

In [1]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

In [2]:

```
df=pd.read_csv("/home/ubuntu/Downloads/HousingData.csv")
df
```

Out[2]:

	CRIM	ZN	INDUS	CHAS	NOX	RM	AGE	DIS	RAD	TAX	PTRATIO	B	L
0	0.00632	18.0	2.31	0.0	0.538	6.575	65.2	4.0900	1	296	15.3	396.90	
1	0.02731	0.0	7.07	0.0	0.469	6.421	78.9	4.9671	2	242	17.8	396.90	
2	0.02729	0.0	7.07	0.0	0.469	7.185	61.1	4.9671	2	242	17.8	392.83	
3	0.03237	0.0	2.18	0.0	0.458	6.998	45.8	6.0622	3	222	18.7	394.63	
4	0.06905	0.0	2.18	0.0	0.458	7.147	54.2	6.0622	3	222	18.7	396.90	
...	...	...	...	...	...	...	...	...	...	...	...	...	...
501	0.06263	0.0	11.93	0.0	0.573	6.593	69.1	2.4786	1	273	21.0	391.99	
502	0.04527	0.0	11.93	0.0	0.573	6.120	76.7	2.2875	1	273	21.0	396.90	
503	0.06076	0.0	11.93	0.0	0.573	6.976	91.0	2.1675	1	273	21.0	396.90	
504	0.10959	0.0	11.93	0.0	0.573	6.794	89.3	2.3889	1	273	21.0	393.45	
505	0.04741	0.0	11.93	0.0	0.573	6.030	NaN	2.5050	1	273	21.0	396.90	

506 rows × 14 columns

In [3]:

```
df=pd.set_option("display.max_rows",None)
```

In [4]:

```
df=pd.read_csv("/home/ubuntu/Downloads/HousingData.csv")
df
```

Out[4]:

	CRIM	ZN	INDUS	CHAS	NOX	RM	AGE	DIS	RAD	TAX	PTRATIO	B	LSTAT	MEDV
0	0.00632	18.0	2.31	0.0	0.5380	6.575	65.2	4.0900	1	296	15.3	396.90	4.98	212.0
1	0.02731	0.0	7.07	0.0	0.4690	6.421	78.9	4.9671	2	242	17.8	396.90	9.14	262.0
2	0.02729	0.0	7.07	0.0	0.4690	7.185	61.1	4.9671	2	242	17.8	392.83	4.03	318.0
3	0.03237	0.0	2.18	0.0	0.4580	6.998	45.8	6.0622	3	222	18.7	394.63	2.94	333.0
4	0.06905	0.0	2.18	0.0	0.4580	7.147	54.2	6.0622	3	222	18.7	396.90	NaN	337.0
5	0.02985	0.0	2.18	0.0	0.4580	6.430	58.7	6.0622	3	222	18.7	394.12	5.21	263.0
6	0.08829	12.5	7.87	NaN	0.5240	6.012	66.6	5.5605	5	311	15.2	395.60	12.43	278.0
7	0.14455	12.5	7.87	0.0	0.5240	6.172	96.1	5.9505	5	311	15.2	396.90	19.15	264.0
8	0.21124	12.5	7.87	0.0	0.5240	5.631	100.0	6.0821	5	311	15.2	386.63	29.93	178.0
9	0.17004	12.5	7.87	NaN	0.5240	6.004	85.9	6.5921	5	311	15.2	386.71	17.10	179.0

In [5]:

```
df.shape
```

Out[5]:

(506, 14)

In [6]:

```
df.isnull().sum()
```

Out[6]:

```
CRIM      20
ZN        20
INDUS     20
CHAS      20
NOX       0
RM        0
AGE       20
DIS       0
RAD       0
TAX       0
PTRATIO   0
B         0
LSTAT     20
MEDV      0
dtype: int64
```

In [7]:

```
df['CRIM'].fillna(df['CRIM'].mean(),inplace=True)
df['ZN'].fillna(df['ZN'].mean(),inplace=True)
df['INDUS'].fillna(df['INDUS'].mean(),inplace=True)
df['CHAS'].fillna(df['CHAS'].mean(),inplace=True)
df['AGE'].fillna(df['AGE'].mean(),inplace=True)
df['LSTAT'].fillna(df['LSTAT'].mean(),inplace=True)
```

In [8]:

df

Out[8]:

	CRIM	ZN	INDUS	CHAS	NOX	RM	AGE	DIS	RAD	TAX	PTRATIO	
0	0.006320	18.000000	2.310000	0.000000	0.5380	6.575	65.200000	4.0900	1	296	15.3	3
1	0.027310	0.000000	7.070000	0.000000	0.4690	6.421	78.900000	4.9671	2	242	17.8	3
2	0.027290	0.000000	7.070000	0.000000	0.4690	7.185	61.100000	4.9671	2	242	17.8	3
3	0.032370	0.000000	2.180000	0.000000	0.4580	6.998	45.800000	6.0622	3	222	18.7	3
4	0.069050	0.000000	2.180000	0.000000	0.4580	7.147	54.200000	6.0622	3	222	18.7	3
5	0.029850	0.000000	2.180000	0.000000	0.4580	6.430	58.700000	6.0622	3	222	18.7	3
6	0.088290	12.500000	7.870000	0.069959	0.5240	6.012	66.600000	5.5605	5	311	15.2	3
7	0.144550	12.500000	7.870000	0.000000	0.5240	6.172	96.100000	5.9505	5	311	15.2	3
8	0.211240	12.500000	7.870000	0.000000	0.5240	5.631	100.000000	6.0821	5	311	15.2	3
9	0.170040	12.500000	7.870000	0.069959	0.5240	6.004	85.900000	6.5921	5	311	15.2	3

In [9]:

df.isnull().sum()

Out[9]:

```
CRIM      0
ZN         0
INDUS      0
CHAS       0
NOX        0
RM         0
AGE        0
DIS        0
RAD        0
TAX        0
PTRATIO    0
B          0
LSTAT      0
MEDV       0
dtype: int64
```

In [10]:

```
x=df.drop(['MEDV'],axis=1)
x
```

Out[10]:

	CRIM	ZN	INDUS	CHAS	NOX	RM	AGE	DIS	RAD	TAX	PTRATIO	
0	0.006320	18.000000	2.310000	0.000000	0.5380	6.575	65.200000	4.0900	1	296	15.3	3
1	0.027310	0.000000	7.070000	0.000000	0.4690	6.421	78.900000	4.9671	2	242	17.8	3
2	0.027290	0.000000	7.070000	0.000000	0.4690	7.185	61.100000	4.9671	2	242	17.8	3
3	0.032370	0.000000	2.180000	0.000000	0.4580	6.998	45.800000	6.0622	3	222	18.7	3
4	0.069050	0.000000	2.180000	0.000000	0.4580	7.147	54.200000	6.0622	3	222	18.7	3
5	0.029850	0.000000	2.180000	0.000000	0.4580	6.430	58.700000	6.0622	3	222	18.7	3
6	0.088290	12.500000	7.870000	0.069959	0.5240	6.012	66.600000	5.5605	5	311	15.2	3
7	0.144550	12.500000	7.870000	0.000000	0.5240	6.172	96.100000	5.9505	5	311	15.2	3
8	0.211240	12.500000	7.870000	0.000000	0.5240	5.631	100.000000	6.0821	5	311	15.2	3
9	0.170040	12.500000	7.870000	0.069959	0.5240	6.004	85.900000	6.5921	5	311	15.2	3

In [11]:

```
y=df['MEDV']
y
```

Out[11]:

```
0    24.0
1    21.6
2    34.7
3    33.4
4    36.2
5    28.7
6    22.9
7    27.1
8    16.5
9    18.9
10   15.0
11   18.9
12   21.7
13   20.4
14   18.2
15   19.9
16   23.1
17   17.5
```

In [12]:

```
from sklearn.model_selection import train_test_split
xtrain,xtest,ytrain,ytest=train_test_split(x,y,test_size=0.2,random_state=0)
```

In [13]:

```
from sklearn.linear_model import LinearRegression
lm=LinearRegression()
model=lm.fit(xtrain,ytrain)
ytrain_pred=lm.predict(xtrain)
ytest_pred=lm.predict(xtest)
```

In [14]:

```
ytrain_pred
```

Out[14]:

```
array([32.81627321, 22.44810156, 28.03766232, 23.75733198,  6.509474
14,
      14.03444098, 22.08820099, 29.25415603, 32.5690748 , 13.056409
04,
      20.22623633, 21.50935141, 13.130115 , 23.97459357,  5.943695
26,
      19.18872715,  9.22208539, 45.332114 , 30.74904711, 17.365082
71,
      17.89690787, 21.8538289 , 23.41427455, 19.26643055, 35.010609
2 ,
      13.80465069, 20.97652211, 35.5947274 , 19.11750028, 13.645653
13,
      13.99390551, 22.3513967 , 14.97018808, 31.41563776, 25.422005
97,
      16.12261841, 24.8151741 ,  9.77305866, 14.99985459, 21.706950
3 ,
      33.07056401, 28.27079451, 24.99059687, 15.53957856, 31.813107
3 ,
      25.31962943, 14.20026892,  7.85538779, 27.95901931, 25.432165
39,
      5.06592749, 28.31123002, 16.96468964, 29.9148188 , 19.391086
37,
      16.30580326, 18.42774446, 12.95370034,  8.93495761, 19.258867
82,
      34.02016271, 32.86420157, 23.95497256, 20.14492197, 22.950837
75,
      26.67615398, 21.10735155, 17.74979344, 32.38278077, 10.582547
31,
      19.13318567, 31.64516362, 18.84520418, 15.78422296, 18.940469
27,
      15.11678414, 24.07458559, 23.66247081, 17.31713844, 13.219102
58,
      20.51857201, 24.22417338, 17.56807076, 25.39478716, 22.844986
87,
      28.06459103, 36.75378827, 16.43162613, 12.07850424, 35.047467
39,
      31.32063269, 20.48029146, 39.84323922, 28.61226369, 28.522332
59,
      17.5436698 , 26.81225348, 40.46530719, 27.56913544, 17.038835
67,
      37.52638916, 35.8229176 , 14.0961138 , 27.81731446, 21.948696
55,
      24.8818496 , 21.19719131, 23.45740081, 28.00487668, 29.574315
57,
      13.98506722, 26.16081663, 22.9901289 , 13.49729741, 14.068399
36,
      25.56540702, 19.52524407, 30.57606788, 10.04770818, 24.371600
39,
      17.17149942, 17.0112022 , 22.84937882, 21.67962795, 12.220635
91,
      25.20446182, 28.42197799, 20.75309338, 12.37886993, 25.113060
51,
      26.49461989, 25.70733316, 23.55013679, 25.89336895, 19.395536
91,
      20.85778701, 36.13738934, 21.1272548 , 36.28019204, 25.817084
```

```
51,
    20.90978431, 15.58319854, 31.98314685, 21.54080142, 28.053963
55,
    14.844027 , 32.73726538, 14.35524691, 1.67919971, 19.447111
42,
    13.87157102, 37.66832351, 16.30974204, 14.5549353 , 26.769510
17,
    23.52976973, 17.9629898 , 31.18728628, 25.18377746, 27.615761
9 ,
    24.80190806, 22.65264492, 22.46031495, 11.15172696, 20.918422
34,
    11.68647636, 17.73974341, 12.46032947, 27.77373401, 15.192514
31,
    15.99965125, 28.65756198, 14.46036826, 21.62746997, 12.642543
29,
    14.45555483, 23.36423324, 21.29797228, 14.93440596, 17.486499
66,
    13.45046456, 24.24295419, 12.50361345, 35.3211587 , 14.037601
67,
    43.21873439, 31.66742609, 34.8841818 , 22.01139076, 15.821800
35,
    26.85423351, 29.2313597 , 13.77457177, 26.727208 , 36.279458
42,
    16.83686285, 11.36251837, 34.42259379, 35.99230127, 17.961330
76,
    21.19847959, 18.52795439, 24.4993728 , 19.58258696, 27.190066
6 ,
    -4.30707212, 20.86540926, 32.56708668, 35.64347839, 25.171237
63,
    26.89080764, 20.27656175, 21.38857599, 15.98382948, 17.861325
17,
    21.07756281, 28.04291179, 19.96986817, 6.97597498, 16.237431
92,
    32.36451304, 35.49074744, 16.44786434, 18.93927412, 22.338548
17,
    6.31311559, 21.58666499, 23.55661668, 15.92257792, 18.476279
52,
    23.05203916, 27.21603015, 25.92177059, 32.83874569, 14.876935
91,
    28.97948914, 25.26104179, 21.07941467, 38.66910342, 20.485238
52,
    23.62439463, 22.76124186, 11.92347536, 19.99770627, 33.427698
53,
    24.80122075, 17.81400458, 33.31685947, 22.07230652, 28.718578
35,
    32.15452084, 36.60591303, 21.92092982, 24.10484055, 23.116761
77,
    32.01346165, 22.26638068, 18.36298427, 21.92771527, 29.118530
42,
    22.86612836, 22.06878986, 17.24027069, 17.3597754 , 16.974728
58,
    16.93337092, 16.77794732, 32.03226006, 23.32756224, 17.499838
57,
    19.30181049, 34.16756391, 14.14434098, 25.93018821, 17.019477
28,
    30.72513401, 30.03828448, 21.25506147, 20.3298965 , 36.068969
82,
    20.55702264, 33.38547011, 21.24908409, 31.57709481, 30.278972
98,
    37.42356288, 25.97972218, 21.09103073, 29.15810192, 15.961089
91,
```

```
26.18276066, 21.50175208, 30.02080424, 10.53454993, 31.348920
03,
6.17985967, 15.25781318, 20.54161826, 35.68172035, 31.937788
97,
12.06580266, 13.62606496, 21.96053599, 34.80135807, 18.762017
57,
18.62293267, 14.91881869, 25.48278612, 41.01931145, 25.348711
74,
42.07887857, 25.45001621, 21.13187064, 12.07131249, 15.869974
46,
14.20602197, 18.51957266, 3.01474407, 27.73065016, 26.412216
64,
41.70198443, 21.64461258, 21.05335926, 34.0757984 , 32.950548
01,
9.64398016, 24.85070793, 43.79024787, 21.78807344, 17.692057
4 ,
26.16218871, 18.58628594, 6.34645098, 18.91507749, 35.663190
94,
16.30328017, 23.75087974, 13.14594653, 24.39977842, 18.271
,
17.18171595, 18.47353031, 33.00268863, 19.48068366, 29.839622
14,
31.93425646, 41.60423999, 18.49648542, 16.12293921, 38.255106
56,
17.76581487, 10.55196656, 14.7688579 , 25.35067163, 19.469649
74,
16.46915179, 26.67604749, 13.37142686, 5.91891273, 18.718167
01,
10.83757858, 28.50859921, 4.96207098, 28.66572375, 32.808750
45,
22.61360755, 16.44226312, 17.93061892, 21.13503675, 34.026546
74,
28.24386741, 19.33997782, 20.46592732, 6.76629059, 28.974808
48,
25.03253899, 22.44360225, 13.7417905 , 24.80011236, 19.463037
25,
8.97368226, 26.78806235, 16.0120091 , 31.5636985 , 31.995305
44,
25.19164941, 18.43188624, 30.61644449, 21.28067267, 25.931139
5 ,
24.31670018, 31.19585054, 24.60225016, 31.38261045, 17.644089
55,
19.80887282, 18.65517922, 41.33253844, 25.56986444, 19.340473
26,
33.40599414, 23.70130494, 18.3221544 , 23.25156216])
```



In [15]:

```
model.predict([[0.006320,18.000000,2.310000,0.000000,0.5380,6.575,65.200000,4.0900,
ytest_pred
```

```
/home/ubuntu/.local/lib/python3.8/site-packages/sklearn/base.py:420: UserWarning: X does not have valid feature names, but LinearRegression was fitted with feature names
  warnings.warn(
```

Out[15]:

```
array([26.175296 , 22.64747588, 29.1456294 , 11.52971235, 21.6531213
4,
      19.42320699, 20.18413017, 21.46914355, 19.1985363 , 19.9822816
2,
      4.32483046, 16.16891668, 16.87682404,  5.31232373, 39.3682786
1,
      33.09358732, 21.9152876 , 36.61918436, 31.52676377, 23.5271348
2,
      24.96022461, 23.69866912, 20.88033802, 30.55074901, 22.7408174
1,
      8.66805959, 17.65119072, 17.93088633, 36.01223185, 21.1629955
6,
      17.83464361, 17.43306603, 19.5240167 , 23.50605522, 28.9726279
3,
      19.21808862, 11.23997435, 23.94256597, 17.86786717, 15.4084980
6,
      26.3630836 , 21.5193299 , 23.78733694, 14.84041522, 23.9445175
,
      24.97067627, 20.11366175, 23.08636158, 10.42208266, 24.5283212
2,
      21.60847326, 18.66228165, 24.53362832, 31.03502944, 12.9745782
6,
      22.38536236, 21.34822822, 16.10928673, 12.37477824, 22.7859671
2,
      18.28714824, 21.91802045, 32.49771603, 31.21256855, 17.4786779
1,
      33.18861907, 19.17896285, 19.94662594, 20.17142015, 23.9022885
7,
      22.81288844, 24.17911208, 30.83402844, 28.87481037, 25.1458172
1,
      5.55072029, 37.0183454 , 24.15428003, 27.67587636, 19.6388464
4,
      28.74874123, 18.83204358, 17.63305678, 37.97947167, 39.4950797
2,
      24.17228966, 25.33605088, 16.75044819, 25.43224687, 16.6508942
6,
      16.49186628, 13.37283452, 24.81689254, 31.21188699, 22.0891919
,
      20.49360168,  0.8229737 , 25.5004737 , 15.5481509 , 17.7290119
3,
      25.77663998, 22.43131323])
```

In [16]:

```
df1=pd.DataFrame(ytrain_pred,ytrain)
df2=pd.DataFrame(ytest_pred,ytest)
df1
df2
```

13.8	14.840415
16.5	23.944517
21.6	24.970676
20.3	20.113662
17.0	23.086362
11.8	10.422083
27.5	24.528321
15.6	21.608473
23.1	18.662282
24.3	24.533628
42.8	31.035029
15.6	12.974578
21.7	22.385362

In [17]:

```
from sklearn.metrics import mean_squared_error,r2_score
mse=mean_squared_error(ytest,ytest_pred)
print (mse)
```

34.98738954423878

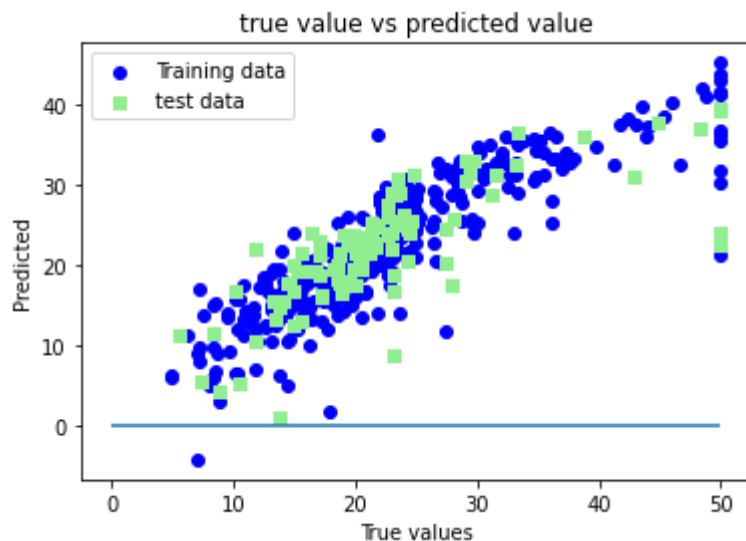
In [18]:

```
mse=mean_squared_error(ytrain,ytrain_pred)
print (mse)
```

20.019115913036593

In [20]:

```
import matplotlib.pyplot as plt
plt.scatter(ytrain,ytrain_pred,c='blue',marker='o',label='Training data')
plt.scatter(ytest,ytest_pred,c='lightgreen',marker='s',label='test data')
plt.xlabel('True values')
plt.ylabel('Predicted')
plt.title("true value vs predicted value")
plt.legend(loc='upper left')
plt.hlines(y=0,xmin=0,xmax=50)
plt.plot()
plt.show()
```



In [ ]:

In [ ]: