회귀분석 프로젝트

분석 모형을 통한 기온 예측

2023.03.29(수)

송수린 | 현정환

프로젝트 수행 과정

01
데이터 수집
크롤링을 통한
기상 데이터 확보

02
데이터 전처리
Python을 이용한
데이터 전처리

03
데이터 분석1
R과 Python을 이용한
단순 & 다중 회귀분석

04
데이터 분석2
Python을 이용한
시계열분석

05
결과 검증 및 결론
예측 값과 실제 값의
비교를 통한 모형 검증
및 기온 예측

01

데이터 수집

데이터 수집

기상 데이터	기상청 기상자료개방포털
103개 관측소	2018년1월1일00시~2022년12월31일23시 시간대별 데이터
대기 정보 데이터	에어코리아 홈페이지 크롤링
59개 관측소	2018년1월1일00시~2022년12월31일23시 시간대별 데이터

데이터 개요

지점명	기상 관측소 지점명	PM2.5	1000L 공기 안에 포함된 먼지 농도
일시	기상 관측 날짜	Ozon	공기중 한시간 평균 오존 농도
지점	기상 관측소 지점 번호	NO2	공기중 한시간 평균 이산화질소 농도
기온(°C)	기상 관측소 기준 1시간동안의 기온 평균	CO	공기중 한시간 평균 일산화탄소 농도
강수량(mm)	기상 관측소 기준 1시간동안의 강수량	SO2	공기중 한시간 평균 이황산가스 농도
풍속(m/s)	기상 관측소 기준 1시간동안의 풍속 평균	증기압(hPa)	대기 중에 포함되어있는 수증기만의 압력
습도(%)	기상 관측소 기준 1시간동안의 습도 평균	이슬점온도(°C)	포화수증기압이 수증기압과 같아지는 온도
현지기압(hPa)	관측 지점(관측소)의 기압	시정(10m)	관측 위치에서 볼 수 있는 거리의 정도
해면기압(hPa)	현지 기압을 평균 해수면의 기압으로 추정한 값	지면온도(°C)	지면의 이물질을 모두 제거 한 후 측정한 온도
일조(hr)	1분마다 수신된 일조시간의 한시간 누적 값	전날기온	측정 일의 24시간 전의 온도(파생변수)
PM10	1000L 공기 안에 포함된 먼지 농도	강수여부	강수량이 존재하는지 여부(파생변수)

02

데이터 전처리

데이터 전처리

기상 데이터 원본

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	D	E	F	G
1	지점	지점명	일시	기온(°C)	강수량(mr	풍속(m/s)	풍향(16방	습도(%)	현지기압(I	해면기압(I	일조(hr)	일사(MJ/m	적설(cm)	전운량(10	운형(운형약어)		증기압(hP	이슬점온도	시정(10m)	지면온도(°C)
2	90	속초	2018-01-01 0:00	-1		1.1	250	23	1019.4	1021.7							1.1	-21	2000	-7
3	90	속초	2018-01-01 1:00	-2.1		1.7	230	28	1019.7	1022							1.1	-21.4	2000	-6.9
4	90	속초	2018-01-01 2:00	-2.1		1.4	160	29	1020.1	1022.4							1	-23	2000	-7.8
5	90	속초	2018-01-01 3:00	-2.2		0.9	230	28	1020.4	1022.7							0.9	-23.9	2000	-8.7
6	90	속초	2018-01-01 4:00	-2		1.2	250	27	1020.4	1022.7							0.9	-24.2	2000	-9.6
7	90	속초	2018-01-01 5:00	-1.7		1.4	270	26	1020.7	1023							0.9	-23.3	2000	-9.7
8	90	속초	2018-01-01 6:00	-1		1.8	270	23	1020.6	1022.9							1	-22.1	2000	-8.7
9	90	속초	2018-01-01 7:00	-1.1		2	290	24	1021.3	1023.6							0.9	-23.6	2000	-8.3
10	90	속초	2018-01-01 8:00	-1.2		1.9	270	25	1021.7	1024	0.1						1.1	-21.7	2000	-8.1
11	90	속초	2018-01-01 9:00	-0.3		3.9	290	23	1022.1	1024.4	1						1.3	-19.6	2000	-4.9
12	90	속초	2018-01-01 10:00	1.1		2.3	290	20	1022.8	1025.1	1						1.3	-19.3	2000	-1.6
13	90	속초	2018-01-01 11:00	2.6		2.1	320	18	1023.1	1025.4	1						1.5	-17.6	2000	3.8
14	90	속초	2018-01-01 12:00	2.7		3	290	19	1022.6	1024.9	1						1.6	-16.9	2000	11.2
15	90	속초	2018-01-01 13:00	3		5.7	290	19	1021.7	1024	1						1.6	-17.3	2000	13
16	90	속초	2018-01-01 14:00	3.7		4.9	290	17	1021.2	1023.5	1						1.7	-16.3	2000	14
17	90	속초	2018-01-01 15:00	3.8		4.1	290	17	1021.3	1023.6	1						1.7	-16.3	2000	10.1
18	90	속초	2018-01-01 16:00	3.8		3.8	290	16	1021.3	1023.6	1						1.8	-16	2000	0
19	90	속초	2018-01-01 17:00	2.7		3.4	270	17	1021.5	1023.8	0.8						1.7	-16.4	2000	-1.2
20	90	속초	2018-01-01 18:00	2		3.3	270	19	1022	1024.3	0						1.6	-17.2	2000	-2
21	90	속초	2018-01-01 19:00	1.1		2.1	270	21	1022.6	1024.9							1.4	-18.6	2000	-3.7
22	90	속초	2018-01-01 20:00	1.3		3.3	270	22	1022.8	1025.1							1.4	-18.7	2000	-3.6
23	90	속초	2018-01-01 21:00	1.5		3.2	270	20	1023	1025.3							1.5	-18	2000	-4
24	90	속초	2018-01-01 22:00	1		1.9	270	22	1023.1	1025.4							1.6	-17.5	2000	-3.3
25	90	속초	2018-01-01 23:00	0.9		1.1	290	21	1023.2	1025.5							1.7	-16.2	2000	-3.5
26	90	속초	2018-01-02 0:00	1.3		2.5	320	21	1022.7	1025							1.5	-18	2000	-4.6

데이터 전처리

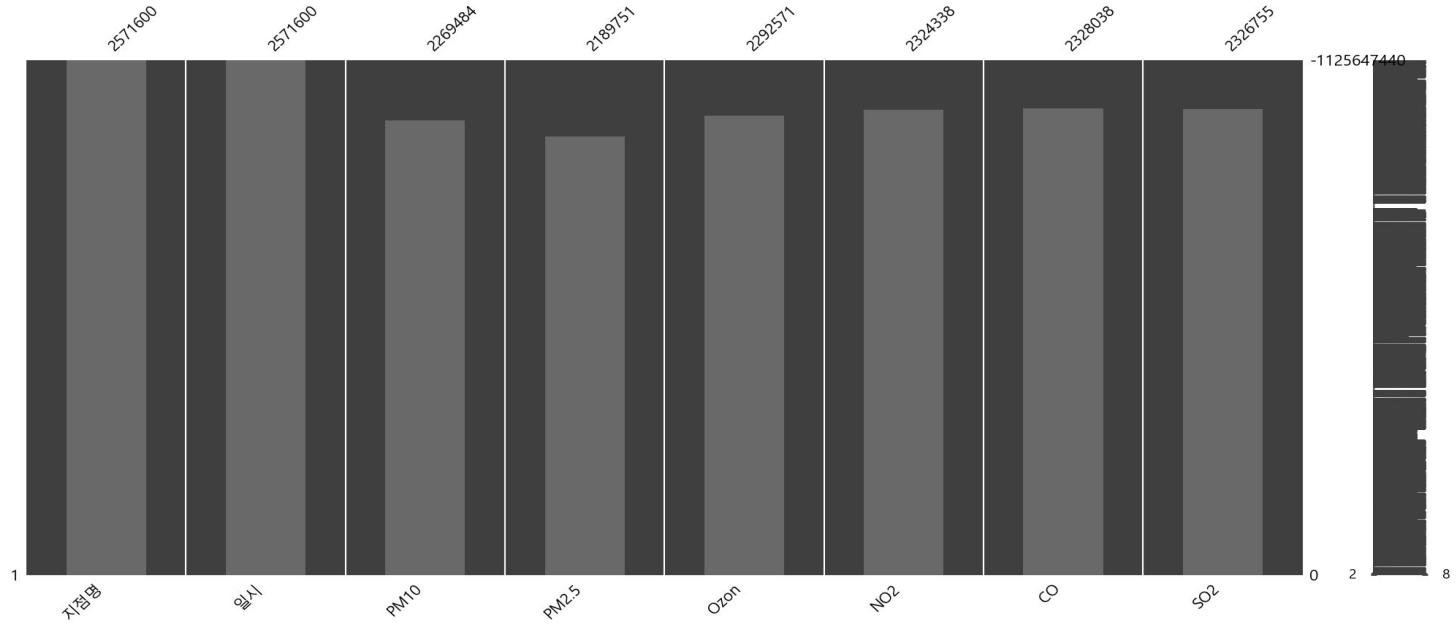
대기질 데이터 원본

백령도 (인천 옹진군 백령면 연화리 산241-2)

날짜	PM_{10}		$PM_{2.5}$		오존		이산화질소		일산화탄소		아황산가스	
(년-월-일:시)	(㎍/㎥)		(㎍/㎥)		(ppm)		(ppm)		(ppm)		(ppm)	
	등급	1시간	등급	1시간	등급	1시간	등급	1시간	등급	1시간	등급	1시간
2018-02-01:01	좋음	21	좋음	7		0.038		0.003		0.1		0.000
2018-01-31:24	좋음	27	좋음	6		0.037		0.003		0.1		0.001
2018-01-31:23	좋음	13	좋음	6		0.038		0.002		0.1		0.001
2018-01-31:22	좋음	21	좋음	6		0.039		0.001		0.1		0.001
2018-01-31:21	좋음	29	좋음	8		0.038		0.002		0.1		0.001
2018-01-31:20	좋음	17	좋음	11		0.038		0.002		0.1		0.002
2018-01-31:19	좋음	29	좋음	11		0.037		0.003		0.1		0.002
2018-01-31:18	좋음	28	좋음	7		0.036		0.003		0.1		0.002
2018-01-31:17	좋음	24	좋음	10		0.037		0.003		0.1		0.002
2018-01-31:16	보통	37	좋음	14		0.037		0.002		0.1		0.002
2018-01-31:15	보통	46	보통	18		0.035		0.004		0.2		0.003
2018-01-31:14	보통	50	보통	21		0.034		0.005		0.3		0.003
2018-01-31:13	보통	49	보통	16		0.033		0.005		0.3		0.003
2018-01-31:12	보통	44	보통	17		0.033		0.004		0.2		0.003

데이터 전처리 - 결측값 대체

기상 데이터 결측값



데이터 전처리 - 결측값 대체

	PM10	PM2.5
PM10	1.000000	0.740917
PM2.5	0.740917	1.000000

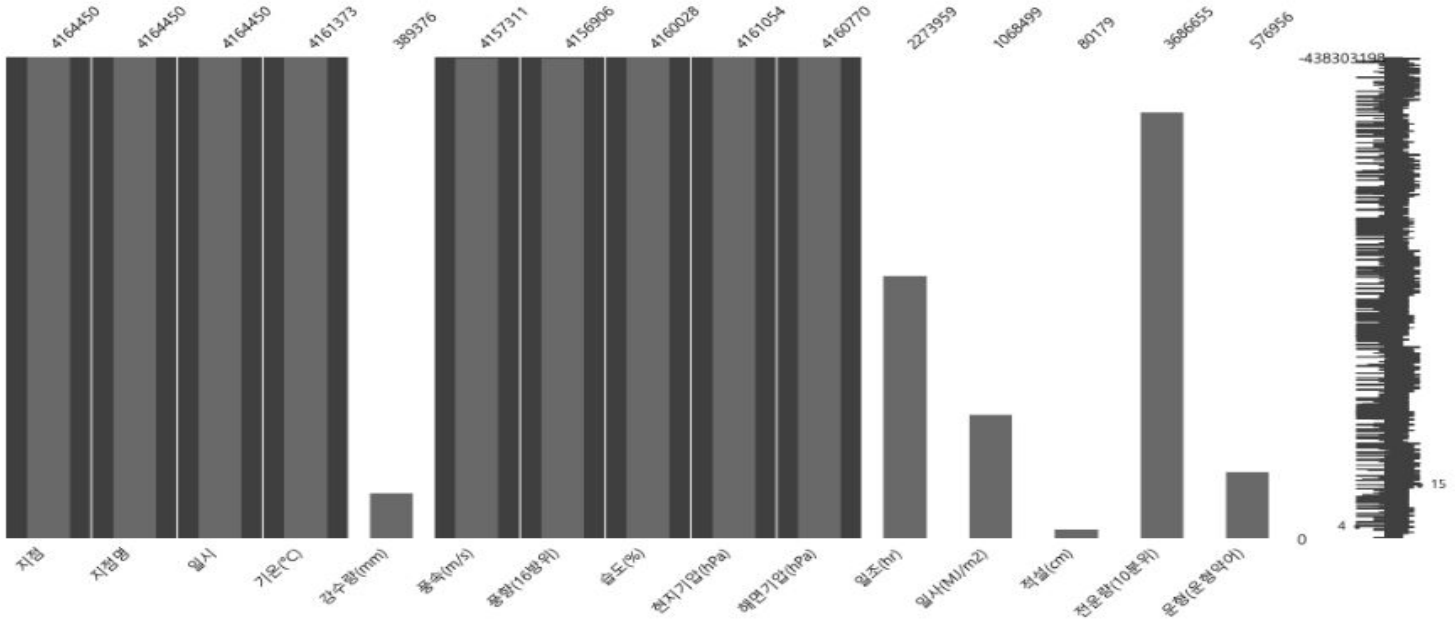
미세먼지와 초미세먼지 농도 간 양의 상관관계 존재
-> 각각의 결측값을 회귀대치법을 이용하여 처리

```
# 미세, 초미세 농도 결측값 회귀추정하여 대체
df['PM10']=df['PM10'].fillna(7.50653+1.4538987*df['PM2.5'])
```

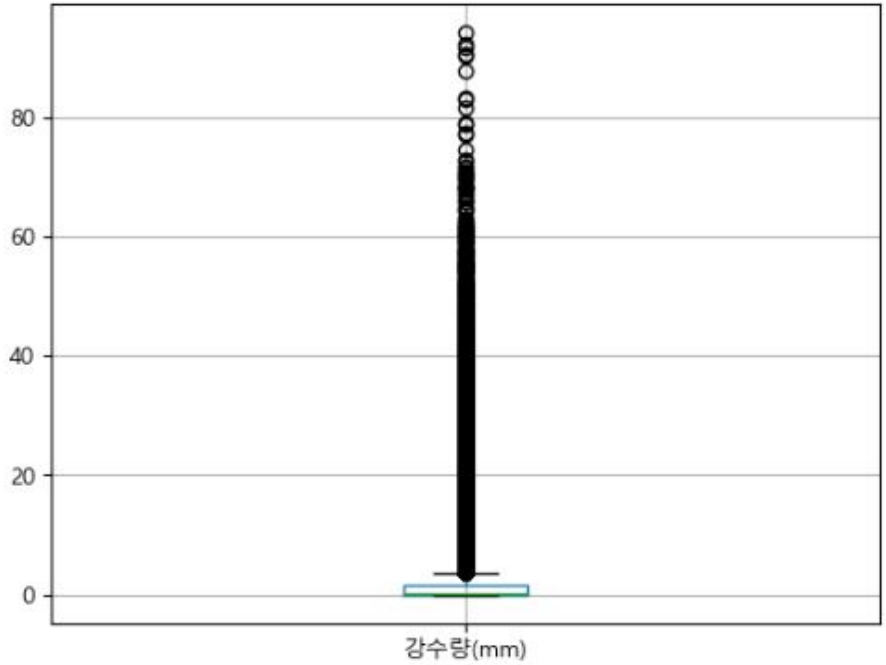
추후 분석에서 초미세먼지 농도는 다중공선성을 고려하여 제거

데이터 전처리 - 결측값 대체

대기질 데이터 결측값



데이터 전처리 - 강수량 데이터 변경



강수량이 없는 날이 더 많아서
강수량이 존재하면 이상치로 출력

-> 강수 여부라는 범주형 파생 변수 생성
(강수량 > 0: 1 , 강수량 = 0: 0)

데이터 전처리 - 데이터 결합

관측소와 일시 별로 열 결합

	지점명	일시	지점	기온(°C)	강수여부	풍속(m/s)	풍향(16방위)	습도(%)	현지기압(hPa)	해면기압(hPa)	...	Ozon	NO2	CO	SO2	증기압(hPa)	이슬점온도(°C)	시정(10m)	지면온도(°C)	월	전날기온
24	철원	2018-08-02 01:00:00	95	28.2	0	1.0	50.0	79.0	987.3	1004.6	...	0.023	0.003	0.3	0.001	30.2	24.2	1522.0	27.7	8	25.0
25	철원	2018-08-02 02:00:00	95	27.3	0	0.7	90.0	85.0	987.5	1004.9	...	0.024	0.003	0.3	0.001	30.7	24.5	1346.0	26.8	8	24.2
26	철원	2018-08-02 03:00:00	95	26.5	0	0.4	0.0	87.0	987.8	1005.2	...	0.024	0.003	0.2	0.001	30.0	24.1	1072.0	26.2	8	23.6
27	철원	2018-08-02 04:00:00	95	26.1	0	1.0	70.0	91.0	987.9	1005.4	...	0.024	0.002	0.2	0.001	30.7	24.5	1125.0	25.7	8	23.7
28	철원	2018-08-02 05:00:00	95	26.5	0	0.6	140.0	90.0	988.4	1005.8	...	0.021	0.002	0.2	0.001	31.1	24.7	1329.0	25.4	8	22.9
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
1933352	거제	2022-12-31 19:00:00	294	3.6	0	1.3	270.0	49.0	1023.7	1029.3	...	0.037	0.008	0.5	0.003	3.9	-6.1	5000.0	0.4	12	3.5
1933353	거제	2022-12-31 20:00:00	294	3.4	0	1.5	290.0	54.0	1023.7	1029.4	...	0.034	0.011	0.5	0.003	4.2	-5.0	4142.0	-0.4	12	3.6
1933354	거제	2022-12-31 21:00:00	294	3.1	0	1.6	290.0	58.0	1024.2	1029.9	...	0.037	0.007	0.5	0.003	4.4	-4.3	3315.0	-1.0	12	1.5

03

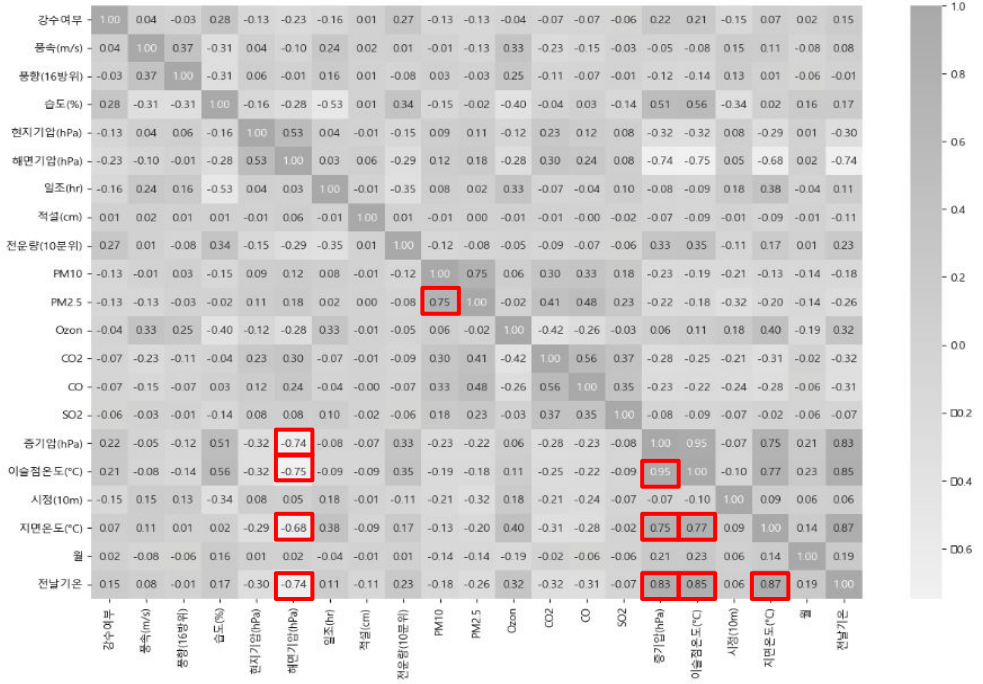
데이터 분석1

R 과 Python을 이용한 회귀 분석

단순회귀분석

독립변수	Adjusted R-squared
풍속	0.004743
일조율	0.03441
적설량	0.01336
미세먼지 농도	0.02109
초미세먼지 농도	0.04356
오존 농도	0.1239

다중회귀분석



해면기압, PM10, PM2.5,증기압, 이슬점온도,지면온도에서 다중공선성 의심 됨

다중회귀분석

범주형 변수 조합	조건 수	설명력
풍향 & 전운량 & 월 &강수여부	6.29e+15	0.945
전운량 & 월 & 강수여부	1.27e+16	0.944
풍향 & 전운량 & 강수여부	5.35e+15	0.939
풍향 & 월 & 강수여부	1.28e+16	0.944
풍향 & 전운량 & 월	1.28e+16	0.945

다중회귀분석

```
# 범위형 변수 제거
X.drop(columns=['풍향(16방위)', '전운량(10분위)', '월'], axis=1, inplace=True)
```

```
# 더미변수 생성
X=pd.get_dummies(data=X, columns=['강수여부'])
```

좋은 모형 예측을 위해 요인이 너무 많은 범주형 변수 삭제

강수량 데이터를 이상치로 반환하는 문제 해결을 위해 '강수 여부' 파생변수 생성

다중회귀분석

```
scaler = StandardScaler()
scaler.fit(X.iloc[:,:-2])
X_scaled = scaler.transform(X.iloc[:,:-2])
X_scaled = pd.DataFrame(X_scaled)
X_scaled.columns = ['풍속(m/s)', '습도(%)', '현지기압(hPa)', '해면기압(hPa)', '일조(hr)', '적설(cm)', 'PM10', 'PM2.5',
                    'O3', 'NO2', 'CO', 'SO2', '증기압(hPa)', '이슬점온도(°C)', '시정(10m)', '지면온도(°C)', '전날기온']
X_scaled = pd.concat([X_scaled, X.iloc[:,-2:]], axis=1)
X_scaled
```

	풍속(m/s)	습도(%)	현지기압(hPa)	해면기압(hPa)	일조(hr)	적설(cm)	PM10	PM2.5	O3	NO2	CO	SO2	증기압(hPa)	이슬점온도(°C)	시정(10m)	지면온도(°C)	전날기온	강수여부_0	강수여부_1
0	-0.543789	0.446784	-1.147573	-1.434567	-0.645615	-0.070183	-0.019538	-0.005901	-0.525180	-0.874219	-0.547369	-1.069525	1.997825	1.491758	-0.335595	1.010295	1.163569	1	0
1	-0.725001	0.724630	-1.134416	-1.398730	-0.645615	-0.070183	0.151909	0.123778	-0.472111	-0.874219	-0.547369	-1.069525	2.055110	1.517753	-0.482683	0.937277	1.084297	1	0
2	-0.906213	0.817246	-1.114681	-1.362894	-0.645615	-0.070183	-0.053827	-0.031837	-0.472111	-0.874219	-1.048907	-1.069525	1.974912	1.483093	-0.711672	0.888599	1.024842	1	0
3	-0.543789	1.002477	-1.108102	-1.339002	-0.645615	-0.070183	-0.019538	-0.005901	-0.472111	-0.966011	-1.048907	-1.069525	2.055110	1.517753	-0.667378	0.848034	1.034751	1	0
4	-0.785405	0.956169	-1.075209	-1.291220	-0.645615	-0.070183	0.151909	0.123778	-0.631320	-0.966011	-1.048907	-1.069525	2.100937	1.535083	-0.496890	0.823694	0.955478	1	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
1933328	-0.362577	-0.942446	1.247015	1.515993	-0.645615	-0.070183	-0.705324	-0.524617	0.217797	-0.415259	0.455707	0.057103	-1.015333	-1.133756	2.571060	-1.204574	-0.966888	1	0
1933329	-0.241769	-0.710908	1.247015	1.527938	-0.645615	-0.070183	-0.671035	-0.489956	0.058587	-0.139884	0.455707	0.057103	-0.980963	-1.038440	1.854007	-1.269479	-0.956979	1	0
1933330	-0.181365	-0.525677	1.279908	1.587666	-0.645615	-0.070183	-0.602456	-0.489956	0.217797	-0.507051	0.455707	0.057103	-0.958049	-0.977785	1.162862	-1.318158	-1.165070	1	0
1933331	-0.302173	-0.386754	1.253594	1.539884	-0.645615	-0.070183	-0.533877	-0.421113	0.217797	-0.507051	0.455707	0.057103	-0.969506	-1.021110	1.062574	-1.391175	-1.244343	1	0
1933332	-0.362577	-0.201523	1.279908	1.587666	-0.645615	-0.070183	-0.533877	-0.489956	0.111657	-0.507051	0.455707	0.057103	-0.980963	-1.055770	0.809349	-1.456080	-1.244343	1	0

변수 간 단위 차이로 인한 문제 방지를 위한 스케일링

다중회귀분석

	VIF Factor	features
0	1.282310	풍속(m/s)
1	6.320909	습도(%)
2	1.508063	현지기압(hPa)
3	3.590454	해면기압(hPa)
4	2.194225	일조(hr)
5	1.026503	적설(cm)
6	2.388284	PM10
7	2.954484	PM2.5
8	2.230332	O3
9	2.190010	NO2
10	1.752756	CO
11	1.262375	SO2
12	12.007960	증기압(hPa)
13	26.795980	이슬점온도(°C)
14	1.350652	시정(10m)
15	9.762461	지면온도(°C)
16	8.544146	전날기온
17	1.010853	강수여부_0
18	1.167441	강수여부_1

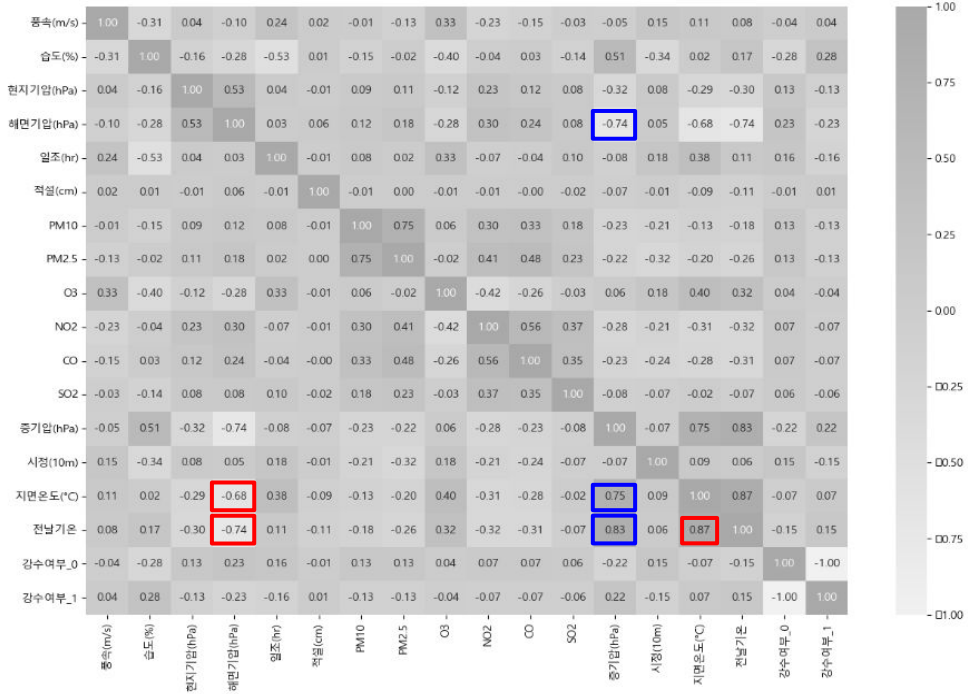
VIF값이 10 넘는 변수 중
VIF값이 가장 큰 이슬점온도 제거

다중회귀분석

	VIF Factor	features
0	1.281796	풍속(m/s)
1	3.286038	습도(%)
2	1.505450	현지기압(hPa)
3	3.554263	해면기압(hPa)
4	2.164952	일조(hr)
5	1.021423	적설(cm)
6	2.386740	PM10
7	2.952681	PM2.5
8	2.137356	O3
9	2.120577	NO2
10	1.751513	CO
11	1.262108	SO2
12	8.312001	증기압(hPa)
13	1.345371	시정(10m)
14	7.892783	지면온도(°C)
15	7.053457	전날기온
16	1.009921	강수여부_0
17	1.153061	강수여부_1

이슬점온도 제거 후 **VIF** 재확인
-> 10을 넘는 **VIF** 없음

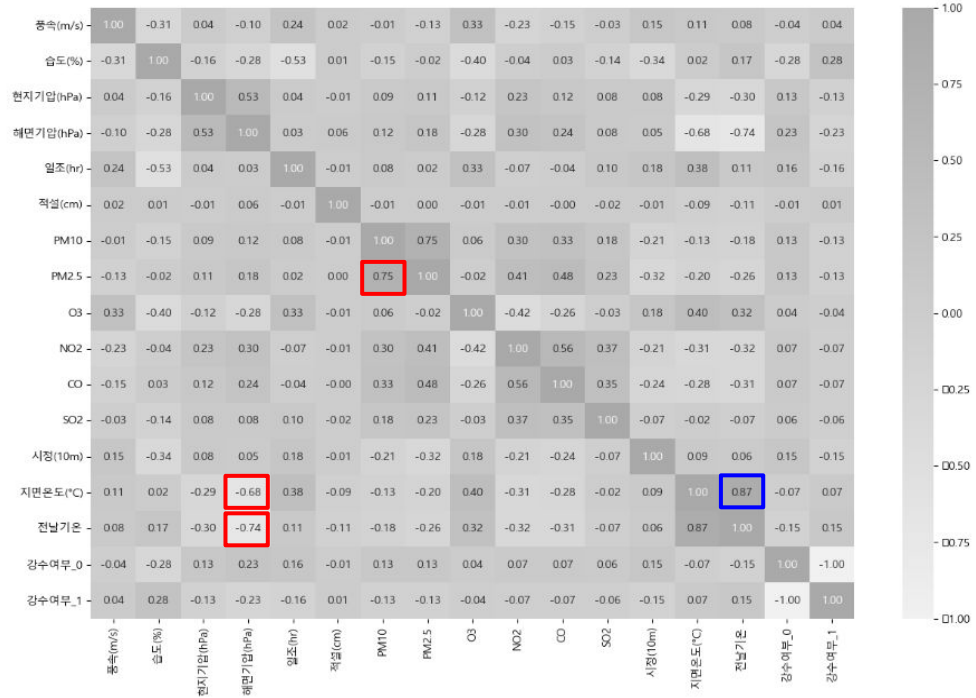
다중회귀분석



1차적으로 변수 제거, 변경 후 상관계수 재측정

-> 상대적으로 다른 변수들과의 상관계수(절대값 0.7이상)가 큰 증기압 제거

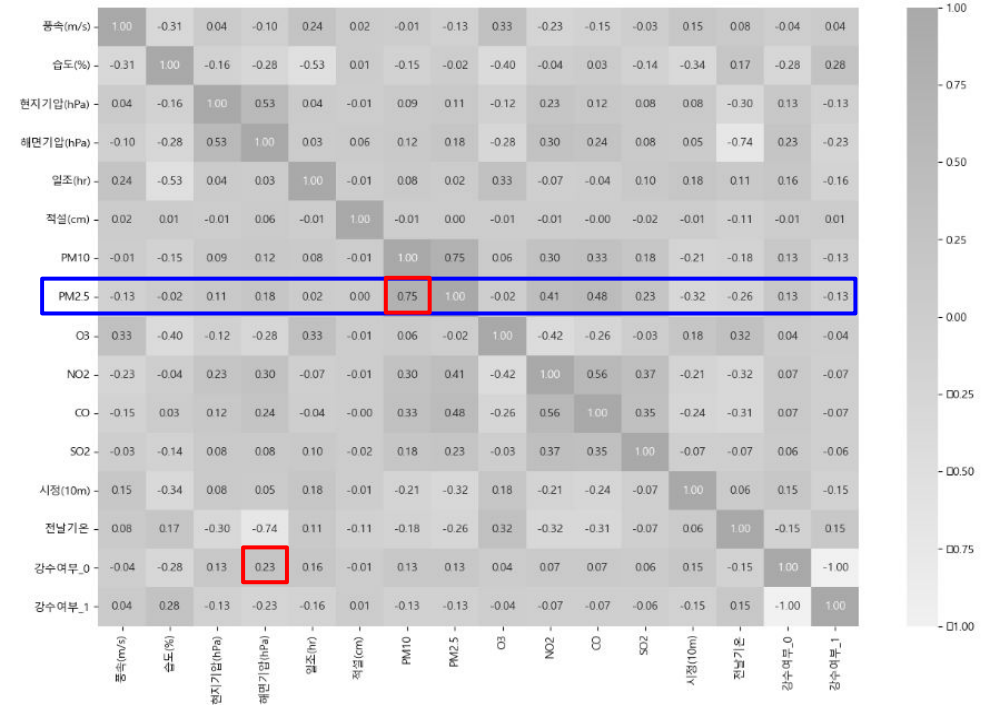
다중회귀분석



증기압 제거 후 상관계수 재측정

-> 상대적으로 다른 변수들과의 상관계수가 큰 지면온도 제거

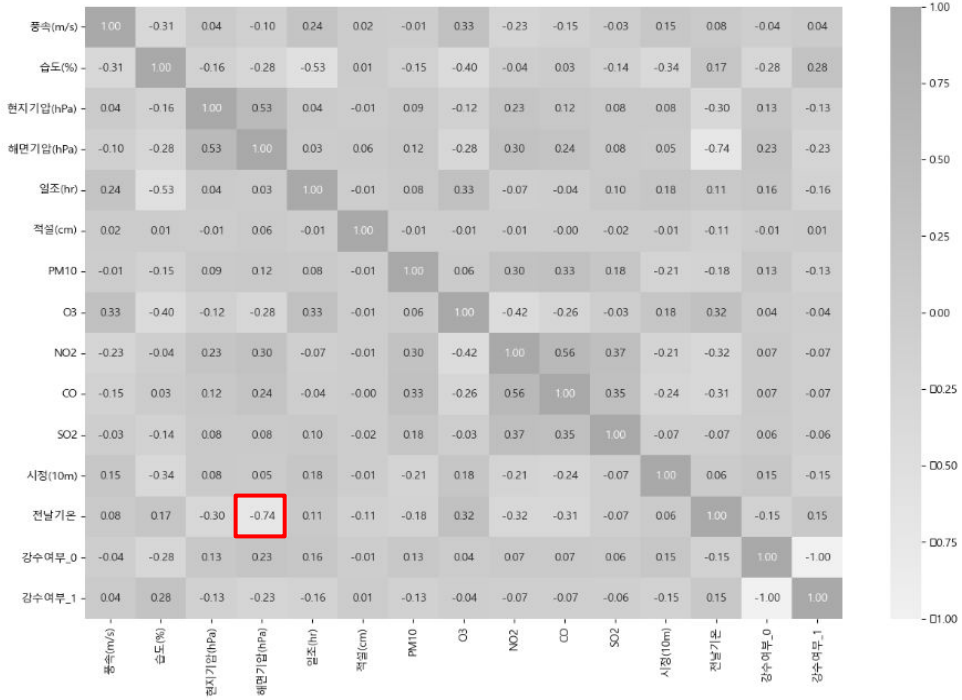
다중회귀분석



지면온도 제거 후 상관계수 재측정

-> 회귀대치법에 사용된
PM2.5 제거

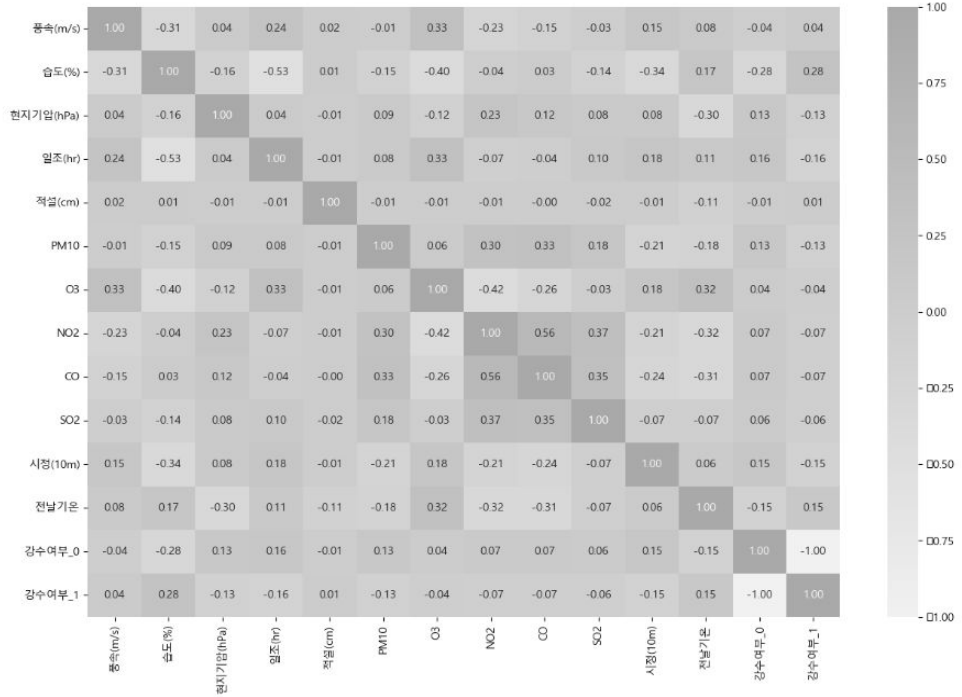
다중회귀분석



PM2.5 제거 후 상관계수 재측정

-> 상관계수(-0.74)가 높은 해면기압과 전날기온 중에 해면기압 제거

다중회귀분석



최종적으로 남은 변수들 간의 상관계수 확인

-> 상관계수가 0.7 넘는 변수 없음

다중회귀분석

OLS Regression Results

Dep. Variable:	기온(°C)	R-squared:	0.886
Model:	OLS	Adj. R-squared:	0.886
Method:	Least Squares	F-statistic:	1.159e+06
Date:	Fri, 24 Mar 2023	Prob (F-statistic):	0.00
Time:	15:51:24	Log-Likelihood:	-5.1113e+06
No. Observations:	1933337	AIC:	1.022e+07
Df Residuals:	1933323	BIC:	1.022e+07
Df Model:	13		
Covariance Type:	nonrobust		

설명력(R-sqared) 88.6%

	coef	std err	t	P>|t|	[0.025	0.975]
풍속(m/s)	-0.2207	0.003	-81.231	0.000	-0.226	-0.215
습도(%)	0.4485	0.004	122.044	0.000	0.441	0.456
현지기압(hPa)	-0.3626	0.003	-137.110	0.000	-0.368	-0.357
일조(hr)	0.8652	0.003	288.235	0.000	0.859	0.871
적설(cm)	-0.1095	0.002	-44.298	0.000	-0.114	-0.105
PM10	0.0112	0.003	4.058	0.000	0.006	0.017
O3	0.7676	0.003	229.583	0.000	0.761	0.774
NO2	0.4047	0.003	115.702	0.000	0.398	0.412
CO	-0.0795	0.003	-25.136	0.000	-0.086	-0.073
SO2	0.1357	0.003	49.533	0.000	0.130	0.141
시정(10m)	0.0102	0.003	3.649	0.000	0.005	0.016
전날기온	9.0517	0.003	2990.167	0.000	9.046	9.058
강수여부_0	13.2997	0.003	5242.329	0.000	13.295	13.305
강수여부_1	12.6012	0.011	1191.834	0.000	12.580	12.622

Omnibus:	177227.170	Durbin-Watson:	0.116
Prob(Omnibus):	0.000	Jarque-Bera (JB):	1195264.510
Skew:	-0.146	Prob(JB):	0.00
Kurtosis:	6.841	Cond. No.	7.19

모든 변수들이 유의함

데이터 분할

```
X_train, X_test, y_train, y_test = train_test_split(X_scaled, y, test_size=0.2, random_state=0)
X_train, X_val, y_train, y_val = train_test_split(X_train, y_train, test_size=0.1, random_state=0)
print(X_train.shape, X_test.shape, X_val.shape)
```

```
(1391999, 14) (386667, 14) (154667, 14)
```

학습용: 72%(약 140만개) **검증용: 20%(약 39만개)** **예측용: 9%(약15만개)**

선형회귀모형

```
%%time
lin_reg = LinearRegression()
model_lin = lin_reg.fit(X_train, y_train)
y_pred = lin_reg.predict(X_test)
rms = np.sqrt(mean_squared_error(y_test, y_pred))
rms
```

```
CPU times: total: 4.97 s
Wall time: 1.65 s
3.4071507092372784
훈련용: 0.886206635097972
검증용: 0.8858739327650216
```

의사결정나무 모형

```
%%time
tree_reg = DecisionTreeRegressor(criterion = 'squared_error',
                                 splitter='best',
                                 max_depth=14,
                                 min_samples_leaf=10,
                                 random_state=0)
model_tree = tree_reg.fit(X_train, y_train)
y_pred = tree_reg.predict(X_test)
rms = np.sqrt(mean_squared_error(y_test, y_pred))
rms
```

```
CPU times: total: 17.5 s
Wall time: 17.5 s
3.14786680742729
훈련용: 0.9104048293242206
검증용: 0.9025829756405 71
```

랜덤포레스트

```
%%time
forest_reg=RandomForestRegressor(random_state=0, n_jobs=-1) #CPU full 사용
model_forest = forest_reg.fit(X_train, y_train)
y_pred=forest_reg.predict(X_test)
rms=np.sqrt(mean_squared_error(y_test, y_pred))
rms
```

```
CPU times: total: 48min 42s
Wall time: 3min 14s
2.8071162898033473
훈련용: 0.9890982332535317
검증용: 0.9225498260530902
```

신경망 모형

```
# 인공신경망
model = Sequential()
model.add(Dense(56, input_shape=(len(X_train.columns),), activation='relu', name='input'))
model.add(Dense(28, activation='relu', name='hidden-1'))
model.add(Dense(14, activation='relu', name='hidden-2'))
model.add(Dense(7, activation='relu', name='hidden-3'))
model.add(Dense(1, name='output')) #항등함수
model.compile(loss='mse', optimizer='adam', metrics=['mse', 'mae'])
model.summary()
```

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 input (Dense)               (None, 56)                840

 hidden-1 (Dense)            (None, 28)                1596

 hidden-2 (Dense)            (None, 14)                406

 hidden-3 (Dense)            (None, 7)                 105

 output (Dense)              (None, 1)                 8

=================================================================
Total params: 2,955
Trainable params: 2,955
Non-trainable params: 0
_________________________________________________________________
```

입력값을 3개의 은닉층을 거쳐 모델 구축

신경망 모형

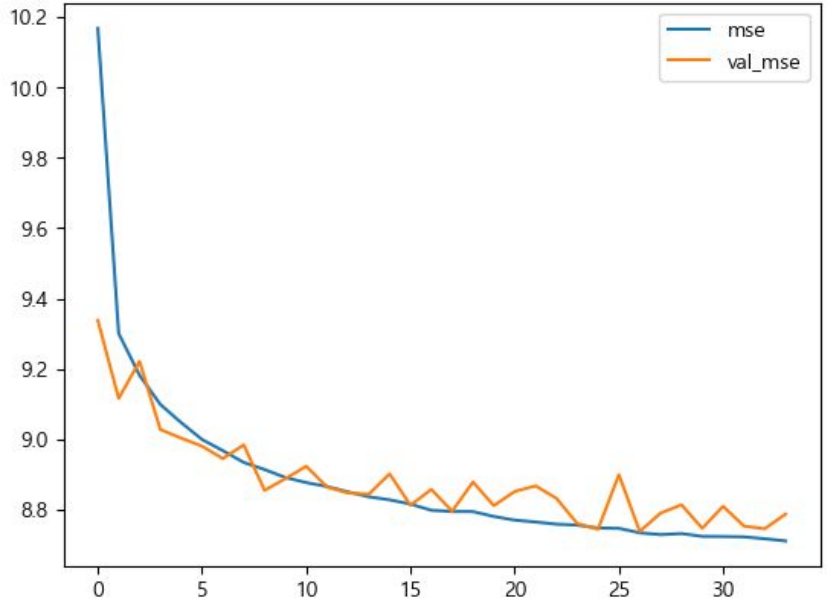
```
np.random.seed(5)
early_stopping = EarlyStopping(monitor='val_loss', patience=7)
hist = model.fit(X_train, y_train, epochs=500, validation_split=0.2, callbacks=[early_stopping])
```

```
Epoch 32/500
34800/34800 [==============================] - 508s 15ms/step - loss: 8.7221 - mse: 8.7221 - mae: 2.2042 - val_loss: 8.7527 - val_mse: 8.7527 - val_mae: 2.1940
Epoch 33/500
34800/34800 [==============================] - 529s 15ms/step - loss: 8.7168 - mse: 8.7168 - mae: 2.2036 - val_loss: 8.7453 - val_mse: 8.7453 - val_mae: 2.1992
Epoch 34/500
34800/34800 [==============================] - 552s 16ms/step - loss: 8.7105 - mse: 8.7105 - mae: 2.2031 - val_loss: 8.7871 - val_mse: 8.7871 - val_mae: 2.2017
CPU times: total: 5h 56min 5s
Wall time: 5h 9min 30s
```

훈련용 데이터 1,391,999개 중에 20%를 검증용 데이터로 사용하여 500번 학습

34번만에 학습 완료

신경망 모형



훈련용과 검증용 평균제곱오차

신경망 모형

```
scores = model.evaluate(X_train, y_train)
print(scores)
scores = model.evaluate(X_test, y_test)
print(scores)
```

```
43500/43500 [==============================] - 388s 9ms/step - loss: 8.7126 - mse: 8.7126 - mae: 2.1946
[8.7125520706176 76, 8.7125520706176 76, 2.1945810317993164]
12084/12084 [==============================] - 106s 9ms/step - loss: 8.7842 - mse: 8.7842 - mae: 2.2008
[8.784239768981934, 8.784239768981934, 2.2007851600646973]
```

```
from sklearn.metrics import r2_score
pred = model.predict(X_test)
r2_score(y_test, pred)
```

```
12084/12084 [==============================] - 57s 5ms/step
0.9136412129234297
```

구축한 모델의 정확도: 약 91%

하이퍼파라미터 튜닝

```
from sklearn.model_selection import RandomizedSearchCV
from scipy.stats import randint
param_distribs={
 'n_estimators': randint(low=1, high=100),
 'max_features': randint(low=1, high=8),
 }
rnd_search=RandomizedSearchCV(forest_reg, param_distributions=param_distribs, cv=5, random_state=2)
rnd_search.fit(X_train, y_train)

cvres=rnd_search.cv_results_
for mean_score, params in zip(cvres["mean_test_score"],cvres["params"]):
    print(np.sqrt(mean_score), params)
```

```
0.9494020395762193 {'max_features': 1, 'n_estimators': 16}
0.960822407849167 {'max_features': 6, 'n_estimators': 73}
0.9600281072257755 {'max_features': 7, 'n_estimators': 44}
0.9616441390617363 {'max_features': 3, 'n_estimators': 76}
0.941576560215161 {'max_features': 1, 'n_estimators': 8}
0.9612396148750093 {'max_features': 3, 'n_estimators': 50}
0.961633901608306 {'max_features': 4, 'n_estimators': 86}
0.9596902415089246 {'max_features': 3, 'n_estimators': 21}
0.9604609364752975 {'max_features': 5, 'n_estimators': 38}
0.9516193955970892 {'max_features': 4, 'n_estimators': 5}
```

최적의 파라미터 조합: **max_features 3개, n_estimators 76개**

하이퍼파라미터 튜닝

```
%%time
model_RSCV_forest=rnd_search.best_estimator_
y_pred=model_RSCV_forest.predict(X_test)
rms=np.sqrt(mean_squared_error(y_test, y_pred))
rms
```

```
CPU times: total: 1min 51s
Wall time: 7.73 s
2.72632709517729
훈련용: 0.9895306843754185
검증용: 0.9269268073037014
```

데이터 분할

```
X_train, X_test, y_train, y_test=train_test_split(X_scaled, y, test_size=0.2, random_state=0)
X_train, X_val, y_train, y_val=train_test_split(X_train, y_train, test_size=0.1, random_state=0)
print(X_train.shape, X_test.shape, X_val.shape)
```

```
(1391998, 16) (386667, 16) (154667, 16)
```

학습용: 72%(약 140만개) **검증용: 20%(약 39만개)** **예측용: 9%(약15만개)**

다중회귀 분석

	VIF Factor	features
0	1.301954	풍속(m/s)
1	27.705264	습도(%)
2	1.506393	현지기압(hPa)
3	3.636364	해면기압(hPa)
4	2.163982	일조(hr)
5	1.026010	적설(cm)
6	2.386472	PM10
7	2.935056	PM2.5
8	2.230409	O3
9	2.188005	NO2
10	1.749850	CO
11	1.263672	SO2
12	11.951279	증기압(hPa)
13	125.647945	이슬점온도(°C)
14	1.375422	시정(10m)
15	10.833174	지면온도(°C)
16	95.678381	전날기온
17	1.011306	강수여부_0
18	1.174433	강수여부_1

VIF값이 10 넘는 변수 중
VIF값이 가장 큰 이슬점온도 제거

다중회귀 분석

	VIF Factor	features
0	1.287559	풍속(m/s)
1	3.634456	습도(%)
2	1.506282	현지기압(hPa)
3	3.619774	해면기압(hPa)
4	2.163135	일조(hr)
5	1.025997	적설(cm)
6	2.386378	PM10
7	2.931069	PM2.5
8	2.208107	O3
9	2.167528	NO2
10	1.749337	CO
11	1.263065	SO2
12	10.977250	증기압(hPa)
13	1.345264	시정(10m)
14	10.716394	지면온도(°C)
15	16.844576	전날기온
16	1.010013	강수여부_0
17	1.154481	강수여부_1

VIF값이 10 넘는 변수 중
VIF값이 가장 큰 전날기온 제거

다중회귀분석

OLS Regression Results

Dep. Variable:	기온(°C)	R-squared:	0.934
Model:	OLS	Adj. R-squared:	0.934
Method:	Least Squares	F-statistic:	1.827e+06
Date:	Tue, 28 Mar 2023	Prob (F-statistic):	0.00
Time:	22:51:27	Log-Likelihood:	-4.5835e+06
No. Observations:	1933332	AIC:	9.167e+06
Df Residuals:	1933316	BIC:	9.167e+06
Df Model:	15		
Covariance Type:	nonrobust		

Omnibus:	239363.563	Durbin-Watson:	0.218
Prob(Omnibus):	0.000	Jarque-Bera (JB):	673408.783
Skew:	-0.680	Prob(JB):	0.00
Kurtosis:	5.551	Cond. No.	8.11

설명력(R-sqared) 93.4%

	coef	std err	t	P>|t|	[0.025	0.975]
풍속(m/s)	-0.2164	0.002	-102.981	0.000	-0.221	-0.212
습도(%)	-1.3063	0.003	-395.543	0.000	-1.313	-1.300
현지기압(hPa)	0.2154	0.002	94.408	0.000	0.211	0.220
해면기압(hPa)	-0.9565	0.003	-275.704	0.000	-0.963	-0.950
일조(hr)	-0.3036	0.003	-113.308	0.000	-0.309	-0.298
적설(cm)	-0.2676	0.002	-142.650	0.000	-0.271	-0.264
PM10	0.1270	0.002	60.032	0.000	0.123	0.131
O3	0.3936	0.003	147.655	0.000	0.388	0.399
NO2	0.4988	0.003	186.366	0.000	0.494	0.504
CO	-0.1307	0.002	-54.427	0.000	-0.135	-0.126
SO2	-0.1076	0.002	-51.429	0.000	-0.112	-0.104
증기압(hPa)	4.6132	0.005	937.235	0.000	4.604	4.623
시정(10m)	0.0059	0.002	2.745	0.006	0.002	0.010
지면온도(°C)	5.4486	0.005	1176.402	0.000	5.439	5.458
강수여부_0	13.2820	0.002	6876.425	0.000	13.278	13.286
강수여부_1	12.8749	0.008	1592.696	0.000	12.859	12.891

선형회귀모형

```
%%time
lin_reg = LinearRegression()
model_lin = lin_reg.fit(X_train, y_train)
y_pred = lin_reg.predict(X_test)
rms = np.sqrt(mean_squared_error(y_test, y_pred))
rms
```

```
Wall time: 932 ms

2.588573861396947

훈련용: 0.9341058319370741
검증용: 0.9340888232034034
```

의사결정나무 모형

```
%%time
tree_reg = DecisionTreeRegressor(criterion = 'squared_error',
                                 splitter='best',
                                 max_depth=14,
                                 min_samples_leaf=10,
                                 random_state=0)
model_tree = tree_reg.fit(X_train, y_train)
y_pred = tree_reg.predict(X_test)
rms = np.sqrt(mean_squared_error(y_test, y_pred))
rms
```

```
Wall time: 13.2 s
```

```
1.1453323535329565
```

```
훈련용: 0.9884563161569282
검증용: 0.9870966938828492
```

랜덤포레스트

```
%%time
forest_reg=RandomForestRegressor(random_state=0, n_jobs=-1) #CPU full 사용
model_forest = forest_reg.fit(X_train, y_train)
y_pred=forest_reg.predict(X_test)
rms=np.sqrt(mean_squared_error(y_test, y_pred))
rms
```

```
CPU times: total: 1h 49min 8s
Wall time: 7min 29s
```

```
0.9814109871679801
```

```
훈련용: 0.9986670825705177
검증용: 0.9905258605204281
```

하이퍼파라미터 튜닝

```
%%time
from sklearn.model_selection import RandomizedSearchCV
from scipy.stats import randint
param_distribs={
 'n_estimators': randint(low=1, high=100),
 'max_features': randint(low=1, high=8),
 }
forest_reg=RandomForestRegressor(random_state=0, n_jobs=-1) #CPU full 사용
rnd_search=RandomizedSearchCV(forest_reg, param_distributions=param_distribs, cv=5, random_state=0)
rnd_search.fit(X_train, y_train)

cvres=rnd_search.cv_results_
for mean_score, params in zip(cvres["mean_test_score"],cvres["params"]):
    print(np.sqrt(mean_score), params)
```

```
0.9947586337229204 {'max_features': 5, 'n_estimators': 48}
0.9950150844322293 {'max_features': 6, 'n_estimators': 65}
0.9942885859458427 {'max_features': 4, 'n_estimators': 68}
0.9920912719210961 {'max_features': 2, 'n_estimators': 84}
0.9948944887288426 {'max_features': 6, 'n_estimators': 37}
0.9952243660888737 {'max_features': 7, 'n_estimators': 89}
0.9857635677303628 {'max_features': 1, 'n_estimators': 13}
0.9936172023084505 {'max_features': 3, 'n_estimators': 66}
0.9950939206446904 {'max_features': 7, 'n_estimators': 40}
```

최적의 파라미터 조합: **max_features 7개, n_estimators 89개**

하이퍼파라미터 튜닝

```
%%time
model_RSCV_forest=rnd_search.best_estimator_
y_pred=model_RSCV_forest.predict(X_test)
rms=np.sqrt(mean_squared_error(y_test, y_pred))
rms
```

```
CPU times: total: 1min 53s
Wall time: 8.01 s
0.9766532213377118
```

```
훈련용: 0.9986621252805558
검증용: 0.9906174969025778
```

04

데이터 분석2

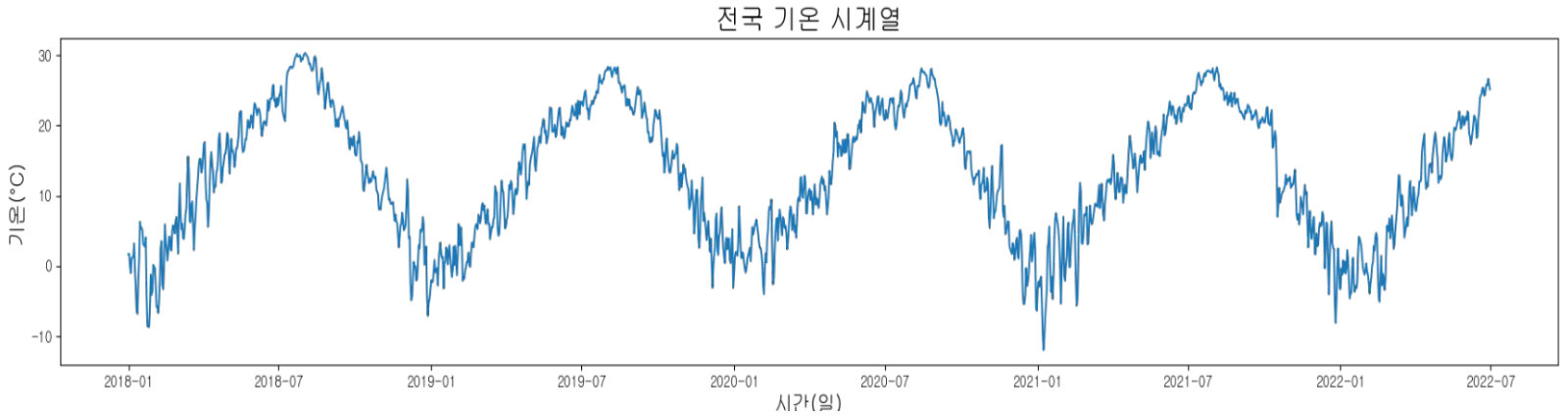
Python을 이용한 시계열 분석

데이터 분할

용도	기준	기간
학습용	2022-06-30 이전	4년 6개월
테스트용	2022-07-01 이후	6개월

시계열 분석

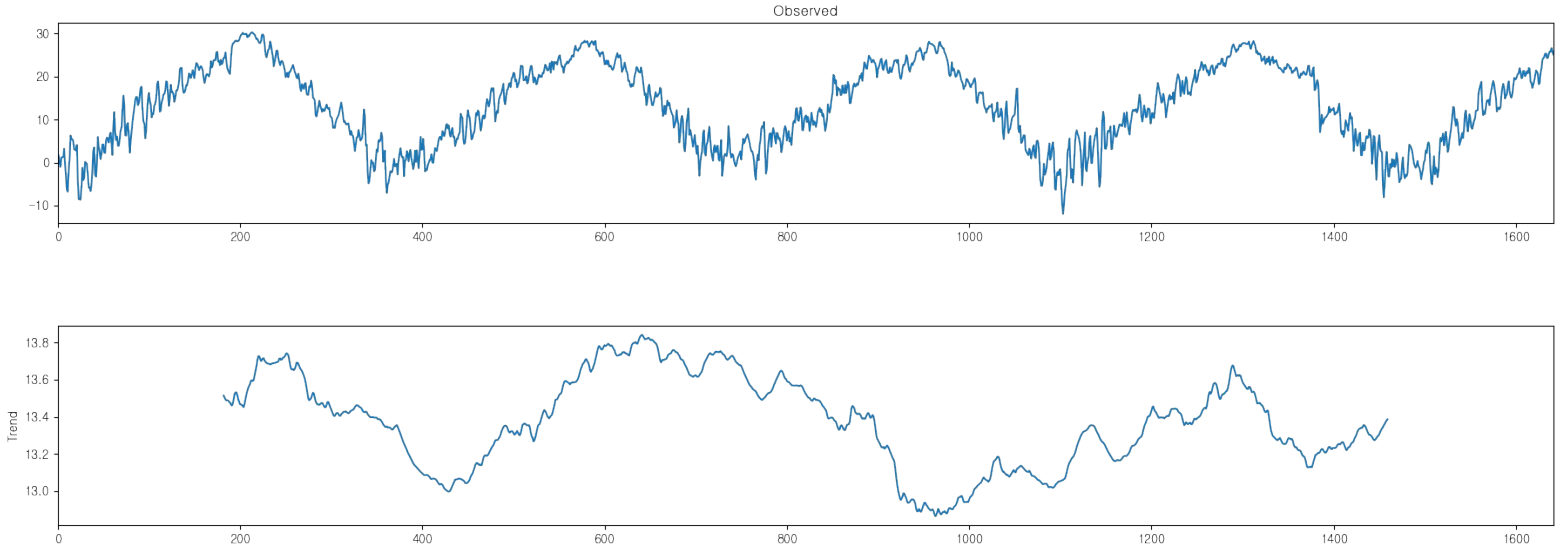
시계열 그래프



우리가 평소 아는 것과 같이 일년을 주기로 계절성을 보임

시계열 분석

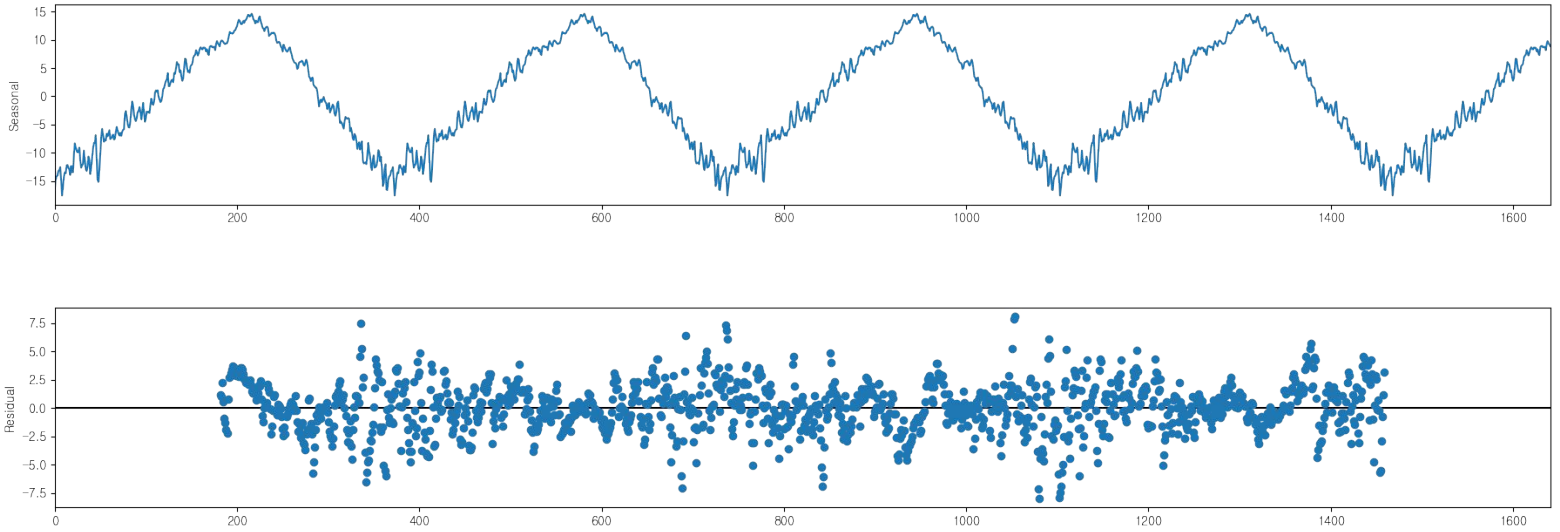
시계열 분해



Trend(추세)에서 패턴이 보임

시계열 분석

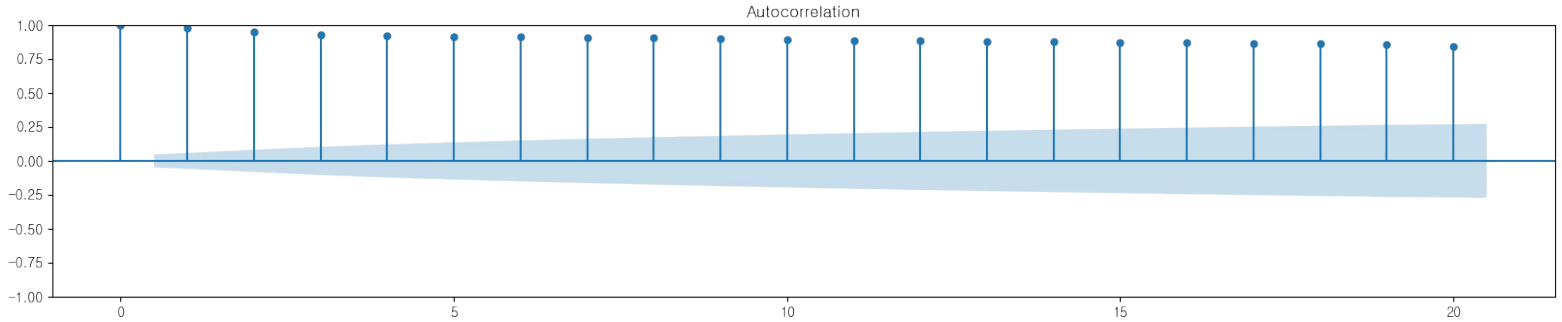
시계열 분해



Seasonal(계절성)과 Residual(불규칙 요소)에서 패턴이 보임

시계열 분석

정상성 확인



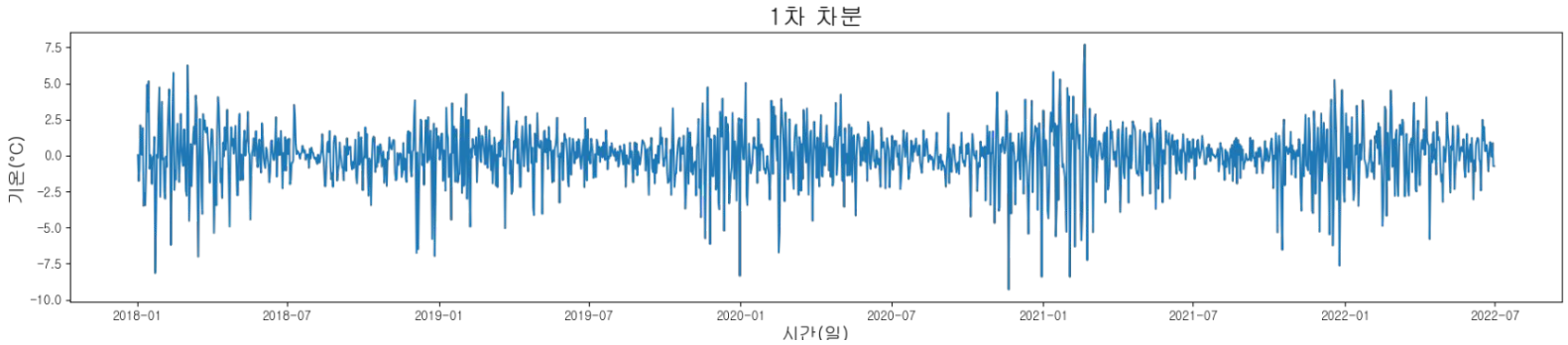
ACF 그래프 모양: **아주 서서히 감소**
ADF 검정 결과: p-value = **0.078**

H0: 정상 시계열이 아니다. -> 채택
H1: 정상 시계열이다.

ACF 그래프의 모양과 ADF 검정 결과를 보면 정상성을 만족하지 않음

시계열 분석

차분



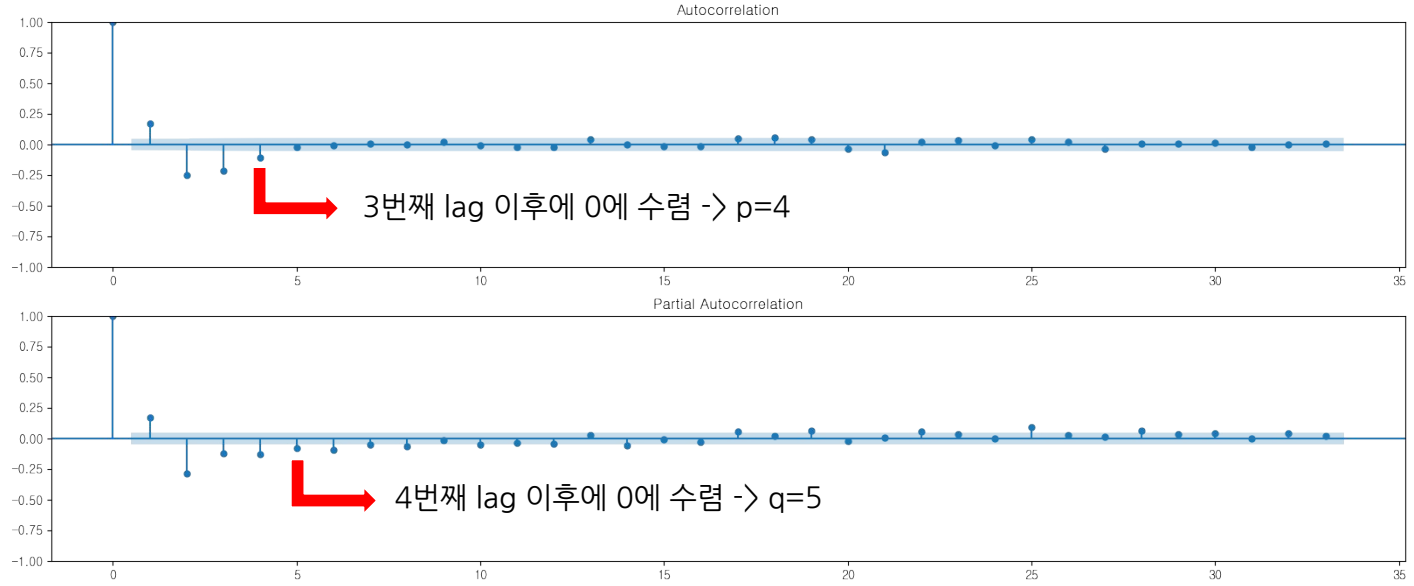
ACF 그래프 모양: **규칙성이 완화 됨**
ADF 검정 결과: p-value = **1.99e-10**

H0: 정상 시계열이 아니다. -> 기각
H1: 정상 시계열이다.

ACF 그래프의 모양과 ADF 검정 결과를 보면 정상성을 만족함

시계열 분석

ACF와 PACF



ARIMA(4,1,5) 를 채택

시계열 분석

ARIMA(4,1,5) 모형

	coef	std err	z	P>|z|	[0.025	0.975]
ar.L1	-0.2912	0.066	-4.399	0.000	-0.421	-0.161
ar.L2	1.4456	0.055	26.180	0.000	1.337	1.554
ar.L3	0.3298	0.062	5.340	0.000	0.209	0.451
ar.L4	-0.5131	0.049	-10.489	0.000	-0.609	-0.417
ma.L1	0.4039	0.066	6.147	0.000	0.275	0.533
ma.L2	-1.7580	0.047	-37.273	0.000	-1.850	-1.666
ma.L3	-0.8049	0.088	-9.111	0.000	-0.978	-0.632
ma.L4	0.8020	0.044	18.254	0.000	0.716	0.888
ma.L5	0.3892	0.029	13.548	0.000	0.333	0.446
sigma2	2.8891	0.072	40.124	0.000	2.748	3.030

모든 p-value 값이 0.05보다 작음 -> 유의한 모형

시계열 분석

ARIMA(4,1,5) 모형

	coef	std err	z	P>|z|	[0.025	0.975]
ar.L1	-0.2912	0.066	-4.399	0.000	-0.421	-0.161
ar.L2	1.4456	0.055	26.180	0.000	1.337	1.554
ar.L3	0.3298	0.062	5.340	0.000	0.209	0.451
ar.L4	-0.5131	0.049	-10.489	0.000	-0.609	-0.417
ma.L1	0.4039	0.066	6.147	0.000	0.275	0.533
ma.L2	-1.7580	0.047	-37.273	0.000	-1.850	-1.666
ma.L3	-0.8049	0.088	-9.111	0.000	-0.978	-0.632
ma.L4	0.8020	0.044	18.254	0.000	0.716	0.888
ma.L5	0.3892	0.029	13.548	0.000	0.333	0.446
sigma2	2.8891	0.072	40.124	0.000	2.748	3.030

Predictions

2022-07-01	23.701025
2022-07-02	24.629320
2022-07-03	24.254841
2022-07-04	25.807694
2022-07-05	22.986242
...	
2022-12-27	-5.580611
2022-12-28	-5.448437
2022-12-29	-3.310956
2022-12-30	-4.221506
2022-12-31	-2.230451

MAPE: 62.04%

모든 p-value 값이 0.05보다 작음 -> 유의한 모형

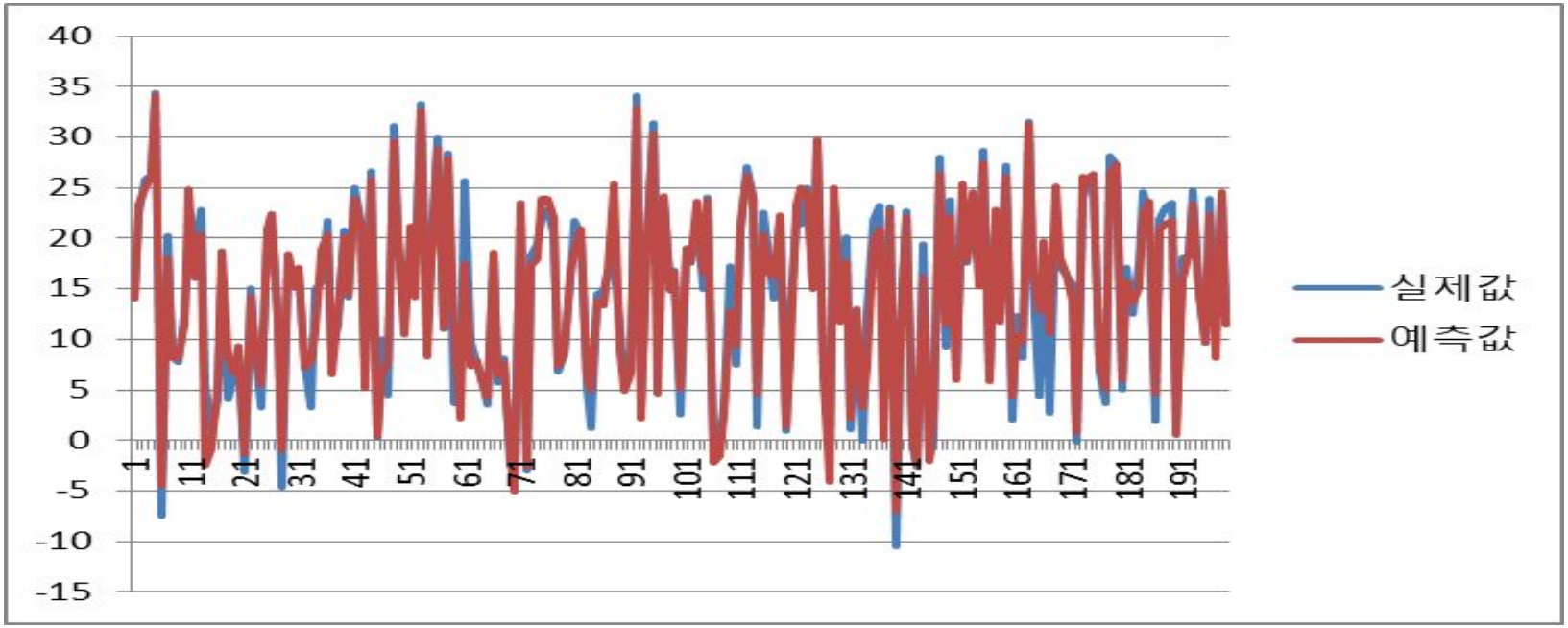
05

결과 검증 및 결론

당일 기온 예측 모델링 결과 요약

모형	선형회귀	의사결정나무	랜덤포레스트	RSCV	인공 신경망
평균제곱오차	3.4	3.15	2.8	2.73	2.95
훈련용 설명력	89%	91%	99%	99%	-
검증용 설명력	89%	90%	92%	93%	91%

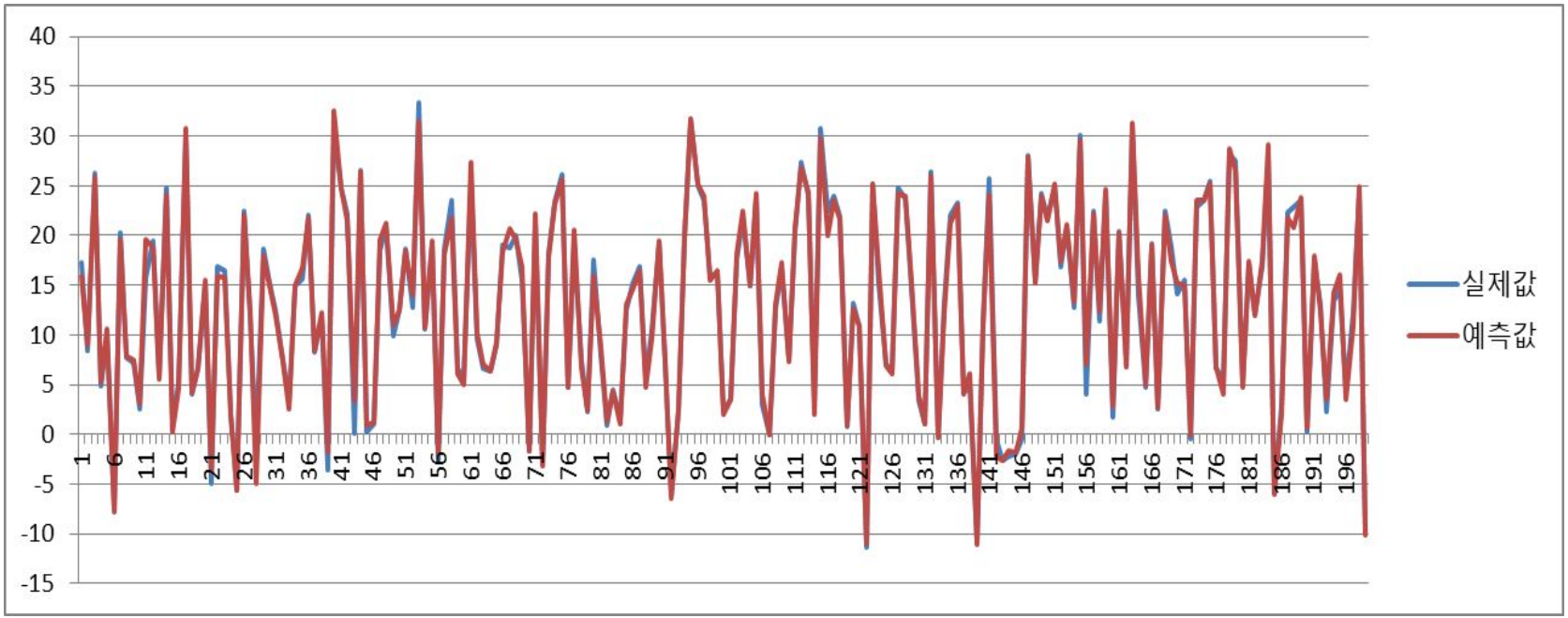
당일 기온 예측
랜덤포레스트 - 하이퍼파라미터 튜닝



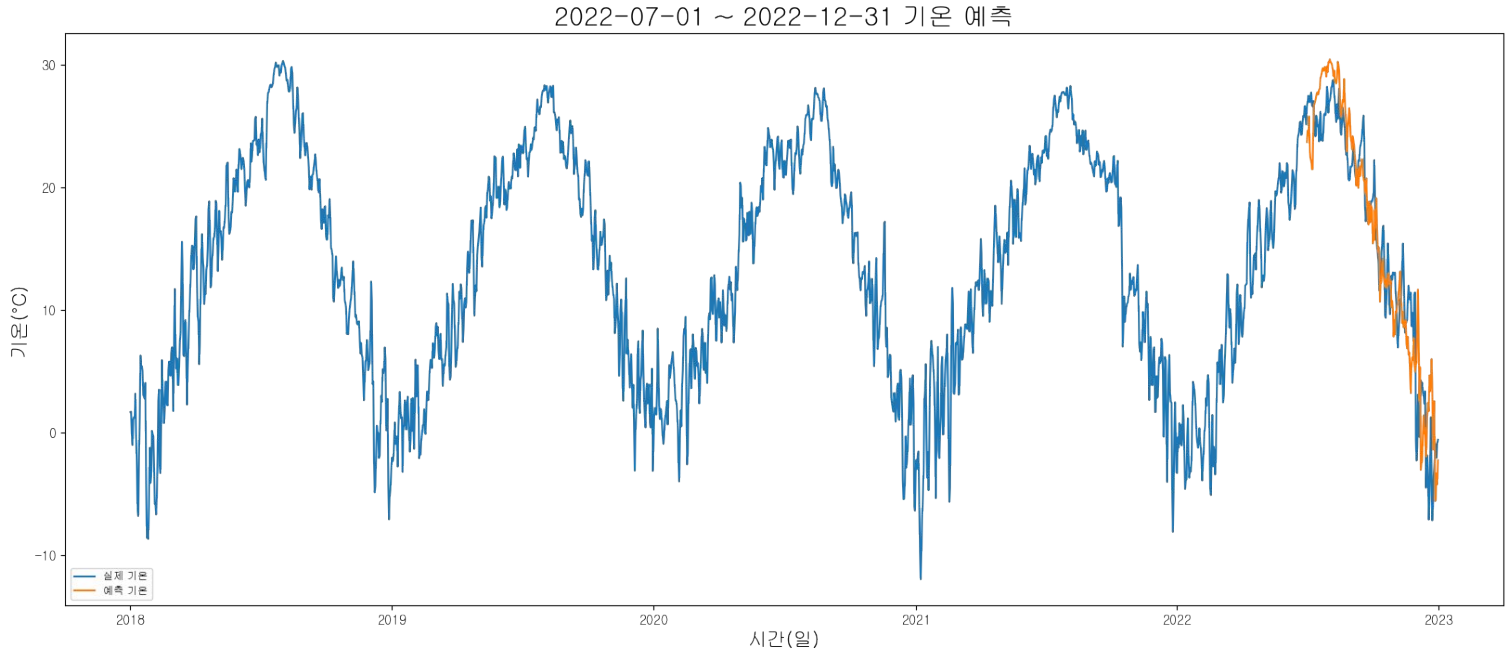
익일 기온 예측 모델링 결과 요약

모형	선형회귀	의사결정나무	랜덤포레스트	RSCV	인공 신경망
평균제곱오차	2.589	1.145	0.98	0.977	-
훈련용 설명력	93.4%	98.8%	99.8%	99.87%	-
검증용 설명력	93.4%	98.7%	99%	99.06%	-

익일 기온 예측
랜덤포레스트 - 하이퍼파라미터 튜닝



6개월 기온 예측 - 시계열 분석



MAPE: 62.04%

한계점 및 개선 과제

- 분석에 사용할 시간이 충분하지 않아 더 많은 파생변수 조합과 다양한 파라미터 조정을 하지 못했다.
- 기온에 영향을 줄 것이라고 예상되는 요인들이 더 있었으나, 데이터를 구하지 못했다.
- 더 다양한 독립변수들을 사용하여 더 좋은 모델을 구축 할 필요가 있어보인다.
- 범주형 변수를 모형 예측에 활용할 수 있는 방법을 찾을 필요가 있다.
- 시계열 예측의 정확도를 더 높일 필요가 있다.

THANK YOU!

Do you have any questions?