```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
import seaborn as sns
from sklearn.pipeline import Pipeline
from sklearn.linear_model import LinearRegression
from sklearn.preprocessing import StandardScaler,PolynomialFeatures
from sklearn.model_selection import cross_val_score
from sklearn.model_selection import train_test_split
from sklearn.linear_model import Ridge
%matplotlib inline

from google.colab import files
uploaded=files.upload()
```

Choose Files No file chosen Upload widget is only available when the cell has been executed in the current browser session. Please rerun this cell to enable.

```
Saving kc_house_data.csv to kc_house_data (3).csv
```

```
df=pd.read_csv('kc_house_data.csv')
df.head()
```

	id	date	price	bedrooms	bathrooms	sqft_living	sqft_lot
0	7129300520	20141013T000000	221900.0	3	1.00	1180	5650
1	6414100192	20141209T000000	538000.0	3	2.25	2570	7242
2	5631500400	20150225T000000	180000.0	2	1.00	770	10000
3	2487200875	20141209T000000	604000.0	4	3.00	1960	5000
4	1954400510	20150218T000000	510000.0	3	2.00	1680	8080

```
df.dtypes
```

```
id                 int64
date              object
price            float64
bedrooms           int64
bathrooms        float64
sqft_living        int64
sqft_lot           int64
floors           float64
waterfront         int64
view               int64
condition          int64
grade              int64
sqft_above         int64
sqft_basement      int64
yr_built           int64
```

```
yr_renovated        int64
zipcode             int64
lat               float64
long              float64
sqft_living15       int64
sqft_lot15          int64
dtype: object
```

```
df.drop(["id","Unnamed: 0"] , axis = 1, inplace = True)
df.describe()
```

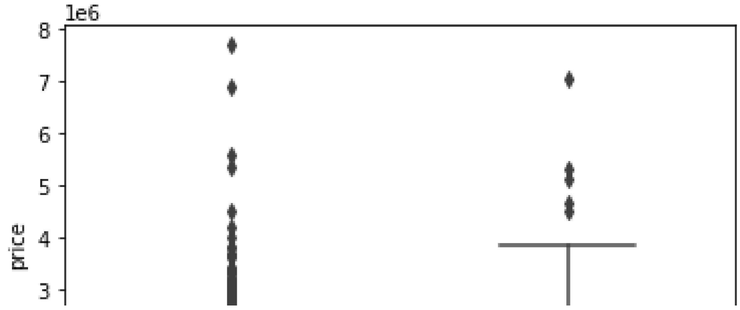
	id	price	bedrooms	bathrooms	sqft_living	sqft_
count	2.161300e+04	2.161300e+04	21613.000000	21613.000000	21613.000000	2.161300e-
mean	4.580302e+09	5.400881e+05	3.370842	2.114757	2079.899736	1.510697e-
std	2.876566e+09	3.671272e+05	0.930062	0.770163	918.440897	4.142051e-
min	1.000102e+06	7.500000e+04	0.000000	0.000000	290.000000	5.200000e-
25%	2.123049e+09	3.219500e+05	3.000000	1.750000	1427.000000	5.040000e-
50%	3.904930e+09	4.500000e+05	3.000000	2.250000	1910.000000	7.618000e-
75%	7.308900e+09	6.450000e+05	4.000000	2.500000	2550.000000	1.068800e-
max	9.900000e+09	7.700000e+06	33.000000	8.000000	13540.000000	1.651359e-

```
df["floors"].value_counts().to_frame()
```

	floors
1.0	10680
2.0	8241
1.5	1910
3.0	613
2.5	161
3.5	8

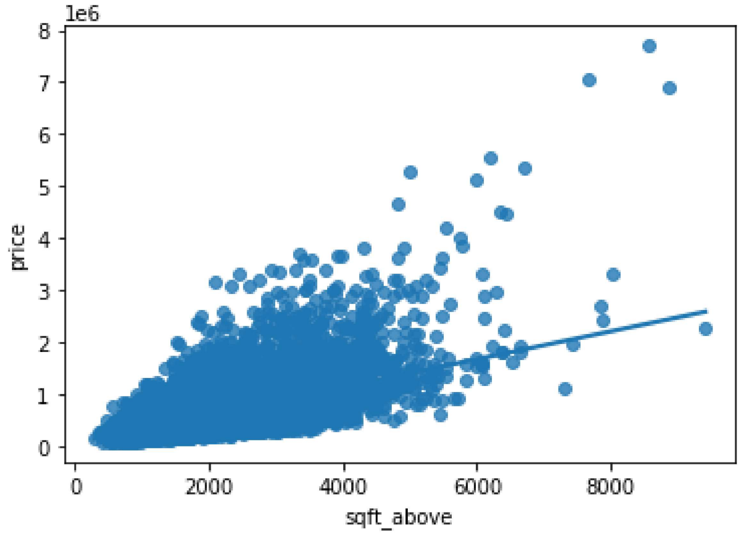
```
sns.boxplot(x="waterfront", y="price", data=df)
```

```
    <matplotlib.axes._subplots.AxesSubplot at 0x7f30e07ec610>
```



```
sns.regplot(x="sqft_above", y="price", data=df, ci = None)
```

```
    <matplotlib.axes._subplots.AxesSubplot at 0x7f30e06cc350>
```



```
X1 = df[['sqft_living']]
Y1 = df['price']
lm = LinearRegression()
lm
lm.fit(X1,Y1)
lm.score(X1, Y1)
```

```
    0.4928532179037931
```

```
features =["floors", "waterfront","lat" ,"bedrooms" ,"sqft_basement" ,"view" ,"bathrooms",
X = df[features]
Y = df['price']
lm.fit(X,Y)
lm.score(X,Y)
```

```
    0.6577027577865877
```

```
Input=[('scale',StandardScaler()),('polynomial', PolynomialFeatures(include_bias=False)),(
pipe=Pipeline(Input)
pipe
pipe.fit(X,Y)
pipe.score(X,Y)
features =["floors", "waterfront","lat" ,"bedrooms" ,"sqft_basement" ,"view" ,"bathrooms",
X = df[features ]
Y = df['price']

x_train, x_test, y_train, y_test = train_test_split(X, Y, test_size=0.15, random_state=1)

print("number of test samples :", x_test.shape[0])
print("number of training samples:",x_train.shape[0])
```

```
     number of test samples : 3242
     number of training samples: 18371
```

```
RigeModel = Ridge(alpha=0.1)
RigeModel.fit(x_train, y_train)
RigeModel.score(x_test, y_test)
```

```
     0.6480374087702245
```

```
pr=PolynomialFeatures(degree=2)
x_train_pr=pr.fit_transform(x_train[features])
x_test_pr=pr.fit_transform(x_test[features])

RigeModel = Ridge(alpha=0.1)
RigeModel.fit(x_train_pr, y_train)
RigeModel.score(x_test_pr, y_test)
```

```
     0.7004432058878023
```