IA0

Yongsung Cho

Computer Science, Oregon State University

AI 534: Machine Learning

Prof. Fern

Due Oct 1, 2021

Part 1

Please check readme.txt file.

Part 2

(a)
Before removing id feature

	id	date	bedrooms	bathrooms	sqft_living	sqft_lot	floors	waterfront	view	condition	...	sqft_above	sqft_basement	yr_built	yr_renovat
0	3066410850	7/9/2014	4	2.50	2720	10006	2.0	0	0	3	...	2720	0	1989	
1	9345400350	7/18/2014	2	2.50	2600	5000	1.0	0	0	5	...	1300	1300	1926	
2	7128300060	7/7/2014	5	1.75	1650	3000	1.5	0	0	3	...	1650	0	1902	
3	2155500030	4/28/2015	4	1.75	1720	9600	1.0	0	0	4	...	1720	0	1969	
4	3999300080	9/4/2014	6	2.25	3830	11180	1.0	0	2	5	...	2440	1390	1962	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
9995	1523059103	9/26/2014	4	2.50	2570	22215	2.0	0	0	5	...	2570	0	1958	
9996	985001015	6/4/2014	1	1.00	790	13062	1.0	0	0	3	...	790	0	1942	
9997	1115100278	3/17/2015	3	1.50	1540	7506	1.0	0	0	5	...	1540	0	1961	
9998	8032700070	11/18/2014	3	2.25	1870	1900	3.0	0	0	3	...	1870	0	2008	
9999	3328500250	5/2/2014	4	2.50	2200	9397	2.0	0	0	3	...	2200	0	1987	

10000 rows × 21 columns

After removing id feature

```
table.drop('id', axis=1)
```

	date	bedrooms	bathrooms	sqft_living	sqft_lot	floors	waterfront	view	condition	grade	sqft_above	sqft_basement	yr_built	yr_renovated	zipcc
0	7/9/2014	4	2.50	2720	10006	2.0	0	0	3	9	2720	0	1989	0	98(
1	7/18/2014	2	2.50	2600	5000	1.0	0	0	5	8	1300	1300	1926	0	98
2	7/7/2014	5	1.75	1650	3000	1.5	0	0	3	8	1650	0	1902	0	98
3	4/28/2015	4	1.75	1720	9600	1.0	0	0	4	8	1720	0	1969	0	98(
4	9/4/2014	6	2.25	3830	11180	1.0	0	2	5	9	2440	1390	1962	0	98(
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
9995	9/26/2014	4	2.50	2570	22215	2.0	0	0	5	7	2570	0	1958	0	98(
9996	6/4/2014	1	1.00	790	13062	1.0	0	0	3	6	790	0	1942	0	98
9997	3/17/2015	3	1.50	1540	7506	1.0	0	0	5	7	1540	0	1961	0	98
9998	11/18/2014	3	2.25	1870	1900	3.0	0	0	3	8	1870	0	2008	0	98
9999	5/2/2014	4	2.50	2200	9397	2.0	0	0	3	8	2200	0	1987	0	98(

10000 rows × 20 columns

I've used pd.drop() function. It removed id feature from dataframe.

Question: Is it a good idea to use this feature in predicting the price of the house? why?

I think using the id feature in predicting the price of the house is essential. So, I shouldn't remove the id feature from this table (data frame). Because if I don't know the id, I can't figure out which one is the correct one. In addition, the duplicated row will be made because the id feature works as an identifier. If we are going to use database, the id feature is going to be key attribute. Therefore, the id feature is essential.

(b) table[['month','day','year']] = table['date'].str.split("/",expand=True)
table = table[['id','month','day','year','bedrooms','bathrooms','sqft_living','sqft_lot','floors','waterfront','view','condition','grade','sqft_above','sqft_basement','yr_built','yr_renovated','zipcode','lat','long','sqft_living15','sqft_lot15','price']]

First of all, I split date feature to month, day, and year. When I run first line of code, new features were on the last part of the table. So, I changed the order of data frame.

	month	day	year	bedrooms	bathrooms	sqft_living	sqft_lot	floors	waterfront	view	...	sqft_above	sqft_basement	yr_built	yr_renovated	zipcode	
0	7	9	2014	4	2.50	2720	10006	2.0	0	0	...	2720	0	1989	0	98074	47
1	7	18	2014	2	2.50	2600	5000	1.0	0	0	...	1300	1300	1926	0	98126	47
2	7	7	2014	5	1.75	1650	3000	1.5	0	0	...	1650	0	1902	0	98144	47
3	4	28	2015	4	1.75	1720	9600	1.0	0	0	...	1720	0	1969	0	98059	47
4	9	4	2014	6	2.25	3830	11180	1.0	0	2	...	2440	1390	1962	0	98008	47
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
9995	9	26	2014	4	2.50	2570	22215	2.0	0	0	...	2570	0	1958	0	98059	47
9996	6	4	2014	1	1.00	790	12062	1.0	0	0	...	790	0	1942	0	98168	47
9997	3	17	2015	3	1.50	1540	7506	1.0	0	0	...	1540	0	1961	0	98155	47
9998	11	18	2014	3	2.25	1870	1900	3.0	0	0	...	1870	0	2008	0	98103	47
9999	5	2	2014	4	2.50	2200	9397	2.0	0	0	...	2200	0	1987	0	98001	47

click to scroll output; double click to hide

10000 rows × 22 columns

This is a result data frame. Finally, I can compare the date of contract easily.

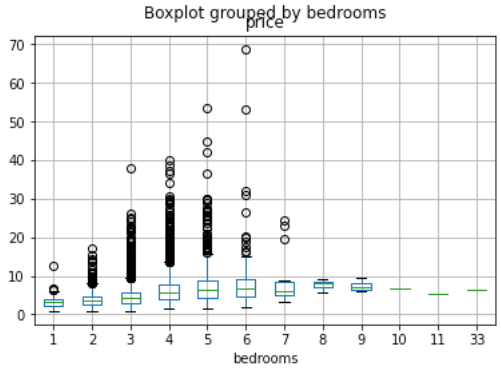
Question: date feature is useful for this problem? Can you think of better ways of using this date feature than splitting them into three numerical features?

I think date feature is useful for this problem. However, previous data feature was inconvenient for compare and calculate. So, splitting date feature is going to be good way to predicting the price of the house.

In the three numerical features, I think that the day feature is useless. Because the most important thing for sale is when it is sold. On the other hand, year and month is more important than day. I think that the price of the house isn't changed by the day of the date.

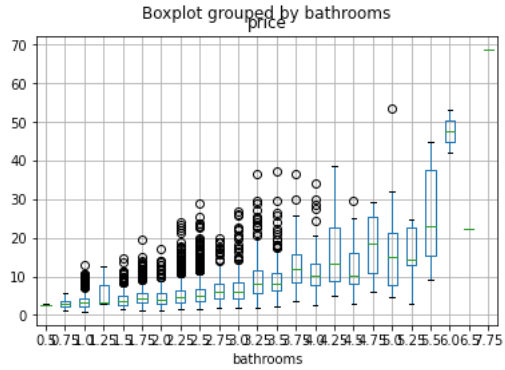
(c) Because of using matplotlib, I can see the plot at a glance and easily understand graphs.

Box plot grouped by bedrooms



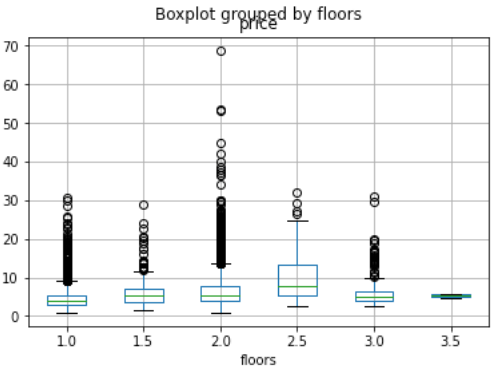
This plot shows that the house with many bedrooms is usually expensive.

Box plot grouped by bathrooms



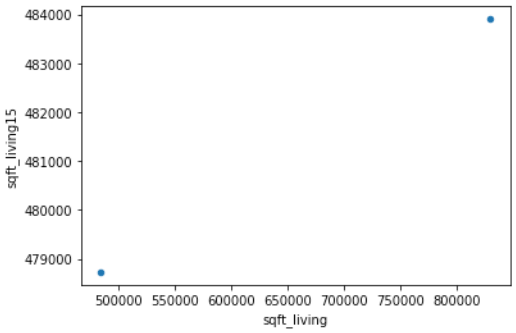
Most of the house have 1 to 4 bathrooms and the house with many bathrooms is usually expensive, like box plot grouped by bedrooms plot.

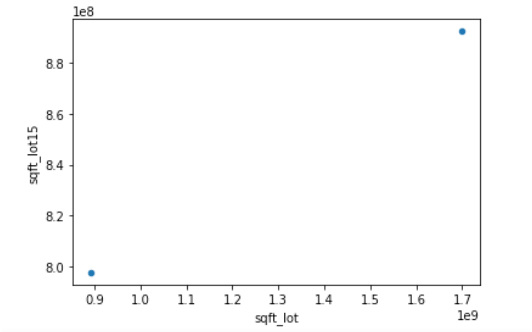
Box plot grouped by floors



The mean of the house with 2.5 floors is highest, but there are not many houses were sold. Most of the house that was sold have 1 or 2 floors.

(d) Scatter plot sqft_living against sqft_living15 / Scatter plot sqft_lot against sqft_lot15





I think this scatter plot doesn't have any meaning. Because co-variance matrix of sqft_living against sqft_living15 has only 4 numbers and it was too big or too small number for showing plot. Sqft_lot against sqft_lot15 co-variance matrix also has same result. Therefore, these features are redundant.