House price prediction

Supervised machine learning

Hayley Hyejeong Lee

Project Goal

To predict the house price using different regression models

To find the best-performing model

Dataset

House sale prices in King County

(May 2014–May 2015)

18 features
(both numerical and categorical features)

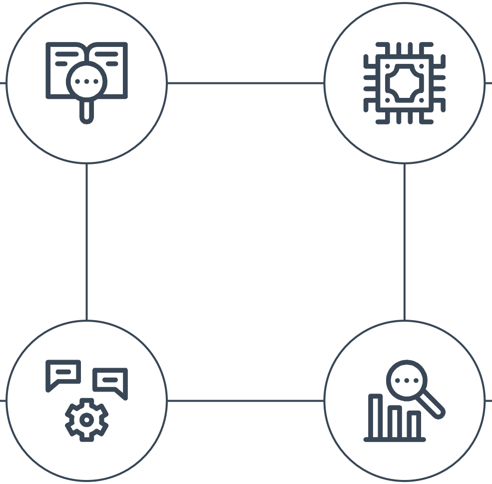
Project Workflow

Data Preprocessing

Data cleaning
Feature selection

Data analysis

Multivariate analysis
Correlation matrix



Model training

Train baseline model
Comparing model performance

Best-performing model

Model evaluation
Hyperparameter tuning

Data preprocessing

Data cleaning

Handling missing values
Check duplicates rows

Drop the columns where bedrooms and bathrooms are 0 & sale_year<yr_bulit

Date to datetime (dtype)
Convert to house_age from sale_year and yr_built

Data analysis

Univariate analysis on target variable: price

Bivariate analysis with other features (grade, sqft_living, lat, long etc)

Correlation matrix for numerical columns to check multicollinearity

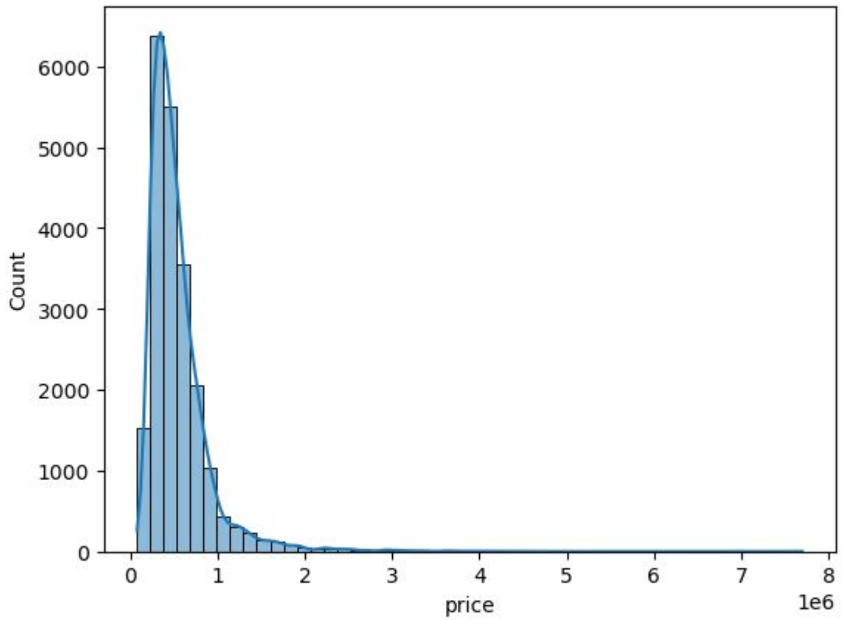
Univariate analysis : Price

Descriptive Statistics

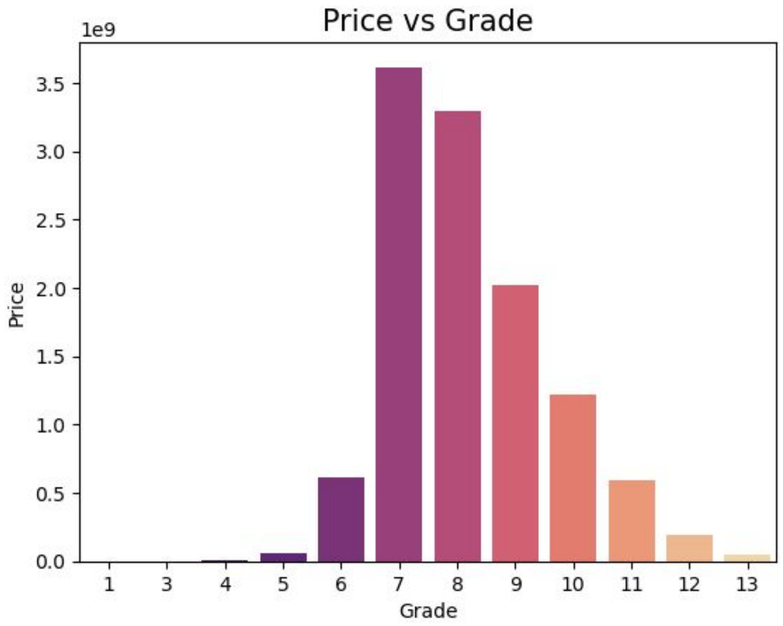
Mean: 5,400,881
Min: 75,000
Max: 7,700,000

Distribution

Skewness (4.02)
-> highly right-skewed
Kurtosis(34.58)
-> heavily tailed
-> contain extreme outliers (may be capping, log transformation)

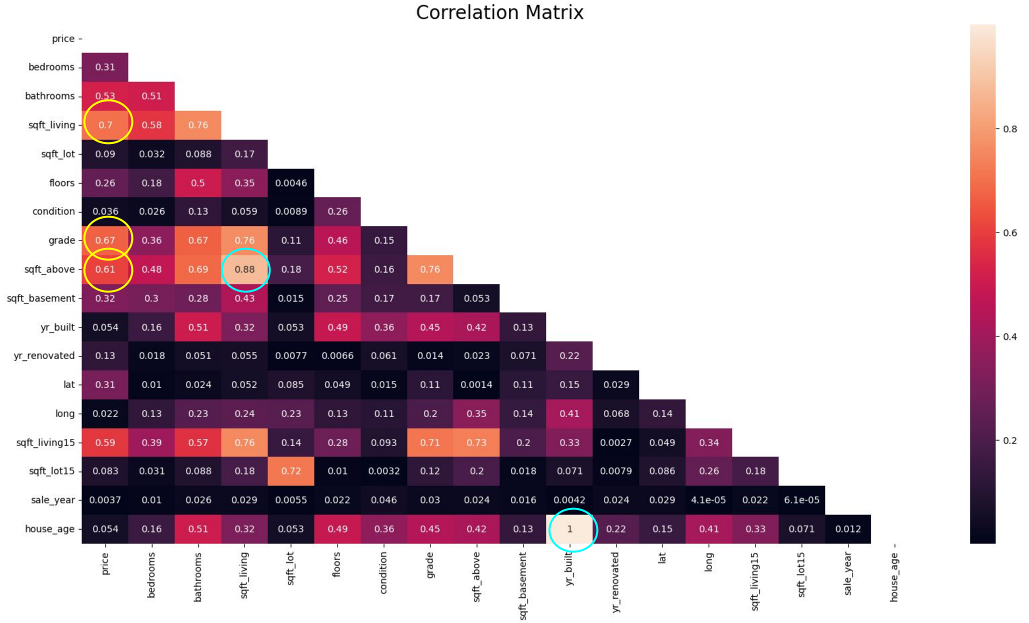


Bivariate analysis



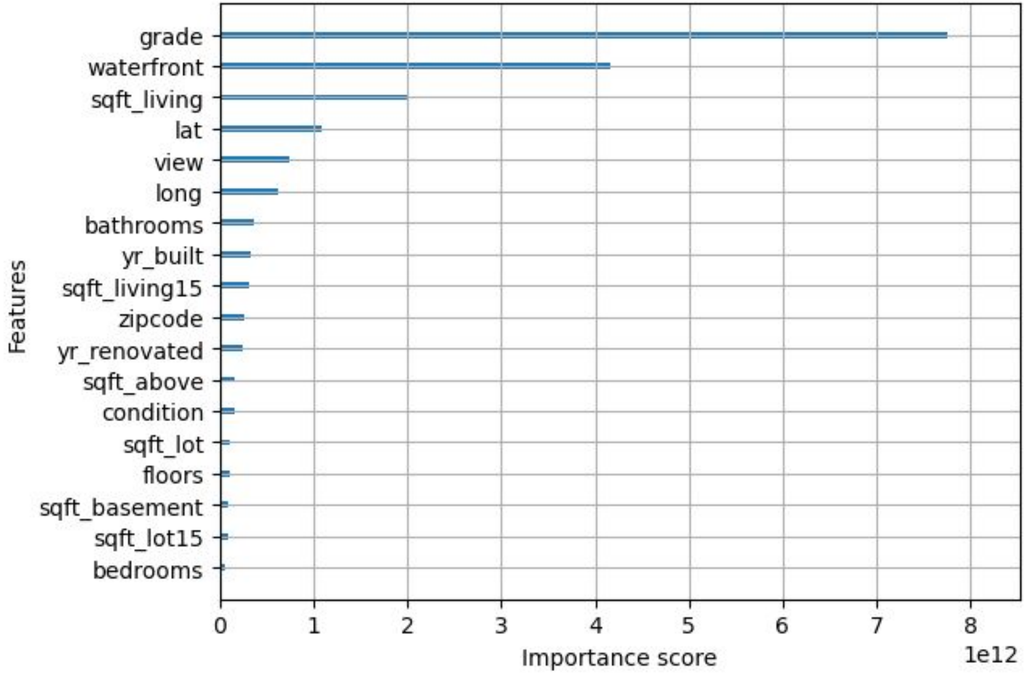
Correlation matrix

Features	Correlation score (with price)
sqft_living	0.702
grade	0.668
sqft_above	0.605
sqft_living15	0.585
bathrooms	0.526



Strong correlation (light blue circle)
Between house_age and yr_built (1)
between sqft_lliving and sqft_above (0.88)

Feature importance



Top 10 features
grade
waterfront
sqft_living
lat
long
view
house_age
zipcode
sqft_living15

Grade, waterfront, sqft_living explain about 70 % of the total model importance (Xgboost)

Interaction: selected features

```
Data columns (total 19 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   price          21585 non-null  float64
 1   bedrooms       21585 non-null  int64
 2   bathrooms      21585 non-null  float64
 3   sqft_living    21585 non-null  int64
 4   sqft_lot       21585 non-null  int64
 5   floors         21585 non-null  float64
 6   waterfront     21585 non-null  int64
 7   view           21585 non-null  int64
 8   condition      21585 non-null  int64
 9   grade          21585 non-null  int64
 10  sqft_basement  21585 non-null  int64
 11  yr_renovated   21585 non-null  int64
 12  zipcode        21585 non-null  int64
 13  lat            21585 non-null  float64
 14  long           21585 non-null  float64
 15  sqft_living15  21585 non-null  int64
 16  sqft_lot15     21585 non-null  int64
 17  sale_year      21585 non-null  int32
 18  house_age      21585 non-null  int64
```

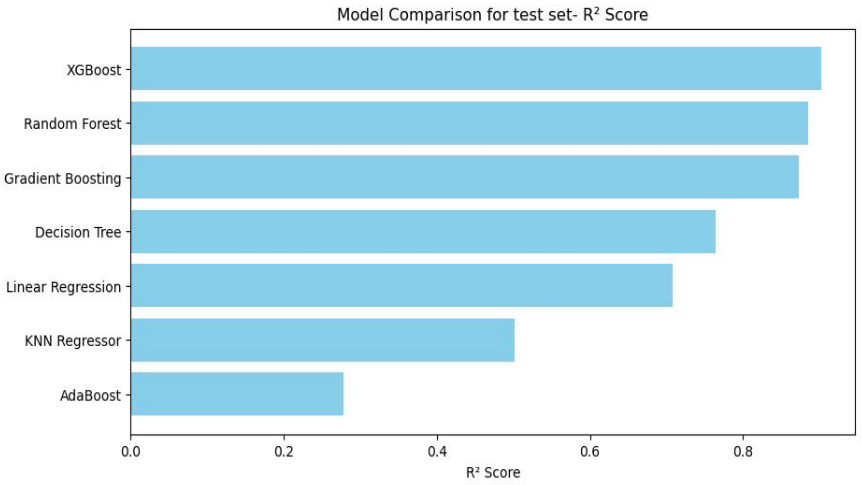
Four columns were removed
= id, date, yr_built, sqft_above

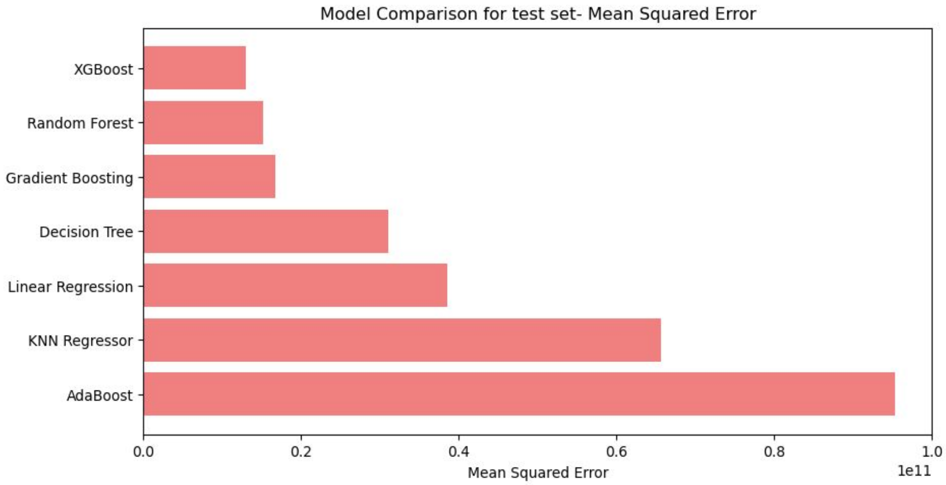
✅ Model Comparison (Train vs Test):

	Model	Train_MSE	Test_MSE	Train_R2	Test_R2
6	XGBoost	2.962096e+09	1.296811e+10	0.978149	0.901710
3	Random Forest	2.401300e+09	1.521478e+10	0.982286	0.884682
5	Gradient Boosting	1.319050e+10	1.671492e+10	0.902697	0.873312
2	Decision Tree	9.108929e+06	3.104438e+10	0.999933	0.764704
0	Linear Regression	4.064180e+10	3.848296e+10	0.700195	0.708325
1	KNN Regressor	4.543097e+10	6.568779e+10	0.664866	0.502130
4	AdaBoost	9.334322e+10	9.526367e+10	0.311429	0.277965

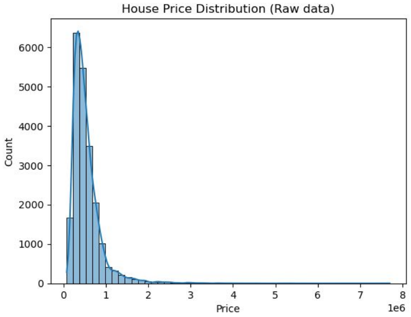
XGBoost showed the highest performance

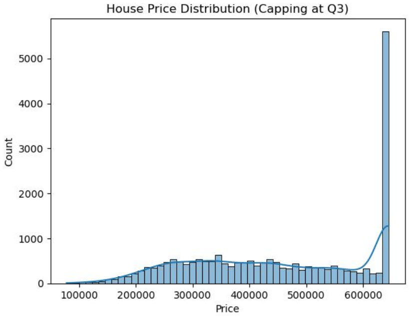
Model evaluation



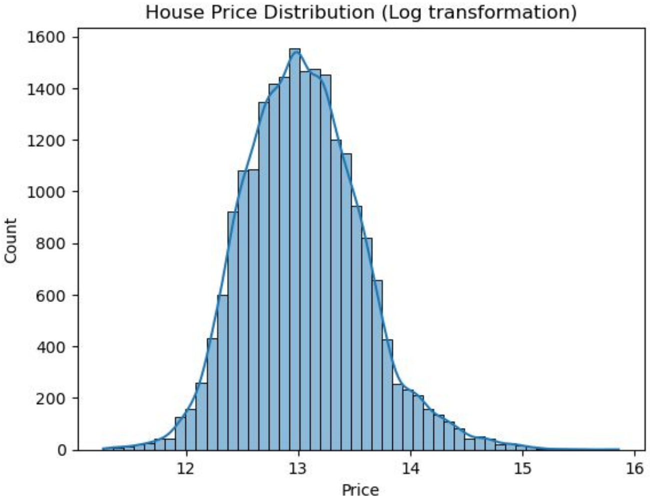


Feature engineering: price





Price capping above Q3



Log transformation

Model improvement

Standardization

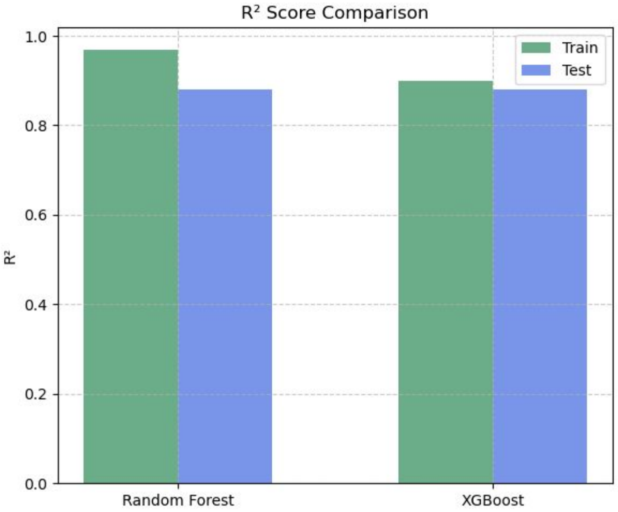
No impact on model performance

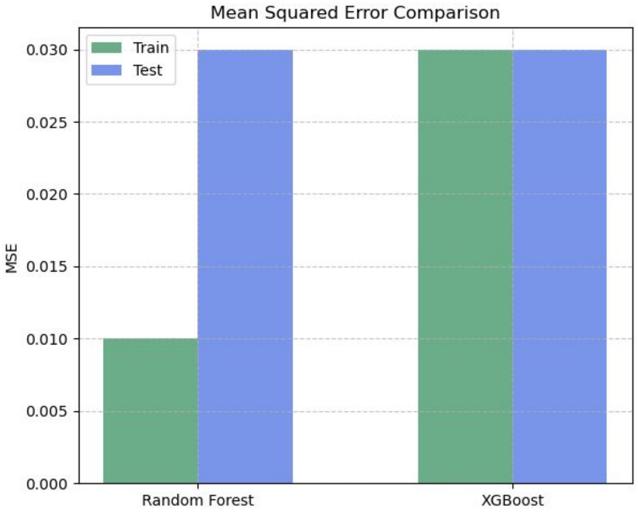
Scaled feature:
sqft_living, sqft_log, sqft_living15
sqft_lot15, sqft_basement

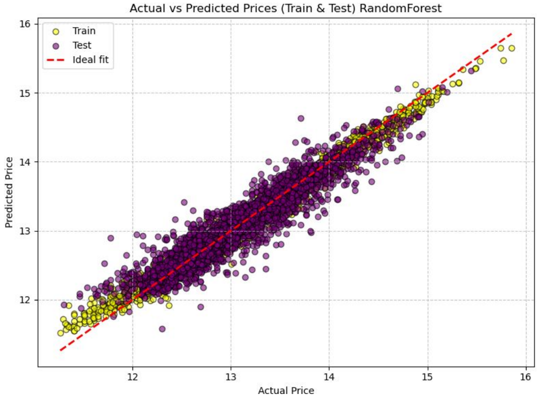
Hyperparameter tuning

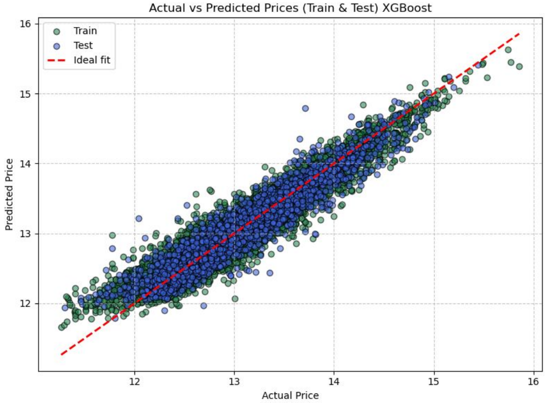
RandomizedSearchCV using XGBoost and Random Forest models

RandomForest vs XGBoost









Summary

XGBoost fits the testing data better (higher R2 score: 0.92)
RandomForest shows slightly more overfitting (gap between train and test, about 10%)
MSE for testing data is identical for both models.

Overall, XGBoost is more balanced, generalized better than RandomForest.

Thanks!