-회귀로 한 이유

=> Regression을 하면 y\_quality가 나올 것.

=> 현재 스마트팩토리 상에서 가장 문제가 되는 상황은, 고객사 및 계열사에게 불량인 제품이 오판별되어 정상으로서 납품되는 상황. 오히려, 정상인 제품을 불량으로 오판정하는 것은 고객 신뢰도를 고려했을 때, 물론 waste가 발생할 수는 있겠지만 정상인 제품을 불량으로 오판정하는 것보다는

=> 따라서, test set에 대한 regression을 한 뒤, threshold를 TRAIN\_SET에서 0으로 판별된 제품의 Y\_QUALITY MAX THRESHOLD를 0과 1을 판별하는 기준점으로 사용하는 것이 아니라, 1로서 판별된 제품의 Y\_Quality Min Threshold를 0과 1을 판별하는 기준점으로 삼는 전략을 취한다.

-왜 그루핑을 했는가?

=> 시각화로 보여주기. 이렇게, ‘공통’으로 존재하는 것으로 보이는 열들이 그룹이 지어진 것을 알 수 있었음.

=> 그 결과 특정 라인을 지나가는 제품에서만 측정되었던 변수들이 존재하였음

=> 라인 내에서도 비결측열(실제로 값을 가지는 FEATURE) 그룹들을 공유하는 행들이 존재함을 확인.

=> 어떠한 feature subset을 사용해야할지 여러 실험을 진행하였음

-기본 베이스라인 + 왜 이번에는 5,6번에 집중했는가?

=> Trainset에서 5,6번 라인에서 생산되는 T 및 O 제품의 비중이 늘어나기도 했으며, 새롭게 추가된 변수들의 경우

-왜 catboost로 했는가?

numeric하게 보여도, 실제로는 value\_counts()를 찍어보았을 때 물론 연속적인 값을 가지는 경우도 있지만, 특정값들을 가지는 사실상 ‘이산형’ 인 경우가 존재. ‘X\_46’가 그 대표적인 예시

이는 결국 값 자체는 numeric이지만 numeric ‘CATEGORY’가 많은 경우이기 떄문에, 범주형데이터를 다루는데 있어 최적화된 catboost를 사용하는 것이 적합하다고 판단하여 본 알고리즘을 사용하였으며, gpu 학습 옵션을 통해 조금 더 빠른 학습 속도 지원

* 결측값 시도를 어떻게 했는가
* Deep Learning의 경우, 결측값을 그냥 무시하는 차원으로 학습이 진행됨
* 결측값 보간을 평균, 최빈, 0 impuation등으로 자체적으로 진행하는 것은 위험이 있을 것으로 예상되었음.
* 특히 0으로 imputation하는 경우 tree-based 알고리즘의 경우 bias를 높일 것으로 예상되었음. Public Score 상으로 0 impuation이 더 좋았으나 overfit우려가 있었으며, repeatedstratifiedkfold 결과 그리 성능은 좋지 않았으므로 overfit 위험에 해당 파일은 최종 submit 고려 대상에서 제외함.
* 5,6번 catboostregression에서는 catboost 알고리즘 자체가 nan-values 들에 대하여 자체적으로 처리를 하는 방식이 있으며, 해당 결측처리를 최소화하기 위해, 5번에서는 특히 비결측열이 적고 비결측열행그룹 내 행이 적은 하나의 그룹씩을 제외한 비결측열행그룹끼리의 교집합을 취하였고, 라인별 특징을 조금이나마 반영하고자 두 교집합 끼리의 합집합으로서 최종 feature subset을 선정하였음.
* 5,6번 라인 간에 비결측열 분포가 비슷하게 관찰되므로 학습 데이터를 최대한 늘리고자 5,6번 라인을 묶어서 진행하였음
* 결측값 BEST
* RepeatedKStratifiedKfold로서 ValidationSet 구축을 통한 metric 확인으로 최대한 일반화된 성능을 내는 모델을 만들고자 하였음