



회귀 분석

인공지능 RPA – 10주차

학습 내용

1. 회귀분석
2. FastAPI - 파일 다운로드

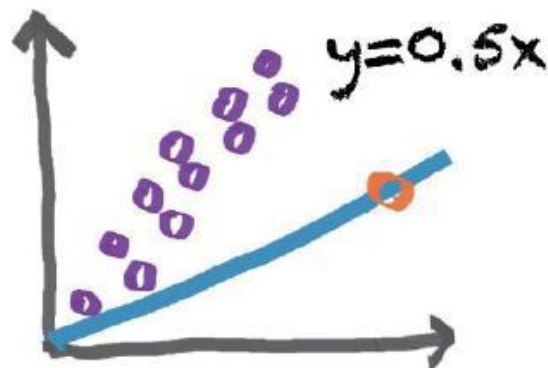
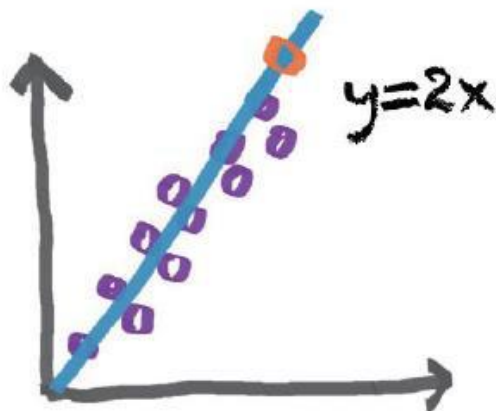


인공지능 실습

_선형 회귀 분석

★ 회귀분석 (Regression Test)

- 회귀분석은 독립변인이 종속변인에 영향을 미치는지 알아보고자 할 때 실시하는 분석방법
- 단순 선형 회귀분석은 독립변수 X(설명변수)에 대하여 종속변수 Y(반응변수)들 사이의 관계를 수학적 모형을 이용하여 규명하는 것
- 규명된 함수식을 이용하여 설명변수들의 변화로부터 종속변수의 변화를 예측하는 분석
- 선형 회귀 모델은 가장 단순하고 학습 속도가 빠르며 인간이 이해하기 쉬워 분류하는 요인의 수가 적을 때는 활용하기가 용이



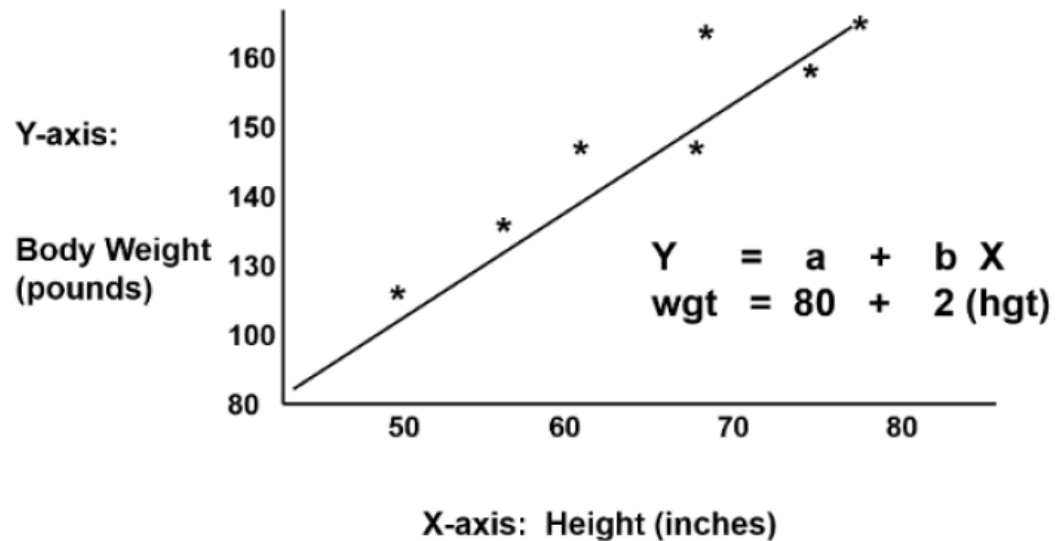
_선형 회귀 분석

'키(Height)에 따른 몸무게(Weight)' 를 예로 들면,

$$\text{Weight} = a + b * \text{Height}$$

가 되며, 결국 Height에 따라 Weight가 결정되므로, Height는 독립변수, Weight는 종속변수.

키(height) inch	몸무게(weight) pound
50	120
58	130
61	145
69	145
70	165
75	155
78	165



파이썬 회귀 분석 프로그래밍

1. 라이브러리 불러오기

```
import statsmodels.formula.api as smf
```

2. 데이터 읽기 및 선별

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

3. 회귀모델 알고리즘 설정

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

- * 파라미터 formula는 '예측하고자 하는 칼럼 이름 ~ 원인이 되는 칼럼 이름'을 설정
- * data 에는 분석할 DataFrame을 넣어주면 된다.

4. 모델링 수행(학습)

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

5. 결과 확인

```
result_lm.summary()
```

6. 예측값 계산

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



회귀분석 프로그래밍 실습 : 회귀 분석

```
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 ~ egg_weight', data = w_n)
```

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

```
result_lm.summary()
```

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



_회귀 분석 결과 해석

OLS Regression Results						
=====						
Dep. Variable:	weight	R-squared:	0.916	3		
Model:	OLS	Adj. R-squared:	0.913			
Method:	Least Squares	F-statistic:	306.0			
Date:	Sat, 21 Oct 2023	Prob (F-statistic):	1.32e-16	1		
Time:	15:16:04	Log-Likelihood:	-63.148			
No. Observations:	30	AIC:	130.3			
Df Residuals:	28	BIC:	133.1			
Df Model:	1					
Covariance Type:	nonrobust					
=====						
	coef	std err	t	P> t	[0.025	0.975]

Intercept	-14.5475	8.705	-1.671	0.106	-32.380	3.285
egg_weight	2.3371	0.134	17.493	0.000	2.063	2.611
=====						
Omnibus:	15.078	Durbin-Watson:	1.998			
Prob(Omnibus):	0.001	Jarque-Bera (JB):	2.750			
Skew:	0.032	Prob(JB):	0.253			
Kurtosis:	1.518	Cond. No.	1.51e+03			
=====						

Notes:

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

[2] The condition number is large, 1.51e+03. This might indicate that there are strong multicollinearity or other numerical problems.



회귀 분석 결과 해석

1. F통계량 - Prob(F-statistic) : 만들어진 결과가 의미가 있는 가 (유의미 한 가)
유의수준 (보통 95%) 이하 경우 유의미 한 것으로 판단

2. 개별 독립변수의 p값 - $P>[t]$: 독립변수가 미치는 영향력은 얼마나 되는 가
유의수준 (보통 95%) 이하 경우 유의미 한 것으로 판단

3. 결정 계수 - R-squared : 회귀식의 판단, 예측 능력이 좋은 가?
0.7 이상이면 우수한 편으로 판단

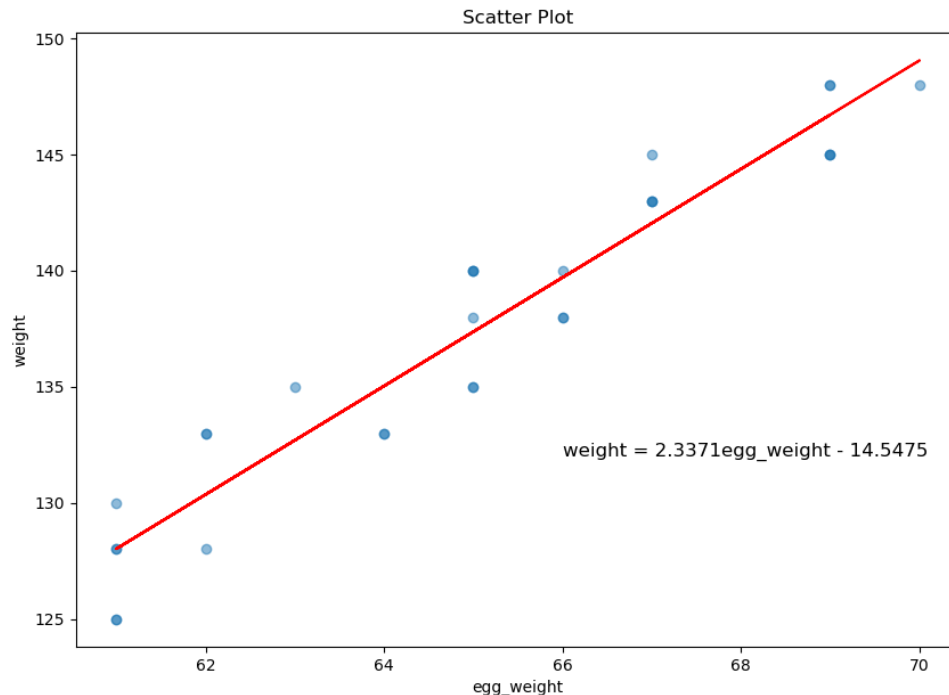
4. 독립변수의 계수 - coef : 결과(변수)에 미치는 영향력
선형 모델에서는 기울기로 나타 남
Intercept는 상수 임 (y절편)

$$[\text{결과}] = [\text{coef값}] \times [\text{독립변수값}] + [\text{Intercept}]$$



_회귀분석 프로그래밍 실습 : 시각화

```
plt.figure(figsize = (10,7))  
plt.scatter(w.egg_weight, w.weight, alpha = .5)  
plt.plot(w.egg_weight, w.egg_weight*2.3371 - 14.5475, color = 'red')  
plt.text(66, 132, 'weight = 2.3371egg_weight - 14.5475', fontsize = 12)  
plt.title('Scatter Plot')  
plt.xlabel('egg_weight')  
plt.ylabel('weight')  
plt.show()
```



_실습

실습 : 실습 1에서 독립변수 중 food 에 대한 회귀분석을 실시하기 (종속변수는 weight)



RPA 실습

_FastAPI

Fast API 파일 다운로드

- 클라이언트 코딩(Front)
- 서버 코딩 (Back)

1. 클라이언트 코딩

: static 폴더안에 html 파일 생성

```
<h2>Download File</h2>
```

```
<input type="text" id="filename" placeholder="Enter filename">
```

```
<button onclick="downloadFile()">Download File</button>
```

Download File

```
<script>
```

```
function downloadFile() {
```

```
    const filename = document.getElementById('filename').value;
```

```
    if (filename.length > 0) {
```

```
        const url = `/files/${filename}`;
```

```
        const a = document.createElement('a');
```

```
        a.href = url;
```

```
        a.download = filename;
```

```
        document.body.appendChild(a);
```

```
        a.click();
```

```
        document.body.removeChild(a);
```

```
    } else {
```

```
        alert('Please enter a filename');
```

```
    }
```

```
}
```

```
</script>
```



FastAPI

2. 서버 코딩 : Form 요청에 대응하는 API

```
from fastapi import File, UploadFile
import shutil
from pathlib import Path
from fastapi.responses import FileResponse
```

```
@app.get("/files/{filename}")
```

```
async def get_file(filename: str):
```

```
    file_path = Path("static/uploads") / filename
```

```
    if file_path.is_file():
```

```
        return FileResponse(path=file_path, filename=filename)
```

```
    else:
```

```
        raise HTTPException(status_code=404, detail="File not found")
```



_FastAPI

3. 업로드 파일 확인

<http://127.0.0.1:8000/filedownload.html>

