**분석 보고서**

1. **패키지 및 데이터 불러오기**

* Python에서 데이터를 분석하기 위해서는 필요한 패키지를 불러오는 것이 필요함.
* 수치계산을 위한 numpy, 데이터프레임 관리를 위한 pandas, 시각화를 위한 matplotlib, seaborn은 필수적이며, 분석 모델에 따라 다양한 패키지가 존재함
* 이번 분석에서는 로지스틱 회귀분석을 통한 분류 분석을 진행하기 때문에 모델 학습을 위한 statsmodel과 학습된 모델의 평가를 위한 sklearn.metrics 패키지를 활용하였음.

**import** **numpy** **as** **np**

**import** **pandas** **as** **pd**

**from** **matplotlib** **import** pyplot **as** plt

**import** **seaborn** **as** **sns**

**import** **statsmodels.api** **as** **sm**

**from** **sklearn.metrics** **import** confusion\_matrix

**from** **sklearn.metrics** **import** classification\_report

**from** **sklearn.metrics** **import** roc\_curve

**from** **sklearn.metrics** **import** auc

path = "D:**\\**Dropbox**\\**5. Kmong\_Work**\\**Kmong\_2022\_전반기**\\**220116 회귀교육자료**\\**"

data = pd.read\_csv(path + "pima-indians-diabetes.csv")

1. **데이터 변수 설명**

* 분석 대상 데이터는 인디언들의 신체 특성에 따른 당뇨병 여부를 나타내는 데이터로 총 9개의 변수로 구성이 되어 있으며, 그 설명은 다음과 같음.

*# 1. Number of times pregnant*

*# 2. Plasma glucose concentration a 2 hours in an oral glucose tolerance test*

*# 3. Diastolic blood pressure (mm Hg)*

*# 4. Triceps skin fold thickness (mm)*

*# 5. 2-Hour serum insulin (mu U/ml)*

*# 6. Body mass index (weight in kg/(height in m)^2)*

*# 7. Diabetes pedigree function*

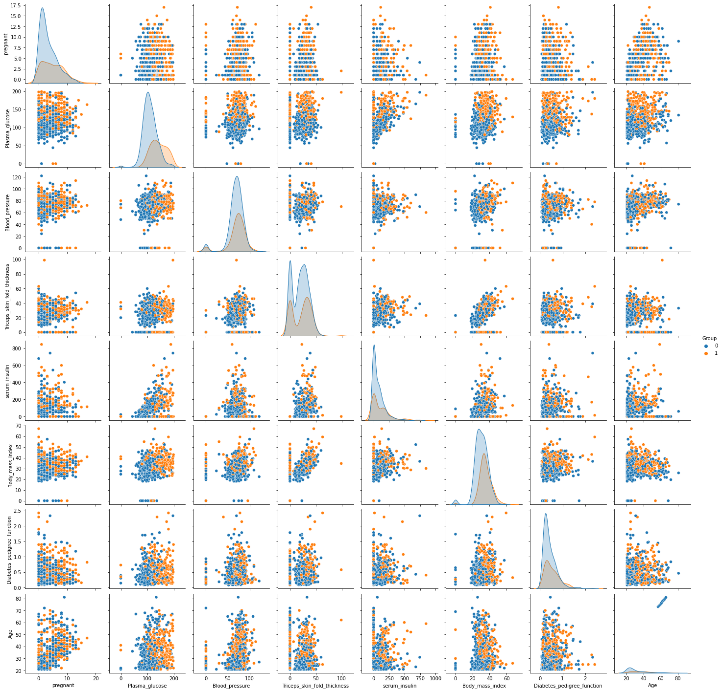
*# 8. Age (years)*

*# 9. Class variable (0 or 1)*

1. **시각화**

* Python에선 seaborn, matplotlib 패키지를 활용하면 변수들의 분포와 그 관계를 시각적으로 확인할 수 있음.
* sns.pairplot()을 활용할 경우, 분석 대상 변수들 간의 산점도 행렬과 분포를 확인할 수 있음.

sns.pairplot(data, hue="Group")



* 위 그림은, 당뇨병 여부에 따른 각 변수들의 분포와 관계를 나타냄. 파랑색이 당뇨병이 없는 그룹이며 주황색이 당뇨병이 있는 그룹임.
* 시각화를 깔끔하게 잘 할 경우, 분석 결과를 많이 설명하지 않아도 한 번에 데이터의 특성을 알아 볼 수 있는 효과가 존재함.

1. **로지스틱 회귀분석 학습**

* 인디언들의 신체 특성이 당뇨병 여부에 어떤 영향을 미치는지 확인하기 위해 분류 모형 중 하나인 로지스틱 회귀분석 학습을 진행
* 학습을 진행하기에 앞서 데이터를 모델 훈련에 활용할 train set와 검증에 활용할 test set으로 분할한 후 학습을 진행하였음

np.random.seed(1234)

sl = np.random.randint(data.shape[0], size=round(data.shape[0] \* 0.8))

train = data.iloc[sl,]

train = train.reset\_index(drop=True)

test = data[~data.index.isin(sl)]

test = test.reset\_index(drop=True)

glm = sm.GLM.from\_formula('Group ~ pregnant + Plasma\_glucose + Blood\_pressure + Triceps\_skin\_fold\_thickness + serum\_insulin + Body\_mass\_index + Diabetes\_pedigree\_function + Age', data=train, family=sm.families.Binomial())

glm\_res = glm.fit()

**print**(glm\_res.summary())

* 로지스틱 회귀분석에 대한 학습이 완료가 되면 결과를 통해 모델의 결과를 확인할 수 있는데, 이는 다음과 같음

===============================================================================================

coef std err z P>|z| [0.025 0.975]

-----------------------------------------------------------------------------------------------

Intercept -8.7824 0.814 -10.788 0.000 -10.378 -7.187

**pregnant 0.0969 0.037 2.627 0.009 0.025 0.169**

**Plasma\_glucose 0.0340 0.004 8.490 0.000 0.026 0.042**

**Blood\_pressure -0.0180 0.006 -3.059 0.002 -0.030 -0.006**

**Triceps\_skin\_fold\_thickness -0.0180 0.007 -2.409 0.016 -0.033 -0.003**

serum\_insulin 0.0008 0.001 0.824 0.410 -0.001 0.003

**Body\_mass\_index 0.1059 0.017 6.168 0.000 0.072 0.140**

**Diabetes\_pedigree\_function 1.0468 0.354 2.957 0.003 0.353 1.741**

**Age 0.0285 0.011 2.591 0.010 0.007 0.050**

===============================================================================================

* 중점적으로 봐야할 값은 coef와 P>|z| 2개의 값으로 coef는 회귀계수(기울기)를 의미하며 P>|z|는 유의확률로 회귀계수의 유의성(당뇨병에 정말 영향을 주는지)을 판단하는 수치임. 유의확률이 0.05보다 작을 경우 회귀계수가 영향력을 가지고 있다고 판단함
* coef가 양수(+)일 경우에는 해당 변수의 값이 높을수록 당뇨병에 걸릴 가능성이 높다는 것을 의미하고 반대로 음수(-)일 경우에는 해당 변수의 값이 높을수록 당뇨병에 걸릴 가능성이 낮다는 것을 의미함.
* serium insulin의 경우 유의확률이 0.41로 0.05보다 높은 것을 확인할 수 있는데, 이는 insulin이 당뇨병에 영향을 주지 않는 것을 의미함. 나머지 모든 변수들은 당뇨병에 유의한 영향을 미치는 것으로 확인이 되었음.

1. **로지스틱 회귀분석을 통한 당뇨병 예측**

* 학습된 모형을 활용할 경우 인디언의 신체 특성에 따른 당뇨병 발병여부를 예측할 수 있음.
* 로지스틱 회귀분석을 통한 예측 결과는 당뇨병에 걸릴 확률(0~1)로 나타남. 만약 발병율이 0.5(50%) 이상인 경우를 당뇨병에 걸렸다고 판단할 경우 그 예측 정확성은 다음과 같음.

predicted = glm\_res.predict(test)

predicted\_c = np.where(predicted > 0.7, 1, 0)

confusion\_matrix(test['Group'], predicted\_c, labels=[1, 0])

array([[ 43, 73],

[ 12, 222]], dtype=int64)

* 분류모델의 경우 실제 Positive, Negative에 대해 모형이 제대로 예측을 했는지 분석을 하는데, 예측값을 기준으로 맞췄을 경우 True, 틀렸을 경우 False로 표현한다. 이를 표로 나타내면 다음과 같음.

|  |  |  |
| --- | --- | --- |
|  | **True Positive** | **True Negative** |
| **Predicted Positive** | True Positive (TP), 43 | False Positive (FP), 73 |
| **Predicted Negative** | False Negative (FN), 12 | True Negative (TN), 222 |

* 모형의 분류 정확도를 계산하면 (43 +222) / (43 + 73 + 12 + 222) = 0.75로 모형의 예측 정확도는 대체로 75%정도 되는 것을 확인할 수 있음.

1. **로지스틱 회귀분석을 통한 예측 시뮬레이션**

* 분류모델을 통해 인디언의 나이 증가에 따른 당뇨병 발병확률을 예측할 수 있음. 예를 들어, 다음의 수치를 가진 인디언이 1년씩 나이를 먹을 경우에 대해 당뇨병 발병확률을 예측하면 다음과 같음.
* 인디언의 특성은 다음과 같음.

|  |  |
| --- | --- |
| **Columns** | **Value** |
| pregnant | 13 |
| Plasma\_glucose | 145 |
| Blood\_pressure | 85 |
| Triceps\_skin\_fold\_thickness | 19 |
| serum\_insulin | 110 |
| Body\_mass\_index | 22.2 |
| Diabetes\_pedigree\_function | 0.245 |

* 이러한 특성을 가진 인디언의 57세부터 66세까지 총 10년간의 당뇨병 발병확률을 예측하면 다음과 같음.

sample = test.loc[10, ['pregnant', 'Plasma\_glucose', 'Blood\_pressure',

'Triceps\_skin\_fold\_thickness', 'serum\_insulin', 'Body\_mass\_index',

'Diabetes\_pedigree\_function']]

simulation = pd.DataFrame()

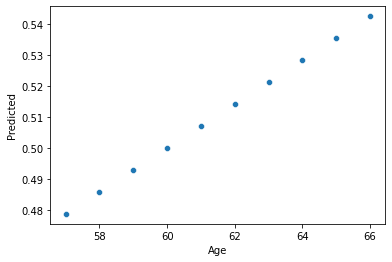
simulation = simulation.append([sample]\*10, ignore\_index=True)

simulation['Age'] = list(range(57, 67))

predicted2 = glm\_res.predict(simulation)

simulation["Predicted"] = predicted2

sns.scatterplot(data=simulation, x="Age", y="Predicted")



* 시간의 흐름에 따라 당뇨병 발병확률이 증가하는 것을 확인할 수 있음
* 예측 모형을 잘 활용할 경우에는 미래 예측 및 시뮬레이션 등의 작업을 보다 더 정교하게 할 수 있는 장점이 존재함