Machine_Learning_Decision_Tree_Regression_And_Cross_Validation

In [83]:

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

Boston House Pricing Dataset

In [49]:

```
data_url = "http://lib.stat.cmu.edu/datasets/boston"
raw_df = pd.read_csv(data_url, sep="\s+", skiprows=22, header=None)
data = np.hstack([raw_df.values[::2, :], raw_df.values[1::2, :2]])
target = raw_df.values[1::2, 2]
```

In [50]:

```
boston_df
```

Out[50]:

	CRIM	ZN	INDUS	CHAS	NOX	RM	AGE	DIS	RAD	TAX	PTRATIO	B
0	0.00632	18.0	2.31	0	0.538	6.575	65.2	4.0900	1	296.0	15.3	396.90
1	0.02731	0.0	7.07	0	0.469	6.421	78.9	4.9671	2	242.0	17.8	396.90
2	0.02729	0.0	7.07	0	0.469	7.185	61.1	4.9671	2	242.0	17.8	392.83
3	0.03237	0.0	2.18	0	0.458	6.998	45.8	6.0622	3	222.0	18.7	394.63
4	0.06905	0.0	2.18	0	0.458	7.147	54.2	6.0622	3	222.0	18.7	396.90
...	...	...	...	...	...	...	...	...	...	...	...	...
501	0.06263	0.0	11.93	0	0.573	6.593	69.1	2.4786	1	273.0	21.0	391.99
502	0.04527	0.0	11.93	0	0.573	6.120	76.7	2.2875	1	273.0	21.0	396.90
503	0.06076	0.0	11.93	0	0.573	6.976	91.0	2.1675	1	273.0	21.0	396.90
504	0.10959	0.0	11.93	0	0.573	6.794	89.3	2.3889	1	273.0	21.0	393.45
505	0.04741	0.0	11.93	0	0.573	6.030	80.8	2.5050	1	273.0	21.0	396.90

506 rows × 14 columns

Independent Features

In [53]:

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

Dependent Features

In [55]:

```
y = boston.target
```

In [56]:

```
X.head()
```

Out[56]:

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

Train_Test_Split

In [57]:

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.33, random_state
```

In [58]:

```
from sklearn.tree import DecisionTreeRegressor
regressor = DecisionTreeRegressor()
```

In [59]:

```
regressor.fit(X_train,y_train)
```

Out[59]:

```
▸ DecisionTreeRegressor
```

In [60]:

```
y_pred = regressor.predict(X_test)
```

In [61]:

```
y_pred
```

Out[61]:

```
array([22. , 27.9,  8.1, 24.7, 15.2, 21.6, 18.8, 17.8, 21.4, 20.1, 19. ,
       19. ,  6.3, 21.1, 16.2, 22. , 20.5, 10.5, 43.1, 14.6, 24.1, 24.6,
       13.6, 20.6, 16.8, 14.6, 22. , 13.4, 19. , 22.7, 19.8, 22.9, 36.5,
       15.3, 17.3, 13.8, 31.2, 18.7, 21.7, 24.7, 19.4, 37.2, 35.2, 19.9,
       22. , 11. , 13.6, 24.7, 27.1, 24.5, 19.1, 35.1, 13.6, 29.4, 43.1,
       20.6, 17.8, 37.3, 22.9, 22.5, 27.5, 29. , 30.1, 18.2, 29.8, 14.4,
       12.1, 22.9, 32.5, 17.3, 22.6, 22.8,  8.4, 18.6, 20.6,  5.6, 19.8,
       35.2, 10.2, 13.1, 22. , 16.3, 17.5, 10.5, 20.3, 25.1, 15.2, 23. ,
       23. , 18. , 22.2,  7.2, 19.8, 17.5, 18.6, 19.8, 50. , 16.3, 11.8,
       16.3, 17.5, 21.2, 14.6, 20.4, 23.7, 11.7, 20.4, 24.8, 19. , 22.2,
        8.4, 16.3, 22.3, 21.4, 31.7, 16.7, 50. , 14.3, 16.2, 23.7, 17.1,
       24.7,  8.3, 18.5, 24.7, 22.9, 23.3, 37.2, 17.5, 46. , 15.4, 25. ,
       18.2, 27.1, 11.8, 21.7, 19.8, 29.6, 24.5, 14.3, 21.7, 23.5, 19.6,
       14.4,  6.3, 20.1, 13.8, 14.9, 15.6, 44.8, 14.1, 17.8, 23. , 18.5,
       16.2, 18.6, 14.6, 22.9, 36.2,  8.3, 21.4, 19.9, 20.4, 22.9, 17.1,
       22.8, 41.7])
```

In [62]:

```
from sklearn.metrics import r2_score
score = r2_score(y_pred, y_test)
```

In [63]:

```
score
```

Out[63]:

```
0.729288261788323
```

Hyperparameter Tunning

In [65]:

```
parameter={
 'criterion':['squared_error','friedman_mse','absolute_error','poisson'],
  'splitter':['best','random'],
  'max_depth':[1,2,3,4,5,6,7,8,10,11,12],
  'max_features':['auto', 'sqrt', 'log2']

}
regressor=DecisionTreeRegressor()
```

https://scikit-learn.org/stable/modules/model_evaluation.html (https://scikit-learn.org/stable/modules/model_evaluation.html)

In [77]:

```
from sklearn.model_selection import GridSearchCV
regressor_cv = GridSearchCV(regressor, param_grid=parameter, cv=5, scoring='neg_mean_squa
```

In [78]:

```
regressor_cv.fit(X_train, y_train)
```

```
  # reshape is necessary to preserve the data contiguity against vs
C:\Users\baps\anaconda3\lib\site-packages\sklearn\tree\_classes.py:277: F
utureWarning: `max_features='auto'` has been deprecated in 1.1 and will b
e removed in 1.3. To keep the past behaviour, explicitly set `max_feature
s=1.0'`.
  # reshape is necessary to preserve the data contiguity against vs
C:\Users\baps\anaconda3\lib\site-packages\sklearn\tree\_classes.py:277: F
utureWarning: `max_features='auto'` has been deprecated in 1.1 and will b
e removed in 1.3. To keep the past behaviour, explicitly set `max_feature
s=1.0'`.
  # reshape is necessary to preserve the data contiguity against vs
C:\Users\baps\anaconda3\lib\site-packages\sklearn\tree\_classes.py:277: F
utureWarning: `max_features='auto'` has been deprecated in 1.1 and will b
e removed in 1.3. To keep the past behaviour, explicitly set `max_feature
s=1.0'`.
  # reshape is necessary to preserve the data contiguity against vs
C:\Users\baps\anaconda3\lib\site-packages\sklearn\tree\_classes.py:277: F
utureWarning: `max_features='auto'` has been deprecated in 1.1 and will b
e removed in 1.3. To keep the past behaviour, explicitly set `max_feature
s=1.0'`.
```

In [80]:

```
regressor_cv.best_params_
```

Out[80]:

```
{'criterion': 'squared_error',
 'max_depth': 11,
 'max_features': 'auto',
 'splitter': 'random'}
```

In [81]:

```
y_pred = regressor_cv.predict(X_test)
```

In [82]:

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

Out[82]:

```
0.6943237659674601
```

In [ ]: