實習期間(2/13~6/30) 實習生:邱翰杰 專案負責人:邱翰杰 領航員:林清河

注意:由於新冠肺炎(COVID-19)影響4/13~6/12為在異地上班。我的實習天數為一週五天，有時會請假，但不多。連結的部分因為很多當時沒有特別記錄下來所以只列出關鍵詞的部分，並且有些連結為新光人壽有擋住的連結，我可以連到是因為在家異地的緣故或是下班後，晚上我花額外的時間自己研究。以下為我在房價預測專案的過程報告，所有建議的部分都該依妳接續狀況來做參考，這邊建議的假設是以重做一次此專案為目的的建議，請斟酌參考。

**房價預測專案過程報告**

一開始來新光的時候還不知道專案會做什麼，也不知道會做房價預測，其實還有做其他工作，而這邊題目為我的專案過程，所以我只打上與我的專案相關的工作項目與過程，且不會細講過程。

由於我所寫的程式是python，所以領航員幫我向公司借了一本python王者之路的書，基本上這一本我是每一章都有看，除了python tkinter套件我，那是UI相關的可以不用看(雖然我也練習了)，你說整本書(並不是每一張都要練習)，沒錯這需要花些時間，如果你認為你有較好的程式能力，我建議可以花五天的時間看完，看完一定還會忘記，不用擔心，因為已經看完了，所以你會知道有哪些方法，便知道如何GOOGLE。

(以下為書中重要套件或要素)

重要的章節要練習一下

os、Datetime、calendar、time、generator、lambda、filter()、map()、reduce()、matplotlib、numpy、pandas、基本資料結構與他的方法、物件導向等等

因為是人工智慧，所以一定是用tensorflow的keras，所以要了解[keras](" https://keras.io/api/models/model/)套件中的許多方法或物件，因為沒有書，所以我幾乎都是看官網document，(建議英文真的看不懂，再看中文，一開始看中文的話，你未來遲早後悔，因為英文會比較清楚，反而是被中文的翻譯給困住)，我會英文看完不懂的話在看中文，中文看不懂的話(這很常有的)，我會GOOGLE或到論壇(stackoverflow、CSDN、geeksforgeeks、github、kaggle等)查資料。

Keras要至少熟悉model的sequential，layer的Dense、dropout這會對你很有用

Loss function 、 activation function 、 optimizer

[Cross validation](#)(CV也是很重要的一環)、你也可以看看callback function

另外，[sklearn](" https://scikit-learn.org/stable/)是資料分系套件(scikit套件有點多，因為他是science kit，幾乎所有有關數據分析的方法都有)，這個基本上是看不完的(看完的話，學習時數會是幾百小時以上)，這邊只是告訴你我很多幾乎都是從他們這邊查詢資料，因為他的架構很好且完整，順帶一提，可以找youtuber StatQuest with John Starmer，他講了許多資料科學的課程，與統計相關的課程。

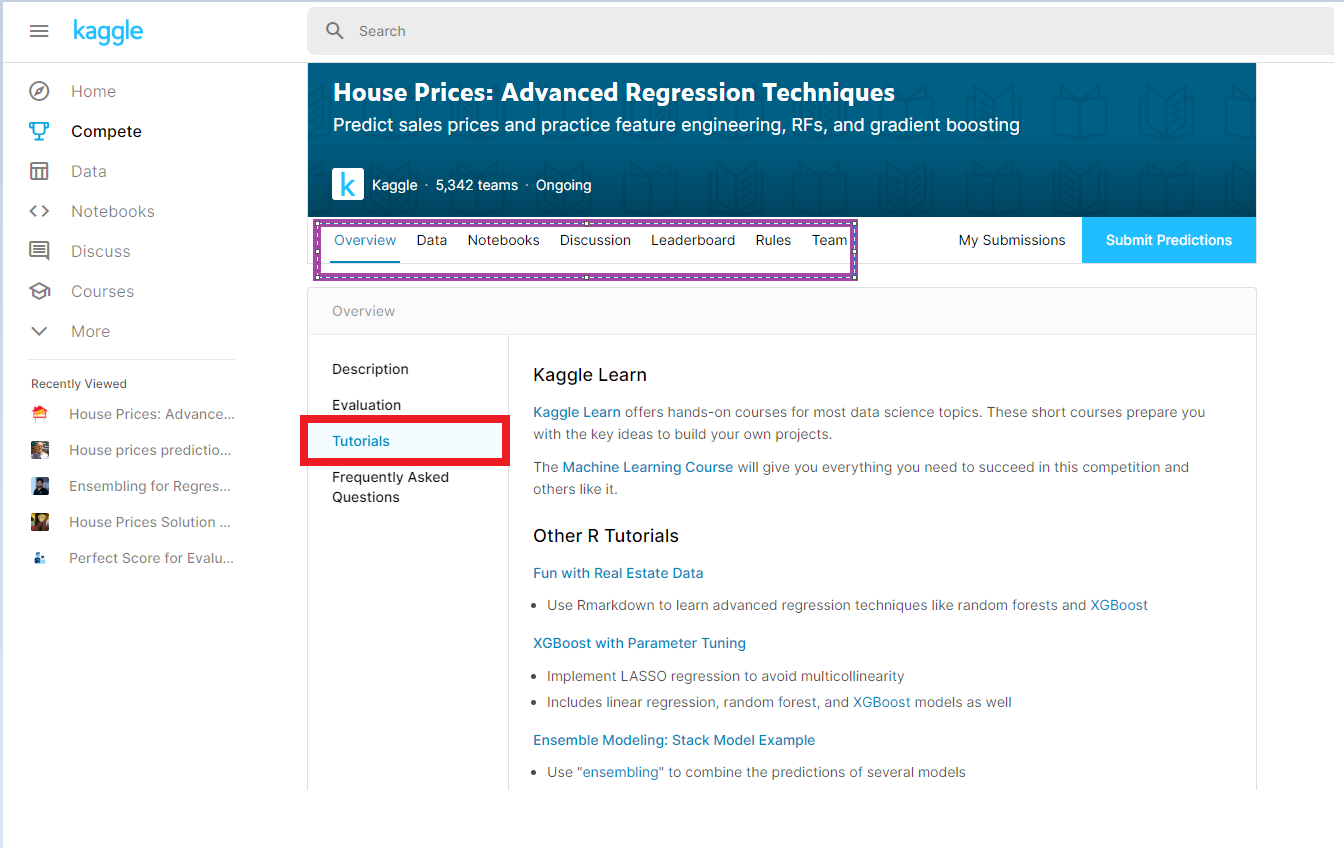
再來是演算法的部分，也就是人工智慧的演算法，因為人工智慧是近年來計算能力的突破導致其快速發展，基本上它與資料探勘是不同的，不過他們都包含在資料科學裡，建議還是把他們都搞清楚。常見四種學習的方法(監督、非監督、半監督、強化)，與一些常見的方法，由於我的專案偏向是監督是學習方法，所以我看了很多方法，決策樹、集成（baging，boosting，random forest）k-NN線性回歸Bayes神經網路邏輯回歸感知器支援向量機（SVM）相關向量機（RVM）等，卷積類神經網絡(CNN)、RNN、LSTM、auto encoder、(人工智能我有查看的書籍有*TensorFlow+Keras深度學習人工智慧實務應用*這本書是別人給我的、*人工智慧與深度學習：理論與Python實踐*剛好這本書的作者是我的專題老師，也剛好我有旁聽他的人工智能的課，這也剛好也是在實習之前。不過我沒什麼問他，因為基本上還是上網查資料比較快與看書，因為這樣學得會比較快，除非狀況處於死胡同裡，我才會去問)看了這些之後，都有助於我理解、表達與傳達我所做的專案的內容與整體框架。Youtuber NTU李宏毅的教學影片、周莫煩教學影片

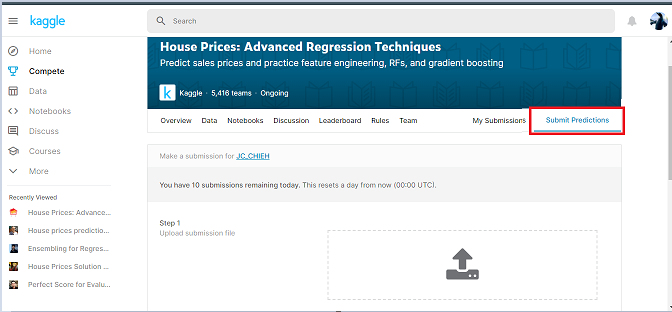
另外，有些在查找資料時會用到數學，所以你像我一樣都還給老師的話，可以回去看大學的書籍(如果你賣掉了或是沒有，可以回學校藉相關書籍，數學的教科書基本上不分新舊版都沒什麼變)，因為數學本來就是工具。例如:優化器、loss function所用的更新方法backpropagation，ElasticNet、LASSO、RIDGE、kernal function等一大堆都會用到微積分的概念，就連在做數據分析時會用到一些統計的數學，而統計的數學也會需要大量的微積分的數學，當然你可以都背起來，沒問題，但這記的不長，就跟打程式一樣，背程式沒有意義，多用幾次、多寫幾次與多思考幾次程式，久了因為理解的過程中，就賦予了結果意義，自然而然就會背起結果，儘管以後可能會忘記怎麼推倒的但還是會記得就像最簡單的九九乘法表，我只記得結果並使用，而不會再一一去推(舉這個例子是因為大家都還知道怎麼推)。也可以看Youtube CUSTCourse李柏堅的微積分與統計的相關課程或是Coursera、MOOC的免費課程。

資料處理與分析的方面要知道如何篩選的語法，這邊熟悉也要需要一點時間，我常常忘記，drop()、fillna()、astype()、reset\_index()、preprocessing.StanstardScaler()、model\_selection.train\_test\_split()、np.log()、np.logp1()、pd.get\_dummies()、labelEncoder、oneHotEncoder()(這個也可以稍微了解一下用法)、[boxcox1P](" https://baike.baidu.com/item/Box-Cox%E5%8F%98%E6%8D%A2/10278422)(這個也可以稍微了解一下用法)、np.expm1()，還有資料格式的轉換、接入訓練模型的輸入與輸出的資料格式(EX:傳入的模型的shape是怎樣)。

繪圖的話，請熟悉[matplotlib](#)(也可以看新光的書)與[seaborn](" https://seaborn.pydata.org/)套件(資源看官方document或是別人的CODE)，這兩個幾乎都是必要知道的，因為很常用到，熟悉他們幫助我很多在分析資料上面。

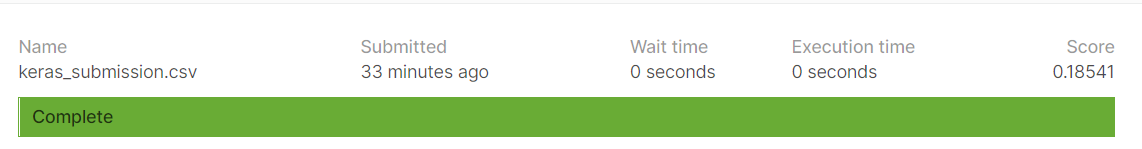
[Kaggle房價預測](#)

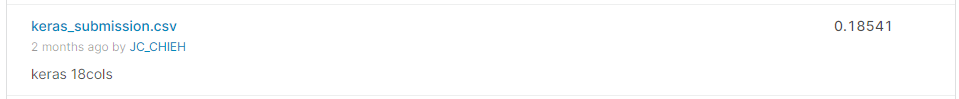
紅框，它裡面也會有有關房價預測的教學。紫框起來的欄位為所有要查詢的基本資訊，幾乎都可以在這裡找到，只是都是英文。



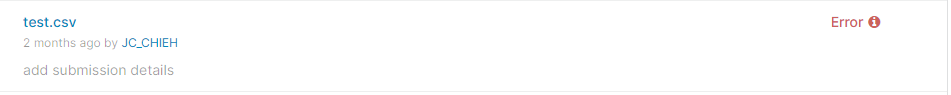
上傳檔案。

上傳成功、你也可以增加註解，如下圖”Keras 18 columns”:





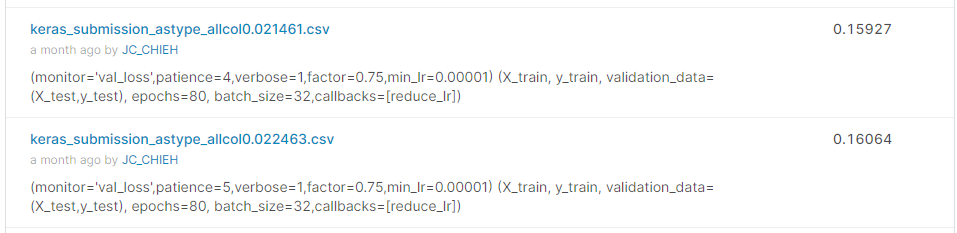
上船失敗、格式不對:

****

上圖成功的keras 18 columns在當時競賽名次(Leaderboard)為第3508名，現在的話為的4254名(2020/6/18)。以這個成績為例，你可以發現名次會隨著時間往後掉，是因為計分上面，Kaggle允許每個人繳交多次成績，並且她會拿取最好的成績與Kaggle上有比賽的成員們進行比較，所以通常大家都會繳交不同版本或是有修改的預測結果，所以名次會隨著時間而漸漸往後掉是正常的。順帶一提，當時我做出擁有最好的loss 值(kaggle上寫的是Score)是0.14459，在當時好像是2900多名左右的樣子，現在也掉到2955明了。

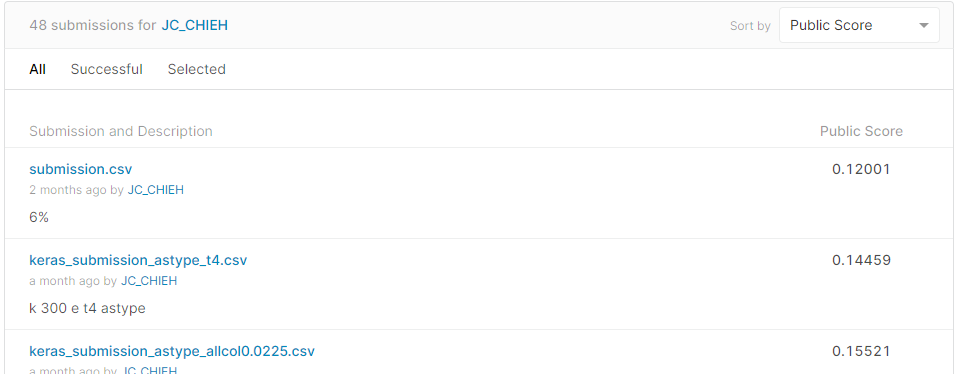
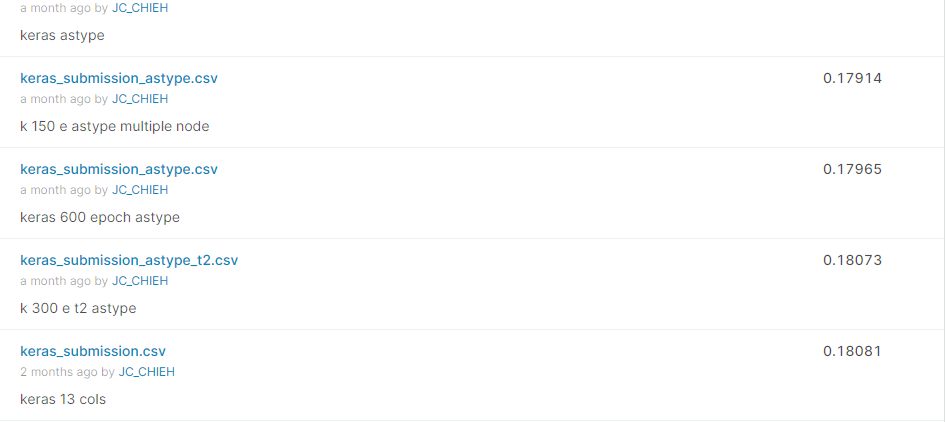
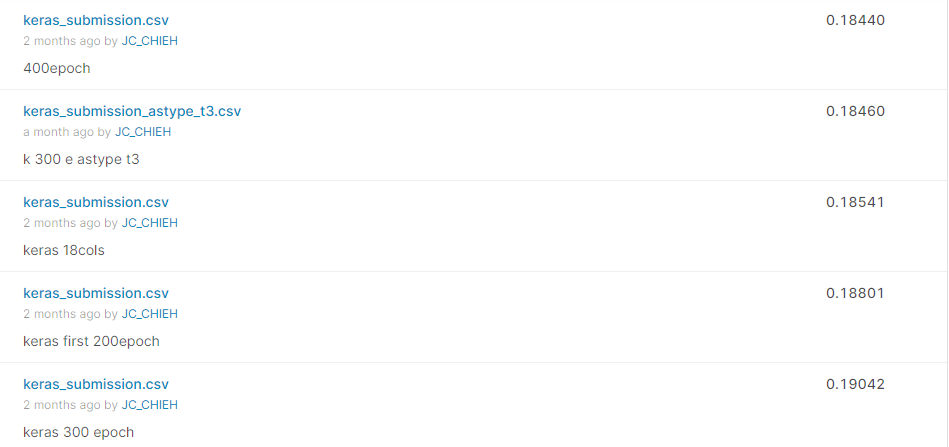
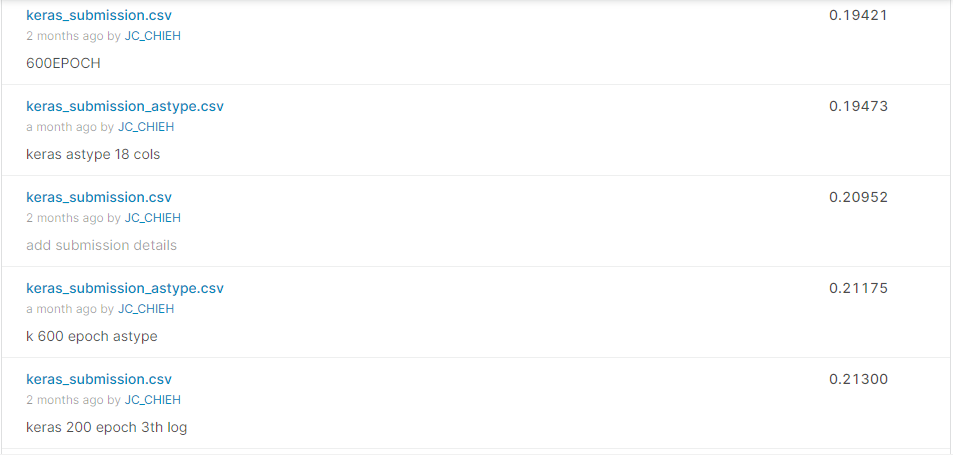
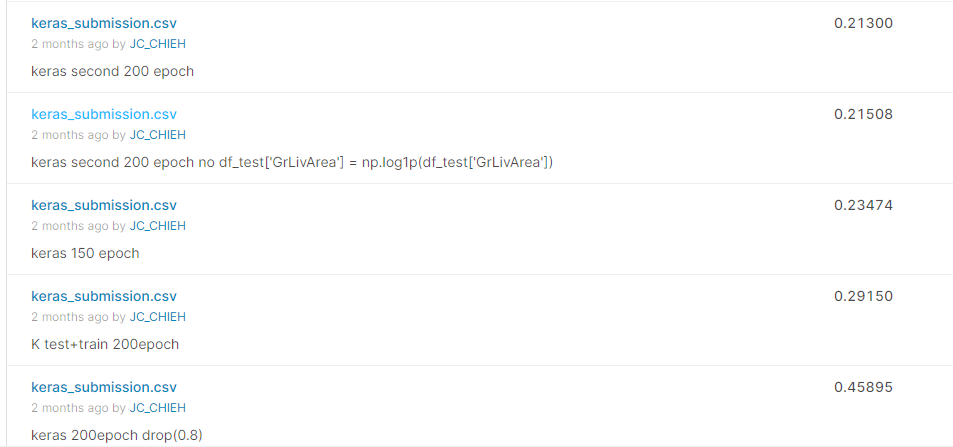
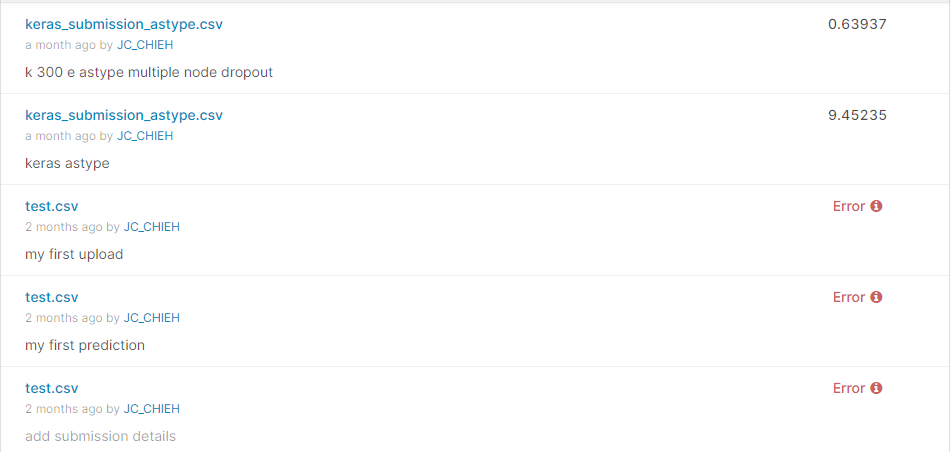
其中也用了不少方法去跑，不過kaggle他是不會讓你無上限的一直繳交檔案的，如果你在一段時間內一直平凡的繳交檔案，他會有一個防止濫用的機制，必須等一段時間後才能再繳交。為此，我幾乎都是先以圖和loss值，先在自己電腦先透過觀看LOSS值和圖形(基本都看loss值)，如果不錯的話才繳交(不是擁有最低的loss值就會好喔，因為她可能會有參數初始化的問題與實際正確答案比對準確率的問題)。所以模型我會訓練模型完後預測，再訓練一次再預測(這個動作可以多做幾次，但是預測出來的CSV檔要換檔名)，所以便會有兩個相同的建模方式的模型，但不同預測結果答案的預測結果，這是由於初始化權重的不同。

以下圖為例，他們建模方式都是一樣的，但結果不同。



最後我在這邊擺上我有預測並且最後有上傳的所有結果，妳可以看一下我繳交檔案的註解，可以透過註解來做標記以管理(不過這邊都是寫給我自己看的)。(注意:有可能因為我預測的CSV裡面可能因為它的loss值不好看，而我沒有上傳，但實際的Score卻是好的情況 ，因為kaggle用來評分的正確答案可能剛好會相對反而準確些的情況發生。所以如果時間夠多的話建議還是把每一次預測出來的結果上傳，但是loss值真的太差的話就不用了。)

圖為以public score來排序，並非以繳交時間來排序，0.12001是參考別人(前6%分數的人)的程式碼，用他建模的方式，來訓練模型，透過訓練完的模型做預測結果後繳交的CSV，建議不要這樣，因為這樣每個人都去看前1%排名人的程式碼(他們的KERNAL)，沒有在建模的方式上或是或是架構上有甚麼改變的話，那根本毫無意義，就失去了比賽的樂趣了。

參考別人的競賽成果網址或是數據分析:

[連結一](#) ([連結二\_keras](" https://www.kaggle.com/hugosjoberg/house-prices-prediction-using-keras) [連結三\_6%](#) [連結四\_1%](#) 這幾個都在kaggle的討論區(Discussion)可以找到，這邊只是把我點過的參考(學習)連結放到這，kaggle請先登入帳號並以Chrome載入連結)，還有清河(我的領航員)有先給我一份練習的房價預測的專案(這個專案是kaggle裡房價預測別的版本的)，可以先練習那份，練習完在到kaggle房價預測的競賽中練習，因為我就是這樣的。

Python其他連結

[pandas相關教學](#) [numpy教學](#) [googleColab](#recent=true)

其他注意事項:

因為是用KERAS跑，我有用到GPU，如果要用GPU跑的話，記得要安裝相關套件，與查看你是屬於哪一種類型的GPU，我是用[cuda GPU](#)，記得到官網查看自己的GPU是該用哪一種版本的驅動應用程式，不要下載錯了。如果要用CPU跑的話可能會跑很久，且也傷電腦，因為我是用我自己的電腦跑的，你也可以用google colab，就不用擔心沒有GPU的問題了，不過筆電通常都會有的。

另外可以多看看討論區，許多人提出的問題都很有意思，或是有些人分享他們自己的程式碼。

這個專案有第二部分是拿新光公司內部放款資料做預測，不過沒有做出來，詳細可以看房價預測碰到問題報告，裡面詳細說明原因。

步驟參考:

資料的資料結構處理、資料部分遺失值處理、前處理資料->建立相關性陣列->選取標籤與特徵->打散標籤與特徵作為交叉驗證資料->建模->訓練模型->繪製訓練後的LOSS結果->運用模型並預測

再次提醒所以上為我在房價預測專案的過程報告，所有建議的部分都該依妳接續狀況來做參考，這邊建議的假設是以重做一次此專案為目的的建議，請斟酌參考。