3월 31일 실습코드

20174119 한지훈

import numpy as np

import matplotlib.pyplot as plt

X = np.array([0.0, 1.0, 2.0])

y = np.array([3.0, 3.5, 5.5])

W = 0 # 기울기

b = 0 # 절편

lrate = 0.01 # 학습률

epochs = 1000 # 반복 횟수

n = float(len(X)) # 입력 데이터의 개수

# 경사 하강법

for i in range(epochs):

y\_pred = W\*X + b # 예측값

dW = (2/n) \* sum(X \* (y\_pred-y))

db = (2/n) \* sum(y\_pred-y)

W = W - lrate \* dW # 기울기 수정

b = b - lrate \* db # 절편 수정

print (W, b)

# 예측값을 만든다.

y\_pred = W\*X + b

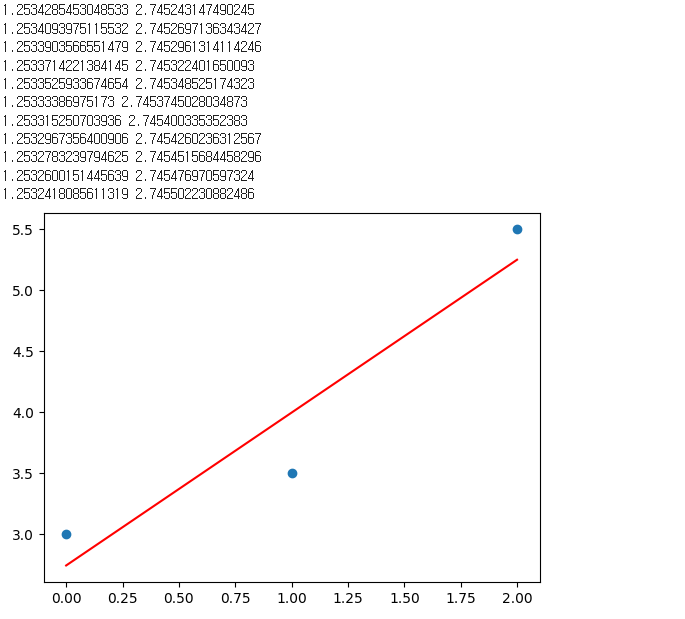
# 입력 데이터를 그래프 상에 찍는다.

plt.scatter(X, y)

# 예측값은 선그래프로 그린다.

plt.plot([min(X), max(X)], [min(y\_pred), max(y\_pred)], color='red')

plt.show()



import matplotlib.pylab as plt

from sklearn import linear\_model

# 선형 회귀 모델을 생성한다.

reg = linear\_model.LinearRegression()

# 데이터는 파이썬의 리스트로 만들어도 되고 아니면 넘파이의 배열로 만들어도 됨

X = [[0], [1], [2]] # 2차원으로 만들어야 함

y = [3, 3.5, 5.5] # y = x + 3

# 학습을 시킨다.

reg.fit(X, y)

reg.coef\_

reg.intercept\_

reg.score(X, y)

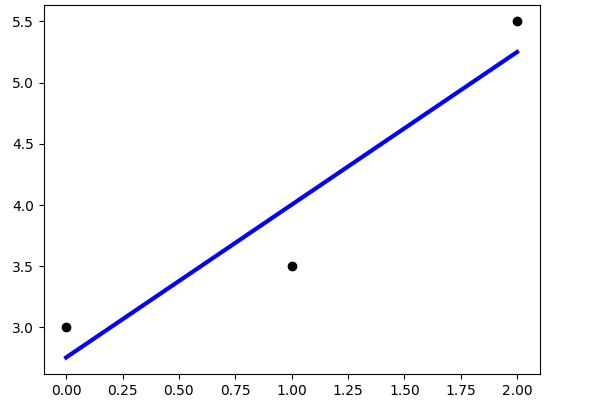
reg.predict([[5]])

plt.scatter(X, y, color='black')

y\_pred = reg.predict(X)

plt.plot(X, y\_pred, color='blue', linewidth=3)

plt.show()



import matplotlib.pylab as plt

from sklearn import linear\_model

reg = linear\_model.LinearRegression()

X = [[174], [152], [138], [128], [186]]

y = [71, 55, 46, 38, 88]

reg.fit(X, y) # 학습

print(reg.predict([[165]]))

# 학습 데이터와 y 값을 산포도로 그린다.

plt.scatter(X, y, color='black')

# 학습 데이터를 입력으로 하여 예측값을 계산한다.

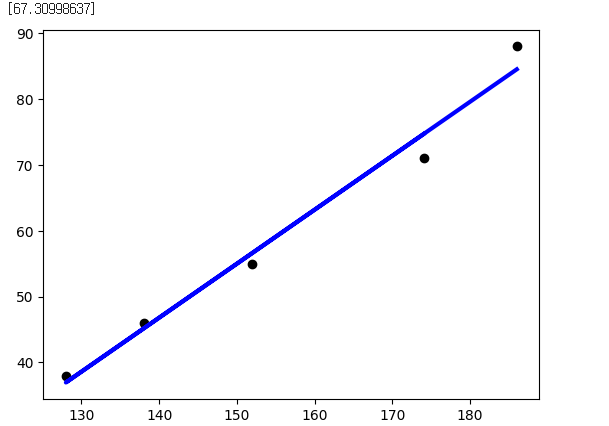
y\_pred = reg.predict(X)

# 학습 데이터와 예측값으로 선그래프로 그린다.

# 계산된 기울기와 y 절편을 가지는 직선이 그려진다.

plt.plot(X, y\_pred, color='blue', linewidth=3)

plt.show()



import matplotlib.pylab as plt

import numpy as np

from sklearn.linear\_model import LinearRegression

from sklearn import datasets

# 당뇨병 데이터 세트를 적재한다.

diabetes\_X, diabetes\_y = datasets.load\_diabetes(return\_X\_y=True)

# 하나의 특징(BMI)만 추려내서 2차원 배열로 만든다. BMI 특징의 인덱스가 2이

diabetes\_X\_new = diabetes\_X[:, np.newaxis, 2]

# 학습 데이터와 테스트 데이터를 분리한다.

from sklearn.model\_selection import train\_test\_split

X\_train, X\_test, y\_train, y\_test = train\_test\_split(diabetes\_X\_new, diabetes\_y, test\_size=0.1, random\_state=0)

regr = linear\_model.LinearRegression()

regr.fit(X\_train, y\_train)

# 테스트 데이터로 예측해보자.

y\_pred = regr.predict(X\_test)

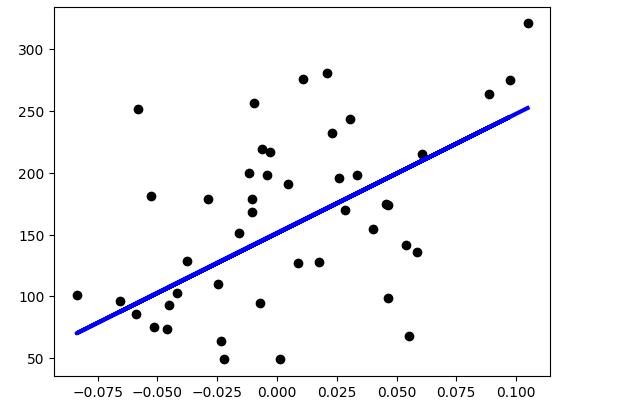
# 실제 데이터와 예측 데이터를 비교해보자.

#plt.plot(y\_test, y\_pred, '.')

plt.scatter(X\_test, y\_test, color='black')

plt.plot(X\_test, y\_pred, color='blue', linewidth=3)

plt.show()



from sklearn.model\_selection import train\_test\_split

from sklearn.linear\_model import LinearRegression

from sklearn.metrics import mean\_squared\_error , r2\_score

import pandas as pd

import seaborn as sns

import numpy as np

import matplotlib.pyplot as plt

from sklearn.datasets import fetch\_california\_housing

california = fetch\_california\_housing()

y=california.target

df\_california=pd.DataFrame(california.data, columns=california.feature\_names)

df\_california

X\_train , X\_test , y\_train , y\_test = train\_test\_split(df\_california , y, test\_size=0.2, random\_state=123)

model\_lr = LinearRegression()

model\_lr.fit(X\_train ,y\_train )

y\_preds = model\_lr.predict(X\_test)

mse = mean\_squared\_error(y\_test, y\_preds)

rmse = np.sqrt(mse)

fig, axs = plt.subplots(figsize=(16,8) , ncols=4 , nrows=2)

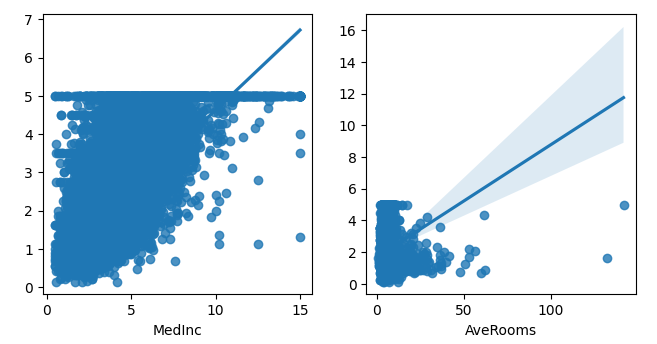
lm\_features = ['MedInc','AveRooms']

for i , feature in enumerate(lm\_features):

row = int(i/4)

col = i%4

sns.regplot(x=feature , y=y,data=df\_california , ax=axs[row][col])



중간 소득과 평균 방의 수와 가격의 연관관계를 선형회귀로 표현