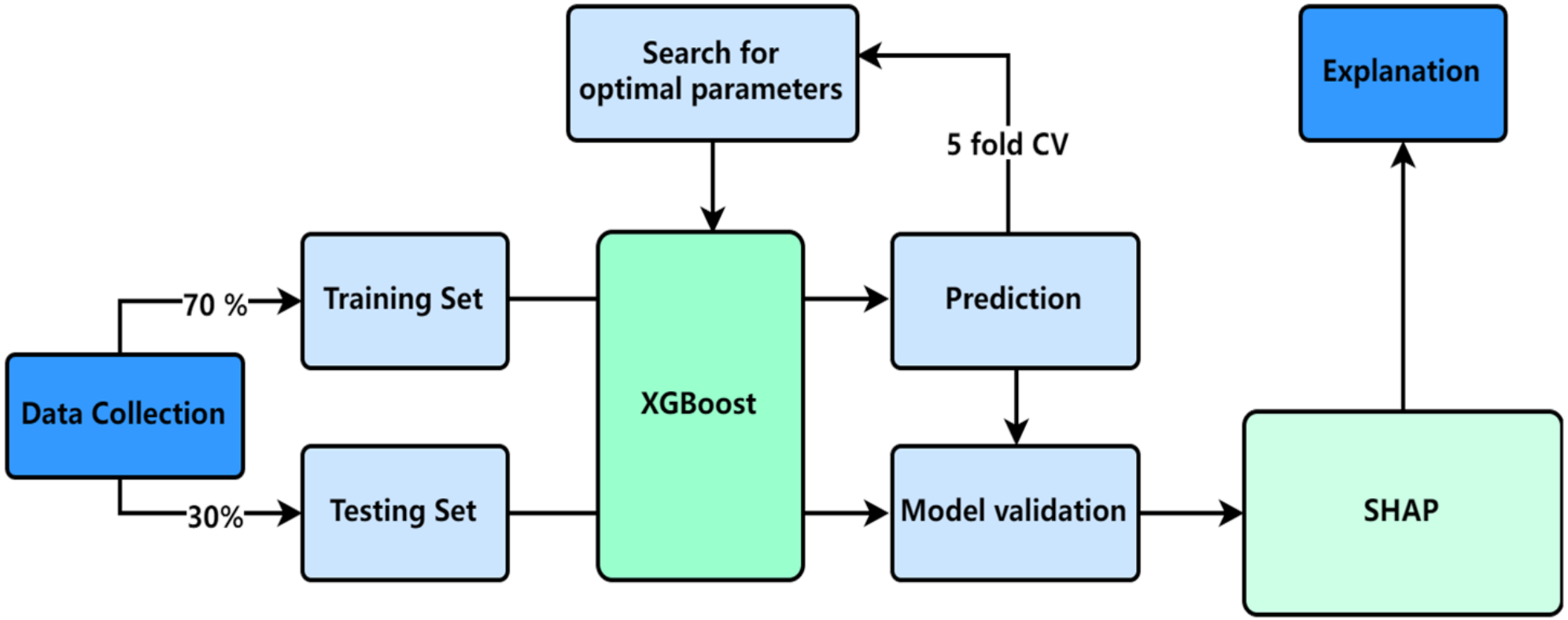
XGBoost

優點	缺點	改善方法
1. 用於分類/迴歸問題 2. 正規化 3. 支援並行處理	1. 失去了線性模型的可解釋性 2. 無法提供表現特徵的正負影響力	利用SHAP解釋Xgboost模型

模型架構圖

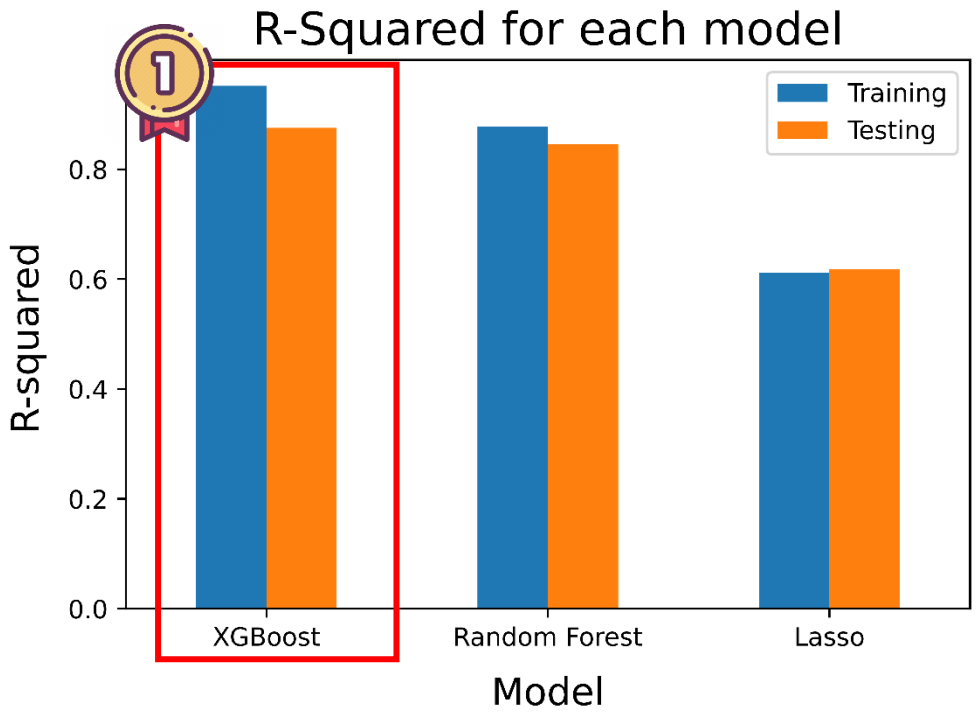

Flow Chart
Data
XGBoost Model
Prediction
SHAP
Explanation

模型架構圖


Search for optimal parameters
5 fold CV
Explanation
70 %
Training Set
Data Collection
30%
Testing Set
XGBoost
Prediction
Model validation
SHAP

模型比較

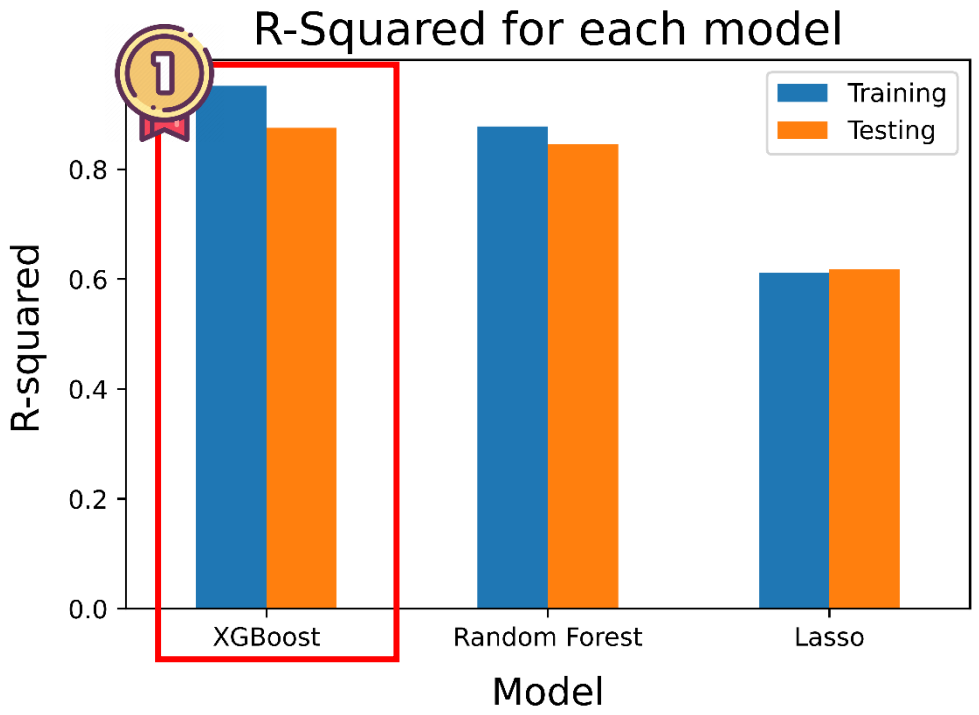
翁婉容



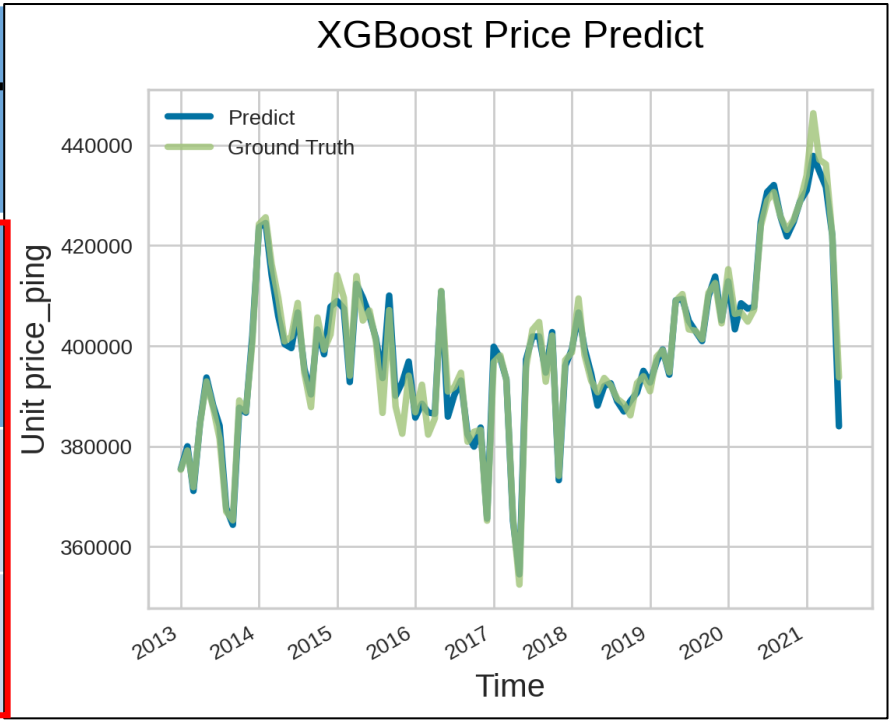
Model	集成方法模型 XGBoost	集成方法模型 Random Forest	單一方法模型 Lasso
R-squared	0.91	0.85	0.62
RMSE	69,789	77,705	122,085

模型比較

翁婉容

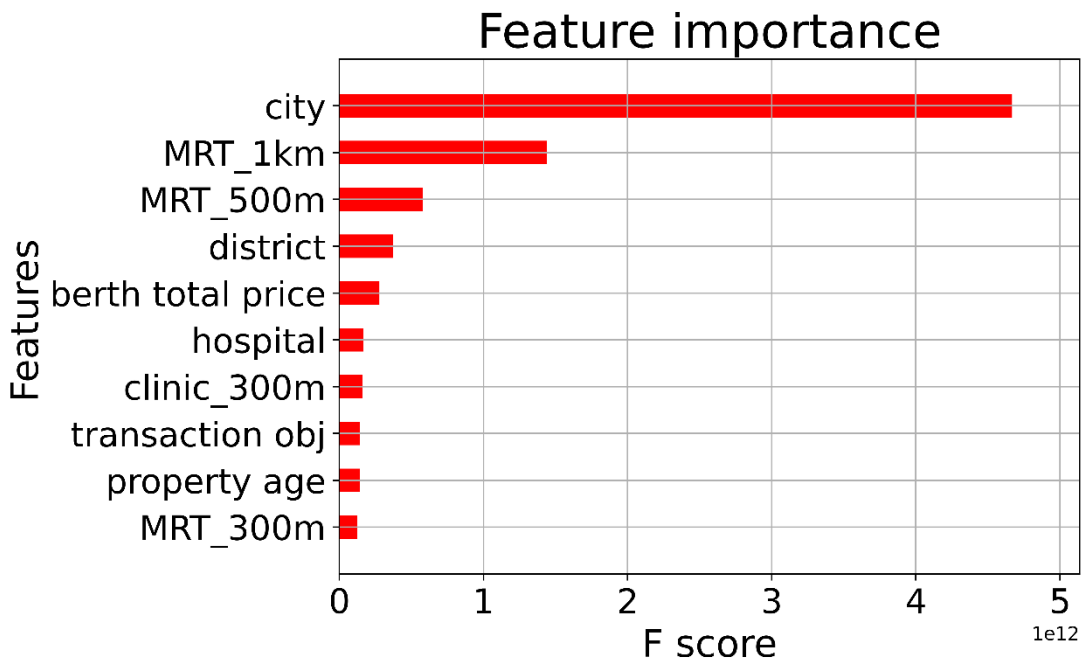


Model	集成方 XGBoost
R-squared	0.91
RMSE	69,789



模型選用- XGBoost

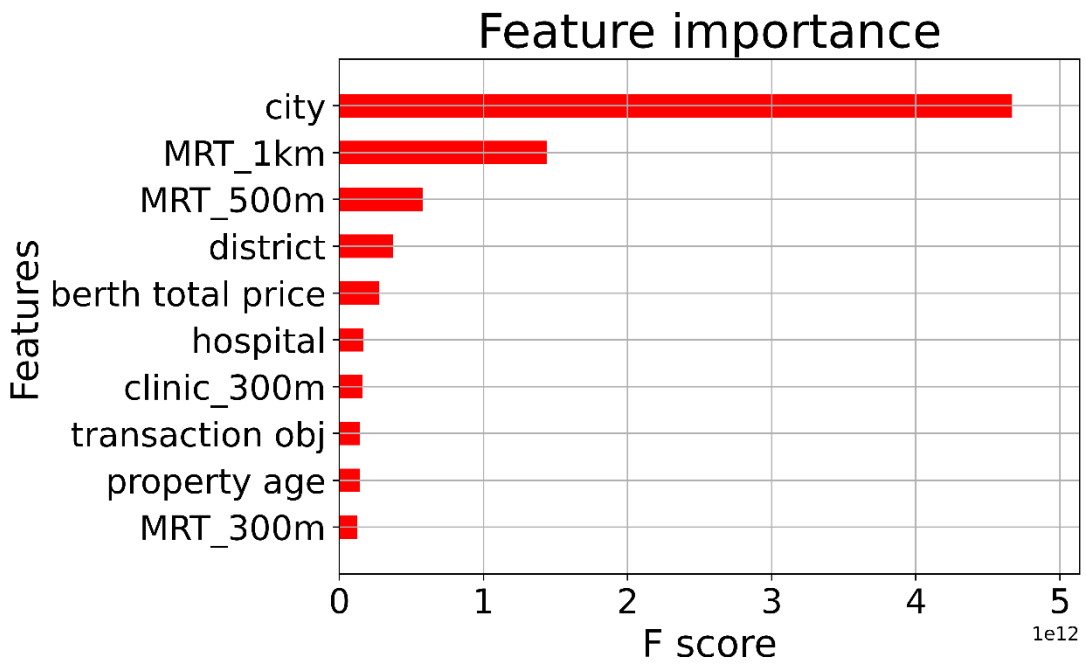
翁婉容



名次	特徵
1	城市
2	方圓1km內有無捷運站
3	方圓500m內有無捷運站
4	行政區
5	車位總價元
6	方圓1km內有無醫院
7	方圓300m內診所數量
8	交易標的
9	屋齡
10	方圓300m內有無捷運站

模型選用- XGBoost

翁婉容

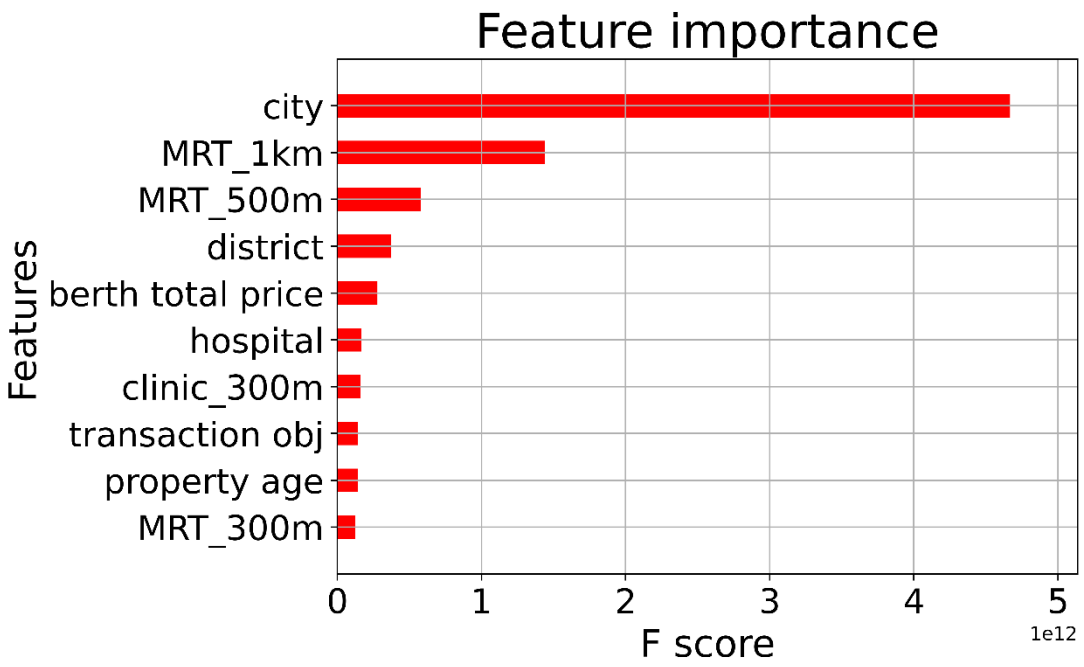


名次	特徵
1	城市
2	方圓1km內有無捷運站
3	方圓500m內有無捷運站
4	行政區
5	車位總價元
6	方圓1km內有無醫院
7	方圓300m內診所數量
8	交易標的
9	屋齡
10	方圓300m內有無捷運站

地理位置最為重要

模型選用- XGBoost

翁婉容



名次	特徵
1	城市
2	方圓1km內有無捷運站
3	方圓500m內有無捷運站
4	行政區
5	車位總價元
6	方圓1km內有無醫院
7	方圓300m內診所數量
8	交易標的
9	屋齡
10	方圓300m內有無捷運站

前十名特徵重要性中，
生活機能佔50%

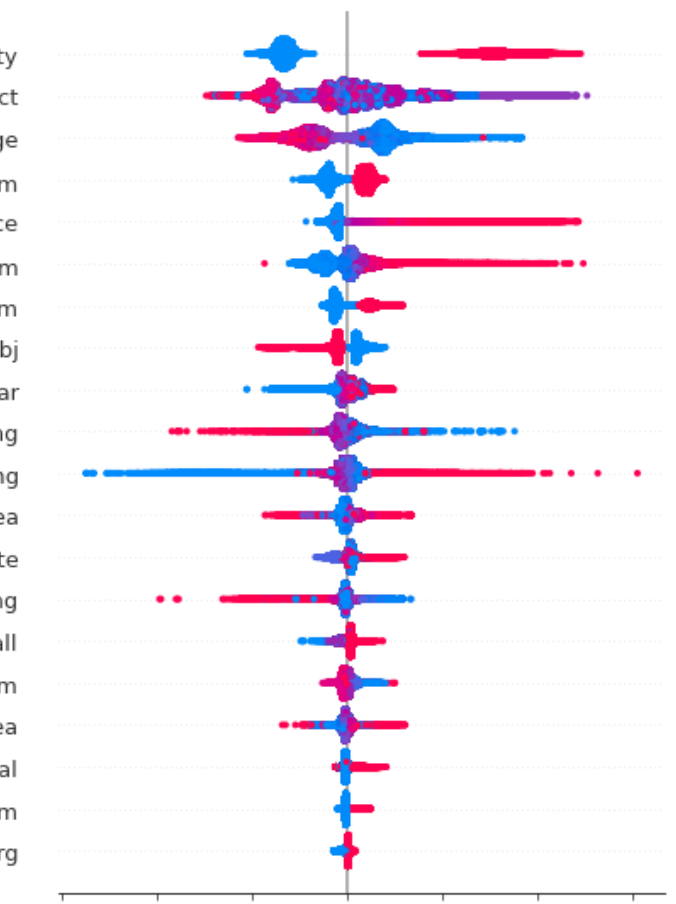
模型解釋- SHAP

翁婉容

特徵重要度

高

低



特徵本身值

大

小

低 ← 高 SHAP Value

Y 軸

- 表示特徵
- 愈上方特徵重要度愈高，反之亦然。
- 特徵排名：
  1. 城市
  2. 行政區
  3. 屋齡
  4. 方圓1km內有無捷運站
  5. 車位總價元

模型解釋- SHAP

翁婉容

特徵重要度

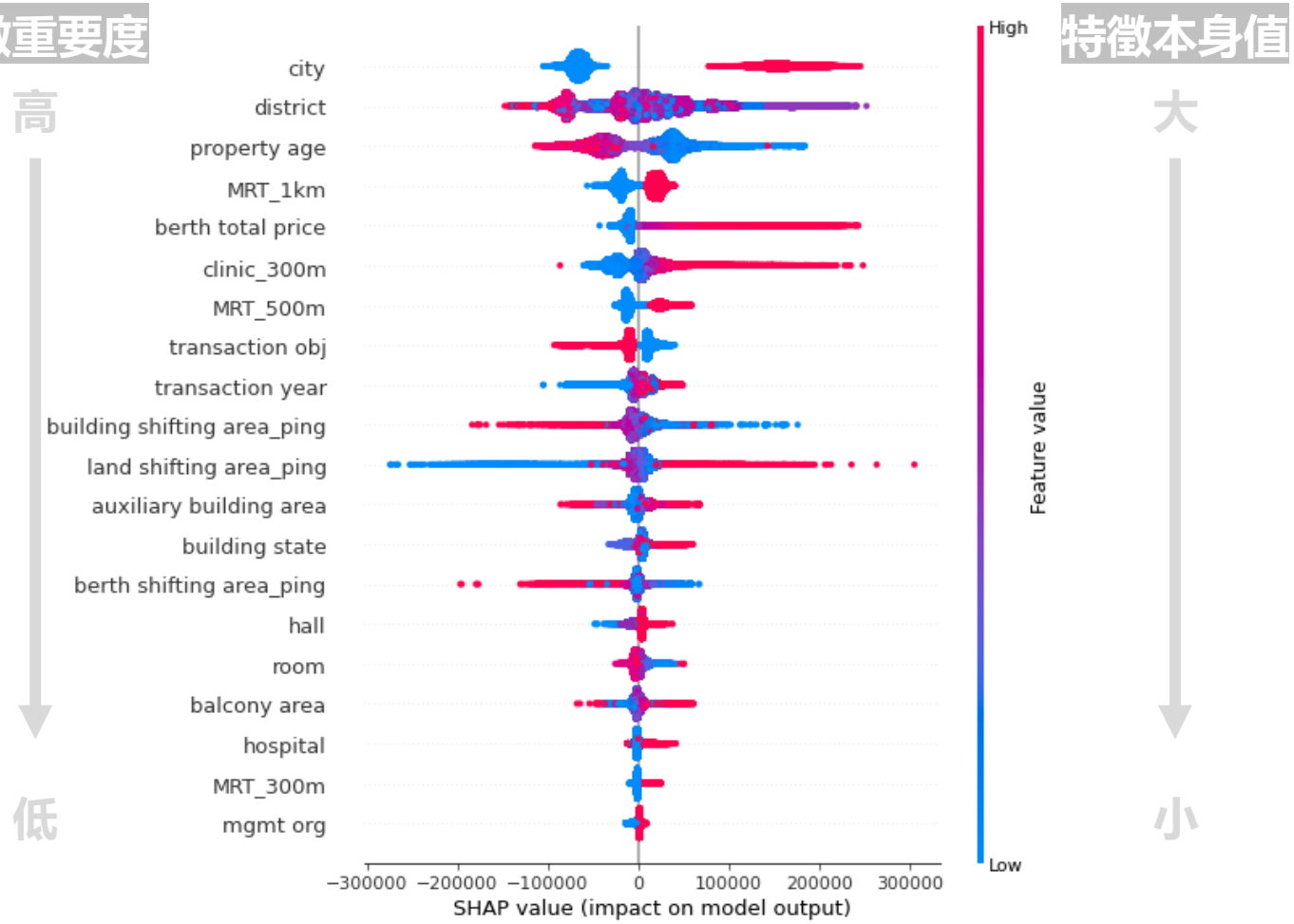
高

低

特徵本身值

大

小



低 ← 高 **SHAP Value**

X 軸

- 表示SHAP value
- X軸愈右方

  → SHAP value愈高

  → 對於房價影響愈大

模型解釋- SHAP

翁婉容

藍色在左, 紅色在右

特徵重要度

高

低

特徵本身值

大

小

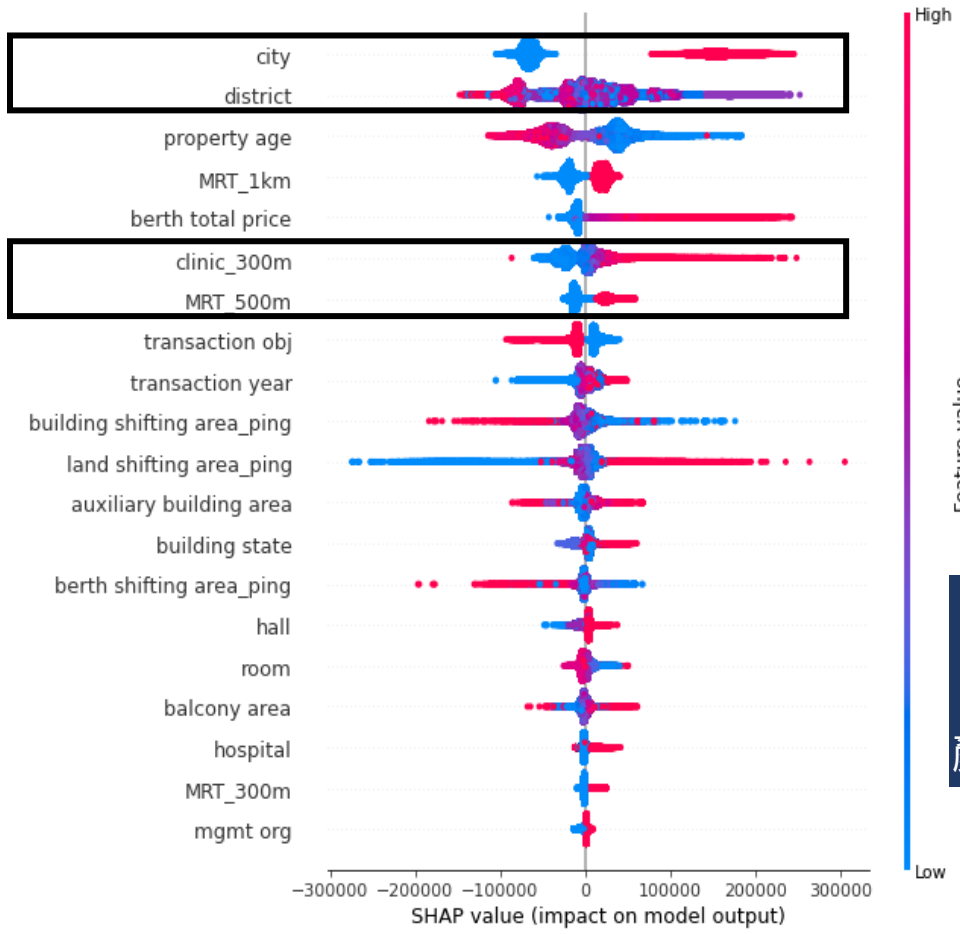
低 ← → 高 SHAP Value

顏色

- 表示特徵本身值的大小
- 紅色在右，藍色在左
  → 此特徵與房價正相關
- 紅色在左，藍色在右
  → 此特徵與房價負相關

模型解釋- SHAP

翁婉容



城市與區域為影響房價最主要因素

→ 地理位置重要性

距離捷運愈近，房價愈高
診所數量愈多，房價愈高

→ 生活機能重要性

X軸:	SHAP value
Y軸:	特徵
顏色:	特徵本身值的大小

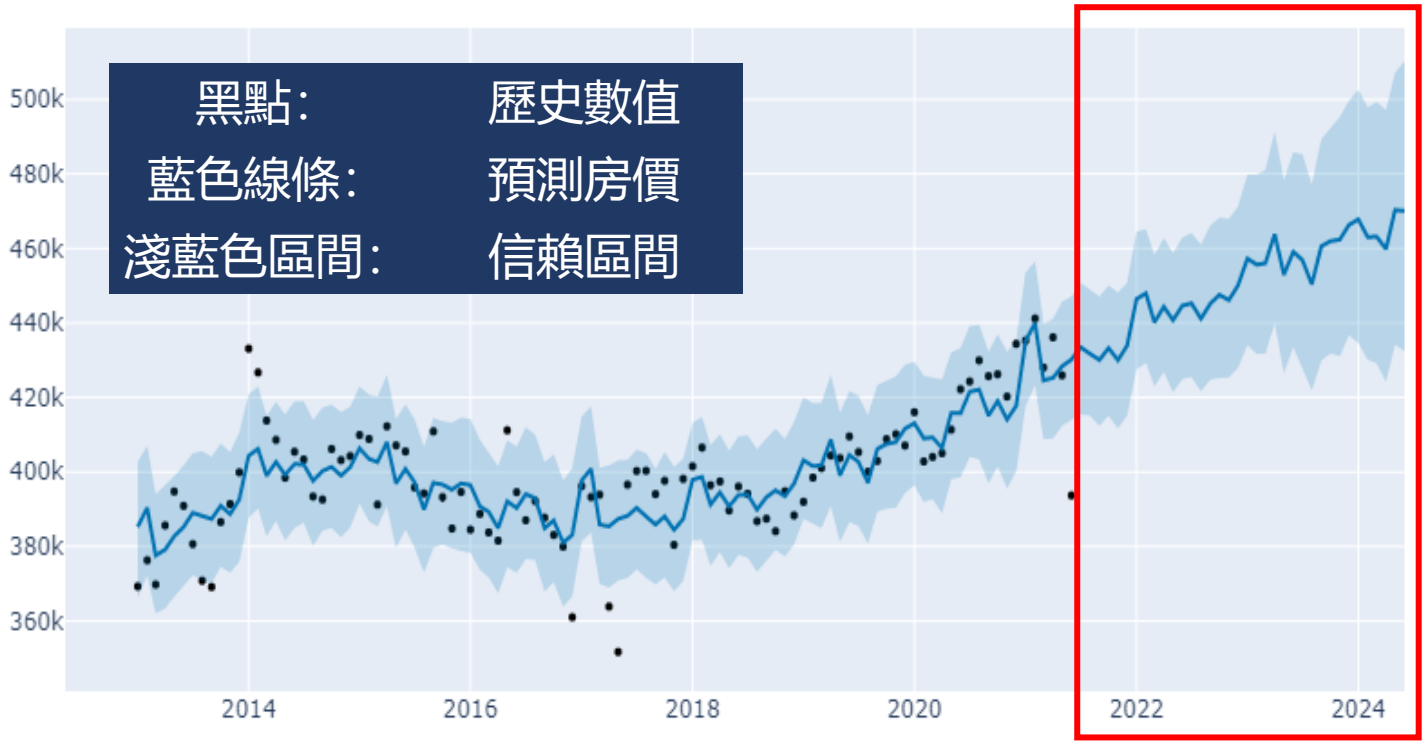
SHAP

- SHAP value反映出每一個樣本中的特徵影響力，並表現特徵影響的正負性
- 買房子要考量的重點為地段佳、生活機能佳、交通便利

時間序列– 雙北整體房價預測

翁婉容

未來預測趨勢

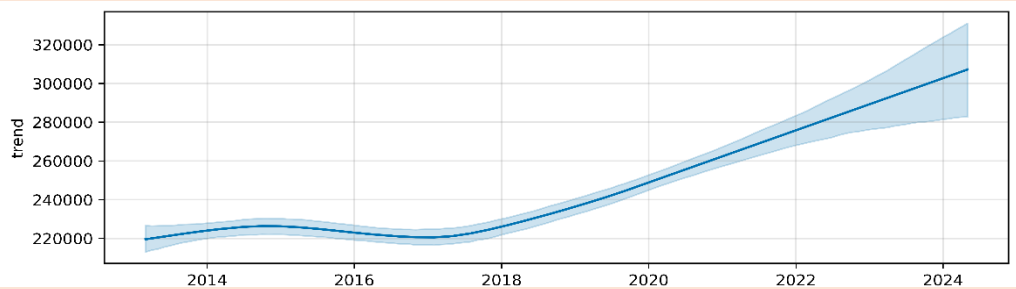
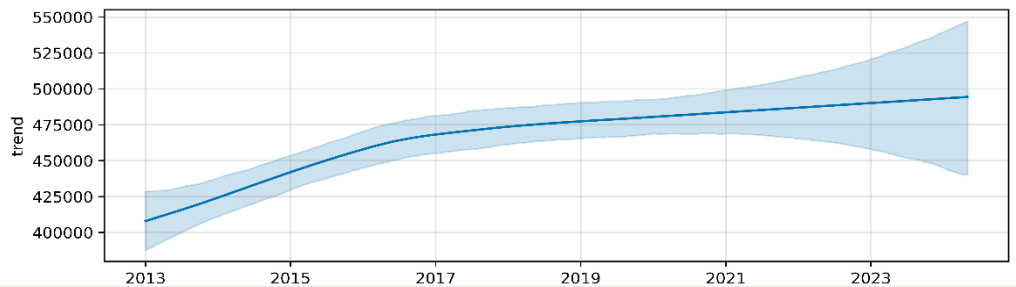
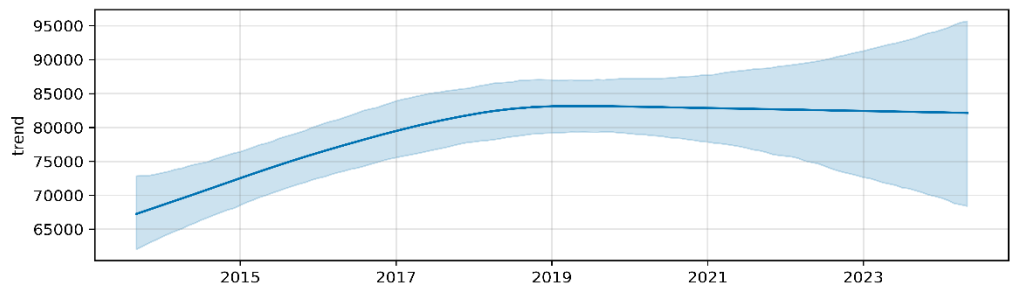


模型預測

- 預測2022年至2024年雙北房價會飆漲，並且不停地突破房價天花板

時間序列- 行政區預測

翁婉容

類型	預測趨勢圖	代表行政區
迅速飆漲型		內湖區、新莊區、樹林區、深坑區、金山區、三芝區
緩慢上漲型		汐止區、大同區、林口區、雙溪區、泰山區、萬華區
持平型		石碇區、萬里區、貢寮區

Chapter 5 視覺化

講者：李惠萍

1 | 製作架構

2 | DEMO

製作架構

李惠萍

網頁架構

前端語言

排版套件

輕量級Web應用框架

互動圖表

視覺化互動式圖表

時間序列預測工具

地圖繪製

部署網站

DEMO

Dream House首頁連結：
https://bdse21-group2-finalproject.herokuapp.com/

Chapter 6 總結

講者：翁婉容

1 | 總結

2 | 未來展望

總結

翁婉容

專題貢獻

雙北實價登錄房產資訊，
融合各大生活機能

分析趨勢、預測房價
了解房市資訊、挖掘升值潛力

提供影響房價之
最關鍵因素

建立網頁讓欲買房者
迅速解析房市資料

未來展望

翁婉容

服務對象
擴展至全台

增加特徵屬性

Ex: 人口分布、所得、教育程度

結合借貸資訊

即時互動式網站

謝謝大家

CALL MMMMEEEE