```
ID
              種類
              地域
               市区町村コード
              都道府県名
               市区町村名
              地区名
               最寄駅: 名称
              最寄駅:距離(分)
              間取り
               面積 (m)
              土地の形状
              延床面積(m)
データについて
               建築年
               建物の構造
               用途
               今後の利用目的
              前面道路:方位
              前面道路:種類
              前面道路:幅員(m)
              都市計画
              建ぺい率 (%)
              容積率(%)
              取引時点
              改装
              取引の事情等
              取引価格(総額)_log 目的変数
                                                                    欠損値をすべて 'missing' で補完する場合
                           欠損値の補完(空白は欠損値とみなされていないため注意が必要)
                                                                他の特徴量を用いて予測し、数値の欠損値を補完する場合
                                              取引時点_何年前:現在から取引時点の差を計算
                                              面積 (m) 容積率 (%) _combi : 面積 (m) * 容積率 (%) / 100
                                              取引時点_enc : 「取引時点」カラムをエンコードして、各ユニークな取引時点を整数に変換する処理
                                              _ [『景寄駅: 距離(分)", "面積(㎡)", "建ペい率(96)", "容積率(96)"]と["ount", "mean", "min", "mea"/を作用させた特徴量:時間的な特徴と地域(都道府県名)に基づいたターゲット変数の過去のカウント、平均値、最小値、最大値を計算
                                              築年数:建築年から築年数を算出する
                                              取引時点_日付 : 四半期の始まりを示す日付を生成
                                              消費税率:当時の消費税率を表す
                           新たに追加した特徴量
EDA+前処理、特徴量エンジニアリング
                                              madori_mean : 間取りごとの取引価格 (総額) _logの平均を取った特徴量を作成する
                                              最寄り駅乗降客数:外部データを用いて最寄駅:名称に紐づける形で追加
                                              緯度:外部データを用いて住所から算出
                                              経度 : 同様に外部データから算出
                                              東京都港区からの距離 : 上記で追加した 緯度 , 経度 から各地点と東京都港区までの距離を算出
                                              *9~12の特徴量はbase2.csvに追加済み
                                                全ての特徴量をダミー化した場合 -> 実行時間の関係で不可能
                                                全ての特徴量をラベルエンコーディングした場合
                           カテゴリカル変数の数値化
                                                全ての特徴量をカウントエンコーディングした場合
                                                全ての特徴量をターゲットエンコーディングをした場合
                                         hold-out法 (都道府県名 で層化)
                       バリデーション方法へ
                                        StratifiedKHold (K=10、同じく 都道府県名 で層化)
                       optunaによるハイパーパラメータの調整
           lightgbm
                       過学習しすぎない程度に学習回数を増やした。(20000~50000)
           catboost
                       学習率は学習回数の関係であまり低くできなかった。
           xgboost
           ランダムフォレスト
価格予測
           Kerasを使用したディープニューラルネットワーク
           MLPRegressor
           pycaret
           スタッキング
                住宅価格の補正を行う場合
                最近の住宅価格の高騰を考慮したほうが良いと考え、1.5%, 3.0%, 5.0%と3パターンで試しに補正を加えてみた。
                精度はあまり変わらなかった。
予測ファイル提出。
                アンサンブルして予測値を出す。
                2パターンの目的変数で予測したもの
                ハイバーバラメータを調整したもの
            今回は他の参加者とは違い、すべての欠損値を補完して予測値を算出したが無理に補完するよりも欠損値を扱えるlightgbmなら精度は良かったかもしれない。
           アンサンブルをする際にもっと特徴の異なるモデルを使った方が良かったかもしれない。(boosting_typeの変更など)
振り返り
           さらに上位を狙うなら特徴量エンジニアリングの部分でまだできたことがあったかも。
しかしながら、今回のコンベティションで特徴量エンジニアリングの大切さを学んだ。
           次回はもっと深層学習の知識をつけて臨みたい。
```