**Linear Regression**

**Import packages**

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import seaborn as sea

**Import data**

data=pd.read\_excel("Linear Regression.xlsx")

**Cheack distribution of data:**

data.isnull().sum()

Out[3]:

|  |  |
| --- | --- |
| 1. price | 0 |
| 1. sqft\_living | 0 |
| 1. bedrooms | 0 |
| 1. bathrooms | 0 |
| 1. floors | 0 |

dtype: int64

**Analysis distribution of data:**

data.skew()

Out[5]:

|  |  |
| --- | --- |
|  |  |
| 1. price | 4.024069 |
| 1. sqft\_living | 1.471555 |
| 1. bedrooms | 1.974300 |
| 1. bathrooms | 0.511108 |
| 1. floors | 0.616177 |

dtype: float64

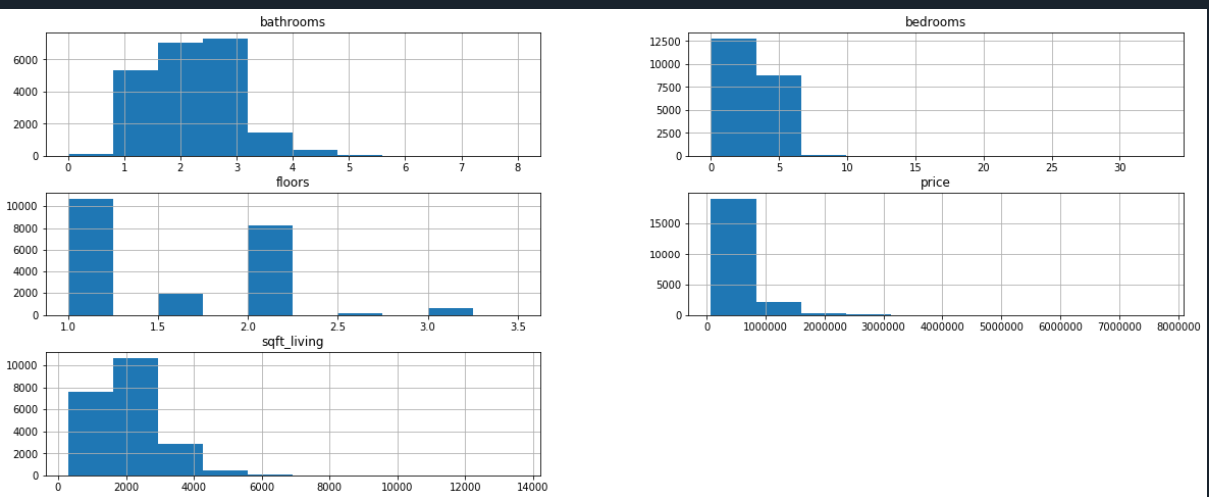
data.columns

Out[4]: Index(['price', 'sqft\_living', 'bedrooms', 'bathrooms', 'floors'], dtype='object')

#EDA:

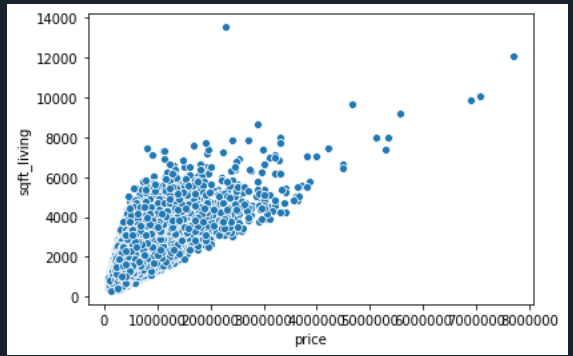
**Histogram**

data.hist(figsize=(20,8))

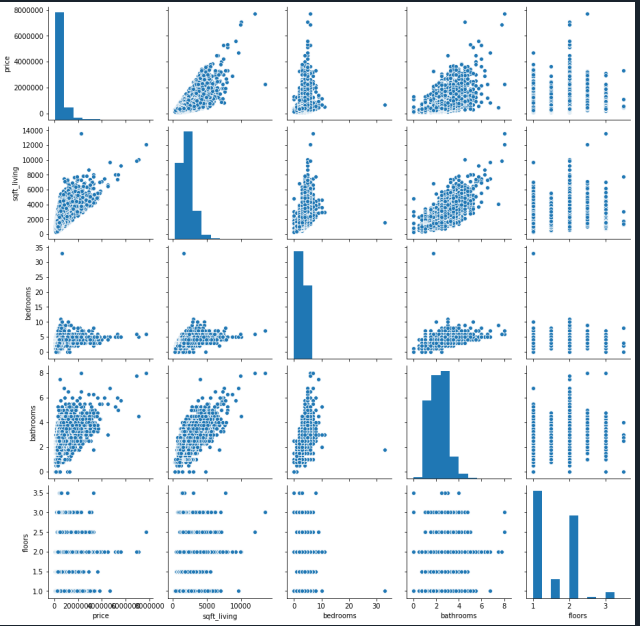


**Scatterplot**

sea.scatterplot(data["price"],data["sqft\_living"])



sea.pairplot(data)



**Split the data:**

x=data.iloc[:,1:5]

y=data.iloc[:,0]

from sklearn.model\_selection import train\_test\_split

xtrain,xtest,ytrain,ytest=train\_test\_split(x, y,test\_size=0.2,random\_state=2)

**Cheack multicolinarity with help of r^2: if executed value is greater then 10 there multicolinarity is existing.**

import statsmodels.api as ap

mod1=ap.OLS(ytrain,xtrain["sqft\_living"]).fit()

print(1/(1-mod1.rsquared))

mod2=ap.OLS(ytrain,xtrain["bedrooms"]).fit()

print(1/(1-mod2.rsquared))

mod3=ap.OLS(ytrain,xtrain["bathrooms"]).fit()

print(1/(1-mod3.rsquared))

mod4=ap.OLS(ytrain,xtrain["floors"]).fit()

print(1/(1-mod4.rsquared))

6.206555943978328

3.4580228989509263

4.378520992307858

3.1612551218409686

**So there is no multi colinarity.**

**Simple linear regression model:**

smod1=ap.OLS(ytrain,xtrain["sqft\_living"]).fit()

**smod1.summary()**

**Out[52]:**

**<class 'statsmodels.iolib.summary.Summary'>**

**"""**

**OLS Regression Results**

**=======================================================================================**

**Dep. Variable: price R-squared (uncentered): 0.839**

**Model: OLS Adj. R-squared (uncentered): 0.839**

**Method: Least Squares F-statistic: 9.002e+04**

**Date: Sun, 09 Aug 2020 Prob (F-statistic): 0.00**

**Time: 07:33:57 Log-Likelihood: -2.4021e+05**

**No. Observations: 17290 AIC: 4.804e+05**

**Df Residuals: 17289 BIC: 4.804e+05**

**Df Model: 1**

**Covariance Type: nonrobust**

**===============================================================================**

**coef std err t P>|t| [0.025 0.975]**

**-------------------------------------------------------------------------------**

**sqft\_living 263.4685 0.878 300.027 0.000 261.747 265.190**

**==============================================================================**

**Omnibus: 13117.379 Durbin-Watson: 1.989**

**Prob(Omnibus): 0.000 Jarque-Bera (JB): 632432.691**

**Skew: 3.196 Prob(JB): 0.00**

**Kurtosis: 31.931 Cond. No. 1.00**

**smod2=ap.OLS(ytrain,xtrain["bedrooms"]).fit()**

**smod2.summary()**

**Out[53]:**

**<class 'statsmodels.iolib.summary.Summary'>**

**"""**

**OLS Regression Results**

**=======================================================================================**

**Dep. Variable: price R-squared (uncentered): 0.711**

**Model: OLS Adj. R-squared (uncentered): 0.711**

**Method: Least Squares F-statistic: 4.250e+04**

**Date: Sun, 09 Aug 2020 Prob (F-statistic): 0.00**

**Time: 07:36:27 Log-Likelihood: -2.4527e+05**

**No. Observations: 17290 AIC: 4.905e+05**

**Df Residuals: 17289 BIC: 4.905e+05**

**Df Model: 1**

**Covariance Type: nonrobust**

**==============================================================================**

**coef std err t P>|t| [0.025 0.975]**

**------------------------------------------------------------------------------**

**bedrooms 1.573e+05 762.969 206.147 0.000 1.56e+05 1.59e+05**

**==============================================================================**

**Omnibus: 14580.571 Durbin-Watson: 1.988**

**Prob(Omnibus): 0.000 Jarque-Bera (JB): 1018319.604**

**Skew: 3.648 Prob(JB): 0.00**

**Kurtosis: 39.882 Cond. No. 1.00**

**smod3=ap.OLS(ytrain,xtrain["bathrooms"]).fit()**

**smod3.summary()**

**Out[55]:**

**<class 'statsmodels.iolib.summary.Summary'>**

**"""**

**OLS Regression Results**

**=======================================================================================**

**Dep. Variable: price R-squared (uncentered): 0.772**

**Model: OLS Adj. R-squared (uncentered): 0.772**

**Method: Least Squares F-statistic: 5.841e+04**

**Date: Sun, 09 Aug 2020 Prob (F-statistic): 0.00**

**Time: 07:37:26 Log-Likelihood: -2.4323e+05**

**No. Observations: 17290 AIC: 4.865e+05**

**Df Residuals: 17289 BIC: 4.865e+05**

**Df Model: 1**

**Covariance Type: nonrobust**

**==============================================================================**

**coef std err t P>|t| [0.025 0.975]**

**------------------------------------------------------------------------------**

**bathrooms 2.546e+05 1053.483 241.684 0.000 2.53e+05 2.57e+05**

**==============================================================================**

**Omnibus: 14086.495 Durbin-Watson: 1.976**

**Prob(Omnibus): 0.000 Jarque-Bera (JB): 817605.317**

**Skew: 3.514 Prob(JB): 0.00**

**Kurtosis: 35.947 Cond. No. 1.00**

**smod4=ap.OLS(ytrain,xtrain["floors"]).fit()**

**smod4.summary()**

**Out[56]:**

**<class 'statsmodels.iolib.summary.Summary'>**

**"""**

**OLS Regression Results**

**=======================================================================================**

**Dep. Variable: price R-squared (uncentered): 0.684**

**Model: OLS Adj. R-squared (uncentered): 0.684**

**Method: Least Squares F-statistic: 3.737e+04**

**Date: Sun, 09 Aug 2020 Prob (F-statistic): 0.00**

**Time: 07:38:02 Log-Likelihood: -2.4605e+05**

**No. Observations: 17290 AIC: 4.921e+05**

**Df Residuals: 17289 BIC: 4.921e+05**

**Df Model: 1**

**Covariance Type: nonrobust**

**==============================================================================**

**coef std err t P>|t| [0.025 0.975]**

**------------------------------------------------------------------------------**

**floors 3.391e+05 1754.438 193.303 0.000 3.36e+05 3.43e+05**

**==============================================================================**

**Omnibus: 13875.605 Durbin-Watson: 1.963**

**Prob(Omnibus): 0.000 Jarque-Bera (JB): 782474.261**

**Skew: 3.439 Prob(JB): 0.00**

**Kurtosis: 35.231 Cond. No. 1.00**

**==============================================================================**

**Warnings:**

**[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.**

**"""**

**Final model:**

finmod=ap.OLS(ytrain,xtrain).fit()

finmod.summary()

Out[18]:

<class 'statsmodels.iolib.summary.Summary'>

"""

OLS Regression Results

=======================================================================================

Dep. Variable: price R-squared (uncentered): 0.843

Model: OLS Adj. R-squared (uncentered): 0.843

Method: Least Squares F-statistic: 2.326e+04

Date: Sat, 08 Aug 2020 Prob (F-statistic): 0.00

Time: 00:22:15 Log-Likelihood: -2.3997e+05

No. Observations: 17290 AIC: 4.800e+05

Df Residuals: 17286 BIC: 4.800e+05

Df Model: 4

Covariance Type: nonrobust

===============================================================================

coef std err t P>|t| [0.025 0.975]

-------------------------------------------------------------------------------

sqft\_living 305.1401 3.444 88.602 0.000 298.390 311.891

bedrooms -4.276e+04 2001.811 -21.359 0.000 -4.67e+04 -3.88e+04

bathrooms 1.14e+04 4257.109 2.679 0.007 3059.885 1.97e+04

floors 1.51e+04 3809.676 3.963 0.000 7631.739 2.26e+04

==============================================================================

Omnibus: 11479.931 Durbin-Watson: 1.987

Prob(Omnibus): 0.000 Jarque-Bera (JB): 415656.238

Skew: 2.688 Prob(JB): 0.00

Kurtosis: 26.411 Cond. No. 5.69e+03

ypr.head(7)

Out[21]:

6638 6.126657e+05

7366 5.660869e+05

3158 3.834414e+05

9117 1.226605e+06

3392 2.034087e+05

305 4.926529e+05

14462 6.743326e+05

dtype: float64

ytest.head()

Out[22]:

6638 735000

7366 1150000

3158 350500

9117 860000

3392 122000

Name: price, dtype: int64

conclusion: this model is accurate 84 percentage.**R-squared (uncentered**): **0.843**