## LINEAR REGRESSION MODEL:

THE MODEL NO : 1

Code:

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

data = pd.read\_excel(r"C:\Users\User\OneDrive\Desktop\Linear Regression.xlsx")

print(data.head())

print(data.isna().sum())

print(data.describe())

print(data.corr())

X = data.iloc[:,0:1]

Y = data.iloc[:,1:2]

print(X)

print(Y)

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

X\_train, X\_test, Y\_train, Y\_test = train\_test\_split(np.array(X), np.array(Y), test\_size = 0.2)

print(X\_train.shape)

print(X\_test.shape)

print(Y\_train.shape)

print(Y\_test.shape)

from sklearn.linear\_model import LinearRegression

lin\_reg = LinearRegression()

lin\_reg.fit(X\_train, Y\_train)

lin\_reg.coef\_

lin\_reg.intercept\_

plt.scatter(X\_train, Y\_train, color = 'red')

plt.plot(X\_train, lin\_reg.predict(X\_train), color = 'green')

plt.xlabel("price")

plt.ylabel("sprt\_living")

plt.show()

ypred = lin\_reg.predict(X\_test)

print(ypred)

data.head()

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

print("the value of the r\_square value", r2\_score(Y\_test, ypred))

print("the value of the RMSE value", np.sqrt(mean\_squared\_error(Y\_test, ypred)))

unseen\_pred = lin\_reg.predict(np.array([[109]]))

print("the unseen data is : ", unseen\_pred)

The output:

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

price 0

sqft\_living 0

bedrooms 0

bathrooms 0

floors 0

dtype: int64

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

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

price

0 221900

1 538000

2 180000

3 604000

4 510000

... ...

21608 360000

21609 400000

21610 402101

21611 400000

21612 325000

[21613 rows x 1 columns]

sqft\_living

0 1180

1 2570

2 770

3 1960

4 1680

... ...

21608 1530

21609 2310

21610 1020

21611 1600

21612 1020

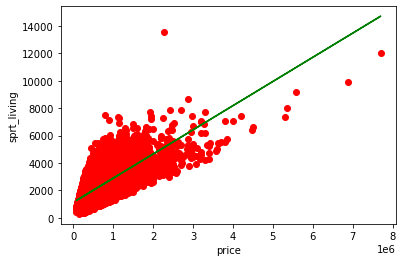
[21613 rows x 1 columns]

(17290, 1)

(4323, 1)

(17290, 1)

(4323, 1)



[[1848.79639838]

[1838.21376849]

[1600.10459583]

...

[2487.28173543]

[2055.15768135]

[1508.38847007]]

the value of the r\_square value 0.5109041317265297

the value of the RMSE value 642.3614948765321

the unseen data is : [[1129.36981658]]

This model is made by the linear regression dataset. This model is made to find there is any correlation for between the “price of the flat and the sprt living”. To check that we used the correlation function and the scatter plot to define that to the customer. The scatter plot shows that there is a significant correlation between the “price and the sprt living” part.

To conclude that the r2\_score and the RMSE value is used. So the output of that shows that there is significant correlation between them. And this model is used to predict the unseen value or the future value.

MODEL NO : 2

The code:

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

data = pd.read\_excel(r"C:\Users\User\OneDrive\Desktop\Linear Regression.xlsx")

print(data.head())

print(data.isna().sum())

print(data.describe())

print(data.corr())

X = data.iloc[:, :1]

Y = data.iloc[:,2:3]

print(X)

print(Y)

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

X\_train, X\_test, Y\_train, Y\_test = train\_test\_split(np.array(X), np.array(Y), test\_size = 0.2)

print(X\_train.shape)

print(X\_test.shape)

print(Y\_train.shape)

print(Y\_test.shape)

from sklearn.linear\_model import LinearRegression

lin\_reg = LinearRegression()

lin\_reg.fit(X\_train, Y\_train)

lin\_reg.coef\_

lin\_reg.intercept\_

plt.scatter(X\_train, Y\_train, color = 'red')

plt.plot(X\_train, lin\_reg.predict(X\_train), color = 'green')

plt.xlabel("price")

plt.ylabel("bedrooms")

plt.show()

ypred = lin\_reg.predict(X\_test)

print(ypred)

data.head()

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

print("the value of the r\_square value", r2\_score(Y\_test, ypred))

print("the value of the RMSE value", np.sqrt(mean\_squared\_error(Y\_test, ypred)))

unseen\_pred = lin\_reg.predict(np.array([[1]]))

print("the unseen data is : ", unseen\_pred)

the output:

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

price 0

sqft\_living 0

bedrooms 0

bathrooms 0

floors 0

dtype: int64

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

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

price

0 221900

1 538000

2 180000

3 604000

4 510000

... ...

21608 360000

21609 400000

21610 402101

21611 400000

21612 325000

[21613 rows x 1 columns]

bedrooms

0 3

1 3

2 2

3 4

4 3

... ...

21608 3

21609 4

21610 2

21611 3

21612 2

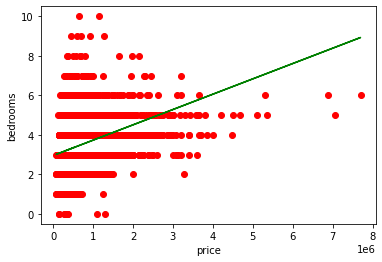
[21613 rows x 1 columns]

(17290, 1)

(4323, 1)

(17290, 1)

(4323, 1)



[[3.21564667]

[3.22712553]

[3.21169379]

...

[3.58535738]

[4.02172452]

[3.36678627]]

the value of the r\_square value 0.08253722556760112

the value of the RMSE value 0.9699884762686669

the unseen data is : [[2.95599737]]

This model is made by the linear regression dataset. This model is made to find there is any correlation for between the “price of the flat and the bedrooms”. To check that we used the correlation function and the scatter plot to define that to the customer. The scatter plot shows that there is a partial correlation between the “price and the bedrooms” part.

To conclude that the r2\_score and the RMSE value is used. So the output of that shows that there is significant correlation between them. And this model is used to predict the unseen value or the future value.

MODEL NO : 3

The code:

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

data = pd.read\_excel(r"C:\Users\User\OneDrive\Desktop\Linear Regression.xlsx")

print(data.head())

print(data.isna().sum())

print(data.describe())

print(data.corr())

X = data.iloc[:, :1]

Y = data.iloc[:,3:4]

print(X)

print(Y)

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

X\_train, X\_test, Y\_train, Y\_test = train\_test\_split(np.array(X), np.array(Y), test\_size = 0.2)

print(X\_train.shape)

print(X\_test.shape)

print(Y\_train.shape)

print(Y\_test.shape)

from sklearn.linear\_model import LinearRegression

lin\_reg = LinearRegression()

lin\_reg.fit(X\_train, Y\_train)

lin\_reg.coef\_

lin\_reg.intercept\_

plt.scatter(X\_train, Y\_train, color = 'red')

plt.plot(X\_train, lin\_reg.predict(X\_train), color = 'green')

plt.xlabel("price")

plt.ylabel("bathrooms")

plt.show()

ypred = lin\_reg.predict(X\_test)

print(ypred)

data.head()

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

print("the value of the r\_square value", r2\_score(Y\_test, ypred))

print("the value of the RMSE value", np.sqrt(mean\_squared\_error(Y\_test, ypred)))

unseen\_pred = lin\_reg.predict(np.array([[3]]))

print("the unseen data is : ", unseen\_pred)

The output:

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

price 0

sqft\_living 0

bedrooms 0

bathrooms 0

floors 0

dtype: int64

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

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

price

0 221900

1 538000

2 180000

3 604000

4 510000

... ...

21608 360000

21609 400000

21610 402101

21611 400000

21612 325000

[21613 rows x 1 columns]

bathrooms

0 1.00

1 2.25

2 1.00

3 3.00

4 2.00

... ...

21608 2.50

21609 2.50

21610 0.75

21611 2.50

21612 0.75

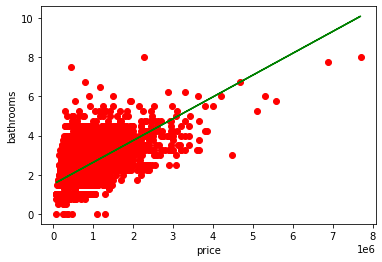
[21613 rows x 1 columns]

(17290, 1)

(4323, 1)

(17290, 1)

(4323, 1)



[[1.74639083]

[2.13403508]

[1.97920016]

...

[2.04603538]

[2.10173139]

[2.24932582]]

the value of the r\_square value 0.2609643914867642

the value of the RMSE value 0.655991275381669

the unseen data is : [[1.5169266]]

This model is made by the linear regression dataset. This model is made to find there is any correlation for between the “price of the flat and the bathrooms”. To check that we used the correlation function and the scatter plot to define that to the customer. The scatter plot shows that there is a partial correlation between the “price and the bathrooms” part.

To conclude that the r2\_score and the RMSE value is used. So the output of that shows that there is partial correlation between them. And this model is used to predict the unseen value or the future value.

MODEL NO: 4

The code:

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

data = pd.read\_excel(r"C:\Users\User\OneDrive\Desktop\Linear Regression.xlsx")

print(data.head())

print(data.isna().sum())

print(data.describe())

print(data.corr())

X = data.iloc[:, :1]

Y = data.iloc[:,4:5]

print(X)

print(Y)

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

X\_train, X\_test, Y\_train, Y\_test = train\_test\_split(np.array(X), np.array(Y), test\_size = 0.2)

print(X\_train.shape)

print(X\_test.shape)

print(Y\_train.shape)

print(Y\_test.shape)

from sklearn.linear\_model import LinearRegression

lin\_reg = LinearRegression()

lin\_reg.fit(X\_train, Y\_train)

lin\_reg.coef\_

lin\_reg.intercept\_

plt.scatter(X\_train, Y\_train, color = 'red')

plt.plot(X\_train, lin\_reg.predict(X\_train), color = 'green')

plt.xlabel("price")

plt.ylabel("floors")

plt.show()

ypred = lin\_reg.predict(X\_test)

print(ypred)

data.head()

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

print("the value of the r\_square value", r2\_score(Y\_test, ypred))

print("the value of the RMSE value", np.sqrt(mean\_squared\_error(Y\_test, ypred)))

unseen\_pred = lin\_reg.predict(np.array([[5]]))

print("the unseen data is : ", unseen\_pred)

The output:

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

price 0

sqft\_living 0

bedrooms 0

bathrooms 0

floors 0

dtype: int64

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

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

price

0 221900

1 538000

2 180000

3 604000

4 510000

... ...

21608 360000

21609 400000

21610 402101

21611 400000

21612 325000

[21613 rows x 1 columns]

floors

0 1.0

1 2.0

2 1.0

3 1.0

4 1.0

... ...

21608 3.0

21609 2.0

21610 2.0

21611 2.0

21612 2.0

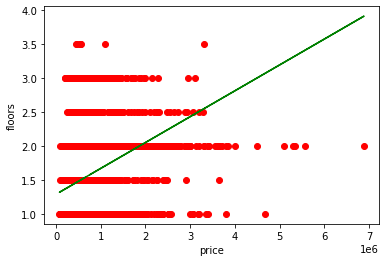
[21613 rows x 1 columns]

(17290, 1)

(4323, 1)

(17290, 1)

(4323, 1)



[[1.40277334]

[1.37896914]

[1.38983255]

...

[1.41727695]

[1.56288472]

[1.56479059]]

the value of the r\_square value 0.06201386306103962

the value of the RMSE value 0.5227417762441187

the unseen data is : [[1.29034851]]

This model is made by the linear regression dataset. This model is made to find there is any correlation for between the “price of the flat and the floors”. To check that we used the correlation function and the scatter plot to define that to the customer. The scatter plot shows that there is a no correlation between the “price and the floors” part.

To conclude that the r2\_score and the RMSE value is used. So the output of that shows that there is no correlation between them. And this model is used to predict the unseen value or the future value.