|  |
| --- |
|  |
| 재생에너지 발전량 예측 경진대회 보고서  (분야 :풍력 발전량 예측) |
|  |

2019. 08.

작성자 : MakinaPebbles(유현우, 전민규)

연락처 : +82-10-2720-7948 (유현우)

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  | **목 차** |  |  |  |  |

|  |
| --- |
| 1. 개요  가. 분석 목적  나. 분석 방향  다. 사용 데이터  라. 문제 정의  2. Exploratory Data Analysis (EDA)  가. 기상관측치와 기상예보의 차이  나. 발전량 데이터의 시계열적 특성  3. Modeling  가. Baseline Model  나. Feature Engineering  다. Data Augmentation  라. 1차 제출 예측 모델  마. 최종 제출 예측 모델  4. Non-Stationary Distribution Issue  가. Using Smaller Dataset  5. Continual Learning을 통한 모델의 산업 적용  가. 모델의 실제 산업에의 적용 |

**1. 개요**

**가. 분석 목적**

**(1) 분석의 목적**

○ 신재생에너지 공급의무화제도(RPS : Renewable Portfolio Standard[[1]](#footnote-2))로 인해 일정 규모(500MW) 이상의 발전설비를 보유한 발전사업자는 총 발전량의 일정비율 이상을 신재생에너지를 이용하여 공급하여야 한다. 또한 **스케쥴링이 가능한 에너지원은 에너지 거래에서 더 높은 가치를 가진다[[2]](#footnote-3).**

○ 다른 에너지원보다 시간, 기후의 영향으로 변동성이 큰 풍력 에너지 발전량을 성공적으로 예측함으로써, 전력 생산 스케쥴링을 가능하게 한다. 이를 통해 안정적인 전력 공급과 효율적인 자원 분배를 목표로 한다.

ㅇ 발전량의 cycle을 확인하여 불규칙한 기상으로 인해 **예측하기 어려운 풍력발전량을 예측**

**나. 분석 방향**

**(1) Time Series Forecasting**

○ 3시간 단위의 개별적인 기상예보를 Input으로 받아들여 예보가 가리키는 시간, +1시간, +2시간의 발전량을 Output으로 내놓는 Time Series Forecasting Model

**(2) LSTM Model**

○ 기상예보의 오차에 종속적인 Time Series Forecasting Model의 한계를 극복하기 위해, 연속적인 위성사진을 Input으로 받아들여 풍력 발전량을 예측하는 LSTM Model.

**(3) Ensemble, Stacking**

○ 앙상블(Ensemble)과 스태킹(Stacking)을 통해 (1), (2)의 각 모델이 가진 강점은 살리고 약점을 상쇄하는 최종 예측 모델을 구축. Stacking에서 1st-Level Model의 Hyperparameter와 Feature Engineering 방법은 Bayesian Optimization을 통해 최적화된다.

**(4) Conditional Variational Autoencoder(CVAE) 활용 Data Augmentation**

○ 공개된 타 발전소 데이터, 기상 데이터를 활용하여 상대적으로 데이터가 부족한 신규 발전소에 대해서도 발전량을 예측. 추가적인 데이터를 학습시킴으로써 기존 모델의 성능 향상

**다. 사용 데이터**

**(1) 기상자료개방포털 기상관측치, 기상예보 데이터**

○ 기상관측치, 기상예보 데이터

○ 기상자료개방포털(data.kma.go.kr)에 공개된 데이터

**(2) SK 웨더 플래닛(Weather Planet) 기상예보 데이터**

○ 기상관측치, 기상예보 데이터

**(3) 한국남부발전 태양광 발전량 데이터**

○ 한경풍력발전, 성산풍력발전

○ 공공데이터포털(data.go.kr)에 공개된 데이터

**라. 문제 정의**

1. **Time Series Forecasting과 Regression**

a. Regression

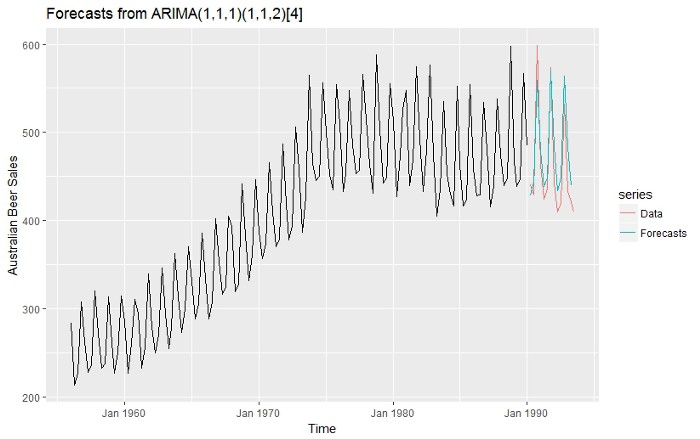
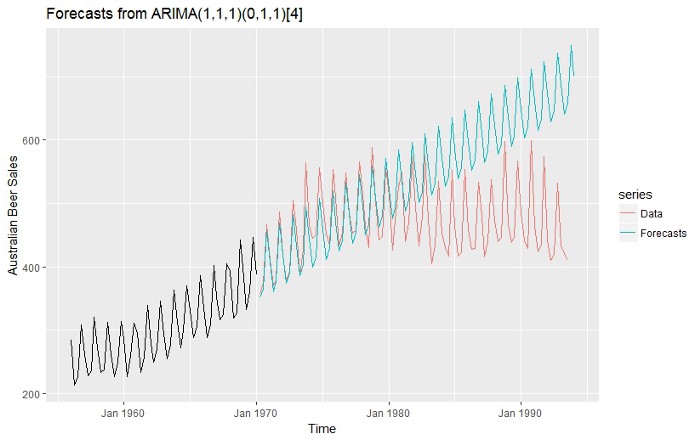
주어진 데이터 집합 에 대해, 종속 변수 와 개의 설명변수 사이의 관계를 모델링하는 문제. 일반적으로 Train Set과 Test Set의 데이터 분포(Distribution)이 같다.

b. Time Series Forecasting

시간에 따라 연속적으로 변하는 를 예측하는 문제. 주어진 변수 로 를 설명하여 미래를 예측하는 문제이다. **Train Set과 Test Set의 데이터 분포(Distribution)이 다른 경우가 많으며**, 이는 Time Series Forecasting을 일반적인 Regression문제와 구분 짓는 요인이 된다. 시간에 따라 기상 예보의 분포와 발전량의 분포가 변하는 풍력 발전량 예측은 Regression이 아닌 Time Series Forecasting 문제이다.

1. **Time Series Forecasting Issues**

a. Non-Stationary Distribution



<Figure. 1> Non-Stationary Distribution을 예측할 때 발생하는 문제

일반적으로 Time Series Forecasting에서 Train Set과 Test Set은 다른 Distribution을 가진다. Figure 1.과 같이 Non-Stationary한 Distribution을 가지는 대상은 T**rain Set에서 x, y의 관계를 모델이 잘 학습하더라도, Test Set에서 예측하려는 변수의 Distribution이 바뀌면 모델이 잘 예측하지 못하게 된다**. Non-Stationary한 Distribution을 가지는 Forecasting 문제는 **Continual Learning을 통해 모델을 지속적으로 새로운 Distribution에 맞게 업데이트함으로써 이 문제를 해결할 수 있다.**

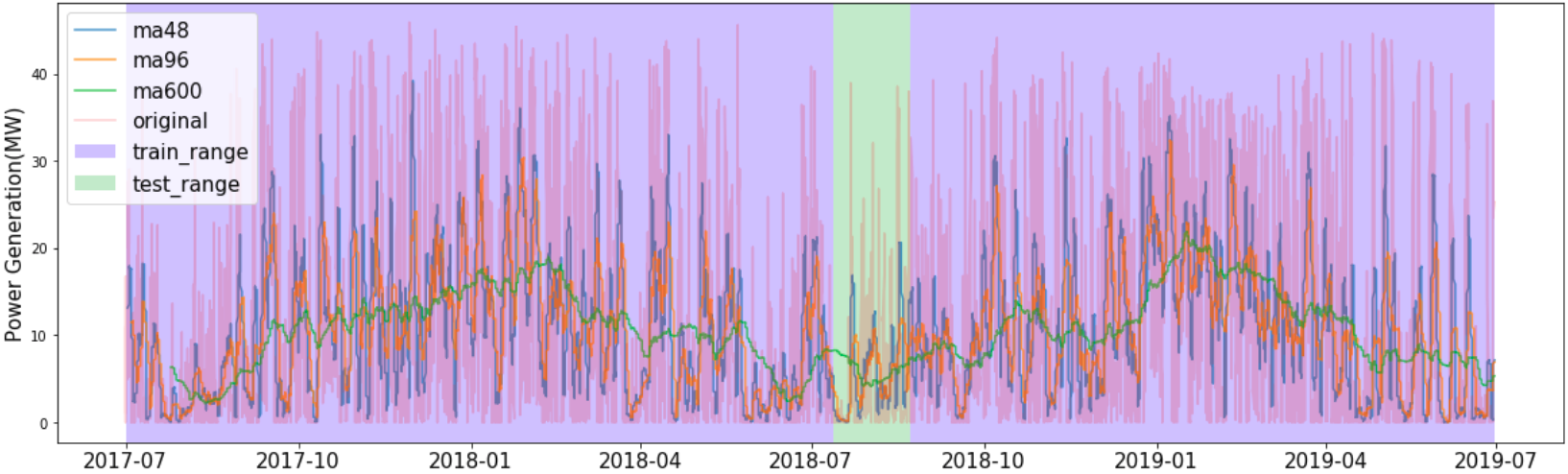
b. Time Series Forecasting Model Validation

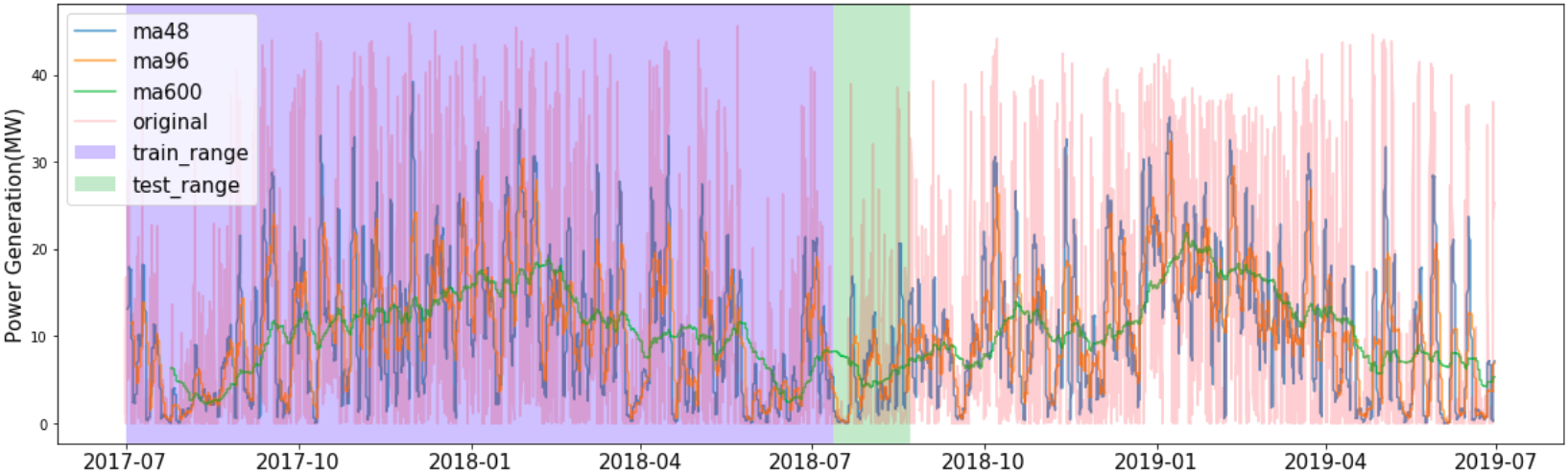
Time Series Forecasting Model은 일반적인 Regression모델과 다른 방법으로 검증되어야 한다. 특히, Cross Validation은 적합하지 않은 검증 방법이다. Cross-Validation을 수행하면 시간 순서상 뒤에 있는 Train Set을 보고 그보다 앞에 있는 Test Set을 예측하게 되는데, 이는 모델을 실제로 운영할 때 일어나지 않는 시나리오이다. 예를들어 Cross-Validation을 사용할때는 극단적으로 2018년 8월2일 14시와 16시 데이터가 train set에 있고, 2018년 8월 2일 15시의 데이터가 test set에 있다면 Interpolation으로 어느정도 값을 맞출 수 있고 이로인해 test score가 잘 나온다고 착각할 수 있다. 그러나 예측 문제에서는 낮은 성능을 보일 가능성이 농후하다. 이와 관련하여 간단한 실험을 해 보았다. 다음은 Time Split으로 Cross Validation을 한 결과이다.

[실험]

총 데이터 : 2017년 7월 1일 00시~2019년 6월30일 00시 (#data : 17435)

test 구간 : 2018년 7월 13일 00시 ~ 2018년 8월23일 23시 (#data : 1008)

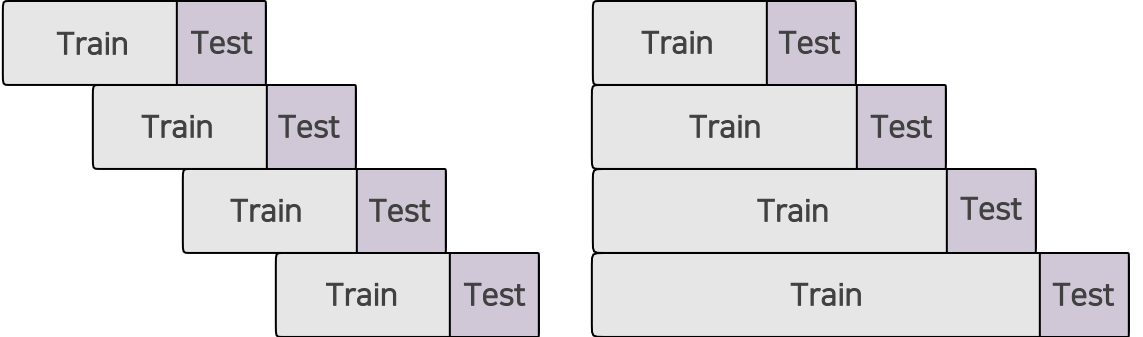
<Figure2. 실험1 train range가 test이외의 모든 구간>

<Figure3. 실험2 train range가 test이전 (2017.7.1~2018.7.12) >

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Train 구간 | Train r2 | Train nMAE | Test r2 | Test nMAE |
| test제외 모든 구간(실험1) | 0.743 | 0.102 | 0.271 | 0.109 |
| Test 이전 구간(실험2) | 0.83 | 0.078 | -0.128 | 0.159 |

실험 1의 경우 실제로 두가지의 경우로 해석할 수 있다. 첫번째는 2년간의 구간을 보고 비슷한 구간을 외워 값을 뱉었을 수도 있고, 앞뒤로 interpolation한 것을 뱉었을 수도 있다. 실제로 실험2의 방법대로 예측해야 하는데, 일반 CV로 했을 경우 실험1의 경우가 발생할 수 있으므로 CV는 적절하지 않다.

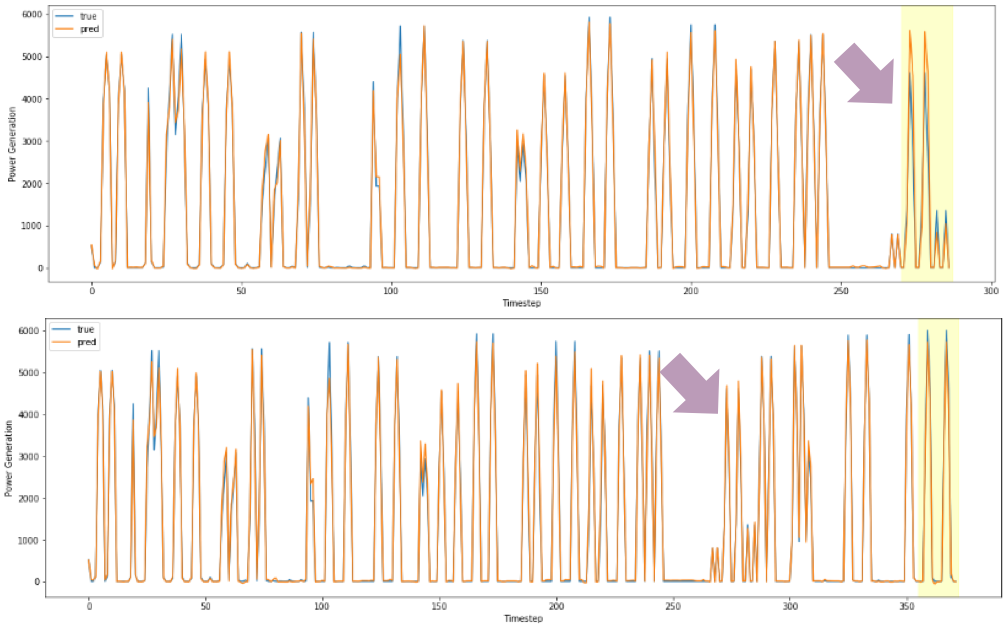
따라서 **모델은 과거를 보고 미래를 예측하는, 실제 상황에서 일어나는 시나리오로만 검증되어야 한다.** 아래와 같은 두 가지 방법이 있다.



<Figure. 4> Walk-Forward Optimization, Nested Cross Validation

1) Walk-Forward Optimization

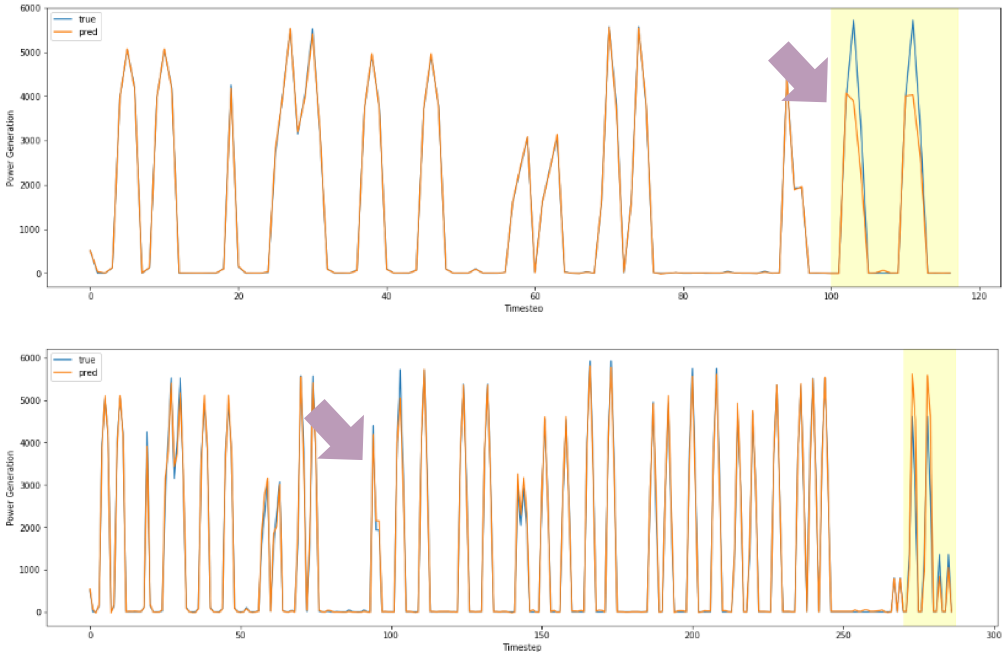
고정된 길이의 Train Set을 보고, 이어지는 Test Set을 예측하도록 하여 모델의 성능을 검증하는 방법. Time Series Forecasting Model을 트레이닝하는 두가지 방법 중 하나이기도 하다. 전 구간이 아닌 주어진 구간만을 학습하므로 **최신의 Distribution만을 학습할 수 있다는 장점이 있다.** 하지만 Train Set의 길이가 충분히 길지 않을 경우, **모델이 주어진 데이터를 설명할만큼 Generalize되지 못할 수 있다는 단점이 있다.**



<Figure 5.> Walk-Forward Optimization의 예

2) Nested Cross Validation

데이터의 시작 구간에서부터 해당 Fold의 끝까지 이어지는 Train Set을 보고, 이어지는 Test Set을 예측하도록 하여 모델의 성능을 검증하는 방법. Forecasting 모델을 트레이닝시키는 방법으로도 활용된다. **Test Set이전의 모든 데이터를 학습함으로 모델에게 충분한 데이터를 제공할 수 있다는 장점이 있다**. 반면, **Non-Stationary한 Distribution을 다루는 경우에 현재와 다른 Distribution에서 온 과거의 데이터가 학습을 방해해 현재 시점에서 가까운 Sample을 학습, 예측하는 성능이 저하될 수 있다.**



<Figure6.> Nested-Cross Validation의 예

c. Evaluation Metrics for Time Series Forecasting

|  |  |  |
| --- | --- | --- |
| nMAE = 0 | R2 = 1 | 모든 시간에 대한 예측 = 모든 시간에 대한 실제값 |
| nMAE > 0 | ... | R2가 커질수록 오차가 점점 작아짐 |
| R2 = 0 | 예측 총 발전량 = 실제 총 발전량 |
| R2 < 0 | 예측 총 발전량 != 실제 총 발전량 |
| … | R2가 작아질수록 오차가 점점 커진다 |

<Figure. 7> nMAE와 R2의 관계

1) R2

Regression문제에서 모델의 성능을 평가하는 지표[[3]](#footnote-4). In-Sample Train, In-Sample Test Set을 가져 Train Set과 Test Set의 평균값이 같은 Regression에서는, 단순히 평균으로 예측하는것 보다 얼마나 잘 예측하느냐가 모델의 성능을 결정하는 중요한 지표가 된다. R2의 범위는 최대값 1, 최소값은 없다. **R2가 1일때 모델은 모든 데이터를 완벽하게 설명하고 있으며, R2가 0일때 모델은 데이터의 총합은 예측하나, 개별 예측은 전혀 수행하지 못함을 의미한다.** R2가 음수일때는 모델이 데이터의 총합(평균값, 여기서는 총 발전량)을 예측하지 못한 것이다. 모델의 성능이 어느정도 보장될 경우에 R2는 0에서 1의 범위를 가지므로, 모델의 객관적인 성능을 빠르게 평가할 수 있는 장점이 있다.

In Sample Train, Out-of-Sample Test를 가지는 Time Series Forecasting에서는 Test Set의 총량(평균값)조차 맞추지 못할 가능성이 높다. 태양광 발전량 예측 문제에서는 R2가 0이상이면 총 발전량을 맞춘 것이고, R2가 1이면 모든 1시간 단위의 발전량을 완벽하게 맞춘 것을 의미한다.

**태양광 발전량과 같이 어느정도 성능이 보장되어 있는 Regression, 혹은 Forecasting 문제에서는 평가 지표로 R2를 사용하여도 무방하다.**

2) MAE

Forecasting 문제에서 모델의 성능을 평가하는 지표[[4]](#footnote-5). In-Sample Train, Out-of-Sample Test를 가져 Train Set과 Test Set의 평균값이 다를 때 주로 사용한다[[5]](#footnote-6). 최소값이 0이고 최대값이 없으므로 상대적인 수치를 통한 모델간 비교만 가능하여 직관적인 해석이 어렵다. 대부분의 Forecasting문제에서 예측 총량을 맞춰 0 이상의 R2를 내놓는 모델을 만드는것이 어렵기 때문에 주로 사용된다. MAE기반의 Metric을 사용하는 또다른 이유는, 다음과 같은 시나리오 때문이다. 모든 예측값이 실제값보다 0.001씩 높을 경우, 예측값의 평균과 실제값의 평균은 0.001의 차이가 나게 된다. 이때 R2는 음수인데, 실제로 모델의 에러는 적기 때문에 모든 예측을 평균값으로 출력한 모델보다 성능이 떨어진다고 단언할 수 없다.

3) Conclusion

풍력 발전량 예측 문제에서 0 이상의R2는 적어도 54시간 간의 총 발전량을 맞췄음을 의미하고, 낮은 MAE는 총 발전량을 맞췄는지 여부는 알 수 없지만 개별 예측의 에러가 적었음을 의미한다. **총 발전량을 맞추는 것이 중요한지, 개별 예측의 에러가 적은 것이 중요한지에 따라 적합한 Metric을 사용하여 모델을 평가하여야 한다. Baseline Model에서 0이상의 R2가 보장된다면 R2를 쓰는것이 결과 해석에 편리하다.**

d. Prediction Uncertainty

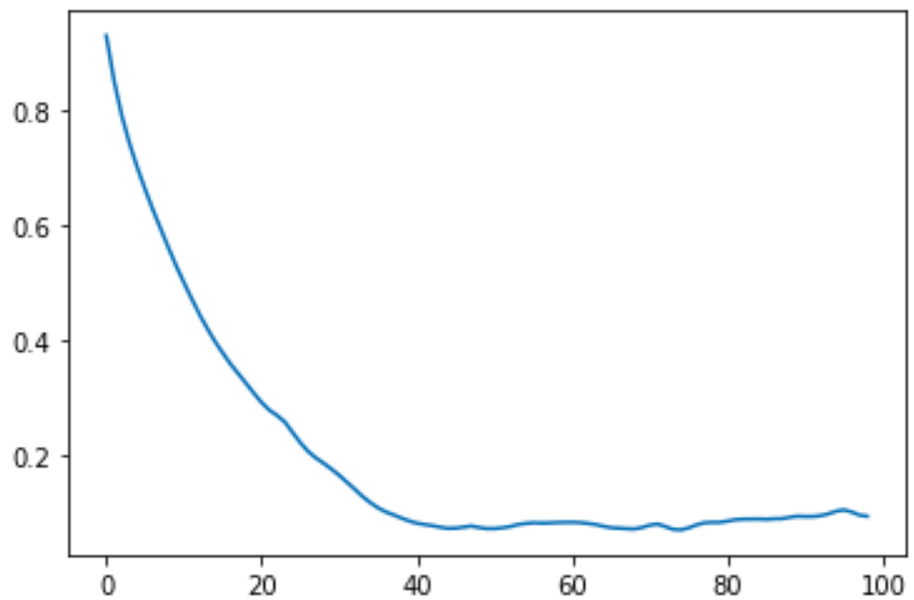
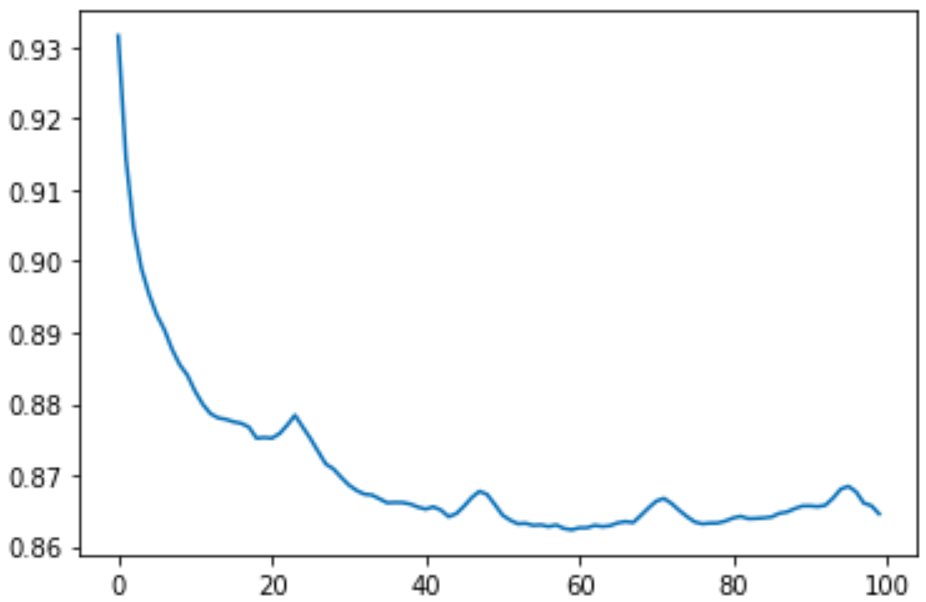
Non-stationary한 Distribution을 다루는 Forecasting 문제의 특성상 Regression보다 모델이 예측을 맞추지 못할 가능성이 높다. 따라서 Test Set이 Out-of-Distribution Sample인지를 확인하여 예측이 틀릴 가능성(Uncertainty)를 제공하는것이 중요하다. Out-of-Distribution Detection[[6]](#footnote-7), Variational Autoencoder를 활용한 Novelty Detection[[7]](#footnote-8)등을 통해 사용자에게 예측의 불확실성을 제공할수 있다. 본 보고서에서는 분석시간의 제약으로 인해 빠르게 적용해볼수 있는 One-Class Support Vector Machine을 통해 Novelty Score를 계산하였다. 남은 기간동안 VAE를 활용한 Novelty Detection을 적용하여 Uncertainty의 신뢰도를 높이고자 한다.

**2. 탐색적 데이터 분석(Exploratory Data Analysis)**

데이터의 특성을 확인해보기 위해 kpx에서 제공받은 한시간 간격의 제주 서귀포 지역 관측치 기상 데이터(이하 kpx데이터)와 발전량 데이터와 같은 지역의 1분 단위 관측치 기상 데이터(이하 kma 데이터)를 사용하여 데이터의 특징을 찾아보고자 탐색적 데이터 분석을 시도했다.

추가적으로, 한국 남부발전 사이트에서 4년치의 제주도 제주시 성산읍 수산 풍력 발전량과 제주도 제주시 한경면 풍력 발전량을 얻게 되어 사용하였다.

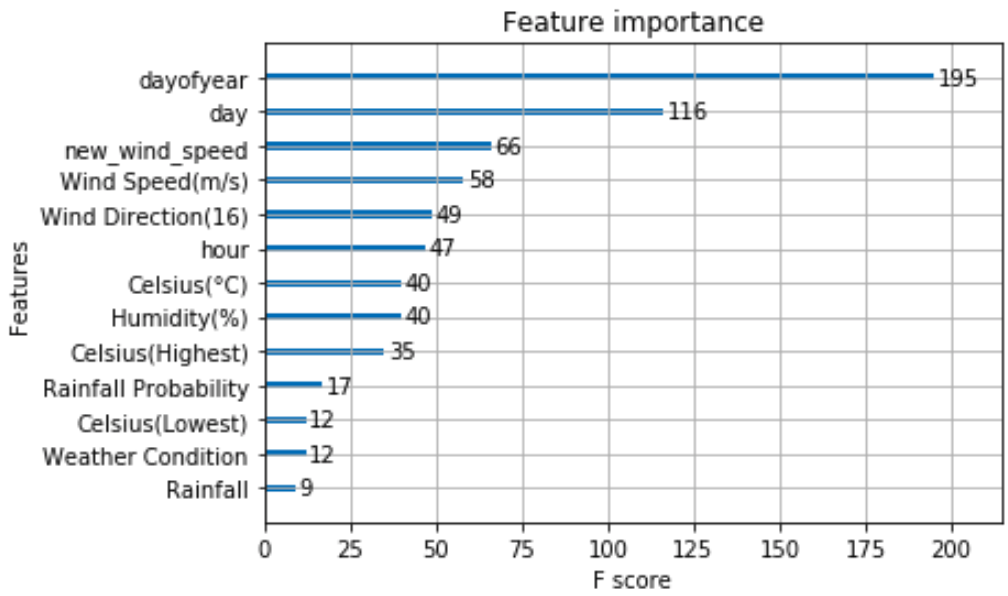
**가. 발전량 데이터의 경향**

****

<Figure7.> 시간별 Auto Correaltion, Auto-Normalized Mutual Information

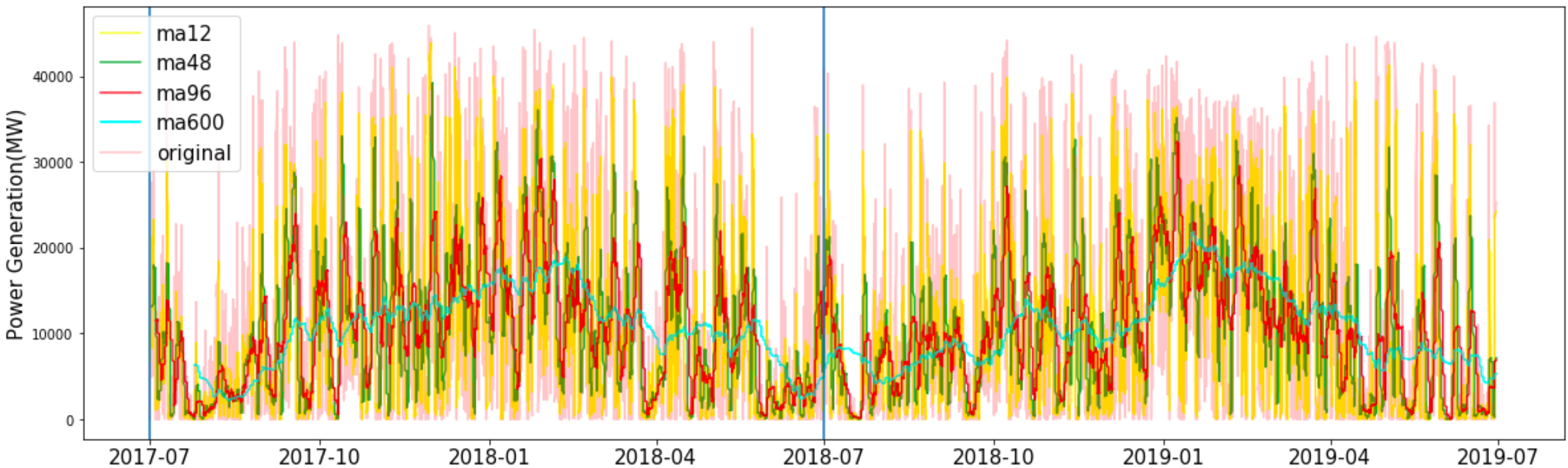
풍력 발전량이 시계열적인 특성을 가지는지 확인해보기 위해 Auto Correaltion, Auto Normalized Mutual Information을 확인하였다.

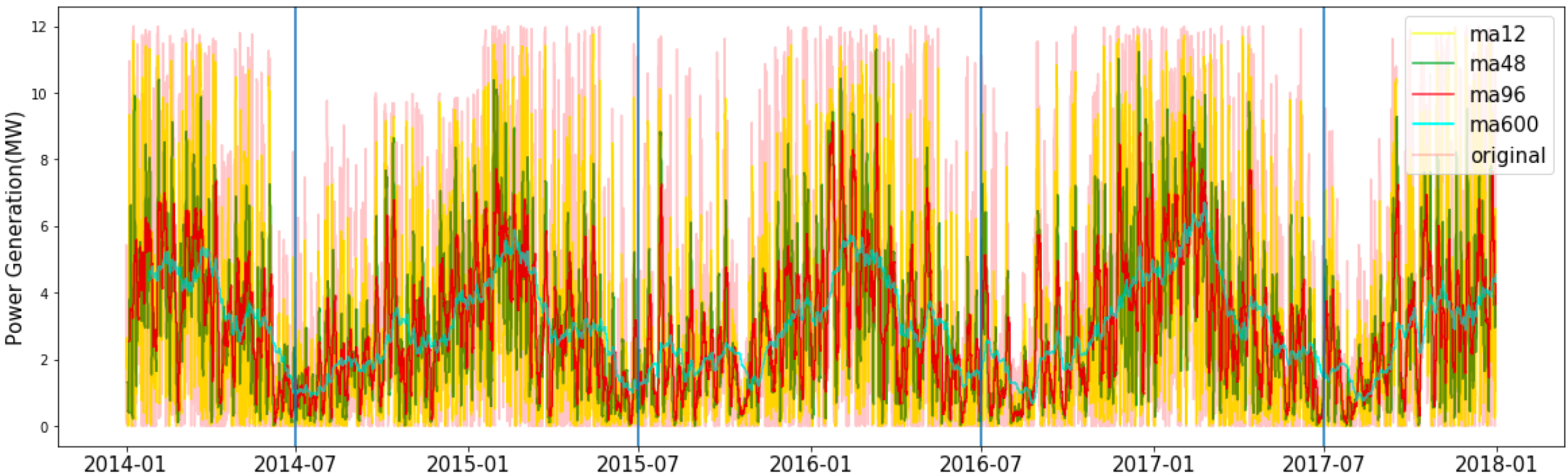
발전량 데이터의 경향성을 확인해 보았을때, 40시간(약2일) 정도 동안은 이전 시간의 영향을 받는다는 것을 확인할 수 있었다. Time Lag(가로축)이 40시간을 넘어가면서 매우 낮은 수준을 상회하는 것을 확인했고, 이후 시간과 관련된 Feature Engineering(Windowing, n hours mean)을 할 때 이 시간 간격을 활용하였다.

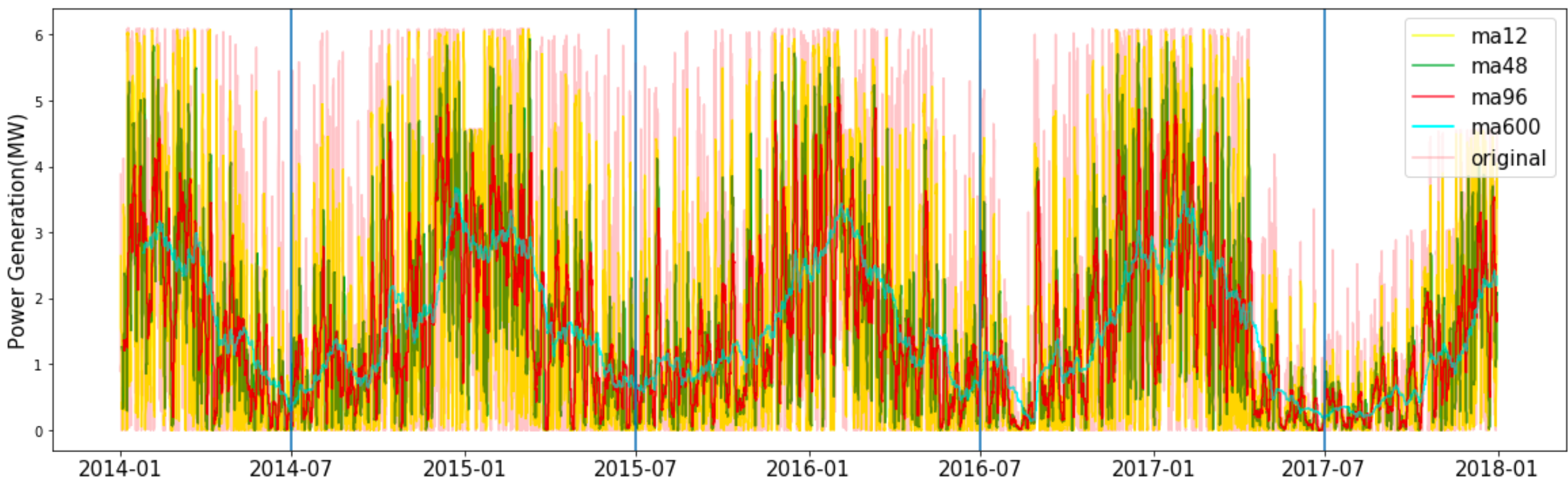


<Figure8.> Feature Importance

Baseline Model로 Feature Imporatance를 확인한 결과 dyaofyear가 가장 높은것을 확인할 수 있었다. dayofyear란 현재 날짜가 1년 기준 몇 일에 해당하는지를 나타내는 변수이며 1~365로 이루어져있다. 즉 1년간 어떤 주기가 있을 것이다라고 추측해볼수 있고 추가적으로 전 구간 발전량에 대해 moving average를 그려보았다.

<Figure9. 제주도 서귀포시 표선면 가시리 풍력 발전량 2017-7~2019-7(2년)>

<Figure10. 제주도 제주시 성산읍 수산 풍력 발전량 2014-1~2018-1(4년)>

<Figure11. 제주도 제주시 한경면 풍력 발전량 2014-1~2018-1(4년)>

위 Figure는 3개의 지역에 대해 발전량 데이터의 plot을 그려본 것이다. Original Plot(moving average가 아닌 발전량 자체)을 찍어 봤을 때 특별한 경향을 찾을 수 없었지만, 12시간, 48시간(2일), 96시간(4일), 600(25일) 간격 moving average를 그려봤을때 cycle을 찾을 수 있었다. 전체적으로 7 월달이 가장 낮고 겨울인 1월 즈음이 가장 높은것을 확인할 수 있었다. 즉 풍력도 결국 시간적 요인의 영향을 받 는다는 것을 알 수 있다. 현재 제공받은 제주도 표선면 가시리의 2년간 발전량 데이터로 미래의 값을 예측을 한다는 것은 실제로 2 배치를 보고 모델을 학습시켜 결과를 내는것과 같아서 모델이 충분한 규칙을 배우기 어려움이 있다. 따라서 이를 극복하기 위해 데이터를 늘리고 cycle을 학습할 수 있는 여러가지 autoregressive한 모델과 과거의 정보를 활용할 수 있는 다양한 EDA방법을 적용하여 문제를 풀고자 한다.

**나. 기존 변수와 발전량과의 관계**

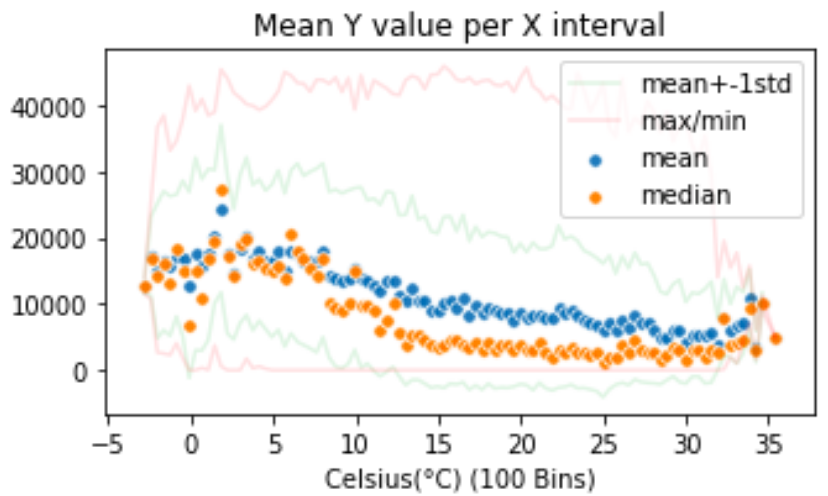
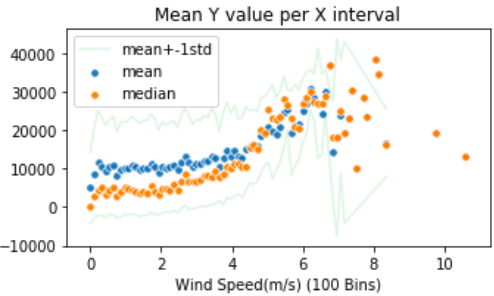
**kpx에서 제공받은 한시간 단위 데이터**

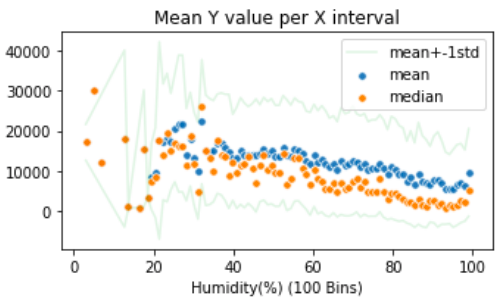
(1) Correlation을 통한 선형 관계 탐색

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 변수 | 풍속^3 | 풍속 | 기온 | 풍향 | 습도 |
| Correlation | 0.13 | 0.12 | -0.31 | 0.1 | -0.2 |

○ Correlation은 변수간의 선형관계를 나타내는 지표이며 -1~1의 범위를 가진다. 0은 선형관계가 없는 것이며, 1은 양의 선형관계로 데이터가 설명됨을 의미한다. 위의 Correlation값은 각 변수가 발전량과 선형관계가 거의 없음을 보여주었다.

(2) Normalized Mutual Information(NMI)을 통한 비선형 관계 탐색



<figure 12 . 각 변수별 구간별 발전량 평균>

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 변수 | 기온 | 습도 | 풍향 | 풍속 | 월 | 일 | 시간 |
| NMI | 0.67 | 0.54 | 0.43 | 0.53 | 0.42 | 0.50 | 0.48 |

○ 의외로 발전량과 풍속간의 Normalized Mutual Information[[8]](#footnote-9)이 기온, 습도보다 낮았다. 실제로 <figure12>를 보면 이는 각 데이터는 moving average로 보면 크게 보면 경향성이 존재하지만 noise가 많이 껴있는 상태이기에 관계가 없는 것처럼 보일 수 있다.

**나. Feature Engineering으로 생성된 변수와 발전량과의 관계**

**1시간 단위 기상 관측치 + 1분 단위 기상 관측치**

현재 제공받은 데이터(kpx데이터)는 1시간 단위의 스냅샷 기상 관측 정보와 1시간 동안 생산된 발전량의 데이터이다. 그러나 이 관측치 데이터는 1시간 동안의 기상정보를 대표하지 못하므로 1분 단위의 기상 관측치(kma데이터)를 1시간 동안 평균 낸 데이터를 사용하여 경향성과 관계를 보고자 하였다.

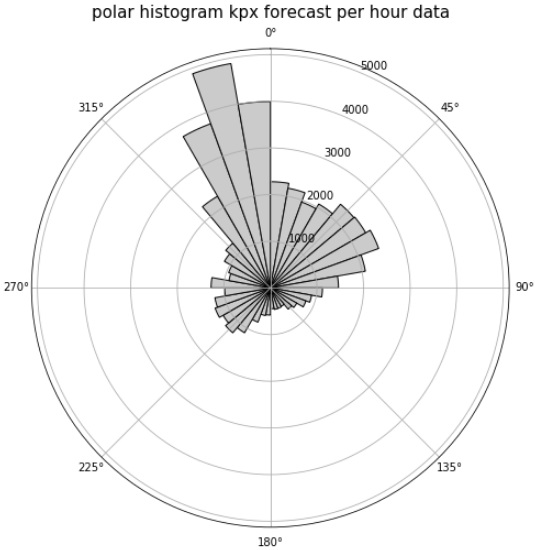
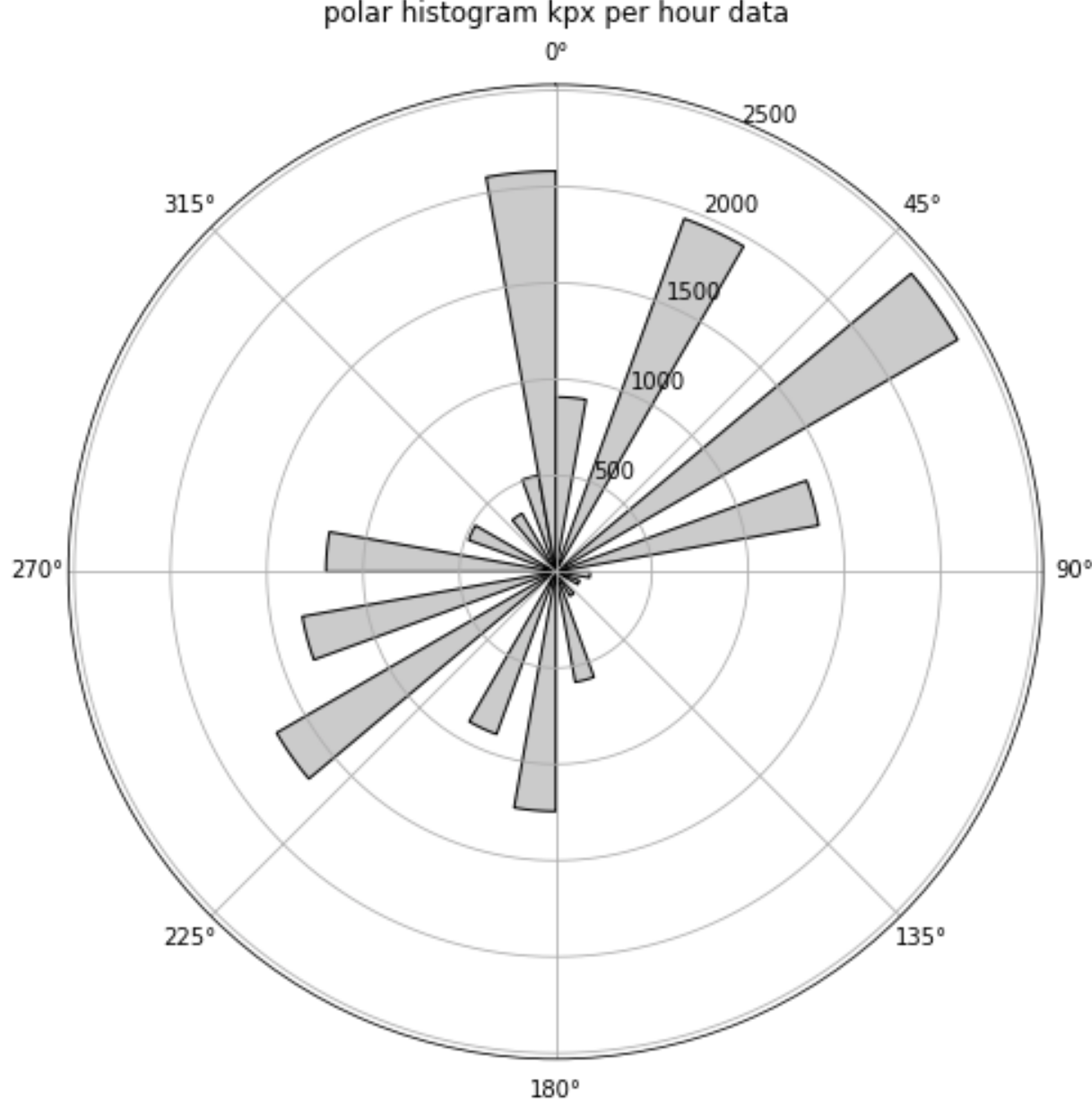
○ Feature Engineering을 하는 목적은 도메인 지식을 활용해 우리가 예측하고자 하는 값(y)를 잘 설명하는 변수를 생성하는 것이다. 이를 통해 모델의 Optimization 부담을 줄이고, 예측 성능을 높인다. Feature Engineering으로 생성된 새로운 변수가 예측하고자 하는 값(y)을 잘 설명하는지를 확인하기 위해 다음과 같은 방법을 사용하였다. XGBosot의 예측 모델 성능 비교를 통해 실제로 얼마나 성능 향상에 기여하는지도 확인하였다.

(1) 풍향, 풍속에 대한 인코딩

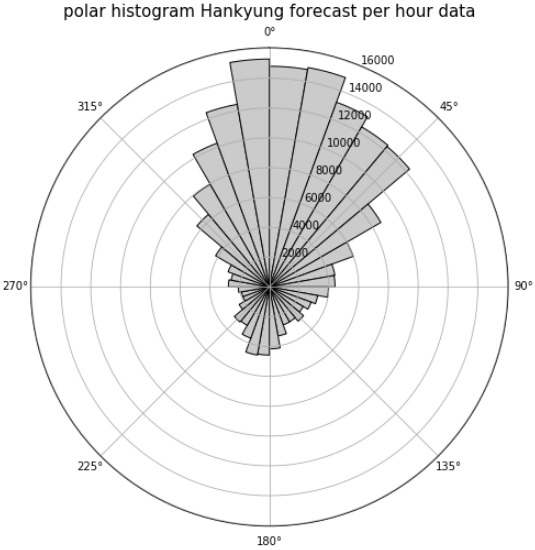
가. 풍향에 cos, sin을 씌움

○ 현재 풍향은 16방위로 numeric type으로 들어가 있지만 이것은 실제 0도와 360도가 같음에도 다르게 표현하는 문제가 발생한다. 따라서 radian 개념을 도입하여 이 문제를 해결하고자 한다.

나. 풍향을 고려한 풍속 인코딩



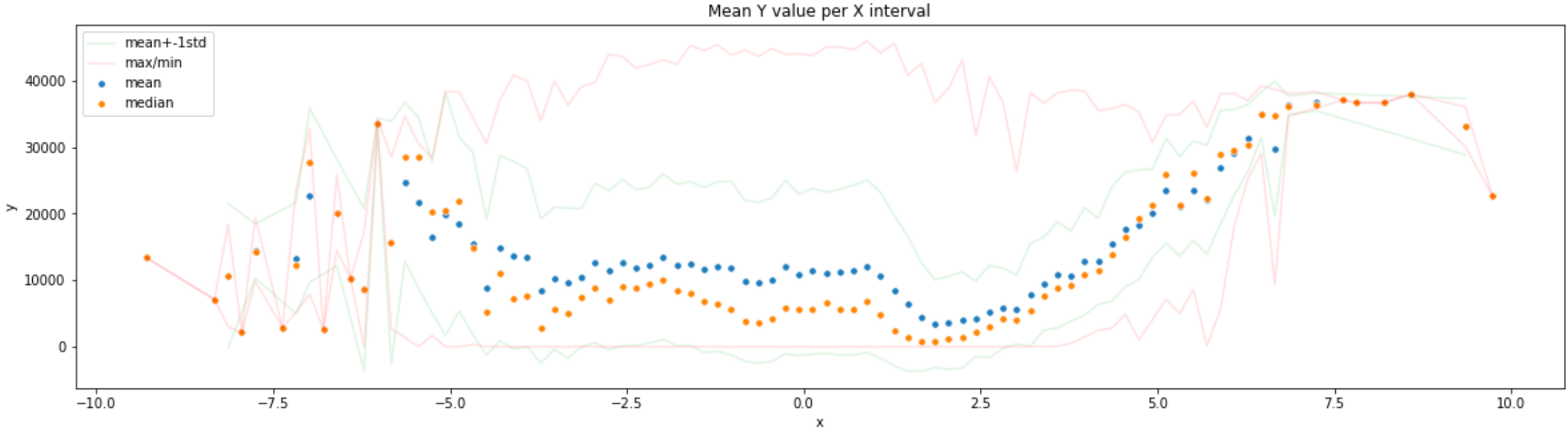
<Figure 13. 가시리 지역 관측치 풍향별 평균 발전량, 예보 데이터 풍향별 평균 발전량>



<Figure 13. 제주 한경 지역 풍향별 평균 발전량>

○ 풍력 터빈은 바람에 의해 돌아가고, 풍속이 높을수록 풍력 발전량이 증가한다는 사실이 알려져있다. 이때 패널이 놓여있는 방향과 바람이 불어오는 방향에 따라 같은 풍속에서도 터빈효율이 달라질 수 있다. 가장 많은 데이터가 갖고 있는 풍향을 발전기가 바라보는 방향이라고 가정하여 Φ라 놓고 현재 데이터의 풍향을 θ라 하면 새로운 풍속(new\_wind\_speed) = 기존 풍속\*(θ - Φ)로 인코딩할 수 있다.

따라서 우리는 히스토그램을 통해 데이터에서 가장 많은 값을 차지하는 50°를 Φ라 가정하고 풍속을 인코딩하여 new\_wind\_speed라는 feature를 추가하였다.

<Figure 14. kma데이터에 대해 새로 인코딩한 풍속의 구간별 발전량 평균>

발전량과 인코딩된 변수들간의 mutual information

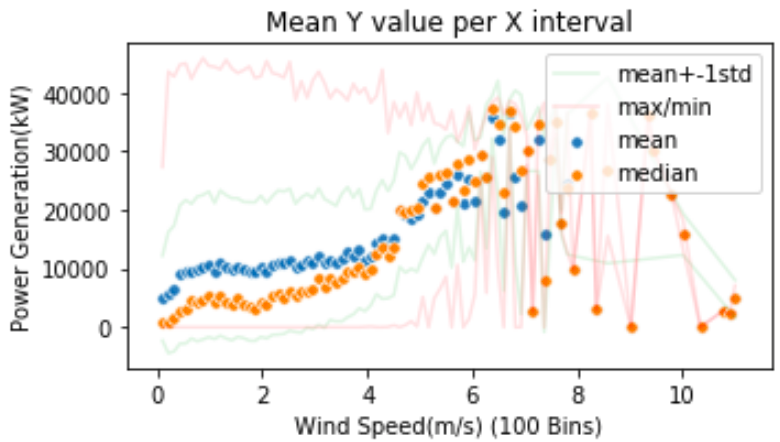
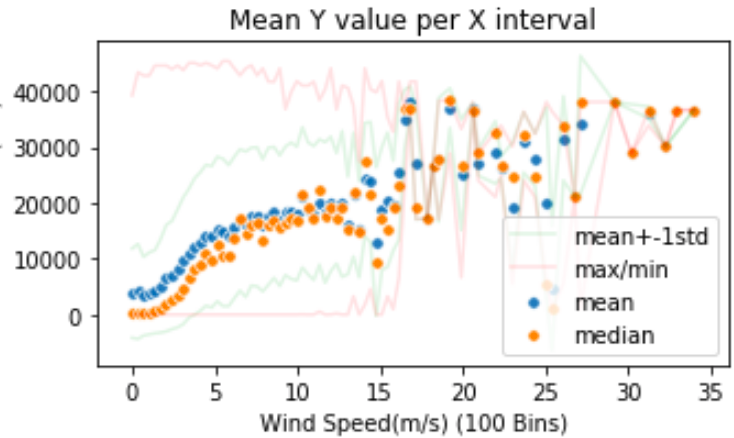
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 변수 | 풍속 | new\_wind\_speed | 풍향 | 풍향 cos | 풍향 sin |
| NMI kpx | 0.53 | 0.69 | 0.43 | 0.42 | 0.4 |

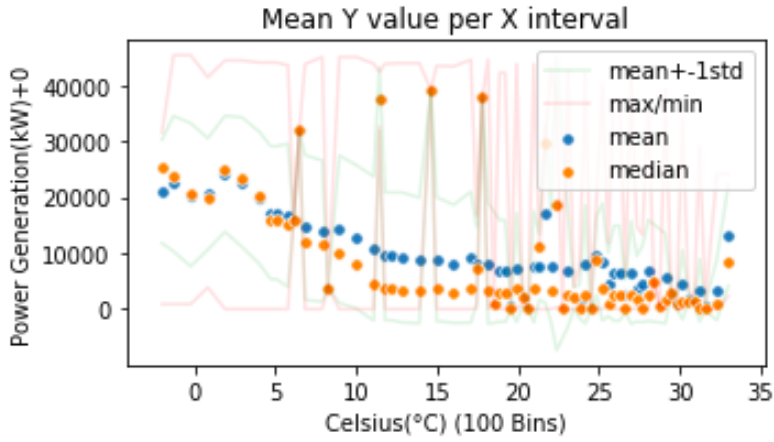
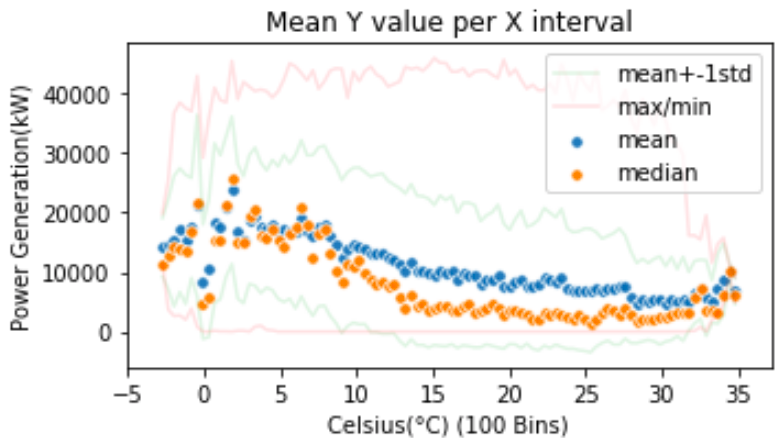
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| NMI kma | 0.61 | 0.93 | 0.66 | 0.93 | 0.93 |

<Figure 15. 인코딩된 풍속과 발전량의 관계>

○ 각도별로 분해된 풍속과 발전량의 관계를 보았을때, 풍속이 증가함에 따라 발전량이 유의미하게 증가하는 경향이 보였다. 추가적으로 풍속은 음수를 가질 수 없기에 기존 풍속\*(cos(θ - Φ)^2)을 사용하여 양수로 만들고 관계를 확인해봤지만, mutual information이 0.83으로 0.92보다 낮았고, 따라서 제곱하지 않은 음수가 나오는 값 그대로 사용하기로 하였다.

**다. 관측치와 예보의 차이**

<Figure16.> 예보에서 풍속-발전량(kW)의 관계, 관측치에서 풍속-발전량(kW)의 관계

<Figure17.> 예보에서 온도-발전량(kW)의 관계, 관측치에서 온도-발전량(kW)의 관계

○ 전체적인 흐름은 비슷한걸 보아 기상관측치-발전량과의 관계를 통해 각 변수가 풍력 발전량에 어떻게 영향을 미치는지 인사이트를 얻을 수 있었으나, 기상예보의 불확실성 때문에 기상예보-발전량과의 관계에서도 같은 영향을 미칠 것이라고 확인하기는 어려웠다. 관측치에서 발견한 관계들을 통해 Feature Engineering을 수행하였다.

**3. Modeling**

**가. Baseline Model**

1. 실험결과

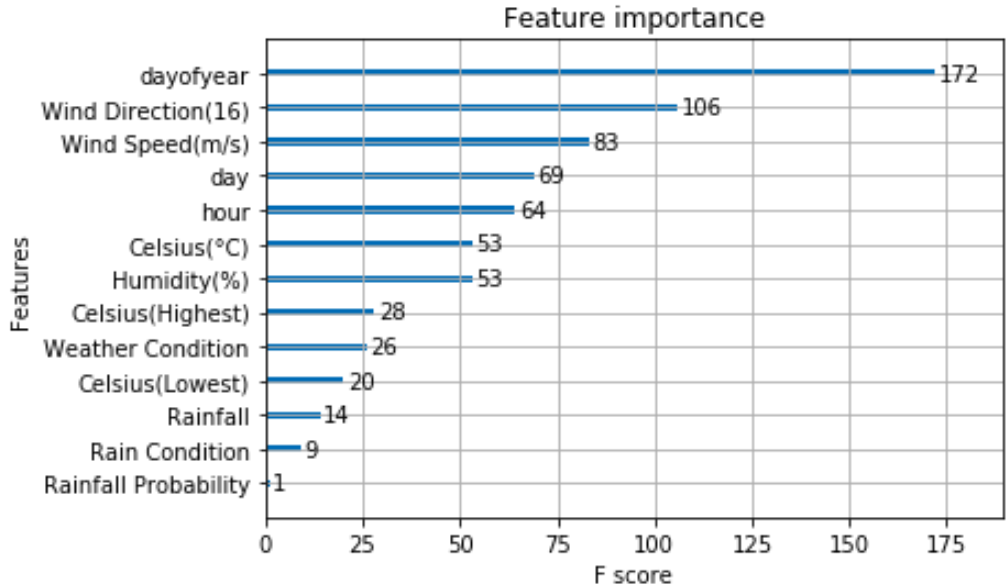
○ 반복적인 실험을 통한 Feature Engineering검증을 위해 Baseline Model로는 Training과Prediction이 빠른 XGBoost를 사용하였다. kpx에서 제공받은 가시리 지역의 예보 데이터를 사용하였고 모든 데이터를 사용하기엔 빠른 실험이 불가능하여 14시에 발표한 예보 데이터만을 가지고 사용했다. 검증 방법은 Walk Forward방식으로 총 데이터를 50 구간으로 나누고 10개의 fold를 간격으로 움직이며 예측하도록 하였다. 즉 각 발전량의 경향이 존재하므로 구간별로 training과 test score의 차이가 따로 있는지 확인하고자 하였다. 아래와 같이 Input, Ouput Data를 정의하였으며 모델의 성능을 객관적으로 평가하기 위해 평가지표는 R2를 사용하였다.

Input : 해당 시간의 기상예보 (Interval : 3시간)

Output : 해당 시간의 발전량, 해당시간+1시간의 발전량, 해당시간+2시간의 발전량

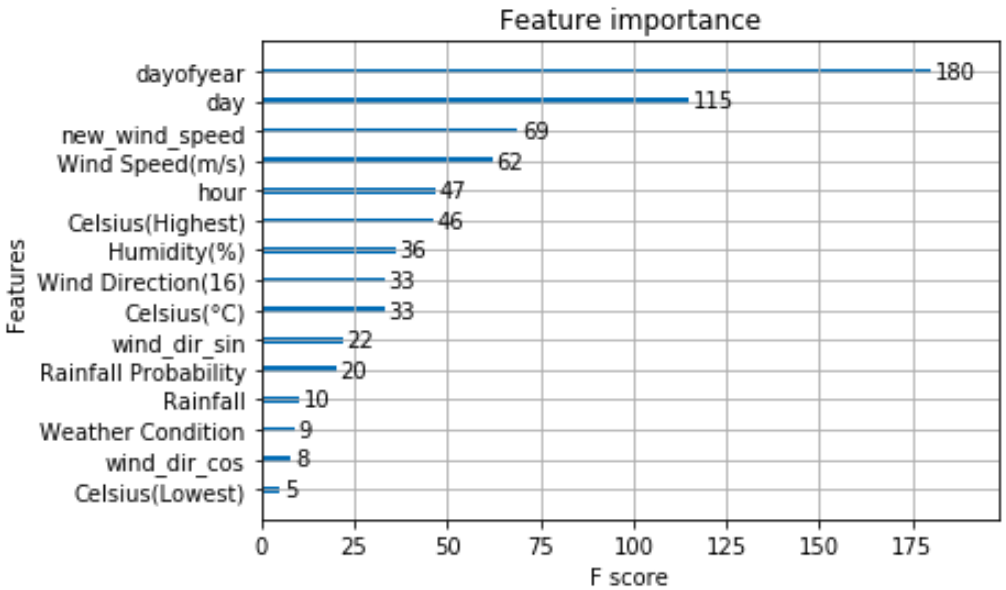
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | 사용모델 | 사용변수 | Train R2 | Test R2 | Train nMAE | Test nMAE |
| 기상예보 | XGBoost | 기온, 전운량, 습도, 강수량, 적설량, 풍향, 풍속 | -0.16 | -5.61 | 0.13 | 0.19 |
| 기상예보+시간 | 기온, 전운량, 습도, 강수량, 적설량, 풍향, 풍속, 월, 일, 시간, 날짜 | 0.3435 | -2.3558 | 0.1106 | 0.1815 |

(2) Baseline Model 선택

<Figure 18 > Baseline Model에서의 Feature Importance

**나. Feature Engineering**

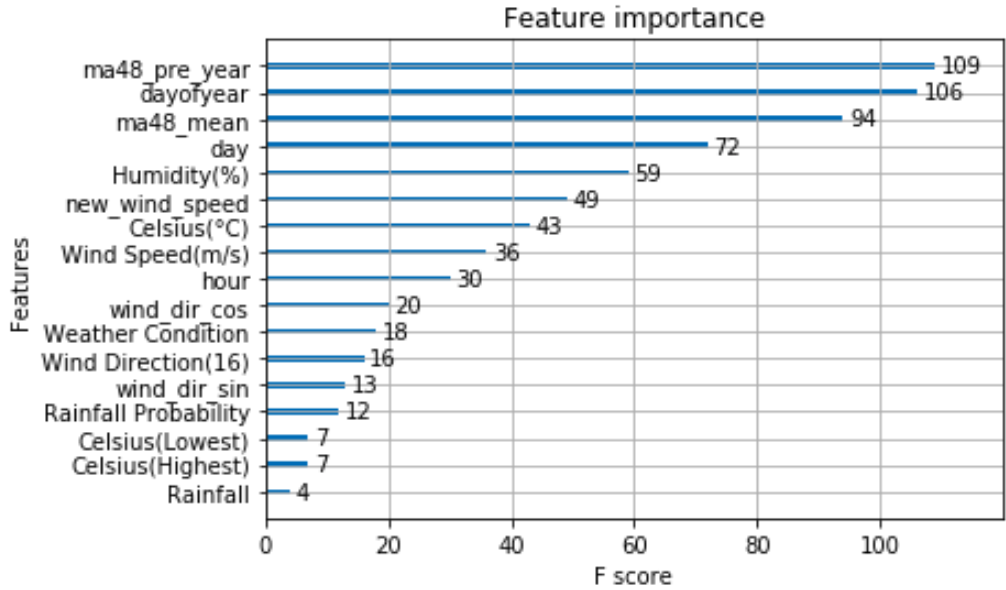
(1) 풍향 풍속 인코딩

<Figure 19. 풍향 풍속 인코딩 변수를 추가했을 때 Feature Importance>

위에서 언급한 풍향 풍속의 인코딩값을 feature로 추가하였을 때 new\_wind\_speed의 Feature importance가 올라가며 test R2도 개선되는걸 알 수 있다. 추가적으로 new\_dir\_cos, new\_dir\_sin만 제거하고 테스트 해봤는데, 성능이 떨어짐을 알 수 있었다.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Train R2 | Test R2 | Train mae | Test mae |
| score | 0.3749 | -2.2549 | 0.1087 | 0.1798 |

(2) 발전량의 Moving Average추가

<Figure 20. 발전량의 Moving Average를 추가했을 때 Feature Importance>

풍향 풍속 인코딩은 고정으로 사용하였고, 전년도 발전량의 48시간 동안의 moving average와 현재 가지고 있는 발전량 데이터의 2년간 평균을 feature로 추가하였다. 위에서 확인한대로 발전량의 cycle정보를 직접적으로 넣어주기 위함이었으며, 예상했던대로 Feature Importance가 가장 큰 것을 확인할 수 있다.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Train R2 | Test R2 | Train mae | Test mae |
| score | 0.4477 | -2.1686 | 0.1037 | 0.1757 |

(3) Windowing

1시간 단위로는 데이터에 노이즈가 너무 많이 껴서 인지, 데이터의 경향을 못잡고 있을 수도 있다고 판단했다. 따라서 mutual information, auto correlation plot에서 확인한 결과를 바탕으로 이전 시간대의 feature들을 사용하려는 시도를 하였다.

풍향 풍속 인코딩과 발전량의 moving average는 고정으로 사용하였고 Autocorrelation을 통해서 확인하였듯이 이전 Timestep의 기상이 현재에도 영향을 줄 수 있기 때문에, 6시간 전, 12시간 전, 24시간 전, 48시간 전의 기상상태를 Feature로 추가하였다.

각각에 대하여 시간별로 비교하였는데, 6시간은 추가하여도 별 효과가 없었고 12시간과 24시간을 넣는게 가장 높은 score를 보였다. 따라서 실제 예측 모델에서는 이전에 48시간을 넣었던 발전량 moving average에 추가로 12, 24시간동안의 moving average도 feature로 추가할 것이다.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Train R2 | Test R2 | Train mae | Test mae |
| 12,24,48 | 0.7195 | -1.1719 | 0.0811 | 0.1500 |
| 6,12,24,48 | 0.7025 | -1.1807 | 0.0828 | 0.1540 |
| 24,48 | 0.6429 | -1.2892 | 0.0891 | 0.1512 |
| 12,24 | 0.6927 | -0.8749 | 0.0844 | 0.1411 |

(4) 이전 시간의 모든 변수의 moving average

풍향 풍속 인코딩과 발전량의 moving average는 고정으로 사용하였고 여기에서 아이디어를 얻어 모든 변수에도 moving average도 적용해보았다. 이때 12시간, 24시간, 48시간의 moving average를 사용하였다.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Train R2 | Test R2 | Train mae | Test mae |
| 24 | 0.6371 | -1.9250 | 0.0888 | 0.1993 |
| 12 | 0.6912 | -1.0955 | 0.0844 | 0.1494 |

(5) Windowing과 Moving Average 모두 적용

Feature Engineering (3)번째와 (4)번째 방식에서 확인하였듯이, 각 feature를 그대로 사용하는 Windowing과 Moving Average사용, 그리고 전년도 발전량의 moving average사용의 효과를 결합하여 사용해보고자 실험해보았다.

12시간, 24시간 이전의 Feature Windowing과 12시간 동안의 Feature moving average의 결합, 12시간 24시간 이전의 Feature Windowing과 24시간 동안의 Feature Moving average를 비교해보았다.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Train R2 | Test R2 | Train mae | Test mae |
| 12,24 / 12 | 0.4173 | -2.900 | 0.1057 | 0.1790 |
| 12,24 / 24 | 0.7617 | -1.2175 | 0.0764 | 0.1462 |

결론적으로 Feature Engineering의 결과는 풍향/풍속의 인코딩 추가, 전년도와 전체 평균의 발전량 moving average, 모든 feature들에 대한 Windowing이 가장 높은 성능을 보였다.

**다. 타 발전소 데이터 활용**

남은 기간동안 공개되어있는 다른 풍력 발전소 데이터를 Training Set에 추가하여 실험할 예정이다. 방법으로는 발전량 데이터를 MinMax Scaling으로 0과 1 사이의 범위로 만들어준 뒤, 가시리 지역 풍력발전기의 최대용량인 45를 곱하여 스케일을 맞춰준 후에 사용할 수 있다. 기상 데이터를 사용할 수도 있을것이지만 단순히 두 지역의 데이터를 그대로 사용하기 보다는 우선은 위 그림에서 확인했듯이 발전량은 다른 지역에 대해서도 어느정도 비슷한 경향을 갖고 있으니 발전량 데이터의 moving average를 feature로 사용할 수 있을 것이다.

또한 간단한 데이터 병합을 통해 타 발전소 데이터가 기존 모델의 성능을 향상시킨다는 것을 확인한 후, 남은 기간동안 Conditional Variational Autoencoder를 통한 발전량 Encoding으로 발전량에 내재된 위치정보, 설비정보를 제거하고 Encoding된 발전량을 맞추도록 트레이닝하여 모델 성능을 더 향상시키고자 한다.

**라. 1차 제출 예측 모델**

(1) Feature Engineering

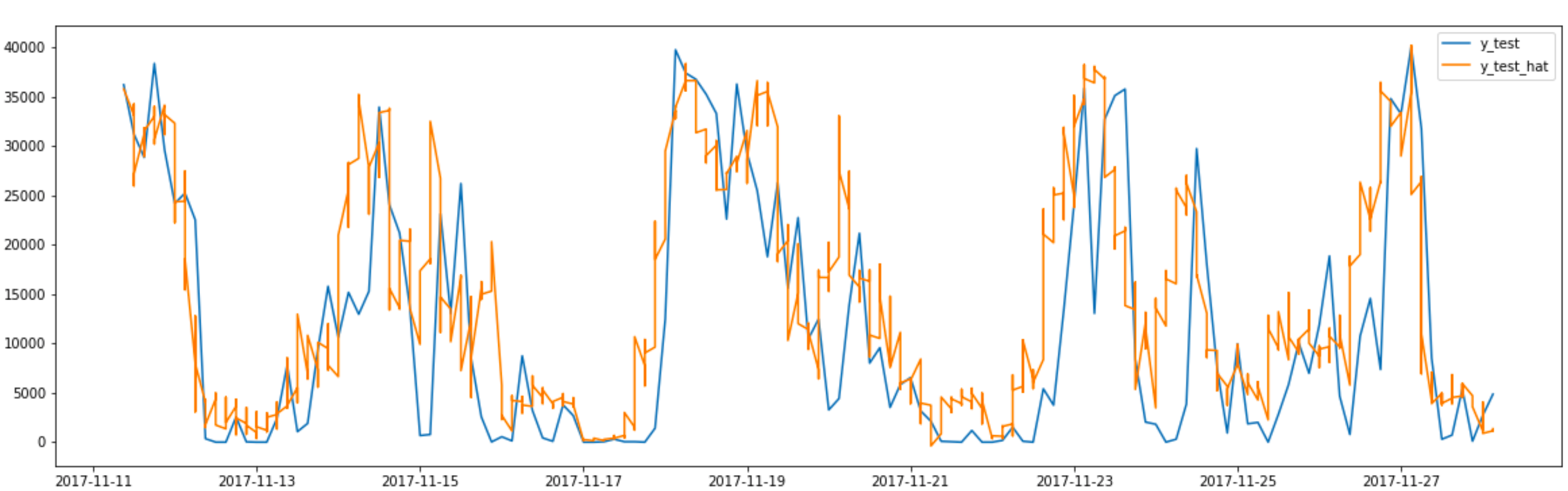
|  |
| --- |
| Feature Engineering |
| 풍향 풍속 인코딩  모든 변수에 대해 이전 12시간, 24시간 Window Feature 추가  전년도 발전량의 12시간, 24시간, 48시간 Moving Average  발전량 12시간, 24시간, 48시간 Moving Average의 총 평균  시간 정보 |

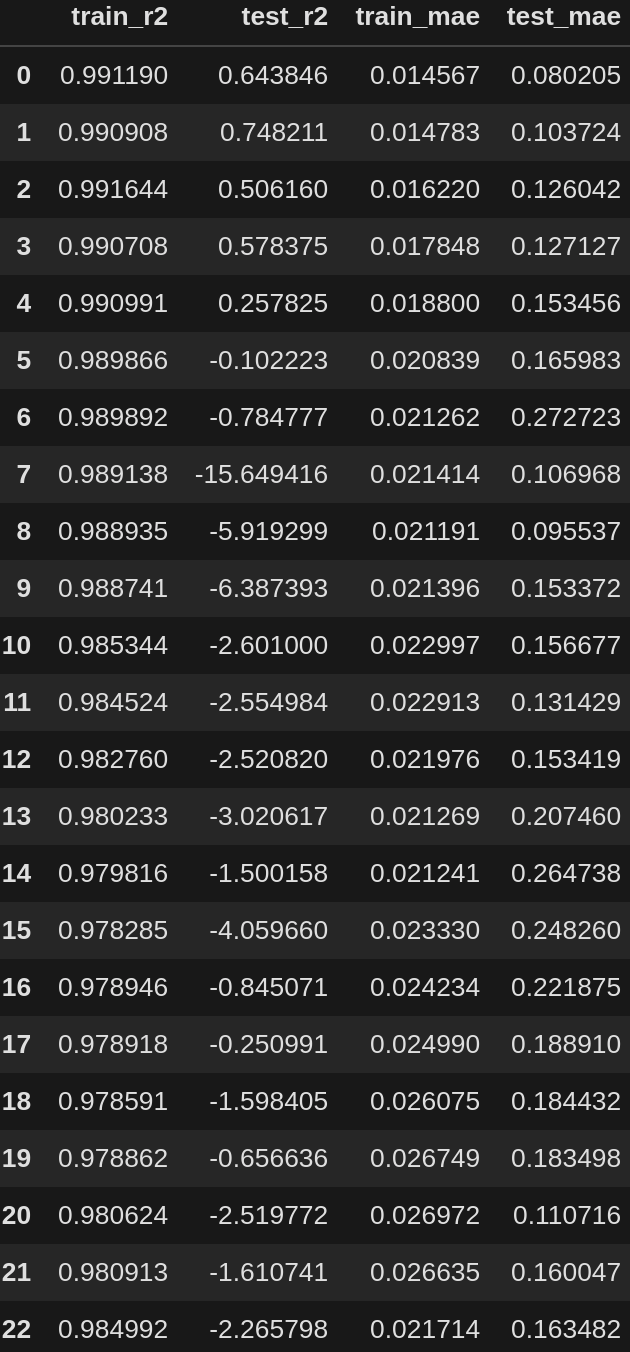
(2) Train Dataset : 전력거래소 데이터

(3) Test Dataset : 전력거래소 데이터

(4) 검증결과 (Walk Forward Validation)

Walk Forward Validation을 이용하여 2017-07부터 2019-07까지 2주 정도를 1Fold로 하여 10구간(20주)을 training하고2 주를 맞추는 방식으로 검증하였다.실제로 현재는 Walk Forward를 사용하였는데 잘 맞추는 구간과 못 맞추는 구간이 따로 존재하였다. 이는 Nested Cross Validation을 통해 적절한 fold를 찾고 구간을 재설정하여 구간별로 학습을 시킨다면 더 최적의 모델을 만들수 있을 것이라 기대한다.

<Figure 15. Test구간에서 y값과 예측값>



결과를 보면 구간별로 잘 맞추는 구간, 잘 못맞추는 구간이 있다. 또한 Train set 16000개에 대해 Train R square값이 대부분이 0.98을 내고있는것을 보면 데이터 사이에 어느정도 규칙성이 있고 모델이 배울 수 있음을 의미한다. Walk-Forward Optimization은 Train Fold의 길이가 모델의 성능에 영향을 미치는 Hyperparameter이다. 남은 기간동안 Walk-Forward Optimization의 적합한 Train Fold길이를 탐색하여, Nested Cross Validation에서 나타나는 문제인 시간이 지남에 따라 모델의 성능이 떨어지는 문제를 해결하고자 한다.

**마. 최종(2차) 제출 예측 모델**

(1) 1st-Level Model 로써 기상 예보를 Input으로 받아 발전량을 예측하는 Forecasting Model (1차 제출)

(2) 1st-Level Model 로써 연속적인 위성사진을 Input으로 받아 발전량을 예측하는 LSTM Model(남은 기간중 모델 구축)

(3) 2nd-Level Model로써 1st- Level Model의 예측값과 Sample을 Input으로 받아 최종 발전량을 출력하는 Fully Connected Neural Network (Stacking)

(4) Train Set을 학습하고, Test Set에 대한 Novelty Detection을 수행하여 개별 예측에 대한 정확도를 알려주는 Variational Autoencoder Model

**4. Non-Stationary Distribution Issue**

**가. Smaller Dataset**

시간에 따라 Distribution이 변하는 Forecasting문제에서는 앞서 Nested Cross Validation결과로 보았듯이, **작은 데이터셋을 사용하는것이 더 유리한 경우가 있다**. 예측 모델은 주어진 데이터를 설명할 수 있을 만큼 충분히 복잡해야 한다. 또한, 시간에 따라 Distribution이 변함과 동시에 x->y의 관계도 변할 수 있다. 이 경우에 모델은 같은 x에서 서로 다른 두 가지 y를 보게 된다. 둘 중에 하나는 반드시 틀릴 수 밖에 없고, 이는 학습을 방해하는 요인이 된다.

결국 모델은 아래와 같은 상반된 목표들을 달성해야만 한다.

**- 최신의 Distribution, x->y 관계를 학습해야 함**

**- 오래된 Distribution, 현재에 유효하지 않은 x->y 관계를 잊어버려야 함**

**- Train Set의 길이는 모델이 Generalized되어 Test Set을 설명할만큼 충분히 길어야 함**

**5. Continual Learning을 통한 모델의 산업 적용**

**가. Conflict Objectives**

(1) 모델은 **최신의 Distribution, x->y 관계를 학습해야 함 (Continual Learning)**

(2) 오래된 Distribution, **현재에 유효하지 않은 x->y 관계를 잊어버려야 함 (Forgetting)**

(3) Train Set의 길이는 모델이 Generalized되어 Test Set을 설명할만큼 충분히 길어야 함 (Hyperparameter Search)

**나. 풍력 발전량 예측에서 요구되는 것**

풍력 발전량 예측 문제에서 요구되는 위와 같은 상충되는 목표들은 일반적인 머신러닝 기법으로 해결하기 어렵다. 앞서 확인하였듯이 Nested Cross Validation은 모델의 성능이 지속적으로 하락하는 문제가 있기에, Walk-Forward Optimization과 Continual Learning, Forgetting을 통한 지속적인 학습 및 모델 운영이 필요하다.

**다. 모델의 실제 산업에의 적용**

모델의 온라인 적용은 배포를 포함한 다음과 같은 운영 측면의 요소들을 고려해야 한다.

(1) Operational aspects of AI

- Continual Learning : **지속적인 학습**

- System Integration : **기존 시스템과의 통합**

- Operator Empowering : **시스템 운영의 주체**

(2) 지속적으로 학습하지 않는 AI 시스템의 문제점

**- 예측할 수 없는 환경변화에 능동적으로 대응할 수 없음**

**- 모델의 초기 성능을 유지하는 것이 불가능**

(3) 지속적으로 학습하는 AI 시스템 구현의 어려움

- 지속적인 Model Builder의 개입을 요구

- 태양광 발전량 예측문제에서는 다음과 같은 방법을 활용할 수 있다.

**1) Time window-based policy**

- 일정한 주기마다 모델을 Walk-Forward Optimization으로 업데이트

**2) On-demand policy**

- 누적 nMAE가 일정 Threshold를 넘으면 모델을 다시 트레이닝

1. 한국에너지공단 신재셍에너지센터(<https://www.knrec.or.kr/business/rps_guide.aspx>

   ) [↑](#footnote-ref-2)
2. ‘Machine Learning can boost the value of wind energy’, <https://deepmind.com/blog/machine-learning-can-boost-value-wind-energy/> [↑](#footnote-ref-3)
3. Coefficient of Determination, [https://en.wikipedia.org/wiki/Coefficient\_of\_determination#cite\_ref-13](https://en.wikipedia.org/wiki/Coefficient_of_determination" \l "cite_ref-13) [↑](#footnote-ref-4)
4. Comparision of Training Approaches for Photovoltaic Forecasts by Means of Machine Learning (2018)’, Alberto Dolara et al. [↑](#footnote-ref-5)
5. Forecasting : Principles and Practice (2013)’, Rob J Hyndman et al. [↑](#footnote-ref-6)
6. ‘Learning Confidence for Out-of-Distribution Detection in Neural Networks (2018)’, Terrance DeVries et al. [↑](#footnote-ref-7)
7. ‘Variational Autoencoder based Anomaly Detection using Reconstruction Probability (2015)’, Jinwon An et al. [↑](#footnote-ref-8)
8. I(X;Y) = H(Y) – H(Y|X) = I(Y;X) , Y를 앎으로써 X의 불확실성이 줄어드는 정도 [↑](#footnote-ref-9)