일반화 선형 모델 (Generalized Linear Model, GLM)

- 일반적인 선형 회귀(OLS)의 가정을 완화하여, 정규분포가 아닌 다양한 확률분포를 따르는 종속변수를 모델링할 수 있도록 확장한 프레임워크
- OLS는 종속변수가 정규분포를 따르고, 독립변수와 선형 관계를 가진다고 가정
- but, 현실의 데이터는 이러한 가정을 만족하지 않는 경우가 많음
    - e.g. 개수 데이터, 비율 데이터, 생존 시간 데이터 등

GLM의 세 가지 구성 요소

1. **확률 성분 (Random Component)**: 종속변수 y가 따르는 확률분포를 지정합니다. (e.g., 정규분포, 푸아송분포, 이항분포, 감마분포 등)
2. **체계적 성분 (Systematic Component)**: 독립변수들의 선형 결합($\eta = \beta_0 + \beta_1x_1 + ...$)을 정의합니다. 이는 선형 예측자(Linear Predictor)라고 불립니다.
3. **연결 함수 (Link Function)**: 확률 성분(종속변수의 기댓값 $E(y)=\mu$)과 체계적 성분(선형 예측자 η)을 연결하는 함수 $g(\cdot)$입니다. 즉, $g(\mu) = \eta$ 입니다. 연결 함수를 통해 종속변수의 제약조건(e.g., 개수는 항상 0 이상)을 만족시키면서 선형 모델을 적용할 수 있습니다.

적용 가능한 상황

- **푸아송 회귀 (Poisson Regression)**: 종속변수가 특정 시간이나 공간 내에서 발생하는 사건의 횟수(count)일 때 사용합니다. (e.g., 하루 동안 웹사이트 방문자 수, 한 달간 특정 지역의 교통사고 건수)
- **감마 회귀 (Gamma Regression)**: 종속변수가 양의 값을 가지며 오른쪽으로 꼬리가 긴 분포(right-skewed)를 따를 때 사용합니다. (e.g., 보험 청구액, 병원 입원 기간, 강우량)
- **음이항 회귀 (Negative Binomial Regression)**: 푸아송 회귀와 같이 카운트 데이터를 다루지만, 데이터의 분산이 평균보다 큰 '과산포(Overdispersion)' 현상이 나타날 때 사용합니다. 푸아송 회귀는 평균과 분산이 같다고 가정하지만, 실제 데이터는 이 가정을 위배하는 경우가 많아 음이항 회귀가 더 적합할 수 있습니다.
- 이 외에도 **로지스틱 회귀**는 종속변수가 이항분포를 따르는 GLM의 한 종류입니다.

구현 방법

GLM은 `statsmodels` 라이브러리를 통해 매우 효과적으로 구현할 수 있습니다. `sm.GLM` 클래스를 사용하며, `family` 인자를 통해 확률분포와 연결 함수를 지정합니다.

용도

- 정규분포 가정을 만족하지 않는 다양한 유형의 종속변수(카운트, 비율, 양의 연속형 등)를 모델링합니다.

주의사항

- **적절한 `family` 선택**: 데이터의 특성에 맞는 확률분포(family)를 선택하는 것이 매우 중요합니다. 잘못된 분포를 선택하면 모델의 해석과 예측 성능이 저하됩니다.
- **과산포 (Overdispersion)**: 푸아송 회귀의 경우, 분산이 평균보다 유의미하게 큰 과산포가 있는지 확인해야 합니다. 과산포가 존재하면 표준오차가 과소추정되어 변수의 유의성이 부풀려질 수 있으므로, 음이항 회귀를 대안으로 고려해야 합니다.

- **노출 (Exposure)**: 카운트 데이터 분석 시, 관찰 기간이나 공간의 크기가 다르다면 이를 '노출' 변수로 처리하여 모델에 반영해야 합니다. (e.g., `exposure` 인자 사용)

1. 푸아송 회귀 (Poisson Regression)

- **상황**: 어떤 상점의 시간대별 고객 방문 횟수를 예측.

```python
import statsmodels.api as sm
import pandas as pd
import numpy as np

# 데이터 생성
np.random.seed(42)
X = pd.DataFrame({
    'hour': np.arange(8, 22).repeat(10), # 오전 8시 ~ 오후 9시
    'is_weekend': np.random.randint(0, 2, 140)
})
# 시간에 따라 방문자 수가 변하고, 주말에 더 많다고 가정
lambda_val = np.exp(0.5 + 0.1 * (X['hour'] - 8) + 0.5 * X['is_weekend'])
y = np.random.poisson(lambda_val)

X_const = sm.add_constant(X)

# 모델 학습
# family=sm.families.Poisson(): 푸아송 분포와 로그 연결 함수(기본값)를 사용
poisson_model = sm.GLM(y, X_const, family=sm.families.Poisson())
poisson_results = poisson_model.fit()

print("--- 푸아송 회귀 결과 ---")
print(poisson_results.summary())
'''
                 Generalized Linear Model Regression Results
==============================================================================
Dep. Variable:                      y   No. Observations:                  140
Model:                            GLM   Df Residuals:                      137
Model Family:                 Poisson   Df Model:                            2
Link Function:                    Log   Scale:                          1.0000
Method:                          IRLS   Log-Likelihood:                -306.35
Date:                Fri, 10 Oct 2025   Deviance:                       178.94
Time:                        16:25:12   Pearson chi2:                     166.
No. Iterations:                     5   Pseudo R-squ. (CS):             0.6142
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
const         -0.2723      0.183     -1.487      0.137      -0.631       0.087
hour           0.0972      0.010      9.302      0.000       0.077       0.118
is_weekend     0.5408      0.084      6.474      0.000       0.377       0.705
==============================================================================
'''
```

결과 해석: coef는 로그 스케일에서의 변화량. np.exp(coef)를 통해 해석.

- 절편(const) = -0.2723 → $e^{-0.2723}=0.762$
  - 해석: hour=0, is_weekend=0일 때의 기대값
  - 실제 데이터의 시간 범위와 맞지 않음(데이터는 hour=8~21) → 절편 단독 해석은 주의 필요
- hour 계수 = 0.0972 → $e^{0.0972}=1.102$
  - 한 시간 증가당 기대 방문자 수가 약 1.102배(약 +10.2%) 증가.
  - 이 때 신뢰구간 [0.077, 0.118]은 np.exp() 적용 시 (1.080, 1.125)이 됨
- is_weekend 계수 = 0.5408 → $e^{0.5408}=1.717$
  - **주말이면 기대 방문자 수가 약 1.717배(약 +71.7%)**이다.
  - 이 때 신뢰구간 [0.377, 0.705]은 np.exp() 적용 시 (1.458, 2.024)이 됨
- 푸아송 회귀 결과, 한 시간 증가당 기대 방문자 수는 **약 10.2%** 증가(95% CI 8.0%–12.5%)하며, 주말이면 기대 방문자 수는 **약 71.7%** 더 많다(95% CI 45.8%–102.4%).

2. 음이항 회귀 (Negative Binomial Regression)

- **상황**: 푸아송 회귀 모델의 잔차 분석 결과 과산포가 의심될 때.

```
# 음이항 회귀 모델 학습
# family=sm.families.NegativeBinomial(): 음이항 분포와 로그 연결 함수 사용
# 단, 해당 방식은 alpha 값 출력이 이루어지지 않아, sm.NegativeBinomial 함수로 진행
neg_binom2 = sm.NegativeBinomial(y, X_const)
neg_binom2_results = neg_binom2.fit()

print(neg_binom2_results.summary())
'''
Optimization terminated successfully.
         Current function value: 2.184673
         Iterations: 15
         Function evaluations: 16
         Gradient evaluations: 16
                     NegativeBinomial Regression Results
==============================================================================
Dep. Variable:                      y   No. Observations:                  140
Model:               NegativeBinomial   Df Residuals:                      137
Method:                           MLE   Df Model:                            2
Date:                Fri, 10 Oct 2025   Pseudo R-squ.:                  0.1195
Time:                        16:57:15   Log-Likelihood:                -305.85
converged:                       True   LL-Null:                       -347.36
Covariance Type:            nonrobust   LLR p-value:                 9.461e-19
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
const         -0.2670      0.192     -1.390      0.164      -0.643       0.109
hour           0.0970      0.011      8.765      0.000       0.075       0.119
is_weekend     0.5367      0.089      6.058      0.000       0.363       0.710
```

```
alpha          0.0261      0.029      0.894      0.371      -0.031       0.083
==============================================================================
'''
```

결과 해석: summary의 [alpha] 값이 음이항 분포의 과산포 파라미터.

- 음이항 회귀에서 `alpha` 값에 대한 가설
  - **귀무가설 H_0**: α = 0 (과산포 없음 → 푸아송으로 충분)
  - **대립가설 H_1**: α > 0 (과산포 존재 → 음이항이 더 적합)
- `alpha` 값의 `coef` = 0.0261, `p-value` = 0.371
- 현재 `alpha` 값은 95% 신뢰구간에서 통계적으로 유의하지 않아, 대립가설을 기각하고 귀무가설을 채택할 수 있다.
  - 과산포가 존재하지 않는다고 해석 가능하고, 푸아송 모델이 음이항 모델보다 적합할 수 있음을 시사.
- `hour` 및 `is_weekend`에 대한 해석은 푸아송 모델 해석과 동일하게, `np.exp(coef)` 적용 후 해석 진행하면 된다.

3. 감마 회귀 (Gamma Regression)

- **상황**: 자동차 사고 당 보험 청구액(양수, skewed)을 예측.

```
# 1. 데이터 생성
np.random.seed(123)
X_gamma = pd.DataFrame({
    'driver_age': np.random.randint(18, 70, 100),
    'car_value': np.random.uniform(500, 50000, 100)
})
# 나이가 적고 차 가격이 비쌀수록 청구액이 높다고 가정
mu = np.exp(10 - 0.02 * X_gamma['driver_age'] + 0.00001 * X_gamma['car_value'])
# 감마 분포는 shape(alpha)와 scale(beta) 파라미터를 가짐. 여기서는 shape=2로 고정.
y_gamma = np.random.gamma(shape=2., scale=mu/2.)

X_gamma_const = sm.add_constant(X_gamma)

# 2. 모델 학습
# family=sm.families.Gamma(link=sm.families.links.log()): 감마 분포와 로그 연결 함수 사용
# 감마 회귀는 역수(inverse) 연결 함수가 기본값이지만, 로그 연결이 해석에 용이할 때가 많음.
gamma_model = sm.GLM(y_gamma, X_gamma_const,
family=sm.families.Gamma(link=sm.families.links.log()))
gamma_results = gamma_model.fit()

print("\n--- 감마 회귀 결과 ---")
print(gamma_results.summary())
'''
                 Generalized Linear Model Regression Results
==============================================================================
Dep. Variable:                      y   No. Observations:                  100
Model:                            GLM   Df Residuals:                       97
```

```
Model Family:                  Gamma   Df Model:                            2
Link Function:                   log   Scale:                         0.53893
Method:                         IRLS   Log-Likelihood:                -1021.3
Date:               Fri, 10 Oct 2025   Deviance:                       65.872
Time:                       17:12:26   Pearson chi2:                     52.3
No. Iterations:                   14   Pseudo R-squ. (CS):             0.2188
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
const          9.6546      0.266     36.302      0.000       9.133      10.176
driver_age    -0.0180      0.005     -3.741      0.000      -0.027      -0.009
car_value   1.743e-05    5.3e-06      3.287      0.001    7.04e-06    2.78e-05
==============================================================================
'''
```

결과 해석 방법

- `summary()` **결과**: 각 모델의 `summary()`는 OLS와 유사한 형태의 결과를 제공합니다.
  - `coef`: 각 독립변수의 회귀 계수입니다. 연결 함수를 거친 스케일에서의 값이므로 해석에 주의해야 합니다. (e.g., 로그 연결 함수면 e^{coef}를 취해 배수 효과로 해석)
  - `P>|z|`: 계수의 유의성을 판단하는 p-value입니다.
  - `Deviance`, `Log-Likelihood`: 모델의 적합도를 나타내는 지표로, 모델 간 비교에 사용됩니다. (e.g., AIC, BIC)
- **과산포 확인**: 푸아송 회귀 후, `poisson_results.pearson_chi2 / poisson_results.df_resid` 값을 계산해볼 수 있습니다. 이 값이 1보다 현저히 크면 과산포를 의심하고 음이항 회귀를 고려합니다.

장단점 및 대안

- **장점**:
  - 다양한 분포의 종속변수를 하나의 통일된 프레임워크로 모델링할 수 있어 유연성이 매우 높습니다.
  - `statsmodels`를 통해 풍부한 통계적 추론 결과(계수 유의성, 신뢰구간 등)를 얻을 수 있습니다.
- **단점**:
  - 데이터에 적합한 분포와 연결 함수를 선택해야 하는 어려움이 있습니다.
  - 모델의 가정이 복잡하여 해석이 OLS보다 어려울 수 있습니다.
- **대안**:
  - **변수 변환 후 OLS**: 종속변수에 로그, 제곱근 등 변환을 적용하여 정규성을 만족시킨 후 OLS를 적용하는 간단한 방법을 시도할 수 있습니다.
  - **준최대가능도(Quasi-Likelihood) 모델**: 종속변수의 정확한 분포를 모르더라도, 평균과 분산의 관계만 가정하여 모델을 추정할 수 있습니다. (e.g., `sm.families.QuasiPoisson`)
  - **머신러닝 모델**: 트리 기반 모델(Random Forest, Gradient Boosting) 등은 분포 가정이 필요 없고 비선형 관계도 잘 다루므로 대안이 될 수 있지만, 통계적 추론보다는 예측에 더 초점을 둡니다.