데이터마이닝 이론 및 응용 7주차 과제

Logistic Regression & Model Assessment

요조[곽지운, 박현준, 이정현, 최동훈]

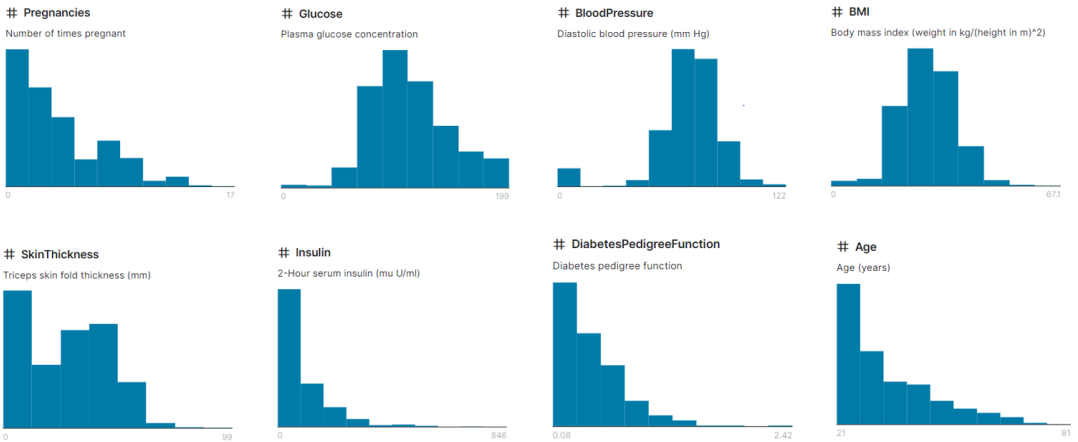
1. 서론

1.1 데이터 설명

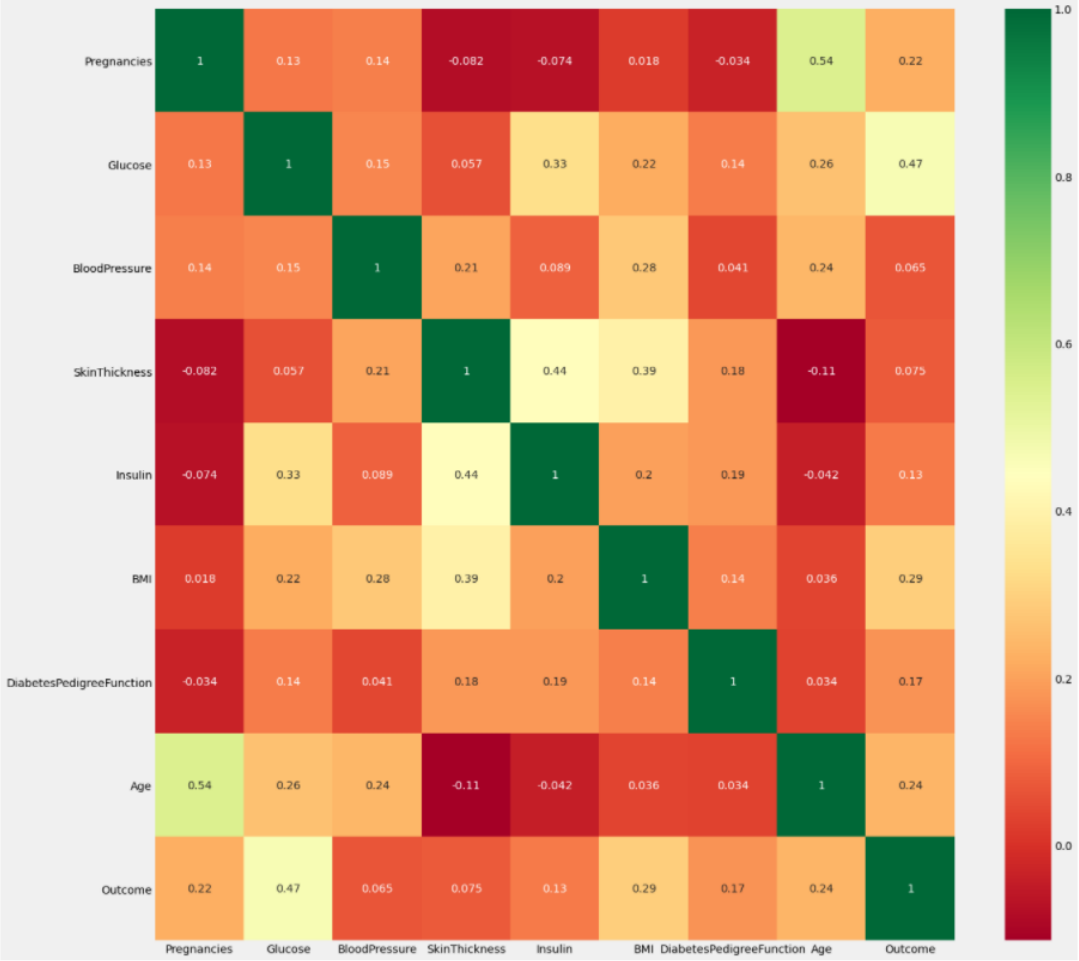
이번 실습에 활용할 데이터로 캐글의 당뇨병 진단 결과 데이터를 선택하였다. 해당 데이터는 미 국립 당뇨, 소화기병 및 신장병 연구소가 발표한 데이터로 총 768명의 21살 이상 여성 북미 원주민의 후손들의 나이, 인슐린 레벨, BMI, 등의 독립변수와 그들의 당뇨 진단 결과를 포함하고 있다.

총 8개의 독립변수와 1개의 종속변수로 구성되어 있으며 독립변수와 각 변수의 데이터가 가진 특징, 데이터 분포는 다음과 같다.

독립변수	평균	표준편차	최소값	최댓값
임신 횟수	3.85회	3.37	0	17
혈중 포도당 농도	12mg/dl	32	0	199
이완기 혈압	69.1mmHg	19.3	0	122
삼두근 피부 주름 두께	20.5mm	15.9	0	99
혈중 인슐린 농도	78.9uU/ml	115	0	128
BMI 지수	32	7.88	0	67.1
가족력에 의한 당뇨 가능성 지표	0.47	0.33	0.08	2.42
나이	33.2	11.8	21	81



독립변수들은 모두 연속형 변수였으며 임신 횟수, 혈중 인슐린 농도, 가족력에 의한 당뇨 가능성 지표, 나이는 좌측으로 치우친 분포를 띠는 것을 그래프를 통해 확인할 수 있었다. 그에 반해 혈중 포도당 농도, 이완기 혈압, BMI 지수는 비교적 정규분포에 가까운 분포를 띠는 것을 확인할 수 있었다.



독립변수들 간의 상관관계가 존재하는지 파악하기 위해 차트를 그려보았고, 차트는 위와 같이 나타났다. 매우 높은 상관관계를 가지는 독립변수는 없는 것으로 파악되었고, 나이와 임신 횟수, 혈중 인슐린 농도와 삼두근 피부 주름 두께정도만이 0.4정도의 양의 상관관계를 갖는다는 것을 확인할 수 있었다.

종속변수는 당뇨병을 진단받았는지를 나타내는 이항변수로, 당뇨병을 진단받은 경우 1, 그렇지 않은 경우 0으로 표시했다. 768명 중 268명이 당뇨병 진단을 받았고, 500명은 당뇨병이 없다고 진단받았다.

1.2 실습 목표

Logistic 회귀는 독립변수와 종속변수의 관계를 fitting하여 새로운 데이터가 들어올 경우, 그에 따른 종속변수를 예측하는 모델이 아닌 분류하는 모델로서, 데이터의 독립변수를 고려해 종속변수가 어떤 클래스에 속하는지에 대한 확률을 제시하는 모델이다.

따라서 위의 데이터를 통해 환자들의 당뇨병 여부에 영향을 주는 요인들을 파악하고 로지스틱 회귀분석을 통해 새로운 환자의 당뇨병 여부를 분류하는 회귀식을 만들고자 한다.

로지스틱 회귀에서 승산 비율에 로짓 변환을 한 Logit(p)함수는 다음과 같이 나타나며, 당뇨병 유무에 대해 어떤 class에 속할지에 대한 확률은 아래의 식과 같이 나타난다. 이 식에 새로운 데이터를 입력하고 결과값을 threshold value와 비교하여 당뇨병 유무를 분류하게 된다.

$$Logit(p) = ln(Odds\ ratio) = ln(p/(1-p)) = \beta_0 + \beta_1 Pregnancies + \cdots + \beta_8 \text{age}$$

$$p = \frac{1}{1 + e^{-(\beta_0 + \beta_1 \text{pregnancies} + \cdots + \beta_8 age)}}$$

2. Logistic Regression

2.1. 데이터 전처리

데이터 값들의 분포와 이상값, 그리고 Null 값을 확인하는 과정을 거쳤다.

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
count	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000
mean	3.845052	120.894531	69.105469	20.536458	79.799479	31.992578	0.471876	33.240885	0.348958
std	3.369578	31.972618	19.355807	15.952218	115.244002	7.884160	0.331329	11.760232	0.476951
min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.078000	21.000000	0.000000
25%	1.000000	99.000000	62.000000	0.000000	0.000000	27.300000	0.243750	24.000000	0.000000
50%	3.000000	117.000000	72.000000	23.000000	30.500000	32.000000	0.372500	29.000000	0.000000
75%	6.000000	140.250000	80.000000	32.000000	127.250000	36.600000	0.626250	41.000000	1.000000
max	17.000000	199.000000	122.000000	99.000000	846.000000	67.100000	2.420000	81.000000	1.000000

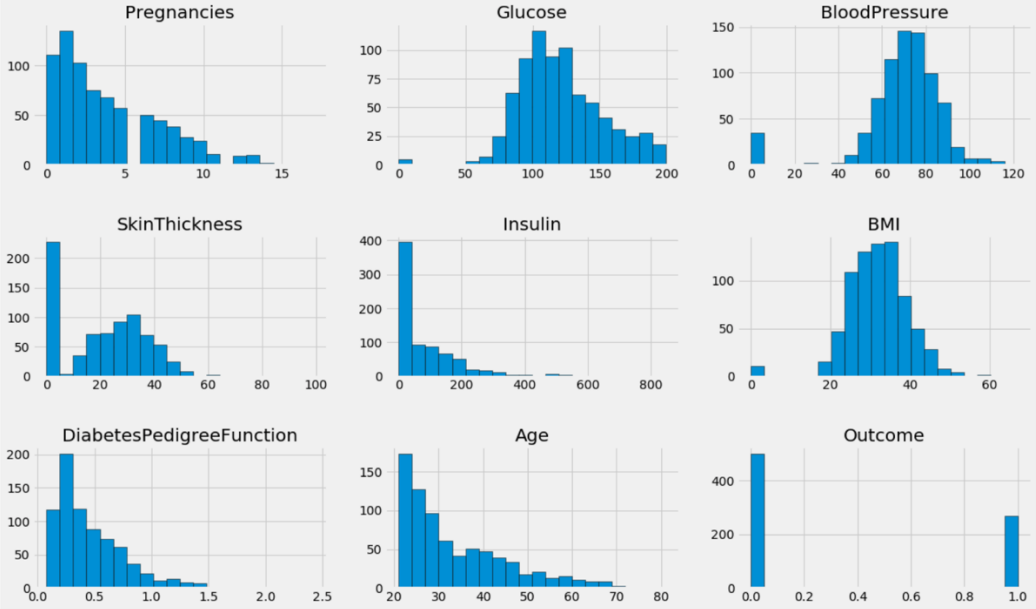
일반적으로 'Glucose', 'BloodPressure', 'SkinThickness', 'Insulin', 'BMI' 데이터에 대해서 Min 값이 0이 나오는 것은 이상하다고 볼 수 있다. 그 이유는 사람이 위 항목에 대해 0인 수치를 기록하면, 그것 자체가 오류이거나 missing vlaue라고 볼 수 있다. 따라서 위 5가지 항목에 대해 적당한 값으로 바꿔주는 과정이 필요하다.

```
## showing the count of Nans
print(diabetes_data_copy.isnull().sum())
```

```
Pregnancies                 0
Glucose                     5
BloodPressure              35
SkinThickness             227
Insulin                   374
BMI                        11
DiabetesPedigreeFunction    0
Age                         0
Outcome                     0
dtype: int64
```

<0 값을 Null 값으로 대체해준 모습>

아래 Histogram은 데이터의 전반적인 형태를 보기 위해 나타났다. 5개의 항목 중 'BMI', 'BloodPressure' 항목은 평균에 가깝게 분포하고 있어 평균으로 Null(0) 값을 대체하고, 'SkinThickness', 'Insulin', 'BMI' 세 항목은 치우쳐져 있는 것을 아래 히스토그램 에서 확인할 수 있다. 그렇기 때문에 평균보다는 중앙값으로 Null(0) 값을 대체하여 진행을 하였다.



위 코드를 통해 Null(0) 값을 대체 해준 뒤 진행을 하였다.

```
diabetes_data_copy['Glucose'].fillna(diabetes_data_copy['Glucose'].mean(), inplace = True)
diabetes_data_copy['BloodPressure'].fillna(diabetes_data_copy['BloodPressure'].mean(), inplace = True)
diabetes_data_copy['SkinThickness'].fillna(diabetes_data_copy['SkinThickness'].median(), inplace = True)
diabetes_data_copy['Insulin'].fillna(diabetes_data_copy['Insulin'].median(), inplace = True)
diabetes_data_copy['BMI'].fillna(diabetes_data_copy['BMI'].median(), inplace = True)
```

```
y.value_counts()
```

```
0    500
1    268
Name: Outcome, dtype: int64
```

y값을 확인한 결과, 다음과 같이 0에 편향되어 있었기 때문에 Scaling후 Shuffle 한 후, imbalanced 한 데이터를 맞춰주기 위한 다음과 같은 과정을 거쳤다.

2.2. Maximum Likelihood

우도는 일어날 가능성으로 나타난 결과에 따라 여러 개의 가능한 가설들을 평가할 수 있는 측도이다. 이때 최대 우도는 결과에 해당하는 각 가설마다 계산된 우도의 값 중 가장 큰 값이다. 즉 관측된 랜덤 표본에 해당되는 여러 가설 중 우도 함수의 값이 최대인 것을 의미한다. 일반적으로 우도함수 L(θ)를 최대화하면서 모수를 추정하는 방법이 최대 우도추정법이다. 위의 과정을 식을 통해 확인해보면,

모든 데이터 들이 독립적이라고 가정하면, 다음과 같은 likelihood 식을 얻을 수 있다.

$$L(\theta) = p(X|\theta) = \prod_{n=1}^{N} p(x_n|\theta)$$

우리는 이 식에 log와 -를 취해서 그 값이 최소가 되는 값을 구함으로써 maximum likelihood를 만들어주는 값을 구한다. 이 식을 log likelihood 라고 한다.

$$E(\theta) = -\ln L(\theta) = -\sum_{n=1}^{N} \ln p(x_n|\theta)$$

그러면 이제 likelihood를 최대화하는 θ 값을 찾을 차례이다. 이를 위해 우리는 log likelihood 식을 미분하고, 이 식이 0이 되는 값을 찾는다. 즉 다음 식을 만족하는 θ값을 찾는 것이다.

$$\frac{\partial}{\partial\theta}E(\theta) = -\frac{\partial}{\partial\theta}\sum_{n=1}^{N}\ln p(x_n|\theta) = -\sum_{n=1}^{N}\frac{\frac{\partial}{\partial\theta}p(x_n|\theta)}{p(x_n|\theta)} \stackrel{!}{=} 0$$

우리는 위 식을 0을호 만드는 parameter $\theta = (\mu, \sigma)$ 값을 찾으면 된다. 그러면 우리는

$$\hat{\mu} = \frac{1}{N}\sum_{n=1}^{N} x_n \quad \hat{\sigma}^2 = \frac{1}{N}\sum_{n=1}^{N}(x_n - \hat{\mu})^2$$

다음과 같은 값을 구할 수 있다. 이처럼 likelihood를 최대화하는 parameter 값을 maximum likelihood estimate 라고 한다. maximum likelihood 기법은 머신러닝에서 모수를 추정할 때, 가장 자주 쓰이는 개념이다. 하지만, maximumlikelihood는 분산을 실제보

다 작게 추정하여 표본에 대해 overfitting 될 수 있다는 한계점을 지닌다.

2.3. 변수 해석 및 Odds Ratio (승산비)

Logit Regression Results

Dep. Variable:	Outcome	**No. Observations:**	375
Model:	Logit	**Df Residuals:**	367
Method:	MLE	**Df Model:**	7
Date:	Fri, 23 Apr 2021	**Pseudo R-squ.:**	0.2807
Time:	01:05:38	**Log-Likelihood:**	-186.96
converged:	True	**LL-Null:**	-259.93
Covariance Type:	nonrobust	**LLR p-value:**	2.889e-28

	coef	std err	z	P>|z|	[0.025	0.975]
Pregnancies	0.4317	0.154	2.809	0.005	0.131	0.733
Glucose	0.9767	0.163	5.998	0.000	0.658	1.296
BloodPressure	-0.0311	0.147	-0.212	0.832	-0.319	0.257
SkinThickness	0.2333	0.168	1.386	0.166	-0.097	0.563
Insulin	0.1647	0.184	0.897	0.370	-0.195	0.525
BMI	0.5515	0.173	3.180	0.001	0.212	0.891
DiabetesPedigreeFunction	0.4465	0.146	3.050	0.002	0.160	0.733
Age	0.1523	0.161	0.948	0.343	-0.163	0.467

위는 로지스틱 회귀분석의 결과 테이블이다. 해당 변수의 회귀계수가 0이다라는 귀무가설 H0를 테이블을 통해 검정을 진행하였다.

$$\log_e(odds_2) - \log_e(odds_1) = \log_e\left(\frac{odds_2}{odds_1}\right) = b_1$$

또한 검정과 함께 해당 식을 이용하여 Odds ratio를 구하여 함께 분석을 진행하였다. 이때, Odds는 p가 성공할 확률을 의미할 때, 성공할 확률을 실패할 확률로 나눈 값이다. 즉 성공할 확률이 높다면 1보다 큰 값을, 실패할 확률이 높다면 1보다 작은 값을 가지게 된다. 본 회귀 모델의 p는 당뇨병에 걸렸을 확률을 의미한다. Odds Ratio는 두 집단에서 한 집단이 다른 집단과 비교하여 성공할 승산의 비에 대한 값을 나타내며, 두 odds를 나눈 값을 의미한다. 다음은 odds와 odds ratio를 계산하는 식이다.

$$odds = \frac{p}{1-p}$$

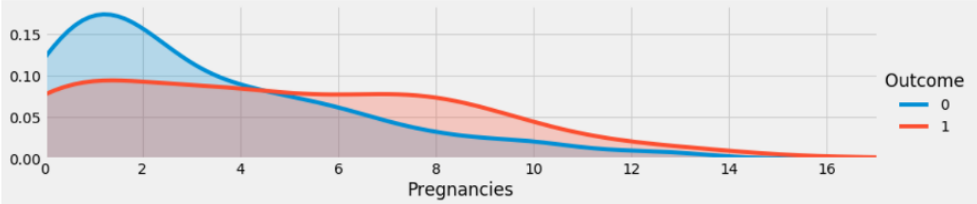
$$\text{odds ratio} = \frac{p1(1-p2)}{p2(1-p1)}$$

```
# Odds ratio
import numpy as np
np.exp(fit_result_sm.params)
```

```
Pregnancies                 1.539928
Glucose                     2.655746
BloodPressure               0.969358
SkinThickness               1.262769
Insulin                     1.179061
BMI                         1.735893
DiabetesPedigreeFunction    1.562836
Age                         1.164517
dtype: float64
```

- Pregnancies

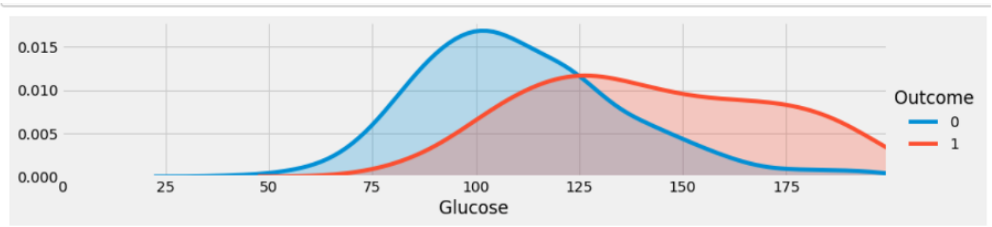
  : 임신 횟수를 나타내는 변수의 경우에는, 회귀계수의 로그 값이 0.4317로 양의 값을 가진 것으로 확인되었다. 또한 z-test의 p-value가 유의수준(0.05)보다 작은 값으로 나왔기 때문에 귀무가설을 기각하여, 종속변수인 당뇨병 발병(Outcome)에 유의미한 영향을 줄 것으로 확인할 수 있었다. Odds Ratio의 경우 1.54가 나온 것으로 보아, Pregnancies 변수의 한 단위가 증가할 때마다 발병이 될(Outcome=1)확률은 1.54배 증가할 것으로 확인할 수 있었다. 또한 임신 횟수에 따른 발병 유무에 영향을 주는 것은 그래프를 통해서도 확인할 수 있었는데, 발병이 안된 경우가 적은 임신 횟수에 몰려있는 것을 볼 수 있었다.



- Glucose

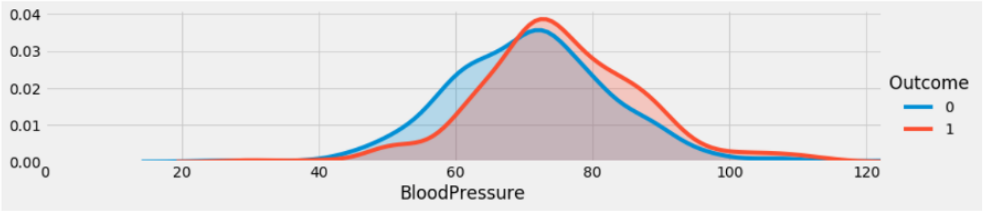
  : 혈중 포도당 농도를 나타내는 변수 Glucose의 경우에는, 회귀계수의 로그 값이 0.9767로 양의 값을 가진 것으로 확인되었다. 또한 z검정의 P-value가 유의수준보다 작은 값으로 나왔기 때문에 귀무가설을 기각하여 종속변수에 유의미한 영향을 준다고 결론을 내었다. Odds ratio의 경우 2.66이라는 변수 중에는 가장 높은 수치가 나왔다. Glucose 변수의 한 단위가 증가할 때마다 발병이 될 확률이 약 2.66배 증가한다는 것이다. 또한 그래프를 통해서도 영역의 확실한 구분을 확인할 수 있

었다.



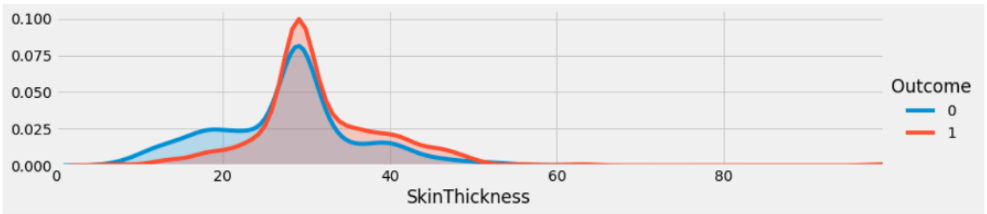
- BloodPressure

  : 이완기 혈압을 나타내는 BloodPressure변수의 경우, 회귀계수의 로그 값이 -0.03111로 음의 값을 가진 것으로 확인되었다. 하지만, z검정의 p-value가 유의수준보다 높은 값으로 나와 귀무가설을 기각하지 않아 종속변수에 유의미한 영향을 주지 않을 것이라고 결론을 내었다. 이는 Odds Ratio를 통해서도 확인이 가능했는데, 0.97이라는 승산비가 도출되었고, 계수 로그값이 음의 값이 나온만큼, 단위가 증가할 때 발병이 될 확률이 0.97배 된다는 의미를 갖는다. 하지만 1에 매우 가까운 숫자이기에 그렇게 큰 영향을 준다고 볼 수는 없었다. 그래프에서도 확인할 수 있듯이 영역 구분이 위 두 변수 대비 크게 되지 않음을 확인할 수 있었다.



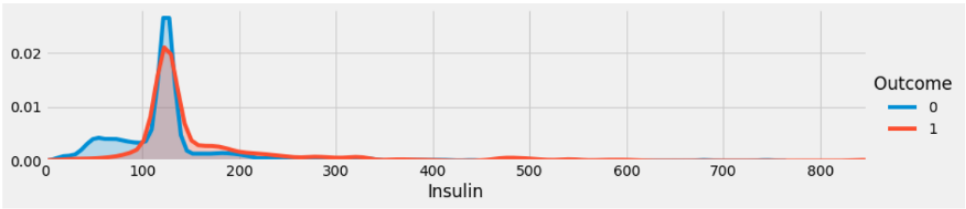
- SkinThickness

  : 삼두근 피부 지름 두께를 나타내는 변수의 경우, 회귀계수의 로그 값이 0.233으로 양의 값을 가졌다. 이 변수도 z검정의 p-value가 유의수준보다 높은 0.166으로 나와 귀무가설을 기각하지 않았다. 따라서 종속변수에 유의미한 영향을 주지 않을 것이라고 결론을 내었다. Odds Ratio의 경우 1.26의 값이 나왔는데 이는 지름 두께의 한 단위가 증가할 때마다 확률이 1.26배 높아진다는 것을 의미하지만, 위 임신 횟수, 포도당 농도와 비교했을 때 낮은 수치임을 확인할 수 있었다. 그래프에서도 영역이 크게 구분되지 않았다.



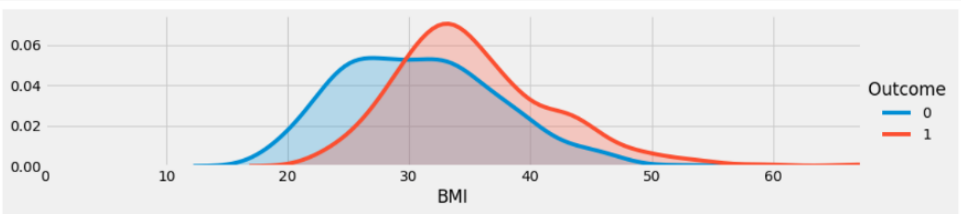
- Insulin

  : 혈중 인슐린 농도를 의미하는 Insulin의 경우 회귀계수의 로그값이 0.1647, z검정의 p-value가 0.37로 도출되었다. 이 변수도 유의수준보다 높아 귀무가설을 기각하지 않았다. 따라서 종속변수에 유의미한 영향을 주지 않을 것이라고 결론을 내었다. Odds Ratio의 경우 1.18의 값이 나왔는데 이는 농도의 한 단위가 증가할 때마다 발병이 될 확률이 1.18배 증가한다는 것을 의미하지만, 이 역시도 낮은 수치임을 확인할 수 있었다. 그래프에서도 인슐린 농도가 낮을 때 Outcome=0인 영역이 구분 되어있지만 전반적으로 영역 구분이 크지 않는 것으로 확인할 수 있었다.



- BMI

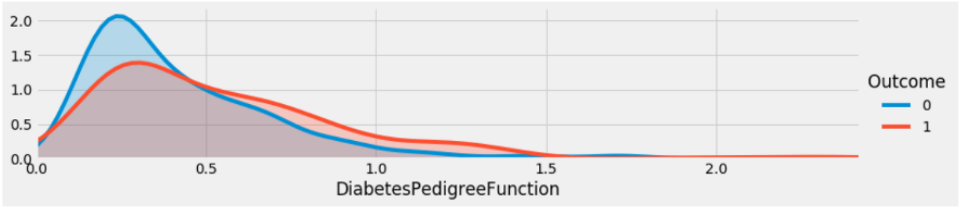
  : BMI지수를 나타내는 변수 BMI의 경우에는, 회귀계수의 로그 값이 0.5515로 양의 값을 가진 것으로 확인되었다. 또한 z검정의 P-value가 유의수준보다 작은 값(0.001)으로 나왔기 때문에 귀무가설을 기각하여 종속변수에 유의미한 영향을 준다고 결론을 내었다. Odds ratio의 경우 1.74이라는 변수 중에는 가장 높은 수치가 나왔다. BMI 변수의 한 단위가 증가할 때마다 발병이 될 확률이 약 1.74배 증가한다는 것이다. 이는 귀무가설을 기각하지 않은 위의 세 가지 변수보다 높은 비율였다. 또한 그래프를 통해서도 영역의 확실한 구분을 확인할 수 있었다.



- DiabetesPedigreeFunction

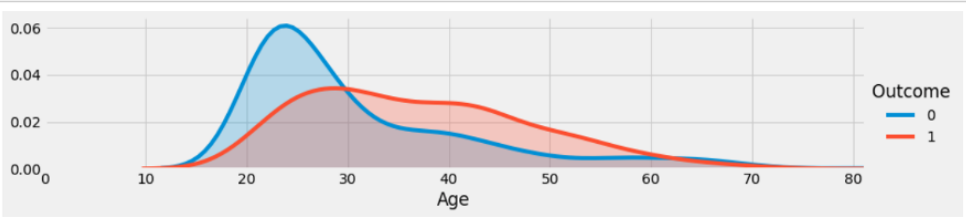
  : 가족력에 의한 당뇨 가능성의 지표를 나타내는 변수의 경우에는, 회귀계수의 로그 값이 0.4465로 양의 값을 가진 것으로 확인되었다. 또한 z검정의 P-value가 유의수준보다 작은 값(0.002)으로 나왔기 때문에 귀무가설을 기각하여 종속변수에 유의미한 영향을 준다고 결론을 내었다. Odds ratio의 경우 1.56이라는 변수 중에는 가장 높은 수치가 나왔다. 이는 해당 변수의 한 단위가 증가할 때마다 발병이 될 확률이 약 1.56배 증가한다는 것이다. 이는 귀무가설을 기각하지 않은 위의 세

가지 변수보다 높은 비율이었다. 또한 그래프를 통해서도 지표가 낮을 수록 발병이 안되는 영역의 확실한 구분을 확인할 수 있었다.



- Age

: Age 변수의 경우 회귀계수의 로그 값이 0.1523으로 양의 값이 나왔다. 또한 z검정의 p-value가 0.343이 나와 귀무가설을 기각하지 않았다. 따라서 종속변수에 유의미한 영향을 주지 않는다고 판단할 수 있었다. Odds Ratio의 경우에도 1.16이 나와 상대적으로 1에 가까운 수치가 나온 것으로 확인할 수 있었다. 하지만 그래프에서는 연령대가 낮을수록 발병이 되지 않는 영역이 뚜렷하게 구분되는 것으로 볼 수 있었다.



2.4. Logistic Regression 결과

위는 로지스틱 회귀분석의 결과 테이블이다. 해당 변수의 회귀계수가 0이다라는 귀무가설 H0를 테이블을 통해 검정을 진행하였다. logistic regression 결과, 데이터들을 다음과 같은 확률로 분류할 수 있다는 것을 확인했다. 그리고 이것을 cut-off 값이 0.5와 0.7인 경우 나눠서 다음 항목에 비교를 하였다.

```
# Probability 값
fit_result_sm.predict(X)
```

```
0      0.854101
1      0.097864
2      0.881776
3      0.071191
4      0.973174
       ...
763    0.669948
764    0.446175
765    0.280920
766    0.400462
767    0.145708
Length: 768, dtype: float64
```

3. Assessment

일련의 과정을 통해 만든 모델을 여러가지 지표를 통해 그 성능을 확인하고자 한다. posterior probability의 threshold를 0.5, 0.7로 나누어 설정한 후 Classification table을 그렸고, 이를 바탕으로 모델의 성능을 평가했다. Classification table의 각 성분과 그 의미는 다음과 같다.

		Predicted value	
		Non-event	Event
Real value	Non-event	(correct-nonevent) (True Negative(TN))	(incorrect-nonevent) (False Positive(FP))
	Event	(incorrect-event) (False Negative(FN))	(correct-event) (True Positive(TP))

성분	의미
True Negative (TN)	0을 예측했고, 실제로 0인 경우
False Negative (FN)	0을 예측했지만, 실제론 1인 경우
False Positive (FP)	1을 예측했지만, 실제론 0인 경우
True Positive (TP)	1을 예측했고, 실제로 1인 경우

이를 바탕으로 여러가지 평가 지표를 만들 수 있으며, 각 지표가 의미하는 바는 다음과 같다.

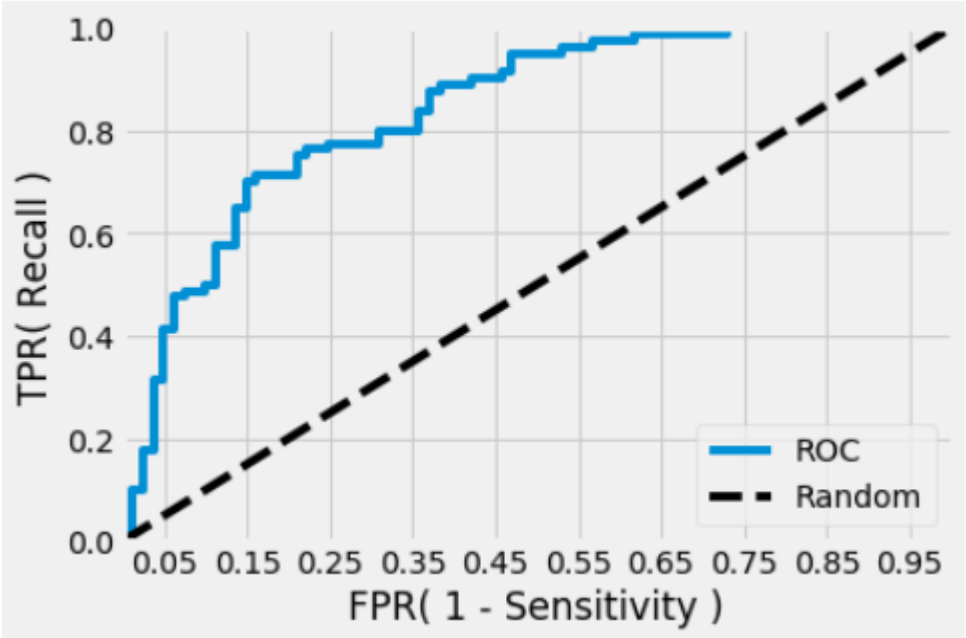
지표	산출 방법	의미
Accuracy	$\frac{TP+TN}{TP+TN+FP+FN}$	전체 사건에서 모델이 분류에 성공한 정도로, 1에 가까울수록 좋다.
Misclassification	$\frac{FP+FN}{TP+TN+FP+FN}$	전체 사건에서 모델이 분류에 실패한 정도로, 0에 가까울수록 좋다.
Sensitivity	$\frac{TP}{TP+FN}$	실제 1인 값들 중 모델이 분류에 성공한 정도로, 1에 가까울수록 좋다.
Specificity	$\frac{TN}{TN+FP}$	실제 0인 값들 중 모델이 분류에 성공한 정도로, 1에 가까울수록 좋다.

Precision	$\frac{TP}{TP+FP}$	모델이 1로 분류했을 때, 실제로 1이 나온 정도로 1에 가까울수록 좋다.
F – measure	$\frac{2TP}{2TP+FP+FN}$	Precision과 Sensitivity의 조화평균으로, 1에 가까울수록 좋다.

posterior probability의 threshold를 나누어 평가하기 전에, ROC curve를 통해 모델의 전체적인 성능을 파악하고자 하였다.

3.1 ROC curve

ROC curve란 Receiver Operating Characteristic Curve의 약자로, y축은 Sensitivity, x축은 1-specify로 그려지는 곡선을 의미한다. 주로 검사도구의 유용성을 판단하고 검사의 정확도를 평가하는데 사용되며, 그래프가 왼쪽 상단에 치우칠수록 더 좋은 분류 모델로 평가한다. 로지스틱 회귀를 통해 만든 모델의 ROC 커브는 다음과 같이 나타났다.



육안으로 그래프를 봤을 때, 어느정도 왼쪽 상단에 치우쳐져 있음을 확인할 수 있지만 보다 정확한 평가를 위해서는 ROC curve의 아래쪽 면적(Area Under Curve, AUC)를 살펴봐야 한다. AUC 면적이 클수록 Sensitivity와 Specificity가 1에 가깝다는 의미이다. 따라서 AUC의 면적이 1에 가까울수록 해당 모델의 정확도가 높으며, 낮을수록 해당 모델의 정확도가 낮다고 할 수 있다. AUC 면적에 대해 Muller(2005)이 언급한 평가 기준은 다음과 같다.

Area Under Curve (AUC)	Evaluation
AUC≥0.9	Excellent
0.8≤AUC<0.9	Good
0.7≤AUC<0.8	Fair
AUC<0.7	Poor

로지스틱 회귀를 통해 구한 모델의 AUC를 구한 결과는 다음과 같다.

```
In [68]: from sklearn.metrics import auc
         fprs , tprs , thresholds = roc_curve(validation_labels, predict_probability_valid_y)
         auc(fprs,tprs)

Out[68]: 0.8402777777777778
```

해당 모델의 AUC는 대략 0.84로 Muller의 평가 기준 상으로 우수한 모델에 속한다고 판단할 수 있다. 이후로는 threshold 값을 0.5일 때, 0.7일 때로 나누어 Classification table을 그리고 각 지표의 수치를 구해 모델의 성능을 평가하고자 하였다.

3.2 Threshold=0.5

1) Classification table

Threshold=0.5		예측값		
		Non-diabetes	Diabetes	
실제값	Non-diabetes	56	25	81
	diabetes	17	63	80
		73	88	161

2) $Accuracy = \frac{56+63}{56+25+17+63} \approx 0.7391$

: 0.5를 threshold값으로 설정한 결과, 해당 모델은 전체 161명의 validation data중 약 74%에 해당하는 119명을 올바르게 분류했다.

3) $Misclassification\ rate = \frac{25+17}{70+11+30+50} \approx 0.2609$

: 전체 161명 중 약 26%에 해당하는 42명은 잘못 분류했다.

4) $Sensitivity\ (Recall) = \frac{63}{17+63} \approx 0.7875$

: 실제로 당뇨병을 진단받은 80명의 사람들 중 약 79%에 해당하는 63명을 당뇨병을 가지고 있을 것이라 분류했다.

5) $Precision = \frac{56}{56+25} \approx 0.6914$

: 정상 진단을 받은 81명의 사람들 중 약 69%에 해당하는 56명을 정상으로 분류했다.

6) $F-measure\ (F1\ score) = \frac{2*recall*precision}{recall+precision} \approx 0.7500$

: F-measure는 recall과 precision의 산술평균이 아닌 조화평균으로, 작은 값을 지닌 recall 쪽으로 치우친 평균이 도출되어 큰 비중을 지닌 precision쪽의 bias가 줄어들게 된다.

3.3 Threshold=0.7

1) Classification table

Threshold=0.7		예측값		
		Non-diabetes	Diabetes	
실제값	Non-diabetes	70	11	81
	diabetes	30	50	80
		100	61	161

2) $Accuracy = \frac{70+50}{70+11+30+50} \approx 0.7453$

: 0.7을 threshold값으로 설정한 결과, 해당 모델은 전체 161명의 validation data중 약 75%에 해당하는 120명을 올바르게 분류했다.

3) $Misclassification\ rate = \frac{11+30}{70+11+30+50} \approx 0.2547$

: 전체 161명 중 약 25%에 해당하는 41명은 잘못 분류했다.

4) $Sensitivity\ (Recall) = \frac{50}{30+50} \approx 0.6250$

: 실제로 당뇨병을 진단받은 80명의 사람들 중 약 62%에 해당하는 50명을 당뇨병을 가지고 있을 것이라 분류했다.

5) $Specificity = \frac{70}{70+11} \approx 0.8642$

: 정상 진단을 받은 81명의 사람들 중 약 86%에 해당하는 70명을 정상으로 분류했다.

6) $Precision = 50 / (11 + 50) \approx 0.8197$

: 당뇨병을 가지고 있을 것이라 분류한 61명 중 약 82%에 해당하는 50명이 실제로 당뇨병을 진단받았다.

7) $F - measure\ (F1\ score) = \frac{2*recall*precision}{recall+precision} \approx 0.7092$

: F-measure는 recall과 precision의 산술평균이 아닌 조화평균으로, 작은 값을 지닌 recall 쪽으로 치우친 평균이 도출되어 큰 비중을 지닌 precision쪽의 bias가 줄어들게 된다.

3.4 비교

Threshold / Measures	0.5	0.7
Accuracy	0.739	0.745
Misclassification Rate	0.261	0.255
Sensitivity (Recall)	0.786	0.625
Specificity	0.691	0.864
Precision	0.716	0.820
F-measure (F1 score)	0.735	0.709

지표에 따라 threshold를 0.5로 설정했을 때 더 크게 나오는 지표도 있었고, threshold를 0.7로 설정했을 때 더 크게 나오는 지표도 있었다. Accuracy, misclassification, specificity, precision의 측면에선 threshold를 0.7로 설정했을 때 모델이 더 우수하다고 할 수 있지만 sensitivity와 F-measure의 측면에선 threshold를 0.5로 설정했을 때 모델이 더 우수하다고 판단할 수 있다.

그러나 해당 데이터는 당뇨병 여부 분류에 관한 모델이므로, 당뇨병을 가진 사람이 당뇨병이 없다고 오판하는 경우(False Negative)가 최대한 적어야 하며 당뇨병을 가졌다고 분류한 사람들 중 실제로 당뇨병을 가진 사람이 최대한 많아야 한다(True Positive).

즉 실제로 당뇨병이 있는 사람들 중 모델이 분류에 성공한 비율인 Sensitivity와 당뇨병이 있을 것이라 분류한 사람들 중 실제로 당뇨병을 가진 사람들의 비율인 Precision이 중요한 지표라 판단하였다.

Sensitivity의 경우 threshold를 0.5로 설정했을 때 0.768, 0.7로 설정했을 때 0.625로 0.5로 설정했을 때가 더 높게 나타났지만 Precision의 경우 반대로 threshold를 0.7로 설정했을 때 0.820으로 0.5로 설정했을 때보다 더 높게 나타났다.

따라서 Sensitivity와 Precision의 조화평균인 F-measure를 최종적으로 threshold 선정에 가장 유의미한 지표라 판단하였다. F-measure의 경우 threshold를 0.5로 설정했을 때 0.735로 threshold를 0.7로 설정했을 때보다 더 높게 나타났다.

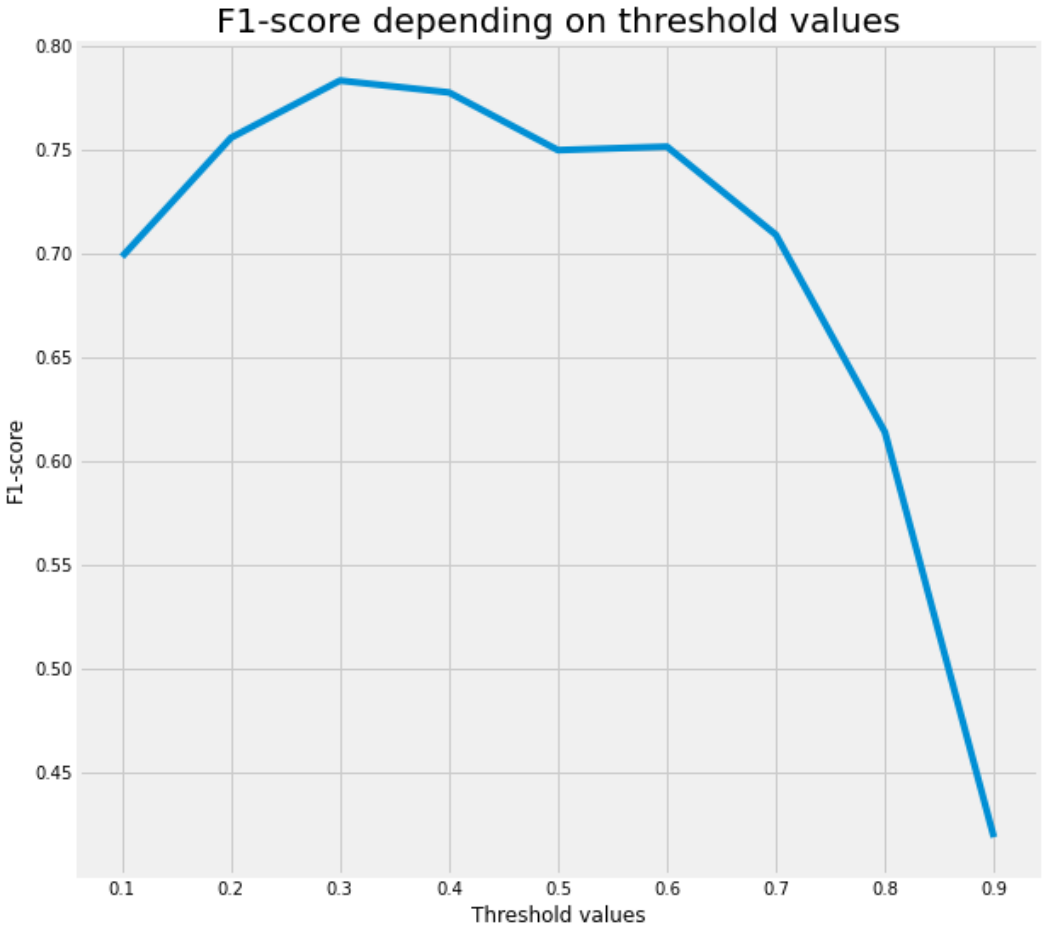
이에 따라 해당 모델은 threshold를 0.5로 설정하는 것이 더 적합하다고 판단했다.

4. 결론

당뇨병 여부에 대한 데이터를 통해 로지스틱 회귀 모델을 만들고 여러가지 지표를 통해 해당 모델의 성능을 평가해본 결과, 모델이 데이터를 적절히 분류하고 있다고 판단할 수 있었다.

모델의 분류 기준, threshold를 데이터의 특성에 맞게 적절하게 선정하기 위해 F-measure를 가장 중요한 지표로 선택한 결과 threshold가 0.5일 때가 0.7일때보다 더 데이터에 적합한 threshold라 판단할 수 있었다.

추가로 비교 과정에서 F-measure를 가장 중요한 지표로 선정했기 때문에, 이를 바탕으로 데이터에 가장 적합한 threshold를 찾기 위해 0.1~0.9까지 threshold중 F-measure의 값이 가장 큰 threshold를 찾고자 했다.



최종적으로 threshold가 0.3일 때 가장 F-measure가 크다는 것을 확인할 수 있었다.

이 분류 모델을 토대로 당뇨병 환자분류 진행 시 최적의 효과를 거둘 수 있을 것으로 기대된다.