PM2.5 Final Report

Three method to linear regression: Gradient descent …know how and why

Maximum likelihood estimation …know how only

Maximum a posterior …know how only

PM2.5 dataset

Features:17D

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
	AMB_TE	CH4	CO	NMHC	NO	NO2	NOx	O3	PM10	RAINFAI	RH	SO2	THC	WD_HR	WIND_D	WIND_SI	WS_HR
	19.5	1.9	0.4175	0.089167	1.054167	10.78333	11.97917	42.95833	54.04167	0	71.29167	2.383333	1.9875	62.04167	62.16667	2.770833	2.320833
	18.70833	1.9625	0.60375	0.173333	5.420833	20.225	25.64583	26.17083	89.375	0.016667	83.66667	2.495833	2.120833	76.25	87.70833	1.529167	1.1125
	19.16667	2.079167	0.721667	0.27375	14.7375	29.58333	44.41667	4.904167	55.29167	0.816667	90.41667	2.408333	2.354167	94.91667	84.375	1.15	0.758333
	20.20833	2.154167	0.996667	0.360417	23.4	22.975	46.41667	9.4375	42.75	0.008333	88.54167	2.9125	2.504167	188.5417	186.7083	0.883333	0.491667
	20.375	2.033333	0.654583	0.324583	15.87391	24.69565	40.52174	7.291667	55.08696	0.008333	87.33333	3.433333	2.345833	119.9167	133.5417	1.3375	0.895833
	18.25	1.9	0.375417	0.16125	1.779167	17.14167	19.02083	24.61667	26.20833	1.825	86.45833	2.258333	2.075	62.04167	60.66667	2.633333	1.8375
	18	1.881818	0.41875	0.135455	2.426087	12.31304	14.68696	38.75	47.5	0	75.04167	2.583333	1.990909	61.33333	62.08333	3.675	2.920833
	16.875	1.891667	0.44875	0.12375	1.3125	14.025	15.44583	36.70833	51.20833	0.083333	77.375	2.420833	1.9875	65.04167	67.5	2.85	2.325
	17.33333	1.9125	0.557917	0.144583	3.441667	17.87083	21.26667	32.91667	49.16667	0	71.54167	3.554167	2.045833	63.66667	61.91667	2.8375	2.204167
	18.375	1.954167	0.542917	0.189583	5.879167	17.65	23.67083	21.25417	45.625	0	81.45833	2.654167	2.145833	94.125	85.79167	1.425	1.029167
	16.875	1.845833	0.415833	0.1325	3.841667	19.50417	23.35833	27.7375	19.625	2.3	89.41667	1.666667	1.991667	68	75.375	1.904167	1.329167
	16.29167	1.854545	0.382174	0.099545	2.104348	10.78261	12.90435	34	32.82609	0.066667	80.125	1.326087	1.945455	64.08333	64.875	3.629167	2.766667
1	14.25	1.9	0.477917	0.117917	1.129167	11.39167	12.31667	33.79167	27.125	0	82.33333	1.1625	1.983333	64.54167	64.45833	3.525	2.641667
	14.66667	1.941667	0.64375	0.17375	3.016667	17.6125	20.625	31	61.33333	0	75.58333	3.420833	2.116667	66.875	66.79167	2.895833	2.266667
	15.625	1.958333	0.653333	0.213333	4.929167	26.125	31.08333	19.45	70.29167	0.083333	83.75	3.1625	2.183333	68.83333	69.25	1.725	1.408333
	18.66667	1.929167	0.515	0.146667	3.020833	17.99167	20.85	26.72083	33.25	0.016667	77	2.466667	2.075	74.91667	80.58333	2.116667	1.683333
	16.875	1.945833	0.51375	0.148333	7.004167	16.49167	23.54167	22.95417	26.45833	2.025	89.83333	1.795833	2.095833	114.8083	126.5833	1.5625	0.925
	14.83333	1.913043	0.542917	0.13913	2.816667	13.4625	16.33333	36.58333	51.3913	0.008696	70.33333	2.083333	2.03913	71.375	71.625	3.3625	2.741667
	14.58333	1.895833	0.5875	0.149167	3.3	15.51667	18.74583	36.58333	72.875	0.033333	74.58333	3.708333	2.05	71.20833	71.58333	3.7	3.0625
	15.5	1.883333	0.511667	0.18	4.416667	19.25	23.5875	26.81667	26.75	0	78	3.0625	2.075	82.58333	83.91667	2.7875	2.266667
	15.54167	1.858333	0.374583	0.15375	5.254167	16.1	21.29583	22.7875	18.45833	0.666667	88.95833	1.9125	2.033333	77.79167	74.41667	2.075	1.495833
	15.25	1.841667	0.329167	0.135417	4.941667	15.8375	20.725	26.81667	14.20833	1.341667	86	2.083333	1.9875	61.70833	61.625	2.9	2.254167
	10.0375	1.866667	0.420417	0.082917	2.216667	10.62083	12.74583	32.29167	34.95833	0.9	84.5	2.1375	1.9375	64.79167	67.20833	4.504167	3.716667
	4.970833	1.804167	0.3225	0.08375	1.604167	8.495833	9.991667	35.5	40.16667	0.258333	70.29167	2.583333	1.908333	77.125	76.79167	3.458333	2.920833
	7.833333	1.841667	0.335	0.098333	2.991667	11.48333	14.43333	33.33333	40.375	0	61.33333	2.270833	1.945833	71.75	69.95833	3.795833	3.045833
	13.87917	1.947826	0.59	0.266957	12.38696	23.82609	36.13043	22.23913	35.73913	0	64.25	3.117391	2.217391	111.9167	108.4583	1.445833	0.991667

Target:

#Date	PM2.5
2016/1/1	27.70833
2016/1/2	46.08333
2016/1/3	38.47826
2016/1/4	23.52174
2016/1/5	31.47826
2016/1/6	6.916667
2016/1/7	21.625
2016/1/8	25.5
2016/1/9	22.29167
2016/1/10	21.16667
2016/1/11	6.375
2016/1/12	11.13043
2016/1/13	11.75
2016/1/14	34.95833
2016/1/15	43.29167
2016/1/16	14.91667
2016/1/17	10.16667
2016/1/18	24.65217
2016/1/19	44
2016/1/20	10.58333
2016/1/21	5.666667
2016/1/22	2.958333
2016/1/23	17.625
2016/1/24	12.75

Four basic function: Polynomial (M=1): $h(x,\theta) = \sum_{i=0}^{17} \theta_i x_i$

Polynomial (M=2): $h(x,\theta) = \sum_{i=0}^{17} \theta_i x_i + \sum_{i=1}^{17} \sum_{j=1}^{i} \theta_n x_i x_j\,, n\ in\ range\ (18{,}171)$

Sigmoidal: $S(x) = \frac{e^x}{1+e^x}$

Gaussian: $f(x,\mu,\sigma) = \frac{1}{\sigma\sqrt{2\pi}} e^{\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)}$

Error estimation: $RMSE$

Polynomial (M=1)

Dataset: $\begin{cases} training: 80\% \\ testing: 20\% \end{cases}$

data

Normalization

Shuffle

Train test split

Linear regression (G.D and MLE)

PM2.5(M=1)—gradient descent

Hypothesis: $h_\theta(x) = \theta_0 x_0 + \theta_1 x_1 + \ldots. + \theta_{17} x_{17}$

Cost function: $J(\theta_0, \theta_1, \ldots., \theta_{16}, \theta_{17}) = \dfrac{1}{2N} \sum_{i=1}^{N} \left(h_\theta\left(x^i\right) - T^i\right)^2$

Gradient descent: $repeat\ before\ convergence\{$

$$\theta_n = \theta_n - \alpha \frac{\partial J(\theta_0, \theta_1, \ldots, \theta_{17})}{\partial \theta_n}$$

$\}$

PM2.5(M=1)—gradient descent

Pseudocode:

Read the data
Split into training set and testing set

↓

Initialize theta
Update theta by gradient descent

↓

Get the hypothesis function

↙ ↘

Use the training target to test the model

Use the testing target to test the model

PM2.5(M=1)hypothesis function

x_i^j:the ith feature in the jth data

$x_0^i = 1$

Just to put the constant of the hypothesis function into the matrix

$$h_\theta(x) = [\theta_0 \quad \theta_1 \quad \dots \quad \theta_{16} \quad \theta_{17}]_{1X18} \cdot \begin{bmatrix} x_0^1 & x_0^2 & \dots & x_0^{N-1} & x_0^N \\ x_1^1 & x_1^2 & \dots & x_1^{N-1} & x_1^N \\ \dots & \dots & \dots & \dots & \dots \\ x_{16}^1 & x_{16}^2 & \dots & x_{16}^{N-1} & x_{16}^N \\ x_{17}^1 & x_{17}^2 & \dots & x_{17}^{N-1} & x_{17}^N \end{bmatrix}_{18XN} = [y^1 \quad y^2 \quad \dots \quad y^{N-1} \quad y^N]_{1XN}$$

Transpose from the original data

Theta_grad: $= \left[\frac{\partial J}{\partial \theta_0} \quad \frac{\partial J}{\partial \theta_1} \quad \dots \quad \frac{\partial J}{\partial \theta_{16}} \quad \frac{\partial J}{\partial \theta_{17}}\right]_{1X18}$

$$= \frac{1}{N}\left[\sum_{i=1}^{N}\left(h_\theta\left(x^i\right) - T^i\right)\cdot x_0^i \quad \sum_{i=1}^{N}\left(h_\theta\left(x^i\right) - T^i\right)\cdot x_1^i \quad \dots \quad \sum_{i=1}^{N}\left(h_\theta\left(x^i\right) - T^i\right)\cdot x_{16}^i \quad \sum_{i=1}^{N}\left(h_\theta\left(x^i\right) - T^i\right)\cdot x_{17}^i\right]_{1X18}$$

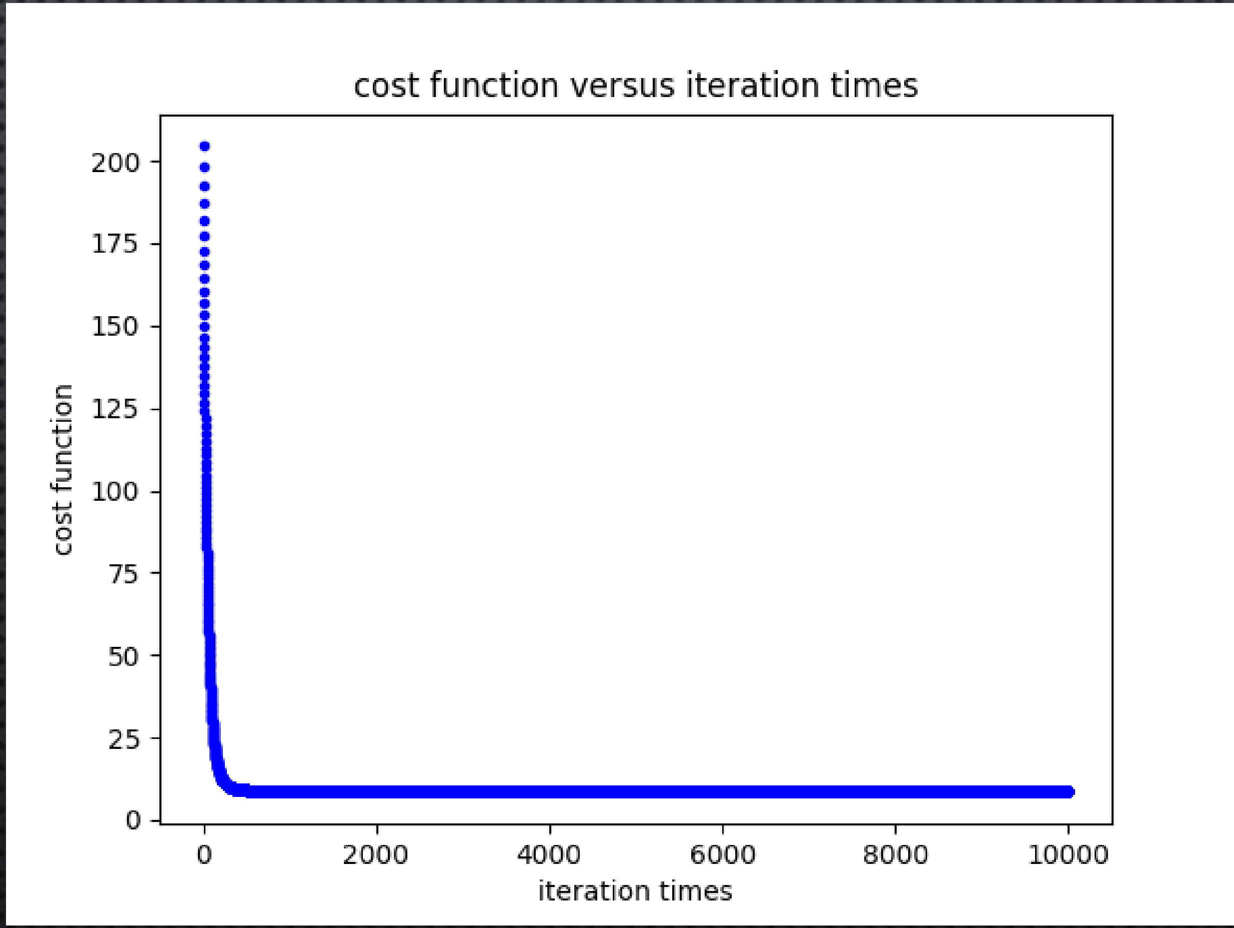
$$= \frac{1}{N}\left([y^1 \quad y^2 \quad \dots \quad y^{N-1} \quad y^N]_{1XN} - [T^1 \quad T^2 \quad \dots \quad T^{N-1} \quad T^N]_{1XN}\right)\cdot \begin{bmatrix} x_0^1 & x_0^2 & \dots & x_0^{N-1} & x_0^N \\ x_1^1 & x_1^2 & \dots & x_1^{N-1} & x_1^N \\ \dots & \dots & \dots & \dots & \dots \\ x_{16}^1 & x_{16}^2 & \dots & x_{16}^{N-1} & x_{16}^N \\ x_{17}^1 & x_{17}^2 & \dots & x_{17}^{N-1} & x_{17}^N \end{bmatrix}_{18XN}$$

Hypothesis: $h_\theta(x) = \theta_0 x_0 + \theta_1 x_1 + \dots. + \theta_{17} x_{17}$

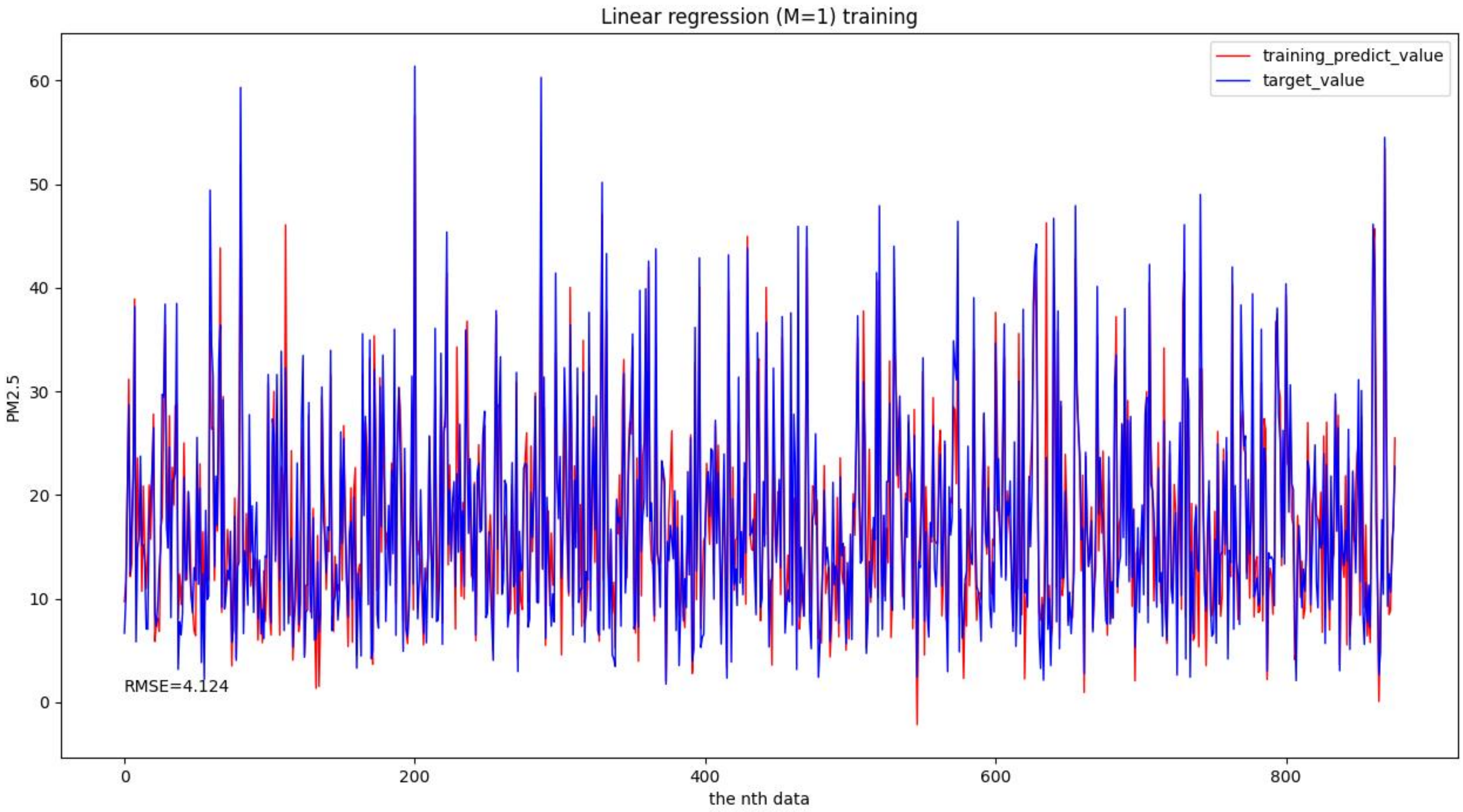
Cost function: $J(\theta_0, \theta_1, \dots., \theta_{16}, \theta_{17}) = \frac{1}{2N}\sum_{i=1}^{N}\left(h_\theta\left(x^i\right) - T^i\right)^2$

Gradient descent: $\theta_n = \theta_n - \alpha \frac{\partial J(\theta_0, \theta_1, \dots, \theta_{17})}{\partial \theta_n}$

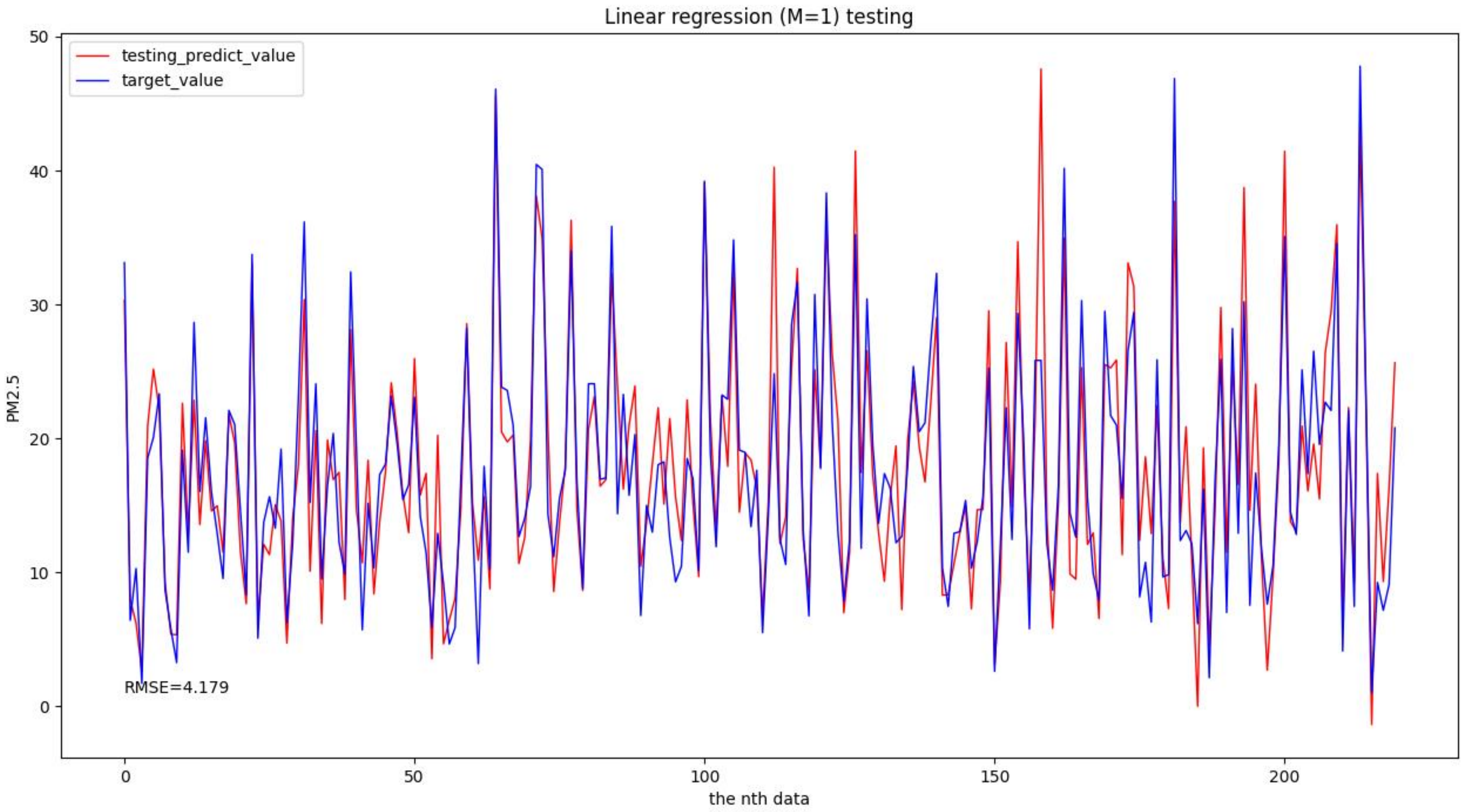
Result:


cost function versus iteration times
cost function
iteration times
200
175
150
125
100
75
50
25
0
0
2000
4000
6000
8000
10000

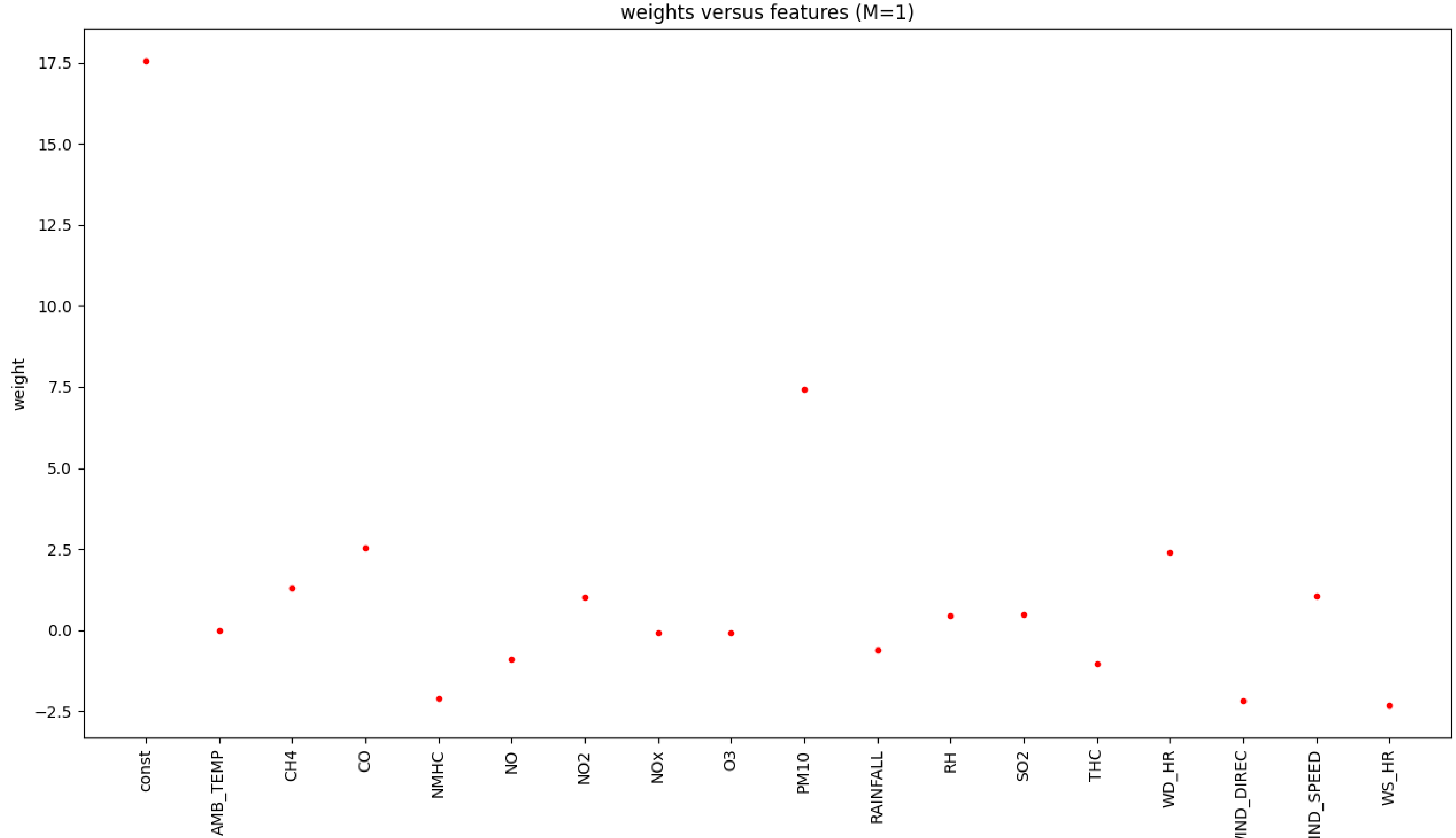
Result:


Linear regression (M=1) training
training_predict_value
target_value
PM2.5
60
50
40
30
20
10
0
RMSE=4.124
0
200
400
600
800
the nth data

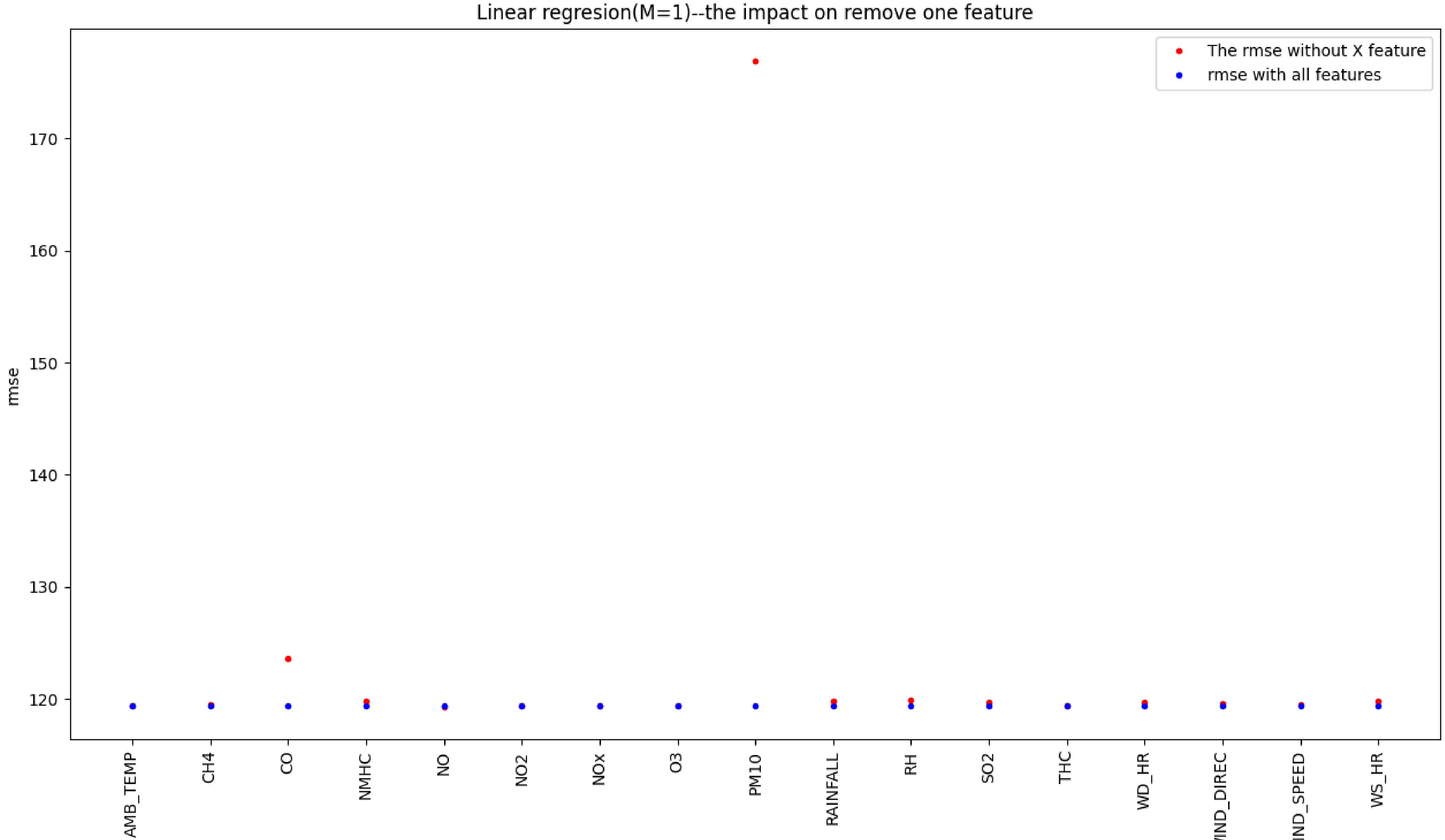
Result:


Linear regression (M=1) testing
testing_predict_value
target_value
PM2.5
the nth data
RMSE=4.179
0
10
20
30
40
50
0
50
100
150
200

Result:


weights versus features (M=1)
weight
17.5
15.0
12.5
10.0
7.5
5.0
2.5
0.0
−2.5
const
AMB_TEMP
CH4
CO
NMHC
NO
NO2
NOx
O3
PM10
RAINFALL
RH
SO2
THC
WD_HR
WS_HR

Result:



Linear regresion(M=1)--the impact on remove one feature
The rmse without X feature
rmse with all features
rmse
170
160
150
140
130
120
AMB_TEMP
CH4
CO
NMHC
NO
NO2
NOx
O3
PM10
RAINFALL
RH
SO2
THC
WD_HR
WIND_DIREC
WIND_SPEED
WS_HR

PM2.5(M=2)—gradient descent

Hypothesis: $h_\theta(x) = \theta_0 x_0 + \theta_1 x_1 + \ldots + \theta_{17} x_{17} +$

$\theta_{18} x_1 x_1 +$

$\theta_{19} x_2 x_1 + \theta_{20} x_2 x_2 +$

$\theta_{21} x_3 x_1 + \theta_{22} x_3 x_2 + \theta_{23} x_3 x_3 +$

$\ldots.$

$\theta_{154} x_{17} x_1 + \theta_{155} x_{17} x_2 + \ldots + \theta_{169} x_{17} x_{16} + \theta_{170} x_{17} x_{17}$

$$\Rightarrow h_\theta(x) = \sum_{i=0}^{17} \theta_i x_i + \sum_{i=1}^{17} \sum_{j=1}^{i} \theta_n x_i x_j \, , n \; in \; range(18{,}171)$$

Cost function: $J(\theta_0, \theta_1, \ldots., \theta_{169}, \theta_{170}) = \dfrac{1}{2N} \displaystyle\sum_{i=1}^{N} \left(h_\theta(x^i) - T^i\right)^2$

Gradient descent: $repeat \; before \; convergence\{$

$$\theta_n = \theta_n - \alpha \frac{\partial J(\theta_0, \theta_1, \ldots, \theta_{170})}{\partial \theta_n}$$

$\}$

PM2.5(*M*=2)—gradient descent

Data preprocess

Append the data (or features) by multiplied two features :

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22……………169 170

1

AMB_TE	CH4	CO	NMHC	NO	NO2	NOx	O3	PM10	RAINFAI	RH	SO2	THC	WD_HR	WIND_D	WIND_SI	WS_HR
19.5	1.9	0.4175	0.089167	1.054167	10.78333	11.97917	42.95833	54.04167	0	71.29167	2.383333	1.9875	62.04167	62.16667	2.770833	2.320833
18.70833	1.9625	0.60375	0.173333	5.420833	20.225	25.64583	26.17083	89.375	0.016667	83.66667	2.495833	2.120833	76.25	87.70833	1.529167	1.1125
19.16667	2.079167	0.721667	0.27375	14.7375	29.58333	44.41667	4.904167	55.29167	0.816667	90.41667	2.408333	2.354167	94.91667	84.375	1.15	0.758333
20.20833	2.154167	0.996667	0.360417	23.4	22.975	46.41667	9.4375	42.75	0.008333	88.54167	2.9125	2.504167	188.5417	186.7083	0.883333	0.491667
20.375	2.033333	0.654583	0.324583	15.87391	24.69565	40.52174	7.291667	55.08696	0.008333	87.33333	3.433333	2.345833	119.9167	133.5417	1.3375	0.895833
18.25	1.9	0.375417	0.16125	1.779167	17.14167	19.02083	24.61667	26.20833	1.825	86.45833	2.258333	2.075	62.04167	60.66667	2.633333	1.8375
18	1.881818	0.41875	0.135455	2.426087	12.31304	14.68696	38.75	47.5	0	75.04167	2.583333	1.990909	61.33333	62.08333	3.675	2.920833
16.875	1.891667	0.44875	0.12375	1.3125	14.025	15.44583	36.70833	51.20833	0.083333	77.375	2.420833	1.9875	65.04167	67.5	2.85	2.325
17.33333	1.9125	0.557917	0.144583	3.441667	17.87083	21.26667	32.91667	49.16667	0	71.54167	3.554167	2.045833	63.66667	61.91667	2.8375	2.204167
18.375	1.954167	0.542917	0.189583	5.879167	17.65	23.67083	21.25417	45.625	0	81.45833	2.654167	2.145833	94.125	85.79167	1.425	1.029167
16.875	1.845833	0.415833	0.1325	3.841667	19.50417	23.35833	27.7375	19.625	2.3	89.41667	1.666667	1.991667	68	75.375	1.904167	1.329167
16.29167	1.854545	0.382174	0.099545	2.104348	10.78261	12.90435	34	32.82609	0.066667	80.125	1.326087	1.945455	64.08333	64.875	3.629167	2.766667
14.25	1.9	0.477917	0.117917	1.129167	11.39167	12.31667	33.79167	27.125	0	82.33333	1.1625	1.983333	64.54167	64.45833	3.525	2.641667
14.66667	1.941667	0.64375	0.17375	3.016667	17.6125	20.625	31	61.33333	0	75.58333	3.420833	2.116667	66.875	66.79167	2.895833	2.266667
15.625	1.958333	0.653333	0.213333	4.929167	26.125	31.08333	19.45	70.29167	0.083333	83.75	3.1625	2.183333	68.83333	69.25	1.725	1.408333
18.66667	1.929167	0.515	0.146667	3.020833	17.99167	20.85	26.72083	33.25	0.016667	77	2.466667	2.075	74.91667	80.58333	2.116667	1.683333
16.875	1.945833	0.51375	0.148333	7.004167	16.49167	23.54167	22.95417	26.45833	2.025	89.83333	1.795833	2.095833	114.8083	126.5833	1.5625	0.925
14.83333	1.913043	0.542917	0.13913	2.816667	13.4625	16.33333	36.58333	51.3913	0.008696	70.33333	2.083333	2.03913	71.375	71.625	3.3625	2.741667
14.58333	1.895833	0.5875	0.149167	3.3	15.51667	18.74583	36.58333	72.875	0.033333	74.58333	3.708333	2.05	71.20833	71.58333	3.7	3.0625
15.5	1.883333	0.511667	0.18	4.416667	19.25	23.5875	26.81667	26.75	0	78	3.0625	2.075	82.58333	83.91667	2.7875	2.266667
15.54167	1.858333	0.374583	0.15375	5.254167	16.1	21.29583	22.7875	18.45833	0.666667	88.95833	1.9125	2.033333	77.79167	74.41667	2.075	1.495833
15.25	1.841667	0.329167	0.135417	4.941667	15.8375	20.725	26.81667	14.20833	1.341667	86	2.083333	1.9875	61.70833	61.625	2.9	2.254167
10.0375	1.866667	0.420417	0.082917	2.216667	10.62083	12.74583	32.29167	34.95833	0.9	84.5	2.1375	1.9375	64.79167	67.20833	4.504167	3.716667
4.970833	1.804167	0.3225	0.08375	1.604167	8.495833	9.991667	35.5	40.16667	0.258333	70.29167	2.583333	1.908333	77.125	76.79167	3.458333	2.920833
7.833333	1.841667	0.335	0.098333	2.991667	11.48333	14.43333	33.33333	40.375	0	61.33333	2.270833	1.945833	71.75	69.95833	3.795833	3.045833
13.87917	1.947826	0.59	0.266957	12.38696	23.82609	36.13043	22.23913	35.73913	0	64.25	3.117391	2.217391	111.9167	108.4583	1.445833	0.991667

18	19	20	21 22……169	170
1 * 1	2 * 1	2 * 2	…	17 * 16
				17 * 17

PM2.5(M=2)—gradient descent

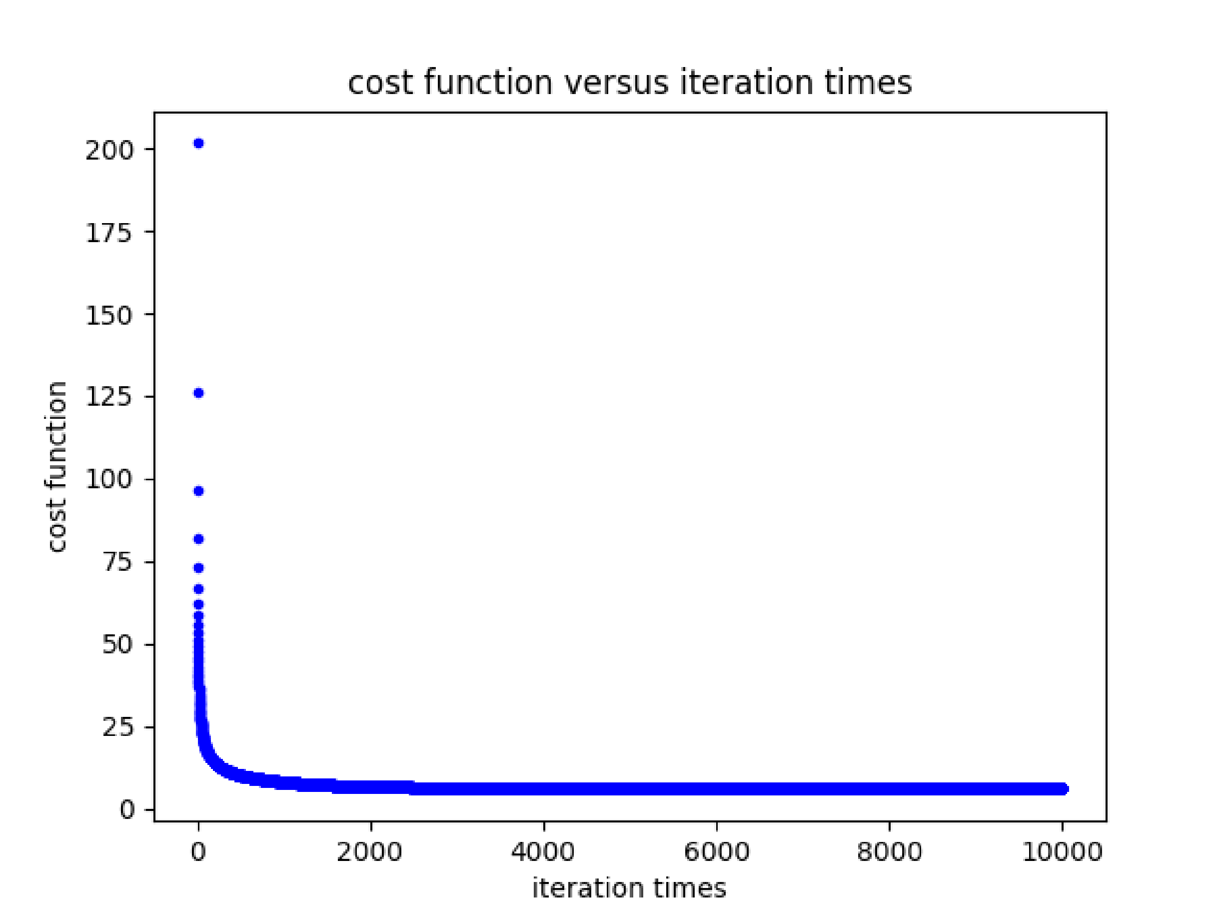
$$h_\theta(x) = \sum_{i=0}^{17} \theta_i x_i + \sum_{i=1}^{17} \sum_{j=1}^{i} \theta_n x_i x_j \text{ , } n \; in \; range(18,171)$$

$$\Rightarrow h_\theta(x) = \sum_{i=0}^{17} \theta_i x_i + \sum_{i=18}^{170} \theta_i x_i = \sum_{i=0}^{170} \theta_i x_i = [\theta_0 \quad \theta_1 \quad \ldots \quad \theta_{169} \quad \theta_{170}]_{1X171} \cdot \begin{bmatrix} x_0^1 & x_0^2 & \ldots & x_0^{N-1} & x_0^N \\ x_1^1 & x_1^2 & \ldots & x_1^{N-1} & x_1^N \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ x_{169}^1 & x_{169}^2 & \ldots & x_{16}^{N-1} & x_{16}^N \\ x_{170}^1 & x_{170}^2 & \ldots & x_{17}^{N-1} & x_{17}^N \end{bmatrix}_{171XN}$$

$$= [y^1 \quad y^2 \quad \ldots \quad y^{N-1} \quad y^N]_{1XN}$$

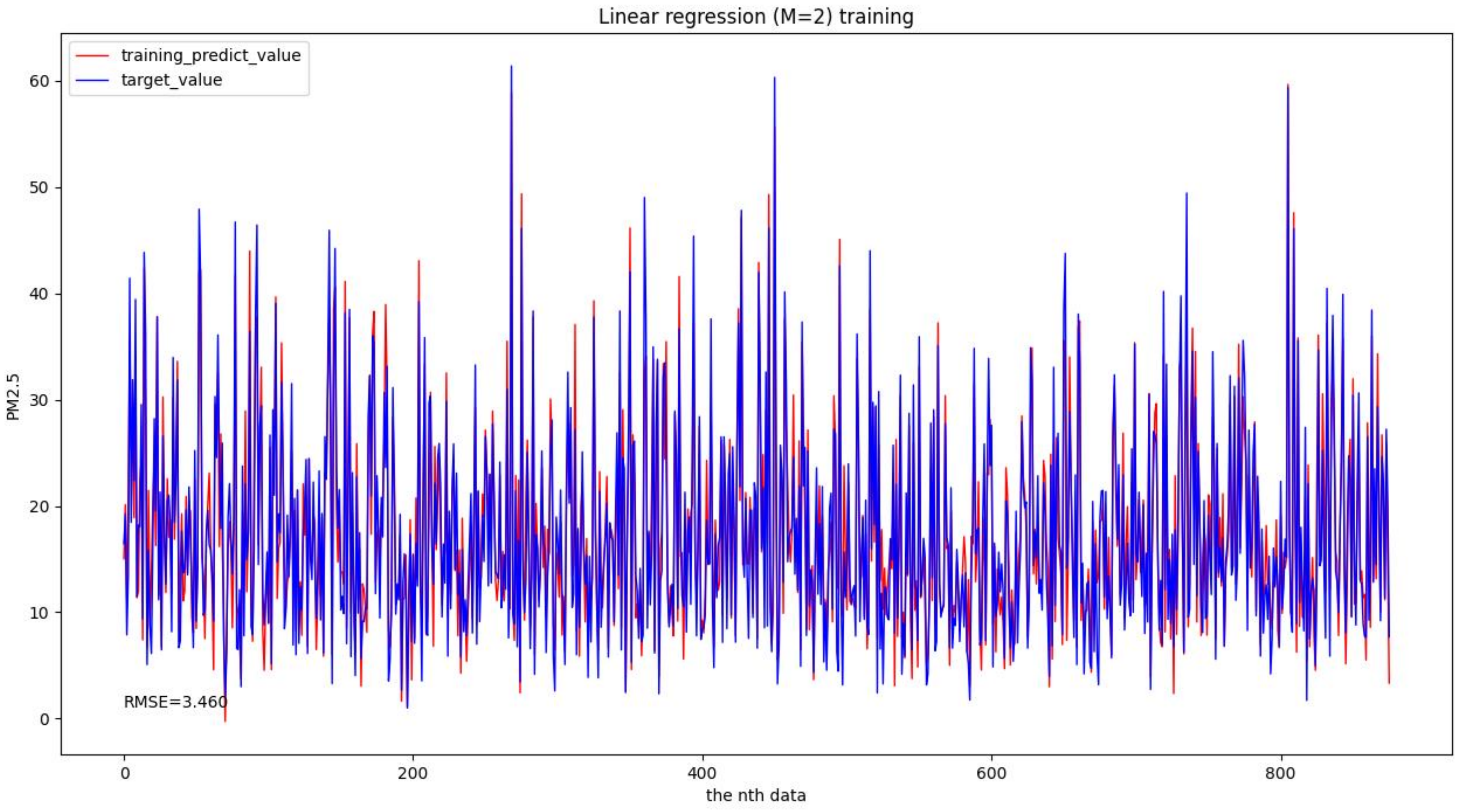
Transpose from the original data

Result:

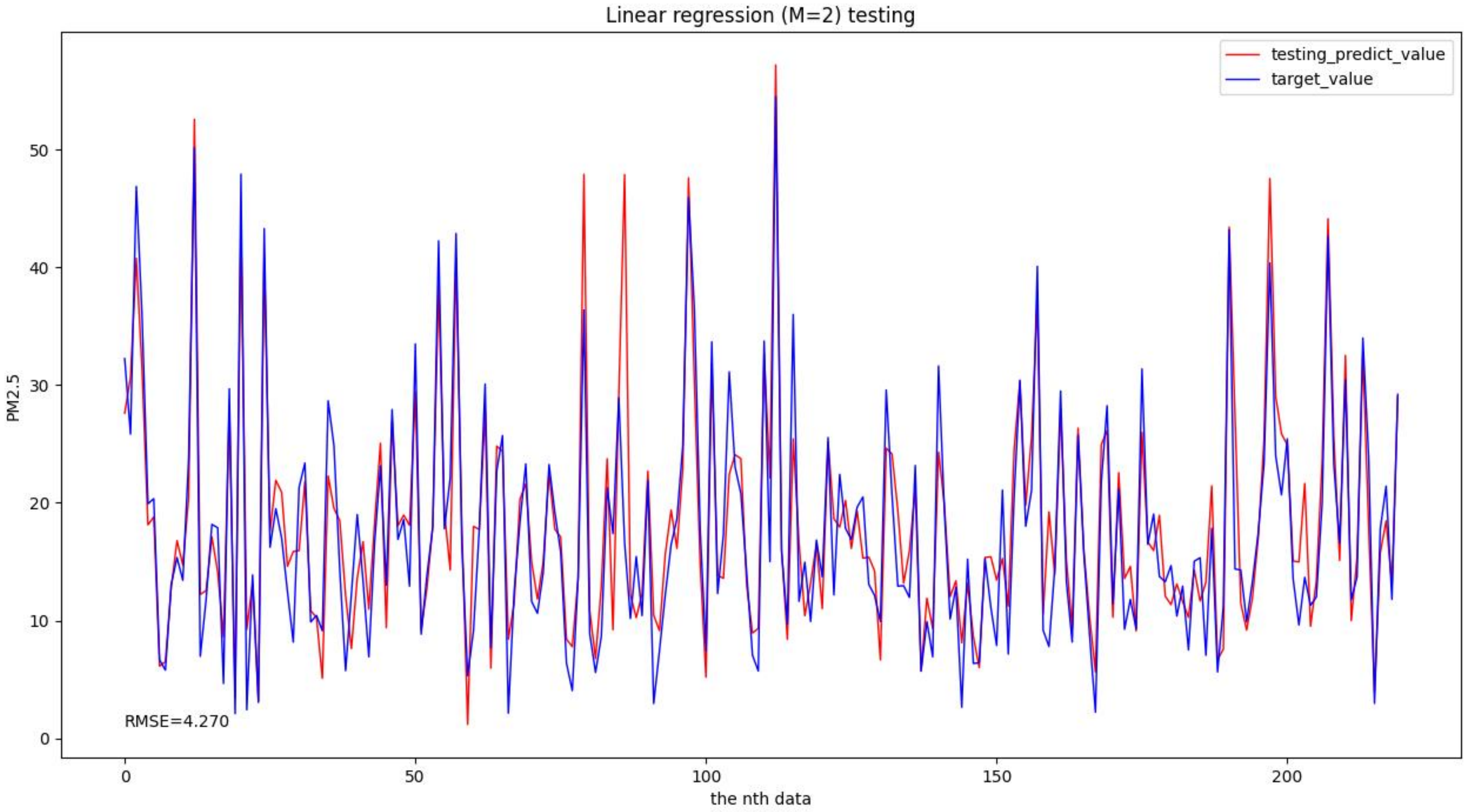


cost function versus iteration times
cost function
iteration times
0
25
50
75
100
125
150
175
200
0
2000
4000
6000
8000
10000

Result:


Linear regression (M=2) training
training_predict_value
target_value
PM2.5
60
50
40
30
20
10
0
RMSE=3.460
0
200
400
600
800
the nth data

Result:



Linear regression (M=2) testing
testing_predict_value
target_value
PM2.5
0
10
20
30
40
50
0
50
100
150
200
the nth data
RMSE=4.270

Model evaluation

RMSE	training	testing
M=1 (gradient descent)	1. 4.116	1. 4.160
	2. 4.093	2. 4.258
	3. 4.190	3. 3.879
	4. 4.066	4. 4.345
	5. 4.093	5. 4.245
	Avg: 4.111	Avg: 4.177
M=2 (gradient descent)	1. 3.361	1. 4.128
	2. 3.365	2. 4.242
	3. 3.414	3. 5.051
	4. 3.351	4. 4.183
	5. 3.408	5. 4.864
	Avg: 3.379	Avg: 4.493

由以上梯度下降實作中可以看到當m=1的時候，training的rmse與testing的rmse差距其實蠻小的，但是當m=2的時候，可以觀察到training的rmse降到很小，表示比起m=1找到更適合的參數去貼合模型，但是當到了testing的時候，可以發現rmse比起training的時候增加了許多，甚至超過了m=1的testing的rmse，因此可以推斷出這邊發生了過擬合的現象，因此面對陌生的資料時會較敏感。

PM2.5--MLE

data

Normalization

Data Transform

Sigmoidal: $S(x) = \frac{e^x}{1 + e^x}$

Gaussian: $f(x, \mu, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} e^{\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)}$

Shuffle

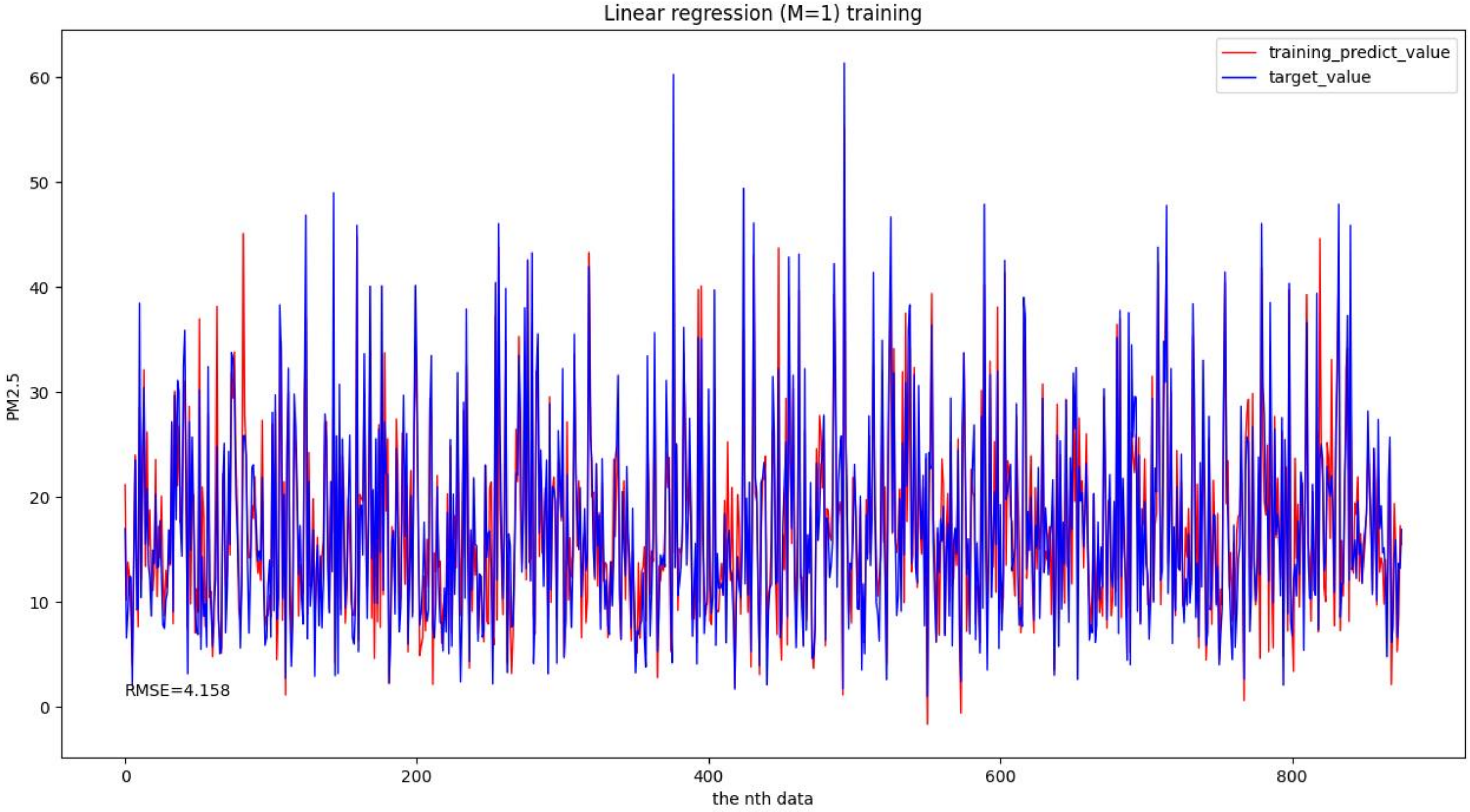
Train test split

Linear regression

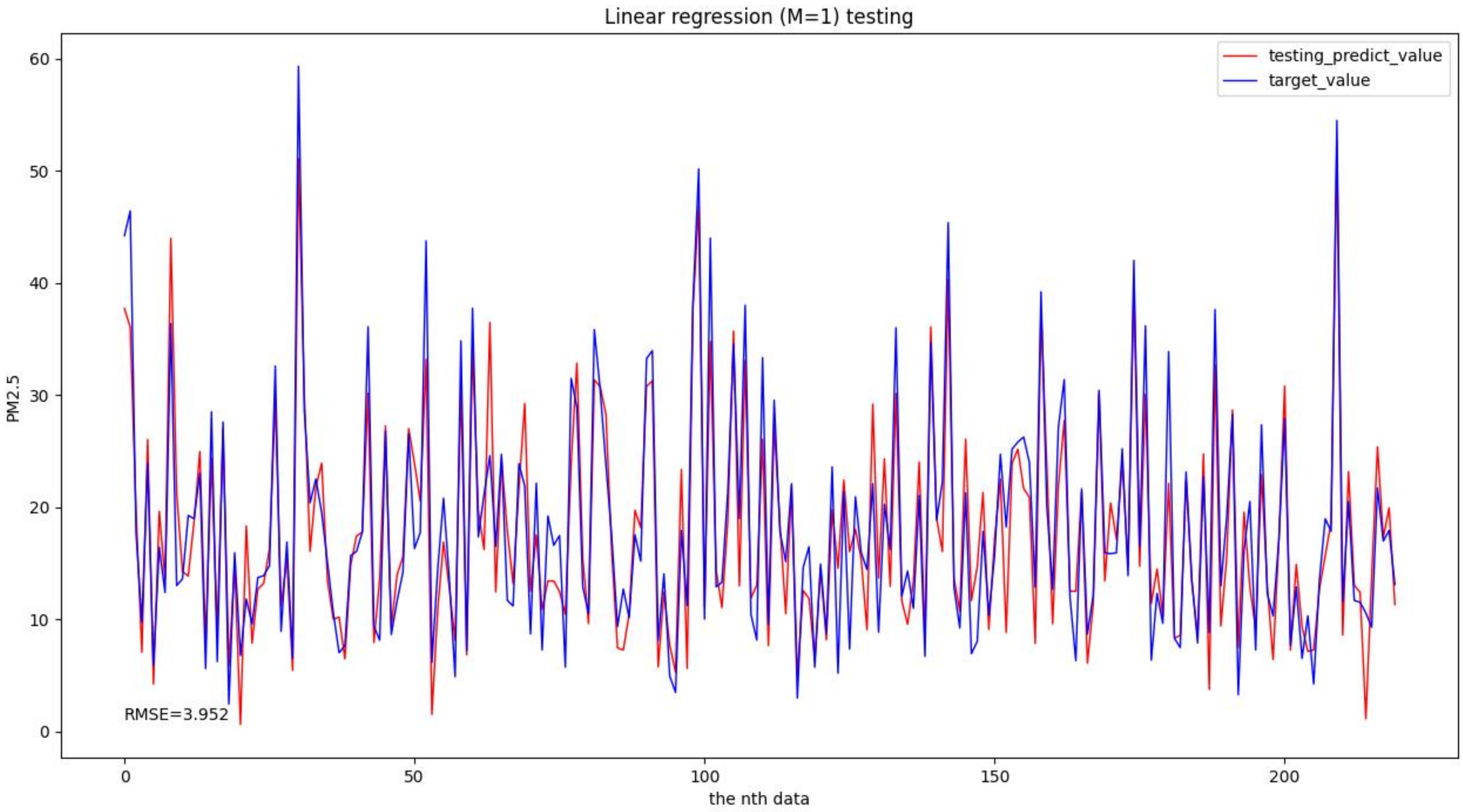
Linear regression: $w = (X^T X)^{-1} X^T Y$

Polynomial (M=1)-MLE

Result:

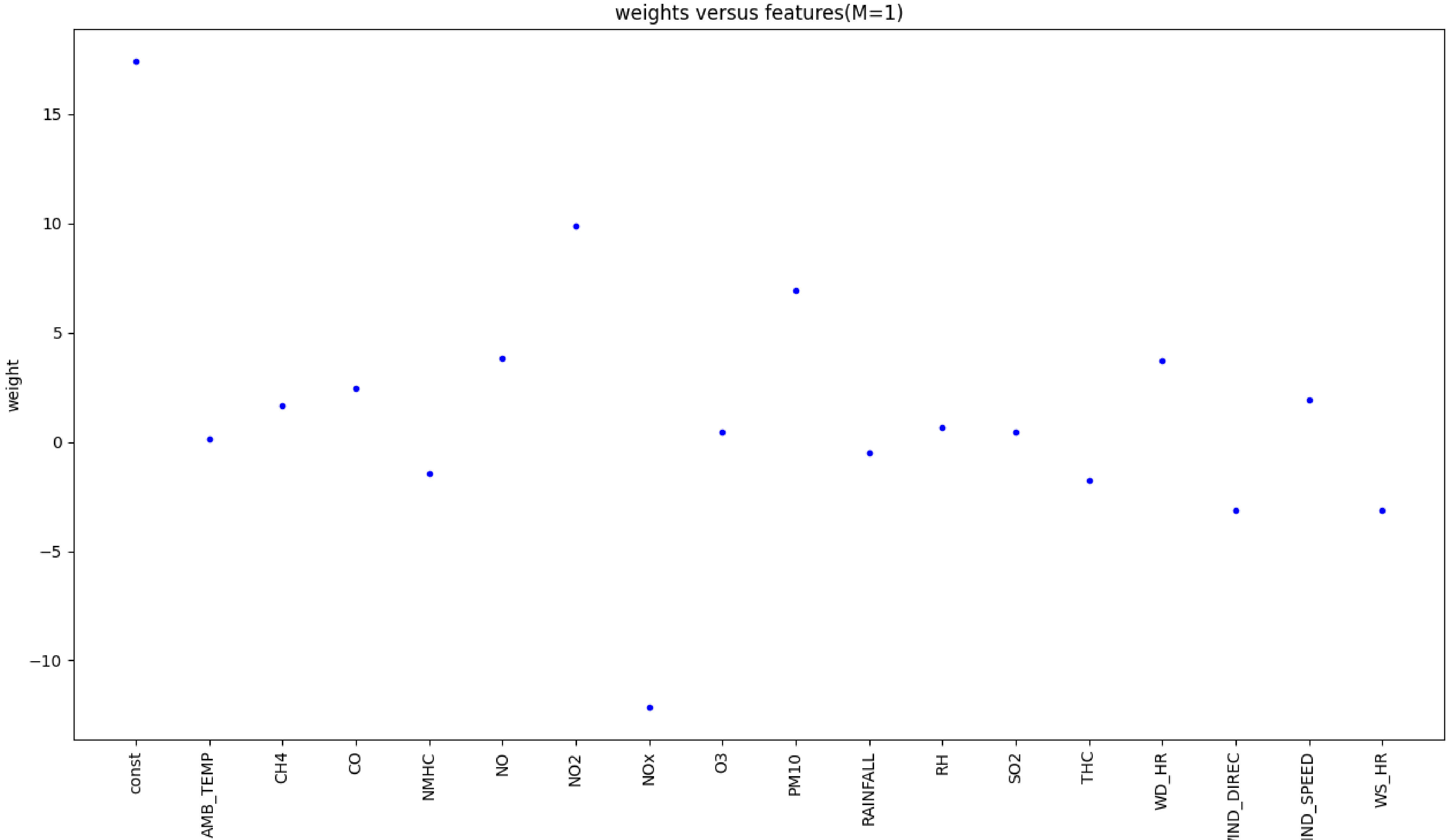

Linear regression (M=1) training
training_predict_value
target_value
PM2.5
the nth data
RMSE=4.158
0
10
20
30
40
50
60
0
200
400
600
800

Result:

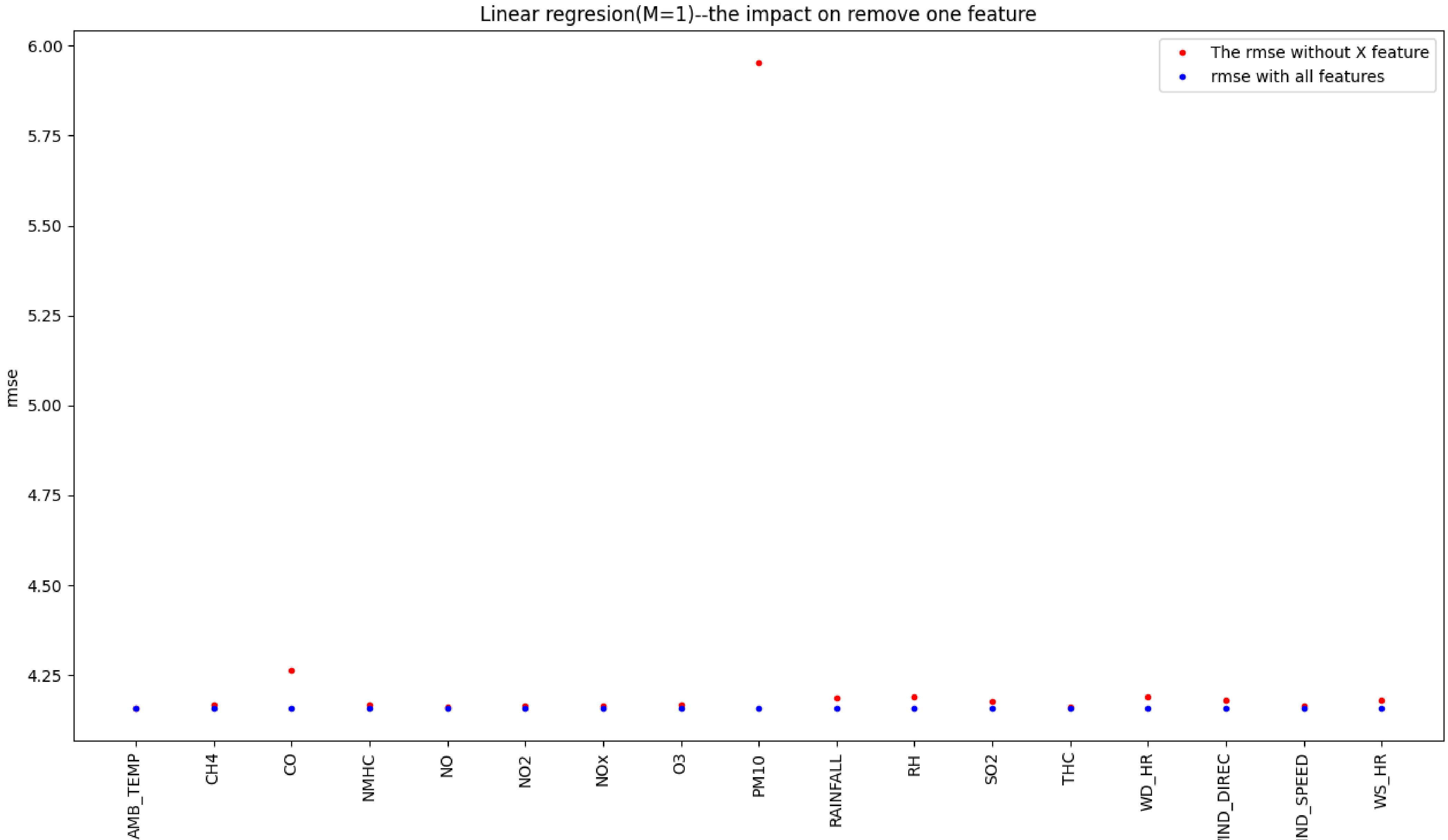


Linear regression (M=1) testing
testing_predict_value
target_value
PM2.5
the nth data
RMSE=3.952
0
10
20
30
40
50
60
0
50
100
150
200

Result:

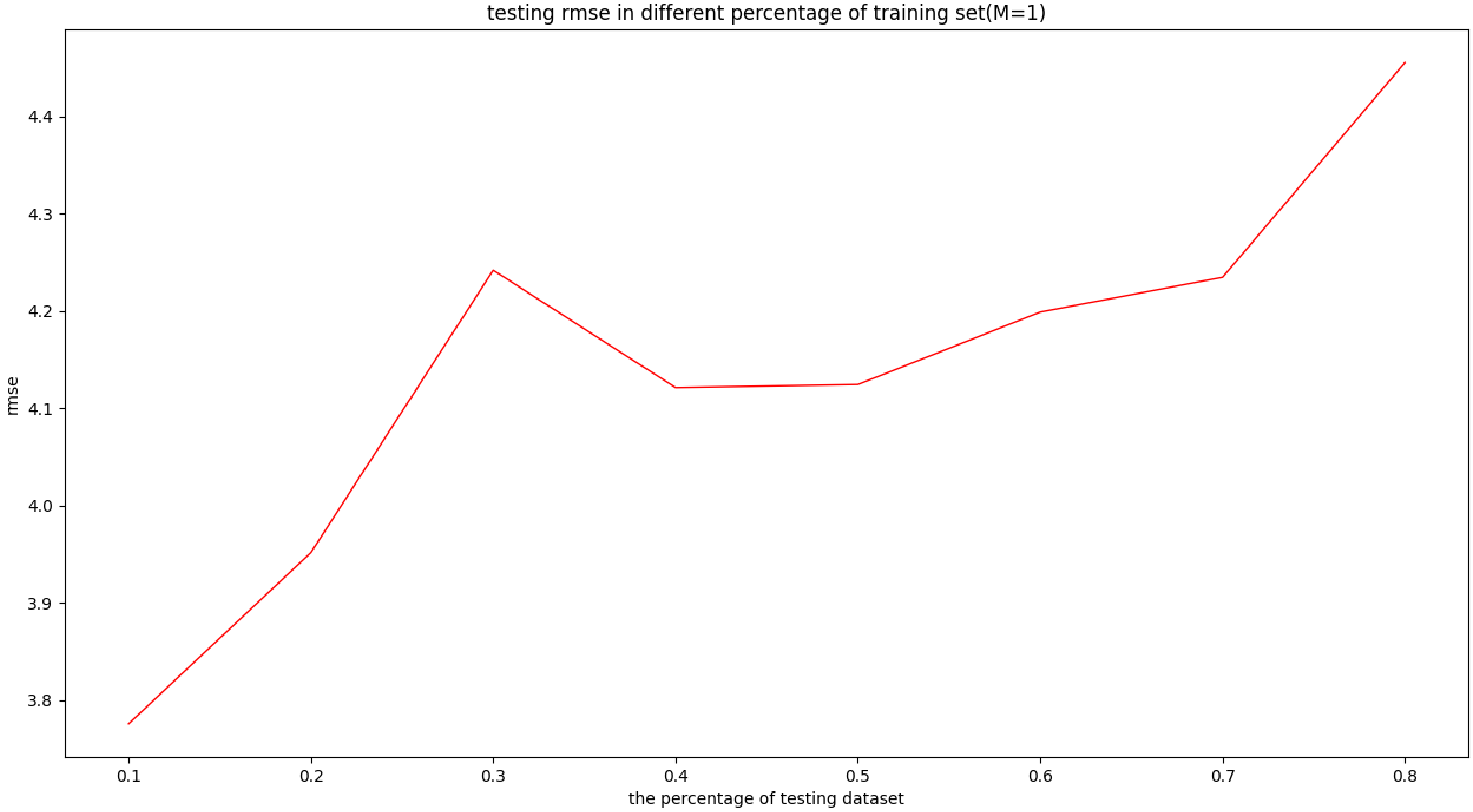

weights versus features(M=1)
weight
15
10
5
0
−5
−10
const
AMB_TEMP
CH4
CO
NMHC
NO
NO2
NOx
O3
PM10
RAINFALL
RH
SO2
THC
WD_HR
WS_HR

Result:

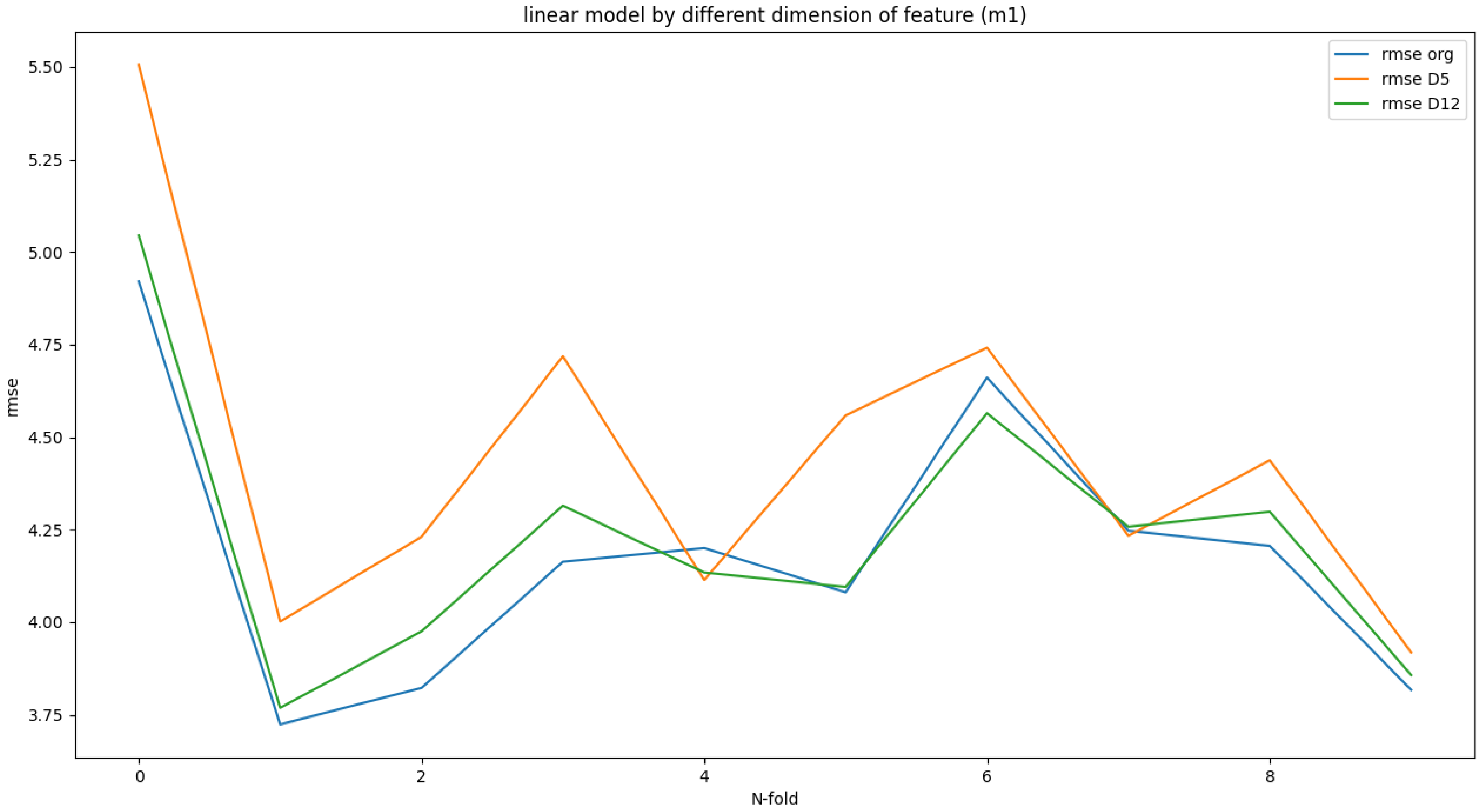


Linear regresion(M=1)--the impact on remove one feature
The rmse without X feature
rmse with all features
rmse
6.00
5.75
5.50
5.25
5.00
4.75
4.50
4.25
AMB_TEMP
CH4
CO
NMHC
NO
NO2
NOx
O3
PM10
RAINFALL
RH
SO2
THC
WD_HR
WS_HR

Result:


testing rmse in different percentage of training set(M=1)
rmse
4.4
4.3
4.2
4.1
4.0
3.9
3.8
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
the percentage of testing dataset

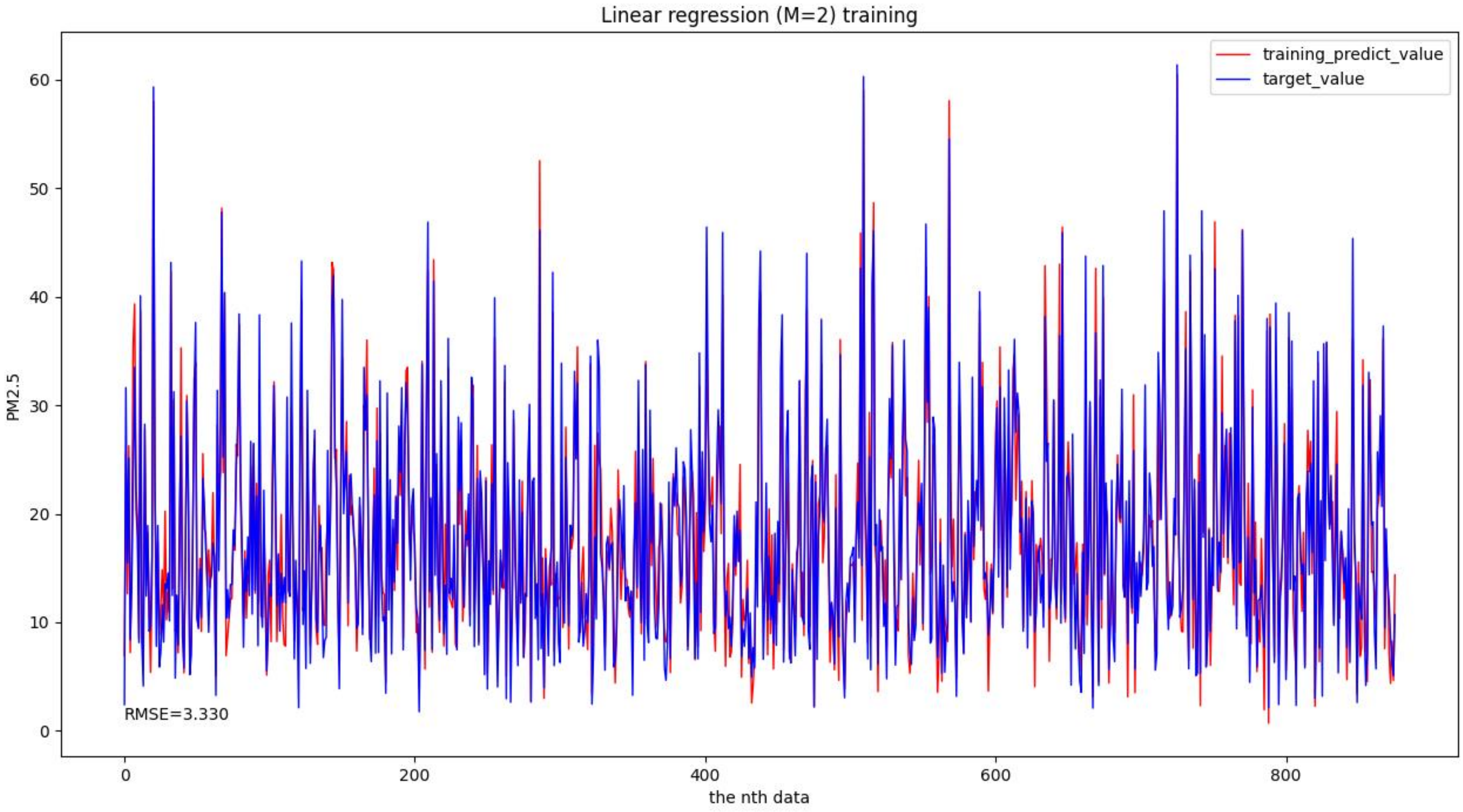
Result:



linear model by different dimension of feature (m1)
rmse org
rmse D5
rmse D12
5.50
5.25
5.00
4.75
4.50
4.25
4.00
3.75
rmse
0
2
4
6
8
N-fold

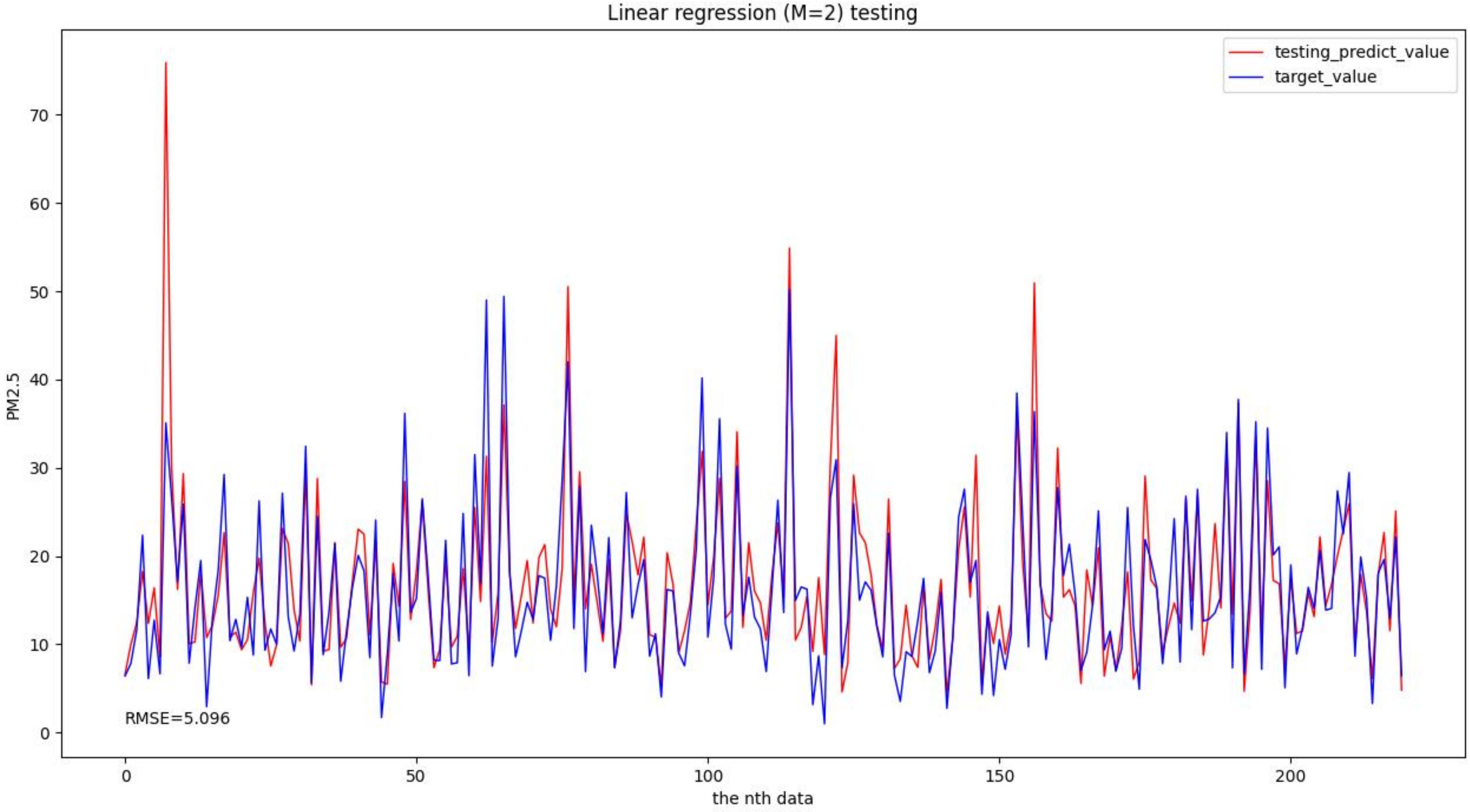
Polynomial (M=2)-MLE

Result:



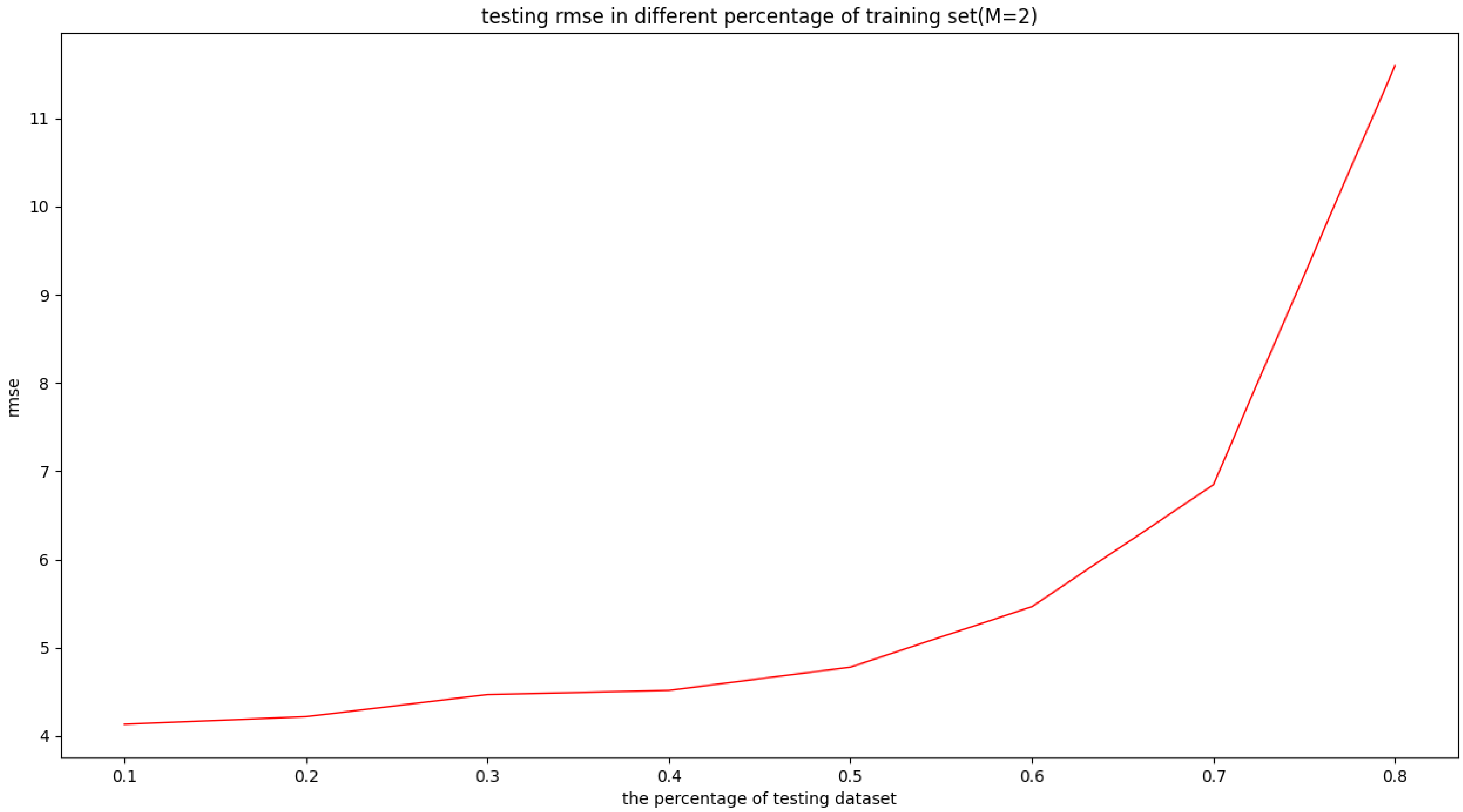
Linear regression (M=2) training
training_predict_value
target_value
PM2.5
60
50
40
30
20
10
0
0
200
400
600
800
the nth data
RMSE=3.330

Result:



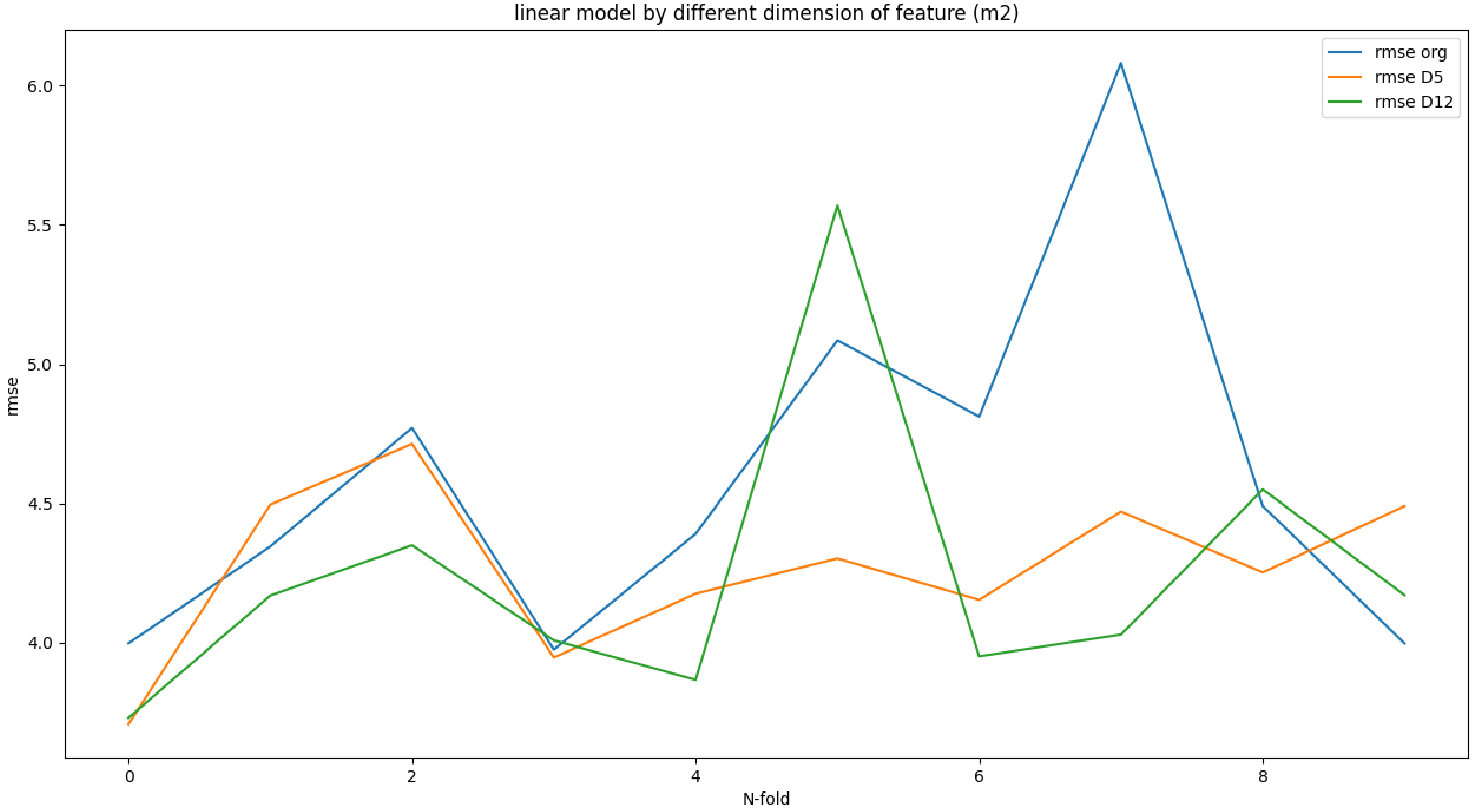
Linear regression (M=2) testing
testing_predict_value
target_value
PM2.5
70
60
50
40
30
20
10
0
RMSE=5.096
0
50
100
150
200
the nth data

Result:



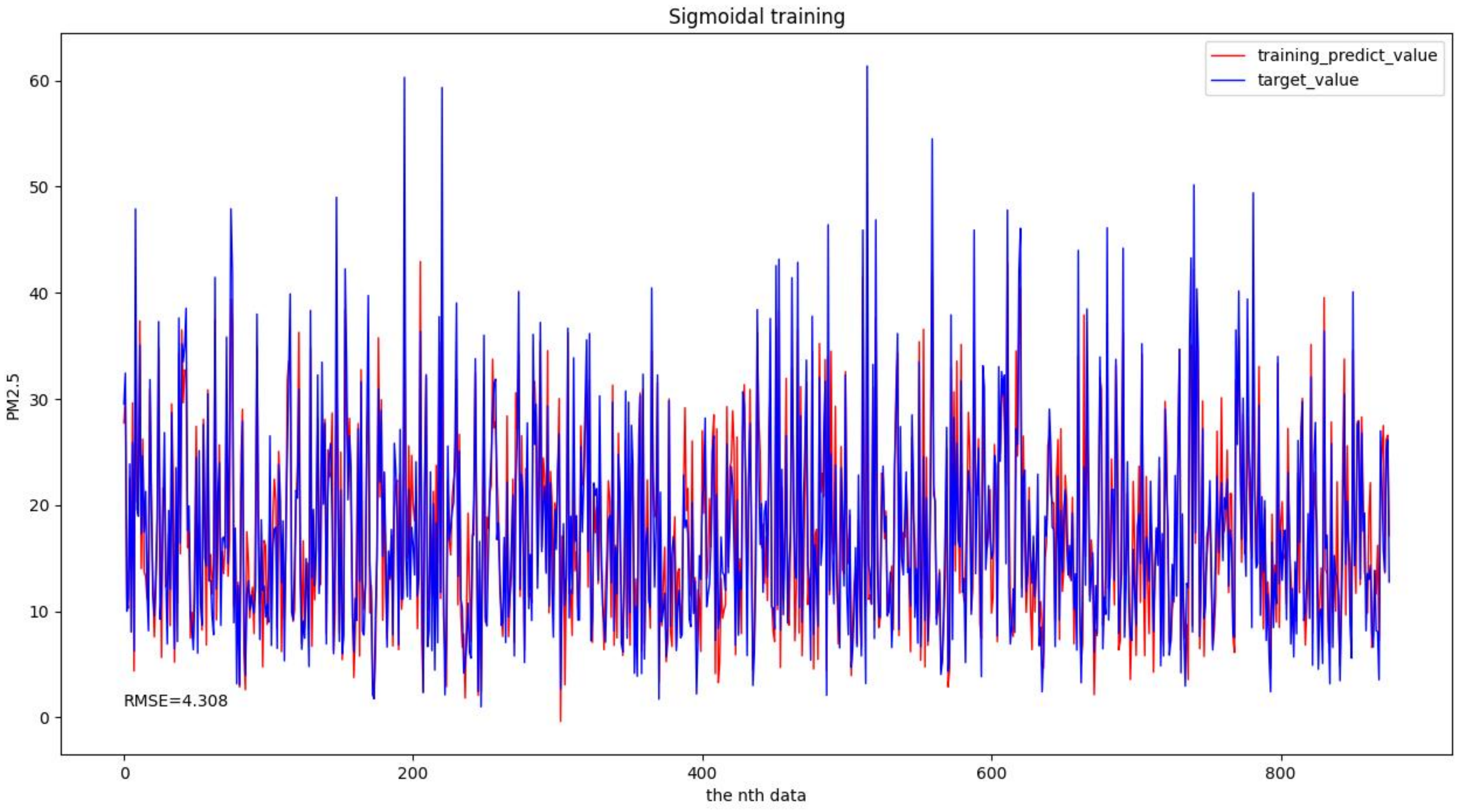
testing rmse in different percentage of training set(M=2)
rmse
4
5
6
7
8
9
10
11
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
the percentage of testing dataset

Result:

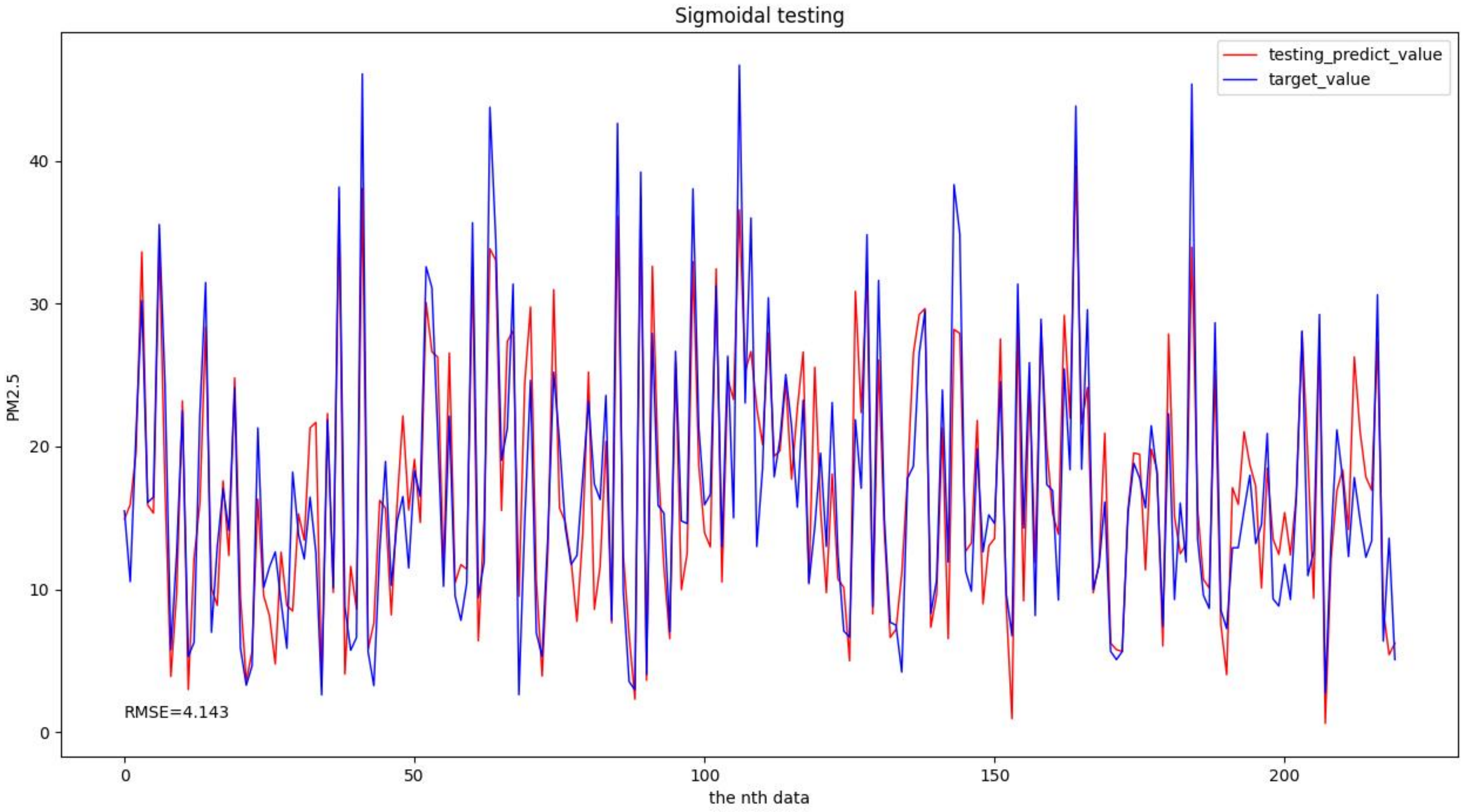

linear model by different dimension of feature (m2)
rmse org
rmse D5
rmse D12
6.0
5.5
5.0
4.5
4.0
rmse
0
2
4
6
8
N-fold

Sigmoidal $S(x) = \dfrac{e^x}{1 + e^x}$

Result:

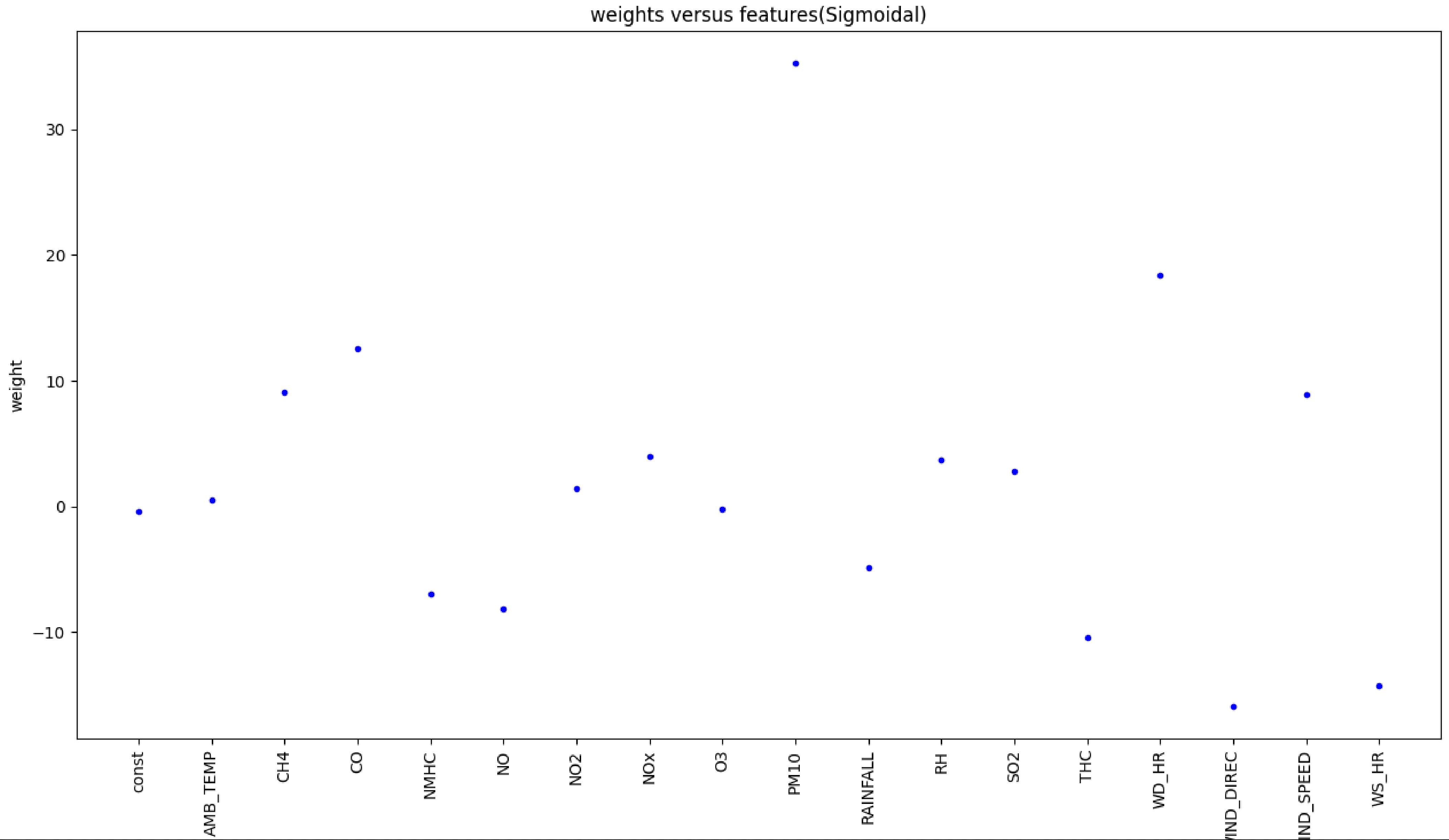

Sigmoidal training
training_predict_value
target_value
60
50
40
30
20
10
0
PM2.5
RMSE=4.308
0
200
400
600
800
the nth data

Result:

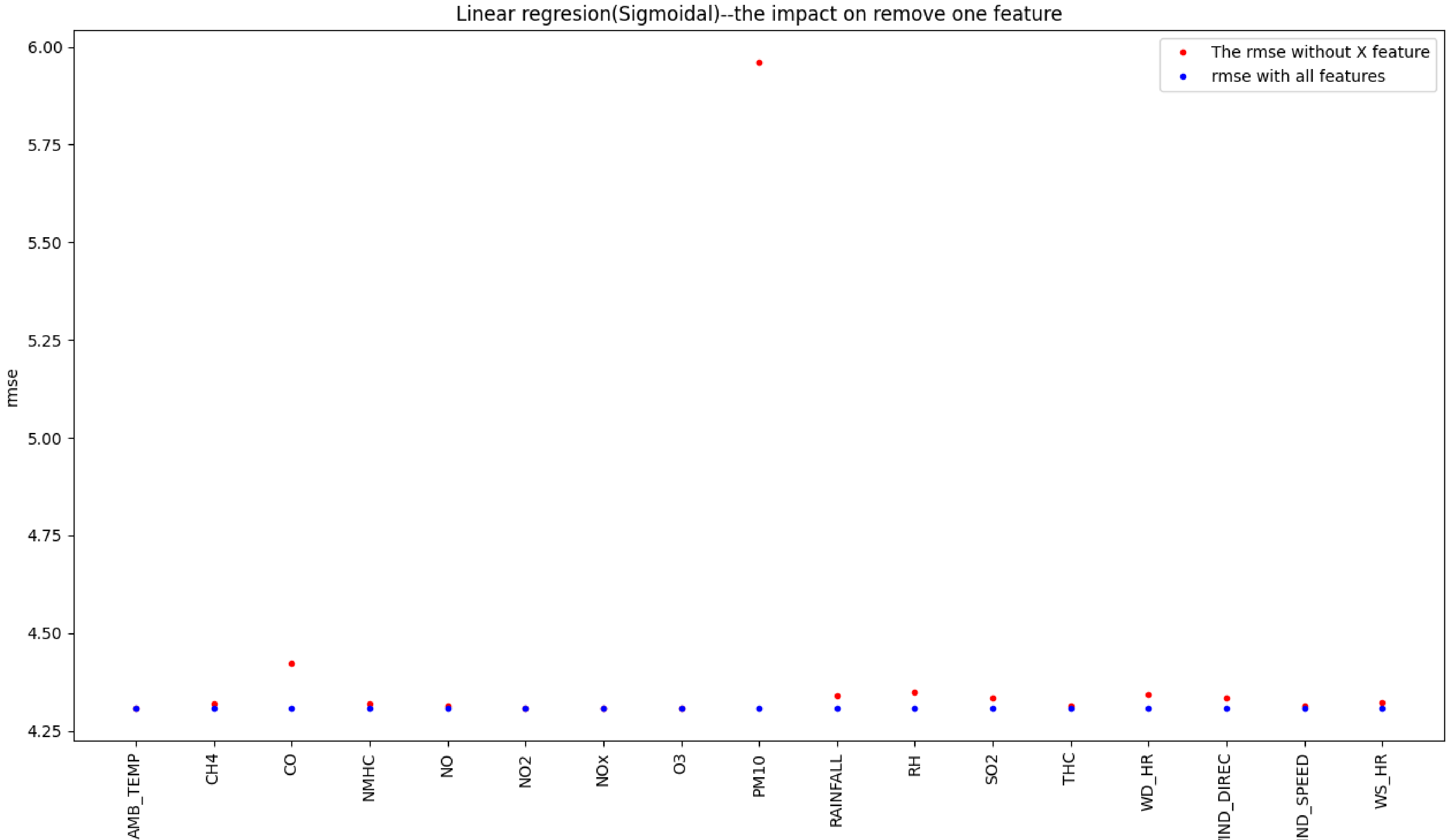


Sigmoidal testing
testing_predict_value
target_value
PM2.5
the nth data
RMSE=4.143
0
10
20
30
40
0
50
100
150
200

Result:

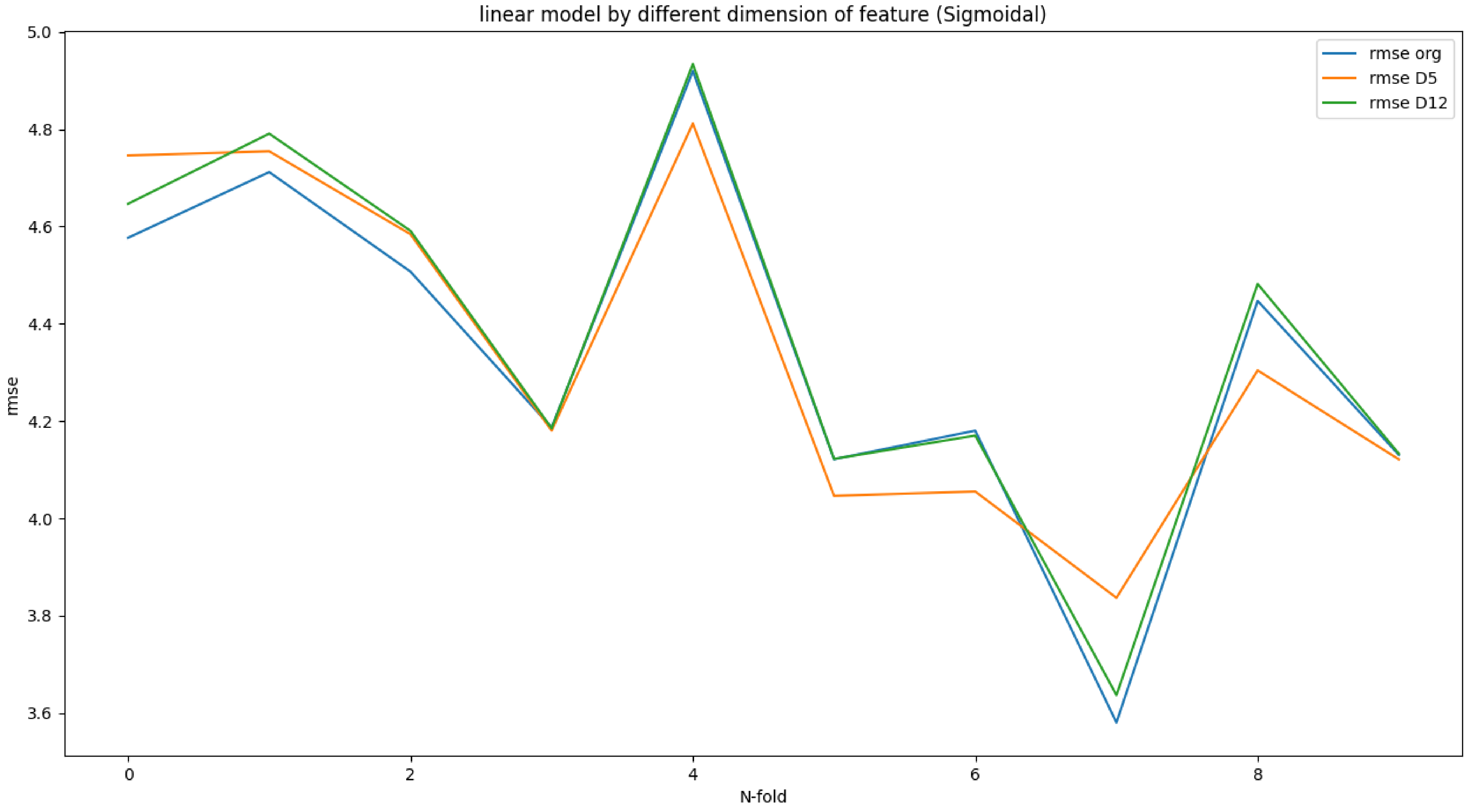

weights versus features(Sigmoidal)
weight
30
20
10
0
−10
const
AMB_TEMP
CH4
CO
NMHC
NO
NO2
NOx
O3
PM10
RAINFALL
RH
SO2
THC
WD_HR
WS_HR

Result:



Linear regresion(Sigmoidal)--the impact on remove one feature
The rmse without X feature
rmse with all features
rmse
6.00
5.75
5.50
5.25
5.00
4.75
4.50
4.25
AMB_TEMP
CH4
CO
NMHC
NO
NO2
NOx
O3
PM10
RAINFALL
RH
SO2
THC
WD_HR
IND_DIREC
IND_SPEED
WS_HR

Result:

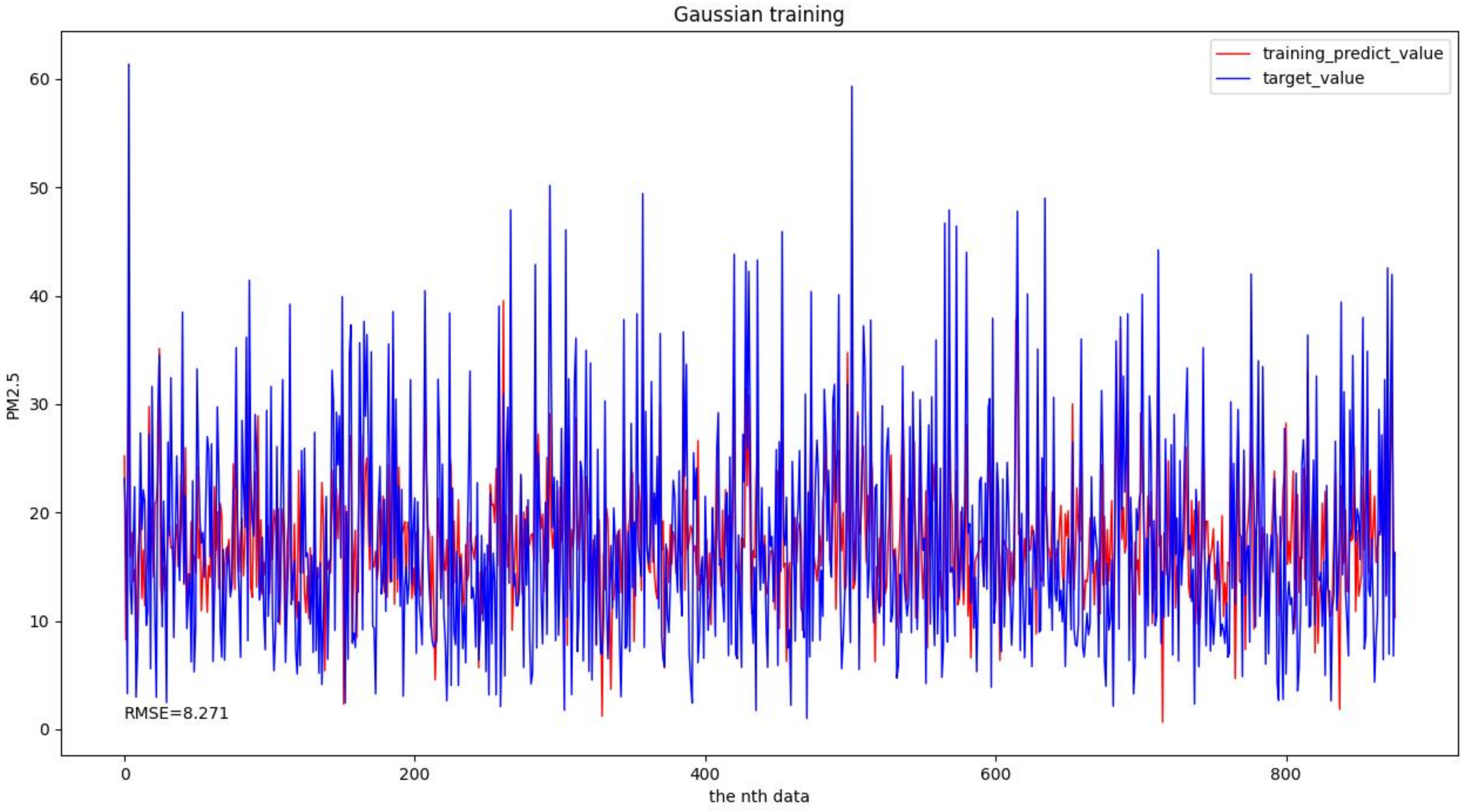


linear model by different dimension of feature (Sigmoidal)
rmse org
rmse D5
rmse D12
rmse
N-fold
5.0
4.8
4.6
4.4
4.2
4.0
3.8
3.6
0
2
4
6
8

Gaussian

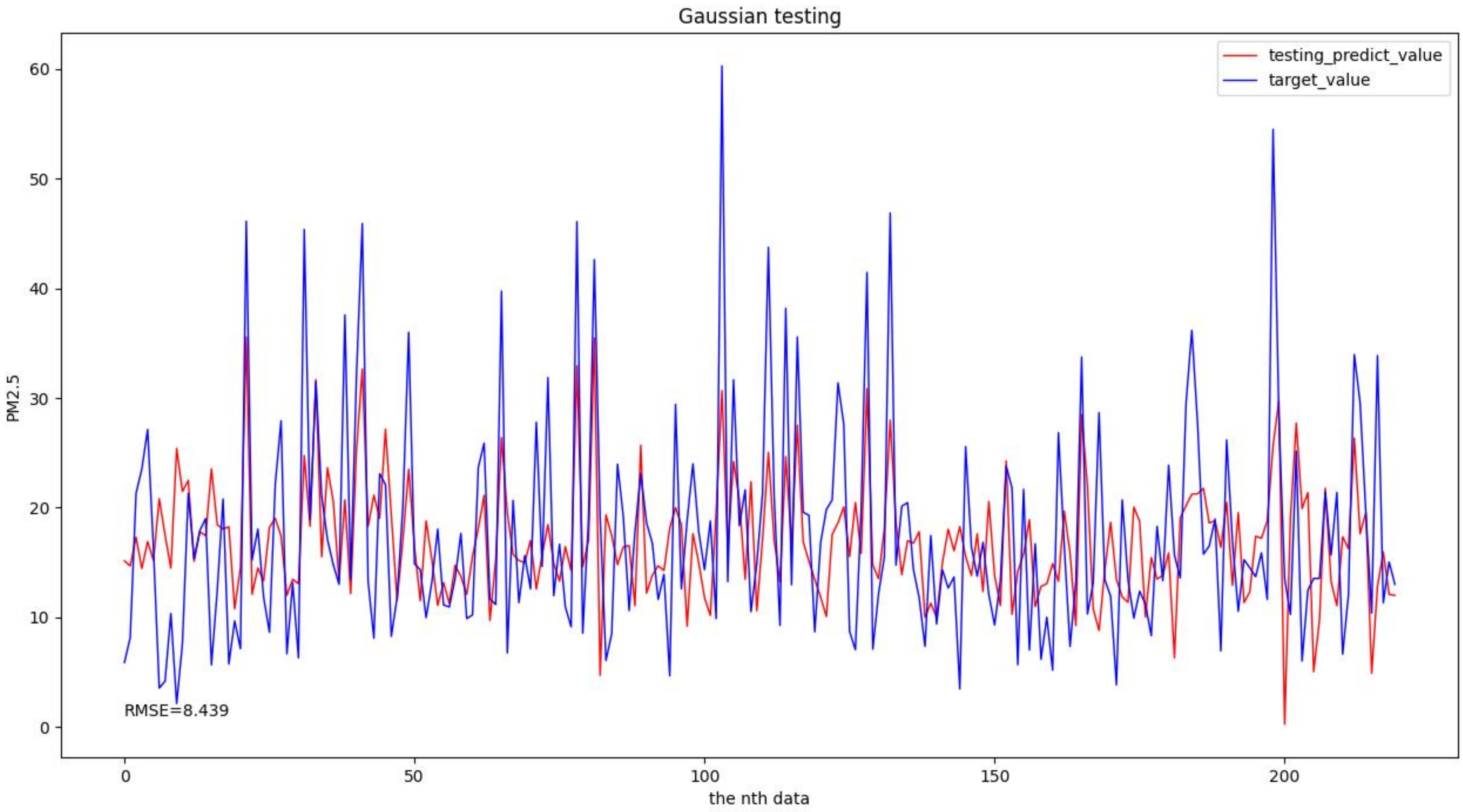
$$f(x, \mu, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} e^{\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)}$$

Result:



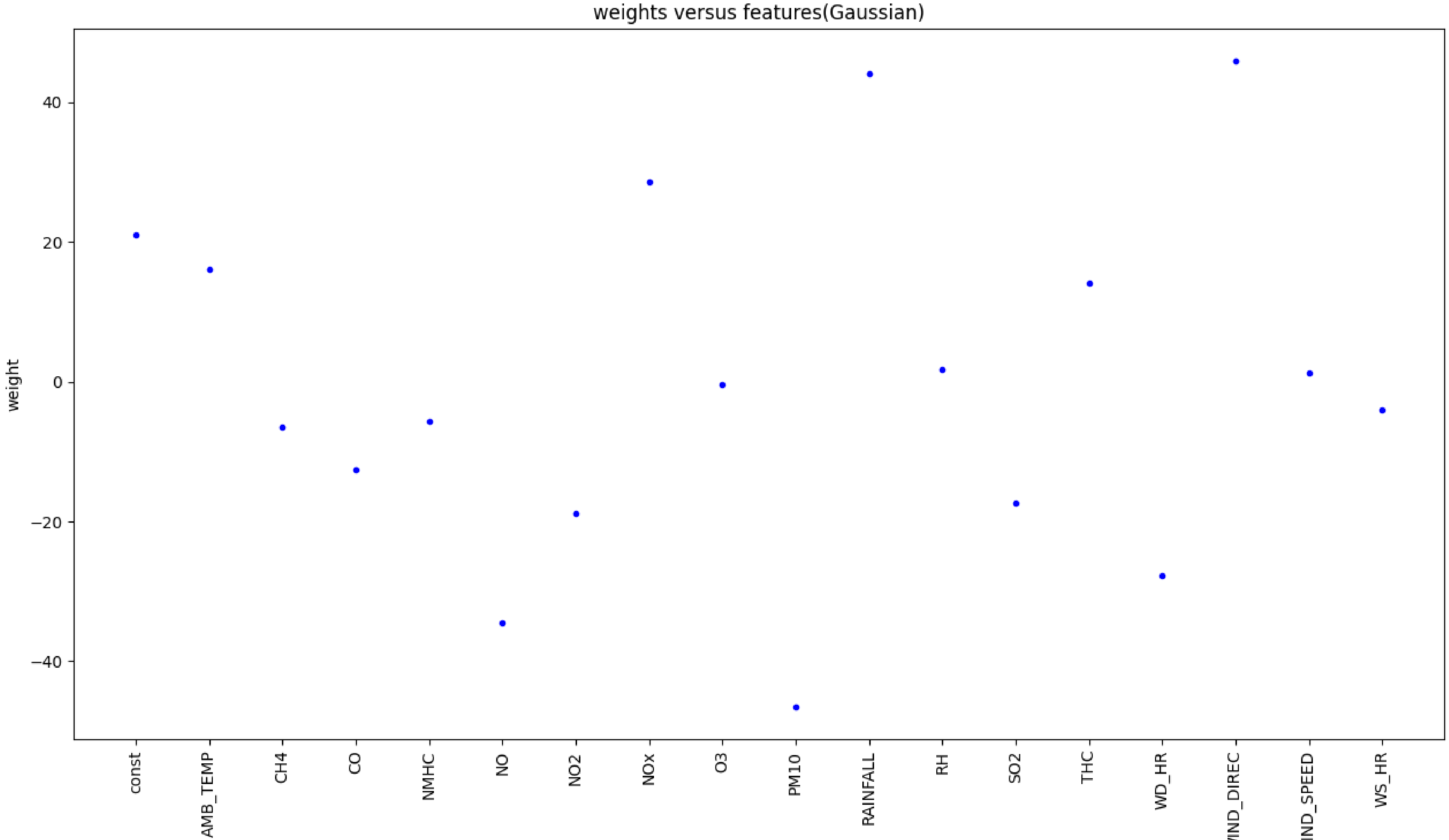
Gaussian training
training_predict_value
target_value
PM2.5
the nth data
RMSE=8.271
0
10
20
30
40
50
60
0
200
400
600
800

Result:



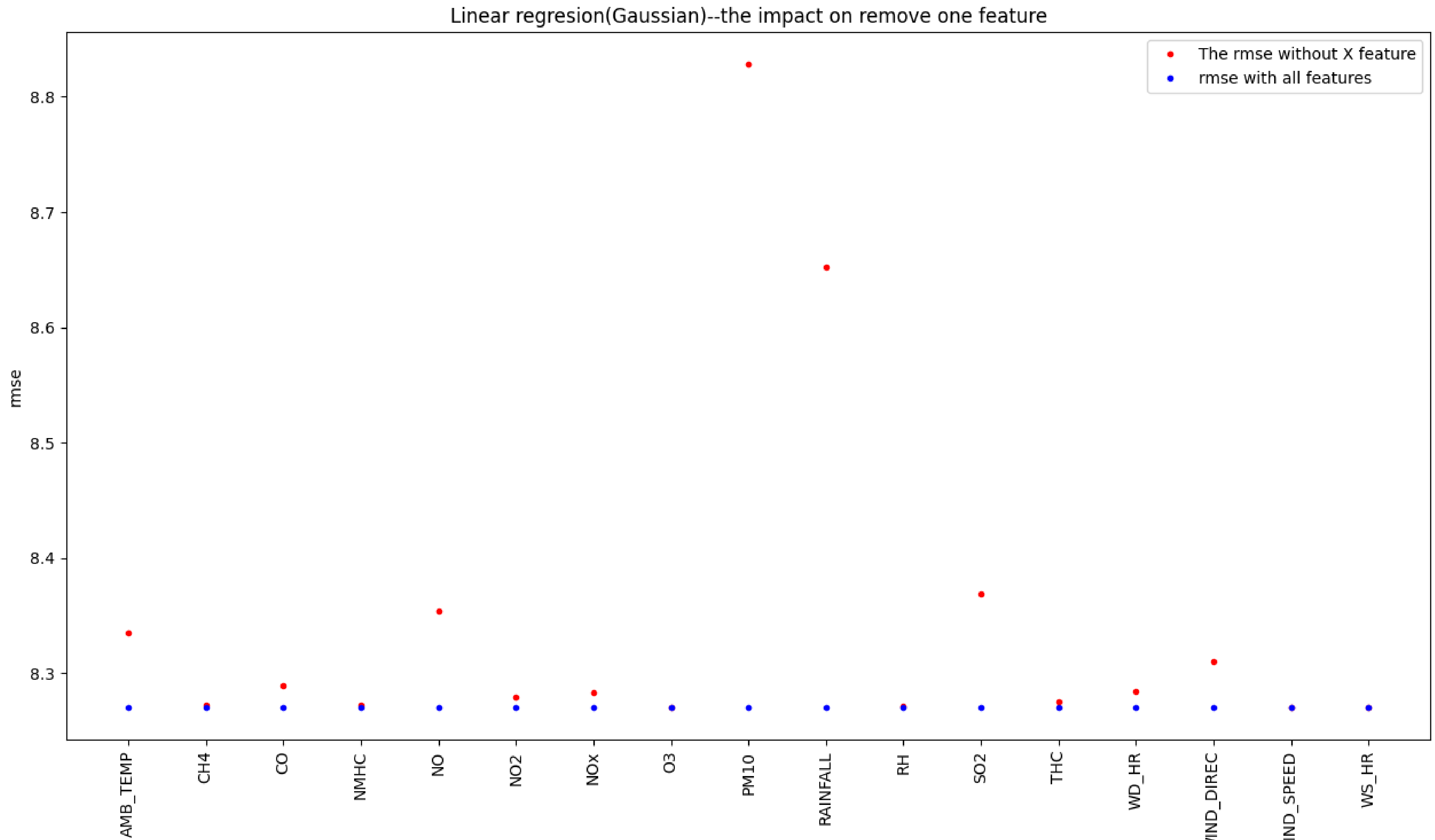
Gaussian testing
testing_predict_value
target_value
PM2.5
RMSE=8.439
the nth data
0
50
100
150
200
0
10
20
30
40
50
60

Result:

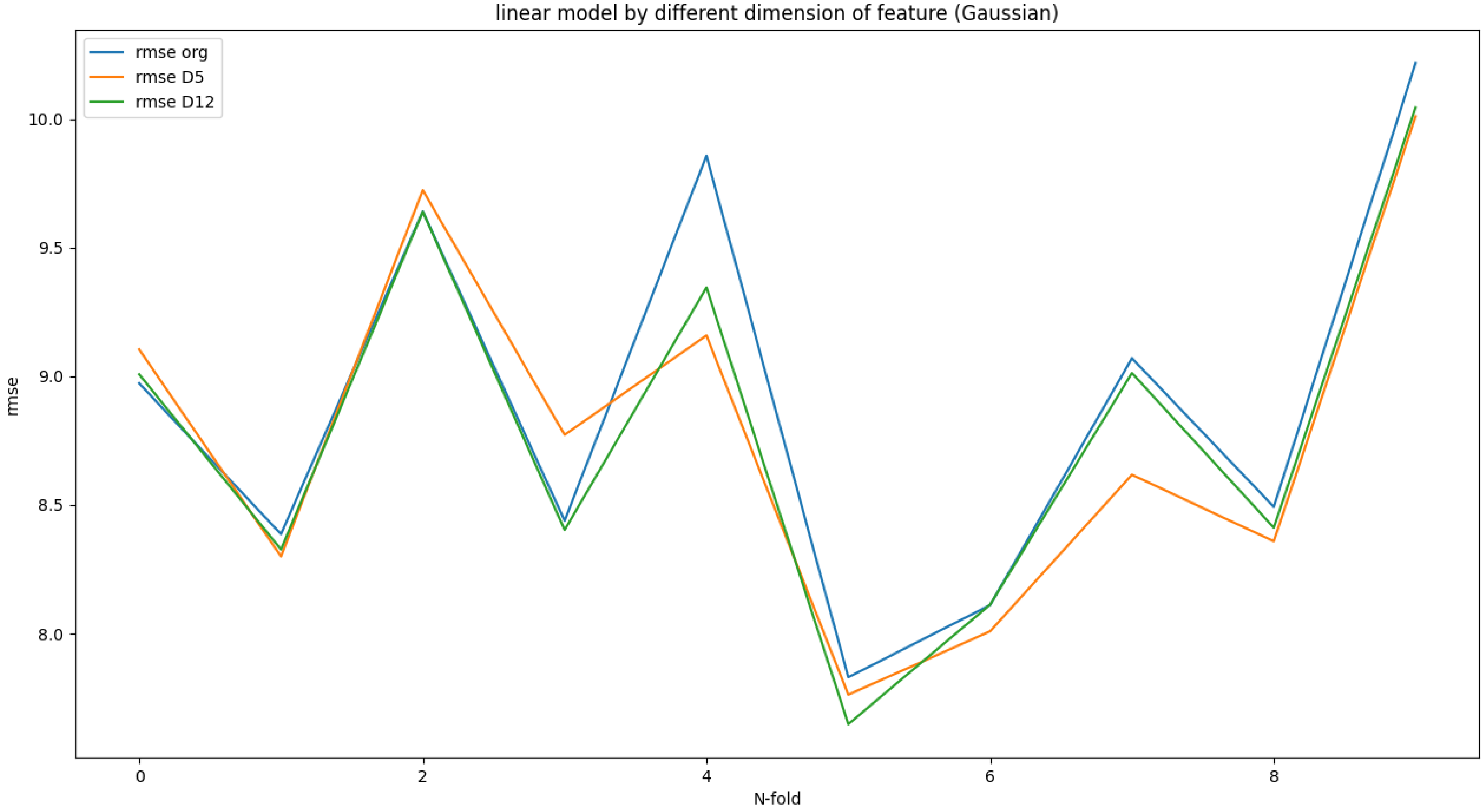


weights versus features(Gaussian)
weight
40
20
0
−20
−40
const
AMB_TEMP
CH4
CO
NMHC
NO
NO2
NOx
O3
PM10
RAINFALL
RH
SO2
THC
WD_HR
IND_DIREC
IND_SPEED
WS_HR

Result:


Linear regresion(Gaussian)--the impact on remove one feature
The rmse without X feature
rmse with all features
rmse
8.8
8.7
8.6
8.5
8.4
8.3
AMB_TEMP
CH4
CO
NMHC
NO
NO2
NOx
O3
PM10
RAINFALL
RH
SO2
THC
WD_HR
WIND_DIREC
WIND_SPEED
WS_HR

Result:



linear model by different dimension of feature (Gaussian)
rmse org
rmse D5
rmse D12
10.0
9.5
9.0
8.5
8.0
rmse
0
2
4
6
8
N-fold

MLE rmse testing comparison:

	Rmse
M=1	5.543
M=2	5.971
Sigmoidal	5.658
Gaussian	8.806

我們可以從這邊testing的rmse中發現Gaussian的模型是四個裡面表現最差的，可能的推論原因是因為Gaussian的basic function較為複雜，因此使其沒有很好的表現。

另外，在這邊我們也透過減少模型的特徵數來觀察是否可以解決過擬合的現象發生，因此我們選了三種模型，分別是不進行任何處理的original(D=17)與考慮影響最大的五個特徵(D=5)及去除影響最小的五個特徵(D=12)，來做比較，從上面的結果圖中可以發現:D=5的fold幾乎都在最下面，而D=12的fold則是與D=17沒有進行特徵篩選的模型很貼合，因此我們可以推斷，對於取D=5的這幾種特徵可以很好的去減少testing的rmse，透過做這個測試，我也了解到可以，不一定是要去使用所有的特徵才能去train出一個很好的model，如何去篩選所需要的特徵，對於model的效率及精準度都是一個很重要的事情。

Maximum a posterior (MAP)

data

Normalization

Shuffle

Function Transform

Sigmoidal: $S(x) = \frac{e^x}{1 + e^x}$

Gaussian: $f(x, \mu, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} e^{\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)}$

Train test split

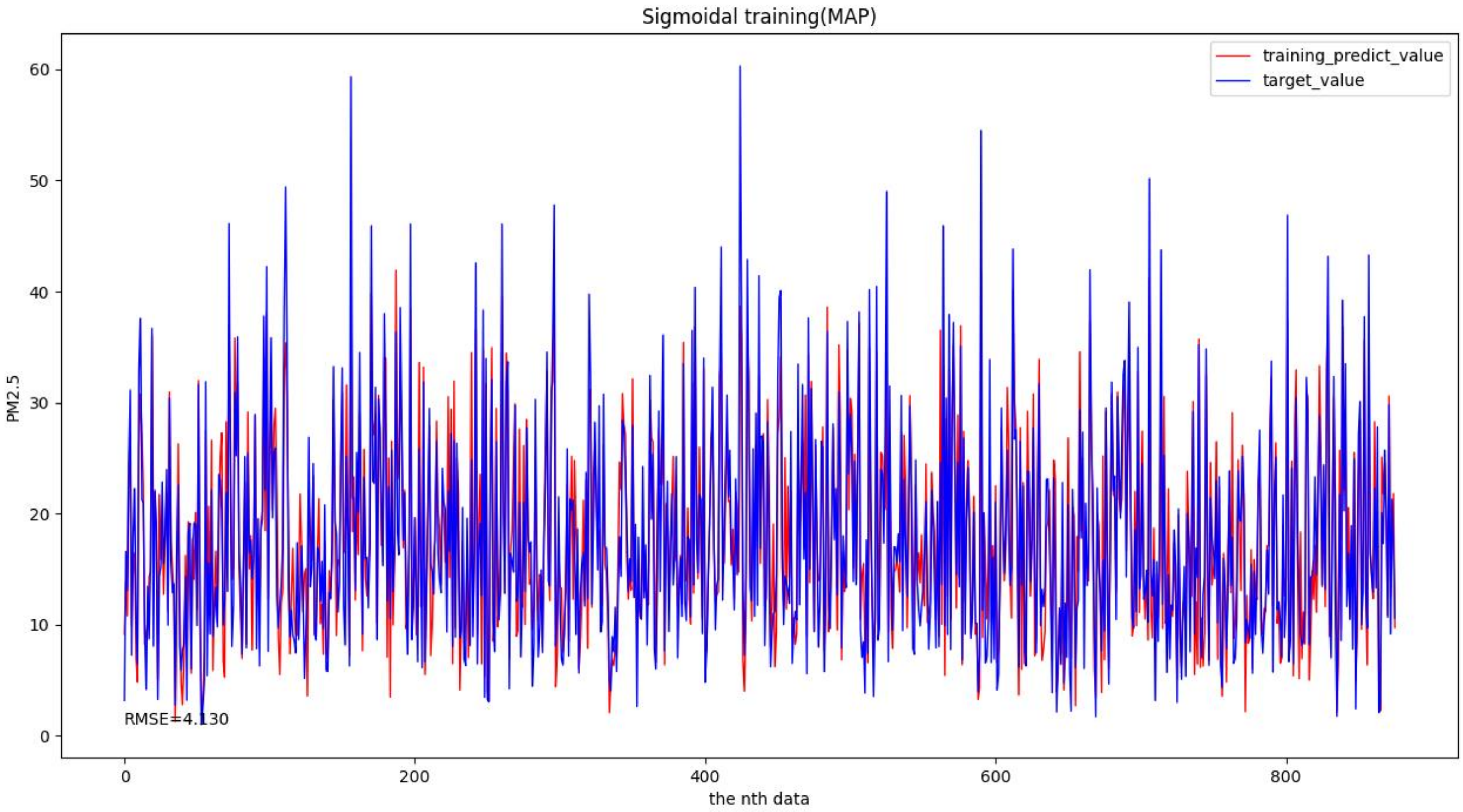
Linear regression(MAP)

How to update the weight: $\theta_{map} = (X^T X + \lambda I)^{-1} X^T y$

With λ=0.0001

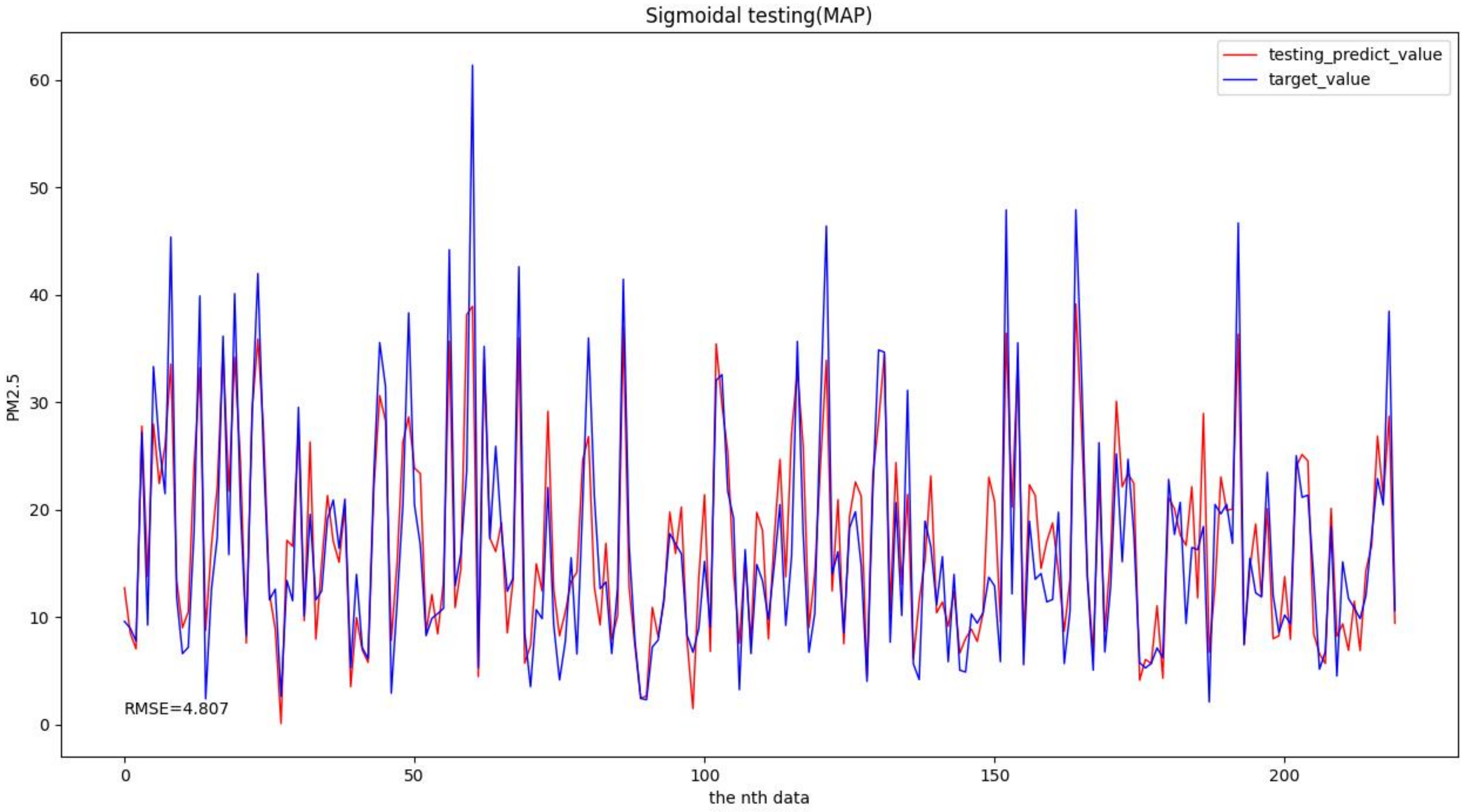
Sigmoidal $S(x) = \frac{e^x}{1 + e^x}$

Result:



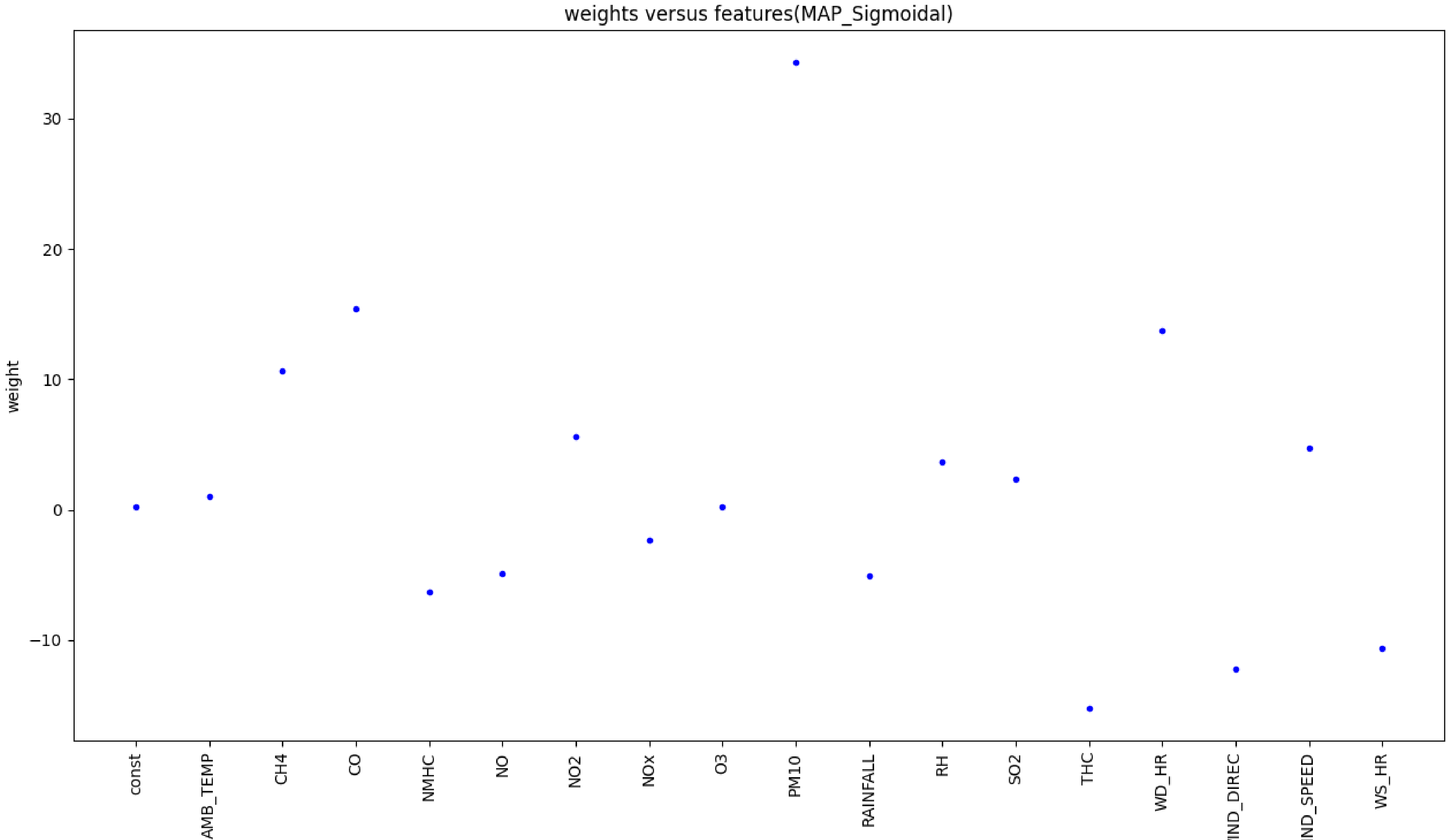
Sigmoidal training(MAP)
training_predict_value
target_value
PM2.5
60
50
40
30
20
10
0
RMSE=4.130
0
200
400
600
800
the nth data

Result:



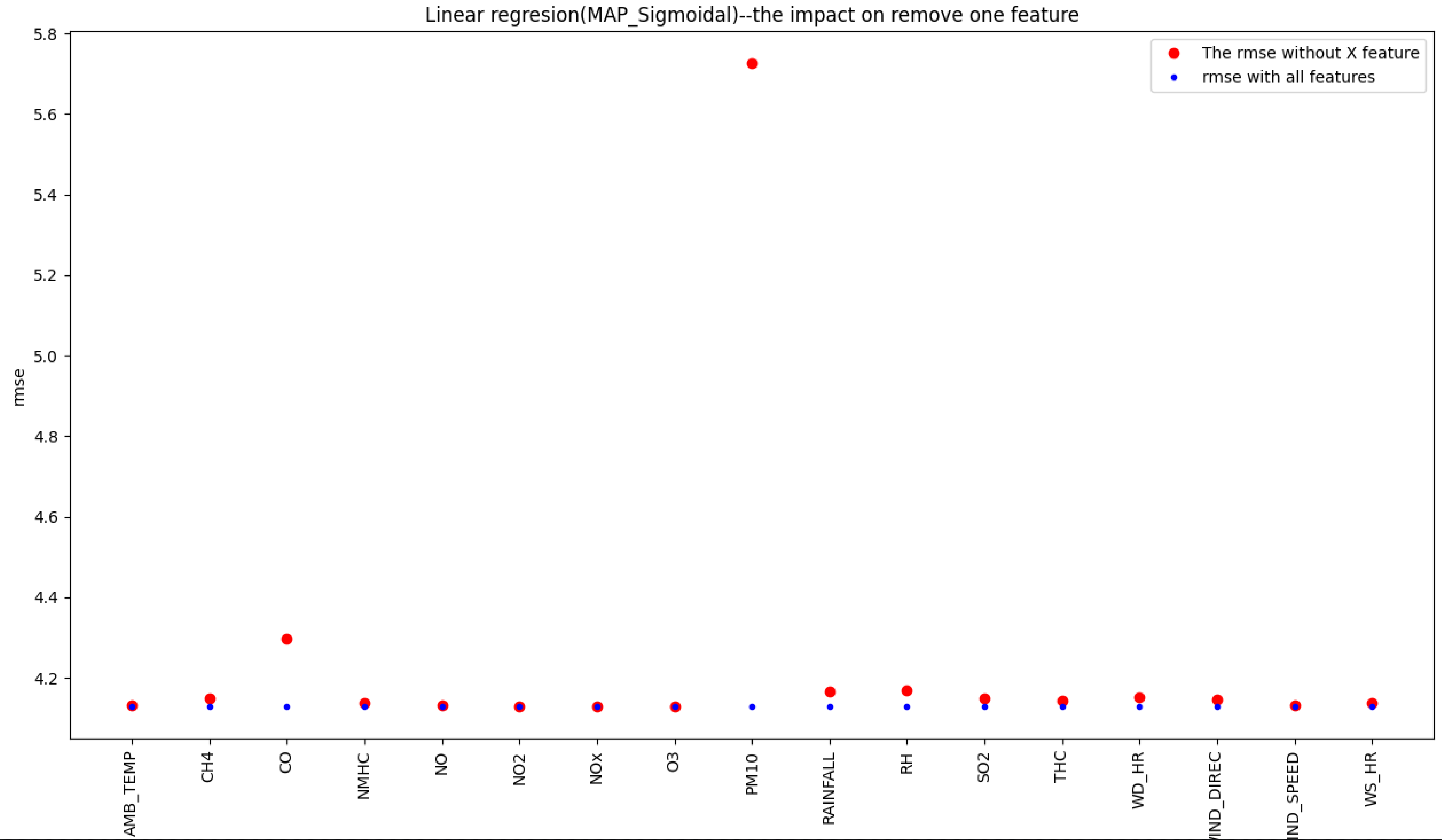
Sigmoidal testing(MAP)
testing_predict_value
target_value
PM2.5
0
10
20
30
40
50
60
RMSE=4.807
0
50
100
150
200
the nth data

Result:



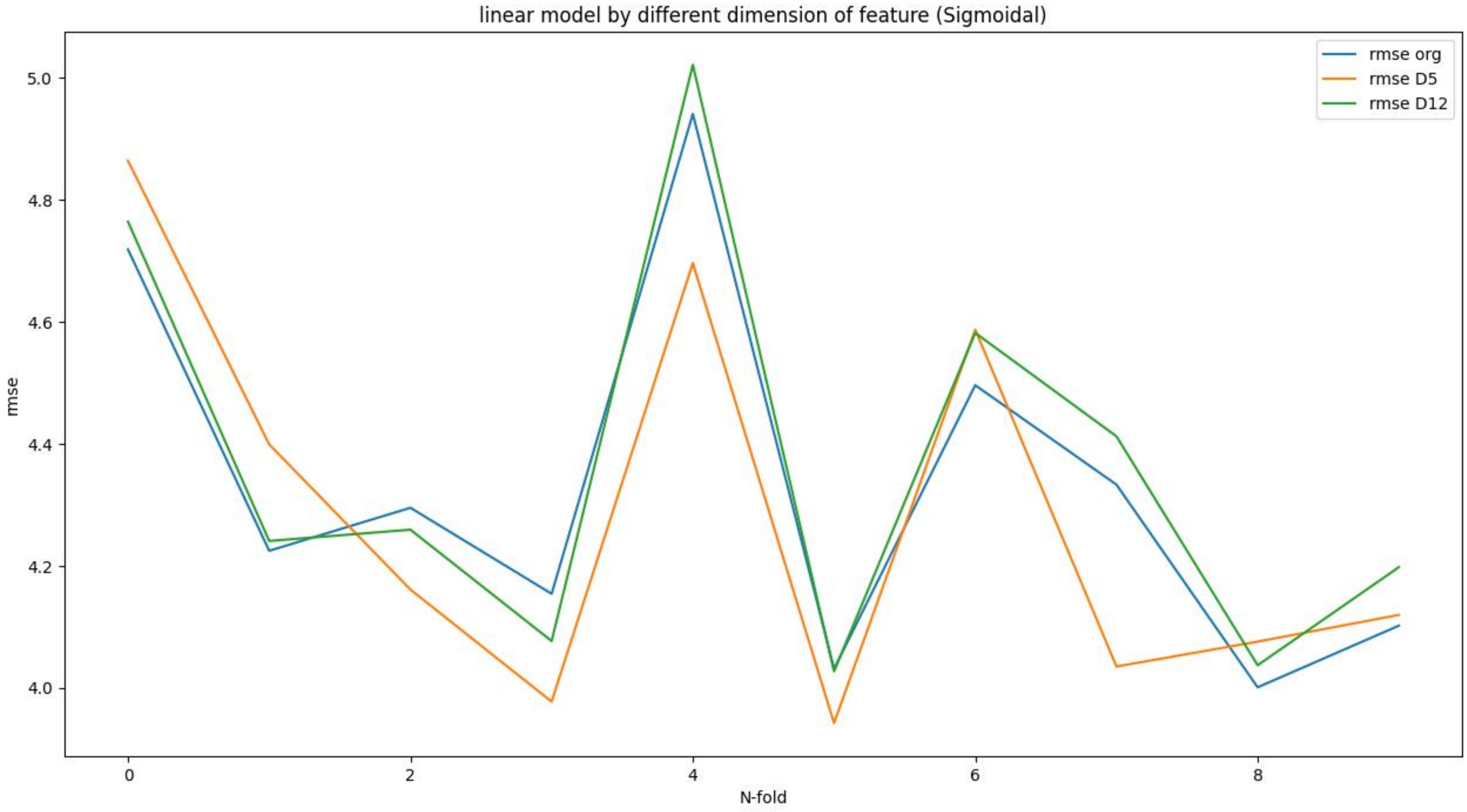
weights versus features(MAP_Sigmoidal)
weight
30
20
10
0
−10
const
AMB_TEMP
CH4
CO
NMHC
NO
NO2
NOx
O3
PM10
RAINFALL
RH
SO2
THC
WD_HR
WS_HR

Result:



Linear regresion(MAP_Sigmoidal)--the impact on remove one feature
The rmse without X feature
rmse with all features
rmse
5.8
5.6
5.4
5.2
5.0
4.8
4.6
4.4
4.2
AMB_TEMP
CH4
CO
NMHC
NO
NO2
NOx
O3
PM10
RAINFALL
RH
SO2
THC
WD_HR
WS_HR

Result:

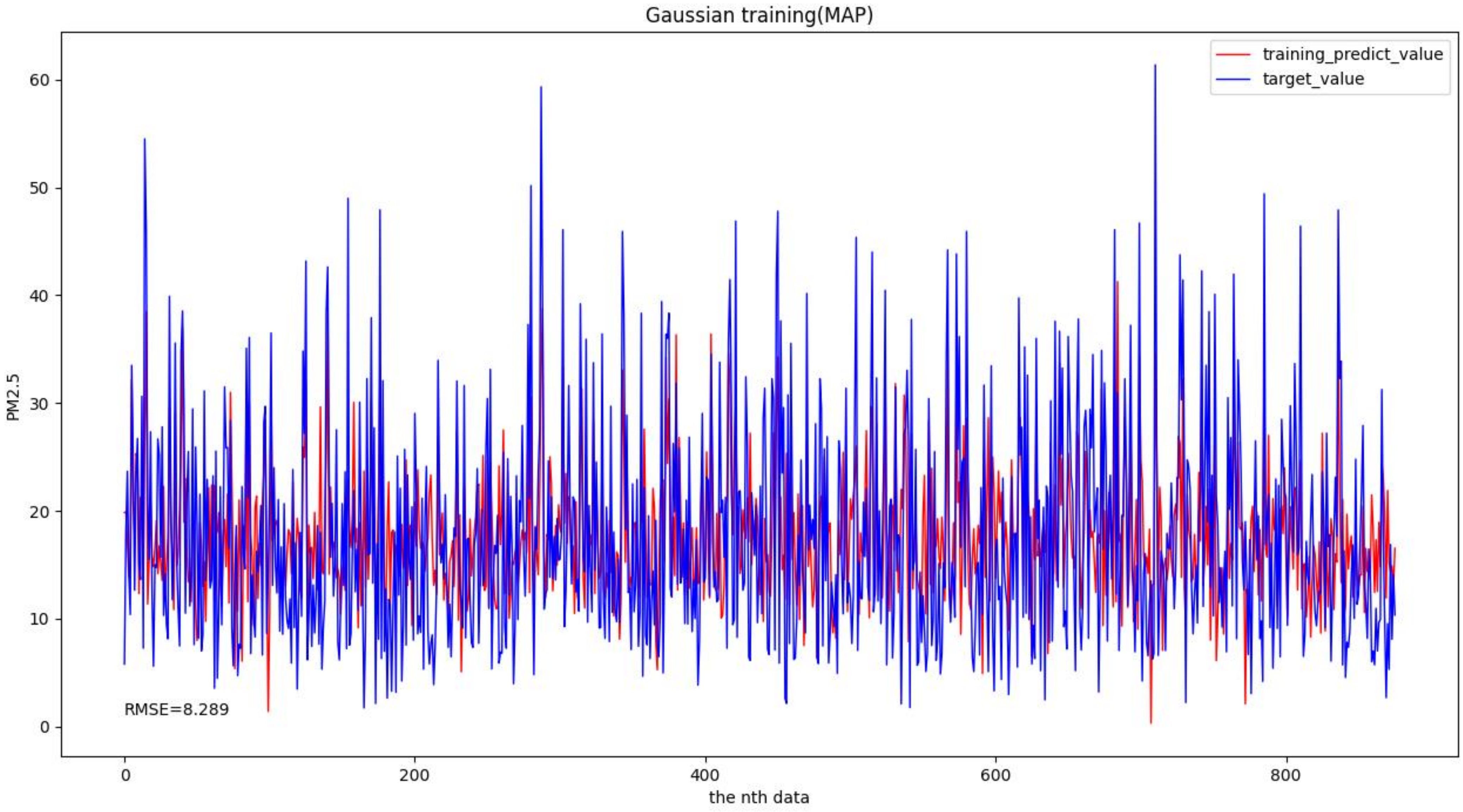


linear model by different dimension of feature (Sigmoidal)
rmse org
rmse D5
rmse D12
rmse
N-fold
5.0
4.8
4.6
4.4
4.2
4.0
0
2
4
6
8

Gaussian

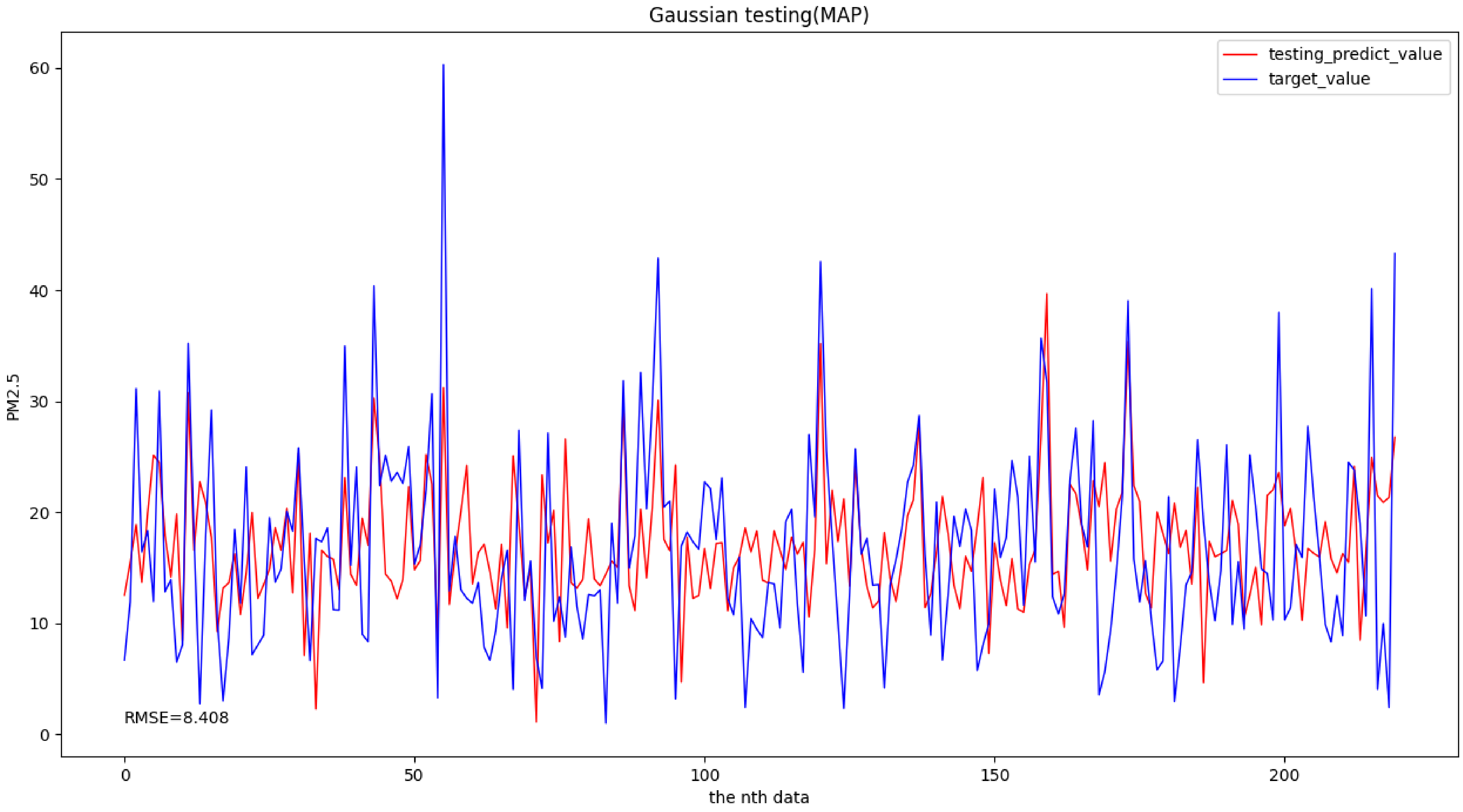
$$f(x, \mu, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} e^{\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)}$$

Result:



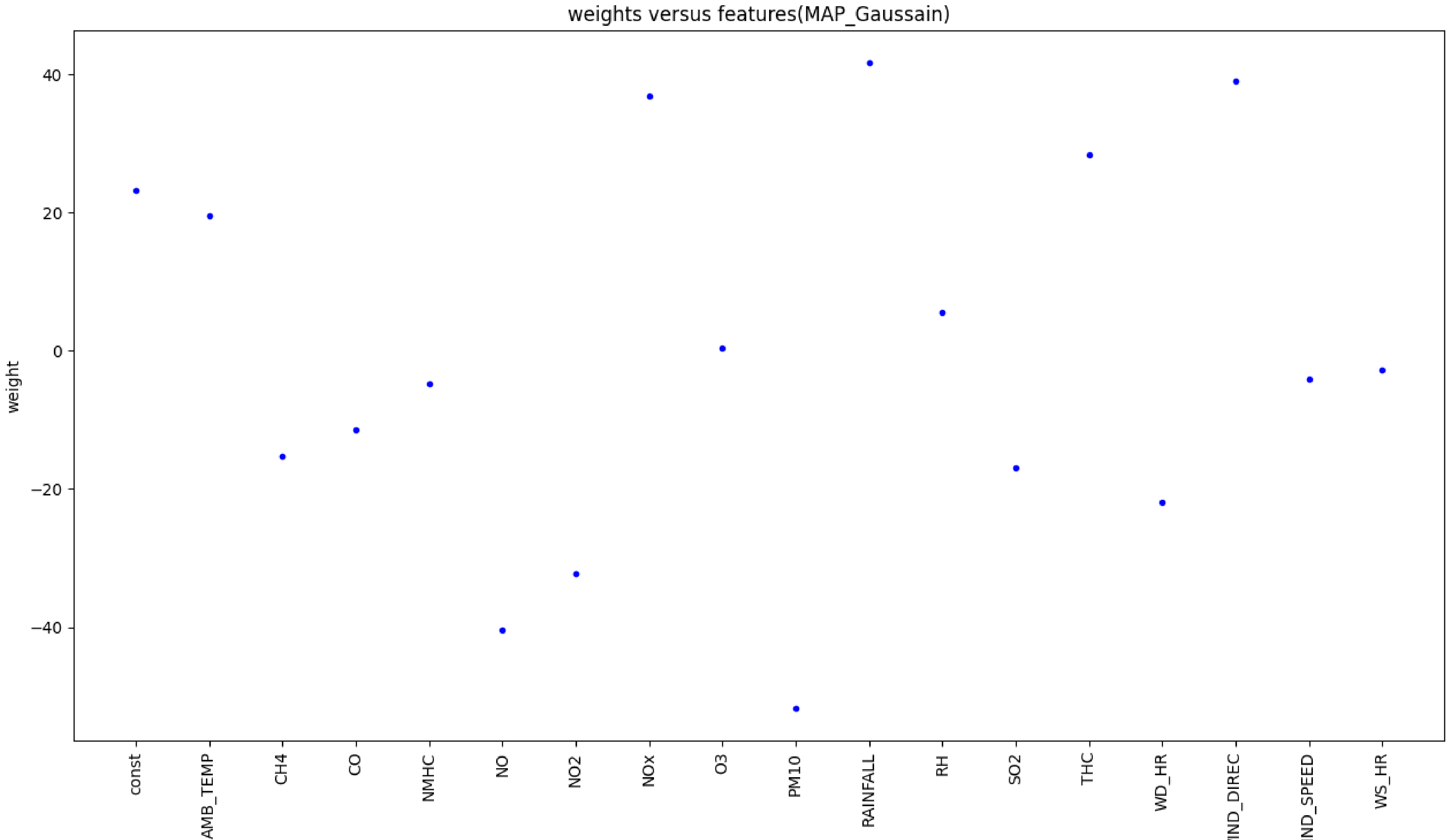
Gaussian training(MAP)
training_predict_value
target_value
PM2.5
0
10
20
30
40
50
60
200
400
600
800
the nth data
RMSE=8.289

Result:

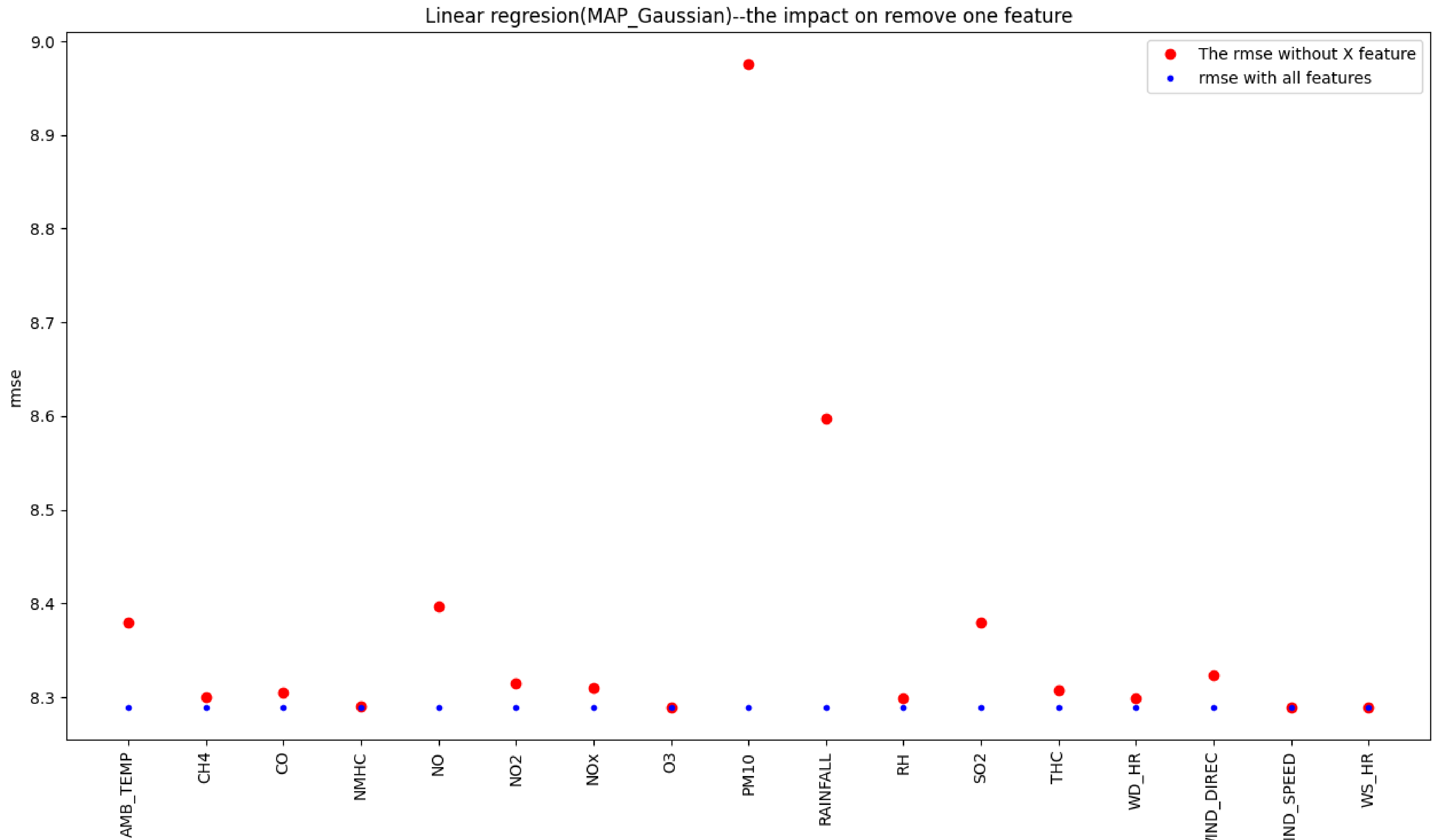


Gaussian testing(MAP)
testing_predict_value
target_value
PM2.5
the nth data
RMSE=8.408
0
10
20
30
40
50
60
0
50
100
150
200

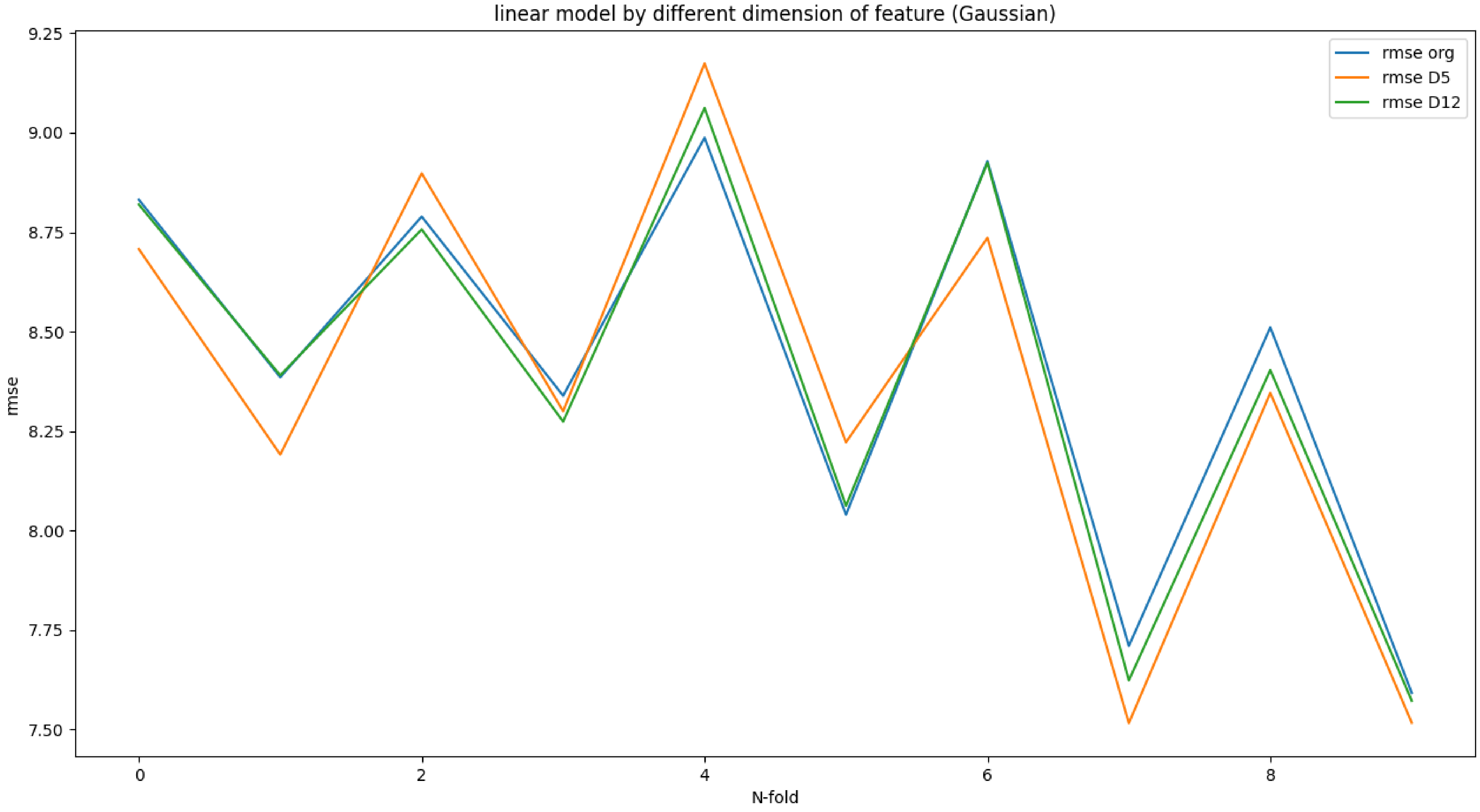
Result:


weights versus features(MAP_Gaussain)
weight
40
20
0
−20
−40
const
AMB_TEMP
CH4
CO
NMHC
NO
NO2
NOx
O3
PM10
RAINFALL
RH
SO2
THC
WD_HR
IND_DIREC
IND_SPEED
WS_HR

Result:


Linear regresion(MAP_Gaussian)--the impact on remove one feature
The rmse without X feature
rmse with all features
rmse
9.0
8.9
8.8
8.7
8.6
8.5
8.4
8.3
AMB_TEMP
CH4
CO
NMHC
NO
NO2
NOx
O3
PM10
RAINFALL
RH
SO2
THC
WD_HR
WS_HR

Result:



linear model by different dimension of feature (Gaussian)
rmse org
rmse D5
rmse D12
9.25
9.00
8.75
8.50
8.25
8.00
7.75
7.50
rmse
0
2
4
6
8
N-fold

Compare testing between MLE and MAP:

Linear regression(rmse)	MLE	MAP
Sigmoidal(Train)	4.125	4.222
Sigmoidal(Test)	5.658	4.453
Gaussian(Train)	8.234	8.275
Gaussian(Test)	8.806	8.678

由上表可以看到MAP的表現結果比MLE小一點點，因此推斷MAP是能夠比MLE在更精準一點的線性回歸方法。