eegshou / dsc-phase-2-project Public

forked from learn-co-curriculum/dsc-phase-2-project

View license

0 stars 178 forks

Star Watch

Code Pull requests Actions Projects Wiki Security Insights Settings

main

This branch is 36 commits ahead of learn-co-curriculum:main.

Contribute Fetch upstream

eegshou ... 3 days ago

View code

README.md

Phase 2 Project

Project Overview

This project analyzes house sales data in a northwestern county using regression model.

Business Problem

After buiding the regression model, the features that are closely related to house price will be identified.

Therefore, some suggestions could be given to both the buyers and sellers.

- For the buyer, they will know the price of the house based on the characteristics of the house, and also, what's the investment value for the house.
- For the seller, they may know whether they can do something to sell the house with a better price.

Data

This project uses the King County House Sales dataset, which can be found in `kc_house_data.csv` in the data folder in this repo. The description of the column names can be found in `column_names.md` in the same folder.

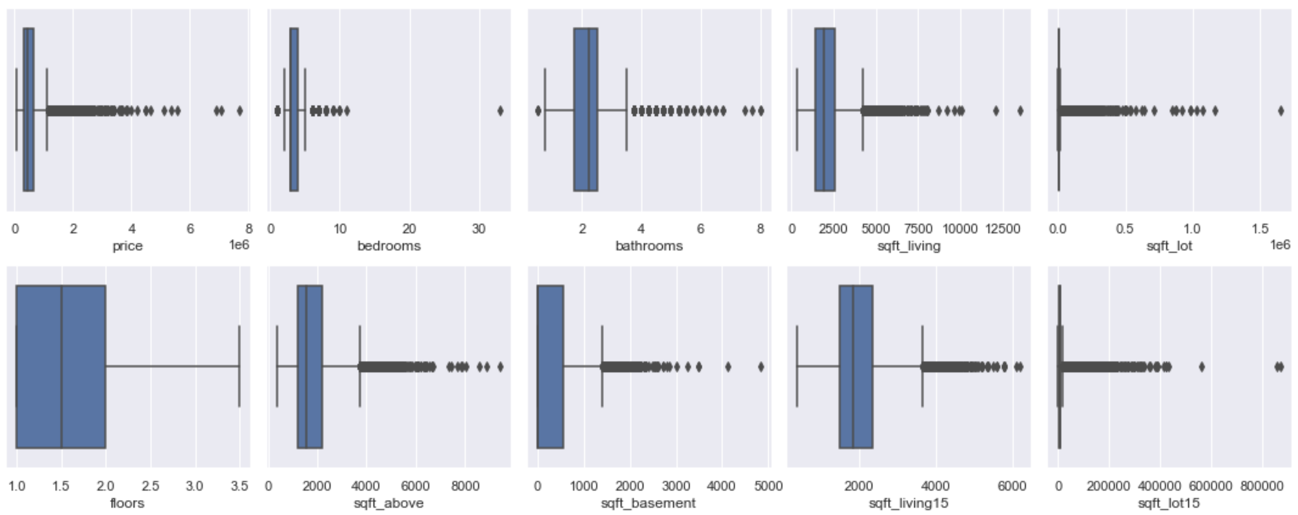
Methods

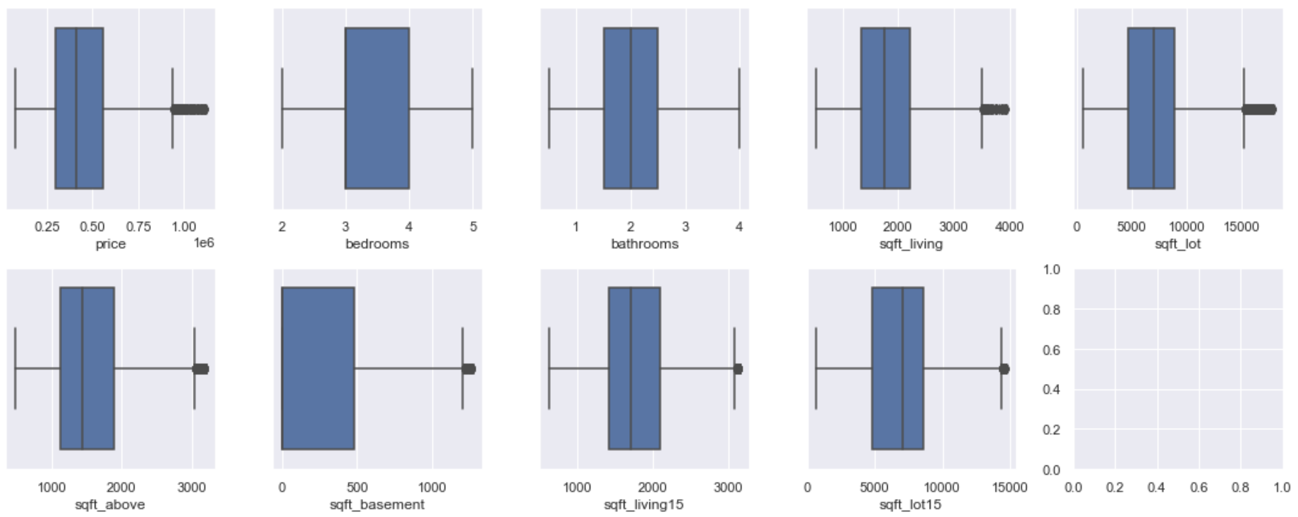
First, I loaded data to check the potential features, and found:

1. There are potentially 19 predictors excluding the id and the target,i.e.,the price
2. A total of 21597 rows, while some rows have null values in some predictors
3. Several predictors' data type need to be changed

Second, I prepare the data by

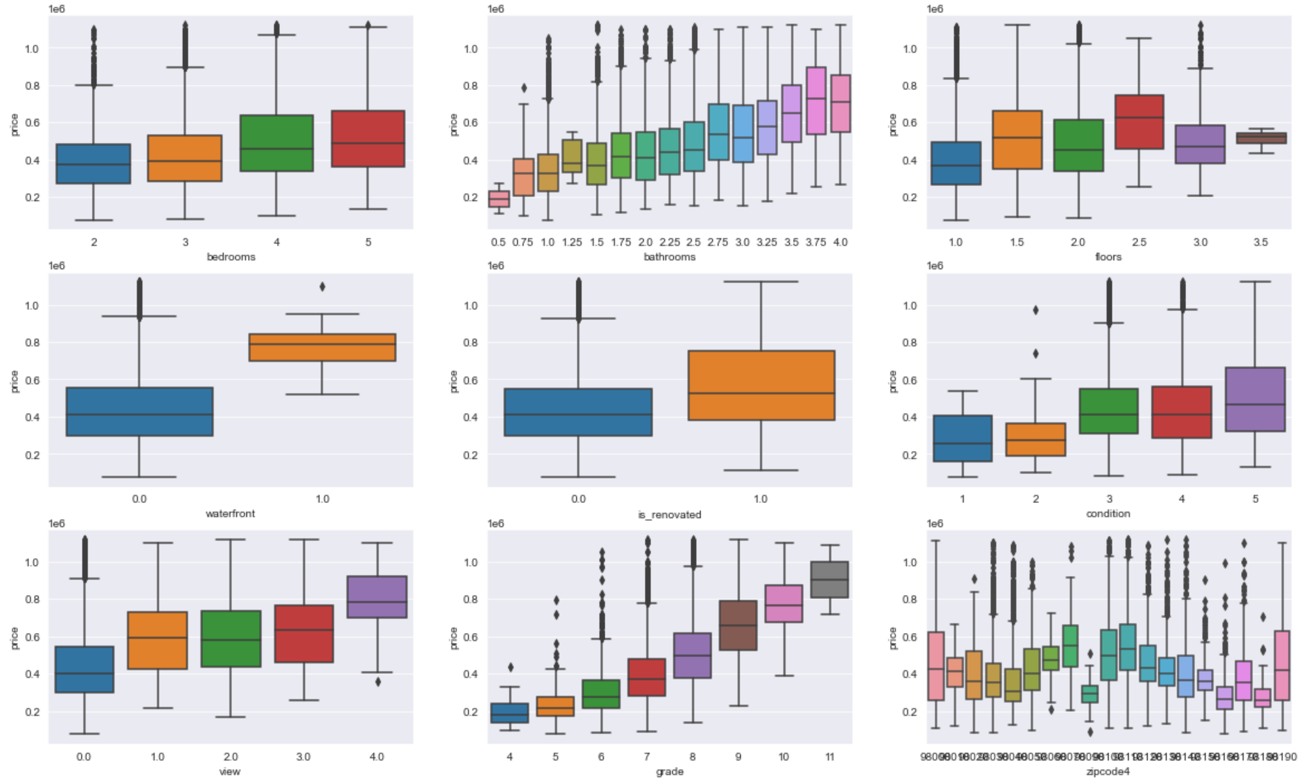
1. dealing with datatyepes for sqft_basement & date
2. dealing with the missing values in waterfront, view, yr_renovated, sqft_basement
3. dealing with outerliers (see the following figures for the data before and after removing outliers)





And finally the histogram of all features are shown below

For the zip code, I only keep the first four digits since if I only keep the first three digits,it will only have two zipcodes And the visualization of categorical variables are shown in the following figure



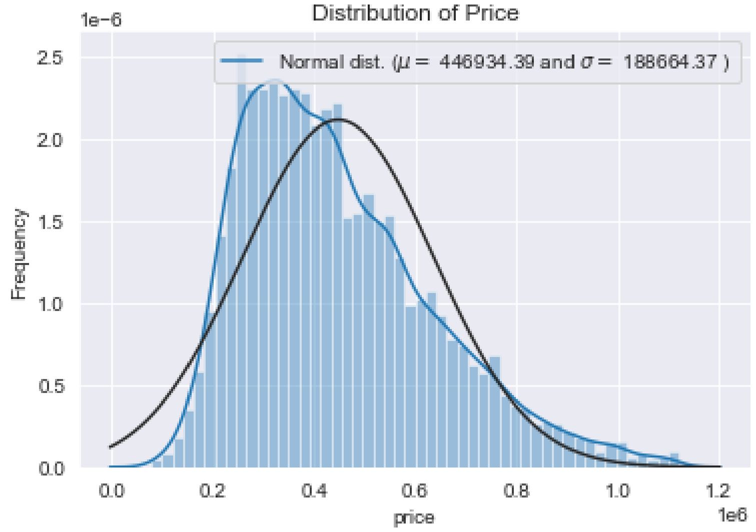
Modeling

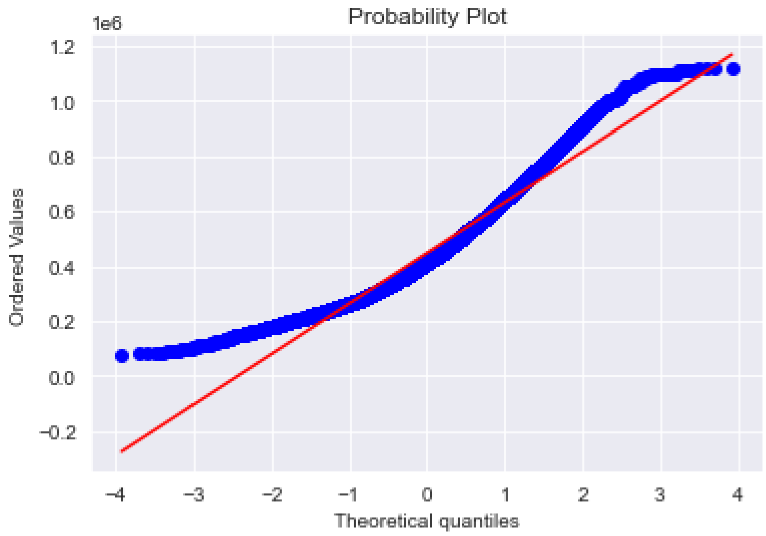
After finishing the data preparation, now I start to build the regression model I drop id, date, yr_renovated,lat, long, zipcode from df since:

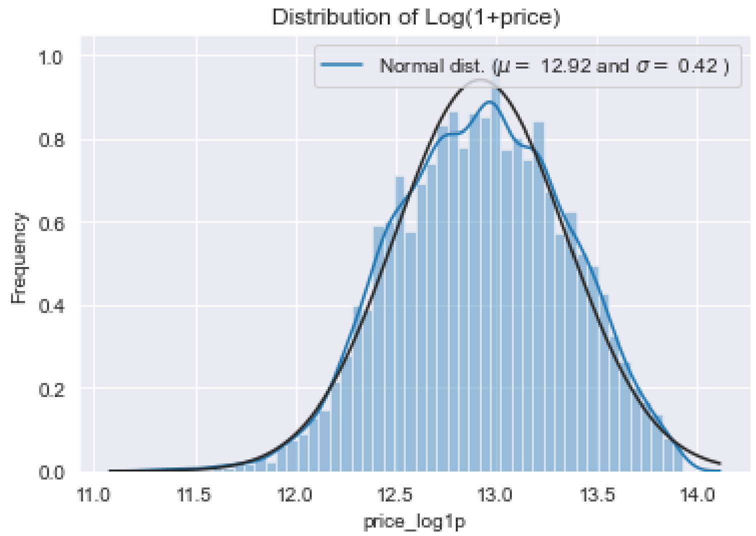
1. the id has not related to the house price
2. the date has been transformed into sold_year and sold_month
3. the yr_renovated has been transformed into is_renovated
4. the lat and long indicate similar information as zipcode
5. zipcode has been transformed into zipcode4 and dummy variables

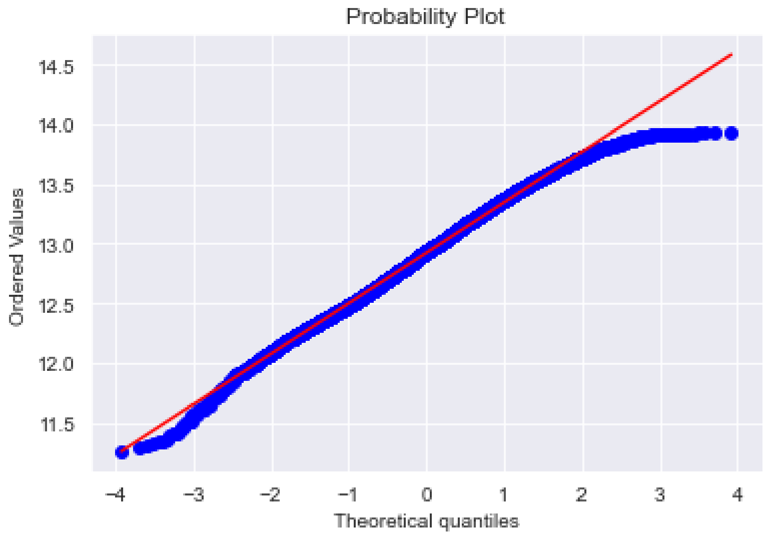
I also scaled individual features in the same scale

For the target, i.e., price, I performed a transformation: log10(1+price), to make it more normal distribution(see the following figures)


Distribution of Price
1e−6
Normal dist. ($\mu =$ 446934.39 and $\sigma =$ 188664.37)
Frequency
price
1e6
0.0
0.5
1.0
1.5
2.0
2.5
0.2
0.4
0.6
0.8
1.0
1.2


Probability Plot
1e6
Ordered Values
Theoretical quantiles
−0.2
0.0
0.2
0.4
0.6
0.8
1.0
1.2
−4
−3
−2
−1
0
1
2
3
4





Now I conducted linear regression modeling. I firstly used all features and found the transformed price has a little better R-squared value. Therefore, I used it as the target for subsequent analysis.

After removing the insignificant Features, and features with collinearity, I obtained the final model shown as below:

OLS Regression Results

Dep. Variable:	price_log1p	**R-squared:**	0.449
Model:	OLS	**Adj. R-squared:**	0.447
Method:	Least Squares	**F-statistic:**	342.6
Date:	Thu, 18 Nov 2021	**Prob (F-statistic):**	0.00
Time:	21:44:44	**Log-Likelihood:**	-3325.9
No. Observations:	12664	**AIC:**	6714.
Df Residuals:	12633	**BIC:**	6945.
Df Model:	30		
Covariance Type:	nonrobust		

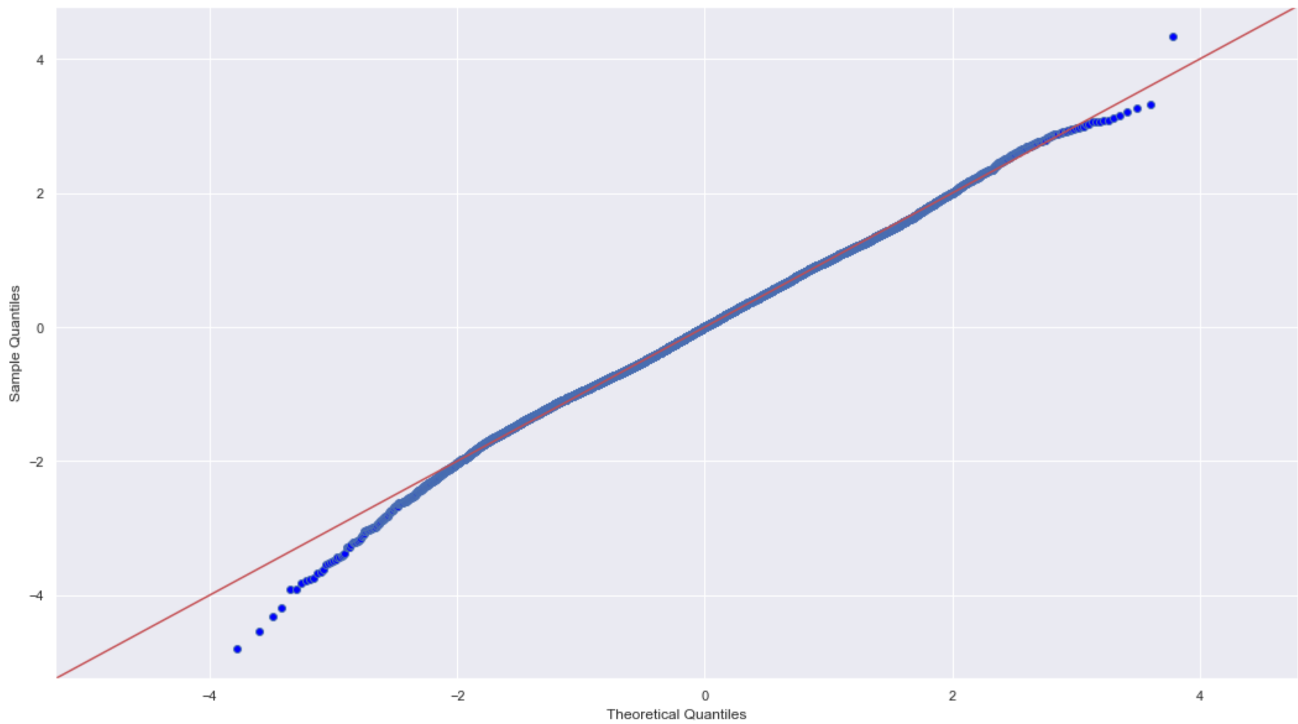
	coef	std err	t	P>|t|	[0.025	0.975]
const	12.3034	0.013	939.565	0.000	12.278	12.329
sqft_lot	0.0019	0.032	0.061	0.951	-0.060	0.064
sqft_living15	1.0967	0.017	62.711	0.000	1.062	1.131
sqft_lot15	-0.1448	0.030	-4.819	0.000	-0.204	-0.086
bedrooms_3	0.0838	0.008	9.855	0.000	0.067	0.100
bedrooms_4	0.1500	0.010	15.497	0.000	0.131	0.169
bedrooms_5	0.1632	0.015	10.968	0.000	0.134	0.192
floors_2.5	0.2261	0.040	5.682	0.000	0.148	0.304
floors_3.0	0.0568	0.017	3.423	0.001	0.024	0.089
waterfront_1.0	0.3989	0.113	3.545	0.000	0.178	0.620
is_renovated_1.0	0.1923	0.017	11.532	0.000	0.160	0.225
view_1.0	0.1377	0.026	5.345	0.000	0.087	0.188
view_2.0	0.1483	0.016	9.056	0.000	0.116	0.180
view_3.0	0.1173	0.027	4.324	0.000	0.064	0.170
view_4.0	0.2436	0.051	4.730	0.000	0.143	0.344
grade_10	0.2589	0.026	9.868	0.000	0.207	0.310

grade_11	0.4181	0.141	2.961	0.003	0.141	0.695
zipcode4_98020	-0.1122	0.011	-9.881	0.000	-0.134	-0.090
zipcode4_98030	-0.0860	0.010	-8.718	0.000	-0.105	-0.067
zipcode4_98040	-0.1567	0.015	-10.469	0.000	-0.186	-0.127
zipcode4_98060	-0.0552	0.027	-2.061	0.039	-0.108	-0.003
zipcode4_98070	0.1285	0.016	7.933	0.000	0.097	0.160
zipcode4_98090	-0.4208	0.024	-17.744	0.000	-0.467	-0.374
zipcode4_98100	0.3045	0.011	26.970	0.000	0.282	0.327
zipcode4_98110	0.3558	0.010	35.292	0.000	0.336	0.376
zipcode4_98120	0.2334	0.013	17.816	0.000	0.208	0.259
zipcode4_98130	0.1495	0.015	10.127	0.000	0.121	0.178
zipcode4_98150	0.0563	0.019	2.944	0.003	0.019	0.094
zipcode4_98160	-0.2256	0.019	-11.681	0.000	-0.263	-0.188
zipcode4_98170	-0.0977	0.018	-5.360	0.000	-0.133	-0.062
zipcode4_98180	-0.2792	0.034	-8.272	0.000	-0.345	-0.213

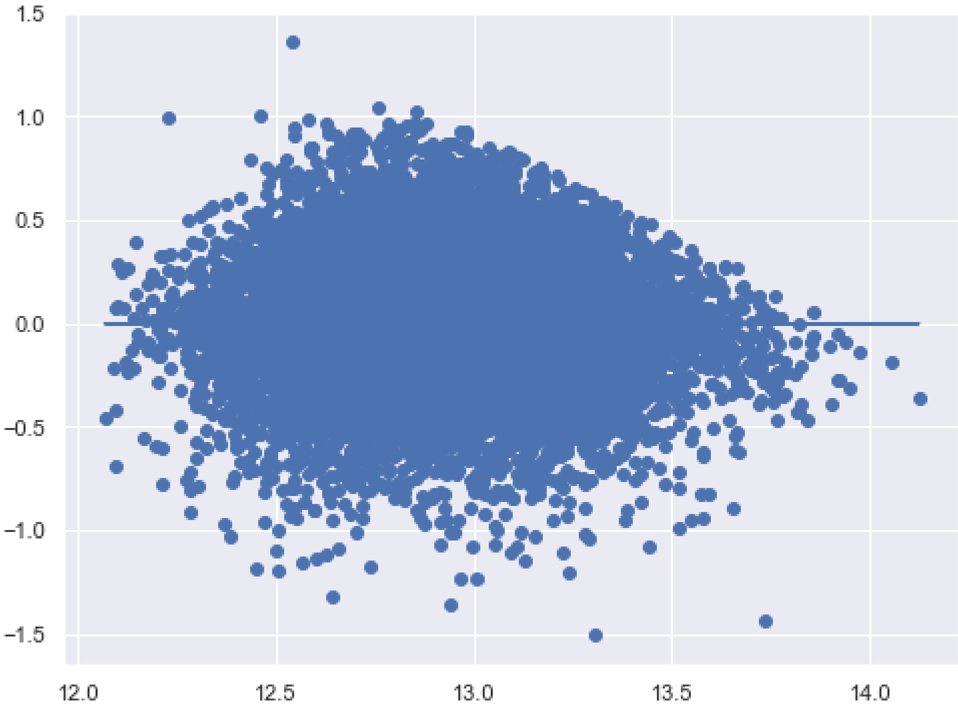
Omnibus:	48.135	**Durbin-Watson:**	2.022
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	59.797
Skew:	-0.069	**Prob(JB):**	1.04e-13
Kurtosis:	3.307	**Cond. No.**	70.9

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

The model's normality is



And the homoscedasticity is



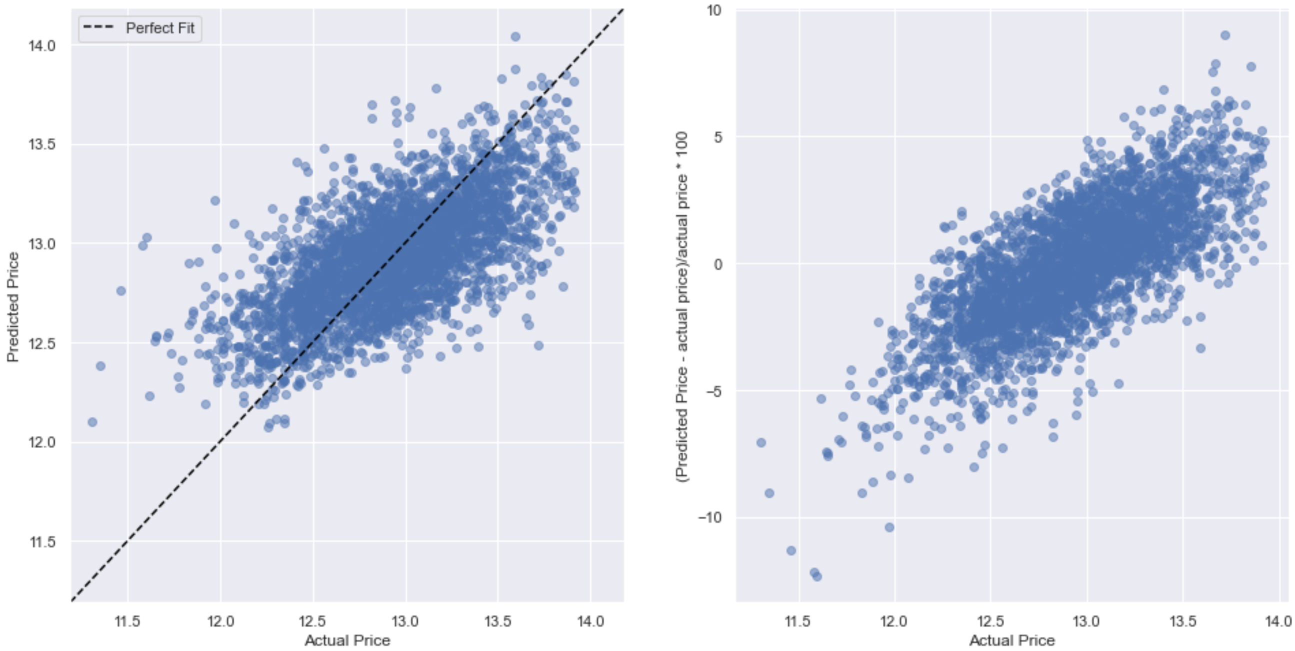
Evaluation

To evaluate the model, I calculated the score for both train and test data, and found they are similar: Test score: 0.44378007421545485 Train score: 0.44859642929281374

I also used cross validation to evaluate the model and found that they are similar Train score: 0.44648159473484395 Validation score: 0.4500640604866031

I also calculated mean squared error, root mean squared error, and Mean absolute error: MSE: 0.08275149712742216 RMSE: 0.28766559948562176 MAE: 0.22861402157360175 R-Squared: 0.5432593071160836

And the comparison of predicted and real values



From above values and plots, the fitted regression model can predict house price very well

Summary

The coefficients of the selected features are:

```
sqft_lot            0.001943
sqft_living15       1.096717
sqft_lot15         -0.144789
bedrooms_3          0.083760
bedrooms_4          0.149979
bedrooms_5          0.163189
floors_2.5          0.226109
floors_3.0          0.056801
waterfront_1.0      0.398927
is_renovated_1.0    0.192348
view_1.0            0.137734
view_2.0            0.148326
view_3.0            0.117277
view_4.0            0.243554
grade_10            0.258860
grade_11            0.418129
zipcode4_98020     -0.112151
zipcode4_98030     -0.086005
zipcode4_98040     -0.156698
zipcode4_98060     -0.055208
zipcode4_98070      0.128485
zipcode4_98090     -0.420847
zipcode4_98100      0.304544
zipcode4_98110      0.355810
zipcode4_98120      0.233377
zipcode4_98130      0.149527
zipcode4_98150      0.056281
zipcode4_98160     -0.225556
zipcode4_98170     -0.097728
zipcode4_98180     -0.279241
Name: Coefficients, dtype: float64

Intercept: 12.30344679761995
```

From coefficients described above, I observed:

1. The grade and sqft_living15 have the strongest relationship with the house price
2. It is interesting to see the sqft_lot15 has the negative relationship with the house price

To address the business question:

1. For buyer, they will know the house price is higer for a house with high grade and sqrt_living15
2. For seller, if they want to sell their house with a higher price, they could add waterfront, improve the grade/condition.

Releases

No releases published
Create a new release

Packages

No packages published
Publish your first package

Languages

- **Jupyter Notebook** 100.0%