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

In [2]:

```
from sklearn.datasets import load_boston
boston_dataset = load_boston()
```

In [3]:

```
boston_dataset.DESCR
```

```
'.. _boston_dataset:\n\nBoston house prices dataset\n---------------------------
\n\n**Data Set Characteristics:**  \n\n    :Number of Instances: 506 \n\n    :Num
ber of Attributes: 13 numeric/categorical predictive. Median Value (attribute 14)
is usually the target.\n\n    :Attribute Information (in order):\n        - CRIM
per capita crime rate by town\n        - ZN       proportion of residential land
zoned for lots over 25,000 sq.ft.\n        - INDUS    proportion of non-retail bu
siness acres per town\n        - CHAS     Charles River dummy variable (= 1 if tr
act bounds river; 0 otherwise)\n        - NOX      nitric oxides concentration (p
```

```
boston = pd.DataFrame(boston_dataset.data, columns=boston_dataset.feature_names)
boston.head()
```

	CRIM	ZN	INDUS	CHAS	NOX	RM	AGE	DIS	RAD	TAX	PTRATIO	B
0	0.00632	18.0	2.31	0.0	0.538	6.575	65.2	4.0900	1.0	296.0	15.3	396.90
1	0.02731	0.0	7.07	0.0	0.469	6.421	78.9	4.9671	2.0	242.0	17.8	396.90
2	0.02729	0.0	7.07	0.0	0.469	7.185	61.1	4.9671	2.0	242.0	17.8	392.83
3	0.03237	0.0	2.18	0.0	0.458	6.998	45.8	6.0622	3.0	222.0	18.7	394.63
4	0.06905	0.0	2.18	0.0	0.458	7.147	54.2	6.0622	3.0	222.0	18.7	396.90

```
boston['MEDV'] = boston_dataset.target
boston.corr().round(2)
```

	CRIM	ZN	INDUS	CHAS	NOX	RM	AGE	DIS	RAD	TAX	PTRATIO	
CRIM	1.00	-0.20	0.41	-0.06	0.42	-0.22	0.35	-0.38	0.63	0.58	0.29	-0
ZN	-0.20	1.00	-0.53	-0.04	-0.52	0.31	-0.57	0.66	-0.31	-0.31	-0.39	0
INDUS	0.41	-0.53	1.00	0.06	0.76	-0.39	0.64	-0.71	0.60	0.72	0.38	-0
CHAS	-0.06	-0.04	0.06	1.00	0.09	0.09	0.09	-0.10	-0.01	-0.04	-0.12	0
NOX	0.42	-0.52	0.76	0.09	1.00	-0.30	0.73	-0.77	0.61	0.67	0.19	-0
RM	-0.22	0.31	-0.39	0.09	-0.30	1.00	-0.24	0.21	-0.21	-0.29	-0.36	0
AGE	0.35	-0.57	0.64	0.09	0.73	-0.24	1.00	-0.75	0.46	0.51	0.26	-0
DIS	-0.38	0.66	-0.71	-0.10	-0.77	0.21	-0.75	1.00	-0.49	-0.53	-0.23	0
RAD	0.63	-0.31	0.60	-0.01	0.61	-0.21	0.46	-0.49	1.00	0.91	0.46	-0
B	0.39	0.18	0.36	0.05	0.38	0.13	0.27	0.29	0.44	0.44	0.18	1
MEDV	-0.39	0.36	-0.48	0.18	-0.43	0.70	-0.38	0.25	-0.38	-0.47	-0.51	0

```
x = boston[['RM','CRIM']]
y = boston['CHAS']
```

```
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)
```

```
from sklearn.linear_model import LinearRegression
model1 = LinearRegression()
model1.fit(x_train,y_train)
y_predict1 = model1.predict(x_test)
print(y_predict1)
```

```
[ 0.08575752  0.08086078  0.07769969  0.07719863  0.10839868  0.06747325
  0.06387316  0.08927898  0.06607958  0.03664805  0.0404299   0.09409896
  0.08482509  0.03821862  0.06428837  0.06699523  0.04649006  0.07374757
  0.11529088  0.12974778  0.07779783  0.12576772  0.04832456  0.07051065
 -0.03102016  0.05577446  0.14667332  0.03694424  0.07010686  0.10408592
  0.10201574  0.05594656  0.07403882  0.07390905  0.09122709  0.07028836
  0.06526836  0.05336953  0.12094477  0.05840215  0.05160834  0.03914058
  0.06026269  0.06442975  0.10873205  0.06101606  0.06355081  0.0800493
  0.14895238  0.07778782  0.0193631   0.06110434  0.04388434  0.09426225
  0.0753434   0.09213827  0.0418488   0.06554942  0.06891212  0.08257058
  0.11707205  0.0510933   0.03503349  0.07120158  0.04752699  0.03322513
  0.07280822  0.10258403  0.07187611  0.07811144  0.06876596  0.07987129
  0.06500802 -0.009518    0.05842426  0.0627456   0.08533715  0.09694721
  0.07159104  0.05458854  0.07809617  0.05705614  0.09788991  0.06949055
  0.07936319  0.02937969  0.10411059  0.04105772  0.09342061  0.09154868
  0.0668481   0.05170299  0.07595712  0.08507912  0.06710871  0.09422403
  0.07160889  0.06084648  0.06334713  0.05688402  0.1284948   0.08285505
  0.0908281   0.04254786 -0.04535859  0.02528951  0.05824596  0.04427793
  0.0597257   0.09901987  0.06527028  0.01943936  0.05994516  0.05078728
  0.14521806  0.04006098 -0.02606613  0.06318421  0.01385115  0.04668944
  0.11384881  0.08424188  0.08346007  0.05774677  0.08061655  0.03838416
  0.05821195]
```

```
import numpy as np
from sklearn.metrics import mean_squared_error,r2_score
print("The RMSE is: ",np.sqrt(mean_squared_error(y_test,y_predict1)))
print("The R2 score is: ",r2_score(y_test,y_predict1))
```

```
    The RMSE is:  0.2567582272592853
    The R2 score is:  -0.0012249470833525233
```

```
from sklearn.ensemble import RandomForestRegressor
from sklearn.datasets import make_regression
X, y = make_regression(n_features=4, n_informative=2,
                       random_state=0, shuffle=False)
regr = RandomForestRegressor(max_depth=2, random_state=0)
regr.fit(X, y)
print(regr.predict([[0, 0, 0, 0]]))
```

```
    [-8.32987858]
```

✓ 0s completed at 11:40 AM ● ✕