

一、動機

現實生活中，大多數的資料雜亂無章，需要經過長時間的預處理，其中一個常見問題便是資料的遺失。遺失原因有很多可能，比如人為疏漏、蒐集過程發生意外等因素，但對於資料缺失的處理，尚須考量資料中隱含資訊的重要性、資料之間是否會互相影響、缺失資料占比大小等問題，若逕行刪除缺失值可能導致我們遺失有用的資訊，那麼，若在無法刪除資料的狀況下，又有什麼方式可以填補遺失的資料？

彌補缺失值的方式主要為刪除或插補，其中，最簡單的方式為使用平均數或中位數填補缺失值，但這樣的方法未必適用於所有資料型態，尤其現在資料類型多元且龐大，插補遺失值尚須根據資料型態有所不同，於類別型資料與連續型資料的填補也需分開處理，為此，過去有許多研究提出各式各樣的演算法處理不同型態的資料，過去幾年中，於填補缺失資料的方法上，以 knn 最受熱議，不論是在離散型的資料或連續型的資料，knn 都能用來填補缺失值，針對離散型資料可以用 knn classifier 進行填補，連續型資料則可用 knn regression 進行填補。於一些具有特殊意義的資料，相較於單純使用平均數與中位數彌補缺失值，knn 或許更為合適，此次針對 KNN 的原理進行一個探討之餘，同時將此方法應用於 Kaggle 競賽平台上的資料進行分析與討論。

二、目的

1. 了解 K Nearest Neighbors 演算法的原理
2. 了解 KNN 演算法於不同資料型態的使用方式
3. 將 KNN 演算法用於資料缺失值的填補並和平台上直接刪去資料的方式進行比較

三、K Nearest Neighbors 方法介紹

(一)、 概念

將缺失值當作預測值，給定 k 值作為需考量的鄰近資料點數量，計算預測值與周圍資料點的加權距離，依據最近的 k 個觀察值判斷其類別或計算預測值。常用的 KNN imputer 即以歐式距離矩陣尋找資料點周遭最近的 k 個樣本，利用最近的 k 個樣本的平均值填補其缺失值。

以下簡述以 k -means 作為歐式距離的範例說明，再簡述 KNN Imputer 內使用的加權歐式距離來說明此次資料分析彌補連續型資料缺失值的使用方法。

歐式距離

$$\text{Define } d(x_i, x'_i) = \sum_{j=1}^p (x_{ij} - x'_{ij})^2 = \|x_i - x'_i\|^2$$

$$\text{and } W(C) = \frac{1}{2} \sum_{k=1}^K \sum_{C(i)=k} \sum_{C(i')=k} d(x_i, x'_i)$$

Goal : find C such that $W(C)$ is minimize

上述 $d(x_i, x'_i)$ 即計算資料點之間的歐式距離，以下簡述的 K-means 與

Influence of the j th attribute x_{ij} ，其目標皆已找到能使整體距離加總最小的 C 值，將各資料點 i 分配至對應的 $C(i)$ ，以決定資料點應被歸類至哪一群。

K-means

計算每個資料點與群中心的距離來決定資料屬於哪一群，其中 m_k 為第 k 群的中心點， $\hat{m}_k = \bar{x}_k$

An iterative descent algorithm for solving

$$C^* = \min_C \sum_{k=1}^K N_k \sum_{C(i)=k} \|x_i - \bar{x}_k\|^2$$

can be obtain by noting that for any set of observation S

$$\bar{x}_S = \operatorname{argmin}_m \sum_{i \in S} \|x_i - m\|^2. \quad (\text{群內資料點與群中心點的距離平均})$$

Hence, we can obtain C^* by solving

$$\min_{C, \{m_k\}_1^K} \sum_{k=1}^K N_k \sum_{C(i)=k} \|x_i - m_k\|^2.$$

✚ 加權歐式距離

✎ Influence of the j th attribute x_{ij}

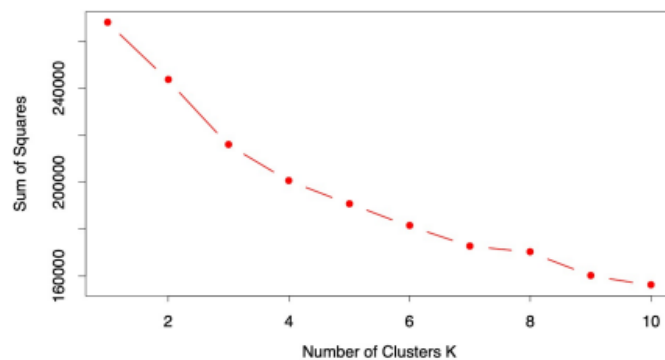
Define $d(x_i, x'_i) = \sum_{j=1}^p w_j \cdot (x_{ij} - x'_{ij})^2$, $\sum_{j=1}^p w_j = 1$ where $w_j =$

$\frac{\text{total number of coordinates}}{\text{number of present coordinates}}$

e.g. $x_i = (3, NA, 5)$, $x'_i = (1, 0, 0)$, $d(x_i, x'_i) = \sqrt{\frac{3}{2} \times \{(3 - 1)^2 + (5 - 0)^2\}}$

(二)、K 值的選取

設定不同的 k 值畫出 MSE，選擇 MSE 斜率變化的轉折點決定分群數量。



四、資料分析

(一)、資料預處理

此次資料分析以 kaggle 平台上西班牙紅酒的資料集作分析，使用資料集內的類別型與數值型的資料預測紅酒價格。

✚ 資料型態

✎ 類別型資料：winery、wine、region、type

✎ 數值型資料：year、rating、num_reviews、price、body、acidity

✚ 遺失值檢查

```
winery      0
wine        0
year        290
rating      0
num_reviews 0
region      0
price       0
type        545
body        1169
acidity     1169
dtype: int64
```

資料共 7500 筆，但類別型資料與數值型資料都有一定比例的缺失值，以下針對不同類型的變數做遺失值填補的說明。

✎ Year：將欄位中含有 N.V.的資料轉換為遺失值

✎ Type：由於此變數為類別型資料，若使用 KNN Imputer 進行遺失值的填

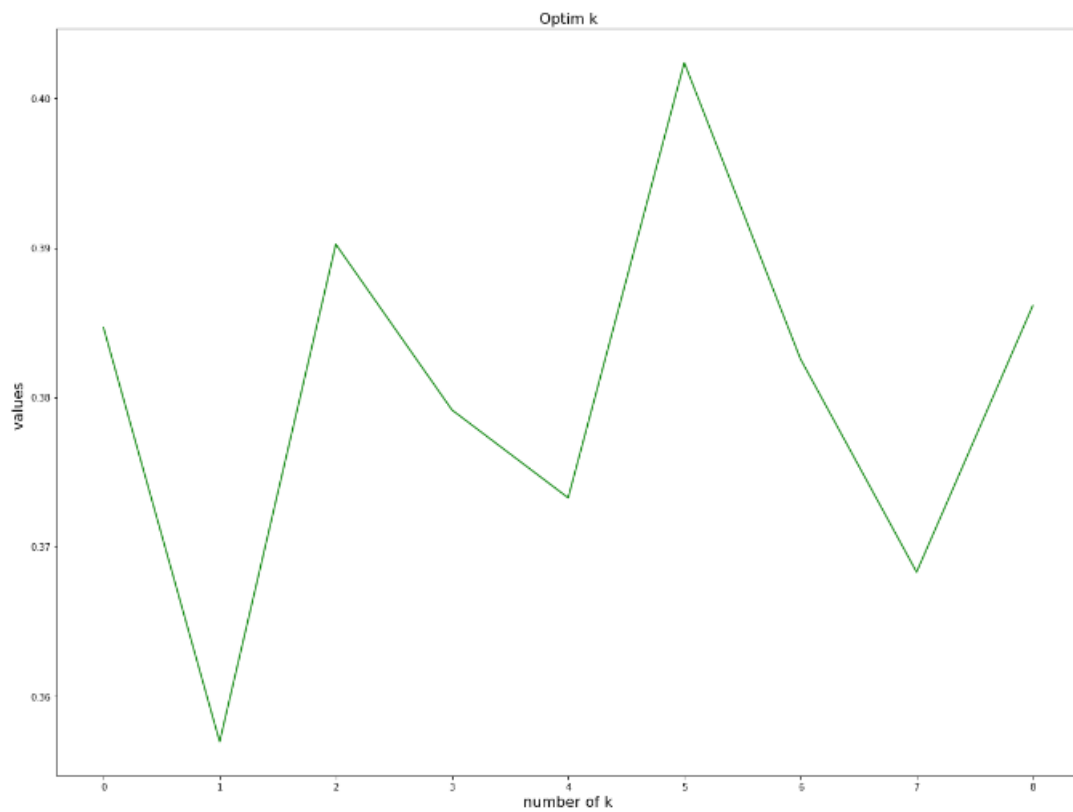
補需先做 Label Encoding 轉換為數值後方能進行填補。

✎ Year、body、acidity：數值型資料則是直接使用 KNN Imputer 進行填補。

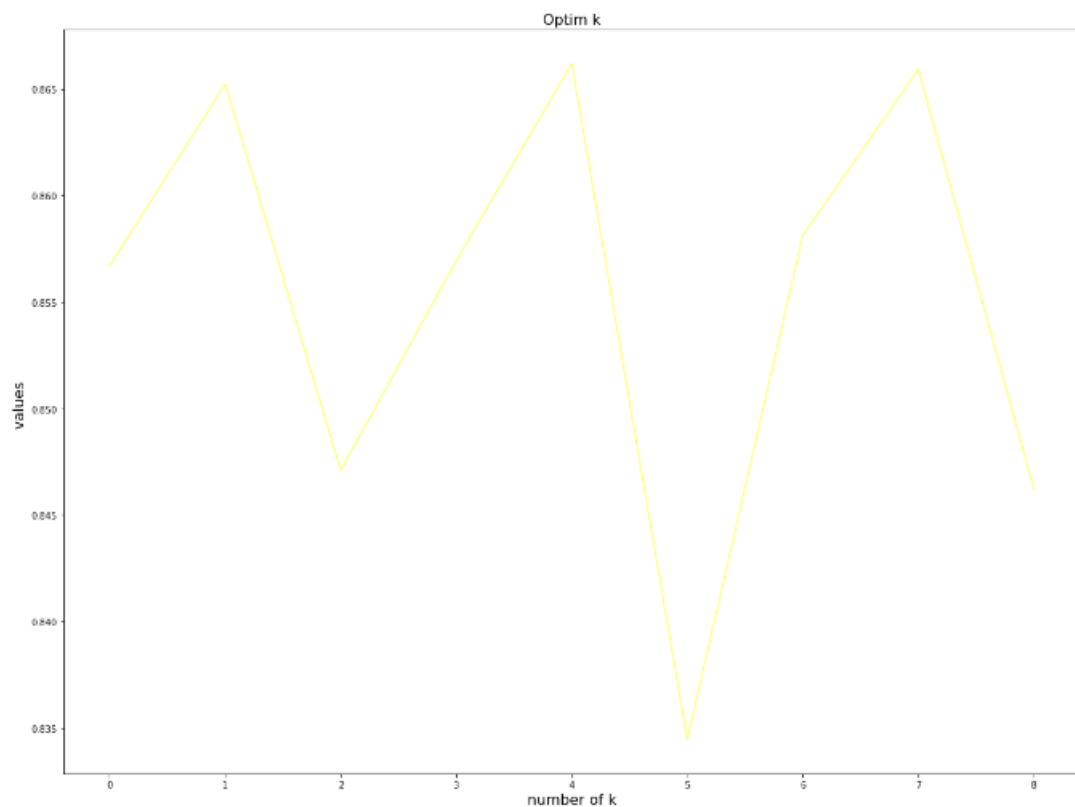
遺失值處理

將轉換完的的資料配飾 Knn Regression，計算 k=1~10 之間的 RMSE 與 R-squared，畫出 RMSE 與 R-squared 的摺線圖選擇合適的 k 值。

```
[{'K': 1, 'RMSE': 0.3846517466314483},
 {'K': 2, 'RMSE': 0.3569349728298786},
 {'K': 3, 'RMSE': 0.3902341987553622},
 {'K': 4, 'RMSE': 0.37914363978838134},
 {'K': 5, 'RMSE': 0.37326131256004086},
 {'K': 6, 'RMSE': 0.40239035711732424},
 {'K': 7, 'RMSE': 0.3825624366952271},
 {'K': 8, 'RMSE': 0.3682890751629718},
 {'K': 9, 'RMSE': 0.38612435908433923}]
```



```
[{'K': 1, 'R-squared': 0.8566937734477911},  
 {'K': 2, 'R-squared': 0.8652420334556328},  
 {'K': 3, 'R-squared': 0.8470956399826355},  
 {'K': 4, 'R-squared': 0.8569483588746638},  
 {'K': 5, 'R-squared': 0.866208282283017},  
 {'K': 6, 'R-squared': 0.8344488127784543},  
 {'K': 7, 'R-squared': 0.8581359296843647},  
 {'K': 8, 'R-squared': 0.8659300388596385},  
 {'K': 9, 'R-squared': 0.8461990540187627}]
```



根據以上結果，選擇 $k=3$ 來進行遺失值填補，並將轉換成數值的類別型資料

轉換回類別型，檢查是否有無法對應的部分，其中仍有 72 筆資料是無法被歸類的，由於佔筆極低，故予以刪除。

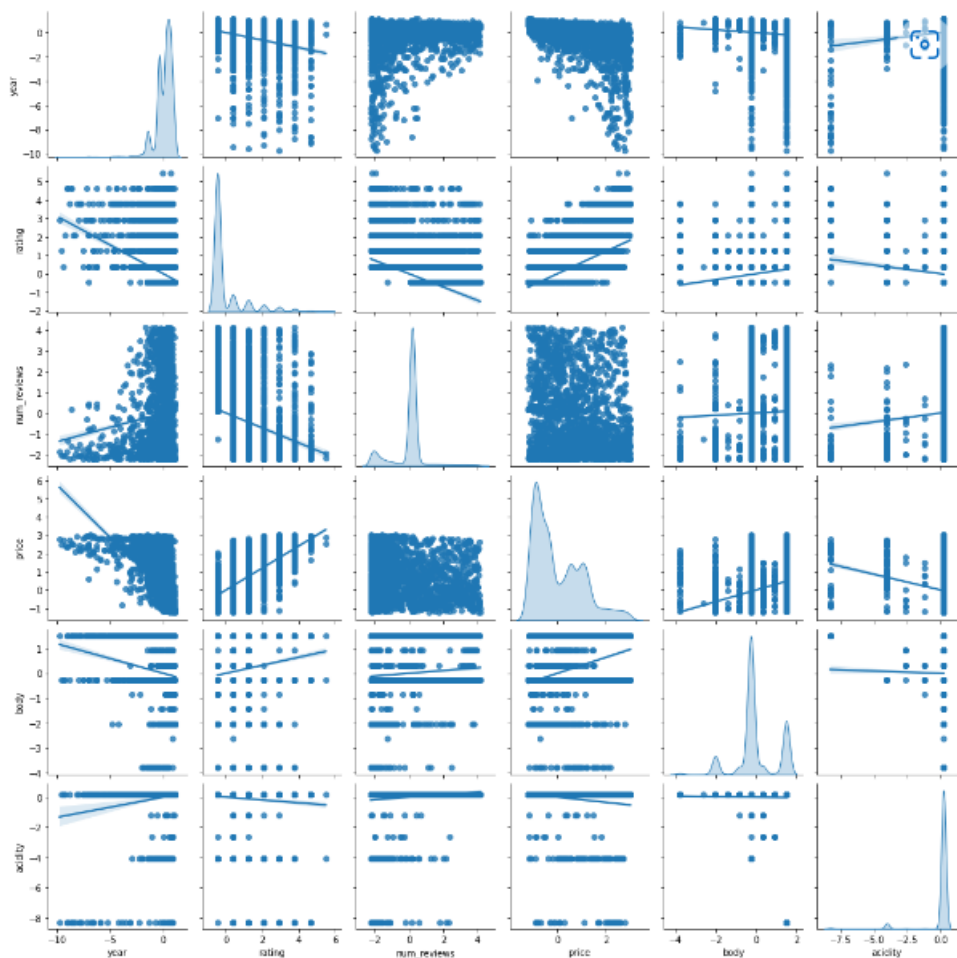
```
winery      0
wine        0
year        0
rating      0
num_reviews 0
region      0
price       0
type       72
body        0
acidity     0
dtype: int64
```

(二)、探索性資料分析

使用填補完後的資料畫出熱力圖、散佈圖、長條圖觀察變數之間的關係。

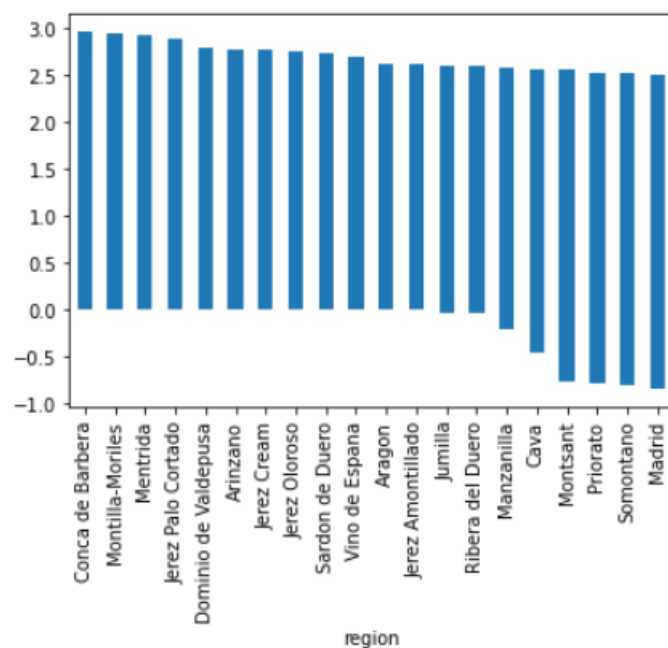


填補完後的資料對照網頁上其他參與者直接刪去資料後的結果相近，並沒有因為填補資料大幅影響資料原有的關係與分布，不論是直接刪除缺失值或填補後資料仍關係與分部接近一致。



從熱力圖與散佈圖可以看到數值型資料與價格之間的關係，其中 rating、body、

acidity 與價格有正相關，排名越高、酒的丹寧與口感好壞與否與價格高低有正向的關係。



類別型資料以 region 為例，其餘類別型資料型態與此相似，難以從中看出其與價格之間的關係。

交叉驗證

使用 knn 補值雖然較單純使用平均值、眾數或中位數考量到的面向更多，但仍需要將填補後的資料分割建模驗證，此次使用資料量共 7500 筆，故將資料隨機切分成訓練、測試、驗證做交叉驗證，計算不同模型經過十次隨機切分資料後的 R-squared 平均與 MSE 平均來選擇此份資料使用的模型，其中訓練：測試：驗證 = 0.5：0.3：0.2。

R-squared

```
{'Linear Regression': 0.6015193571882733,  
 'Lasso': -0.0006865952840817702,  
 'Ridge': 0.6043572863690513,  
 'KNeighbors Regressor': 0.8485122782552459,  
 'Random Forest Regressor': 0.8825035344166551}
```

MSE(下圖數值取正值後方為 mse)


```
{'Linear Regression': -0.4012606467452372,
 'Lasso': -1.0080332737302093,
 'Ridge': -0.39078443579101974,
 'KNeighbors Regressor': -0.14995290118491042,
 'Random Forest Regressor': -0.11915092337975401}
```

R-squared & MSE、RMSE of Random Forest Regression

```
score of train: 0.9828039705874156   mse: 0.10969621774785066
score of test: 0.8922935773749382   rmse: 0.33120419343337226
```

R-squared & MSE、RMSE of KNN Regression


```
score of train: 0.905350470885817   mse: 0.1421366596434931
score of test: 0.8619604364548552   rmse: 0.3770101585415081
```

從上述結果，Knn regression 和 Random Forest Regressor 的 R-squared 較大且 MSE 較小，且套用到測試資料後，Random Forest Regressor 的 R-squared 於訓練集與測試集的差距較 Knn regression 大，故此次選用 Knn regression 建模。

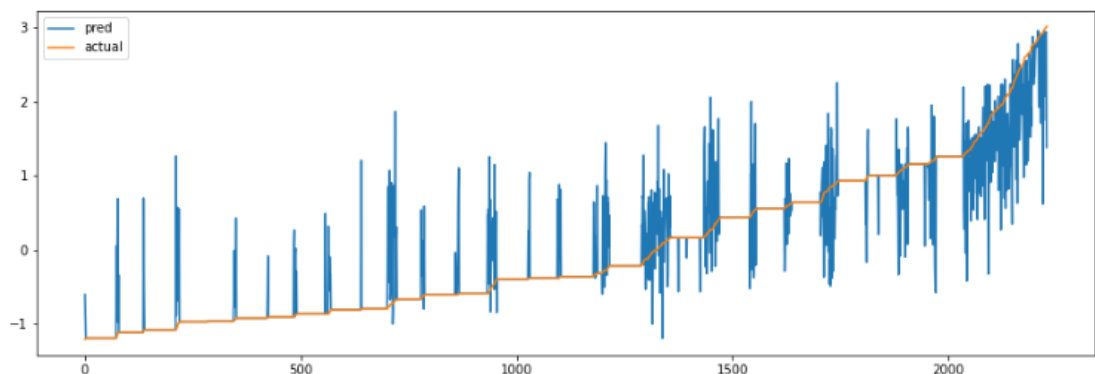


建模

將資料以 7：3 的比例分成訓練集與測試集，並計算測試與訓練資料的 R-squared、預測結果與測試資料之間的 MSE、RMSE 作為模型配適評估標準。

 使用類別型資料與數值型資料

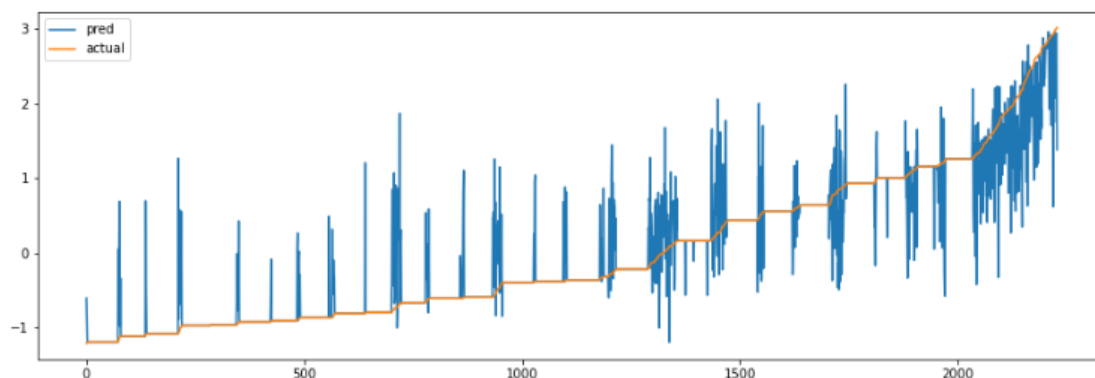
```
score of train: 0.905350470885817   mse: 0.1421366596434931
score of test: 0.8619604364548552   rmse: 0.3770101585415081
```



R-squared 相較於平台上直接刪去缺失值的預測結果高，MSE 也小一些。

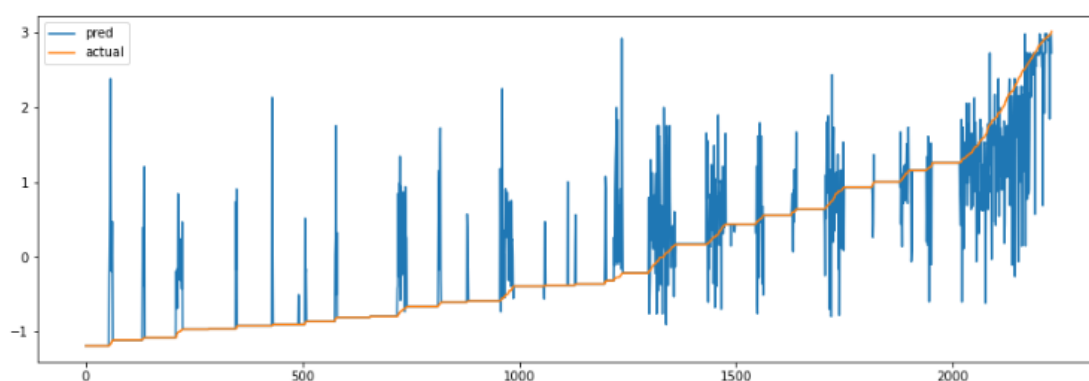
使用數值型資料

score of train: 0.8943807774329595 mse: 0.15912515006654784
score of test: 0.847408078650668 rmse: 0.3989049386339405



使用類別型資料

score of train: 0.879560706365859 mse: 0.20474276846476117
score of test: 0.7955751789883969 rmse: 0.45248510303076406



從上述結果可以看到，若單純使用類別型變數預測價格的 mse，相較於使用數值型或使用全部資料來的高一些，且訓練集與測試集的 R-squared 差距也較單純使用數值型資料與全部資料來的大，且加入類別型資料後並沒有讓 mse 與 R-squared 明顯下降或上升，由此推論數值型的資料對價格的影響較大。