Relax

建模流程

EDA -> 製造特徵 -> 填補缺失值 -> 特徵篩選 ->
初步建模 -> 調整參數 -> stacking

特徵工程 (Feature Engineering)

1.將預測目標改成每平方的價格
Code : np.log1p(df[total_price] / df[building_area])

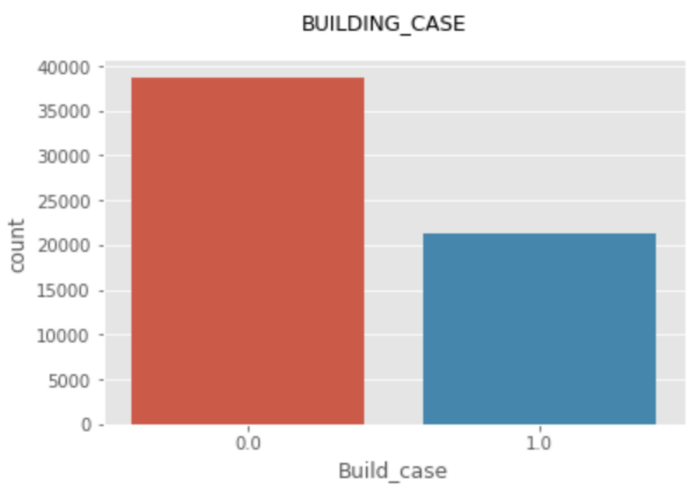
2. 合併城市、鄉鎮、建築型態對目標取中位數
Code : Groupby([city, town, building_type])[target].median()

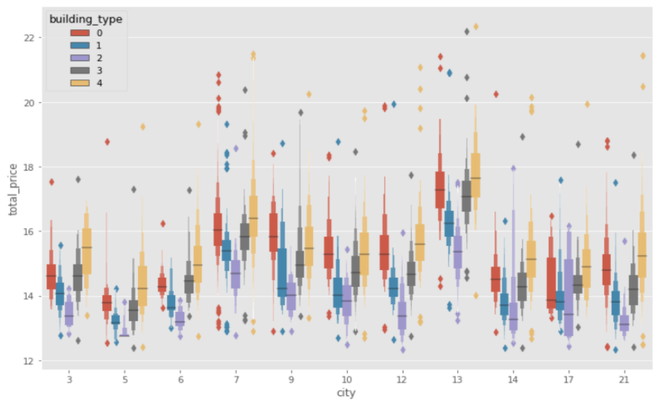
3. 找出透天和大樓，並對大樓做平均所在的樓層，透天則填1
Code : df[txn_floor] / df[total_floor]

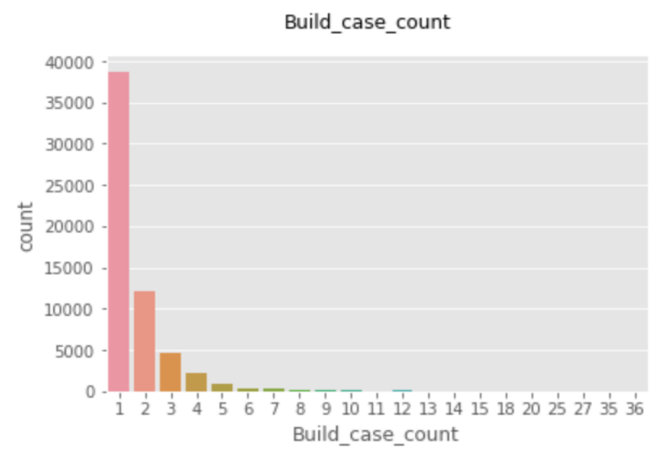
4. 對於週遭的建築物密度做遞增或遞減的判斷(類似速度和等加速度)
Code : (df[I_50] – df[I_10]) / 40
Code : (df[I_100] – df[I_50]) / 50 - (df[I_50] – df[I_10]) / 40

5. 對各類的最近建築物做統計特徵
Code : df[Min_col].std(axis = 1)

6. 找出非建商的建築物，將其複製
Code : Clone df.loc[~df[col].duplicated(keep = False)]







填補缺失值

1. 將parking_way = 2 的 parking_area 和 parking_price 填上0

2. 將 village_income_median 的缺失值，填上同 city, town 的中位數

特徵篩選

刪掉一些無用的特徵，並擷取feature_importance前300

初步建模、自動調參

使用 Lightgbm、使用 hyperopt 調整參數

1.參數 parameters

A.General

- n_estimators = 1000000(每個 fold 最大 training rounds)
- learning_rate = 0.01(default=0.1, 值越小 accuracy 越高)
- early_stopped_rounds = 3000(避免過度 training 造成 over-fitting)

B.Leaf-wise (Best-first) Tree (避免 over-fitting, under-fitting)

- num_leaves = 30(should be smaller than 2^max_depth)
- max_depth = -1
  Note : 根據 Lightgbm 官方文件，deeper tree depth 可能造成 over-fitting，
  在我們的 model 中使用 early_stopped_rounds 來控制 tree depth 因此未設置 max_depth
- min_data_in_leaf = 50(太大會 under-fitting)

C.Hyperopt

給定 parameters 之範圍，加上 Evaluation functions，找出最佳參數

2.K-Fold Cross Validation

- 使用10 folds
- 合成每次 fold 結果 (predictions/10) 為最終預測結果

圖示▼
以5 folds為例