In [1]:

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

In [2]:

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

Out[2]:

	CRIM	ZN	INDUS	CHAS	NOX	RM	AGE	DIS	RAD	TAX	PTRATIO	B	LSTAT
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	4.98
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	9.14
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	4.03
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	2.94
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	5.33

In [5]:

```
from sklearn.model_selection import train_test_split
```

In [4]:

```
x = boston.drop('LSTAT',1)
y = boston['LSTAT']
```

In [7]:

```
x_train,x_test,y_train,y_test  = train_test_split(x,y)
```

In [8]:

```
from sklearn.linear_model import LinearRegression
model  = LinearRegression()
```

In [9]:

```
model.fit(x_train,y_train)
```

Out[9]:

```
LinearRegression()
```

In [10]:

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

In [14]:

```
from sklearn.metrics import r2_score,mean_squared_error
```

In [13]:

```
r2_score(y_pred,y_test)
```

Out[13]:

```
0.4133945586344202
```

In [16]:

```
j = mean_squared_error(y_pred,y_test)
```

In [19]:

```
import numpy as np
j/np.mean(y_test) * 100
```

Out[19]:

```
114.64320722496011
```

In [ ]:

```

```