통계 스터디 실습 _ 당뇨데이터

1. 목표

- 통계 데이터셋으로 데이터분석 진행 해보기
- 1. 통계기법 사용
- 2. 머신러닝 기법 사용

2. 데이터분석

- 1. 기초 통계량 확인
- 2. 결측치 확인
- 3. 정규화
- 4. 변수선택
- 5. 모델링
- 6. 검정

1. 기초통계량 확인

data

	age	sex	bmi	bp	s1	s2	s3	s4	s5	s6	target
0	0.038076	0.050680	0.061696	0.021872	-0.044223	-0.034821	-0.043401	-0.002592	0.019907	-0.017646	151.0
1	-0.001882	-0.044642	-0.051474	-0.026328	-0.008449	-0.019163	0.074412	-0.039493	-0.068332	-0.092204	75.0
2	0.085299	0.050680	0.044451	-0.005670	-0.045599	-0.034194	-0.032356	-0.002592	0.002861	-0.025930	141.0
3	-0.089063	-0.044642	-0.011595	-0.036656	0.012191	0.024991	-0.036038	0.034309	0.022688	-0.009362	206.0
4	0.005383	-0.044642	-0.036385	0.021872	0.003935	0.015596	0.008142	-0.002592	-0.031988	-0.046641	135.0
...	...	...	...	...	...	...	...	...	...	...	...
437	0.041708	0.050680	0.019662	0.059744	-0.005697	-0.002566	-0.028674	-0.002592	0.031193	0.007207	178.0
438	-0.005515	0.050680	-0.015906	-0.067642	0.049341	0.079165	-0.028674	0.034309	-0.018114	0.044485	104.0
439	0.041708	0.050680	-0.015906	0.017293	-0.037344	-0.013840	-0.024993	-0.011080	-0.046883	0.015491	132.0
440	-0.045472	-0.044642	0.039062	0.001215	0.016318	0.015283	-0.028674	0.026560	0.044529	-0.025930	220.0
441	-0.045472	-0.044642	-0.073030	-0.081413	0.083740	0.027809	0.173816	-0.039493	-0.004222	0.003064	57.0

442 rows × 11 columns

1. 기초통계량 확인

- 1. 데이터는 442 x 11 442개의 행과 11개의 열로 이루어져있음
- 2. 10개의 특성 (feature)를 가지고 있고 1개의 target을 가지고 있음
- 3. 데이터의 type은 float로 수치형 데이터로 되어 있음 -> 회귀분석 진행가능
- 4. 다만 성별의 경우 애매할 수 있음

```
data.dtypes
```

	0
age	float64
sex	float64
bmi	float64
bp	float64
s1	float64
s2	float64
s3	float64
s4	float64
s5	float64
s6	float64
target	float64

```
data
```

	age	sex	bmi	bp	s1	s2	s3	s4	s5	s6	target
0	0.038076	0.050680	0.061696	0.021872	-0.044223	-0.034821	-0.043401	-0.002592	0.019907	-0.017646	151.0
1	-0.001882	-0.044642	-0.051474	-0.026328	-0.008449	-0.019163	0.074412	-0.039493	-0.068332	-0.092204	75.0
2	0.085299	0.050680	0.044451	-0.005670	-0.045599	-0.034194	-0.032356	-0.002592	0.002861	-0.025930	141.0
3	-0.089063	-0.044642	-0.011595	-0.036656	0.012191	0.024991	-0.036038	0.034309	0.022688	-0.009362	206.0
4	0.005383	-0.044642	-0.036385	0.021872	0.003935	0.015596	0.008142	-0.002592	-0.031988	-0.046641	135.0
...	...	...	...	...	...	...	...	...	...	...	...
437	0.041708	0.050680	0.019662	0.059744	-0.005697	-0.002566	-0.028674	-0.002592	0.031193	0.007207	178.0
438	-0.005515	0.050680	-0.015906	-0.067642	0.049341	0.079165	-0.028674	0.034309	-0.018114	0.044485	104.0
439	0.041708	0.050680	-0.015906	0.017293	-0.037344	-0.013840	-0.024993	-0.011080	-0.046883	0.015491	132.0
440	-0.045472	-0.044642	0.039062	0.001215	0.016318	0.015283	-0.028674	0.026560	0.044529	-0.025930	220.0
441	-0.045472	-0.044642	-0.073030	-0.081413	0.083740	0.027809	0.173816	-0.039493	-0.004222	0.003064	57.0

442 rows × 11 columns

1. 기초통계량 확인

- 현재 데이터셋을 보면 age가 0.038 , sex가 일정한 2값이 반복 되는것을 확인
  - 이 값은 정규화가 되어 있다는 것을 알 수 있음
  - 따라서 정규화는 빼고 진행

	age	sex	bmi	bp
0	0.038076	0.050680	0.061696	0.021872
1	-0.001882	-0.044642	-0.051474	-0.026328
2	0.085299	0.050680	0.044451	-0.005670
3	-0.089063	-0.044642	-0.011595	-0.036656
4	0.005383	-0.044642	-0.036385	0.021872
...	...	...	...	...
437	0.041708	0.050680	0.019662	0.059744
438	-0.005515	0.050680	-0.015906	-0.067642
439	0.041708	0.050680	-0.015906	0.017293
440	-0.045472	-0.044642	0.039062	0.001215
441	-0.045472	-0.044642	-0.073030	-0.081413

1. 기초통계량 확인

- 1. 결측치는 없다고 판단
- 2. 이상치 또한 없다고 판단

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 442 entries, 0 to 441
Data columns (total 11 columns):
 #   Column  Non-Null Count  Dtype  
---  ------  --------------  -----  
 0   age     442 non-null    float64
 1   sex     442 non-null    float64
 2   bmi     442 non-null    float64
 3   bp      442 non-null    float64
 4   s1      442 non-null    float64
 5   s2      442 non-null    float64
 6   s3      442 non-null    float64
 7   s4      442 non-null    float64
 8   s5      442 non-null    float64
 9   s6      442 non-null    float64
 10  target  442 non-null    float64
dtypes: float64(11)
memory usage: 38.1 KB
```

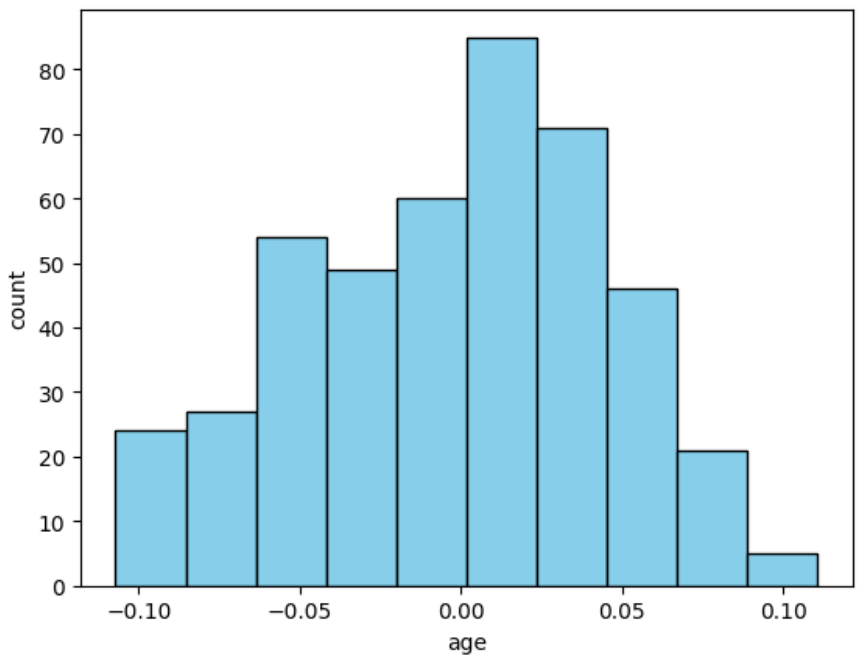
```
# 소수점 형식 설정 (지수 표기 대신 소수점)
pd.options.display.float_format = '{:.2f}'.format

# describe 출력
print(data.describe())
```

```
          age    sex    bmi     bp     s1     s2     s3     s4     s5     s6  \
count  442.00 442.00 442.00 442.00 442.00 442.00 442.00 442.00 442.00 442.00   
mean    -0.00   0.00  -0.00  -0.00  -0.00   0.00  -0.00  -0.00   0.00   0.00   
std      0.05   0.05   0.05   0.05   0.05   0.05   0.05   0.05   0.05   0.05   
min     -0.11  -0.04  -0.09  -0.11  -0.13  -0.12  -0.10  -0.08  -0.13  -0.14   
25%     -0.04  -0.04  -0.03  -0.04  -0.03  -0.03  -0.04  -0.04  -0.03  -0.03   
50%      0.01  -0.04  -0.01  -0.01  -0.00  -0.00  -0.01  -0.00  -0.00  -0.00   
75%      0.04   0.05   0.03   0.04   0.03   0.03   0.03   0.03   0.03   0.03   
max      0.11   0.05   0.17   0.13   0.15   0.20   0.18   0.19   0.13   0.14   

       target  
count  442.00  
mean   152.13  
std     77.09  
min     25.00  
25%     87.00  
50%    140.50  
75%    211.50  
max    346.00  
```

1. 기초통계량 확인

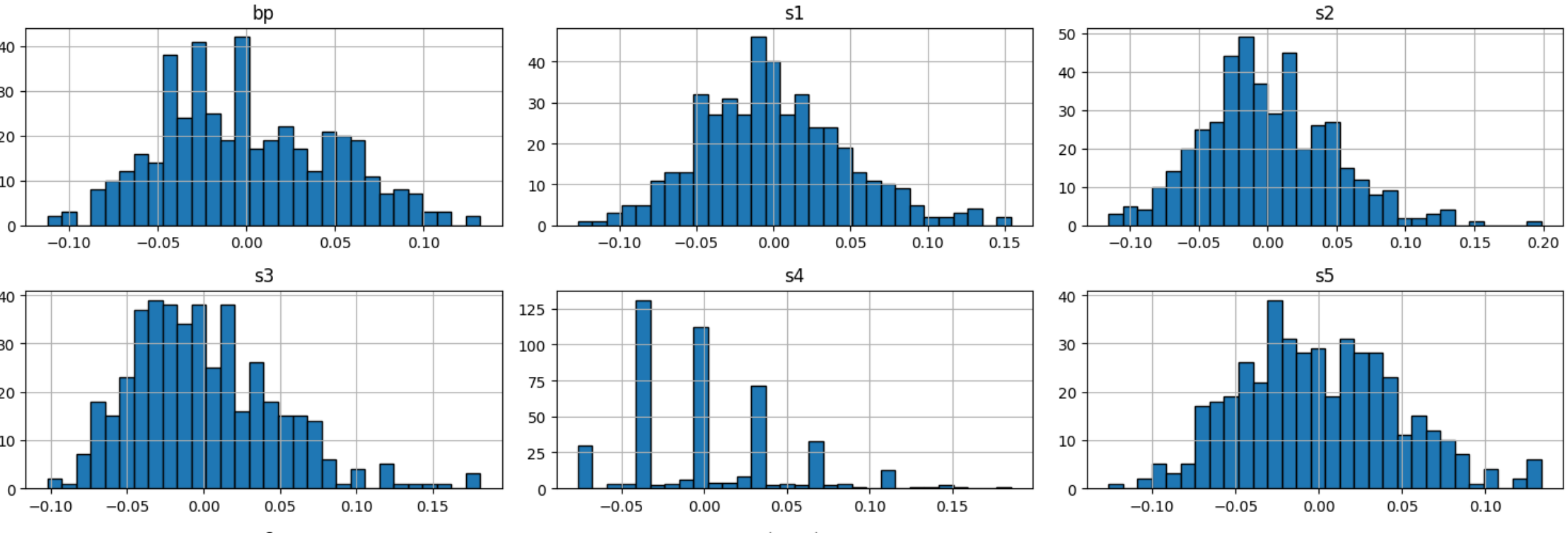
- Age의 경우 정규분포라고 판단 됌

```
import matplotlib.pyplot as plt
column_name = 'age'
plt.hist(data[column_name], bins=10, color='skyblue', edgecolor='black')

plt.xlabel(column_name)
plt.ylabel('count')
plt.show()
```



1. 기초통계량 확인

```
# Histograms for numerical columns
data.hist(figsize=(15, 10), bins=30, edgecolor='black')
plt.tight_layout()
plt.show()
```



2. 통계 vs 머신러닝

- 통계적 기법
  - “무엇이 정확한가 " 를 넘어서 “왜 이런 결과가 나왔는 가 " 를 설명할 수 있는 도구
  - 예측보다 해석이 중요한 경우가 있음(비즈니스, 정책, 의학)
  - 의사 결정의 근거 제공 : 어떤 변수에 중점을 둬야 하는가
  - 변수의 중요성 평가 : “어떤 변수를 조정해야 종속변수에 큰 영향을 미치는가?”를 평가

- 머신러닝
  - 블랙박스 모델로 어떤 변수가 어떤 영향을 주었는지 판단 할 수 없음
  - 모델에 대한 설명보다는 예측이 중요함

- 통계적 가설검정은 결론적으로 모델의 신뢰도를 점검한다.

3. 통계적 가설 검정

- P_value
  - 어떤 사건이 우연히 발생할 확률
  - P_value 가 0.05보다 작다 = 어떤 사건이 우연히 일어날 확률이 5%보다 작다.
  - -> 실제로 우리가 현재 검정을 하는 이 값이 우연히 일어 났을리가 없다.
  - -> 따라서 기각한다.

  - P_value가 0.05보다 크다 = 어떤 사건이 우연히 발생할 확률이 있다.
  - -> 실제로 우리가 현재 검정을 하는 이값이 우연히 일어날 수 있다.
  - -> 따라서 기각을 못한다.

3. 통계적 가설검정

- 모델에 대한 검정
- 1. 모형이 얼마나 설명력을 갖는가?
  - 결정계수 R^2으로 확인 (정답 맞춘 비율이 1에 가까울 수록 높음)
- 2. 모형이 통계적으로 유의한가?
  - F-검정과 p-value를 통해
  - F-검정을 통해 이 모델의 적어도 하나의 회귀계수가 0이 아님을 알 수 있음
- 3. 회귀계수가 유의한가?
  - P-value -> 기울기가 유의한지 (이때 가설은 회귀계수가 영향이 없다가 가설)
  - 따라서 p_value값이 0.05아래 즉 적은값을 선택해야 함

4. 모델 설정

- 통계 모델에서 제일 중요한 것은 독립변수를 선택하는것
- 1. 다때려박기
  - 다 때려 박아보고 문제되는거 하나씩 빼기
- 2. 상관관계 분석
  - 상관관계가 높은 것 위주로 분석
- 3. 다중 공선성 확인
  - 다 때려박았을때 다중공선성으로 문제되는거 제거 하기

5. 변수 선택

- Sex를 제외하고 상관관계가 있는 것으로 파악
- 하지만 성별의 경우 원래 categorical해서 상관관계가 없는 걸로 판단
- 추가적으로 상관관계가 있다고 모든 변수가 유의미하지는 않고 상관관계가 없다고 무의미하지 않음

```
# 상관관계 분석
# 현재 target과 상관관계를 보았을때 sex를 제외하고 상관관계가 다 있음
data.corr()
```

	target
age	0.187889
sex	0.043062
bmi	0.586450
bp	0.441482
s1	0.212022
s2	0.174054
s3	-0.394789
s4	0.430453
s5	0.565883
s6	0.382483
target	1.000000

5. 변수 선택

- 다중공선성이 5는 위험 10은 있다고 파악
- 다중공선성이 높은 것은 독립변수들끼리 상관관계가 높아 회귀분석에서 회귀 계수를 설정하기 어려움을 말함

```
vif = pd.DataFrame()
vif["Variable"] = data.columns
vif["VIF"] = [variance_inflation_factor(data.values, i) for i in range(data.shape[1])]
print(vif)
```

```
   Variable        VIF
0       age   1.217315
1       sex   1.283075
2       bmi   1.532949
3        bp   1.468583
4        s1  59.257108
5        s2  39.213144
6        s3  15.403044
7        s4   8.893714
8        s5  10.125073
9        s6   1.485021
10   target   1.118065
```

6. 실험 1

모든 변수로 진행

train-set

r-squre: 0.518

age, s1, s2, s3, s4, s6 유의하지 않음

test-set

r-squre : 0.45

rmse : 53.0

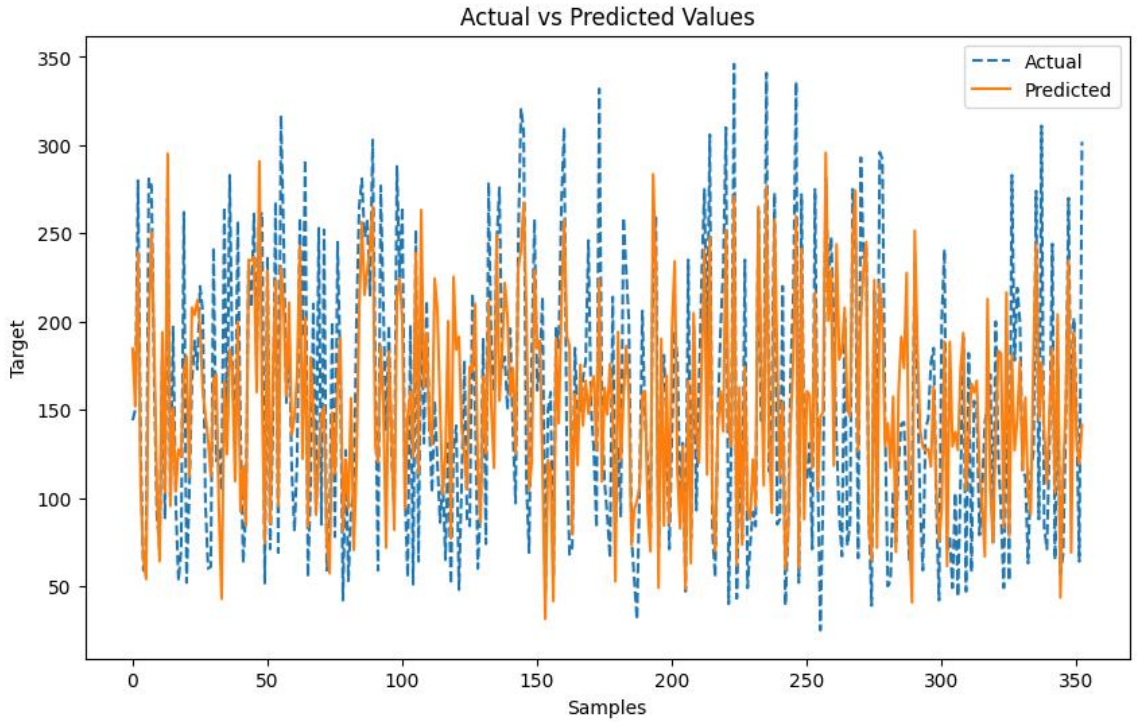
OLS Regression Results

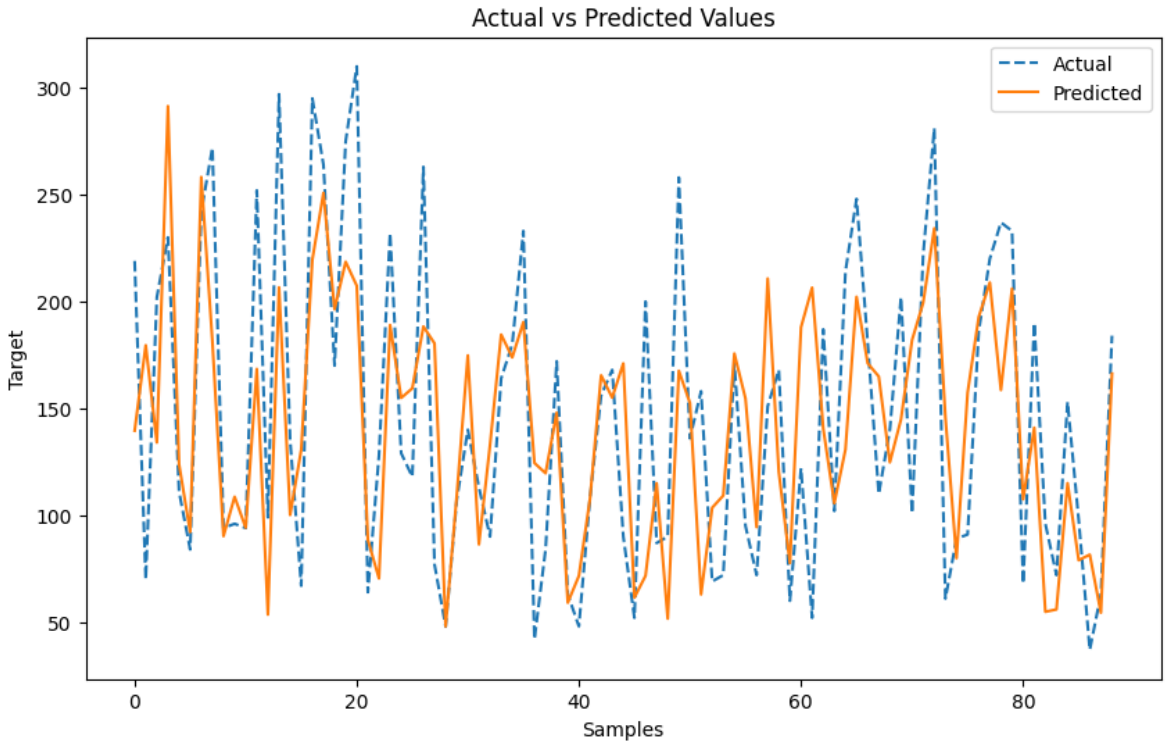
Dep. Variable:	target	R-squared:	0.518
Model:	OLS	Adj. R-squared:	0.507
Method:	Least Squares	F-statistic:	46.27
Date:	Mon, 13 Jan 2025	Prob (F-statistic):	3.83e-62
Time:	12:38:06	Log-Likelihood:	-2386.0
No. Observations:	442	AIC:	4794.
Df Residuals:	431	BIC:	4839.
Df Model:	10		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	152.1335	2.576	59.061	0.000	147.071	157.196
age	-10.0099	59.749	-0.168	0.867	-127.446	107.426
sex	-239.8156	61.222	-3.917	0.000	-360.147	-119.484
bmi	519.8459	66.533	7.813	0.000	389.076	650.616
bp	324.3846	65.422	4.958	0.000	195.799	452.970
s1	-792.1756	416.680	-1.901	0.058	-1611.153	26.802
s2	476.7390	339.030	1.406	0.160	-189.620	1143.098
s3	101.0433	212.531	0.475	0.635	-316.684	518.770
s4	177.0632	161.476	1.097	0.273	-140.315	494.441
s5	751.2737	171.900	4.370	0.000	413.407	1089.140
s6	67.6267	65.984	1.025	0.306	-62.064	197.318

Omnibus:	1.506	Durbin-Watson:	2.029
Prob(Omnibus):	0.471	Jarque-Bera (JB):	1.404
Skew:	0.017	Prob(JB):	0.496
Kurtosis:	2.726	Cond. No.	227.

7. 시각화 – 실험 1


Actual vs Predicted Values
Actual
Predicted
Target
Samples
350
300
250
200
150
100
50
0
50
100
150
200
250
300
350


Actual vs Predicted Values
Actual
Predicted
Target
Samples
300
250
200
150
100
50
0
20
40
60
80

6. 실험 2

age, s1, s2, s3, s4, s6 제외

train-set

r-squre: 0.487

모든 변수 유의함

test-set

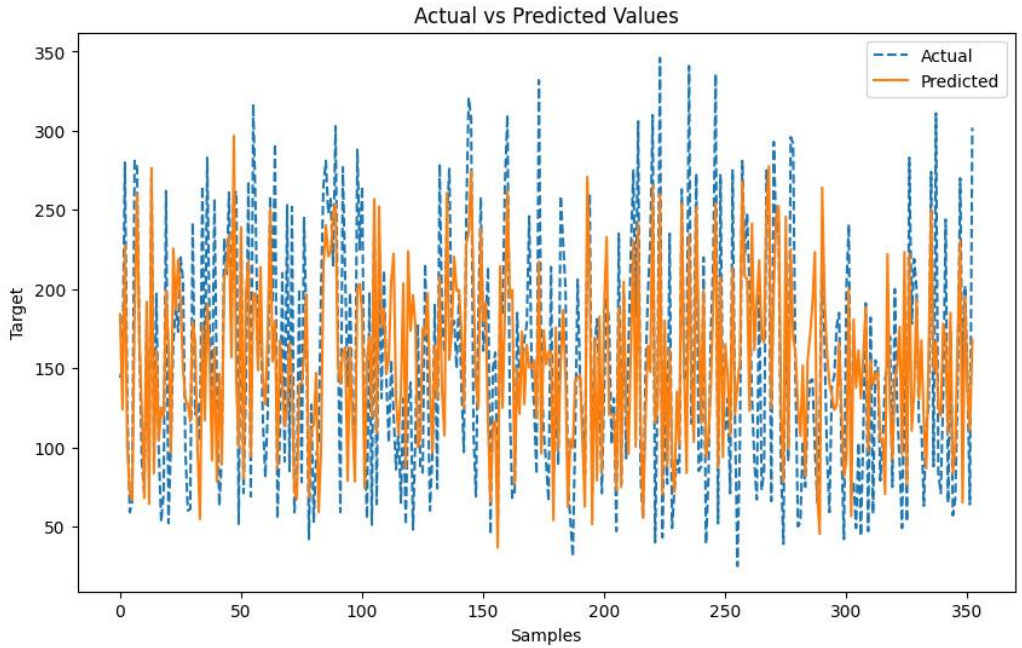
r-squre : 0.467

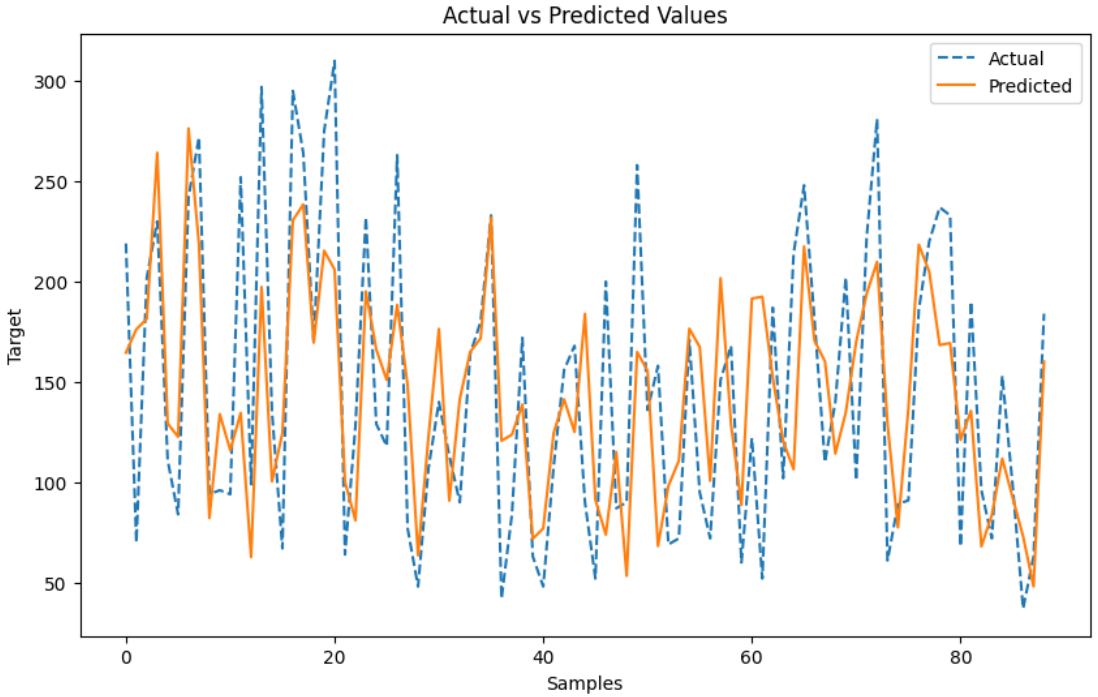
rmse : 53.0

```python
# 앞선 p-value에서 유의하지않은 변수를 차단

import statsmodels.api as sm
X = data[['sex', 'bmi', 'bp', 's5']]
y = data['target']

X = sm.add_constant(X)  # 회귀모델의 절편(intercept) 포함
model = sm.OLS(y, X).fit()

# 결과 출력
print(model.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                 target   R-squared:                       0.487
Model:                            OLS   Adj. R-squared:                  0.482
Method:                 Least Squares   F-statistic:                     103.6
Date:                Mon, 13 Jan 2025   Prob (F-statistic):           5.42e-62
Time:                        12:38:06   Log-Likelihood:                -2399.8
No. Observations:                 442   AIC:                             4810.
Df Residuals:                     437   BIC:                             4830.
Df Model:                           4
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const        152.1335      2.639     57.648      0.000     146.947     157.320
sex         -136.7580     57.304     -2.387      0.017    -249.383     -24.132
bmi          598.2839     64.365      9.295      0.000     471.781     724.786
bp           292.9722     63.935      4.582      0.000     167.314     418.630
s5           554.4326     64.427      8.606      0.000     427.807     681.059
==============================================================================
Omnibus:                        5.261   Durbin-Watson:                   1.982
Prob(Omnibus):                  0.072   Jarque-Bera (JB):                4.282
Skew:                           0.145   Prob(JB):                        0.118
Kurtosis:                       2.614   Cond. No.                         28.5
==============================================================================

...
```

7. 시각화 – 실험 2


Actual vs Predicted Values
Actual
Predicted
Target
Samples
0
50
100
150
200
250
300
350
50
100
150
200
250
300
350


Actual vs Predicted Values
Actual
Predicted
Target
Samples
0
20
40
60
80
50
100
150
200
250
300

6. 실험 3

```
# 높은 VIF를 가진 변수 제거
X_reduced = data.drop(['s1', 's2'], axis=1)

# VIF 재계산
from statsmodels.stats.outliers_influence import variance_inflation_factor
vif = pd.DataFrame()
vif["Variable"] = X_reduced.columns
vif["VIF"] = [variance_inflation_factor(X_reduced.values, i) for i in range(X_reduced.shape[1])]
print(vif)
```

```
  Variable       VIF
0      age  1.207569
1      sex  1.279405
2      bmi  1.510066
3       bp  1.463022
4       s3  2.440660
5       s4  3.155841
6       s5  1.997245
7       s6  1.484402
8   target  1.116354
```

다중 공선성이 높은 S1,S2를 제거하니 다중 공선성이 해결 됨

6. 실험 3

s1,s2제외

train-set

r-squre: 0.511

age, s4, s6 회귀계수 유의하지 않음

t값이 크면 현재 가설인 회귀계수가 0이다가 기각되서 대립가설인 회귀계수가 유의

test-set

r-squre : 0.46

rmse : 53.0

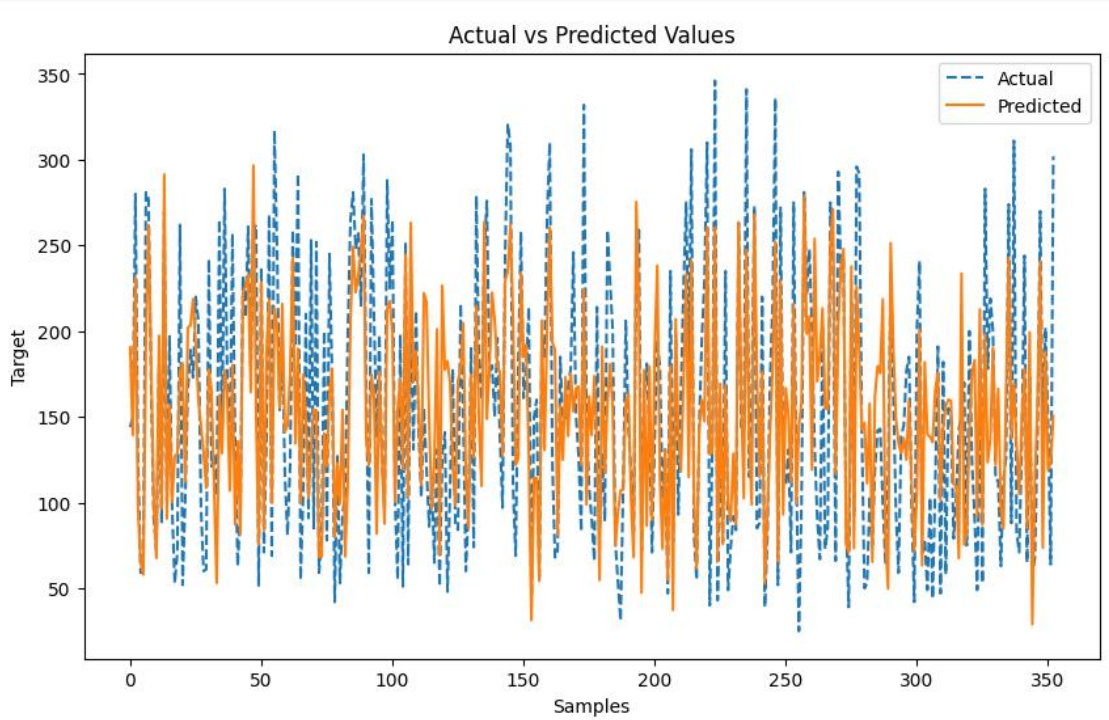
OLS Regression Results

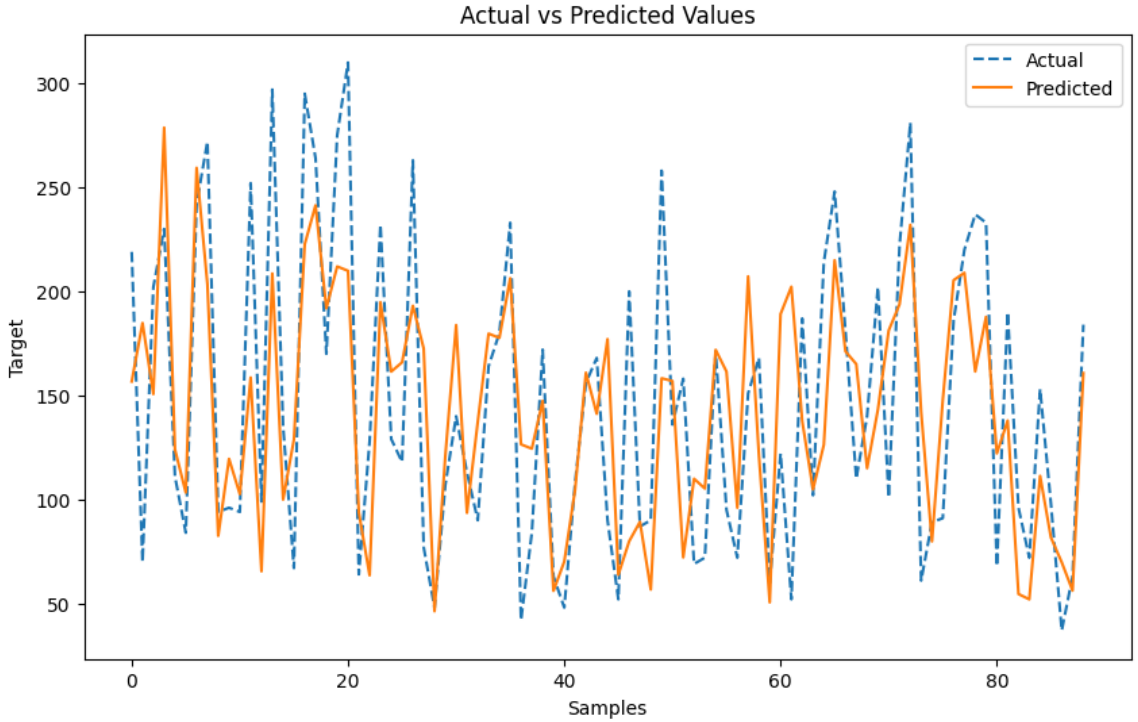
Dep. Variable:	target	R-squared:	0.511
Model:	OLS	Adj. R-squared:	0.502
Method:	Least Squares	F-statistic:	56.57
Date:	Mon, 13 Jan 2025	Prob (F-statistic):	1.26e-62
Time:	12:38:06	Log-Likelihood:	-2389.1
No. Observations:	442	AIC:	4796.
Df Residuals:	433	BIC:	4833.
Df Model:	8		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	152.1335	2.588	58.790	0.000	147.047	157.220
age	-16.0121	59.784	-0.268	0.789	-133.515	101.491
sex	-232.6033	61.424	-3.787	0.000	-353.329	-111.877
bmi	518.0802	66.336	7.810	0.000	387.699	648.461
bp	315.1609	65.611	4.804	0.000	186.206	444.116
s3	-346.3372	84.812	-4.084	0.000	-513.031	-179.643
s4	-110.4294	96.631	-1.143	0.254	-300.354	79.495
s5	499.0379	76.469	6.526	0.000	348.742	649.334
s6	67.4445	66.275	1.018	0.309	-62.816	197.705

Omnibus:	2.301	Durbin-Watson:	2.000
Prob(Omnibus):	0.317	Jarque-Bera (JB):	1.954
Skew:	0.032	Prob(JB):	0.376
Kurtosis:	2.681	Cond. No.	46.7

7. 시각화 – 실험 3


Actual vs Predicted Values
Actual
Predicted
Target
Samples
350
300
250
200
150
100
50
0
50
100
150
200
250
300
350


Actual vs Predicted Values
Actual
Predicted
Target
Samples
300
250
200
150
100
50
0
20
40
60
80

6. 실험 4

s1,s2, age, s4, s6 제외

train-set

r-squre: 0.509

모든 변수 유의 함

test-set

r-squre : 0.46

rmse : 53.0

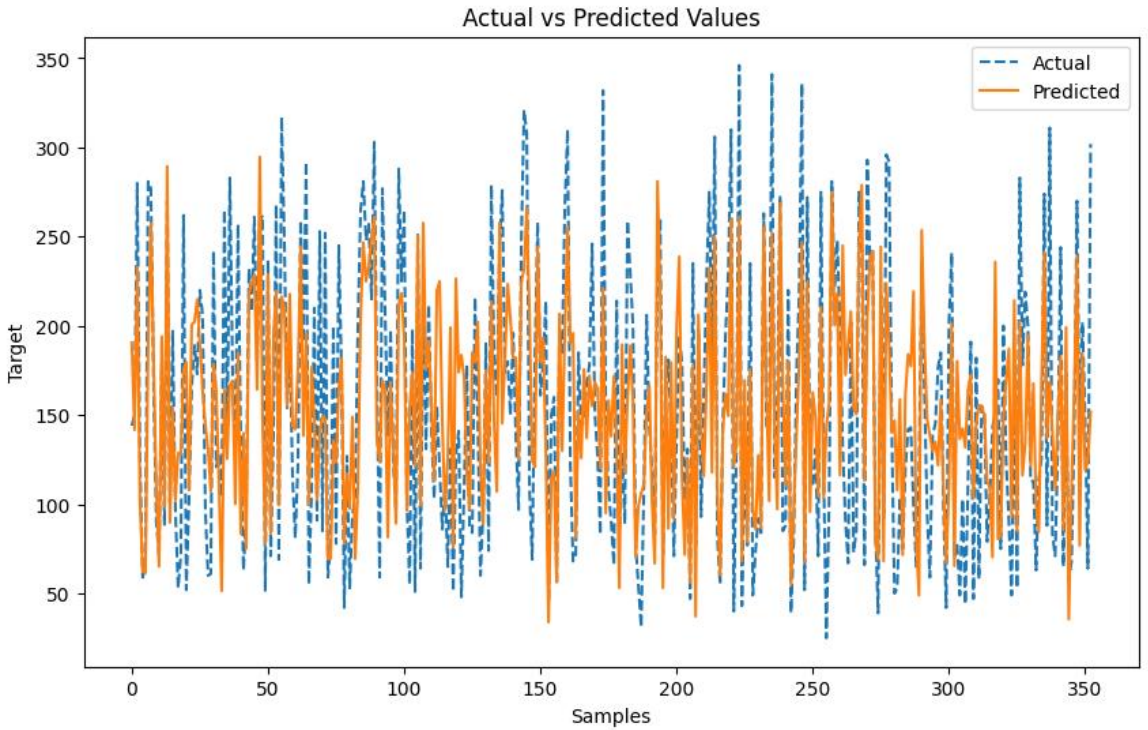
OLS Regression Results

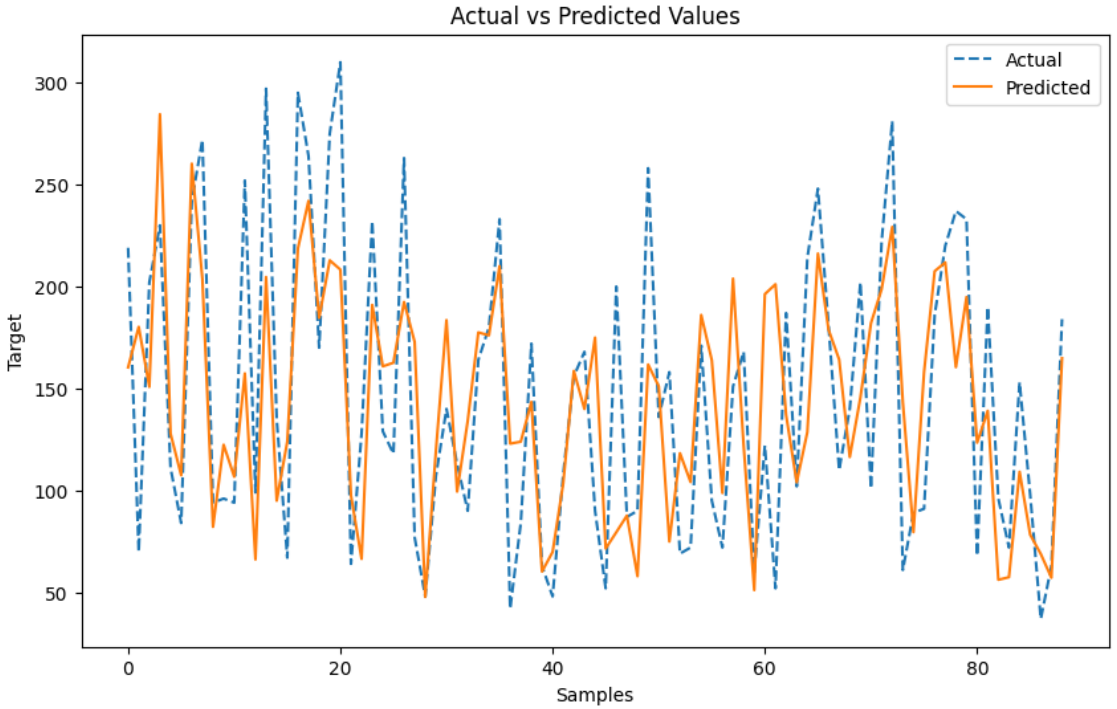
Dep. Variable:	target	R-squared:	0.509
Model:	OLS	Adj. R-squared:	0.503
Method:	Least Squares	F-statistic:	90.26
Date:	Mon, 13 Jan 2025	Prob (F-statistic):	4.75e-65
Time:	12:38:06	Log-Likelihood:	-2390.1
No. Observations:	442	AIC:	4792.
Df Residuals:	436	BIC:	4817.
Df Model:	5		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	152.1335	2.585	58.849	0.000	147.053	157.214
sex	-235.7724	60.469	-3.899	0.000	-354.620	-116.925
bmi	523.5678	65.293	8.019	0.000	395.239	651.897
bp	326.2311	63.084	5.171	0.000	202.245	450.217
s3	-289.1148	65.646	-4.404	0.000	-418.136	-160.094
s5	474.2902	65.683	7.221	0.000	345.195	603.386

Omnibus:	2.465	Durbin-Watson:	1.990
Prob(Omnibus):	0.291	Jarque-Bera (JB):	2.099
Skew:	0.051	Prob(JB):	0.350
Kurtosis:	2.678	Cond. No.	32.7

7. 시각화 – 실험 4


Actual vs Predicted Values
Actual
Predicted
Target
Samples
350
300
250
200
150
100
50
0
50
100
150
200
250
300
350


Actual vs Predicted Values
Actual
Predicted
Target
Samples
300
250
200
150
100
50
0
20
40
60
80

6. 실험 5

- 주성분 분석을 이용한 다중공선성 제거

```
from sklearn.decomposition import PCA
# PCA 적용
pca = PCA(n_components=2)  # 주성분 수 선택
X_pca = pca.fit_transform(data[['s1','s2','s3','s4','s5','s6']])
```

```
주성분 기여율: [0.54594331 0.21808828]
누적 기여율: [0.54594331 0.76403159]
```

PCA

```
import statsmodels.api as sm
X = new_data[['sex','bmi','bp','s1','s2']]
y = data['target']
```

다중공선성이 높은 feature들의 차원을 6개에서 2개로 줄임
2개가 약 76%를 설명함

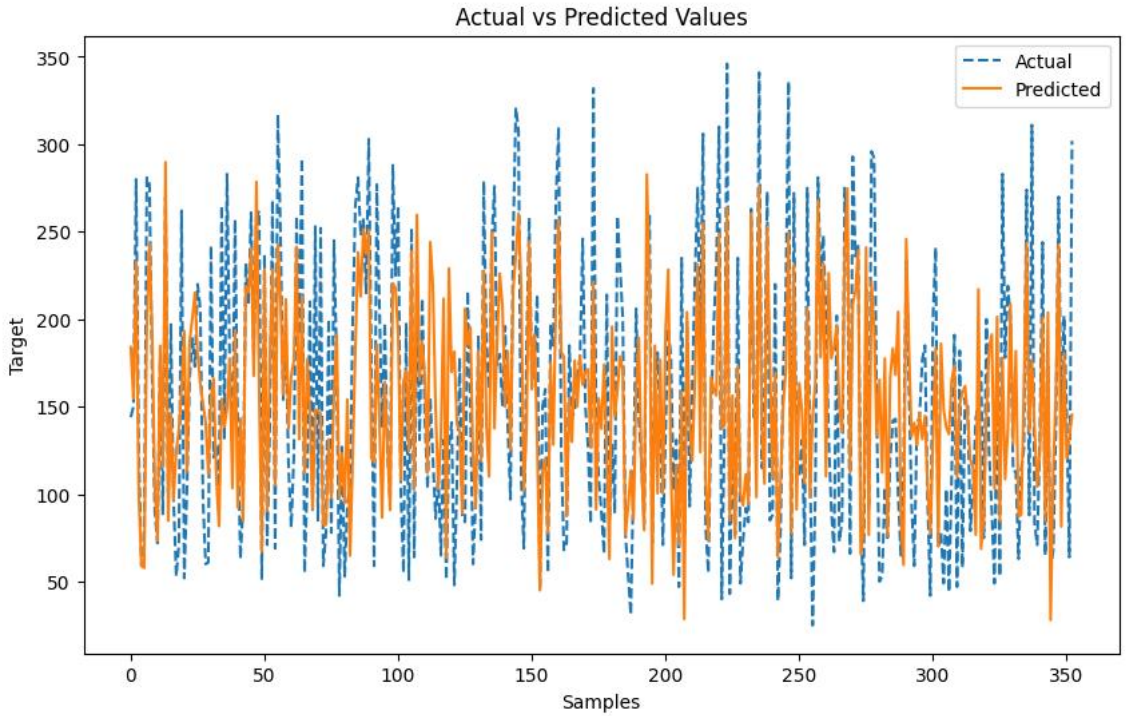
OLS Regression Results

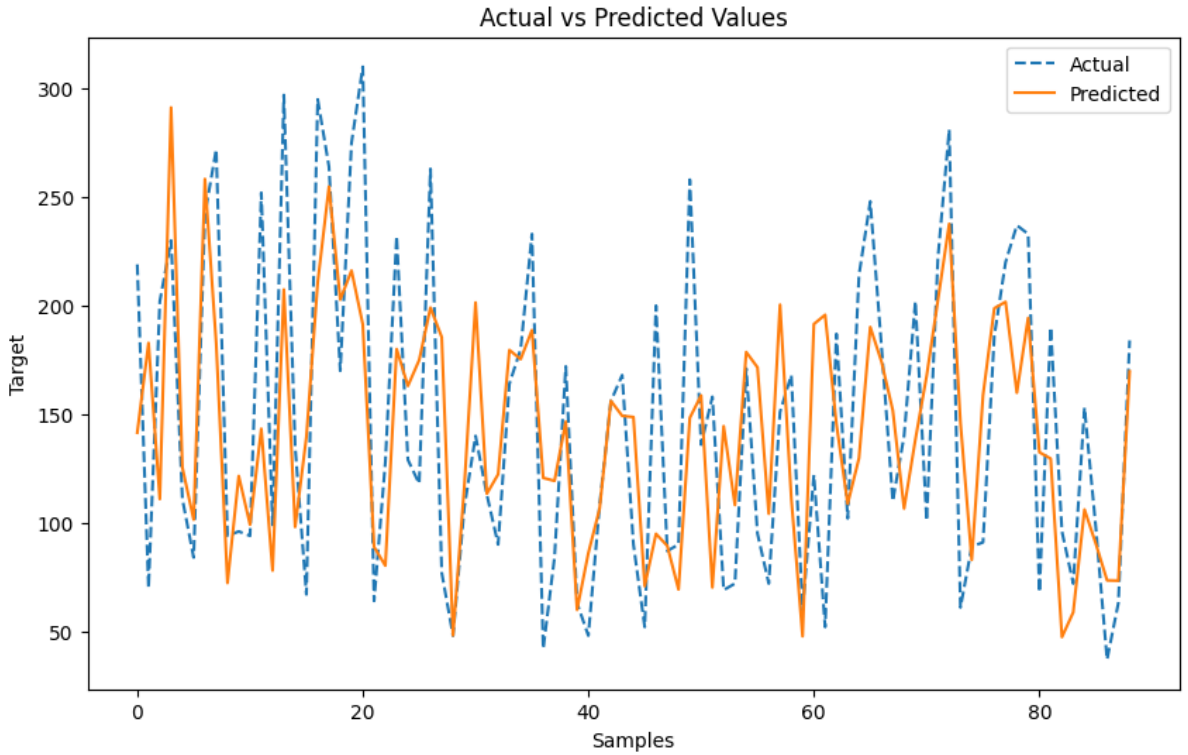
Dep. Variable:	target	R-squared:	0.489
Model:	OLS	Adj. R-squared:	0.483
Method:	Least Squares	F-statistic:	83.41
Date:	Mon, 13 Jan 2025	Prob (F-statistic):	2.40e-61
Time:	13:25:39	Log-Likelihood:	-2398.8
No. Observations:	442	AIC:	4810.
Df Residuals:	436	BIC:	4834.
Df Model:	5		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	152.1335	2.637	57.702	0.000	146.952	157.315
sex	-291.8792	61.126	-4.775	0.000	-412.017	-171.741
bmi	539.8853	67.378	8.013	0.000	407.460	672.311
bp	378.3984	62.915	6.014	0.000	254.743	502.053
s1	253.4718	37.036	6.844	0.000	180.680	326.264
s2	-333.9744	51.674	-6.463	0.000	-435.536	-232.413

Omnibus:	3.160	Durbin-Watson:	1.941
Prob(Omnibus):	0.206	Jarque-Bera (JB):	2.473
Skew:	0.031	Prob(JB):	0.290
Kurtosis:	2.639	Cond. No.	30.2

7. 시각화 – 실험 5


Actual vs Predicted Values
Actual
Predicted
Target
Samples
350
300
250
200
150
100
50
0
50
100
150
200
250
300
350


Actual vs Predicted Values
Actual
Predicted
Target
Samples
300
250
200
150
100
50
0
20
40
60
80

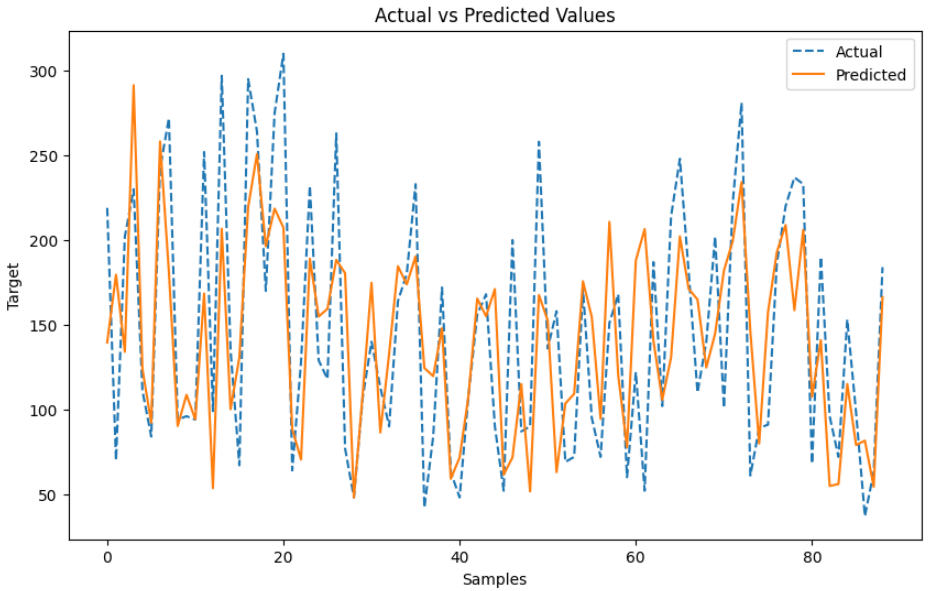
8. 결론

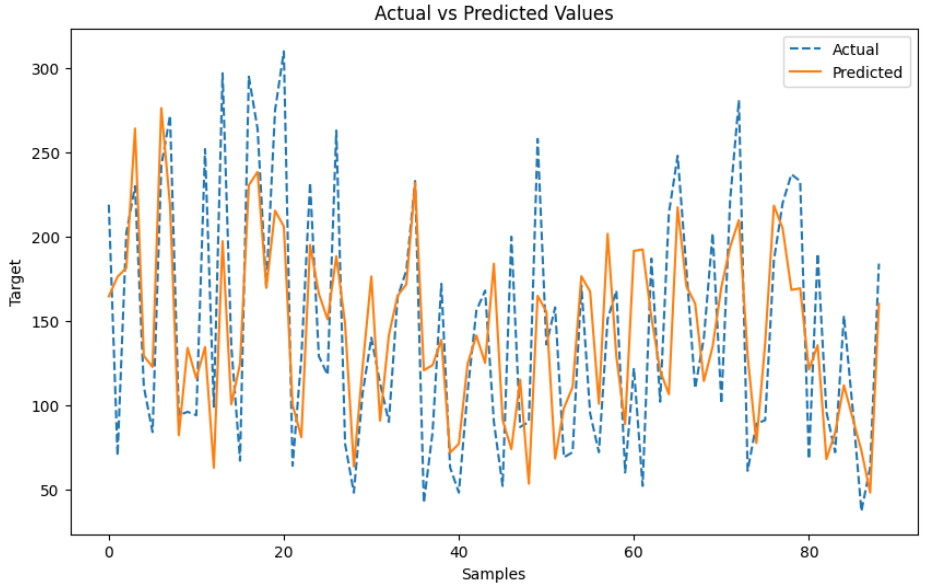
	변수	변수 개수	Train R-squre	Test R-squre	Test Rmse	모든 회귀 계수 유의하지 않음
실험1	'age','sex', 'bmi', 'bp', 's1', 's2', 's3', 's4', 's5', 's6'	10	0.518	0.452	53.85	O
실험2	'sex', 'bmi', 'bp', 's5'	4	0.487	0.467	53.10	X
실험3	'age', 'sex', 'bmi', 'bp', 's3', 's4', 's5', 's6'	8	0.511	0.468	53.04	O
실험4	'sex', 'bmi', 'bp', 's3', 's5'	5	**0.509**	**0.469**	**53.02**	X
실험5	'sex','bmi','bp','s7','s8'	5	0.489	0.412	55.79	X

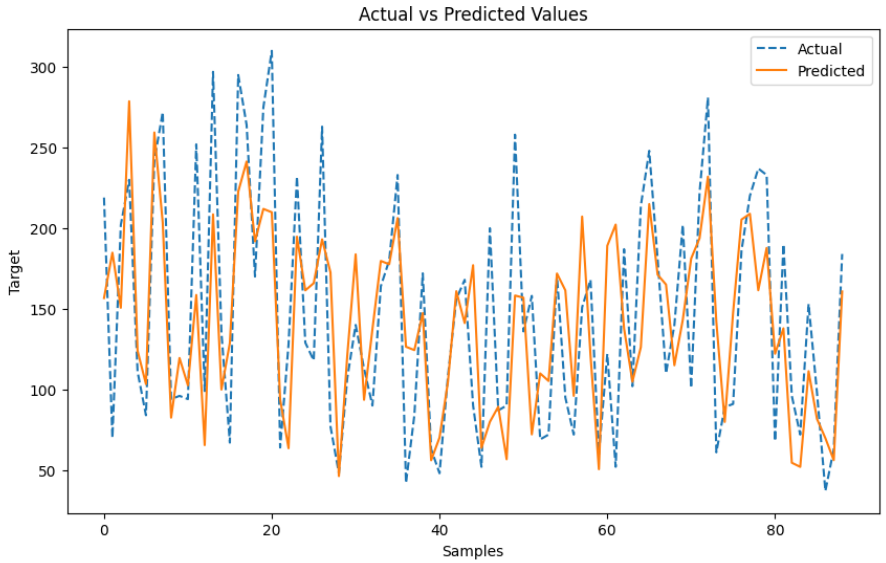
실험 1, 2 다 때려보고 p_value 높은값 빼기

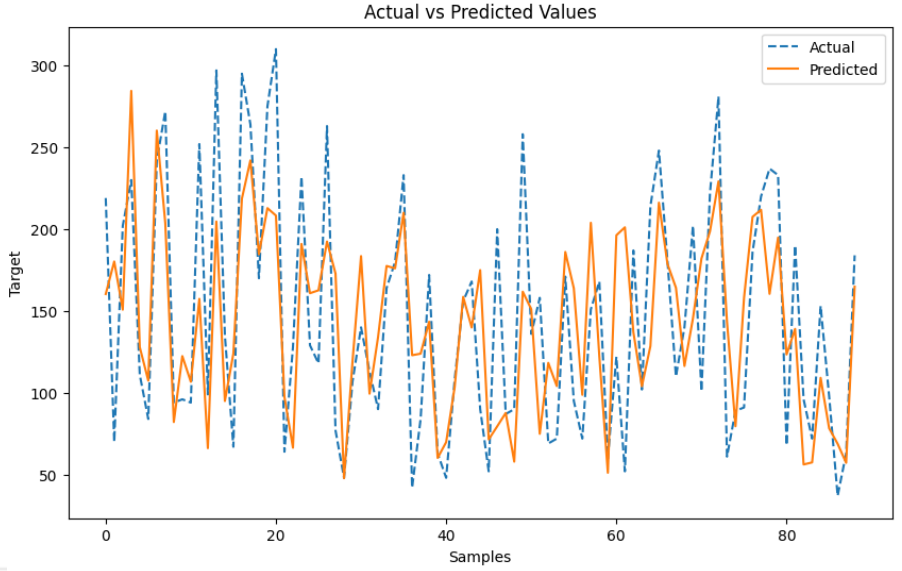
실험 3, 4, 5 다중 공선성 확인하고 해결해보기

6. 결론


Actual vs Predicted Values
Actual
Predicted
Target
300
250
200
150
100
50
0
20
40
60
80
Samples


Actual vs Predicted Values
Actual
Predicted
Target
300
250
200
150
100
50
0
20
40
60
80
Samples


Actual vs Predicted Values
Actual
Predicted
Target
300
250
200
150
100
50
0
20
40
60
80
Samples


Actual vs Predicted Values
Actual
Predicted
Target
300
250
200
150
100
50
0
20
40
60
80
Samples

9. 머신러닝 실험 1

```
# 데이터 준비
X = data[['age', 'sex', 'bmi', 'bp', 's1', 's2', 's3', 's4', 's5', 's6']].values
y = data['target'].values

# 데이터 나누기
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=5)


# 데이터를 Tensor로 변환
X_train_tensor = torch.tensor(X_train, dtype=torch.float32)
X_test_tensor = torch.tensor(X_test, dtype=torch.float32)
y_train_tensor = torch.tensor(y_train, dtype=torch.float32).view(-1, 1)  # 출력값 형태 (n, 1)
y_test_tensor = torch.tensor(y_test, dtype=torch.float32).view(-1, 1)

# MLP 모델 정의
class MLPModel(nn.Module):
    def __init__(self, input_size, hidden_size, output_size):
        super(MLPModel, self).__init__()
        self.fc1 = nn.Linear(input_size, hidden_size)  # 첫 번째 레이어
        self.relu = nn.ReLU()  # 활성화 함수
        self.fc2 = nn.Linear(hidden_size, hidden_size)  # 두 번째 레이어
        self.fc3 = nn.Linear(hidden_size, output_size)  # 출력 레이어

    def forward(self, x):
        x = self.fc1(x)
        x = self.relu(x)
        x = self.fc2(x)
        x = self.relu(x)
        x = self.fc3(x)
        return x
```

```
Train Loss (MSE): 2717.8386
Train RMSE: 52.1329
Train R² : 0.5334
```

```
Test Loss (MSE): 2893.6426
Test RMSE: 53.7926
Test R² : 0.5411
```

9. 머신러닝 실험 2, 3, 4

```
# 데이터 준비
X = data[['sex', 'bmi', 'bp', 's5']].values
y = data['target'].values

# 데이터 나누기
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=5)
```

```
Train Loss (MSE): 2954.0618
Train RMSE: 54.3513
Train R² : 0.4928

Test Loss (MSE): 3054.3647
Test RMSE: 55.2663
Test R² : 0.5156
```

```
# 데이터 준비
X = data[['sex', 'bmi', 'bp', 's3','s5']].values
y = data['target'].values

# 데이터 나누기
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=5)
```

```
Train Loss (MSE): 2728.0825
Train RMSE: 52.2310
Train R² : 0.5316

Test Loss (MSE): 2961.0903
Test RMSE: 54.4159
Test R² : 0.5304
```

```
X = new_data[['sex','bmi','bp','s1','s2']].values
y = data['target'].values

# 데이터 나누기
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=5)
```

```
Train Loss (MSE): 2821.5713
Train RMSE: 53.1185
Train R² : 0.5156

Test Loss (MSE): 3259.1372
Test RMSE: 57.0889
Test R² : 0.4831
```

9. 머신러닝 실험 결과

	변수	변수 개수	Train R-squre	Test R-squre	Test Rmse
실험1	'age','sex', 'bmi', 'bp', 's1', 's2', 's3', 's4', 's5', 's6'	10	**0.533**	**0.541**	**53.79**
실험2	'sex', 'bmi', 'bp', 's5'	4	0.492	0.515	55.26
실험4	'sex', 'bmi', 'bp', 's3', 's5'	5	0.531	0.530	55.41
실험5	'sex','bmi','bp','s7','s8'	5	0.515	0.483	57.08

10. 최종 결과

	변수	변수 개수	Train R-squre	Test R-squre	Test Rmse	Train R-squre	Test R-squre	Test Rmse
실험1	'age','sex', 'bmi', 'bp', 's1', 's2', 's3', 's4', 's5', 's6'	10	0.518	0.452	53.85	**0.533**	**0.541**	**53.79**
실험2	'sex', 'bmi', 'bp', 's5'	4	0.487	0.467	53.10	0.492	0.515	55.26
실험3	'age', 'sex', 'bmi', 'bp', 's3', 's4', 's5', 's6'	8	0.511	0.468	53.04			
실험4	'sex', 'bmi', 'bp', 's3', 's5'	5	**0.509**	**0.469**	**53.02**	0.531	0.530	55.41
실험5	'sex','bmi','bp','s7','s8'	5	0.489	0.412	55.79	0.515	0.483	57.08

머신러닝은 블랙박스이기 때문에 각 변수의 회귀계수를 알 수 없음

Test-성능에서 실험 1이 제일 성능이 좋음

통계검증에서는 실험 4가 제일 test 와 train의 모델을 제일 잘 설명 rmse또한 낮음