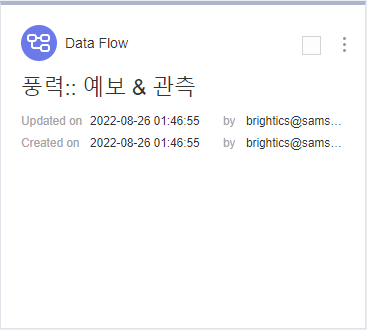
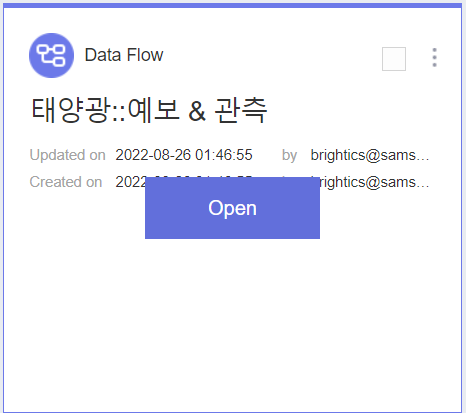
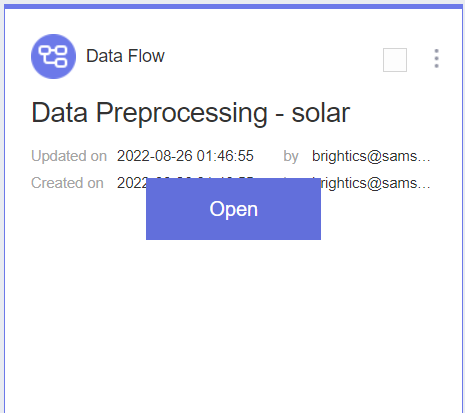
**경진대회 코드 설명자료**

**팀 명 : AIVLE**

1. **브라이틱스 프로젝트 내 모델 설명**

* 프로젝트 내에는 3가지 모델로 구성되어 있다.
* Data Preprocessing – solar : 14시 예보데이터를 활용한 일조, 일사, 전운량 예측
* 태양광 :: 예보 & 관측 : 태양광의 발전량을 예측하는 모델
* 풍력 :: 예보 & 관측 : 풍력의 발전량을 예측하는 모델

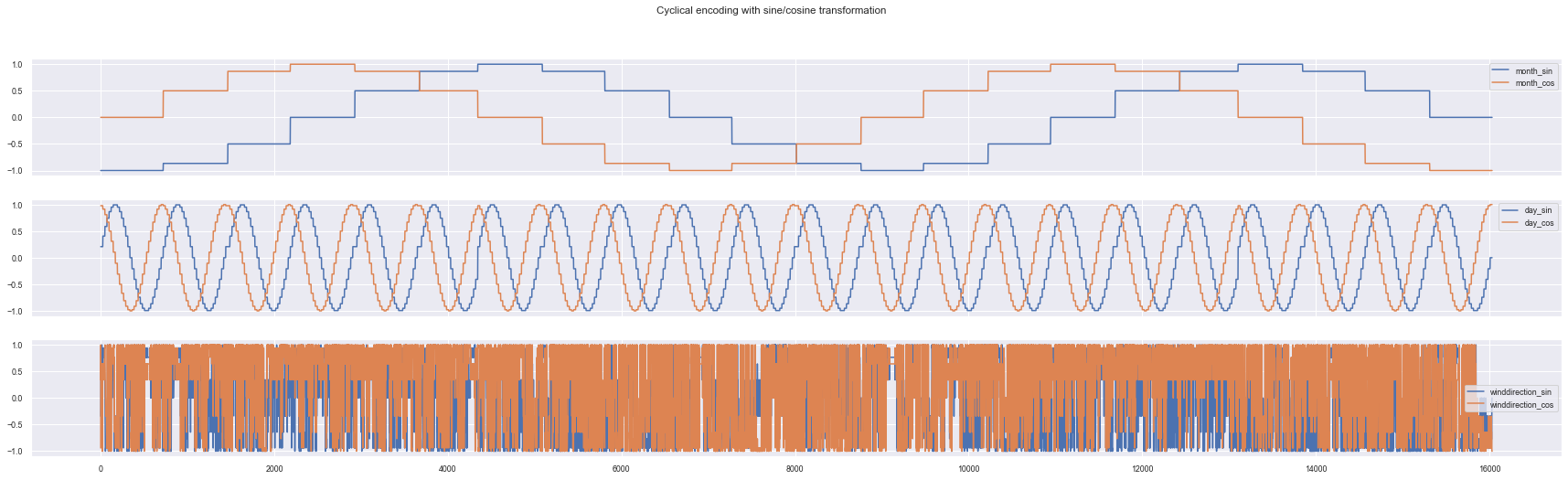


1. **Input 파일 (활용 데이터) 설명**

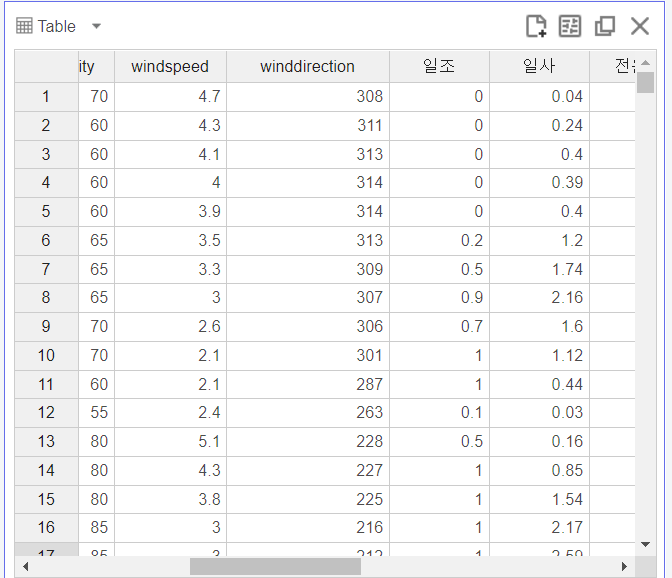
* solar\_power\_2204 : 4월까지의 태양광 발전량 데이터, 주최측에서 제공
* solar\_forecast\_final : 주최측에서 제공한 기온, 습도, 풍량, 풍향 예보데이터 중, 14시 예보데이터 중 2022년 3월 1일부터 2022년 6월 30일까지만 뽑은 데이터
* solar\_predict\_new : Data Preprocessing - solar에서 14시 예보데이터로 예측한 2022년 3월 1일부터 2022년 6월 30일까지 일사, 일조, 전운량 데이터
* solar\_weather\_actual\_810 : 기상청 API를 이용해 크롤링한 실제 일사, 일조, 전운량 데이터를 주최측에서 제공한 실제 관측 데이터와 병합한 데이터
* wind\_2204 : 4월까지의 태양광 발전량 데이터, 주최측에서 제공
* wind\_forecast\_weather : 주최측에서 제공한 기온, 습도, 풍량, 풍향 예보데이터
* weather\_wind\_actual : 주최측에서 제공한 실제 관측 데이터

1. **탐색적 자료 분석 (Exploratory Data Analysis)**

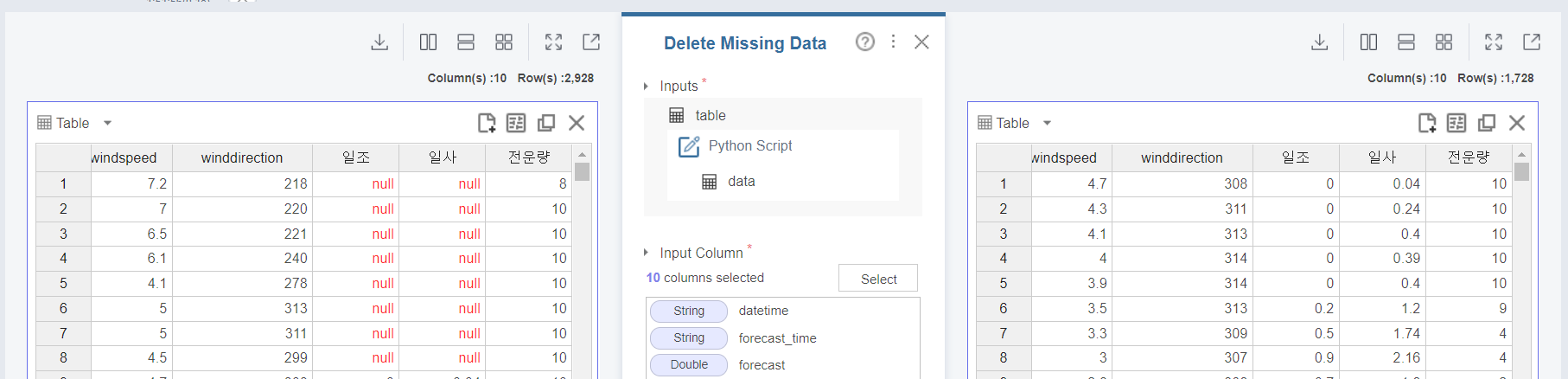
* 시간 데이터의 경우 더미 변수로 취급하지 않고, 명확한 주기성과 연속성이 존재하도록 하기 위해서 sin변환과 cos변환을 진행
* sin변환과 cos변환을 한 month, day, hour를 새로운 변수로 넣어줌



* 변환 시 위와 같은 결과를 얻을 수 있음. 시간은 이제 더미변수로 끊어져 있지 않고 연속적인 값을 얻게 됨
* 풍향 데이터는 아래와 같이 구성되어 있음

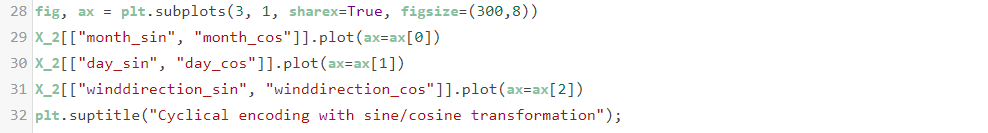


* 풍향은 보통 각도에 따라 더미변수로 묶어주지만(북서풍, 북풍, 서풍 등) 바람의 방향 또한 연속적인 변수라고 판단하여 sin변환과 cos변환을 통해 연속성을 지켜주려고 함
* 그러므로, 최종적으로 sin, cos변환을 통해 연속성을 얻은 변수는 month, day, hour, winddirection임.
* 모든 분석과정에서 위와 같은 sin,cos변환을 전부 시행하였음.
* 데이터에 결측치가 존재한다면 행 기준으로 데이터를 모두 제거함. 이 때 brightics studio안의 Delete Missing Data모듈을 사용해 결측치를 모두 제거함. 모든 분석 과정에서 결측치에 해당하는 부분은 모두 삭제함.

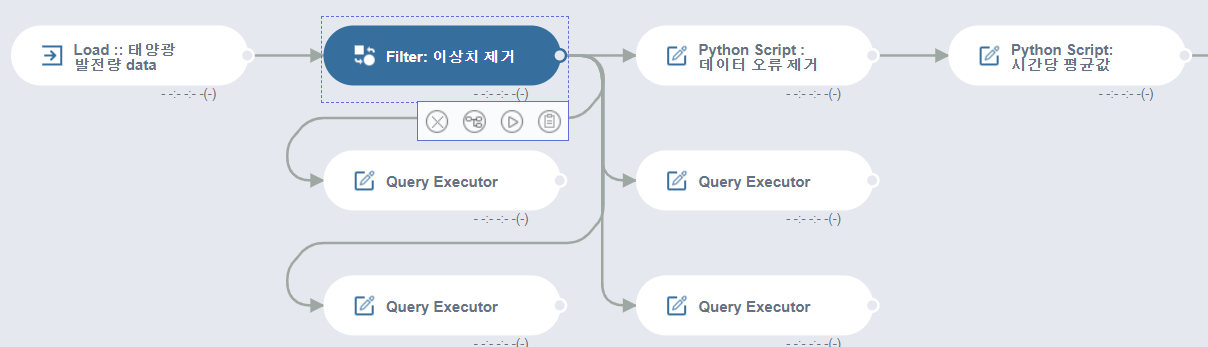


* 아래는 sin,cos변환 코드임

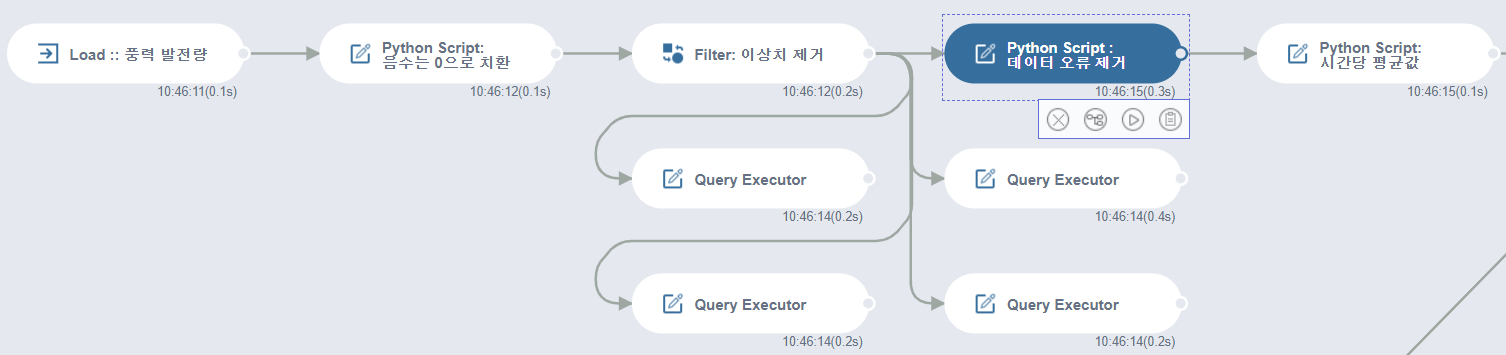




* 태양광 발전량 데이터에 한정해서, 이상치와 데이터의 오류를 제거함.(베이스라인을 따름) 또한, 10분 주기 데이터를 1시간 주기 데이터로 변경시킴

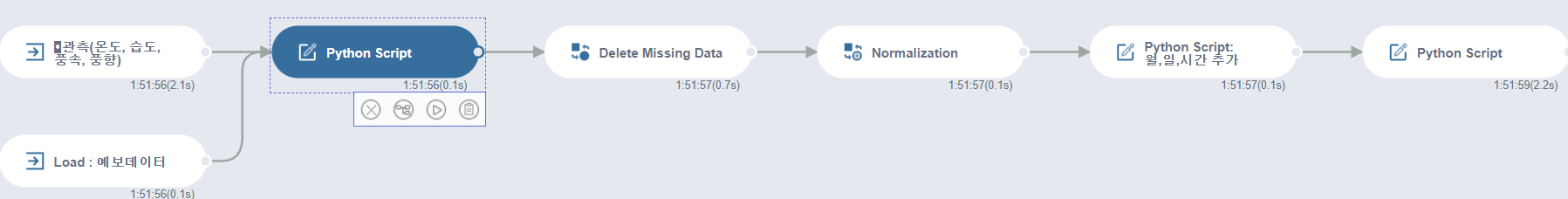


* 풍력 발전량의 경우 이상치와 데이터의 오류를 제거하고(베이스라인을 따름.)음수로 나오는 값이 있어, 0으로 치환시킴. 이 또한 마찬가지로 10분 주기 데이터를 1시간 주기 데이터로 변경시킴

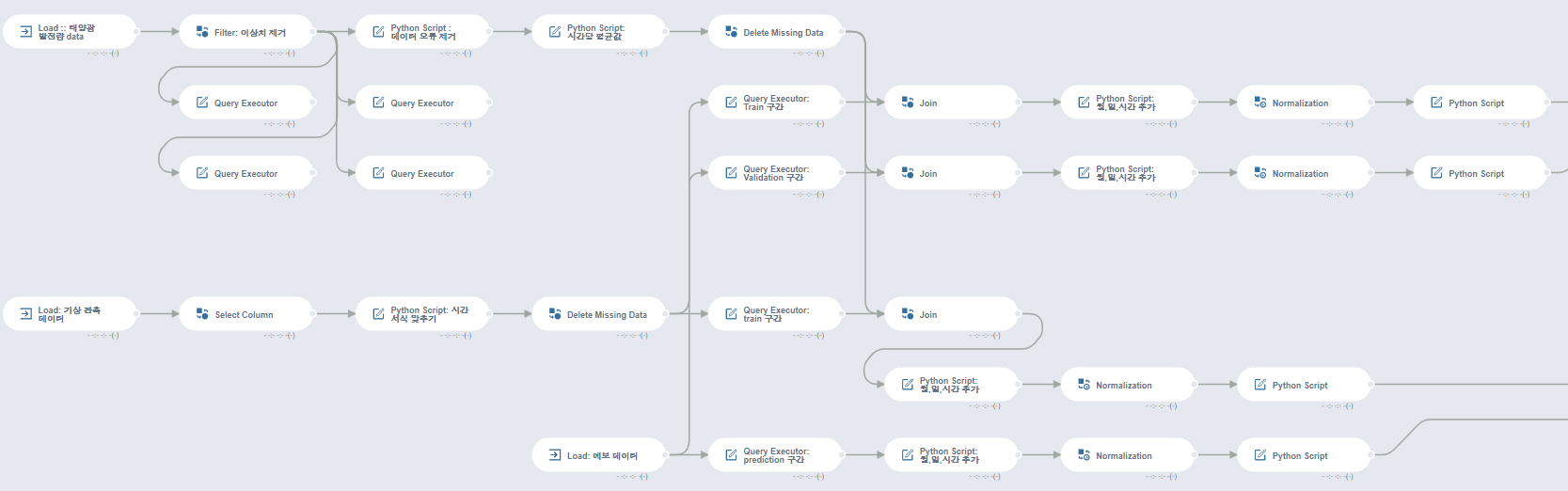


1. **변수 선택 및 모델 구축 (Feature Engineering & Initial Modeling)**

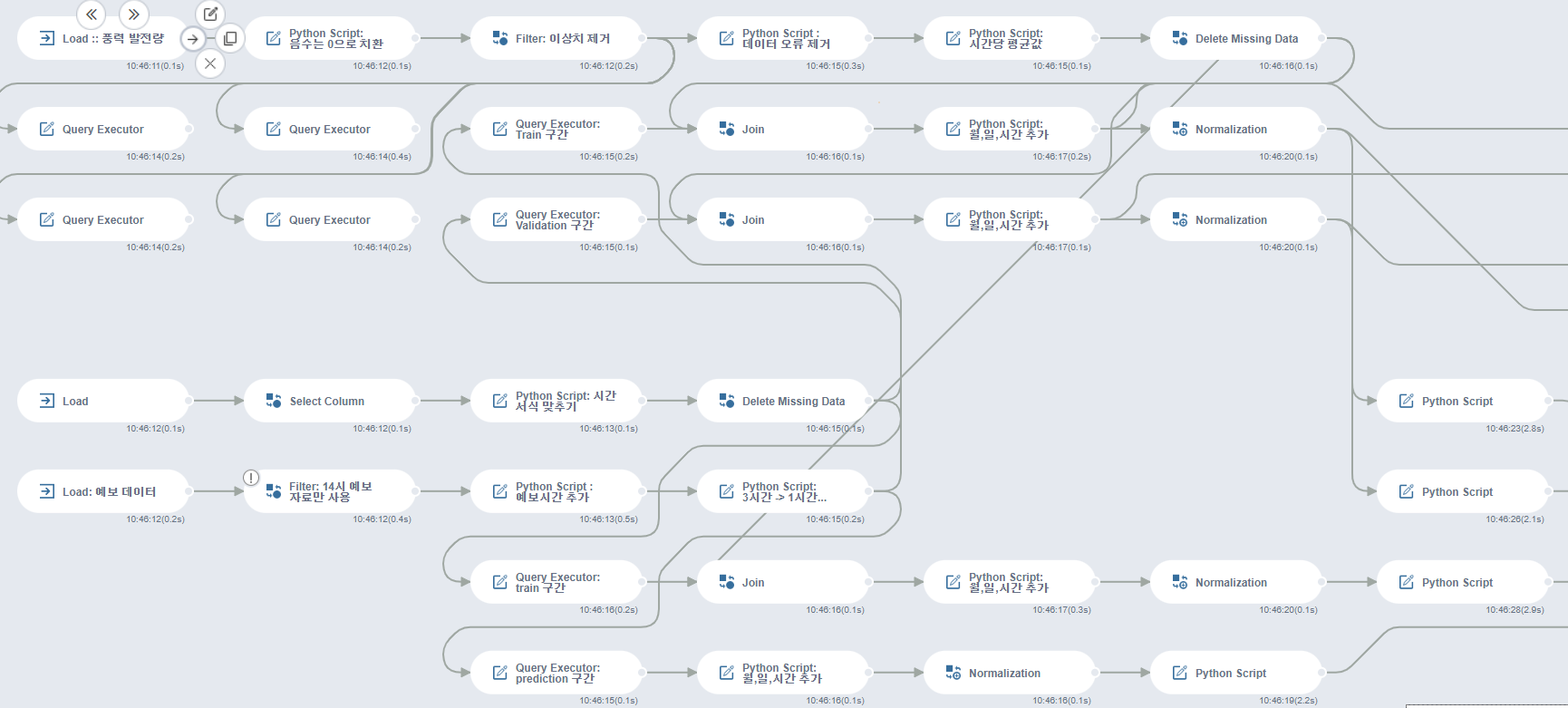
* Data Preprocessing



* 결과 값을 비교하기 위해서 사용될 실제 데이터와 일조, 일사, 전운량을 예측할 예보데이터를 사용
* Python Script를 보면, 실제 데이터에서는 모델 학습시 target으로 사용할 일조, 일사, 전운량 데이터를 가져오고, 모델의 features로 들어갈 데이터로는 오직 예보데이터를 사용
* 과거의 일조, 일사, 전운량으로 모델을 학습시키고, 이를 통해 평가를 거치는 구조로 모델은 구성되어 있음.
* 위에서 설명했던 것과 마찬가지로, 월, 일, 시간, 풍향에 대한 데이터는 sin,cos변환을 사용함.
* Scaler의 경우 Robust Scaler를 사용함.
* 태양광 :: 예보 & 관측



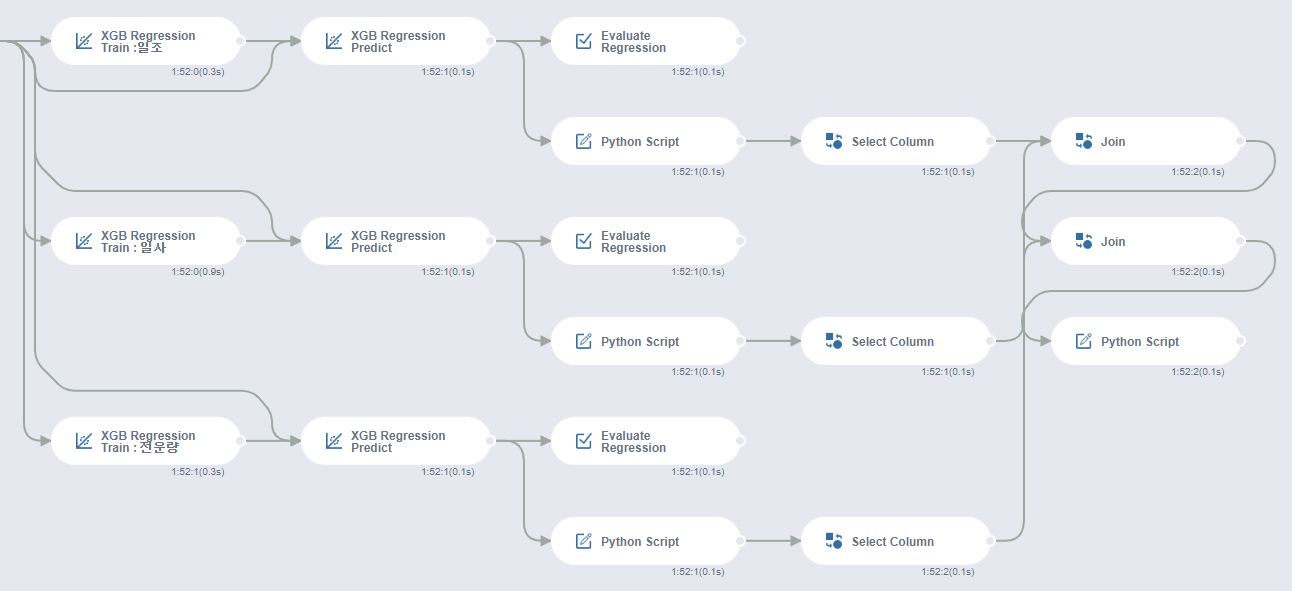
* 위에서 언급했다시피, 발전량 데이터의 경우 이상치, 오류제거 후 평균값으로 데이터를 변형한 후, 결측치를 모두 제거
* 기상 관측 데이터의 경우 결측치를 모두 제거
* 예보 데이터는 Data Preprocessing - solar에서 예측한 일조, 일사, 전운량 데이터와 주최측에서 제공한 14시 예보데이터를 가져옴.
* 이 때, 3개의 데이터 모두 Standard Scaler를 적용하고, sin, cos 변환을 시행함.
* 풍력 :: 예보 & 관측



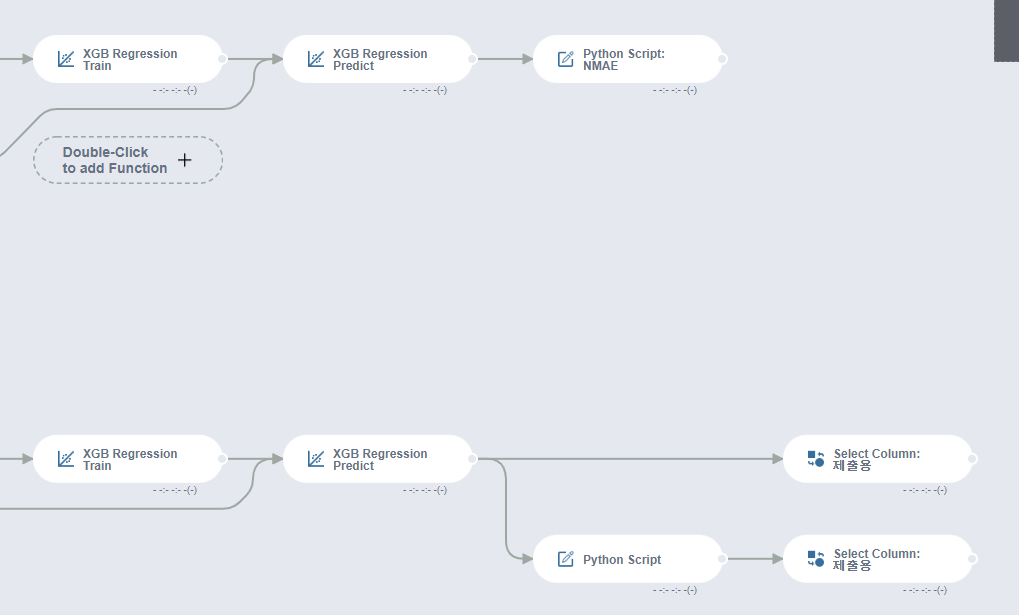
* 위에서 언급했다시피, 발전량 데이터는 음수인 경우 0으로 치환하고, 이상치와 데이터 오류를 제거해 평균값으로 데이터를 변형 후, 결측치를 모두 제거
* 실제 데이터는 결측치를 제거
* 예보 데이터는 14시에 예보한 데이터만 사용하도록 했으며 시간 간격으로 예측된 데이터 이므로, 1시간 단위로 데이터를 변경하고 선형 보간법으로 결측치를 보간
* 3개의 모든 데이터에 대해서 MinMax Scaler를 통해 데이터를 스케일링한 후, sin,cos변환을 통해 hour, month, day, winddirection을 연속형 변수로 변경함.

1. **모델 학습 및 검증 (Model Tuning & Evaluation)**

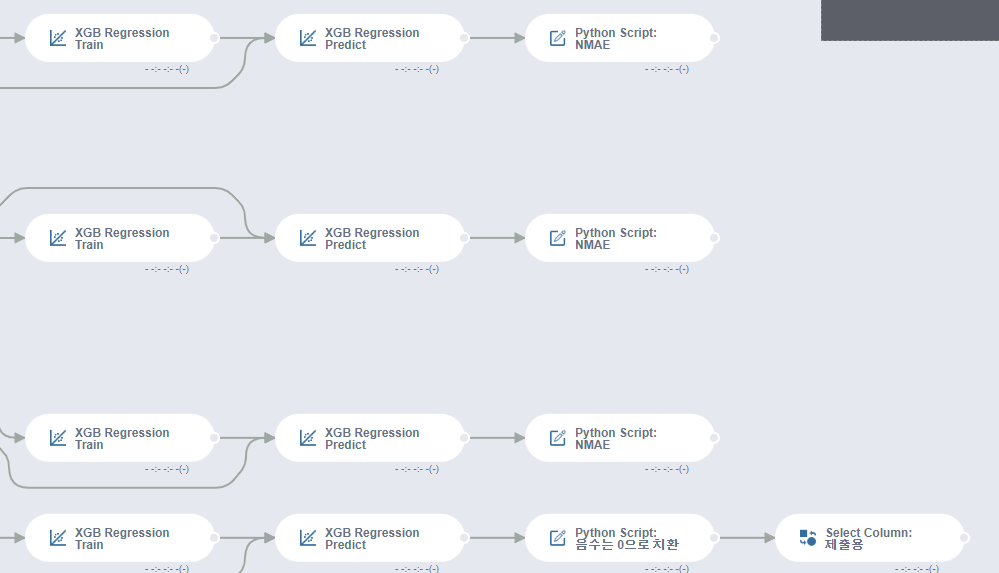
* 모델의 학습은 모두 XGBoost를 이용함.
* Data Preprocessing



* 일조, 일사, 전운량을 예측한 결과, r2 score가 0.633, 0.837, 0.613으로 어느정도 잘 맞추는 것을 알 수 있었음
* 모델을 trainin할 때 사용된 데이터는 14시 예보데이터 였으며, 기온과 습도, 풍속은 모두 로버스트 변환을 시행하였고, 풍속의 경우 sin,cos 변환을 시행
* 예측한 결과를 모두 join하여 하나의 데이터 셋으로 export함
* 태양광 :: 예보 & 관측



* 태양광 예보의 경우 validation score가 8.70611이었음.
* Standard Scaler의 성능이 예상보다 좋지 않아 이를 사용하지 않고, 원래있던 값으로 예측을 진행.
* Month, day, hour, winddirection은 모두 sin, cos변환을 시행.
* 풍력 :: 예보 & 관측



* 풍력의 경우도 MinMax Scaler의 성능이 좋지 않아 사용하지 않았으며, Month, day, hour, winddirection은 모두 sin, cos변환을 시행.
* Validation score가 11.7118이 나오는 것을 확인할 수 있었음
* 예측 결과가 0이 나오면 모두 음수로 치환함.

1. **결과 및 결언 (Conclusion & Discussion)**

* 시간을 sincos으로 변형시키는 것이 성능을 더 좋게 할 수 있었음. 시간을 단지 더미변수로만 생각하지 말고, sincos변환을 통해 연속적으로 나타내는 아이디어가 좋았다고 생각함.
* 파생변수를 더 많이 생성하였어야 했는데 생각나는 파생변수가 많이 없었음. 파생변수에 대해 많은 고민이 필요해보임.