



신경망 모델 실습

인공지능 RPA – 11주차

학습 내용

1. 회귀분석 결과의 평가
2. 인공신경망
3. QR Code 만들기



인공지능 실습

_복습

복습 문제 : ch5-1.csv 독립변수 중 food에 대한 회귀분석을 실시하고 그래프를 그리시오
(종속변수는 weight)

OLS Regression Results						
=====						
Dep. Variable:	weight	R-squared:	0.770			
Model:	OLS	Adj. R-squared:	0.762			
Method:	Least Squares	F-statistic:	93.81			
Date:	Sat, 16 Nov 2024	Prob (F-statistic):	1.94e-10			
Time:	10:06:17	Log-Likelihood:	-78.279			
No. Observations:	30	AIC:	160.6			
Df Residuals:	28	BIC:	163.4			
Df Model:	1					
Covariance Type:	nonrobust					
=====						
	coef	std err	t	P> t	[0.025	0.975]

Intercept	78.1551	6.169	12.669	0.000	65.519	90.791
food	4.6684	0.482	9.686	0.000	3.681	5.656
=====						
Omnibus:	1.785	Durbin-Watson:	2.054			
Prob(Omnibus):	0.410	Jarque-Bera (JB):	1.239			
Skew:	0.238	Prob(JB):	0.538			
Kurtosis:	2.125	Cond. No.	128.			
=====						



_실습

```
import pandas as pd
import matplotlib.pyplot as plt
import statsmodels.formula.api as smf
```

```
w = pd.read_csv('ch5-1.csv')
w_n = w.iloc[:,1:5]
```

```
model_lm = smf.ols(_____)
```

```
result_lm = model_lm.fit()
```

```
result_lm.summary()
```

```
print(result_lm.summary())
```



_실습

```
import pandas as pd
import matplotlib.pyplot as plt
import statsmodels.formula.api as smf

w = pd.read_csv('ch5-1.csv')
w_n = w.iloc[:,1:5]

model_lm = smf.ols(formula = 'weight ~ food', data = w_n)

result_lm = model_lm.fit()

result_lm.summary()

print(result_lm.summary())
```



_회귀 분석 결과 평가

모델이 얼마나 쓸 만한지 평가하기

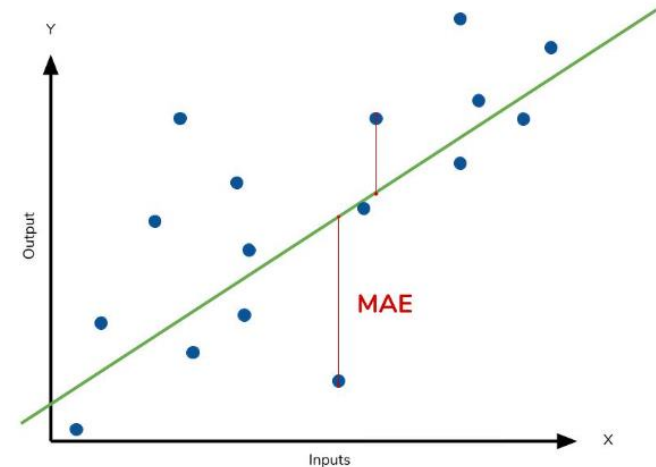
MAE(Mean of Absolute Error) : 모델의 예측값과 실제값의 차이의 절대값의 평균
- 절대값을 취하기 때문에 가장 직관적으로 알 수 있는 지표이다. (해석에 용이하다.)

MSE(Mean of Squared Error) : 모델의 예측값과 실제값의 차이의 제곱값의 평균
- 제곱을 하기 때문에 MAE와는 다르게 모델의 예측값과 실제값 차이의 면적의(제곱)합 (평균제곱오차)

RMSE(Root Mean Squared Error) : MSE에 루트를 씌워 사용한다.
- RMSE를 사용하면 오류 지표를 실제값과 유사한 단위로 다시 변환하여 해석을 쉽게한다.

R-squared (Coefficient of determination, 결정계수) : R-squared는 현재 사용하고 있는 x변수가 y변수의 분산을 얼마나 줄였는가이다.

- y평균값 모델(기준모델)을 사용했을 때 대비 우리가 가진 x변수를 사용함으로써 얻는 성능 향상의 정도
- 값이 **1에 가까우면 데이터를 잘 설명하는 모델**이고 0에 가까울수록 설명을 못하는 모델이라고 생각할 수 있다.



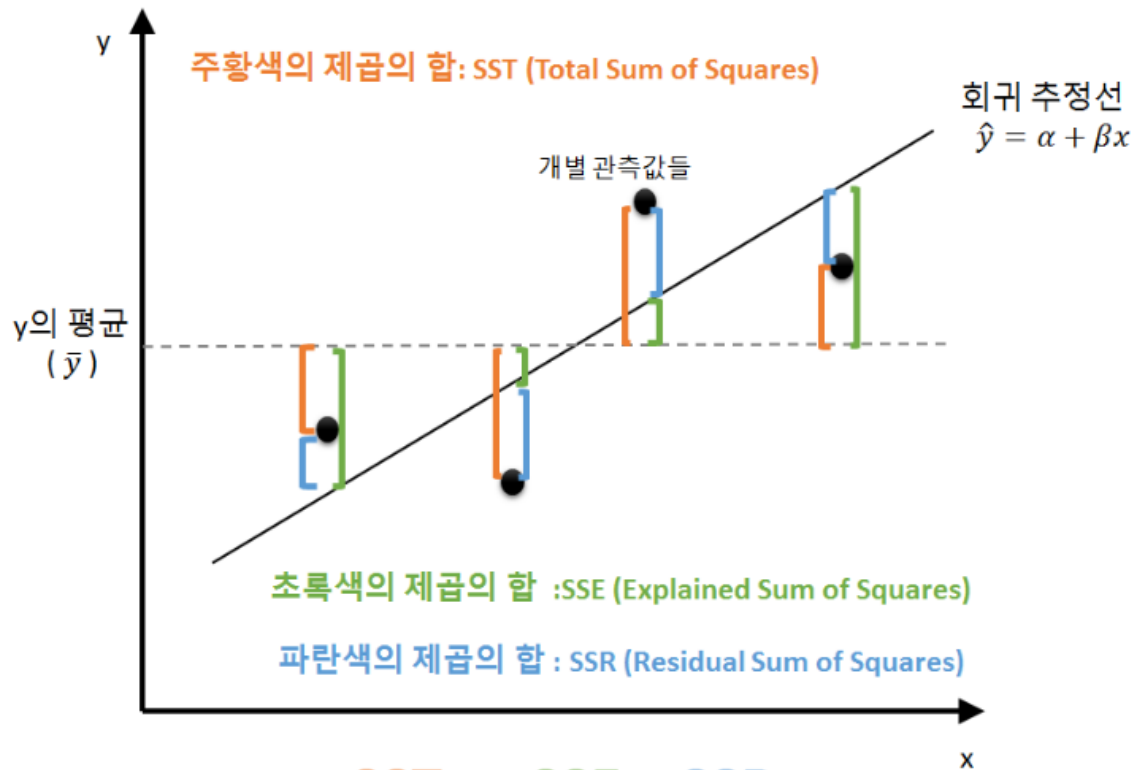
v "

$$R^2 = 1 - \frac{\sum (y_i - \hat{y}_i)^2}{\sum (y_i - \bar{y})^2} \quad R^2 = \frac{SSR}{SST} = 1 - \frac{SSE}{SST}$$



_회귀 분석 결과 평가

실제값, 예측값, 평균



$$SST = SSE + SSR$$

총변동 = 설명된 변동 + 설명안된 변동

$$SST = \sum (y_i - \bar{y})^2$$

$$SSR = \sum (y_i - \hat{y}_i)^2$$

$$SSE = \sum (\hat{y}_i - \bar{y})^2$$

$$\sum_{i=1}^n (y_i - \bar{y})^2 = \sum_{i=1}^n (y_i - \hat{y}_i)^2 + \sum_{i=1}^n (\hat{y}_i - \bar{y})^2$$

SST

SSE

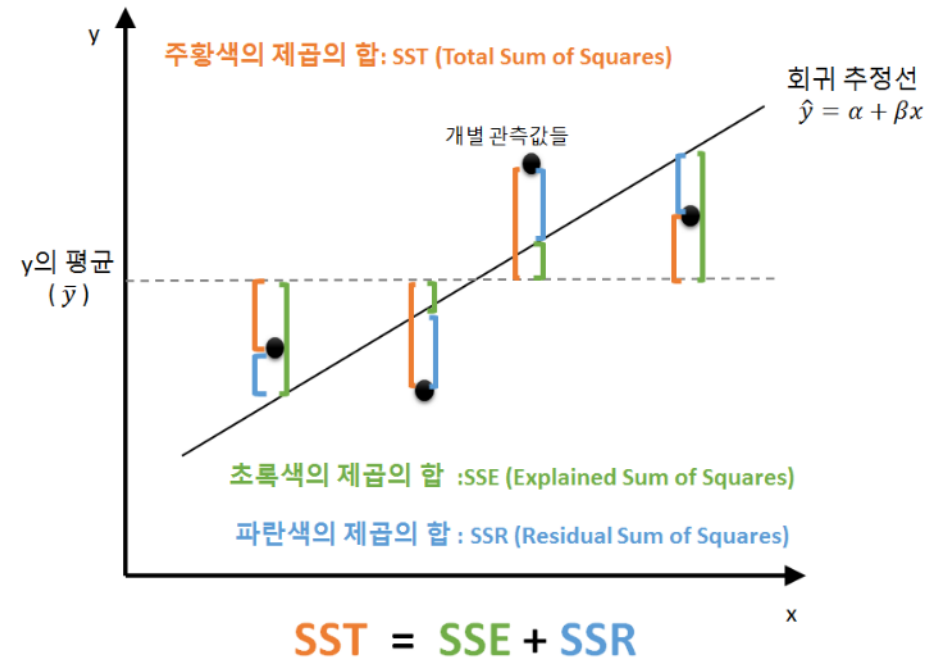
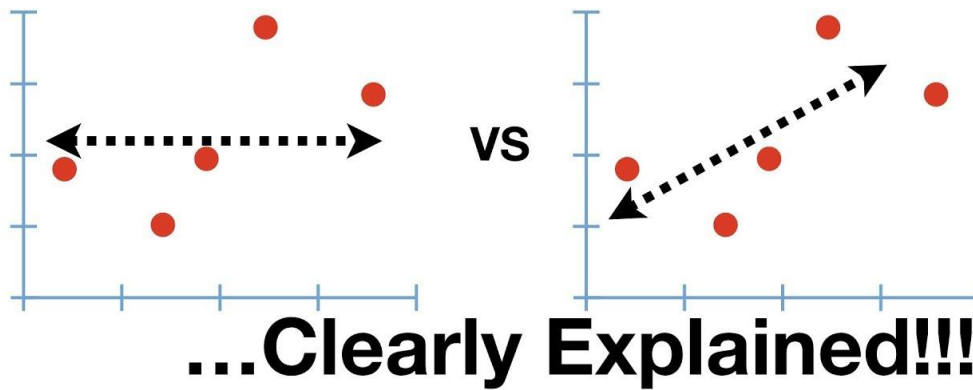
SSR



_회귀 분석 결과 평가

$$R^2 = 1 - \frac{\sum (y_i - \hat{y}_i)^2}{\sum (y_i - \bar{y})^2} \quad R^2 = \frac{SSR}{SST} = 1 - \frac{SSE}{SST}$$

R^2 (R squared)....



회귀분석 프로그래밍 실습 : 결과 평가

라이브러리 설치 : `pip install scikit-learn`

```
from sklearn.metrics import mean_squared_error, mean_absolute_error, r2_score
import numpy as np
```

```
predicted_values = result_lm.predict()
```

```
mse = mean_squared_error(w_n['weight'], predicted_values)
```

```
mae = mean_absolute_error(w_n['weight'], predicted_values)
```

```
rmse = np.sqrt(mse)
```

```
r_squared = r2_score(w_n['weight'], predicted_values)
```

```
print("Mean Squared Error (MSE):", mse)
```

```
print("Mean Absolute Error (MAE):", mae)
```

```
print("Root Mean Squared Error (RMSE):", rmse)
```

```
print("R-squared:", r_squared)
```



회귀분석 프로그래밍 실습 : 결과 평가

[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

Mean Squared Error (MSE): 10.812834224598946

Mean Absolute Error (MAE): 2.713012477718354

Root Mean Squared Error (RMSE): 3.2882874303501732

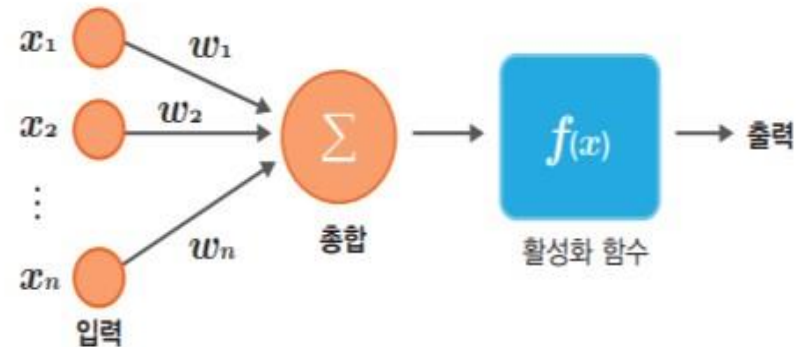
R-squared: 0.7701353268580156



_인공신경망 개념

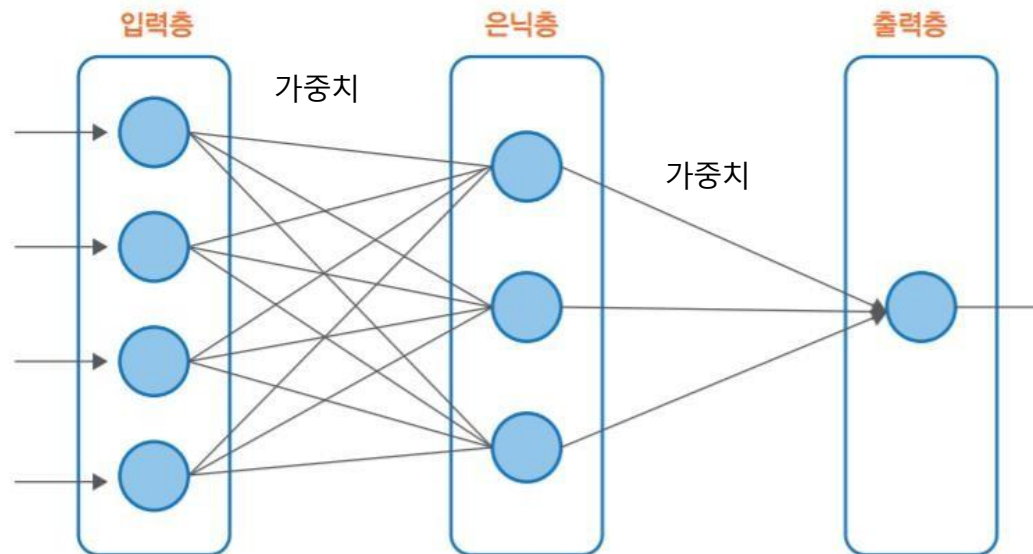
인공신경망(ANN - Artificial Neural Network)

- 인간의 뇌에는 수많은 뉴런이 존재하고 그 뉴런들은 시냅스로 서로 연결되어 있는데, 이를 신경망(Neural Network)이라고 부름
- 인공신경망은 사람의 신경망 구조에서 착안해 만들어졌기 때문에 뉴런들의 연결, 즉 신경망을 인공적으로 흉내 낸 것



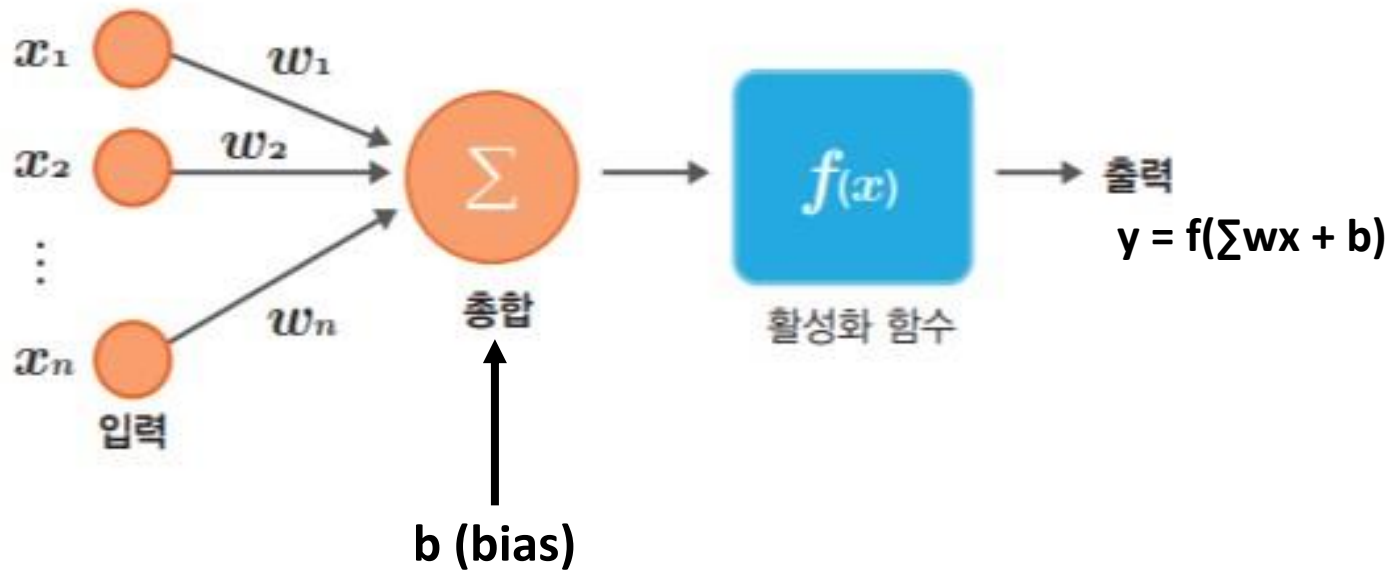
_다층 퍼셉트론 (MLP)

- 퍼셉트론을 여러 층 쌓아 올린 다층퍼셉트론 Multi-Layer Perceptron(MLP)
= 인공신경망(Artificial Neural Networks)
- 다층 퍼셉트론의 구조는 입력층Input Layer과 은닉층Hidden Layer, 출력층Output Layer으로 구성
- 은닉층은 데이터의 입출력 과정에서 직접적으로 보이지 않지만 숨겨진 특징을 학습하는 역할을 함



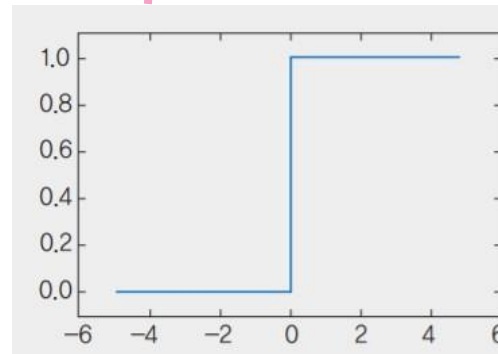
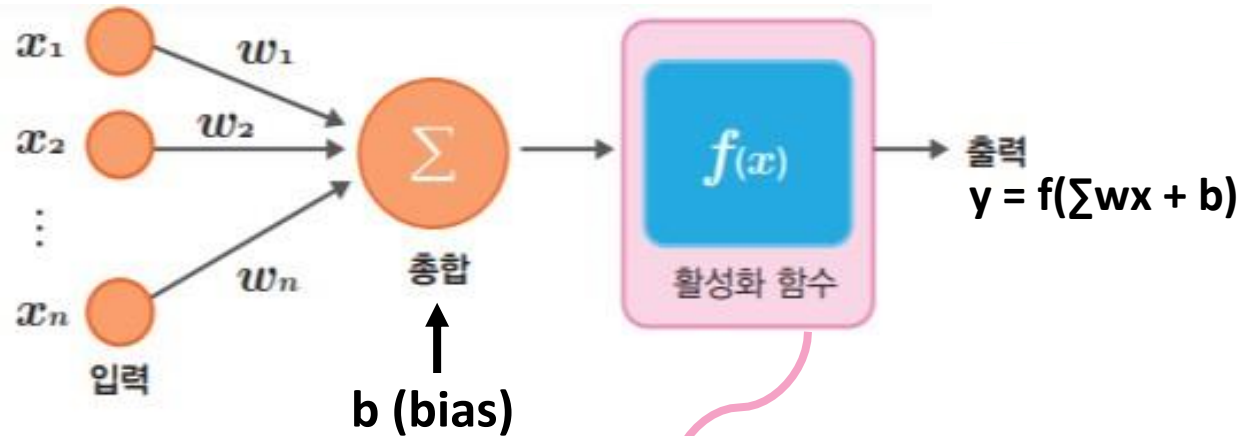
_인공신경망 : 계산 방법

1. 입력 층에서 데이터 값을 입력 받음 : x
2. 입력의 중요도에 따라 가중값 지정 : w
3. 입력 x 와 가중 값 w 를 곱하여 누적한 값에 바이어스 b 를 더함



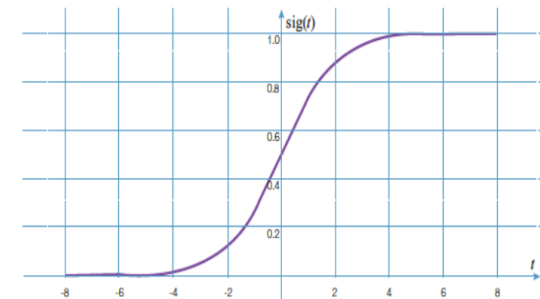
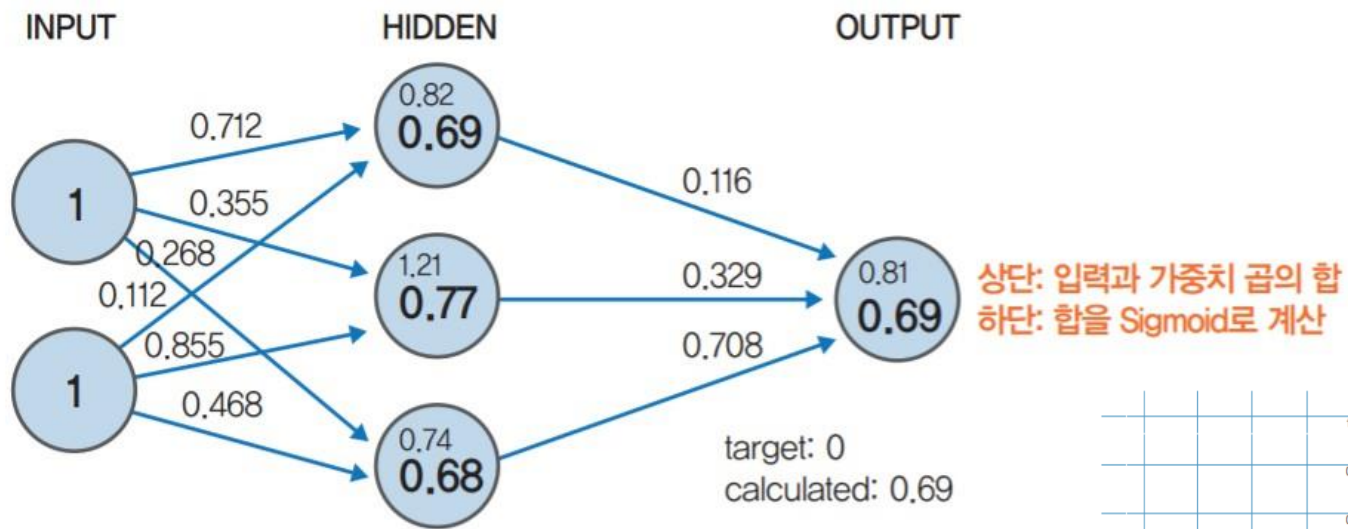
_인공신경망 : 계산 방법

4. 합한 값을 활성화 함수로 출력값을 계산
(계단함수 활용)



_다층 퍼셉트론 계산 방법

여러 개의 입력값과 가중치 곱의 합을 활성화 함수로 계산해 여러 개 뉴런에서 출력



_인공신경망 프로그래밍 실습

1. 데이터 로딩

- import pandas as pd
- w = pd.read_csv("ch7-1.csv")

2. 데이터 분할

- 독립변수(원인) / 종속변수(결과) 분할

- from sklearn.model_selection import train_test_split
- x_data = w.iloc[:,0:2].values
- y_data = w.iloc[:,2].values

- 학습데이터 / 시험데이터 분할

- x_train, x_test, y_train, y_test = train_test_split(x_data, y_data, test_size=0.2)
- print(len(pd.DataFrame(x_train)), len(pd.DataFrame(x_test)), end='\n\n')
- print(len(pd.DataFrame(y_train)), len(pd.DataFrame(y_test)), end='\n\n')



_인공신경망 프로그래밍 실습

3. 학습 수행 : 모델 구축

- from sklearn.neural_network import MLPRegressor
- model_mlp = **MLPRegressor().fit**(x_train, y_train)
- print(model_mlp.get_params(), end='\n\n')

4. 예측값 생성

- y_pred_mlp = **model_mlp.predict(x_test)**
- print(y_train, end='\n\n')
- print(y_pred_mlp, end='\n\n')

	egg_weight	acc_food	real	predict
0	52	9940	4060	3928.411748
1	64	10436	4098	4124.209883
2	56	9940	4074	3928.313133
3	66	10208	4114	4034.020601
4	75	10510	4130	4153.194649
5	71	10384	4127	4103.479067
6	71	10540	4134	4165.153787
7	69	10409	4135	4113.412144
8	57	9982	4079	3944.893211
9	63	10346	4102	4088.652968
10	72	10494	4131	4146.942998

5. 데이터 비교 확인

- df_x_test = pd.DataFrame(x_test, columns=['egg_weight', 'acc_food'])
- df_y_pred = pd.DataFrame(y_pred_mlp, columns=['predict'])
- df_y_test = pd.DataFrame(y_test, columns=['real'])
- df = pd.concat([df_x_test, df_y_test, df_y_pred], axis=1)
- print(df, end='\n\n')



_인공신경망 프로그래밍 실습

6. 모델 성능 확인 : 회귀성능 지표 계산

- `from sklearn.metrics import mean_squared_error, mean_absolute_error, r2_score`
- `R2 = r2_score(y_test, y_pred_mlp)`
- `print("R2 = ", R2, end='\n\n')`

```
R2 = -27.364223504883803
```



RPA 실습

_QR코드 만들기 프로그래밍

1. 라이브러리 설치 및 불러오기

- pip install qrcode
- import qrcode

2. QR코드 생성할 문자열

```
qr_data = "www.naver.com"
```

3. QR코드 이미지 생성

```
qr_img = qrcode.make(qr_data)
```

4. 이미지 저장

```
경로 지정 : save_path = 'qr_data.png'  
저장 : qr_img.save(save_path)
```



_QR코드 만들기 프로그래밍 실습

```
import qrcode # pip install qrcode
```

```
qr_data = 'www.naver.com'  
qr_img = qrcode.make(qr_data)
```

```
save_path = 'qr_data.png'  
qr_img.save(save_path)
```



_QR코드 만들기 프로그래밍 실습2

실습 : 자신의 학번, 이름, 전공 정보를 입력 받아 그 정보에 대한 QR코드 만들기
my_info_data.png로 저장

