



DACON 동서발전 태양광 발전량 예측 AI 경진대회

- 통계학과 19 이나운 (12기)
- 수학과 18 이두형 (12기)
- 통계학과 19 기다연 (13기)
- 의예과 20 박재찬 (13기)



동서발전 태양광 발전량 예측 AI 경진대회

주제: 시간대별 태양광 발전량 예측

배경:

태양광 발전은 매일 기상 상황과 계절에 따른 일사량의 영향을 받습니다.

이에 대한 예측이 가능하다면 보다 원활한 전력 수급 계획이 가능합니다.

인공지능 기반 태양광 발전량 예측 모델을 만들어주세요.



site_info.csv (발전소 정보) : 발전소 발전용량(MW), 주소, 설치각, 입사각, 위도, 경도

energy.csv (발전소별 발전량) : 당진수상/당진자재창고/당진/울산 태양광 발전량. 1시간 단위 계량
(2018/03/01 00:00~2021/01/31 23:00)

dangjin_fcst_data.csv (당진지역 발전소 동네 예보) : 온도, 습도, 풍속, 풍향, 하늘상태(1,2,3,4)
(2018/03/01 00:00~2021/03/01 23:00)

dangjin_obs_data.csv (당진지역 발전소 기상 관측 자료) : 일시, 기온, 풍속, 풍향, 습도, 전운량(10분위)
(2018/03/01 00:00~2021/01/31 23:00)

ulsan_fcst_data.csv (울산지역 발전소 동네 예보) : 온도, 습도, 풍속, 풍향, 하늘상태(1,2,3,4)
(2018/03/01 00:00~2021/03/01 23:00)

ulsan_obs_data.csv (울산지역 발전소 기상 관측 자료) : 일시, 기온, 풍속, 풍향, 습도, 전운량(10분위)
(2018/03/01 00:00~2021/01/31 23:00)

-> 2021년 2월 1일 00:00부터 2021년 7월 8일 23:00 예측

(1) 외부 관측 데이터 활용 -> 변수 추가 (미세먼지, 이슬점)

- 기상자료개방포털 (종관기상관측)

<https://data.kma.go.kr/data/grnd/selectAsosRltmList.do?pgmNo=36>

- 에어코리아

[에어코리아 \(airkorea.or.kr\)](http://airkorea.or.kr)

참고문헌

- [논문] 기계학습을 이용한 태양광 발전량 예측 및 결함 검출 시스템 개발

scienceon.kisti.re.kr

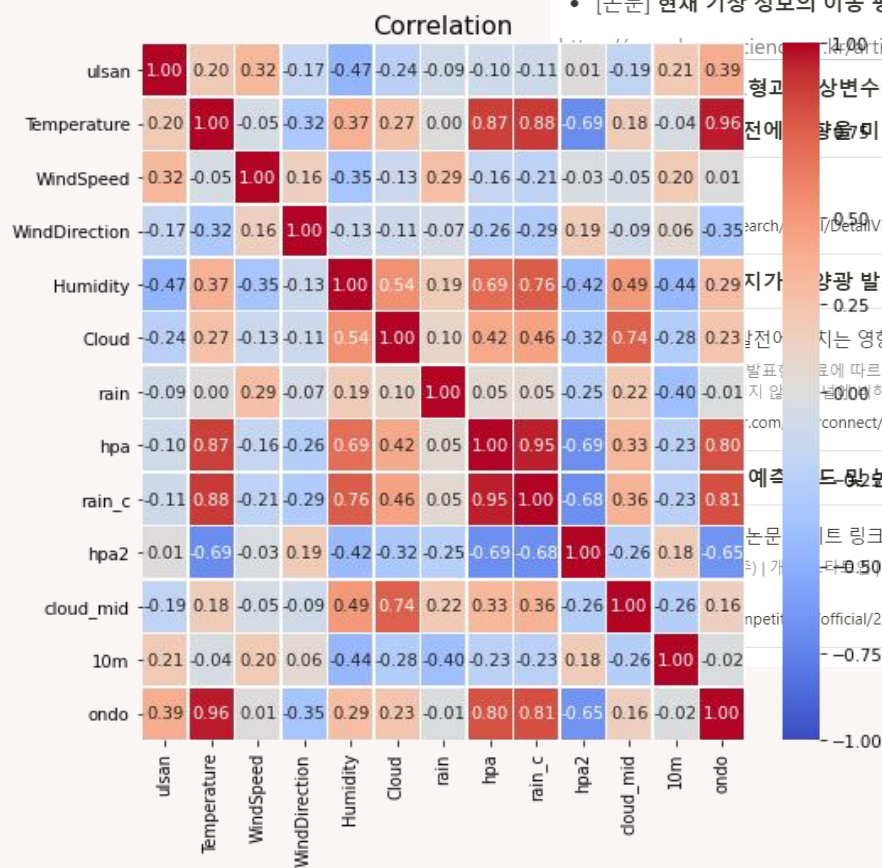
<https://scienceon.kisti.re.kr/srch/selectPORSrchArticle.do?cn=JAKO201631642279260&dbt=NART>

- [논문] 현재 기상 정보의 이동 평균을 사용한 태양광 발전량 예측

scienceon.kisti.re.kr/srch/selectPORSrchArticle.do?cn=JAKO201627035791567.pdf

형고 상변수를 활용한 태양광 발전량 예측 연구

전에 상을 미치는 요소 분석을 통한 연간 발전량 예측에 관한 연구



scienceon.kisti.re.kr/srch/selectPORSrchArticle.do?cn=JAKO201627035791567.pdf

지가 태양광 발전에 미치는 영향

하는 영향 (2)

발표한 내용에 따르면, 미세먼지, 황사 등에 대한 노출이
증가함에 따라 최대 35%의 발전 효율이 감소했다"

예측 모델 논문 사이트 링크

논문 사이트 링크 - Time Series Forecasting

5) | 개월 | 시계열 | NMAE

npetition/official/235720/talkboard/402851?page=1&...

-0.75

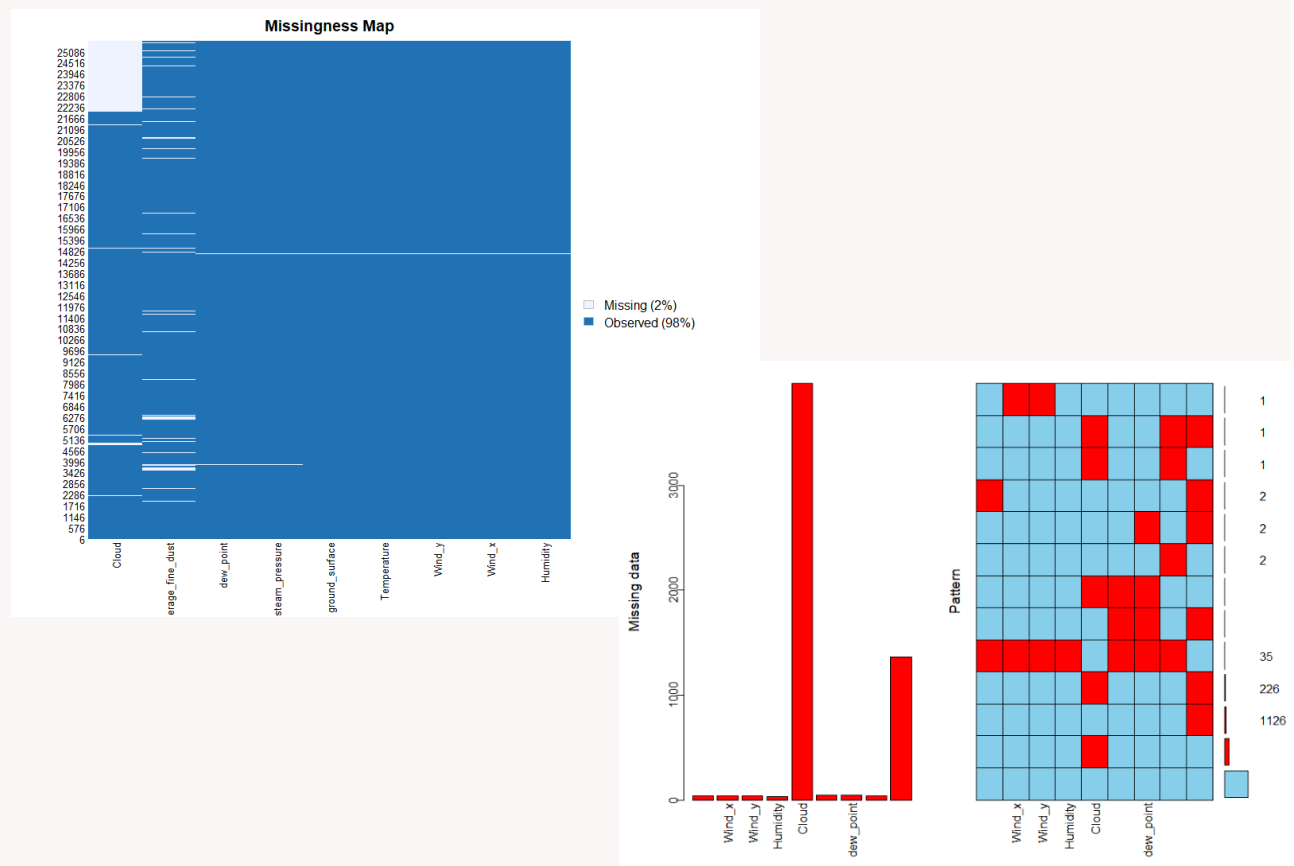
-1.00

SolarConnect

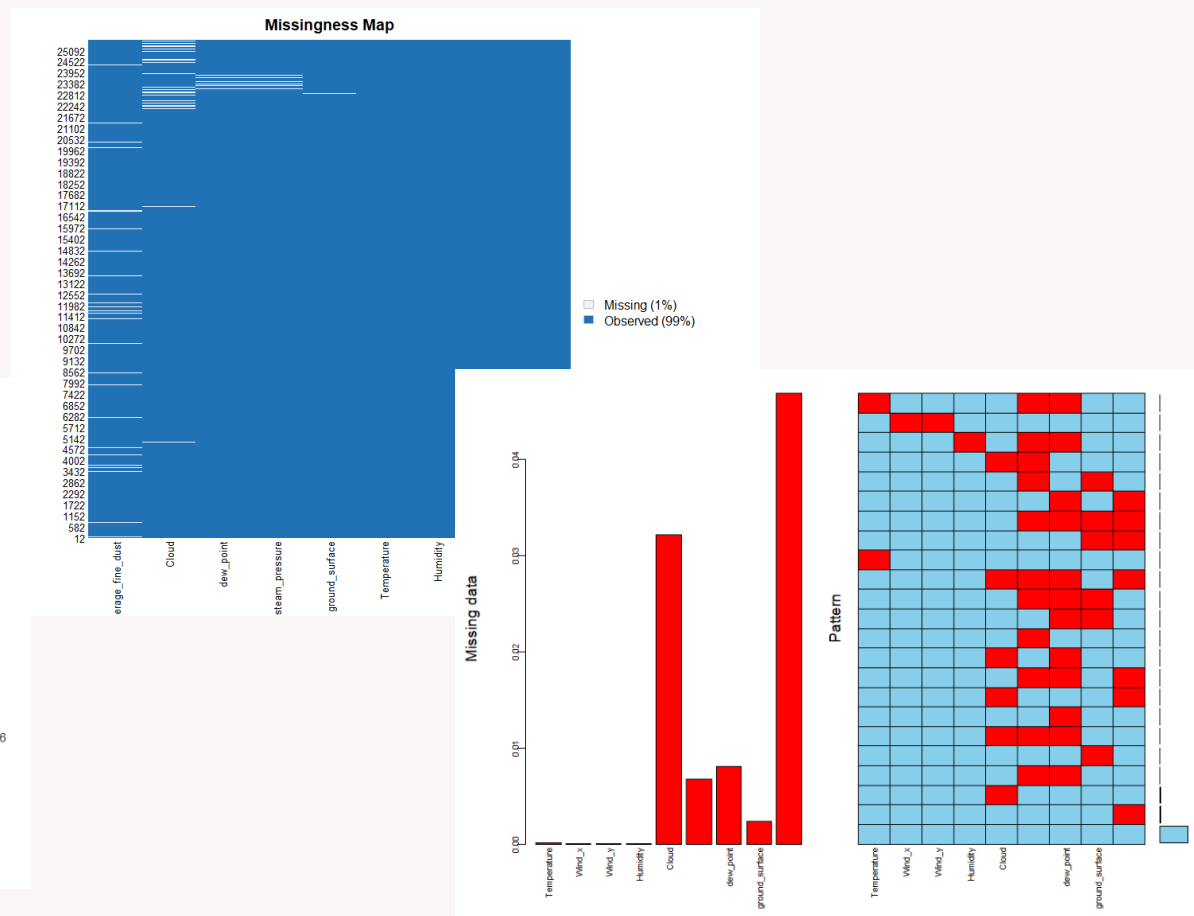
미세먼지가
태양광 발전에 미치는 영향(2)

(2) NA 데이터 전처리 (MICE/PPCA)

<dangjin>



<ulsan>



(2) NA 데이터 전처리 : MICE

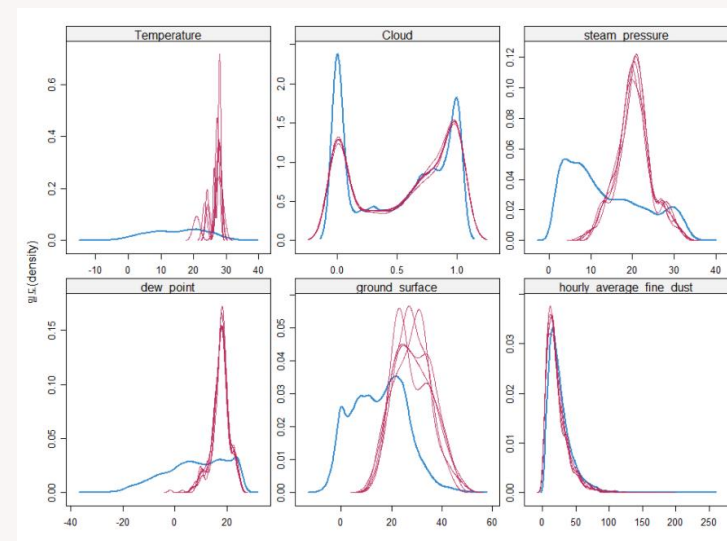
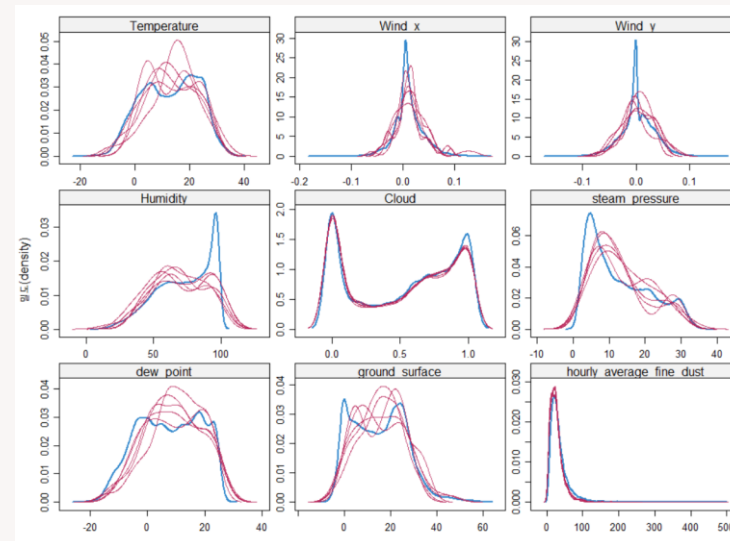
MICE(multivariate imputation by chained equations)

1. 결측값을 그럴듯한 값으로 대체하여 유사 대체 완료

자료 m개 생성

2. 각각 유사 대체 완료 자료 분석

3. 개별적 분석 결과를 결합하여 종합적 결론 도출



(2) NA 데이터 전처리 : PPCA

```
1 ppca(Matrix, nPcs = 2, seed = NA, threshold = 1e-05,
2     maxIterations = 1000, ...)
```

Arguments

| | |
|----------------------|--|
| Matrix | matrix – Data containing the variables in columns and observations in rows. The data may contain missing values, denoted as NA . |
| nPcs | numeric – Number of components to estimate. The preciseness of the missing value estimation depends on the number of components, which should resemble the internal structure of the data. |
| seed | numeric Set the seed for the random number generator. PPCA creates fills the initial loading matrix with random numbers chosen from a normal distribution. Thus results may vary slightly. Set the seed for exact reproduction of your results. |
| threshold | Convergence threshold. |
| maxIterations | the maximum number of allowed iterations |
| ... | Reserved for future use. Currently no further parameters are used. |

Examples

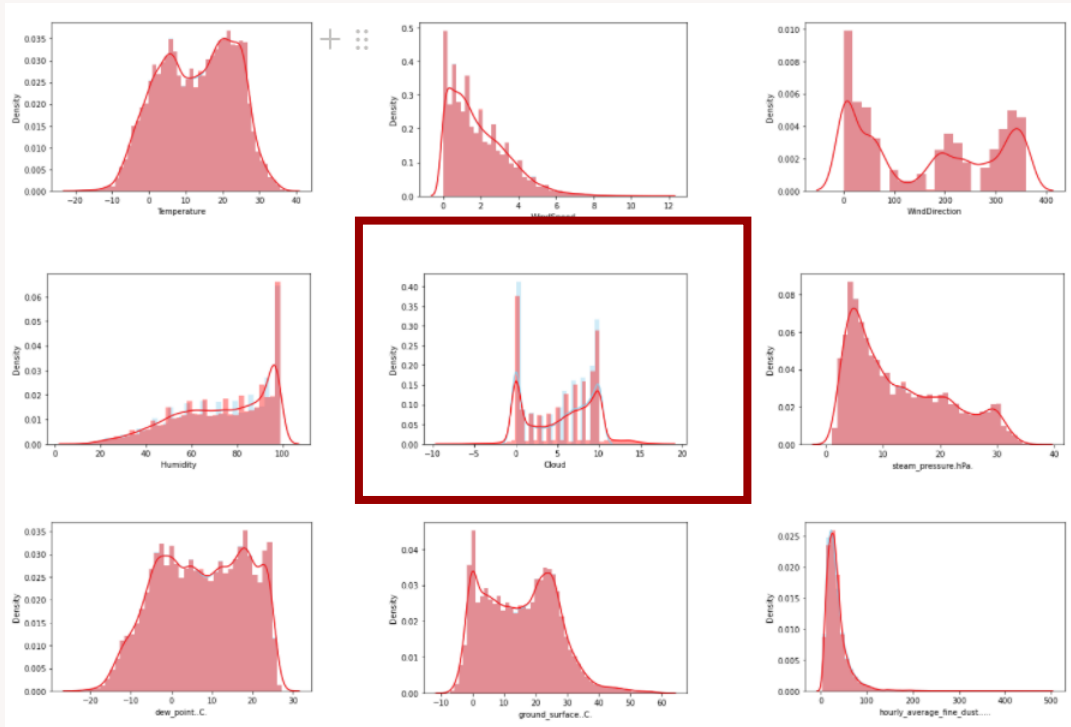
```
1 ## Load a sample metabolite dataset with 5% missing values (metaboliteData)
2 data(metaboliteData)
3 ## Perform probabilistic PCA using the 3 largest components
4 result <- pca(t(metaboliteData), method="ppca", nPcs=3, seed=123)
5 ## Get the estimated complete observations
6 cObs <- completeObs(result)
7 ## Plot the scores
8 plotPcs(result, type = "scores")
```

ppca 함수 주요 파라미터

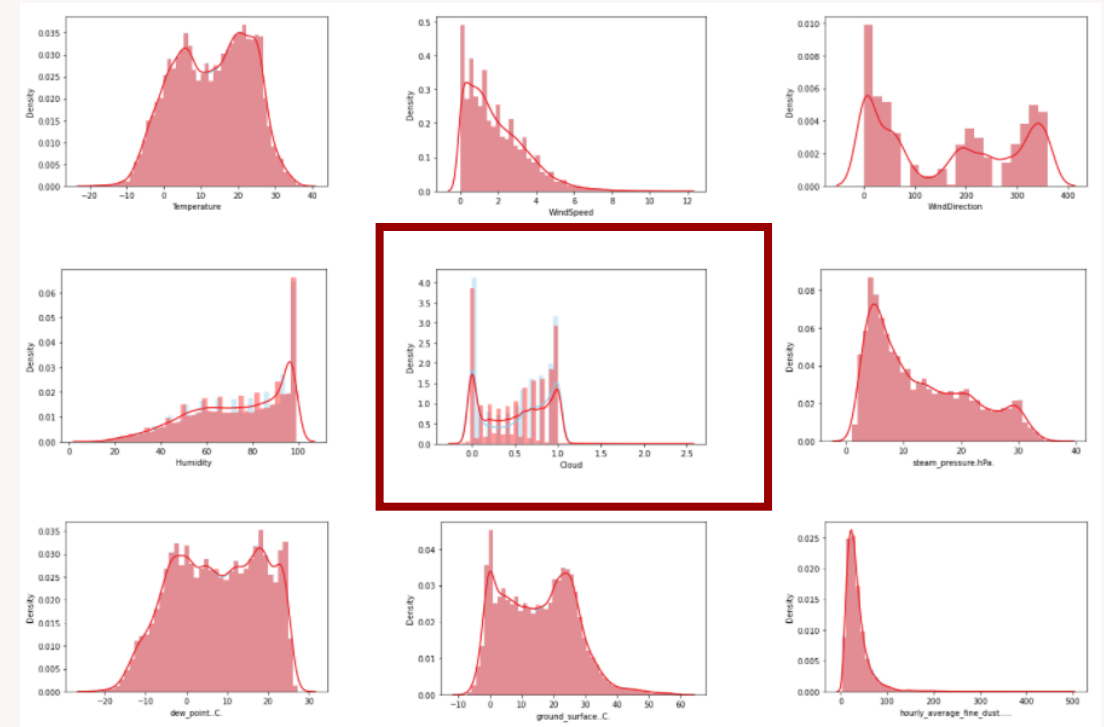
- Matrix: 결측치 처리에 쓰일 데이터
- nPcs: 예측할 요소의 개수 (입력 데이터에 의존)
- maxIteration: 최대 반복 횟수

(2) NA 데이터 전처리 : PPCA

1) 당진 (Cloud 변수/10)



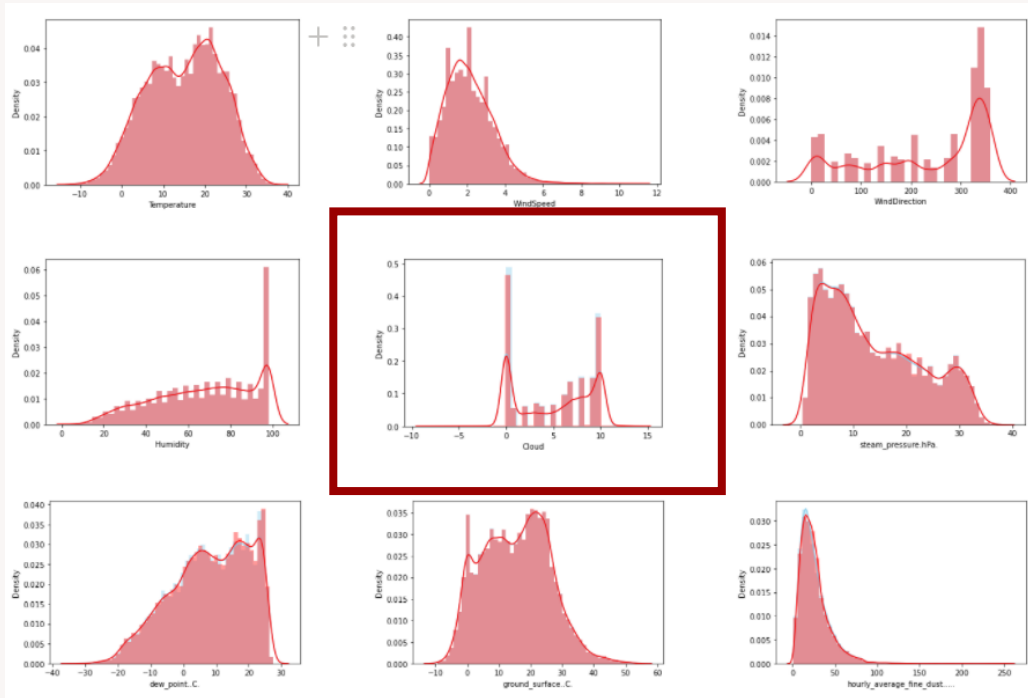
2) 당진 (Cloud 변수 범주화)



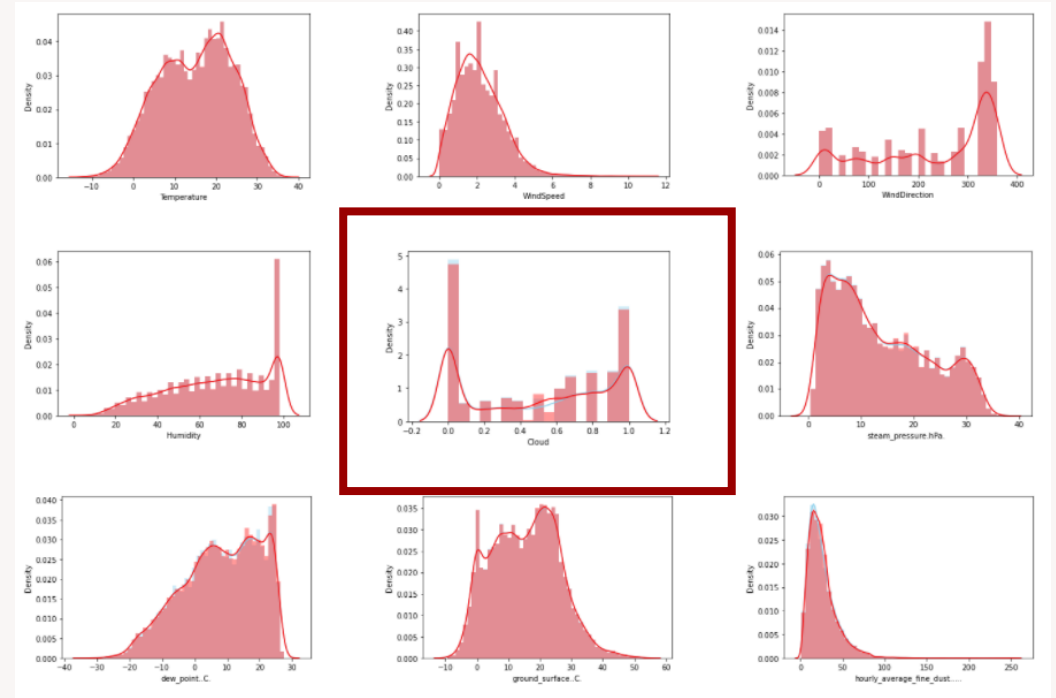
Cloud 변수 / 10 > Cloud 변수 범주화

(2) NA 데이터 전처리 : PPCA

1) 울산 (Cloud 변수/10)



2) 울산 (Cloud 변수 범주화)



Cloud 변수 / 10 > Cloud 변수 범주화

(2) NA 데이터 전처리 : PPCA

| | X | Temperatu | WindSpee | WindDirec | Humidity | Cloud | steam_pre | dew_point | ground_su | hourly_average_fine_dus |
|----|----|-----------|----------|-----------|----------|----------|-----------|-----------|-----------|-------------------------|
| 1 | 0 | 3.1 | 3.6 | 340 | 96 | 0.587038 | 7.3 | 2.5 | 2.7 | 13 |
| 2 | 1 | 2.8 | 0.7 | 140 | 97 | 0.681275 | 7.2 | 2.3 | 2.7 | 11 |
| 3 | 2 | 2.6 | 3.2 | 320 | 95 | 0.382004 | 7 | 1.8 | 2.5 | 10 |
| 4 | 3 | 2 | 1.9 | 230 | 97 | 0.051127 | 6.8 | 1.5 | 2.1 | 19 |
| 5 | 4 | 2.2 | 2.1 | 180 | 97 | 0.12323 | 6.9 | 1.7 | 2.3 | 21 |
| 6 | 5 | 4.1 | 4.4 | 270 | 97 | 0.695503 | 7.9 | 3.6 | 2.8 | 28 |
| 7 | 6 | 3.5 | 7.9 | 320 | 93 | 0.184502 | 7.3 | 2.4 | 2.9 | 27 |
| 8 | 7 | 2.2 | 6.4 | 290 | 86 | -2.25808 | 6.1 | 0 | 1.5 | 76 |
| 9 | 8 | 1 | 7.7 | 320 | 82 | -2.53824 | 5.4 | -1.7 | 1.5 | 59 |
| 10 | 9 | 0.3 | 8.9 | 320 | 71 | -3.29991 | 4.4 | -4.3 | 3.1 | 54 |
| 11 | 10 | 0.6 | 7.9 | 320 | 63 | -3.7432 | 4 | -5.6 | 5.7 | 63 |
| 12 | 11 | 0.5 | 9.1 | 320 | 58 | -3.91567 | 3.7 | -6.8 | 7.5 | 61 |
| 13 | 12 | 0.7 | 6.7 | 320 | 60 | -3.43272 | 3.9 | -6.1 | 9.9 | 63 |
| 14 | 13 | 1.5 | 6.5 | 320 | 60 | -3.05194 | 4.1 | -5.4 | 10.9 | 63 |
| 15 | 14 | 0.1 | 5.1 | 340 | 56 | -4.15424 | 3.5 | -7.6 | 8.2 | 62 |
| 16 | 15 | 0.4 | 5 | 320 | 56 | -4.15825 | 3.5 | -7.3 | 8.4 | 67 |
| 17 | 16 | 0.3 | 6 | 290 | 56 | -4.40242 | 3.5 | -7.4 | 6.5 | 67 |
| 18 | 17 | -0.5 | 4.6 | 320 | 59 | -4.92324 | 3.5 | -7.5 | 3.3 | 71 |
| 19 | 18 | -1.3 | 5.4 | 290 | 62 | -4.97212 | 3.5 | -7.6 | 0.8 | 61 |
| 20 | 19 | -1.7 | 4.4 | 320 | 63 | -5.22753 | 3.4 | -7.8 | 0 | 64 |
| 21 | 20 | -1.8 | 4.3 | 320 | 63 | -4.97268 | 3.4 | -7.9 | 0 | 53 |

Cloud 변수

1) 0보다 작은 음수의 경우
: 0으로 통일

2) 10보다 큰 수의 경우
: 10으로 통일

(3) Forecast data 보간 (cubic interpolation)

예보 발표 시간 몇 시간 후 예측?

| Forecast time | forecast | Temperature | Humidity | WindSpeed | WindDirection | Cloud |
|---------------------|----------|-------------|----------|-----------|---------------|-------|
| 2018-03-01 11:00:00 | 4 | 0 | 60 | 7.3 | 309 | 2 |
| 2018-03-01 11:00:00 | 7 | -2 | 60 | 7.1 | 314 | 1 |
| 2018-03-01 11:00:00 | 10 | -2 | 60 | 6.7 | 323 | 1 |
| 2018-03-01 11:00:00 | 13 | -2 | 55 | 6.7 | 336 | 1 |
| 2018-03-01 11:00:00 | 16 | -4 | 55 | 5.5 | 339 | 1 |

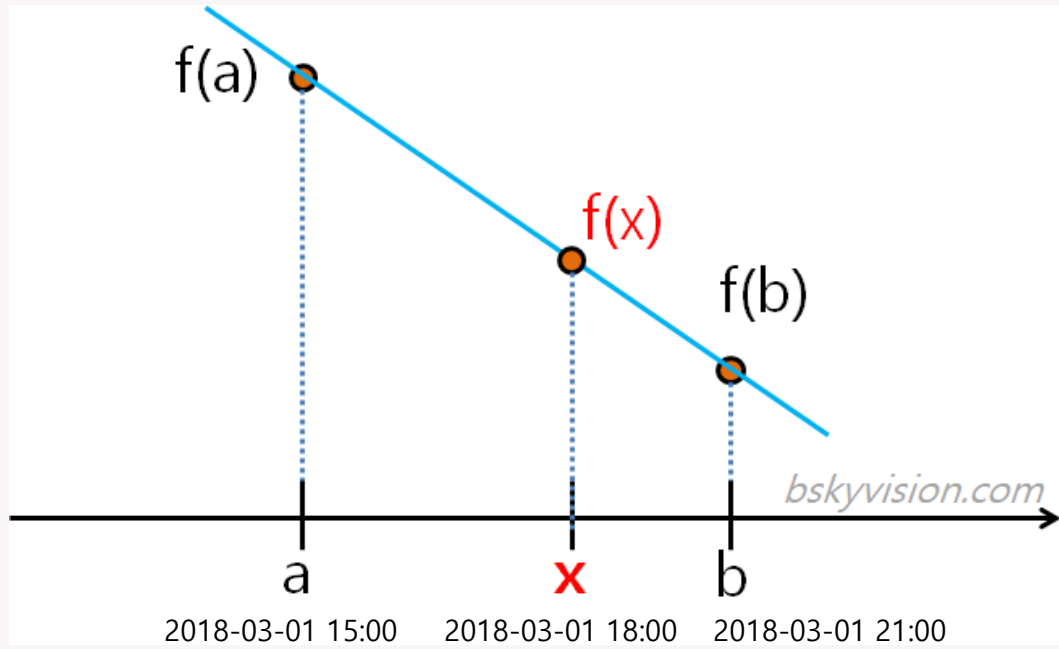
Target time

→ 2018-03-01 15:00
→ 2018-03-01 18:00
→ 2018-03-01 21:00
→ 2018-03-02 00:00
→ 2018-03-02 03:00

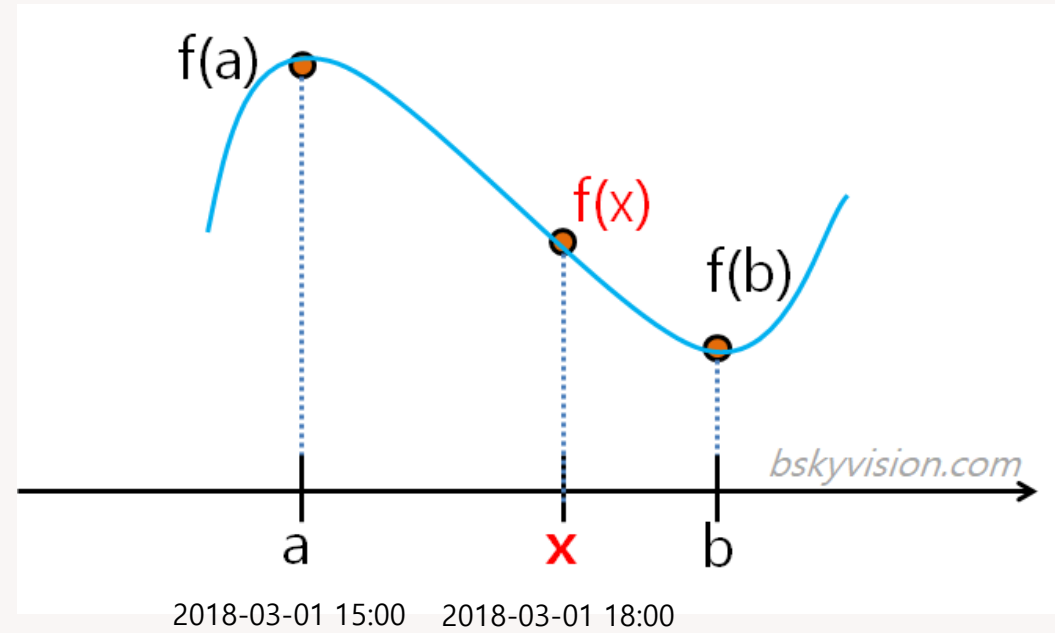
Forecast 데이터에서, 예보 target time은 3시간 간격으로 주어져 있음.

→ 주어진 3시간 간격 데이터 사이의 데이터를 보간해야 함.

(3) Forecast data 보간 (cubic interpolation)



선형 보간 (linear interpolation)



삼차 보간 (cubic interpolation)

(3) Forecast data 보간 (cubic interpolation)

| date | Temperatrue | Humidity | Cloud |
|--------------------|--------------|-------------|-------------|
| 2018-03-02 1:00:00 | -2.104535188 | 50.96545273 | 0.24995661 |
| 2018-03-02 2:00:00 | -2.48362815 | 52.77236218 | 0.249965288 |
| 2018-03-02 3:00:00 | -3 | 55 | 0.25 |
| 2018-03-02 4:00:00 | -3.51637185 | 57.22763782 | 0.250034712 |
| 2018-03-02 5:00:00 | -3.895464812 | 59.03454727 | 0.25004339 |
| 2018-03-02 6:00:00 | -4 | 60 | 0.25 |

보간한 데이터

보간한 데이터

(4) 파생변수 생성

- 태양의 위치는 태양광 발전과 직접적인 연관이 있음.
- 태양은 천구상에서 일주운동(지구의 자전)과 연주운동(지구의 공전)을 함.
- 같은 시간일지라도 날짜가 바뀌면, 태양의 위치는 바뀜.

(4) 파생변수 생성

```
def sun_system(date, lat, longt):
    from datetime import datetime, timedelta
    import math
    pi = math.pi
    lat = lat / 180 * pi
    temp_long = 135 - longt

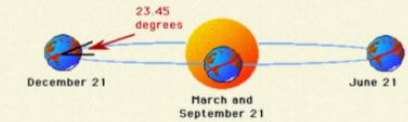
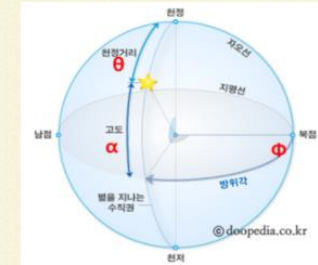
    N = int((date - datetime(int(date.year), 1, 1, 0, 0)).days) + 1
    h = ((int(date.hour) - 12) * 15 - temp_long) / 180 * pi
    # h = ((int(date.hour) - 12) * 15) / 180 * pi
    delta = -23.44 * math.cos(2*pi/365*(N+10)) / 180 * pi

    sin_alpha = math.sin(lat) * math.sin(delta) + math.cos(lat) * math.cos(delta) * math.cos(h)
    sin_theta = math.sqrt(1 - sin_alpha*sin_alpha)
    sin_phi = -1 * math.sin(h) * math.cos(delta) / sin_theta
    cos_phi = (math.sin(delta) * math.cos(lat) - math.cos(h) * math.cos(delta) * math.sin(lat)) / sin_theta

    result = {
        'sin(elev ang)' : [sin_alpha],
        'sin(azimuth ang)' : [sin_phi],
        'cos(azimuth ang)' : [cos_phi]
    }
    df = pd.DataFrame(result)
    return(df)
```

○ 각 변수 설명(모든 값은 degree) 및 관련 설명 그림

- 적위(δ , declination of the Sun) : 날마다 변하는 태양 빛과 적도가 이루는 각도(지구 공통)
- 지역의 위도(φ , latitude) : 한국은 33 ~ 43도(평균 38도)
- 지역의 경도(longitude) : 한국은 135도(영국 런던 그리니치천문대 기준)
- 시간각(h, hour angle) : 1시간을 15도씩으로 계산하되, 남중고도 12시를 0h 하고 남중고도 직전은 (-), 직후는 (+).
- 태양천정각(θ , solar zenith angle) : 천정(하늘)과 태양이 이루는 각도
- 태양고도(α , solar elevation angle) : 지면과 태양이 이루는 각도($90 - \theta$)
- 태양방위(Φ , solar azimuth angle) : 북쪽을 기준으로 시계방향으로 설정(북>동>남>서), xyz 좌표 방향과 반대



$$\delta = -23.44^\circ \times \cos[360^\circ/365 \times (N+10)]$$

$$\sin \alpha = \sin \varphi \sin \delta + \cos \varphi \cos \delta \cos h$$

$$\sin \Phi = \frac{-\sin h \cos \delta}{\sin \theta} \dots (1\text{식})$$

$$\cos \Phi = \frac{\sin \delta \cos \varphi - \cos h \cos \delta \sin \varphi}{\sin \theta} \dots (2\text{식})$$

| sin(elev ang) | sin(azimuth ang) | cos(azimuth ang) |
|---------------|------------------|------------------|
| 0.381913 | 0.860378 | -0.509657 |
| 0.530157 | 0.726173 | -0.687512 |
| 0.636285 | 0.511144 | -0.859495 |
| 0.693064 | 0.202606 | -0.979260 |
| 0.696624 | -0.156348 | -0.987702 |

(4) 파생변수 생성

- 바람에 대한 변수는 WindSpeed(m/s)와 WindDirection(각도)으로 주어져 있음.
- 각도는 0과 360이 이어져야 하므로, 이를 벡터변환(좌표변환)해준다.

```
dangjin_fcst['Wind_x'] = dangjin_fcst['WindSpeed'] * np.cos(dangjin_fcst['WindDirection']/180*np.pi)  
dangjin_fcst['Wind_y'] = dangjin_fcst['WindSpeed'] * np.sin(dangjin_fcst['WindDirection']/180*np.pi)
```

| WindSpeed | WindDirection |
|-----------|---------------|
| 7.3 | 309.0 |
| 7.1 | 314.0 |
| 7.1 | 314.0 |
| 6.7 | 323.0 |
| 6.7 | 323.0 |



| Wind_x | Wind_y |
|----------|-----------|
| 4.594039 | -5.673166 |
| 4.932074 | -5.107313 |
| 5.350858 | -4.032161 |
| 6.120755 | -2.725136 |
| 5.134692 | -1.971024 |

(4) 파생변수 생성

- 온도와 상대습도로 이슬점을 구할 수 있다.

```
def dew_temp(X):  
    c = 243.12  
    b = 17.62  
    gamma = (b * X['Temperatrue']) / (c + X['Temperatrue']) + np.log(X['Humidity']/100)  
    dp = (c*gamma) / (b-gamma)  
    X['dew_point'] = dp
```

(4) 파생변수 생성

- Hour와 day of year는 주기성을 띄기 때문에, sin과 cos으로 주기성을 설정해줄 수 있다.

```
def prep(X):  
    dew_temp(X)  
    X['dayofyear'] = X['date'].dt.dayofyear  
    X['hour'] = X['date'].dt.hour  
    X['dayofyear'] = X['dayofyear'].astype('float')  
    X['hour'] = X['hour'].astype('float')  
    X['sin_hour'] = np.sin(2*np.pi*(X.hour/24))  
    X['cos_hour'] = np.cos(2*np.pi*(X.hour/24))  
  
    X['sin_dayofyear'] = np.sin(2*np.pi*(X.dayofyear/365))  
    X['cos_dayofyear'] = np.cos(2*np.pi*(X.dayofyear/365))  
    X = X.drop(['hour', 'dayofyear'], axis=1)
```

(4) 파생변수 생성

- Forecast의 feature들은 그대로, Observation의 feature들은
3시간, 6시간, 12시간, 24시간 단위로 이동평균 값을 feature로 사용한다.

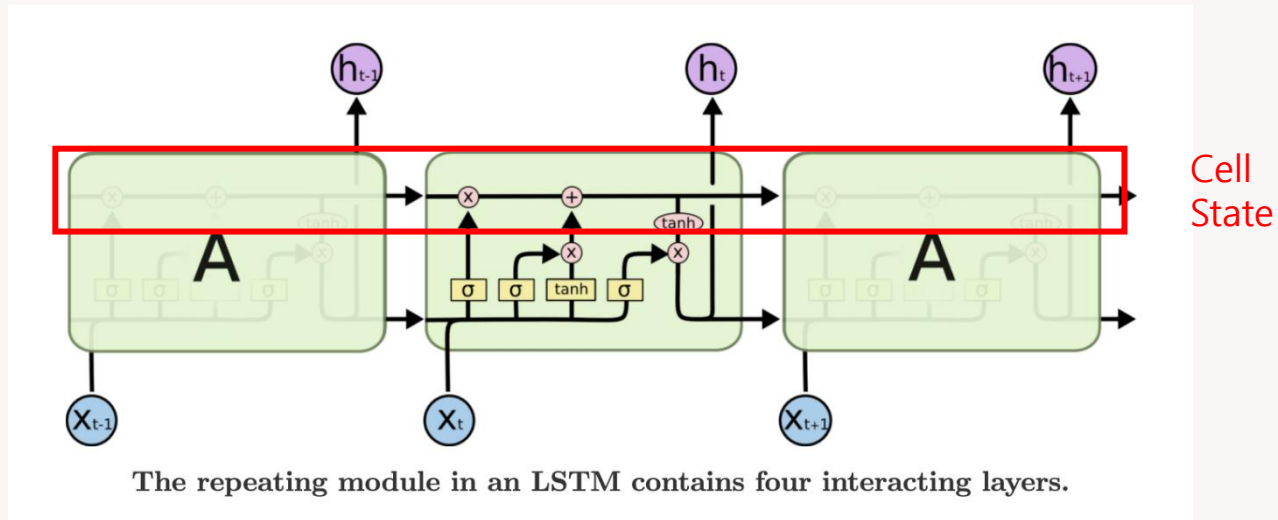
(ex. 05-27 19:00:00의 예측값 + 05-26 19:00:00을 기준으로 3시간, 6시간, 12시간, 24시간 전의 이동평균)

| | T3 | Wx3 | Wy3 | H3 | C3 | dp3 | FD3 | T6 | Wx6 | Wy6 | ... | H24 | C24 | dp24 | FD24 |
|---------------------|----------|----------|-----------|-----------|----------|------------|-----------|----------|----------|-----------|-----|-----------|----------|-----------|-----------|
| date | | | | | | | | | | | | | | | |
| 2018-03-02 23:00:00 | 3.000000 | 2.547811 | -2.765482 | 30.666667 | 0.666667 | -12.766667 | 22.333333 | 4.316667 | 1.610254 | -1.399842 | ... | 47.333333 | 4.166667 | -5.070833 | 17.833333 |
| 2018-03-03 00:00:00 | 2.100000 | 2.735749 | -2.833886 | 34.666667 | 0.633333 | -11.933333 | 25.666667 | 3.433333 | 1.759413 | -1.593653 | ... | 44.791667 | 4.041667 | -5.887500 | 18.708333 |
| 2018-03-03 01:00:00 | 1.300000 | 2.855183 | -1.330378 | 38.000000 | 0.633333 | -11.500000 | 25.000000 | 2.583333 | 2.280685 | -1.525249 | ... | 42.500000 | 3.875000 | -6.612500 | 18.750000 |
| 2018-03-03 02:00:00 | 0.466667 | 2.498318 | -0.387623 | 41.666667 | 0.633333 | -11.100000 | 27.000000 | 1.733333 | 2.523064 | -1.576552 | ... | 41.083333 | 3.875000 | -7.187500 | 19.666667 |

Lstm

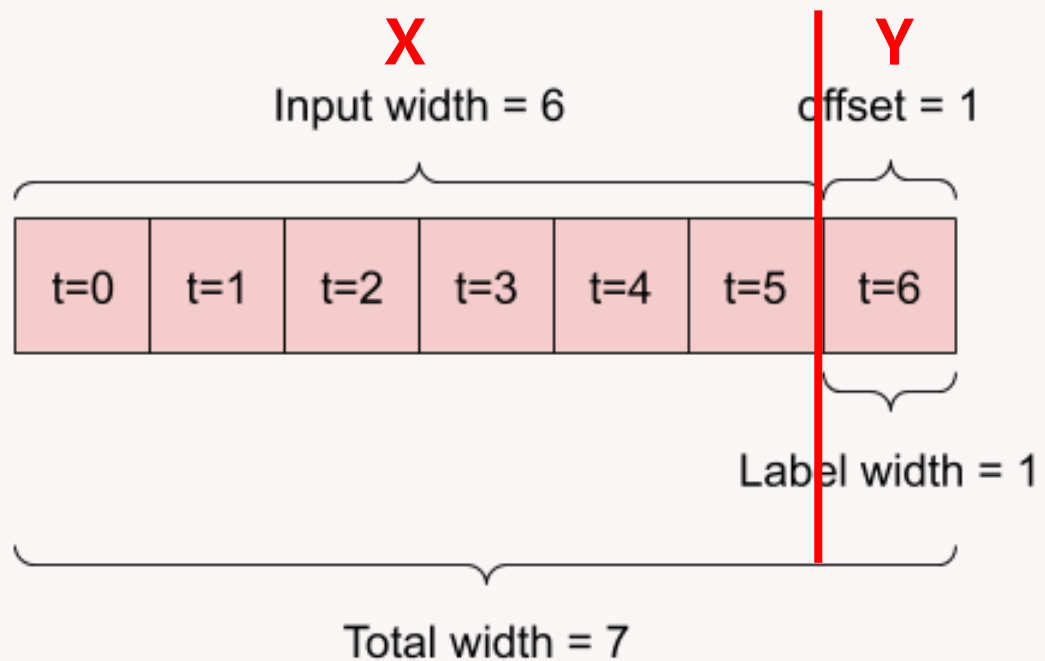
: RNN에 장단기 메모리 셀을 추가하여 기존 RNN을 개선한 모델

: 시간적 상관성과 기상 요인과의 상관성으로 인한 발전량 예측의 어려움을 해결할 수 있는 모델



Lstm

: LSTM을 시계열 예측에 활용하기 위해서는,
input 데이터를 재구성하여 지도학습 형태로 만들어줘야함.



Sarimax : SARIMA + exogeneous variable

SARIMA(Seasonal Autoregressive Integrated Moving Average):

trend에 대하여 ARIMA를 수행하고, 계절성에 대해서 추가적으로 ARIMA를 수행한 것

SARIMA

(p, d, q)

$(P, D, Q)_m$

↑

↑

Non-seasonal part
of the model

Seasonal part
of the model

where m = seasonal lag of observations.

p : pacf에서 수렴하기 직전값 (0과 2 사이 범위)

d : adf테스트 및 추세의 가시적 확인으로 추세여부 확인 (0과 2 사이 범위)

q : acf에서 수렴하기 직전값 (0과 2 사이 범위)

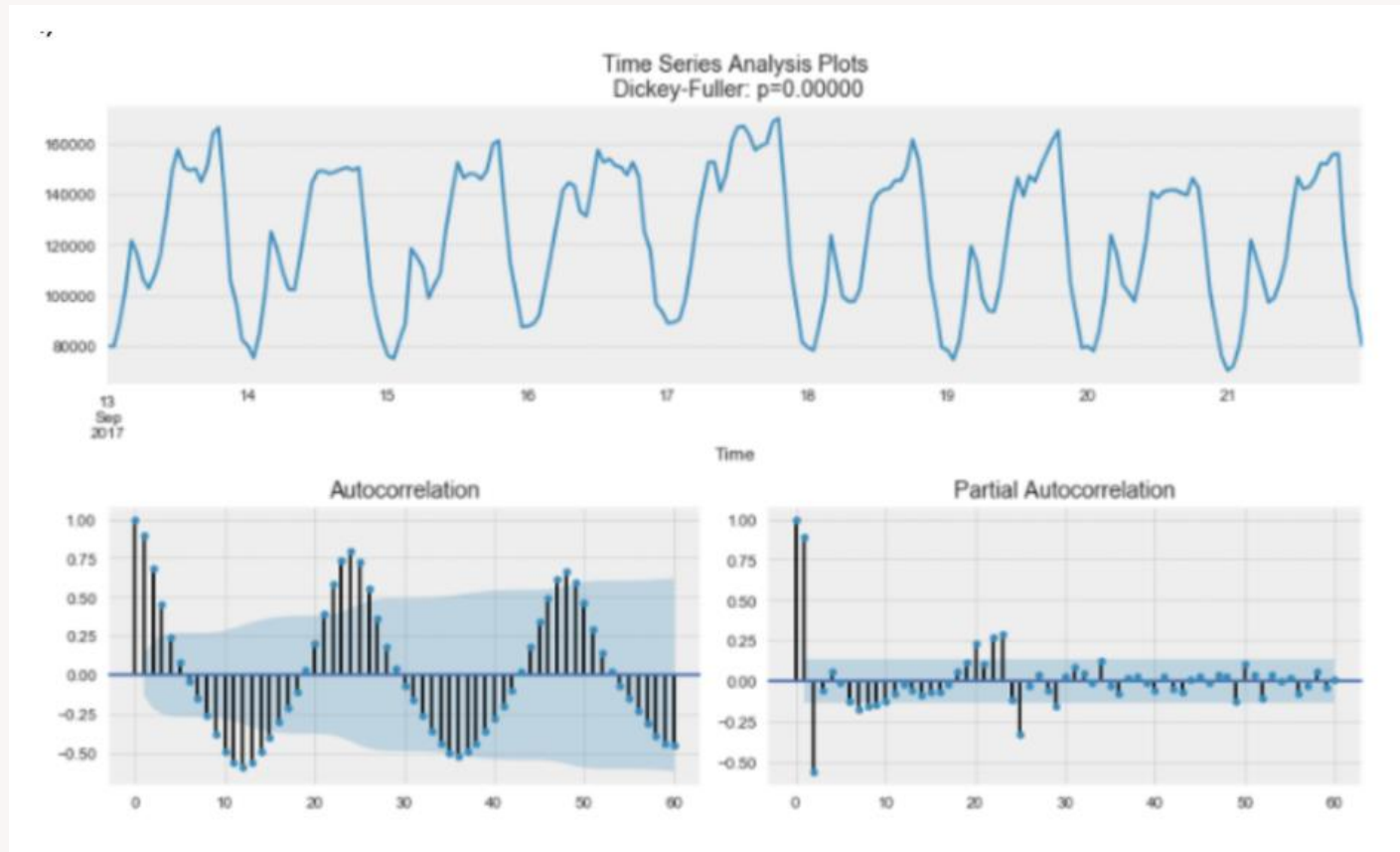
P : pacf에서 계절성이 주기가 몇 번 반복 됐는지 확인

D : 계절성이 있는지 확인하고 계절성 차분의 필요성에 따라 1 or 0

Q : acf기준 계절성 주기가 몇 번 반복 됐는지 확인

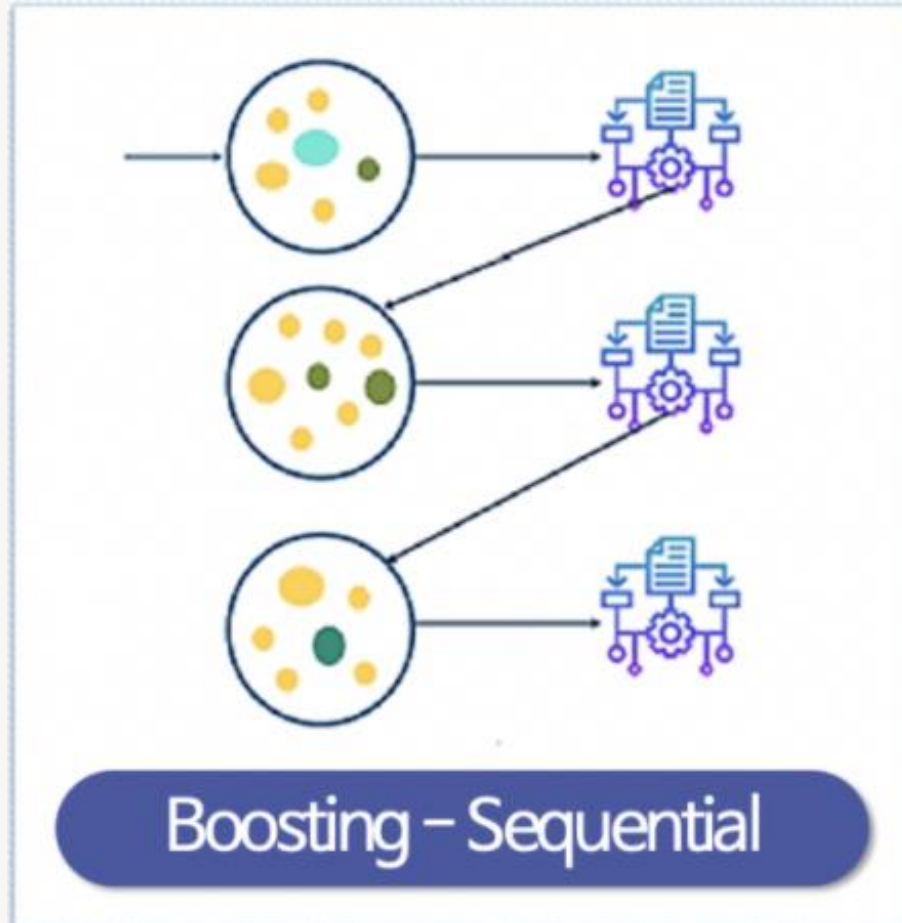
m : 계절성 주기 (계절주기가 1년이기에 12로 설정)

Sarimax : SARIMA + exogeneous variable



+ PCA

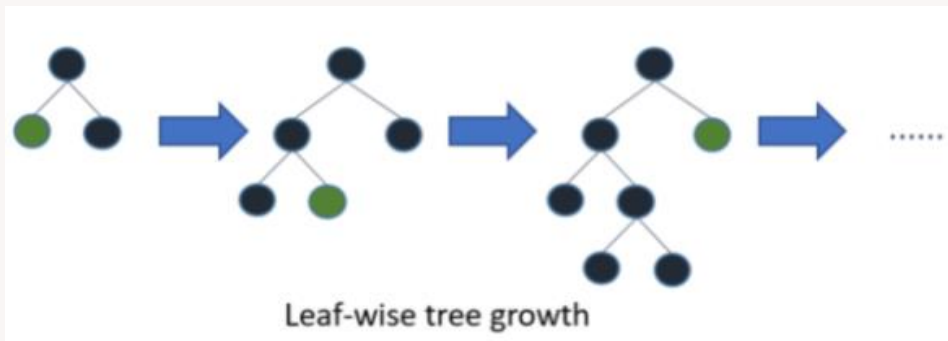
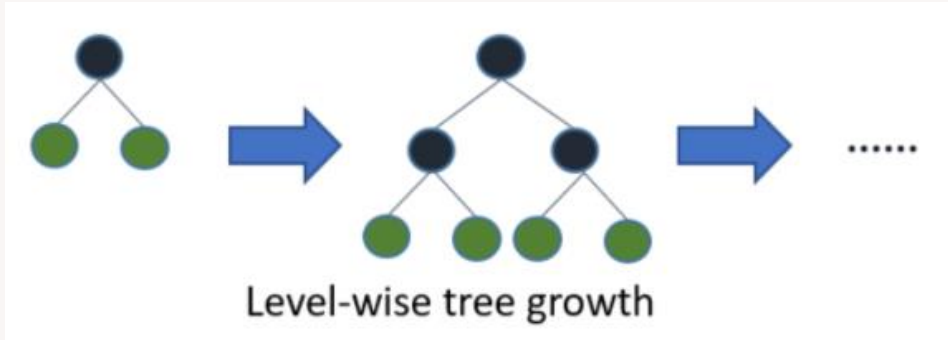
Light GBM



- Boosting

여러 개의 알고리즘(Tree)이 순차적으로 학습-예측을 하면서 이전 모델의 잔차를 이용하여 모델을 발전시키는 머신러닝 방법

Light GBM



- LightGBM

- Level-wise tree growth
- Fast
- Available to use GPU
- Robust to Collinearity problem (Tree based)
- Overfitting 민감

동서발전 태양광 발전량 예측 AI 경진대회
태양광 | 한국동서발전(주) | 개인/스타트업 | 시계열 | NMAE

🏆 상금 : 총 1,600만원




🕒 2021.04.07 ~ 2021.06.08 18:00 [+ Google Calendar](#)

👤 941팀 📅 D-12

[대회안내](#) [데이터](#) [코드 공유](#) [토론](#) [대회문의](#) [리더보드](#) [팀](#) [제출](#)

PUBLIC RANKING CHART [순위기준](#)

● WINNER ● 1% ● 4% ● 10%

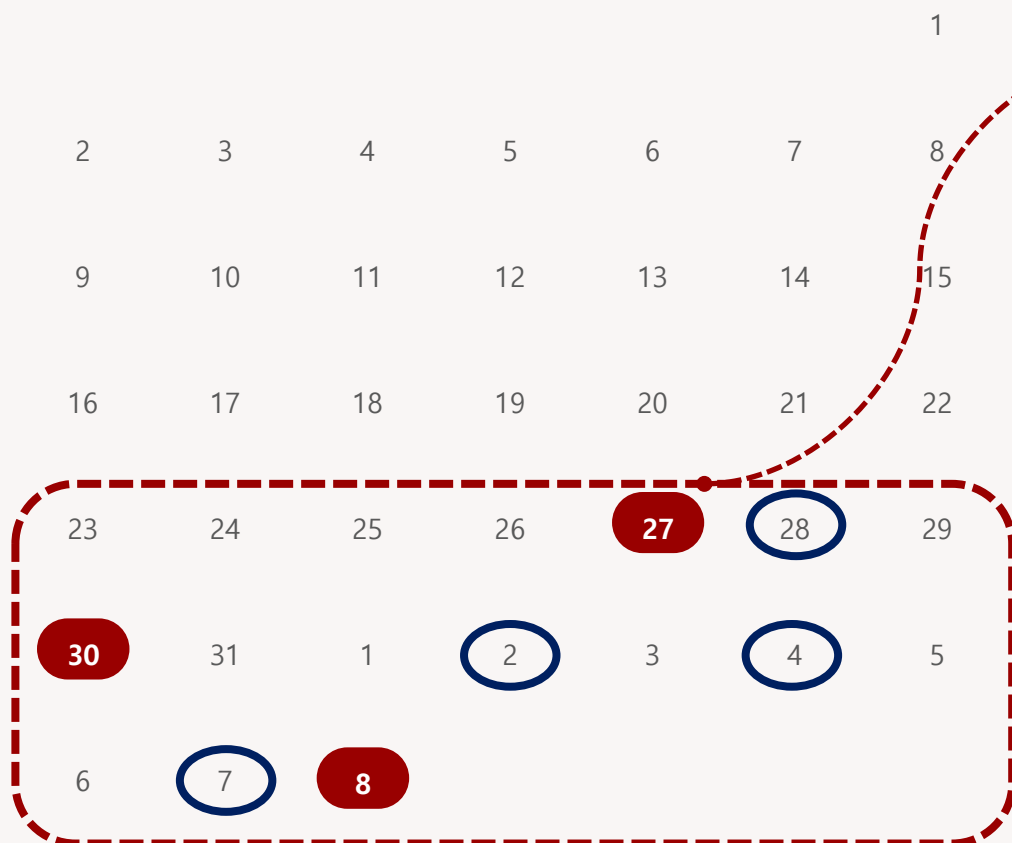
| # | 팀 | 팀 멤버 | 점수 | 제출수 | 등록일 |
|----|-----------|---|---------|-----|--------|
| 21 | 이두형 |  | 7.5782 | 12 | 18분 전 |
| 1 | SHOWMAKER |  | 5.81993 | 8 | 20시간 전 |
| 2 | 양현준 |  | 6.09819 | 24 | 하루 전 |

점수(NMAE-10): 7.5782

941팀 중 "21등!"

2021년 5월

SUN MON TUE WED THU FRI SAT



- EDA, 전처리, 모델 만들기 (5월 3일~ 5월 26일)

- 모델 hyperparameter 조정 (5월 26일 ~ 6월 8일)
: 정기 회의 X 4

-> LSTM, LGBM : Hyperparameter 조정

-> SARIMAX : exogeneous 변수 조정
(auto_arima 함수 사용)

-> 5월 30일 팀 병합 마감.

-> 6월 8일 public 평가 마감.

An aerial photograph of a city, likely Seattle, showing a large body of water (Puget Sound) in the foreground, a dense urban area with many buildings, and a range of mountains in the background. The image is slightly blurred and has a dark, muted color palette. The text "THANK YOU" is overlaid in the center in a large, white, sans-serif font.

THANK YOU