In [1]:
```
#Aim: Hyperparameter tuning for lasso regression can be done in python without using lassoCV API
```

In [1]:
```
import pandas as pd
df=pd.read_csv("Boston_Housing.csv")
df.head(3)
```

Out[1]:

| | CRIM | ZN | INDUS | CHAS | NOX | RM | AGE | DIS | RAD | TAX | PTRATIO | B | LSTAT | MEDV |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 0.00632 | 18.0 | 2.31 | 0 | 0.538 | 6.575 | 65.2 | 4.0900 | 1 | 296 | 15.3 | 396.90 | 4.98 | 24.0 |
| **1** | 0.02731 | 0.0 | 7.07 | 0 | 0.469 | 6.421 | 78.9 | 4.9671 | 2 | 242 | 17.8 | 396.90 | 9.14 | 21.6 |
| **2** | 0.02729 | 0.0 | 7.07 | 0 | 0.469 | 7.185 | 61.1 | 4.9671 | 2 | 242 | 17.8 | 392.83 | 4.03 | 34.7 |

In [2]:
```
x= df.iloc[:,:-1]
x.shape
```

Out[2]:
```
(506, 13)
```

In [3]:
```
y = df.iloc[:,-1]
y
```

Out[3]:
```
0      24.0
1      21.6
2      34.7
3      33.4
4      36.2
       ...
501    22.4
502    20.6
503    23.9
504    22.0
505    11.9
Name: MEDV, Length: 506, dtype: float64
```

In [4]:
```
from sklearn.linear_model import Lasso
model = Lasso()
```

In [5]:
```
from sklearn.model_selection import train_test_split
xtrain,xtest,ytrain,ytest = train_test_split(x,y,test_size=0.25, random_state=1)
model.fit(xtrain,ytrain)
```

Out[5]:
```
Lasso()
```

In [6]:
```
from sklearn.model_selection import RepeatedKFold
cv = RepeatedKFold(n_splits = 10, n_repeats = 3, random_state = 1)
```

In [7]:
```
from sklearn.metrics import r2_score
ypred = model.predict(xtest)
r2_score(ytest, ypred)
```

Out[7]:
```
0.6621980770523261
```

In [8]:
```
from sklearn.preprocessing import StandardScaler
sc= StandardScaler()
x_sc = sc.fit_transform(x)
xtrain,xtest,ytrain,ytest  = train_test_split(x_sc, y, test_size=0.25, random_state=1)
model1= Lasso()
parms = {'alpha':[0.00001,0.0001,0.001,0.01]}
from sklearn.model_selection import GridSearchCV
search = GridSearchCV(model1, parms, cv=cv)
result = search.fit(x_sc, y)
result.best_params_
```

Out[8]:
```
{'alpha': 0.01}
```

In [9]:
```
model2 = Lasso(alpha=0.01)
model2.fit(xtrain, ytrain)
```

Out[9]:
```
Lasso(alpha=0.01)
```

In [10]:
```
ypred2 = model2.predict(xtest)
r2_score(ytest,ypred2)
```

Out[10]:
```
0.7787372388293925
```

In [ ]:

In [ ]: