연습문제 (2) - 2 - 추가 내용

`neighborhood` 변수를 포함한 경우에 대한 결과 체크

```
from pandas import read_excel, DataFrame, merge, melt
from matplotlib import pyplot as plt
import seaborn as sb


import sys
import os
sys.path.append(os.path.dirname(os.path.dirname(os.getcwd())))
from helper import my_ols, scalling, get_best_features, setCategory
```

```
df = read_excel("https://data.hossam.kr/E04/manhattan.xlsx")
df
```

	rent	bedrooms	bathrooms	size_sqft	min_to_subway	floor	building_
0	2550	0.0	1	480	9	2.0	17
1	11500	2.0	2	2000	4	1.0	96
2	4500	1.0	1	916	2	51.0	29

	rent	bedrooms	bathrooms	size_sqft	min_to_subway	floor	building_
3	4795	1.0	1	975	3	8.0	31
4	17500	2.0	2	4800	3	4.0	136
...	...	...	...	...	...	...	...
3534	4210	1.0	1	532	3	8.0	16
3535	6675	2.0	2	988	5	10.0	9
3536	1699	0.0	1	250	2	5.0	96
3537	3475	1.0	1	651	6	5.0	14
3538	4500	1.0	1	816	4	11.0	9

3539 rows × 17 columns

```
df.value_counts('neighborhood')
```

```
neighborhood
Upper West Side       579
Upper East Side       500
Midtown East          460
Midtown West          314
Financial District    268
Chelsea               182
Flatiron              132
```

```
Tribeca                  119
Midtown                  119
East Village             108
Battery Park City        104
Midtown South             85
Central Harlem            82
West Village              67
Greenwich Village         66
Gramercy Park             61
Soho                      58
Washington Heights        54
Lower East Side           41
East Harlem               41
Central Park South        23
Hamilton Heights          16
Morningside Heights       13
Inwood                    12
Nolita                     9
Chinatown                  8
Roosevelt Island           5
Long Island City           4
Stuyvesant Town/PCV        3
Little Italy               3
West Harlem                2
Manhattanville             1
Name: count, dtype: int64
```

```
df2 = df.drop('borough', axis=1)
df3 = setCategory(df2, 'neighborhood')
```

```
df3.value_counts('neighborhood')
```

```
neighborhood
28    579
27    500
18    460
20    314
7     268
3     182
8     132
26    119
17    119
6     108
0     104
19     85
1      82
31     67
10     66
9      61
24     58
29     54
15     41
5      41
2      23
11     16
21     13
12     12
22      9
4       8
23      5
```

```
14      4
25      3
13      3
30      2
16      1
Name: count, dtype: int64
```

```
x_train_std_df, y_train_std_df = scalling(df3, 'rent')
```

```
x_train_std_df.head()
```

	bedrooms	bathrooms	size_sqft	min_to_subway	floor	building_age_yr
0	-1.397410	-0.611790	-0.962011	0.730862	-0.904097	-0.888763
1	0.669863	1.056257	2.218694	-0.176116	-0.995343	1.117593
2	-0.363774	-0.611790	-0.049651	-0.538908	3.566974	-0.584000
3	-0.363774	-0.611790	0.073811	-0.357512	-0.356619	-0.533206
4	0.669863	1.056257	8.077886	-0.357512	-0.721604	2.133470

```
y_train_std_df.head()
```

	rent
0	-0.818669

	rent
1	2.011480
2	-0.202044
3	-0.108760
4	3.908786

```
feature, topfeat_df = get_best_features(x_train_std_df)
feature
```

```
[pca] >Extracting column labels from dataframe.
[pca] >Extracting row labels from dataframe.
[pca] >The PCA reduction is performed to capture [95.0%] explained varia
[pca] >Fit using PCA.
[pca] >Compute loadings and PCs.
[pca] >Compute explained variance.
[pca] >Number of components is [12] that covers the [95.00%] explained v
[pca] >The PCA reduction is performed on the [15] columns of the input d
[pca] >Fit using PCA.
[pca] >Compute loadings and PCs.
[pca] >Outlier detection using Hotelling T2 test with alpha=[0.05] and n
[pca] >Multiple test correction applied for Hotelling T2 test: [fdr_bh]
[pca] >Outlier detection using SPE/DmodX with n_std=[3]
```

```
['no_fee',
 'has_elevator',
 'min_to_subway',
 'size_sqft',
 'has_roofdeck',
 'has_gym',
 'has_patio',
 'neighborhood',
 'has_dishwasher',
 'building_age_yrs']
```

```
mdf = merge(x_train_std_df, y_train_std_df, left_index=True, right_index
ols_result = my_ols(mdf, y='rent', x=feature)
ols_result.summary
```

OLS Regression Results

Dep. Variable:	rent	**R-squared:**	0.757
Model:	OLS	**Adj. R-squared:**	0.756
Method:	Least Squares	**F-statistic:**	1098.
Date:	Thu, 27 Jul 2023	**Prob (F-statistic):**	0.00
Time:	10:49:22	**Log-Likelihood:**	-2519.3
No. Observations:	3539	**AIC:**	5061.
Df Residuals:	3528	**BIC:**	5129.

Df Model:	10		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	-9.346e-17	0.008	-1.13e-14	1.000	-0.016	0.016
no_fee	-0.0095	0.009	-1.087	0.277	-0.027	0.008
has_elevator	0.0036	0.012	0.308	0.758	-0.019	0.026
min_to_subway	-0.0252	0.008	-2.970	0.003	-0.042	-0.009
size_sqft	0.8593	0.008	101.882	0.000	0.843	0.876
has_roofdeck	0.0040	0.010	0.383	0.701	-0.016	0.024
has_gym	-1.995e-05	0.012	-0.002	0.999	-0.023	0.023
has_patio	0.0015	0.008	0.179	0.858	-0.015	0.018
neighborhood	0.0008	0.008	0.100	0.921	-0.016	0.017
has_dishwasher	0.0008	0.009	0.087	0.931	-0.017	0.019
building_age_yrs	-0.1477	0.009	-16.951	0.000	-0.165	-0.131

Omnibus:	885.489	**Durbin-Watson:**	2.058
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	11063.662
Skew:	0.831	**Prob(JB):**	0.00
Kurtosis:	11.501	**Cond. No.**	2.71

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