**import pandas as pd**

**import numpy as np**

**import matplotlib.pyplot as plt**

**import seaborn as seabornInstance**

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

**from sklearn.linear\_model import LinearRegression**

**from sklearn import metrics**

**from sklearn import preprocessing**

**dataset=pd.read\_excel("Linear Regression.xlsx",sheet\_name=0)**

**dataset.head()**

Out[226]:

price sqft\_living bedrooms bathrooms floors

0 221900 1180 3 1.00 1.0

1 538000 2570 3 2.25 2.0

2 180000 770 2 1.00 1.0

3 604000 1960 4 3.00 1.0

4 510000 1680 3 2.00 1.0

**min\_max\_scaler = preprocessing.MinMaxScaler()**

**x = dataset.values**

**min\_max\_scaler = preprocessing.MinMaxScaler()**

**x\_scaled = min\_max\_scaler.fit\_transform(x)**

**df = pd.DataFrame(x\_scaled)**

**normalized\_df=(df-df.mean())/df.std()**

**dataset.shape**

Out[242]: (21613, 5)

**dataset.describe()**

Out[243]:

price sqft\_living bedrooms bathrooms floors

count 2.161300e+04 21613.000000 21613.000000 21613.000000 21613.000000

mean 5.400881e+05 2079.899736 3.370842 2.114757 1.494309

std 3.671272e+05 918.440897 0.930062 0.770163 0.539989

min 7.500000e+04 290.000000 0.000000 0.000000 1.000000

25% 3.219500e+05 1427.000000 3.000000 1.750000 1.000000

50% 4.500000e+05 1910.000000 3.000000 2.250000 1.500000

75% 6.450000e+05 2550.000000 4.000000 2.500000 2.000000

max 7.700000e+06 13540.000000 33.000000 8.000000 3.500000

**dataset.isnull().any()**

Out[244]:

price False

sqft\_living False

bedrooms False

bathrooms False

floors False

dtype: bool

**X = dataset[['sqft\_living','bedrooms','bathrooms','floors']]**

**y= dataset.price**

**plt.figure(figsize=(15,10))**

Out[248]: <Figure size 1080x720 with 0 Axes><Figure size 1080x720 with 0 Axes>

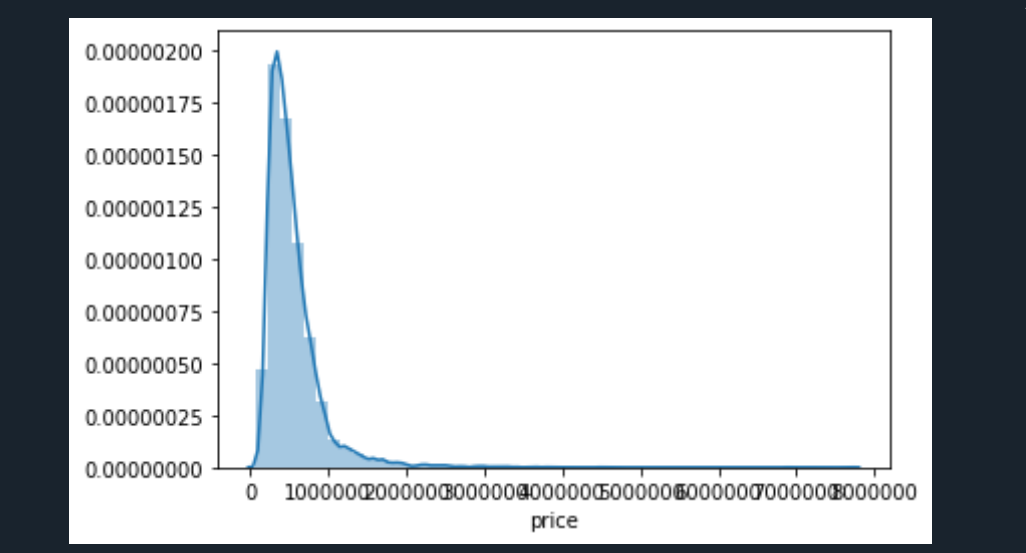
**plt.tight\_layout()**

<Figure size 432x288 with 0 Axes>

**seabornInstance.distplot(dataset['price'])**

Out[250]: <matplotlib.axes.\_subplots.AxesSubplot at 0x1cbfb077188>

Figures now render in the Plots pane by default. To make them also appear inline in the Console, uncheck "Mute Inline Plotting" under the Plots pane options menu.



**X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.25, random\_state=0)**

**regressor = LinearRegression()**

**regressor.fit(X\_train, y\_train)**

Out[253]: LinearRegression(copy\_X=True, fit\_intercept=True, n\_jobs=None, normalize=False)

**coeff\_df = pd.DataFrame(regressor.coef\_, X.columns, columns=['coefficient'])**

**print(coeff\_df)**

coefficient

sqft\_living 310.855577

bedrooms -54235.212195

bathrooms 1633.656805

floors 2182.860836

**y\_pred = regressor.predict(X\_test)**

**df = pd.DataFrame({'Actual': y\_test, 'Predicted': y\_pred})**

**df1 = df.head(25)**

**print(df1)**

Actual Predicted

17384 297000 4.158436e+05

722 1578000 1.315221e+06

2680 562100 4.133612e+05

18754 631500 3.174044e+05

14554 780000 8.508212e+05

16227 485000 4.294474e+05

6631 340000 4.494329e+05

19813 335606 7.054872e+05

3367 425000 6.270052e+05

21372 490000 1.198298e+06

3268 732000 5.925116e+05

20961 389700 4.512073e+05

21456 450000 2.883928e+05

3880 357000 3.292657e+05

17472 960000 8.508212e+05

7618 257000 3.466066e+05

1091 448000 3.917363e+05

1560 610000 5.225146e+05

8945 230950 2.552333e+05

8439 377500 3.394082e+05

13058 375000 4.334817e+05

12080 410000 5.506254e+05

7417 459000 4.537429e+05

3101 190000 2.550687e+05

18769 585000 5.968454e+05

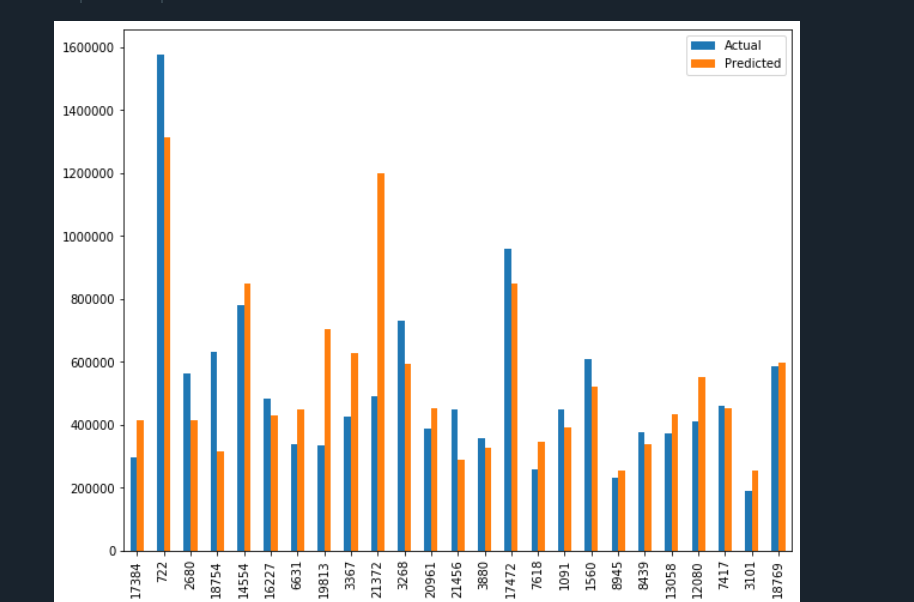
**df1.plot(kind='bar',figsize=(10,8))**

Out[260]: <matplotlib.axes.\_subplots.AxesSubplot at 0x1cbfb439d48>

**plt.grid(which='major', linestyle='-', linewidth='0.5', color='green')**

**plt.grid(which='minor', linestyle=':', linewidth='0.5', color='black')**

**plt.show()**



**print('Mean Absolute Error:', metrics.mean\_absolute\_error(y\_test, y\_pred))**

Mean Absolute Error: 169669.0001695836

**print('Mean Squared Error:', metrics.mean\_squared\_error(y\_test, y\_pred))**

Mean Squared Error: 66379302151.73661

**print('Root Mean Squared Error:', np.sqrt(metrics.mean\_squared\_error(y\_test, y\_pred)))**

Root Mean Squared Error: 257641.80978974784

**dataset.corr()**

Out[267]:

price sqft\_living bedrooms bathrooms floors

price 1.000000 0.702035 0.308350 0.525138 0.256794

sqft\_living 0.702035 1.000000 0.576671 0.754665 0.353949

bedrooms 0.308350 0.576671 1.000000 0.515884 0.175429

bathrooms 0.525138 0.754665 0.515884 1.000000 0.500653

floors 0.256794 0.353949 0.175429 0.500653 1.000000

**Inference:**

**By this analysis we can say**

**1.All the independent variables like sqft\_living , bedrooms , bathrooms , floors are positively correlated .**

**2. price and bedrooms are weakly correlated as the value is .308 bcz it is lesser than .5**

**3. price and sqft\_living are strongly correlated as the value is .702 bcz it is is above .5**

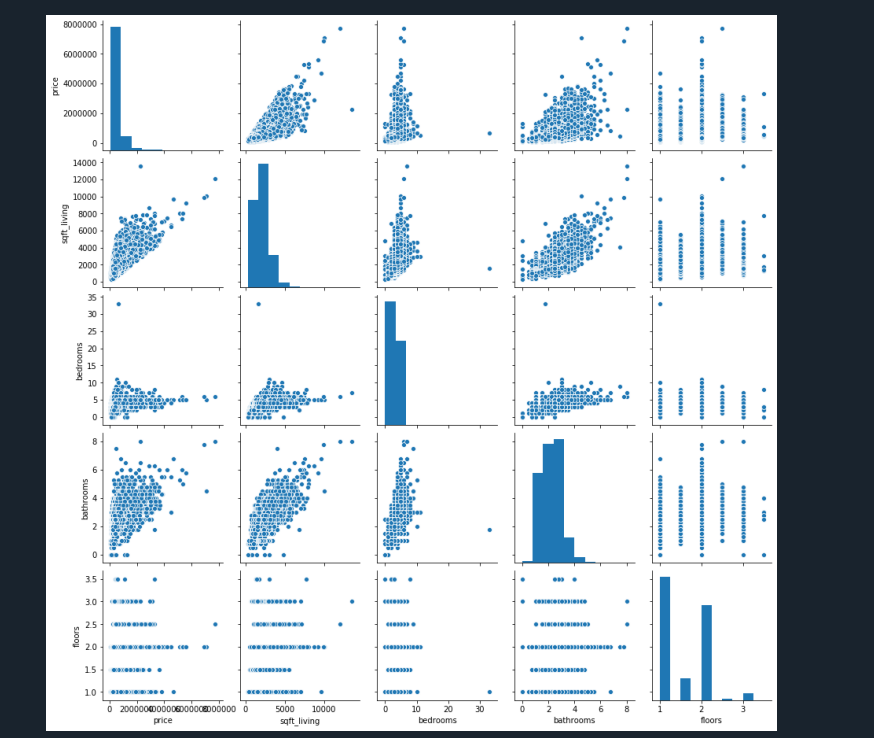
**4.price and bathrooms are moderately correlated as the value is .525 which is almost near to .5**

**5. price and floor are weakly correlated a sthe value is .256 which is lesser than .5**

import seaborn as sns

sns.pairplot(dataset)

Out[269]: <seaborn.axisgrid.PairGrid at 0x1cbfb5f6048>



dataset.hist(figsize=(14,12), bins=50)

Out[270]:

array([[<matplotlib.axes.\_subplots.AxesSubplot object at 0x000001CBFAA39AC8>,

<matplotlib.axes.\_subplots.AxesSubplot object at 0x000001CBFAE6CB08>],

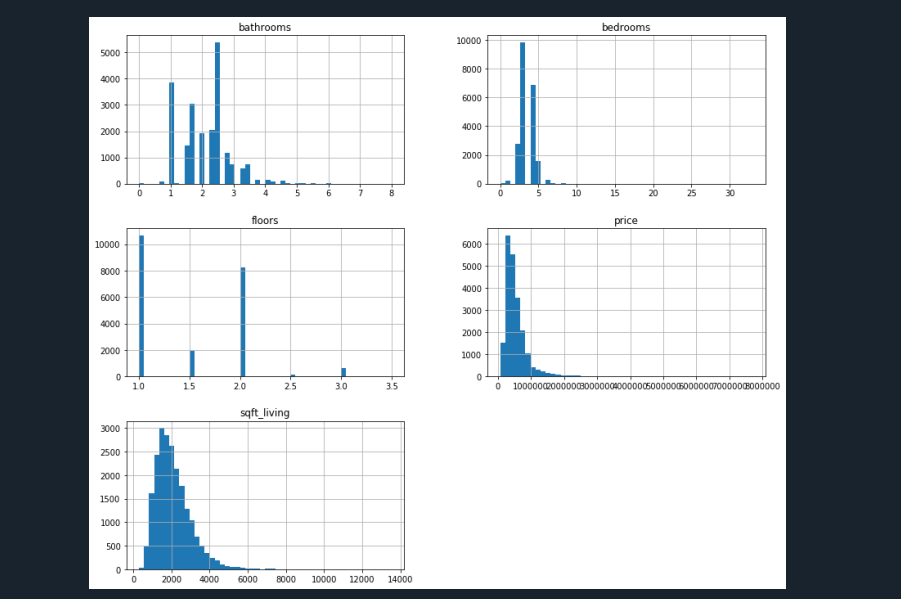
[<matplotlib.axes.\_subplots.AxesSubplot object at 0x000001CBFD0CE508>,

<matplotlib.axes.\_subplots.AxesSubplot object at 0x000001CBFD105888>],

[<matplotlib.axes.\_subplots.AxesSubplot object at 0x000001CBFD140288>,

<matplotlib.axes.\_subplots.AxesSubplot object at 0x000001CBFD175C48>]],

dtype=object)



from sklearn.metrics import mean\_squared\_error,r2\_score

print(f"RMSE Valueis {np.sqrt(mean\_squared\_error(y\_test, y\_pred))}")

RMSE Valueis 257641.80978974784

print(f"R-Square score is {r2\_score(y\_test, y\_pred)}")

R-Square score is 0.5003085644927433

**So here we got R Square vale 0.5003085644927433 so model is 50%**