In [1]:

```
from sklearn import datasets
```

In [3]:

```
# What is 'datasets'?
datasets
```

Out[3]:

```
<module 'sklearn.datasets' from 'C:\\Users\\vtaor\\Anaconda3\\lib\\site-pa
ckages\\sklearn\\datasets\\__init__.py'>
```

In [4]:

```
# What is inside of datasets?
dir(datasets)
```

Out[4]:

```
['__all__',
 '__builtins__',
 '__cached__',
 '__doc__',
 '__file__',
 '__loader__',
 '__name__',
 '__package__',
 '__path__',
 '__spec__',
 '_svmlight_format',
 'base',
 'california_housing',
 'clear_data_home',
 'covtype',
 'dump_svmlight_file',
 'fetch_20newsgroups',
 'fetch_20newsgroups_vectorized',
 'fetch_california_housing',
 'fetch_covtype',
 'fetch_kddcup99',
 'fetch_lfw_pairs',
 'fetch_lfw_people',
 'fetch_mldata',
 'fetch_olivetti_faces',
 'fetch_openml',
 'fetch_rcv1',
 'fetch_species_distributions',
 'get_data_home',
 'kddcup99',
 'lfw',
 'load_boston',
 'load_breast_cancer',
 'load_diabetes',
 'load_digits',
 'load_files',
 'load_iris',
 'load_linnerud',
 'load_sample_image',
 'load_sample_images',
 'load_svmlight_file',
 'load_svmlight_files',
 'load_wine',
 'make_biclusters',
 'make_blobs',
 'make_checkerboard',
 'make_circles',
 'make_classification',
 'make_friedman1',
 'make_friedman2',
 'make_friedman3',
 'make_gaussian_quantiles',
 'make_hastie_10_2',
 'make_low_rank_matrix',
 'make_moons',
 'make_multilabel_classification',
 'make_regression',
 'make_s_curve',
 'make_sparse_coded_signal',
```

```
 'make_sparse_spd_matrix',
 'make_sparse_uncorrelated',
 'make_spd_matrix',
 'make_swiss_roll',
 'mldata',
 'mldata_filename',
 'olivetti_faces',
 'openml',
 'rcv1',
 'samples_generator',
 'species_distributions',
 'svmlight_format',
 'twenty_newsgroups']
```

In [2]:

```
from datasets import load_boston
'''
The reason this threw an error is because it went to the following path as instructed b
y import
C:\\Users\\vtaor\\Anaconda3\\lib\\site-packages\\
There it could not find datasets.
'''
```

```
---------------------------------------------------------------------------
-
ModuleNotFoundError                       Traceback (most recent call las
t)
<ipython-input-2-58e09abc0b3d> in <module>
----> 1 from datasets import load_boston
      2 '''
      3 The reason this threw an error is because it went to the following
path as instructed by import
      4 C:\\Users\\vtaor\\Anaconda3\\lib\\site-packages\\
      5 There it could not find datasets.

ModuleNotFoundError: No module named 'datasets'
```

In [5]:

```
# Getting the Boston housing prices dataset form datasets
from sklearn.datasets import load_boston
boston = load_boston()
```

In [6]:

```
print(type(boston))
print('**********')

# The type is <class 'sklearn.utils.Bunch'> But it works like a Dictionary
# So use the method keys to find out what is in it
print(boston.keys())
print('-----------')
#print(boston)
```

```
<class 'sklearn.utils.Bunch'>
**********
dict_keys(['data', 'target', 'feature_names', 'DESCR', 'filename'])
-----------
```

In [7]:

```
#Let's find out what some keys are.
print('the list of keys in boston are: ',boston.keys())
print('*******************************')
print('These are the columns in the ',boston['feature_names'])
print('File name of this is: ',boston['filename'])
```

```
the list of keys in boston are:  dict_keys(['data', 'target', 'feature_nam
es', 'DESCR', 'filename'])
*******************************
These are the columns in the  ['CRIM' 'ZN' 'INDUS' 'CHAS' 'NOX' 'RM' 'AGE'
'DIS' 'RAD' 'TAX' 'PTRATIO'
 'B' 'LSTAT']
File name of this is:  C:\Users\vtaor\Anaconda3\lib\site-packages\sklearn
\datasets\data\boston_house_prices.csv
```

In [8]:

```
#Usual imports
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
```

In [29]:

```
#boston_housing_from_csv = pd.read_csv('C:\Users\vtaor\Anaconda3\lib\site-packages\skle
arn\datasets\data\boston_house_prices.csv')
#boston_housing_from_csv.head()
# I need to figure this out exactly what this means
```

```
  File "<ipython-input-29-9efd92f3279d>", line 1
    boston_housing_from_csv = pd.read_csv('C:\Users\vtaor\Anaconda3\lib\si
te-packages\sklearn\datasets\data\boston_house_prices.csv')
                                                              ^
SyntaxError: (unicode error) 'unicodeescape' codec can't decode bytes in p
osition 2-3: truncated \UXXXXXXXX escape
```

In [9]:

```
# Build a data frame from the dataset
# Use the method info() to find out shape, data types, non-null values, names of the co
lumns etc.
boston_housing = pd.DataFrame(boston['data'], columns=boston['feature_names'])
boston_housing.info()
# Notice there is no 'Price' column as that is the 'Target' Key in the dict
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 506 entries, 0 to 505
Data columns (total 13 columns):
CRIM       506 non-null float64
ZN         506 non-null float64
INDUS      506 non-null float64
CHAS       506 non-null float64
NOX        506 non-null float64
RM         506 non-null float64
AGE        506 non-null float64
DIS        506 non-null float64
RAD        506 non-null float64
TAX        506 non-null float64
PTRATIO    506 non-null float64
B          506 non-null float64
LSTAT      506 non-null float64
dtypes: float64(13)
memory usage: 51.5 KB
```

In [13]:

```
# Create a dataframe called 'price' from 'boston['target']'
price = pd.DataFrame(boston['target'], columns=['Price'])
# Concatenate two dataframes into new one called 'boston_housing_price'
boston_housing_price = pd.concat([boston_housing,price],axis=1)
# Again check with the info() method
boston_housing_price.info()
```

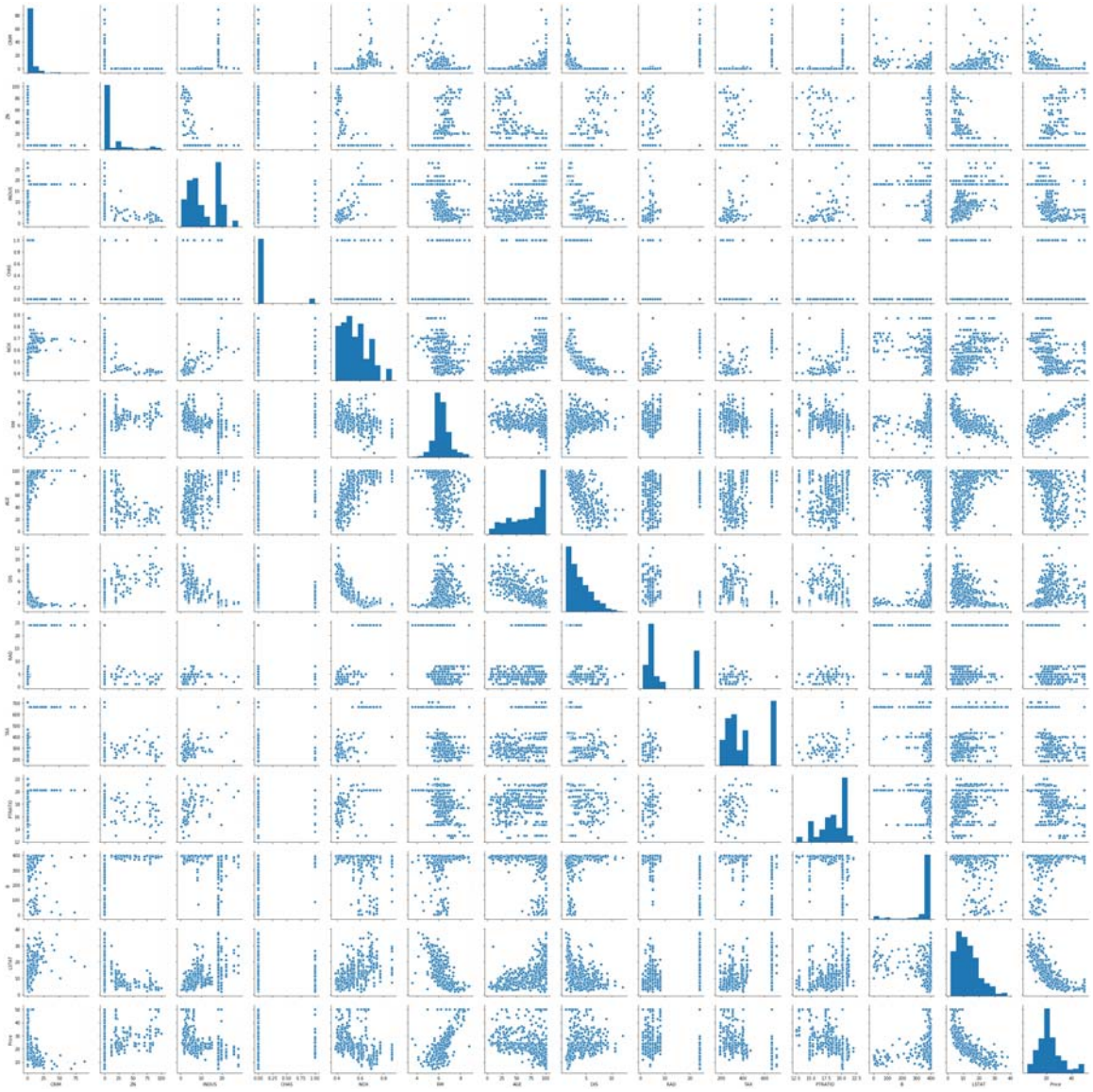
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 506 entries, 0 to 505
Data columns (total 14 columns):
CRIM       506 non-null float64
ZN         506 non-null float64
INDUS      506 non-null float64
CHAS       506 non-null float64
NOX        506 non-null float64
RM         506 non-null float64
AGE        506 non-null float64
DIS        506 non-null float64
RAD        506 non-null float64
TAX        506 non-null float64
PTRATIO    506 non-null float64
B          506 non-null float64
LSTAT      506 non-null float64
Price      506 non-null float64
dtypes: float64(14)
memory usage: 55.4 KB
```

In [14]:

```
sns.pairplot(boston_housing_price)
```

Out[14]:

```
<seaborn.axisgrid.PairGrid at 0x26152031b38>
```



In [19]:

```
# Above plot is not helpful
# try corr() heatmap
figure_ = plt.figure(figsize=(12,8))
sns.heatmap(data=boston_housing_price.corr(),annot=True)
# 14 squares for 14 columns. If not increase the size of figure
# I had to increase the size so it could show the corr.coef. on each cell
```

Out[19]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x2615da8b7f0>
```

In [21]:

```
# Machine learning section
from sklearn.model_selection import train_test_split
```

In [23]:

```
X_train, X_test, y_train, y_test = train_test_split(boston_housing, boston_housing_pric
e['Price'], test_size=0.33, random_state=42)
```

In [24]:

```
# Training the model
from sklearn.linear_model import LinearRegression
lm = LinearRegression()
lm.fit(X_train, y_train)
print('The intercept is: ', lm.intercept_)
print('The set of coefficiets is: ', lm.coef_)
```

```
The intercept is:  33.33497575563571
The set of coefficiets is:  [-1.28749718e-01  3.78232228e-02  5.82109233e-
02  3.23866812e+00
 -1.61698120e+01  3.90205116e+00 -1.28507825e-02 -1.42222430e+00
  2.34853915e-01 -8.21331947e-03 -9.28722459e-01  1.17695921e-02
 -5.47566338e-01]
```

In [25]:

```
# Predictions
pred_values = lm.predict(X_test)
pred_ratio = pred_values/y_test
print(pred_values)
```

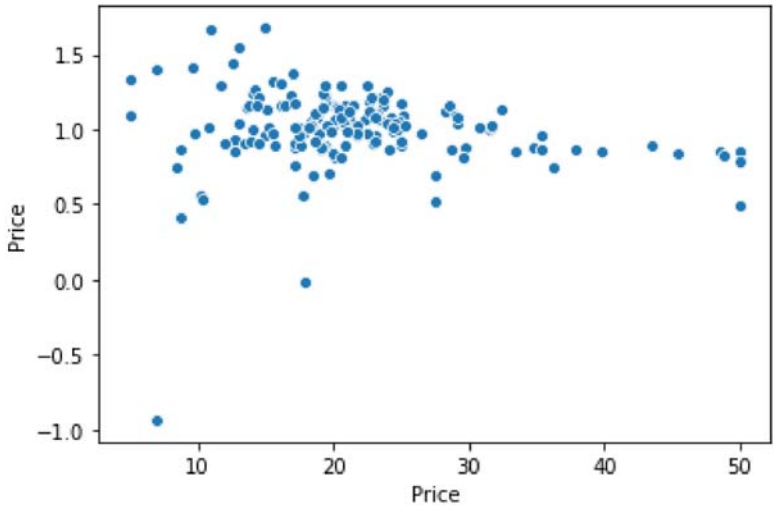
```
[28.53469469 36.6187006  15.63751079 25.5014496  18.7096734  23.16471591
 17.31011035 14.07736367 23.01064388 20.54223482 24.91632351 18.41098052
 -6.52079687 21.83372604 19.14903064 26.0587322  20.30232625  5.74943567
 40.33137811 17.45791446 27.47486665 30.2170757  10.80555625 23.87721728
 17.99492211 16.02608791 23.268288   14.36825207 22.38116971 19.3092068
 22.17284576 25.05925441 25.13780726 18.46730198 16.60405712 17.46564046
 30.71367733 20.05106788 23.9897768  24.94322408 13.97945355 31.64706967
 42.48057206 17.70042814 26.92507869 17.15897719 13.68918087 26.14924245
 20.2782306  29.99003492 21.21260347 34.03649185 15.41837553 25.95781061
 39.13897274 22.96118424 18.80310558 33.07865362 24.74384155 12.83640958
 22.41963398 30.64804979 31.59567111 16.34088197 20.9504304  16.70145875
 20.23215646 26.1437865  31.12160889 11.89762768 20.45432404 27.48356359
 10.89034224 16.77707214 24.02593714  5.44691807 21.35152331 41.27267175
 18.13447647  9.8012101  21.24024342 13.02644969 21.80198374  9.48201752
 22.99183857 31.90465631 18.95594718 25.48515032 29.49687019 20.07282539
 25.5616062   5.59584382 20.18410904 15.08773299 14.34562117 20.85155407
 24.80149389 -0.19785401 13.57649004 15.64401679 22.03765773 24.70314482
 10.86409112 19.60231067 23.73429161 12.08082177 18.40997903 25.4366158
 20.76506636 24.68588237  7.4995836  18.93015665 21.70801764 27.14350579
 31.93765208 15.19483586 34.01357428 12.85763091 21.06646184 28.58470042
 15.77437534 24.77512495  3.64655689 23.91169589 25.82292925 23.03339677
 25.35158335 33.05655447 20.65930467 38.18917361 14.04714297 25.26034469
 17.6138723  20.60883766  9.8525544  21.06756951 22.20145587 32.2920276
 31.57638342 15.29265938 16.7100235  29.10550932 25.17762329 16.88159225
  6.32621877 26.70210263 23.3525851  17.24168182 13.22815696 39.49907507
 16.53528575 18.14635902 25.06620426 23.70640231 22.20167772 21.22272327
 16.89825921 23.15518273 28.69699805  6.65526482 23.98399958 17.21004545
 21.0574427  25.01734597 27.65461859 20.70205823 40.38214871]
```

In [27]:

```
sns.scatterplot(x=y_test,y=pred_ratio)
# It shows that model does a good job of predicting values and pred_values/y_test is ve
ry colse to 1 after a threshold
# let's set the threshold to 12
# X-axis is the Price
# Y-axis shows the corresponding ratio of 'Price' to 'predicted values of Price'
```

Out[27]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x2615ddffeb8>
```



In [28]:

```
f = lambda x: x>12
greater_12 = y_test.apply(f)
greater_12.head()
```

Out[28]:

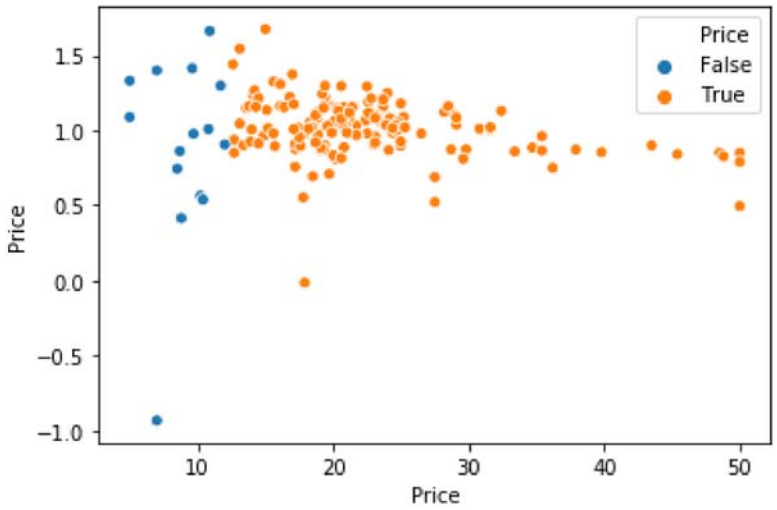
```
173    True
274    True
491    True
72     True
452    True
Name: Price, dtype: bool
```

In [29]:

```
#Re-plotting the above graph with threshold of 12
sns.scatterplot(x=y_test,y=pred_ratio, hue=greater_12)
```

Out[29]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x2615ddff4a8>
```

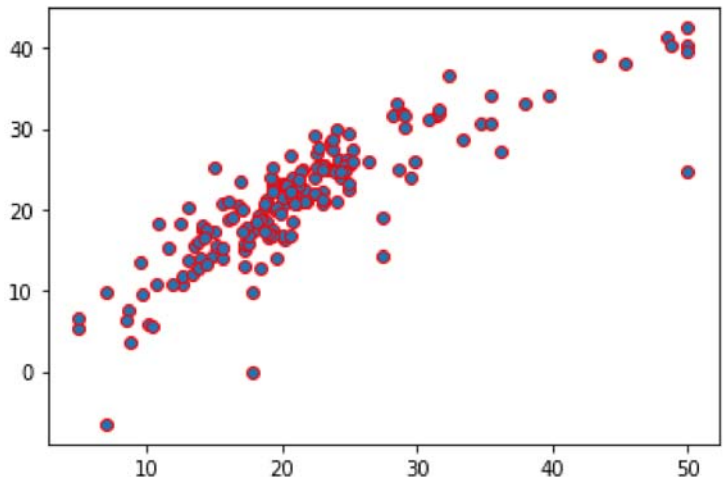


In [30]:

```
# Another way to visualize the accuracy of the model is to plot test values against pre
dicted values
plt.scatter(y_test,pred_values,edgecolors='red')
```

Out[30]:

```
<matplotlib.collections.PathCollection at 0x2615e208ef0>
```



In [33]:

```
# Residuals are the difference in the actual value and the predicted value 'y_test' min
us 'a'

# Residuals should have a normally distributed nature for a good fit

residuals = y_test -pred_values
residuals.head()
```

Out[33]:

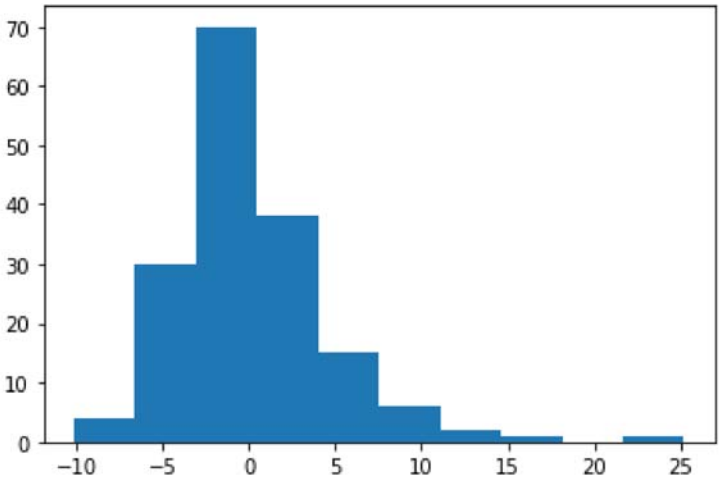
```
173   -4.934695
274   -4.218701
491   -2.037511
72    -2.701450
452   -2.609673
Name: Price, dtype: float64
```

In [34]:

```
plt.hist(residuals)
```

Out[34]:

```
(array([ 4., 30., 70., 38., 15.,  6.,  2.,  1.,  0.,  1.]),
 array([-10.13780726,  -6.60417592,  -3.07054458,   0.46308675,
          3.99671809,   7.53034943,  11.06398077,  14.5976121 ,
         18.13124344,  21.66487478,  25.19850611]),
 <a list of 10 Patch objects>)
```



In [37]:

```
cpv = pd.concat(y_test, pred_values)
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
<ipython-input-37-eeba2ea4a458> in <module>
----> 1 cpv = pd.concat(y_test, pred_values)

~\Anaconda3\lib\site-packages\pandas\core\reshape\concat.py in concat(objs, axis, join, join_axes, ignore_index, keys, levels, names, verify_integrity, sort, copy)
    226                       keys=keys, levels=levels, names=names,
    227                       verify_integrity=verify_integrity,
--> 228                       copy=copy, sort=sort)
    229     return op.get_result()
    230

~\Anaconda3\lib\site-packages\pandas\core\reshape\concat.py in __init__(self, objs, axis, join, join_axes, keys, levels, names, ignore_index, verify_integrity, copy, sort)
    242             raise TypeError('first argument must be an iterable of pandas '
    243                             'objects, you passed an object of type '
--> 244                             '"{name}"'.format(name=type(objs).__name__))
    245
    246         if join == 'outer':

TypeError: first argument must be an iterable of pandas objects, you passed an object of type "Series"
```