Step 1

```
from sklearn.datasets import load_boston
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd

boston = load_boston()
x_data = boston.data
y_data = boston.target
name_data = boston.feature_names

x_train, x_test, y_train, y_test = train_test_split(x_data, y_data, test_size=0.2)
```

Step 2(a)

```
regr=LinearRegression()
regr.fit(x_train,y_train)
y_predict=regr.predict(x_test)
```

Step 2(b)

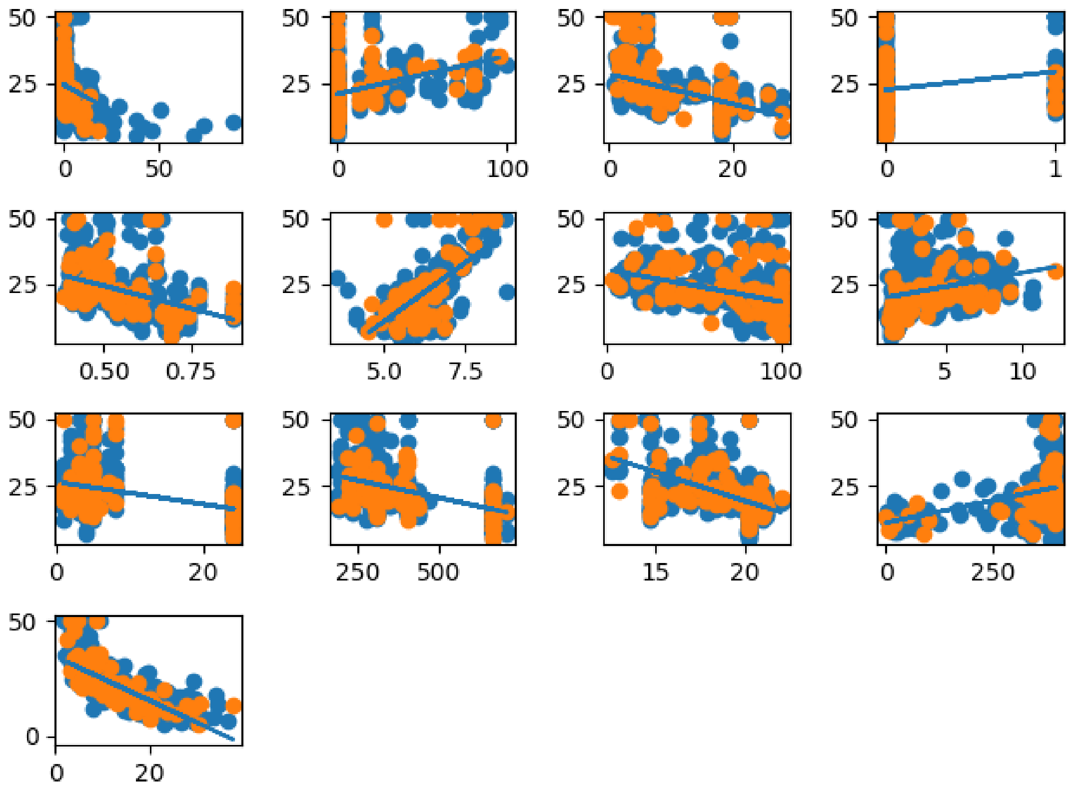
```
# The coefficients
print('Coefficients:', regr.coef_)
# The mean squared error
print("Mean squared error: %.2f" % (np.sum((y_predict-y_test)**2)/len(y_test)))
# Explained variance score : 1 is perfect prediction
print('Variance score: %.2f' % regr.score(x_test,y_test))
```

```
Coefficients: [-7.34277509e-02  5.47532507e-02  4.55890003e-02  2.30802249e+00
 -1.61582610e+01  3.51367092e+00  7.02593744e-03 -1.48021165e+00
  2.90886957e-01 -1.26316371e-02 -9.46241536e-01  1.04397651e-02
 -6.02614052e-01]
Mean squared error: 19.24
Variance score: 0.71
```

Step 3(a) & Step 3(b)

6

```
plt.figure()
for i in range(13):
    plt.subplot(4, 4, i + 1)
    x=pd.DataFrame(boston.data[:,i])
    x_train, x_test, y_train, y_test = train_test_split(x, y_data, test_size=0.2)
    lr = LinearRegression()
    lr.fit(x_train, y_train)
    y_pre =lr.predict(x_test)
    plt.scatter(x_train, y_train, label='train')
    plt.scatter(x_test, y_test, label='test')
    plt.plot(x_test.values.reshape(-1,1) , y_pre,label='line')
    print('for fit on feature:'+boston.feature_names[i])
    # The coefficients
    print('Coefficients:', lr.coef_)
    # The mean squared error
    print("Mean squared error: %.2f" % (np.sum((y_pre-y_test)**2)/len(y_test)))
    # Explained variance score : 1 is perfect prediction
    print('Variance score: %.2f' % lr.score(x_test,y_test))
plt.show()
```



```
for fit on feature:CRIM
Coefficients: [-0.41583276]
Mean squared error: 69.03
Variance score: 0.17
for fit on feature:ZN
Coefficients: [0.15021427]
Mean squared error: 82.82
Variance score: 0.02
for fit on feature:INDUS
Coefficients: [-0.62453467]
Mean squared error: 78.40
Variance score: 0.22
for fit on feature:CHAS
Coefficients: [5.31689753]
Mean squared error: 74.01
Variance score: 0.01
for fit on feature:NOX
Coefficients: [-33.86356381]
Mean squared error: 45.66
Variance score: 0.23
```

```
for fit on feature:RM
Coefficients: [9.58784094]
Mean squared error: 48.39
Variance score: 0.36
for fit on feature:AGE
Coefficients: [-0.12952383]
Mean squared error: 70.30
Variance score: 0.07
for fit on feature:DIS
Coefficients: [1.16523221]
Mean squared error: 66.77
Variance score: 0.03
for fit on feature:RAD
Coefficients: [-0.42622777]
Mean squared error: 87.65
Variance score: 0.07
for fit on feature:TAX
Coefficients: [-0.0246286]
Mean squared error: 76.94
Variance score: 0.25
```

```
for fit on feature:PTRATIO
Coefficients: [-2.20605439]
Mean squared error: 72.17
Variance score: 0.18
for fit on feature:B
Coefficients: [0.03250238]
Mean squared error: 88.37
Variance score: 0.13
for fit on feature:LSTAT
Coefficients: [-0.93338223]
Mean squared error: 38.34
Variance score: 0.53
```

Step 4(a) & Step 4(b)

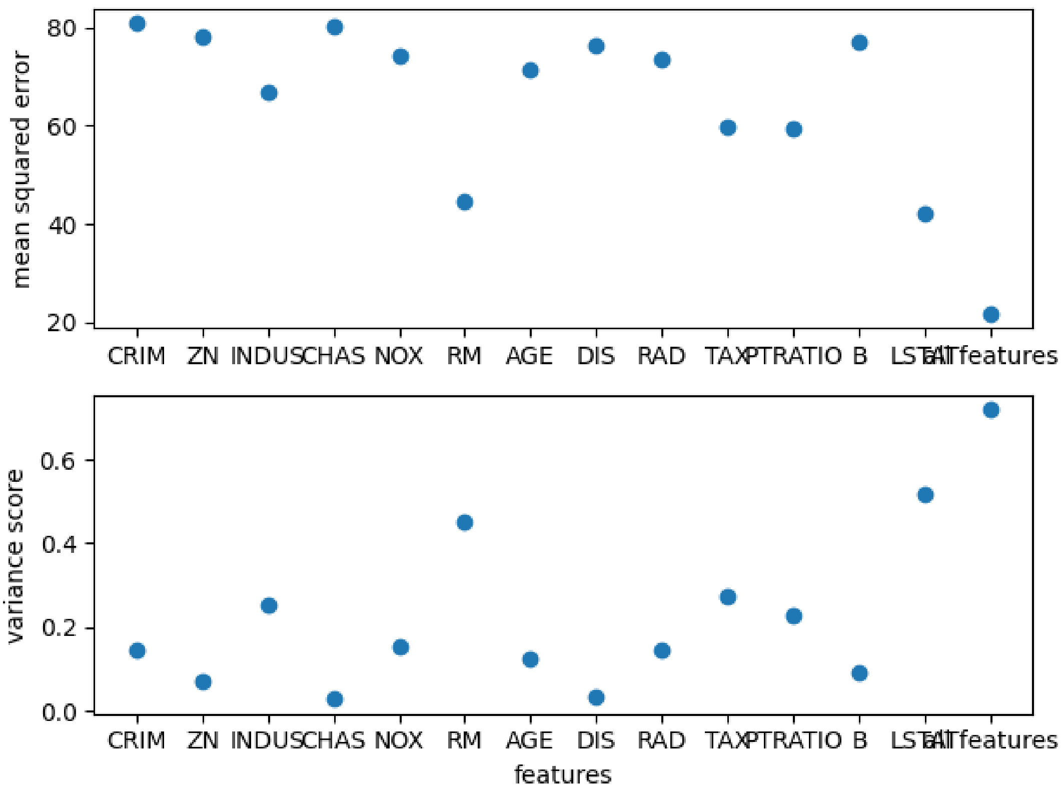
```
coef_temp=[]
mse_all=[]
vs_all=[]
coef_feas = []
mse_feas = []
vs_feas = []
for j in range(10):
    x_train, x_test, y_train, y_test = train_test_split(x_data, y_data, test_size=0.2)
    regr=LinearRegression()
    regr.fit(x_train,y_train)
    y_predict=regr.predict(x_test)
    # The coefficients
    coef_temp.append(regr.coef_)
    # The mean squared error
    mse_all.append((np.sum((y_predict-y_test)**2)/len(y_test)))
    # Explained variance score : 1 is perfect prediction
    vs_all.append(regr.score(x_test,y_test))
for m in range(10):
    temp1 = []
```

```
    temp2 = []
    temp3 = []
    for i in range(13):
        x=pd.DataFrame(boston.data[:,i])
        x_train, x_test, y_train, y_test = train_test_split(x, y_data, test_size=0.2)
        lr = LinearRegression()
        lr.fit(x_train, y_train)
        y_pre =lr.predict(x_test)
        # The coefficients
        temp1.append(lr.coef_)
        # The mean squared error
        temp2.append((np.sum((y_pre-y_test)**2)/len(y_test)))
        # Explained variance score : 1 is perfect prediction
        temp3.append(lr.score(x_test,y_test))
    coef_feas.append(temp1)
    mse_feas.append(temp2)
    vs_feas.append(temp3)
coef_all=[]
coef=[]
mse=[]
vs=[]
for n in range(13):
    temp_all=0
    temp=0
    temp_mse=0
```

```
    temp_vs=0
    for a in range(10):
        temp_all+=coef_temp[a][n]
        temp += coef_feas[a][n]
        temp_mse+=mse_feas[a][n]
        temp_vs+=vs_feas[a][n]
    temp_all=temp_all/10
    temp=temp/10
    temp_mse=temp_mse/10
    temp_vs=temp_vs/10
    coef_all.append(temp_all)
    coef.append(temp)
    mse.append(temp_mse)
    vs.append(temp_vs)
    print('for feature:' + boston.feature_names[n])
    print("Averange of coefficients: %.2f" % temp[0])
    print("Averange of mean squared error: %.2f" % mse[n])
    print("Averange of variance score: %.2f" % vs[n])

print("for all features:")
print("Averange of coefficients:",coef_all)
print("Averange of mean squared error:%.2f" % (sum(mse_all) / len(mse_all)))
print("Averange of variance score:%.2f" % (sum(vs_all) / len(vs_all)))
```

```
plt.figure()
name_data=name_data.tolist()
name_data.append("all features")
mse.append((sum(mse_all) / len(mse_all)))
vs.append((sum(vs_all) / len(vs_all)))
plt.subplot(2,1,1)
s1=plt.scatter(name_data,mse)
plt.ylabel('mean squared error')
plt.subplot(2,1,2)
s2=plt.scatter(name_data,vs)
plt.ylabel('variance score')
plt.xlabel('features')
plt.show()
```



for feature:CRIM

Averange of coefficients: -0.40

Averange of mean squared error: 79.80

Averange of variance score: 0.13

for feature:ZN

Averange of coefficients: 0.14

Averange of mean squared error: 73.95

Averange of variance score: 0.12

for feature:INDUS

Averange of coefficients: -0.66

Averange of mean squared error: 65.51

Averange of variance score: 0.20

for feature:CHAS
Averange of coefficients: 6.35
Averange of mean squared error: 85.87
Averange of variance score: 0.02
for feature:NOX
Averange of coefficients: -33.79
Averange of mean squared error: 70.44
Averange of variance score: 0.17
for feature:RM
Averange of coefficients: 9.04
Averange of mean squared error: 44.17
Averange of variance score: 0.48
for feature:AGE
Averange of coefficients: -0.13
Averange of mean squared error: 75.93
Averange of variance score: 0.11
for feature:DIS
Averange of coefficients: 1.09
Averange of mean squared error: 75.33
Averange of variance score: 0.05
for feature:RAD
Averange of coefficients: -0.40
Averange of mean squared error: 69.59
Averange of variance score: 0.13
for feature:TAX
Averange of coefficients: -0.03
Averange of mean squared error: 71.56
Averange of variance score: 0.19
for feature:PTRATIO
Averange of coefficients: -2.18
Averange of mean squared error: 62.48
Averange of variance score: 0.22
for feature:B
Averange of coefficients: 0.03
Averange of mean squared error: 79.74
Averange of variance score: 0.09
for feature:LSTAT
Averange of coefficients: -0.95
Averange of mean squared error: 41.53
Averange of variance score: 0.52
for all features:
Averange of coefficients: [-0.11013804250703951, 0.046916026226885496, 0.04544679054483304, 2.5107724659777184, -18.18201901800465, 3.744648567675101, 0.0007449163224672557, -1.4480315814174243,

0.3157197892154096, -0.012792258938602715, -0.9806929298028028, 0.009035502171159568, -0.5362748321901103]
Averange of mean squared error:23.47
Averange of variance score:0.71

1. Based upon the linear models you generated, which feature appears to be most predictive for the target feature? Note that you can answer this question based upon the output provided for the linear models.

LSTAT and RM. Because the mean squared error of these features are low and the variance score of these features are high.

2. Suppose you need to select two features for a linear regression model to predict the target feature. Which two features would you select? Why?

LSTAT and RM. Because these two characteristics have a relatively obvious linear relationship with housing prices

3. Examine all the plots and numbers you have, do you have any comments on them? Do you find any surprising trends? Do you have any idea about what might be causing this surprising trend in the data? This is a descriptive question meant to encourage you to interpret your results and express yourself.

The prediction effect is the best when all 13 features are used to predict. Some features (such as CHAS and RAD) do not have much influence on the prediction effect. It may be because their data distribution is extreme and there is no obvious linear relationship.