Simple Linear Regression on Linear Regression set

```
import numpy as np
import pandas as pd
dataset=pd.read_excel("Linear Regression.xlsx",sheet_name=0)
dataset.head()
```

Out[4]:

	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

```
dataset.describe()
```

Out[5]:

	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

```
x=dataset.iloc[:,:1]
```

```
x.head(2)
```

Out[7]:

	price
0	221900

```
1  538000
```

```
y=dataset.iloc[:,1:2]
```

```
y.head(2)
Out[9]:
   sqft_living
0         1180
1         2570
```

```
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=0)
```

```
X_train.shape
Out[12]: (17290, 1)
```

```
y_train.shape
Out[13]: (17290, 1)
```

```
X_test.shape
Out[14]: (4323, 1)
```

```
y_test.shape
Out[15]: (4323, 1)
```

```
from sklearn.linear_model import LinearRegression
lin_reg=LinearRegression()
```

```
lin_reg.fit(X_train,y_train)
Out[19]: LinearRegression(copy_X=True, fit_intercept=True, n_jobs=None, normalize=False)
```

```
lin_reg.coef_
Out[20]: array([[0.00174499]])

lin_reg.intercept_
Out[21]: array([1134.23538784])

ypred=lin_reg.predict(X_test)

ypred
Out[23]:
array([[1652.49798531],
       [3887.83262287],
       [2115.09534083],
       ...,
       [1779.79514519],
       [1657.73296104],
       [2139.26347879]])

from sklearn.metrics import  mean_squared_error,r2_score

RMSE=np.sqrt(mean_squared_error(y_test,ypred))

r_square=r2_score(y_test,ypred)

print("The R square value is :-",r_square)
The R square value is :- 0.4813925927530154

print("The RMSE Value is:-",RMSE)
The RMSE Value is:- 647.3715598616461
```

48.13% the model this predicting good

647 .37 is the total mean error value

Multiple Linear Regression on Linear Regression set

dataset.head(5)

Out[31]:

	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

dataset.describe()

Out[33]:

	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

X=dataset.iloc[:,1:]

X.head()

Out[5]:

	sqft_living	bedrooms	bathrooms	floors
0	1180	3	1.00	1.0
1	2570	3	2.25	2.0
2	770	2	1.00	1.0
3	1960	4	3.00	1.0

4 1680 3 2.00 1.0

```
y=dataset.iloc[:,0:1]

y.head()
Out[7]:
   price
0  221900
1  538000
2  180000
3  604000
4  510000

from sklearn.model_selection import train_test_split

X_train,X_test,y_train,y_test = train_test_split(X,y,test_size=0.3,random_state=2)

dataset.shape
Out[10]: (21613, 5)

from sklearn.linear_model import LinearRegression

mul_reg=LinearRegression()

mul_reg.fit(X_train,y_train)
Out[13]: LinearRegression(copy_X=True, fit_intercept=True, n_jobs=None, normalize=False)

ypred=mul_reg.predict(X_test)

ypred
Out[15]:
array([[609328.77721079],
```

```
       [585038.87031192],
       [415562.45223517],
       ...,
       [599102.75434259],
       [339784.19873135],
       [516024.79183523]])
```

```
from sklearn.metrics import r2_score,mean_squared_error
```

```
print("The R-square...",r2_score(y_test,ypred))
```

The R-square... 0.5105722437453337

```
print("The RMSE value is...", np.sqrt(mean_squared_error(y_test,ypred)))
```

The RMSE value is... 261133.29646851748

51.05% the model this predicting good

261133.29 is the total mean error value