信義房屋-B組

利用公開房地產數據 進行物件分群及房價預測

組員：詹雅婷、劉祖維、陳永靖、張威廉、黃敏瑄、葉柏妤
Mentor：數據科學家 劉秝瑋、數據工程師 柯雅云、 經理 張玄江

目錄

痛點分析

房仲

媒合效率低且耗時

買賣雙方因價格談不攏導致媒合成功率低且耗時，最終造成**客戶流失率高**

不同房仲價格建議沒有一致

房仲因**經驗不同**，**提供給買賣方的建議價格有落差**，致買賣方對房仲抱持著懷疑的態度

買方&賣方

難以掌握房產價格

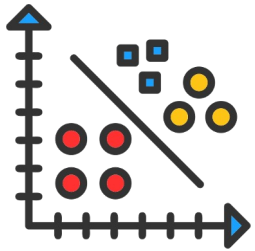
房產交易低頻，且市場變化快速，買賣方難以獲得完整資訊

蒐集可比較物件過程耗時

雖內政部提供實價登錄資訊，但**缺乏整合系統或介面**來獲取所有可比較物件

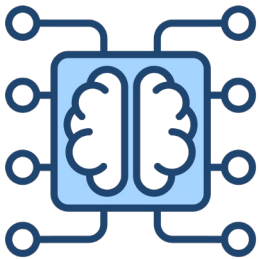
專案介紹

專案內容



相似物件分群

利用非監督式學習對樣本分群，將分群結果加入資料集，成為新的特徵



建立模型預測單價

使用機器學習，透過調整模型參數，建立準確率高的價格預測模型

樣本

來源

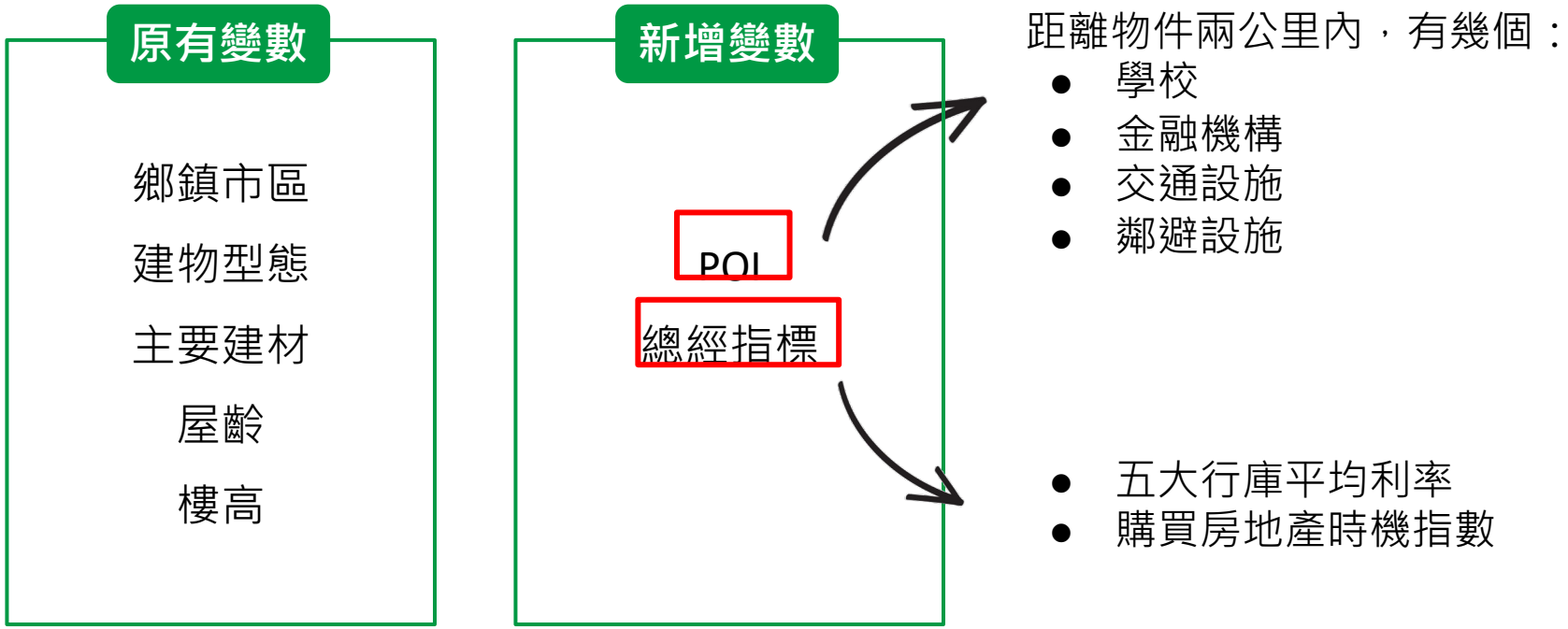
台北市實價登錄資料集

時間

2021/01 ~ 2024/03

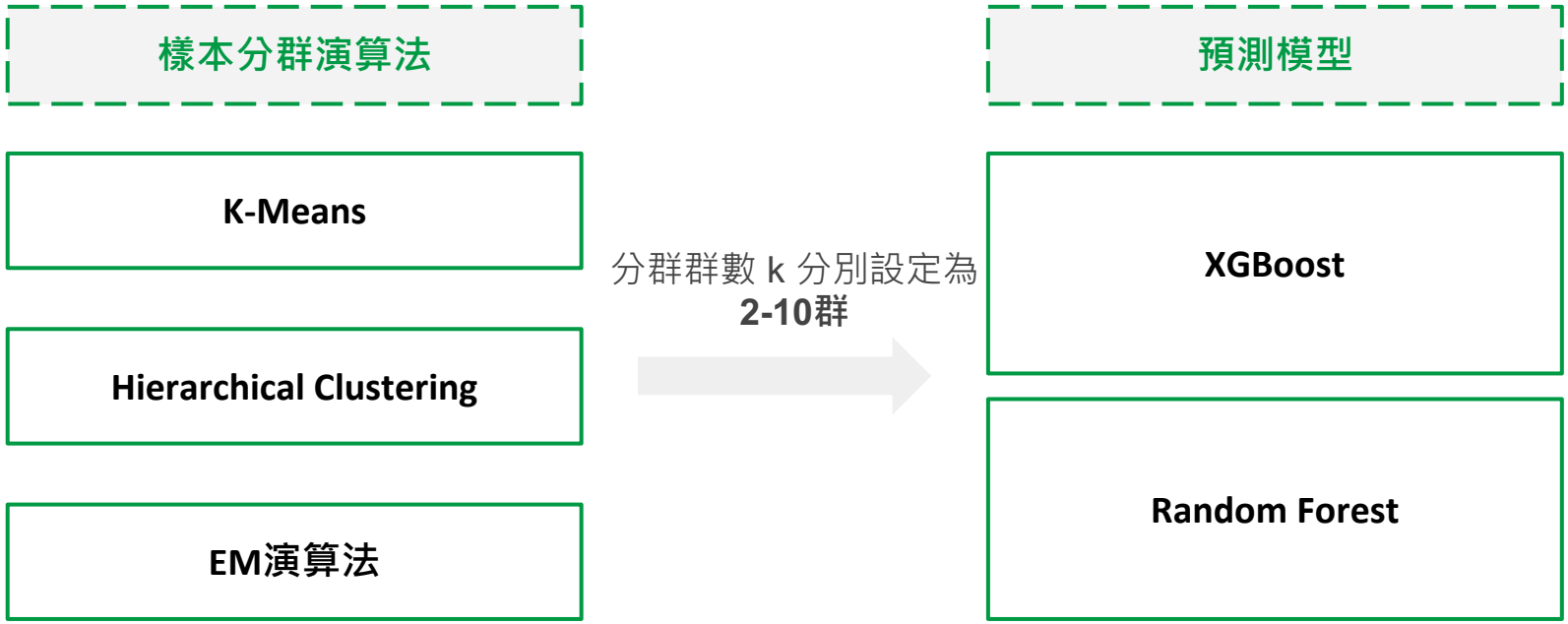
- 排除以下條件的樣本：
  - ✓ 計算邏輯錯誤
  - ✓ 離群值
  - ✓ 特徵為空值

變數介紹


原有變數
鄉鎮市區
建物型態
主要建材
屋齡
樓高
新增變數
POI
總經指標
距離物件兩公里內，有幾個：
學校
金融機構
交通設施
鄰避設施
五大行庫平均利率
購買房地產時機指數

模型設計及結果

模型設計


樣本分群演算法
K-Means
Hierarchical Clustering
EM演算法
分群群數 k 分別設定為
2-10群
預測模型
XGBoost
Random Forest

模型選擇方式

RMSE 越小越好
Hit Rate 越大越好

透過 **RMSE** 和 **Hit Rate** 兩指標來決定最適分群預測模型組合

		K-Means	Hierarchical Clustering	EM演算法
RF	RMSE	13.79萬元	13.76萬元	13.72萬元
	Hit Rate	57.94%	58.05%	58.60%
XG-Boost	RMSE	10.40 萬元	10.38 萬元	10.38 萬元
	Hit Rate	74.43%	74.48%	74.38%

Hit Rate 定義：模型預測值與實際值之間相對誤差小於 **15%** 的比例

選擇最終模型

最後決定使用預測結果最好 **XGBoost 和階層式分群（7 群）**為最終的預測與分群模型。
但可以看出分群並沒有顯著提升預測效果。

	MSE	MAE	RMSE	R2	Hit Rate
分群前	107.89	7.44	10.39 萬元	0.80	74.23%
分群後	107.71	7.42	10.38 萬元	0.80	74.48%

預測模型實作範例

Input 物件


```
{'鄉鎮市區': 0.0, '土地移轉總面積_坪': 5.59625, '建物型態': 0.0, '主要建材': 0.0,
'屋齡': 36.66666667, '樓高': 12.0, '土地筆數': 1.0, '建物筆數': 1.0, '房數':
0.0, '廳數': 0.0, '衛數': 0.0, '管理組織': 1.0, ... 'POI_民間機構': 4073.0,
'POI_鄰避設施': 1102.0, '五大行庫平均購屋貸款利率': 1.851, '購買房地產時機指數':
102.65, '建物移轉總面積(不含車位)_坪': 53.9176, '7群': 4.0}
```


Output

實際價格	63.06 萬元/坪
預測價格	66.01 萬元/坪
Hit Rate	(within 15% error): Yes

預測模型實作範例

Input 物件


```
{'鄉鎮市區': 0.0, '土地移轉總面積_坪': 5.59625, '建物型態': 0.0, '主要建材': 0.0,
'屋齡': 36.66666667, '樓高': 12.0, '土地筆數': 1.0, '建物筆數': 1.0, '房數':
0.0, '廳數': 0.0, '衛數': 0.0, '管理組織': 1.0, ... 'POI_民間機構': 4073.0,
'POI_鄰避設施': 1102.0, '五大行庫平均購屋貸款利率': 1.851, '購買房地產時機指數':
102.65, '建物移轉總面積(不含車位)_坪': 53.9176, '7群': 4.0}
```


Output

相似物件

* 程式篩選順序：分群結果相同 -> 鄉鎮市相同 -> 土地、建物筆數相同
-> 房廳數相近 -> 屋齡相近 -> 購買房地產時機指數相近

* 結果:

經度	緯度	鄉鎮市	屋齡	...	單價/坪
121.5259	25.0670	0	30.8		$61.56萬元
121.5365	25.0622	0	36.6		$74.02萬元
121.5245	25.0548	0	38.4		$79.99萬元
121.5245	25.0548	0	38.4		$79.98萬元

功能介紹

功能介紹｜針對不同使用者提供合適平台

使用者	平台	功能	時程規劃
仲介	擴充信義房屋後台資訊平台	物件預估售價與其他詳細資訊	先上線測試及穩定功能
賣方 買方	Line 聊天機器人	價格預測功能 市場情報查詢功能	待系統精準度穩定後上線至 Line，提供給買賣方使用

功能介紹｜仲介 擴充信義房屋後台資訊平台功能

功能介紹｜仲介 擴充信義房屋後台資訊平台功能

信義房屋資訊平台

近期成交案件

登入新案源

...

預測房價

信義房屋資訊平台

本物件單價預估為每坪___萬元

★ 已成交相似物件
- ● 同區域單價相似
- ● POI資訊相似

★ 已委託本公司物件(銷售中)
- ● 同社區
- ● 其他社區

★ 其他市場相似物件(銷售中)
- ● 同社區
- ● 其他社區

可再點進去物件了解詳細資訊，包括委託銷售期間、POI、總價預估等資訊

功能介紹｜賣方和買方 擴增信義房屋現有的line聊天機器人

立即加信義房屋好友!

功能介紹｜賣方 價格預測

Step 1：點選價格預測

Step 2：依序輸入持有的房屋條件

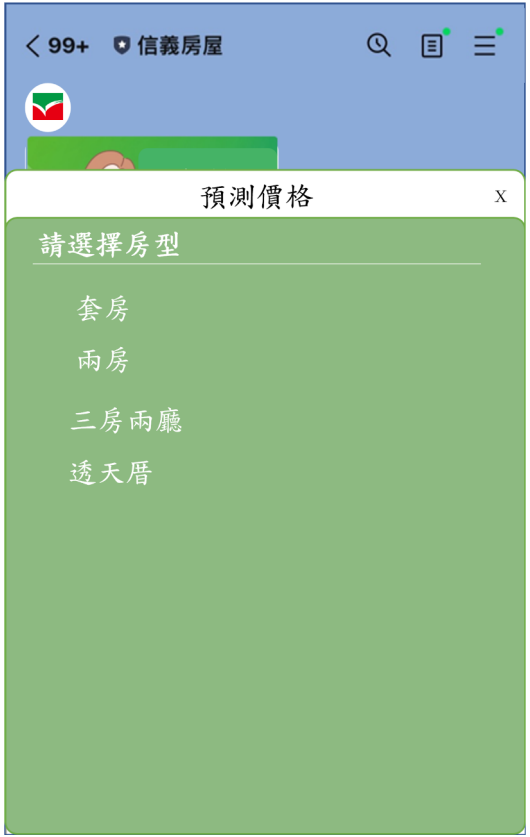
Step 3：呈現預測單價與相似物件

功能介紹｜賣方 價格預測

Step 2：依序輸入持有的房屋條件

1. 房屋所在地
2. 房型
3. 屋齡
4. 樓高
5. 預計賣出時間

…



功能介紹｜買方 市場情報

Step 1：點選市場情報

Step 2：輸入理想的房屋條件

Step 3：呈現市場行情價 格與相似物件

功能介紹｜買方 市場情報

Step 2：輸入理想的房屋條件

預計可輸入的條件

1. 房屋所在地
2. 房型
3. 屋齡
4. 樓高
5. 房間數量
6. 衛浴間數
7. 總廳數

…

功能介紹｜買方 市場情報

Step 3：
呈現目前市場行情價格與相似物件

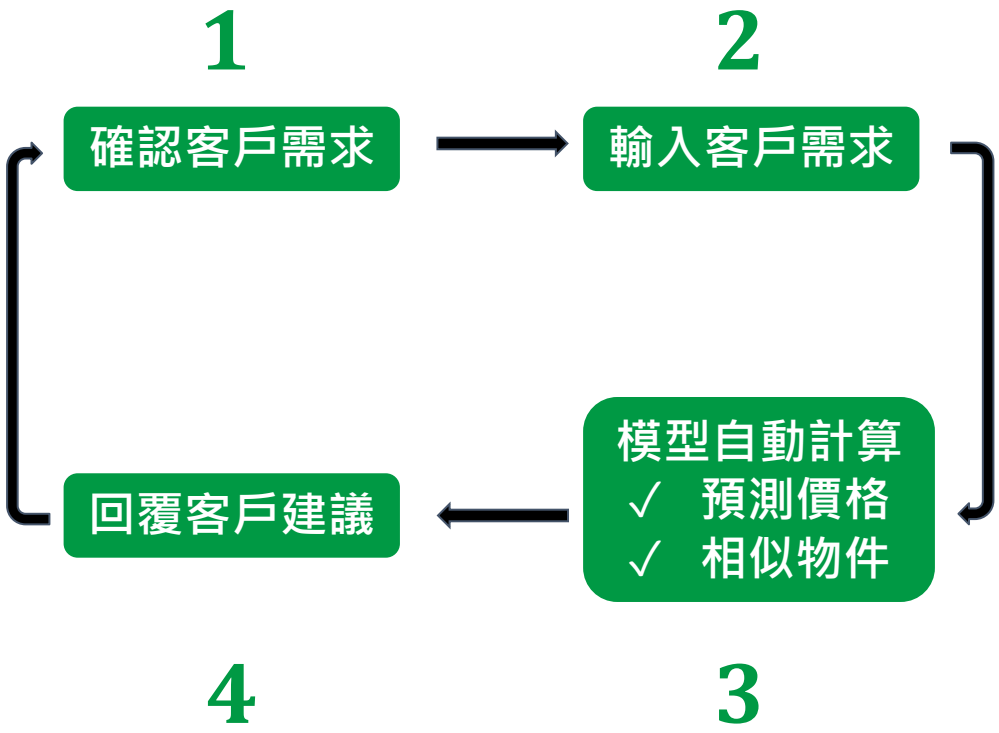
點擊查看相似社區

網站呈現該房屋與其社區

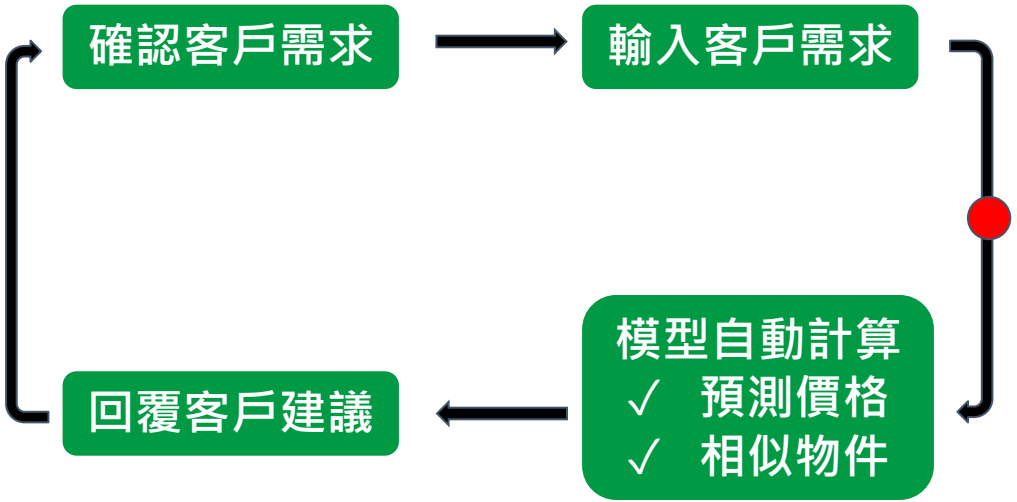
消費者除解此價格能購買的房屋類型外，也能查看相似社區是否有理想的房屋正在出售

應用場景

使用情境-房仲

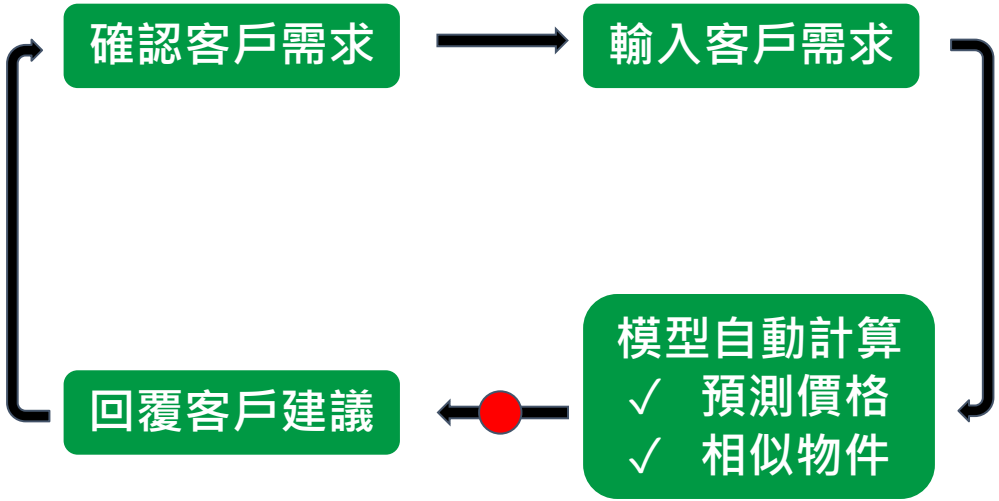


使用情境-房仲



使用模型提供準確且合理的預測，**解決不同房仲價格建議不一樣的問題**

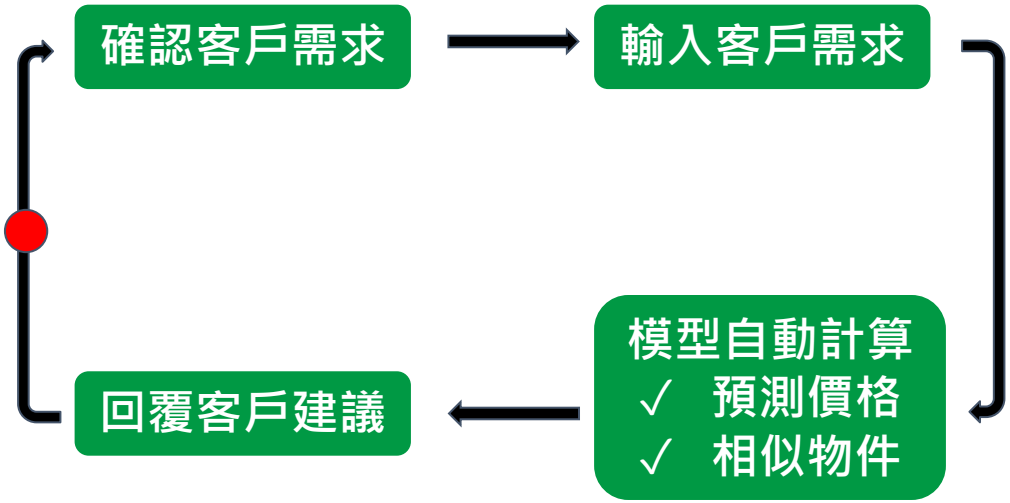
使用情境-房仲



模型迅速提供房仲合理建議價格和相似物件資訊，**縮短交易談判時間**

使用情境-房仲

買賣雙方長期累積良好的回饋，提升買賣雙方對信義房屋的信任和商譽，**提高成交率**



使用情境-買方和賣方

1 確認需求條件

2 輸入需求條件

3 模型自動計算
✓ 預測價格
✓ 相似物件

4 回傳客戶結果

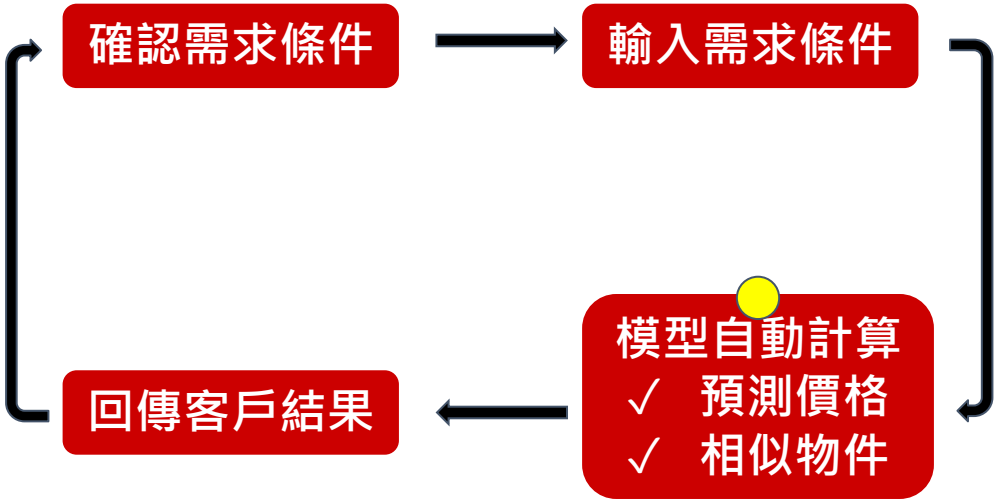
使用情境-買方和賣方

確認需求條件 → 輸入需求條件 → 模型自動計算
- ✓ 預測價格
- ✓ 相似物件

→ 回傳客戶結果 → 確認需求條件

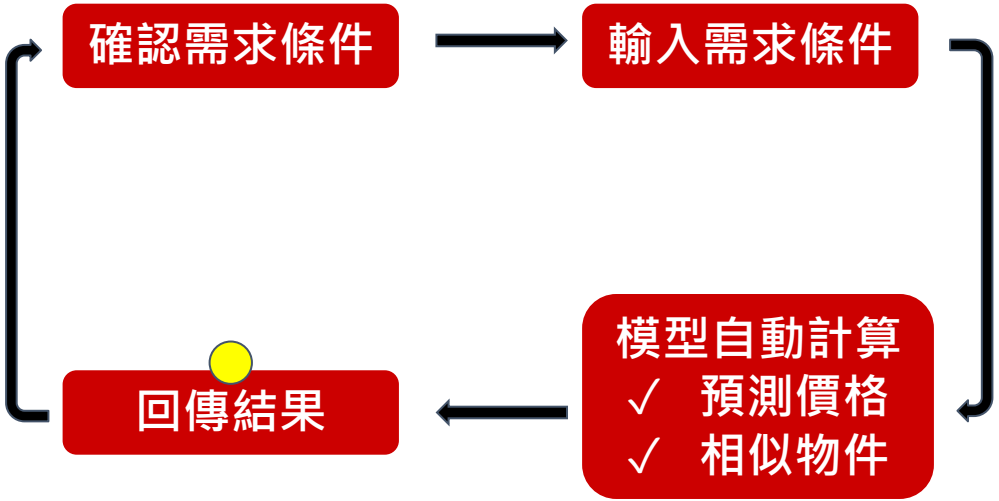
使用Line聊天機器人，可**同時預測價格並蒐集相似物件**

使用情境-買方和賣方



納入POI和總經等指標，
提供**更準確的相似物件**

使用情境-買方和賣方



協助買家和賣家更全面地
了解房地產價格和潛在的
投資機會

分工

分工

姓名	工作內容
詹雅婷	建立分群模型（Hierarchical及EM演算法）、聯絡業師、製作簡報（功能介紹）
劉祖維	資料前處理、製作簡報（痛點介紹、專案介紹、應用場景）
陳永靖	建立預測模型random forest、分析分群數量最佳解、製作簡報（模型設計）
張威廉	製作簡報（痛點介紹）、會議記錄、錄音會議
黃敏瑄	建立預測模型 Xgboost、分析分群數量最佳解、製作簡報（模型設計）
葉柏妤	建立分群模型（K-Means及DBSCAN）、製作簡報（功能介紹）

Q&A

附錄

資料前處理

- **排除樣本數過少的特殊樣本**：地上權、裝潢或傢俱、瑕疵
- **排除計算邏輯錯誤的樣本**：
  - ✓ 車位移轉總面積_坪=0，但車位總價_萬元不為0
  - ✓ 車位移轉總面積_坪=0，車位總價_萬元=0，但交易標的為房地(土地+建物)+車位
  - ✓ 土地筆數=0，車位總價_萬元不為0，但交易標的為建物
- **排除主要建材欄位為空值的樣本**

資料前處理

- **排除樓高為空值的樣本**
- **排除屋齡為空值的樣本**
- **排除樓別為空值的樣本**
- **排除掉交易標的為「建物」的樣本**
- **排除所有和車位相關的特徵(車位坪數、車位價格、車位筆數)**：因為模型預測的y_label是單價，單價和車位無關
- **排除離群值**：利用IQR的方法排除離群值樣本

特徵工程

- **最高樓層、最低樓層、移轉樓別總數：**針對「樓別」這個欄位，用最低樓別/最高樓別兩個欄位來填數字格式的變數。例如”一層”填1/1，”地下一層、一層、三層”則填-1 / 3；透天厝一樣用樓高從一層開始推樓高=4填1/4。移轉樓別總數則是移轉層數的數量，例如：”地下一層、一層、三層”則填3，”五層”則填1，透天厝則根據其樓高推算
- **備註-特殊**：該row的「備註」欄位出現「特殊」這兩個字

特徵工程

- **建物移轉總面積(不含車位)_坪**：原本有一個特徵是「建物移轉總面積_坪」，但我們現在要預測的y_label是單價，和車位無關，所以要扣除掉車位的坪數
- **主要用途**、**主要建材**：按照連結的邏輯歸類

特徵工程

- **分群使用的資料集：**排除掉分群用不到的特徵，再使用pd.get_dummies將類別變數轉換成one-hot encoding。對於POI為空值的欄位，採用fillna函數補0，最後再使用PCA對資料集進行降維，降維後的資料保留95%的變異
- **預測使用的資料集**：排除掉預測模型用不到的特徵後，使用Label encoder將類別變數轉換成數字表示。對於POI為空值的欄位，採用fillna函數補0

變數介紹

原有變數

鄉鎮市區

建物型態

主要建材

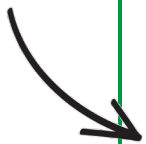
屋齡

樓高

新增變數

POI

總經指標



當指標數字尚未公布時，利用過去的指標數字，結合迴歸模型或機器學習模型作為未來指標的替代值

預測模型實作程式碼

實作測試

```python
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from modelFunction import split_data, hyperparameter_tuning_random, train_model, Score

# Set the path of the data
DATA_PATH_HIERARCHICAL = '/Users/vanessahuang/Desktop/BI_capstone/信義房屋/Final/Data/階層式分群資料集.csv'

# Load the data
data_hierarchical = pd.read_csv(DATA_PATH_HIERARCHICAL, encoding="Big5")

# Select the columns to use: group 7 with hierarchical clustering
other_columns = list(data_hierarchical.columns[:37])
selected_columns = other_columns + [data_hierarchical.columns[42]]
data_hierarchical_group = data_hierarchical[selected_columns]

# Split the data into training and testing sets
X_train, X_test, y_train, y_test = split_data(data_hierarchical_group, '單價_萬元/坪', 0.2, 42)

# Find the best hyperparameters for the model
best_params = hyperparameter_tuning_random(cv=3, n_iter=10, n_jobs=-1, X_train=X_train, y_train=y_train)
model, mse, mae, rmse, r2 = train_model(best_params['max_depth'], best_params['learning_rate'], best_params['n_estimators'], X_train, y_train, X_test, y_test)

# Randomly select a sample from the test set
selected_index = np.random.choice(X_test.index)
selected_object = X_test.loc[selected_index]
actual_price = y_test.loc[selected_index]

# Predict the price of the selected object
predicted_price = model.predict(selected_object.values.reshape(1, -1))[0]

# Calculate the hit rate
relative_error = abs((predicted_price - actual_price) / actual_price)
hit_rate = relative_error < 0.15

selected_object_dict = selected_object.to_dict()

# Print the results
print("Selected object features:")
print(selected_object_dict)
print(f"\nActual price: {actual_price:.2f} 萬元")
print(f"Predicted price: {predicted_price:.2f} 萬元")
print(f"Hit rate (within 15% error): {'Yes' if hit_rate else 'No'}")
```

[10] ✓ 1m 8.0s Python

```
Selected object features:
{'鄉鎮市區': 0.0, '土地移轉總面積_坪': 5.59625, '建物型態': 0.0, '主要建材': 0.0, '屋齡': 36.66666667, '樓高': 12.0, '土地筆數': 1.0, '建物筆數': 1.0, '房數': 0.0, '廳數': 0.0, '衛數': 0.0, '管理組織': 1.0, '電梯': 1.0, '主要用途': 1.0, '經度': 121.5364676, '緯度':

Actual price: 63.06 萬元
Predicted price: 66.01 萬元
Hit rate (within 15% error): Yes
```

信義房屋