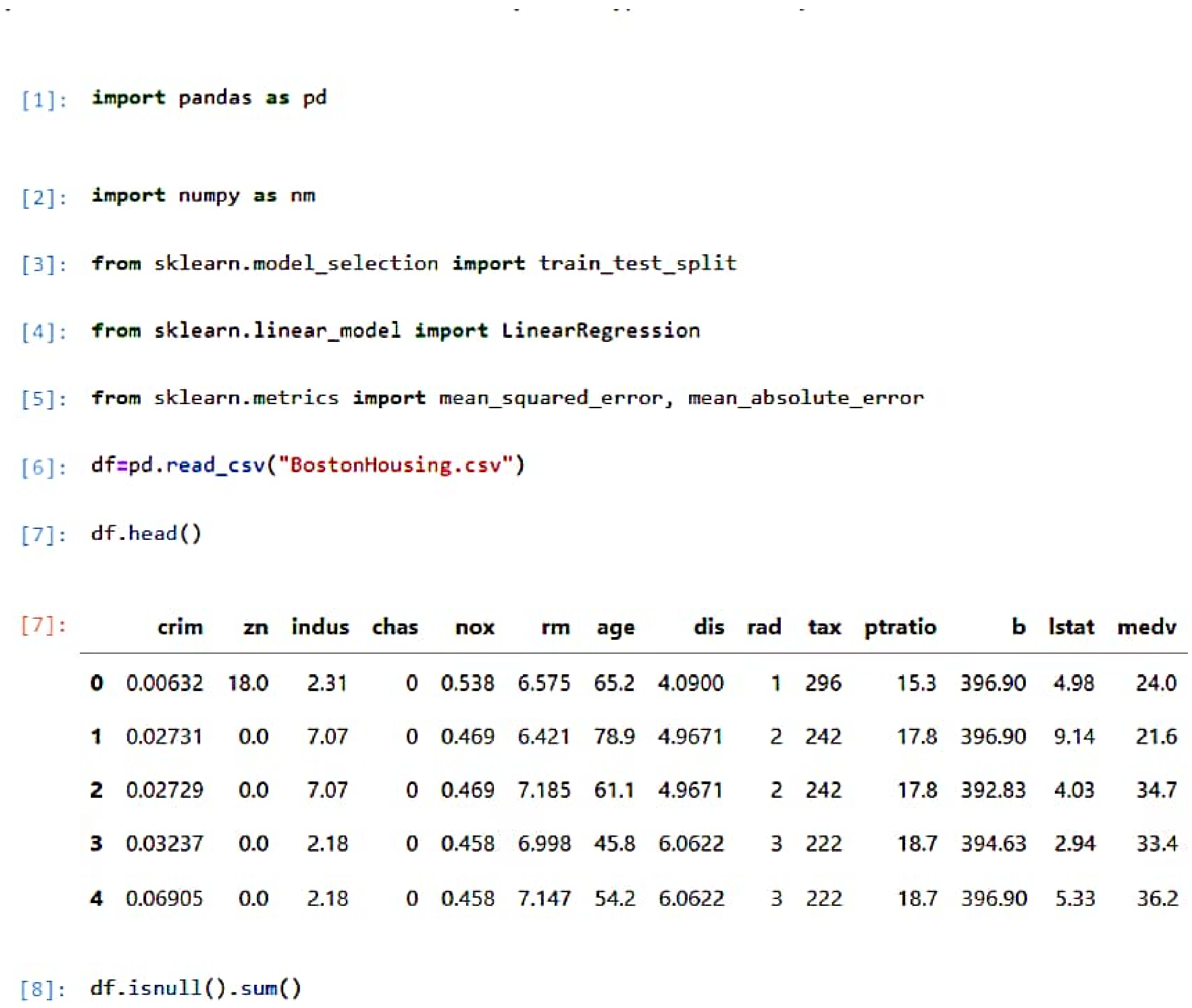
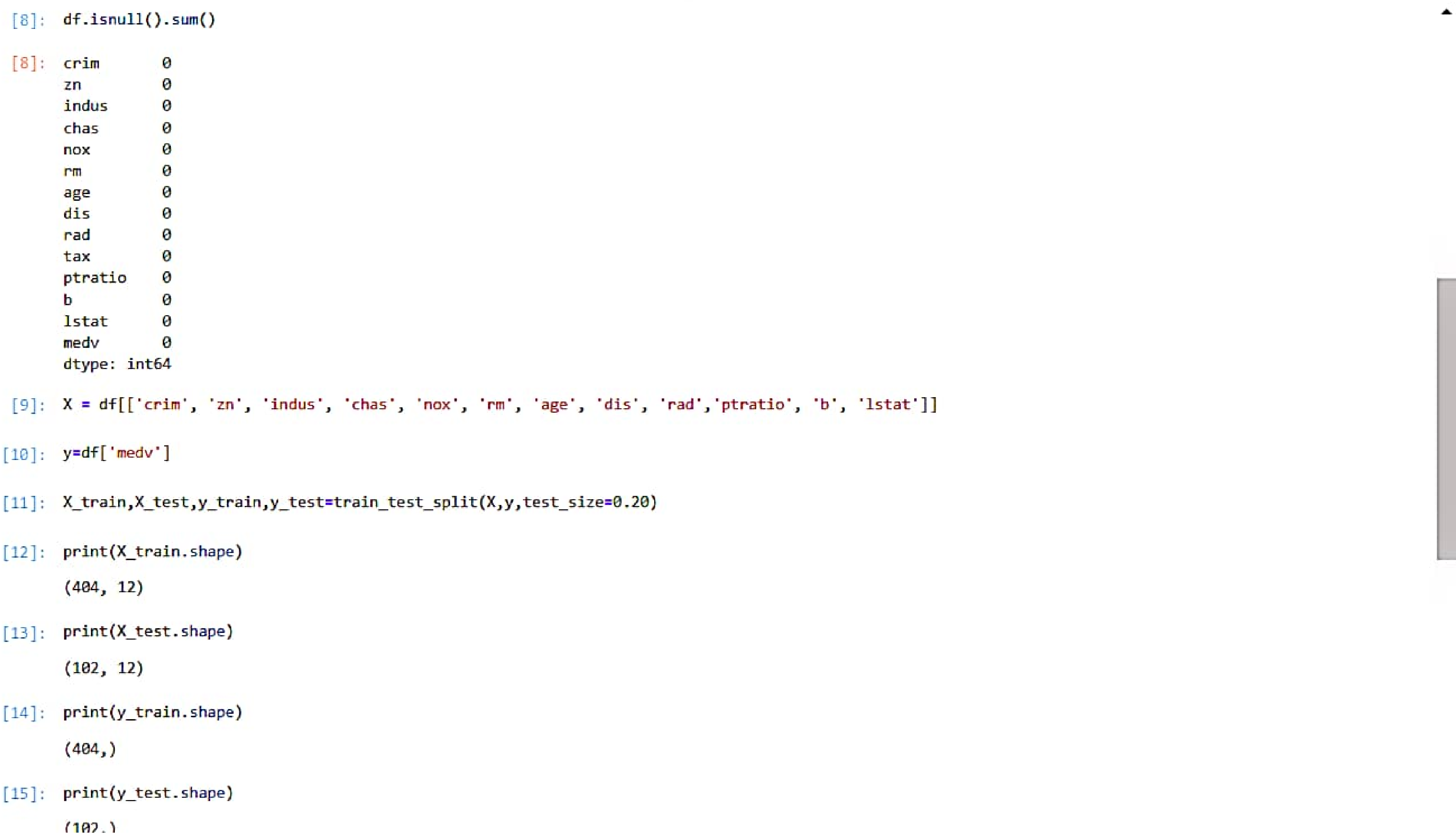


```
[1]: import pandas as pd

[2]: import numpy as nm

[3]: from sklearn.model_selection import train_test_split

[4]: from sklearn.linear_model import LinearRegression

[5]: from sklearn.metrics import mean_squared_error, mean_absolute_error

[6]: df=pd.read_csv("BostonHousing.csv")

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

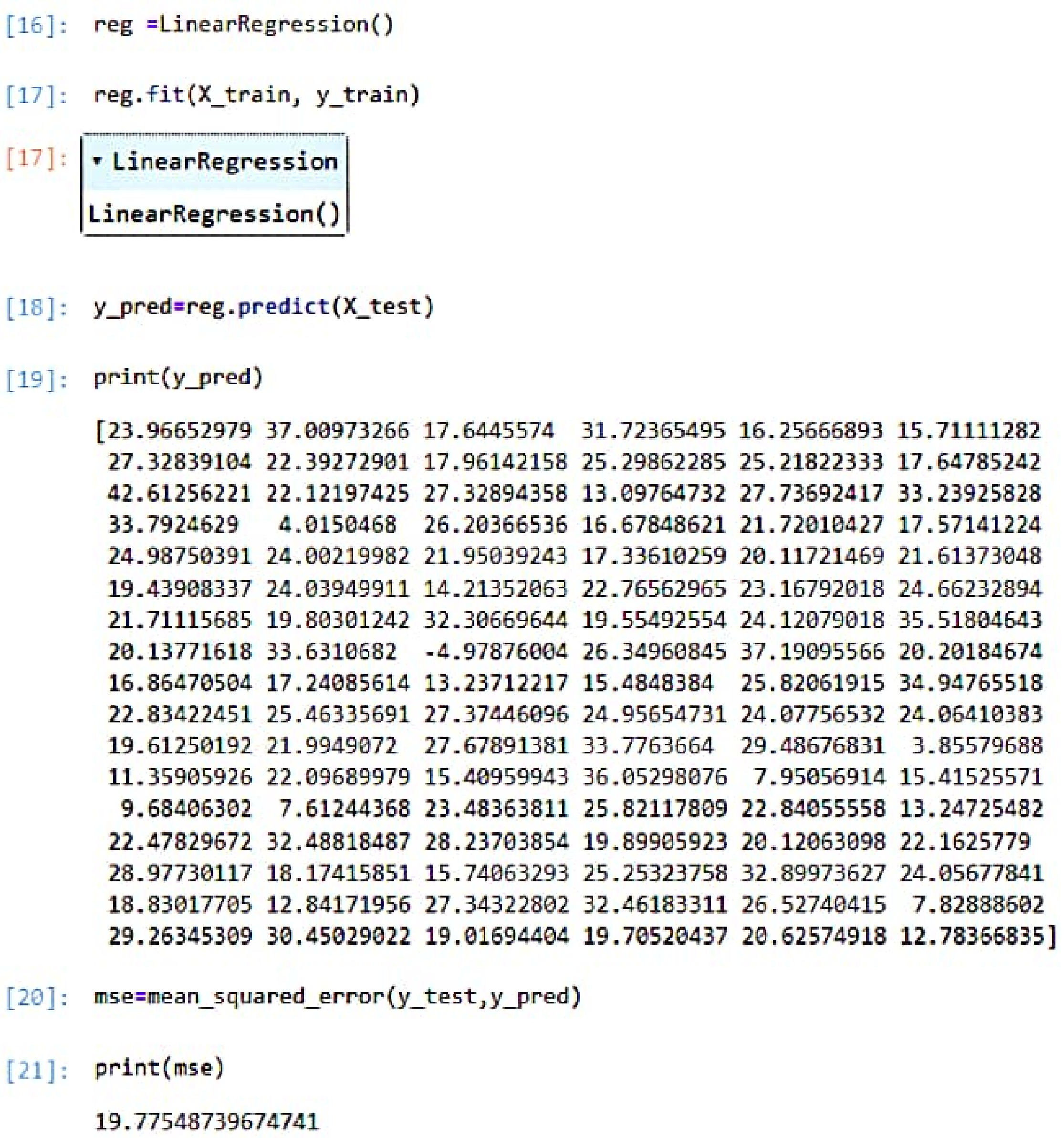
[7]:

	crim	zn	indus	chas	nox	rm	age	dis	rad	tax	ptratio	b	lstat	medv
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

```
[8]: df.isnull().sum()
```



```
[8]:  df.isnull().sum()

[8]:  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

[9]:  X = df[['crim', 'zn', 'indus', 'chas', 'nox', 'rm', 'age', 'dis', 'rad','ptratio', 'b', 'lstat']]

[10]: y=df['medv']

[11]: X_train,X_test,y_train,y_test=train_test_split(X,y,test_size=0.20)

[12]: print(X_train.shape)

      (404, 12)

[13]: print(X_test.shape)

      (102, 12)

[14]: print(y_train.shape)

      (404,)

[15]: print(y_test.shape)

      (102,)
```



```
[16]:  reg =LinearRegression()

[17]:  reg.fit(X_train, y_train)

[17]:  ▾ LinearRegression
       LinearRegression()

[18]:  y_pred=reg.predict(X_test)

[19]:  print(y_pred)

       [23.96652979 37.00973266 17.6445574  31.72365495 16.25666893 15.71111282
        27.32839104 22.39272901 17.96142158 25.29862285 25.21822333 17.64785242
        42.61256221 22.12197425 27.32894358 13.09764732 27.73692417 33.23925828
        33.7924629   4.0150468  26.20366536 16.67848621 21.72010427 17.57141224
        24.98750391 24.00219982 21.95039243 17.33610259 20.11721469 21.61373048
        19.43908337 24.03949911 14.21352063 22.76562965 23.16792018 24.66232894
        21.71115685 19.80301242 32.30669644 19.55492554 24.12079018 35.51804643
        20.13771618 33.6310682  -4.97876004 26.34960845 37.19095566 20.20184674
        16.86470504 17.24085614 13.23712217 15.4848384  25.82061915 34.94765518
        22.83422451 25.46335691 27.37446096 24.95654731 24.07756532 24.06410383
        19.61250192 21.9949072  27.67891381 33.7763664  29.48676831  3.85579688
        11.35905926 22.09689979 15.40959943 36.05298076  7.95056914 15.41525571
         9.68406302  7.61244368 23.48363811 25.82117809 22.84055558 13.24725482
        22.47829672 32.48818487 28.23703854 19.89905923 20.12063098 22.1625779
        28.97730117 18.17415851 15.74063293 25.25323758 32.89973627 24.05677841
        18.83017705 12.84171956 27.34322802 32.46183311 26.52740415  7.82888602
        29.26345309 30.45029022 19.01694404 19.70520437 20.62574918 12.78366835]

[20]:  mse=mean_squared_error(y_test,y_pred)

[21]:  print(mse)

       19.77548739674741
```