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


df=pd.read_csv('Boston.csv')
df.head(10)
```

	Unnamed: 0	crim	zn	indus	chas	nox	rm	age	dis	rad	tax	ptratio	black	lstat	m
0	1	0.00632	18.0	2.31	0	0.538	6.575	65.2	4.0900	1	296	15.3	396.90	4.98	2
1	2	0.02731	0.0	7.07	0	0.469	6.421	78.9	4.9671	2	242	17.8	396.90	9.14	2
2	3	0.02729	0.0	7.07	0	0.469	7.185	61.1	4.9671	2	242	17.8	392.83	4.03	3
3	4	0.03237	0.0	2.18	0	0.458	6.998	45.8	6.0622	3	222	18.7	394.63	2.94	3
4	5	0.06905	0.0	2.18	0	0.458	7.147	54.2	6.0622	3	222	18.7	396.90	5.33	3
5	6	0.02985	0.0	2.18	0	0.458	6.430	58.7	6.0622	3	222	18.7	394.12	5.21	2
6	7	0.08829	12.5	7.87	0	0.524	6.012	66.6	5.5605	5	311	15.2	395.60	12.43	2
7	8	0.14455	12.5	7.87	0	0.524	6.172	96.1	5.9505	5	311	15.2	396.90	19.15	2
8	9	0.21124	12.5	7.87	0	0.524	5.631	100.0	6.0821	5	311	15.2	386.63	29.93	1
9	10	0.17004	12.5	7.87	0	0.524	6.004	85.9	6.5921	5	311	15.2	386.71	17.10	1

```
df.drop(columns=['Unnamed: 15','Unnamed: 16'],inplace=True)
```

```
---------------------------------------------------------------------------
KeyError                                  Traceback (most recent call last)
<ipython-input-3-21a87a6ff3ee> in <cell line: 1>()
----> 1 df.drop(columns=['Unnamed: 15','Unnamed: 16'],inplace=True)

                                 5 frames
/usr/local/lib/python3.10/dist-packages/pandas/core/indexes/base.py in drop(self, labels, errors)
   6932         if mask.any():
   6933             if errors != "ignore":
-> 6934                 raise KeyError(f"{list(labels[mask])} not found in axis")
   6935             indexer = indexer[~mask]
   6936         return self.delete(indexer)

KeyError: "['Unnamed: 15', 'Unnamed: 16'] not found in axis"
```

SEARCH STACK OVERFLOW

```
df.drop(columns=['cat. mdev'],inplace=True)
```

```
---------------------------------------------------------------------------
KeyError                                  Traceback (most recent call last)
<ipython-input-7-bfe64fbe6df8> in <cell line: 1>()
----> 1 df.drop(columns=['cat. mdev'],inplace=True)

                                 5 frames
/usr/local/lib/python3.10/dist-packages/pandas/core/indexes/base.py in drop(self, labels, errors)
   6932         if mask.any():
   6933             if errors != "ignore":
-> 6934                 raise KeyError(f"{list(labels[mask])} not found in axis")
   6935             indexer = indexer[~mask]
   6936         return self.delete(indexer)

KeyError: "['cat. mdev'] not found in axis"
```

SEARCH STACK OVERFLOW

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

```
Unnamed: 0    0
crim          0
zn            0
indus         0
chas          0
nox           0
rm            0
age           0
```

```
dis           0
rad           0
tax           0
ptratio       0
black         0
lstat         0
medv          0
dtype: int64
```

```
df.info
```

```
<bound method DataFrame.info of      Unnamed: 0     crim    zn  indus  chas    nox     rm   age     dis  rad  \
0             1  0.00632  18.0   2.31     0  0.538  6.575  65.2  4.0900    1
1             2  0.02731   0.0   7.07     0  0.469  6.421  78.9  4.9671    2
2             3  0.02729   0.0   7.07     0  0.469  7.185  61.1  4.9671    2
3             4  0.03237   0.0   2.18     0  0.458  6.998  45.8  6.0622    3
4             5  0.06905   0.0   2.18     0  0.458  7.147  54.2  6.0622    3
..          ...      ...   ...    ...   ...    ...    ...   ...     ...  ...
501         502  0.06263   0.0  11.93     0  0.573  6.593  69.1  2.4786    1
502         503  0.04527   0.0  11.93     0  0.573  6.120  76.7  2.2875    1
503         504  0.06076   0.0  11.93     0  0.573  6.976  91.0  2.1675    1
504         505  0.10959   0.0  11.93     0  0.573  6.794  89.3  2.3889    1
505         506  0.04741   0.0  11.93     0  0.573  6.030  80.8  2.5050    1

     tax  ptratio   black  lstat  medv
0    296     15.3  396.90   4.98  24.0
1    242     17.8  396.90   9.14  21.6
2    242     17.8  392.83   4.03  34.7
3    222     18.7  394.63   2.94  33.4
4    222     18.7  396.90   5.33  36.2
..   ...      ...     ...    ...   ...
501  273     21.0  391.99   9.67  22.4
502  273     21.0  396.90   9.08  20.6
503  273     21.0  396.90   5.64  23.9
504  273     21.0  393.45   6.48  22.0
505  273     21.0  396.90   7.88  11.9

[506 rows x 15 columns]>
```

```
df.describe()
```

	zn	indus	chas	nox	rm	age	dis	rad	tax	p
0000	506.000000	506.000000	506.000000	506.000000	506.000000	506.000000	506.000000	506.000000	506.000000	506
3636	11.136779	0.069170	0.554695	6.284634	68.574901	3.795043	9.549407	408.237154	18	
2453	6.860353	0.253994	0.115878	0.702617	28.148861	2.105710	8.707259	168.537116	2	
0000	0.460000	0.000000	0.385000	3.561000	2.900000	1.129600	1.000000	187.000000	12	
0000	5.190000	0.000000	0.449000	5.885500	45.025000	2.100175	4.000000	279.000000	17	
0000	9.690000	0.000000	0.538000	6.208500	77.500000	3.207450	5.000000	330.000000	19	
0000	18.100000	0.000000	0.624000	6.623500	94.075000	5.188425	24.000000	666.000000	20	
0000	27.740000	1.000000	0.871000	8.780000	100.000000	12.126500	24.000000	711.000000	22	

```
df.corr()['medv'].sort_values()
```

```
lstat        -0.737663
ptratio      -0.507787
indus        -0.483725
tax          -0.468536
nox          -0.427321
crim         -0.388305
rad          -0.381626
age          -0.376955
Unnamed: 0   -0.226604
chas          0.175260
dis           0.249929
black         0.333461
zn            0.360445
rm            0.695360
medv          1.000000
Name: medv, dtype: float64
```

```
x=df.loc[:,['lstat','ptratio','rm']]
```

```
y=df.loc[:,['medv']]
x.shape,y.shape
```

```
((506, 3), (506, 1))
```

```
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test= train_test_split(x,y,test_size=0.25,random_state=10)
```

```
from sklearn.preprocessing import StandardScaler
```

```
scaler=StandardScaler()
```

```
scaler.fit(x_train)
```

```
StandardScaler
StandardScaler()
```

```
x_train=scaler.transform(x_train)
x_test=scaler.transform(x_test)
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/base.py:439: UserWarning: X does not have valid feature names, but StandardScaler
  warnings.warn(
```

```
from keras.models import Sequential
from keras.layers import Dense
```

```
model=Sequential()
```

```
model.add(Dense(128,input_shape=(3,),activation='relu',name='input'))
```

```
model.add(Dense(64,activation='relu',name='layer_1'))
```

```
model.add(Dense(1,activation='linear',name='output'))
```

```
model.compile(optimizer='adam',loss='mse',metrics=['mae'])
```

```
model.summary()
```

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 input (Dense)               (None, 128)               512

 layer_1 (Dense)             (None, 64)                8256

 output (Dense)              (None, 1)                 65

=================================================================
Total params: 8,833
Trainable params: 8,833
Non-trainable params: 0
_________________________________________________________________
```

```
model.fit(x_train,y_train,epochs=100,validation_split=0.05)
```

```
12/12 [==============================] - 0s 5ms/step - loss: 22.9162 - mae: 3.4304 - val_loss: 103.6793 - val_mae: 6.1071
Epoch 76/100
12/12 [==============================] - 0s 5ms/step - loss: 22.8394 - mae: 3.4263 - val_loss: 103.8781 - val_mae: 6.1151
Epoch 77/100
12/12 [==============================] - 0s 5ms/step - loss: 22.8000 - mae: 3.4608 - val_loss: 100.5202 - val_mae: 5.9332
Epoch 78/100
12/12 [==============================] - 0s 6ms/step - loss: 22.6695 - mae: 3.4470 - val_loss: 99.9743 - val_mae: 5.9010
Epoch 79/100
12/12 [==============================] - 0s 8ms/step - loss: 22.9583 - mae: 3.4546 - val_loss: 102.8521 - val_mae: 6.0587
Epoch 80/100
12/12 [==============================] - 0s 7ms/step - loss: 22.7592 - mae: 3.3581 - val_loss: 102.3310 - val_mae: 6.0257
Epoch 81/100
12/12 [==============================] - 0s 7ms/step - loss: 22.9553 - mae: 3.5196 - val_loss: 95.7791 - val_mae: 5.6427
Epoch 82/100
12/12 [==============================] - 0s 9ms/step - loss: 22.6123 - mae: 3.3604 - val_loss: 106.0574 - val_mae: 6.2260
Epoch 83/100
12/12 [==============================] - 0s 8ms/step - loss: 22.8430 - mae: 3.4676 - val_loss: 97.7250 - val_mae: 5.7647
Epoch 84/100
12/12 [==============================] - 0s 7ms/step - loss: 22.5640 - mae: 3.3866 - val_loss: 104.4883 - val_mae: 6.1470
Epoch 85/100
12/12 [==============================] - 0s 9ms/step - loss: 22.6047 - mae: 3.4208 - val_loss: 102.7465 - val_mae: 6.0605
Epoch 86/100
12/12 [==============================] - 0s 7ms/step - loss: 22.5795 - mae: 3.4452 - val_loss: 97.1084 - val_mae: 5.7365
Epoch 87/100
12/12 [==============================] - 0s 7ms/step - loss: 22.3302 - mae: 3.3870 - val_loss: 101.5904 - val_mae: 5.9975
Epoch 88/100
12/12 [==============================] - 0s 8ms/step - loss: 22.4942 - mae: 3.3957 - val_loss: 103.0592 - val_mae: 6.0786
Epoch 89/100
12/12 [==============================] - 0s 8ms/step - loss: 22.3668 - mae: 3.3965 - val_loss: 95.9972 - val_mae: 5.6756
Epoch 90/100
12/12 [==============================] - 0s 9ms/step - loss: 22.2884 - mae: 3.3401 - val_loss: 104.0749 - val_mae: 6.1359
Epoch 91/100
12/12 [==============================] - 0s 9ms/step - loss: 22.3638 - mae: 3.3967 - val_loss: 99.5180 - val_mae: 5.8919
Epoch 92/100
12/12 [==============================] - 0s 8ms/step - loss: 22.2430 - mae: 3.3583 - val_loss: 99.0132 - val_mae: 5.8628
Epoch 93/100
12/12 [==============================] - 0s 8ms/step - loss: 22.2950 - mae: 3.3528 - val_loss: 107.0681 - val_mae: 6.2876
Epoch 94/100
12/12 [==============================] - 0s 9ms/step - loss: 22.2542 - mae: 3.4541 - val_loss: 89.2288 - val_mae: 5.2566
Epoch 95/100
12/12 [==============================] - 0s 6ms/step - loss: 23.6318 - mae: 3.4615 - val_loss: 111.2662 - val_mae: 6.4834
Epoch 96/100
12/12 [==============================] - 0s 6ms/step - loss: 21.8952 - mae: 3.3746 - val_loss: 93.5801 - val_mae: 5.4945
Epoch 97/100
12/12 [==============================] - 0s 9ms/step - loss: 22.4717 - mae: 3.4544 - val_loss: 101.2379 - val_mae: 5.9701
Epoch 98/100
12/12 [==============================] - 0s 11ms/step - loss: 22.6548 - mae: 3.3946 - val_loss: 109.3008 - val_mae: 6.3823
Epoch 99/100
12/12 [==============================] - 0s 8ms/step - loss: 22.3242 - mae: 3.4857 - val_loss: 93.0246 - val_mae: 5.4955
Epoch 100/100
12/12 [==============================] - 0s 9ms/step - loss: 22.3740 - mae: 3.3742 - val_loss: 104.6419 - val_mae: 6.1621
<keras.callbacks.History at 0x7fe35012fe50>
```

```
output=model.evaluate(x_test,y_test)
```

```
4/4 [==============================] - 0s 6ms/step - loss: 339.4626 - mae: 17.5330
```

```
print(f"mean Sqaured Error:{output[0]}")
```

```
mean Sqaured Error:339.462646484375
```

```
print(f"mean Absolute Error :{output[1]}")
```

```
mean Absolute Error :17.532989501953125
```

```
y_pred=model.predict(x=x_test)
```

```
4/4 [==============================] - 0s 5ms/step
```

```
print(*zip(y_pred,y_test))
```

```
(array([6.6393113], dtype=float32), 'medv')
```

✓ 0s completed at 11:44