


# 스마트 공장 제품 품질 상태 분류 AI 오프라인 해커톤

팀 주혁이

강민규 김영준 배홍섭 전주혁 최다희

# Contents

- 
-  **01** Intro
  - 02** Feature Engineering
  - 03** Validation
  - 04** Modeling
  - 05** Outro
-

## 대회 배경 및 목표

제조 지능화를 통해  
공정 과정에서 발생하는 이벤트에 신속 대응 및 안전  
성과 효율성 극대화하기 위한 방안 도모



품질 편차를 최소화해 생산 경제성과 안정성을 확보  
할 수 있도록 생산된 제품이 적정 기준을 충족하는지  
판단하고 분류하는 AI 모델 개발

## 제공데이터

### Train.csv :

학습 데이터셋(1132개),  
PRODUCT\_ID, Y\_Class(3개),  
Y\_Quality, LINE, PRODUCT\_CODE, X  
Feature(3326개)

### Test.csv :

테스트 데이터셋(535개),  
PRODUCT\_ID, LINE, PRODUCT\_CODE,  
X Feature(3326개)

## 평가 항목

- 모델 성능
- Feature 상관관계분석
- Feature Selection, 결측치보간
- Validation set 구축 전략
- 모델 적용 가능성

## How to Access Data

**아이디어** 실제 공정은 **Shut Down** 기간이 존재



**분석** PRODUCT CODE와 LINE 피쳐 존재



**아이디어** 같은 PRODUCT CODE라도 측정한 기계가 다르다면?




**검증** 데이터가 나누어지는 특성 발견

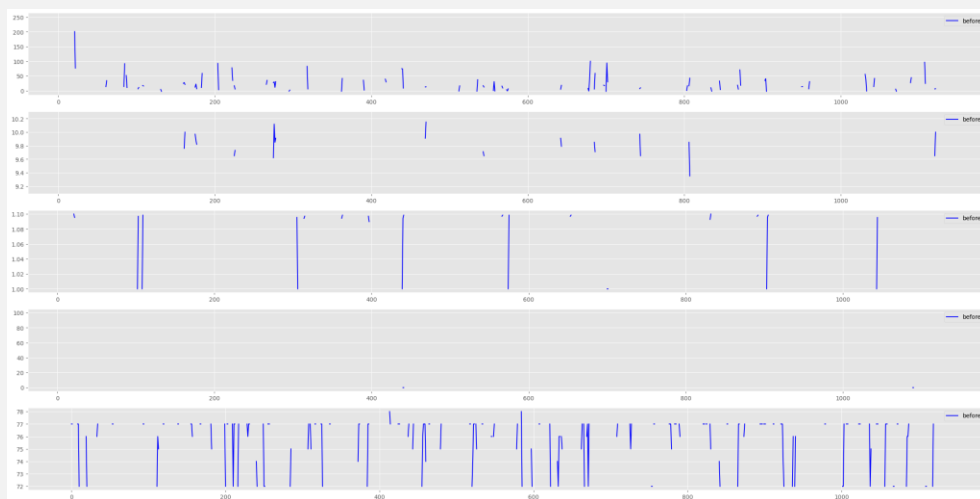
**적용** 다른 측정 기기들을 기반으로 split

	LA	3A	1N	1O	1P	1Q	3B	1L	1T	1U	1V
1	X_241	X_242	X_243	X_244	X_245	X_246	X_247	X_248	X_249	X_250	X_251
2							1	88	0	45	11
3							1	91	1	45	12
4	999	36	36	3	6						
5						1	89	0	45	11	45
6	999	36.3	36.3	3	3						
7						1	93	4	45	11	0
8	999	36	36	3	6						
9						2	95	124	45	11	0
10	999	36	36	3	6						
11						2	88	0	45	11	90
12						2	87	0	45	11	126
13	999	35.8	35.8	3	3						
14						2	88	0	45	11	120
15						1	96	27	45	12	0
16	999	34.4	34.4	3	6						
17	999	36	36	3	3						
18						1	89	0	45	11	91
19	999	34.7	34.7	3	6						
20	999	36	36	3	3						
21						1	92	152	45	11	0
22	999	35.7	35.7	3	3						
23						2	89	149	45	11	0
24	999	35	35	3	6						
25	999	35	35	3	3						
26						1	95	5	45	12	0
27	999	36	36	3	6						
28	999	35	35	3	3						
29						11	90	0	45	11	43
30						1	89	0	45	11	90
31	999	36.4	36.4	3	3						
32						4	93	150	45	12	0
33	999	34.4	34.4	3	3						
34						1	96	95	45	12	0
35						2	91	0	45	11	67

# Contents

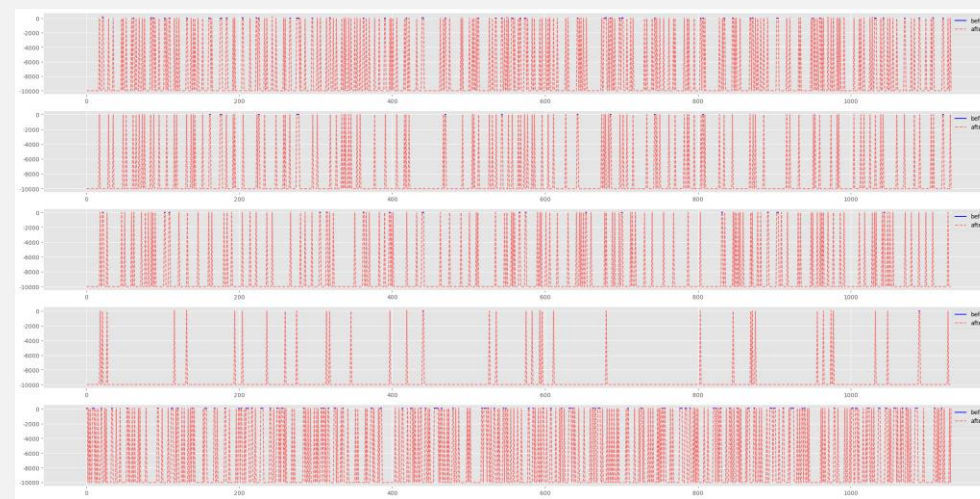
- 
- 01** Intro
  -  **02** Feature Engineering
  - 03** Validation
  - 04** Modeling
  - 05** Outro
-

## Before



&lt; X\_1408 외 데이터 결측치 처리 전 &gt;

## After



&lt; X\_1408 외 데이터 결측치 처리 후 &gt;

제품 코드별, 생산 라인별 고유 칼럼 존재로 인해 결측치가 많아 트리기반 모델 학습에 있어서 일  
반적인 통계량은 모델 학습에 방해가 될 것이라 판단하여 **-9999** 로 결측치를 대체함

## Correlation

<b>X_129</b>	0.275226
--------------	----------

<b>X_128</b>	0.272317
--------------	----------

<b>X_382</b>	0.257216
--------------	----------

<b>X_1219</b>	0.237186
---------------	----------

<b>X_1525</b>	0.236820
---------------	----------

<b>X_3302</b>	0.188313
---------------	----------

<b>X_3266</b>	0.188305
---------------	----------

<b>X_3221</b>	0.188299
---------------	----------

<b>X_3265</b>	0.188276
---------------	----------

<b>X_3262</b>	0.188276
---------------	----------

< A\_31 상관관계 높은 피처 >   < T\_ 31 상관관계 높은 피처 >

Train 데이터셋에 대하여 Y\_Quality 와 X\_Feature 상관관계를 확인한 결과 전체 3326개 칼럼 중 3102개의 칼럼이 상관관계가 0.1 이하로 대부분의 칼럼이 Y\_Quality 와 낮은 상관성을 보임



PRODUCT\_CODE 별로 데이터셋을 나눈 후 Y\_Quality 와 X\_Feature 사이 상관관계가 높은 피처가 있음을 확인



**PRODUCT\_CODE 별로 학습을 나눠서 진행**



## Correlation outro




X\_Feature 사이 상관성이 매우 높음

실제 공정은 연속적 프로세스 과정이지만 변수가 비식별화되어있고 **TIME\_STAMP** 누락으로 인해 제공받은 데이터의 경우 time delay를 고려하기 어려웠음

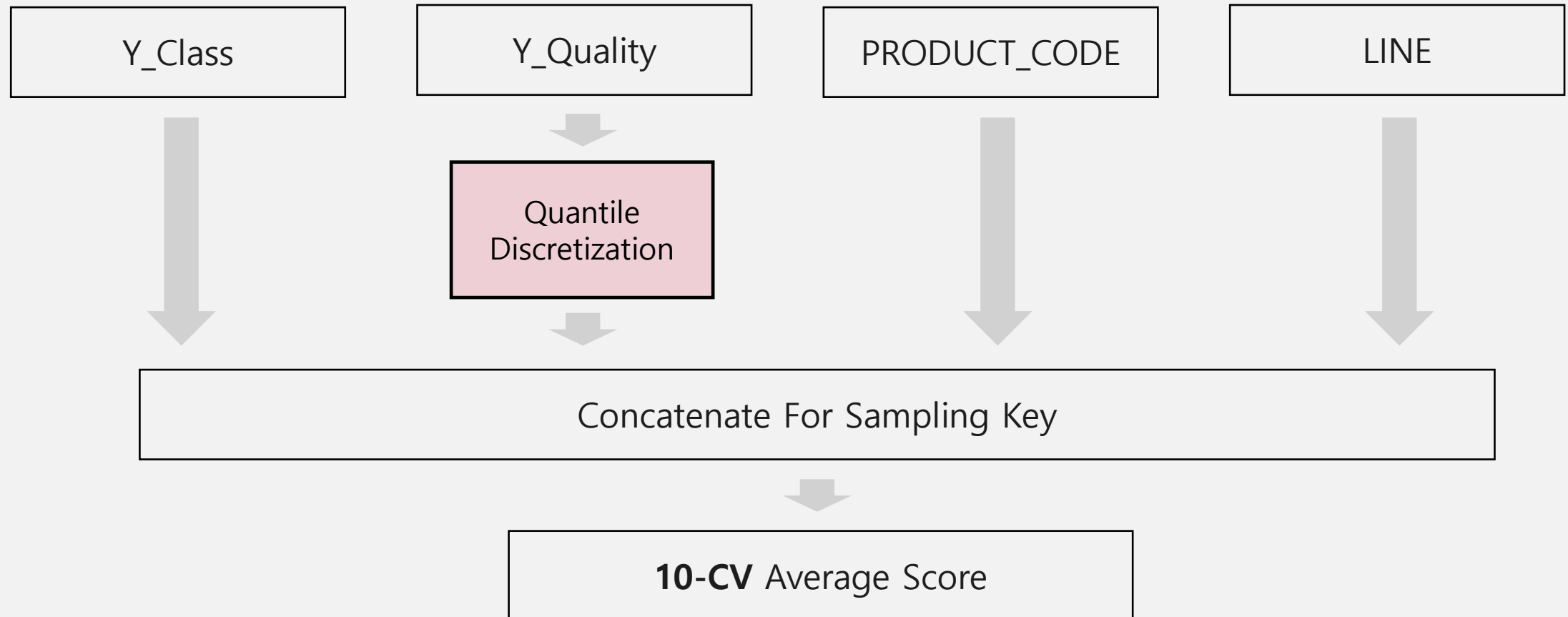
실제 공정 과정에서는 시계열적 요소를 반영하여 더 좋은 성능을 이끌 수 있을 것으로 해석됨



# Contents

- 
- 01** Intro
  - 02** Feature Engineering
  -  **03** Validation
  - 04** Modeling
  - 05** Outro
-

## Validation System - Stratified Sampling



## Thresholding Strategy – Variant of 'MetaClass' Algorithm

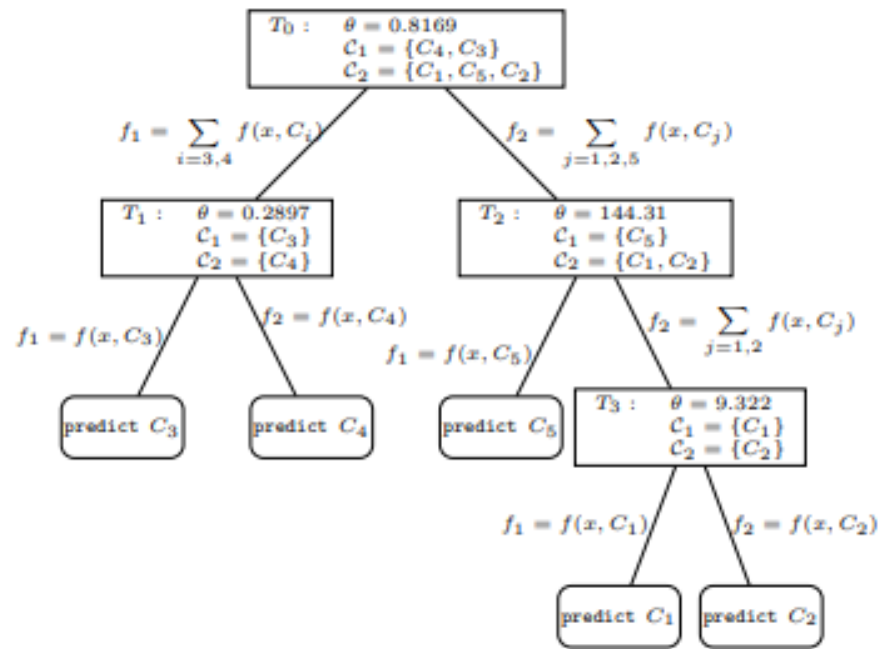
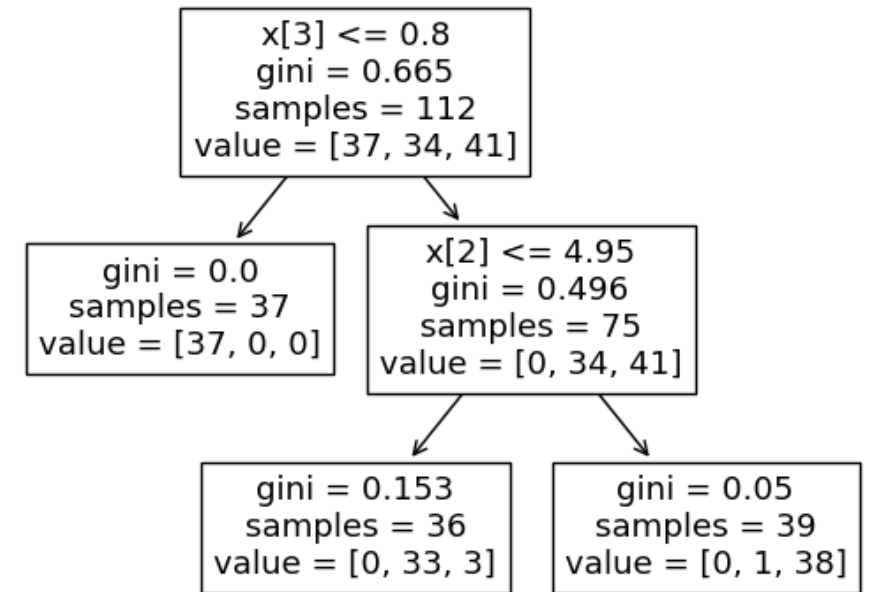


Figure 1. Example run of MetaClass on Nursery, a 5-class problem.

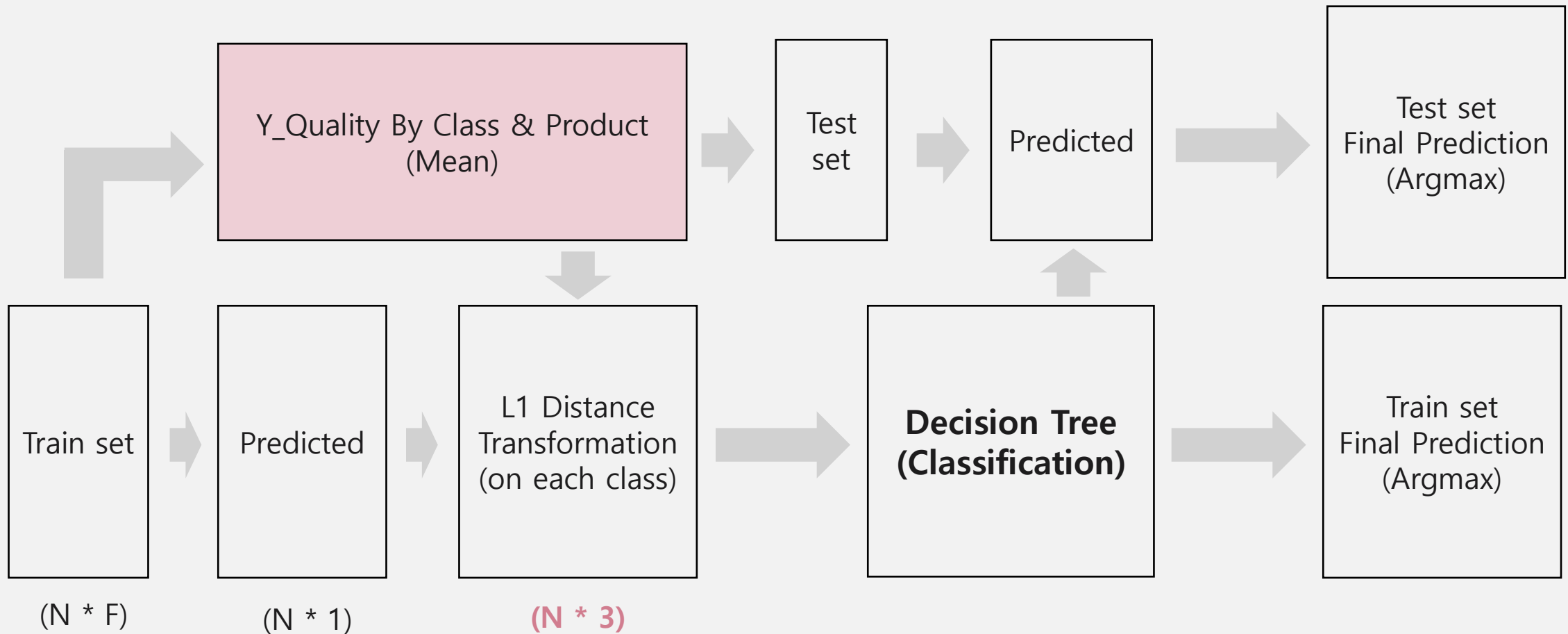
Converted version in ML

## Decision Tree Structure




\* loss functions is gini, use all features

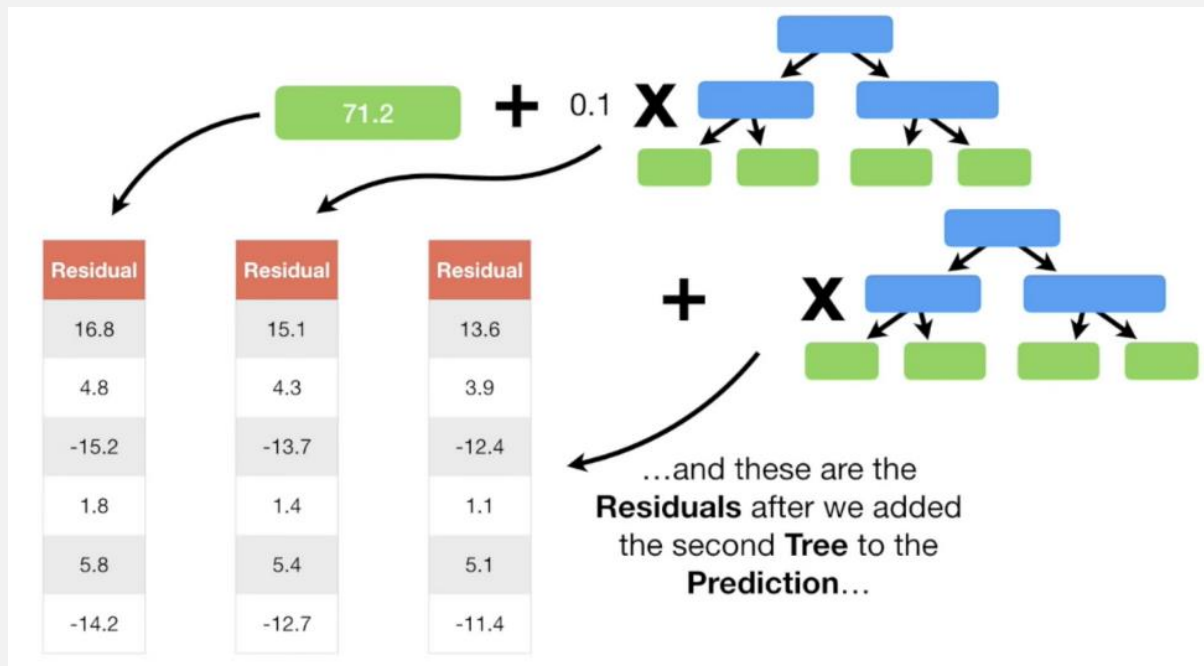
## Thresholding Strategy – Variant of 'MetaClass' Algorithm



# Contents

- 
- 01** Intro
  - 02** Feature Engineering
  - 03** Validation
  -  **04** Modeling
  - 05** Outro
-

## Catboost



## 특징

- Ordering Boosting & Random Permutation
  - Prevent Overfitting
- Auto Ordered Target Encoding & One-hot Encoding
- Categorical Feature Combinations
  - Feature Cardinality ↓
  - 연속형 변수가 아닌 수치형 변수는 범주형 변수로 볼 수 있음
- Optimized Parameter Tuning

```
train.columns[train.nunique() <= 10][3:]  
Index(['X_3', 'X_4', 'X_5', 'X_6', 'X_8', 'X_9', 'X_10', 'X_14', 'X_15',  
      'X_16',  
      ...,  
      'X_3299', 'X_3301', 'X_3303', 'X_3304', 'X_3305', 'X_3306', 'X_3307',  
      'X_3308', 'X_3313', 'X_3325'],  
      dtype='object', length=1622)
```

전체 데이터셋 중 Feature Cardinality 가 10 이하인 Feature 개수 1622개 임을 확인할 수 있다.

## ● 학습시간

Intel(R) Xeon(R) CPU @ 2.20GHz

Model 1, 2 : 약 70sec

Model 3, 4 : 약 45sec

Model 5 : 약 5sec

“공정적용을 위한 빠른 학습이가능”

## 2 현업 적용 가능성


- 앙상블보다 구조가 단순하여 이해가 쉽고 현업 환경에서 시간과 비용 절약 가능
- 빠른 예측속도로 고성능이 요구되는 실시간 시스템에서 유리
- 적은 컴퓨팅 자원을 필요로 하여 제한된 환경에서도 모델을 효과적으로 사용할 수 있음

Valid : 0.68004

Test : 0.69342



# Contents

- 
- 01** Intro
  - 02** Feature Engineering
  - 03** Validation
  - 04** Modeling
  -  **05** Outro
-

[참고\_추가적인 시도 가능성]

## Transpose

IF 시간 정보와 Domain을 활용 가능할 때

-> 각 피쳐들마다 Y\_Quality의 반응에 영향을 미치는 시간이 다를 것

-> 그 시간을 파악한다면 Transpose를 통해 새로운 피쳐들을 만들어 예측 성능이 높아질 것

	A	B	C
1		DSL D-95	FIC21185(F1 Flow)
2	2015-01-01 0:00	396.6	550.4523
3	2015-01-01 1:00	396.6	551.4136
4	2015-01-01 2:00	396.6	550.1314
5	2015-01-01 3:00	396.6	550.2445
6	2015-01-01 4:00	396.6	550.4148
7	2015-01-01 5:00	396.6	551.0786
8	2015-01-01 6:00	396.6	550.369
9	2015-01-01 7:00	396.6	551.0783



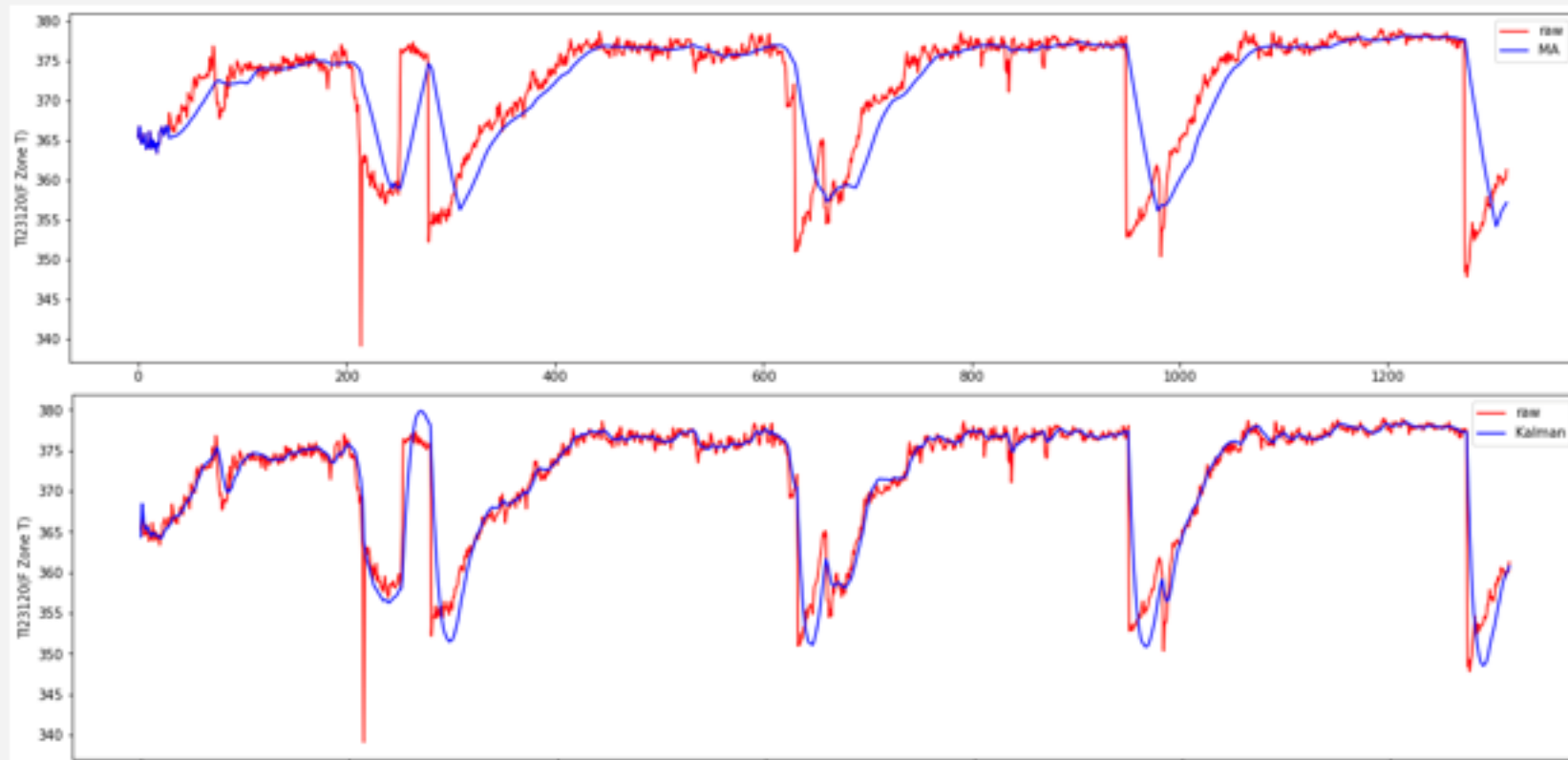
	A	B	C
1		DSL D-95	Before_2Hour_FIC21185
2	2015-01-01 7:00	396.6	551.0786



[참고\_추가적인 시도 가능성]

## Smoothing

-> Kalman filter, Low pass filter, MA, log ..etc 사용



### Moving Average(30)

-> 중심극한정리의 기본값인 30으로 설정

-> 분산과 측정오차 보정

### Kalman Filter

-> 이전 상태와 현재 측정값을 이용하여 예측값을 계산하고, 측정값과 예측값을 조합하여 보다 정확한 추정값을 계산하는 통계적인 방법

감사합니다

팀 주혁이