Data Analytics I

May 2, 2022

[1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

[2]:
```
from sklearn.datasets import load_boston
boston = load_boston()
```

[3]:
```
data = pd.DataFrame(boston.data)
```

[4]:
```
data.columns = boston.feature_names
data.head()
```

[4]:
```
      CRIM    ZN  INDUS  CHAS    NOX     RM   AGE     DIS  RAD    TAX  \
0  0.00632  18.0   2.31   0.0  0.538  6.575  65.2  4.0900  1.0  296.0
1  0.02731   0.0   7.07   0.0  0.469  6.421  78.9  4.9671  2.0  242.0
2  0.02729   0.0   7.07   0.0  0.469  7.185  61.1  4.9671  2.0  242.0
3  0.03237   0.0   2.18   0.0  0.458  6.998  45.8  6.0622  3.0  222.0
4  0.06905   0.0   2.18   0.0  0.458  7.147  54.2  6.0622  3.0  222.0

   PTRATIO       B  LSTAT
0     15.3  396.90   4.98
1     17.8  396.90   9.14
2     17.8  392.83   4.03
3     18.7  394.63   2.94
4     18.7  396.90   5.33
```

[5]:
```
data['PRICE'] = boston.target
```

[6]:
```
data
```

[6]:
```
        CRIM    ZN  INDUS  CHAS    NOX     RM   AGE     DIS  RAD    TAX  \
0    0.00632  18.0   2.31   0.0  0.538  6.575  65.2  4.0900  1.0  296.0
1    0.02731   0.0   7.07   0.0  0.469  6.421  78.9  4.9671  2.0  242.0
2    0.02729   0.0   7.07   0.0  0.469  7.185  61.1  4.9671  2.0  242.0
3    0.03237   0.0   2.18   0.0  0.458  6.998  45.8  6.0622  3.0  222.0
4    0.06905   0.0   2.18   0.0  0.458  7.147  54.2  6.0622  3.0  222.0
..       …    …     …    …     …     …    …      …  …     …
```

```
501  0.06263   0.0  11.93   0.0  0.573  6.593  69.1  2.4786  1.0  273.0
502  0.04527   0.0  11.93   0.0  0.573  6.120  76.7  2.2875  1.0  273.0
503  0.06076   0.0  11.93   0.0  0.573  6.976  91.0  2.1675  1.0  273.0
504  0.10959   0.0  11.93   0.0  0.573  6.794  89.3  2.3889  1.0  273.0
505  0.04741   0.0  11.93   0.0  0.573  6.030  80.8  2.5050  1.0  273.0

     PTRATIO       B  LSTAT  PRICE
0       15.3  396.90   4.98   24.0
1       17.8  396.90   9.14   21.6
2       17.8  392.83   4.03   34.7
3       18.7  394.63   2.94   33.4
4       18.7  396.90   5.33   36.2
..       …       …     …      …
501     21.0  391.99   9.67   22.4
502     21.0  396.90   9.08   20.6
503     21.0  396.90   5.64   23.9
504     21.0  393.45   6.48   22.0
505     21.0  396.90   7.88   11.9

[506 rows x 14 columns]
```

[7]:

```
data.isnull().sum()
```

[7]:

```
CRIM       0
ZN         0
INDUS      0
CHAS       0
NOX        0
RM         0
AGE        0
DIS        0
RAD        0
TAX        0
PTRATIO    0
B          0
LSTAT      0
PRICE      0
dtype: int64
```

[17]:

```
x = data.drop(['PRICE'], axis=1)
y = data['PRICE']
```

[18]:

```
from sklearn.model_selection import train_test_split
xtrain, xtest, ytrain, ytest = train_test_split(x, y, test_size = 0.2,␣
 ↪random_state = 0)
```

[19]:

```python
from sklearn.linear_model import LinearRegression
lr = LinearRegression()
model = lr.fit(xtrain,ytrain)
```

[20]:

```python
ytrain_pred = lr.predict(xtrain)
ytest_pred = lr.predict(xtest)
```

[24]:

```python
df = pd.DataFrame(ytrain_pred,ytrain)
df = pd.DataFrame(ytest_pred,ytest)
```

[30]:

```python
from sklearn.metrics import mean_squared_error, r2_score
mse = mean_squared_error(ytest,ytest_pred)
print(mse)
```

```
33.448979997676524
```

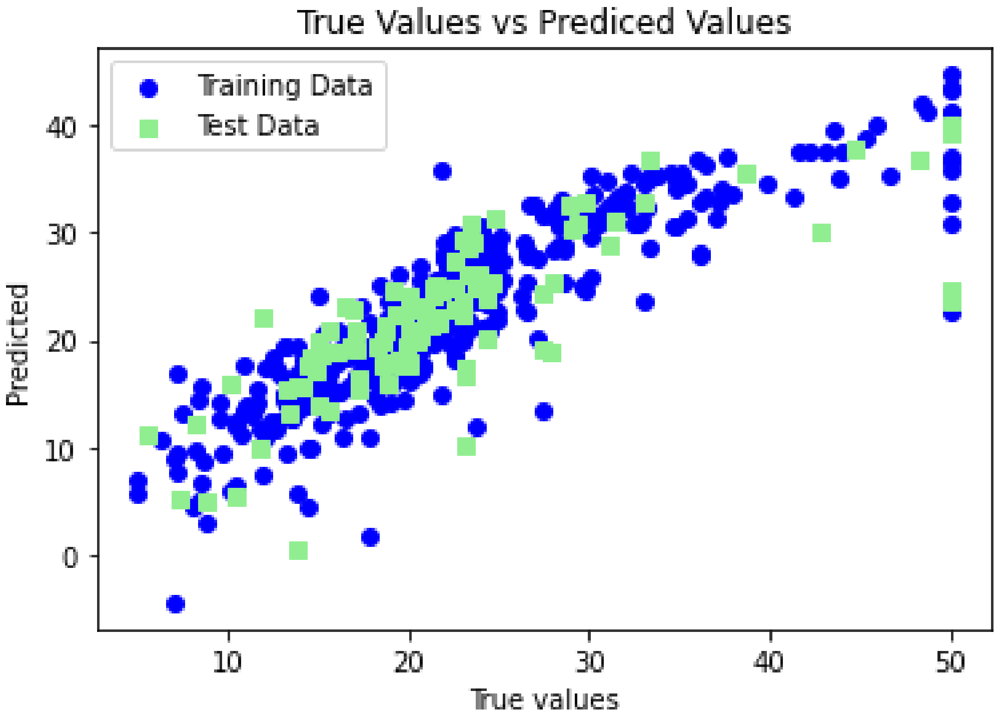
[29]:

```python
mse = r2_score(ytrain,ytrain_pred)
print(mse)
```

```
0.7730135569264234
```

[25]:

```python
plt.scatter(ytrain, ytrain_pred, c='blue', marker='o', label="Training Data")
plt.scatter(ytest,ytest_pred, c='lightgreen', marker='s', label='Test Data')
plt.xlabel("True values")
plt.ylabel("Predicted")
plt.title("True Values vs Prediced Values")
plt.legend(loc='upper left')
plt.plot()
plt.show()
```


True Values vs Prediced Values
Training Data
Test Data
Predicted
40
30
20
10
0
10
20
30
40
50
True values