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

```
dataset=pd.read_csv('Boston-house-price-data.csv')
X=dataset.iloc[:,:-1].values
Y=dataset.iloc[:,-1].values
```

```
dataset.head()
```

	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	296.0	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	242.0	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	242.0	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	222.0	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	222.0	18.7	396.90	5.33	36.2

```
dataset.shape
```

```
(506, 14)
```

```
dataset.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 506 entries, 0 to 505
Data columns (total 14 columns):
 #   Column   Non-Null Count  Dtype  
---  ------   --------------  -----  
 0   CRIM     506 non-null    float64
 1   ZN       506 non-null    float64
 2   INDUS    506 non-null    float64
 3   CHAS     506 non-null    int64  
 4   NOX      506 non-null    float64
 5   RM       506 non-null    float64
 6   AGE      506 non-null    float64
 7   DIS      506 non-null    float64
 8   RAD      506 non-null    int64  
```

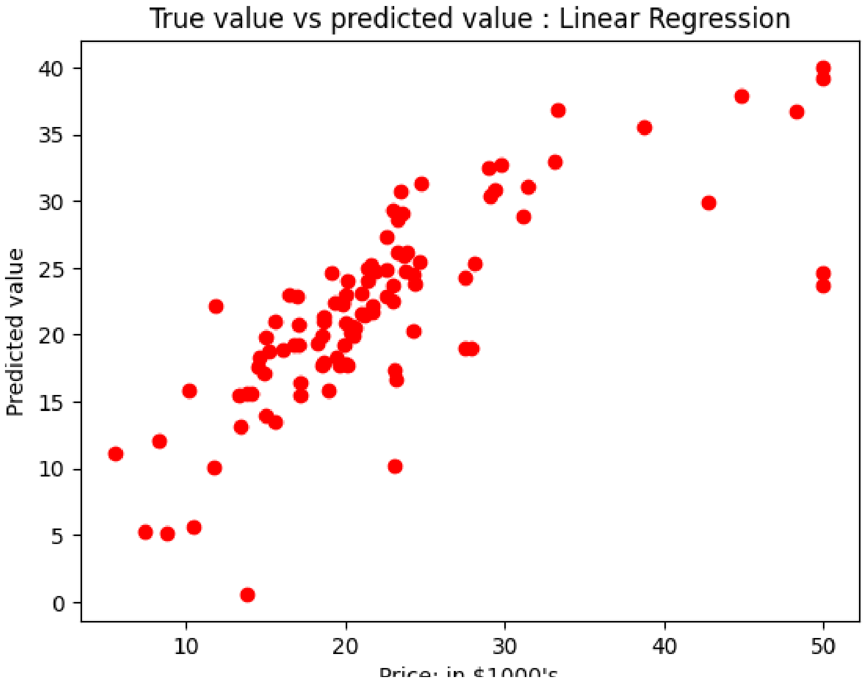
```
      9   TAX      506 non-null    float64
      10  PTRATIO  506 non-null    float64
      11  B        506 non-null    float64
      12  LSTAT    506 non-null    float64
      13  MEDV     506 non-null    float64
     dtypes: float64(12), int64(2)
     memory usage: 55.5 KB
```

```
x=dataset.iloc[:,:-1].values
y=dataset.iloc[:,-1].values

from sklearn.model_selection import train_test_split

xtrain, xtest, ytrain, ytest = train_test_split(x, y, test_size =0.2,random_state = 0)

print("xtrain shape : ", xtrain.shape)
print("xtest shape  : ", xtest.shape)
print("ytrain shape : ", ytrain.shape)
print("ytest shape  : ", ytest.shape)
```

```
     xtrain shape :  (404, 13)
     xtest shape  :  (102, 13)
     ytrain shape :  (404,)
     ytest shape  :  (102,)
```

```
from sklearn.linear_model import LinearRegression
regressor = LinearRegression()
regressor.fit(xtrain, ytrain)

y_pred = regressor.predict(xtest)
```

```
plt.scatter(ytest, y_pred, c = 'red')
plt.xlabel("Price: in $1000's")
plt.ylabel("Predicted value")
plt.title("True value vs predicted value : Linear Regression")
plt.show()
```

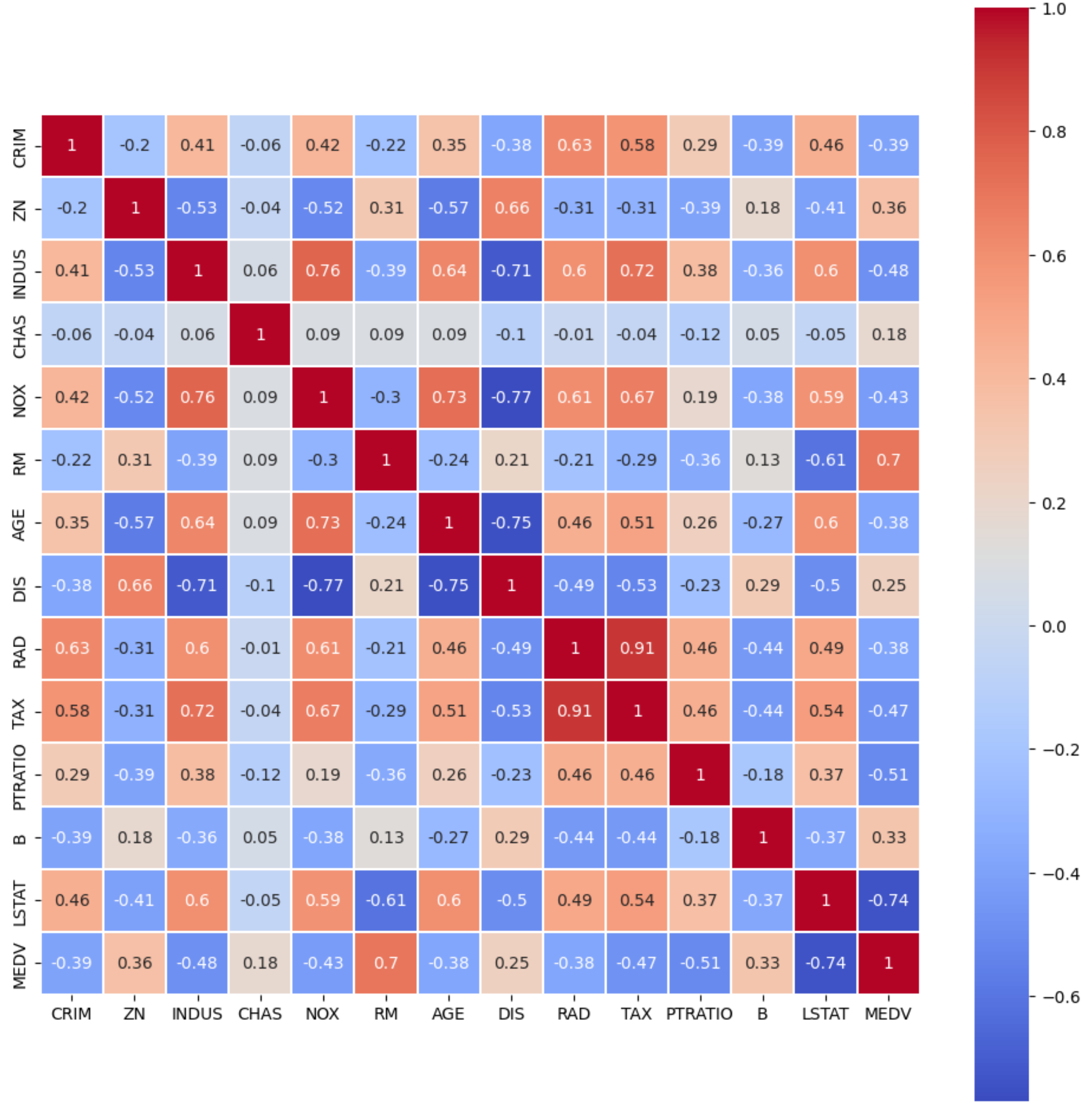


```
from sklearn.metrics import mean_squared_error, mean_absolute_error
mse = mean_squared_error(ytest, y_pred)
mae = mean_absolute_error(ytest,y_pred)
print("Mean Square Error : ", mse)
print("Mean Absolute Error : ", mae)
```

```
     Mean Square Error :  33.44897999767653
     Mean Absolute Error :  3.8429092204444966
```

```
import seaborn as sns
plt.figure(figsize=(12,12))
sns.heatmap(data=dataset.corr().round(2),annot=True,cmap='coolwarm',linewidths=0.2,square=True)
```

```
<Axes: >
```


CRIM ZN INDUS CHAS NOX RM AGE DIS RAD TAX PTRATIO B LSTAT MEDV
CRIM 1 -0.2 0.41 -0.06 0.42 -0.22 0.35 -0.38 0.63 0.58 0.29 -0.39 0.46 -0.39
ZN -0.2 1 -0.53 -0.04 -0.52 0.31 -0.57 0.66 -0.31 -0.31 -0.39 0.18 -0.41 0.36
INDUS 0.41 -0.53 1 0.06 0.76 -0.39 0.64 -0.71 0.6 0.72 0.38 -0.36 0.6 -0.48
CHAS -0.06 -0.04 0.06 1 0.09 0.09 0.09 -0.1 -0.01 -0.04 -0.12 0.05 -0.05 0.18
NOX 0.42 -0.52 0.76 0.09 1 -0.3 0.73 -0.77 0.61 0.67 0.19 -0.38 0.59 -0.43
RM -0.22 0.31 -0.39 0.09 -0.3 1 -0.24 0.21 -0.21 -0.29 -0.36 0.13 -0.61 0.7
AGE 0.35 -0.57 0.64 0.09 0.73 -0.24 1 -0.75 0.46 0.51 0.26 -0.27 0.6 -0.38
DIS -0.38 0.66 -0.71 -0.1 -0.77 0.21 -0.75 1 -0.49 -0.53 -0.23 0.29 -0.5 0.25
RAD 0.63 -0.31 0.6 -0.01 0.61 -0.21 0.46 -0.49 1 0.91 0.46 -0.44 0.49 -0.38
TAX 0.58 -0.31 0.72 -0.04 0.67 -0.29 0.51 -0.53 0.91 1 0.46 -0.44 0.54 -0.47
PTRATIO 0.29 -0.39 0.38 -0.12 0.19 -0.36 0.26 -0.23 0.46 0.46 1 -0.18 0.37 -0.51
B -0.39 0.18 -0.36 0.05 -0.38 0.13 -0.27 0.29 -0.44 -0.44 -0.18 1 -0.37 0.33
LSTAT 0.46 -0.41 0.6 -0.05 0.59 -0.61 0.6 -0.5 0.49 0.54 0.37 -0.37 1 -0.74
MEDV -0.39 0.36 -0.48 0.18 -0.43 0.7 -0.38 0.25 -0.38 -0.47 -0.51 0.33 -0.74 1
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4
−0.6

```
df1 = dataset[['RM','TAX','PTRATIO','LSTAT']]
df1.head()
```

	RM	TAX	PTRATIO	LSTAT
0	6.575	296.0	15.3	4.98
1	6.421	242.0	17.8	9.14
2	7.185	242.0	17.8	4.03
3	6.998	222.0	18.7	2.94
4	7.147	222.0	18.7	5.33

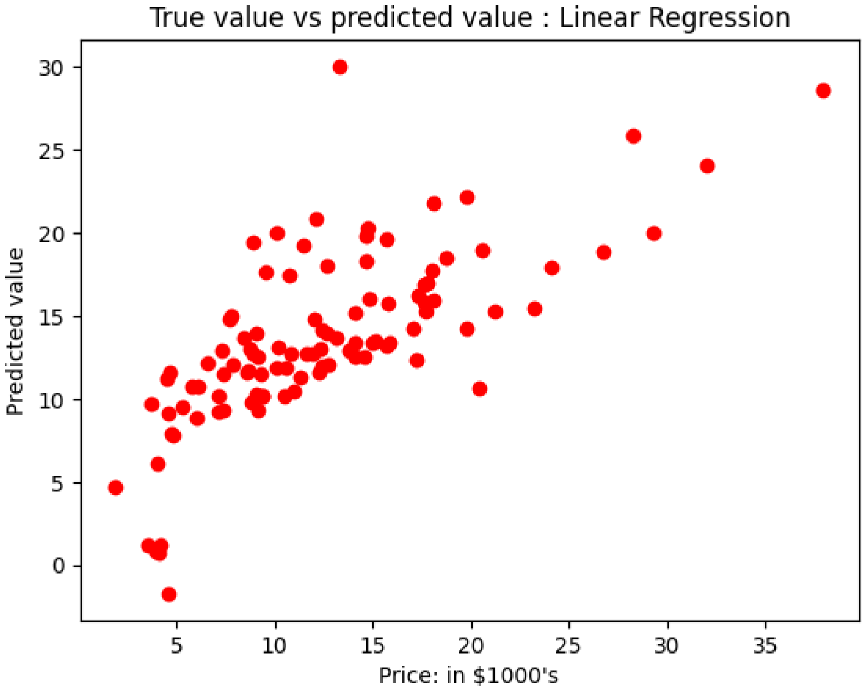
```
df1.shape
```

```
(506, 4)
```

```
x=df1.iloc[:,:-1].values
y=df1.iloc[:,-1].values

from sklearn.model_selection import train_test_split

xtrain, xtest, ytrain, ytest = train_test_split(x, y, test_size =0.2,random_state = 0)

print("xtrain shape : ", xtrain.shape)
print("xtest shape  : ", xtest.shape)
print("ytrain shape : ", ytrain.shape)
print("ytest shape  : ", ytest.shape)
```

```
xtrain shape :  (404, 3)
xtest shape  :  (102, 3)
ytrain shape :  (404,)
ytest shape  :  (102,)
```

```
from sklearn.linear_model import LinearRegression

regressor = LinearRegression()

regressor.fit(xtrain, ytrain)

y_pred = regressor.predict(xtest)


plt.scatter(ytest, y_pred, c = 'red')
plt.xlabel("Price: in $1000's")
plt.ylabel("Predicted value")
plt.title("True value vs predicted value : Linear Regression")
plt.show()
```



```
from sklearn.metrics import mean_squared_error, mean_absolute_error
mse = mean_squared_error(ytest, y_pred)
mae = mean_absolute_error(ytest,y_pred)
```

```
print("Mean Square Error : ", mse)
print("Mean Absolute Error : ", mae)
# ,'MEDV' removed end maI
```

```
     Mean Square Error :  21.714680825959494
     Mean Absolute Error :  3.616572667697949
```