MLR_HOUSING

```
import numpy as np

import pandas as pd

import seaborn as sns

import matplotlib.pyplot as plt


%matplotlib inline


#importing dataset using panda

dataset = pd.read_csv(r"D:\Samson - All Data\Naresh IT Institute\New
folder\House_data.csv")


#checking if any value is missing

print(dataset.isnull().any())


#checking for categorical data

print(dataset.dtypes)


# Dropping the id and date column

dataset = dataset.drop(['id','date'], axis = 1)


#understanding the distribution with seaborn

with sns.plotting_context("notebook",font_scale=0.8):

   g = sns.pairplot(dataset[['sqft_lot','sqft_above','price','sqft_living','bedrooms']],

             hue='bedrooms', palette='tab20',height=1)

g.set(xticklabels=[]);
```

```
#separating independent and dependent variable
X = dataset.iloc[:,1:].values
y = dataset.iloc[:,0].values
#splitting dataset into training and testing dataset
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.2, random_state = 0)

from sklearn.linear_model import LinearRegression
regressor = LinearRegression()
regressor.fit(X_train, y_train)

# Predicting the Test set results
y_pred = regressor.predict(X_test)

#Backward Elimination
import statsmodels.api as sm
def backwardElimination(x, SL):
    numVars = len(x[0])
    temp = np.zeros((21613,19)).astype(int)
    for i in range(0, numVars):
        regressor_OLS = sm.OLS(y, x).fit()
        maxVar = max(regressor_OLS.pvalues).astype(float)
        adjR_before = regressor_OLS.rsquared_adj.astype(float)
        if maxVar > SL:
            for j in range(0, numVars - i):
                if (regressor_OLS.pvalues[j].astype(float) == maxVar):
                    temp[:,j] = x[:, j]
                    x = np.delete(x, j, 1)
            tmp_regressor = sm.OLS(y, x).fit()
            adjR_after = tmp_regressor.rsquared_adj.astype(float)
            if (adjR_before >= adjR_after):
                x_rollback = np.hstack((x, temp[:,[0,j]]))
                x_rollback = np.delete(x_rollback, j, 1)
                print (regressor_OLS.summary())
                return x_rollback
            else:
                continue
    regressor_OLS.summary()
    return x

SL = 0.05
X_opt = X[:, [0, 1, 2, 3, 4, 5,6,7,8,9,10,11,12,13,14,15,16,17]]
X_Modeled = backwardElimination(X_opt, SL)
```