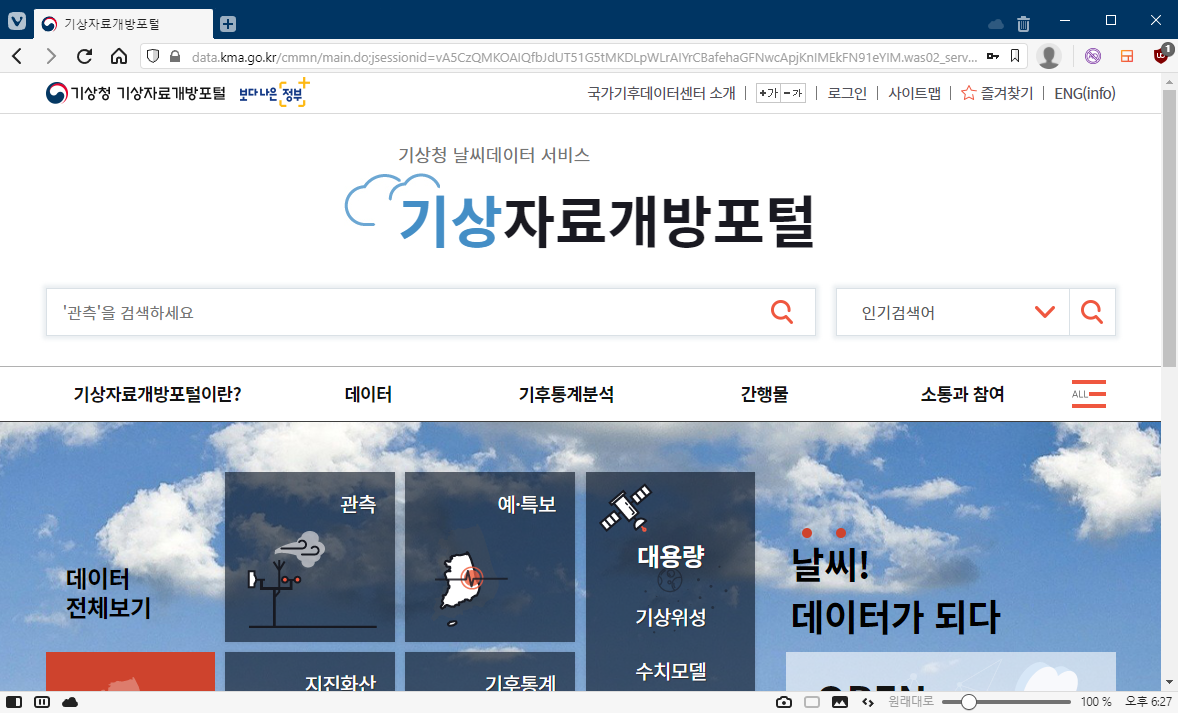
데이터 마이닝 보고서

컴퓨터정보통신공학과 B789029 성정현

**주제** **:** **기상 분류 분석**

**데이터셋 선택 : 기상청 기상자료개방포털** <https://data.kma.go.kr/cmmn/main.do>

**텍스트이(가) 표시된 사진

자동 생성된 설명**

Data - 기후통계분석(조건별통계)

<https://data.kma.go.kr/climate/RankState/selectRankStatisticsDivisionList.do?pgmNo=179>

Target - 기상예보(중기예보)

<https://data.kma.go.kr/data/weatherReport/mrfList.do?pgmNo=646>

**데이터셋 설명:**

20160101~20211127

한 도시의 기상, 강수량, 바람, 습도, 일조일사 5개 데이터셋

전국의 중기예보 1개 데이터셋

한 도시에 총 6개 데이터셋

총 7개 도시 데이터 이용. 42개 데이터셋 merge

**데이터 소개**

기온

**테이블이(가) 표시된 사진

자동 생성된 설명**

평균기온, 최고기온, 최저기온, 일교차 (기온의 모든 단위는 섭씨)

최고기온시각, 최저기온시각 (시:분)

습도

테이블이(가) 표시된 사진

자동 생성된 설명

평균습도, 최저습도 (단위 : %rh)

일조일사

테이블이(가) 표시된 사진

자동 생성된 설명

일조합, 일조율, 일사합 ( 해가 떠있던 시간, 하루 중 해가 떠있던 시간비, 태양 복사열의 양)

바람

테이블이(가) 표시된 사진

자동 생성된 설명

평균풍속, 최대풍속, 최대순간풍속 (m/s)

최대풍속시각, 최대순간풍속시각 (시:분)

최대풍속풍향, 최대순간풍속풍향 (degree)

강수량

테이블이(가) 표시된 사진

자동 생성된 설명

강수량, 1시간최다강수량 (mm)

1시간최다강수량시각(시:분

중기예보

텍스트, 낱말맞추기게임이(가) 표시된 사진

자동 생성된 설명

발표시각 : 예보를 발표한 시각

예보시각 : 예보가 발현될 시각

예보 : 예보시각에 발생할 날씨

**데이터 전처리**

**텍스트, 스크린샷, 실내, 여러개이(가) 표시된 사진

자동 생성된 설명**

‘지점번호’와 같이 필요 없는 열을 제거

‘\t최고기온시각’와 같은 공백있는 열 이름을 모두 수정

테이블이(가) 표시된 사진

자동 생성된 설명

다음과 같이 변경되었다.

열 수정을 거친 [기온, 강수량, 바람, 습도, 일조일사] 5개 데이터셋을 merge



**target**

텍스트, 낱말맞추기게임이(가) 표시된 사진

자동 생성된 설명

발표시각은 예보 발표 시각으로 예보가 발생하는 시각이 아님 -> 필요 없으므로 제거

지역은 위에서 전처리한 데이터셋과 같은 지역으로 설정해야함 -> 여기서는 ‘서울’로 필터링

테이블이(가) 표시된 사진

자동 생성된 설명

다음과 같이 변경

테이블이(가) 표시된 사진

자동 생성된 설명

‘예보시각’을 기준으로 정렬(sort)한 모습

2016-01-01에 예보가 4개

2016-01-02에 예보가 6개가 있다.

앞서 전처리한 데이터셋은 하루에 하나의 데이터가 존재한다.

따라서 예보도 하루에 하나만 있어야 병합이 가능하다.

날씨는 일반적으로 하루 중 가장 나쁜 날씨를 예보로 사용한다.

대표 날씨를 선정하기 위해 [맑음 < 구름 < 흐림 < 소나기 < 비 < 눈] 의 가중치를 두었다.

텍스트이(가) 표시된 사진

자동 생성된 설명

텍스트이(가) 표시된 사진

자동 생성된 설명각 날짜마다 예보들을 수집하고

예보들의 가장 나쁜 날씨를 선택하게 했다.

ex)구름조금, 구름많음, 흐리고 눈 -> 흐리고 눈



하루에 하나의 예보로 변경되었다.

이제 앞서 전처리한 데이터셋과 예보 데이터셋을 병합했다.

텍스트, 실내, 스크린샷, 여러개이(가) 표시된 사진

자동 생성된 설명

테이블이(가) 표시된 사진

자동 생성된 설명

다시 위의 과정을 거쳐

부산, 제주, 광주, 충주, 안동, 전주, 대전 7개 도시 데이터셋을 생성했다.

서울을 포함한 총 8개 도시 데이터셋을 합쳐 15106행 데이터셋을 완성했다.



**결측값**

텍스트, 컴퓨터이(가) 표시된 사진

자동 생성된 설명

결측값이 매우 많은 열들이 존재했다.

강수량테이블이(가) 표시된 사진

자동 생성된 설명 데이터를 확인해보았다.

일사합 데이터도 마찬가지로

기상청에서는 소수 첫번째 자리까지 반올림해 표현하고 있다.

0표기는 0.01와 같이 강수량이 존재한다는 말이다.

강수량이 진짜 0이면 결측치로 기록하지 않는다.

따라서 이들을 모두 0으로 채워주었다.

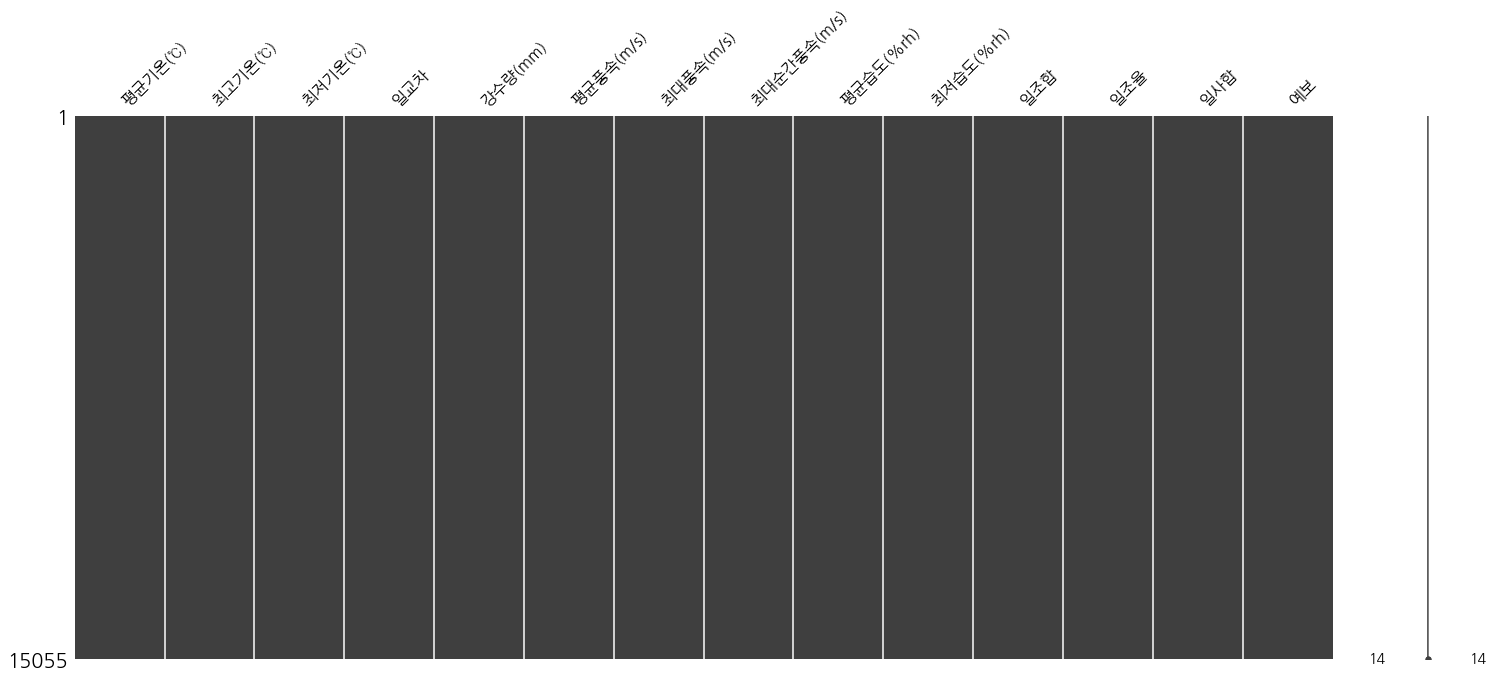
그 외는 모두 드랍해주었다.

**열 제거**

또한 시계열을 이용한 예측이 아닌, 기상 관측 자료만으로 분류 분석을 진행할 것으므로

시계열 데이터는 모두 제거해주었다.

방향 데이터는 각도를 나타내므로 사용에 어려움이 있어서 제거해 주었다.



결론적으로 다음과 같이 열이 줄고, 결측치가 모두 제거되고, 일부 행이 사라졌다.

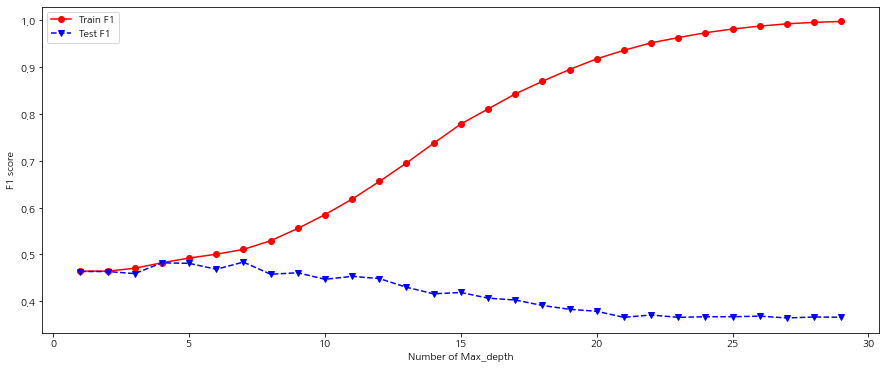
**데이터 변환**

데이터 : 예보를 제외한 기상 관측 자료

타겟 : 예보

**데이터 마이닝**

우선 성능 테스트를 Decision Tree 모델을 사용했다.



최고 0.48의 Test F1 score를 나타냈다.

**아웃라이어**

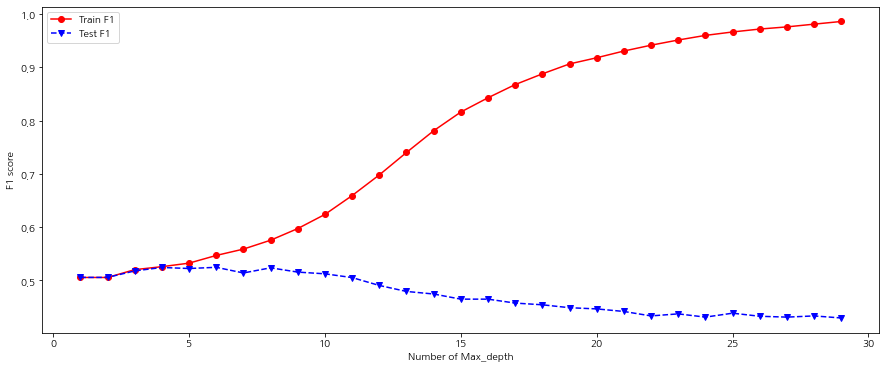
성능 향상을 위해 이상치를 제거해 주었다.

텍스트이(가) 표시된 사진

자동 생성된 설명

다음 코드를 통해 예보 종류 별 이상치를 제거해 주었다.

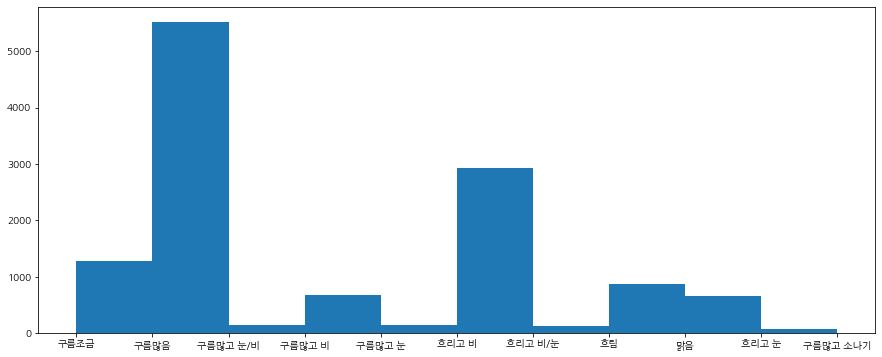
날씨마다 이상 기온에서 갑작스런 기상 이변이 발생하는 경우가 생길 수 있기 때문.



Decision Tree 모델 결과 최고 0.52의 Test F1 score가 나타났다.

소폭 상승했지만 여전히 성능이 좋지 않다.

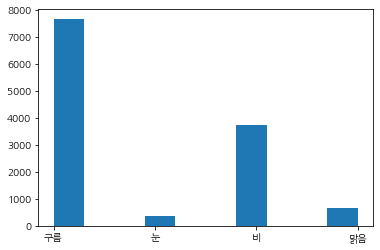
이번에는 Target을 확인해보았다.



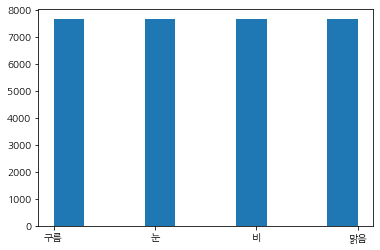
Target의 종류가 많고, 이름이 혼란을 주는 경우도 있고, 데이터셋의 불균형이 매우 심했다.

구름많고 비, 구름많고 눈/비, 흐리고 비, 흐리고 소나기 등등 분별이 어려운 예보가 많다.

이런 비슷한 날씨들을 서로 같은 날씨로 묶어주었다.



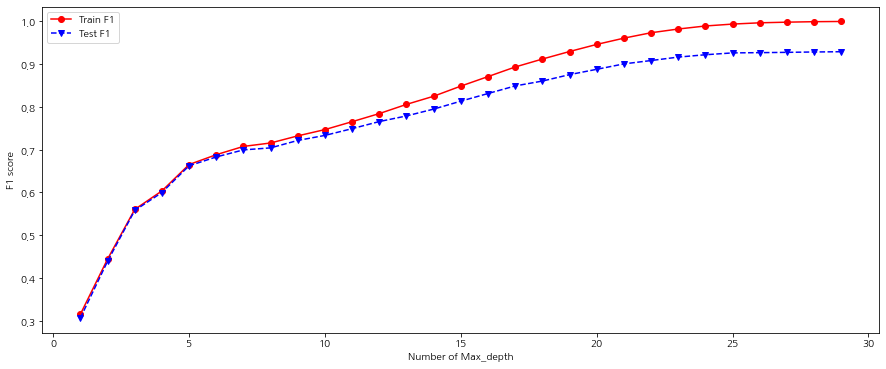
여전한 불균형이 있다.



랜덤오버샘플링을 통해

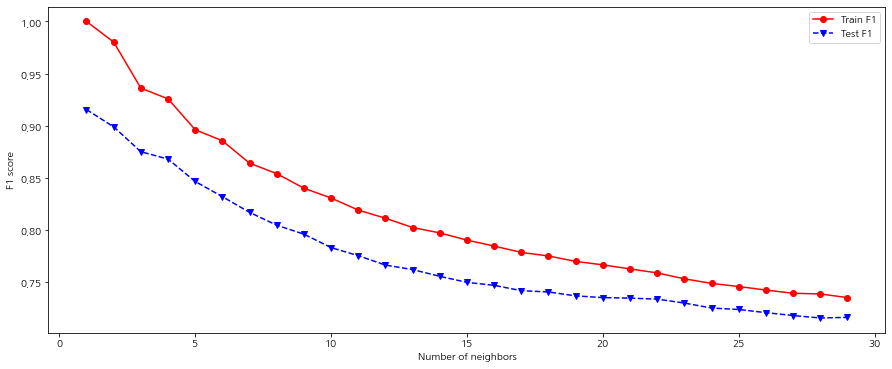
가중치를 균등하게 만들어주었다.

**Decision Tree**



Decision Tree 모델 수행 결과 Test F1 score가 매우 좋아졌다.

**KNN**



KNN의 결과 역시 0.91의 높은 성능을 보여준다.

**SVM**

svm의 분류 모델인 SVC에 다양한 파라미터를 대입해 GridSearch를 해보았다.

텍스트이(가) 표시된 사진

자동 생성된 설명

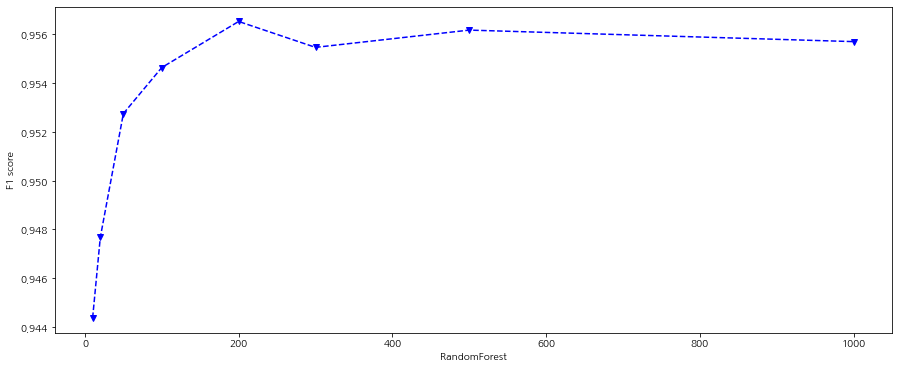


결과는 0.815로 이전 모델들보다 성능이 낮게 나왔다.

**앙상블 모델**

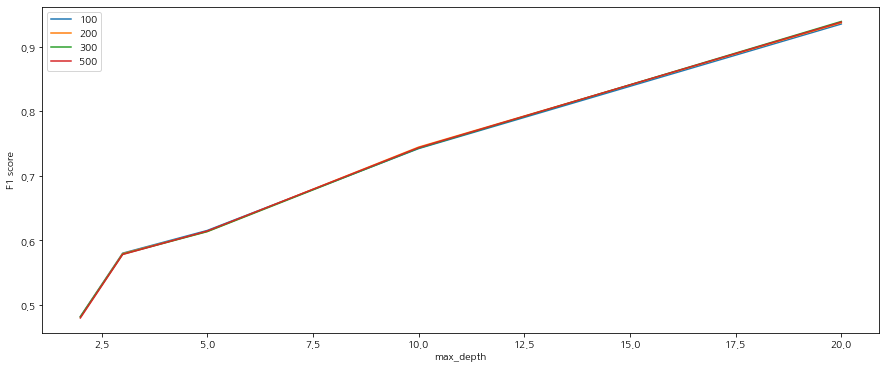
다양한 파라미터를 대입하여 성능을 비교했다.

**Random Forest**

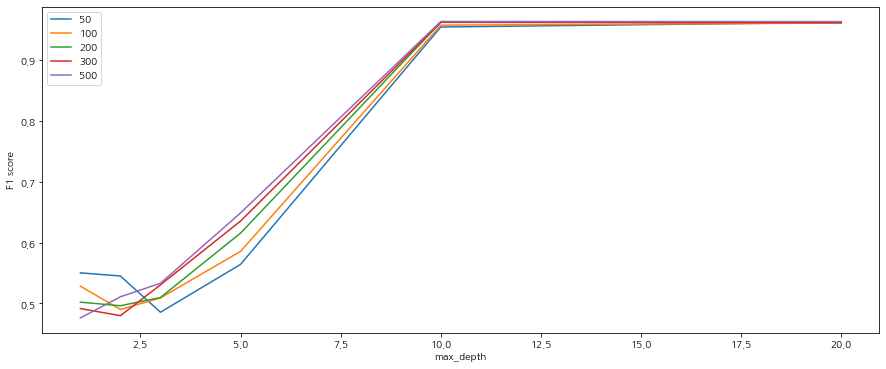
****

앙상블 모델로 랜덤포레스트를 적용했더니 훨씬 더 좋은 결과가 나왔다.

Bagging

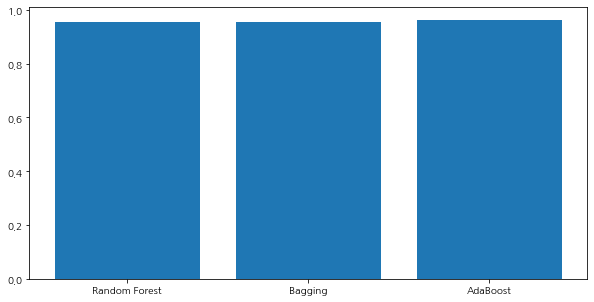
배깅 역시 매우 높은 성능을 보여주었다.

Adaboost



Adaboost 역시 성능이 매우 높았다.

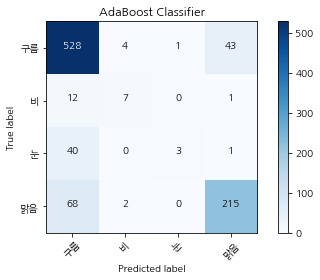
**Test set 결과 평가**

앙상블**** 모델이 기존 모델들보다 고성능이다.

가장 좋은 모델은 AdaBoost

(max\_depth = 10, n\_estimators=500)

F1 score = 0.969

**앞서 사용하지 않은 새로운 도시의 데이터로 성능 평가**

**텍스트, 영수증이(가) 표시된 사진

자동 생성된 설명**

**F1 score = 0.808**로 준수한 성능 보여주었지만, ‘눈’, ‘맑음’이 낮다. 이 날씨들이 표본이 다른 날씨에 매우 적어서 생긴 overfitting 문제이다.

**토론(정리)**

Target의 종류를 줄여서 성능이 크게 증가했지만 아쉬움이 남았다.

우선 일반 PC는 세분화된 Target을 분류하기에는 성능이 부족하다.

또한 태풍과 같은 기상 이변과

산, 평지, 바다와 같은 지형,

그리고 가장 중요한 계절성을 지닌 시계열 데이터를 차분하여 사용해야

더 정확한 기상을 분류 및 예측을 할 수 있다.

현재 내 수준에서는 딥러닝을 수행하기에 무리가 있어서 시계열 데이터를 제외했지만

기회가 된다면 도전해보고 싶다.

데이터셋 분석에 대해서는 참고한 아이디어가 없다.

처음부터 정리되지 않은 별개의 데이터셋이기 때문이다.

Kaggle과 같은 정리된 데이터로 대체할 수도 있었지만,

이해하기 쉬운 컬럼명과 익숙한 지역명으로 진행하고 싶었다.

**참고**

SVM 참고

<https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html>

<https://bskyvision.com/163>

<https://todayisbetterthanyesterday.tistory.com/32>

Ensemble

<https://m.blog.naver.com/slykid/221658390208>