Feature Selection- With Correlation

In this step we will be removing the features which are highly correlated

from sklearn.datasets import load_boston import pandas as pd import matplotlib.pyplot as plt

In [61]:
```
data=load_boston()
```

In [62]:
```
data.feature_names
```

Out[62]:
```
array(['CRIM', 'ZN', 'INDUS', 'CHAS', 'NOX', 'RM', 'AGE', 'DIS', 'RAD',
       'TAX', 'PTRATIO', 'B', 'LSTAT'], dtype='<U7')
```

In [63]:
```
df=pd.DataFrame(data.data,columns=data.feature_names)
```

In [64]:
```
df.head()
```

Out[64]:

	CRIM	ZN	INDUS	CHAS	NOX	RM	AGE	DIS	RAD	TAX	PTRATIO	B	LSTAT
0	0.00632	18.0	2.31	0.0	0.538	6.575	65.2	4.0900	1.0	296.0	15.3	396.90	4.98
1	0.02731	0.0	7.07	0.0	0.469	6.421	78.9	4.9671	2.0	242.0	17.8	396.90	9.14
2	0.02729	0.0	7.07	0.0	0.469	7.185	61.1	4.9671	2.0	242.0	17.8	392.83	4.03
3	0.03237	0.0	2.18	0.0	0.458	6.998	45.8	6.0622	3.0	222.0	18.7	394.63	2.94
4	0.06905	0.0	2.18	0.0	0.458	7.147	54.2	6.0622	3.0	222.0	18.7	396.90	5.33

In [65]:
```
df['MEDV']=data.target
```

In [66]:
```
df.head()
```

Out[66]:

	CRIM	ZN	INDUS	CHAS	NOX	RM	AGE	DIS	RAD	TAX	PTRATIO	B	LSTAT	MEDV
0	0.00632	18.0	2.31	0.0	0.538	6.575	65.2	4.0900	1.0	296.0	15.3	396.90	4.98	24.0
1	0.02731	0.0	7.07	0.0	0.469	6.421	78.9	4.9671	2.0	242.0	17.8	396.90	9.14	21.6
2	0.02729	0.0	7.07	0.0	0.469	7.185	61.1	4.9671	2.0	242.0	17.8	392.83	4.03	34.7
3	0.03237	0.0	2.18	0.0	0.458	6.998	45.8	6.0622	3.0	222.0	18.7	394.63	2.94	33.4
4	0.06905	0.0	2.18	0.0	0.458	7.147	54.2	6.0622	3.0	222.0	18.7	396.90	5.33	36.2

In [67]:
```
x=df.drop('MEDV',axis=1)
y=df['MEDV']
```

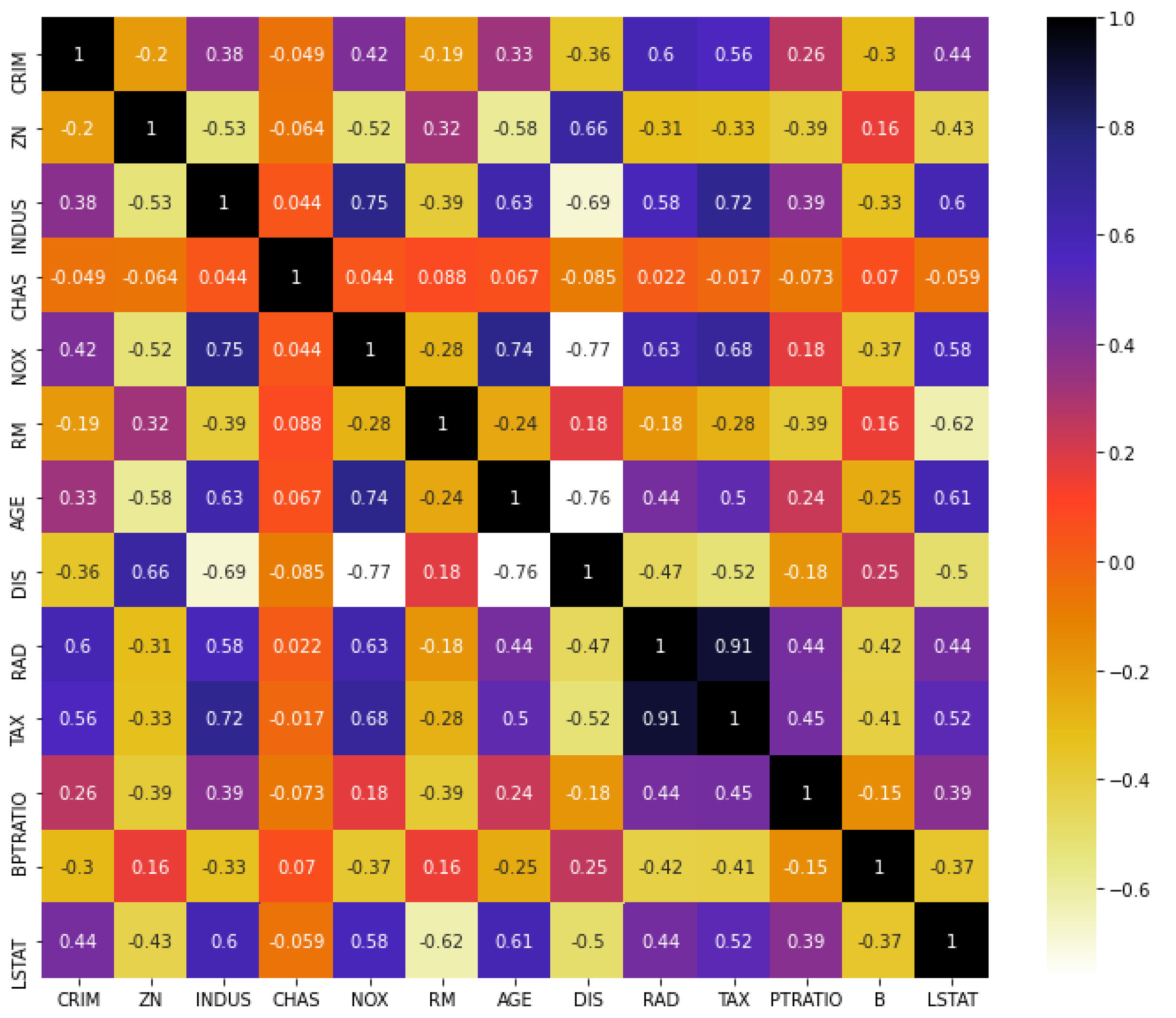
In [68]:
```
from sklearn.model_selection import train_test_split
```

In [69]:
```
xtrain,xtest,ytrain,ytest=train_test_split(x,y,random_state=0,test_size=.30)
```

In [70]:
```
import seaborn as sns
import matplotlib.pyplot as plt


plt.figure(figsize=(12,10))
sns.heatmap(xtrain.corr(),annot=True,cmap=plt.cm.CMRmap_r)
```

Out[70]: <AxesSubplot:>



In [71]:
```
# with the following function we can select highly correlated features
# it will remove the first feature that is correlated with anything other feature

def correlation(dataset, threshold):
    col_corr = set()  # Set of all the names of correlated columns
    corr_matrix = dataset.corr()
    for i in range(len(corr_matrix.columns)):
        for j in range(i):
            if abs(corr_matrix.iloc[i, j]) > threshold: # we are interested in absolute coeff value
                colname = corr_matrix.columns[i]  # getting the name of column
                col_corr.add(colname)
    return col_corr
```

In [72]:
```
corr_features=correlation(xtrain,0.7)
```

In [73]:
```
len(corr_features)
```

Out[73]: 4

In [74]:
```
corr_features
```

Out[74]: {'AGE', 'DIS', 'NOX', 'TAX'}

In [75]:
```
xtrain.drop(corr_features,axis=1)
```

Out[75]:

	CRIM	ZN	INDUS	CHAS	RM	RAD	PTRATIO	B	LSTAT
141	1.62864	0.0	21.89	0.0	5.019	4.0	21.2	396.90	34.41
272	0.11460	20.0	6.96	0.0	6.538	3.0	18.6	394.96	7.73
135	0.55778	0.0	21.89	0.0	6.335	4.0	21.2	394.67	16.96
298	0.06466	70.0	2.24	0.0	6.345	5.0	14.8	368.24	4.97
122	0.09299	0.0	25.65	0.0	5.961	2.0	19.1	378.09	17.93
...	...	...	...	...	...	...	...	...	...
323	0.28392	0.0	7.38	0.0	5.708	5.0	19.6	391.13	11.74
192	0.08664	45.0	3.44	0.0	7.178	5.0	15.2	390.49	2.87
117	0.15098	0.0	10.01	0.0	6.021	6.0	17.8	394.51	10.30
47	0.22927	0.0	6.91	0.0	6.030	3.0	17.9	392.74	18.80
172	0.13914	0.0	4.05	0.0	5.572	5.0	16.6	396.90	14.69

354 rows × 9 columns

In [76]:
```
xtest.drop(corr_features,axis=1)
```

Out[76]:

	CRIM	ZN	INDUS	CHAS	RM	RAD	PTRATIO	B	LSTAT
329	0.06724	0.0	3.24	0.0	6.333	4.0	16.9	375.21	7.34
371	9.23230	0.0	18.10	0.0	6.216	24.0	20.2	366.15	9.53
219	0.11425	0.0	13.89	1.0	6.373	5.0	16.4	393.74	10.50
403	24.80170	0.0	18.10	0.0	5.349	24.0	20.2	396.90	19.77
78	0.05646	0.0	12.83	0.0	6.232	5.0	18.7	386.40	12.34
...	...	...	...	...	...	...	...	...	...
4	0.06905	0.0	2.18	0.0	7.147	3.0	18.7	396.90	5.33
428	7.36711	0.0	18.10	0.0	6.193	24.0	20.2	96.73	21.52
385	16.81180	0.0	18.10	0.0	5.277	24.0	20.2	396.90	30.81
308	0.49298	0.0	9.90	0.0	6.635	4.0	18.4	396.90	4.54
5	0.02985	0.0	2.18	0.0	6.430	3.0	18.7	394.12	5.21

152 rows × 9 columns