|  |
| --- |
|  |
| 재생에너지 발전량 예측 경진대회 보고서  (분야 : 태양광 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  나. 맑은날, 흐린날 발전량 예측 성능의 차이  다. 흐린날 발전량 예측이 잘 되지 않는 이유  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-1))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-2).

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

- 발전량이 일정하고 안정적인 맑은 날씨에 비해 **예측이 어려운 흐린날의 태양광 발전량 예측을 수행**하여 태양광 발전량 예측의 정확도를 향상시킨다.

**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) CNN LSTM**

- 기상예보의 오차에 종속적인 Time Series Forecasting Model의 한계를 극복하기 위해, 연속적인 위성사진을 Input으로 받아들여 태양광 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, 일사량 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

- 기상청에서 제공하는 예보데이터보다 커버하는 범위가 넓으며, 훨씬 세밀한 구역에 대한 정보를 제공해 기상정보를 제공하는 거리와 발전소 간의 차이로 발생하는 오차를 줄일수 있다

**(3) 한국농어촌공사 태양광 발전량 데이터**

- 영암태양광발전소, 진도태양광발전소

- 공공데이터포털(data.go.kr)에 공개된 데이터

- 전력거래소에서 주어진 데이터 외에 추가로 학습시켜 모델의 성능을 향상시킨다

**(4) 한국남부발전 태양광 발전량 데이터**

- 부산복합, 부산복합자재창고, 부산수처리장, 부산신항, 인천수산정수장, 하동공설운동장, 하동변전소, 하동보건소, 하동정수장, 하동태양광

- 공공데이터포털(data.go.kr)에 공개된 데이터

- 전력거래소에서 주어진 데이터 외에 추가로 학습시켜 모델의 성능을 향상시킨다

**라. 문제 정의**

1. **Time Series Forecasting과 Regression**

a. Regression

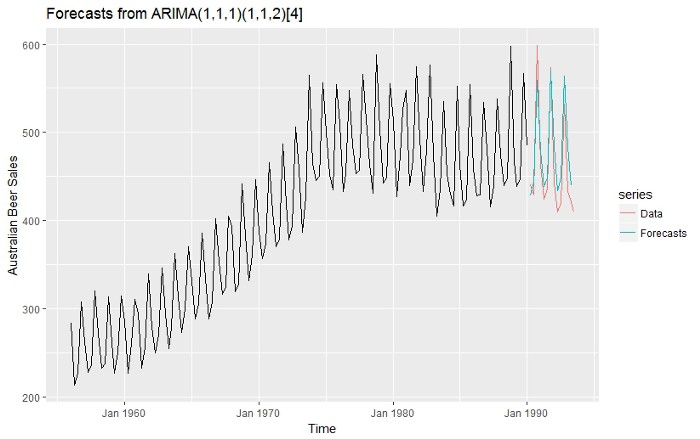
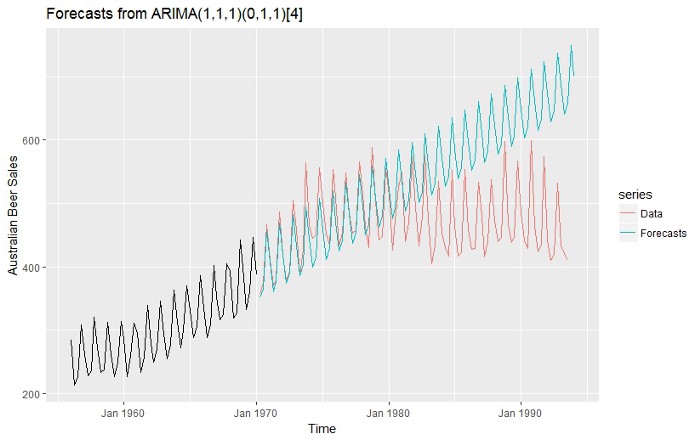
주어진 데이터 집합 에 대해, 종속 변수 와 개의 설명변수 사이의 관계를 모델링하는 문제. 일반적으로 Train Set과 Test Set의 데이터 분포(Distribution)이 같다.

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 태양광 발전량 예측은 Regression이 아닌 Time Series Forecasting 문제이다.

1. **Time Series Forecasting Issues**

a. Non-Stationary Distribution

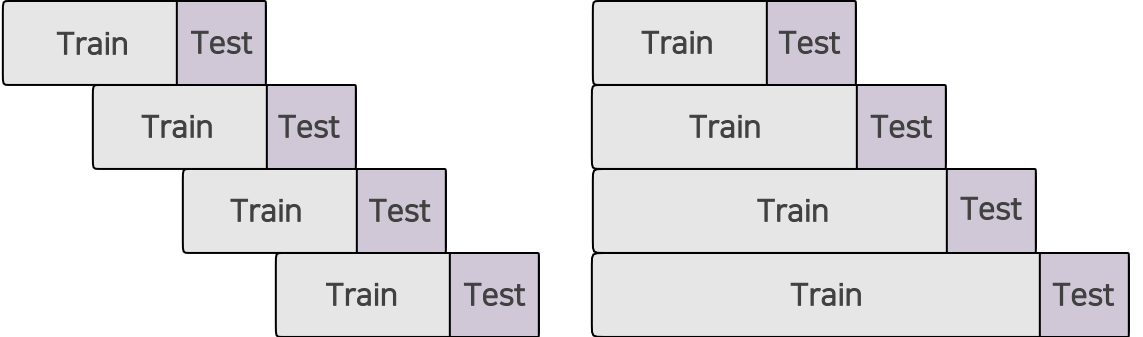


<Figure 1.> Non-Stationary Distribution을 예측할 때 발생하는 문제

일반적으로 Time Series Forecasting에서 Train Set과 Test Set은 다른 Distribution을 가진다. Figure 1.과 같이 Non-Stationary한 Distribution을 가지는 대상은 **Train 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**.

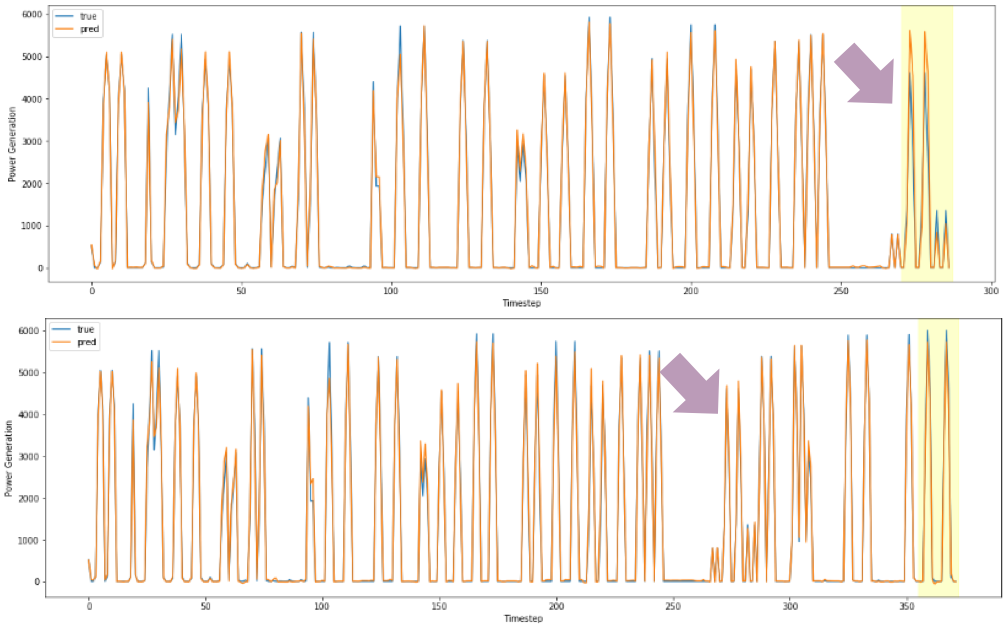
b. Time Series Forecasting Model Validation

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



<Figure 2.> Walk-Forward Optimization, Nested Cross Validation

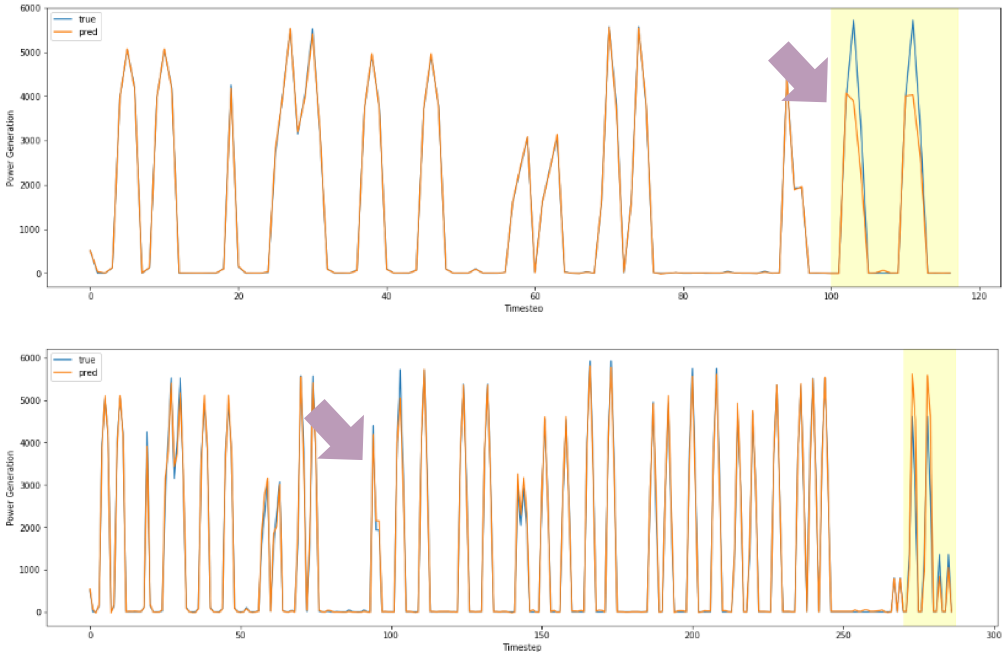
1) Walk-Forward Optimization



<Figure 3.> Walk-Forward Optimization의 예

고정된 길이의 Train Set을 보고, 이어지는 Test Set을 예측하도록 하여 모델의 성능을 검증하는 방법. Time Series Forecasting Model을 학습시키는 방법으로도 활용된다. 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.**

2) Nested Cross Validation



<Figure 4.> Nested-Cross Validation의 예

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.**

c. Evaluation Metrics for Time Series Forecasting

|  |  |  |
| --- | --- | --- |
| nMAE = 0 | R2 = 1 | 모든 1시간단위 예측값 = 모든 1시간단위 실측값 |
| 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 증가함 |

<Figure 5.> nMAE와 R2의 관계

1) R2

Regression문제에서 모델의 성능을 평가하는 지표[[3]](#footnote-3). 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-4). In-Sample Train, Out-of-Sample Test를 가져 Train Set과 Test Set의 평균값이 다를 때 주로 사용한다[[5]](#footnote-5). 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

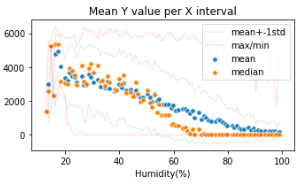
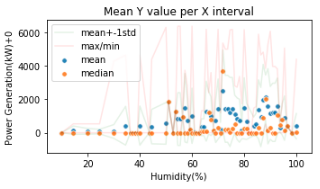
태양광 발전량 예측 문제에서 0 이상의R2는 적어도 52시간 간의 총 발전량을 맞췄음을 의미하고, 낮은 MAE는 총 발전량을 맞췄는지 여부는 알 수 없지만 개별 예측의 에러가 적었음을 의미한다. **총 발전량을 맞추는 것이 중요한지, 개별 예측의 에러가 적은 것이 중요한지에 따라 적합한 Metric을 사용하여 모델을 평가하여야 한다. Baseline Model에서 0이상의 R2가 보장된다면 R2를 쓰는것이 결과 해석에 편리하다.**

d. Prediction Uncertainty

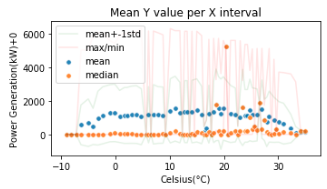
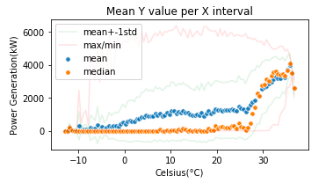
Non-stationary한 Distribution을 다루는 Forecasting 문제의 특성상 Regression보다 모델이 예측을 맞추지 못할 가능성이 높다. 따라서 Test Set이 Out-of-Distribution Sample인지를 확인하여 예측이 틀릴 가능성(Uncertainty)를 제공하는것이 중요하다. Out-of-Distribution Detection[[6]](#footnote-6), Variational Autoencoder를 활용한 Novelty Detection[[7]](#footnote-7)등을 통해 사용자에게 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. EDA**

**가. 기상관측치와 기상예보의 차이**

****

<Figure 6.> 관측치에서 습도-발전량의 관계, 예보에서 습도-발전량의 관계

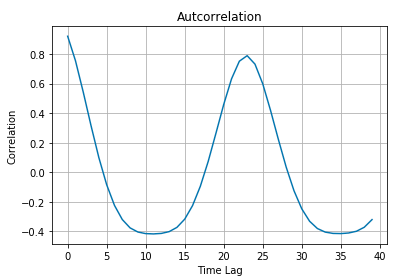
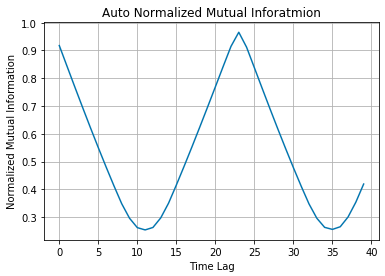


<Figure 7.> 관측치에서 온도-발전량의 관계, 예보에서 온도-발전량의 관계

Correlation, Normalized Mutual Information, Directed Information을 통해 각 변수와 발전량의 관계를 조사하였다. 기상관측치와 태양광발전량과의 관계를 통해 각 변수가 태양광 발전량에 어떻게 영향을 미치는지 인사이트를 얻을 수 있었으나, 기상예보의 불확실성 때문에 기상예보-발전량과의 관계에서도 같은 영향을 미칠 것이라고 기대하기는 어려웠다. Figure 6., Figure 7.과 같이 관측치에서 습도와 온도는 태양광 발전량과 관계가 있는 것처럼 보이지만, 기상 예보상의 습도와 온도는 아무런 관계가 없는 것처럼 보이기 때문이다.

본래 가지고 있는 변수가 10개 안팎이었기 때문에 EDA를 통해 변수를 선택하기 보다는, 관측치에서 관찰할 수 있었던 관계들과 도메인 지식을 통해 Feature Engineering을 수행하였다.

**나. 발전량 데이터의 시계열적 특성**

<Figure 8.> Autocorrelation, Auto-Normalized Mutual Information

태양광 발전량은 시계열적인 특성을 가지는 데이터이므로 Cycle을 확인하기 위해 Autocorrelation, Auto Normalized Mutual Information을 확인하였다. Autocorrleation을 통해서는 Time Lag(x축)에 대한 선형적인 관계를, Auto-Normalized Mutual Information을 통해서는 Time Lag에 대한 비선형적인 관계를 확인하였다. 이미 알려진 24시간의 주기를 Max Peak을 통해 나타났다. 추가적으로, Time Lag이 5시간일때 Correlation이 0이며, 10시간일때 -0.4로 최소인것을 확인할수 있었다. 이 시간 간격을 활용하여 시간과 관련된 Feature Engineering (Windowing, n hours mean)을 수행하였다.

**3. Modeling**

**가. Baseline Model**

1. 실험결과

반복적인 실험을 통한 Feature Engineering검증을 위해 Baseline Model로는 Training과Prediction이 빠른 XGBoost를 사용하였다. 아래와 같이 Input, Ouput Data를 사용하였으며 모델의 성능을 객관적으로 평가하기 위해 평가지표는 R2를 사용하였다.

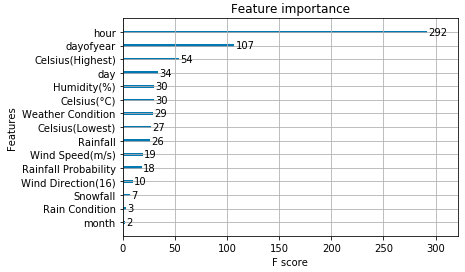
Input : 해당 시간의 기상예보 (Interval : 3시간)

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

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 사용모델 | 사용변수 | Train R2 | Test R2 |
| 기상예보 | XGBoost | 기온, 전운량, 습도, 강수량, 적설량, 풍향, 풍속 | -3.30 | -3.43 |
| 시간 | 월, 일, 시간, 날짜 | 0.72 | 0.31 |
| 기상예보+시간 | 기온, 전운량, 습도, 강수량, 적설량, 풍향, 풍속, 월, 일, 시간, 날짜 | 0.83 | 0.72 |

<Table 1.> Baseline Model Experiment

(2) Baseline Model 선택



<Figure 9.> Baseline Model에서의 Feature Importance

○ 태양과 지표면사이의 거리가 일사량에 가장 크게 영향을 미치는 태양광 발전량의 특성상 시간에 관련된 변수를 넣었을 때와 넣지 않았을 때의 성능 차이가 매우 컸다. 주기를 가진 태양광 발전량을 표현하기 위해 시간 변수를 추가한 모델을 Baseline Model로 활용한다.

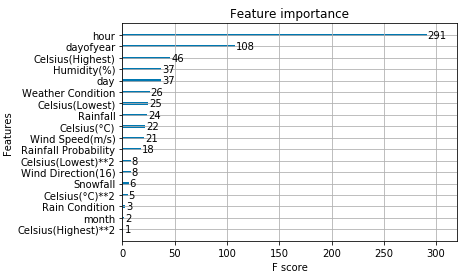
**나. Feature Engineering**

(1) Feature Engineering을 통한 모델의 예측 성능 향상

○ Feature Engineering을 하는 목적은 도메인 지식을 활용해 우리가 예측하고자 하는 값()을 잘 설명하는 변수()를 만들어내는 것이다. 이를 통해 모델의 Optimization 부담을 줄이고, 예측 성능을 높인다. Feature Engineering으로 추가된 변수를 학습한 Model과 Baseline Model을 비교함으로써 각 Feature Engineering의 성능 향상에 대한 기여도를 평가하였다

(2) Feature Engineering Method

○ Polynomial Features



<Figure 10.> Poly\_dim=3일때의 Feature Importance

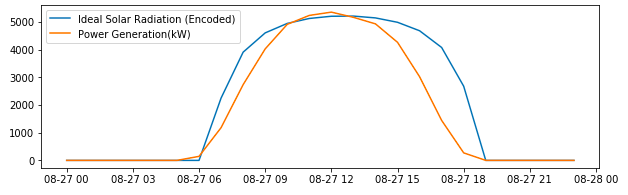
Degree=3 일때 Feature Importance를 확인할 결과 2차항의 기온과 최대기온만 유의미한 중요도를 보이고, 그 외의 변수들의 중요도는 0인것을 확인할 수 있다. Degree 3이상의 Polynomial Feature는 성능 향상에 기여하지 않음을 확인할 수 있었다.

○ Scaling : Tree기반의 모델인 XGBoost 특성상 성능 변화가 없었다. 하지만 Neural Net을 사용할 경우 반드시 필요하다.

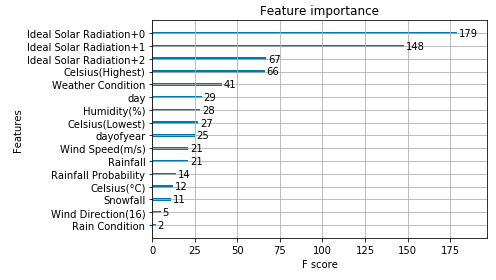
○ Skewness 보정 : 마찬가지로 Tree기반의 모델인 XGBoost에서는 별 효과가 없었다.

○ Polar Coordinates : 0도와 360도가 다르게 표현되어있는 Degree를 활용한 각도 대신, 각 Degree의 cos, sin값을 Feature로 추가하여 구분하였다.

○ 이론적인 일사량



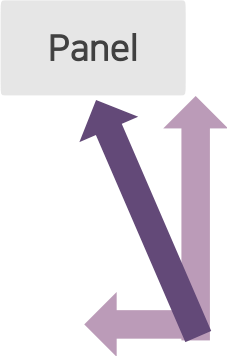
<Figure 11. 이론적인 일사량(하늘색)과 태양광 발전량(주황색)>



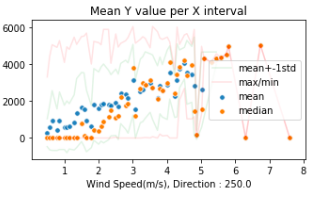
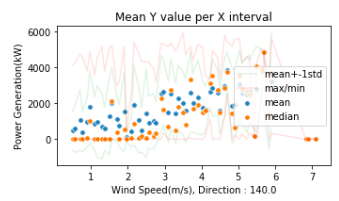
<Figure 12. > 이론적인 일사량의 Feature Importance

위도, 경도, 시간을 통해 이론적인 일사량을 계산해주는 Python Package ‘PySolar’를 통해 이론적인 일사량을 계산하여 변수로 추가하였다. 24시간의 주기를 가지는 태양광 발전량의 특성상 가장 높은 중요도를 보이는 것을 확인할 수 있다.

○ 풍속 벡터 분해



<Figure 13. > 패널을 직접적으로 냉각시키는 남풍 성분으로 풍속 벡터를 분해



<Figure 14. Vector 분해된 풍속과 발전량의 관계>

태양광 패널은 바람에 의해 냉각되고, 냉각이 잘 될수록 발전효율이 올라간다[[8]](#footnote-8). 이에 따라 풍속이 높을수록 태양광 발전량이 증가할 것으로 기대할 수 있다. 이때 패널이 놓여있는 방향과 바람이 불어오는 방향에 따라 같은 풍속에서도 냉각효율이 달라질 수 있다. 이에 태양광 패널이 주로 놓여있는 방향(남쪽) 성분으로 벡터를 분해하여, 해당 성분의 풍속만 변수로 추가하였다.

○ Windowing : Autocorrelation을 통해서 확인하였듯이 이전 Timestep의 기상이 현재에도 영향을 줄 수 있기 때문에, 3시간 전, 6시간전, 9시간전의 기상상태를 Feature로 추가하였다.

(3) 실험결과

|  |  |  |
| --- | --- | --- |
|  | **Train R2** | **Test R2** |
| Baseline Model | 0.8205 | 0.7456 |
| Polynomial (d=2) | 0.8231 | 0.7502 |
| Polynomial (d=3) | 0.8231 | 0.7502 |
| X Scaling | 0.8205 | 0.7455 |
| Y Scaling | 0.8204 | 0.7457 |
| X, Y Scaling | 0.8204 | 0.7257 |
| Skewness 보정 | 0.8205 | 0.7455 |
| Polar Coordinates | 0.8208 | 0.7471 |
| 이론적 일사량 | 0.8712 | 0.7735 |
| 풍향 인코딩 | 0.8215 | 0.7421 |
| Windowing (-3) | 0.8220 | 0.7502 |
| Windowing (-6) | 0.8238 | 0.7464 |
| Windowing (-9) | 0.8239 | 0.7483 |

<Table 2.> Feature Engineering Result

이론적 일사량, Windowing이 XGBoost의 예측 성능을 향상시킨것을 확인할 수 있었다. X, Y Scaling과 Skewness보정은 Tree기반의 모델에서는 동작하지 않지만, 다른 모델에서는 동작하기 때문에 Bayesian Optimization에서 결정할 Feature Engineering으로 포함시켰다.

**다. Data Augmentation (타 발전소 데이터 활용)**

|  |  |  |  |
| --- | --- | --- | --- |
| **Train Set** | **Test Set** | **Train R2** | **Test R2** |
| kpx | kpx | 0.8205 | 0.7456 |
| kpx+타발전소 | kpx | 0.7992 | 0.7585 |
| kpx+타발전소 | kpx 14시예보 | 0.7992 | 0.7603 |

<Table 3.> Data Augmentation

빠른 실험을 위해 타 발전소 발전량을 kpx데이터의 발전량과 동일한 스케일로 스케일링하고, 카테고리 변수 Location을 추가하여 학습시켰다. 다른 조건과 설비를 가진 발전소의 데이터를 Condition(Location)을 통해 구분하며 학습하여Test Score가 향상되었다.

모델의 Complexity가 제한되어 있음에 따라 Train Score는 떨어졌지만, 모델이 잘 Generalize되어 Unseen Data에 대한 예측 성능이 올라갔음이 Test Score의 향상으로 나타났다.

Test Set을 전체 예보가 아닌 1차 제출 과제인 14시 예보만으로 실험해보았는데, Test Score가 향상됨을 확인할 수 있었다.

간단한 데이터 병합을 통해 타 발전소 데이터가 기존 모델의 성능을 향상시킨다는 것을 확인하였다. 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) 1st-level Model : Bayesian Optimization으로 모델별 최적의 Feature Engineering 방법, Hyperparameter 탐색

|  |  |  |
| --- | --- | --- |
| Model | Hyperparameter | Feature Engineering |
| XGBoost | Colsample\_bytree=0.5698  Max\_depth=7  Learning\_rate=0.1  Subsample=0.8  Gamma=0 | Windowing(+3)  Windowing(-3 |
| CatBoost | Border\_count=163  Depth=10  Learning\_rate=0.03 | Polar Coordinates  Windowing (-3)  Windowing (+3) |
| LightGBM | Num\_leaves=132  Learning\_rate=0.034  Subsample\_for\_bin=30  Min\_child\_samples=438  Reg\_alpha=0.050  Reg\_lambda=0.1166  Colsample\_bytree=0.8211 | Polar Coordinates  Windowing (-3)  Windowing (+3) |

<Table 4.> Bayesian Optimzation으로 최적화한 1st-Level Models

(2) 2nd-level Model : Fully Connected Neural Network로 Stacking , 최종 예측결과를 출력한다.

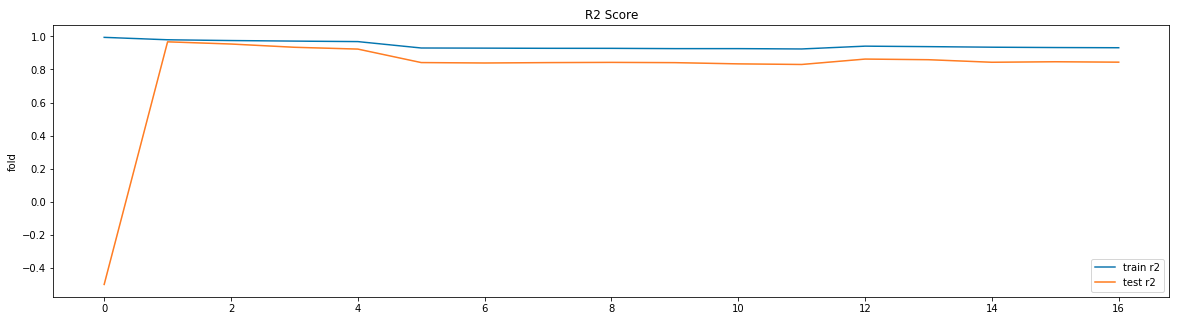
(3) Train Dataset : 전력거래소, 한국농어촌공사, 한국남부발전 데이터

(4) Test Dataset : 전력거래소 데이터

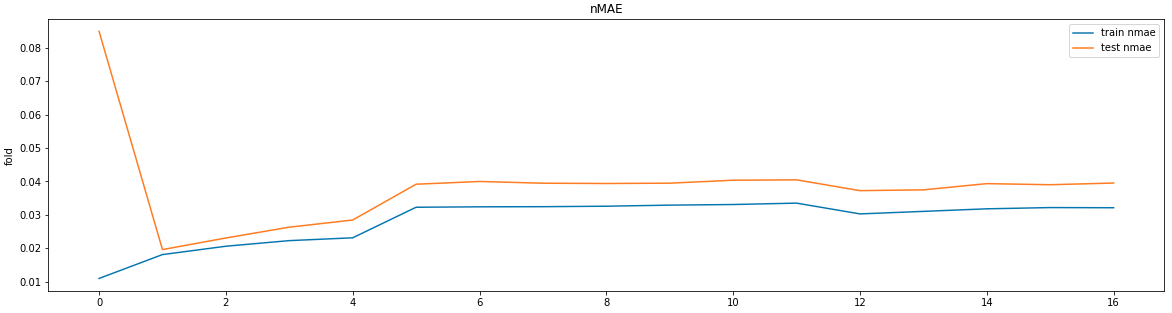
(5) 사용변수 : 기상예보에서 제공되는 변수, 앞서 제시된 Feature Engineering으로 만들어진 파생변수, 시간과 관련된 변수(년, 월, 일)

(6) 검증결과 (Nested Cross Validation)

Nested Cross Validation을 이용하여 2017-07부터 2019-07까지 1개월을 1Fold로 하여 검증하였다.



<Figure 15. Train R2 Score와 Test R2 Score>



<Figure 15. Train nMAE와 Test nMAE>

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Train Fold** | **Test Fold** | **Train R2** | **Test R2** | **Train nMAE** | **Test nMAE** |
| 2017.1~2017.6 | 2017.7 | 0.9945 | -0.5016 | 0.0109 | 0.0848 |
| 2017.1~2017.7 | 2017.8 | 0.979901 | 0.967460 | 0.018101 | 0.019596 |
| 2017.1~2017.8 | 2017.9 | 0.975175 | 0.954151 | 0.020599 | 0.023064 |
| 2017.1~2017.9 | 2017.10 | 0.971863 | 0.934466 | 0.022275 | 0.026301 |
| 2017.1~2017.10 | 2017.11 | 0.968852 | 0.923557 | 0.023122 | 0.028439 |
| 2017.1~2017.11 | 2017.12 |  |  |  |  |
| 2017.1~2017.12 | 2018.1 | 0.963709 | 0.912754 | 0.024057 | 0.029932 |
| 2017.1~2018.1 | 2018.2 | 0.959084 | 0.903958 | 0.024898 | 0.031523 |
| 2017.1~2018.2 | 2018.3 | 0.956089 | 0.890277 | 0.025742 | 0.034004 |
| 2017.1~2018.3 | 2018.4 | 0.953730 | 0.878335 | 0.026866 | 0.035094 |
| 2017.1~2018.4 | 2018.5 | 0.951828 | 0.865854 | 0.027712 | 0.036699 |
| 2017.1~2018.5 | 2018.6 | 0.946290 | 0.862264 | 0.028981 | 0.037062 |
| 2017.1~2018.6 | 2018.7 | 0.943455 | 0.870820 | 0.029675 | 0.036335 |
| 2017.1~2018.7 | 2018.8 | 0.941295 | 0.863015 | 0.030269 | 0.037228 |
| 2017.1~2018.8 | 2018.9 | 0.938277 | 0.859216 | 0.031025 | 0.037463 |
| 2017.1~2018.9 | 2018.10 | 0.934971 | 0.843686 | 0.031791 | 0.039325 |
| 2017.1~2018.10 | 2018.11 | 0.932850 | 0.846831 | 0.032161 | 0.038996 |
| 2017.1~2018.11 | 2018.12 | 0.931605 | 0.844233 | 0.032110 | 0.039501 |
| 2017.1~2018.12 | 2019.1 | 0.926213 | 0.833989 | 0.033078 | 0.040347 |
| 2017.1~2019.1 | 2019.2 | 0.924230 | 0.830298 | 0.033492 | 0.040460 |
| 2017.1~2019.2 | 2019.3 | 0.941295 | 0.863015 | 0.030269 | 0.037228 |
| 2017.1~2019.3 | 2019.4 | 0.938277 | 0.859216 | 0.031025 | 0.037463 |
| 2017.1~2019.4 | 2019.5 | 0.934971 | 0.843686 | 0.031791 | 0.039325 |
| 2017.1~2019.5 | 2019.6 | 0.932850 | 0.846831 | 0.032161 | 0.038996 |
| 2017.1~2019.6 | 2019.7 | 0.931605 | 0.844233 | 0.032110 | 0.039501 |

<Table 5.> Nested Cross Validation으로 검증한 태양광 발전량 예측 모델

2017년 12월은 기상예보 결측치 때문에 검증할 수 없었다.

Train Set으로 예측 시점 이전의 모든 데이터를 사용함으로써, Train Set에 길어짐에 따라 Train Score와 Test Score가 지속적으로 떨어지는 것을 확인할 수 있다. **과거의 Distribution과 현재의 Distribution을 모두 학습할 수 없는 Model의 제한된 Complexity때문에 Train Score가 계속 떨어지고, 앞쪽의 데이터가 새로운 Distribution을 학습하는 것을 방해하기 때문에 Test Score또한 계속 떨어지는 것을 확인할 수 있다**.

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.

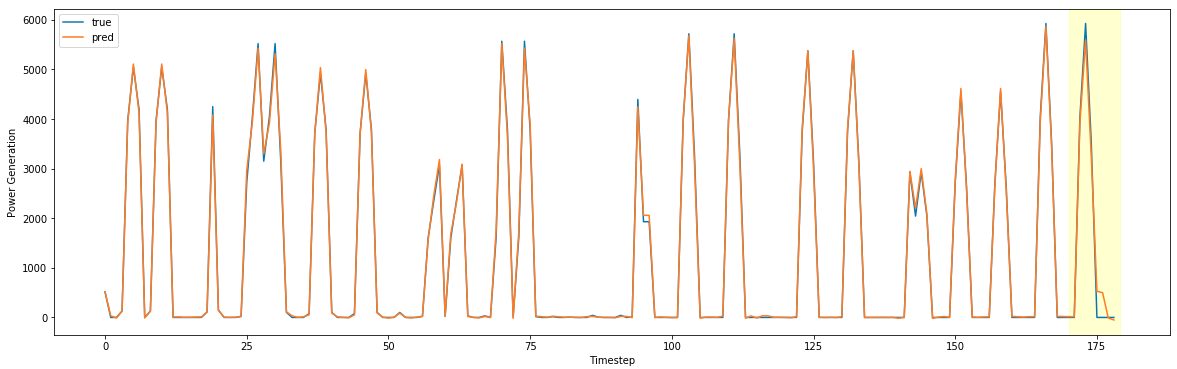
결국 모델은 아래와 같은 상반된 목표들을 달성해야만 한다.

**- 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**

**나. 맑은 날 발전량 예측, 흐린 날 발전량 예측**

****

<Figure 16. 맑은날 발전량 예측>

Figure 16. 은 138시간의 발전량을 학습시켜 52시간을 예측한(연한 노란색 구간) 결과이다. **Train R2 0.99, Test R2 0.98로 그리 많지 않은 데이터로도 예측을 완벽하게 수행해내는 것을 알 수 있다. 뚜렷한 주기를 가지고 있고, Distribution이 Stationary한 맑은 날의 발전량을 예측하는 것은 그리 어렵지 않다.**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  | Train r2 | Test r2 | Train nMAE | Test nMAE |
| 맑은날 | Nested CV | 0.9915 | 0.8833 | 0.0118 | 0.0476 |
| WF Opt. (4) | 0.9987 | 0.8680 | 0.0137 | 0.0514 |
| WF Opt. (9) | 0.9912 | 0.8874 | 0.0118 | 0.0467 |
| 흐린날 | Nested CV | 0.9478 | 0.2875 | 0.0149 | 0.0522 |
| WF Opt. (4) | 0.9301 | -0.091 | 0.0159 | 0.0622 |
| WF Opt. (9) | 0.9447 | 0.2300 | 0.0148 | 0.0542 |

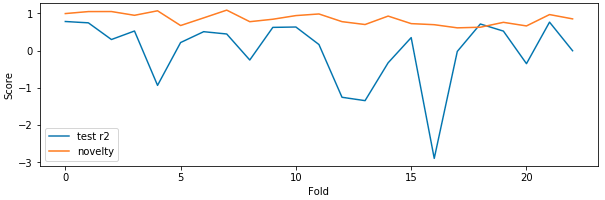
<Table 6.> 맑은날과 흐린날 예측의 성능 차이

여기서 맑은날이란 기상청 예보데이터에서 하늘상태=1, 또는 ‘맑음’이라고 발표된 예보를 지칭한다. 흐린날은 기상청 예보데이터에서 하늘상태=4, 또는 ‘비’라고 발표된 예보를 의미한다.

Nested CV, Walk-Forward Optimization (Train Fold의 개수가 4개, 9개) 총 3가지 검증방법에 대해서 Test Score가 맑은날에 비해 훨씬 떨어지는 것을 확인하였다. 예측이 쉬운 맑은 날의 태양광 발전량에 비해 흐린날의 태양광 발전량이 잘 예측되지 않는 이유는 다음과 같다.

**다. 흐린날 발전량 예측이 잘 안되는 이유**

(1) Novelty (Unseen Pattern)

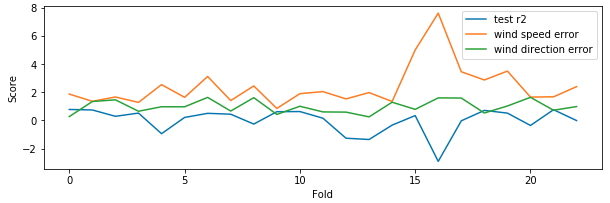
<Figure 17. Fold에 따른 Novelty Score와 Test Score>

**Stationary한 맑은날 기상의 Distribution에 비해, 흐린날의 기상은 훨씬 복잡하고 다양한 패턴을 가지고 있다**. 이는 흐린날의 Walk-Forward Optimization Test Score가 Nested CV에 비해 확연히 작은것으로 설명할 수 있다. Training Fold의 개수가 4개일때 맑은날은 Test Score가 조금밖에 떨어지지 않는 반면에, 흐린날은 크게 떨어진다. Training Fold가 9개가 되어서 Train Set에서 충분한 데이터를 보고 난 뒤에야 흐린날의 WF Opt. Test Score는 Nested CV와 비슷한 수준으로 올라온다. 훨씬 다양한 패턴을 가지고 있기에, 더 많은 데이터를 봐야 학습이 가능한 것이다.

Novelty Score는 1이 Seen, -1이 Unseen으로 정의되어 있다. Novelty Score와 Test Score의 Correlation이 0.3242인 것으로, **Unseen인 Pattern이 많을수록 예측성능이 떨어지는 것을 확인할 수 있다.**

모델을 실제로 운용할 때에는, **Novelty Score를 발전량 예측의 신뢰도로 사용하여 모델의 사용편의성을 높인다.**

(2) 기상예보의 부정확성

****<Figure 18. Fold에 따른 풍속, 풍향의 관측치-예보간 오차와 Test Score>

**모델이 기상예보와 발전량의 관계를 제대로 학습하였으나, Test Set의 Input인 기상예보가 부정확하여 예측에 실패하는 경우이다**. 위의 그래프는 과거의 기상 관측치와 기상예보가 얼마나 틀렸는지를 나타낸 것이다. 풍속과 풍향의 Error가 증가할때, Test Score가 크게 떨어지는 것을 확인할 수 있다. Correlation을 보았을때도 Test Score와 풍속, 풍향의 값이 각각 -0.5336, -0.2129로 Error가 클수록 Test Score가 떨어지는 것을 알 수 있다.

반대로 **Train Set에 틀린 예보가 들어가 모델이 잘못된 관계를 학습하기도 한다. 이런 경우는 Train Set의 관측치-예보간 오차 Score를 조사하여 예측이 크게 틀린 예보를 Train Set에서 배제함으로써 방지할 수 있다.**

미리 과거의 관측치를 볼 수 있는 Train Set과 달리 미래에 Test Set에 들어올 예보는 틀릴지 안 틀릴지 알 수 없기 때문에 위와 같은 방법으로 해결할 수 없다. 이 문제는 **기상예보에 의존하지 않고 위성사진을 활용하는 CNN LSTM모델을 통해 해결할 수 있다.** 남은 기간 동안 CNN LSTM과 1차 제출한 모델을 Stacking으로 함께 활용하여 예측 정확도를 높인다.

**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. 태양광 발전량 예측에서 요구되는 것**

태양광 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-1)
2. ‘Machine Learning can boost the value of wind energy’, <https://deepmind.com/blog/machine-learning-can-boost-value-wind-energy/> [↑](#footnote-ref-2)
3. Coefficient of Determination, <https://en.wikipedia.org/wiki/Coefficient_of_determination#cite_ref-13> [↑](#footnote-ref-3)
4. Comparision of Training Approaches for Photovoltaic Forecasts by Means of Machine Learning (2018)’, Alberto Dolara et al. [↑](#footnote-ref-4)
5. Forecasting : Principles and Practice (2013)’, Rob J Hyndman et al. [↑](#footnote-ref-5)
6. ‘Learning Confidence for Out-of-Distribution Detection in Neural Networks (2018)’, Terrance DeVries et al. [↑](#footnote-ref-6)
7. ‘Variational Autoencoder based Anomaly Detection using Reconstruction Probability (2015)’, Jinwon An et al. [↑](#footnote-ref-7)
8. ‘A collection of resource for the photovoltaic educator’, Christian Honsberg et al. [↑](#footnote-ref-8)