PROJECT : Banglore place price predict

Import required libraries

In [94]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
df=pd.read_csv('/content/Bengaluru_House_Data.csv')
df
```

Out[94]:

	area_type	availability	location	size	society	total_sqft	bath	balcony	price
0	Super built-up Area	19-Dec	Electronic City Phase II	2 BHK	Coomee	1056	2.0	1.0	39.07
1	Plot Area	Ready To Move	Chikka Tirupathi	4 Bedroom	Theanmp	2600	5.0	3.0	120.00
2	Built-up Area	Ready To Move	Uttarahalli	3 BHK	NaN	1440	2.0	3.0	62.00
3	Super built-up Area	Ready To Move	Lingadheeranahalli	3 BHK	Soiewre	1521	3.0	1.0	95.00
4	Super built-up Area	Ready To Move	Kothanur	2 BHK	NaN	1200	2.0	1.0	51.00
...	...	...	...	...	...	...	...	...	...
13315	Built-up Area	Ready To Move	Whitefield	5 Bedroom	ArsiaEx	3453	4.0	0.0	231.00
13316	Super built-up Area	Ready To Move	Richards Town	4 BHK	NaN	3600	5.0	NaN	400.00
13317	Built-up Area	Ready To Move	Raja Rajeshwari Nagar	2 BHK	Mahla T	1141	2.0	1.0	60.00
13318	Super built-up Area	18-Jun	Padmanabhanagar	4 BHK	SollyCl	4689	4.0	1.0	488.00
13319	Super built-up Area	Ready To Move	Doddathoguru	1 BHK	NaN	550	1.0	1.0	17.00

13320 rows × 9 columns

In [95]:
```
df.head()
```

Out[95]:

	area_type	availability	location	size	society	total_sqft	bath	balcony	price
0	Super built-up Area	19-Dec	Electronic City Phase II	2 BHK	Coomee	1056	2.0	1.0	39.07
1	Plot Area	Ready To Move	Chikka Tirupathi	4 Bedroom	Theanmp	2600	5.0	3.0	120.00
2	Built-up Area	Ready To Move	Uttarahalli	3 BHK	NaN	1440	2.0	3.0	62.00
3	Super built-up Area	Ready To Move	Lingadheeranahalli	3 BHK	Soiewre	1521	3.0	1.0	95.00
4	Super built-up Area	Ready To Move	Kothanur	2 BHK	NaN	1200	2.0	1.0	51.00

In [96]:
```
df.tail()
```

Out[96]:

	area_type	availability	location	size	society	total_sqft	bath	balcony	price
13315	Built-up Area	Ready To Move	Whitefield	5 Bedroom	ArsiaEx	3453	4.0	0.0	231.0
13316	Super built-up Area	Ready To Move	Richards Town	4 BHK	NaN	3600	5.0	NaN	400.0
13317	Built-up Area	Ready To Move	Raja Rajeshwari Nagar	2 BHK	Mahla T	1141	2.0	1.0	60.0
13318	Super built-up Area	18-Jun	Padmanabhanagar	4 BHK	SollyCl	4689	4.0	1.0	488.0
13319	Super built-up Area	Ready To Move	Doddathoguru	1 BHK	NaN	550	1.0	1.0	17.0

In [97]:
```
df.dtypes
```

Out[97]:
```
area_type        object
availability     object
location         object
size             object
society          object
total_sqft       object
bath            float64
balcony         float64
price           float64
dtype: object
```

Checking missing values

In [98]:
```
df.isna().sum()
```

Out[98]:
```
area_type          0
availability       0
location           1
size              16
society         5502
total_sqft         0
bath              73
balcony          609
price              0
dtype: int64
```

In [99]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 13320 entries, 0 to 13319
Data columns (total 9 columns):
 #   Column        Non-Null Count  Dtype  
---  ------        --------------  -----  
 0   area_type     13320 non-null  object 
 1   availability  13320 non-null  object 
 2   location      13319 non-null  object 
 3   size          13304 non-null  object 
 4   society       7818 non-null   object 
 5   total_sqft    13320 non-null  object 
 6   bath          13247 non-null  float64
 7   balcony       12711 non-null  float64
 8   price         13320 non-null  float64
dtypes: float64(3), object(6)
memory usage: 936.7+ KB
```

In [100…
```
for col in df.columns:
  print(df[col].value_counts())
  print('**'*100)
```

```
Super built-up  Area    8790
Built-up  Area          2418
Plot  Area              2025
Carpet  Area              87
Name: area_type, dtype: int64
********************************************************************************************************
Ready To Move    10581
18-Dec             307
18-May             295
18-Apr             271
18-Aug             200
                  ...
15-Aug               1
17-Jan               1
16-Nov               1
16-Jan               1
14-Jul               1
Name: availability, Length: 81, dtype: int64
********************************************************************************************************
Whitefield                        540
Sarjapur  Road                    399
Electronic City                   302
Kanakpura Road                    273
Thanisandra                       234
                                 ...
Bapuji Layout                       1
1st Stage Radha Krishna Layout      1
BEML Layout 5th stage               1
singapura paradise                  1
Abshot Layout                       1
Name: location, Length: 1305, dtype: int64
********************************************************************************************************
2 BHK         5199
3 BHK         4310
4 Bedroom      826
4 BHK          591
3 Bedroom      547
1 BHK          538
2 Bedroom      329
5 Bedroom      297
6 Bedroom      191
1 Bedroom      105
8 Bedroom       84
7 Bedroom       83
5 BHK           59
9 Bedroom       46
6 BHK           30
7 BHK           17
1 RK            13
```

```
10 Bedroom      12
9 BHK            8
8 BHK            5
11 BHK           2
11 Bedroom       2
10 BHK           2
14 BHK           1
13 BHK           1
12 Bedroom       1
27 BHK           1
43 Bedroom       1
16 BHK           1
19 BHK           1
18 Bedroom       1
Name: size, dtype: int64
**********************************************************************************************
GrrvaGr    80
PrarePa    76
Sryalan    59
Prtates    59
GMown E    56
           ..
Amionce     1
JaghtDe     1
Jauraht     1
Brity U     1
RSntsAp     1
Name: society, Length: 2688, dtype: int64
**********************************************************************************************
1200    843
1100    221
1500    205
2400    196
600     180
       ...
3580      1
2461      1
1437      1
2155      1
4689      1
Name: total_sqft, Length: 2117, dtype: int64
**********************************************************************************************
2.0     6908
3.0     3286
4.0     1226
1.0      788
5.0      524
6.0      273
7.0      102
```

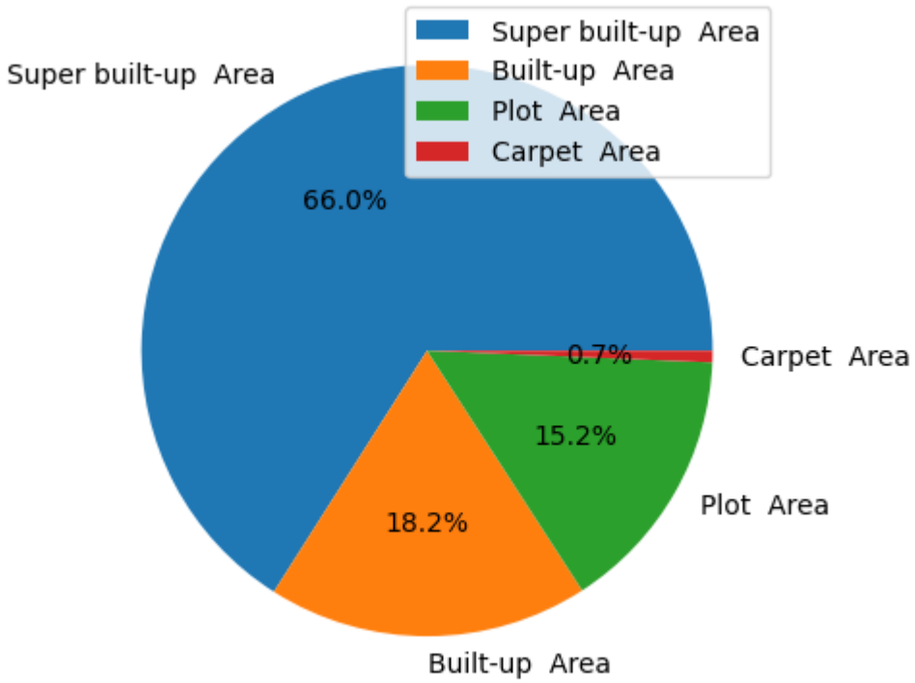
```
8.0       64
9.0       43
10.0      13
12.0       7
13.0       3
11.0       3
16.0       2
27.0       1
40.0       1
15.0       1
14.0       1
18.0       1
Name: bath, dtype: int64
**********************************************************************************************
2.0    5113
1.0    4897
3.0    1672
0.0    1029
Name: balcony, dtype: int64
**********************************************************************************************
75.00     310
65.00     302
55.00     275
60.00     270
45.00     240
         ...
351.00      1
54.10       1
80.64       1
32.73       1
488.00      1
Name: price, Length: 1994, dtype: int64
**********************************************************************************************
```

In [101…
```
area_cnt=df['area_type'].value_counts()
```

Area types Range

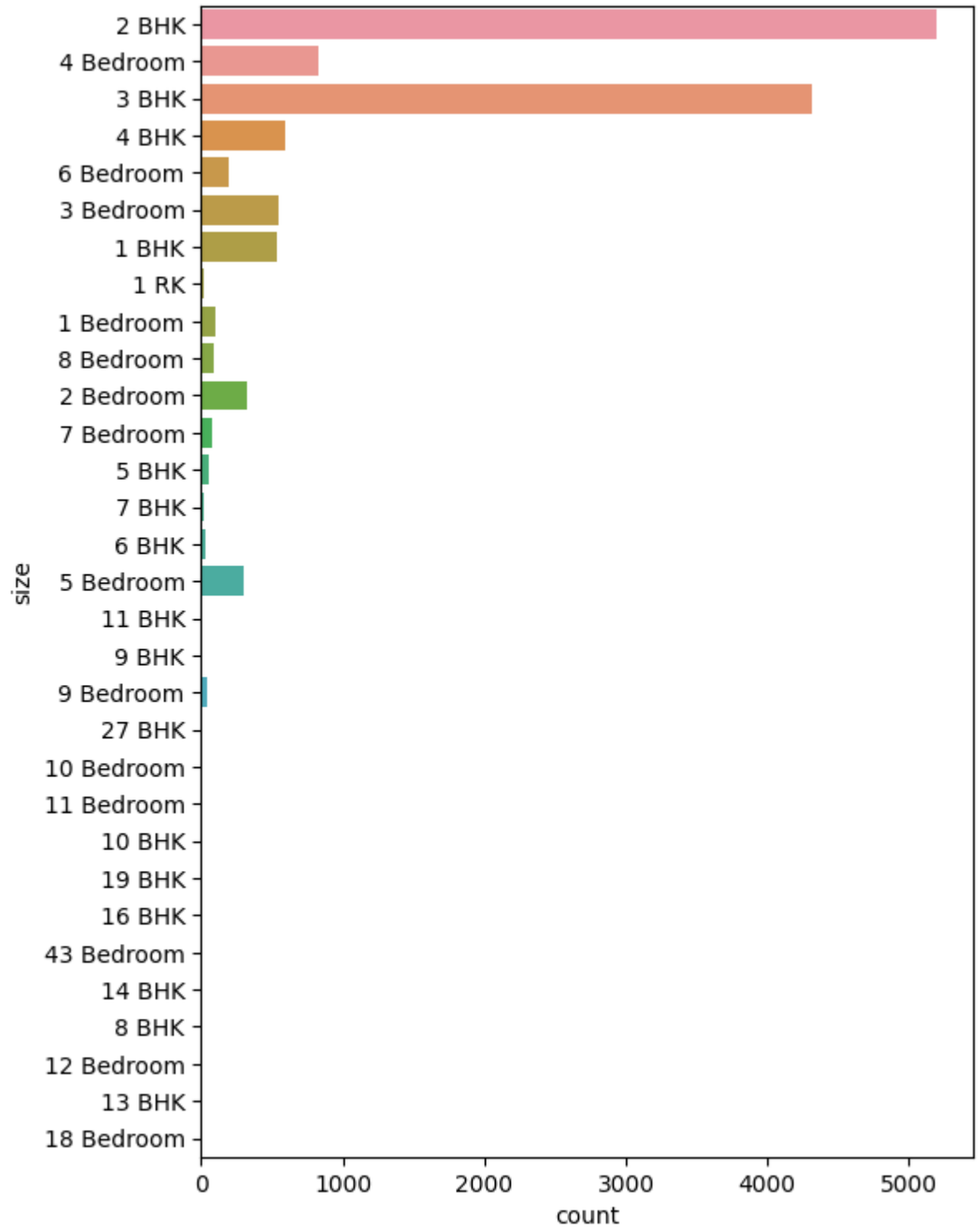
In [102…
```
area=area_cnt.values
area_label=area_cnt.index
plt.pie(area,labels=area_label,autopct='%1.1f%%')
plt.legend(loc='best')
```

Out[102… <matplotlib.legend.Legend at 0x7d5f9ffebf70>



In [103…
```
plt.figure(figsize=(6,9))
sns.countplot(y='size',data=df)
```

Out[103… <Axes: xlabel='count', ylabel='size'>



Remove unwanted columns

In [104…

```
df.drop(['area_type','availability','society','balcony'],axis=1,inplace=True)
```

In [105…
```
df
```

Out[105…

	location	size	total_sqft	bath	price
0	Electronic City Phase II	2 BHK	1056	2.0	39.07
1	Chikka Tirupathi	4 Bedroom	2600	5.0	120.00
2	Uttarahalli	3 BHK	1440	2.0	62.00
3	Lingadheeranahalli	3 BHK	1521	3.0	95.00
4	Kothanur	2 BHK	1200	2.0	51.00
...	...	...	...	...	...
13315	Whitefield	5 Bedroom	3453	4.0	231.00
13316	Richards Town	4 BHK	3600	5.0	400.00
13317	Raja Rajeshwari Nagar	2 BHK	1141	2.0	60.00
13318	Padmanabhanagar	4 BHK	4689	4.0	488.00
13319	Doddathoguru	1 BHK	550	1.0	17.00

13320 rows × 5 columns

Filling the missing values

In [106…
```
df['location']=df['location'].fillna(df['location'].mode()[0])
df['size']=df['size'].fillna(df['size'].mode()[0])
df['bath']=df['bath'].fillna(df['bath'].mean())
```

In [107…
```
df.isna().sum()
```

Out[107…
```
location      0
size          0
total_sqft    0
bath          0
price         0
dtype: int64
```

In [108…
```
df['size']=df['size'].str.replace('BHK','')
df['size']=df['size'].str.replace('Bedroom','')
df['size']=df['size'].str.replace('RK','')
df
```

Out[108…

	location	size	total_sqft	bath	price
0	Electronic City Phase II	2	1056	2.0	39.07
1	Chikka Tirupathi	4	2600	5.0	120.00
2	Uttarahalli	3	1440	2.0	62.00
3	Lingadheeranahalli	3	1521	3.0	95.00
4	Kothanur	2	1200	2.0	51.00
...	...	...	...	...	...
13315	Whitefield	5	3453	4.0	231.00
13316	Richards Town	4	3600	5.0	400.00
13317	Raja Rajeshwari Nagar	2	1141	2.0	60.00
13318	Padmanabhanagar	4	4689	4.0	488.00
13319	Doddathoguru	1	550	1.0	17.00

13320 rows × 5 columns

Convert into Object to Int datatype

In [109…
```
df['size']=df['size'].astype(int)
```

In [110…
```
df.dtypes
```

Out[110…
```
location       object
size            int64
total_sqft     object
bath          float64
price         float64
dtype: object
```

In [111…
```
df['total_sqft'].unique()
```

Out[111…
```
array(['1056', '2600', '1440', ..., '1133 - 1384', '774', '4689'],
      dtype=object)
```

Convert the correct Sqft

In [112…
```
def convertRange(x):
  temp=x.split('-')
  if len(temp)==2:
    return (float(temp[0])+float(temp[1]))/2
  try:
```

```
        return float(x)
    except:
        return None
```

In [113…
```
df['total_sqft']=df['total_sqft'].apply(convertRange)
df.head()
```

Out[113…

	location	size	total_sqft	bath	price
0	Electronic City Phase II	2	1056.0	2.0	39.07
1	Chikka Tirupathi	4	2600.0	5.0	120.00
2	Uttarahalli	3	1440.0	2.0	62.00
3	Lingadheeranahalli	3	1521.0	3.0	95.00
4	Kothanur	2	1200.0	2.0	51.00

Price per sqft

In [114…
```
df['price_per_sqft']=df['price']*100000/df['total_sqft']
```

In [115…
```
df.head()
```

Out[115…

	location	size	total_sqft	bath	price	price_per_sqft
0	Electronic City Phase II	2	1056.0	2.0	39.07	3699.810606
1	Chikka Tirupathi	4	2600.0	5.0	120.00	4615.384615
2	Uttarahalli	3	1440.0	2.0	62.00	4305.555556
3	Lingadheeranahalli	3	1521.0	3.0	95.00	6245.890861
4	Kothanur	2	1200.0	2.0	51.00	4250.000000

In [116…
```
df['total_sqft'].unique()
```

Out[116…
```
array([1056. , 2600. , 1440. , ..., 1258.5,  774. , 4689. ])
```

In [117…
```
df.isna().sum()
```

Out[117…
```
location           0
size               0
total_sqft        46
bath               0
price              0
price_per_sqft    46
dtype: int64
```

In [118…
```
df['total_sqft']=df['total_sqft'].fillna(df['total_sqft'].mean())
```

In [119…
```
df.dtypes
```

Out[119…
```
location           object
size                int64
total_sqft        float64
bath              float64
price             float64
price_per_sqft    float64
dtype: object
```

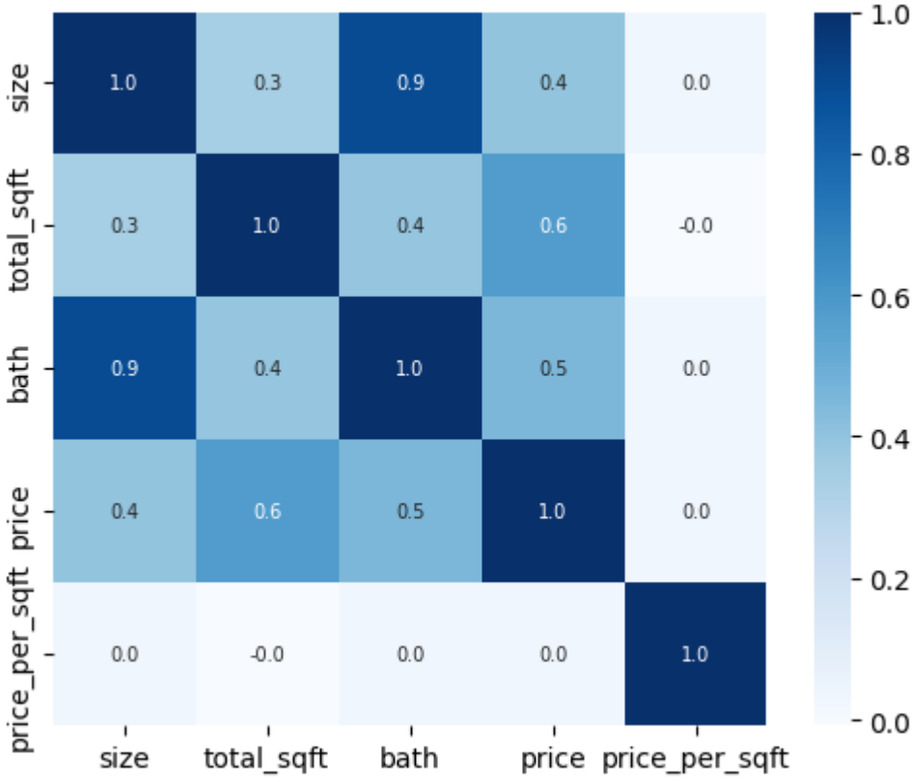
Display the correlation of data

In [120…
```
corr=df.corr()
```

```
<ipython-input-120-0014364bc22a>:1: FutureWarning: The default value of numeric_only in DataFrame.corr is deprecated. In a future version, it w
ill default to False. Select only valid columns or specify the value of numeric_only to silence this warning.
  corr=df.corr()
```

In [121…
```
sns.heatmap(corr,cbar=True,square=True,fmt='.1f',annot=True,annot_kws={'size':7},cmap='Blues')
```

Out[121…
```
<Axes: >
```



Encoding columns using get dummies

In [122…
```
df1=pd.get_dummies(df['location'],drop_first=True)
```

In [123…
```
df1.head()
```

Out[123…

	Banaswadi	Basavangudi	Bhoganhalli	Devarabeesana Halli	Devarachikkanahalli	Electronic City	Mysore Highway	Rachenahalli	Sector 1 HSR Layout	Thanisandra	...	rr nagar	sankeswari	sapthagir Layou
0	0	0	0	0	0	0	0	0	0	0	...	0	0	
1	0	0	0	0	0	0	0	0	0	0	...	0	0	
2	0	0	0	0	0	0	0	0	0	0	...	0	0	
3	0	0	0	0	0	0	0	0	0	0	...	0	0	
4	0	0	0	0	0	0	0	0	0	0	...	0	0	

5 rows × 1304 columns

In [124…
```
dfe=pd.concat([df1,df],axis=1)
```

```
dfe
```

Out[124…

	Banaswadi	Basavangudi	Bhoganhalli	Devarabeesana Halli	Devarachikkanahalli	Electronic City	Mysore Highway	Rachenahalli	Sector 1 HSR Layout	Thanisandra	...	tc.palya	vinayakanagar
0	0	0	0	0	0	0	0	0	0	0	...	0	0
1	0	0	0	0	0	0	0	0	0	0	...	0	0
2	0	0	0	0	0	0	0	0	0	0	...	0	0
3	0	0	0	0	0	0	0	0	0	0	...	0	0
4	0	0	0	0	0	0	0	0	0	0	...	0	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...
13315	0	0	0	0	0	0	0	0	0	0	...	0	0
13316	0	0	0	0	0	0	0	0	0	0	...	0	0
13317	0	0	0	0	0	0	0	0	0	0	...	0	0
13318	0	0	0	0	0	0	0	0	0	0	...	0	0
13319	0	0	0	0	0	0	0	0	0	0	...	0	0

13320 rows × 1310 columns

In [125…

```
dfe.drop(['location','price_per_sqft'],axis=1,inplace=True)
dfe.head()
```

Out[125…

	Banaswadi	Basavangudi	Bhoganhalli	Devarabeesana Halli	Devarachikkanahalli	Electronic City	Mysore Highway	Rachenahalli	Sector 1 HSR Layout	Thanisandra	...	singapura paradise	t.c palya	tc.palya	v
0	0	0	0	0	0	0	0	0	0	0	...	0	0	0	
1	0	0	0	0	0	0	0	0	0	0	...	0	0	0	
2	0	0	0	0	0	0	0	0	0	0	...	0	0	0	
3	0	0	0	0	0	0	0	0	0	0	...	0	0	0	
4	0	0	0	0	0	0	0	0	0	0	...	0	0	0	

5 rows × 1308 columns

Separate X and Y

In [126…
```
x=dfe.iloc[:,:-1].values
y=dfe.iloc[:,-1].values
```

Data into Training and Testing

In [127…
```
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.30,random_state=42)
x_train
```

Out[127…
```
array([[0.000e+00, 0.000e+00, 0.000e+00, ..., 3.000e+00, 2.000e+03,
        3.000e+00],
       [0.000e+00, 0.000e+00, 0.000e+00, ..., 3.000e+00, 2.401e+03,
        3.000e+00],
       [0.000e+00, 0.000e+00, 0.000e+00, ..., 3.000e+00, 1.870e+03,
        3.000e+00],
       ...,
       [0.000e+00, 0.000e+00, 0.000e+00, ..., 2.000e+00, 1.060e+03,
        2.000e+00],
       [0.000e+00, 0.000e+00, 0.000e+00, ..., 6.000e+00, 1.200e+03,
        4.000e+00],
       [0.000e+00, 0.000e+00, 0.000e+00, ..., 3.000e+00, 1.885e+03,
        3.000e+00]])
```

In [128…
```
x_test
y_train
y_test
```

Out[128…
```
array([ 64.8, 125. ,  60. , ..., 235. ,  41.5,  72. ])
```

Model Creation using

- LinearRegression**
- RandomForestRegressor
- XGBRegressor

LinearRegression

In [129…
```
from sklearn.linear_model import LinearRegression
model=LinearRegression()
model.fit(x_train,y_train)
y_pred=model.predict(x_test)
y_pred
```

Out[129…
```
array([ 22.03037049, 100.99941869,  90.28129489, ..., 238.97657366,
        66.56751911,  55.34846665])
```

In [130…
```
df2=pd.DataFrame({'act_value':y_test,'pred_value':y_pred,'diff':y_test-y_pred})
```

```
df2
```

Out[130…

	act_value	pred_value	diff
0	64.80	22.030370	42.769630
1	125.00	100.999419	24.000581
2	60.00	90.281295	-30.281295
3	110.00	124.085317	-14.085317
4	210.00	149.845355	60.154645
...	...	...	...
3991	45.03	86.591596	-41.561596
3992	74.00	71.464494	2.535506
3993	235.00	238.976574	-3.976574
3994	41.50	66.567519	-25.067519
3995	72.00	55.348467	16.651533

3996 rows × 3 columns

In [131…
```
from sklearn.metrics import mean_absolute_percentage_error
print('MAPE: ',mean_absolute_percentage_error(y_test,y_pred))
```

```
MAPE:  3491066.5333451806
```

In [132…
```
from sklearn.metrics import r2_score
print('R2_SCORE: ',r2_score(y_test,y_pred))
```

```
R2_SCORE:  -1.0851471553275484e+16
```

RandomForestRegressor

In [134…
```
from sklearn.ensemble import RandomForestRegressor
rfg=RandomForestRegressor()
rfg.fit(x_train,y_train)
y_pred2=rfg.predict(x_test)
y_pred2
```

Out[134…
```
array([ 43.27228333,  96.03271429,  44.16734921, ..., 168.84      ,
        52.6237    ,  59.63096667])
```

In [135…
```
df3=pd.DataFrame({'act_value':y_test,'pred_value':y_pred2,'diff':y_test-y_pred2})
df3
```

Out[135…

	act_value	pred_value	diff
0	64.80	43.272283	21.527717
1	125.00	96.032714	28.967286
2	60.00	44.167349	15.832651
3	110.00	119.180000	-9.180000
4	210.00	159.770000	50.230000
...	...	...	...
3991	45.03	47.769900	-2.739900
3992	74.00	48.880000	25.120000
3993	235.00	168.840000	66.160000
3994	41.50	52.623700	-11.123700
3995	72.00	59.630967	12.369033

3996 rows × 3 columns

In [136…
```
from sklearn.metrics import mean_absolute_percentage_error
print('MAPE: ',mean_absolute_percentage_error(y_test,y_pred2))
```

```
MAPE:  0.23334932304132378
```

In [137…
```
from sklearn.metrics import r2_score
print('R2 SCORE: ',r2_score(y_test,y_pred2))
```

```
R2 SCORE:  0.5884429368529496
```

XGBRegressor

In [138…
```
from xgboost import XGBRegressor
xgb=XGBRegressor()
xgb.fit(x_train,y_train)
y_pred3=xgb.predict(x_test)
y_pred3
```

Out[138…
```
array([ 49.21886 , 104.56958 ,  57.43917 , ..., 170.5894  ,  68.550186,
        49.21886 ], dtype=float32)
```

In [139…
```
df4=pd.DataFrame({'act_value':y_test,'pred_value':y_pred3,'diff':y_test-y_pred3})
df4
```

Out[139…

	act_value	pred_value	diff
0	64.80	49.218861	15.581139
1	125.00	104.569580	20.430420
2	60.00	57.439171	2.560829
3	110.00	155.423645	-45.423645
4	210.00	223.124313	-13.124313
...	...	...	...
3991	45.03	74.053177	-29.023177
3992	74.00	63.746330	10.253670
3993	235.00	170.589401	64.410599
3994	41.50	68.550186	-27.050186
3995	72.00	49.218861	22.781139

3996 rows × 3 columns

In [140…

```
from sklearn.metrics import mean_absolute_percentage_error
print('MAPE: ',mean_absolute_percentage_error(y_test,y_pred3))
```

```
MAPE:  0.3046481532986313
```

In [141…

```
from sklearn.metrics import r2_score
print('R2 SCORE: ',r2_score(y_test,y_pred3))
```

```
R2 SCORE:  0.6796815271794361
```