In [1]:
```
from sklearn.datasets import load_boston  # Import dataset module
boston_dataset = load_boston()
```

In [2]:
```
import matplotlib.pyplot as plt

import pandas as pd
import seaborn as sns
```

Mean Squared Error

MSE = 1/n*Sum(y_pred-y_test)^2

In [3]:
```
from sklearn.datasets import load_boston  # Import dataset module
boston_dataset = load_boston()
```

In [4]:
```
import pandas as pd
import numpy as np

import matplotlib.pyplot as plt
import seaborn as sns
```

In [5]:
```
import pandas as pd
boston = pd.DataFrame(boston_dataset.data, columns = boston_dataset.feature
boston.head()
```

Out[5]:

	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 [6]:
```
boston['MEDV'] = boston_dataset.target
```

In [7]:
```
boston.describe()
```

Out[7]:

	CRIM	ZN	INDUS	CHAS	NOX	RM	AGE	
count	506.000000	506.000000	506.000000	506.000000	506.000000	506.000000	506.000000	506.00
mean	3.613524	11.363636	11.136779	0.069170	0.554695	6.284634	68.574901	3.79
std	8.601545	23.322453	6.860353	0.253994	0.115878	0.702617	28.148861	2.10
min	0.006320	0.000000	0.460000	0.000000	0.385000	3.561000	2.900000	1.12
25%	0.082045	0.000000	5.190000	0.000000	0.449000	5.885500	45.025000	2.10
50%	0.256510	0.000000	9.690000	0.000000	0.538000	6.208500	77.500000	3.20
75%	3.677083	12.500000	18.100000	0.000000	0.624000	6.623500	94.075000	5.18
max	88.976200	100.000000	27.740000	1.000000	0.871000	8.780000	100.000000	12.12

In [8]:
```
x,y = boston.drop('MEDV', axis=1),boston['MEDV']
```

In [12]:
```
from sklearn.neighbors import KNeighborsRegressor
model = KNeighborsRegressor()
```

In [13]:
```
model.fit(x,y)
```

Out[13]:
```
KNeighborsRegressor()
```

In [14]:
```
y_pred = model.predict(x)
```

In [15]:
```
from sklearn.metrics import r2_score,mean_squared_error,mean_absolute_error
```

In [16]:
```
mean_squared_error(y_pred,y)
```

Out[16]:
```
23.966862450592888
```

Mean Absolute Error

MSE = 1/n*Sum(|y_pred-y_test|)

In [17]:
```
mean_absolute_error(y_pred,y)
```

Out[17]:
```
3.3752173913043477
```

Root Mean Squared Error

MSE = 1/n*(Sum(y_pred-y_test)^2)^1/2

In [18]:
```
mean_squared_error(y_pred,y,squared = False)
```

Out[18]:
```
4.8955962303475244
```

R2 Score

In [19]:
```
r2_score(y_pred,y)
```

Out[19]:
```
0.5884292317985158
```

In [ ]:
```

```