로지스틱 회귀를 이용한 사업체 유형과 일 · 가정 양립제도 상관관계 분석

202347001 배예진

https://github.com/daraeyaa/logistic-regression/

목차

1. 문제 정의

「남녀고용평등과 일·가정 양립 지원에 관한 법률」

→ 헌법의 평등이념에 의거해 고용에 있어 남녀의 평등한 기회와 대우를 보장하고,
모성을 보호하고 여성고용을 촉진하여 남녀고용평등을 실현함과 동시에
근로자의 일과 가정의 양립을 지원하여 모든 국민의 삶의 질을 높이는 것을 목적으로 한 법률이다.

→ 제도의 도입실태를 파악하고자 전국의 상시근로자 5인 이상 사업체 중 업종별, 사업체 규모별로 층화 추출된 5,070개 표본사업체의 인사담당자를 대상으로 「2021년 일·가정 양립 실태 조사」를 실시함.

1. 문제 정의

〈표 1〉 업종 및 사업체 규모별 최종 조사 현황

(단위: 개)

업종 \ 사업체 규모	5~9인	10~29인	30~99인	100~299인	300인 이상	전체
전 체	1,603	1,373	1,001	664	429	5,070
광업	31	27	19	4	2	83
제조업	137	128	94	67	46	472
전기, 가스, 증기 및 공기 조절 공급업	25	23	32	21	16	117
수도, 하수 및 폐기물 처리, 원료 재생업	60	57	45	24	3	189
건설업	119	97	69	47	34	366
도매 및 소매업	169	132	75	53	31	460
운수 및 창고업	71	77	60	43	26	277
숙박 및 음식점업	180	117	59	36	22	414
정보통신업	75	67	56	42	29	269
금융 및 보험업	74	78	69	40	29	290
부동산업	95	85	40	31	21	272
전문, 과학 및 기술 서비스업	97	93	72	48	35	345
사업시설 관리, 사업 지원 및 임대 서비스업	68	69	51	44	33	265
교육 서비스업	112	81	85	47	29	354
보건업 및 사회복지 서비스업	115	106	84	55	38	398
예술·스포츠 및 여가관련 서비스업	79	57	32	29	15	212
협회 및 단체·수리 및 기타 개인 서비스업	96	79	59	33	20	287

→ 업종별, 규모별 추출된 5,070 개의 표본사업체

→ 주제 : **로지스틱 회귀를 이용한 사업체 유형으로 일 · 가정 양립제도 사용 여부 예측하기**

2. 사용한 데이터

- 고용노동부의 2021년 일 · 가정 양립 실태조사 데이터
  https://www.data.go.kr/data/15100310/fileData.do

- 실제 사용된 조사표, CSV, CSV 항목 및 코드집 파일을 받을 수 있음.

- 조사 내용

사업체 일반현황, 임신 및 출산지원제도의 활용 실태, 육아 등 돌봄지원제도의 활용 실태,
유연근로 및 일하는 문화, 남녀고용 평등기회 등 5개 영역의 총 96개 문항으로 구성

2. 사용한 데이터

- CSV

	A	B	업종	규모	지역	AE	AF	AG
1	아이디	리스트아이디	info1	info2	info3	b1a11	b1a12	b1a13
2	61232	100003	8	1	16	1	3	
3	62016	100004	15	2	3	2	1	2
4	30281	100005	3	5	10	1	1	2
5	30582	100006	10	1	1	4		
6	61299	100009	2	1	1	1	1	1
7	61426	100010	8	4	9	1	2	2
8	62015	100013	14	4	1	1	1	1
9	62003	100014	15	1	16	1	1	2
10	30258	100016	12	1	16	1	1	2

- 항목 및 코드집 (출산전후휴가 코드 내용)

인지도	1	잘 알고 있다	b1a11
인지도	2	대충 알고 있다	b1a11
인지도	3	들어본 적은 있다	b1a11
인지도	4	모른다	b1a11
활용 가능 여부	1	필요한 사람은 모두 자유롭게 활용 가능	b1a12
활용 가능 여부	2	활용가능하나, 직장 분위기, 대체인력 확보 어려움 등으로 인해 충분히 사용하지 못함	b1a12
활용 가능 여부	3	전혀 활용할 수 없음	b1a12
2021년 활용실적	1	있다	b1a13
2021년 활용실적	2	없다	b1a13

2. 사용한 데이터

[사용 변수]

- 사업체 유형 : 업종, 규모, 지역, 노조 유무
- 제도 : 출산전후휴가, 배우자출산휴가, 임신기근로시간단축제도, 육아휴직제도

[데이터 처리]

활용 가능 여부	1	필요한 사람은 모두 자유롭게 활용 가능
활용 가능 여부	2	활용가능하나, 직장 분위기, 대체인력 확보 어려움 등으로 인해 충분히 사용하지 못함
활용 가능 여부	3	전혀 활용할 수 없음

1 : 활용 가능
0 : 활용 불가능 (활용가능여부 2,3)

3. 성능평가방법

- **옵티마이저를 변경하여 분류 예측 결과의 정확도 평가**

- SGD (Stochastic Gradient Descent)
- AdaGrad (Adaptive Gradient)
- RMSProp
- Adam

4. 코드 분석

1) 데이터 불러오기

A	B	C	D	E	F	G	H	I	J	K	L	M	N
아이디	리스트아이	info1	info2	info3	a4	b1a12	b4a12	b5a12	b7a12	c1a12	c17a12	c20a12	c22a12
61232	100003	8	1	16	1	3	3	2		3			
62016	100004	15	2	3	1	1							
30281	100005	3	5	10	2	1	1	2	1	1	1	1	
30582	100006	10	1	1	2					1			
61299	100009	2	1	1	1	1	1	1	1	1	1	1	3

기존 CSV 에서 사용 변수만 정리하여 사용

- 사업체 유형 : 업종(info1), 규모(info2), 지역(info3), 노조 유무(a4)
- 제도 : 출산전후휴가(b1a12), 배우자출산휴가(b4a12), 임신기근로시간단축제도(b5a12), 육아휴직제도(c1a12) 활용가능여부 등

4. 코드 분석

1) 데이터 불러오기

```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt

df = pd.read_csv('/content/dataset_2.csv', encoding='cp949', index_col=0)
print(df)
```

4. 코드 분석

2) 데이터 전처리

활용 가능 여부	1	필요한 사람은 모두 자유롭게 활용 가능
활용 가능 여부	2	활용가능하나, 직장 분위기, 대체인력 확보 어려움 등으로 인해 충분히 사용하지 못함
활용 가능 여부	3	전혀 활용할 수 없음

```
# 값이 2 또는 3인 경우를 0으로 바꾸기
columns_to_replace = ['b1a12', 'b4a12', 'b5a12', 'b7a12', 'c1a12', 'c17a12', 'c20a12', 'c22a12']

df[columns_to_replace] = df[columns_to_replace].replace({2: 0, 3: 0})

# NULL 값 확인하고 0으로 바꾸기
df[columns_to_replace] = df[columns_to_replace].fillna(0)

# 결과 확인
print(df.head())
```

4. 코드 분석

3) 모델 학습

```
cols = ['info1', 'info2', 'info3', 'a4', 'c1a12']

import torch
import torch.nn as nn
import torch.nn.functional as F
import torch.optim as optim

data = torch.from_numpy(df[cols].values).float()
data.shape

=> torch.Size([5070, 5])
```

4. 코드 분석

3) 모델 학습

```
# Split x and y.

x = data[:, :-1]
y = data[:, -1:]

print(x.shape, y.shape)
=> torch.Size([5070, 4]) torch.Size([5070, 1])

# Define configurations.
n_epochs = 200000
learning_rate = 1e-2
print_interval = 10000
```

4. 코드 분석

3) 모델 학습

```
class MyModel(nn.Module):
    def __init__(self, input_dim, output_dim):
        self.input_dim = input_dim
        self.output_dim = output_dim

        super().__init__()

        self.linear = nn.Linear(input_dim, output_dim)
        self.act = nn.Sigmoid()

    def forward(self, x):
        # |x| = (batch_size, input_dim)
        y = self.act(self.linear(x))
        # |y| = (batch_size, output_dim)

        return y
```

모델 : Linear Layer + Sigmoid Function

4. 코드 분석

3) 모델 학습

```
model = MyModel(input_dim=x.size(-1), output_dim=y.size(-1))
crit = nn.BCELoss()

# Choose the optimizer
optimizer = optim.SGD(model.parameters(), lr=learning_rate)

#optimizer = Adagrad, RMSprop, Adam
```

손실함수 : BCELoss

옵티마이저 선택

4. 코드 분석

3) 모델 학습

```
for i in range(n_epochs):
    y_hat = model(x)
    loss = crit(y_hat, y)

    optimizer.zero_grad()
    loss.backward()

    optimizer.step()

    if (i + 1) % print_interval == 0:
        print('Epoch %d: loss=%.4e' % (i + 1, loss))
```

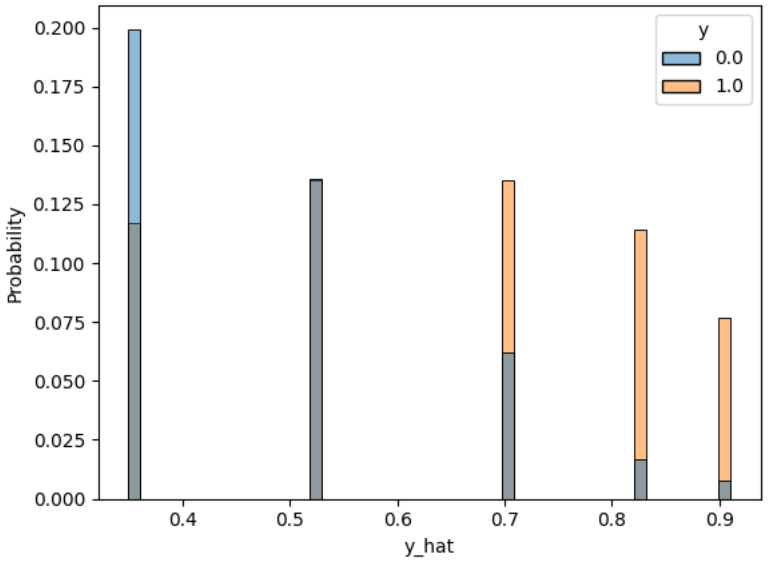
4. 코드 분석

4) 결과 확인 (정확도 확인)

```
correct_cnt = (y == (y_hat > .5)).sum()
total_cnt = float(y.size(0))

print('Accuracy: %.4f' % (correct_cnt / total_cnt))
```

- 규모와 육아휴직 활용가능여부를 변수로 사용
  => Accuracy: 0.6602



5. 성능

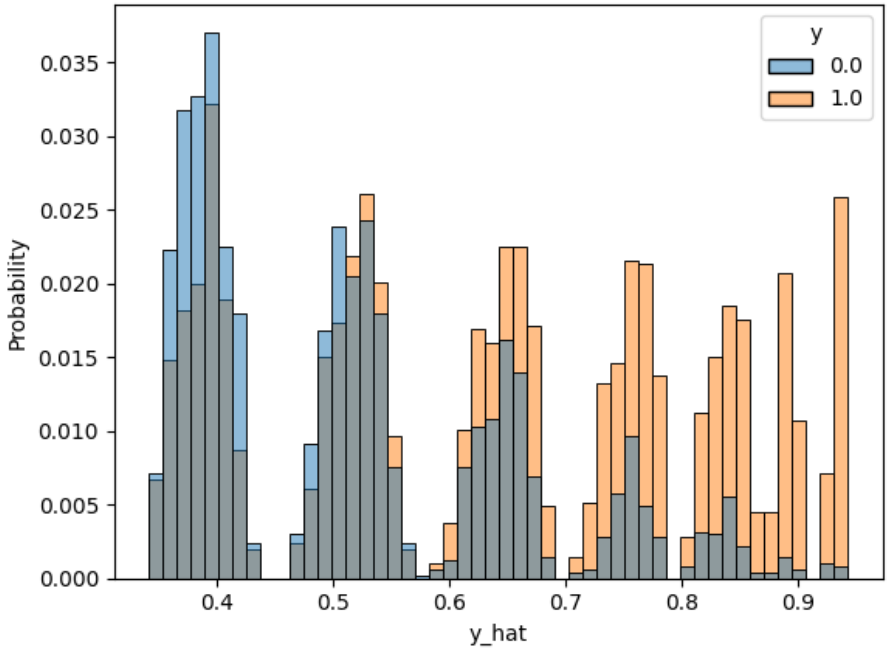
- c1a12(육아휴직제도 사용가능여부)

- SGD : Accuracy: 0.6244

- RMSprop : Accuracy: 0.6398

- Adagrad : Accuracy: 0.6438

- **Adam : Accuracy: 0.6444**



6. 결론

- 성능 평가 결과 -> 적합한 데이터가 아니라 예측 정확도가 높게 나오지 못함.
- 업종, 규모, 지역, 노조 유무

  규모와 노조 유무가 가장 연관이 있는 데이터로 파악됨.
  업종과 지역은 크게 상관이 없었다.
- 옵티마이저에서는 Adam의 성능이 가장 좋았음.