

ML_5_Assignment

March 18, 2019

```
In [1]: """In this assignment students will build the random forest model after normalizing the  
variable to house pricing from boston data set."""
```

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn import datasets
boston = datasets.load_boston()
features = pd.DataFrame(boston.data, columns = boston.feature_names)
targets = boston.target
features.head(5)
```

```
Out[1]:
```

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

	PTRATIO	B	LSTAT
0	15.3	396.90	4.98
1	17.8	396.90	9.14
2	17.8	392.83	4.03
3	18.7	394.63	2.94
4	18.7	396.90	5.33

```
In [2]: # The target column contains the prices which we are going to predict.  
# Lets add the target data to the dataframe as a new column = 'Price'  
features['Price'] = boston.target  
features.head(5)
```

```
Out[2]:
```

	CRIM	ZN	INDUS	CHAS	NOX	RM	AGE	DIS	RAD	TAX	\
0	0.00632	18.0	2.31	0.0	0.538	6.575	65.2	4.0900	1.0	296.0	
1	0.02731	0.0	7.07	0.0	0.469	6.421	78.9	4.9671	2.0	242.0	
2	0.02729	0.0	7.07	0.0	0.469	7.185	61.1	4.9671	2.0	242.0	
3	0.03237	0.0	2.18	0.0	0.458	6.998	45.8	6.0622	3.0	222.0	

```
4  0.06905   0.0   2.18   0.0  0.458  7.147  54.2  6.0622  3.0  222.0
```

	PTRATIO	B	LSTAT	Price
0	15.3	396.90	4.98	24.0
1	17.8	396.90	9.14	21.6
2	17.8	392.83	4.03	34.7
3	18.7	394.63	2.94	33.4
4	18.7	396.90	5.33	36.2

```
In [3]: # Splitting into train and test parts
```

```
X = features.drop(['Price'],axis = 1).values
y = features['Price'].values
X_train,X_test,y_train,y_test = train_test_split(X,y,test_size = 0.30,random_state = 4)
```

```
In [4]: # Apply standard Scaling
```

```
from sklearn.preprocessing import StandardScaler
sc_X = StandardScaler()
X_train = sc_X.fit_transform(X_train)
X_test = sc_X.fit_transform(X_test)
```

```
In [5]: # Applying the model
```

```
from sklearn.ensemble import RandomForestRegressor
regressor = RandomForestRegressor()
regressor.fit(X_train,y_train)
y_pred = regressor.predict(X_test)
```

```
C:\ProgramData\Anaconda3\lib\site-packages\sklearn\ensemble\forest.py:246: FutureWarning: The number of trees hyperparameter is deprecated in favor of n_estimators (from 10 in version 0.20 to 100 in 0.22.", FutureWarning)
```

```
In [6]: # Getting the r2 score
```

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

```
Out[6]: 0.8182031508239296
```