Beijing Housing Price Prediction Using Random Forest Regression

Final report for DATA1030, Fall 2021 at Brown University

Ying Sun DSI

Github link: https://github.com/YingSun0314/YingSun_DATA1030_project.git

1. Introduction

Beijing's housing prices have always been one of the concerns of the people of the whole country and even scholars all over the world. Because Beijing's fast-developing economy, rich history, and functions of the capital make many people want to come here to work and live. However, due to the impact of policies, regions, and apartment types, etc., Beijing's housing prices have risen and fallen significantly. If Beijing housing prices can be predicted, it can not only guide the government to specify policies, but also allow people to buy reasonably priced houses or invest.

In order to predict Beijing's housing prices, I used data from Lianjia, Beijing's largest housing trading platform. There are 297701 data points and 19 features in the dataset. The type and description is as below. I choose Total Price as my target variable, whose type is float64, and plan to do a regression to predict the house price using the target variable and all features dependent to it.

Feature Name	Type	Description
tradeTime	Int32	the time of transaction
followers	Int64	the number of people follow the transaction
price	Int64	the average price by square meter
Square	Float64	the square of house
livingRoom	Float64	the number of livingroom
drawingRoom	Float64	the number of drawingroom
kitchen	Int64	the number of kitchen
bathRoom	Float64	the number of bathroom
floor	Float64	the height of the house.
buildingType	Float64	including tower(1) , bungalow(2), combination of plate and tower(3), plate(4).
constructionTime	Float64	the time of construction
renovationCondition	Int64	including other(1), rough(2),Simplicity(3), hardcover(4)
buildingStucture	Int64	including unknow(1), mixed(2), brick and wood(3), brick and concrete(4),steel(5) and steel-concrete composite (6).
ladderRatio	Float64	the proportion between number of residents on the same floor and number of elevator of ladder. It describes how many ladders a resident have on average.
elevator	Float64	have (1) or not have elevator(0)
fiveYearsProperty	Float64	if the owner have the property for less than 5 years (0), more than 5 years (1)
subway	Float64	Have (1) or not have (0)
district	Int64	Dongcheng(1), Fengtai(2), Tongzhou(3), Daxing(4), Fangshan(5), Changping(6), Chaoyang(7), Haidian(8), Shijingshan(9), Xicheng(10), Pinggu(11), Mentougou(11), Shunyi(12)
communityAverage	Float64	The average housing price in the community

Table 1. Feature Description

Since it is a well-documented and comprehensive database, many people have studied this before and the previous work using this dataset are posted on Kaggle. Younes(2019) found building types, building structure, renovation, five-year policy, and district all influence the price of the house in Beijing. The time series forecasting intuitively and clearly shows the trend of Beijing housing prices over time, and making the monthly and annual price changes obvious. But he only analyzed the relationship between the target variable, housing price, and the single feature. He also didn't predict the house price in Beijing. There is no any numeric result in his research. Another Kaggle Expert, Vignesh, focus on regression with Scikit-Learn, including linear regression, Robust regression, Ridge regression, and Lasso regression. However, he just introduced how the methods fit the dataset, but didn't do data processing at all.

2. Exploratory Data Analysis

In this part, I deal with some values that doesn't make sense or was selected wrong, and generate a description of each column. Then I focus on columns. I perform EDA on each column, two columns of a feature and the target variable, based on the type of the column, and a scatter matrix.

2.1. EDA on each column

I firstly create two lists for both continuous and categorical data. Note that followers (number of people follow the information of the house) is not continuous technically, while I process them as continuous to see the trend clearly.

For the target variable, Total Price, I graph another histogram with log scaling to show all of the data clearly. I also apply this to other plots in this part if most of the data are concentrated to a specific portion. The Figure 1 below indicates that this is a right-tailed distribution of total price, and most of house price are between 1.5 to 4 million yuan.

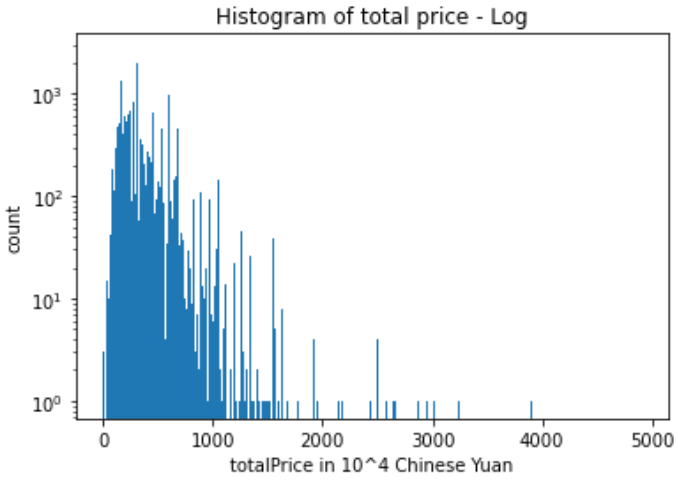

Figure 1. Histogram of Total Price with Log Scaling

2.2. EDA on two columns and Scatter matrix

Same in the part 2.1., I separate features into two groups, categorical and continuous. Note that the target variable is continuous. Therefore, for continuous target variable and a categorical feature, I can use violin plot, box plot and category-specific histograms. Box plot is better here because it clearly shows the percentile of the data and the position of the outliers. For continuous target variable and a continuous feature, I can use scatter and heatmap. I also generated a scatter matrix, which shows the relationship between each continuous feature. For example, Figure 2 indicates that the price of the house with elevator is only a little bit higher than that of the house with no elevators. Figure 3 shows the relationship between total square area and the total price. The graph indicates that there is a positive relationship between total square area and total price, the less the total square area, the correlation stronger.

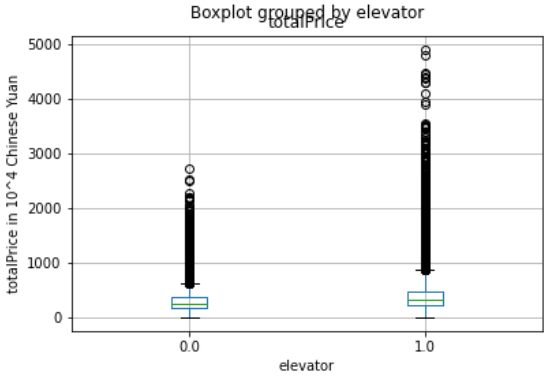

Figure 2. Bar plot of Elevator

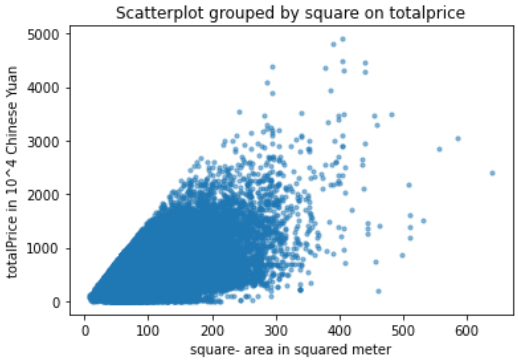

Figure 3. Scatter plot of Square

3. Methods

3.1. Data Splitting and Preprocessing

The dataset is IID, independent and identically distributed. Because each data point are selected from the same generative process and independent to each other, based on the creator of this dataset, Tue. The dataset creator also mentioned the dataset is iid.

The dataset has no group structure. Because all samples are selected in the same way. It is a time-series data. Because there is a correlation between the house price and the traded date.

Generally, I use the basic split to split the dataset with test size is 20%, then use KFold split with the number of split 4 for the other 80% of the data, because it is a big dataset and I could do k-fold cross validation later in the ML-models part. Since the dataset is balanced generally since my goal is to predict the house price. Splitting data helps to find the best hyper-parameters of my ML algorithms and show how the model will perform on previously unseen data by applying the model on the final set. Based on the target variable, total price, I use this split method. If I study the features, I probably use other split methods. For example, if I want to predict the district given all other variables, I will use group shuffle split and KFold, since I want to make sure that district in test groups are not in the other groups to avoid the bias.

Then, I apply scalers and encoders on features based on the type of them. 19 features in total are in the preprocessed data. For all continuous features, I used Standard Scalar. Because there is not reasonably bounded for tradeTime, followers, price, square, ladderRatio, and community Average, so using MinMax Scalar might be wrong. For example, there could be 10,000 followers tracking information on one popular house despite the fact that a house usually has 20 followers on average. For categorical features, I used One Hot Encoding for buildingType, renovationCondition, buildingStructure, elevator, fiveYearsProperty, subway, and

district. Because the categories of each feature are not ordered or can be ranked. For livingRoom, drawingRoom, kitchen, bathroom, constructionTime, and floor, I use ordinal encoding, because they have a natural rank ordering. After doing this, I get transformed data in training, validation, and test groups.

3.2**. Model Selection**

Using the splitting and preprocessing mentioned in part 3.1, four machine learning models were trained: linear regression with three regularizations, random forest, support vector regression, and K-Neighbors regression. All of them were hyperparameter tuned and returned best models and test scores and repeated on 10 random states for repeatable testing and to measure the variance and uncertainty with the model prediction. The name of models, parameters tuned and values tried for each model are as below:

Name of Models	**Parameters**
linear regression with l1 regularization	alpha: np.logspace(-7,0,5)
linear regression with l2 regularization	alpha: np.logspace(-7,0,5)
linear regression with an elastic net	alpha: np.logspace(-7,0,5)
	l1_ratio: [0.1,0.2,0.3]
random forest	max_depth: [1,5,10,30]
	min_samples_split: [2,6,10,16]
support vector regression	C: [0.01, 0.1, 1, 10]
	gamma: [0.001, 0.01, 0.1, 1]
K-Neighbors regression	n_neighbors: [1,10,30,100]

Table 2. Parameters used for tuning of each model

The metric I used to evaluate my models performance is root mean square error (RMSE), because it penalizes the large prediction errors vi-a-vis Mean Absolute Error (MAE) and it is better than mean square error (MSE) because it has the same units as the dependent variable. The uncertainties due to splitting are if the seed or random variables change, the result will change. There is also uncertainty due to non-deterministic ML methods I used, random forest. The model is sensitive and will generates variable results. If we start the same random forest learning algorithm, with the same datasets, at different times, we will get different forests. This is because the algorithm selects random subsets of the features and/or datapoints to learn on, and, if different seeds are used, the subsets will be different each time. (John, 2019)

4. Result

I chose the baseline as the average of the target value. The baseline models returned an average RMSE score of 272.48, and the standard deviation of it is 5.52. Random forest model was the most predictive. The performance of the ML models including the pipeline is as below. The random forest model returned the average RMSE score of 23.98, and standard deviation of that is 5.72. Comparing them, my model achieved an RMSE that is 45 standard deviations below the baseline.

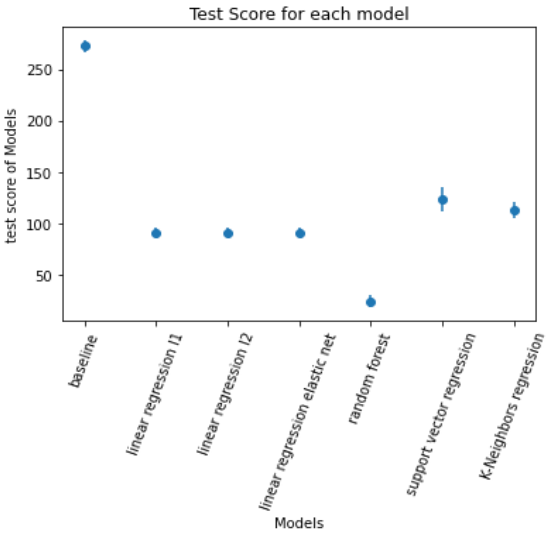

Figure 4. RMSE scores for each model and baseline

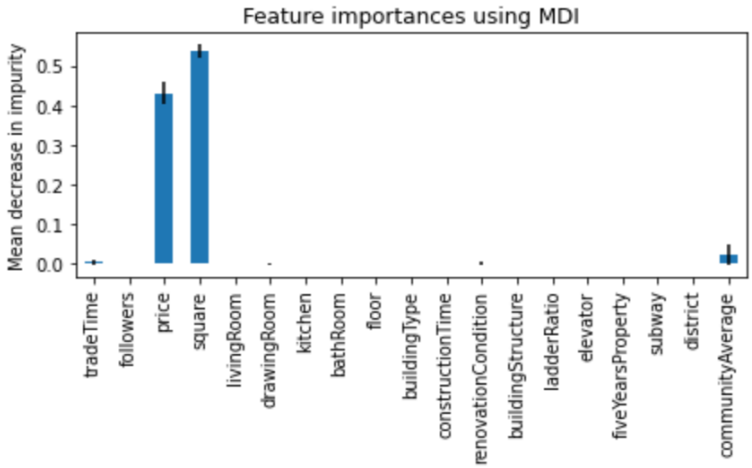

Figure 5. Feature importance using MDI

I also calculated three different global feature importance for random forest using permutation, mean decrease in impurity, and random forest regressor's feature importance. I found that all of them reveals the higher similar results, that square and price are the most important features, and community average is relatively small important. For other features, they are not important so least important feature is not meaningful. I also calculate SHAP values for local feature importance. The results indicate that the global importance using SHAP method is highly similar as the three global importance results. The other thing is that not the larger the house, the more important the area is, based on the figure below. The unexpected thing is that the importance of district is so low. The fact is that, in Beijing, the resources for education and work are extremely unbalanced, and housing prices in different districts are therefore very different. But the results of my model show that district is not one of the important factors affecting Beijing's housing prices.

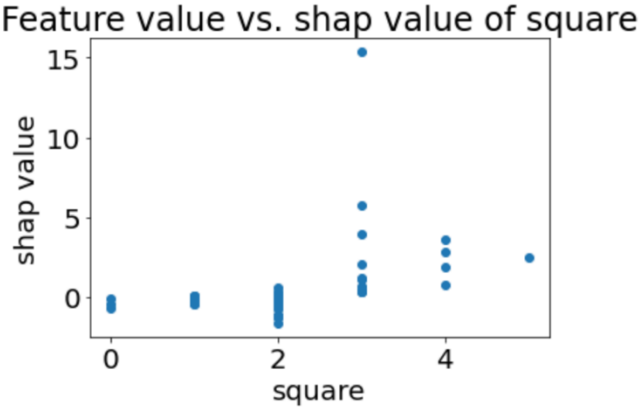

Figure 6. Feature value VS shap value of square

5. Outlook

The result shows that price per square meter of the house and the total size of the house are two important feature deciding Beijing housing price. This makes sense because most property markets use these attributes for pricing. Also, the importance of features like which district the house is in does not indicate by the models. The possible improvement is to do the feature selection and only keep some valuable features to analyze.

The weakness of the K-neighbors model is that, due to the big number of data points, I could try several relative larger numbers of neighbors like 70, 200, to find the best parameter.

The additional techniques I could have used are Decision Tree Regression and neutral network regression for this regression problem. Also, the data like the salary of people living in Beijing in each month, and management fees, decoration fees, taxes of the house, could be collected to prove the model performance.

6. Reference

Qichen Qiu, Housing price in Beijing, 2018, Kaggle, url = https://www.kaggle.com/ruiqurm/lianjia

Youns, Housing price in Beijing EDA/ARIMA, 2019, Kaggle, url = https://www.kaggle.com/eraw0x/house-prices-in-beijing-eda-arima

Aadhav Vignesh, Regression with Scikit-Learn: Practical ML #1, 2020, Kaggle, url= https://www.kaggle.com/aadhavvignesh/regression-with-scikit-learn-practical-ml-1

John Doucette, Are decision tree learning algorithms deterministic? 2019. url = https://ai.stackexchange.com/questions/11576/are-decision-tree-learning-algorithms-deterministic

Github link: https://github.com/YingSun0314/YingSun_DATA1030_project.git