1 데이터 설명 및 출처

1 Airkorea

출처 : https://www.airkorea.or.kr/web/(에어코리아)

날짜(월-일-시)	PM10(㎍/㎥)	PM2.5(㎍/㎥)	오 존(ppm)	이산화질소(ppm)	일산화탄소(ppm)	아황산가스(ppm)
01-01-01	32	15	0.019	0.018	0.6	0.002
01-01-02	25	15	0.018	0.019	0.6	0.003
01-01-03	25	18	0.017	0.017	0.6	0.002
01-01-04	25	15	0.009	0.023	0.7	0.003
01-01-05	28	15	0.014	0.019	0.6	0.002
01-01-06	22	17	0.011	0.021	0.6	0.003
01-01-07	28	18	0.005	0.025	0.6	0.003
01-01-08	26	9	0.004	0.026	0.7	0.002
01-01-09	28	13	0.007	0.026	0.7	0.003
01-01-10	21	19	0.012	0.025	0.7	0.003

- 미세먼지 및 대기질에 관한 데이터 (세종시)
  <1달씩 24개의 파일>
- 1시간 단위, 2021, 2022 2년의 데이터. **17520개, 7개의 변수**
- **PM10 - 미세먼지**

좋음	보통	나쁨	매우 나쁨
0~30	31~80	81~150	151 이상

2 기상청 날씨데이터 서비스 기상자료개방포털

출처 : https://data.kma.go.kr/cmmn/main.do(기상자료개방포털)

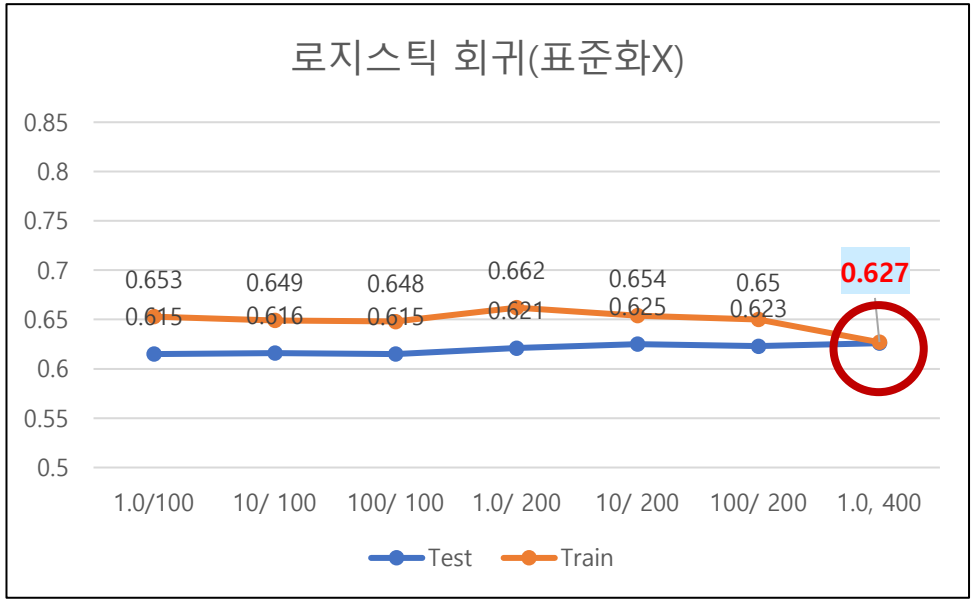
지점	지점명	일시	기온(℃)	강수량(mm)	풍속(m/s)	풍향(16방위)	습도(%)	증기압(hPa)	이슬점온도(℃)	현지기압(hPa)	적설(cm)	시정(10m)	지면온도(℃)
239	세종	2022-01-01 00:00	-7.9	NaN	0.2	0.0	68.0	2.3	-12.7	1021.6	NaN	1588.0	-7.1
239	세종	2022-01-01 01:00	-8.0	NaN	0.4	0.0	65.0	2.2	-13.4	1021.8	NaN	1515.0	-7.3
239	세종	2022-01-01 02:00	-9.2	NaN	1.2	320.0	70.0	2.1	-13.6	1022.3	NaN	2020.0	-7.7
239	세종	2022-01-01 03:00	-9.1	NaN	0.3	0.0	74.0	2.3	-12.8	1022.2	NaN	2001.0	-8.0
239	세종	2022-01-01 04:00	-9.3	NaN	0.0	0.0	76.0	2.3	-12.7	1021.7	NaN	2056.0	-8.3
239	세종	2022-01-01 05:00	-9.8	NaN	0.3	0.0	77.0	2.3	-13.0	1021.4	NaN	2145.0	-8.4
239	세종	2022-01-01 06:00	-10.6	NaN	1.3	230.0	78.0	2.1	-13.6	1021.2	NaN	1705.0	-8.7
239	세종	2022-01-01 07:00	-10.4	NaN	0.0	0.0	79.0	2.2	-13.3	1021.5	NaN	1727.0	-8.6
239	세종	2022-01-01 08:00	-10.3	NaN	0.2	0.0	81.0	2.3	-12.9	1021.6	NaN	1798.0	-8.4
239	세종	2022-01-01 09:00	-7.9	NaN	0.8	110.0	65.0	2.2	-13.3	1021.7	NaN	4109.0	-4.9

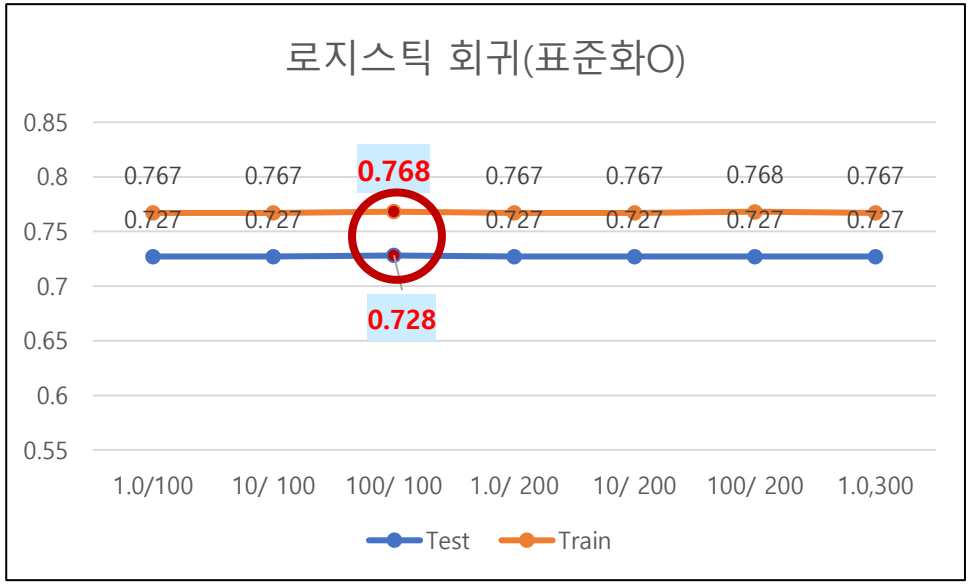
- 기상에 관한 데이터 (세종시) <1년씩 2개의 파일>
- 1시간 단위, 2021, 2022 2년의 데이터. **17520개, 15개의 변수**
- 미세먼지 값을 4개로 범주화하여 분석 진행

- 최종적으로 2019, 2020, 2021, 2022 4년의 데이터. **31440개, 16개의 변수**

2-1 로지스틱 회귀분석

1 로지스틱 회귀분석 결과



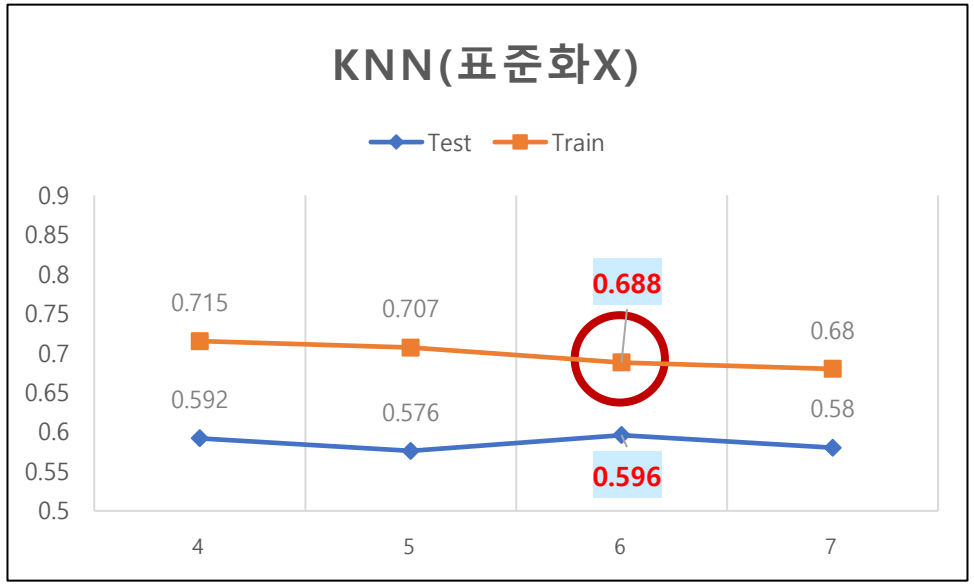


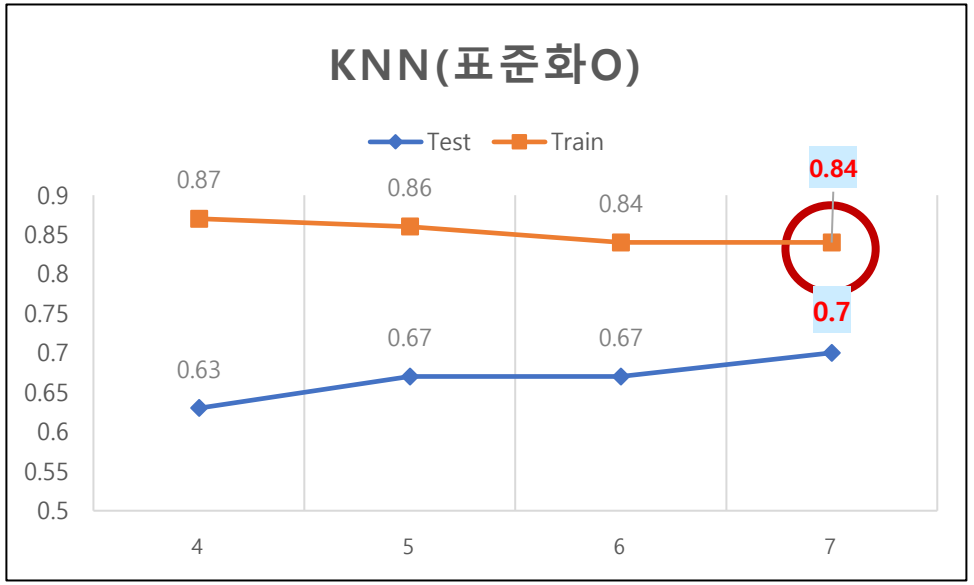
- Ex) 1.0(C) / 100(max_iter) 하이퍼파라미터
- 표준화X 그리드서치로 확인한 C= 1.0, max_iter= 400일 때 과적합 X, 성능도 다른 설정값과 비슷함.
- 표준화O 그리드서치와 다른 설정값과 차이가 거의 없음. C=100, max_tier=100일 때 성능이 가장 좋음

  -> 정확도를 높이기 위해 다른 모델도 돌려봄

2-2 KNN

2 KNN 결과

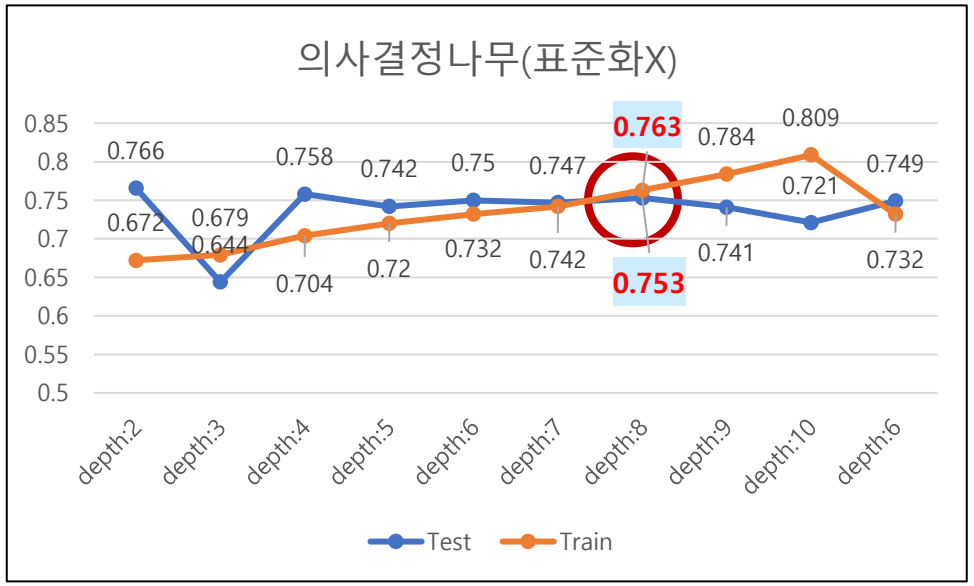


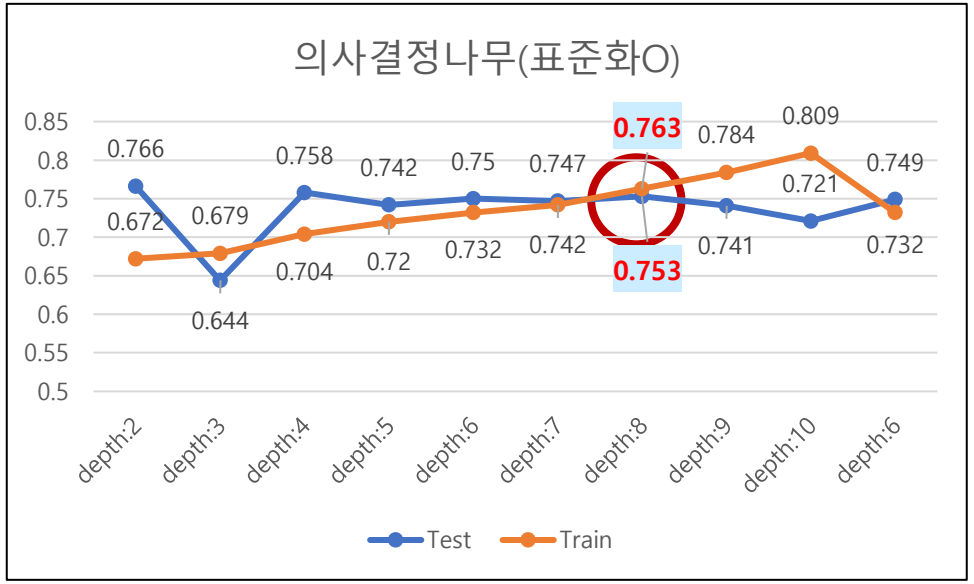


- Ex.) 4,5,6,7 n_neighbors의 개수 하이퍼파라미터
- 성능은 표준화 했을 때 가장 좋았지만 KNN은 과적합이 발생했다고 판단.
- 표준화X 그리드 서치 -> 6개의 n_neighbors
- 표준화O 그리드 서치 -> 7개의 n_neighbors

2-3 의사결정나무

3 의사결정나무 결과



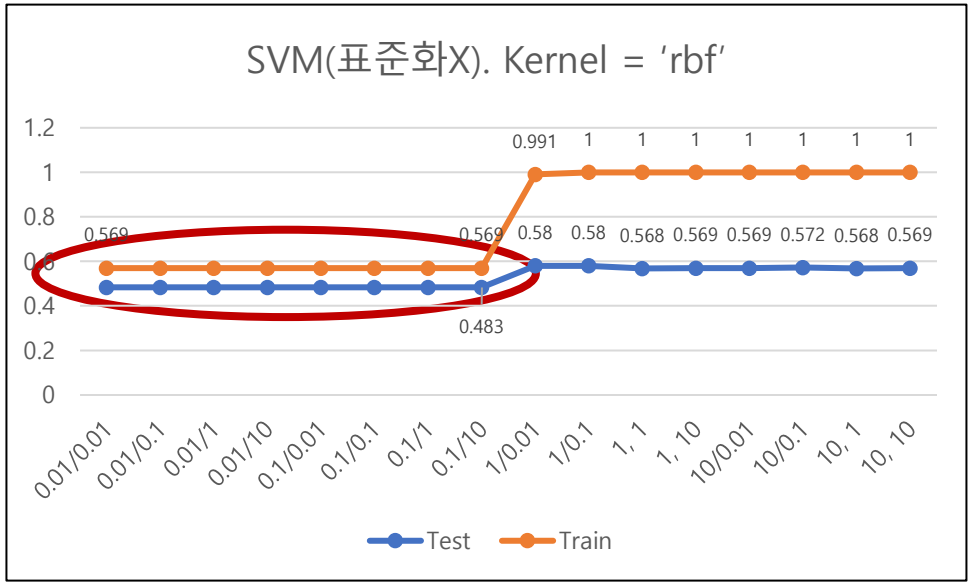


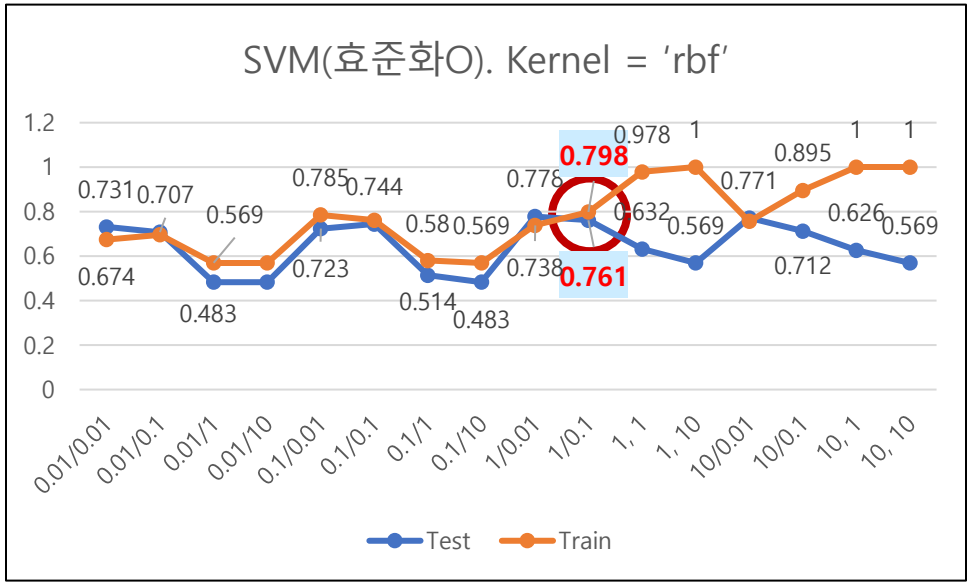
- Ex) depth:2 -> tree의 깊이 하이퍼파라미터
- 성능은 표준화 했을 때와 표준화 하지 않았을 때 모두 동일
- 표준화X&O depth가 낮을때는 과소적합. 이후 depth 8부터 과소적합 문제가 해결
- Depth 8 : Train - 0.763 / Test - 0.753 정확도 가장 높음

  -> 현재까지 가장 성능이 좋음, 이후 SVM과 비교

2-4 SVM. Kernel = 'rbf'

4 SVM 결과

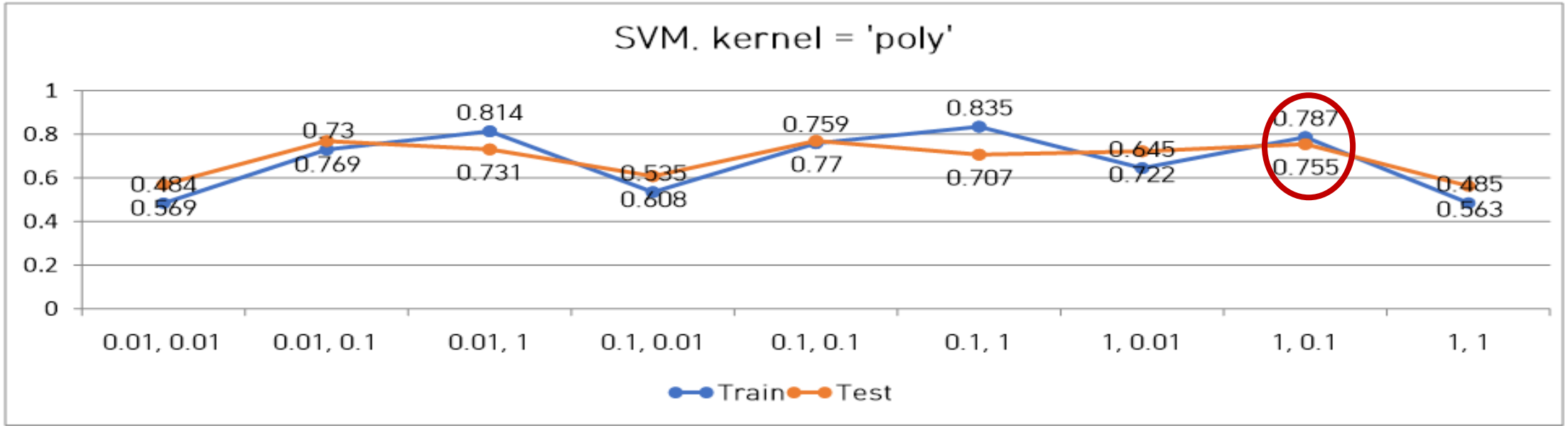




- 하이퍼파라미터 : kenel = 'rbf', C, gamma
- 표준화O: C=1, gamma: 0.1 했을때 train: 0.798/ test 0.761
- Hyperparameter의 설정을 위와 같이 했을 때 가장 좋은 성능을 보임
- 이후 kernel 을 rbf 가 아닌 다항식 커널 poly 를 사용 후 비교

2-5 SVM. Kernel = ‘poly’

5 SVM 결과(표준화 O)

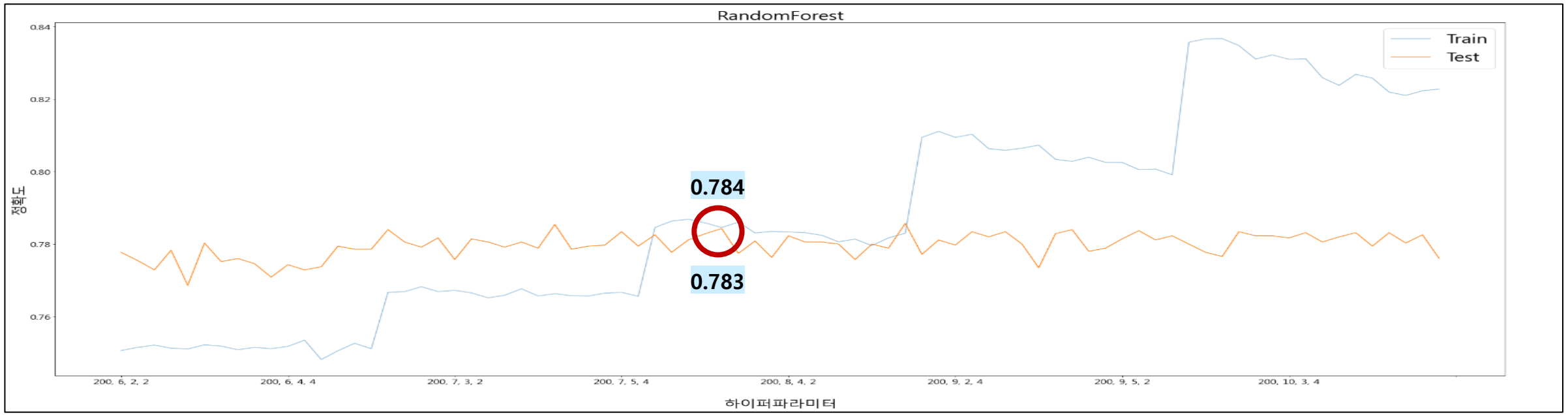


- 하이퍼파라미터 : kenel = ‘poly’, C, gamma
- 표준화O: kenel = ‘poly’, C = 1, gamma: 0.1 했을때 train: 0.787, test 0.755
- Hyperparameter의 설정을 위와 같이 했을 때 가장 좋은 성능을 보임

3-1 RandomForest

1 RandomForest 결과

(1. n_estimators=200 2. max_depth(6, 7, 8, 9, 10) 3. min_samples_leaf(2, 3, 4, 5) 4. min_samples_split(2, 3, 4, 5)

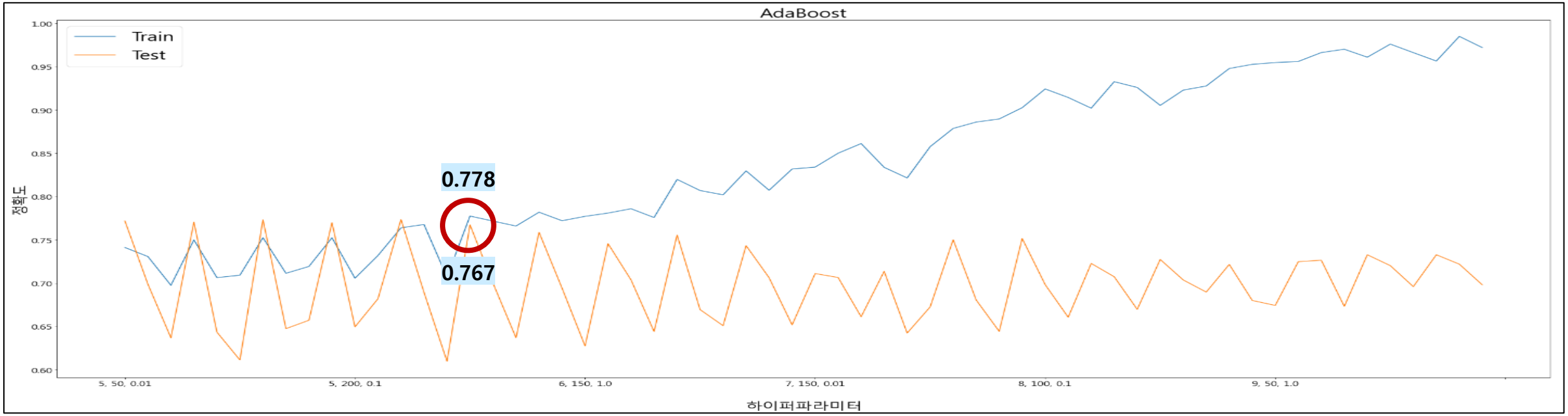


- 특이사항 Max_depth에 따라서 Train 학습 정도가 눈에 띄게 달라짐
- Max_depth가 커지면서 과소적합에서 과대적합의 문제로 바뀜
- n_estimators: 200, max_depth: 8, min_samples leaf: 2, min_samples_split: 2 일 때
- Train: 0.784, Test: 0.783 으로 가장 좋은 성능

3-2 AdaBoost

2 AdaBoost 결과

(1. max_depth(5, 6, 7, 8, 9) 2. n_estimators(50, 100, 150, 200) 3. learning_rate = (0.01, 0.1, 1)

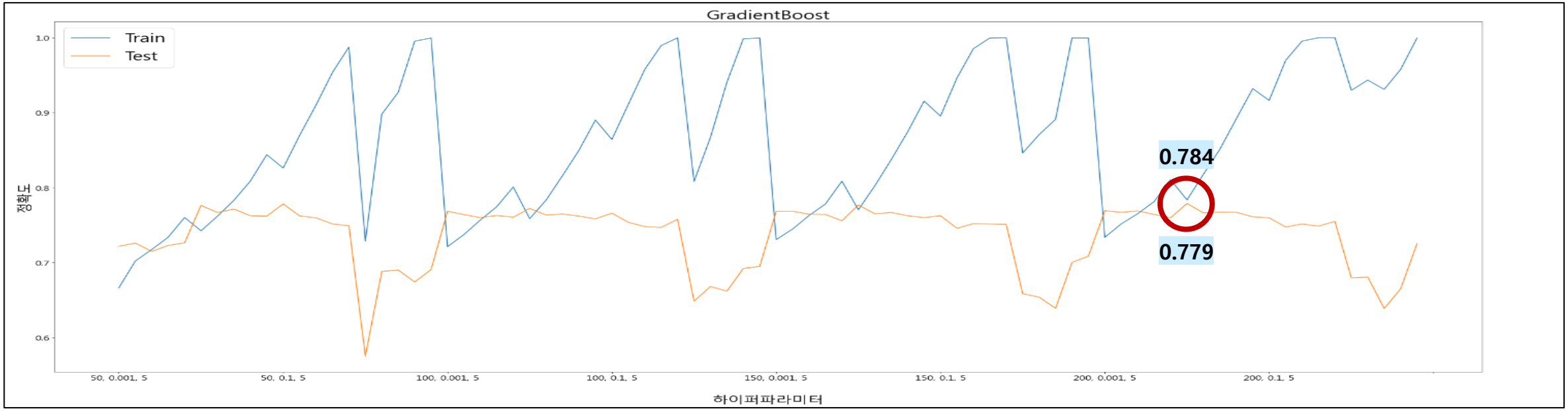


- 특이사항 learning_rate에 따라 패턴 있고, max_depth와 n_estimators는 커질수록 과대적합 발생
- max_depth: 8, n_estimators: 50, learning_rate: 0.01 일 때 Train: 0.778, Test: 0.767 으로 가장 좋은 성능

3-3 GradientBoost

3 GradientBoost 결과

(1. n_estimators(50, 100, 150, 200) 2. learning_rate = (0.001, 0.01, 0.1, 1) 3. max_depth(5, 6, 7, 8, 9)

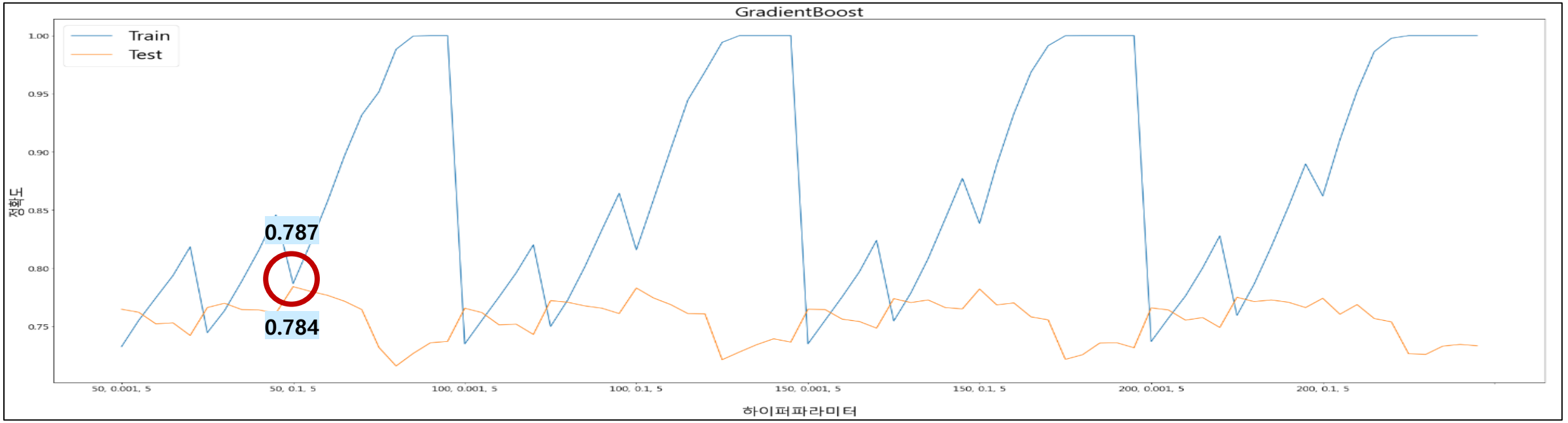


- 특이사항 n_estimators에 따라 패턴이 있고, max_depth가 커짐에 따라 과대적합이 발생
- n_estimators: 200, learning_rate: 0.01, max_depth: 5 일 때 Train: 0.784, Test: 0.779 로 가장 좋은 성능

3-4 XGBoost

4 XGBoost 결과

(1. n_estimators(50, 100, 150, 200) 2. , learning_rate = (0.001, 0.01, 0.1, 1) 3.max_depth(5, 6, 7, 8, 9)



- 특이사항 n_estimators에 따라 패턴이 있고, max_depth가 커짐에 따라 과대적합이 발생
- n_estimators: 50, learning_rate: 0.1, max_depth: 5 일 때 Train: 0.786, Test: 0.784 로 가장 좋은 성능

4 결론 및 한계점

결론

- 앙상블(XGBoost) 기법을 사용하였을 때 약 80퍼 정도의 정확도를 가지고 미세먼지를 예측 가능
- 기여도가 높은 날씨데이터인 기온, 이슬점 온도 등 날씨를 보고 미세먼지를 어느정도 예측할 수 있음.
- Grid Search도 진행하였고, 이와 하이퍼파라미터의 관계를 Plot으로 확인.

한계점

- 모델을 학습할 때 2019~2022년도까지 4년치 데이터만 사용 -> 23년도 3월, 6월을 예측할 때 경향 및 추세를 잘 파악 못했을 수도 있음.
- 미세먼지와의 상관관계가 높지 않은 데이터들로 분석을 진행했기 때문에 정확도 증가에 한계가 있음.
- 상대적으로 19년도 20년도에는 코로나 영향을 받아 적은 개수의 값들은 예측에 어려움을 겪음.