4주. Regression			
학번	3 2 1 7 0 5 7 8	이름	김산

※ 이번 실습에 사용된 데이터셋은 공지에 있는 데이터셋 압축파일에 포함되어 있음

BostonHousing 데이터셋은 보스턴 지역의 지역정보 및 평균주택 가격 (medv) 정보를 담고 있다.

BostonHousing dataset을 가지고 단순 선형 회귀 분석을 하고자 한다.

Q1 lstat (소득분위가 하위인 사람들의 비율) 로 medv (주택가격)을 예측하는 단순 선형회귀 모델을 만드시오 (tain, test 나누지 않음). 모델의 내용을 보이시오

Source code :

```
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd

from sklearn.linear_model import LinearRegression

dataset = pd.read_csv("./Data/BostonHousing.csv")
lstat = dataset['lstat']
medv = dataset['medv']

lstat = np.array(lstat).reshape(lstat.size, 1);
medv = np.array(medv).reshape(medv.size, 1);

# Define learning method
model = LinearRegression()

# Train the model using lstat set
model.fit(lstat, medv)

print('Coefficients : {0:.2f}, Intercept {1:.3f}'\
    .format(model.coef_[0][0], model.intercept_[0]))
```

실행화면 캡쳐:

```
Coefficients : -0.95, Intercept 34.554
```

Q2. 모델에서 만들어진 회귀식을 쓰시오 (medv = W x lstat + b 의 형태)

$$y = -0.95x + 34.554$$

Q3. 회귀식을 이용하여 lstat 의 값이 각각 2.0, 3.0, 4.0, 5.0 일 때 medv 의 값을 예측하여 제시하시오.

Source code :

```
test = np.array([[2.0],[3.0],[4.0],[5.0]])
model.predict(test)
```

실행화면 캡쳐:

```
array([[32.65374217],
       [31.70369282],
       [30.75364346],
       [29.80359411]])
```

Q4. 데이터셋의 모든 lstat 값을 회귀식에 넣어 mdev 의 값을 예측 한 뒤 mean square error를 계산하여 제시하시오

Source code :

```
pred = model.predict(lstat)
print('mean_squared_error : {0:.2f}' .\
    format(mean_squared_error(medv, pred)))
```

실행화면 캡쳐:

```
mean_squared_error : 38.48
```

BostonHousing dataset을 가지고 다중 선형 회귀 분석을 하고자 한다.

Q5. **lstat** (소득분위가 하위인 사람들의 비율), **ptratio**(초등교사비율), **tax**(세금), **rad**(고속도로접근성)로 **medv** (주택가격)을 예측하는 단순 선형회귀 모델을 만드시오 (tain, test 나누지 않음). 모델의 내용을 보이시오

Source code :

```
df = pd.read_csv("./Data/BostonHousing.csv")

df_x = df[['lstat', 'ptratio', 'tax', 'rad']]
df_y = df[['medv']]

# Define learning method
model = LinearRegression()

#Train the model using lstat set
model.fit(df_x, df_y)

print('Coefficients : {0:.2f},{1:.2f},{2:.2f},{3:.2f}, Intercept {4:.3f}'\
    .format(model.coef_[0][0],model.coef_[0][1],model.coef_[0][2],model.coef_[0][3],\
       model.intercept_[0]))
```

실행화면 캡쳐:

```
Coefficients : -0.81,-1.23,-0.02,0.33, Intercept 58.546
```

Q6. 모델에서 만들어진 회귀식을 쓰시오

$$medv = -0.81lstat + -1.23ptratio + -0.02tax + 0.33rad + 58.546$$

Q7. lstat, ptratio, tax, rad 의 값이 다음과 같을 때 mdev 의 예측값을 보이시오.

lstat	ptratio	tax	rad
2.0	14	296	1
3.0	15	222	2
4.0	15	250	3

Source code :

```
test = np.array([[2.0, 14, 296, 1],[3.0, 15, 222, 2],[4.0, 15, 250,
3]]) . reshape(3,-1)
model.predict(test)
```

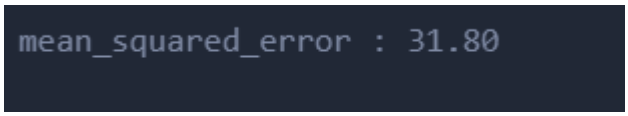
실행화면 캡쳐:

```
array([[35.5479738 ],
       [34.95427204],
       [34.04856204]])
```

Q8. 데이터셋의 모든 lstat, ptratio, tax, rad 값을 회귀식에 넣어 mdev 의 값을 예측한 뒤 mean square error를 계산하여 제시하시오.

```
pred = model.predict(df_x)
print('mean_squared_error : {0:.2f}' .\
    format(mean_squared_error(df_y, pred)))
```

실행화면 캡쳐:



Q9. lstat 하나만 가지고 모델을 만든 경우와 4개 변수를 가지고 모델을 만든 경우 어느쪽이 더 좋은 모델이라고 할수 있는가? 그 이유는?

4개의 변수를 가지고 예측모델을 만든 경우 mean squared error가 더 낮기 때문에 4 개의 변수를 가지고 만든 모델이 1 개의 변수를 가지고 만든 모델 보다 더 좋은 모델이라 할 수 있습니다 .

ucla_admit.csv 파일은 미국 UCLA 의 대학원 입학에 대한 정보를 담고 있다. 컬럼(변수)에 대한 설명은 다음과 같다.
admit : 합격여부 (1:합격, 0:불합격)
gre : GRE 점수
gpa : GPA 점수
rank : 성적 석차

이 데이터셋에 대해 다음의 문제를 해결하시오

Q10. **gre**, **gpa**, **rank**를 가지고 합격여부를 예측하는 logistic regression 모델을 만드시오. (train, test를 나누되 test 의 비율은 30% 로 하고 random_state 는 1234 로 한다)

```
train_X, test_X, train_y, test_y = \
    train_test_split(df_x,df_y,test_size=0.3,random_state=1234)

model = LogisticRegression()
model.fit(train_X, train_y)

print('Coefficients : {0:.3f},{1:.3f},{2:.3f}, Intercept {3:.3f}'\
    .format(model.coef_[0][0],model.coef_[0][1],model.coef_[0][2],\
        model.intercept_[0]))
```

실행화면 캡쳐:

```
Coefficients : 0.001,0.150,-0.668, Intercept -0.073
```

Q11. 모델을 테스트 하여 training accuracy 와 test accuracy를 보이시오

```
train_pred_y = model.predict(train_X)
test_pred_y = model.predict(test_X)

train_acc = accuracy_score(train_y, train_pred_y)
test_acc = accuracy_score(test_y, test_pred_y)

print('Training Accuracy : {0:3f}'.format(train_acc))
print('Test Accuracy : {0:3f}'.format(test_acc))
```

실행화면 캡쳐:

```
Training Accuracy : 0.671429
Test Accuracy : 0.741667
```

Q12. **gre, gpa, rank 가 다음과 같을 때 합격 여부를 예측하여 보이시오**

gre	gpa	rank
400	3.5	5
550	3.8	2
700	4.0	2

```
test2_X = np.array([[400, 3.5, 5],[550, 3.8, 2],[700, 4.0, 2]]).reshape(3,-1)
test2_y = model.predict(test2_X)
test2_y
```

실행화면 캡쳐:

```
array([0, 0, 0])
```

Q13.이번에는 **gre, gpa만 가지고 합격 여부를 예측하는 모델을 만드시오**
(train, test를 나누되 test 의 비율은 30% 로 하고 random_state 는 1234 로 한다)

```
df = pd.read_csv("./Data/ucla_admit.csv")
df_x = df[['gre','gpa']]
df_y = df['admit']

train_X, test_X, train_y, test_y = \
    train_test_split(df_x,df_y,test_size=0.3,random_state=1234)

model = LogisticRegression()
model.fit(train_X, train_y)

print('Coefficients : {0:.3f},{1:.3f}, Intercept {2:.3f}'\
    .format(model.coef_[0][0],model.coef_[0][1],\
        model.intercept_[0]))
```

실행화면 캡쳐:

```
Coefficients : 0.002,0.504, Intercept -3.369
```

Q14. 모델을 테스트 하여 training accuracy 와 test accuracy를 보이시오

```
train_pred_y = model.predict(train_X)
test_pred_y = model.predict(test_X)

train_acc = accuracy_score(train_y, train_pred_y)
test_acc = accuracy_score(test_y, test_pred_y)

print('Training Accuracy : {0:3f}'.format(train_acc))
print('Test Accuracy : {0:3f}'.format(test_acc))
```

실행화면 캡쳐:

```
Training Accuracy : 0.625000
Test Accuracy : 0.825000
```

Q15. 3가지 변수로 모델을 만든 경우와 2가지 변수로 모델을 만든 경우를 비교하여 어떤 모델이 더 좋은 모델인지 자신의 의견을 제시하시오

두 경우 모두 training Accuracy에서는 큰 차이를 보이지 않지만 , 2 가지 변수로 모델을 만든 경우에서 test Accuracy가 크게 앞서기 때문에 2 가지 변수로 만든 ㅂ 모델이 더 좋은 모델인 것 같습니다 .