Beijing House Pricing Prediction

Springboard Capstone Project 1

Nancy Mao

Contents

The Problem

The Beijing housing market is highly competitive. Figuring out the best selling price is the number one decision to make when someone is going to sell a house. The purpose of the project is to build machine learning regression model to predict house pricing in Beijing, China.

Data Set

The Beijing house price data from 2011 to 2017 fetched from lianjia.com can be found on Kaggle: House Price in Beijing.

Data Cleaning

Data Set

The data set has 26 columns with each row represents a particular house transaction.

Columns Included in original dataset:

- url
- id
- Lng
- Lat
- Cid
- tradeTime
- DOM
- followers
- totalPrice
- price
- square
- livingRoom
- drawingRoom
- kitchen
- bathRoom
- floor
- buildingType
- constructionTime
- renovationCondition
- buildingStructure
- ladderRatio
- elevator
- fiveYearsProperty
- subway
- district
- communityAverage

Data Wrangling

The following steps are performed in the data cleaning process:

- Review column names and rename as needed (some original column names are misleading)
- Clean and set index column
- Visualize and find columns have the most missing values
- Convert Chinese characters to English
- For numerical variables:
  - cleaned and performed imputation on missing values
- For categorical variables
  - Split column that has more than one features into separate columns
  - Cleaned and performed imputation on missing values
  - Created labels as needed
  - Converted to categorical dtype where applicable
- Detect and deal with outliers

Missing Values and Outliers

Imputation of Missing Value

- Missing values mainly appear in four columns: **construction time**, **building type**, **community average** and **DOM**
- construction time, building type: Filled missing values with the mode by each community
- community average (price per sq. meter) : Filled missing values with most recent value by community for each year
- DOM: Filled missing values with the mode by community by year

Outliers

- Extreme values appear in two columns: **ladder ratio** and **total price**.
- Only 2 lines showing as outlier in ladder ratio. Removed those lines from the data set
- Total price can be validated based on the product of price per sq. meter and total sq. meters of the house. I compared the original total price values with the calculated values. Then removed lines that have significant differences between the two values.

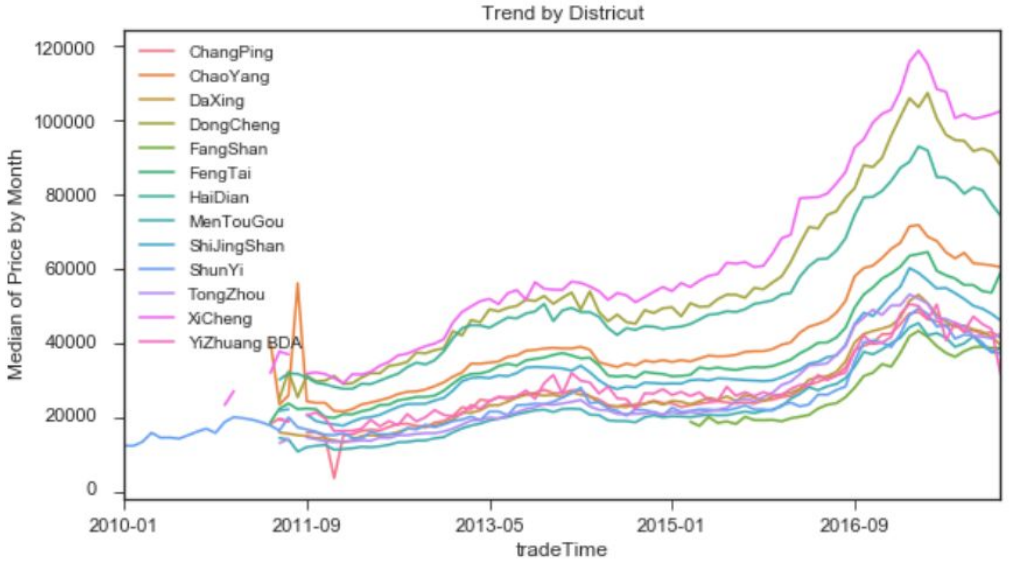
Dataset Summary

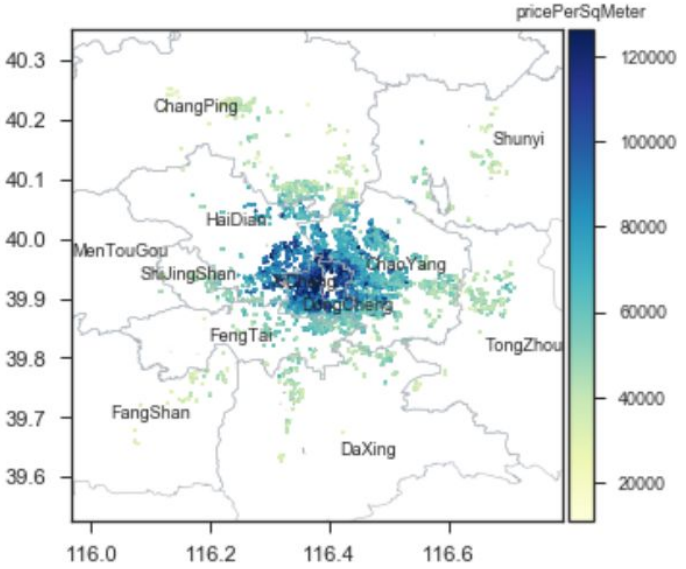
- Number of lines remained: 316, 222
- 2, 622 (0.8%) lines are dropped during the data cleaning process.

Numeric Variables	Categoricals	Objects	DateTime	Targets
Lng Lat Cid DOM followers squareMeters bedRoom livingRoom kitchen bathRoom buildingType constructionTime buildingStructure ladderRatio communityAverage floorPosition buildingFloors	buildingType renovationCondition buildingStructure elevator fiveYearsProperty subway district floorPosition	url	tradeTime	totalPrice pricePerSqMeter

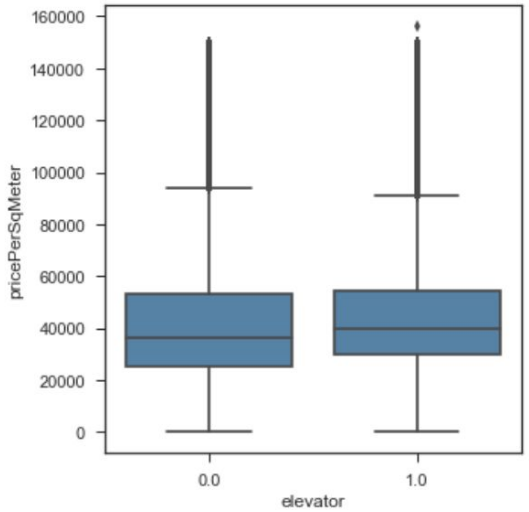
EDA and Initial Findings

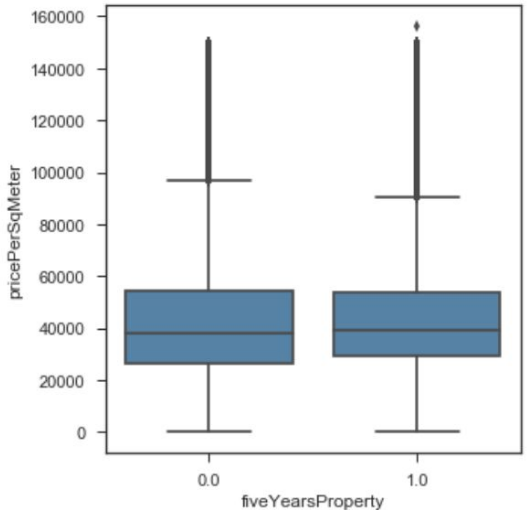
House Price Price by District

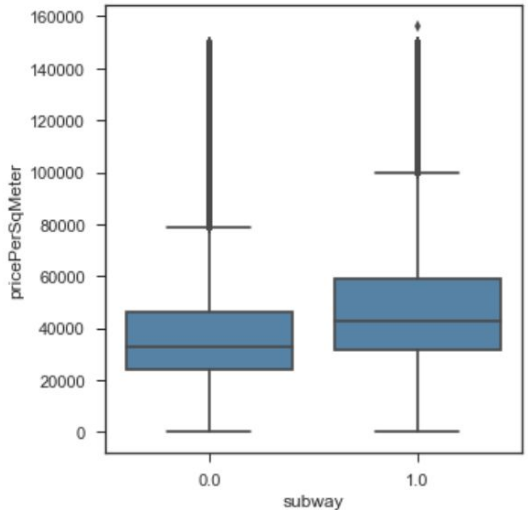




Elevator, Five Years Property, Subway







elevator

Permutation Test	P-value: 0.0000
Bootstrap Test	P-value: 0.0000
Mann-Whitney Test	P-value: 0.0000
Welch's T-test	P-value:1.97e-172

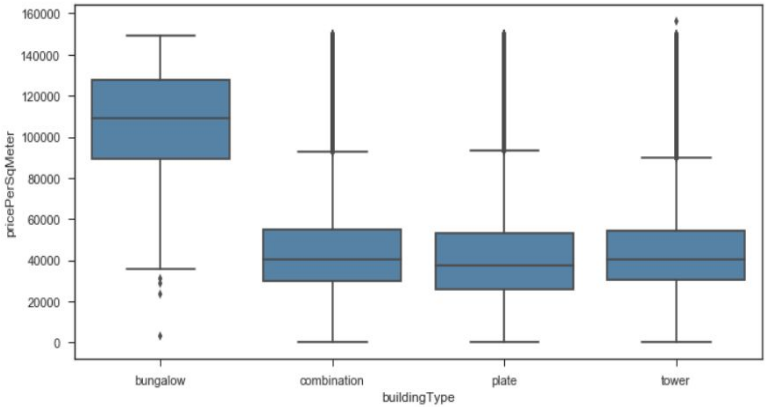
fiveYearsProperty

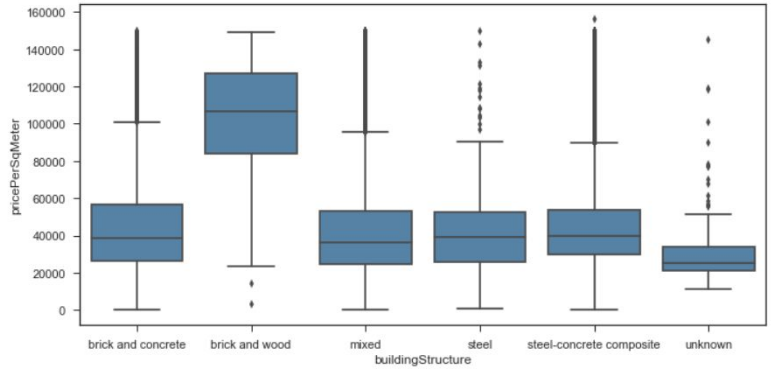
Permutation Test	P-value: 0.0000
Bootstrap Test	P-value: 0.0000
Mann-Whitney Test	P-value: 2.14e-116
Welch's T-test	P-value: 1.54e-14

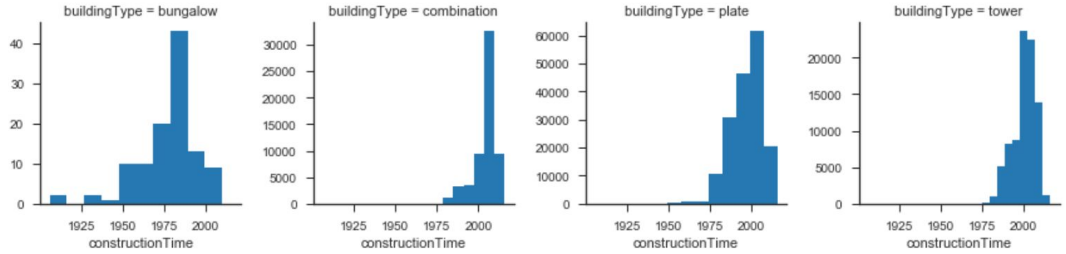
Subway

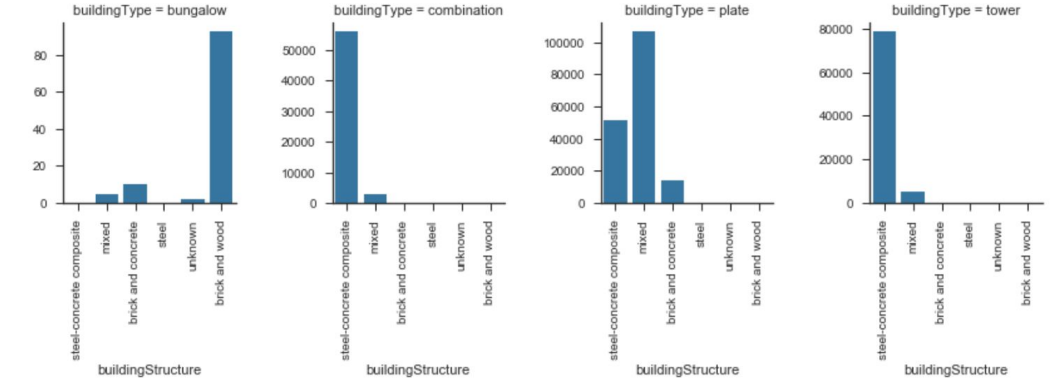
Permutation Test	P-value: 0.0000
Bootstrap Test	P-value: 0.0000
Mann-Whitney Test	P-value: 0.0000
Welch's T-test	P-value: 0.0000

Building Type and Building Structure

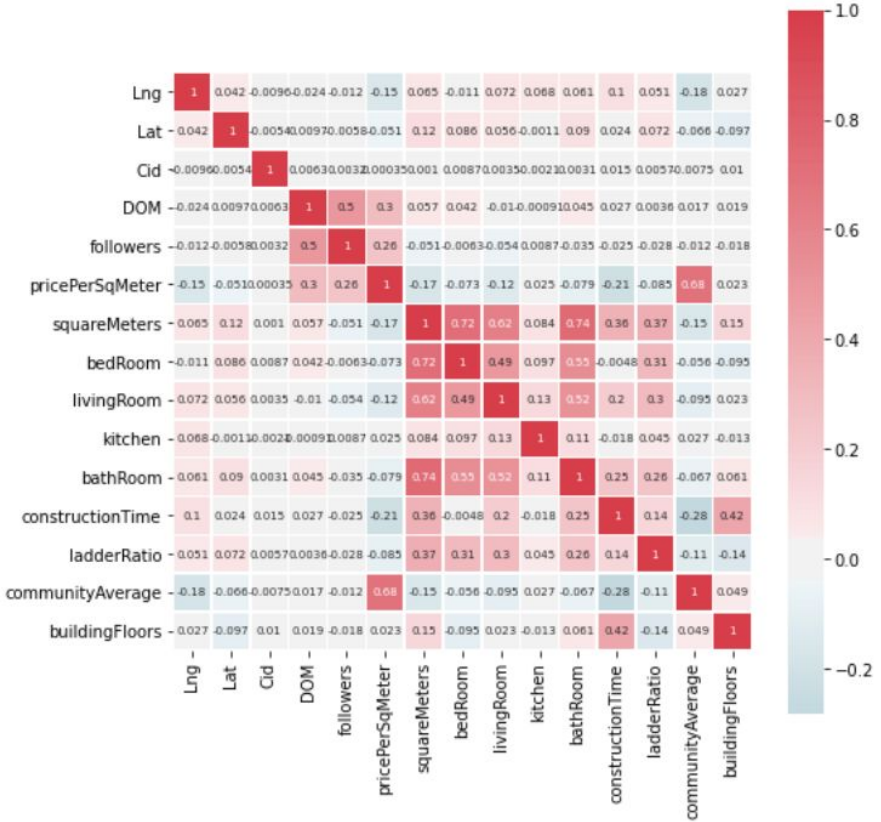








Correlation Heat Map



Modeling

Feature Engineering and Selection

- New features created
  - 'FloorNumber': Indicates which floor the house is at. Estimated based on 'floorPosition' and 'buildingFloors'.

- Feature selection
  - 19 features are included to build the models
  - 'Lng', 'Lat', 'tradeTime', 'DOM', 'followers','bedRoom', 'livingRoom', 'kitchen', 'bathRoom', 'buildingType', 'constructionTime', 'renovationCondition', 'buildingStructure', 'elevator', 'fiveYearsProperty', 'subway', 'district', 'communityAverage', 'floorNumber

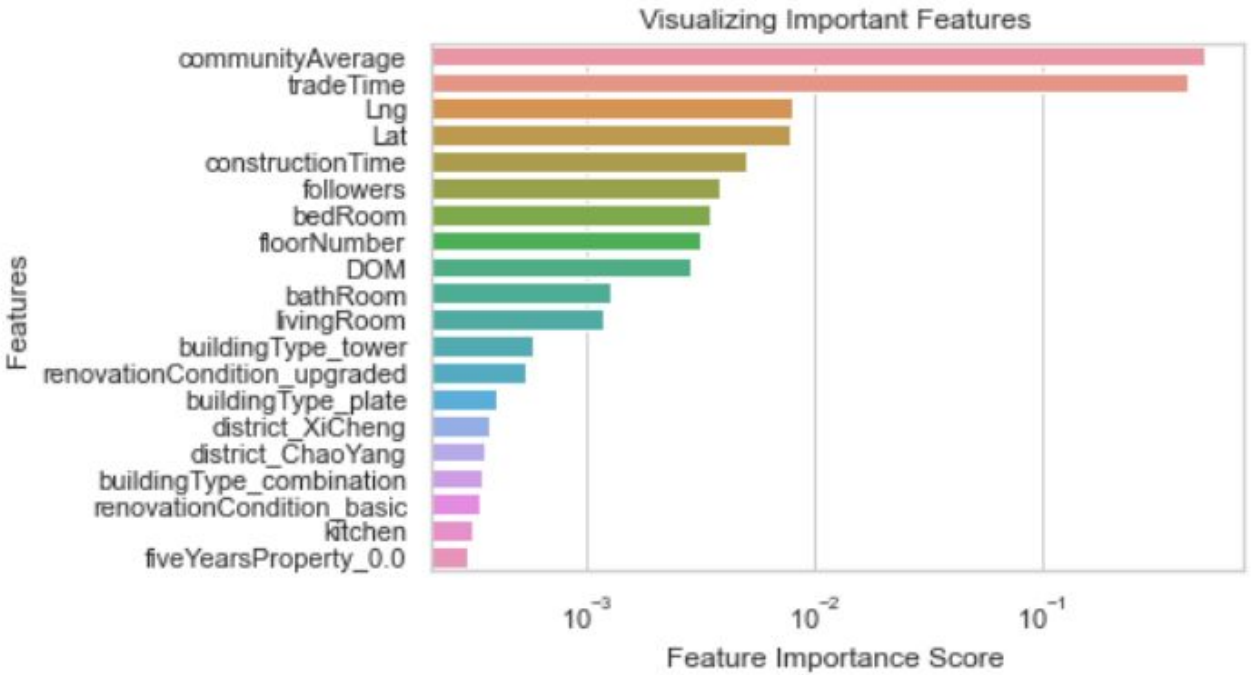
Model Comparison and Performance

- Model Process
  - Train-Test-Split data
  - Scale numeric features (StandardScaler)
  - Encode categorical features (OneHotEncoder)
  - Build models
    - Linear Regression
    - Ridge Regression
    - Lasso Regression
    - Linear SVR
    - **Random Forest**
    - Gradient Boosting
  - Evaluate and choose best-performance model
  - Tune parameters on the chosen model
  - Evaluate

	model	train_RMSE	test_RMSE
0	LR	8944.587	8962.582
1	RIDGE	8941.056	8958.310
2	LASSO	22216.348	22037.518
3	LSVR	10765.068	10709.196
4	RF	3710.491	5204.277
5	GB	1338.953	5150.458

Chosen model

Feature Importance



Parameter Tuning

Hyper Parameter	Best Params
'max_depth'	40
'min_samples_leaf'	2
'n_estimators'	100

Model Performance with Tuning:

- RMSE on test set: 5103.17