Loading Random Forest Regressor Model

In [1]:
```
from joblib import load
model = load("RANDOM_FOREST_REGRESSOR.joblib")
```

Giving Feature Information (To help user in INPUT)

In [2]:
```
print("Feature Information:")
print("")
print("1.  CRIM      per capita crime rate by town")
print("2.  ZN        proportion of residential land zoned for lots over 25,000 sc
print("3.  INDUS     proportion of non-retail business acres per town")
print("4.  CHAS      Charles River dummy variable (= 1 if tract bounds river; 0 c
print("5.  NOX       nitric oxides concentration (parts per 10 million)")
print("6.  RM        average number of rooms per dwelling")
print("7.  AGE       proportion of owner-occupied units built prior to 1940")
print("8.  DIS       weighted distances to five Boston employment centres")
print("9.  RAD       index of accessibility to radial highways")
print("10. TAX       full-value property-tax rate per $10,000")
print("11. PTRATIO   pupil-teacher ratio by town")
print("12. B         1000(Bk - 0.63)^2 where Bk is the proportion of blacks by tc
print("13. LSTAT     % lower status of the population")

print("")
print("Label Information: ")
print("")
print("14. MEDV     Median value of owner-occupied homes in $1000's")
```

```
Feature Information:

1.  CRIM      per capita crime rate by town
2.  ZN        proportion of residential land zoned for lots over 25,000 sq.ft.
3.  INDUS     proportion of non-retail business acres per town
4.  CHAS      Charles River dummy variable (= 1 if tract bounds river; 0 otherw
ise)
5.  NOX       nitric oxides concentration (parts per 10 million)
6.  RM        average number of rooms per dwelling
7.  AGE       proportion of owner-occupied units built prior to 1940
8.  DIS       weighted distances to five Boston employment centres
9.  RAD       index of accessibility to radial highways
10. TAX       full-value property-tax rate per $10,000
11. PTRATIO   pupil-teacher ratio by town
12. B         1000(Bk - 0.63)^2 where Bk is the proportion of blacks by town
13. LSTAT     % lower status of the population

Label Information:

14. MEDV     Median value of owner-occupied homes in $1000's
```

Taking input of Features of a house to predict its price

Here we will take input as a list of 13 features. It must be converted into a DDL(Double Dimensional List) because .predict() function takes DDL.

In [3]:
```
temp = eval(input("Enter a list of 13 features of a house to predict it's price:
predicted_label = model.predict([temp])
print("")
print("")
print("")
print("")
print("The Price of house is : ", "$", predicted_label[0]*1000)
print("")
print("")
print("")
print("")
```

```
Enter a list of 13 features of a house to predict it's price: [0.06905,0,2.18,
0,0.458,7.147,54.2,6.0622,3,222,18.7,396.9,5.33]




The Price of house is :  $ 35242.99999999996




C:\Users\prath\Desktop\Coding\Machine Learning\learning_ml\lib\site-packages\sk
learn\base.py:450: UserWarning: X does not have valid feature names, but Random
ForestRegressor was fitted with feature names
  warnings.warn(
```