```
import pandas as pd
data = pd.read_csv("https://trello-attachments.s3.amazonaws.com/5cf2142046ceb163a0e4b189/5
# select Y dependent variable and X Independent variable
Y = data["price"]
independent_variables = data.columns
independent_variables = independent_variables.delete(0)
X = data[independent_variables]


# fit the ordinary least square regression model
import statsmodels.api as sm
model = sm.OLS(Y,X)

#Train the model
model = model.fit()

# check the model summary
model.summary()
```

OLS Regression Results

Dep. Variable:	price	**R-squared (uncentered):**	0.956
Model:	OLS	**Adj. R-squared (uncentered):**	0.956
Method:	Least Squares	**F-statistic:**	1067.
Date:	Thu, 03 Sep 2020	**Prob (F-statistic):**	0.00
Time:	13:56:12	**Log-Likelihood:**	-6034.8
No. Observations:	546	**AIC:**	1.209e+04
Df Residuals:	535	**BIC:**	1.214e+04
Df Model:	11		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
lotsize	3.4431	0.339	10.144	0.000	2.776	4.110
bedrooms	1095.9263	842.938	1.300	0.194	-559.947	2751.800
bathrms	1.402e+04	1466.301	9.561	0.000	1.11e+04	1.69e+04
stories	6526.5732	925.283	7.054	0.000	4708.940	8344.206
driveway	5665.6447	1854.971	3.054	0.002	2021.724	9309.565
recroom	4659.4642	1896.548	2.457	0.014	933.870	8385.059
fullbase	5306.1054	1583.810	3.350	0.001	2194.856	8417.355
gashw	1.285e+04	3218.757	3.993	0.000	6529.985	1.92e+04
airco	1.28e+04	1549.330	8.260	0.000	9754.655	1.58e+04
garagepl	4379.7318	833.106	5.257	0.000	2743.173	6016.291
prefarea	9561.2358	1661.849	5.753	0.000	6296.687	1.28e+04

Omnibus:	101.942	**Durbin-Watson:**	1.576
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	279.382
Skew:	0.915	**Prob(JB):**	2.15e-61
Kurtosis:	5.988	**Cond. No.**	2.74e+04

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 2.74e+04. This might indicate that there are

```
# calculate variance inflation factor
```

```
# calculate variance inflation factor
from statsmodels.stats.outliers_influence import variance_inflation_factor as vif
for i in range(len(independent_variables)):
  vif_list = [vif(data[independent_variables].values,index) for index in range(len(indepen
  mvif = max(vif_list)
  print("Max VIF value is:", mvif)
  drop_index = vif_list.index(mvif)
  print("For the independent variable", independent_variables[drop_index])
  if mvif>10:
    print("Deleting", independent_variables[drop_index])
    independent_variables = independent_variables.delete(drop_index)
print("Final independent variables",independent_variables)
```

```
Max VIF value is: 15.213540834822062
For the independent variable bedrooms
Deleting bedrooms
Max VIF value is: 7.738793387948324
For the independent variable bathrms
Max VIF value is: 7.738793387948324
For the independent variable bathrms
Max VIF value is: 7.738793387948324
For the independent variable bathrms
Max VIF value is: 7.738793387948324
For the independent variable bathrms
Max VIF value is: 7.738793387948324
For the independent variable bathrms
Max VIF value is: 7.738793387948324
For the independent variable bathrms
Max VIF value is: 7.738793387948324
For the independent variable bathrms
Max VIF value is: 7.738793387948324
For the independent variable bathrms
Max VIF value is: 7.738793387948324
For the independent variable bathrms
Max VIF value is: 7.738793387948324
For the independent variable bathrms
Max VIF value is: 7.738793387948324
For the independent variable bathrms
Final independent variables Index(['lotsize', 'bathrms', 'stories', 'driveway', 'recr
       'gashw', 'airco', 'garagepl', 'prefarea'],
      dtype='object')
```

```
Y = data["price"]
X = data[independent_variables]
model = sm.OLS(Y,X)
model = model.fit()

model.summary()
```

OLS Regression Results

Dep. Variable:	price	**R-squared (uncentered):**	0.956
Model:	OLS	**Adj. R-squared (uncentered):**	0.955
Method:	Least Squares	**F-statistic:**	1172.
Date:	Thu, 03 Sep 2020	**Prob (F-statistic):**	0.00
Time:	13:56:12	**Log-Likelihood:**	-6035.7
No. Observations:	546	**AIC:**	1.209e+04
Df Residuals:	536	**BIC:**	1.213e+04
Df Model:	10		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
lotsize	3.5725	0.325	11.001	0.000	2.935	4.210
bathrms	1.482e+04	1331.794	11.127	0.000	1.22e+04	1.74e+04
stories	7079.3352	822.349	8.609	0.000	5463.914	8694.756
driveway	6097.9741	1826.098	3.339	0.001	2510.788	9685.160
recroom	4474.0007	1892.393	2.364	0.018	756.584	8191.417
fullbase	5788.5832	1540.712	3.757	0.000	2762.009	8815.157
gashw	1.294e+04	3220.058	4.020	0.000	6619.038	1.93e+04
airco	1.264e+04	1545.505	8.178	0.000	9603.420	1.57e+04
garagepl	4372.5314	833.623	5.245	0.000	2734.962	6010.101
prefarea	9463.4765	1661.216	5.697	0.000	6200.185	1.27e+04

```
user_input = {}
for var in independent_variables:
  temp = input(" Enter " +var+ " : ")
  user_input[var] = temp
user_df = pd.DataFrame(data=user_input, index=[0], columns=independent_variables)
import sklearn.linear_model as lm
lr = lm.LinearRegression()
lr.fit(X,Y)
price = lr.predict(user_df)
print("Price of House is USD", int(price[0]))
```

```
 Enter lotsize : 2000
 Enter bathrms : 1
 Enter stories : 1
 Enter driveway : 1
 Enter recroom : 1
 Enter fullbase : 1
 Enter gashw : 1
 Enter airco : 0
 Enter garagepl : 1
 Enter prefarea : 0
Price of House is USD 62598
```

```
user_df
```

	lotsize	bathrms	stories	driveway	recroom	fullbase	gashw	airco	garagepl	pr
0	2000	1	1	1	1	1	1	0	1	