

Final project in Module 2

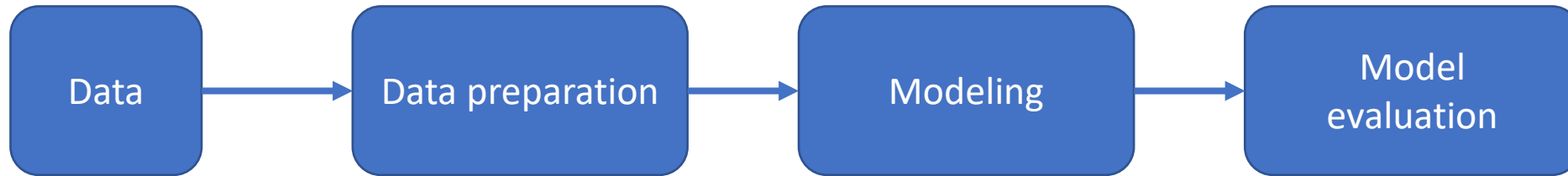
Guofa Shou

Self-paced

Outline

- The aim of the project:
 - To provide suggestions regarding the price of house
 - For house buyer, they will know the approximate price of the house and the investment value of the house for future sale, based on the characteristics of the house
 - For house seller, they will know what they need to do to sell the house with a better price
- A linear regression model is built based on the selected house characteristics to achieve the aim of the project
- Conclusion

A linear regression model



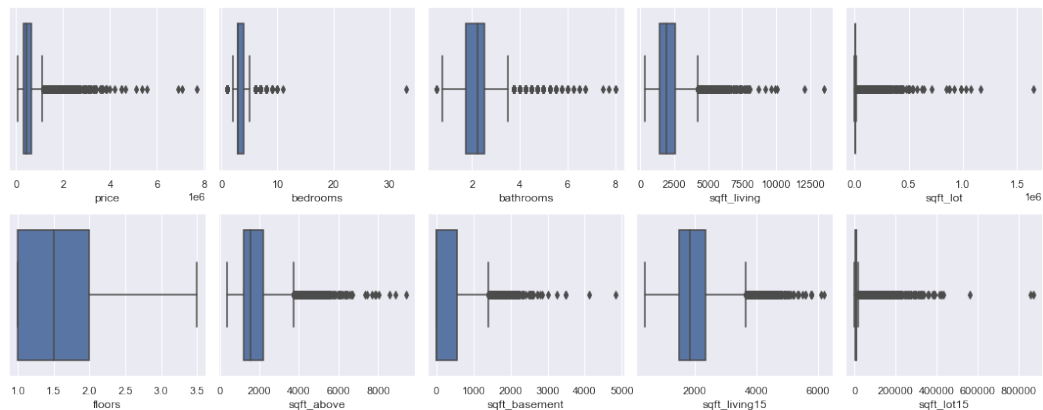
Data

- Data source: kc_house_data.csv
- Data understanding:
 - A total of 19 predictors available after excluding the id and the target (price)
 - A total of 21597 rows, while some rows have null values in some predictors
 - Several predictors' data type need to be changed, e.g., date and sqft_basement

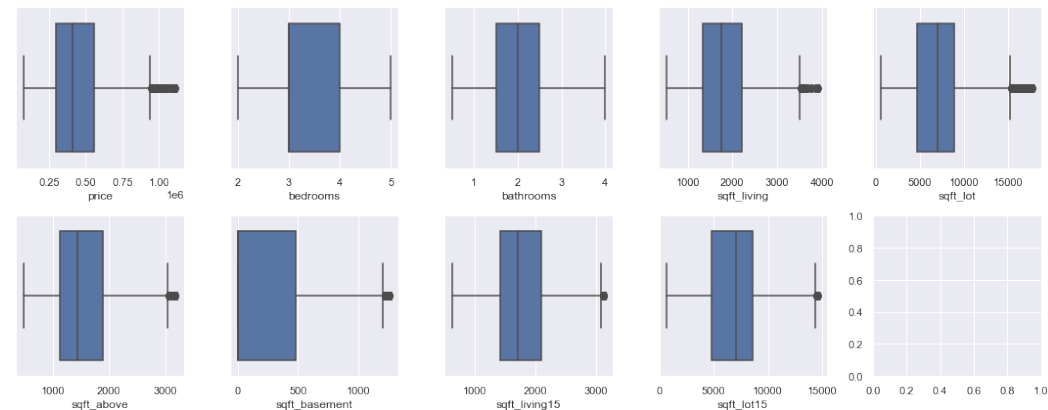
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 21597 entries, 0 to 21596
Data columns (total 21 columns):
#   Column                Non-Null Count  Dtype
---  -
0   id                    21597 non-null  int64
1   date                 21597 non-null  object
2   price               21597 non-null  float64
3   bedrooms            21597 non-null  int64
4   bathrooms           21597 non-null  float64
5   sqft_living         21597 non-null  int64
6   sqft_lot            21597 non-null  int64
7   floors              21597 non-null  float64
8   waterfront          19221 non-null  float64
9   view                21534 non-null  float64
10  condition            21597 non-null  int64
11  grade               21597 non-null  int64
12  sqft_above          21597 non-null  int64
13  sqft_basement       21597 non-null  object
14  yr_built            21597 non-null  int64
15  yr_renovated        17755 non-null  float64
16  zipcode             21597 non-null  int64
17  lat                 21597 non-null  float64
18  long                21597 non-null  float64
19  sqft_living15       21597 non-null  int64
20  sqft_lot15          21597 non-null  int64
dtypes: float64(8), int64(11), object(2)
memory usage: 3.5+ MB
```

Data Preparation

- Deal with data types: sqft_basement and date
- Deal with missing values:
 - View and sqft_basement: drop those rows with null values
 - Waterfront, and yr_renovated: they are over 10% of null values with special process
- Deal with outliers:



Original

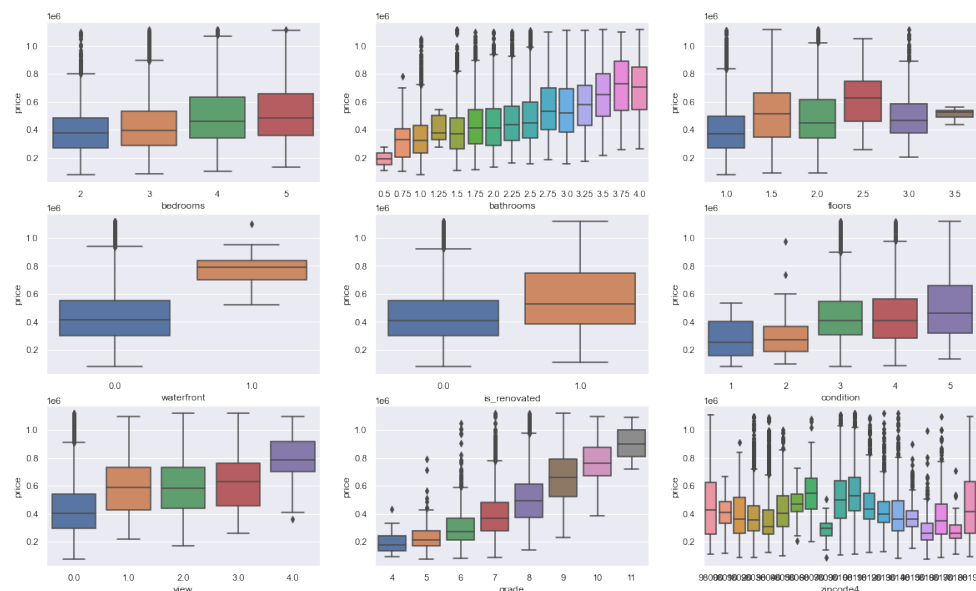


After removing outliers

Data Preparation

- Deal with categorical variables
 - 'bedrooms', 'bathrooms', 'floors', 'waterfront', 'is_renovated', 'condition', 'view', 'grade', 'zipcode4'

Visualization of categorical variables



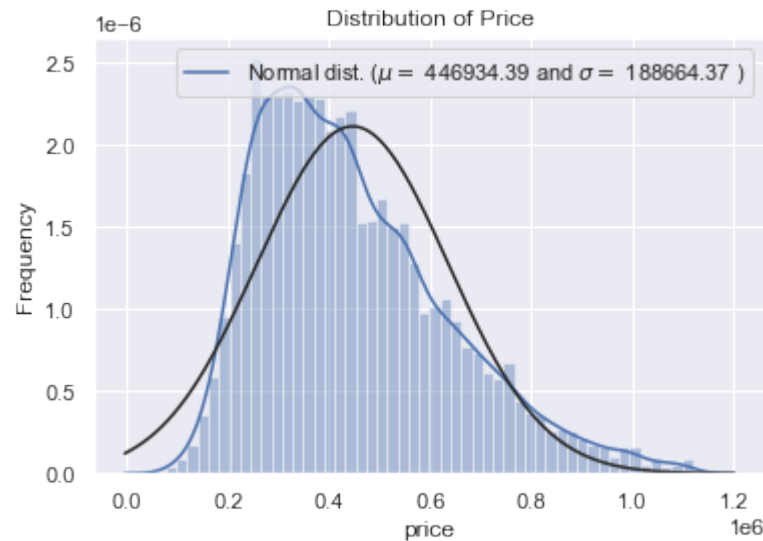
The predictors

```
df_scl.columns
```

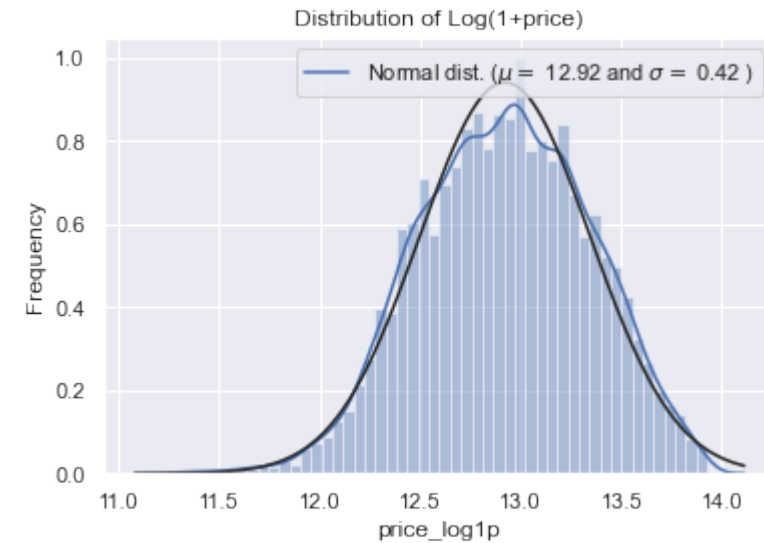
```
Index(['price', 'sqft_living', 'sqft_lot', 'sqft_above', 'sqft_basement',  
      'yr_built', 'sqft_living15', 'sqft_lot15', 'year_sold', 'month_sold',  
      'age_sold', 'bedrooms_3', 'bedrooms_4', 'bedrooms_5', 'bathrooms_0.75',  
      'bathrooms_1.0', 'bathrooms_1.25', 'bathrooms_1.5', 'bathrooms_1.75',  
      'bathrooms_2.0', 'bathrooms_2.25', 'bathrooms_2.5', 'bathrooms_2.75',  
      'bathrooms_3.0', 'bathrooms_3.25', 'bathrooms_3.5', 'bathrooms_3.75',  
      'bathrooms_4.0', 'floors_1.5', 'floors_2.0', 'floors_2.5', 'floors_3.0',  
      'floors_3.5', 'waterfront_1.0', 'is_renovated_1.0', 'condition_2',  
      'condition_3', 'condition_4', 'condition_5', 'view_1.0', 'view_2.0',  
      'view_3.0', 'view_4.0', 'grade_5', 'grade_6', 'grade_7', 'grade_8',  
      'grade_9', 'grade_10', 'grade_11', 'zipcode4_98010', 'zipcode4_98020',  
      'zipcode4_98030', 'zipcode4_98040', 'zipcode4_98050', 'zipcode4_98060',  
      'zipcode4_98070', 'zipcode4_98090', 'zipcode4_98100', 'zipcode4_98110',  
      'zipcode4_98120', 'zipcode4_98130', 'zipcode4_98140', 'zipcode4_98150',  
      'zipcode4_98160', 'zipcode4_98170', 'zipcode4_98180', 'zipcode4_98190'],  
      dtype='object')
```

Data Preparation

- Deal with target variable: price
 - Logarithm transform to make it more normal distribution



Log(1+price)



Modeling

- Regression model with all available predictors
 - Using price or log-transformed price as the target

```
=====
                        OLS Regression Results
=====
Dep. Variable:          price      R-squared:                0.597
Model:                  OLS        Adj. R-squared:            0.595
Method:                 Least Squares  F-statistic:            291.5
Date:                  Thu, 18 Nov 2021  Prob (F-statistic):      0.00
Time:                  22:28:20      Log-Likelihood:         -1.6601e+05
No. Observations:      12664        AIC:                   3.322e+05
Df Residuals:          12599        BIC:                   3.326e+05
Df Model:               64
Covariance Type:       nonrobust
=====
```

```
=====
                        OLS Regression Results
=====
Dep. Variable:          price_log1p  R-squared:                0.598
Model:                  OLS        Adj. R-squared:            0.595
Method:                 Least Squares  F-statistic:            292.2
Date:                  Thu, 18 Nov 2021  Prob (F-statistic):      0.00
Time:                  22:28:20      Log-Likelihood:         -1332.8
No. Observations:      12664        AIC:                   2796.
Df Residuals:          12599        BIC:                   3280.
Df Model:               64
Covariance Type:       nonrobust
=====
```

From R-squared values, it seems log-transformed price is a little better

Modeling

- Regression model with all predictors that are significantly related to the target: $p < 0.05$

	coef	std err	t	P> t	[0.025	0.975]
const	7.4111	0.082	90.876	0.000	7.251	7.571
sqft_living	0.1990	0.012	15.974	0.000	0.175	0.223
sqft_lot	-0.0778	0.028	-2.814	0.005	-0.132	-0.024
sqft_above	0.2990	0.015	19.764	0.000	0.269	0.329
sqft_basement	0.1290	0.011	12.253	0.000	0.108	0.150
yr_built	3.4663	0.041	83.645	0.000	3.385	3.548
sqft_living15	0.4635	0.020	23.218	0.000	0.424	0.503
sqft_lot15	-0.0924	0.027	-3.486	0.000	-0.144	-0.040
year_sold	0.0106	0.008	1.280	0.201	-0.006	0.027
month_sold	0.0086	0.014	0.632	0.527	-0.018	0.035
age_sold	3.9109	0.042	94.086	0.000	3.829	3.992
bedrooms_3	-0.0321	0.008	-4.033	0.000	-0.048	-0.016
bedrooms_4	-0.0521	0.010	-5.354	0.000	-0.071	-0.033
bedrooms_5	-0.0706	0.015	-4.865	0.000	-0.099	-0.042
bathrooms_0.75	0.5217	0.057	9.211	0.000	0.411	0.633
bathrooms_1.0	0.4406	0.016	28.375	0.000	0.410	0.471
...

Predictors: 68 -> 56

OLS Regression Results

Dep. Variable:	price_log1p	R-squared:	0.597
Model:	OLS	Adj. R-squared:	0.595
Method:	Least Squares	F-statistic:	346.1
Date:	Thu, 18 Nov 2021	Prob (F-statistic):	0.00
Time:	22:28:20	Log-Likelihood:	-1338.6
No. Observations:	12664	AIC:	2787.
Df Residuals:	12609	BIC:	3197.
Df Model:	54		
Covariance Type:	nonrobust		

Modeling

- Regression model after further excluding some predictors with high collinearity

Predictors: 56 -> 30

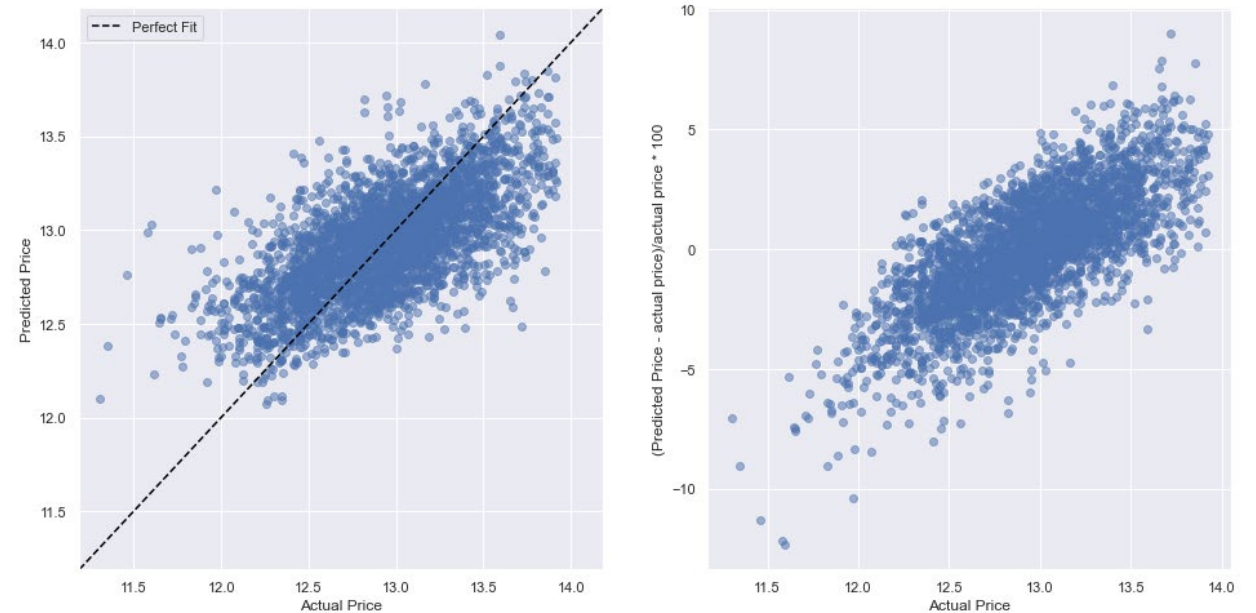
OLS Regression Results

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:	23:18:42	Log-Likelihood:	-3325.9
No. Observations:	12664	AIC:	6714.
Df Residuals:	12633	BIC:	6945.
Df Model:	30		
Covariance Type:	nonrobust		

```
['sqft_lot',  
'sqft_living15',  
'sqft_lot15',  
'bedrooms_3',  
'bedrooms_4',  
'bedrooms_5',  
'floors_2.5',  
'floors_3.0',  
'waterfront_1.0',  
'is_renovated_1.0',  
'view_1.0',  
'view_2.0',  
'view_3.0',  
'view_4.0',  
'grade_10',  
'grade_11',  
'zipcode4_98020',  
'zipcode4_98030',  
'zipcode4_98040',  
'zipcode4_98060',  
'zipcode4_98070',  
'zipcode4_98090',  
'zipcode4_98100',  
'zipcode4_98110',  
'zipcode4_98120',  
'zipcode4_98130',  
'zipcode4_98150',  
'zipcode4_98160',  
'zipcode4_98170',  
'zipcode4_98180']
```

Model evaluation

- Cross-validation
 - Train score: 0.4465
 - Test score: 0.4501
- Mean squared error, root mean squared error, mean absolute error, and R-squared value
 - MSE: 0.1007
 - RMSE: 0.3174
 - MAE: 0.2519
 - R-Squared: 0.4437
- Examine the predicted values with the real values



The fitted regression model can predict house price very well

Conclusion

- Observations from coefficients
 - The grade and sqft_living15 have the strongest relationship with the house price
 - It is interesting to see sqft_lot15 has a negative relationship with the house price
 - Waterfront_1.0 and grade_11 also have a positive relationship with the price
 - For some zip codes, e.g., 98100 and 98110, they have high positive relationships with the price
- To address the business questions:
 - For buyer, they will know that the house price is higher for a house with the high grade and sqrt_living15 values
 - For seller, if they want to sell their house with a higher price, they could add waterfront and improve the grade.

```
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
```