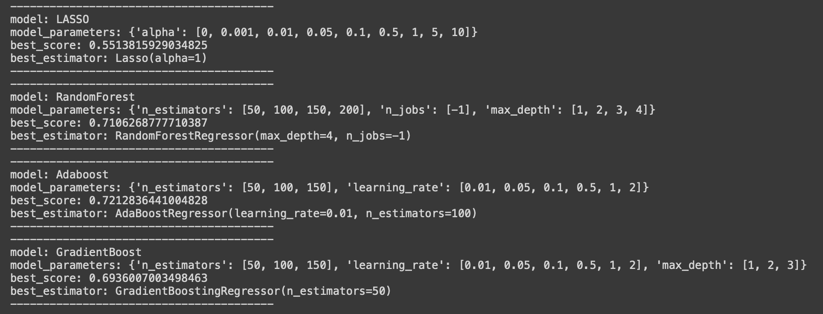
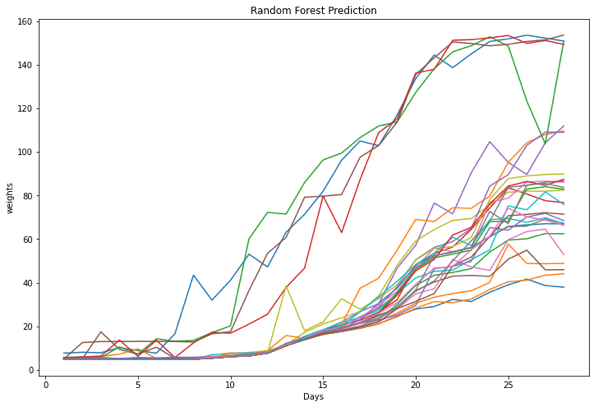
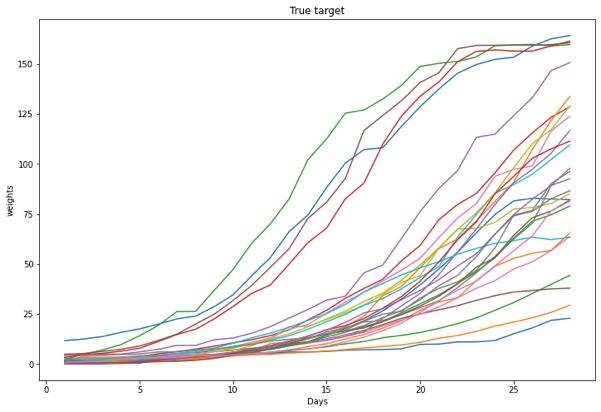
위의 preprocessing을 거친 데이터들을 불러와 traditional model 들에 적용을 시켜보았다. 데이터가 시계열 데이터라 traditional model에 이 데이터를 적용하게 된다면, 시계열 데이터는 하루의 각 시간대별로 순차적으로 모델에 들어가게 되는데 traditional 모델에서는 00시부터 23시까지의 각 15개 feature들로 쭉 펴야 넣을 수 있게된다. 이렇게 함으로써 갖는 의미는 예를들면 00시의 내부온도, 01시의 내부온도, 02시의 내부온도 등의 시간대별 같은 feature들이 있는데, 이들을 독립적으로 보겠다는 것이다. 이는 실제로 순서가 있는 데이터에 대해서 순서를 제거함으로써 더 greedy한 방식으로 예측을 하겠다는 것이다. 여기서 greedy 한 방식이라 표현을 한 것은, 예를 들어 (00시의 데이터)->(01시의 데이터)->(02시의 데이터) ... 라는 큰 흐름을 다 보는 대신에 (00시 데이터, 01시 데이터, 02시 데이터...) 이런 식으로 단편적으로만 보아 더 greedy한 방식이 된다고 하였다.

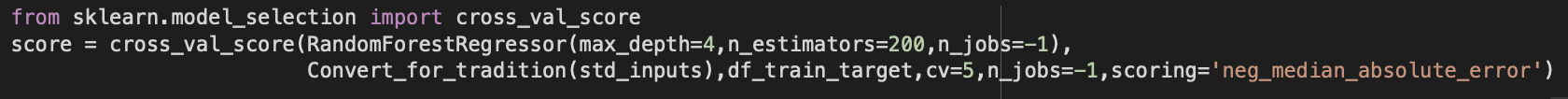
이렇게 처리를 한 데이터는 원 시계열 데이터 shape (700, 24 , 15) 를 (700, 24\*15)로 바꾸어주어 입력을 넣었고, standard scaling을 해서 넣어주었다. 여기서 사용한 모델은 LASSO, AdaBoost,GradientBoost, RandomForest 등이다. 문제가 단순할 것을 대비해 LASSO 같이 단순한 모형부터 복잡한 Boost regressor까지 확인해보았다. grid search를 통해 hyperparameter를 설정하였으며 measurement는 score를 사용하였다.



결과는 AdaBoost가 가장 좋게 나왔으며 RandomForest와 비슷한 수준을 나타내었다. RandomForst 가 interpretablity가 높기 때문에 RandomForest를 기반으로 더 살펴보면



원본 시계열 데이터를 따라가지는 못하지만 어느정도의 경향을 따라가는 것을 볼 수 있다. 본 문제에서 measurement를 MAE로 측정을 하기 때문에 RandomForest에 대해서 MAE 측정을 해 본 결과 8 에 가까운 점수가 나왔으며, 이는 NN 모델을 사용한 예측에서 실험해본 최저 MAE값이었던 4의 2배가 조금 안된다.



여러 traditional 모델을 사용해본 결과, 원본 데이터의 시간적 흐름을 무시하고 각 시간들의 feature를 독립적으로 보는 것에 대하여 한계가 있다고 생각했다. 그리고 traditional model들에서 데이터의 시간적 흐름을 무시함으로써 데이터 자체의 정보가 손실되어야 하는 상태에서, 데이터의 개수 또한 너무 적어 단순히 traditional model의 complexity를 늘리는 것은 overfitting의 위험이 특별히 더 클것으로 예상이 되고, 그로인해 sequantial information을 적용할 수 있어 데이터를 최대한 활용할 수 있는 Neural Network 가 더 적합하다고 생각하였다.