Forecasting Shanghai Second-hand House Price

Group 11
Chen Taoyue 1155141543
DAI Qiyu 1155141616
JIANG Yunhui 1155141677
LI Linyuan 1155141569
MAK Man Fung 1155141893
QIN Zihao 1155124920

Contents

1 Introduction 1

2 Data 1

2.1 Data Description . . . 1

2.2 Data Splitting and Visualization . . . 2

2.3 Data Cleaning . . . 2

2.4 Data Transformation . . . 2

3 Model Building 3

3.1 Overall Workflow . . . 3

3.2 Models . . . 3

4 Variable Importance 5

5 Conclusion 6

6 Appendix 8

6.1 Data description . . . 8

6.2 Visualization . . . 10

6.3 Result . . . 11

1 Introduction

In developed Chinese cities, the property price is crucial to all potential and existing holders. According to an official article posted by YUNXIOK [1], the price of Shanghai's property is volatile, which has soared three times in the past two decades but slumped this year. Also, most of the apartments transacted in Shanghai are second-hand apartments, whose price is more unpredictable than that of new flats. Such fluctuating patterns in property prices and the lack of information hurt the stability of buyers and sellers and increase the difficulty for government to regulate.

On the other hand, thanks to the advancement in information technology, the availability of big data on housing information is tremendously increased. More importantly, we are able to invoke modern machine-learning techniques in data mining. Thus motivated, this project aims to convey insights into the Shanghai property market via a data-driven approach from two perspectives:

- Forecasting the price of pre-owned properties by regression models.
- Identifying factors that influence the price significantly.

The remainder of this report is organized as follows. We first answer questions on where our data comes from and how to process it in Section 2. Then, section 3 presents the model-building approaches. Several important determinants of house price are emphasized in Section 4, and Section 5 concludes. Additional plots and numerical results are contained in Appendix.

2 Data

2.1 Data Description

This project uses python to automatically collect the pre-owned apartments' information from the website of *Lianjia*,which is the largest online property agency in mainland China. The reasons for using such a data source are twofold. Firstly, it can provide us with a rich-data environment. After dropping the irrelevant columns, there are still 21,299 records, each described by a long vector of 22 features. For a detailed explanation of the variables, please check Appendix 6.1. Secondly, the dataset is more up-to-date than off-the-shelf data on websites like Kaggle since it is collected at the beginning of this project. Thus, real-time updates of housing information in Shanghai are included. Combined, our project can adequately reflect the underlying pattern of the current housing market.

[1] YUNXINOK: https://www.yun-xk.com/details/fj/930.html

2.2 Data Splitting and Visualization

The acquired dataset is randomly divided into two parts, with 80% training data and 20% testing data. We calculate the model parameters with training data, and then assess the model performance in the testing sample.

Before data pre-processing, we first visualize the training dataset to gain insights. Firstly, looking at the density curve for the price, it is observed that the distribution is right-skewed. Hence, a log transformation is conducted. We also compare the boxplots before and after taking the log. As shown in Figure 1 (refer to Appendix 6.2), the log transformation significantly reduces the number of outliers, which leads us to expect that it can improve the fitting of models.

2.3 Data Cleaning

Upon closely examining our dataset, we find that it contains information on two categories, with 15 columns containing categorical features and 7 columns including numerical attributes.

To deal with textual categorical features, We first filter out trivial characters. Then dummy encoding is employed to convert each categorical feature with n levels into (n-1) dummy variables. Furthermore, numerical variables are extracted by string manipulation functions. For instance, for contents like "five bedrooms two parlous one kitchen two toilets" in Chinese, four corresponding values could be obtained as features of the apartments' layout. After the above procedure, the data sample has 75 dimensions total, with 63 dummies and 12 numerical variables.

2.4 Data Transformation

We then computed the proportion of missing data for each column. The number of missing data is reasonably small or even zero for most features, which indicates the good quality of our data. Nevertheless, to avoid a waste of information, we decide to utilize the k-nearest neighbors algorithm to impute missing data instead of directly discarding them.

A suite of different k is compared in order to prevent our imputation strategy from hurting model prediction performance. That is, we feed the training data into a pipeline consisting of an imputer and a random forest regressor. Then, following the philosophy of cross-validation, the prediction error for a grid hyperparameters is recorded. Given the mean and standard deviation of negative RMSE across different folds for each k, 33 is selected. After plugging in the optimal number of neighbors, a KNN imputer is calculated on all training data and then applied to the testing set.

Also, we observe that the scales of numerical features differ significantly, which may impact the models' sensitivity to outliers. Therefore, standardization is required to address the problem. Similar to imputation, we normalize the whole dataset using the mean and standard deviation of the training set.

At the end of data pre-processing, an additional visualization step is conducted. By projecting all inputs on a three-dimensional scatter plot using principal component analysis (PCA), a crucial information is gained. As displayed in Appendix 6.2, Figure 2, our data is well-separated looking from different angles. Hence, the upcoming models are expected to perform well.

3 Model Building

3.1 Overall Workflow

Before diving into the core part of this section, we first overview the workflow comprised of three stages. Firstly, we select the model with an optimal level of complexity using a grid search with five-fold cross-validation. Then, the full training data is employed for fitting. Lastly, we evaluate the models' performance on new data points using a testing set.

When choosing the model parameters, the parameter candidate sequence is adjusted until the chosen value lies in the middle of the set rather than the boundary. Additionally, considering the huge volume of the dataset, we allow Python to run all computer processors in a parallel manner for acceleration.

In order to assess the goodness of models, three metrics are employed: R-squared, root-mean-squared-error (RMSE), and mean-absolute error (MAE). R-squared represents the proportion of variance for the dependent variable explained by the model. RMSE and MAE quantify how deviated the model prediction is from the truth. Moreover, they share the same units as the target variable and thus have an intuitive meaning. Let $\hat{y}$ and y be the estimate and true value of the forecasting target; their mathematical definitions are

$$R^2 = \frac{\sum_{i=1}^{N}(\hat{y}-\bar{\hat{y}})^2}{\sum_{i=1}^{N}(y-\bar{y})^2},\ MSE = \sqrt{\sum_{i=1}^{N}(\hat{y}-y)^2},\ MAE = \sum_{i=1}^{N}|\hat{y}-y|$$

3.2 Models

1. **OLS and Penalized Regression**

   The starting point of our model construction efforts is ordinary least squares (OLS) regression. It has the benefits of being both computationally efficient and simple to understand. However, the fitting performance of both in-sample and out-of-sample is not unsatisfactory, with an R-squared around 85%. Clearly, the reason is its inflexibility since linear restrictions are imposed.

It should be mentioned that people usually include penalized regression in machine learning projects as a common practice. However, given the current situation, we don't bother to discuss them in detail. From a theoretical point of view, the difference in OLS training and testing fitting is insignificant, implying that imposing regularization on coefficients will not help too much; empirically, as illustrated in Appendix 6.3, table 1, the advantages of LASSO, RIDGE, and Elastic Net are not obvious in terms of forecasting the testing set. Moreover, almost no parameters are shrunk exactly to zero for LASSO, meaning the interpretability is low.

2. **SVR**

   The first model for dealing with the large bias of OLS prediction is Support Vector Machine Regressor (SVR). The objective of SVR is to find a hyperplane with most instances around it as the decision boundary. Furthermore, the kernel function could be applied to implement its non-linear structure. The best kernel is selected to be Radial Basis Function using five-fold cross-validation, and the results of the tuned model show that the high bias has been alleviated to some extent.

3. **KNN**

   Furthermore, the k-nearest neighbors (KNN) algorithm is carried out to boost the in-sample fitting performance, considering the flexibility due to its non-parametric nature. This intelligent method works by directly computing the weighted mean of the k nearest neighbors for a new point. Two crucial parameters, k and the weighting scheme, are again pinned down by cross-validation. Notably, the selected k given by 1, which causes perfect in-sample fitting. However, the error on testing data is even larger than that of OLS, implying the considerable variance of such a complex model.

4. **Decision Tree**

   We then applied tree-based methods, which are among the most popular approaches in machine learning. A decision tree is implemented by doing recursive binary partitions. By default, the package in python tends to split the data until only pure leaf nodes are left, causing the instability of model prediction errors. Therefore, we similarly conducted a grid search to determine the depth and other tree properties.

   Regarding the results of the decision tree, there are two key remarks. Firstly, though tuning parameters are carefully chosen, the gap between training and testing accuracy is still remarkable. Such a high variance issue should be tackled. Secondly, there is still room for reducing in-sample error, which is attributed to the bias of the model. Being aware of these limitations, ensemble learning methods are further developed afterward.

5. **Random Forest**

   We first utilize a random forest algorithm to handle the overfitting problem of the decision tree. There, each tree is constructed independently on a bootstrap sample of input points and by selecting a random subset of all features. As for making forecasts, a data point will be passed by multiple trees, and the final decision is made by taking the average.

   Given the above bagging (bootstrap and aggregation) mechanism, random forest is hard to encounter overfitting. Therefore, tuning the depth of the tree will not be that rewarding, and we simply set it to be the optimal value of the decision tree (30). The grid search is performed on two parameters, the number of features used in each tree and the number of trees in the whole forest. From Table 1 in Appendix, random forest mitigated the overfitting problem and increased out-of-sample R-square.

6. **Boosting**

   Additionally, we experiment with three boosting algorithms. Unlike random forests, which grow multiple trees independently, boosting algorithm's implementation is sequential. Specifically, the Ada boosting regressor grows the next tree while assigning higher weights to instances with larger errors in the previous step. As for gradient boosting, it adds a new decision tree that best reduces the loss function before. Furthermore, compared with gradient boosting, the extension of XGB lies in the regularization of the estimation procedure, which controls overfitting.

   After choosing the optimal learning rate and other parameters using grid search, it is found that the Ada and XGB boosting perform well, with an R-squared over 91% on the testing set. However, gradient boosting overfits a bit.

Table 1 in Appendix 6.3 summarizes the overall performance of all models. After comparison, we claim that random forest, Ada boosting, and XGB boosting outperform the other methods. Thus, in the next section, we compute the importance of features based on these three models as they have superb prediction ability.

4 Variable Importance

In this project, feature importance could be interpreted as predictive power. Moreover, in the tree model context, it is measured by the improvement in the split criterion. Notice that if a feature participates in multiple nodes, an average should be taken.

Several most important variables are reviewed here. On the one hand, random forest and Ada boosting agree that area_size and inner are the two most essential predictors. On the other hand, XGB boosting ranks office and Chongming at the top. One possible reason why XGB holds a distinct view from the other two might be its regularization, which leads to a different objective function and/or additional constraints. Additionally, notice that area_size is a numerical variable, and the remaining four are dummies. Specifically, inner indicates whether a house is located in the inner string of Shanghai; office refers to places with commercial use instead of residential purpose; Chongming and Jianshan are names of districts. A more comprehensive comparison can be seen in Appendix 6.2, Figure 3.

5 Conclusion

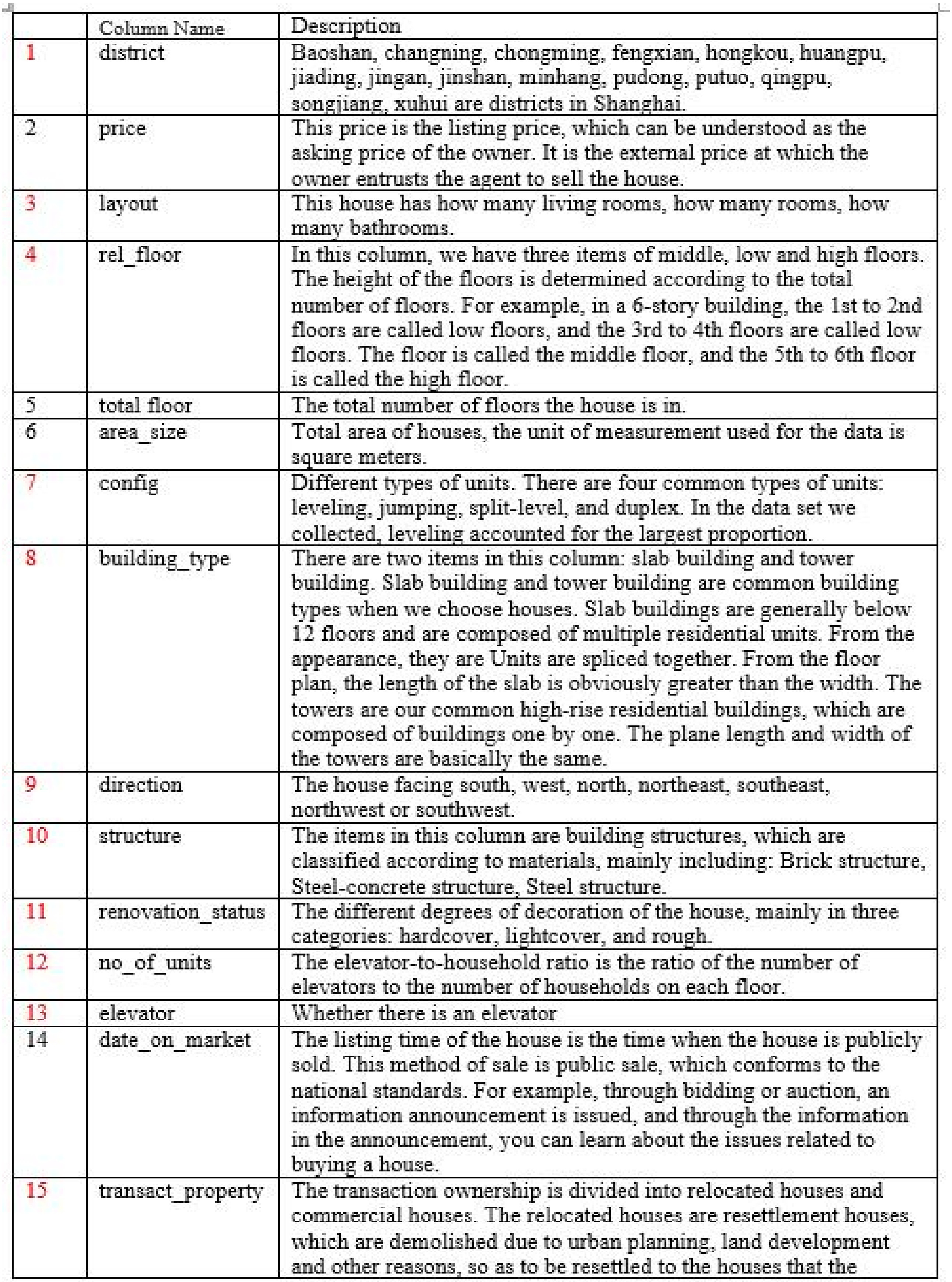
In summary, we scrapped an updated and large dataset from the Lianjia website. After data-pre-processing, the performance of the difference estimation method in predicting Shanghai housing prices is evaluated. Based on the results, we recommend three algorithms with out-of-sample R-squared of around 92%: Random Forest, Ada boosting, and XGB boosting. Finally, the most important variables are identified.

With the information provided in this project, buyers and sellers of properties in Shanghai will benefit from the mitigated difficulties of assessing apartment prices. Also, our model is worth being used as a reference by the government and private institutions like consulting firms and equity research companies when studying Shanghai real estate market issues.

Nevertheless, we note that there are two main caveats to our analysis. Firstly, information from the *Lianjia* website like apartment reviews is not incorporated. These types of textual data require deep learning methods to process, which, unfortunately, are beyond the course scope. Secondly, out of *Lianjia* website, factors such as government regulation and economic environment should also be taken into consideration. It is hoped that these limitations will be circumvented in the future.

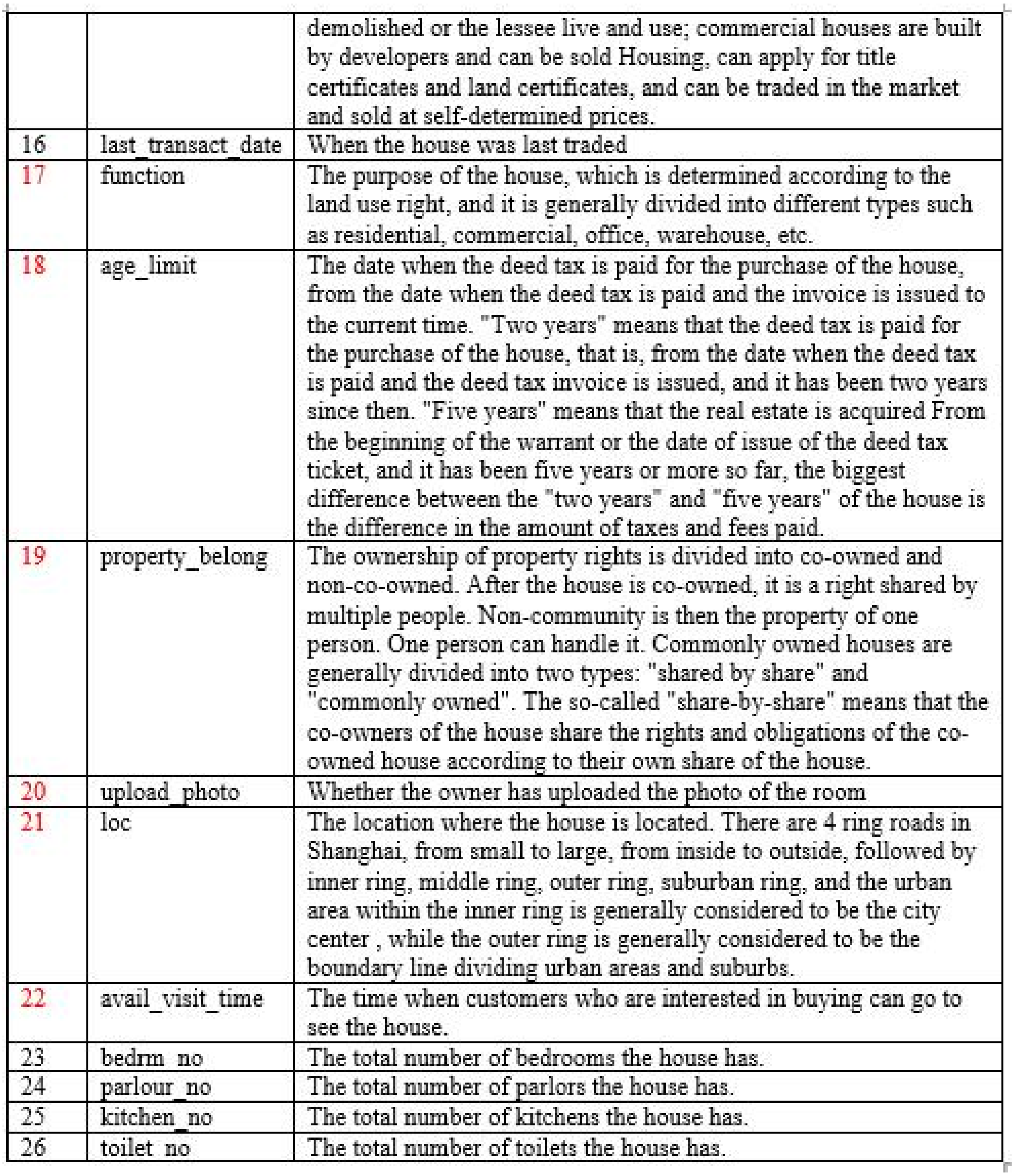
6 Appendix

6.1 Data description



	Column Name	Description
1	district	Baoshan, changning, chongming, fengxian, hongkou, huangpu, jiading, jingan, jinshan, minhang, pudong, putuo, qingpu, songjiang, xuhui are districts in Shanghai.
2	price	This price is the listing price, which can be understood as the asking price of the owner. It is the external price at which the owner entrusts the agent to sell the house.
3	layout	This house has how many living rooms, how many rooms, how many bathrooms.
4	rel_floor	In this column, we have three items of middle, low and high floors. The height of the floors is determined according to the total number of floors. For example, in a 6-story building, the 1st to 2nd floors are called low floors, and the 3rd to 4th floors are called low floors. The floor is called the middle floor, and the 5th to 6th floor is called the high floor.
5	total floor	The total number of floors the house is in.
6	area_size	Total area of houses, the unit of measurement used for the data is square meters.
7	config	Different types of units. There are four common types of units: leveling, jumping, split-level, and duplex. In the data set we collected, leveling accounted for the largest proportion.
8	building_type	There are two items in this column: slab building and tower building. Slab building and tower building are common building types when we choose houses. Slab buildings are generally below 12 floors and are composed of multiple residential units. From the appearance, they are Units are spliced together. From the floor plan, the length of the slab is obviously greater than the width. The towers are our common high-rise residential buildings, which are composed of buildings one by one. The plane length and width of the towers are basically the same.
9	direction	The house facing south, west, north, northeast, southeast, northwest or southwest.
10	structure	The items in this column are building structures, which are classified according to materials, mainly including: Brick structure, Steel-concrete structure, Steel structure.
11	renovation_status	The different degrees of decoration of the house, mainly in three categories: hardcover, lightcover, and rough.
12	no_of_units	The elevator-to-household ratio is the ratio of the number of elevators to the number of households on each floor.
13	elevator	Whether there is an elevator
14	date_on_market	The listing time of the house is the time when the house is publicly sold. This method of sale is public sale, which conforms to the national standards. For example, through bidding or auction, an information announcement is issued, and through the information in the announcement, you can learn about the issues related to buying a house.
15	transact_property	The transaction ownership is divided into relocated houses and commercial houses. The relocated houses are resettlement houses, which are demolished due to urban planning, land development and other reasons, so as to be resettled to the houses that the

(a) Data Description

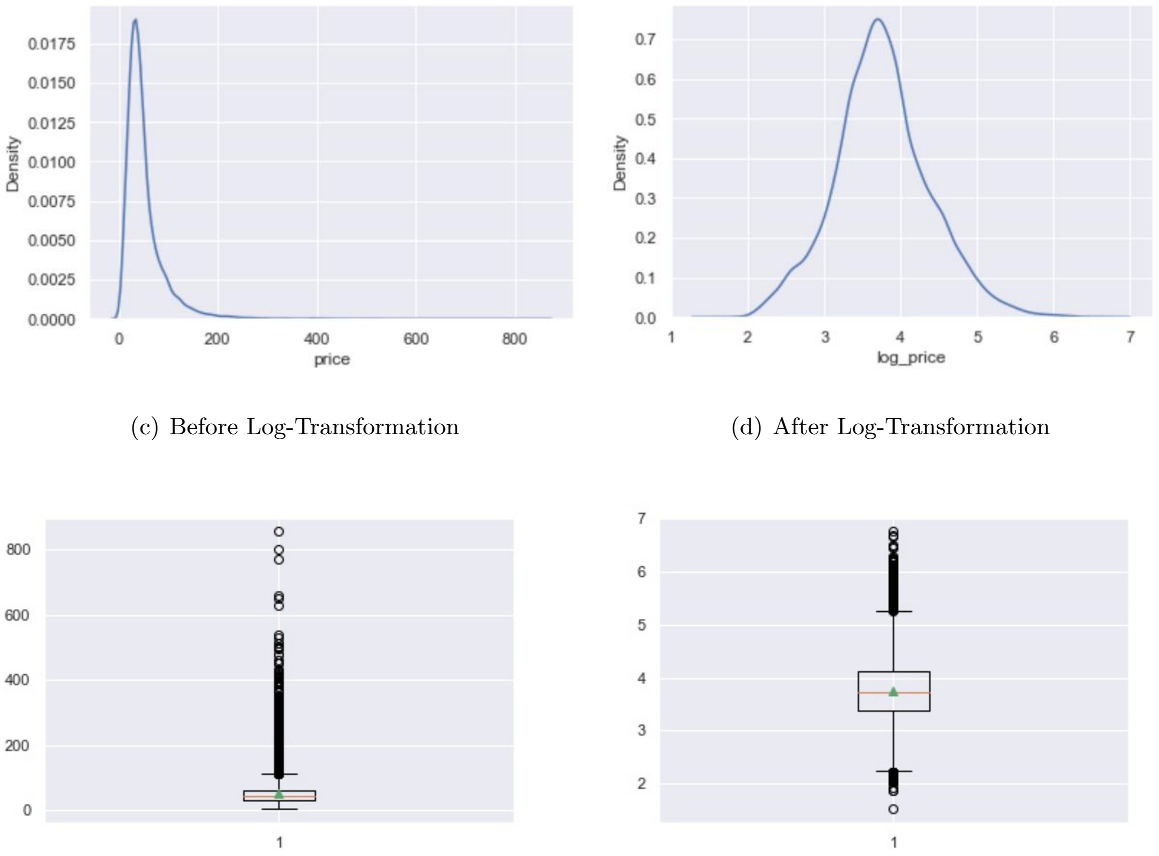


		demolished or the lessee live and use; commercial houses are built by developers and can be sold Housing, can apply for title certificates and land certificates, and can be traded in the market and sold at self-determined prices.
16	last_transact_date	When the house was last traded
17	function	The purpose of the house, which is determined according to the land use right, and it is generally divided into different types such as residential, commercial, office, warehouse, etc.
18	age_limit	The date when the deed tax is paid for the purchase of the house, from the date when the deed tax is paid and the invoice is issued to the current time. "Two years" means that the deed tax is paid for the purchase of the house, that is, from the date when the deed tax is paid and the deed tax invoice is issued, and it has been two years since then. "Five years" means that the real estate is acquired From the beginning of the warrant or the date of issue of the deed tax ticket, and it has been five years or more so far, the biggest difference between the "two years" and "five years" of the house is the difference in the amount of taxes and fees paid.
19	property_belong	The ownership of property rights is divided into co-owned and non-co-owned. After the house is co-owned, it is a right shared by multiple people. Non-community is then the property of one person. One person can handle it. Commonly owned houses are generally divided into two types: "shared by share" and "commonly owned". The so-called "share-by-share" means that the co-owners of the house share the rights and obligations of the co-owned house according to their own share of the house.
20	upload_photo	Whether the owner has uploaded the photo of the room
21	loc	The location where the house is located. There are 4 ring roads in Shanghai, from small to large, from inside to outside, followed by inner ring, middle ring, outer ring, suburban ring, and the urban area within the inner ring is generally considered to be the city center , while the outer ring is generally considered to be the boundary line dividing urban areas and suburbs.
22	avail_visit_time	The time when customers who are interested in buying can go to see the house.
23	bedrm_no	The total number of bedrooms the house has.
24	parlour_no	The total number of parlors the house has.
25	kitchen_no	The total number of kitchens the house has.
26	toilet_no	The total number of toilets the house has.

(b) Data Description

The serial numbers of some columns are marked in red, indicating that they are categorical features and will be transformed to dummy variables after data processing.

6.2 Visualization



(c) Before Log-Transformation

(d) After Log-Transformation

(e) Before Log-Transformation

(f) After Log-Transformation

Figure 1: Density plot of price and Bloxplot of Price

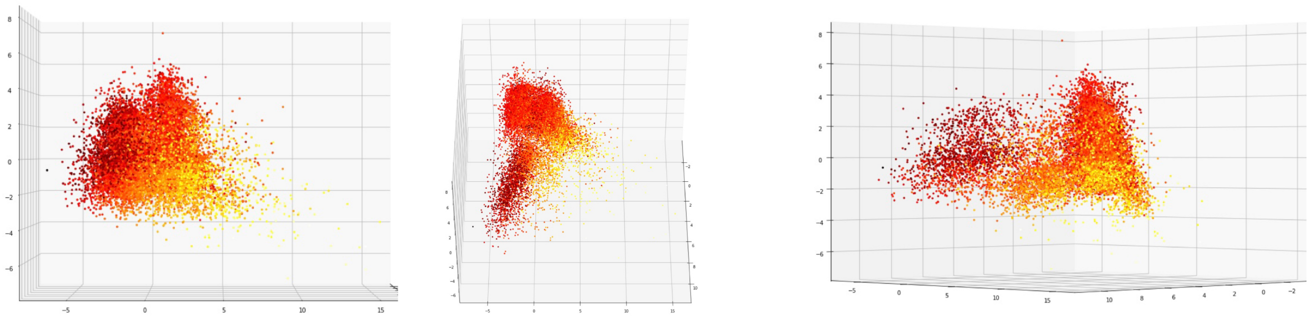
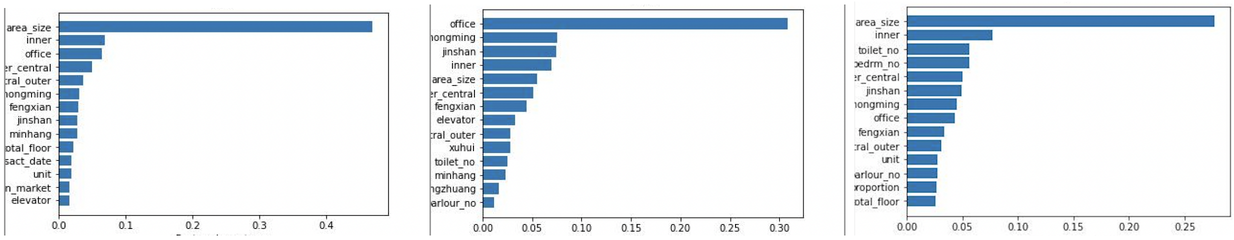


Figure 2: principal component analysis



(a) Ada boosting

(b) XGB boosting

(c) Random Forest

Figure 3: Feature Importance

6.3 Result

	R-squared		**RMSE**		**MAE**	
	training	**testing**	**training**	**testing**	**training**	**testing**
OLS	**0.8516**	**0.8450**	0.2476	0.2557	0.1886	0.1948
Lasso	**0.8515**	**0.8455**	**0.2476**	**0.2554**	**0.1886**	**0.1946**
Ridge	**0.8516**	**0.8454**	**0.2476**	**0.2554**	**0.1886**	**0.1946**
Elastic_net	0.8454	0.8365	0.2527	0.2627	0.1891	0.1962
KNN	1	0.7901	0.0	0.2976	0.0	0.2271
SVR	0.9429	0.8888	0.1535	0.2166	0.1134	0.1541
Decision Tree	0.9206	0.8741	0.1811	0.2305	0.137	0.1679
Random Forest	0.9892	0.9172	0.0667	0.187	0.0484	0.1345
Ada boosting	0.9998	0.9191	0.0101	0.1848	0.0029	0.1269
Grandient boosting	0.9954	0.8584	0.0436	0.2445	0.0336	0.1709
XGB boosting	0.9867	0.9197	0.0742	0.1841	0.0584	0.1334

Table 1: Overall results