國立中正大學數學研究所

碩士論文

機器學習模型在預測房價上的表現：以臺中市為例

Performance of Machine Learning Models in Predicting Housing Prices: Taking Taichung City as an Example

指導教授：陳孟豁 博士

研究生：莊佳樺 撰

中華民國 113 年 6 月

摘要

在物價上漲、疫情過後的嚴苛背景之下，理解房價的變動與預測趨勢是件很重要的事情，本研究旨在用機器學習技術，收集內政部不動產實價登錄 2020 年至 2023 年台中市房價，並一併考慮了經濟指標、屋齡與不動產周遭環境等影響房價之因素，以 2020 年至 2022 年不動產買賣資料做訓練，並以 2023 年不動產買賣資料驗證其預測可行性，期望可提供一個可靠的參考給欲購買不動產者。

在機器學習演算法上，透過了解線性迴歸中的梯度下降法 (Gradient Descent)、決策樹迴歸 (Decision Tree Regressor)、隨機森林迴歸 (Random Forest Regressor) 與支援向量迴歸 (Suppoer Vector Regressor) 的演算法運作，並比較四個模型這在訓練集資料的表現，取較優的實驗結果去驗證新資料。

根據實驗結果，四種模型的差異不大，在訓練與驗證中，隨機森林迴歸的表現都是最好的，其次是支援向量迴歸。以模型特徵權重來說，影響房價最多的因素是建物移轉總面積，其次是屋齡與周遭環境，在現實生活中，這些因素確實會影響買房的意願與價格。綜合上述的驗證，此研究可以成為可靠的參考。

關鍵字: 台中市、房價、機器學習、線性回歸、梯度下降、決策樹迴歸、隨機森林迴歸、支援向量迴歸

ABSTRACT

Under the harsh background of rising prices and the aftermath of the epidemic, it is important to understand the changes in real estate prices and predict the trend. The purpose of this study is to collect the real estate prices of Taichung City from the Ministry of the Interior (MOI) for the period of 2020 to 2023 by using machine learning techniques, and to verify the feasibility of the predictions by using the data of the real estate purchase and sale of real estate for 2022 to 2022, taking into consideration the factors affecting the prices of real estate such as the economic indicators, the age of the house, and the real estate surrounding environment. The data of real estate transactions from 2020 to 2022 is used for training, and the data of real estate transactions in 2023 is used to verify the feasibility of the prediction, which is expected to provide a reliable reference for those who want to buy real estate.

In terms of machine learning algorithms, we understand the operation of Linear Regression with Gradient Descent, Decision Tree Regression, Random Forest Regression, and Support Vector Regression, and compare the performance of the four models on the training data, so that the better experimental results can be used to validate the new data.

According to the experimental results, there is not much difference among the four models, and Random Forest Regression has the best performance in both training and validation, followed by Support Vector Regression. In terms of the weights of the model features, the factor that affects the price of a house most is the total area transferred, followed by the age of the house and the surrounding environment, which do affect the willingness to buy a house and the price of a house in real life. Taken together, this study can serve as a reliable reference.

Keywords：Taichung City、Housing Prices、Machine Learning、Linear Regression、Gradient Descent、Decision Tree Regression、Random Forest Regression、Support Vector Regression

目錄

圖目錄

表目錄

第一章　緒論

1.1　研究背景與動機

在 2023 與 2024 年的今天，大家都身處於萬物皆漲、薪水沒漲的時代，有許多年輕人都意識到了或許打拼一輩子都很難有屬於自己的一個家，所以‘躺平族’這一個名詞開始成為當代年輕人的目標，‘躺平族’意旨與其為了滿足社會期望而努力奮鬥，不如選擇躺平，過著無慾無求，只為滿足自己能夠維持最低生活標準的處世態度。

在 2019 年爆發的 Covid-19 新冠疫情對全世界包括臺灣的經濟環境都造成了重大的影響，當大家都以為會跟 2003 年的 SARS 一樣，疫情會使房價降到最低點，待疫情回穩之後才會趨於正常，但是這次的疫情卻不僅沒有下跌還一路往上升，其中的原因可能為政府的防疫政策、台商回流等等，這無疑對多數人都是雪上加霜。身處這些嚴苛環境下的我們，如果以買房當作目標，除了等待政府的政策、市場交易、提升自己多賺點錢外，還能做哪些事來做準備呢?

以 2020 年 Q1 開始至 2023 年 Q4 的臺中市不動產買賣為例，查看每個月的買賣平均可以發現是一個穩定上漲的趨勢，在這種環境下，對於那些仍然希望擁有自己的家的人來說，理解房價的變動和預測未來的趨勢是一個很重要的事情。然而，不論在疫情爆發前後，房價的變動都受到許多因素的影響，在這種情況下，機器學習似乎提供了一種可靠的方案，好讓我們能透過大量的數據訓練出可靠的模型去預測房價，並根據預測結果，提供給還在努力的人們一個目標，也給欲買房的人們一個參考。

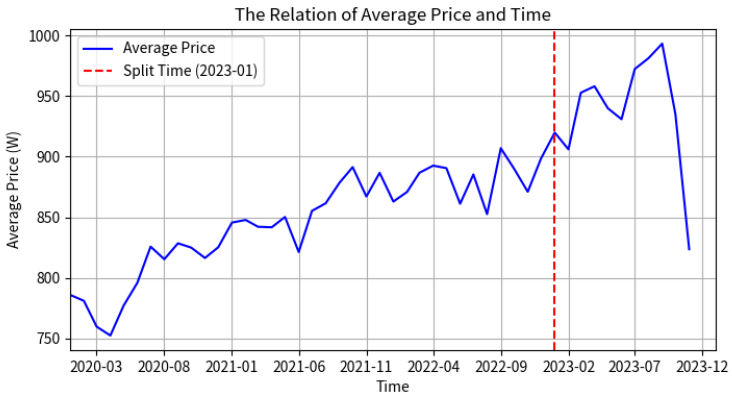


圖 1. 2020 至 2023 臺中市房價當月平均折線圖

1.2 研究目的

本研究透過內政部不動產實價查詢服務網與政府開放式平台收集臺中市 2020 年至 2023 年的臺中市不動產買賣資訊，將資料整理完後分割成 2020 年至 2022 年不動產賣賣資料作為訓練、2023 年作為驗證。透過線性迴歸、決策樹迴歸、隨機森林迴歸、支援向量迴歸模型預測房價，並衡量各模型的預測結果，再使用此結果去驗證新資料。

1.3 論文架構

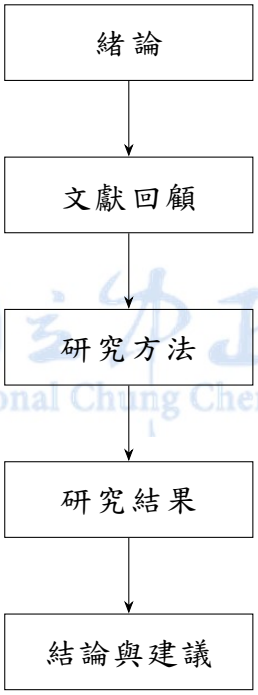


圖 2. 論文架構

第二章　文獻回顧

1. 張建發 (2019)

該研究由經政策、經濟指標、政治變數與國際情勢等面向去探討影響房價之因素，先初步擬定這 4 大構面與 21 項指標後，再經德爾菲法挑選出該 4 大構面與其中 17 項指標，經挑選出的指標有: 財經政策的貨幣供給額 (M1b)、經濟指標的國民所得、政治變數的政治穩定性和國際情勢的國際局勢穩定度等等。再透過層級程序分析法訂定權重。經研究得出結論為若要調節房市，透過財經政策是最好的方法，且財經政策中權重較高的指標為: 利率、稅制與貨幣供給額 (M1b)，而政治變數與國際局勢影響較低。

2. 邱國祥 (2020)

該研究使用 2018 年至 2019 年台中市原市區房屋、土地買賣的實價登錄資料，模型上則使用多元線性迴歸分析 (Multiple Linear Regression Analysis)、正則項迴歸分析 (Lasso and Ridge Regression Analysis)、隨機森林迴歸 (RandomForest Regression) 以及極限梯度提升法 (eXtreme Gradient Boosting)，並請使用 RMSE 與 MAPE 作為模型的衡量指標。經研究發現，多元線性回歸在經過特徵選後 RMSE 與 MAPE 反而都提高；Lasso 迴歸在 α=0.01 時有最低的 RMSE，為 241.48；Ridge 迴歸在 α=1 實有最低的 RMSE，為 235.60，比較所有的實驗結果，極限梯度提升法有著最好的實驗結果，RMSE 為 215.50、MAPE 為 24.404%。

3. 陳玟寧 (2022)

該研究使用 2016 年至 2020 年的臺北市不動產公開資料，並使用 Lasso 迴歸 (Lasso Regression)、嶺迴歸 (Ridge Regression)、支援向量迴歸 (Support Vector Regressor)、核嶺迴歸 (Kernel Ridge Regression) 與彈性網路 (Elastic) 對原始數據集與刪除非住宅的數據集進行預測比較。經研究結果顯示，在原始數據的預測比較上，支援向量迴歸有最好的表現；在刪除非住宅的數據集上亦同；最後使用 Minitab 的最佳子集迴歸選擇每個最佳誤差的各種預測變數組合模型後，得出對於影響房價的重要變數為: 建物移轉總面積平方公尺、主建物面積、建物現況格局-房、車位總價元、主要建材、鄉鎮市區與交易標的，這些都是在該類別中使用次數最多的變數，經實驗結果表示，所使用的變數越多，各模型的表現都會越好。在新資料的表現上，支援向量迴歸是最好的演算法，以平均來說 R^2 為 0.19151；MAPE 為

22.83939%；MAE 為 6882667。

4. 廖思閔 (2023)

該研究利用桃園市 2017 至 2021 實價登錄資料來預測房價，並使用了自動化機器學習所建議之隨機森林 (RandomForest, RF)、梯度提升機 (Grandient Boosting Machine, GBM) 和極限梯度提升 (eXtreme Gradient Boosting, XGBoost) 模型來評估預測的表現，分別探討了在不同變數個數下與不同的超參數組合在模型上有何種表現，經研究發現各模型在只有 20 個變數的表現最佳，以 R^2 來說，隨機森林來到 0.8032；梯度提升機 0.8113；極限梯度提升 0.7368。經超參數優化後，梯度提升機的表現最好，在訓練資料不相同的情況下，各模型取 5 次的實驗結果，梯度提升機的模型表現有著最好的平均值，分別為 R^2=0.8348；RMSE=8827802；MAE=1254301；MAPE=12.28%。

5. 王尹暘 (2023)

該研究聚焦於利用決策樹模型預測台南世紀之門大樓的房價，採用 CART 和 CHAID 兩種決策樹方法。研究分析了從實價登錄系統和政府機關公開數據中收集的房價數據，並進行了深入的比較分析。研究結果表明，CHAID 方法在預測準確度上優於 CART 方法，且兩者的 R^2 均超過 0.75，顯示出對房價的高解釋能力。此研究不僅提供了一種有效的房價預測工具，也增強了對資料透明度和市場條件影響房價的理解。

第三章　研究方法

3.1　研究流程

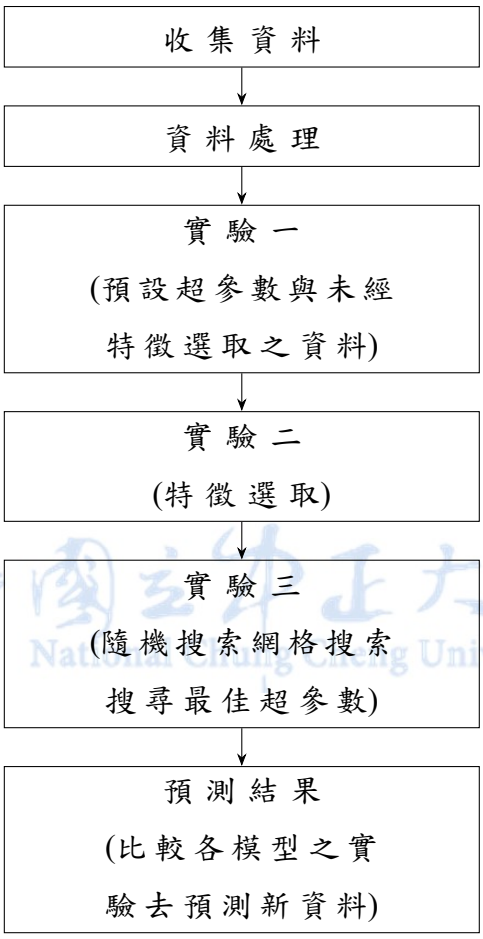


圖 3. 研究流程

3.2　資料來源與資料處理

- 資料來源為內政部不動產交易實價查詢服務網之 2020 至 2022 臺中市行政區不動產實價價登錄資料，以 2020 至 2022 年資料共 168550 筆資料做訓練並以 2023 年份之資料共 50362 筆做驗證。

- 於內政部不動產資訊平台網站下方的住宅資訊動態看板，新增市場經濟欄位，分別是: 五大行庫平均房貸利率 (%)、消費者物價指數、M1b 貨幣供給額 (億元) 以及經濟成長率。

圖 4. 內政部不動產交易實價查詢服務網

圖 5. 住宅資訊動態看板

- 原始資料欄位為: 鄉鎮市區、交易標的、土地位置建物門牌、土地移轉總面積平方公尺、都市土地使用分區、非都市土地使用分區、非都市土地使用編定、交易年月日、交易筆棟數、移轉層次、總樓層數、建物型態、主要用途、主要建材、建築完成年月、建物移轉總面積平方公尺、建物現況格局-房、建物現況格局-廳、建物現況格局-衛、建物現況格局-隔間、有無管理組織、總價元、單價元平方公尺、車位類別、備註、編號、主建物面積、附屬建物面積、陽台面積、電梯、移轉編號。

- 預先確認所有資料都在指定的日期中，訓練集資料為 2020/01/01 至 2022/12/31；驗證集資料為 2023/01/01 至 2023/12/31。

- 將都市土地使用分區分類為: 住、農、工、商、其他。

- 刪除交易標的為土地之資料。

- 將交易筆棟數拆分為: 土地數量、建物數量、車位數量。

- 過濾掉主要用途欄位非住家用之資料。
  訓練集資料該欄位各值之原始數量為: 住家用:99840、住商用:7254、商業用:3917、其他:1564、辦公用:1526、...、管理室、停車空間:1 和住宅、樓梯間:1。修正後為住家用:99840。
  驗證集資料同理，修正後為住家用:17518。

- 檢查土地位置建物門牌的完整度，將‘台中市’改為‘臺中市’、若地址沒有其鄉鎮市區的值，就補上該列鄉鎮市區的值，讓土地位置建物門牌統一為‘臺中市 XX 區 XXXXXXXXXX’。

修改前		修改後	
鄉鎮市區	土地位置建物門牌	鄉鎮市區	土地位置建物門牌
西屯區	台中市西屯區文華路 100 號	西屯區	臺中市西屯區文華路 100 號
南區	大慶街二段 130 號	南區	臺中市南區大慶街二段 130 號

Table 1: 土地位置建物門牌修改前後範例

- 依照臺中市 2022 各行政區人口統計，將訓練集資料的鄉鎮市區依照人口數量由多到少排列，再取總人口數平均以上的行政區，並將平均以下歸類為其他。

總人數為: 2,814,459 人；共 29 個行政區; 平均人口為 97050 人。大於人口數量的行政區由多到少排列為: 北屯區、西屯區、大里區、太平區、南屯區、豐原區、北區、南區、西區、潭子區。再將此排列在最後新增一值為其他，並由大到小將行政區與其他替換成數字: 10 至 0。

驗證集資料總人口為: 2,845,909 人，平均人口為 98134 人，達於總人口數之行政區由大到小排列為: 北屯區、西屯區、大里區、太平區、南屯區、豐原區、北區、南區、西區、潭子區、沙鹿區。再將此排列在最後新增一值為其他，並由大到小將行政區與其他替換成數字: 11 至 0。

- 新增屋齡欄位，計算方法為: (交易年月日 - 建築完成年月) / 365 , 以年為單位，並取到小數點第一位。

- 將總樓層數與移轉層次轉換成整數型態，例如:

	修改前	修改後
移轉層次	地下一層、一層、兩層	3
總樓層數	十五層	15

Table 2: 移轉層次與總樓層數修改前後範例

- 將建物型態少於其他之項目都歸類為其他。

訓練集資料該欄位各值之原始數量為: 住宅大樓 (11 層含以上有電梯):37060、透天厝:17161、華廈 (10 層含以下有電梯):13077、公寓 (5 樓含以下無電梯):6132、套房 (1 房 1 廳 1 衛):1180、其他:24、店面 (店鋪):21、辦公商業大樓:10、工廠:1、倉庫:1。

修正後為: 住宅大樓 (11 層含以上有電梯):37060、透天厝:17161、華廈 (10 層含以下有電梯):13077、公寓 (5 樓含以下無電梯):6132、套房 (1 房 1 廳 1 衛):1180、其他:57。

驗證集資料為: 住宅大樓 (11 層含以上有電梯):8863、透天厝:3405、華廈 (10 層含以下有電梯):3394、公寓 (5 樓含以下無電梯):1511、套房 (1 房 1 廳 1 衛):134、其他:2。已同為訓練集資料修正後結果，無須修改。

- 過濾掉主要建材欄位非鋼筋混凝土造、加強磚造、鋼骨鋼筋混凝土造、鋼筋混凝土加強磚造之資料。
- 將土地移轉總面積平方公尺、建物移轉總面積平方公尺、主建物面積、附屬建物面積、陽台面積轉換成坪數並將欄位名稱的平方公尺更改成坪數。(坪數 = 平方公尺 ×0.3025)
- 將都市土地使用分區、有無管理組織、建物現況格局-隔間、建物型態、主要建材計算每個類別的數量，並將該類別轉換成該類別的數量，例如: 得知有無管理組織欄位分別為: 有或無，其出現次數分別為 53820 與 20847，將有替換成 56664、將無替換成 28221，這個的做法相較 OneHotEncoder 更能減少資料的維度，從而減少過擬合的風險。
- 刪除總價元為 0 與其離群值之資料。

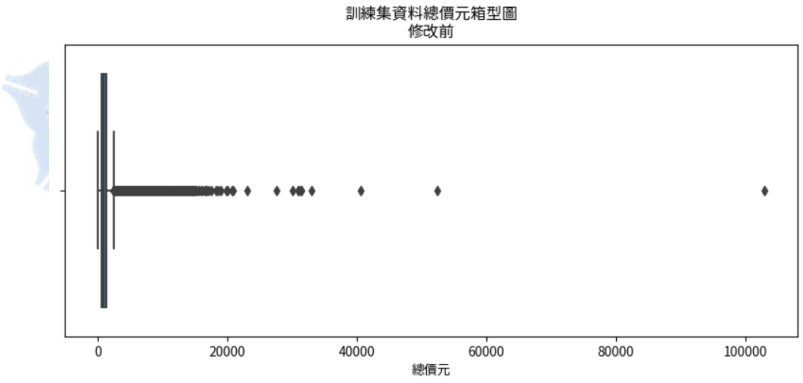


圖 6. 訓練集資料總價元箱型圖 - 修改前

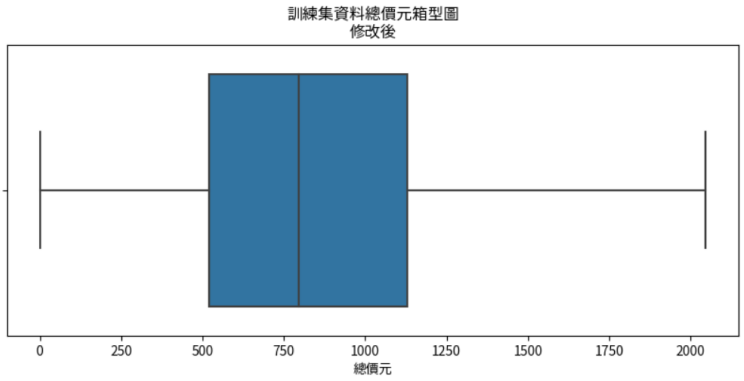


圖 7. 訓練集資料總價元箱型圖 - 修改後

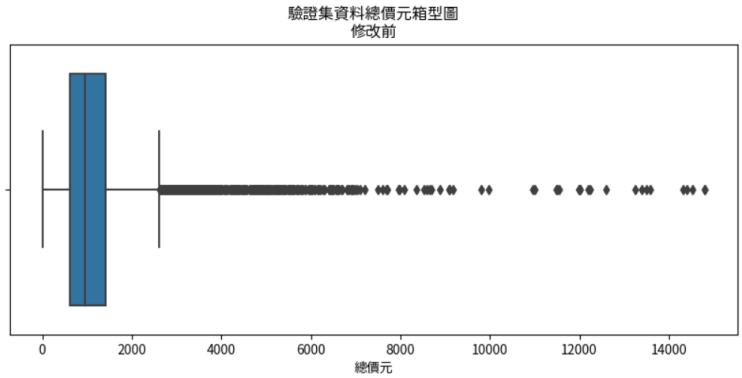


圖 8. 驗證集資料總價元箱型圖 - 修改後

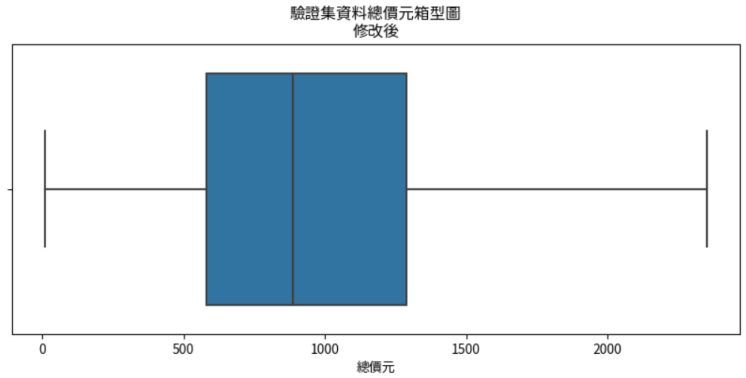


圖 9. 驗證集資料總價元箱型圖 - 修改後

- 將總價元/10000 以便將來在訓練模型時，能避免數值過大導致的數值不穩定性的問題。

- 於臺中市政府資料開放平台找到臺中市重要地標 (111 年版) 之後，經整理之後共 6679 筆資料，地標大分類有: 便利商店、公共設施、文教機構、醫療保健、金融機構、百貨公司、飯店 (餐飲)、育樂場所與運輸服務。再依照地標名稱透過爬蟲獲取各地標之經緯度。這些分類佔比如下圖所示:

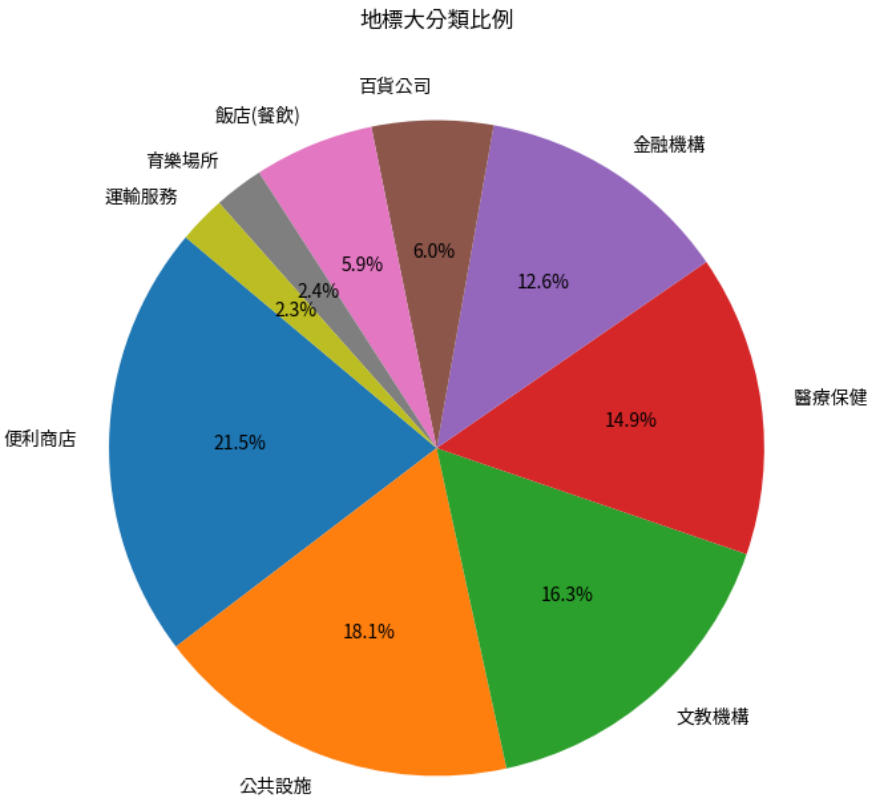


圖 10. 地標大分類比例

圖 11. 臺中市政府資料開放式平台

- 依照土地位置建物門牌的地址，透過爬蟲獲取經緯度後，利用 Haversine 公式計算土地建物與地標之距離

Haversine 公式是一種根據兩點的經緯度來確定大圓上兩點之間的距離計算方法，

公式為:

$$d = 2R\sin^{-1}\left(\sqrt{\sin^2\left(\frac{\theta_2-\theta_1}{2}\right)+\cos\theta_1\cos\theta_2\sin^2\left(\frac{\lambda_2-\lambda_1}{2}\right)}\right) \quad (1)$$

其中 R = 6371 為地球半徑 (單位: 公里)

(λ_1, θ_1) 為第一個地點的經緯度，並轉換成弧度。

(λ_2, θ_2) 為第二個地點的經緯度，並轉換成弧度。

新增 points 欄位，以考量周遭環境對建物的影響。計算方法為若此地標距離建物 0.5 公里內得 4 分; 距離在 0.5 公里到 2 公里內得 1.5 分; 距離在 2 公里到 3.5 公里內得 0.5 分; 距離大於 3.5 公里到 5 公里內得 0.1 分; 距離大於 5 公里得 0 分，將上述得分加總得到 points。

- 刪除空白值過多、已經使用完並且已生成新欄位之欄位與對模型性能影響不大之欄位。

  刪除的欄位為: 土地位置建物門牌、交易年月日、建築完成年月、非都市土地使用分區、非都市土地使用編定、交易標的、交易筆棟數、單價元平方公尺、備註、編號、電梯、車位類別、車位移轉總面積平方公尺、車位總價元、移轉編號、車位數量、主要用途、經度、緯度。

經上述資料處理過程 2020 至 2022 年資料共 66479 筆，共 25 個欄位；2023 年資料共 15916 筆，共 25 個欄位。此 25 個欄位分別為: 鄉鎮市區、土地移轉總坪數、都市土地使用分區、移轉層次、總樓層次、建物型態、主要建材、建物移轉總坪數、建物現況格局-房、建物現況格局-廳、建物現況格局-衛、建物現況格局-隔間、有無管理組織、總價元、主建物坪數、附屬建物坪數、陽台坪數、土地數量、建物數量、屋齡、五大行庫平均房貸利率 (%)、消費者物價指數、M1b 貨幣供給額 (億元) 與 points。

3.3 機器學習

機器學習是人工智慧的分支，是現今當代人工智慧發展當中，最重要的技術之一，其目標是能讓電腦從過去的資料中學習與改進，並且不需要程式碼持續編譯，機器學習應用程式會隨著使用不斷改善，存取的資料越多、準確度也會越高。機器學習主要分成 4 種方法，分別是監督式學習、非監督式學習、半監督式學習以及強化學習，以下是這 4 種方法之比較。

機器學習方法	定義	過程
監督式學習	模型通過學習從標記的訓練數據中進行預測或分類	訓練階段，模型接收由帶有標籤的特徵和相應輸出組成的數據集。模型通過學習特徵和輸出之間的映射來進行訓練。在測試階段，模型用於預測新數據的輸出
非監督式學習	模型在訓練數據中沒有標籤，並開始使用所有相關且可存取的資料來識別模式和關聯性	模型通常試圖通過聚類（Clustering）或降維（Dimensionality Reduction）等技術，發現數據中的模式
半監督式學習	半監督式學習是監督式學習和非監督式學習的結合，其中模型在訓練數據中同時擁有標籤和沒有標籤的數據	使用以標記的資料進行訓練，再用已訓練好資料訓練未知的資料，並將已標記之資料加入訓練資料中
強化學習	智能體（agent）通過與環境互動來學習。智能體會接收獎勵或懲罰，並嘗試最大化獎勵	智能體透過對現在環境的觀察，得出一個使其獎勵最大化的決策，並更新現在的策略，以便在未來獲得更好的獎勵

本研究用機器學習中的監督式學習，並使用 Python 中的 scikit-learn 模組進行模型的建模、訓練、驗證以及測試。

scikit-learn 是 Python 當中非常強大的機器學習套件，其中包括了提供許多機器學習的演算法、內建數據集可供練習、模型評估或選擇等，在機器學習的研究中，帶來非常大的便利性。

3.4 資料標準化

資料標準化是將資料之特徵值縮放到平均值為 0，標準差為 1 的範圍內。其公式如下:

$$x' = \frac{x - \bar{x}}{\sigma} \tag{2}$$

x' 為縮放結果，x 為縮放前之資料，$\bar{x}$ 為該欄特徵之平均值，σ 為標準差

由於一筆數據是由眾多特徵所組成，這些特徵的分布狀況可能都不盡相同，而這些因素可能會影響某些模型的收斂速度，如梯度下降 (Gradient Descent) 等，為避免這些問題的發生以及提升模型性能或泛化能力，所以在建模前，將數據進行資料標準化的動作，是非常重要的步驟。

3.5 線性迴歸 (Linear Regression)

以機器學習的角度而言，線性迴歸是標籤 (依變數 Dependent variable, y) 與特徵 (自變數 Independent variables, $x_1, x_2, ..., x_i$) 之間的線性關係，一般分成兩種:

- 簡單線性迴歸 (Simple Linear Regression)

  如果標籤只受單一特徵影響就是簡單線性迴歸，可以用以下公式表示:

$$y_i = \beta_0 + \beta_1 x_i + \varepsilon_i \tag{3}$$

  其中 y_i 表示第 i 個標籤，β_0 為截距，β_1 為特徵權重；ε_i 是隨機誤差項。

- 多重線性迴歸 (Multiple Linear Regression)

  在很多情況下，特徵可能不只收到一個特徵的影響，可能會有很多個，這時候就很適合用多重線性迴歸，公式表示如下:

$$y_i = \beta_0 + \beta_1 x_{1i} + \beta_2 x_{2i} + ... + \beta_k x_{ki} + \varepsilon_i \tag{4}$$

  其中，x_{ji} 表示第 i 個樣本的第 j 個特徵，$(\beta_1, \beta_2, \cdots, \beta_k)$ 為特徵權重。

為了使線性迴歸模型能夠適應數據，會採用梯度下降 (Gradient Descent) 的方法進行模型參數的學習。該演算法是一種迭代求解最優解的方法。在每次迭代中，梯度下降法會沿著梯度方向（即損失函數下降最快的方向）更新參數，使預測值與實際值誤差最小，算法如下:

1. 以多元線性迴歸為例，可以將多元線性迴歸用下列表示，並假設此為預測線:

$$\hat{y_i} = w_1 x_1 + w_2 x_2 + \cdots + w_i x_i + b \tag{5}$$

   其中 $\hat{y_i}$ 為預測值；$(x_1, x_2, \cdots, x_i)$ 為特徵；$(w_1, w_2, \cdots w_i)$ 為權重；b 為誤差項

2. 設定一個損失函數 $L(w_i, b)$，表示實際值與預測值之間的誤差，通常以均方誤差 (Mean Squared Error, MSE) 作為損失函數，表示為:

$$L(w_i, b) = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y_i})^2 \tag{6}$$

3. 先隨機定義一個初始值，以損失函數分別計算該值之 w_i 與 b 方向之斜率且令為 ∇w_i 與 ∇b，可將損失函數寫成

$$L(w_i, b) = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y_i})^2 = \frac{1}{n}\sum_{i=1}^{n}(y_i - (w_1x_1 + w_2x_2 + \cdots + w_ix_i + b))^2 \tag{7}$$

計算 w_i 方向斜率之損失函數，即對 w_i 做偏微分:

$$\nabla w_i = \frac{\partial L(w_i, b)}{\partial w_i} = \frac{1}{n}\sum_{i=1}^{n} 2 \cdot (-x_i) \cdot (y_i - (w_1x_1 + w_2x_2 + \cdots + w_ix_i + b)) \tag{8}$$

故 w_i 方向斜率之損失函數為:

$$\nabla w_i = \frac{1}{n}\sum_{i=1}^{n} -2 \cdot x_i \cdot (y_i - \hat{y_i}) \tag{9}$$

b 方向斜率同理，其損失函數為:

$$\nabla b = \frac{1}{n}\sum_{i=1}^{n} -2 \cdot (y_i - \hat{y_i}) \tag{10}$$

4. 設定一個學習率 α，讓 w_i 方向、b 方向斜率進行 m 次迭代，使得預測值與真實值誤差達到最低點。學習率控制著每次迭代的更新幅度。學習率過大可能導致算法不收斂，而學習率過小可能導致算法收斂速度過慢。每次迭代過程如下:

$$w_{it+1} = w_{it} - \alpha \cdot \nabla w_{it} \tag{11}$$

$$b_{t+1} = b_t - \alpha \cdot \nabla b_t \tag{12}$$

w_{it} 表示當下權重之值，w_{it+1} 為下一次迭代後的權重值；b_t 表示當下誤差項之值，b_{t+1} 為下一次迭代後的誤差值，其中 t 表示迭代次數，$t = 0, 1, 2, \ldots, m$。

5. 當迭代結束時，會生成一組最佳的 w_i 與 b，使損失函數降到最低點。在訓練模型之後，可以使用驗證集來評估模型的性能。如果模型在驗證集上的表現不佳，則可能需要調整模型的參數或重新訓練模型。

值得注意的是，針對多重線性迴歸，亦或是現實生活中的情況，往往很難直觀的去找到最低點，因為都是由預設定義的初始點去做迭代，一步步去試探並找出最小值，所以可能會因為定義的初始值不同，而導致不同的結果。

3.6 決策樹 (Decision Tree)

決策樹屬於監督式學習演算法，它通過對數據集的特徵閥值條件反覆的進行分割，最終形成一棵樹狀結構。它模擬了人類做決策的過程，其優點在於高效率、淺顯易懂、對於決策過程有很高的解釋性；缺點是當特徵數量多時，模型會變得過於複雜且容易因為模型超參數調整不當而過度擬合。

對於決策樹的分類或迴歸，都有適合的演算法去做處理，分類問題適合用 ID3(Iterative Dichotomiser 3)、C4.5；迴歸則適合用 CART(Classification and Regression Trees)，CART 亦可處理分類問題。以下使用 CART 算法做說明:

對於決策樹，可以用以下遞迴式表示:

$$G(x) = \sum_{c=1}^{C} [\![b(x) = c]\!] \, G_c(x) \tag{13}$$

其中，C 表示在每次分割中生成的子節點數量，在 CART 中，C 的值固定為 2，表示每次劃分將節點分為 2 類；$b(x)$ 為分支標準 (Branching Criteria)，代表該節點的條件判斷式；G_c 表示為第 c 個節點下的子樹遞迴式。而每一個子樹的遞迴式為:

$$G_c(x) = \sum_{c=1}^{C} [\![b_c(x) = c]\!] \, G_{ci}(x) \tag{14}$$

即眾多的子樹 $G_c(x)$ 依照分支標準構成一整棵樹 $G(x)$。

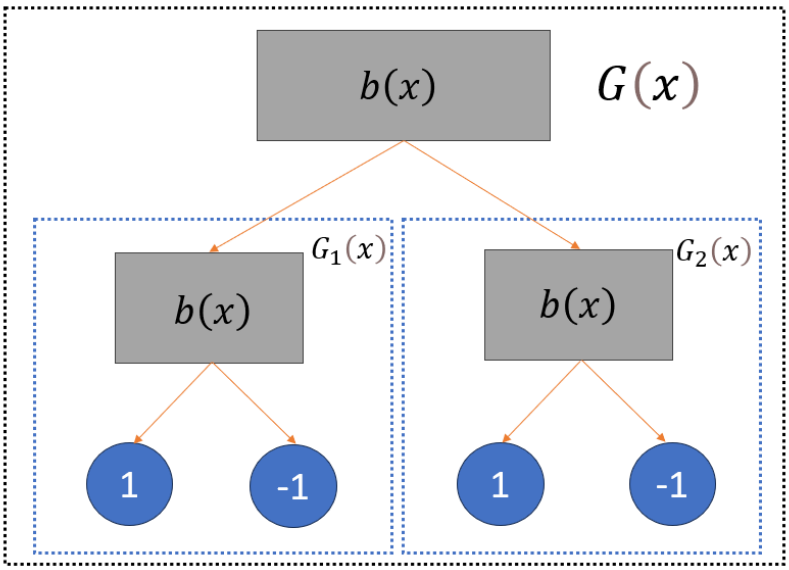


圖 12. 簡易決策樹結構圖

每次劃分資料的方法叫做決策樹樁 (Decision Stump)，代表只對資料做一次分割，即從根節點直接通到葉節點的一個樹狀結構，在圖 12 中，$G_1(x)$ 與 $G_2(x)$ 是子樹也是一個決策樹樁。

在 CART 的分類樹中，所使用的分類基準為基尼不純度 (Gini impurity)，代表著分類乾淨的程度，其值介於 $[0, 1]$，越接近 0 代表分類的越乾淨，反之，值越接近 1，公式表示為:

$$GI(D) = 1 - \sum_{i=1}^{n} p_i^2 \tag{15}$$

其中，n 代表資料集 D 中，不同分類區塊的數量，p_i 為該區塊中不同類別的數據點所佔的比例。而對於機器學習來說，其核心概念為最小化預測值與實際值的差異，所以會選擇一個特徵和一個閾值來最小化每一個分類區塊或是每一個樹樁中節點的不純度:

$$\min_{feature,split} \frac{|D_{left}|}{|D|} GI(D_{left}) + \frac{|D_{right}|}{|D|} GI(D_{right}) \tag{16}$$

這裡的 feature 為該節點所要考慮的特徵；split 為該特徵的分割值，D_{left} 與 D_{right} 表示資料集 D 依照 feature 與 split 分割成的左右兩個子集合，$|D|$、$|D_{left}|$ 與 $|D_{right}|$ 表示 D、D_{left} 與 D_{right} 中數據點的個數；$\frac{|D_{left}|}{|D|}$ 與 $\frac{|D_{right}|}{|D|}$ 為左右子樹中樣本所佔總樣本的比例。

而整個分類樹遞迴式的中止條件為:

- 該節點的 Gini impurity 為 0，表示該節點的數據完美分類。

- 當一個節點的數據量小於某個預設的閾值時停止分割，可自行定義，以防止過度擬合。
- 達到樹的最大深度，也是防止過度擬合的自訂值。

在最後所產生的葉子節點中，分類樹會是該節點上最多數據點的類別標籤，迴歸樹則是該節點上所有數據點目標值的平均。

而在迴歸的任務中，同樣是使用 CART 演算法，跟分類樹不同的是，分類樹是使用基尼不純度 (Gini impurity) 去衡量分類純度，迴歸樹通常是使用均方誤差 (MSE) 來衡量分割後數據的同質性。故可以表示成:

$$\min_{feature,split} \frac{|D_{left}|}{|D|} MSE_{left} + \frac{|D_{right}|}{|D|} MSE_{right} \tag{17}$$

而這裡的均方誤差表示為:

$$\begin{aligned} MSE_{left} &= \frac{1}{|D_{left}|} \sum_{i \in D_{left}} (y_i - \bar{y}_{left})^2 \\ MSE_{right} &= \frac{1}{|D_{right}|} \sum_{i \in D_{right}} (y_i - \bar{y}_{right})^2 \end{aligned} \tag{18}$$

其中，y_i 是子集中數據點的目標值，$\bar{y}_{left}$ 與 $\bar{y}_{right}$ 分別為 D_{left} 與 D_{right} 中所有目標值之平均值。

而整個迴歸樹遞迴式的中止條件為:

- 達到自訂義樹的最大深度，避免過度擬合並減少計算複雜度。
- 節點中的樣本數小於自訂義的閥值，避免在數據稀疏的情況下繼續分割。

3.7 隨機森林 (Random Forest)

隨機森林是一種處理分類或迴歸的監督式學習演算法，以決策樹演算法為基礎，透過 Bagging (Bootstrap Aggregation) 方法將各棵決策樹結合起來，形成一片森林。

Bagging 屬於集成學習 (Ensemble Learning) 中的一種方法，集成學習方法的目的在於結合多個模型的預測來提升整體的預測性能，而 Bagging 就是通過取後放回的抽樣 (Boostrap) 來生成多個樣本子集，然後使用每個樣本子集來訓練模型，這些模型可以是同質的或異質的，最終，Bagging 將這些模型的預測結果進行平均或投票，以得到最終的集成預測結果。所以隨機森林的構建就是隨機抽取訓練集與特徵來構建決策樹，並將這個過程重複數次，最後再把這些樹用投票的方式結合起來，就可以得到隨機森林。

- 分類 (Classification)

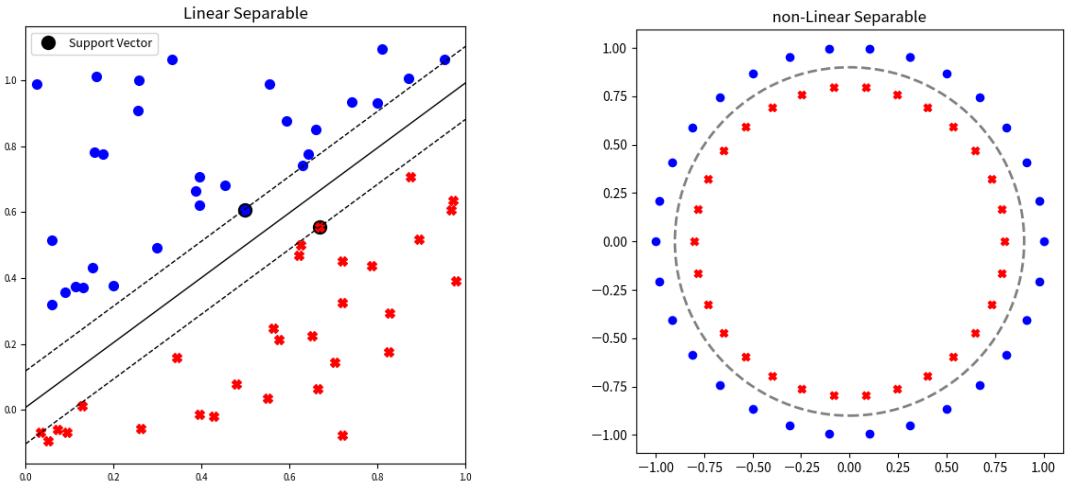
  在分類任務中，隨機森林使用多棵決策樹來預測目標變量的類別。每棵樹對數據集進行訓練，並對每個觀測值進行預測。最終的預測結果是基於所有樹的投票得出的，即取得最多票數的類別作為最終的分類結果。

- 迴歸 (Regression)

  在迴歸任務中，隨機森林同樣使用多棵決策樹來預測目標變量的數值。每棵樹對數據集進行訓練，並對每個觀測值進行數值預測。最終的預測結果是所有樹預測值的平均或加權平均。

3.8 支援向量機 (Support Vector Machine)

支援向量機是一種處理分類或迴歸的監督式學習演算法，目標是找到一個最大化邊際的超平面，以區分不同類別的數據點。一般而言，支援向量機可以分為線性可分與非線性可分，線性可分的目的是在二維平面中，找到一條讓兩類別之間間隔寬度最大的直線，而兩類別離此條直線最近的點就稱為支援向量。然而現實中的資料很難在二維平面上做出分類，可以將資料映射到更高維度使得可以將此資料成功地做出分類，這就是非線性可分，如圖 13 所示，支援向量機可以使用不同的核函數 (kernel) 成功地做到這件事情。且在分類任務中所決定之邊際又可區分為硬性邊界 (Hard-Margin) 與軟性邊界 (Soft-Margin)，兩者差別在於是否容忍誤差值。



a. 線性可分　　b. 非線性可分

圖 13. 支援向量機

- 分類 (Classification)

  支援向量在分類的任務中與上述的說明大致相同，使用適合的 kernel 將資料成功地做分類。

  支援向量機在線性可分情況下的硬性邊界標準分類問題為:

$$\min_{w,b} \frac{1}{2} w^T w$$
$$\text{subject to} \quad y_i(w^T x_i + b) \geq 1 \quad \text{for } i = 1, 2, \cdots, n \tag{19}$$

  求解 w、b 後即可找到一條使得邊際最大的那條決策邊界，也能進一步討論支援向量在何種情況會發生。

- 迴歸 (Regression)

  在迴歸任務中，支援向量機的目標是擬合一個迴歸函數來逼近輸出值，並預測連續值的輸出，同時保持輸出值與實際值之間的誤差最小化。假設最後的擬合迴歸線為:

$$f(x) = w^T x + b \tag{20}$$

  支援向量迴歸的非線性標準問題為:

$$\min_{w,b,\xi^+,\xi^-} \frac{1}{2} w^T w + C \sum_{i=1}^{n} (\xi_i^+ + \xi_i^-)$$
$$\text{subject to} \quad -\varepsilon - \xi_i^- \leq w^T z_i + b - y_i \leq \varepsilon + \xi_i^+, \ \xi_i^+ \geq 0, \xi_i^- \geq 0 \tag{21}$$

其中，C 為正則化參數，用來權衡誤差值在模型中的重要性，ε 為自定義的參數，定義邊際的長度大小，ξ_i^+ 與 ξ_i^- 分別表示在邊際上下的數據點離邊際的距離。

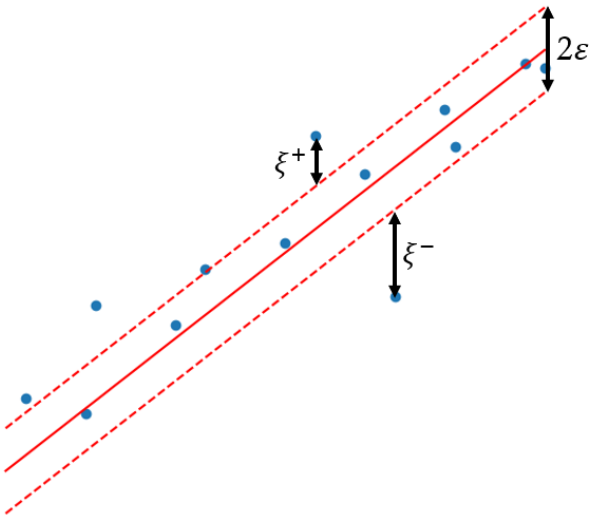


圖 14. 支持向量迴歸

再來將其根據拉格朗日函式轉化成對偶問題可得:

$$\min_{\alpha^+,\alpha^-} \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}(\alpha_i^+ - \alpha_i^-)(\alpha_j^+ - \alpha_j^-)\underbrace{z_i^T z_j}_{\mathbf{K(x_i,x_j)}} + \sum_{i=1}^{n}\left[\alpha_i^+(\varepsilon + y_i) + \alpha_i^-(\varepsilon - y_i)\right]$$

$$\text{subject to } \sum_{i=1}^{n}(\alpha_i^+ - \alpha_i^-) = 0,\ w_i = -\sum_{i=1}^{n}(\alpha_i^+ - \alpha_i^-)z_i,\ 0 \le \alpha_i^+ \le C,\ 0 \le \alpha_i^- \le C \tag{22}$$

其中 $\mathbf{K(x_i, x_j)}$ 為核函數映射，叫做核計巧 (Kernel Trick) 就是將非線性可分的資料映射到更高維度的空間，使該資料在此空間是線性可分，而常見的核計巧有:

1. 線性核函數 (Linear Kernel):

$$K(x, x') = (x \cdot x') \tag{23}$$

2. 多項式核函數 (Polynoimal Kernel):

$$K(x, x') = (\gamma(x \cdot x') + r)^Q \tag{24}$$

3. 徑向基函數 (Radius Basis Function, RBF Kernel):

$$K(x, x') = \exp(-\frac{||x - x'||^2}{2\sigma^2}) = \exp(-\gamma||x - x'||^2) \tag{25}$$

4. S 型核函數 (Sigmoid Kernel):

$$\tanh(\gamma(x \cdot x') + r) \tag{26}$$

其中，Q 為多項式核函數中指定多項式的最高次方、γ 為適用於多項式核函數、徑向基函數與 S 型核函數之核係數；r 為適用於多項式核函數與 S 型核函數的獨立項係數

此時即可求解 α^+、α^-，就可將此結果去計算最後的擬合迴歸線

$$\begin{aligned} w_i &= -\sum_{i=1}^{n}(\alpha_i^+ - \alpha_i^-)z_i \\ b &= y_i + \sum_{i=1}^{n}(\alpha_i^+ - \alpha_i^-)K(x, x') \end{aligned} \tag{27}$$

上下界方程式通過支援向量，而支持項量發生在 $\alpha_i^+ > 0$、$\alpha_i^- > 0$、$\xi_i^+ = \xi_i^- = 0$ 的條件下，則此時的 b^+、b^- 為

$$\begin{aligned} b^+ &= y_{sv}^+ + \sum_{i=sv}^{n}(\alpha_i^+ - \alpha_i^-)K(x_{sv}^+, x_i) \\ b^- &= y_{sv}^- + \sum_{i=sv}^{n}(\alpha_i^+ - \alpha_i^-)K(x_{sv}^-, x_i) \end{aligned} \tag{28}$$

接下來即可得出上下界方程式:

$$\begin{aligned} f_+(x) &= -\sum_{i=sv}^{n}(\alpha_i^+ - \alpha_i^-)K(x, x') + b^+ \\ f_-(x) &= -\sum_{i=sv}^{n}(\alpha_i^+ - \alpha_i^-)K(x, x') + b^- \end{aligned} \tag{29}$$

3.9 衡量指標

- 均方根誤差 (Root Mean Squared Error, RMSE)

  在之前提到的梯度下降使用之損失函數為 MSE，然而 MSE 也是一種權衡模型好壞的指標，而 RMSE 相較 MSE 而言，數值更小，有助於更好理解模型的預測誤差，也對較大的誤差有更強的懲罰效果，其公式為:

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{30}$$

  其值範圍為 $[0, \infty)$，值越小表示模型表現越好。

- 決定係數 (R-squared, R^2) 決定系數，是一個迴歸模型性能的評估指標，用於衡量模型解釋資料的變異程度。R^2 的計算方式如下:

$$R^2 = 1 - \frac{SSE}{SST} \tag{31}$$

其中，SSE (Sum of Squared Errors)，為殘差平方和，表示模型預測值和實際觀測值之間的差異的平方和，公式表示如下:

$$SSE = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 \tag{32}$$

SST (Total Sum of Squares)，為總平方和，表示實際觀測值的變異的平方和。

$$SST = \sum_{i=1}^{n} (y_i - \bar{y}_i)^2 \tag{33}$$

以上，y_i 是真實值；$\hat{y}_i$ 是預測值；$\bar{y}_i$ 是真實值之平均。
R^2 的範圍會介於 $(-\infty, 1]$ 之間，數值越大，表示模型對目標變數解釋度越好。

3.10 過擬合 (Overfitting)

- 當一個模型在訓練集的表現上非常好，但在測試集或新數據上的表現卻大幅下降，這種現象稱為過擬合 (Overfitting)，意味著模型的泛化能力不足。

- 用來偵測是否發生過擬合有許多種方式，常見的方法為交叉驗證，就是將數據集分割成多個子集，模型在一部分子集上進行訓練，而在另一部分子集上進行測試，以確保模型在未見數據上的表現，並反覆做訓練及驗證。再將訓練集分成驗證集，常見的訓練集、驗證集、測試集的分割比例為 60%、20% 和 20%，在用訓練集訓練完模型時，如果模型在驗證集的解釋能力 (R^2) 或是誤差值比訓練集來的差，那就表示發生了過擬合。

- 用來避免過擬合的常見方法有提前停止模型訓練、降低模型複雜度、進行特徵選擇 (Feature Selection)、正則化 (Regularization)，正則化分為 L1 正則化和 L2 正則化，它們的作用是在模型的損失函數中加入一個懲罰項，以限制模型參數的大小，透過對參數增加約束使得模型不會過分地擬合訓練數據。L1 正則化使用參數的絕對值加入損失函數作為懲罰項，傾向於產生稀疏模型，使其中一些參數等於零；而 L2 正則化使用參數的平方，促使所有參數都趨於較小的值，但不會使它們完全為零。特徵選擇會在下個章節做說明。

3.11 特徵選擇 (Feature Selection)

上一節有提到特徵選擇是模型用來預防過擬合的方法，它通過選擇重要的特徵，忽略不重要的特徵，來降低模型的複雜度，常見的特徵選擇方法分成三大類:

- 過濾法 (Filter Methods)

  通過評估各個特徵與目標變量之間的關聯性來進行特徵選擇，通常是設定一個閾值進而確定特徵子集。例如: 方差閾值 (VarianceThreshold)、相關係數 (Correlation Coefficient) 等。其優點是運算速度快且不占用電腦資源，缺點是可能會忽略特徵子集之間的交互作用，因為其篩選方法是單一特徵與目標變量的計算結果。

- 包裝法 (Wrapper Methods)

  在模型的訓練過程中，結合搜尋策略評估不同的特徵子集，以尋找在模型訓練中能夠達到最佳性能的特徵組合。優點是因為直接使用模型效能去評估，所以能更好的考慮特徵之間的交互作用，進而獲取最佳特徵子集，缺點在於因需反覆訓練模型，計算成本會過高，再加上若搜尋策略不當，也會產生過擬合的問題。常見的包裝法為: 前向搜尋 (Forward Selection)、後向搜尋 (Backward Selection)、逐步搜尋 (Stepwise Selection) 等。

- 嵌入法 (Embedded Methods)

  直接使用模型的訓練過程中的特徵重要性或權重，以選擇最佳的特徵子集。其結合了過濾法與包裝法的優點，能跟包裝法一樣考慮特徵之間的交互作用，也有著過濾法的運算速度，常見的方法為: 正則化、隨機森林或決策樹的特徵權重等。

3.12 超參數優化

在使用機器學習模型中，有許多超參數可供調整，透過調整超參數，可使模型的效能顯著性的提高，並進一步提升其泛化能力。而隨機搜索 (Random Search)、網格搜索 (Grid Search) 與 K 折交叉驗證 (K-fold Cross-Validation) 的搭配使用能使各模型找到其最適合的超參數組合。

- 隨機搜索 (Random Search)

  隨機搜索是一種通用的超參數優化方法，其做法是在超參數的範圍內進行隨機抽

樣，並評估模型的性能。此方法優點在於其高效性，能夠在相對較少的試驗次數內找到較好的超參數組合。

- 網格搜索 (Grid Search)
  網格搜索是一種系統性的搜索方法，它在預先定義的超參數範圍中，生成一個超參數的網格，然後對每一個點進評估。這種方法的優勢在於其全面性，會嘗試所有可能的超參數組合，但也因此需要更多的計算資源。

- K 折交叉驗證 (K-fold Cross-Validation)
  屬於交叉驗證中的其中一種方法，在這種方法中，數據集被分成 K 個子集，之後輪流把其中一份當作測試集，然後模型在 K-1 個子集上進行訓練。這個過程被重複 K 次。在超參數優化中，K 折交叉驗證可以幫助評估每組超參數的性能，以更穩健的方式進行模型選擇。

這三種方法的搭配使用童常為先使用隨機搜索來探索大範圍的參數空間，然後在找到的最佳區域內使用網格搜索進行細節的調整，並通過 K 折交叉驗證來評估這些超參數的性能，以確保最終選擇的超參數組合在不同數據集上都具有較好的泛化性能。

第四章　研究結果

- 隨機種子 (random_state) 統一為 42，為確保結果的可重現性。
- 每組實驗都會用驗證集進行是否過擬和的評估。
- 訓練集訓練結果為第一個表格，驗證集為第二個表格。
- 觀察模型訓練與超參數優化耗時，為在模型比較時作為參考。
- 比較模型各實驗之誤差高低，最後選擇誤差值最小的實驗階段去預測 2023 年之房價資料。

4.1　實驗一: 使用自訂義或預設超參數

- 使用自定義或模型預設超參數進行訓練，且不對特徵做處理。

以下為各模型之超參數解釋:

4.1.1　線性迴歸-梯度下降

- m: 迭代次數，自訂義為 10000 次。
- α: 學習率，自訂義為 0.01。

4.1.2　決策樹迴歸

- criterion: 節點劃分的標準。提供的超參數有: squared_error (均方誤差)、friedman_mse(改良版均方誤差)、absolute_error (絕對值誤差)、poisson (布阿松偏差)，預設為 squared_error。
- splitter: 節點劃分的策略。提供的超參數有: random(隨機)、best(最佳)，預設為 best。
- max_depth : 樹的最大深度，控制樹的最大深度，以防止過擬和。輸入為整數，預設為 None。

- min_samples_split : 內部節點再劃分所需的最小樣本數，如果節點的樣本數少於此值，則不會劃分。輸入為整數或浮點數，預設為 2。
- min_samples_leaf : 葉節點所需的最小樣本數，如果葉節點的樣本數少於此值，則不會劃分。輸入為整數或浮點數，預設為 1。
- max_feature : 在劃分節點時要考慮的特徵數量。輸入為整數、浮點數、sqrt 或 log2，預設為 None。
- random_state : 控制每次運行時劃分的隨機性。輸入為整數，預設為 None。
- max_leaf_nodes : 限制最大葉子節點個數。輸入為整數，預設為 None。
- min_impurity_decrease : 如果分裂將導致不純度減少大於或等於此值，則分裂將被採納。輸入為浮點數，預設為 0.0。
- ccp_alpha : 控制最小成本複雜度修剪的程度。輸入為非負浮點數，預設為 0.0。
- monotonic_cst : 對每個特徵施加單調性約束的陣列。輸入為-1、0 或 1，預設為 None。

4.1.3 隨機森林迴歸

隨機森林迴歸與決策樹迴歸的超參數大致相同，以下為隨機森林迴歸才有的超參數進行說明。

- n_estimators：隨機森林中決策樹的數量。輸入為整數，預設為 100。
- min_weight_fraction_leaf: 葉子節點的最小樣本權重和。輸入為浮點數，預設為 0.0。
- bootstrap: 是否在構建樹時使用放回抽樣。輸入為布林值，預設為 True。
- n_jobs: 並行運行的工作數量。輸入為整數，預設為 None。
- verbose: 控制擬合和預測時的詳細程度。輸入為整數，預設為 0。
- warm_start: 指定是否在訓練過程中使用之前訓練的模型參數作為初始化。輸入為布林值，預設為 False。

- max_samples: 控制隨機森林中每顆決策樹的訓練樣本的數量或比例。輸入為整數或浮點數，預設為 None。

4.1.4 支援向量迴歸

- kernal : 控制模型在高維空間中擬和數據的方式。提供的超參數有:linear (線性)、poly (多項式)、rbf (徑向基函數)、sigmoid (雙曲正切函數) 與 precomputed (預先計算的核矩陣)，預設為 rbf。
- degree: 多項數核函數的次數，僅適用於多項式 (poly)，在其他核函數會被忽略。輸入為非負整數，預設為 3。
- gamma : 適用於 rbf、ploy、sigmoid 核函數之核係數。輸入為 auto、scale 或浮點數，預設為 scale，浮點數需為非負數。
- coef0 : 核函數的獨立項，僅在 poly 和 sigmoid 中有意義。輸入為浮點數，預設為 0.0。
- tol : 在模型訓練中控制何時停止的條件。輸入為浮點數，預設為 0.001。
- C : 正則化參數。輸入為浮點數且嚴格為正，預設為 1.0。
- epsilon: 指定了在訓練損失函數中預測值與實際值之間的距離小於此值的點不會受到影響，反之模型會給予該值一相關的誤差值。輸入為浮點數，預設為 0.1。
- shrinking : 是否開啟收縮啟發式，以提高模型訓練速度。輸入為布林值，預設為 True。
- cache_size : 控制核矩陣的緩存大小 (MB)。輸入為浮點數，預設為 200。
- verbose : 是否在訓練過程中詳細輸出訓練過程。輸入為布林值，預設為 False。
- max_iter : 限制模型訓練的最大迭代次數。輸入為整數，預設為-1。

4.1.5 實驗一結果

模型	RMSE	R^2	模型訓練時間 (秒)
線性迴歸-梯度下降	238.954103	0.709837	68.637
決策樹迴歸	233.213284	0.724769	1.163
隨機森林迴歸	164.568875	0.862948	120.024
支援向量迴歸	252.492223	0.677384	76.207

Table 3: 實驗一之訓練集誤差

模型	RMSE	R^2
線性迴歸-梯度下降	231.924972	0.727802
決策樹迴歸	227.681914	0.737605
隨機森林迴歸	162.001809	0.867157
支援向量迴歸	253.881837	0.673741

Table 4: 實驗一之驗證集誤差

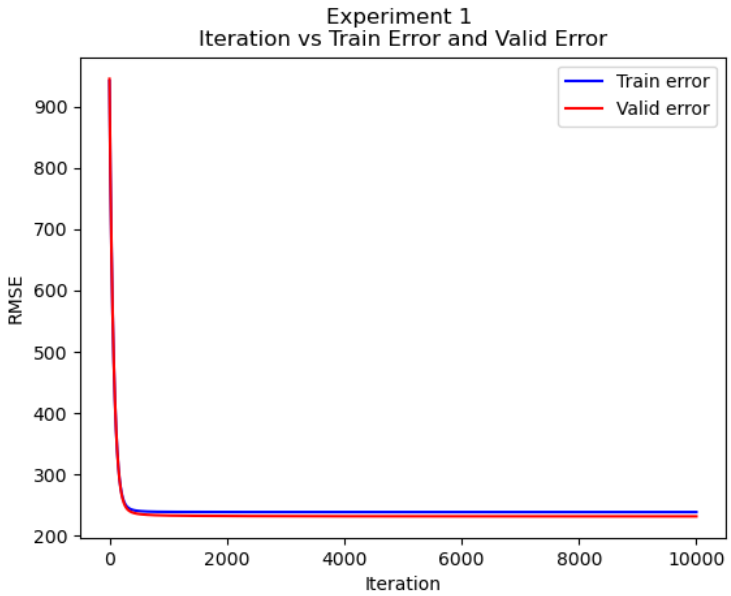


圖 15. 梯度下降 10000 次迭代圖

根據實驗一結果，在四個模型中，無論是訓練集或是驗證集，都是隨機森林迴歸表現最好，它的 RMSE 值最低；R^2 值最高。這表明隨機森林迴歸模型對數據的擬合能力最強。然而在模型的訓練時間上，決策樹迴歸模型的訓練快其他三者非常多，這也進一步驗證了決策樹模型訓練速度快的特點。也可以發現的是支援向量迴歸有輕微過擬合的現象，待之後的實驗進行改進。

4.2 實驗二: 使用特徵選擇

- 實驗二使用特徵選擇中的嵌入法進行特徵篩選。線性迴歸-梯度下降、隨機森林迴歸、支援向量迴歸都使用隨機森林迴歸模型中的特徵重要性進行篩選，決策數迴歸則使用該模型本身的特徵重要性。
- 所篩選出的每一個特徵為該特徵之權重大於等於所有特徵權重中位數的特徵。
- 超參數部分與實驗一相同。

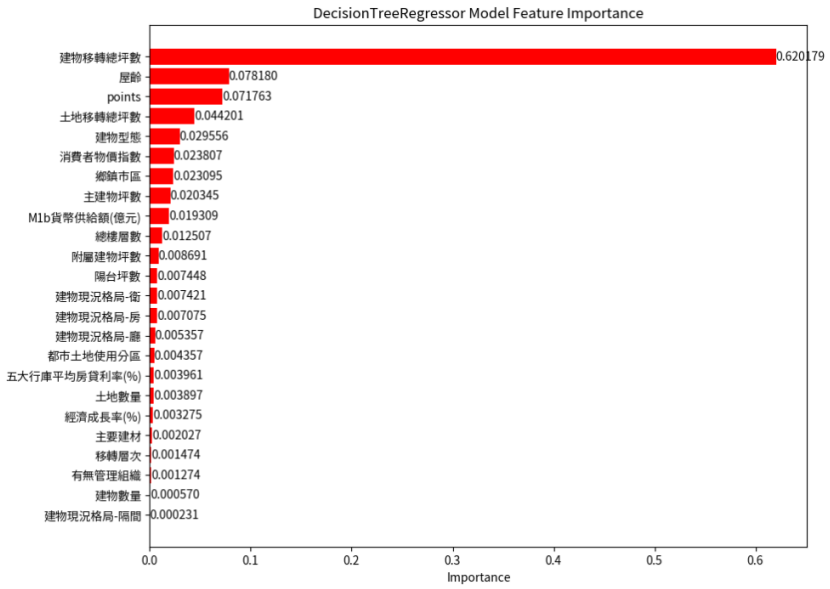


圖 16. 決策樹迴歸模型特徵權重

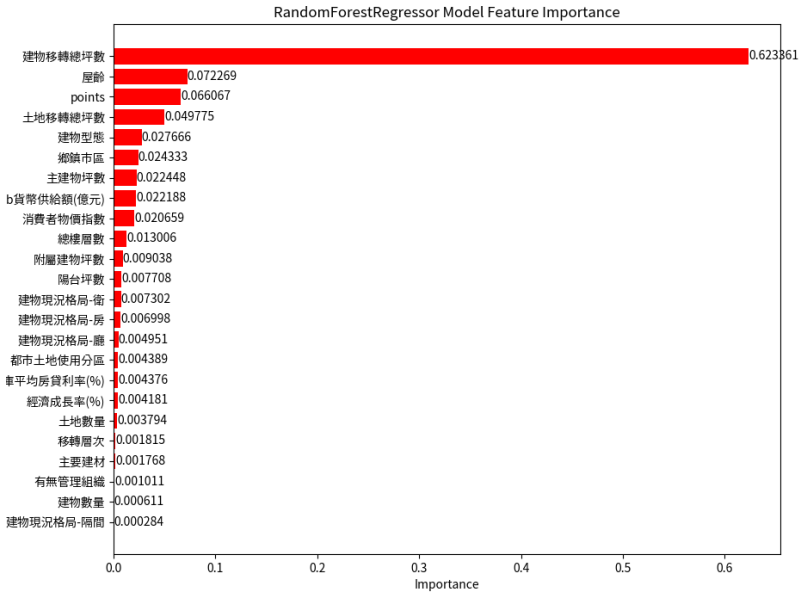


圖 17. 隨機森林迴歸模型特徵權重

	決策樹迴歸		隨機森林迴歸	
中位數	0.007435		0.007505	
權重	特徵	權重	特徵	權重
	建物移轉總坪數	0.620179	建物移轉總坪數	0.623361
	屋齡	0.078180	屋齡	0.072269
	points	0.071763	points	0.066067
	土地移轉總坪數	0.044201	土地移轉總坪數	0.049775
	建物型態	0.029556	建物型態	0.027666
	消費者物價指數	0.023807	鄉鎮市區	0.024333
	鄉鎮市區	0.023095	主建物坪數	0.022448
	主建物坪數	0.020345	M1b 貨幣供給額 (億元)	0.022188
	M1b 貨幣供給額 (億元)	0.019309	消費者物價指數	0.020659
	總樓層數	0.012507	總樓層數	0.013006
	附屬建物坪數	0.008691	附屬建物坪數	0.009038
	陽台坪數	0.007448	陽台坪數	0.007708

Table 5: 特徵選擇所篩選之特徵

4.2.1 實驗二結果

模型	RMSE	R^2	模型訓練時間 (秒)
線性迴歸-梯度下降	265.508670	0.641763	55.639
決策樹迴歸	235.315390	0.719785	1.022
隨機森林迴歸	168.316809	0.856634	115.617
支援向量迴歸	243.192237	0.700712	70.516

Table 6: 實驗二之訓練集誤差

模型	RMSE	R^2
線性迴歸-梯度下降	258.160059	0.662737
決策樹迴歸	233.451993	0.724137
隨機森林迴歸	166.094951	0.860359
支援向量迴歸	243.782654	0.699182

Table 7: 實驗二之驗證集誤差

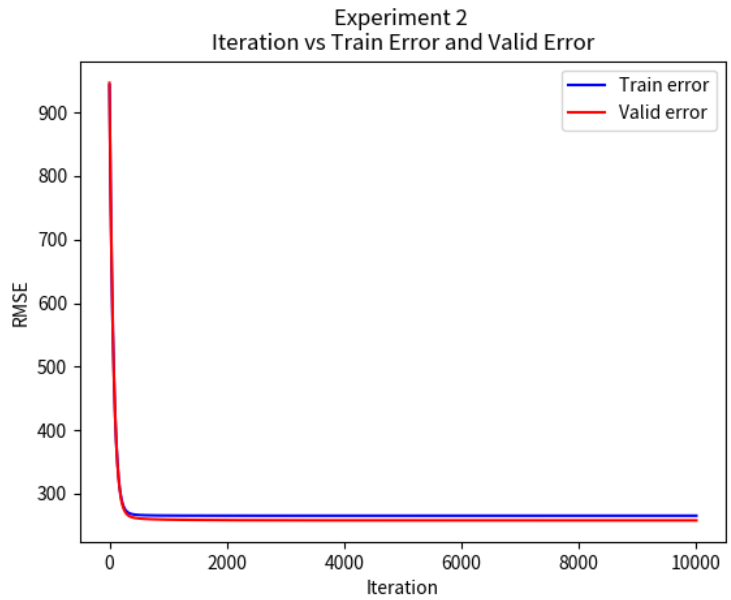


圖 18. 實驗二之梯度下降 10000 次迭代圖

在進行特徵選擇後，四種模型在訓練集和驗證集上的 RMSE 和 R^2 值都有一定程度的變化，說明特徵選擇對模型性能有一定影響。隨機森林迴歸在經過特徵選擇之後，無論是訓練集或驗證集的表現仍是四種模型中誤差最低；支持向量迴歸相較其他模型，誤差值相比實驗一來的小，但是過擬合的問題還是存在，可是訓練集誤差與驗證集誤差也縮小了，或許在超參數優化之後，可以使模型泛化能力與性能更好。

4.3 實驗三: 超參數優化

- 延用實驗二特徵選取完的訓練集，分別對四個模型做隨機搜索與網格搜索並搭配 K 折交叉驗證來搜索影響模型性能較大的超參數並組合成一組誤差值最小的超參數組合，再利用搜索出來的超參數去訓練模型，並依據模型的誤差去判斷是否過擬合，再決定要使用哪個階段的實驗去預測新資料。

4.3.1 線性迴歸-梯度下降

線性迴歸-梯度下降的隨機搜索範圍與結果如下表所示。隨機選擇 20 組參數組合，然後對每個組合進行 10 折交叉驗證擬合。在隨機搜索 20 組超參數組合後，得到最佳的超參數組合為 { 學習率 (α) = 0.08 , 迭代次數 (m) = 5000 }；10 折驗證集 RMSE 平均為 300.034798；共耗時 2769.854 秒。

超參數	隨機搜索範圍	隨機搜索結果
學習率 (α)	[0.01 , 0.02 , 0.03 , $\cdots$ 0.3]	0.08
迭代次數 (m)	{ 5000 , 10000 , 15000 , 20000 }	5000
平均誤差	300.034798	
隨機搜索時間 (秒)	2769.854	

Table 8: 線性迴歸-梯度下降之隨機搜索

將隨機搜索的結果稍做微調進行網格搜索，設定學習率 (α) 的搜索列表為其 $\frac{2}{3}$ 倍、學習率 (α) 本身與其 $\frac{3}{2}$ 倍；迭代次數的搜索列表為次數本身與其正負 1000 次之間，這樣設定的目的是不要讓其數值離本身太遠或太近，可以涵蓋到潛在的相關範圍。兩搜索列表共會產生 $3 \times 3 = 9$ 個網格，每個網格進行 10 折交叉驗證擬合後，得到的最佳超參數組合為 { 學習率 (α) = 0.053333 , 迭代次數 (m) = 4000 }，10 折驗證集 RMSE 平均為 300.034519；共耗時 463.621 秒。

超參數	網格搜索範圍	網格搜索結果
學習率 (α)	[$\frac{2}{3}\times 0.08$, 0.08, $\frac{3}{2}\times 0.08$]	0.053333
迭代次數 (m)	{ 3000 , 4000 , 5000 }	4000
平均誤差	300.034519	
網格搜索時間 (秒)	463.621	

Table 9: 線性迴歸-梯度下降之網格搜索

4.3.2 決策樹迴歸

決策樹迴歸的隨機搜索範圍與結果如下表所示，隨機選擇 20 組參數組合，然後對每個組合進行 10 折交叉驗證擬合。在隨機搜索 20 組超參數組合後，得到最佳的超參數組合為 { criterion = poisson , splitter = best , max_depth = 50 , min_samples_split = 26 , min_samples_leaf = 25 , max_feature = 12 }；10 折驗證集 RMSE 平均為 196.932024；共耗時 1664.938 秒。

超參數	隨機搜索範圍	隨機搜索結果
criterion	squared_error、absolute_error、friedman_mse、poisson	poisson
splitter	best、random	best
max_depth	None、{ 5 , 10 , 15 , 20, ⋯ , 50 }	50
min_samples_split	{ 2 , 6 , 10 , ⋯ , 41 }	26
min_samples_leaf	{ 1 , 5 , 9 , ⋯ , 40 }	25
max_feature	None、sqrt、log2、{ 1 , 2 , 3 , ⋯ , 12 }	12
平均誤差	196.932024	
隨機搜索時間 (秒)	1664.938	

Table 10: 決策樹迴歸之隨機搜索

將隨機搜索的結果稍做微調進行網格搜索，網格搜索之範圍與結果如下表，共會產生 1 × 1 × 1 × 3 × 3 × 1 = 9 個網格，每個網格進行 10 折交叉驗證擬合後，得到的最佳超參數組合為 { criterion = possion , splitter = best , max_depth = 48 , min_samples_split = 25 , min_samples_leaf = 25 , max_feature = 12 }；10 折驗證集 RMSE 平均為 197.862825；共耗時 165.743 秒。

超參數	網格搜索範圍	網格搜索結果
criterion	poisson	poisson
splitter	best	best
max_depth	{ 48 , 49 , 50 }	48
min_samples_split	{ 25 , 26 , 27 }	25
min_samples_leaf	{ 24 , 25 , 26 }	25
max_feature	{ 12 }	12
平均誤差	197.862825	
網格搜索時間 (秒)	165.743	

Table 11: 決策樹迴歸之網格搜索

4.3.3 隨機森林迴歸

隨機森林迴歸的隨機搜索範圍與結果如下表所示，隨機選擇 20 組參數組合，然後對每個組合進行 10 折交叉驗證擬合。在隨機搜索 20 組超參數組合後，得到最佳的超參數組合為 { criterion = squared error , max_depth = None , max_features = log2 , min_samples_leaf = 1 , min_samples_split = 2 , n_estimators = 550 }；10 折驗證集 RMSE 平均為 165.985468；n. 搜索共耗時 709333.216 秒。

超參數	隨機搜索範圍	隨機搜索結果
criterion	squared_error、absolute_error、friedman_mse、poisson	squared_error
max_features	None、sqrt、log2、{ 1 , 2 , 3 ，⋯ , 13}	log2
max_depth	None、{ 5 , 10 , 15, ⋯ , 30 }	None
min_samples_split	{ 2 , 4 , 6 , ⋯ , 10 }	2
min_samples_leaf	{ 1 , 4 , 7 , 10 }	1
n_estimators	{ 100 , 150 , 200 , ⋯ , 1000 }	550
平均誤差	165.985468	
隨機搜索時間 (秒)	709333.216	

Table 12: 隨機森林迴歸之隨機搜索

將隨機搜索的結果稍做微調進行網格搜索，網格搜索之範圍與結果如下表，共會產生 $1 \times 1 \times 1 \times 2 \times 2 \times 5 = 20$ 格網格，每個網格進行 10 折交叉驗證擬合後，得到的最佳超參數組合為 { criterion = squared error , max_depth = None , max_features = log2 , min_samples_leaf = 1 , min_samples_split = 2 , n_estimators = 570 }；10 折驗證集 RMSE 平均為 165.965751；共耗時 30857.770 秒。

超參數	網格搜索範圍	網格搜索結果
criterion	squared_error	squared_error
max_feature	{ log2 }	log2
max_depth	{ None }	None
min_samples_split	{ 2 , 3 }	2
min_samples_leaf	{ 1 , 2 }	1
n_estimators	{ 530 , 540 , 550 , 560 , 570 }	570
平均誤差	165.965751	
網格搜索時間 (秒)	30857.770	

Table 13: 隨機森林迴歸之網格搜索

4.3.4 支援向量迴歸

隨機森林的隨機搜索範圍與結果如下表所示，隨機選擇 20 組參數組合，然後對每個組合進行 10 折交叉驗證擬合。C 與 epsilon 輸入為浮點數，故搜索範圍設定為在 $[a, b]$ 之間的連續均勻分布中隨機取一浮點數。在隨機搜索 20 組超參數組合後，得到最佳的超參數組合為 { kernal = rbf , gamma = auto , C = 15.233283 , epsilon = 1.065632 }；10 折驗證集 RMSE 平均為 200.604644；共耗時 14243.901 秒。

可以看到搜索出來的 C 與 epsilon 都分別超過指定的範圍，屬於為正常現象，是因為計算機中浮點數精度的限制所導致的，且超出去的範圍往往不會太大，所以才被認為是正常現象。

超參數	隨機搜索範圍	隨機搜索結果
kernel	linear、rbf	rbf
gamma	scale、auto	auto
C	{ 1 , 2 , 3, ⋯ 15 }	15.233283
epsilon	{ 0.1 , ⋯ , 1 }	1.065632
平均誤差	200.604644	
隨機搜索時間 (秒)	14243.901	

Table 14: 支援向量迴歸之隨機搜索

將隨機搜索的結果稍做微調進行網格搜索，網格搜索之範圍與結果如下表，為實驗便利性，下方表格令 C 與 e 為隨機搜索之結果。此搜索範圍共會產生 $1 \times 1 \times 3 \times 3 = 9$ 格網格。每個網格進行 10 折交叉驗證擬合後，得到的最佳超參數組合為 { kernal = rbf , gamma = auto , C = 22.849925 , epsilon = 1.598448 }；10 折驗證集 RMSE 平均為 197.379146；共耗時 7009.077 秒。

超參數	網格搜索範圍	網格搜索結果
kernel	rbf	rbf
gamma	auto	auto
C	$\{\frac{2}{3}\times C, C, \frac{3}{2}\times C\}$	22.849925
epsilon	$\{\frac{2}{3}\times \varepsilon, \varepsilon, \frac{3}{2}\times \varepsilon\}$	1.598448
平均誤差	197.379146	
網格搜索時間 (秒)	7009.077	

Table 15: 支援向量迴歸之網格搜索

4.3.5 實驗三結果

模型	RMSE	R^2	模型訓練時間 (秒)
線性迴歸-梯度下降	265.508670	0.641763	22.916
決策樹迴歸	198.626665	0.800352	0.355
隨機森林迴歸	165.605775	0.861215	194.519
支援向量迴歸	197.495173	0.802620	82.389

Table 16: 實驗三之訓練集誤差

模型	RMSE	R^2
線性迴歸-梯度下降	258.159976	0.662737
決策樹迴歸	198.186577	0.801186
隨機森林迴歸	162.232843	0.866778
支援向量迴歸	195.629884	0.806282

Table 17: 實驗三之驗證集誤差

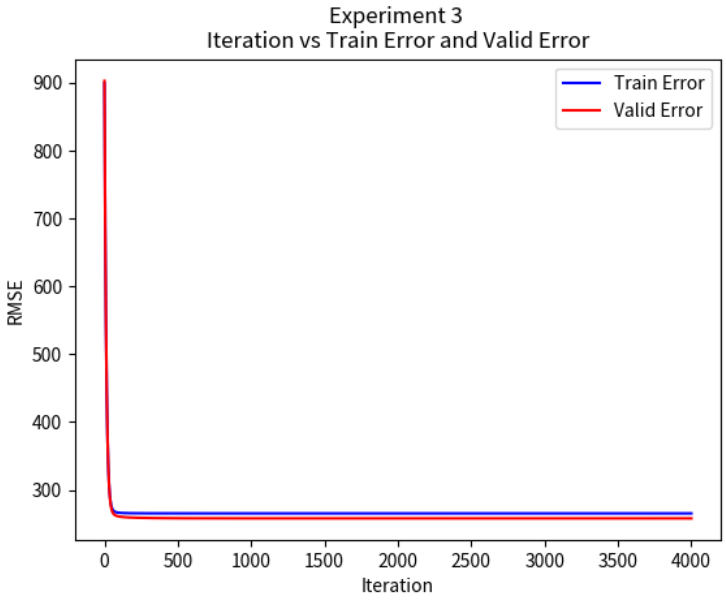


圖 19. 實驗三之梯度下降 4000 次迭代圖

各個模型在經過 3 次實驗後，可以明確的比較出各個模型在哪個階段的實驗結果較佳，決策樹迴歸與支援向量迴歸在經過特徵選擇與超參數優化過後的預測性能都比以往實驗還要好，而且支援向量迴歸也透過超參數優化解決了過擬合的問題，隨機森林迴歸一樣是所有模型中結果最好的，線性迴歸-梯度下降模型進行超參數優化之後與自訂義超參數的結果沒有太大的不同，而且反而沒經過特徵選擇和超參數優化的模型效能還比較好。隨機森林迴歸亦是在實驗一效能最佳，不過差距非常小，故不需要再做優化的動作。

根據上述對模型的觀察與結論，再將線性迴歸-梯度下降用原始數據以及同樣的搜索範圍做超參數優化，以得出最好的實驗結果。

線性迴歸-梯度下降使用原始數據與同樣的搜索範圍作隨機搜索，一樣搜索 20 次超參數組合，每個組合進行 10 折交叉驗證後，最佳的組合為 { 學習率 (α) = 0.013 , 迭代次數 (m) = 5000 }；10 折驗證集 RMSE 平均為 253.532077；共耗時 3489.352 秒。

超參數	隨機搜索範圍	隨機搜索結果
學習率 (α)	[0.01 , 0.02 , 0.03 , $\cdots$ 0.3]	0.013
迭代次數 (m)	{ 5000, 10000, 15000, 20000 }	5000
平均誤差	253.532077	
網格搜索時間	3489.352	

Table 18: 線性迴歸-梯度下降之隨機搜索

將上述結果進行網格搜索，網格搜索範圍如下表所示，在產生 9 個網格，並對這 9 個網格進行 10 折交叉驗證後，搜索結果為 { 學習率 (α) = 0.086667 , 迭代次數 (m) = 4000 }，10 折驗證集 RMSE 平均為 253.532054；共耗時 582.341 秒。

超參數	網格搜索範圍	網格搜索結果
學習率 (α)	{ $\frac{2}{3}\times$0.013, 0.013, $\frac{3}{2}\times$0.013 }	0.086667
迭代次數 (m)	{ 4000, 5000, 6000 }	4000
平均誤差	253.532054	
網格搜索時間	582.341	

Table 19: 線性迴歸-梯度下降之網格搜索

將搜索出來的超參數套入模型當中，所得出之結果之訓練集 RMSE 為 238.953940；R^2 為 0.709837；驗證集 RMSE 為 231.921063；R^2 為 0.727811 模型訓練時間為 30.249 秒。相比實驗一，此階段使用原始數據做超參數優化，在改變之後的學習率與更小的迭代次數即有較優的模型成效。

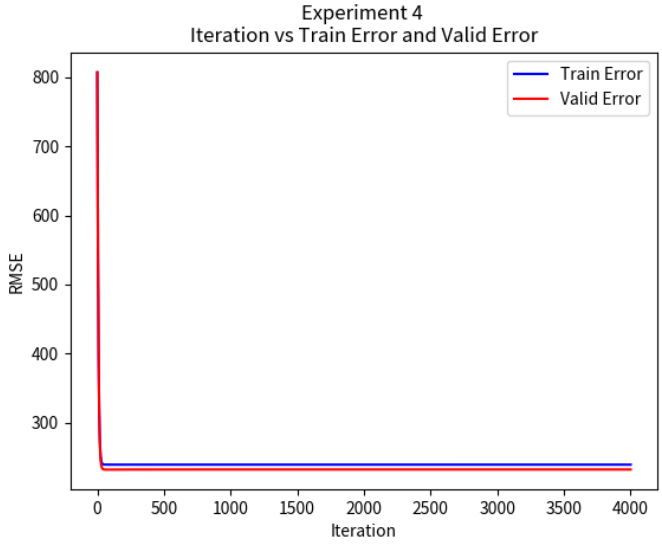


圖 20. 實驗四之梯度下降 4000 次迭代圖

4.4 實驗四: 預測新資料

將以上已訓練完成的模型去預測 112 年資料，所得出結果如下表所示

模型	RMSE	R^2
線性迴歸-梯度下降	307.185532	0.628644
決策樹迴歸	303.906527	0.636529
隨機森林迴歸	279.523733	0.692513
支援向量迴歸	290.015641	0.668997

Table 20: 預測結果

在四種模型中，隨機森林迴歸模型在預測 112 年資料時表現最佳。其 RMSE 為 279.523733，這是四種模型中最好的。同時，其 R^2 為 0.692513 也是四種模型中最高的，表示該模型能夠解釋資料變異的比例最高。因此，可以得出，隨機森林迴歸模型是適合用來預測 112 年房價資料的模型。

第五章　結論與建議

5.1　結論

- 決策樹迴歸、支援向量迴歸經過特徵選擇與超參數優化之後，都達成了使用較少的特徵進而使模型訓練時間減少且模型效能提高的這件事；隨機森林迴歸則是經過特徵選擇與超參數優化才使模型與實驗一模型效能接近；線性迴歸-梯度下降經過特徵選擇後，模型效能下降，經優化也沒讓效能比原始效能還要好，可以猜測特徵之間的交互作用對線性迴歸模型有巨大的影響。

- 在各個實驗中，隨機森林迴歸的模型表現始終是所有模型當中最好的，且在預測新資料上也是如此，更加驗證了隨機森林模型準確度高的優點，但是相較的所付出的時間也相對較多，如果考量模型訓練、優化時間與預測結果，決策樹是一個比較好的選擇，該模型在這三點考量中均有比較優秀的權衡。

5.2　建議

- 觀察線性迴歸-梯度下降模型的迭代圖可以發現，在前面幾次的迭代就已經降到最低點附近，後面的迭代也趨於平緩，欲尋求更好的誤差值，才設定較大的迭代次數，若考量效率方面，觀察迭代圖之後即可將迭代次數設定的小一點，可使效率提高不少。

- 因考量了模型訓練或是超參數優化的耗時，故只考慮影響模型較大的或是較多研究者使用之超參數，若考慮更多的超參數應該有助於模型效果的提升。

- 本研究使用了網路爬蟲獲取建物和地標經緯度後去計算建物與地標之距離，並新增一欄位 points，為了就是要查看建物周遭的公共設施是否會對房價的造成重大的影響，但是相關程度不高。可能是因為在做爬蟲時，獲取的經緯度有些微的誤差，亦或是計算分數設的標準不夠好，如果未來能夠能對此多加實驗與改進，或是能更加深入去探討蛋黃區與非蛋黃區建物的周遭環境，應該能更好的驗證周遭環境對買房的影響程度。

附件

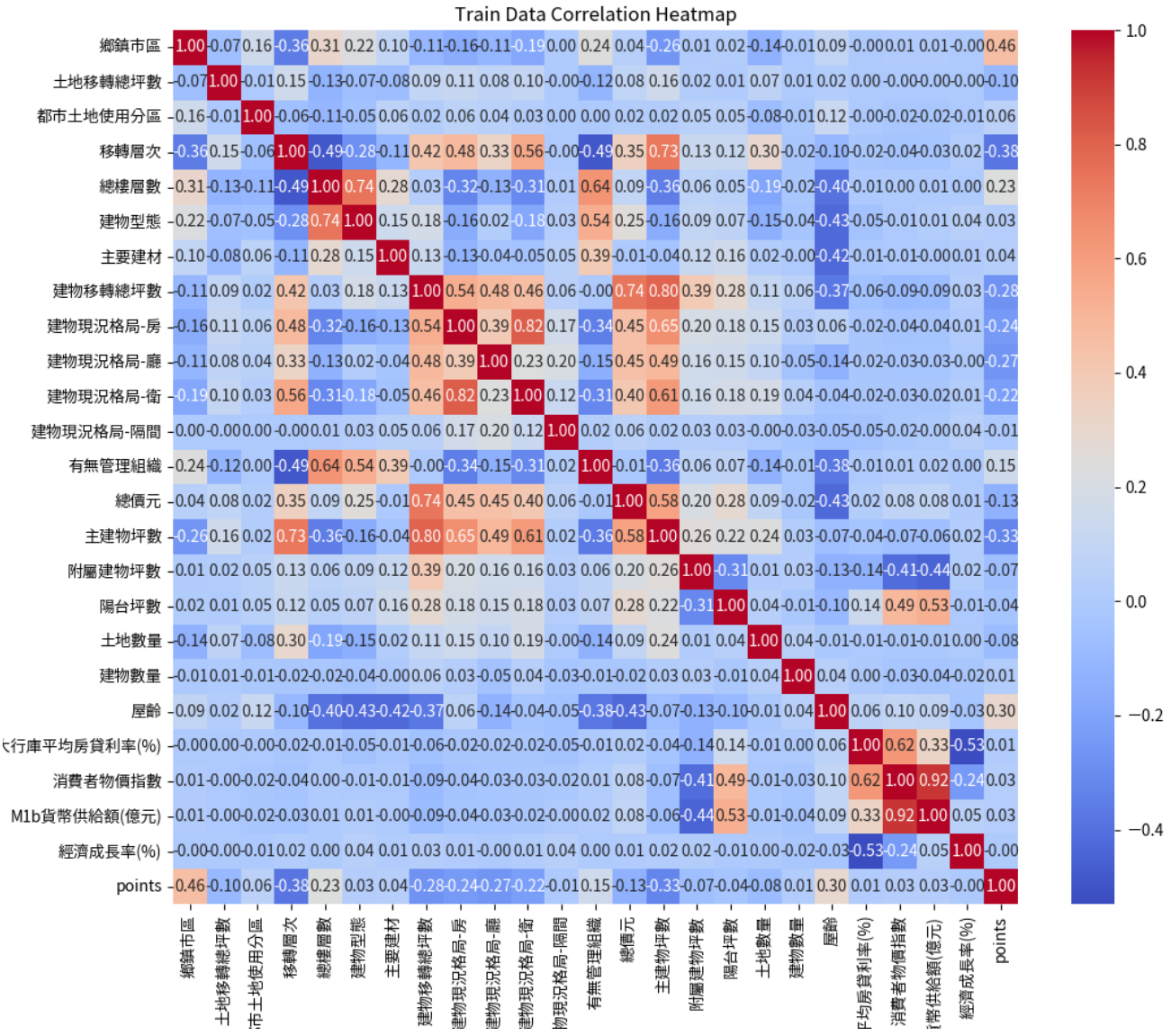


圖 21. 訓練集資料 (2020-2022) 欄位相關係數熱度圖

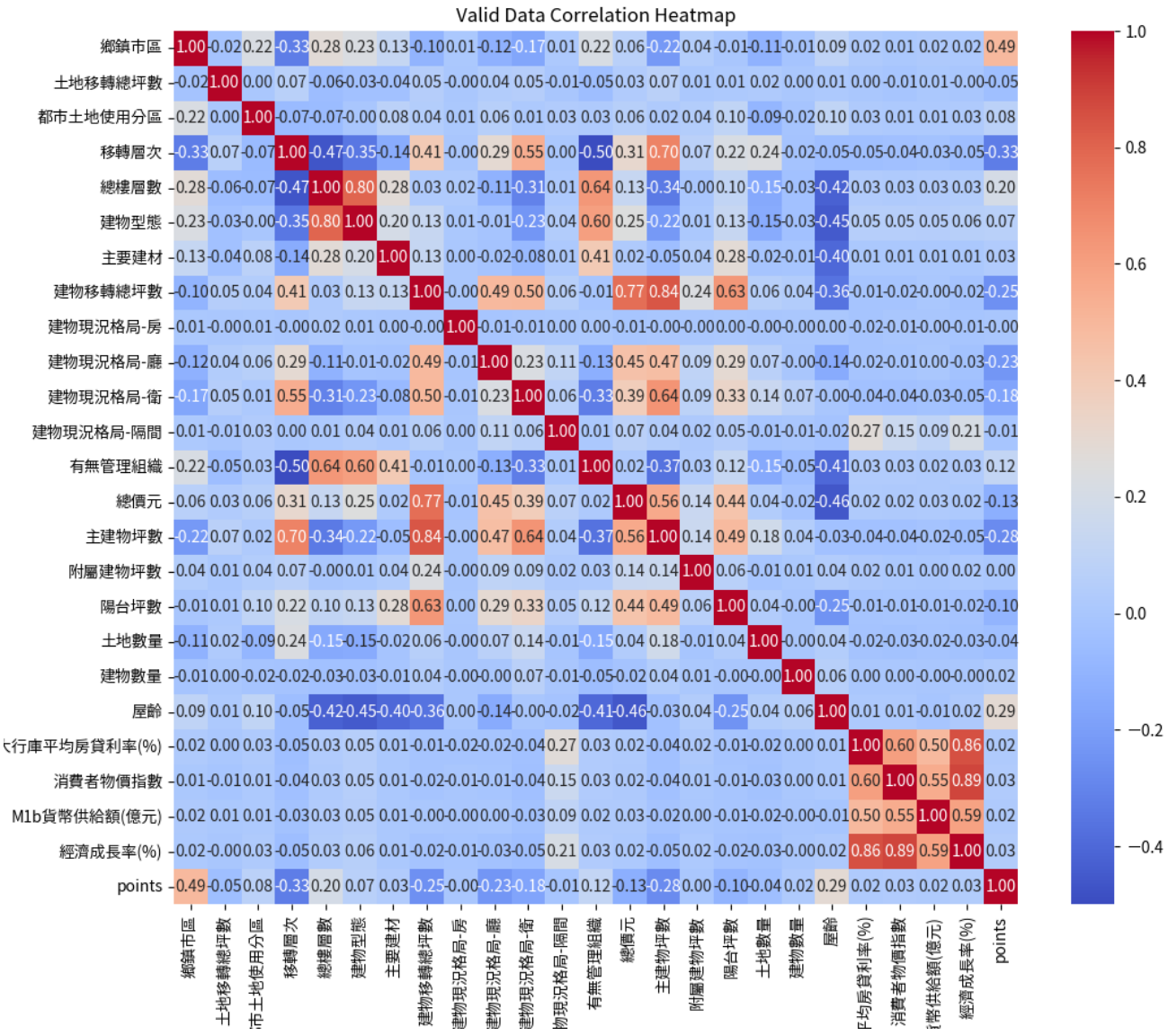


圖 22. 驗證集資料 (2023) 欄位相關係數熱度圖

參考文獻

中英文文獻

張建發 (2019)。影響房價關鍵因素之探討。國立臺北科技大學管理學院工業工程與管理 EMBA 專班碩士學位論文。

邱國祥 (2020)。以多元線性迴歸與機器學習膜性預估不動攢價格-以台中市實價登錄為例。國立中興大學應用數學系碩士學位論文。

陳玟寧 (2022)。迴歸機器學習應用於房價預測-以台北市實價登錄為例。明志科技大學工業工程管理系碩士班碩士論文。

廖思閔 (2023)。應用機器學習於預測桃園市房價。元智大學工業工程與管理研究所碩士論文。

王尹暘 (2023)。以決策樹預測台南世紀之門房價。國立成功大學土木工學系碩士論文。

Rana ORTAC-KABAOGLU(2011)。A SUPPORT VECTOR REGRESSION METHOD FOR REDUCING THE HIGH-ORDER SYSTEMS TO FIRST-ORDER PLUS TIME-DELAY FORMS。Istanbul University Electrical-Electronics Engineering。

網站

Coursera, 機器學習基石, 林軒田教授

Coursera, 機器學習技法, 林軒田教授

Medium, 機器學習 _ 學習筆記系列, 劉智皓