In [1]:
```
import numpy as np
import pandas as pd
import warnings
warnings.filterwarnings("ignore")
df=pd.read_csv("Boston.csv")
df.drop(df.columns[0],axis=1,inplace=True)
df.rename(columns={'medv':'Price'},inplace=True)
df
```

Out[1]:

	crim	zn	indus	chas	nox	rm	age	dis	rad	tax	ptratio	black	lstat	Price
0	0.00632	18.0	2.31	0	0.538	6.575	65.2	4.0900	1	296	15.3	396.90	4.98	24.0
1	0.02731	0.0	7.07	0	0.469	6.421	78.9	4.9671	2	242	17.8	396.90	9.14	21.6
2	0.02729	0.0	7.07	0	0.469	7.185	61.1	4.9671	2	242	17.8	392.83	4.03	34.7
3	0.03237	0.0	2.18	0	0.458	6.998	45.8	6.0622	3	222	18.7	394.63	2.94	33.4
4	0.06905	0.0	2.18	0	0.458	7.147	54.2	6.0622	3	222	18.7	396.90	5.33	36.2
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
501	0.06263	0.0	11.93	0	0.573	6.593	69.1	2.4786	1	273	21.0	391.99	9.67	22.4
502	0.04527	0.0	11.93	0	0.573	6.120	76.7	2.2875	1	273	21.0	396.90	9.08	20.6
503	0.06076	0.0	11.93	0	0.573	6.976	91.0	2.1675	1	273	21.0	396.90	5.64	23.9
504	0.10959	0.0	11.93	0	0.573	6.794	89.3	2.3889	1	273	21.0	393.45	6.48	22.0
505	0.04741	0.0	11.93	0	0.573	6.030	80.8	2.5050	1	273	21.0	396.90	7.88	11.9

506 rows × 14 columns

In [2]:
```
x=df.iloc[:,:-1]
x
```

Out[2]:

	crim	zn	indus	chas	nox	rm	age	dis	rad	tax	ptratio	black	lstat
0	0.00632	18.0	2.31	0	0.538	6.575	65.2	4.0900	1	296	15.3	396.90	4.98
1	0.02731	0.0	7.07	0	0.469	6.421	78.9	4.9671	2	242	17.8	396.90	9.14
2	0.02729	0.0	7.07	0	0.469	7.185	61.1	4.9671	2	242	17.8	392.83	4.03
3	0.03237	0.0	2.18	0	0.458	6.998	45.8	6.0622	3	222	18.7	394.63	2.94
4	0.06905	0.0	2.18	0	0.458	7.147	54.2	6.0622	3	222	18.7	396.90	5.33
...	...	...	...	...	...	...	...	...	...	...	...	...	...
501	0.06263	0.0	11.93	0	0.573	6.593	69.1	2.4786	1	273	21.0	391.99	9.67
502	0.04527	0.0	11.93	0	0.573	6.120	76.7	2.2875	1	273	21.0	396.90	9.08
503	0.06076	0.0	11.93	0	0.573	6.976	91.0	2.1675	1	273	21.0	396.90	5.64
504	0.10959	0.0	11.93	0	0.573	6.794	89.3	2.3889	1	273	21.0	393.45	6.48
505	0.04741	0.0	11.93	0	0.573	6.030	80.8	2.5050	1	273	21.0	396.90	7.88

506 rows × 13 columns

In [3]:
```
y=df.Price
y
```

Out[3]:
```
0      24.0
1      21.6
2      34.7
3      33.4
4      36.2
       ...
501    22.4
502    20.6
503    23.9
504    22.0
505    11.9
Name: Price, Length: 506, dtype: float64
```

In [4]:
```
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=10)
```

In [5]:
```
from sklearn.linear_model import LinearRegression
model=LinearRegression()
model.fit(x_train,y_train)
```

Out[5]:
```
LinearRegression()
```

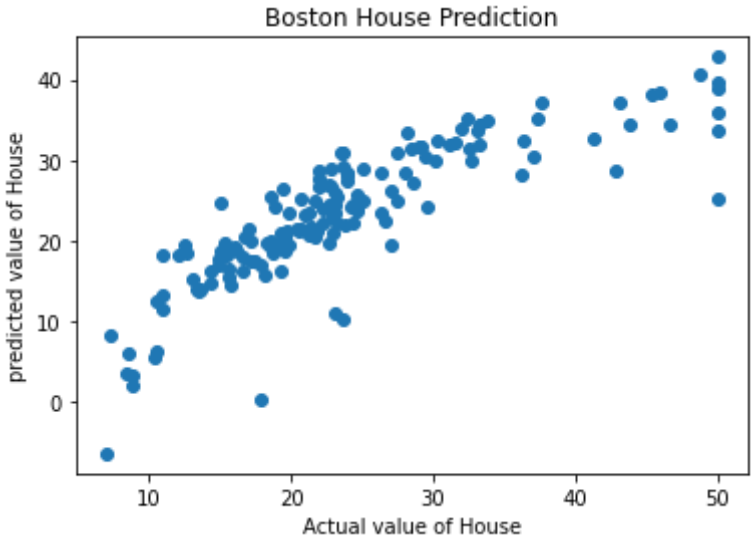
In [6]:
```
y_predict=model.predict(x_test)
y_predict
```

Out[6]:
```
array([31.4243217 , 31.96785487, 30.93785448, 22.34313349, 18.83846235,
       16.20617519, 35.92908162, 14.74157477, 25.07700756, 37.13230282,
       21.47652971, 30.92661826, 28.07823424, 34.02599249, 33.7778476 ,
       40.63701192, 24.25899783, 23.43019291, 25.547906  , 21.34469147,
       32.65467539, 17.80506124, 25.46149722, 25.0207691 , 32.51742137,
       20.51357936, 19.47165255, 16.87107974, 38.44316206,  0.3888111 ,
       32.39559257, 32.15518102, 26.05305015, 23.82049084, 20.56494632,
       19.66990981,  3.53212643, 35.21058387, 27.03280773, 27.67994129,
       34.36642896, 29.82003002, 18.31717228, 31.55109654, 17.93465111,
       28.4618882 , 19.39950216, 21.60782793, 38.10391926, 16.45101411,
       24.51003632, 19.57072199, 24.53359986, 34.34589029, 26.74381857,
       34.86340026, 21.02859444, 19.77400901, 18.68461884, 24.64911818,
       19.89923131, 23.52871967, 39.56690084, 42.81848202, 30.38116596,
       16.96889789, 23.83550068,  3.18952008, 31.52469303, 28.66536677,
       18.40003941, 27.1252398 , 19.56625808, 25.28579071, 25.04642543,
       10.30562974, 38.97693514,  8.17901104, 18.51160158, 30.32060192,
       22.85826904, 20.98739809, 20.03256896, 28.70004491, 30.83986409,
       28.22575374, 26.26715448, 31.56221764, 22.16821072, -6.40058282,
       21.56768106, 19.91022274, 24.98025707, 23.46247865, 19.27692546,
       18.74766072, 27.39474857, 22.21870161, 26.82617865, 23.38366235,
       23.91947259, 19.26003338, 21.09721048, 10.97767848, 13.82483416,
       20.72378817, 23.53977916, 13.92807616, 28.87070782, 15.75931299,
       15.18847286, 22.31904753, 26.57865025, 28.92895886, 24.2980433 ,
       18.18585276, 16.29471281, 17.40018499, 15.5361744 , 21.171008  ,
       33.71665888, 30.02422884, 21.13474092, 13.93326869, 16.19662736,
       29.30890971, 13.1822158 , 22.06473155, 24.33753588, 31.89978763,
       33.44295992,  5.90754764, 35.2142013 , 24.1925307 , 17.45166483,
       24.25550318, 28.41864261, 34.52462088,  6.2544935 ,  2.12089952,
       28.40739835, 12.49858357, 18.22016397, 19.69247114,  5.52373677,
       14.45613717, 37.18224665, 25.78695839, 23.32233462, 26.36634945,
       11.47817153, 20.44901468])
```

In [7]:
```
from sklearn.metrics import mean_squared_error
mean_squared_error(y_test,y_predict)
```

Out[7]:
```
29.32659652612324
```

In [8]:
```
import matplotlib.pyplot as plt
plt.scatter(y_test,y_predict)
plt.xlabel("Actual value of House")
plt.ylabel("predicted value of House")
plt.title("Boston House Prediction")
plt.show()
```



In [ ]: