Multivariate Data Analysis Assignment #2

Logistic Regression & Dimensionality Reduction

산업경영공학부 2020170856 이우진

[Q1] 본인이 스스로 Logistic Regression 모델을 적용하는데 적합한 데이터셋을 선정하고 선정 이유를 설명하시오. 데이터셋 탐색은 아래에서 제시된 Data Repository를 포함하여 여러 Repository를 검색해서 결정하시오. 보고서에는 데이터를 다운로드할 수 있는 링크를 반드시 제공하시오.

Dataset : **Breast Cancer Data Set**

다운로드 링크 : https://www.kaggle.com/datasets/erdemtaha/cancer-data

선정 이유 : Logistic Regression을 이용해 분류 문제를 해결해야 하므로, 범주형의 종속변수를 가지며, 설명변수와 높은 관련성이 있을 것으로 예상되는 데이터셋을 선정하였다. 해당 데이터셋은 악성 종양, 양성 종양을 나타내는 범주형 종속변수를 가지며, 설명 변수로는 종양의 반지름, 둘레, 면적, 조밀도 등 종양의 여러 특징들을 가지고 있다. 이러한 설명변수들은 악성 종양과 양성 종양 간에 뚜렷한 차이를 보일 것으로 예상된다. 따라서 설명 변수들이 종속 변수인 종양의 유형(악성, 양성)을 진단하는 데 높은 관련성을 보일 것으로 생각하였고, 로지스틱 회귀 분석에 적합한 데이터셋이라고 판단하였다. 또한, 종속 변수로 사용할 diagnosis를 제외하고, 총 30개의 설명 변수들을 포함하고 있으며, 569개의 적지 않은 관측치를 가지고 있기에, 해당 데이터셋을 선정하였다.

[Q2] 해당 데이터셋의 종속변수와 설명변수는 어떤 것들이 있는가? 분석 전에 아래 두 가지 질문에 대해서 스스로 생각해보고 답변을 하시오.

해당 데이터 셋에는 1개의 종속 변수와, ID를 제외한 30개의 설명 변수가 있다.

<종속 변수>

- diagnosis : 종양의 진단 (악성-M, 양성-B)

  본 과제에서는 악성(M)을 1로, 양성(B)를 0으로 변환하여 분석을 진행하였다.

<설명 변수>

- radius_mean : 종양의 평균 반지름
- texture_mean : 종양의 평균 텍스처(standard deviation of gray-scale values)
- perimeter_mean : 종양의 평균 둘레
- area_mean : 종양 영역의 평균 면적

- smoothness_mean : 종양 표면의 평균 부드러움(local variation in radius lengths)

- compactness_mean : 종양의 평균 조밀도(perimeter^2 / area - 1.0)

- concavity_mean : 종양의 평균 오목함(severity of concave portions of the contour)

- concave points_mean : 종양의 평균 오목한 점의 수

- symmetry_mean : 종양 모양의 대칭성 평균

- fractal_dimension_mean : 종양의 평균 프랙탈 차원 (종양의 모양을 설명하는 척도)

- radius_se : 반경 표준오차

- texture_se: 텍스처 표준오차

- perimeter_se : 둘레 표준오차

- area_se: 면적 표준오차

- smoothness_se : 부드러움 표준오차

- compactness_se : 조밀도 표준오차

- concavity_se : 오목함 표준오차

- concave points_se : 오목한 점의 수 표준오차

- symmetry_se : 대칭성 표준오차

- fractal_dimension_se : 프랙탈 차원 표준오차

- radius_worst : 종양의 최대 반지름

- texture_worst : 종양의 최대 텍스처

- perimeter_worst : 종양의 최대 둘레

- area_worst : 종양 영역의 최대 면적

- smoothness_worst : 종양 표면의 최대 부드러움

- compactness_worst : 종양 모양의 최대 조밀도

- concavity_worst : 종양 윤곽의 최대 오목함

- concave points_worst : 종양 윤곽의 최대 오목한 점의 수

- symmetry_worst : 종양 모양의 최대 대칭성

- fractal_dimension_worst : 종양의 최대 프랙탈 차원

2-1. 이 데이터에서 제공된 설명변수들 중에서 높은 상관관계가 있을 것으로 예상되는 변수들은 어떤 것들이 있는가? 왜 그렇게 생각하는가?

위의 설명변수들은 아래와 같이 크게 10가지로 나눌 수 있다.

(radius, texture, perimeter, area, smoothness, compactness, concavity, concave points, symmetry, fractal_dimension)

이렇게 10개의 feature들의 평균, 표준오차, 최댓값을 각각 따로 나타냈기 때문에 총 30개의 설명변수가 만들어졌는데, 이때, 각 feature들의 mean과 worst는 높은 양의 상관관계를 나타낼 것으로 예상된다. 이는 평균과 최댓값이 일반적으로 비례 관계를 가지는 경향이 있기 때문이다.

또한, radius, perimeter, area 간의 높은 양의 상관관계가 예상된다. 종양은 원의 형태를 보이기 때문에 radius의 크기가 크다면 일반적으로 perimeter와 area의 값도 크다고 할 수 있다. 즉, radius_mean, perimeter_mean, area_mean, radius_worst, perimeter_worst, area_worst 이렇게 6개의 변수들끼리 높은 상관관계가 있을 것으로 예상된다. 그리고 악성 종양의 경우, 양성 종양보다 radius, perimeter, area가 클 것으로 예상되기 때문에 위의 변수들과 종속변수인 diagnosis도 높은 양의 상관관계가 예상된다. 해당 feature들의 se값도 diagnosis와 높은 양의 상관관계가 예상되는데, 악성 종양이 양성 종양에 비해 큰 종양과 작은 종양의 차이 즉, 편차가 크다고 생각하기 때문에 그에 따라 표준오차 se도 클 것으로 예상된다.

마지막으로, compactness, concavity, concave points도 높은 상관관계가 예상된다. compactness 값은 (perimeter^2 / area - 1.0) 식을 통해 나온 값인데, 이 값이 크다면 area에 비해 perimeter값이 크다는 것이고, 이는 종양이 볼록한 부분이 많은 일그러진 형태를 가졌다고 해석할 수 있다. 따라서 compactness 값이 클 때, 오목함을 나타내는 concavity의 값도 높을 것이고, concave points도 같이 높아질 가능성이 높다. 따라서, compactness_mean, concavity_mean, concave points_mean, compactness_worst, concavity_worst, concave points_worst 이렇게 6개의 변수들끼리 높은 양의 상관관계가 있을 것으로 예상된다.

이 외에도 종양의 특징 및 수치들이 독립적이기보다 복합적으로 연결되어 있기 때문에, 전체적으로 상관관계가 높은 변수들이 다소 나타날 것으로 예상된다.

2-2. 제공된 설명변수들 중에서 종속 변수를 예측하는데 필요하지 않을 것으로 예상되는 변수들은 어떤 것들이 있는가? 왜 그렇게 생각하는가?

종속변수 diagnosis를 예측하는데 필요하지 않을 것으로 예상되는 변수에는 크게 보면, symmetry, fractal_dimension이 있다. symmetry과 fractal_dimesion은 각각 종양의 대칭성과 종양의 프렉탈 차원 즉, 공간 패턴을 설명하는 척도이다.

해당 변수는 cancer image를 통해 나온 수치인데, 악성 종양과 양성 종양 이미지를 비교해보았을

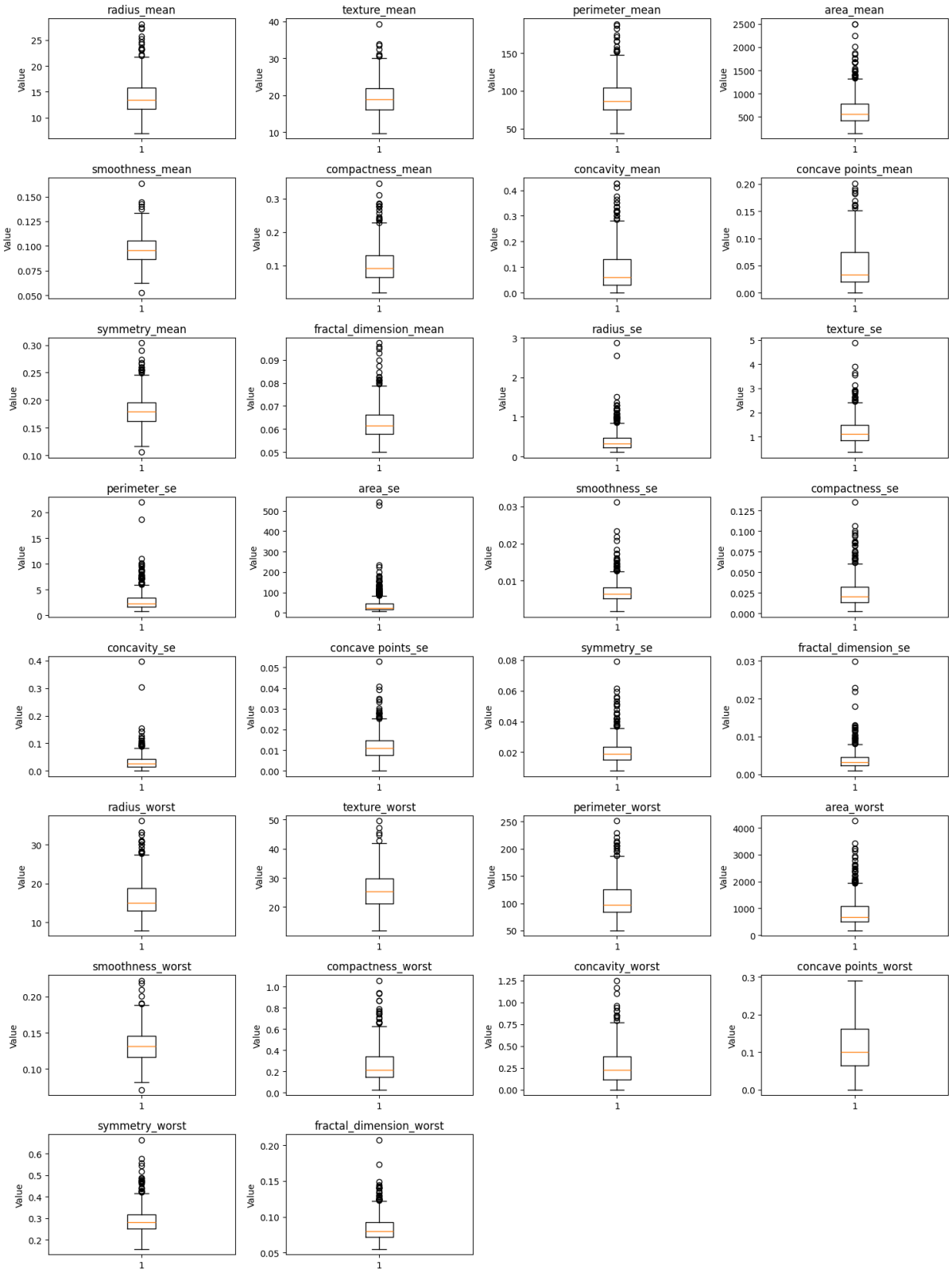
때, area 같은 특징은 뚜렷한 차이가 보이지만, 3차원이 아닌 2차원 사진으로, symmetry, fractal_dimension 차이를 확인하기엔 다소 무리가 있다고 판단된다. 또한, 악성 종양, 양성 종양 둘 다 결론적으로 같은 종양이기 때문에 유의미한 대칭성 차이, 공간 패턴 차이가 존재한다고 보기 어려울 거 같고, 만약 실제로 그 차이가 존재한다고 하더라도, 사진을 찍는 각도에 따라 결과가 달라질 수 있다고 생각하기 때문에, 위의 변수들은 종속 변수와 직접적인 상관관계가 낮을 것으로 예상된다. 따라서 symmetry_mean, symmetry_se, symmetry_worst, fractal_dimesion_mean, fractal_dimesion_se, fractal_dimesion_worst 이렇게 6개 변수는 종속변수 예측에 필요하지 않을 것으로 예상된다.

[Q3] 모든 연속형 숫자 형태를 갖는(명목형 변수 제외) 개별 입력 변수들에 대하여 각각 다음과 같은 단변량 통계량을 계산하고 Box plot을 도시하시오: Mean, Standard deviation, Skewness, Kurtosis. 전체 변수 중에서 정규 분포를 따른다고 할 수 있는 변수들은 몇 개인가? 정규분포를 따른다고 가정한 근거는 무엇인가?

개별 입력 변수들에 대한 단변량 통계량 계산을 진행하였고 결과는 아래와 같다. 이때, 명목형 변수는 제외하고 수치형 변수만을 사용하였다.

	Mean	STD	Skewness	Kurtosis
radius_mean	14.127292	3.524049	0.939893	0.827584
texture_mean	19.289649	4.301036	0.648734	0.741145
perimeter_mean	91.969033	24.298981	0.988037	0.953165
area_mean	654.889104	351.914129	1.641391	3.609761
smoothness_mean	0.096360	0.014064	0.455120	0.837945
compactness_mean	0.104341	0.052813	1.186983	1.625140
concavity_mean	0.088799	0.079720	1.397483	1.970592
concave points_mean	0.048919	0.038803	1.168090	1.046680
symmetry_mean	0.181162	0.027414	0.723695	1.266117
fractal_dimension_mean	0.062798	0.007060	1.301047	2.969017
radius_se	0.405172	0.277313	3.080464	17.521162
texture_se	1.216853	0.551648	1.642100	5.291753
perimeter_se	2.866059	2.021855	3.434530	21.203775
area_se	40.337079	45.491006	5.432816	48.767196
smoothness_se	0.007041	0.003003	2.308344	10.367537
compactness_se	0.025478	0.017908	1.897202	5.050966
concavity_se	0.031894	0.030186	5.096981	48.422562
concave points_se	0.011796	0.006170	1.440867	5.070840
symmetry_se	0.020542	0.008266	2.189342	7.816388
fractal_dimension_se	0.003795	0.002646	3.913617	26.039950
radius_worst	16.269190	4.833242	1.100205	0.925288
texture_worst	25.677223	6.146258	0.497007	0.211809
perimeter_worst	107.261213	33.602542	1.125188	1.050243
area_worst	880.583128	569.356993	1.854468	4.347331
smoothness_worst	0.132369	0.022832	0.414330	0.502760
compactness_worst	0.254265	0.157336	1.469667	3.002120
concavity_worst	0.272188	0.208624	1.147202	1.590568
concave points_worst	0.114606	0.065732	0.491316	-0.541367
symmetry_worst	0.290076	0.061867	1.430145	4.395073
fractal_dimension_worst	0.083946	0.018061	1.658193	5.188111

각 변수들에 대한 box plot은 아래와 같다.



먼저, Skewness와 Kurtosis를 기준으로 정규분포 가정을 확인하려면, 두 값이 0에 가까워야한다. 그러나 이 통계만으로 해당 변수가 정규분포를 따르는지 완벽하게 판단하기는 어렵기 때문에 boxplot의 형태도 같이 확인하였다.

'절댓값 2 이하의 Skewness'와 '절댓값 6 이하의 Kurtosis'를 동시에 만족하는 변수가 정규분포를 따른다고 했을 때, 'radius_se', 'perimeter_se', 'area_se', 'smoothness_se', 'concavity_se', 'symmetry_se', 'fractal_dimension_se'를 제외한 나머지 변수 23개가 정규 분포를 따른다고 할 수 있다.

이때, 한 가지 기준만으로 정규성을 판단하기 어렵기 때문에 boxplot의 형태도 종합적으로 고려해보았고, 위에서 정규 분포를 따른다고 할 수 있는 변수 중, 'concavity_mean', 'concave points_mean', 'area_worst', 'compactness_worst', 'concavity_worst' 이렇게 5개의 변수는 다소 아래로 치우친 box plot을 보인다는 점에서 정규 분포를 따른다고 보기 어렵다고 판단된다.

따라서, 총 30개의 변수 중 'radius_se', 'perimeter_se', 'area_se', 'smoothness_se', 'concavity_se', 'symmetry_se', 'fractal_dimension_se', 'concavity_mean', 'concave points_mean', 'area_worst', 'compactness_worst', 'concavity_worst' 이렇게 12개를 제외한 18개의 변수가 정규 분포를 따른다고 할 수 있다.

[Q4] [Q3]의 Box plot을 근거로 각 변수들에 대한 이상치(값이 너무 크거나 작은 객체) 조건을 정의하고, 해당하는 객체들을 데이터셋에서 제거해 보시오.

위의 Box plot을 보았을 때, 대부분의 설명변수에서 상위 범위에 이상치들이 많이 분포하고 있음을 확인할 수 있다. 이러한 이상치들은 모델 학습에 있어서 다른 데이터들에 비해 회귀계수(coef) 변화에 큰 영향을 미치기 때문에 최적해를 찾는 것이 더 어려워질 수 있다. 따라서 이들을 제거하는 것이 성능 향상에 더 도움이 될 수 있다.

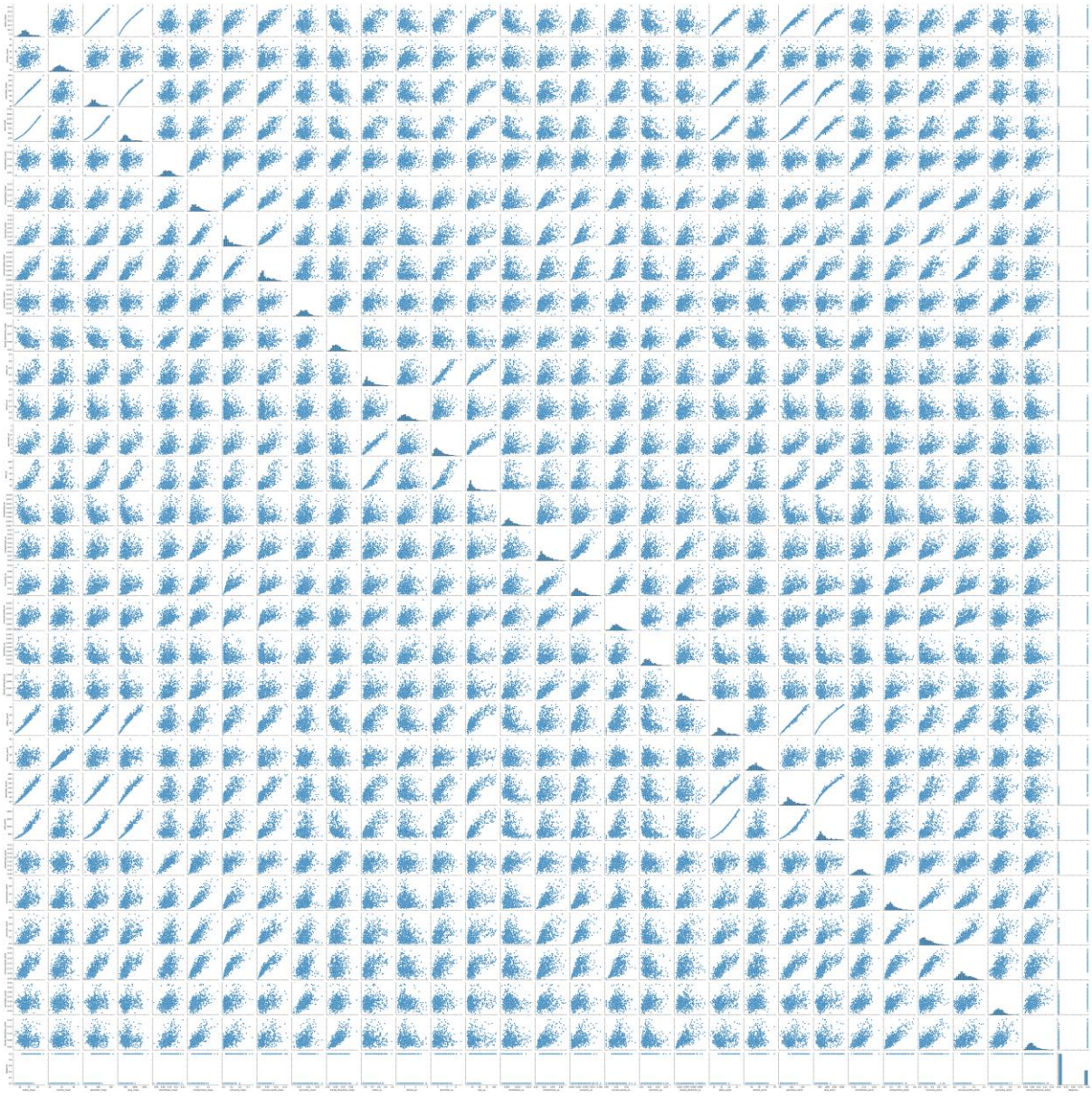
Q1을 1사분위수, Q3를 3사분위수라고 할 때, 보통 Q1, Q3에서 IQR(Q3-Q1)의 1.5배 떨어진 거리를 최소, 최대로 정의하고 이 밖의 값들은 이상치(outlier)라고 간주한다. 즉, boxplot에서 구간 (Q1 – 1.5 * IQR , Q3 +1.5 * IQR)에 포함되지 않는 데이터를 이상치로 정의하였고, 데이터셋에서 이상치를 제거해보았다. 그 결과 569개의 데이터에서 277개의 데이터만 남았고, 약 50%에 해당하는 데이터가 제거되었다. 이 경우 분석에 필요한 중요한 데이터들을 많이 잃게 되므로, 제거된 데이터셋의 비율이 크다는 점을 고려하여, 1.5배 대신 3배를 초과하는 극단적 이상치를 제거하는 방식을 택하였다. 그 결과 569개의 데이터 중 494개의 데이터가 남았다.

다음 각 물음에 대해서는 [Q4]에서 제거된 객체들을 제외시킨 데이터프레임을 사용하여 답하시오.

[Q5] 가능한 모든 두 쌍의 입력 변수 조합에 대한 산점도(scatter plot)을 도시하고 적절한 정량적 지표를 사용하여 상관관계를 판단해 보시오.

변수 간 상관관계를 Scatter plot으로 시각화 해보았고, 그 결과는 아래와 같다.

▶ Scatter plot

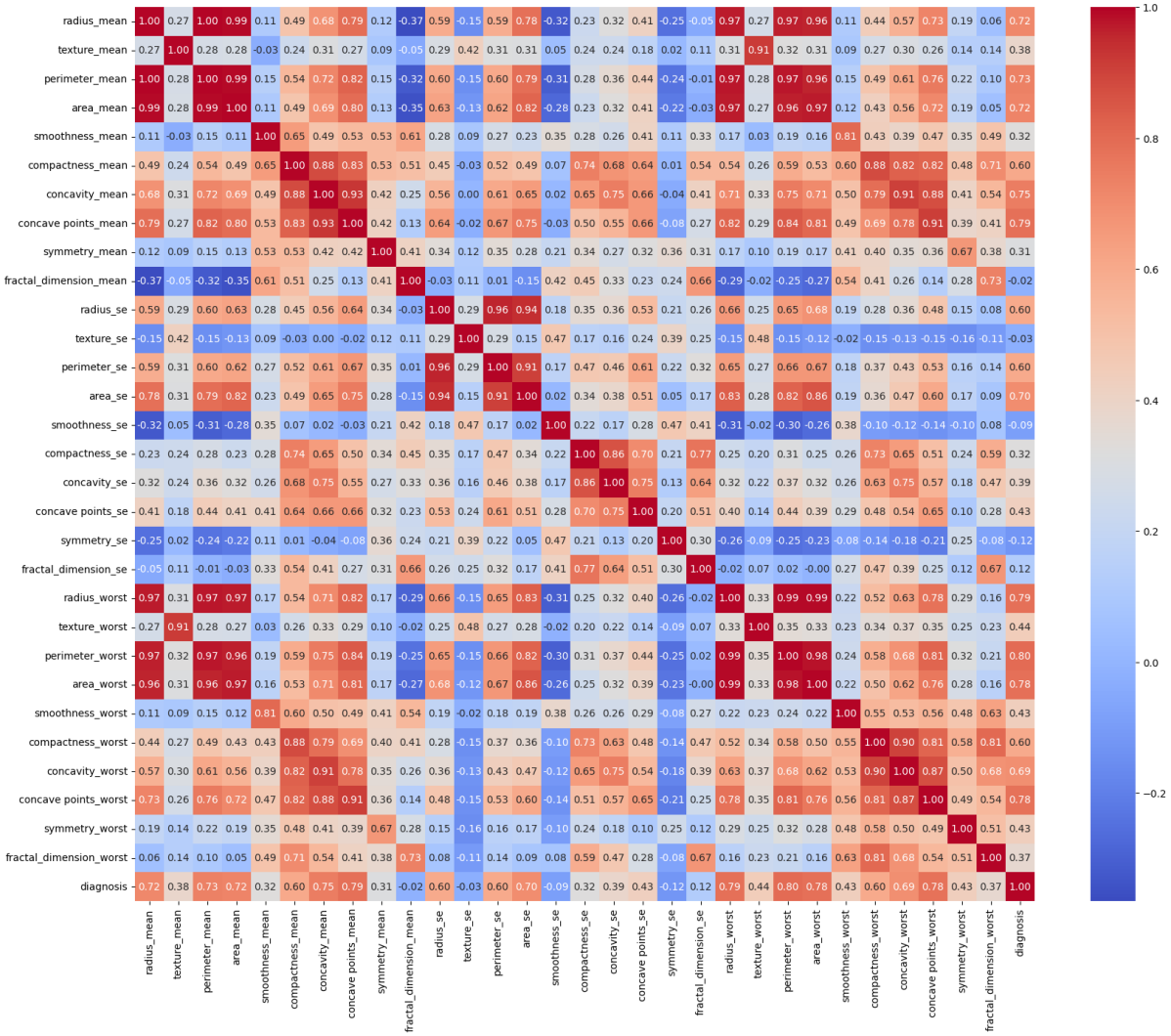


해당 그림을 통해 상관관계가 높은 변수들을 파악할 수 있다. 데이터가 골고루 퍼져 있으면 상관관계가 낮고, 대각선에 집중되어 있으면 상관계수가 높다고 판단할 수 있다.

Scatter plot을 보았을 때, 높은 양의 상관관계를 보이는 변수 조합이 많은 것을 확인할 수 있는데, 이는 [Q2]에서 설명한 것과 같이, 종양의 특징 및 수치들이 독립적이기보다 복합적으로 연결되어 있기 때문에, 전체적으로 상관관계가 높은 변수 조합이 많다고 해석할 수 있다.

상관관계를 보다 정확하게 판단하기 위해 상관계수를 나타내는 Heatmap 도 시각화 해보았고, 그 결과는 아래와 같다.

▶ Heatmap



상관계수는 -1부터 1까지의 값을 가지며, 0에서 멀리 떨어질수록 강한 선형관계를 가진다고 말할 수 있다. 이때, 상관계수가 양수인 경우 양의 상관관계, 음수인 경우 음의 상관관계를 보인다고 할 수 있다. 여기서도 상관계수가 1에 가까운 수치가 많은 것을 보아, 많은 변수 조합이 강한 양의 상관관계를 보이는 것을 확인할 수 있고, 반면 상관계수가 -0.4 이하인 값이 없는 것을 보아 강한 음의 상관관계를 보이는 변수 조합은 없는 것을 알 수 있다.

5-1. 어떤 두 조합의 변수들이 서로 강한 상관관계가 있다고 할 수 있는가?

일반적으로 두 변수의 상관관계를 확인할 때, 상관계수의 절댓값이 0.7 이상인 경우, 두 변수는 강한 상관관계를 가진다고 판단한다. 해당 기준을 가지고 상관관계 분석을 진행하였다.

설명 변수는 크게 radius, texture, perimeter, area, smoothness, compactness, concavity, concave points, symmetry, fractal_dimension 이렇게 10개의 feature로 나눌 수 있고, 각각의 feature마다 평균, 표준오차, 최댓값을 따로 나타냈기 때문에 총 30개의 변수가 나타난다. 이때 각 feature의 mean과 worst의 상관계수를 보면 radius : 0.97, texture : 0.91, perimeter : 0.97, area : 0.97,

smoothness : 0.81, compactness : 0.88, concavity : 0.91, concave points : 0.91, symmetry : 0.67, fractal_dimension : 0.73으로 대부분 강한 상관관계를 가지고 있는 것을 확인할 수 있다. symmetry의 경우 0.67으로, 0.70보다는 작은 값을 보이지만 0.7과 거의 근접한 수치를 보여주기 때문에, 이 정도의 수치 차이는 감수하고 강한 상관관계가 있다고 말할 수 있다.

일반적으로 평균이 높다면, 최댓값도 높을 가능성이 크기 때문에 납득할 수 있는 결과라고 할 수 있다. 가능한 변수의 조합이 너무 많아 아래의 표로 상관관계 분석 결과를 표로 정리하였다.

상관관계	변수
강한 (양의) 상관관계 (상관계수 절댓값 0.7 이상)	radius_mean, perimeter_mean radius_mean, area_mean radius_mean, cancave points_mean radius_mean, area_se radius_mean, radius_worst radius_mean, perimeter_worst radius_mean, area_worst radius_mean, cancave points_worst radius_mean, diagnosis texture_mean, texture_worst perimeter_mean, area_mean perimeter_mean, concavity_mean perimeter_mean, concave points_mean perimeter_mean, area_se perimeter_mean, radius_worst perimeter_mean, perimeter_worst perimeter_mean, area_worst perimeter_mean, concave points_worst perimeter_mean, diagnosis area_mean, concave points_mean area_mean, area_se area_mean, radius_worst area_mean, perimeter_worst area_mean, area_worst area_mean, concave points_worst area_mean, diagnosis smoothness_mean, smoothness_worst compactness_mean, concavity_mean compactness_mean, concave points_mean compactness_mean, compactness_se compactness_mean, compactness_worst

	compactness_mean, concavity_worst compactness_mean, concave points_worst compactness_mean, fractal_dimension_worst concavity_mean, concave points_mean concavity_mean, concavity_se concavity_mean, radius_worst concavity_mean, perimeter_worst concavity_mean, area_worst concavity_mean, compactness_worst concavity_mean, concavity_worst concavity_mean, concave points_worst concavity_mean, diagnosis concave points_mean, area_se concave points_mean, radius_worst concave points_mean, perimeter_worst concave points_mean, area_worst concave points_mean, concavity_worst concave points_mean, concave points_worst concave points_mean, diagnosis fractal_dimensinal_mean, fractal_dimensinal_worst radius_se, perimeter_se radius_se, area_se perimeter_se, area_se area_se, radius_worst area_se, perimeter_worst area_se, area_worst area_se, diagnosis compactness_se, concavity_se compactness_se, concave points_se compactness_se, fractal_dimension_se compactness_se, compactness_worst concavity_se, concave_points_se concavity_se, concavity_worst radius_worst, perimeter_worst radius_worst, area_worst radius_worst, concave points_worst radius_worst, diagnosis perimeter_worst, area_worst perimeter_worst, concave points_worst

	perimeter_worst, diagnosis area_worst, concave points_worst area_worst, diagnosis compactness_worst, concavity_worst compactness_worst, concave points_worst compactness_worst, fractal_dimension_worst concavity_worst, concave points_worst concave points_worst, diagnosis

변수의 조합이 많으므로, 크게 10개의 feature (radius, texture, perimeter, area, smoothness, compactness, concavity, concave points, symmetry, fractal_dimension) 간의 상관관계를 확인해보면, (radius, perimeter, area, concavity, concave points)의 mean, worst 변수가 서로 강한 양의 상관관계를 가지는 것을 확인할 수 있다. 이는 보통 radius가 크면 perimeter와 area도 같이 커지고, 크기가 큰 종양이 상대적으로 오목함의 정도나 개수가 많다는 것으로 이해할 수 있다.

또한, (compactness, concavity, concave points)와 관련된 변수들도 서로 강한 양의 상관관계를 가지는 것을 확인할 수 있는데 compactness (perimeter^2 / area - 1.0)값이 크다는 것은 area 대비 perimeter값이 크다는 것이고, 종양이 올록볼록하게 생겼다는 것으로 해석할 수 있다. 따라서, compactness, concavity, concave points의 상관관계가 강한 것을 이해할 수 있다.

이 외에도 강한 양의 상관관계를 가지는 조합들이 존재하는데, 이는 설명변수들의 특징들이 독립적이기보다 복합적으로 연결되어 있기 때문에, 전체적으로 상관관계가 높은 조합이 많다고 판단할 수 있다. 반면, 강한 음의 상관관계를 가지는 조합은 없는 것을 알 수 있다.

5-2. 강한 상관관계가 존재하는 변수 조합들 중에 대표 변수를 하나씩만 선택해서 전체 변수의 개수를 감소시켜 보시오.

회귀 분석에서는 변수 간의 상관계수가 높을 경우, 다중공산성 문제가 발생할 수 있다. 이는 회귀계수의 추정을 어렵게 만들며, 통계적으로 유의미한 결과를 도출하는 데 방해가 될 수 있다. 따라서 이를 해결하기 위해 [Q5-1]에서 도출한 변수 간의 상관관계를 토대로, 상관관계가 높은 변수들 중 대표 변수를 선택하여 변수의 개수를 줄이려고 한다.

우선, radius, texture, perimeter, area, smoothness, compactness, concavity, concave points, symmetry, fractal_dimension 이렇게 10개의 feature를 보았을 때, 각 feature의 mean값과 worst값이 강한 상관관계를 보이는 것을 확인할 수 있다. 따라서, 각 feature의 mean과 worst 중에서 diagnosis와 조금 더 강한 상관관계를 보이는 worst를 대표로 선택하고자 한다.

mean에 해당하는 10개의 변수를 제거한 나머지 20개의 변수의 상관관계를 확인하고, 강한 상관관계를 보이는 변수들 중에서 종속 변수에 대한 상관계수가 가장 높은 변수를 대표로 사용하고자 한다. 그 결과는 아래와 같다.

(compactness_se, concavity_se, concave points_se)가 서로 강한 상관관계를 보이고 있고, 따라서 그 중 종속변수와 가장 상관관계가 강한 'concave points_se'를 대표로 사용하고자 한다.

(radius_se, perimeter_se, area_se)가 서로 강한 상관관계를 보이기 때문에 'area_se'를 대표로 선택하려고 했지만, 이때 문제가 하나 발생한다. area_se는 (radius_worst, perimeter_worst, area_worst)와도 강한 상관관계를 가지고 있어, 여기서 또 대표를 뽑게 된다면 perimeter_worst가 뽑히게 된다. 그러나 radius_se, perimeter_se는 perimeter_worst와 강한 상관관계를 가지고 있지 않다. 즉, (radius_se, perimeter_se, area_se)에서 대표로 뽑힌 'area_se'가 'perimeter_worst'에 의해 제거가 되면, radius_se, perimeter_se에 대한 정보가 많이 사라질 수 있다고 생각하였기 때문에 (radius_se, perimeter_se, area_se)의 대표로 'area_se' 대신 'radius_se'를 선택하였다.

(compactness_worst, concavity_worst, concave points_worst) 조합들도 서로 강한 상관관계를 보이기 때문에, 이 중 대표로 'concave points_worst'를 사용하려고 했지만, 여기서도 위와 같은 문제가 발생한다. 'concave points_worst'는 (radius worst, perimeter_worst, area_worst)와도 서로 강한 상관관계를 가지고 있기 때문에, 여기서 또 대표를 뽑게 된다면 'perimeter_worst'가 뽑히게 된다. 그러나, 해당 변수는 compactness_worst, concavity_worst와 강한 상관관계를 가지지 않기 때문에 compactness_worst, concavity_worst의 중요한 정보들이 사라질 수 있다고 판단하였다. 따라서 (compactness_worst, concavity_worst, concave points_worst) 중에서 'concave points_worst' 대신 'concavity_worst'를 대표로 선택하였다.

(radius worst, perimeter_worst, area_worst)의 변수 조합은 상관계수가 1에 가까울 정도로 높은 상관관계를 보이고, 종속변수와의 상관계수도 거의 차이가 없기 때문에, 최대한 다중공산성 문제를 피하고자 그나마 다른 변수들과의 상관계수가 작은 'radius_worst'를 대표로 선택하였다.

결론적으로 전체 30개의 변수 중에서 12개의 변수('radius_se', 'texture_se', 'smoothness_se', 'concave points_se', 'symmetry_se', fractal_dimension_se', 'radius_worst', 'texture_worst', 'smoothness_worst', 'concavity_worst', 'symmetry_worst', 'fractal_dimension_worst')로 변수의 개수를 감소시킬 수 있다.

[Q6] 전체 데이터셋을 70%의 학습 데이터와 30%의 테스트 데이터로 분할한 후 모든 변수를 사용하여 Logistic Regression 모델을 학습해 보시오. 이 때 70:30으로 구분하는 random seed를 저장하시오.

```
from sklearn.model_selection import train_test_split
seed = 12345
test_size = 0.3
cancer_trn_data, cancer_valid_data = train_test_split(new_cancer_data,test_size=test_size,random_state=seed)

# input, output variables in numpy array
x_trn, y_trn = cancer_trn_data.iloc[:,:-1],cancer_trn_data.iloc[:,-1]
x_tst, y_tst = cancer_valid_data.iloc[:,:-1],cancer_valid_data.iloc[:,-1]
```

데이터셋 분할 시 random seed는 12345로 하였다. 학습 데이터셋을 이용하여 logistic Regression 모델을 학습시키고 p-value를 확인한 결과는 아래와 같다.

	P-value
constant	0.9877
radius_mean	0.8473
texture_mean	0.9836
perimeter_mean	0.9317
area_mean	0.7649
smoothness_mean	0.9987
compactness_mean	0.9955
concavity_mean	0.9899
concave points_mean	0.9970
symmetry_mean	0.9949
fractal_dimension_mean	0.9999
radius_se	0.9990
texture_se	0.8063
perimeter_se	0.8089
area_se	0.8161
smoothness_se	1.0000
compactness_se	0.9999
concavity_se	0.9995
concave points_se	0.9999
symmetry_se	0.9998
fractal_dimension_se	1.0000
radius_worst	0.6068
texture_worst	0.4422
perimeter_worst	0.6264
area_worst	0.2427
smoothness_worst	0.9970
compactness_worst	0.9749
concavity_worst	0.9324
concave points_worst	0.9889
symmetry_worst	0.9828
fractal_dimension_worst	0.9994

6-1. 유의수준 0.05에서 유효한 변수의 수는 몇 개인지 확인하고 각 변수들이 본인의 상식선에서 실제로 유효하다고 할 수 있는지 판단해 보시오.

유의수준 0.05에서 p-value 값이 0.05보다 작은 변수들이 통계적으로 유효하다고 판단하는데, 결과를 확인해보면 현재 모델에서 유효한 변수는 존재하지 않는다. 이는 이전에 수행한 상관관계 분석 결과와 연관이 있을 수 있다. 변수들 간에 높은 상관관계가 많이 관찰되었는데, 이는 변수들이 서로 독립적이지 않다는 것을 의미하고 이로 인해 다중공산성 문제가 발생했을 가능성이 있다. 이러한 문제로 통계적으로 유의미한 결과를 도출하지 못했다고 해석할 수 있다.

따라서 p-value 해석 결과, 해당 모델에서 유효한 변수는 0개라고 나오지만, 정확한 분석을 위해서 다중공산성 문제를 해결한 후에 p-value를 다시 확인해볼 필요가 있다.

6-2. [Q2-2]에서 정성적으로 선택했던 변수들의 P-value를 확인하고 해당 변수가 모델링 측면에서 실제로 유효하지 않는 것인지 확인해 보시오.

[Q2-2]에서 정성적으로 선택했던 변수들은 symmetry_mean, symmetry_se, symmetry_worst, fractal_dimesion_mean, fractal_dimesion_se, fractal_dimesion_worst 이렇게 6개의 변수이다. 이들의 p-value를 살펴보면 모두 유효하지 않은 변수임을 확인할 수 있다. 하지만, 위에서 말한 것과 같이 변수들 간 상관관계가 높고 독립적이지 않아 다중공산성 문제가 발생했을 가능성이 크다. 따라서 통계적으로 유의미한 결과를 도출했다고 보기 어렵고, 위의 변수들이 모델링 측면에 있어 유효하지 않는지 제대로 판단하기 위해서는 다중공산성 문제를 해결한 후에 p-value를 다시 확인해볼 필요가 있다.

6-3. 학습 데이터와 테스트 데이터에 대한 Confusion Matrix를 생성하고 Simple Accuracy, Balanced Correction Rate, F1-Measure를 산출하여 비교해 보시오.

- 학습 데이터에 대한 Confusion Matrix -

Confusion Matrix		Predicted	
		Benign	Malignant
Actual	Benign	228	7
	Malignant	9	101

- 테스트 데이터에 대한 Confusion Matrix –

Confusion Matrix		Predicted	
		Benign	Malignant
Actual	Benign	101	2
	Malignant	2	44

- 학습/테스트 데이터 Simple Accuracy, Balanced Correction Rate, F1-Measure –

	Simple Accuracy	BCR	F1-Measure
Train	0.9536	0.9438	0.9266
Test	0.9732	0.9685	0.9565

학습 데이터셋의 평가 결과와 테스트 데이터셋의 평과 결과를 확인했을 때, ACC, BCR, F1-Measure 모두 0.9가 넘는 높은 수치를 보이는 것을 보아 전체적으로 Benign, Malignant 분류를 잘 해준다는 것을 알 수 있다. 이때, 모든 지표(ACC, BCR, F1-Measure)의 수치가 학습 데이터보다 테스트 데이터에서 더 높은 것을 보아 테스트 데이터셋에서의 성능이 더 뛰어난 것을 알 수 있다. 또한, 테스트 데이터셋에서 뛰어난 성능을 보이는 것을 확인했을 때, 학습 과정에서 과적합이 발생하지 않았다고 해석할 수 있고, 현재 모델이 분류 작업을 잘 수행하고 있다고 판단할 수 있다.

Accuracy를 보면 train셋, test셋에서 각각 95%, 97% 정도로 높은 수치를 보이고 있는데, BCR, F1은 Accuracy보다 낮은 수치를 보인다. 이는 Benign 탐지에 비해 Malignant 탐지 비율이 낮기 때문이라고 할 수 있다. 종합적으로 보았을 때, 모든 평가지표의 수치가 0.9 이상으로 높기 때문에 해당 모델의 분류 성능이 뛰어난 편이라고 생각한다.

6-4. 학습 데이터와 테스트 데이터에 대한 AUROC를 산출하는 함수를 직접 작성하고 이를 사 용하여 학습/테스트 데이터셋에 대한 AUROC를 비교해 보시오.

AUROC를 산출 코드를 아래와 같이 직접 작성하였다.

```python
def calculate_auroc(y_true, y_scores):
    y_true = np.array(y_true)

    # 예측 확률을 기준으로 내림차순 정렬
    sorted_indices = np.argsort(y_scores)[::-1]
    y_true = y_true[sorted_indices]

    # 누적 합계 계산 (누적합이 배열로 나타남)
    cum_positive = np.cumsum(y_true == 1)  # 실제 악성 샘플의 누적 합계
    cum_negative = np.cumsum(y_true == 0)  # 실제 양성 샘플의 누적 합계

    # TPR과 FPR 계산
    tpr = cum_positive / np.sum(y_true == 1)
    fpr = cum_negative / np.sum(y_true == 0)

    # AUROC 계산
    auroc = np.trapz(tpr, fpr)
    return [round(auroc,4)]
```

이를 사용하여 학습/데이터셋에 대한 AUROC를 구한 값은 아래와 같다.

	auroc
Train	0.9906
Test	0.9932

6-3에서 분류 성능이 뛰어난 것을 확인할 수 있었는데, AUROC 점수도 동일하게 학습 데이터셋, 테스트 데이터셋 모두 매우 높게 나타난 것을 확인할 수 있고, 따라서 해당 모델이 overfitting 되지 않았으며 높은 분류 성능을 가지고 있음을 알 수 있다. 또한, 위에서 Simple Accuracy, Balanced Correction Rate, F1-Measure을 보았을 때, 테스트셋에서의 수치가 더 높은 것을 확인하였는데, AUROC에서도 동일하게 테스트셋에서 더 성능이 좋은 것을 확인할 수 있다.

[Q7] [Q5]에서 변수 간 상관관계를 기준으로 선택한 변수들만을 사용하여 [Q6]에서 사용한 학습/테스트 70:30 분할 데이터로 Logistic Regression 모델을 학습해 보시오

[Q5]에서 변수 간 상관관계를 기준으로 선택한 변수는 총 12개로, 변수의 목록은 아래와 같다.

('radius_se', 'texture_se', 'smoothness_se', 'concave points_se', 'symmetry_se', fractal_dimension_se', 'radius_worst', 'texture_worst', 'smoothness_worst', 'concavity_worst', 'symmetry_worst', 'fractal_dimension_worst')

```
new_input_idx = [10, 11, 14, 17, 18, 19, 20, 21, 24, 26, 28, 29]

# input, output variables in numpy array
x_trn, y_trn = cancer_trn_data.iloc[:,:-1],cancer_trn_data.iloc[:,-1]
x_tst, y_tst = cancer_valid_data.iloc[:,:-1],cancer_valid_data.iloc[:,-1]
new_x_trn = x_trn.iloc[:,new_input_idx]
new_x_tst = x_tst.iloc[:,new_input_idx]
```

위의 코드와 같이 사용할 변수의 인덱스를 지정하여 학습/테스트 데이터를 만들었고, 나머지는 모두 동일하게 학습을 진행하였다.

7-1. 유의수준 0.05에서 유효한 변수의 수는 몇 개인지 확인하고 [Q6-1]의 결과와 비교하시오.

p-value를 확인한 결과는 아래와 같다.

	P-value
constant	0.0023
radius_se	0.2672
texture_se	0.0865
smoothness_se	0.9999
concave points_se	0.9997
symmetry_se	0.9985
fractal_dimension_se	1.0000
radius_worst	0.0107
texture_worst	0.1570
smoothness_worst	0.9902
concavity_worst	0.1043
symmetry_worst	0.9316
fractal_dimension_worst	0.9936

유의수준 0.05에서 유효한 변수는 'radius_worst' 1개이다. [Q6-1]의 결과에서는 유효한 변수가 0개였으며, 그때 'radius_worst'의 p-value는 0.6068이었다. 즉, 변수 제거가 이루어진 후 'radius_worst'가 종속변수인 'diagnosis'에 미치는 영향이 커졌다고 해석할 수 있다.

그 원인으로 'radius_worst'와 상관관계가 높았던 변수들이 제거됨으로써 다중공산성 문제가 어느 정도 해결이 되었고, 따라서 'radius_worst'의 유의성이 증가했다고 해석이 가능하다. 또한, 악성 종양이 양성 종양에 비해, 'radius_worst'가 크기 때문에 해당 변수는 종속변수 'diagnosis'와 선형성이 있다고 할 수 있고, 따라서 해당 변수가 유효하다는 점은 상식 선에서 이해할 수 있다.

7-2. 학습 데이터와 테스트 데이터에 대한 Confusion Matrix를 생성하고 Simple Accuracy, Balanced Correction Rate, F1-Meausre를 산출한 뒤, [Q6-3]의 결과와 비교해 보시오.

- 학습 데이터에 대한 Confusion Matrix -

Confusion Matrix		Predicted	
		Benign	Malignant
Actual	Benign	233	2
	Malignant	14	96

- 테스트 데이터에 대한 Confusion Matrix –

Confusion Matrix		Predicted	
		Benign	Malignant
Actual	Benign	101	2
	Malignant	5	41

- 학습/테스트 데이터 Simple Accuracy, Balanced Correction Rate, F1-Measure –

	Simple Accuracy	BCR	F1-Measure
Train	0.9536	0.9302	0.9231
Test	0.9530	0.9349	0.9214

[Q6-3]과 Confusion Matrix를 비교해보았을 때, 학습/테스트 데이터 모두 실제 Malignant인 것을 맞추는 것은 줄어들었지만, 실제 Benign인 것을 맞추는 것은 학습 데이터에서는 더 좋아졌고, 테스트 데이터에서는 그대로인 것을 알 수 있다. 즉, 변수를 선택하여 새롭게 학습한 모델은 'Malignant'보다 'Benign'에 더 focus되어 있는 모델일 수 있다.

		Simple Accuracy	BCR	F1-Measure
[Q6-3]	Train	0.9536	0.9438	0.9266
	Test	0.9732	0.9685	0.9565
[Q7-2]	Train	0.9536	0.9302	0.9231
	Test	0.9530	0.9349	0.9214

[Q6-3]과 [Q7-2]의 Simple Accuracy, Balanced Correction Rate, F1-Measure를 비교해보면, 학습 데이터에서의 Simple Accuracy 값만 동일하고, 나머지 모든 부분에서 수치가 감소한 것을 확인할 수 있다. 즉, 변수를 삭제하기 이전이 더 높은 성능을 보이는 것을 알 수 있지만, 변수 삭제 이후에도 모든 수치가 0.9 이상인 점을 고려하였을 때, 성능 감소 폭이 크지 않다고 할 수 있다. 이때, 증가한 'Benign' 탐지 비율보다, 감소한 'Malignant' 탐지 비율이 더 크기 때문에, BCR, F1-Measure 값이 감소한 것을 확인할 수 있다.

7-3. 학습/테스트 데이터셋에 대한 AUROC를 산출하여 [Q6-4]의 결과와 비교해 보시오.

	auroc
Train	0.9906
Test	0.9932

	auroc
Train	0.9821
Test	0.9932

좌측의 표가 변수 제거 전 AUROC, 우측의 표가 변수 제거 후 AUROC이다. AUROC를 비교해보았을 때, 학습 데이터셋에서는 AUROC값이 감소하였지만, 테스트 데이터셋에서는 ACC, BCR, F1-Measure 모두 감소하였음에도 AUROC값은 차이가 없는 것을 확인할 수 있다. 이는 ACC, BCR, F1-Measure과 다르게 AUROC는 cut-off의 영향을 받지 않기 때문에 가능한 결과라고 할 수 있다. 또한, 변수 제거 전, 후 모두 학습 데이터셋보다 테스트 데이터셋에서의 AUROC값이 더 1에 가까우므로, 두 모델 모두 테스트 데이터셋에서의 성능이 더 좋다고 할 수 있다.

[Q8] [Q6]에서 생성한 학습 데이터를 이용하여 Logistic Regression 에 Forward Selection, Backward Elimination, Stepwise Selection 을 적용해보시오. 각 방법론마다 Training dataset 에 대한 AUROC 및 소요 시간, Validation dataset 에 대한 AUROC, Accuracy, BCR, F1- Measure 를 산출하시오.

[Q6]에서 생성한 학습 데이터에 Forward Selection, Backward Elimination, Stepwise Selection을 차례로 수행해 보았고 그 결과는 아래와 같다. 이때, logistic Regression의 main arguments 중 하나인 solver는 모두 'saga'를 사용하였다.

8-1. Forward Selection

1) Forward Selection 방법을 통해 선택된 변수는 다음과 같다. (6개)

['texture_mean' 'concavity_mean' 'radius_worst' 'compactness_worst' 'concavity_worst' 'concave points_worst']

2) Training dataset에 대한 AUROC 및 소요 시간은 다음과 같다. (시간의 단위는 seconds)

```
AUROC : [0.988]
소요 시간 : 59.81606340408325
```

3) Validation dataset에 대한 AUROC, Accuracy, BCR, F1- Measure 등 평가 결과는 다음과 같다.

	TPR(Recall)	Precision	TNR	ACC	BCR	F1	AUROC
Forward Selection	0.9348	0.9773	0.9903	0.9732	0.9621	0.9556	0.9954

8-2. Backward Elimination

1) Backward Elimination 방법을 통해 선택된 변수는 다음과 같다. (29개)

['radius_mean' 'texture_mean' 'perimeter_mean' 'area_mean' 'smoothness_mean' 'compactness_mean' 'concavity_mean' 'concave points_mean' 'symmetry_mean' 'fractal_dimension_mean' 'radius_se' 'texture_se' 'perimeter_se' 'smoothness_se' 'compactness_se' 'concavity_se' 'concave points_se' 'symmetry_se' 'fractal_dimension_se' 'radius_worst' 'texture_worst' 'perimeter_worst' 'area_worst' 'smoothness_worst' 'compactness_worst' 'concavity_worst' 'concave points_worst' 'symmetry_worst' 'fractal_dimension_worst']

2) Training dataset에 대한 AUROC 및 소요 시간은 다음과 같다. (시간의 단위는 seconds)

```
AUROC : [0.9721]
소요 시간 : 57.916446685791016
```

3) Validation dataset에 대한 AUROC, Accuracy, BCR, F1- Measure 등 평가 결과는 다음과 같다.

	TPR(Recall)	Precision	TNR	ACC	BCR	F1	AUROC
Forward Selection	0.9348	0.9773	0.9903	0.9732	0.9621	0.9556	0.9954
Backward Elimination	0.8696	0.9524	0.9806	0.9463	0.9234	0.9091	0.9591

8-3. Stepwise Selection

1) Stepwise Selection 방법을 통해 선택된 변수는 다음과 같다. (15개)

['texture_mean' 'smoothness_mean' 'compactness_mean' 'concavity_mean' 'concave points_mean' 'fractal_dimension_mean' 'smoothness_se' 'compactness_se' 'concavity_se' 'concave points_se' 'symmetry_se' 'radius_worst' 'compactness_worst' 'concavity_worst' 'concave points_worst']

2) Training dataset에 대한 AUROC 및 소요 시간은 다음과 같다. (시간의 단위는 seconds)

```
AUROC : [0.9883]
소요 시간 : 127.41714882850647
```

3) Validation dataset에 대한 AUROC, Accuracy, BCR, F1- Measure 등 평가 결과는 다음과 같다.

	TPR(Recall)	Precision	TNR	ACC	BCR	F1	AUROC
Forward Selection	0.9348	0.9773	0.9903	0.9732	0.9621	0.9556	0.9954
Backward Elimination	0.8696	0.9524	0.9806	0.9463	0.9234	0.9091	0.9591
Stepwise Selection	0.9348	0.9773	0.9903	0.9732	0.9621	0.9556	0.9956

8-4. 3가지 변수 선택 기법 성능 비교

변수 선택 부분에서 전체 30개의 변수 중, Forward Selection은 6개의 변수, Backward Elimination은 29개의 변수, Stepwise Selection은 15개의 변수를 선택하였다. 즉, 변수감소율은 다음과 같이 나타낼 수 있다. Forward Selection > Stepwise Selection > Backward Selection

시간적인 부분에서 Forward Selection과 Backward Elimination은 비슷하게 약 1분 소요된 반면 Stepwise Selection은 그보다 2배 더 긴 2분이 소요되었다. 즉, 소요 시간은 다음과 같이 나타낼 수 있다. Forward Selection = Backward Selection < Stepwise Selection

성능적인 부분에서, Stepwise Selection와 Forward Selection을 비교했을 때, 학습 데이터셋에서의 AUROC, 테스트 데이터셋에서의 AUROC 모두 거의 차이가 없고, ACC, BCR, F1 지표에서는 모두 동일한 값을 가지기 때문에, 둘의 성능적인 차이는 없다고 할 수 있다. 이때, Stepwise Selection과 Forward Selection은 AUROC, ACC, BCR, F1 모두 0.95가 넘는 높은 수치를 보이기 때문에, 둘은 매우 뛰어난 성능을 보인다고 할 수 있다. 반면, Backward Elimination은 Stepwise Selection, Forward Selection에 비해 AUROC, F1, BCR은 약 4%, ACC는 약 3% 낮은 수치를 보이는 것을 알 수 있고, 성능적으로 둘에 비해 부족한 것을 알 수 있다. 즉, 성능은 다음과 같은 순서로 나타낼 수 있다. Stepwise Selection = Forward Selection > Backward Elimination

종합적으로 보면, Forward Selection과 Stepwise Selection이 가장 뛰어난 성능을 보이는데, 둘 중 Forward Selection이 Stepwise Selection보다 변수 감소율이 높고, 소요 시간도 적기 때문에 해당 데이터셋에서는 Forward Selection 기법을 적용하는 것이 적합하다고 할 수 있다.

[Q9] AUROC를 Fitness function으로 하는 Genetic Algorithm 기반의 변수 선택 함수를 작성해 보시오. 작성한 함수를 이용하여 GA 기반 변수 선택을 수행하고, 선택된 변수를 사용한 Logistic Regression의 Validation dataset에 대한 분류 성능(AUROC, Accuracy, BCR, F1-Measure), 변수 감소율, 수행 시간의 세 가지 관점에서 Forward Selection, Backward Elimination, Stepwise Selection 방식을 사용한 Logistic Regression과 비교해 보시오.

[Q8]에서 사용한 데이터로 Genetic Algorithm을 수행한다. 이때, 염색체의 우열을 가릴 수 있는 정량적 지표로 AUROC를 사용한다.

1) Genetic Algorithm을 통해 선택된 변수는 다음과 같다. (20개)

['radius_mean', 'texture_mean', 'perimeter_mean', 'area_mean', 'smoothness_mean', 'symmetry_mean', 'fractal_dimension_mean', 'radius_se', 'perimeter_se', 'area_se', 'compactness_se', 'concave points_se', 'radius_worst', 'texture_worst', 'perimeter_worst', 'area_worst', 'smoothness_worst', 'compactness_worst', 'concavity_worst', 'fractal_dimension_worst']

2) Training dataset에 대한 AUROC 및 소요 시간은 다음과 같다. (시간의 단위는 seconds)

```
AUROC : [0.9719]
소요 시간 : 18.575902462005615
```

3) Validation dataset에 대한 AUROC, Accuracy, BCR, F1- Measure 등 평가 결과는 다음과 같다.

	TPR(Recall)	Precision	TNR	ACC	BCR	F1	AUROC
Forward Selection	0.9348	0.9773	0.9903	0.9732	0.9621	0.9556	0.9954
Backward Elimination	0.8696	0.9524	0.9806	0.9463	0.9234	0.9091	0.9591
Stepwise Selection	0.9348	0.9773	0.9903	0.9732	0.9621	0.9556	0.9956
Genetic Algorithm	0.8696	0.9524	0.9806	0.9463	0.9234	0.9091	0.9591

4) Forward Selection, Backward Elimination, Stepwise Selection, Genetic Algorithm으로 변수 선택을 하고, 선택된 변수를 사용한 Logistic Regression의 Validation dataset에 대한 평가까지 진행한 것을 정리한 표는 아래와 같다.

	변수감소율	수행시간	AUROC	Accuracy	BCR	F1
Forward Selection	80.0%	59.816 sec	0.9954	0.9732	0.9621	0.9556
Backward Elimination	3.3%	57.916 sec	0.9591	0.9463	0.9234	0.9091
Stepwise Selection	50.0%	127.417 sec	0.9956	0.9732	0.9621	0.9556
Genetic Algorithm	33.3%	18.576 sec	0.9591	0.9463	0.9234	0.9091

Genetic Algorithm은 수행 시간 부분에서 가장 적은 시간을 소요하였고, 변수감소율 부분에서는 Backward Elimination보단 높지만, Forward Selection과 Stepwise보단 낮은 수치를 보이고 있다. AUROC, Accuracy, BCR, F1은 Backward Elimination과 정확하게 동일한 것을 확인할 수 있다. 즉, Genetic Algorithm의 분류 성능은 Backward Elimination과 차이가 없고, Forward Selection, Stepwise Selection보다 낮은 성능을 보이는 것을 확인할 수 있다.

5) 종합 평가

마지막으로 종합 평가를 진행하면 아래와 같다.

변수 감소율 : Forward Selection > Stepwise Selection > Genetic Algorithm > Backward Elimination

수행 시간 : Genetic Algorithm < Backward Elimination = Forward Selection < Stepwise Selection

성능 지표 : Stepwise Selection = Forward Selection > Genetic Algorithm = Backward Elimination

지금까지의 결과를 보면, Forward Selection이 Stepwise Selection과 함께 가장 뛰어난 성능을 보이고, 변수감소율도 가장 높으며, Genetic Algorithm 다음으로 빠른 수행시간을 가지고 있어, Forward Selection이 현재 데이터셋에서 가장 적합한 변수 선택 기법이라고 판단할 수 있다.

그러나 현재 Genetic Algorithm의 성능이 다른 변수 선택 기법에 비해 높지는 않지만, Genetic Algorithm은 다양한 하이퍼파라미터가 있어, 파라미터의 조합에 따라 성능에 큰 차이가 있을 수 있다. 따라서 Genetic Algorithm에서 여러 파리마터의 조합을 탐색하는 과정을 거치고 변수 선택을 진행한다면 결과가 달라질 수 있음에 유의해야 한다.

[Q10] Genetic Algorithm에서 변경 가능한 하이퍼파라미터들(population size, Cross-over rate, Mutation rate 등) 중 세 가지를 선택하고 각각의 하이퍼파라미터마다 최소 세 가지 이상의 후보 값들을 선정(최소 27가지 이상의 조합)하여 각 조합에 대한 변수 선택 결과에 대해 본인만의 생각을 더해 해석해보시오.

Genetic Algorithm에는 여러 하이퍼파라미터들이 존재한다. population_size, n_gen, mutation_rate, crossover_rate, n_parents, init_rate 등 여러 파라미터들이 있는데, 이번 과제에서는 population_size, crossover_rate, mutation_rate를 조정하면서 여러 성능 지표들이 어떻게 변화하는지 살펴보려고 한다.

이때, population_size는 한 세대에 포함되는 개체의 수를 나타내고, crossover_rate는 교차 연산이 일어날 확률을 나타내며, mutation_rate는 돌연변이가 일어날 확률을 나타낸다.

```
# 하이퍼파라미터 후보 값들
population_sizes = [10, 20, 30]
crossover_rates = [0.1, 0.3, 0.5]
mutation_rates = [0.05, 0.1, 0.2]
```

위와 같이 population_size에서는 [10, 20, 30]을 후보 값으로 설정하였고, crossover_rate에서는 [0.1, 0.3, 0.5]를 후보 값으로, mutation_rate에서는 [0.05, 0.1, 0.2]를 후보 값으로 설정하였다. 따라서 총 27가지의 조합에 대한 Genetic Algorithm 결과를 확인할 수 있다.

평가 지표로는 기존과 동일하게 Validation dataset에 대한 분류 성능(AUROC, Accuracy, BCR, F1-Measure)을 사용하고자 한다.

아래는 각 파라미터의 조합에 따른 분류 성능을 나타낸 표와 선택한 변수의 개수 및 변수감소율을 나타낸 것이다.

	population_size	crossover_rate	mutation_rate	TPR(Recall)	Precision	TNR	ACC	BCR	F1	AUROC
combi_1	10.0	0.1	0.05	0.8696	0.9524	0.9806	0.9463	0.9234	0.9091	0.9591
combi_2	10.0	0.1	0.10	0.8696	0.9524	0.9806	0.9463	0.9234	0.9091	0.9591
combi_3	10.0	0.1	0.20	0.8696	0.9524	0.9806	0.9463	0.9234	0.9091	0.9588
combi_4	10.0	0.3	0.05	0.8696	0.9524	0.9806	0.9463	0.9234	0.9091	0.9591
combi_5	10.0	0.3	0.10	0.8696	0.9524	0.9806	0.9463	0.9234	0.9091	0.9591
combi_6	10.0	0.3	0.20	0.8696	0.9524	0.9806	0.9463	0.9234	0.9091	0.9588
combi_7	10.0	0.5	0.05	0.8696	0.9524	0.9806	0.9463	0.9234	0.9091	0.9591
combi_8	10.0	0.5	0.10	0.8696	0.9524	0.9806	0.9463	0.9234	0.9091	0.9591
combi_9	10.0	0.5	0.20	0.8913	0.9762	0.9903	0.9597	0.9395	0.9318	0.9869
combi_10	20.0	0.1	0.05	0.8696	0.9524	0.9806	0.9463	0.9234	0.9091	0.9591
combi_11	20.0	0.1	0.10	0.8696	0.9524	0.9806	0.9463	0.9234	0.9091	0.9591
combi_12	20.0	0.1	0.20	0.8696	0.9756	0.9903	0.9530	0.9280	0.9196	0.9884
combi_13	20.0	0.3	0.05	0.8696	0.9524	0.9806	0.9463	0.9234	0.9091	0.9591
combi_14	20.0	0.3	0.10	0.8696	0.9524	0.9806	0.9463	0.9234	0.9091	0.9591
combi_15	20.0	0.3	0.20	0.8696	0.9302	0.9709	0.9396	0.9189	0.8989	0.9624
combi_16	20.0	0.5	0.05	0.8696	0.9524	0.9806	0.9463	0.9234	0.9091	0.9591
combi_17	20.0	0.5	0.10	0.8696	0.9524	0.9806	0.9463	0.9234	0.9091	0.9591
combi_18	20.0	0.5	0.20	0.8696	0.9302	0.9709	0.9396	0.9189	0.8989	0.9624
combi_19	30.0	0.1	0.05	0.8696	0.9302	0.9709	0.9396	0.9189	0.8989	0.9736
combi_20	30.0	0.1	0.10	0.8696	0.9302	0.9709	0.9396	0.9189	0.8989	0.9662
combi_21	30.0	0.1	0.20	0.8696	0.8696	0.9417	0.9195	0.9049	0.8696	0.9766
combi_22	30.0	0.3	0.05	0.8696	0.9302	0.9709	0.9396	0.9189	0.8989	0.9664
combi_23	30.0	0.3	0.10	0.8696	0.9302	0.9709	0.9396	0.9189	0.8989	0.9662
combi_24	30.0	0.3	0.20	0.8696	0.8696	0.9417	0.9195	0.9049	0.8696	0.9766
combi_25	30.0	0.5	0.05	0.8913	0.9535	0.9806	0.9530	0.9349	0.9214	0.9717
combi_26	30.0	0.5	0.10	0.8696	0.9302	0.9709	0.9396	0.9189	0.8989	0.9662
combi_27	30.0	0.5	0.20	0.8261	0.8636	0.9417	0.9060	0.8820	0.8444	0.9612

```
combi_1 => 변수의 개수 : 22개 / 변수감소율 : 0.27
combi_2 => 변수의 개수 : 22개 / 변수감소율 : 0.27
combi_3 => 변수의 개수 : 19개 / 변수감소율 : 0.37
combi_4 => 변수의 개수 : 22개 / 변수감소율 : 0.27
combi_5 => 변수의 개수 : 22개 / 변수감소율 : 0.27
combi_6 => 변수의 개수 : 19개 / 변수감소율 : 0.37
combi_7 => 변수의 개수 : 22개 / 변수감소율 : 0.27
combi_8 => 변수의 개수 : 22개 / 변수감소율 : 0.27
combi_9 => 변수의 개수 : 15개 / 변수감소율 : 0.5
combi_10 => 변수의 개수 : 20개 / 변수감소율 : 0.33
combi_11 => 변수의 개수 : 20개 / 변수감소율 : 0.33
combi_12 => 변수의 개수 : 16개 / 변수감소율 : 0.47
combi_13 => 변수의 개수 : 20개 / 변수감소율 : 0.33
combi_14 => 변수의 개수 : 20개 / 변수감소율 : 0.33
combi_15 => 변수의 개수 : 19개 / 변수감소율 : 0.37
combi_16 => 변수의 개수 : 20개 / 변수감소율 : 0.33
combi_17 => 변수의 개수 : 20개 / 변수감소율 : 0.33
combi_18 => 변수의 개수 : 19개 / 변수감소율 : 0.37
combi_19 => 변수의 개수 : 20개 / 변수감소율 : 0.33
combi_20 => 변수의 개수 : 20개 / 변수감소율 : 0.33
combi_21 => 변수의 개수 : 13개 / 변수감소율 : 0.57
combi_22 => 변수의 개수 : 20개 / 변수감소율 : 0.33
combi_23 => 변수의 개수 : 20개 / 변수감소율 : 0.33
combi_24 => 변수의 개수 : 13개 / 변수감소율 : 0.57
combi_25 => 변수의 개수 : 19개 / 변수감소율 : 0.37
combi_26 => 변수의 개수 : 20개 / 변수감소율 : 0.33
combi_27 => 변수의 개수 : 19개 / 변수감소율 : 0.37
```

먼저, 분류 성능 표에서 ACC, BCR, F1을 종합적으로 보았을 때, population_size 값이 30인 것이 10, 20인 것에 비해 다소 낮은 수치를 보이는 것을 확인할 수 있다. 특히, population_size 값이 30이면서 mutation_rate가 0.20으로 높은 combi_21, combi_24, combi_27이 다른 combi에 비해서 ACC, BCR, F1 모두 현저히 낮은 수치를 보이는 것을 확인할 수 있다. 이들의 공통점은 population_size 값이 제일 큰 30이고, mutation_rate 값도 제일 큰 0.20이라는 점이다. 이때, 너무 높은 돌연변이율은 Genetic Algorithm의 수렴 속도를 늦추는데, population_size가 큰 상황에서 mutation_rate도 높다면, 돌연변이의 수가 많다는 것이고, 그 결과 성능 하락으로 이어졌다는 해석이 가능하다. 실제로, population_size 값이 30일 때, mutation_rate가 0.05인 것과 0.20인 것을 비교하면, 0.05인 것이 0.20인 것보다 ACC, BCR, F1 값 모두 높은 것을 확인할 수 있다.

crossover_rate에 따른 차이도 있는지 확인해보았지만, ACC, BCR, F1, AUROC, 변수감소율 등 모든 지표에서 뚜렷한 성능의 차이를 발견하지 못하였다.

다음으로 mutation_rate 값이 높은 경우, 변수감소율이 높은 것을 확인할 수 있다. 위의 27가지 경우를 살펴보면, 전체 30개의 변수 중 평균적으로 선택한 변수의 개수가 20개이고, 제거된 변수는 약 10개인 것을 알 수 있다. 이때, mutation이 발생하면 선택한 변수를 제거하거나 제거된 변수를 재선택하는 것이 가능한데, 평균적으로 선택한 변수의 수가 더 많으므로 mutation이 일어났을 때, 변수를 선택하는 쪽보다 제거하는 쪽의 확률이 더 크다는 것을 알 수 있다. 결론적으로 해당 데이터셋에서는 mutation이 일어나면, 변수를 새롭게 선택할 확률보다 제거할 확률이 더 크기 때문에, mutation_rate 값이 크다면 변수감소율도 크다는 것을 알 수 있다.

각 파라미터들의 조합에서 'ACC, BCR, F1, AUROC, 변수감소율'을 종합적으로 살펴보았을 때, 높은 성능을 보이는 조합은 combi_9, combi_12, combi_25가 있다. 이 중, combi_9이 모든 성능지표(ACC, BCR, F1, AUROC)에서 가장 뛰어난 성능을 보이고, 변수감소율도 가장 큰 것을 확인할 수 있다. 결론적으로 27개의 조합 중에서는 (population_size = 10.0, crossover_rate = 0.5, mutation_rate = 0.20)이 현재 데이터셋에서 가장 최적의 파라미터 조합인 것을 알 수 있다.

combi_9, combi_12, combi_25 조합의 변수 선택 결과는 다음과 같다.

combi_9 (15개) : ['radius_mean', 'texture_mean', 'smoothness_mean', 'compactness_mean', 'concavity_mean', 'fractal_dimension_mean', 'radius_se', 'perimeter_se', 'area_se', 'concave points_se', 'fractal_dimension_se', 'texture_worst', 'smoothness_worst', 'compactness_worst', 'fractal_dimension_worst']

combi_12 (16개) : ['radius_mean', 'texture_mean', 'smoothness_mean', 'compactness_mean', 'concavity_mean', 'fractal_dimension_mean', 'radius_se', 'perimeter_se', 'area_se', 'concave points_se', 'fractal_dimension_se', 'texture_worst', 'smoothness_worst', 'compactness_worst', 'fractal_dimension_worst']

combi_25 (19개) : ['radius_mean', 'texture_mean', 'perimeter_mean', 'smoothness_mean', 'concavity_mean', 'concave points_mean', 'fractal_dimension_mean', 'radius_se', 'perimeter_se', 'area_se', 'compactness_se', 'concavity_se', 'symmetry_se', 'radius_worst', 'texture_worst', 'perimeter_worst', 'smoothness_worst', 'concave points_worst', 'fractal_dimension_worst']

높은 성능을 보이는 조합에 모두 포함되는 공통변수는 ['radius_mean', 'texture_mean', 'smoothness_mean', 'concavity_mean', 'fractal_dimension_mean', 'radius_se', 'perimeter_se', 'area_se', 'texture_worst', 'smoothness_worst', 'fractal_dimension_worst']로 나타났다.

그러나 ['radiuse_se', 'perimeter_se', 'area_se']가 서로 상관계수가 0.9가 넘어갈 정도로 강한 상관관계를 가지고 있고, ['texture_mean', 'text_worst'], ['radius_mean', 'radius_se']도 강한 상관관계를 가지고 있는 것을 보아 아직 다중공산성과 같은 문제를 제대로 해결하지 못한 모습을 보이고 있다. 이때, combi_9, combi_12, combi_25는 공통적으로 선택한 변수 외에도 더 많은 변수를 가지고 있으므로 해당 문제는 더 커질 것이라 판단되고, 따라서 이들은 통계적으로 유의미한 결과를 도출하지 못했다고 해석이 가능하다.

실제로 변수 선택 기법으로 forward selection을 사용했을 때, 훨씬 더 적은 6개의 변수 ['texture_mean' 'concavity_mean' 'radius_worst' 'compactness_worst' 'concavity_worst' 'concave points_worst']로 combi_9, combi_12, combi_25보다 ACC, BCR, F1, AUROC, 변수감소율에서 모두 더 높은 수치를 보이고, 따라서 더 성능이 뛰어난 것을 확인할 수 있다.

즉, 이번 과제의 데이터셋에서는 변수 선택 기법으로 forward selection이 가장 적합하다고 결론을 내릴 수 있다.