House Price Prediction (MLR)

```
# Import neccessary packages
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
%matplotlib inline

# Import dataset
house=pd.read_csv(r"D:\Full Stack Data Science\4 Sep (Multiple
Regression)\MLR\House_data.csv")
house

# Splitting data dependent and independent
x=house.iloc[:,3:]
x

y=house.iloc[:,2]
y

# Splitting data into train and test
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)

# Fit Multiple linear regression for training data
from sklearn.linear_model import LinearRegression
model=LinearRegression()
model.fit(x_train,y_train)

# Predicting the Test set
y_pred=model.predict(x_test)

# Building model using Backward Elimination
import statsmodels.formula.api as sm
x=np.append(arr=np.ones((21613,1)).astype(int),values=x,axis=1)


import statsmodels.api as sm
x_opt=x[:,[0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18]]

# Ordinal Least Square
OLS=sm.OLS(endog=y,exog=x_opt).fit()
OLS.summary()
```

House Price Prediction (MLR)

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                  price   R-squared:                       0.700
Model:                            OLS   Adj. R-squared:                  0.700
Method:                 Least Squares   F-statistic:                     2960.
Date:                Mon, 04 Sep 2023   Prob (F-statistic):               0.00
Time:                        22:14:47   Log-Likelihood:            -2.9460e+05
No. Observations:               21613   AIC:                         5.892e+05
Df Residuals:                   21595   BIC:                         5.894e+05
Df Model:                          17
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const        6.69e+06   2.93e+06      2.282      0.022    9.44e+05    1.24e+07
x1         -3.577e+04   1891.843    -18.906      0.000   -3.95e+04   -3.21e+04
x2          4.114e+04   3253.678     12.645      0.000    3.48e+04    4.75e+04
x3           110.4417      2.270     48.661      0.000     105.993     114.890
x4             0.1286      0.048      2.683      0.007       0.035       0.223
x5          6689.5501   3595.859      1.860      0.063    -358.599    1.37e+04
x6           5.83e+05   1.74e+04     33.580      0.000    5.49e+05    6.17e+05
x7          5.287e+04   2140.055     24.705      0.000    4.87e+04    5.71e+04
x8          2.639e+04   2351.461     11.221      0.000    2.18e+04     3.1e+04
x9          9.589e+04   2152.789     44.542      0.000    9.17e+04       1e+05
x10           70.7864      2.253     31.413      0.000      66.370      75.203
x11           39.6588      2.647     14.985      0.000      34.471      44.846
x12        -2620.2232     72.659    -36.062      0.000   -2762.640   -2477.806
x13           19.8126      3.656      5.420      0.000      12.647      26.978
x14         -582.4199     32.986    -17.657      0.000    -647.074    -517.765
x15         6.027e+05   1.07e+04     56.149      0.000    5.82e+05    6.24e+05
x16        -2.147e+05   1.31e+04    -16.349      0.000    -2.4e+05   -1.89e+05
x17           21.6814      3.448      6.289      0.000      14.924      28.439
x18           -0.3826      0.073     -5.222      0.000      -0.526      -0.239
==============================================================================
Omnibus:                    18384.201   Durbin-Watson:                   1.990
Prob(Omnibus):                  0.000   Jarque-Bera (JB):          1868224.491
Skew:                           3.566   Prob(JB):                         0.00
Kurtosis:                      47.985   Cond. No.                     3.34e+17
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly
specified.
[2] The smallest eigenvalue is 1.97e-21. This might indicate that there are
strong multicollinearity problems or that the design matrix is singular.
"""
```

```
# Removing x5 whose p-value is greater than 0.05
x_opt=x[:,[0,1,2,3,4,6,7,8,9,10,11,12,13,14,15,16,17,18]]

# Ordinal Least Square
OLS=sm.OLS(endog=y,exog=x_opt).fit()
OLS.summary()
```

House Price Prediction (MLR)

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                  price   R-squared:                       0.700
Model:                            OLS   Adj. R-squared:                  0.699
Method:                 Least Squares   F-statistic:                     3145.
Date:                Mon, 04 Sep 2023   Prob (F-statistic):               0.00
Time:                        22:21:14   Log-Likelihood:            -2.9460e+05
No. Observations:               21613   AIC:                         5.892e+05
Df Residuals:                   21596   BIC:                         5.894e+05
Df Model:                          16
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const       5.741e+06   2.89e+06      1.989      0.047    8.28e+04    1.14e+07
x1         -3.586e+04   1891.248    -18.962      0.000   -3.96e+04   -3.22e+04
x2          4.272e+04   3141.958     13.596      0.000    3.66e+04    4.89e+04
x3           109.9751      2.256     48.752      0.000     105.554     114.397
x4             0.1266      0.048      2.643      0.008       0.033       0.221
x5          5.831e+05   1.74e+04     33.585      0.000    5.49e+05    6.17e+05
x6          5.297e+04   2139.565     24.756      0.000    4.88e+04    5.72e+04
x7          2.614e+04   2347.831     11.133      0.000    2.15e+04    3.07e+04
x8          9.624e+04   2144.551     44.878      0.000     9.2e+04       1e+05
x9            72.3464      2.092     34.590      0.000      68.247      76.446
x10           37.6292      2.412     15.604      0.000      32.902      42.356
x11        -2590.7927     70.920    -36.531      0.000   -2729.801   -2451.784
x12           20.1729      3.651      5.526      0.000      13.017      27.328
x13         -576.6895     32.844    -17.559      0.000    -641.065    -512.314
x14         6.044e+05   1.07e+04     56.494      0.000    5.83e+05    6.25e+05
x15        -2.168e+05   1.31e+04    -16.568      0.000   -2.42e+05   -1.91e+05
x16           20.9673      3.426      6.119      0.000      14.251      27.683
x17           -0.3874      0.073     -5.291      0.000      -0.531      -0.244
```

Find intercept value and replace constant as intercept.

```
# Import neccessary packages
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
%matplotlib inline

# Import dataset
house=pd.read_csv(r"D:\Full Stack Data Science\4 Sep (Multiple
Regression)\MLR\House_data.csv")
house

# Splitting data dependent and independent
x=house.iloc[:,3:]
x

y=house.iloc[:,2]
y

# Splitting data into train data and test data
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)
```

House Price Prediction (MLR)

```
# Multiple linear regression
from sklearn.linear_model import LinearRegression
model=LinearRegression()
model.fit(x_train,y_train)

y_pred=model.predict(x_test)
c=model.intercept_
c

# Building model using Backward Elimination
import statsmodels.formula.api as sm

x=np.append(arr= np.full((21613, 1),4166134.8),values=x,axis=1)

import statsmodels.api as sm
x_opt=x[:,[0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18]]
# Ordinal Least Square
OLS=sm.OLS(endog=y,exog=x_opt).fit()
OLS.summary()
```

OLS Regression Results

Dep. Variable:	price	R-squared:	0.700
Model:	OLS	Adj. R-squared:	0.700
Method:	Least Squares	F-statistic:	2960.
Date:	Tue, 05 Sep 2023	Prob (F-statistic):	0.00
Time:	10:48:33	Log-Likelihood:	-2.9460e+05
No. Observations:	21613	AIC:	5.892e+05
Df Residuals:	21595	BIC:	5.894e+05
Df Model:	17		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	1.6059	0.704	2.282	0.022	0.227	2.985
x1	-3.577e+04	1891.843	-18.906	0.000	-3.95e+04	-3.21e+04
x2	4.114e+04	3253.678	12.645	0.000	3.48e+04	4.75e+04
x3	110.4429	2.270	48.661	0.000	105.994	114.891
x4	0.1286	0.048	2.683	0.007	0.035	0.223
x5	6689.5501	3595.859	1.860	0.063	-358.599	1.37e+04
x6	5.83e+05	1.74e+04	33.580	0.000	5.49e+05	6.17e+05
x7	5.287e+04	2140.055	24.705	0.000	4.87e+04	5.71e+04
x8	2.639e+04	2351.461	11.221	0.000	2.18e+04	3.1e+04
x9	9.589e+04	2152.789	44.542	0.000	9.17e+04	1e+05
x10	70.7852	2.253	31.412	0.000	66.368	75.202
x11	39.6576	2.646	14.985	0.000	34.470	44.845
x12	-2620.2232	72.659	-36.062	0.000	-2762.640	-2477.806
x13	19.8126	3.656	5.420	0.000	12.647	26.978
x14	-582.4199	32.986	-17.657	0.000	-647.074	-517.765
x15	6.027e+05	1.07e+04	56.149	0.000	5.82e+05	6.24e+05
x16	-2.147e+05	1.31e+04	-16.349	0.000	-2.4e+05	-1.89e+05
x17	21.6814	3.448	6.289	0.000	14.924	28.439
x18	-0.3826	0.073	-5.222	0.000	-0.526	-0.239

House Price Prediction (MLR)

```
# Removing x5 whose p-value is greater than 0.05
x_opt=x[:,[0,1,2,3,4,6,7,8,9,10,11,12,13,14,15,16,17,18]]
# Ordinal Least Square
OLS=sm.OLS(endog=y,exog=x_opt).fit()
OLS.summary()
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                  price   R-squared:                       0.700
Model:                            OLS   Adj. R-squared:                  0.699
Method:                 Least Squares   F-statistic:                     3145.
Date:                Tue, 05 Sep 2023   Prob (F-statistic):               0.00
Time:                        10:48:48   Log-Likelihood:            -2.9460e+05
No. Observations:               21613   AIC:                         5.892e+05
Df Residuals:                   21596   BIC:                         5.894e+05
Df Model:                          16
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const          1.3781      0.693      1.989      0.047       0.020       2.736
x1         -3.586e+04   1891.248    -18.962      0.000   -3.96e+04   -3.22e+04
x2          4.272e+04   3141.958     13.596      0.000    3.66e+04    4.89e+04
x3           109.9753      2.256     48.753      0.000     105.554     114.397
x4             0.1266      0.048      2.643      0.008       0.033       0.221
x5          5.831e+05   1.74e+04     33.585      0.000    5.49e+05    6.17e+05
x6          5.297e+04   2139.565     24.756      0.000    4.88e+04    5.72e+04
x7          2.614e+04   2347.831     11.133      0.000    2.15e+04    3.07e+04
x8          9.624e+04   2144.551     44.878      0.000     9.2e+04       1e+05
x9            72.3462      2.092     34.590      0.000      68.247      76.446
x10           37.6290      2.412     15.604      0.000      32.902      42.356
x11        -2590.7927     70.920    -36.531      0.000   -2729.801   -2451.784
x12           20.1729      3.651      5.526      0.000      13.017      27.328
x13         -576.6895     32.844    -17.559      0.000    -641.065    -512.314
x14         6.044e+05   1.07e+04     56.494      0.000    5.83e+05    6.25e+05
x15        -2.168e+05   1.31e+04    -16.568      0.000   -2.42e+05   -1.91e+05
x16           20.9673      3.426      6.119      0.000      14.251      27.683
x17           -0.3874      0.073     -5.291      0.000      -0.531      -0.244
```

Interpretation:

When we use constant as **1 or Intercept** value the model will give same result.

When we set the constant value manually (like 0 or any other constant) Then model will be change. It does not gives accurate result.