In [1]:
```
#Practical no.9
#Analyzing how penalty impact the feature in Lasso regression
import sklearn
import pandas as pd
df=pd.read_csv("BostonHousing.csv")
df
```

Out[1]:

	crim	zn	indus	chas	nox	rm	age	dis	rad	tax	ptratio	b	lstat	med
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.
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.
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.
3	0.03237	0.0	2.18	0	0.458	6.998	45.8	6.0622	3	222	18.7	394.63	2.94	33.
4	0.06905	0.0	2.18	0	0.458	7.147	54.2	6.0622	3	222	18.7	396.90	5.33	36.
...	...	...	...	...	...	...	...	...	...	...	...	...	...	.
501	0.06263	0.0	11.93	0	0.573	6.593	69.1	2.4786	1	273	21.0	391.99	9.67	22.
502	0.04527	0.0	11.93	0	0.573	6.120	76.7	2.2875	1	273	21.0	396.90	9.08	20.
503	0.06076	0.0	11.93	0	0.573	6.976	91.0	2.1675	1	273	21.0	396.90	5.64	23.
504	0.10959	0.0	11.93	0	0.573	6.794	89.3	2.3889	1	273	21.0	393.45	6.48	22.
505	0.04741	0.0	11.93	0	0.573	6.030	80.8	2.5050	1	273	21.0	396.90	7.88	11.

506 rows × 14 columns

In [2]:
```
df.head(3)
```

Out[2]:

	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 [3]:
```
df.sample(3)
```

Out[3]:

	crim	zn	indus	chas	nox	rm	age	dis	rad	tax	ptratio	b	lstat	me
330	0.04544	0.0	3.24	0	0.460	6.144	32.2	5.8736	4	430	16.9	368.57	9.09	1(
241	0.10612	30.0	4.93	0	0.428	6.095	65.1	6.3361	6	300	16.6	394.62	12.40	2(
264	0.55007	20.0	3.97	0	0.647	7.206	91.6	1.9301	5	264	13.0	387.89	8.10	3(

In [4]:
```
x=df.iloc[:,:-1]
x
```

Out[4]:

	crim	zn	indus	chas	nox	rm	age	dis	rad	tax	ptratio	b	lstat
0	0.00632	18.0	2.31	0	0.538	6.575	65.2	4.0900	1	296	15.3	396.90	4.98
1	0.02731	0.0	7.07	0	0.469	6.421	78.9	4.9671	2	242	17.8	396.90	9.14
2	0.02729	0.0	7.07	0	0.469	7.185	61.1	4.9671	2	242	17.8	392.83	4.03
3	0.03237	0.0	2.18	0	0.458	6.998	45.8	6.0622	3	222	18.7	394.63	2.94
4	0.06905	0.0	2.18	0	0.458	7.147	54.2	6.0622	3	222	18.7	396.90	5.33
...	...	...	...	...	...	...	...	...	...	...	...	...	...
501	0.06263	0.0	11.93	0	0.573	6.593	69.1	2.4786	1	273	21.0	391.99	9.67
502	0.04527	0.0	11.93	0	0.573	6.120	76.7	2.2875	1	273	21.0	396.90	9.08
503	0.06076	0.0	11.93	0	0.573	6.976	91.0	2.1675	1	273	21.0	396.90	5.64
504	0.10959	0.0	11.93	0	0.573	6.794	89.3	2.3889	1	273	21.0	393.45	6.48
505	0.04741	0.0	11.93	0	0.573	6.030	80.8	2.5050	1	273	21.0	396.90	7.88

506 rows × 13 columns

In [5]:
```
y=df.iloc[:,-1]
y
```

Out[5]:
```
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 [6]:
```
from sklearn.preprocessing import StandardScaler
sc=StandardScaler()
x_sc=sc.fit_transform(x)
x_sc
```

Out[6]:
```
array([[-0.41978194,  0.28482986, -1.2879095 , ..., -1.45900038,
         0.44105193, -1.0755623 ],
       [-0.41733926, -0.48772236, -0.59338101, ..., -0.30309415,
         0.44105193, -0.49243937],
       [-0.41734159, -0.48772236, -0.59338101, ..., -0.30309415,
         0.39642699, -1.2087274 ],
       ...,
       [-0.41344658, -0.48772236,  0.11573841, ...,  1.17646583,
         0.44105193, -0.98304761],
       [-0.40776407, -0.48772236,  0.11573841, ...,  1.17646583,
         0.4032249 , -0.86530163],
       [-0.41500016, -0.48772236,  0.11573841, ...,  1.17646583,
         0.44105193, -0.66905833]])
```

In [20]:
```
from sklearn.linear_model import Lasso
names=x.columns
```

In [21]:
```
def lasso(alphas):
    df1=pd.DataFrame()
    df1['FeatureName']=names
    for alpha in alphas:
        lasso=Lasso(alpha = alpha)
        lasso.fit(x_sc,y)
        column_name='Alpha= %f' %alpha
        df1[column_name]=lasso.coef_
    return df1
lasso([0.0001,0.001,0.01,0.5,1,10,100])
```

Out[21]:

	FeatureName	Alpha= 0.000100	Alpha= 0.001000	Alpha= 0.010000	Alpha= 0.500000	Alpha= 1.000000	Alpha= 10.000000	Alpha= 100.000000
0	crim	-0.927866	-0.925348	-0.900245	-0.115265	-0.000000	-0.0	-0.0
1	zn	1.081086	1.076739	1.035916	0.000000	0.000000	0.0	0.0
2	indus	0.139960	0.131471	0.046924	-0.000000	-0.000000	-0.0	-0.0
3	chas	0.681771	0.682060	0.684152	0.397079	0.000000	0.0	0.0
4	nox	-2.055877	-2.048349	-1.980551	-0.000000	-0.000000	-0.0	-0.0
5	rm	2.674402	2.675950	2.687312	2.974259	2.713355	0.0	0.0
6	age	0.019026	0.015049	0.000000	-0.000000	-0.000000	-0.0	-0.0
7	dis	-3.103667	-3.100300	-3.058301	-0.170569	-0.000000	0.0	0.0
8	rad	2.660381	2.643836	2.481844	-0.000000	-0.000000	-0.0	-0.0
9	tax	-2.074993	-2.058853	-1.899442	-0.000000	-0.000000	-0.0	-0.0
10	ptratio	-2.060372	-2.058263	-2.038645	-1.598449	-1.343549	-0.0	-0.0
11	b	0.849183	0.848414	0.839724	0.543139	0.180957	0.0	0.0
12	lstat	-3.743418	-3.741514	-3.730874	-3.666144	-3.543381	-0.0	-0.0

In [ ]:

In [ ]: