Business Anaytics

Assignment 03

15146314 Yang, Seunghyuck

Add categorical variables to variable set

Variable set from last assignment

```
#%% Training with bare setting
X = house[['bathrooms', 'sqft_living', 'grade', 'sqft_above',
           'sqft_basement', 'sqft_living15']]
y = house['price']
```

Result after adding categorical variables

OLS Regression Results

Dep. Variable:	price	R-squared:	0.544
Model:	OLS	Adj. R-squared:	0.544
Method:	Least Squares	F-statistic:	5160.
Date:	Sun, 20 Sep 2020	Prob (F-statistic):	0.00
Time:	22:46:48	Log-Likelihood:	-2.9911e+05
No. Observations:	21613	AIC:	5.982e+05
Df Residuals:	21607	BIC:	5.983e+05
Df Model:	5		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	-6.469e+05	1.35e+04	-47.870	0.000	-6.73e+05	-6.2e+05
bathrooms	-3.546e+04	3425.567	-10.353	0.000	-4.22e+04	-2.87e+04
sqft_living	136.7857	2.475	55.271	0.000	131.935	141.636
grade	1.11e+05	2462.309	45.090	0.000	1.06e+05	1.16e+05
sqft_above	28.1505	2.433	11.572	0.000	23.382	32.919
sqft_basement	108.6352	2.658	40.866	0.000	103.425	113.846
sqft_living15	22.8201	4.027	5.667	0.000	14.927	30.713

Omnibus:	17285.229	Durbin-Watson:	1.981
Prob(Omnibus):	0.000	Jarque-Bera (JB):	1134486.304
Skew:	3.366	Prob(JB):	0.00
Kurtosis:	37.849	Cond. No.	8.58e+15

1 2

Add categorical variables into dummy variables

```
X2 = X
varCtgr = ['view', 'condition', 'grade']
for c in varCtgr:
    dummy = pd.get_dummies(house[c], prefix = c, drop_first = True)
    X2 = pd.concat((X2, dummy), axis = 1)
```

Result before adding categorical variables

OLS Regression Results

Dep. Variable:	price	R-squared:	0.633
Model:	OLS	Adj. R-squared:	0.633
Method:	Least Squares	F-statistic:	1620.
Date:	Sun, 20 Sep 2020	Prob (F-statistic):	0.00
Time:	22:43:09	Log-Likelihood:	-2.9677e+05
No. Observations:	21613	AIC:	5.936e+05
Df Residuals:	21589	BIC:	5.938e+05
Df Model:	23		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	656.2243	2.49e+05	0.003	0.998	-4.87e+05	4.88e+05
bathrooms	1350.8798	3188.055	0.424	0.672	-4897.943	7599.703
sqft_living	93.7533	2.318	40.442	0.000	89.209	98.297
grade	9.078e+04	2.65e+04	3.426	0.001	3.88e+04	1.43e+05
sqft_above	23.5450	2.277	10.340	0.000	19.082	28.008
sqft_basement	70.2083	2.532	27.730	0.000	65.246	75.171
sqft_living15	10.2173	3.683	2.774	0.006	2.998	17.436
view_1	1.512e+05	1.24e+04	12.153	0.000	1.27e+05	1.76e+05
view_2	9.089e+04	7522.134	12.083	0.000	7.61e+04	1.06e+05
view_3	1.576e+05	1.03e+04	15.316	0.000	1.37e+05	1.78e+05
view_4	5.081e+05	1.3e+04	39.119	0.000	4.83e+05	5.34e+05
condition_2	-2.347e+04	4.48e+04	-0.524	0.600	-1.11e+05	6.44e+04
condition_3	-2.301e+04	4.17e+04	-0.552	0.581	-1.05e+05	5.87e+04
condition_4	2.085e+04	4.17e+04	0.500	0.617	-6.09e+04	1.03e+05
condition_5	9.786e+04	4.2e+04	2.332	0.020	1.56e+04	1.8e+05
grade_3	-1.667e+05	2.15e+05	-0.773	0.439	-5.89e+05	2.56e+05
grade_4	-2.504e+05	1.52e+05	-1.642	0.101	-5.49e+05	4.84e+04
grade_5	-3.613e+05	1.21e+05	-2.979	0.003	-5.99e+05	-1.24e+05
grade_6	-4.184e+05	9.46e+04	-4.425	0.000	-6.04e+05	-2.33e+05
grade_7	-4.734e+05	6.83e+04	-6.930	0.000	-6.07e+05	-3.39e+05
grade_8	-4.918e+05	4.25e+04	-11.573	0.000	-5.75e+05	-4.09e+05
grade_9	-4.48e+05	1.87e+04	-23.961	0.000	-4.85e+05	-4.11e+05
grade_10	-3.45e+05	1.75e+04	-19.721	0.000	-3.79e+05	-3.11e+05
grade_11	-1.531e+05	4.12e+04	-3.713	0.000	-2.34e+05	-7.23e+04
grade_12	2.297e+05	6.86e+04	3.347	0.001	9.52e+04	3.64e+05
grade_13	1.391e+06	1.03e+05	13.468	0.000	1.19e+06	1.59e+06

Omnibus:	13447.050	Durbin-Watson:	1.982
Prob(Omnibus):	0.000	Jarque-Bera (JB):	571338.641
Skew:	2.385	Prob(JB):	0.00
Kurtosis:	27.732	Cond. No.	1.08e+16

1 2 3

Interpretation

1.
R-squared value increased
F-statistic is decreased

➡ Statistical Uncertainty has increased

2.
P > | t | is increased significantly
for 'const', 'bathrooms'

➡ Statistical Uncertainty has increased

3.
P > | t | is unusually high
for some variables

➡ 1. Variable 'condition' is not appropriate factor to predict target
2. Variable 'grade' has potential as factor after preprocessing (removing 3, 4)

Ideas to utilize zipcode, lat, and long

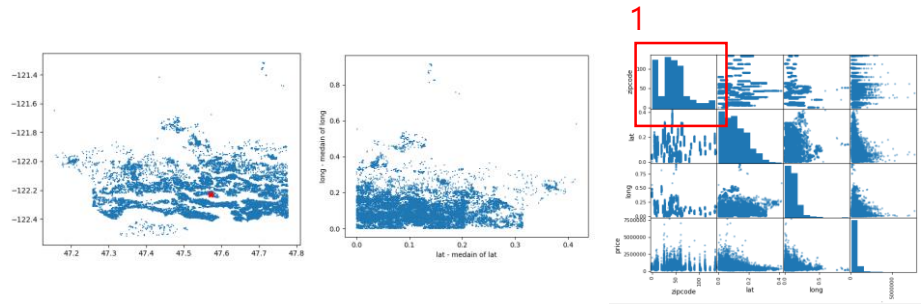
Without other resources:

They are related with location

➡ it would be possible
to rearrange them with new standard

➡ How about designating center and
rearrange them by distance from center?

➡
```
plt.scatter(house['lat'], house['long'], marker = '+', s = 1)
plt.scatter(house['lat'].median(), house['long'].median(), marker = 'X', s = 50, c ='r')
```
```
reLat = house['lat'].apply(lambda x: abs(house['lat'].median() - x))
reLong = house['long'].apply(lambda x: abs(house['long'].median() - x))
plt.scatter(reLat, reLong, marker = '+', s = 1)
plt.xlabel("lat - medain of lat")
plt.ylabel("long - medain of long")
```

➡ 

Result
I might be able to predict correlation between 'price' and 'distance'
only in case of far from center.

1.
Even after arrange variables based on center,
they are not well centralized

➡ Maybe we can split them
into two groups
By using k-means

With other resources:

Problem is how to distinguish in same location

➡ Rural area, which means far away from center, is secured
How about near center?

➡ Need criterion to use in center to classify:
Possibly size of residual space

➡ Especially in urban area, size of house is directly connected
to price of house

➡ Other resources which could be regarded:
'sqft_xxx'