**國立空中大學**

**「Python 程式設計與實務應用」**

**期末小組專題**

**「房價預測器」**

**應用程式開發手冊**

導師：賴素純

班級：ZZZ002

組別：第11組

組名：8089

組員（tronclass平台組員順序）：

112222911詹秉蒼、112224868蔡惠婷、112224871蔡婷羽、113122776張心齡、113122877陳銘泓、113170973余誼姍

1. **專題主題**

房價預測器

1. **專題簡介**

* 本專案開發之軟體，透過爬取內政部實價登錄資料，用最真實的成交資料作為依據，以統計學方法科學地預測房價行情。
* 在現今臺灣房價飛漲，房價所得比不斷上升的年代，購屋已不易，又若購屋前沒有做足功課及進行相關統計分析，不熟悉欲購置區域的行情，就容易受騙，以高於行情的價格成交。
* 購屋前的資料蒐集及統計分析費時又費力，而透過本軟體，購屋者可藉由篩選特定條件的房屋交易資料，並輸入欲購置的時間及物件坪數等相關資料，就可以快速得到欲購買標的之預測房價。有了房屋預測價格，購屋時便不易遭屋主或房仲之話術欺騙，導致買貴。
* 本軟體可減輕購屋者購屋前資料收集分析的負擔，增加決策正確性及速度，並提供議價依據（議價籌碼）。

1. **小組分工**

提案發想：蔡婷羽（112224871）

需求規格化：蔡婷羽（112224871）

專案統籌規劃：詹秉蒼（112222911）

需求分析：全體組員

程式撰寫：

* GUI介面、數據驗證：陳銘泓（113122877）
* 資料表設計與優化：詹秉蒼（112222911）
* 交易資料下載：詹秉蒼（112222911）
* 交易資料檔彙整與分類：余誼姍（113170973）、張心齡（113122776）
* 資料庫查詢語句與條件產出：蔡惠婷（112224868）
* 預測模型與演算法：蔡婷羽（112224871）
* 整合程式碼：詹秉蒼（112222911）

開發文件製作：詹秉蒼（112222911）、蔡婷羽（112224871）、陳銘泓（113122877）

使用手冊製作：余誼姍（113170973）

執行檔編譯（Windows）：陳銘泓（113122877）

執行檔編譯（Mac OS）：待辦

整合測試：全體組員

專題報告：張心齡（113122776）

報告投影片製作：張心齡（113122776）

宣傳影片/海報製作：余誼姍（113170973）

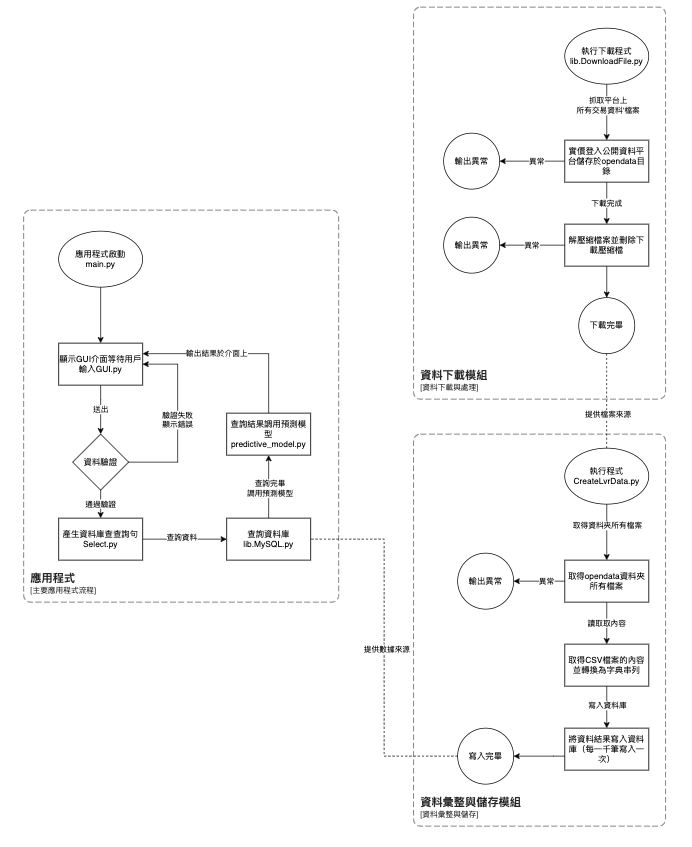
1. **開發環境**

* 語言：Python 3.12.7
* 依賴套件：
  + 檔案下載：requests
  + 檔案解壓縮：zipfile
  + CSV檔案處理：os、sys、csv、datetime、collections、json
  + 資料庫操作：mysql.connector
  + GUI介面與資料驗證：tkinter、customtkinter、functools. Partial
  + 預測模型：pandas、numpy
* 開發系統：Windows、MacOS
* 流程圖：Draw.io
* 資料庫：MySQL 8.1.30
* 版本控制：Git、GitHub
* IDE：Microsoft Visual Studio Code

1. **系統開發流程**

* 需求分析

由全體組員進行討論，對於提案內容進行流程圖繪製、功能分解、使用套件、分配開發人員、制定函數、類別接口與參數相關工作分派。

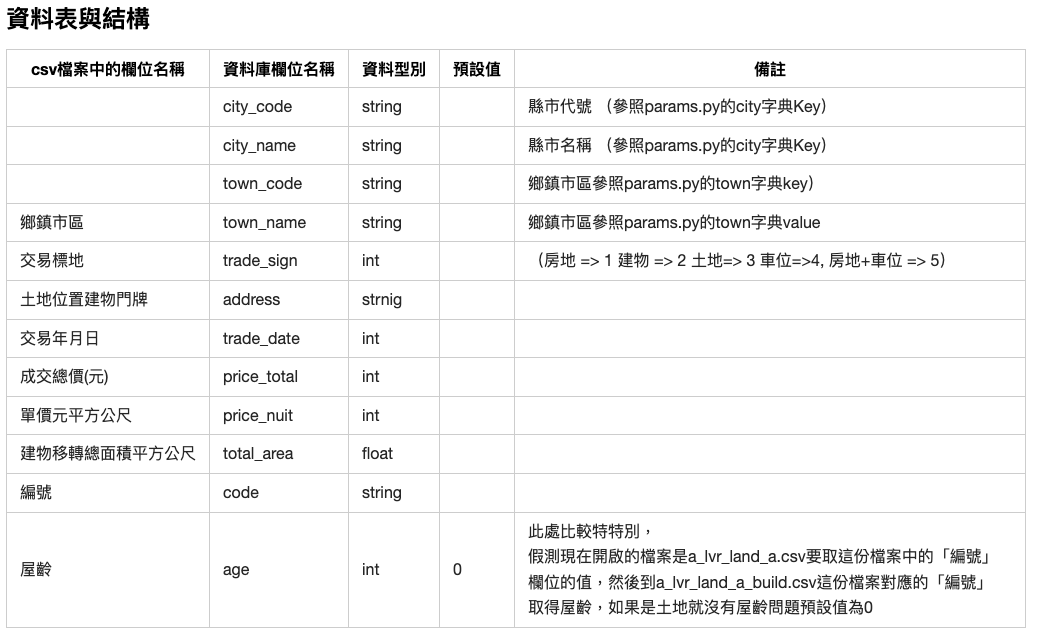


* 定義功能的輸出、輸入與邏輯規則

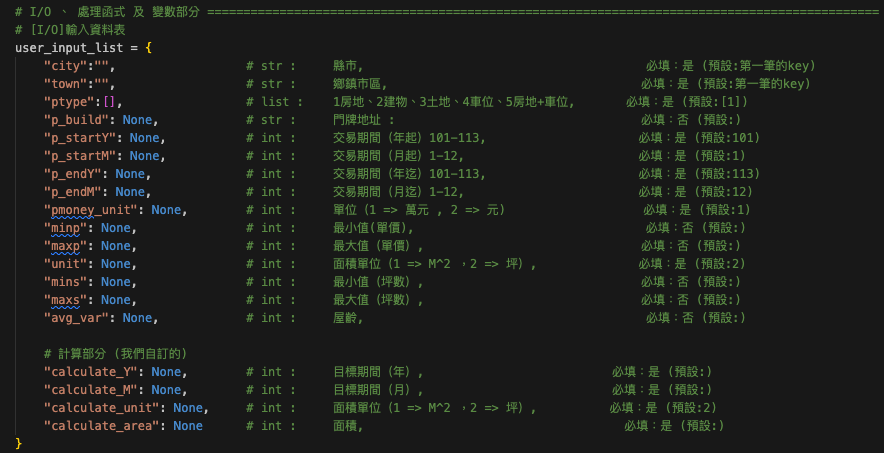
資料下載邏輯



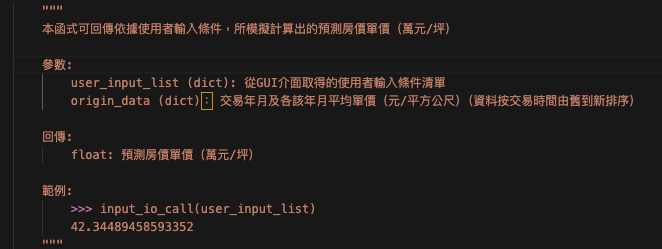
CSV資料轉換為資料庫格式與注意事項



GUI介面提供的輸出內容



預測模型的輸入與輸出



* 開發項目分工與制定完成日期

於每次會議討論工作項目、分配工作並追蹤進度

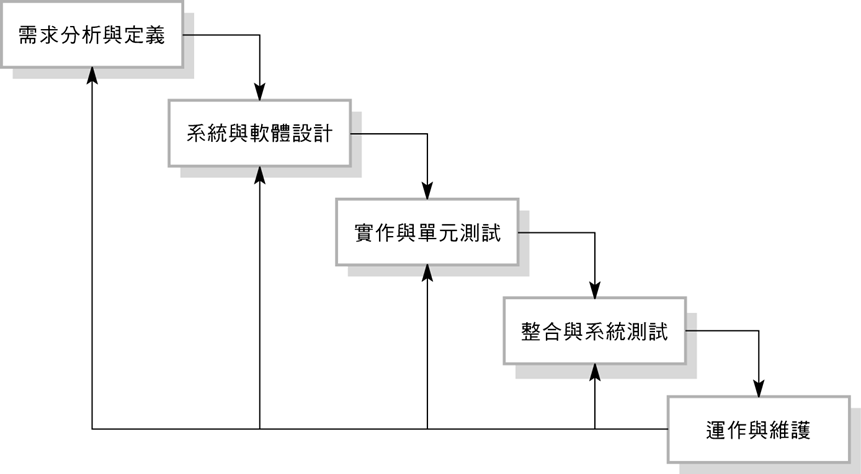
|  |  |
| --- | --- |
| 會議時間 | 討論內容 |
| 113年11月5日  19：00～20：00 | 1. 流程圖解說  2. 討論分工  3. 討論資料型態  4. 相關知識教學（網際網路、requests、API等概念）  5. 初步擬定作業時程 |
| 113年11月19日  21：00～22：20 | 1. 運行一遍程式的運作結果（由組長看完後合併到develop分支，由組長統一展示）  2. 各程式間的串接輸出與輸入的分析（所有屬性是否都存在、型別是否正常）  3. 指派程式串接事務，以及完成時間（１人，串接有問題，原開發者協助除錯與調整）  4. 著手進行文件、報告相關事宜（共同討論如何分工）  5. 作業時程規劃 |
|  |  |

* 時程圖（甘特圖）

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 時間  任務名稱 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
| 題目發想、提案報告撰寫、需求分析 |  |  |  |  |  |  |  |  |  |  |  |  |
| 需求規格化、程式碼撰寫、開發 |  |  |  |  |  |  |  |  |  |  |  |  |
| 測試（單元測試、整合測試、跨平台測試） |  |  |  |  |  |  |  |  |  |  |  |  |
| 報告投影片製作 |  |  |  |  |  |  |  |  |  |  |  |  |
| 面授報告 |  |  |  |  |  |  |  |  |  |  |  |  |
| 開發文件、使用手冊、宣傳影片或海報製作 |  |  |  |  |  |  |  |  |  |  |  |  |

註：1表示專案進行第一週（10/13～10/19）；2表示專案進行第二週（10/20～10/26）；以下以此類推。

* 軟體開發程序
  + 軟體開發的程序有四種：瀑布式、漸進式、需求規格化、組合式的軟體開發程序。
  + 本專案採傳統階梯式的軟體開發程序，亦即瀑布式的軟體開發程序，下圖為階梯式的軟體開發模型，也就是Waterfall model。
  + 此軟體開發程序，講求軟體生命週期（Software life cycle），將軟體發展劃分為明確階段。此開發模式為標準的程序，為大多數人所接受，有利於軟體專案的管理。
  + 由前揭甘特圖可見得，本專案開發程序中，有明確的階段。在經過系統需求分析後，進行需求規格化，接著進行系統開發實作，實作的過程中，亦同時進行單元測試，實作完畢後，進行整合測試，故屬瀑布式的軟體開發程序。

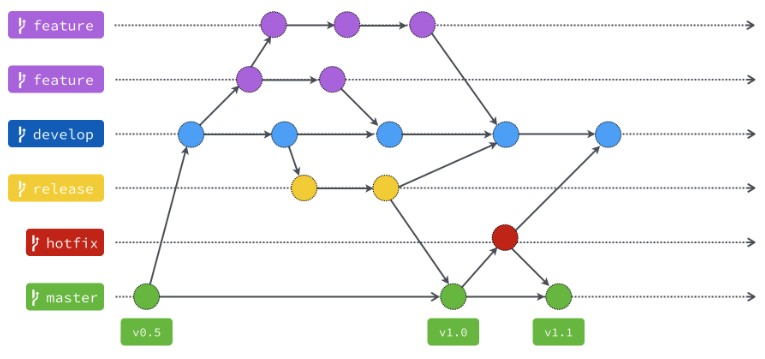


(圖片來源：<https://wayne265265.pixnet.net/blog/post/113080214>)

* 版本控制協作流程

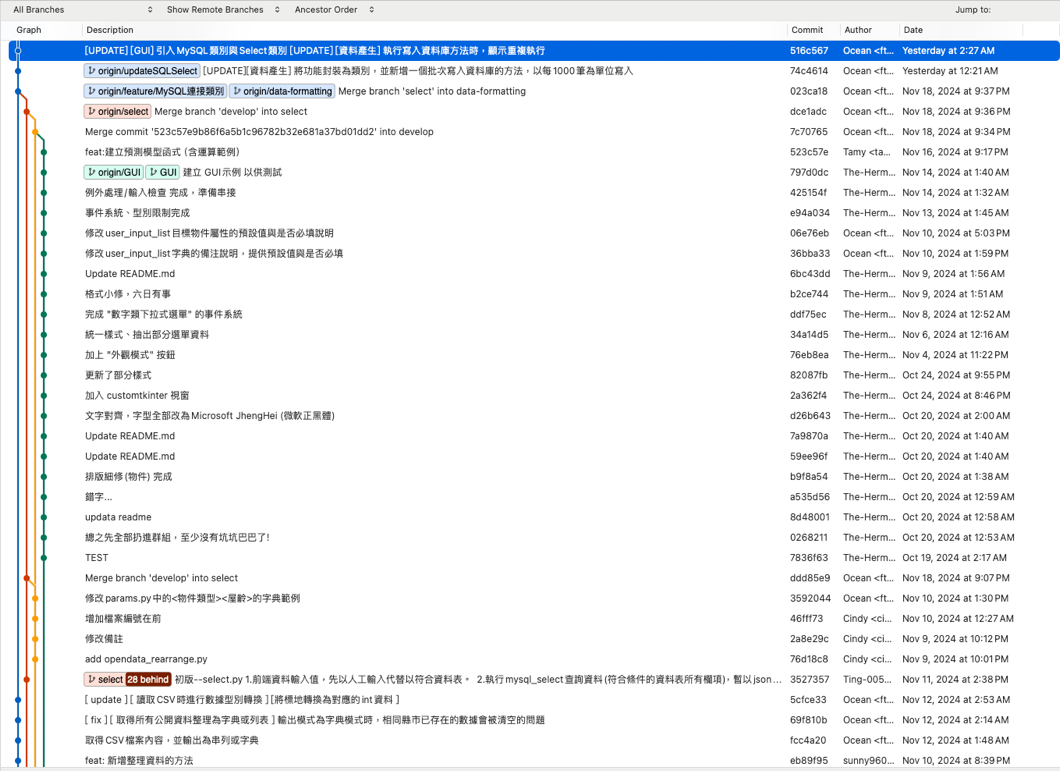
採用現今最流行的GitFlow流程，將Branch（分支）狀態系統化。

組員開發各自所負責之功能時，於各自之feature分支進行開發，功能開發完畢後，合併至develop分支，待功能測試完畢，整體程式可正常執行，即將develop分支合併至main分支（master分支），即成為穩定上線版本。



（圖片來源：<https://gitbook.tw/chapters/gitflow/why-need-git-flow>）

下圖為本專案版本控制協作流程歷程節錄：



* 開發挑戰問題解決

**問題一**：實價登入網站的版本更新，無法獲取資料來源。

解決方式：使用爬蟲抓取「[不動產成交案件實際資訊資料供應系統](https://plvr.land.moi.gov.tw/Index" \t "_blank)」中的公開資料，來解決無法即時獲取交易資料問題，作為解決方案。

**問題二**：下載之數據量高達460萬餘筆，使用資料庫儲存時，遭遇效能問題。

解決方式：透過資料庫的分析語句工具對相關的查詢語句，進行索引的優化與調教。

* 整合測試

由組員各自在本機上進行完整的測試。

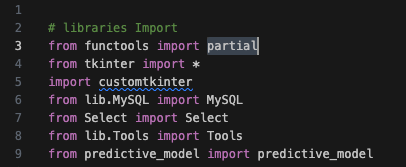
* 問題修正

測試階段遇到問題反映給原作者，由原作者完成修正後，再進行測試。

1. **程式設計主要技巧**

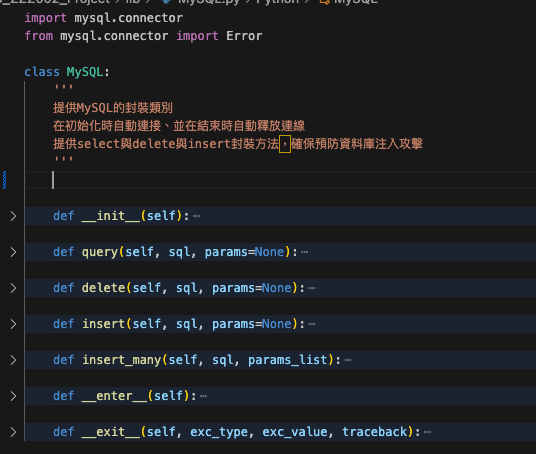
* 導入模組

導入Python原生、第三方模組加速建構，以及組員開發之各實作程式，使開發工作可分工各自進行，不互相干擾。



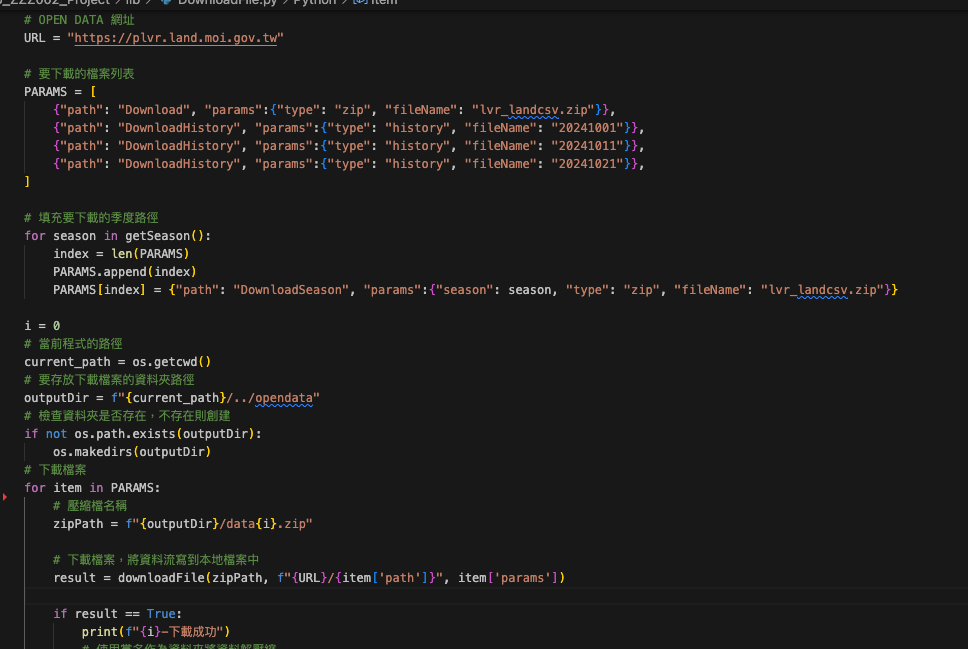
* 類別方法封裝

下圖將資料庫的新增（C）、查詢（R）、刪除（D）、更新（U）各方法封裝為類別



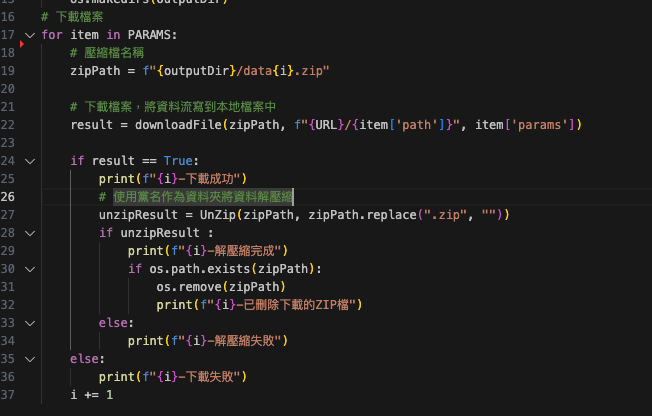
* HTTP訪問

使用requests套件，下載「[不動產成交案件實際資訊資料供應系統](https://plvr.land.moi.gov.tw/Index" \t "_blank)」中的交易資料。下圖為先產生所有欲下載的連結，透過http GET方法將所有公開檔案下載到本地端。

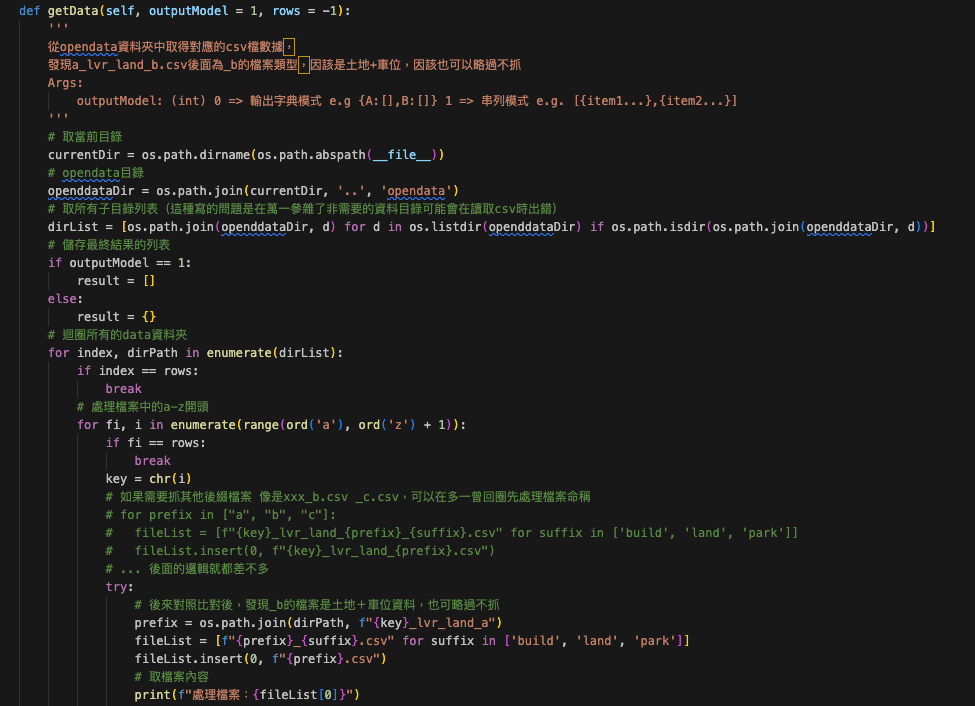


* 檔案解壓縮與解析

將下載的檔案（壓縮檔進行解壓縮）完成後刪除，避免佔用過多空間。



使用CSV套件以取得檔案內容，並彙整成資料庫所需格式。



* 資料庫批次寫入

因搜集到的數據量超過百萬筆（約460萬餘筆），若一次性寫入資料庫，將造成大量數據庫開銷，故採用每一千筆批次寫入，使資料庫的IO達到較良好的效能。



* 圖形化使用者介面（Graphical User Interface, GUI）

透過圖形化使用者介面搜集之用戶對資料的篩選條件，以此為基礎搜尋資料庫中之對應資料，並生成預測結果。



* 數據驗證

提醒用戶必要輸入的數據，以及進行數據型態的控制與驗證。



1. **原始碼重點解說**

* 資料下載與解壓縮（lib.DownloadFile.py）：

主要目的用於下載自101年起至當下之實價登錄資料，並將其解壓縮後儲存於本地端。主要邏輯片段：

|  |
| --- |
| 函數主要產生101年到當年的年+季以[明國年] S [第幾季]為返回格式  透過迴圈產生出對應公開資料平台的下載點URL  以字典串列形式返回，字典中包含檔名、路徑、類型    將資料解壓縮後，再刪除壓縮檔以節省儲存空間 |
|  |

* 資料庫連接與CRUD操作（lib.MySQL.py）

主要用於封裝mysql.connector套件中的資料操作方法，方便重複調用，以及於每次操作資料庫完畢後，自動釋放連線資源。

|  |
| --- |
| 封裝CRUD Create 、Read、Update、Delete資料操作方法，並再提供一個針對批次寫入的方法    透過\_\_enter\_\_、\_\_exit\_\_在結束時自動釋放資料庫連線 |
|  |

* 寫入資料庫（CreateLvrData.py）

主要用於提取所有下載的CSV檔案中的內容，並整理為可寫入資料庫的數據型態，將資料儲存於MySQL資料庫中。

|  |
| --- |
| CreateLvrData.getData()方法  取得lib.DownloadFile.py下載的CSV檔案並從中提取數據，進行數據的資料轉換    CreateLvrData.getCsv()方法  對CSV中的數據進行預處理與轉換    CreateLvrData.insertSQL ()方法  將取得的數據以每批一千筆的的速率寫入資料庫 |

* GUI介面（資料搜集）

|  |
| --- |
|  |

* 數據驗證

|  |
| --- |
|  |

* 資料庫語句生成（Select.py）

主要用於接收GUI介面所搜集到的用戶輸入資料，並產生對應的資料庫查詢語句

|  |
| --- |
| Select.createQuery()方法  處理傳入的數據，產生對應的資料庫查詢語句，若數據沒有值或者為None時，將不會產生對應的欄位查詢語句 |

* 預測模型（predictive\_model.py）

用於接收透過資料庫查詢語句挑出的資料，並使用統計方法，運算出預測不動產單價後，將其回傳。

|  |
| --- |
| predictive\_model.predictive\_model()方法  本方法有兩個參數，第一個參數為由GUI介面取得之使用者輸入條件清單；第二個參數為透過資料庫查詢語句挑出的資料（交易年月及各該年月平均單價（元/平方公尺）（資料按交易時間由舊到新排序））。  引入Pandas及NumPy模組以進行數據分析    本方法將交易時間作為解釋變數（自變數），將不動產交易單價作為被解釋變數（依變數），透過統計學之「最小平方法」，求出簡單線性迴歸式，亦即不動產單價預測公式，再透過使用者輸入的欲購置時間，以產生對應該時點的預測不動產單價 |

* 結果輸出

|  |
| --- |
|  |