

In [6]:

```
import pandas as pd
import numpy as np
housing=pd.read_csv("Z:/College/3.2/ML LAB/Final lab/datasets/Housing.csv")
housing.head()
print(housing.info())
print(housing.describe())
varlist=['mainroad','guestroom','basement','hotwaterheating','airconditioning','prefarea']

def binary_map(x):
    return x.map({'yes':1,'no':0})

housing[varlist]=housing[varlist].apply(binary_map)
housing
status=pd.get_dummies(housing['furnishingstatus'],drop_first=True)
status
housing=pd.concat([housing,status],axis=1)
housing.drop(['furnishingstatus'],axis=1,inplace=True)
housing

from sklearn.model_selection import train_test_split
np.random.seed(0)
df_train,df_test=train_test_split(housing,train_size=0.7,test_size=0.3,random_state=100)

from sklearn.preprocessing import MinMaxScaler
scaler=MinMaxScaler()
num_vars=['area','bedrooms','bathrooms','stories','parking','price']
df_train[num_vars]=scaler.fit_transform(df_train[num_vars])
df_train
y_train=df_train.pop('price')
x_train=df_train

import statsmodels.api as sm
x_train_lm=sm.add_constant(x_train)
lr_1=sm.OLS(y_train,x_train_lm).fit()
lr_1.summary()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 545 entries, 0 to 544
Data columns (total 13 columns):
price                545 non-null int64
area                 545 non-null int64
bedrooms             545 non-null int64
bathrooms            545 non-null int64
stories              545 non-null int64
mainroad             545 non-null object
guestroom            545 non-null object
basement             545 non-null object
hotwaterheating      545 non-null object
airconditioning      545 non-null object
parking              545 non-null int64
prefarea             545 non-null object
furnishingstatus     545 non-null object
dtypes: int64(6), object(7)
memory usage: 55.4+ KB
None
```

	price	area	bedrooms	bathrooms	stories \
count	5.450000e+02	545.000000	545.000000	545.000000	545.000000
mean	4.766729e+06	5150.541284	2.965138	1.286239	1.805505
std	1.870440e+06	2170.141023	0.738064	0.502470	0.867492
min	1.750000e+06	1650.000000	1.000000	1.000000	1.000000
25%	3.430000e+06	3600.000000	2.000000	1.000000	1.000000
50%	4.340000e+06	4600.000000	3.000000	1.000000	2.000000
75%	5.740000e+06	6360.000000	3.000000	2.000000	2.000000
max	1.330000e+07	16200.000000	6.000000	4.000000	4.000000

	parking
count	545.000000
mean	0.693578
std	0.861586
min	0.000000
25%	0.000000
50%	0.000000
75%	1.000000
max	3.000000

Out[6]:

OLS Regression Results

Dep. Variable:	price	R-squared:	0.681
Model:	OLS	Adj. R-squared:	0.670
Method:	Least Squares	F-statistic:	60.40
Date:	Wed, 21 Jun 2023	Prob (F-statistic):	8.83e-83
Time:	04:32:58	Log-Likelihood:	381.79
No. Observations:	381	AIC:	-735.6
Df Residuals:	367	BIC:	-680.4
Df Model:	13		
Covariance Type:	nonrobust		

	coef	std err	t	P> t	[0.025	0.975]
const	0.0200	0.021	0.955	0.340	-0.021	0.061
area	0.2347	0.030	7.795	0.000	0.175	0.294
bedrooms	0.0467	0.037	1.267	0.206	-0.026	0.119
bathrooms	0.1908	0.022	8.679	0.000	0.148	0.234
stories	0.1085	0.019	5.661	0.000	0.071	0.146
mainroad	0.0504	0.014	3.520	0.000	0.022	0.079
guestroom	0.0304	0.014	2.233	0.026	0.004	0.057
basement	0.0216	0.011	1.943	0.053	-0.000	0.043
hotwaterheating	0.0849	0.022	3.934	0.000	0.042	0.127
airconditioning	0.0669	0.011	5.899	0.000	0.045	0.089
parking	0.0607	0.018	3.365	0.001	0.025	0.096
prefarea	0.0594	0.012	5.040	0.000	0.036	0.083
semi-furnished	0.0009	0.012	0.078	0.938	-0.022	0.024
unfurnished	-0.0310	0.013	-2.440	0.015	-0.056	-0.006