STAT4011 Project 1

Forecasting Shanghai House Price

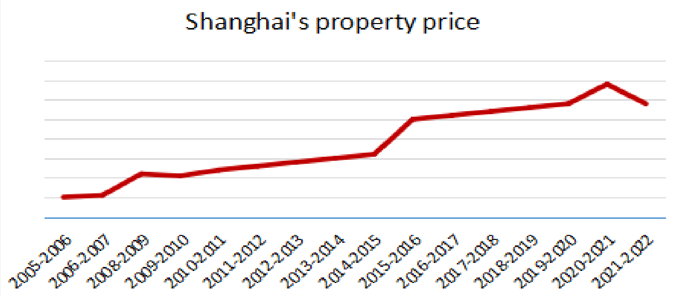
Group 11
DAI Qiyu 1155141616
CHEN Taoyue 1155141543
JIANG Yun Hui 1155141677
LI Linyuan 1155141569
MAK Man Fung 1155141893
QIN Zihao 1155124920

1. Introduction

Introduction

Motivation

- **The second-hand property price increased significantly in Shanghai**



- **Difficult for buyers and sellers to estimate price of second-hand property**

Introduction

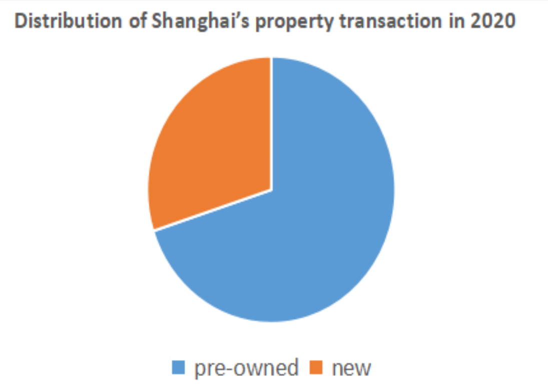
Motivation

Objective

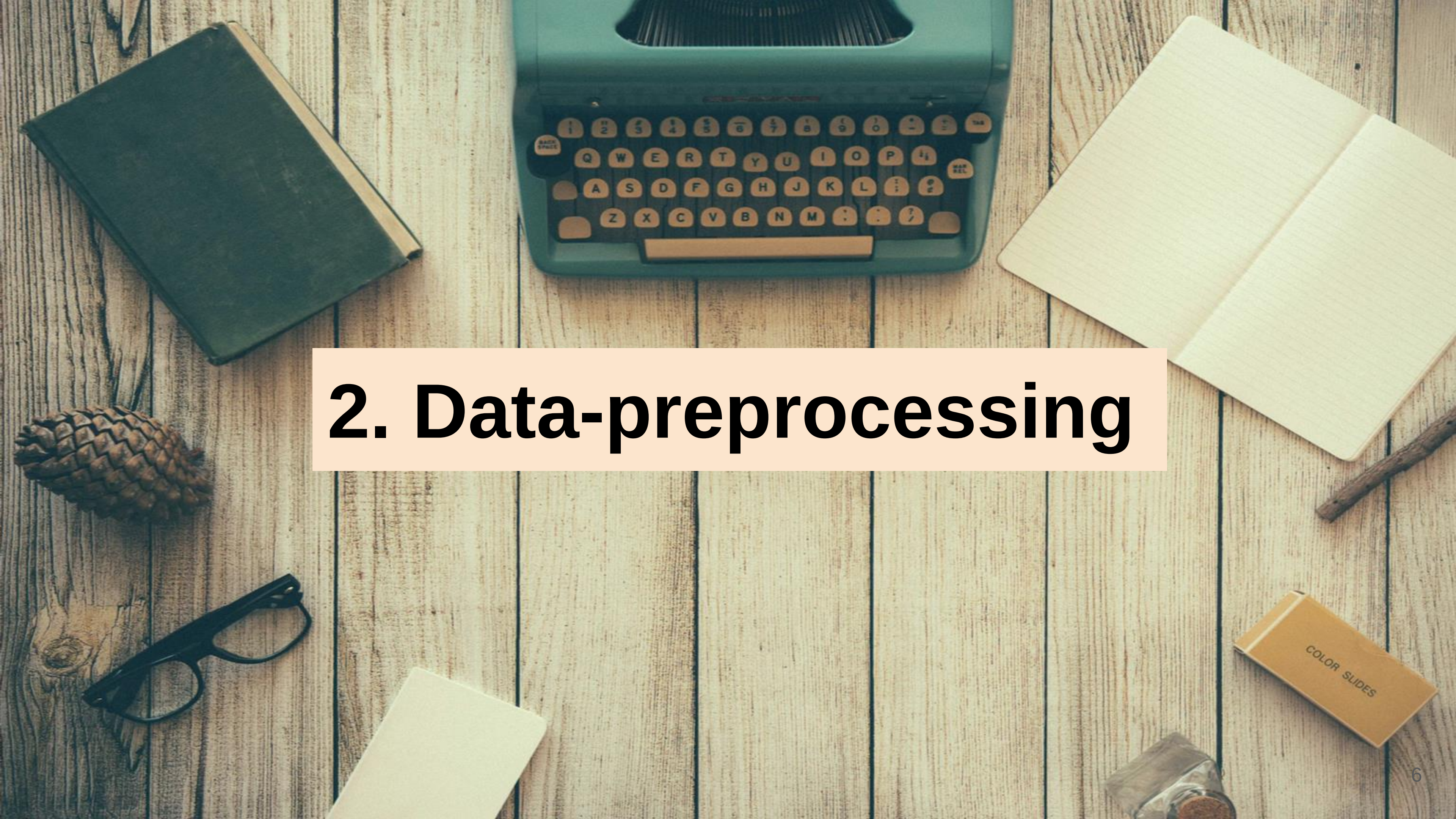
- **Utilize Shanghai second hand property dataset from *Lianjia* to**

  1. Forecaste the price of second hand property by using regression analysis

  2. Identify factors which influence the price significantly



Contents



2. Data-preprocessing

Data-preprocessing

Data Description

- ### Data Source

  We use python to automatically scrap it from *Lianjia.com*

- ### Advantage of our data

  1. Large Volumn - 21299 records, each described by 22 features
  2. Updated - on the market of the nearest twenty days

Data Splitting

Randomly select

80% → Training dataset

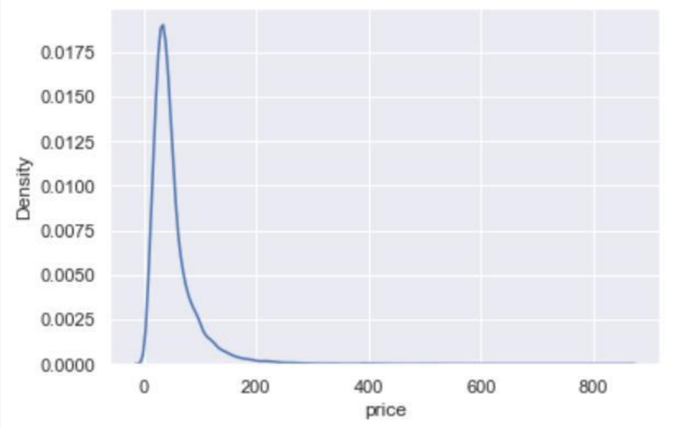
20% → Testing dataset

Data-preprocessing

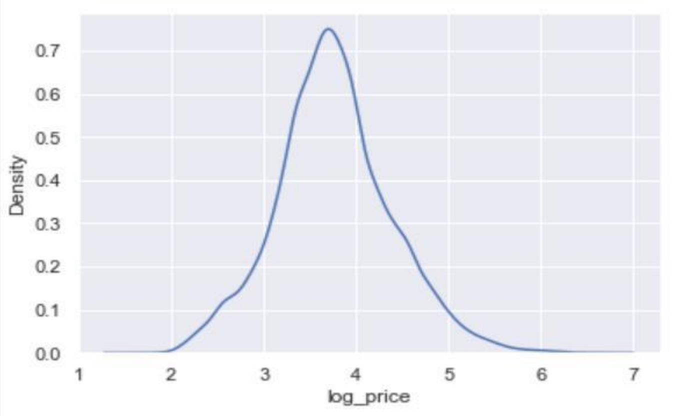
Visualization

- **Density plot of price**

Using log-transformation to reduce the skewness



Take log



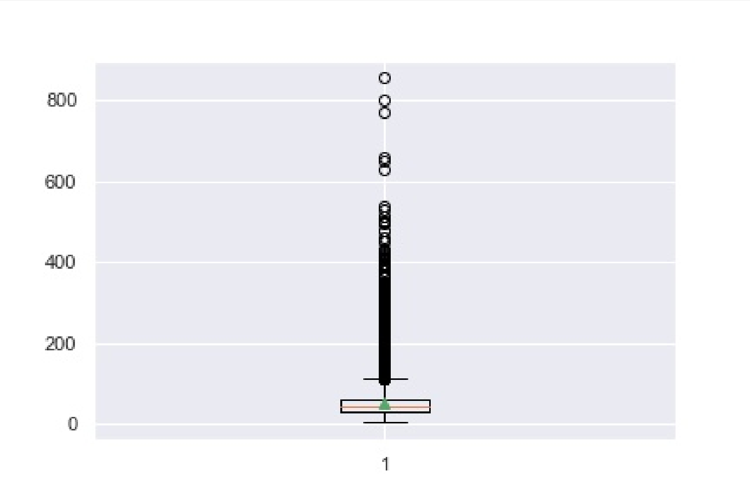
Data-preprocessing

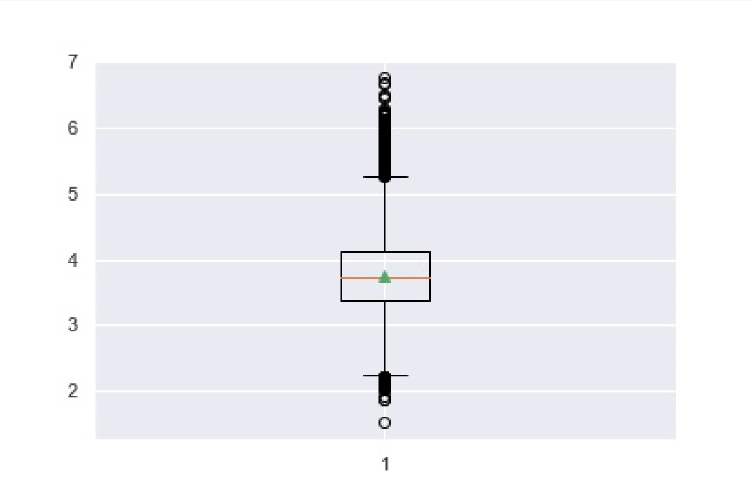
Visualization

- Density plot of price

- Bloxplot for Price

  The number of outliers is reduced after log-transformation





Data-preprocessing

Data Transformation

- **Dummy encoding for 15 categorical features -> 63 dummy factors**

  A categorical variable with n features -> (n-1) dummy variables

id	X
1	a
2	b
3	c

Dummy encoding

id	a	b
1	1	0
2	0	1
3	0	0

- **Extraction from 7 columns of text data -> 12 numerical features**

  E.g: extracting (5, 2, 1, 2) from text information “five rooms two living rooms one kitchen two toilets”

Data-preprocessing

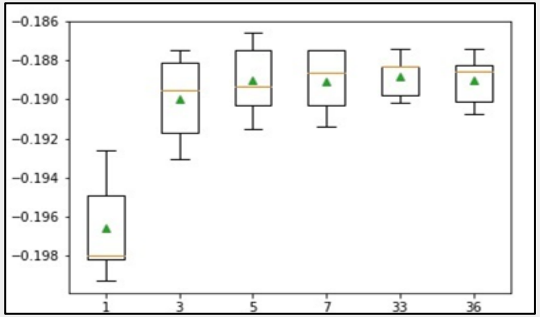
Missing data imputation

- **Summarizing missing data**

1. Proportion of missing data is small or zero
2. Imputation is still needed to avoid a waste of information

- **Missing data imputation**

1. Pin down optimal hyperparameter with a pipeline consisting of
(1) KNN imputer (2) random forest regressor in a five-fold cross validation process
2. Fit the KNN imputer on training data and then apply to testing data



Data-preprocessing

Standardization

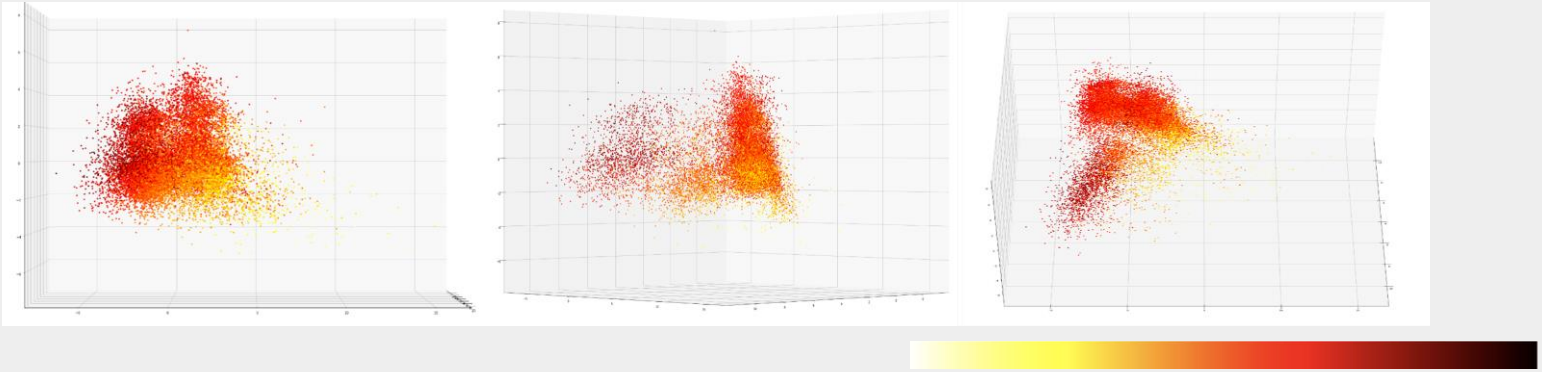
- Scale of numerical features differ greatly

total_floo	area_size	elevator	date_on_r	last_transa	bedrm_no	parlour_nc	kitchen_nc	toilet_no
6	52.01	0	42	5155	2	2	1	1
6	49.77	0	26	2092	1	1	1	1
5	65.75	0	184	4573	2	1	1	1
6	86.37	0	199	4090	2	2	1	1
21	71.22	1	68	1543	2	1	1	1
6	162.14	0	367	2384	5	2	1	2
30	185.87	1	513	6702	3	2	1	2
12	94.73	1	121	2311	3	1	1	1
5	84.77	0	90	4776	2	2	1	1
6	85.95	0	213	4387	2	2	1	1

- Use the mean and variance from training data to perform standarization

Data-preprocessing

Further visualization



[min(y), max(y)]

Explanantion

- Apply PCA to project the dataset to a three-dimensioanl space
- Looking it from three angles, our data is well-separated

3. Regression Analysis

Regression Analysis

Main Workflow

- **Model Selection** → Select model with optimal complexity by 5-folds cross-validation
- **Model Training** → Fit all training observations with selected model
- **Model Evaluation** → Assess the goodness of model on new data points with testing set

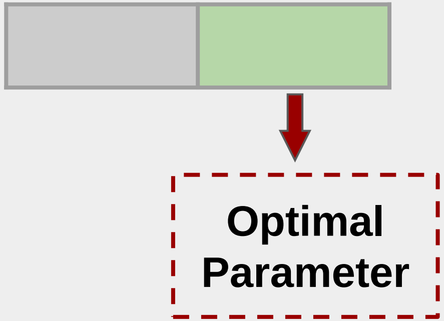
Regression Analysis

Model Selection Strategies

- **Parameter selection**

  Adjust parameter condidate sequence:

  until → the picked one lies in the middle of set (instead of boundary)



- **Computation Speed**

  Allow python to do parallel computing for acceleration

Regression Analysis

Model Evaluation Strategies

R-square	$\frac{\sum_{i=1}^{N}(\hat{y}-\bar{y})^2}{\sum_{i=1}^{N}(y-\bar{y})^2}$
RMSE	$\sqrt{\frac{1}{N}\sum_{i=1}^{N}(y-\hat{y})^2}$
MAE	$\frac{1}{N}\sum_{i=1}^{N}\lvert y-\hat{y}\rvert$

- **R-Square**

  How well the model explains the variation of target variable

- **Root-Mean-Squared Error (RMSE) and Mean-Absolute-Error**

  How deviate our prediction from the truth

  Shares the same unit of dependent variable

OLS

Advantage

- Computationally efficient
- Easily undertstand

VS

Drawback

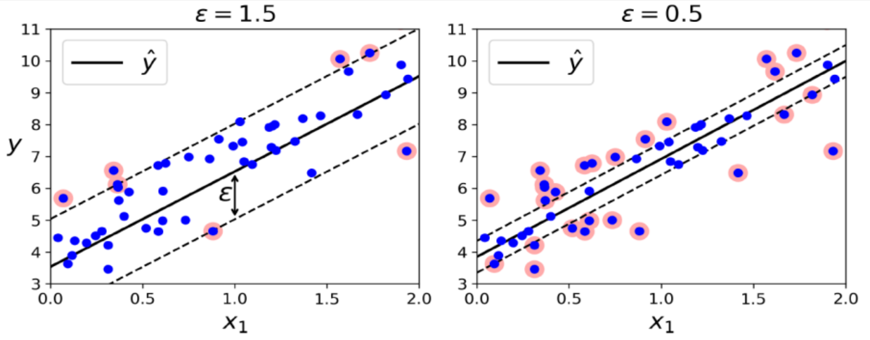
- **Inflexibility:** imposed linear restrictions

Results

	R-square	RMSE	MAE
Training	0.8516	0.2476	0.1886
Testing	0.8450	0.2557	0.1948

Not Satisfactory

Support Vector Machine Regression



Principle

- Fit more instances on the street along the hyperplane

Grid Search Using 5-folds cross-validation

- The kernel function controls non-linearity
- Radial basis function kernel is selected

Support Vector Machine Regression

Results

- ➢ Better than OLS
- ➢ Slightly overfit

	R-square	RMSE	MAE
Training	0.9429	0.1535	0.1134
Testing	0.8888	0.2166	0.1541

K-Nearest Neighbor Regression

Advantage

- Non-parametric in the sense that the number of parameters is unrestricted

Grid Search Using 5-folds cross-validation

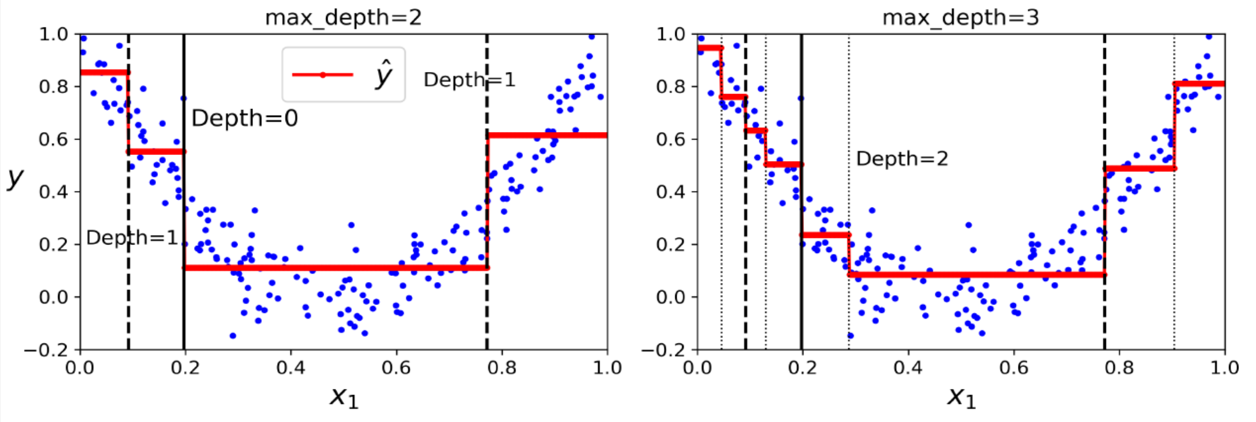
- The most crucial parameter: K=1

Results

	R-square	RMSE	MAE
Training	1.0	0.0	0.0
Testing	0.7901	0.2976	0.2271

Severe overfitting problem

Decision Tree Regression



Advantage

- ➢ **Flexible**： piecewise decision boundary
- ➢ **Robust**： not sensitive to outliers

Decision Tree Regression

Grid Search Using 5-folds cross-validation

- **Depth of tree should be selected**

  Otherwise, python will split data until only pure leaf nodes left

- **We also tuned 4 other paramaters**

  E.g., number of features to consider when looking for the best split

Results

	R-squared	RMSE	MAE
Training	0.9206	0.1811	0.137
Testing	0.8741	0.2305	0.1679

Decision Tree Regression

Results

	R-squared	RMSE	MAE
Training	0.9206	0.1811	0.1370
Testing	0.8741	0.2305	0.1609

Limitations

- **Variance - Solve by Random Forest**

  Gap between training and testing accuracy

- **Bias - Solve by Boosting Algorithms**

  There is still space to improve model fitting

Random Forest

Principle

- Pass data points by each tree trees and average the result

Grid Search Using 5-folds cross-validation

- Random forest is hard to overfit

  Tuning the tree depth is not rewarding; Set to be 30 here.

- number of features used in each tree, number of trees in the whole forest

Results

	R-squared	RMSE	MAE
Training	0.9892	0.0667	0.0484
Testing	0.9172	0.187	0.1345

Boosting Algorithms

Principle

- Ada Boosting

  Grow additional trees, assigning higher weight to instances with larger error before

- Gradient Boosting

  Add at each step a new decision tree that best reduces the loss function before

- XGB Boosting

  Regularization could be implemented compared with Gradient boosting

Grid Search Using 5-folds cross-validation

- We tuned learning rate for all of these three models
- There are also other paramters selected

Boosting Algorithms

Results

	R-squared	RMSE	MAE
Ada (training)	0.9998	0.0101	0.0029
Ada (testing)	0.9191	0.1848	0.1269
Gradient (training)	0.9954	0.0436	0.0336
Gradient (testing)	0.8583	0.2445	0.1709
XGB (training)	0.9867	0.0742	0.0584
XGB (testing)	0.9197	0.1841	0.1334

- Ada and XGB are excellent
- Gradient boosting overfits

Overall Comparison

Results on Testing Data

	R-squared	RMSE	MAE
OLS	0.8450	0.2557	0.1948
KNN	0.7901	0.2976	0.2271
SVR	0.8888	0.2166	0.1541
Decision Tree	0.8741	0.2305	0.1697
Random Forest	0.9172	0.1870	0.1345
Ada boosting	0.9191	0.1848	0.1269
Gradient boosting	0.8584	0.2445	0.1709
XGB boosting	0.9197	0.1841	0.1334

4. Variable Importance

Variable Importance

Criteria

- **Predictive power**

  In tree-based models, it is the reduction of node impurity

- **Based on the best three models mentioned before**

  Random forest, Ada boosting, XGB boosting

Results

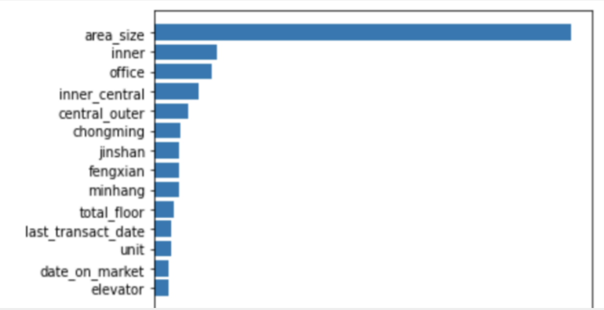
- **Random forest and Ada boosting**

  Area-size (numerical), Inner (dummy), Office (dummy)

- **XGB boosting**

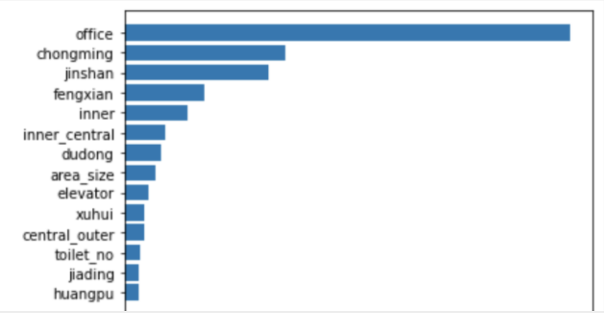
  Office, Jinshan (dummy), Nongming (dummy)

Variable Importance (Visualization)



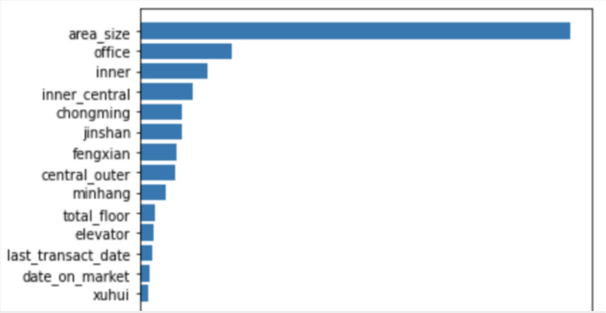
XGB boosting

area_size | inner | office



Ada boosting

office | Nongming | jinshan



Gradient boosting

area_size | inner | office

5. Conclusion

Summary

Data → Second-hand house data scraped from Lianjia.com

Pre-Processing → Visualization, Cleaning, Transformation

Model Building → Our best model gives a R-squared around 0.92

Variable Importance → Variables with good explanatroy power are identified


Q&A


THANK YOU.