House Price Prediction

程序执行

- 将regression与dataset文件夹放置在同一目录下，配置好pycharm和s3的连接信息，执行regression_house.py。
- 基本信息输出

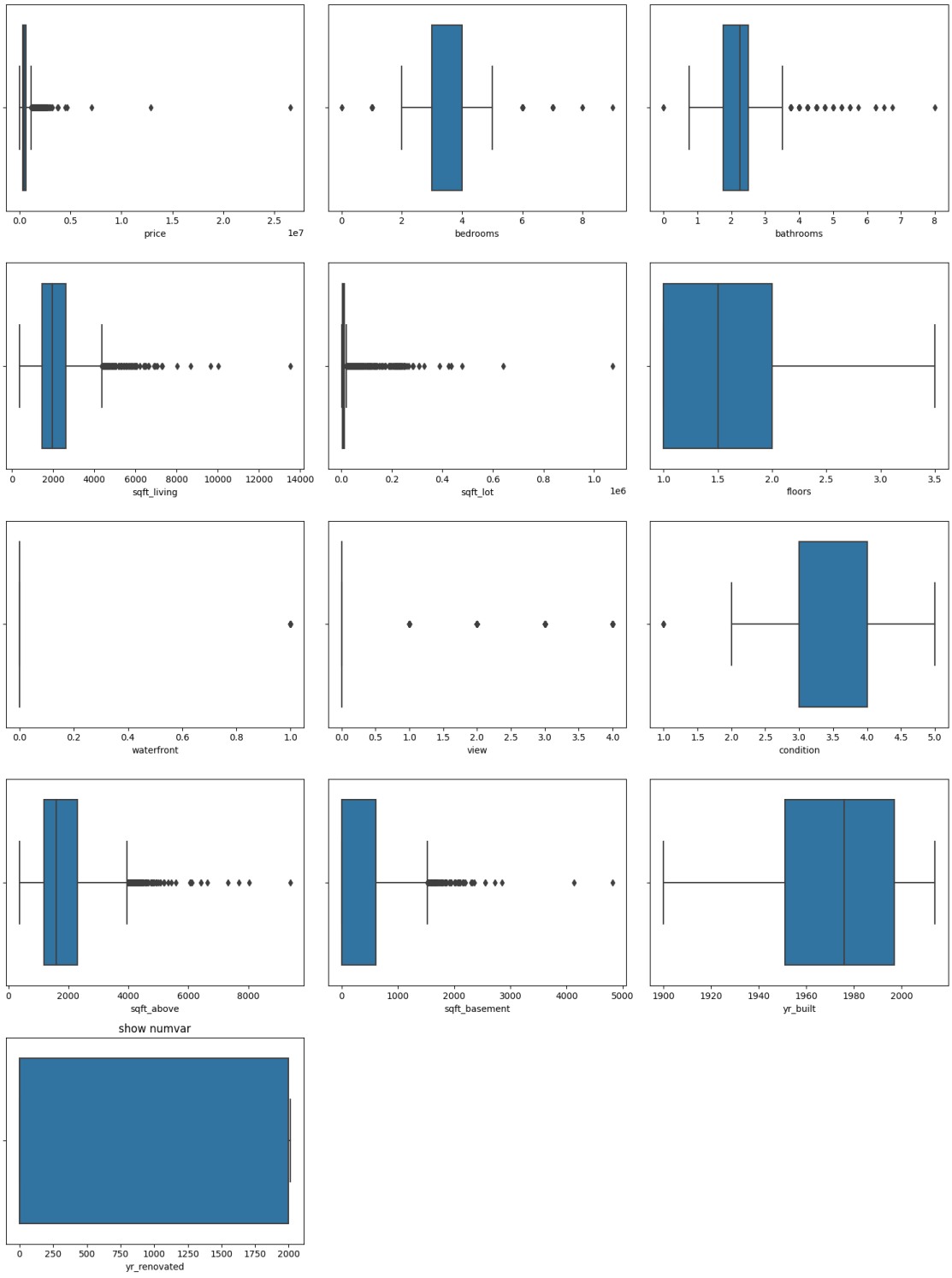
```
基本信息
                   date         price  bedrooms  ...       city  statezip  country
0     2014-05-02 00:00:00  3.130000e+05       3.0  ...  Shoreline  WA 98133      USA
1     2014-05-02 00:00:00  2.384000e+06       5.0  ...    Seattle  WA 98119      USA
2     2014-05-02 00:00:00  3.420000e+05       3.0  ...       Kent  WA 98042      USA
3     2014-05-02 00:00:00  4.200000e+05       3.0  ...   Bellevue  WA 98008      USA
4     2014-05-02 00:00:00  5.500000e+05       4.0  ...    Redmond  WA 98052      USA
...                   ...           ...       ...  ...        ...       ...      ...
4595  2014-07-09 00:00:00  3.081667e+05       3.0  ...    Seattle  WA 98133      USA
4596  2014-07-09 00:00:00  5.343333e+05       3.0  ...   Bellevue  WA 98007      USA
4597  2014-07-09 00:00:00  4.169042e+05       3.0  ...     Renton  WA 98059      USA
4598  2014-07-10 00:00:00  2.034000e+05       4.0  ...    Seattle  WA 98178      USA
4599  2014-07-10 00:00:00  2.206000e+05       3.0  ...  Covington  WA 98042      USA

[4600 rows x 18 columns]
空数据信息
date             0
price            0
bedrooms         0
bathrooms        0
sqft_living      0
sqft_lot         0
floors           0
waterfront       0
view             0
condition        0
sqft_above       0
sqft_basement    0
yr_built         0
yr_renovated     0
street           0
city             0
statezip         0
```

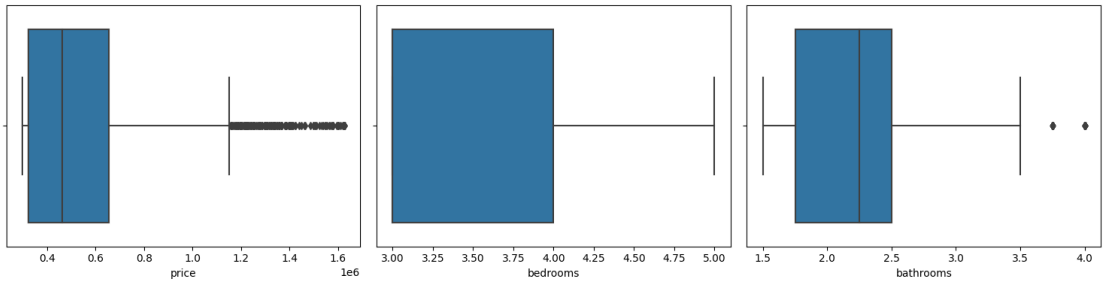
```
数字信息
              price     bedrooms  ...     yr_built  yr_renovated
count  4.600000e+03  4600.000000  ...  4600.000000   4600.000000
mean   5.519630e+05     3.400870  ...  1970.786304    808.608261
std    5.638347e+05     0.908848  ...    29.731848    979.414536
min    0.000000e+00     0.000000  ...  1900.000000      0.000000
25%    3.228750e+05     3.000000  ...  1951.000000      0.000000
50%    4.609435e+05     3.000000  ...  1976.000000      0.000000
75%    6.549625e+05     4.000000  ...  1997.000000   1999.000000
max    2.659000e+07     9.000000  ...  2014.000000   2014.000000

[8 rows x 13 columns]
非数字信息
                        date                street     city  statezip country
count                   4600                  4600     4600      4600    4600
unique                    70                  4525       44        77       1
top     2014-06-23 00:00:00  2520 Mulberry Walk NE  Seattle  WA 98103     USA
freq                     142                     4     1573       148    4600
r2s=0.4918357881060522
```

- 剔除离群值前属性数值分布输出

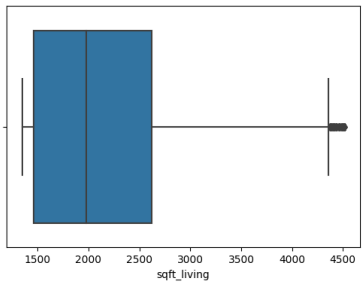


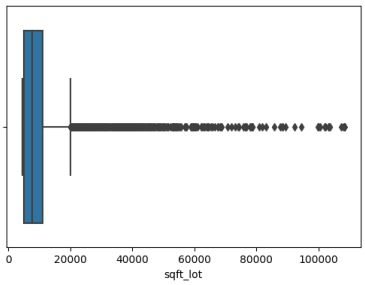
- 剔除离群值后属性数据分布输出

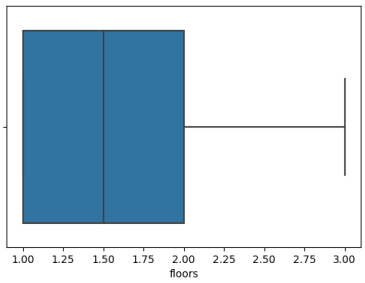

0.4 0.6 0.8 1.0 1.2 1.4 1.6
price
1e6

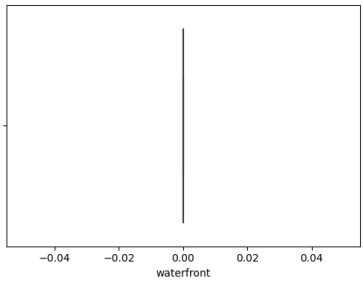
3.00 3.25 3.50 3.75 4.00 4.25 4.50 4.75 5.00
bedrooms

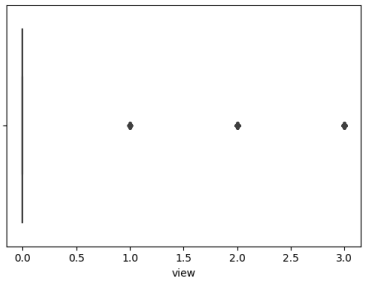
1.5 2.0 2.5 3.0 3.5 4.0
bathrooms

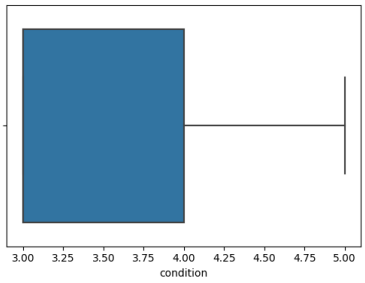

1500 2000 2500 3000 3500 4000 4500
sqft_living

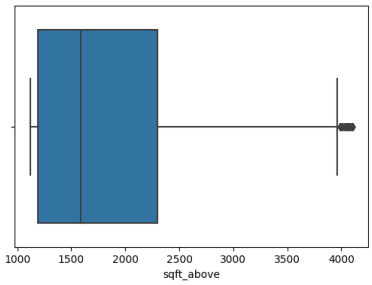

0 20000 40000 60000 80000 100000
sqft_lot

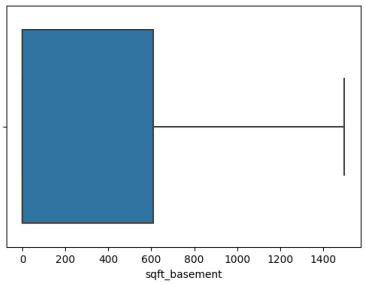

1.00 1.25 1.50 1.75 2.00 2.25 2.50 2.75 3.00
floors

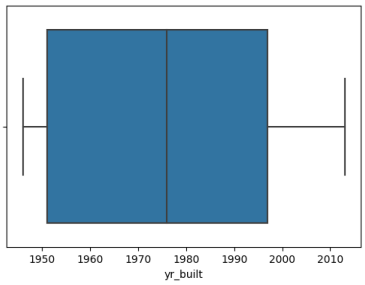

−0.04 −0.02 0.00 0.02 0.04
waterfront

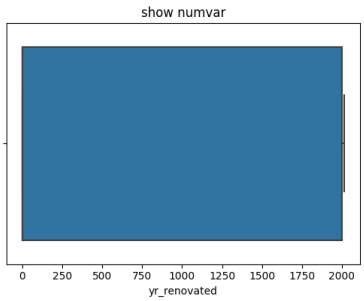

0.0 0.5 1.0 1.5 2.0 2.5 3.0
view


3.00 3.25 3.50 3.75 4.00 4.25 4.50 4.75 5.00
condition

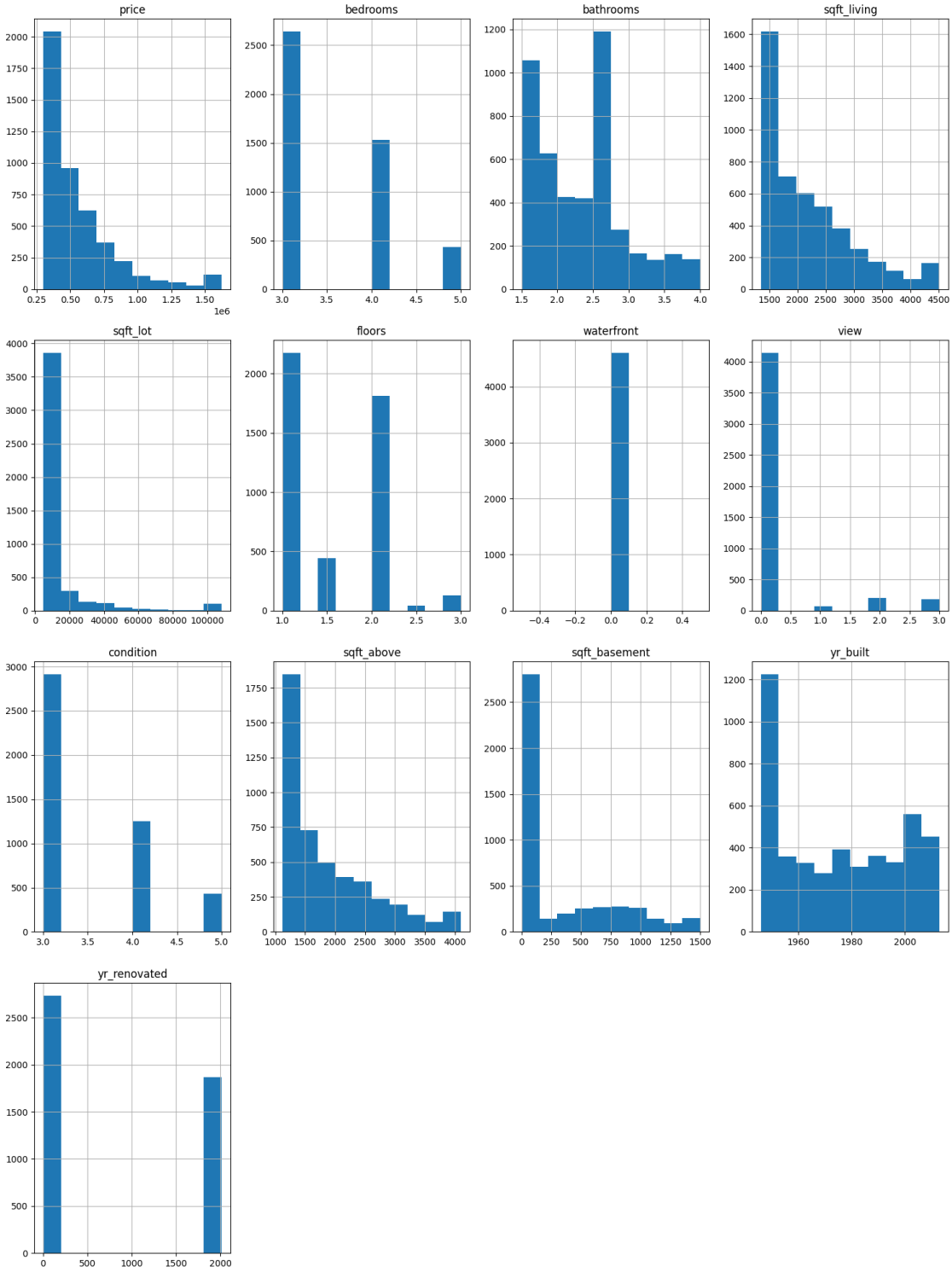

1000 1500 2000 2500 3000 3500 4000
sqft_above


0 200 400 600 800 1000 1200 1400
sqft_basement

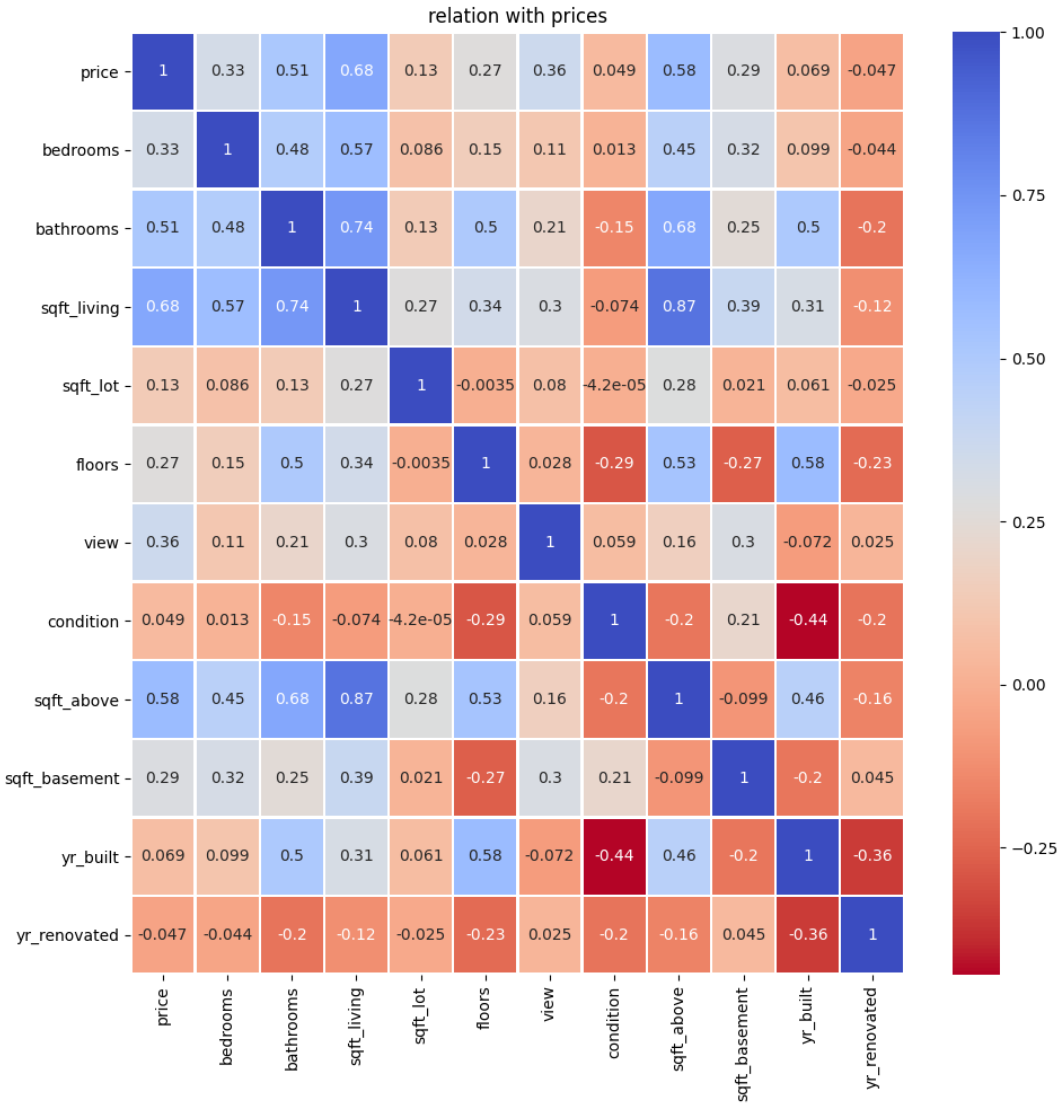

1950 1960 1970 1980 1990 2000 2010
yr_built


show numvar
0 250 500 750 1000 1250 1500 1750 2000
yr_renovated

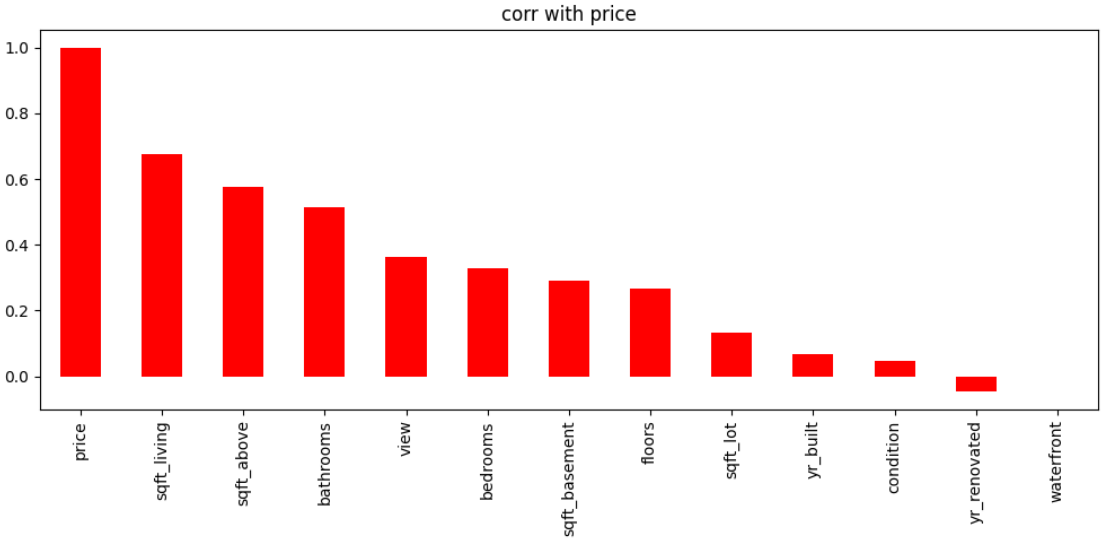
- 直方图分析输出



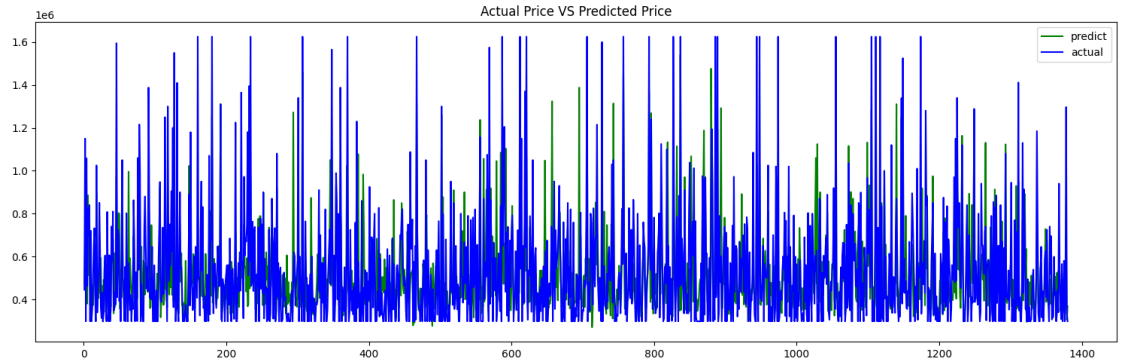
- 属性两两间相关性分析输出



- 各属性按照price的相关性排序结果



- 模型训练后预测效果



- r2score

```
r2s=0.5321643069270936
```

- 结果csv存在s3中

```
[user29@i-D6BAA5B0 homework]$ ls
dataset   __init__.py  regression_house.py  result.csv
```

- 预测结果

num	actual	predict
0	572000	402191.7
1	559950	363604.6
2	1625200	939293.1
3	355000	452829.6
4	385000	456116.2
5	819000	943405
6	395000	585080
7	297785.7	425231.9
8	335000	523959.9
9	605000	483538.2
10	645000	515185.3
11	297785.7	387446.5
12	491234	350260.9
13	581000	521608.7
14	642000	550022.6
15	806000	410616.7
16	548800	424393.3
17	467000	453213.5
18	345000	444349.7
19	955000	1210394
20	613000	378940.1

21	510000	350766.5
22	562000	632164.8
23	379900	380459.3
24	358000	343449.3
25	297785.7	347127.1
26	436500	366711.3
27	297785.7	341029.5
28	550000	435772.1
29	460000	451565.6
30	613000	421644.9
31	495000	802207.7
32	941000	582405.6
33	532000	409007.5

分析预测流程

基本信息分析

- 基本信息
- 查看空数据信息
- 数字信息
- 非数字信息

```
def showEda(df):
    print("基本信息")
    print(df)
    print("空数据信息")
    print(df.isnull().sum())
    print("数字信息")
    print(df.describe(include=np.number))
    print("非数字信息")
    print(df.describe(include=object))
```

单变量分析

- 剔除离群值

```
def outlier_tret(x):
    upper=x.quantile(0.98)
    lower=x.quantile(0.2)
    x=np.where(x>upper,upper,x)
    x=np.where(x<lower,lower,x)
    return x
data1[num_var] = data1[num_var].apply(lambda x: outlier_tret(x))
```

- 直方图分析

```
def showHist(data):
    data1=data.copy()
    data1.hist(figsize=(15, 20))
    plt.title("hist")
    plt.show()
```

- 剔除取值过多或过少的属性

```
x = df.drop(['date', 'price', 'city', 'street', 'statezip', 'country'],
axis=1)
```

双变量分析

- 用seaborn对变量两两之间的关系进行分析

```
def relaAna(data):
    data1=data.copy()
    plt.figure(figsize=(10, 10))
    sns.heatmap(data=data1.drop(columns='waterfront').corr(), linewidths=1,
cmap='coolwarm_r', annot=True)
    plt.title("relation with prices")
    plt.show()
```

- 将各属性按照price的相关性进行排序

```
def showCorr():
    corr = data1.corr()["price"].sort_values(ascending=False)
    plt.figure(figsize=(10, 5))
    corr.plot(kind='bar', color='red')
    plt.title("corr with price")
    plt.show()
```

选择模型并训练绘图

```
def train_predict(trainx,trainy,testx,testy):
    model.fit(trainx,trainy)
    pred1=model.predict(testx)
    np_testx=np.array(testx)
    np_testy=np.array(testy)
```

```
    np_predy=np.array(pred1)
    np_predy=np.exp(np_predy)-1
    data={
        "num":np.array(range(testx.shape[0])),
        "actual":np_testy,
        "predict":np_predy
    }
    dfy=DataFrame(data)
    dfy.to_csv("./result.csv")
    r2s=r2_score(np_testy,np_predy)
    num=np_testx.shape[0]
    plt.figure(figsize=(15, 5))

plt.plot(np.linspace(1.0,num,num=num),np_predy,color='green',label="predict")
    plt.plot(np.linspace(1.0,num,num=num),np_testy,color='blue',label="actual")
    plt.title("Actual Price VS Predicted Price")
    plt.legend()
    plt.show()
    print("r2s="+str(r2s))
```