

```
In [2]: #Experiment No.1
```

```
In [ ]: # Aim:To perform and analysis of Linear Regression Algorithm

#Name: Sakshi Padmakar Yeole
#Class: 3rd yr(B)
#Subject:ET-II
#Roll no.:69
```

```
In [2]: import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np
```

```
In [3]: import os
```

```
In [4]: os.getcwd()
```

Out[4]: 'C:\\\\Users\\hp'

```
In [5]: os.chdir("C:\\\\Users\\hp\\Downloads")
```

```
In [6]: df=pd.read_csv("House Price Prediction Dataset (1).csv")
```

```
In [7]: df.head()
```

Out[7]:

	Id	Area	Bedrooms	Bathrooms	Floors	YearBuilt	Location	Condition	Garage	Price
0	1	1360	5	4	3	1970	Downtown	Excellent	No	149919
1	2	4272	5	4	3	1958	Downtown	Excellent	No	424998
2	3	3592	2	2	3	1938	Downtown	Good	No	266746
3	4	966	4	2	2	1902	Suburban	Fair	Yes	244020
4	5	4926	1	4	2	1975	Downtown	Fair	Yes	636056

```
In [8]: df.tail()
```

Out[8]:

	Id	Area	Bedrooms	Bathrooms	Floors	YearBuilt	Location	Condition	Garage	Price
1995	1996	4994	5	4	3	1923	Suburban	Poor	No	295620
1996	1997	3046	5	2	1	2019	Suburban	Poor	Yes	580929
1997	1998	1062	5	1	2	1903	Rural	Poor	No	476925
1998	1999	4062	3	1	2	1936	Urban	Excellent	Yes	161119
1999	2000	2989	5	1	3	1903	Suburban	Fair	No	482525

```
In [9]: df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2000 entries, 0 to 1999
Data columns (total 10 columns):
#   Column      Non-Null Count  Dtype
---  -
0   Id           2000 non-null   int64
1   Area         2000 non-null   int64
2   Bedrooms     2000 non-null   int64
3   Bathrooms    2000 non-null   int64
4   Floors       2000 non-null   int64
5   YearBuilt    2000 non-null   int64
6   Location     2000 non-null   object
7   Condition    2000 non-null   object
8   Garage       2000 non-null   object
9   Price        2000 non-null   int64
dtypes: int64(7), object(3)
memory usage: 156.4+ KB
```

```
In [10]: df.describe()
```

Out[10]:

	Id	Area	Bedrooms	Bathrooms	Floors	YearBuilt	Price
count	2000.000000	2000.000000	2000.000000	2000.000000	2000.000000	2000.000000	2000.000000
mean	1000.500000	2786.209500	3.003500	2.55250	1.993500	1961.446000	537676.855000
std	577.494589	1295.146799	1.424606	1.10899	0.809188	35.926695	276428.845719
min	1.000000	501.000000	1.000000	1.00000	1.000000	1900.000000	50005.000000
25%	500.750000	1653.000000	2.000000	2.00000	1.000000	1930.000000	300098.000000
50%	1000.500000	2833.000000	3.000000	3.00000	2.000000	1961.000000	539254.000000
75%	1500.250000	3887.500000	4.000000	4.00000	3.000000	1993.000000	780086.000000
max	2000.000000	4999.000000	5.000000	4.00000	3.000000	2023.000000	999656.000000

```
In [11]: df.shape
```

Out[11]: (2000, 10)

```
In [12]: df.size
```

Out[12]: 20000

```
In [13]: df.ndim
```

Out[13]: 2

```
In [14]: df.columns
```

Out[14]: Index(['Id', 'Area', 'Bedrooms', 'Bathrooms', 'Floors', 'YearBuilt', 'Location', 'Condition', 'Garage', 'Price'], dtype='object')

```
In [15]: df.isnull().sum()
```

Out[15]: Id 0  
Area 0  
Bedrooms 0  
Bathrooms 0  
Floors 0  
YearBuilt 0  
Location 0  
Condition 0  
Garage 0  
Price 0  
dtype: int64

```
In [19]: df.drop("Id",axis=1,inplace=True)
```

```
In [20]: df.drop("Area",axis=1,inplace=True)
```

```
In [21]: df.head()
```

Out[21]:

	Bedrooms	Bathrooms	Floors	YearBuilt	Location	Condition	Garage	Price
0	5	4	3	1970	Downtown	Excellent	No	149919
1	5	4	3	1958	Downtown	Excellent	No	424998
2	2	2	3	1938	Downtown	Good	No	266746

3	4	2	2	1902	Suburban	Fair	Yes	244020
4	1	4	2	1975	Downtown	Fair	Yes	636056

```
In [23]: X=df.drop("Floors",axis=1)
```

```
In [25]: y=df["Floors"]
```

```
In [27]: # Splitting testdata into x_train,x_test,y_train,y_test
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=42)
```

```
In [34]: print(X_train)
```

	Bedrooms	Bathrooms	YearBuilt	Location	Condition	Garage	Price
968	4	4	1933	Urban	Excellent	No	434325
240	3	3	1970	Downtown	Good	No	614772
819	3	4	1993	Urban	Good	Yes	922811
692	2	3	1972	Rural	Poor	Yes	794314
420	2	4	1981	Rural	Excellent	Yes	796988
...	...	...	...	...	...	...	...
1130	5	3	1931	Downtown	Good	Yes	387088
1294	3	3	1941	Downtown	Fair	Yes	942215
860	2	3	1973	Downtown	Good	No	323139
1459	5	4	1955	Downtown	Good	Yes	274997
1126	2	1	2006	Suburban	Good	No	644206

[1600 rows x 7 columns]

```
In [35]: print(X_test)
```

	Bedrooms	Bathrooms	YearBuilt	Location	Condition	Garage	Price
1860	1	4	1901	Urban	Fair	No	514764
353	1	1	1901	Urban	Fair	No	694256
1333	5	4	2013	Urban	Excellent	No	66375
905	3	1	1996	Downtown	Good	Yes	650243
1289	5	2	1994	Downtown	Good	Yes	223285
...	...	...	...	...	...	...	...
965	1	3	1970	Rural	Poor	No	386940
1284	3	2	2013	Downtown	Poor	Yes	632570
1739	4	4	1939	Downtown	Poor	Yes	494049
261	5	2	1910	Downtown	Good	Yes	562193
535	5	1	1938	Urban	Good	Yes	447530

[400 rows x 7 columns]

```
In [36]: print(y_train)
```

```
968    3
240    1
819    1
692    1
420    1
...
1130   3
1294   2
860    3
1459   1
1126   3
Name: Floors, Length: 1600, dtype: int64
```

```
In [37]: print(y_test)
```

```
1860    2
353     2
1333    1
905     2
1289    3
...
965     3
1284    2
1739    2
```

261 2  
535 3  
Name: Floors, Length: 400, dtype: int64

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js