LINEAR REGRESSION RESULT FOR BEDROOM ATTRIBUTE:

STEPS:

1. Import pandas package.

2. Loading the dataset.

```
In [1]: import os
        os.chdir(r"C:\Users\User\Downloads")
        import pandas as pd
        dataset=pd.read_excel("Linear Regression.xlsx",sheet_name=0)
        dataset.head()
```

Out[1]:

	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

3. Slicing the dataset to obtain independent variable “Price”.

```
In [15]: x=dataset.iloc[:,:1]
         x.head()
```

Out[15]:

	price
0	221900
1	538000
2	180000
3	604000
4	510000

4. Slicing the dataset to obtain dependent variable “Bathroom”.

```
: y.head()
```

:

	bedrooms
0	3
1	3
2	2
3	4
4	3

5. Import train_test_split model and assigning training and testing attributes (train=80%, test=20%).

   from sklearn.model_selection import train_test_split
   x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.2,random_state=2)

6. Import linear regression medal and optimizing the training variables.

   from sklearn import linear_model
   lin_reg=linear_model.LinearRegression()
   lin_reg.fit(x_train,y_train)

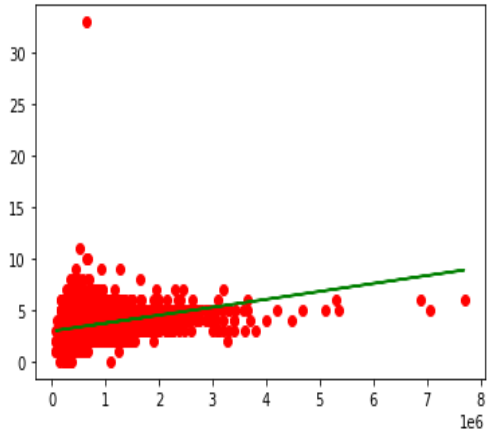
7. Displaying constant and intercept values for the model.

```
In [20]: lin_reg.coef_
Out[20]: array([[1.10105199e-06]])

In [21]: lin_reg.intercept_
Out[21]: array([1.5185242])
```

8. Scatter plot for training dataset.

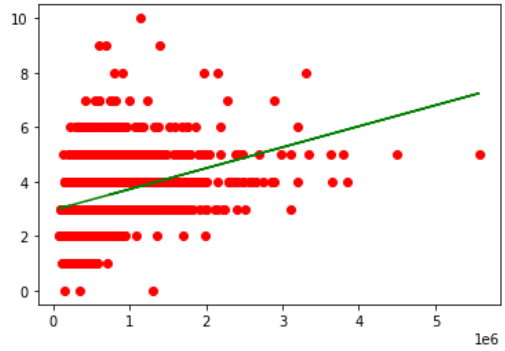
```
import matplotlib.pyplot as plt
plt.scatter(x_train,y_train,color='red')
plt.plot(x_train,lin_reg.predict(x_train),color='green')
plt.show()
```



Here the red spots indicates actual value and the green line indicates regression result.

9. Testing the model and plotting the graph.

```
plt.scatter(x_test,y_test,color='red')
plt.plot(x_test,lin_reg.predict(x_test),color='green')
plt.show()
```



10. Obtaing the r square error and root mean square error values.

```
from sklearn.metrics import mean_squared_error, r2_score
from sklearn.metrics import mean_squared_error, r2_score
import numpy as np
rmse=np.sqrt(mean_squared_error(y_test,ypred))
r_square=r2_score(y_test,ypred)
```

```
r_square
```

```
0.10933217805863649
```

```
rmse
```

```
0.8736548931683022
```

Here we observe that rsquare value is 0.1 hence the model is poor fit.