제3유형_로지스틱 회귀분석

타이타닉 데이터 불러오기(생존자 예측 데이터)

In [1]:
```
###############  복사 영역  ###############
# 데이터 불러오기
import pandas as pd
import numpy as np

# Seaborn의 내장 타이타닉 데이터셋을 불러옵니다.
import seaborn as sns
df = sns.load_dataset('titanic')
###############  복사 영역  ###############
```

In [2]:
```
print(df.head())
```

```
   survived  pclass     sex   age  sibsp  parch     fare embarked  class  \
0         0       3    male  22.0      1      0   7.2500        S  Third
1         1       1  female  38.0      1      0  71.2833        C  First
2         1       3  female  26.0      0      0   7.9250        S  Third
3         1       1  female  35.0      1      0  53.1000        S  First
4         0       3    male  35.0      0      0   8.0500        S  Third

     who  adult_male deck  embark_town alive  alone
0    man        True  NaN  Southampton    no  False
1  woman       False    C    Cherbourg   yes  False
2  woman       False  NaN  Southampton   yes   True
3  woman       False    C  Southampton   yes  False
4    man        True  NaN  Southampton    no   True
```

In [3]:
```
# 분석 데이터 설정
df = df[ ['survived','sex','sibsp','fare'] ] # sex:성별, sibsp:탑승한 부모 및 자녀 수, fare:요금

print(df.head())
```

```
   survived     sex  sibsp     fare
0         0    male      1   7.2500
1         1  female      1  71.2833
2         1  female      0   7.9250
3         1  female      1  53.1000
4         0    male      0   8.0500
```


✔ 회귀식 : P(1일 확률) = 1 / (1+exp(-f(x)))

- f(x) = b0 + b1x1 + b2x2 + b3x3
- ln(P/1-P) = b0 + b1x1 + b2x2 + b3x3

  (P=생존할 확률, x1=sex, x2=sibsp, x3=fare)

In [4]:
```
# 데이터 전처리
# 변수처리
# 문자형 타입의 데이터의 경우 숫자로 변경해준다.
# *** 실제 시험에서 지시사항을 따를 것 ***

# 성별을 map 함수를 활용해서 각각 1과 0에 할당한다.(여성을 1, 남성을 0)
# (실제 시험의 지시 조건에 따를 것)
df['sex'] = df['sex'].map({'female': 1,
                           'male': 0 })
print(df.head())
```

```
   survived  sex  sibsp     fare
0         0    0      1   7.2500
1         1    1      1  71.2833
2         1    1      0   7.9250
3         1    1      1  53.1000
4         0    0      0   8.0500
```

In [5]:
```
print(df.info())
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 891 entries, 0 to 890
Data columns (total 4 columns):
 #   Column    Non-Null Count  Dtype
---  ------    --------------  -----
 0   survived  891 non-null    int64
 1   sex       891 non-null    int64
 2   sibsp     891 non-null    int64
 3   fare      891 non-null    float64
dtypes: float64(1), int64(3)
memory usage: 28.0 KB
None
```


1. sklearn 라이브러리 활용

1. sklearn 라이브러리 활용

In [6]:
```
# 독립변수와 종속변수 설정
x = df.drop(['survived'], axis=1)  # x = df [ ['sex','age','fare'] ]
y = df['survived']
```

(주의) LogisticRegression() 객체안에 반드시 penalty= None 으로 입력해야 함

In [7]:
```
# 모델링
from sklearn.linear_model import LogisticRegression  # 회귀는 LinearRegression

# 반드시 penalty= None 으로 입력할 것해야 함, default='l2'
model1 = LogisticRegression(penalty= None)
model1.fit(x, y)
```


Out[7]:
```
LogisticRegression(penalty=None)
```

In [8]:
```
# 로지스틱회귀분석 관련 지표 출력

# 1. 회귀계수 출력 : model.coef_
print(np.round(model1.coef_, 4) )        # 전체 회귀계수
print(np.round(model1.coef_[0,0], 4) )  # x1 의 회귀계수
print(np.round(model1.coef_[0,1], 4) )  # x2 의 회귀계수
print(np.round(model1.coef_[0,2], 4) )  # x3 의 회귀계수

# 2. 회귀계수(절편) : model.intercept_
print(np.round(model1.intercept_, 4) )
```

```
[[ 2.5668 -0.4017  0.0138]]
2.5668
-0.4017
0.0138
[-1.6964]
```

✔ 회귀식 : P(1일 확률) = 1 / (1+exp(-f(x)))

- f(x) = b0 + b1x1 + b2x2 + b3x3
- ln(P/1-P) = b0 + b1x1 + b2x2 + b3x3

  (P=생존할 확률, x1=sex, x2=sibsp, x3=fare) ### 결과 : ln(P/1-P) = -1.6964 + 2.5668sex - 0.4017sibsp + 0.0138fare

In [9]:
```
# 3-1.로지스틱 회귀모형에서 sibsp 변수가 한단위 증가할 때 생존할 오즈가 몇 배 증가하는지
#    반올림하여 소수점 셋째 자리까지 구하시오.

# exp(b2) 를 구하면 된다.
result = np.exp(model1.coef_[0,1]) # 인덱싱 주의하세요.
print(round(result, 3))

# 해석 : sibsp 변수가 한 단위 증가할 때 생존할 오즈가 0.669배 증가한다.
#        생존할 오즈가 33% 감소한다. (생존할 확률이 감소한다)
```

```
0.669
```

In [10]:
```
# 3-2.로지스틱 회귀모형에서 여성일 경우 남성에 비해 오즈가 몇 배 증가하는지
#    반올림하여 소수점 셋째 자리까지 구하시오.

# exp(b1) 를 구하면 된다.
result2 = np.exp(model1.coef_[0,0]) # 인덱싱 주의하세요.
print(round(result2, 3))

# 해석 : 여성일 경우 남성에 비해 생존할 오즈가 13.024배 증가한다.
#        생존할 오즈가 13배 증가한다. (생존할 확률이 증가한다)
```

```
13.024
```

2. statsmodels 라이브러리 사용

(주의) 실제 오즈가 몇 배 증가했는지 계산하는 문제가 나온다면

sklearn 라이브러리를 사용하여 회귀계수를 직접구해서 계산할 것(소수점이 결과값에 영향을 줄 수 있음!)

In [11]:
```
# 모델링
import statsmodels.api as sm

x = sm.add_constant(x)        # 주의 : 상수항 추가해줘야 함
model2 = sm.Logit(y, x).fit() # 주의할 것 : y, x 순으로 입력해야 함
summary = model2.summary()
print(summary)
```

```
Optimization terminated successfully.
         Current function value: 0.483846
         Iterations 6
                           Logit Regression Results
==============================================================================
Dep. Variable:               survived   No. Observations:                  891
Model:                          Logit   Df Residuals:                      887
Method:                           MLE   Df Model:                            3
Date:                Fri, 10 Nov 2023   Pseudo R-squ.:                  0.2734
Time:                        23:17:02   Log-Likelihood:                -431.11
converged:                       True   LL-Null:                       -593.33
Covariance Type:            nonrobust   LLR p-value:                5.094e-70
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
const         -1.6964      0.129    -13.134      0.000      -1.950      -1.443
sex            2.5668      0.179     14.321      0.000       2.216       2.918
sibsp         -0.4017      0.095     -4.222      0.000      -0.588      -0.215
fare           0.0138      0.003      5.367      0.000       0.009       0.019
==============================================================================
```


(결과 비교해보기) 두 라이브러리 모두 같은 결과값을 출력

- ✔ 회귀식 : P(1일 확률) = 1 / (1+exp(-f(x)))
- f(x) = b0 + b1x1 + b2x2 + b3x3
- ln(P/1-P) = b0 + b1x1 + b2x2 + b3x3
  (P=생존할 확률, x1=sex, x2=sibsp, x3=fare) #### 1. sklearn : ln(P/1-P) = -1.6964 + 2.5668sex - 0.4017sibsp + 0.0138fare #### 2. statsmodel : ln(P/1-P) = -1.6964 + 2.5668sex - 0.4017sibsp + 0.0138fare