Untitled50

July 17, 2019

In [334]:
```
# First let's import all the packages that we will need in order to analyze and  vis

import seaborn as sns
from sklearn.datasets import load_boston
import matplotlib.pyplot as plt
import pandas as pd
import numpy as np

boston_dataset = load_boston()
```

In [341]:
```
# We will try to make a Linear Regression model on the Boston data.
boston = pd.DataFrame(boston_dataset.data, columns=boston_dataset.feature_names)
boston['MEDV'] = boston_dataset.target
boston.head()

# In order to know more about the data's features we can use the DESCR:
print(boston_dataset.DESCR)
```

```
Boston House Prices dataset
===========================

Notes
------
Data Set Characteristics:

    :Number of Instances: 506

    :Number of Attributes: 13 numeric/categorical predictive

    :Median Value (attribute 14) is usually the target

    :Attribute Information (in order):
        - CRIM     per capita crime rate by town
        - ZN       proportion of residential land zoned for lots over 25,000 sq.ft.
        - INDUS    proportion of non-retail business acres per town
        - CHAS     Charles River dummy variable (= 1 if tract bounds river; 0 otherwise)
        - NOX      nitric oxides concentration (parts per 10 million)
        - RM       average number of rooms per dwelling
```

```
        - AGE      proportion of owner-occupied units built prior to 1940
        - DIS      weighted distances to five Boston employment centres
        - RAD      index of accessibility to radial highways
        - TAX      full-value property-tax rate per $10,000
        - PTRATIO  pupil-teacher ratio by town
        - B        1000(Bk - 0.63)^2 where Bk is the proportion of blacks by town
        - LSTAT    % lower status of the population
        - MEDV     Median value of owner-occupied homes in $1000's

    :Missing Attribute Values: None

    :Creator: Harrison, D. and Rubinfeld, D.L.

This is a copy of UCI ML housing dataset.
http://archive.ics.uci.edu/ml/datasets/Housing


This dataset was taken from the StatLib library which is maintained at Carnegie Mellon Universi

The Boston house-price data of Harrison, D. and Rubinfeld, D.L. 'Hedonic
prices and the demand for clean air', J. Environ. Economics & Management,
vol.5, 81-102, 1978.   Used in Belsley, Kuh & Welsch, 'Regression diagnostics
...', Wiley, 1980.   N.B. Various transformations are used in the table on
pages 244-261 of the latter.

The Boston house-price data has been used in many machine learning papers that address regressi
problems.

**References**

   - Belsley, Kuh & Welsch, 'Regression diagnostics: Identifying Influential Data and Sources o
   - Quinlan,R. (1993). Combining Instance-Based and Model-Based Learning. In Proceedings on th
   - many more! (see http://archive.ics.uci.edu/ml/datasets/Housing)
```

In [340]:
```
# The correlation matrix will help us see if there  are any strong correlations betw
correlation_matrix = boston.corr().round(2)
sns.heatmap(data=correlation_matrix, annot=True)
# We can see that there is a strong correlation (0.7) between the variables MEDV and
# We will try to predict the median price (MEDV) using the variable RM  (average num
```

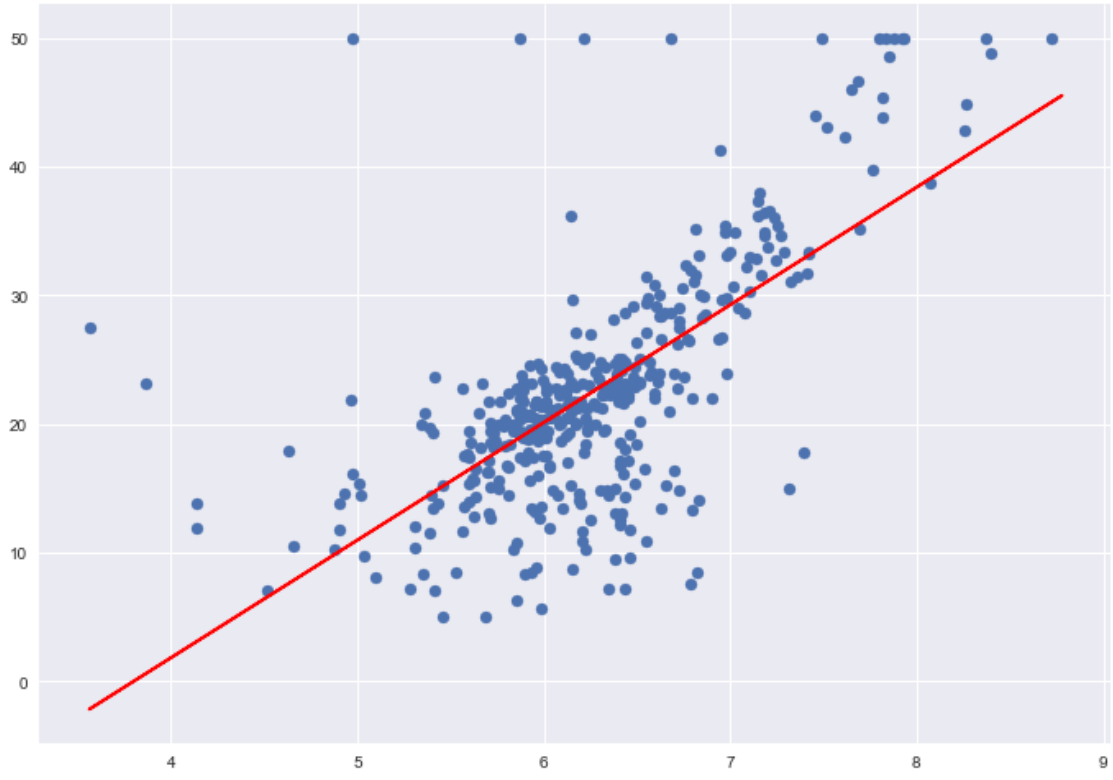
Out[340]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x1a14d95f90>
```

In [373]:
```
# Splitting our data into train and test sets:
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error

X = boston['RM']
X = X.values.reshape(-1, 1)
y  = boston['MEDV']

X_train, X_test, Y_train, Y_test = train_test_split(X, y, test_size=0.2, random_state
print(X_train.shape)
print(X_test.shape)
print(Y_train.shape)
print(Y_test.shape)

# Starting our linear regression model:

regr = LinearRegression()
plt.scatter(X_train, Y_train)
regr.fit(X_train, Y_train)
y_predicted = regr.predict(X_train)
print(regr.coef_)
```

```
print(regr.intercept_)
y_line = [9.14438088*x-34.75540260183401 for x in X]
plt.plot(X, y_line, c='r')
```

```
(404, 1)
(102, 1)
(404,)
(102,)
[9.14438088]
-34.75540260183401
```

Out[373]: [<matplotlib.lines.Line2D at 0x1a1bda0e50>]



In [377]:
```
#  Let's evaluate our model using RMSE:
Y_train_predict = regr.predict(X_train)
rmse = (np.sqrt(mean_squared_error(Y_train, Y_train_predict)))

print("The model performance for training set")
print("------------------")
print('RMSE is {}'.format(rmse))
print("\n")
```

```
        # model evaluation for testing set
        Y_test_predict = regr.predict(X_test)
        rmse = (np.sqrt(mean_squared_error(Y_test, Y_test_predict)))

        print("The model performance for testing set")
        print("------------------")
        print('RMSE is {}'.format(rmse))
```

```
The model performance for training set
------------------
RMSE is 6.66794746405


The model performance for testing set
------------------
RMSE is 6.35272414813
```