import numpy as np  
import matplotlib.pyplot as plt  
  
import pandas as pd  
import seaborn as sns  
  
%matplotlib inline

boston\_dataset = pd.read\_csv("BostonHousing.csv")

boston\_dataset.head()

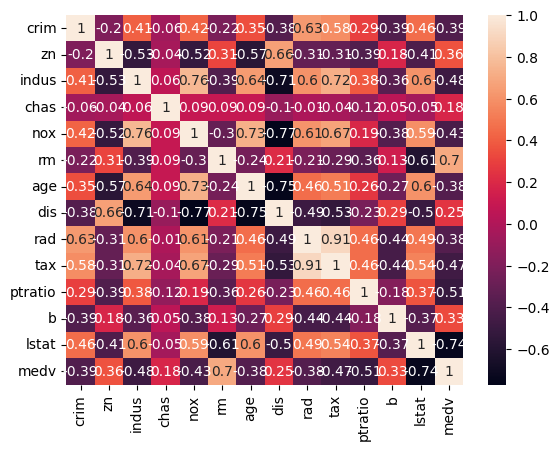
crim zn indus chas nox rm age dis rad tax ptratio \  
0 0.00632 18.0 2.31 0 0.538 6.575 65.2 4.0900 1 296 15.3   
1 0.02731 0.0 7.07 0 0.469 6.421 78.9 4.9671 2 242 17.8   
2 0.02729 0.0 7.07 0 0.469 7.185 61.1 4.9671 2 242 17.8   
3 0.03237 0.0 2.18 0 0.458 6.998 45.8 6.0622 3 222 18.7   
4 0.06905 0.0 2.18 0 0.458 7.147 54.2 6.0622 3 222 18.7   
  
 b lstat medv   
0 396.90 4.98 24.0   
1 396.90 9.14 21.6   
2 392.83 4.03 34.7   
3 394.63 2.94 33.4   
4 396.90 5.33 36.2

boston\_dataset.isnull().sum()

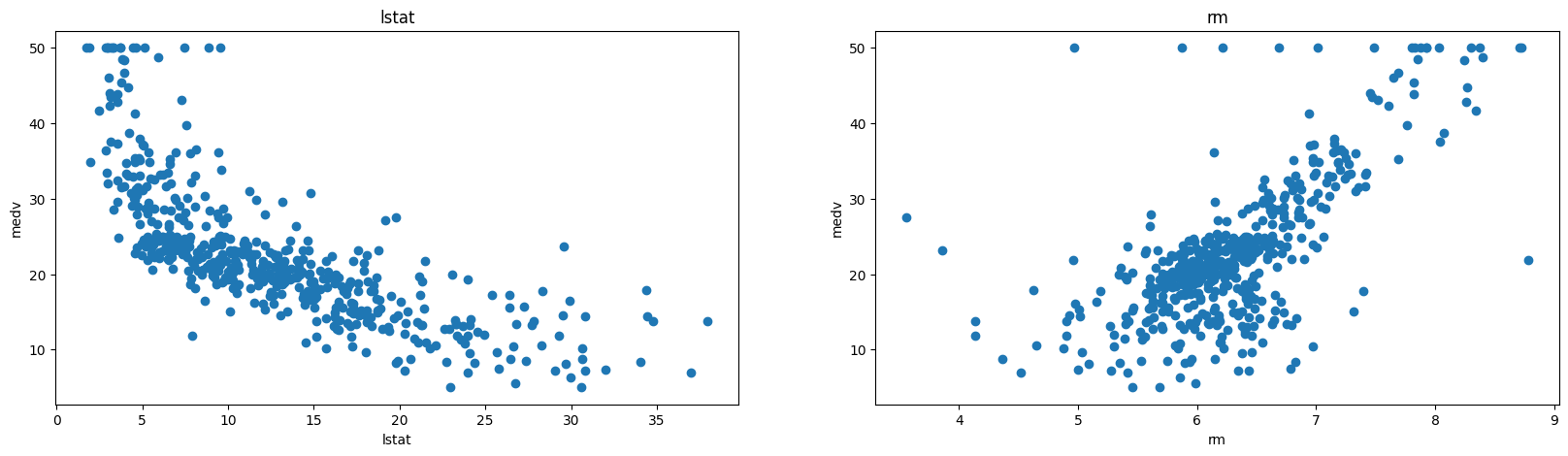
crim 0  
zn 0  
indus 0  
chas 0  
nox 0  
rm 0  
age 0  
dis 0  
rad 0  
tax 0  
ptratio 0  
b 0  
lstat 0  
medv 0  
dtype: int64

correlation\_matrix = boston\_dataset.corr().round(2)  
# annot = True to print the values inside the square  
sns.heatmap(data=correlation\_matrix, annot=True)

<Axes: >



plt.figure(figsize=(20, 5))  
  
features = ['lstat', 'rm']  
target = boston\_dataset['medv']  
  
for i, col in enumerate(features):  
 plt.subplot(1, len(features) , i+1)  
 x = boston\_dataset[col]  
 y = target  
 plt.scatter(x, y, marker='o')  
 plt.title(col)  
 plt.xlabel(col)  
 plt.ylabel('medv')



X = pd.DataFrame(np.c\_[boston\_dataset['lstat'], boston\_dataset['rm']], columns = ['lstat','rm'])  
Y = boston\_dataset['medv']

from sklearn.model\_selection import train\_test\_split  
  
X\_train, X\_test, Y\_train, Y\_test = train\_test\_split(X, Y, test\_size = 0.2, random\_state=5)  
print(X\_train.shape)  
print(X\_test.shape)  
print(Y\_train.shape)  
print(Y\_test.shape)

(404, 2)  
(102, 2)  
(404,)  
(102,)

from sklearn.linear\_model import LinearRegression  
from sklearn.metrics import mean\_squared\_error  
import statsmodels.api as sm  
  
lin\_model = LinearRegression()  
lin\_model.fit(X\_train, Y\_train)

LinearRegression()

model = sm.OLS(Y\_train, X\_train).fit()

print(model.summary())

OLS Regression Results   
=======================================================================================  
Dep. Variable: medv R-squared (uncentered): 0.947  
Model: OLS Adj. R-squared (uncentered): 0.947  
Method: Least Squares F-statistic: 3581.  
Date: Tue, 04 Jun 2024 Prob (F-statistic): 6.67e-257  
Time: 22:52:50 Log-Likelihood: -1272.2  
No. Observations: 404 AIC: 2548.  
Df Residuals: 402 BIC: 2556.  
Df Model: 2   
Covariance Type: nonrobust   
==============================================================================  
 coef std err t P>|t| [0.025 0.975]  
------------------------------------------------------------------------------  
lstat -0.6911 0.036 -19.367 0.000 -0.761 -0.621  
rm 4.9699 0.081 61.521 0.000 4.811 5.129  
==============================================================================  
Omnibus: 121.894 Durbin-Watson: 2.063  
Prob(Omnibus): 0.000 Jarque-Bera (JB): 389.671  
Skew: 1.370 Prob(JB): 2.42e-85  
Kurtosis: 6.954 Cond. No. 4.70  
==============================================================================  
  
Notes:  
[1] R² is computed without centering (uncentered) since the model does not contain a constant.  
[2] Standard Errors assume that the covariance matrix of the errors is correctly specified.