IT@COOKBOOK

데이터 과학 기반의 파이썬 빅데이터 분석

Chapter 11 분류 분석

목차

학습목표

- 로지스틱 회귀의 이진 분류를 이해한다.
- 로지스틱 회귀 분석을 이용하여 질병 진단을 할 수 있다.
- 결정 트리의 다중 분류를 이해한다.
- 결정 트리 분석을 이용하여 움직임을 분류할 수 있다.

01. [로지스틱 회귀 분석] 특징 데이터로 유방암 진단하기

■ 분석 미리보기

특징 데이터로 유방암 진단하기	
목표	로지스틱 회귀 분석을 이용해 유방암에 영향을 미치는 특징 데이터를 분석하고 유방암 여부를 진단하는 예측 모델을 생성한다.
핵심 개념	로지스틱 회귀, 시그모이드 함수, 성능 평가 지표, 오차 행렬, 정밀도, 재현율, F1 스코어, ROC 기반 AUC 스코어
데이터 준비	유방암 진단 데이터: 사이킷런 내장 데이터셋
데이터 탐색	1. 사이킷런 데이터셋에서 제공하는 설명 확인: b_cancer.DESCR 2. 사이킷런 데이터셋에 지정된 X 피처와 타깃 피처 결합 3. 로지스틱 회귀 분석을 위해 X 피처 값을 정규 분포 형태로 스케일링: b_cancer_scaled = scaler.fit_transform(b_cancer.data)
분석 모델 구축	사이킷런의 로지스틱 회귀 모델 구축
결과 분석	성능 평가 지표 계산: confusion_matrix, accuracy_score, precision_score, recall_score, f1_score, roc_auc_score

01. [로지스틱 회귀 분석] 특징 데이터로 유방암 진단하기

■ 목표설정

- 목표: 유방암 특징을 측정한 데이터에 로지스틱 회귀 분석을 수행하여 유방암 발생을 예측

■ 핵심 개념 이해

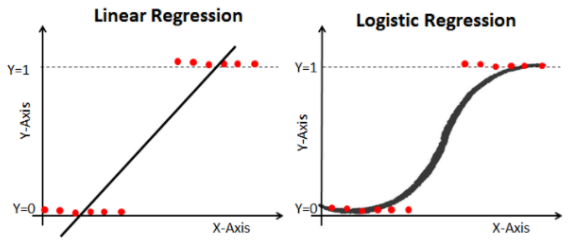
▪ 로지스틱 회귀

- 분류에 사용하는 기법으로 선형 회귀와 달리 S자 함수를 사용하여 참(True, 1)과 거짓(False, 0)을 분류

▪ 시그모이드 함수

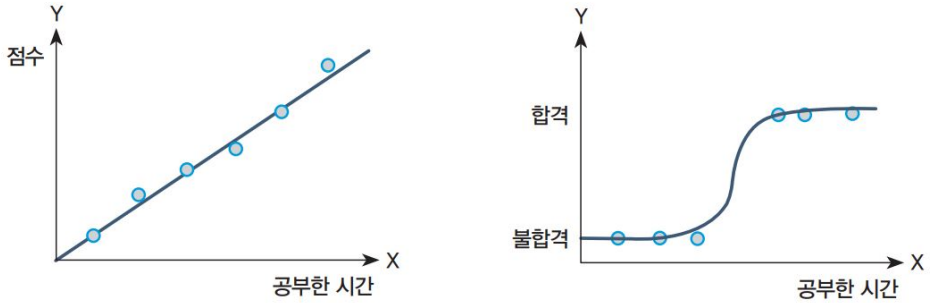
- 로지스틱 회귀에서 사용하는 S자 함수
- x의 값이 커지면 y의 값은 1에 근사하게 되고 x의 값이 작아지면 y의 값은 0에 근사하게 되어 S자 형태의 그래프가 됨
- 두 개의 값을 분류하는 이진 분류에 많이 사용

- 방정식 $y = \frac{1}{1+e^{ax+b}}$



공부 시간	1	3	5	7	9	11
점수	40	55	65	70	80	95

공부 시간	1	3	5	7	9	11
점수	불합격	불합격	불합격	합격	합격	합격



(a) 선형 회귀와 선형 함수 (b) 로지스틱 회귀와 S자 함수

그림 11-1 선형 회귀와 로지스틱 회귀 비교

01. [로지스틱 회귀 분석] 특징 데이터로 유방암 진단하기

■ 핵심 개념 이해

▪ 로지스틱 회귀

- 선형 회귀 모델은 실제값과 예측값의 오차에 기반한 지표를 사용
- 로지스틱 회귀 모델은 이진 분류 결과를 평가하기 위해 오차 행렬에 기반한 성능 지표인 정밀도, 재현율, F1 스코어, ROC_AUC를 사용

선형 회귀분석 : 종속변수가 연속형(숫자)

로지스틱 회귀분석 : 종속형 변수가 범주형이며 0/1 값을 갖는 경우

로지스틱회귀 : https://nittaku.tistory.com/478

logit : https://opentutorials.org/module/3653/22995

01. [로지스틱 회귀 분석] 특징 데이터로 유방암 진단하기

■ 핵심 개념 이해

▪ 오차 행렬(혼동행렬, Confusion Matrix)

- 행렬을 사용해 이진 분류의 예측 오류를 나타내는 지표,
- 사이킷런에서는 오차 행렬을 구하기 위해 confusion_matrix 함수를 제공
- 행은 실제 클래스의 Negative/Positive 값 / 열은 예측 클래스의 Negative/ Positive

  TN: Negative가 참인 경우 TP: Positive가 참인 경우

  FN: Negative가 거짓인 경우 FP: Positive가 거짓인 경우

- `

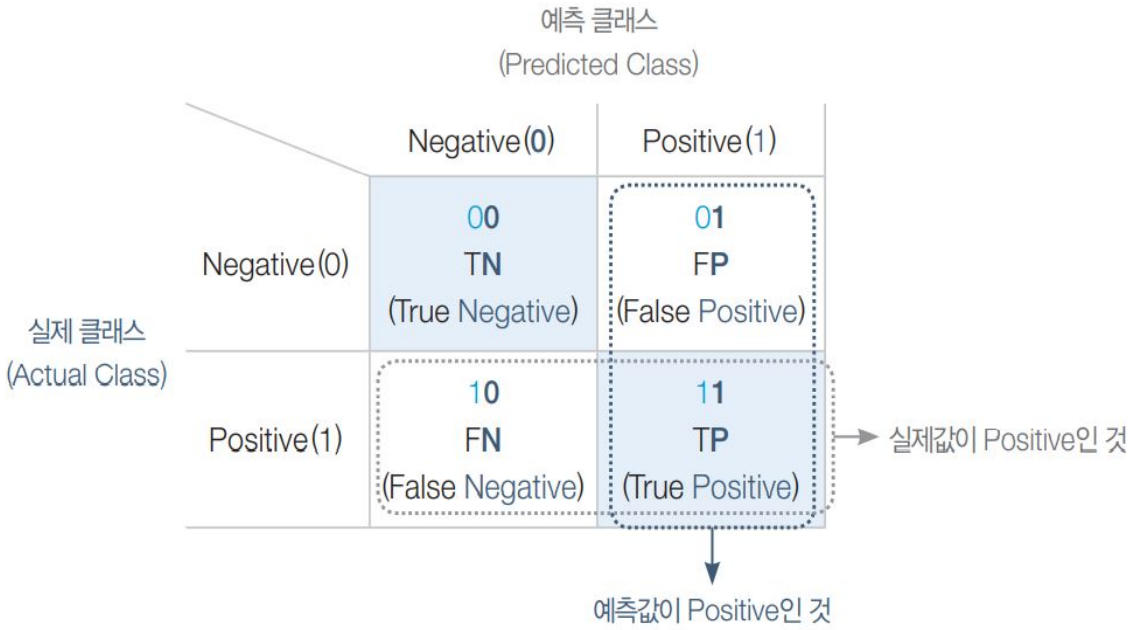


그림 11-2 오차 행렬

01. [로지스틱 회귀 분석] 특징 데이터로 유방암 진단하기

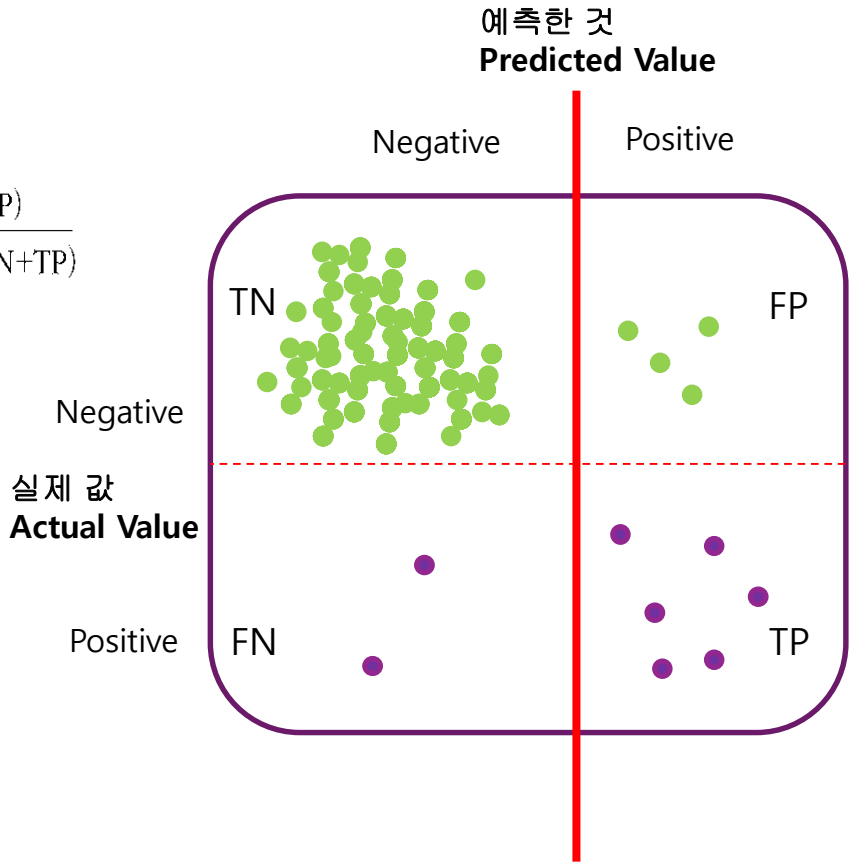
■ 핵심 개념 이해

▪ 정확도(accuracy)

- 정확도 $= \frac{\text{예측 결과와 실제값이 동일한 건수}}{\text{전체 데이터 수}} = \frac{(TN+TP)}{(TN+FP+FN+TP)}$

정확도 예
100명중 8명의 암환자가 있을 때 진단을 통하여 10명이 암이라고 판정했다. 10명중 6명은 실제 암이고 4명은 암환자가 아니었다.
정확도= (88+6)/100=0.94

	진단 Negative	진단 Positive
실제 Negative(90)	TN=88	FP=4
실제 Positive(10)	FN= 2	TP=6



01. [로지스틱 회귀 분석] 특징 데이터로 유방암 진단하기

■ 핵심 개념 이해

	진단 Negative	진단 Positive
실제 Negative(90)	TN=88	FP=4
실제 Positive(10)	FN= 2	TP=6

100명중 8명의 암환자가 있을 때 진단을 통하여 10명이 암이라고 판정했다. 10명중 6명은 실제 암이고 4명은 암환자가 아니었다.

▪ 정밀도(precision)

- 예측이 Positive인 것(FP+TP) 중에서, 참인 것(TP)의 비율을 의미
- 정밀도는 Positive 예측 성능을 더 정밀하게 평가하기 위한 지표로 사용
- 사이킷런에서는 정밀도를 구하기 위해 precision_score 함수를 제공

• 정밀도 $= \frac{TP}{(FP+TP)}$

정밀도 = TP/(FP+TP) = 6/10 = 0.6

▪ 재현율(recall) , 민감도(sensitivity), TPR(true positive rate)

재현률 = TP/(FN+TP) = 6/(2+6)=6/8 = 0.75

- 실제값이 Positive인 것(FN+TP) 중에서 참인 것(TP)의 비율을 의미
- 실제 Positive인 데이터를 정확히 예측했는지 평가하는 지표 (민감도 또는 TPR)
- 사이킷런에서는 재현율을 구하기 위해 recall_score 함수를 제공

• 재현율 $= \frac{TP}{(FN+TP)}$

▪ F1 스코어

F1 score= 2(0.6*0.75) / (0.6+0.75) = 0.78

- 정밀도와 재현율을 결합한 평가 지표
- 정밀도와 재현율이 서로 트레이드 오프 관계(상충 관계)인 문제점을 고려하여 정확한 평가를 위해 많이 사용
- 사이킷런에서는 F1 스코어를 구하기 위해 f1_score 함수를 제공

• F1 스코어 $= \frac{2}{\frac{1}{재현율}+\frac{1}{정밀도}} = 2\times\frac{정밀도\times재현율}{정밀도+재현율}$

01. [로지스틱 회귀 분석] 특징 데이터로 유방암 진단하기

■ 핵심 개념 이해

▪ 정밀도(precision) 비교

▪ 암이라고 추정하는 진단 숫자 (FP+TP=10)

	진단 Negative	진단 Positive
실제 Negative(92)	TN=88	FP=4
실제 Positive(8)	FN= 2	TP=6

100명중 8명의 암환자가 있을 때 진단을 통하여 10명이 암이라고 판정했다. 10명중 6명은 실제 암이고 4명은 암환자가 아니었다.

정확도 = (TN+TP)/T = (88+6)/100 = 0.94
정밀도 = TP/(FP+TP) = 6/10 = 0.6
재현률 = TP/(FN+TP) = 6/(2+6)=6/8 = 0.75
F1 score= 2(0.6*0.75) / (0.6+0.75) = 0.78

▪ 암이라고 추정하는 진단 숫자 (FP+TP=20)

	진단 Negative	진단 Positive
실제 Negative(92)	TN=79	FP=13
실제 Positive(8)	FN= 1	TP=7

100명중 8명의 암환자가 있을 때 진단을 통하여 20명이 암이라고 판정했다. 10명중 7명은 실제 암이고 13명은 암환자가 아니었다.

정확도 = (TN+TP)/T = (79+7)/100 = 0.86
정밀도 = TP/(FP+TP) = 7/20 = 0.35
재현률 = TP/(FN+TP) = 7/(1+7) = 0.89
F1 score= 2(0.35*0.89) / (0.35+0.89) = 0.50

▪ 암이라고 추정하는 진단 숫자 (FP+TP=30)

	진단 Negative	진단 Positive
실제 Negative(92)	TN=70	FP=22
실제 Positive(8)	FN= 0	TP=8

100명중 8명의 암환자가 있을 때 진단을 통하여 30명이 암이라고 판정했다. 10명중 8명은 실제 암이고 22명은 암환자가 아니었다.

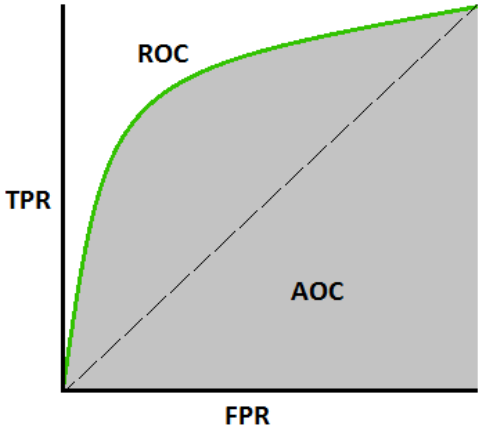
정확도 = (TN+TP)/T = (79+7)/100 = 0.78
정밀도 = TP/(FP+TP) = 8/30 = 0.27
재현률 = TP/(FN+TP) = 8/(0+8) = 1.0
F1 score= 2(0.27*1.0) / (0.27+1.0) = 0.42

01. [로지스틱 회귀 분석] 특징 데이터로 유방암 진단하기

■ 핵심 개념 이해

▪ ROC 기반 AUC 스코어 (ROC: receiver operating characteristic, AUC: area under the curve)

- 오차 행렬의 FPR이 변할 때 TPR이 어떻게 변하는지를 나타내는 곡선
  - » FPR : 실제 Negative인 데이터를 Positive로 거짓(False)으로 예측한 비율(specificity(특이도) = 1- FPR = TN/(TN+FP))
    - FPR $= \frac{FP}{(FP+TN)}$
  - » TPR: 실제 Positive인 데이터를 Positive로 예측한 비율(=**재현율**) $= \frac{TP}{(FN+TP)}$
- ROC 기반의 AUC 값은 ROC 곡선 밑의 면적을 구한 것으로 1에 가까울수록 좋은 성능을 의미
- AUC 값이 커지려면 FPR이 작을때 TPR이 커야함
- 사이킷런에서는 ROC 기반의 AUC를 구하기 위해 roc_auc_score 함수를 제공



	진단Negative(90)	진단Positive(10)
실제Negative(92)	TN=88	FP=4
실제Positive(8)	FN= 2	TP=6

FPR = FP/(FP+TN) = 4/(4+88)
TPR(재현률) = TP/(FN+TP) = 6/(2+6)

Understanding AUC-ROC : https://towardsdatascience.com/understanding-auc-roc-curve-68b2303cc9c5
AUC-ROC Curve : https://bioinformaticsandme.tistory.com/328

01. [로지스틱 회귀 분석] 특징 데이터로 유방암 진단하기

■ 핵심 개념 이해

▪ ROC 기반 AUC 스코어 (ROC: receiver operating characteristic, AUC: area under the curve)

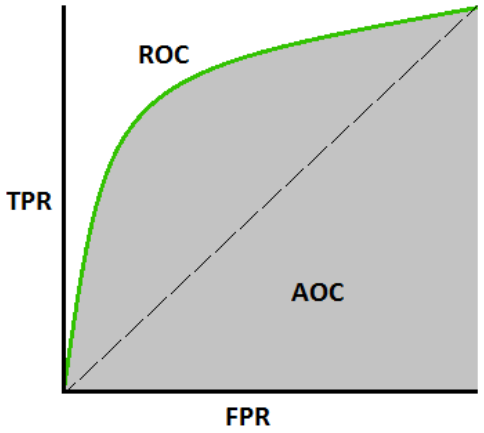
- TPR 과 FPR 은 서로 비례
  조금만 1일꺼 같아도 1이라고 판정하는 모델은 TPR, FPR 함께상승. 반대도 동일

- 단점, 잘예측한 TPR 높이면 잘못예측한 FPR 또한 높아짐
  따라서 FPR 대비 TPR 좋게 나오도록 => 시각화 한것이 ROC 커브

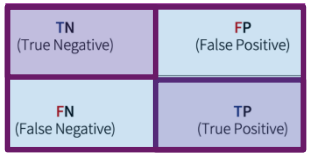
- ROC curve (Receiver Operating Characteristics)
  장점 : 면적을 측정하여 TPR FPR 복합적으로 평가가능
  면적을 area under the curve:AUC 라고 하고 1에 가까울수록 성능이 좋고, 0.5에 가까울수록 성능이 나쁨

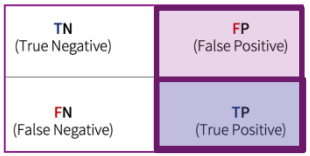
- FPR , TPR , thresholds 해석
  case1 : 의사가 모든 환자들을 암 확률이 어느정도만 되어도(threshold가 낮음)
  암 환자로 판정하면 TPR(실제 암이걸린환자를 암이라 판정)과
  FPR(암이걸리지 않았지만 암이라 판정) 이 함께 높아짐
  => threshold 낮다 => TPR & FPR 비율 높음
  case2 : 의사가 모든 환자들을 암 확률이 매우높아야만(threshold가 높음)
  암 환자로 판정하면 TPR(실제 암이걸린환자를 암이라 판정)과
  FPR(암이걸리지 않았지만 암이라 판정) 이 함께 낮아진다.
  => threshold 높다 => TPR & FPR 비율 낮음

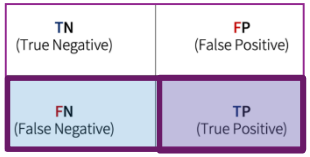
- 따라서 threshold 를 낮출수록 TPR & FPR 비율이 높아지고,
  threshold 를 높일수록 TPR & FPR 비율이 낮아진다.
  그리고 보통 FPR, TPR 순서로 확률 분포가 위치하여 threshold를 낮추면
  TPR이 먼저 증가하고, threshold를 높이면 FPR이 먼저 사라진다.

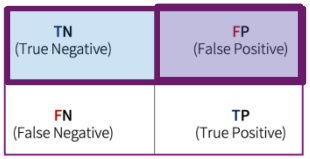


분류성능 측정하는법 : https://wotres.tistory.com/entry/%EB%B6%84%EB%A5%98-%EC%84%B1%EB%8A%A5-%EC%B8%A1%EC%A0%95%ED%95%98%EB%8A%94%EB%B2%95-Accuracy-Precision-Recall-F1-score-ROC-AUC-in-python?category=930448









(정리)

분자

(분모)

TN (True Negative)	**FP** (False Positive)
FN (False Negative)	**TP** (True Positive)

TN (True Negative)	**FP** (False Positive)
FN (False Negative)	**TP** (True Positive)

TN (True Negative)	**FP** (False Positive)
FN (False Negative)	**TP** (True Positive)

TN (True Negative)	**FP** (False Positive)
FN (False Negative)	**TP** (True Positive)

01. [로지스틱 회귀 분석] 특징 데이터로 유방암 진단하기

■ 데이터 준비 및 탐색

- 사이킷런에서 제공하는 데이터셋

표 11-1 사이킷런에서 제공하는 주요 데이터셋

데이터셋	샘플 갯수	독립 변수	종속 변수	데이터 로드 함수
보스톤 주택 가격 데이터	506	13개	주택 가격	load_boston()
붓꽃(아이리스) 데이터	150	4개	붓꽃 종류: setosa, versicolor, virginica	load_iris()
당뇨병 환자 데이터	442	10개	당뇨병 수치	load_diabetes()
숫자 0~9를 손으로 쓴 흑백 데이터	1797	64개	숫자: 0~9	load_digits()
와인의 화학 성분 데이터	178	13개	와인 종류: 0, 1, 2	load_wine()
체력 검사 데이터	20	3개	체력 검사 점수	load_linnerud()
유방암 진단 데이터	569	30개	악성(malignant), 양성(benign): 1, 0	load_ breast_ cancer()

01. [로지스틱 회귀 분석] 특징 데이터로 유방암 진단하기

■ 데이터 준비 및 탐색

- 유방암 진단 데이터

특성	설명
id	환자 식별 번호
dignosis	유방암 종양(M=악성, B=양성)
radius	세포의 크기
texture	질감(흑백 처리했을때의 표준편차 값으로 계산)
perimeter	둘레
area	면적
smoothness	매끄러움(반경의 국소적 변화 측정)
compactness	작은 정도(perimeter2/area−1perimeter2/area−1로 계산)
concavity	오목함(윤곽의 오목한 부분의 정도)
concave points	오목한곳의 수
symmetry	대칭성
fractal dimension	프랙탈 차원(coastlineapproximation−1coastlineapproximation−1로 계산)

01. [로지스틱 회귀 분석] 특징 데이터로 유방암 진단하기

■ 데이터 준비 및 탐색

- 사이킷런의 유방암 진단 데이터셋 사용하기

1. 데이터 준비하기

In [1]:	**import** numpy **as** np **import** pandas **as** pd **from** sklearn.datasets **import load_breast_cancer**
In [2]:	b_cancer = **load_breast_cancer()**

In [1]: 사이킷런에서 제공하는 데이터셋sklearn.datasets중에서 유방암 진단 데이터셋을 사용하기 위해 load_breast_cancer를 임포트

In [2]: 데이터셋을 로드하여 객체b_cancer를 생성

01. [로지스틱 회귀 분석] 특징 데이터로 유방암 진단하기

■ 데이터 준비 및 탐색

- 사이킷런의 유방암 진단 데이터셋 사용하기

2. 데이터 탐색하기

In [3]:
```
print(b_cancer.DESCR)
```

In [4]:
```
b_cancer_df = pd.DataFrame(b_cancer.data, columns = b_cancer.feature_names)
```

In [5]:
```
b_cancer_df['diagnosis']= b_cancer.target
```

In [6]:
```
b_cancer_df.head()
```

Out:[6]

	mean radius	mean texture	mean perimeter	mean area	mean smoothness	mean compactness	mean concavity	mean concave points	mean symmetry	mean fractal dimension	...	worst texture	worst perimeter	worst area	worst smoothness
0	17.99	10.38	122.80	1001.0	0.11840	0.27760	0.3001	0.14710	0.2419	0.07871	...	17.33	184.60	2019.0	0.1622
1	20.57	17.77	132.90	1326.0	0.08474	0.07864	0.0869	0.07017	0.1812	0.05667	...	23.41	158.80	1956.0	0.1238
2	19.69	21.25	130.00	1203.0	0.10960	0.15990	0.1974	0.12790	0.2069	0.05999	...	25.53	152.50	1709.0	0.1444
3	11.42	20.38	77.58	386.1	0.14250	0.28390	0.2414	0.10520	0.2597	0.09744	...	26.50	98.87	567.7	0.2098
4	20.29	14.34	135.10	1297.0	0.10030	0.13280	0.1980	0.10430	0.1809	0.05883	...	16.67	152.20	1575.0	0.1374

5 rows × 31 columns

In [3]: 데이터셋에 대한 설명을 확인

In [4]: 데이터셋 객체의 data 배열b_cancer.data, 즉 독립 변수 X가 되는 피처를 DataFrame 자료형으로 변환하여 b_cancer_df를 생성

In [5]: 유방암 유무 class로 사용할 diagnosis 컬럼을 b_cancer_df에 추가하고 데이터셋 객체의 target 컬럼b_cancer.target을 저장

In [6]: b_cancer_df의 데이터 샘플 5개를 출력b_cancer_df.head( )하여 확인

01. [로지스틱 회귀 분석] 특징 데이터로 유방암 진단하기

■ 데이터 준비 및 탐색

- 사이킷런의 유방암 진단 데이터셋 사용하기

3. 데이터셋의 크기와 독립 변수 X가 되는 피처에 대한 정보를 확인

In [7]:	print('유방암 진단 데이터셋 크기: ', b_cancer_df.shape)
Out:[7]	유방암 진단 데이터셋 크기: (569, 31)
In [8]:	b_cancer_df.head()
Out:[8]	<class 'pandas.core.frame.DataFrame'> RangeIndex: 569 entries, 0 to 568 Data columns (total 31 columns): mean radius 569 non-null float64 mean texture 569 non-null float64 mean perimeter 569 non-null float64 mean area 569 non-null float64 mean smoothness 569 non-null float64 mean compactness 569 non-null float64 mean concavity 569 non-null float64 mean concave points 569 non-null float64 mean symmetry 569 non-null float64 mean fractal dimension 569 non-null float64 radius error 569 non-null float64 texture error 569 non-null float64 perimeter error 569 non-null float64 area error 569 non-null float64 ....

Out:[8]	.... smoothness error 569 non-null float64 compactness error 569 non-null float64 concavity error 569 non-null float64 concave points error 569 non-null float64 symmetry error 569 non-null float64 fractal dimension error 569 non-null float64 worst radius 569 non-null float64 worst texture 569 non-null float64 worst perimeter 569 non-null float64 worst area 569 non-null float64 worst smoothness 569 non-null float64 worst compactness 569 non-null float64 worst concavity 569 non-null float64 worst concave points 569 non-null float64 worst symmetry 569 non-null float64 worst fractal dimension 569 non-null float64 diagnosis 569 non-null int32 dtypes: float64(30), int32(1) memory usage: 135.7 KB

In [7]: b_cancer_df.shape를 사용하여 데이터셋의 행의 개수(데이터 샘플 개수)와 열의 개수(변수 개수)를 확인
행의 개수가 569이므로 데이터 샘플이 569개, 열의 개수가 31이므로 변수가 31개 있음

In [8]: b_cancer_df에 대한 정보를 확인b_cancer_df.info( ) / 30개의 피처(독립 변수 X) 이름과 1개의 종속 변수 이름을 확인 가능
diagnosis는 악성이면 1, 양성이면 0의 값이므로 유방암 여부에 대한 이진 분류의 class로 사용할 종속 변수가 됨

01. [로지스틱 회귀 분석] 특징 데이터로 유방암 진단하기

■ 데이터 준비 및 탐색

▪ 사이킷런의 유방암 진단 데이터셋 사용하기

4. 로지스틱 회귀 분석에 피처로 사용할 데이터를 평균이 0, 분산이 1이 되는 정규 분포 형태로 맞춤

In [9]:	from sklearn.preprocessing import StandardScaler scaler = StandardScaler()
In [10]:	b_cancer_scaled = scaler.fit_transform(b_cancer.data)
In [11]:	print(b_cancer.data[0])
Out:[11]	[1.799e+01 1.038e+01 1.228e+02 1.001e+03 1.184e-01 2.776e-01 3.001e-01 1.471e-01 2.419e-01 7.871e-02 1.095e+00 9.053e-01 8.589e+00 1.534e+02 6.399e-03 4.904e-02 5.373e-02 1.587e-02 3.003e-02 6.193e-03 2.538e+01 1.733e+01 1.846e+02 2.019e+03 1.622e-01 6.656e-01 7.119e-01 2.654e-01 4.601e-01 1.189e-01]
In [12]:	print(b_cancer_scaled[0])
Out:[12]	[1.09706398 -2.07333501 1.26993369 0.9843749 1.56846633 3.28351467 2.65287398 2.53247522 2.21751501 2.25574689 2.48973393 -0.56526506 2.83303087 2.48757756 -0.21400165 1.31686157 0.72402616 0.66081994 1.14875667 0.90708308 1.88668963 -1.35929347 2.30360062 2.00123749 1.30768627 2.61666502 2.10952635 2.29607613 2.75062224 1.93701461]

In [9]: 사이킷런의 전처리 패키지에 있는 정규 분포 스케일러를 임포트하고 사용할 객체scaler를 생성

In [10]: 피처로 사용할 데이터b_cancer.data에 대해 정규 분포 스케일링을 수행scaler.fit_ transform( )하여 b_cancer_scaled에 저장

In [11]~[12]: 정규 분포 스케일링 후에 값이 조정된 것을 확인

01. [로지스틱 회귀 분석] 특징 데이터로 유방암 진단하기

■ 분석 모델 구축 및 결과 분석

1. 로지스틱 회귀를 이용하여 분석 모델 구축하기

In [13]:	**from** sklearn.linear_model import **LogisticRegression** **from** sklearn.model_selection **import** train_test_split
In [14]:	*#X, Y 설정하기* Y = b_cancer_df['diagnosis'] X = b_cancer_scaled
In [15]:	*#훈련용 데이터와 평가용 데이터 분할하기* X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size = 0.3, random_state = 0)
In [16]:	*#로지스틱 회귀 분석: (1) 모델 생성* lr_b_cancer = **LogisticRegression()**
In [17]:	*#로지스틱 회귀 분석: (2) 모델 훈련* lr_b_cancer.fit(X_train, Y_train)
Out:[17]	LogisticRegression()
In [18]:	*#로지스틱 회귀 분석: (3) 평가 데이터에 대한 예측 수행 -> 예측 결과 Y_predict 구하기* Y_predict = lr_b_cancer.**predict**(X_test)

In [13]: 필요한 모듈을 임포트

In [14]: diagnosis를 Y, 정규 분포로 스케일링한 b_cancer_scaled를 X로 설정

In [15]: 전체 데이터 샘플 569개를 학습 데이터:평가 데이터=7:3으로 분할test_size=0.3함

In [16]: 로지스틱 회귀 분석 모델 객체lr_b_cancer를 생성

In [17]: 학습 데이터X_train, Y_train로 모델 학습을 수행fit( )함

In [18]: 학습이 끝난 모델에 대해 평가 데이터 XX_test를 가지고 예측을 수행predict( )하여 예측값 YY_predict를 구함

01. [로지스틱 회귀 분석] 특징 데이터로 유방암 진단하기

■ 분석 모델 구축 및 결과 분석

▪ 로지스틱 회귀를 이용하여 분석 모델 구축하기

- 주피터 노트북 버전 확인
  - 실습하는 아나콘다의 주피터 노트북 버전에 따라 실행 결과가 조금 다르게 나타날 수 있음
  - 노트북 화면 상단의 [Help]- [About] 메뉴에서 확인

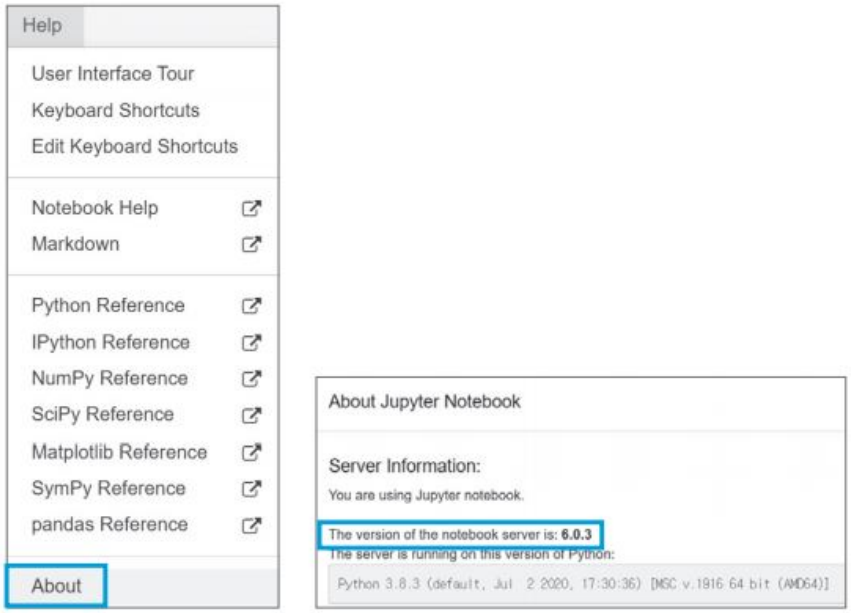


그림 11-3 주피터 노트북 버전 확인

01. [로지스틱 회귀 분석] 특징 데이터로 유방암 진단하기

■ 분석 모델 구축 및 결과 분석

2. 생성한 모델의 성능 확인하기

In [19]:	from sklearn.metrics import confusion_matrix, accuracy_score from sklearn.metrics import precision_score, recall_score, f1_score, roc_auc_score
In [20]:	*#오차 행렬* confusion_matrix(Y_test, Y_predict)
Out:[20]	array([[60, 3], [1, 107]], dtype = int64)
In [21]:	acccuracy = accuracy_score(Y_test, Y_predict) precision = precision_score(Y_test, Y_predict) recall = recall_score(Y_test, Y_predict) f1 = f1_score(Y_test, Y_predict) roc_auc = roc_auc_score(Y_test, Y_predict)
In [22]:	print('정확도: {0:.3f}, 정밀도: {1:.3f}, 재현율: {2:.3f}, F1: {3:.3f}'.format(acccuracy,precision,recall,f1))
Out:[22]	정확도: 0.977, 정밀도: 0.973, 재현율: 0.991, F1: 0.982
In [23]:	print('ROC_AUC: {0:.3f}'.format(roc_auc))
Out:[23]	ROC_AUC: 0.972

In [19]: 필요한 모듈을 임포트

In [20]: 평가를 위해 7:3으로 분할한 171개의 test 데이터에 대해 이진 분류의 성능 평가 기본이 되는 오차 행렬을 구함
실행 결과를 보면 TN이 60개, FP가 3개, FN이 1개, TP가 107개인 오차 행렬이 구해짐

In [21]: 성능 평가 지표인 정확도, 정밀도, 재현율, F1 스코어, ROC-AUC 스코어를 구함

In [22]~[23]: 성능 평가 지표를 출력하여 확인