通常課題1

氏名:薛丁銘

所属:東京大学　工学系研究科　システム創成学専攻

学籍番号:37-237256

1. Introduction

The common topic 1 tackles the problem of price predicting of Tokyo real estate.

Firstly, I am required to imagine that I am intending to purchase some real estate in the first quarter of 2023. From the official website of Land General Information System the detailed history data of land transaction happened in Tokyo can be acquired. These data could be used for model training and by introducing the corresponding features in test dataset, the prices in the first quarter could be predicted. Except for simply predicting the prices, another task is to predict with two different training datasets, which are the most recent 50000 pieces of transaction data (referred as recent data) as well as randomly selected 50000 pieces of transaction data (referred as random data) and compare their training accuracy.

In this topic, I introduced several models for price prediction including 3 different linear regression models, random forest and gradient boosting. For the random forest and gradient boosting, I also tried hyperparameter learning for better hyperparameter optimization. As for the conclusion, Extreme Gradient Boosting model shows the highest quality of price prediction and linear regression model preformed worst. Besides, compared to the result of price prediction in the whole Japan, the importance of features differs significantly and the performance decreased in every models. More importantly, prediction accuracy of recent data is better than random data for the case of most models, indicating that for the sequence data like price is quite time sensitive and it shares high possibility of inheriting features from nearby time periods. In addition, I streamlined the code by extracting common parts and abstracting them into functions, making the code more concise and efficient. Through the coding I gained improvements in my coding ability. Large langue model(LLM) such as ChatGPT was also used in this work mostly for model selection, reference code searching and slightly for report proofreading.

2. Problem Analysis

The common topic 1 appears to be a simple prediction problem with a sub mission of using two different training datasets for training and comparing the performance on the same test dataset. I divide the whole work into two parts: data preprocessing and model training.

In data preprocessing period, the first step is to download the history transaction datasets and divide it into the training and test datasets. There should be data cleaning for handling missing/incorrect data, data transformation for converting data into suitable format such as data of time period and room size as well as converting the input dataframe into suitable format such as two-dimensional matrix as machine learning models input, data reduction for reducing the unnecessary features.

In the model training period, I applied different linear regression models, Random Forest and Extreme Gradient Boosting for training. For Random Forest and XGBoosting model, hyperparameter learning was used for better parameter optimizations. R-squared score and MSE are mainly used for evaluating the accuracy. Considering that the features are mainly sparse matrix, lasso linear regression model was introduced expecting for better results.

Dingming Xue, The University of Tokyo, Graduate School of Engineering, Department of System Innovation

3. Methodologies and Experiments

3.1 Data Preprocessing

This paragraph discusses the data preprocessing period of the work. The data preprocessing period includes data download, data cleaning, data reduction and data transformation. The necessary packages were imported and the number of data for training is set as a global variable as shown in Figure 3.1.1.

```
#import the necessary packages
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
import datetime
import jaconv
import re
import pickle as pkl
import warnings
import statsmodels.api as sm
from sklearn.model_selection import train_test_split
from sklearn.model_selection import GridSearchCV
from sklearn import linear_model
from sklearn.linear_model import LinearRegression
from sklearn.metrics import r2_score, mean_squared_error
from sklearn.linear_model import Lasso
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_squared_error
from sklearn.preprocessing import StandardScaler
from sklearn.ensemble import RandomForestRegressor
import xgboost as xgb

# To ignore the wanrings. Got many warnings from the data type
warnings.filterwarnings("ignore")
plt.rcParams['font.family'] = 'MS Gothic'

NUMBER_OF_DATA = 50000
```

Figure 3.1.1 Package Import and Global Variables

All the existing transaction data before the first quarter of 2023 was downloaded to the local but I encountered the problem of encoding error. The encoding of the csv file was JIS and I used Notepad++ to transfer the csv files into UTF-8 format. I used sample method of Dataframe to randomly select 50000 pieces of data from the history transaction data. As for the most recent data, I firstly checked whether the length of the data in the last quarter of 2022 is more than 50000 as shown in Figure 3.1.2. If so, I would randomly select 50000 pieces of data from it. If not, I would sort the history transaction data by transaction time and pick the most recent 50000 pieces.

```
# Import data from csv file
house_price_test = pd.read_csv("D:\\code\\school\\データサイエンス超入門\\通常課題１\\data\\utf\\13_Tokyo_20231_20231.csv")
house_price_history_data = pd.read_csv("D:\\code\\school\\データサイエンス超入門\\通常課題１\\data\\utf\\13_Tokyo_20053_20224.csv")

# house_price_history_data = pd.read_csv("school||データサイエンス超入門||通常課題１||data||utf||SuperD_Class2_tochi_sample_xdm.csv")
house_price_alltime = house_price_history_data
house_price_recent50000 = house_price_history_data[house_price_history_data["取引時点"] == '2022年第４四半期']
code_reference = pd.read_csv("D:\\code\\school\\データサイエンス超入門\\通常課題１\\data\\utf\\SuperD_Class2_shicyoukuson_code_utf8.c

# get the most recent 50000 data
if len(house_price_recent50000) > NUMBER_OF_DATA:
    house_price_recent50000 = house_price_recent50000.sample(n = NUMBER_OF_DATA, random_state = 42)
else:
    house_price_history_data.sort_values(by = '取引時点', ascending= False, inplace = True)
    house_price_recent50000 = house_price_history_data.iloc[0:NUMBER_OF_DATA, :]
# get random 50000 data
house_price_random50000 = house_price_history_data.sample(n = NUMBER_OF_DATA, random_state= 42)
```

Figure 3.1.2 Data Sampling

I created several methods for data transformation and got the input matrix I wanted. However, after the dummy convert, the dataframe was filled was Boolean data of true and false. The machine learning model could not recognize the Boolean data. In that case it was necessary to transform the Boolean data into 0 and 1 as shown in Figure 3.1.3(a) and Figure 3.1.3(b).

```
## Datatype Adjustment Before
recent50000_feature_df.head(10)
```

Out[67]:

	area	longitude	latitude	種類_中古マンション等	種類_宅地(土地)	種類_宅地(土地と建物)	種類_林地	種類_農地	取引の事情等_その他事情有り	取引の事情等_瑕疵有りの可能性	取引の事情等_私道を含む取引	取引の事情等_調停・競売等	取引の事情等_調停・競売等	取引の事情等_調停・競売等、私道を含む取引	取引の事情等_関係者間取引	取引の事情等_関係者間取引、私道を含む取引	取引の事情等_隣地の購入	取引の事情等_隣地の購入、私道を含む取引
0	15	139.753595	35.694003	True	False	False	False	False	False	False	False	False	False	False	False	False	False	False
6179	30	139.751599	35.658068	True	False	False	False	False	False	False	False	False	False	False	False	False	False	False
81982	120	139.653247	35.646572	False	False	True	False	False	False	False	False	False	False	False	False	False	False	False
81984	190	139.653247	35.646572	False	True	False	False	False	False	False	False	False	False	False	False	False	False	False
81986	30	139.653247	35.646572	True	False	False	False	False	False	False	False	False	False	False	False	False	False	False
81988	30	139.653247	35.646572	True	False	False	False	False	False	False	False	False	False	False	False	False	False	False
81990	30	139.653247	35.646572	True	False	False	False	False	False	False	False	False	False	False	False	False	False	False
81992	30	139.653247	35.646572	True	False	False	False	False	False	False	False	False	False	False	False	False	False	False
81994	30	139.653247	35.646572	True	False	False	False	False	False	False	False	False	False	False	False	False	False	False
36522	45	139.730186	35.609226	True	False	False	False	False	False	False	False	False	False	False	False	False	False	False

Figure 3.1.3(a) Dataframe Before Adjustment

```
## Datatype Adjustment After
recent50000_feature_df.head(10)
```

Out[74]:

	area	longitude	latitude	種類_中古マンション等	種類_宅地(土地)	種類_宅地(土地と建物)	種類_林地	種類_農地	取引の事情等_その他事情有り	取引の事情等_瑕疵有りの可能性	取引の事情等_私道を含む取引	取引の事情等_調停・競売等	取引の事情等_調停・競売等	取引の事情等_調停・競売等、私道を含む取引	取引の事情等_関係者間取引	取引の事情等_関係者間取引、私道を含む取引	取引の事情等_隣地の購入	取引の事情等_隣地の購入、私道を含む取引
0	15.0	139.753595	35.694003	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
6179	30.0	139.751599	35.658068	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
81982	120.0	139.653247	35.646572	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
81984	190.0	139.653247	35.646572	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
81986	30.0	139.653247	35.646572	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
81988	30.0	139.653247	35.646572	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
81990	30.0	139.653247	35.646572	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
81992	30.0	139.653247	35.646572	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
81994	30.0	139.653247	35.646572	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
36522	45.0	139.730186	35.609226	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

Figure 3.1.3(b) Dataframe After Adjustment

Up till now, the feature and label dataframe of both training datasets and test datasets are prepared.

3.2 Model Selection

In this work, several models were selected and deployed for acquiring prediction results. The OLS model of Statsmodels package and LinearRegression model of Sklearn package were used firstly for prediction. However, the outcome showed very low performance on the test dataset. The average R-squared score was around 0.4. Considering that the features were sparse because of creating the dummy features, I introduced Lasso Linear Regression model hoping to solve the problem. However, the result still did not work out well. Therefore, random forest and XGBoosting was introduced and acquired relatively good results.

4. Training Results

4.1 Linear Regression Model

For the prediction result of Linear Regression models, I meanly focus on the R-squared score and the MSE score. The prediction result of Linear Regression model of Sklearn is as shown in Figure 4.1.1. The result shows that using recent data as training data has better performance on the predicting result since the R-square score of recent data is larger an the MSE score is smaller. However, the coefficients indicates that the regression result is not similar between using recent data and random data.

```
########## Linear regression result on recent 50000 data powered by Sklearn ##########

Training result powered by Sklearn
R-squared: 0.44812038429006373
Mean Squared Error: 0.4941691968771306
area: 55792072848766.85
longitude: 55792072849299.836
latitude: 55792072848799.836
種類_中古マンション等: 55792072772489.71
種類_宅地(土地): 55792072772420.12
種類_宅地(土地と建物): 55792072772503.164
種類_林地: 55792072769334.22
種類_農地: 55792072769194.945
取引の事情等_その他事情有り: 55792072848534.445
取引の事情等_瑕疵有りの可能性: 55792072848519.46
取引の事情等_私道を含む取引: 55792072848724.86
取引の事情等_調停・競売等: 55792215794605.84
取引の事情等_調停・競売等: 55791929902822.78
取引の事情等_調停・競売等、私道を含む取引: 55792072848510.6
取引の事情等_関係者間取引: 55792072848611.234
取引の事情等_関係者間取引、私道を含む取引: 55792072848425.39
取引の事情等_隣地の購入: 55792072848748.805
取引の事情等_隣地の購入、私道を含む取引: 55792072848736.78

########## Linear regression result on random 50000 data powered by Sklearn ##########

Training result powered by Sklearn
R-squared: 0.3875772132773805
Mean Squared Error: 0.5483813281899448
area: 19303.356449482013
longitude: 19779.382917566545
latitude: 19340.030319312544
種類_中古マンション等: -49161.78953753291
種類_宅地(土地): -49225.25566050825
種類_宅地(土地と建物): -49134.26941786124
種類_林地: -51980.879472066496
種類_農地: -50898.05771383226
取引の事情等_その他事情有り: 19206.315610851754
取引の事情等_瑕疵有りの可能性: 19242.578791766675
取引の事情等_私道を含む取引: 19255.81445420904
取引の事情等_調停・競売等: 19229.811649556927
取引の事情等_調停・競売等: 19229.81164955751
取引の事情等_調停・競売等、私道を含む取引: 18993.64292495218
取引の事情等_関係者間取引: 19195.377669752135
取引の事情等_関係者間取引、私道を含む取引: 19086.75549151491
取引の事情等_隣地の購入: 19282.205805345497
取引の事情等_隣地の購入、私道を含む取引: 19255.168067608127
```

Figure 4.1.1 Training Results of Sklearn Linear Regression Model

Then the Linear Regression model of Statsmodels Package was introduced to compared the results as shown in Figure 4.1.2(a) and Figure 4.1.2(b). In this result we can see that the random data has out-performed the recent data regarding to it's higher R-squared score and lower MSE score. This no doubts leads to the uncertainty of the model.

```
########## OLS result on recent 50000 data powered by Statismodel ##########

Training result powered by Stats Model
                            OLS Regression Results
==============================================================================
Dep. Variable:                      y   R-squared:                       0.417
Model:                            OLS   Adj. R-squared:                  0.417
Method:                 Least Squares   F-statistic:                     2236.
Date:                Sat, 13 Jan 2024   Prob (F-statistic):               0.00
Time:                        15:24:21   Log-Likelihood:                -53290.
No. Observations:               50000   AIC:                         1.066e+05
Df Residuals:                   49983   BIC:                         1.068e+05
Df Model:                          16
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
x1            20.1897      0.018   1149.779      0.000      20.155      20.224
x2           552.6181      6.032     91.616      0.000     540.796     564.441
x3            53.2894      8.169      6.524      0.000      37.279      69.300
x4         -7.618e+04    909.630    -83.748      0.000    -7.8e+04   -7.44e+04
x5         -7.625e+04    909.517    -83.835      0.000    -7.8e+04   -7.45e+04
x6         -7.617e+04    909.491    -83.746      0.000   -7.79e+04   -7.44e+04
x7         -7.934e+04    908.576    -87.319      0.000   -8.11e+04   -7.76e+04
x8         -7.947e+04    916.436    -86.721      0.000   -8.13e+04   -7.77e+04
x9          -211.5372    102.489     -2.064      0.039    -412.417     -10.658
x10         -227.1812     98.434     -2.308      0.021    -420.113     -34.249
x11          -21.7966      4.050     -5.382      0.000     -29.734     -13.859
x12          -81.4844     11.712     -6.957      0.000    -104.440     -58.529
x13         -236.3229     56.204     -4.205      0.000    -346.482    -126.163
x14         -135.4514     19.884     -6.812      0.000    -174.425     -96.478
x15         -321.3151     76.470     -4.202      0.000    -471.197    -171.433
x16            2.0818     17.177      0.121      0.904     -31.586      35.750
x17           -9.1092     53.364     -0.171      0.864    -113.702      95.484
==============================================================================
Omnibus:                    19489.593   Durbin-Watson:                   1.684
Prob(Omnibus):                  0.000   Jarque-Bera (JB):           405460.979
Skew:                          -1.371   Prob(JB):                         0.00
Kurtosis:                      16.679   Cond. No.                     4.19e+05
==============================================================================
```

Figure 4.1.2(a) Result of Recent Data by OLS Model

```
########## OLS result on random 50000 data powered by Statismodel ##########

Training result powered by Stats Model
                            OLS Regression Results
==============================================================================
Dep. Variable:                      y   R-squared:                       0.452
Model:                            OLS   Adj. R-squared:                  0.451
Method:                 Least Squares   F-statistic:                     2572.
Date:                Sat, 13 Jan 2024   Prob (F-statistic):               0.00
Time:                        15:24:21   Log-Likelihood:                -51824.
No. Observations:               50000   AIC:                         1.037e+05
Df Residuals:                   49983   BIC:                         1.038e+05
Df Model:                          16
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
x1            20.0386      0.016   1264.548      0.000      20.008      20.070
x2           496.0196      5.901     84.055      0.000     484.453     507.586
x3            56.7227      8.977      6.319      0.000      39.128      74.318
x4         -6.844e+04    889.371    -76.952      0.000   -7.02e+04   -6.67e+04
x5          -6.85e+04    889.260    -77.033      0.000   -7.02e+04   -6.68e+04
x6         -6.841e+04    889.251    -76.932      0.000   -7.02e+04   -6.67e+04
x7         -7.126e+04    889.558    -80.105      0.000    -7.3e+04   -6.95e+04
x8         -7.018e+04    895.690    -78.348      0.000   -7.19e+04   -6.84e+04
x9           -76.9996     54.072     -1.424      0.154    -182.981      28.981
x10          -40.7410     77.537     -0.525      0.599    -192.715     111.233
x11          -27.4938      3.911     -7.030      0.000     -35.160     -19.828
x12         -123.5348      6.917    -17.860      0.000    -137.092    -109.977
x13         -289.6635     27.786    -10.425      0.000    -344.125    -235.203
x14          -87.9358     14.670     -5.994      0.000    -116.688     -59.183
x15         -196.5504     46.421     -4.234      0.000    -287.537    -105.564
x16           -1.1065     12.295     -0.090      0.928     -25.204      22.991
x17          -28.1425     32.731     -0.860      0.390     -92.295      36.010
==============================================================================
Omnibus:                    14362.797   Durbin-Watson:                   2.016
Prob(Omnibus):                  0.000   Jarque-Bera (JB):           186508.104
Skew:                          -1.017   Prob(JB):                         0.00
Kurtosis:                      12.241   Cond. No.                     4.19e+05
==============================================================================
```

Figure 4.1.2(b) Result of Random Data by OLS Model

Then the Lasso linear regression model was introduced to seek for a balance. However, the result was not good as shown in Figure 4.1.3. The low quality of regression makes the result lack of reliability.

```
########## Lasso Linear regression result on recent 50000 data powered by Sklearn ##########

Training result powered by Lasso Regression Model
R-squared: 0.3083153099156911
Mean Squared Error: 0.6193547615478922
area: 0.35994179982690644
longitude: 0.2526910318828098
latitude: 0.04689790517341657
種類_中古マンション等: -0.1511908170316618
種類_宅地(土地): 0.0
種類_宅地(土地と建物): 0.11376278479012235
種類_林地: -0.2983943790334744
種類_農地: -0.10471444095515355
取引の事情等_その他事情有り: -0.0
取引の事情等_瑕疵有りの可能性: -0.0
取引の事情等_私道を含む取引: -0.022293517963485887
取引の事情等_調停・競売等: -0.019718728012610945
取引の事情等_調停・競売等: -1.1368683772161606e-16
取引の事情等_調停・競売等、私道を含む取引: -0.001923769718001568
取引の事情等_関係者間取引: -0.005787655340749497
取引の事情等_関係者間取引、私道を含む取引: -0.001243623033162912
取引の事情等_隣地の購入: -0.0
取引の事情等_隣地の購入、私道を含む取引: -0.0


########## Lasso Linear regression result on random 50000 data powered by Sklearn ##########

Training result powered by Lasso Regression Model
R-squared: 0.25294559833201735
Mean Squared Error: 0.668934425528458
area: 0.39349361920221543
longitude: 0.21605504000905304
latitude: 0.060993804950860334
種類_中古マンション等: -0.15241287718942378
種類_宅地(土地): 0.0
種類_宅地(土地と建物): 0.12258829446468954
種類_林地: -0.21062078051107544
種類_農地: -0.03926293250293
取引の事情等_その他事情有り: -0.0
取引の事情等_瑕疵有りの可能性: -0.0
取引の事情等_私道を含む取引: -0.022652923355671786
取引の事情等_調停・競売等: -0.05329224879725695
取引の事情等_調停・競売等: -4.365574568510052e-16
取引の事情等_調停・競売等、私道を含む取引: -0.02078645788805254
取引の事情等_関係者間取引: -0.008531878859217898
取引の事情等_関係者間取引、私道を含む取引: -0.00305115104758715
取引の事情等_隣地の購入: -0.003957861185836493
取引の事情等_隣地の購入、私道を含む取引: -0.0
```

Figure 4.1.3 Training Results of Lasso Linear Regression Model

In this case, there's not necessary to continue to experiment on the linear regression models. Then Random Forest Model and XGBoosting were used as shown in the next paragraph.

4.2 Random Forest Model

Random Forest Model is expected to acquire better results and indeed the results were better than Linear regression. The result is as shown in Figure 4.2.

The parameters were selected by the hyperparameter learning for acquiring better results. From the importance map we can see that area, longitude, latitude and categories are the key features which effects the price. The R-square score and MSE score are improved

```
########## Random Forest regression result on recent 50000 data ##########

Training result powered by Random Forest
R-squared: 0.6493390980041452
Mean Squared Error: 0.3139920580190073
Model Feature Importance:
area: 0.49783569226178437
longitude: 0.21837676690822386
latitude: 0.10518142564308096
種類_中古マンション等: 0.06386214681689198
種類_宅地(土地): 0.014130451689590689
種類_宅地(土地と建物): 0.04817384026835631
種類_林地: 0.03903759927012275
種類_農地: 0.006078036290530012
取引の事情等_その他事情有り: 4.613916200476083e-05
取引の事情等_瑕疵有りの可能性: 0.00010914025270560788
取引の事情等_私道を含む取引: 0.0030940417050418323
取引の事情等_調停・競売等: 0.0008898558918701198
取引の事情等_調停・競売等: 0.000921836984808377
取引の事情等_調停・競売等、私道を含む取引: 0.00022267162127460774
取引の事情等_関係者間取引: 0.0011320768370293717
取引の事情等_関係者間取引、私道を含む取引: 0.0001643802750880655
取引の事情等_隣地の購入: 0.0005208982057416415
取引の事情等_隣地の購入、私道を含む取引: 0.00022299991585473958


########## Random Forest regression result on random 50000 data ##########

Training result powered by Random Forest
R-squared: 0.561558933159116
Mean Squared Error: 0.3925929925858849
Model Feature Importance:
area: 0.5336872752361422
longitude: 0.18150961560781098
latitude: 0.10248605996142193
種類_中古マンション等: 0.07973957774455698
種類_宅地(土地): 0.018524736084031303
種類_宅地(土地と建物): 0.05037191130565045
種類_林地: 0.01845314273242555
種類_農地: 0.0015884150864960873
取引の事情等_その他事情有り: 6.325332911477334e-05
取引の事情等_瑕疵有りの可能性: 3.435190432721187e-05
取引の事情等_私道を含む取引: 0.002795763070736946
取引の事情等_調停・競売等: 0.0036924182618140186
取引の事情等_調停・競売等: 0.00373957353329977
取引の事情等_調停・競売等、私道を含む取引: 0.000953867112667997
取引の事情等_関係者間取引: 0.0011239973410640857
取引の事情等_関係者間取引、私道を含む取引: 0.00023050353027831037
取引の事情等_隣地の購入: 0.0008683486952355248
取引の事情等_隣地の購入、私道を含む取引: 0.00013718946292589433
```

Figure 4.2 Training Results of Random Forest Model

by 20%. Considering that house price prediction is a very complicated task, the R-square score of 0.649 is good as the result. Meanwhile, the comparison shows that with a good prediction model recent data is better for house price prediction.

4.3 XGBoosting Model

XGboosting can be considered as an update of Random Forest Model. The training result of XGboosting model is as shown in Figure 4.3.1.

XGboosting model has a slight improvement upon Random Forest Model. The result also indicates that using recent data for prediction is better. However, as for the importance map, the result was completely different from the Random Forest Model. In this case, the category has become the most import feature as shown in Figure 4.3.2.

5. Conclusion

In this topic several models are used for real estate prediction such as linear regression model, random forest and xgboost. For most of the models, using recent data as training dataset has better prediction accuracy. For the comparison between models, xgboost is the best and linear regression models are the worst. Data normalization is required to improve the prediction result but it depends on the model. The prediction is not precise because for the same model (such as linear regression), using different can lead to different coefficients of the features. There's another important finding that by increasing the size of training data, the result could become worse. This topic enhances my ability of manipulating the models and improved my ability of coding. I also put some efforts in making the code easy to read. ChatGPT played an important role in this work because I used it for supporting my coding. It greatly improved my speed of coding.

```
########## Grident Boosting regression result on recent 50000 data ##########

Training result powered by Grident Boosting
R-squared: 0.6869800840964437
Mean Squared Error: 0.2802872149021509
Model Feature Importance:
area: 0.04973591864109039
longitude: 0.035686880350112915
latitude: 0.03204427286982536
種類_中古マンション等: 0.008472505025565624
種類_宅地(土地): 0.00765584921464324
種類_宅地(土地と建物): 0.015589170157909393
種類_林地: 0.5688657164573669
種類_農地: 0.22708047926425934
取引の事情等_その他事情有り: 0.0033060505520552397
取引の事情等_瑕疵有りの可能性: 0.002921097446233034
取引の事情等_私道を含む取引: 0.005610044114291668
取引の事情等_調停・競売等: 0.006256985478103161
取引の事情等_調停・競売等、私道を含む取引: 0.005677361041307449
取引の事情等_関係者間取引: 0.013353920541703701
取引の事情等_関係者間取引、私道を含む取引: 0.006233620457351208
取引の事情等_隣地の購入: 0.005283720791339874
取引の事情等_隣地の購入、私道を含む取引: 0.00622638501226902


########## Grident Boosting regression result on recent 50000 data ##########

Training result powered by Grident Boosting
R-squared: 0.606243817120161
Mean Squared Error: 0.352580836689945
Model Feature Importance:
area: 0.049682535231113434
longitude: 0.03855186700820923
latitude: 0.02676851488649845
種類_中古マンション等: 0.007659765426069498
種類_宅地(土地): 0.008722295984625816
種類_宅地(土地と建物): 0.02508227340877056
種類_林地: 0.669140100479126
種類_農地: 0.10146109759807587
取引の事情等_その他事情有り: 0.0016833031550049782
取引の事情等_瑕疵有りの可能性: 0.0016154288314282894
取引の事情等_私道を含む取引: 0.005324224010109901
取引の事情等_調停・競売等: 0.010084372013807297
取引の事情等_調停・競売等、私道を含む取引: 0.012964829802513123
取引の事情等_関係者間取引: 0.01341442670673132
取引の事情等_関係者間取引、私道を含む取引: 0.014389580115675926
取引の事情等_隣地の購入: 0.007850238122045994
取引の事情等_隣地の購入、私道を含む取引: 0.005605142563581467
```

Figure 4.3.1 Training Results of XGBoosting Model

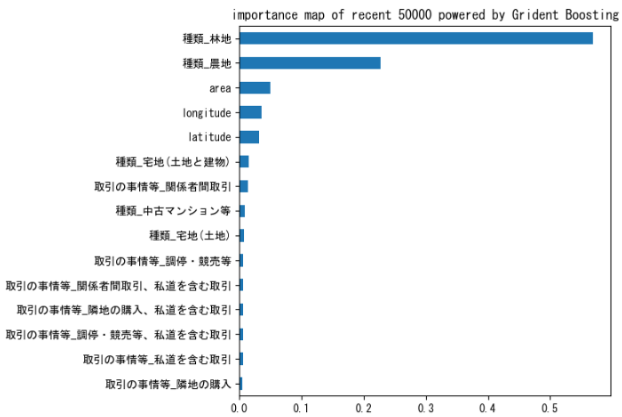

Figure 4.3.2 Importance map of XGBoosting Model