import numpy as np  
import pandas as pd  
import statsmodels.formula.api as sms

house = # load the file kc\_house\_data.csv

model = # fit a linear model using the sms.ols function and the house dataframe

area = float(input())

prediction = # use the model.predict function to find the predicted value for price using   
            # the area value for the predictor

print(prediction)

PLEASE PROVIDE THE CODE IN PYTHON AND PLEASE MAKE SURE THE OUTPUT MATCHES THE SNAPSHOT'S OUTPUT.

Answer and explanations

#importing required libraries

import numpy as np

import pandas as pd

import statsmodels.formula.api as sms

# load the file kc\_house\_data.csv

house = pd.read\_csv("kc\_house\_data.csv")

# fit a linear model using the sms.ols function and the house dataframe

model = sms.ols(formula='price~sqft\_living',data=house).fit()

area = float(input())#taking input

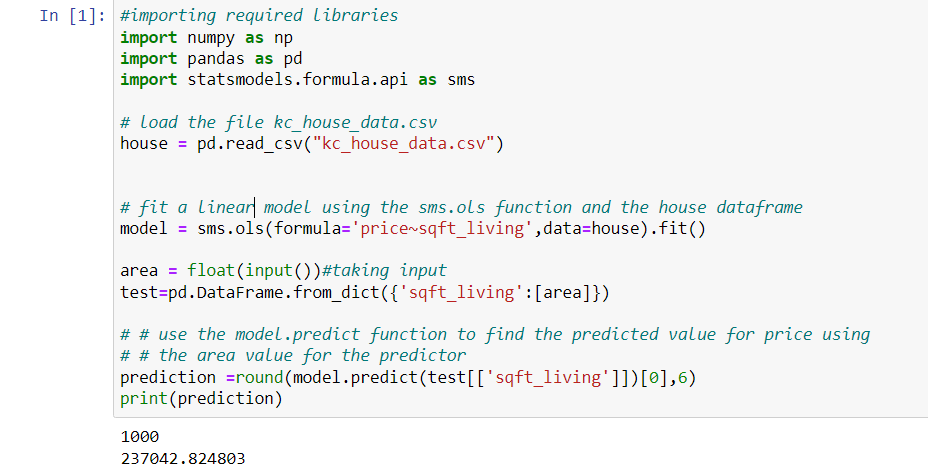
test=pd.DataFrame.from\_dict({'sqft\_living':[area]})

# # use the model.predict function to find the predicted value for price using

# # the area value for the predictor

prediction =round(model.predict(test[['sqft\_living']])[0],6)

print(prediction)



**Explanation:**

Step 1:

Import the necessary libraries

import pandas as pd

import statsmodels.api as sm

Step 2:

df = pd.read\_csv("kc\_house\_data.csv")

df

Step 3:

#set the target as Y variable and feature as X variable

X = df['sqft\_living']

y = df['price']

#create constant before fitting the model

X = sm.add\_constant(X)

#fit the model to the target and feature

model = sm.OLS(y, X).fit()

#Make a prediction, please be noted that value of 1 there is necessary which is based on the math of OLS

# and this is a constant value and does not overall affect the [redicted value

model.predict([1,1000])

Step 4:

#set the target as Y variable and feature as X variable

X = df['zipcode']

y = df['price']

#create constant before fitting the model

X = sm.add\_constant(X)

#fit the model to the target and feature

model = sm.OLS(y, X).fit()

#Make a prediction, please be noted that value of 1 there is necessary which is based on the math of OLS

# and this is a constant value and does not overall affect the [redicted value

model.predict([1,98144])

Step 5:

#set the target as Y variable and feature as X variable

X = df['bathrooms']

y = df['price']

#create constant before fitting the model

X = sm.add\_constant(X)

#fit the model to the target and feature

model = sm.OLS(y, X).fit()

#Make a prediction, please be noted that value of 1 there is necessary which is based on the math of OLS

# and this is a constant value and does not overall affect the [redicted value

model.predict([1,5])