	용해공정의 로트별 부적합 분석 및 생산성 향상 지침서	문서번호	PAS-I-G-001(0)
		제,개정일	2022. 11. 09
		작성부서	서울과학기술대학팀
		페이지	1 / 18

용해공정의 로트별 부적합 분석 및 생산성 향상 지침서 (가이드)

[표준 작성자 및 제·개정 이력]

번호	구분	제·개정 일자	작성자	주요 제·개정 사유	승인		
					팀장	부서장	본부장
0	제정	2022.11.09	신호준	용해공정의 부적합률 분석 및 생산성 제고를 위함	신호준	박종성	김재호

I. 분석배경

1.1 분석배경

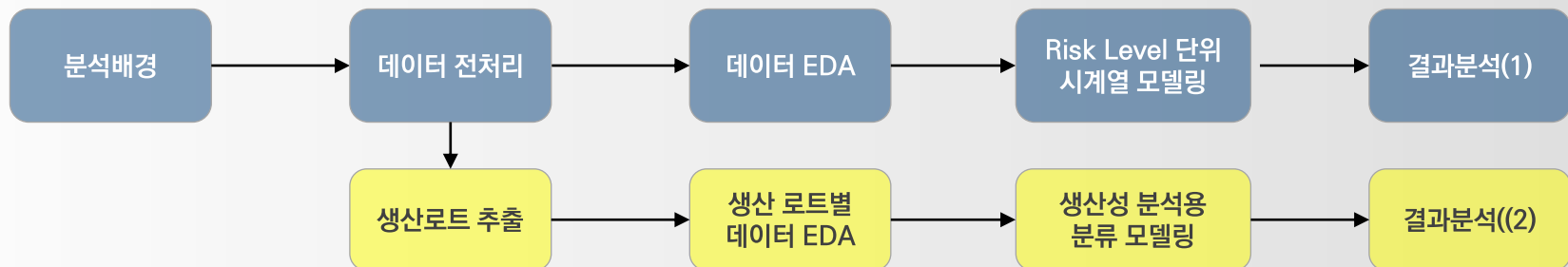
- 생산 과정에 투입된 각 생산 요소(원료·동력·기계·노동·자본 등)가 그 산출에 공헌한 정도를 생산 요소의 생산성이라고 하는데 일반적으로 투입된 생산 요소의 양과 생산된 생산물 양의 비율을 말하는데, 주어진 용해공정 데이터셋을 통해 용해공정의 로트별 생산시간, 생산량 정보와 품질정보를 비교하여 분류모델을 만들고 해석할 수 있다면 제조 원가를 절감하여 자원을 최적화하고, 불명확한 불량원인도 찾을 수 있을 것으로 판단함

1.2 분석목표

- 불량비율에 따른 리스크를 Level 별 구분하여 시계열 데이터 분석 모델링을 진행하고, 보다 신속한 이상탐지 가능여부 확인과 구간별 차이를 비교
- 용해탱크 내용물 중량을 추정하여 생산로트를 구분하고, 생산 로트별 품질 및 생산성 확인 후 이를 모두 만족하는 그룹에 대한 모델 분석 및 해석 실시

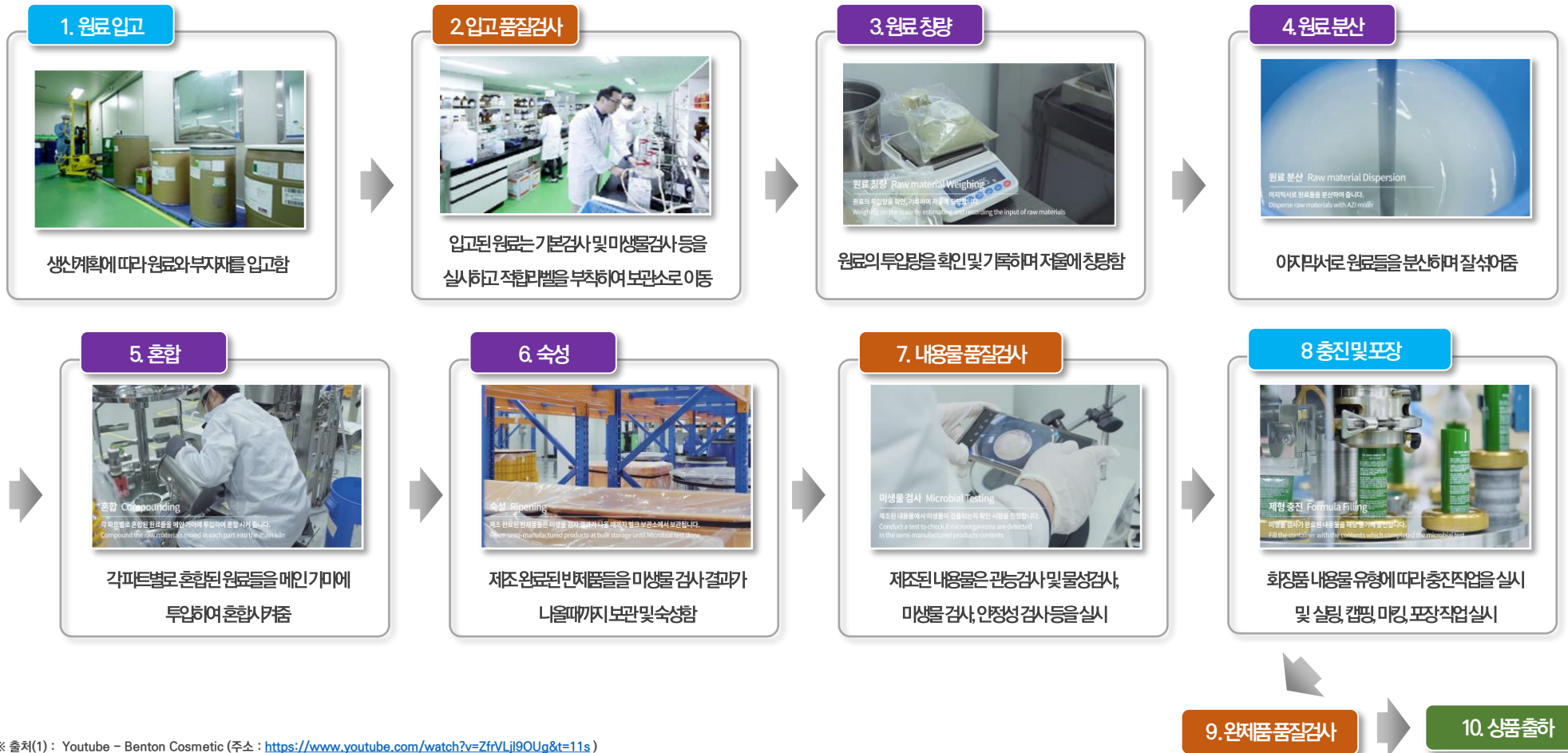
1.3 데이터 분석 프로세스

- 데이터 분석은 시계열 데이터 모델을 활용한 분석과 공정별 생산로트를 추출하여 로트별 품질 수준과 생산성을 비교하는 분류(Classification) 모델을 활용하여 분석을 진행
- 시계열 데이터 분석 모델은 공정 작업을 진행하였던 전 구간을 범위로 하는 분석과 NG가 발생했던 구간을 RISK Level로 구분하여 비교



1.3 화장품 제조공정 프로세스에 대한 이해

- 제조공정은 크게 원료/부자재 입고, 제조, 충전/포장 3단계로 나눌 수 있으며 이를 세부적으로 나누면 아래와 같이 진행된다.



2.1 용해온도와 교반속도의 사이클 비교

- 1분 단위로 “MELT_TEMP” 와 “MOTOR_SPPED” 가 반복되는 사이클을 보여주고 있음

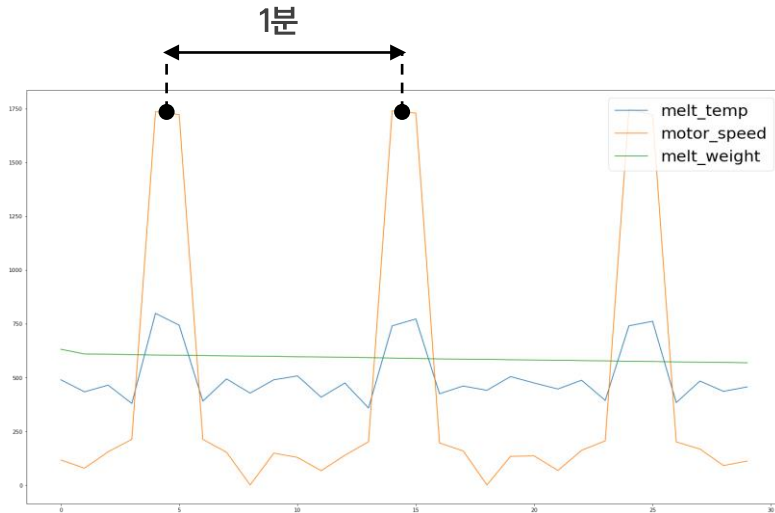


그림00. 변수별사이클비교(60초간격)

분말 원료 투입

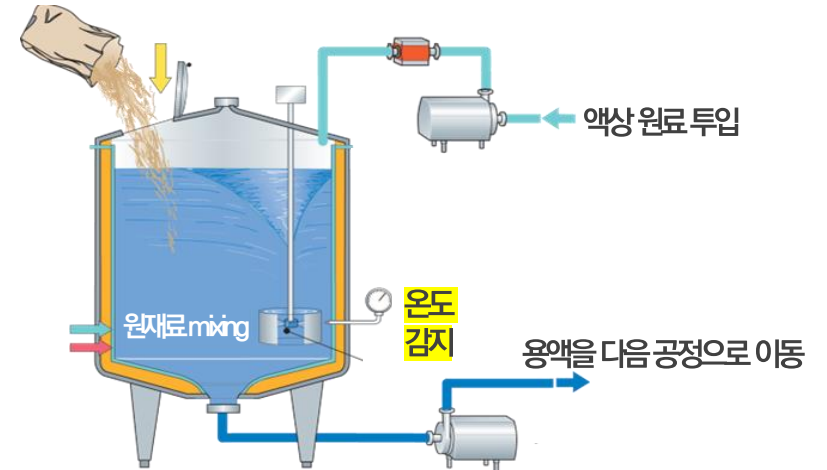


그림00. 원료탱크혼합과정및온도감지프로세스

2.2 생산/설비 관점에서 사이클 비교

- 용해탱크 외벽에 Heating or Cooling Oil Line이 있고, 용액의 온도를 제어하기 위해 Motor로 용액을 Mixing 하는 과정을 진행
- 1분 단위로 Motor 회전수가 급격하게 올라가면, 모터 주변의 Oil Line에서 온도 감지가 일어나기 때문에 두변수의 사이클은 동일할 수 있음

➡ 따라서 60초 구간 중 30초, 36초 사이에서 용해온도와 교반속도가 높게 측정되는 것을 별도의 이상치로 간주하여 전처리할 필요가 없다고 판단

2.3 품질 변동의 원인 : 우연원인과 이상원인

- 우연원인 : 생산 조건이 엄격하게 관리 된 상태에서 발생하는 어느 정도의 불가피한 변동을 일으키는 원인
- 이상원인 : 작업자 부주의, 불량자재 사용, 생산설비 이상, 공정 변화, 부적절 교정 등 만성적으로 존재하는 것이 아닌 산발적으로 발생하는 원인

2.4 MELT_WEIGHT 전처리를 위한 변수 탐색

- MELT_WEIGHT 변수의 평균은 582.9 / 최대값은 55,252 / 표준편차는 1,217.6 수준으로 확인함
- 변수 탐색 결과 공정이 관리된 상태에서 나타나는 우연원인이 아니라 1회 사이클 당 MAX 중량의 최소 10배에서 100배까지 급격하게 증가하는 **이상원인들이 다수 존재**하는 것으로 확인하였으며, 해당 부분에 대한 이상치 처리를 적절하게 진행해 주는 것이 필요함

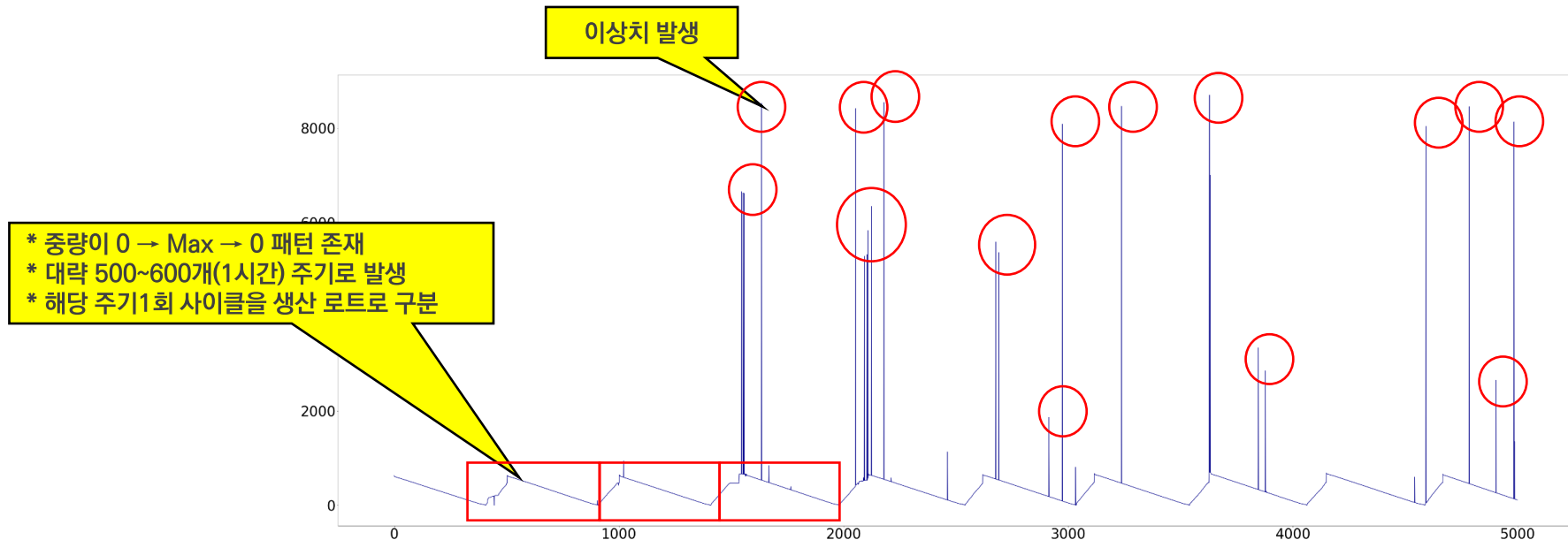


그림0, MELT_WEIGHT 시각화탐색

II. 데이터 전처리 (MELT_WEIGHT)

2.5 MELT_WEIGHT 변수 전처리

- MELT_WEIGHT 변수는 **생산 로트를 구분할 때 사용**해야 하는 **분석과정에서 중요도가 매우 높은 변수**로 해당 변수 전처리에 많은 시간과 인적자원을 할당 시키며 진행하였음
- MELT_WEIGHT 전처리 과정은 데이터의 손실을 최소화 하는 방향으로 크게 3단계 전처리 과정을 통해서 진행하였음

1차 전처리 (구간 필터링)

```
for i in range(len(df)):
    if df['MELT_WEIGHT'][i] >= 52727:
        df['MELT_WEIGHT'][i] = 0.01*df['MELT_WEIGHT'][i]

    elif 52726 >= df['MELT_WEIGHT'][i] >= 28112:
        df['MELT_WEIGHT'][i] = np.NaN

    elif 28111 >= df['MELT_WEIGHT'][i] >= 28078:
        df['MELT_WEIGHT'][i] = 0.1*(df['MELT_WEIGHT'][i]-8000)

    elif 28077 >= df['MELT_WEIGHT'][i] >= 22425:
        df['MELT_WEIGHT'][i] = np.NaN

    elif 22424 >= df['MELT_WEIGHT'][i] >= 10000:
        df['MELT_WEIGHT'][i] = 0.1*df['MELT_WEIGHT'][i]

    elif 9999 >= df['MELT_WEIGHT'][i] >= 8946:
        df['MELT_WEIGHT'][i] = np.NaN

    elif 8945 >= df['MELT_WEIGHT'][i] >= 8000:
        df['MELT_WEIGHT'][i] = df['MELT_WEIGHT'][i]-8000

    elif 7999 >= df['MELT_WEIGHT'][i] >= 2911:
        df['MELT_WEIGHT'][i] = 0.1*df['MELT_WEIGHT'][i]
```

1-1

2차 전처리 (100개 구간별 1.5 IQR 탐지)

```
# 2차-1 4분위수 활용
# Q3+1.5*IQR 이상 Q1-1.5*IQR 이하 null 처리
for i in range(0, len(df), 100):
    eda_2 = df['MELT_WEIGHT'].loc[i:i+99]
    q1 = eda_2.describe()['25%']
    q3 = eda_2.describe()['75%']
    iqr = q3 - q1
    lowest = q1 - 1.5*iqr
    highest = q3 + 1.5*iqr
    for j in range(len(eda_2)):
        if df['MELT_WEIGHT'][i+j] > highest:
            df['MELT_WEIGHT'][i+j] = np.NaN

        elif df['MELT_WEIGHT'][i+j] < lowest:
            df['MELT_WEIGHT'][i+j] = np.NaN
```

2-1

```
# 2차-2 null data를 앞, 뒤 data의 평균값으로 대체
for i in range(1, len(df)-1):
    if pd.isnull(df['MELT_WEIGHT'][i]):
        df['MELT_WEIGHT'][i] = (df['MELT_WEIGHT'][i-1] + df['MELT_WEIGHT'][i+1])/2
```

2-2

```
# 2차-3 null data를 앞 data의 값으로 대체
for i in range(1, len(df)):
    if pd.isnull(df['MELT_WEIGHT'][i]):
        df['MELT_WEIGHT'][i] = df['MELT_WEIGHT'][i-1]
```

2-3

```
# 2차-4 null data를 뒤 data의 값으로 대체
for i in range(0, len(df)-1):
    if pd.isnull(df['MELT_WEIGHT'][i]):
        df['MELT_WEIGHT'][i] = df['MELT_WEIGHT'][i+1]
```

2-4

3차 전처리 (직전 데이터와 편차 비교 보정)

```
# MELT_WEIGHT data 한 칸씩 이동
df['MELT_WEIGHT_SHIFT'] = df['MELT_WEIGHT'].shift(1)
df

# MELT_WEIGHT 원래 data와 이동 data 차이 column 생성
df['diff'] = df['MELT_WEIGHT'] - df['MELT_WEIGHT_SHIFT']
df
```

3-1

```
# 3차-1 차이가 1000보다 크거나 -1000보다 작으면 null 값으로 대체
for i in range(1, len(df)):
    if df['diff'][i] >= 1000:
        df['MELT_WEIGHT'][i] = np.NaN

    elif df['diff'][i] <= -1000:
        df['MELT_WEIGHT'][i] = np.NaN
```

3-2

```
# 3차-2 null data를 앞, 뒤 data의 평균값으로 대체
for i in range(1, len(df)-1):
    if pd.isnull(df['MELT_WEIGHT'][i]):
        df['MELT_WEIGHT'][i] = (df['MELT_WEIGHT'][i-1] + df['MELT_WEIGHT'][i+1])/2
```

3-3

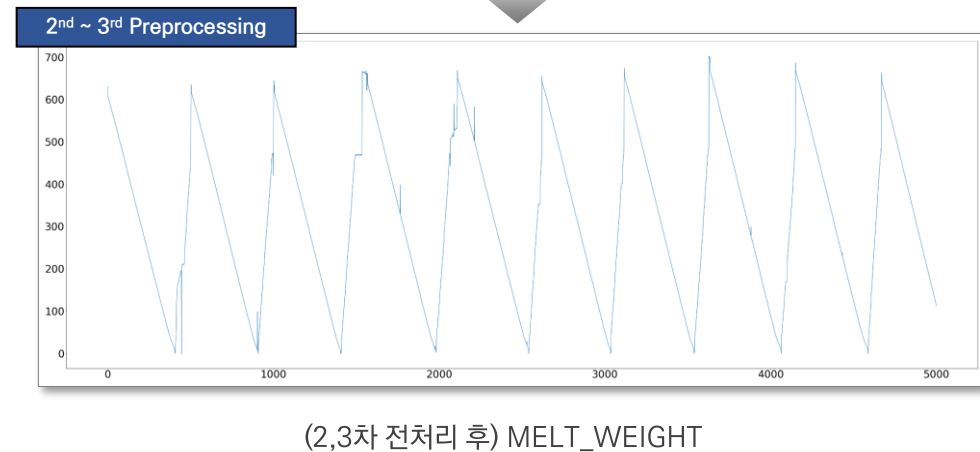
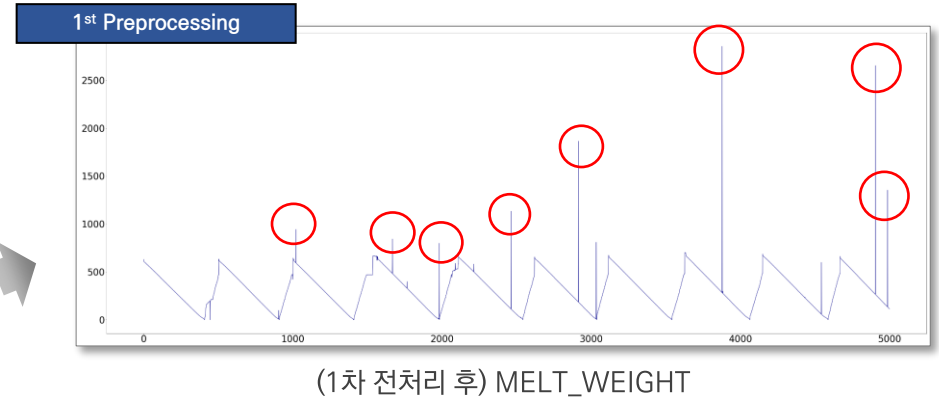
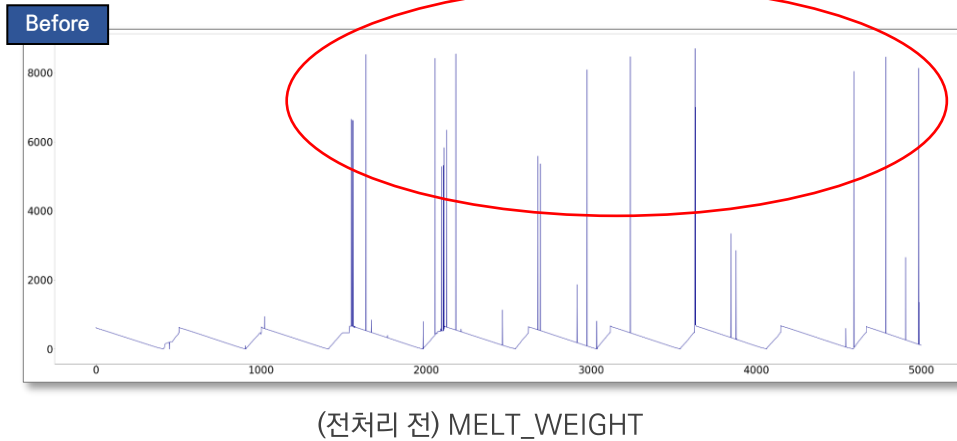
```
# 3차-3 null data를 앞 data의 평균값으로 대체
for i in range(1, len(df)):
    if pd.isnull(df['MELT_WEIGHT'][i]):
        df['MELT_WEIGHT'][i] = df['MELT_WEIGHT'][i-1]
```

3-4

```
# 3차-4 null data를 뒤 data의 값으로 대체
for i in range(0, len(df)-1):
    if pd.isnull(df['MELT_WEIGHT'][i]):
        df['MELT_WEIGHT'][i] = df['MELT_WEIGHT'][i+1]
```

2.6 MELT_WEIGHT 전처리 결과 확인

- 1단계 전처리 프로세스를 통해 센서 오류 또는 비주기적 이상원인이 크게 감지되었던 이상치들이 대폭 조정되었음
- 2~3단계 전처리 프로세스를 통해 이상치로 감지 된 구간들이 앞, 뒤 구간의 평균값 또는 이전값, 이후값 들로 조건에 맞도록 조정 되면서 비교적 안정적인 제조 사이클 형태로 보정을 완료함



2.7 MELT_WEIGHT 변수를 통한 생산로트 구분

- MELT_WEIGHT 변수를 확인 한 결과 내용물 중량이 0으로 시작해서 최대 중량까지 오른 후, 다시 0으로 내리는 사이클을 보여주고 있었음
- 따라서 해당 주기 (1회 사이클)를 용해 공정의 생산 로트 개념으로 구분할 수 있다고 판단하였으며, 전처리를 진행하여 분류작업을 완료하였음
- 전처리 결과 총 1,472개의 생산로트로 구분 되었으며, 데이터 분석시에는 **최소 생산중량과 최소 작업시간** 등을 고려하여 분석할 예정

```
1 zero_point = df.index[df['MELT_WEIGHT'] == 0].tolist()
2 zero_point
3
4 st = 0
5 li = []
6 for i in range(len(zero_point)):
7     globals()['rot_'+str(i+1)] = df.loc[st:zero_point[i]-1]
8     li.append('rot_'+str(i+1))
9     st = zero_point[i]
10 globals()['rot_'+str(len(zero_point)+1)] = df.loc[zero_point[-1]:]
11 li.append('rot_'+str(len(zero_point)+1))
```

생산로트추출코드

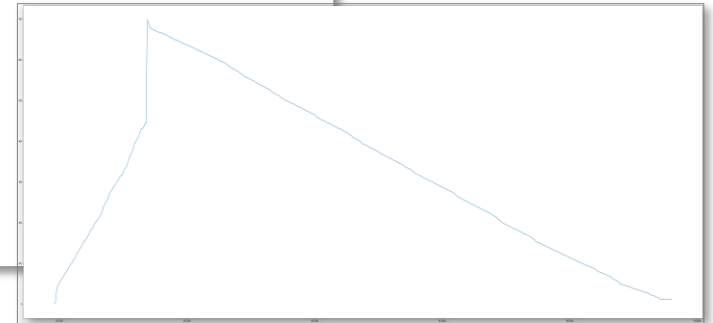
```
1 print(len(li))
2 print(rot_1422)
```

1472

	STD_DT	NUM	MELT_TEMP	MOTORSPEED	MELT_WEIGHT	INSP	TAG	#
811096	2020-04-29 7:49	811096	403	197	0.0	3.19	OK	
811097	2020-04-29 7:49	811097	446	192	4.0	3.19	OK	
811098	2020-04-29 7:49	811098	407	85	37.0	3.19	OK	
811099	2020-04-29 7:49	811099	487	155	46.0	3.20	OK	
811100	2020-04-29 7:50	811100	490	147	52.0	3.20	OK	
...
811577	2020-04-29 8:37	811577	472	163	11.0	3.19	OK	
811578	2020-04-29 8:37	811578	477	70	12.0	3.19	OK	
811579	2020-04-29 8:37	811579	470	146	12.0	3.20	OK	
811580	2020-04-29 8:38	811580	495	125	11.0	3.19	OK	
811581	2020-04-29 8:38	811581	469	0	12.0	3.19	OK	

	MELT_WEIGHT_SHIFT	diff
811096	10.0	-10.0
811097	0.0	4.0
811098	4.0	33.0
811099	37.0	9.0
811100	46.0	6.0
...
811577	12.0	-1.0
811578	11.0	1.0
811579	12.0	0.0
811580	12.0	-1.0
811581	11.0	1.0

[486 rows x 9 columns]



생산로트추출결과확인

3.1 분석실습 가이드북의 문제점 확인

- 해당 데이터셋은 총 835,200개의 인덱스로 구성 TARGET 값은 OK가 658,133개 (79%) , NG가 177,067개 (21%)인 불균형 데이터셋으로 확인
- 해당 데이터셋을 단위 시간 별 NG 비율로 시각화한 결과, 특정 구간에 NG 그룹들이 군집화 되어 있음을 확인할 수 있었음
- 때문에 가이드북 에서 제시하는 방법으로 훈련-테스트 데이터를 분리한다면 테스트 셋 30%에 불균형 비율이 크게 발생되고, OK로만 예측했을 경우 Accuracy가 80% 이상인 모델을 만들 수 있음
- 가이드북에서는 accuracy, F1-score을 OK 관점으로만 제시하였는데, 공정에 사용되는 모델 특성상 NG를 맞추는 것이 더 중요하다고 판단함

[단계 ④] 훈련/테스트 데이터 분리

④-1. 훈련/테스트 데이터 분리

- 딥러닝 모델에 데이터를 훈련시키기 위해서는 데이터를 적절히 분리하는 작업이 필요하다. 학습용 데이터인 훈련 데이터와 성능평가용 데이터인 테스트 데이터를 나눈다. 해당 분석에서는 데이터의 비율을 7:3으로 맞추어 활용하였으며, 특정 날짜를 timestamp로 지정하여 timestamp를 기준으로 분리하는 것도 가능하다.

```
split_date = int(df1.shape[0]*0.7) # df1.shape[0]*0.7 = 584840
train = df1[:split_date]
test = df1[split_date:]
```

[그림 41] 훈련/테스트 데이터 분리를 위한 코드

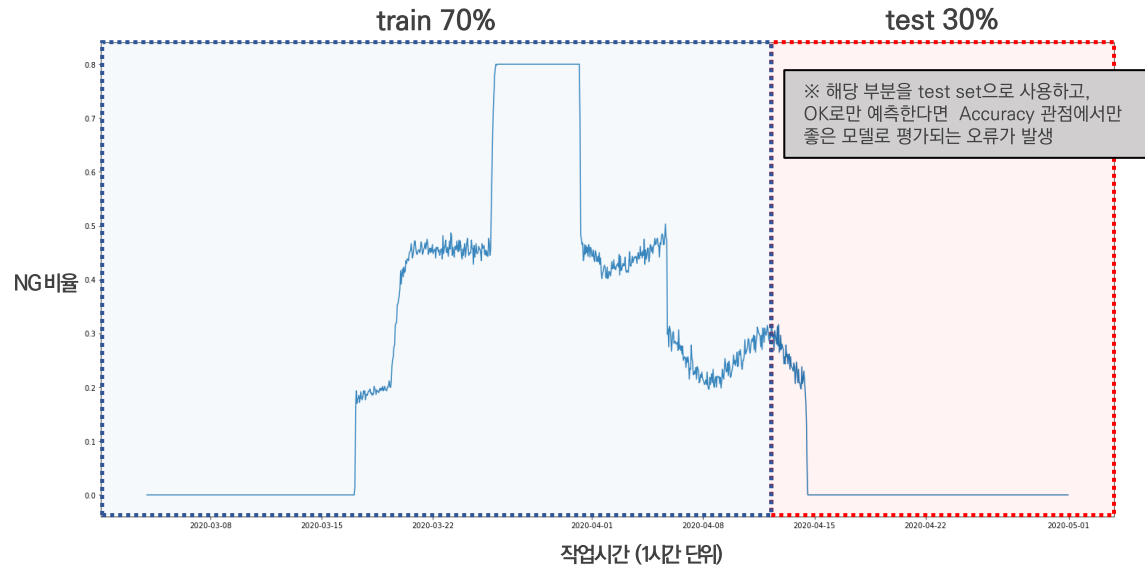


그림0,용해공정의단위시간별 NG비율선형차트

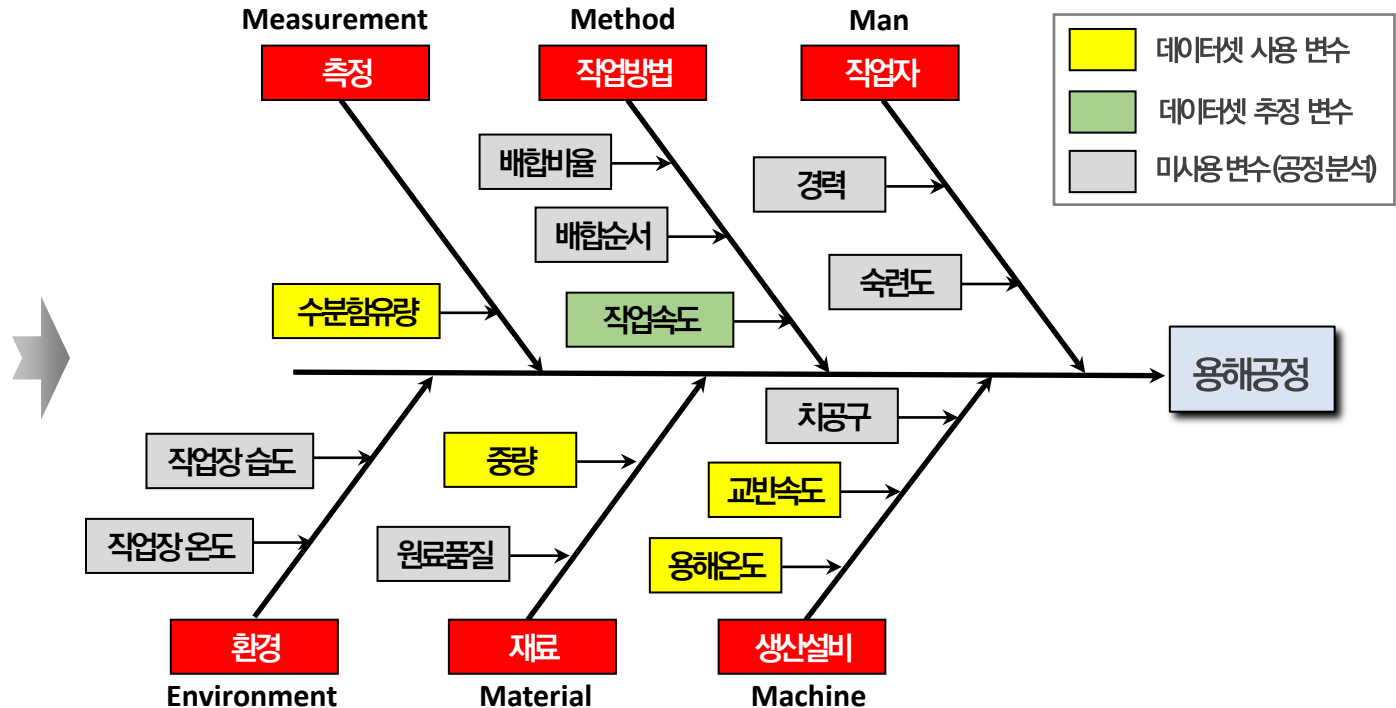
- ✓ NG비율을 시간별로 구분하여 분석해본 결과 시간 인덱스가 다른 Feature들에 비해 큰 비중으로 결과에 영향을 주는 것으로 확인하였음
- ✓ 해당 데이터셋은 NG의 관점에서 평가할 수 있는 F1 score, confusion matrix를 적극 활용하는 것이 필요함

3.2 5M+1E 특성요인도(Cause and Effect diagram)

- 5M+1E 이란 문제의 원인이나 해결해야 되는 과제를 누락없이 분석하기 위해, 원인의 범위를 5M(Man, Machine, Method, Material, Measurement) + 1E(Environment) 로 분류하여 문제의 근본원인을 도출하는 분석 기법
- 특성요인도는 결과에 원인이 어떻게 영향을 미치고 있는가를 나타낸 그림으로, 물고기 뼈다귀 모양을 닮아 『어골도 (Fish-Bone Diagram)』로 불림
- 특성요인도는 공정 관리 및 품질 관리를 위해 사용하며 이상 발생의 원인을 찾아서 조치를 취할 때 사용하며, 5M+1E 를 활용하여 함께 사용 가능함

변수	설명
STD_DT	날짜, 시간 ⇒ 작업속도 추정 기능
NUM	인덱스
MELT_TEMP	용해온도
MOTORSPEED	용해교반속도
MELT_WEIGHT	용해탱크 내용량 (중량)
INSP	수분함유량(%)

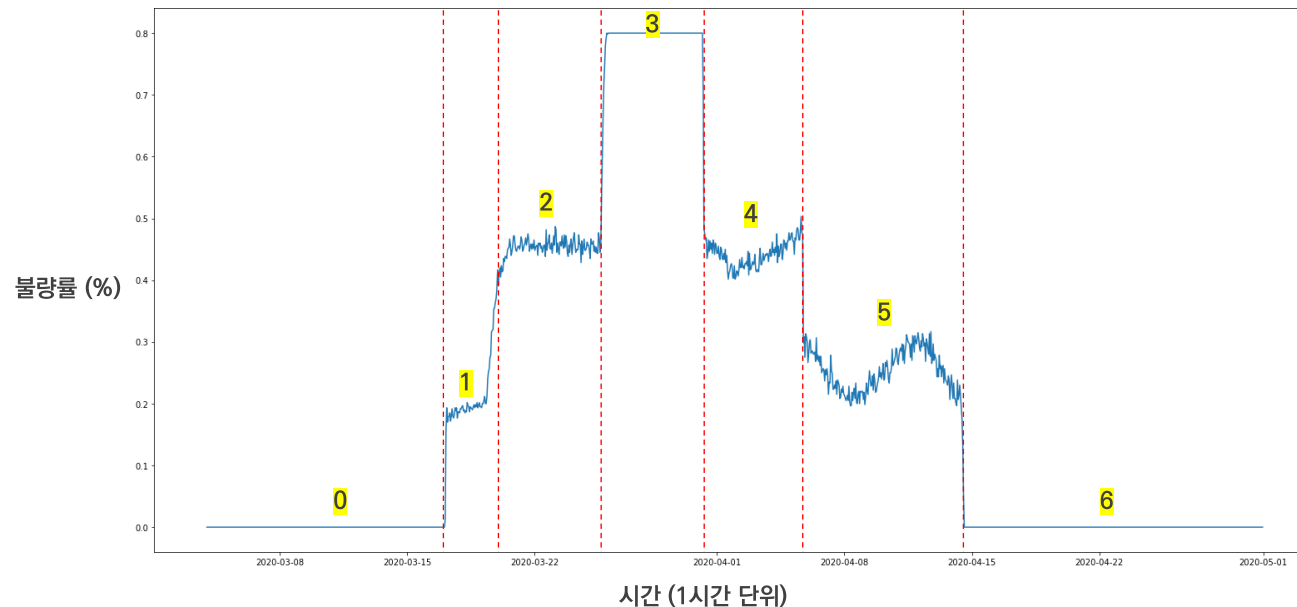
경진대화용 용해탱크 데이터셋.csv




➡ 데이터셋에 제공된 변수 이외에 다른 외부 요인으로 인해 분석결과가 영향받을 수 있음을 염두해 두고 분석하는 것이 필요

4.1 군집별 NG 예측 시계열 모델링

- 아래 그림과 같이 NG 비율에 따라 시간별 트렌드가 군집화 되는 구간으로 분류하였고, 해당 구간별로 개별적인 LSTM 분석을 진행
- LSTM 모델링은 가이드북과 동일한 조건에서 구성을 하였고, SMOTE는 제외하고 실험 준비를 하였음
- NG를 예측 TARGET으로 고려하였고, {1:NG, 0:OK} 으로 라벨링 진행함
- 0 ~ 5번 구간 내에서 Train : Test = 7 : 3 비율로 나누어서 학습을 진행함 (6번 구간은 100% OK 영역으로 별도의 모델링 분석은 진행하지 않음)

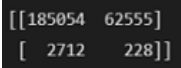
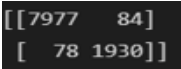
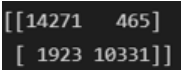
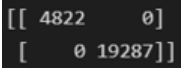
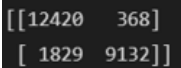



	용해공정의 로트별 부적합 분석 및 생산성 향상 지침서	문서번호	PAS-I-G-001(0)
		제,개정일	2022. 11. 09
	Ⅳ. 시계열 데이터 분석 모델링 (군집별)	작성부서	서울과학기술대학팀
		페이지	12 / 17

4.2 군집별 NG 예측 시계열 모델링 결과

- 공정은 연속적으로 진행 중이기 때문에 현재 시점의 NG/OK 비율을 알고 있다는 가정
- 전체 구간으로 통합하여 예측하는 것에 비해 성능이 월등하게 향상됨
- NG에 대한 f1 score는 0.01에서 약 0.9로 상승
- OK에 대한 f1 score는 0.85에서 0.9 이상으로 상승

[구간별 모델 예측 결과]

구간	NG	OK	Accuracy	Confusion matrix
1~6번 전체	0.01	0.85	0.74	
1번 구간	0.96	0.99	0.98	
2번 구간	0.90	0.92	0.91	
3번 구간	1.00	1.00	1.00	
4번 구간	0.89	0.92	0.91	
5번 구간	0.75	0.93	0.89	
6번 구간	-	-	-	-

4.3 Risk Level 단위별 NG 예측 시계열 모델링 분석



• Risk Level 단위로 분류

- Level 1 (불량률 15% 이하) : 0번 구간+ 6번 구간
- Level 2 (불량률 15% ~ 40% 미만) : 1번 구간 + 5번 구간
- Level 3 (불량률 40% ~ 60% 미만) : 2번 구간 + 4번 구간
- Level 4 (불량률 60 % 이상) : 3번 구간

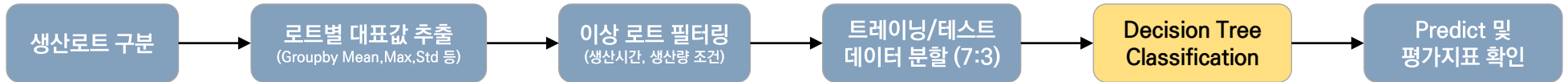
4.4 Risk Level 단위별 NG 예측 시계열 모델링 분석 결과

Risk Level	NG	OK	accuracy	Confusion matrix
Level 1~4	0.01	0.85	0.74	<pre>[[185054 62555] [2712 228]]</pre>
Level 2	0.65	0.91	0.85	<pre>[[34358 1001] [6066 6624]]</pre>
Level 3	0.90	0.93	0.91	<pre>[[27123 1040] [3327 19260]]</pre>

- ✓ 공정은 연속적으로 진행 중이기 때문에 현재 risk level을 알고 있다는 가정함
- ✓ Risk Level 로 관리 시 예측 모델의 크기를 줄일 수 있음
- ✓ Risk Level 1은 NG가 없는 구간이며, 4는 이전 페이지 (3구간) 의 결과와 동일하여 추가 실험을 진행하지 않음
- ✓ Risk를 Level 단위로 분류하여 관리를 하는 것도 비교적 좋은 성능을 나타냄

5.1 모델분석 프로세스 및 모델선정 이유

- 분석모델로는 대표적인 White-Box 모델인 Decision Tree Classification을 사용하였으며, 이유는 해당 데이터셋 자체가 생산 및 공정 조건을 탐색하고 설명하는 것이 중요하다고 판단함.
- 현재 데이터셋 기준으로 로트별 조건인 경우 GPU 환경을 사용하지 않아도 될 만큼 데이터가 크지 않으므로 가볍게 사용할 수 있는 모델을 선택함



5.2 생산성 & 품질 분류모델 데이터셋

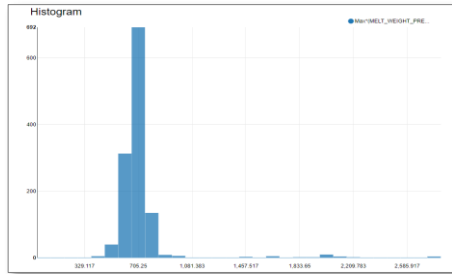
- 용해탱크 시계열 공정 데이터를 로트 단위로 Groupby하여 평균, 최대값, 표준편차, 분산 등 대표값을 추출하여 구성
- 로트단위로 추출 시, 날짜, 시간 변수의 영향을 최소화 하기 위해 Lot No.를 Random으로 재배열
- NG_RATE는 해당 로트의 Index num개수의 합을 NG발생 개수의 합으로 나누어서 계산함
- 해당 로트의 생산량은 MELT_WEIGHT의 MAX값으로 정의하였으며, 이것을 Count(Num)으로 나누면 시간당 생산속도를 추정할 수 있음

모델링시사용제외

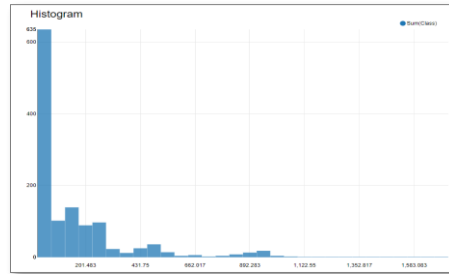
Random LOT No.	Count *(NUM)	Sum (Class)	MELT_WEIGHT				MELT_TEMP					MOTORSPEED					INSP			NG_RATE	Pduction_RATE	4-Class
			Max	Mean	Stdev	Variance	Max	Min	Mean	Stdev	Variance	Max	Min	Mean	Stdev	Variance	Max*	Min	Mean			
1	582	0	688	311.54	192.95	37230.80	815	332	508.43	127.05	16141.71	1774	0	458.40	639.64	409137.56	3.22	3.18	3.19	0.00	1.18	A
2	448	180	507	243.61	142.39	20275.98	794	347	509.63	127.47	16248.64	1771	0	457.12	638.73	407972.84	3.22	3.18	3.19	0.40	1.13	C
3	792	0	703	391.50	198.21	39287.99	810	344	508.02	127.92	16362.32	1775	0	459.73	638.02	407073.53	3.22	3.18	3.19	0.00	0.89	B
⋮																						
1197	1336	1070	672	513.33	204.15	41677.81	805	337	509.36	128.93	16623.09	1775	0	458.34	639.13	408482.98	3.22	3.18	3.19	0.80	0.50	D
1198	504	0	627	297.52	176.38	31108.51	801	343	509.92	128.89	16612.55	1768	0	462.97	643.52	414120.10	3.22	3.18	3.19	0.00	1.24	A
1199	575	460	707	332.33	195.95	38397.02	789	343	508.71	128.05	16396.51	1761	0	460.31	640.85	410683.67	3.23	3.18	3.19	0.80	1.23	C

5.3 생산로트 데이터셋 시각화 (히스토그램)

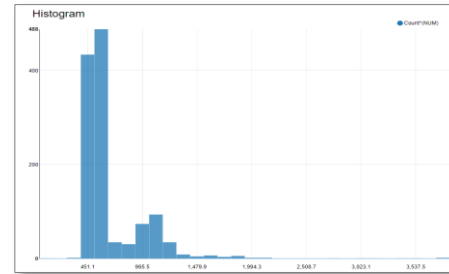
- 생산로트 데이터셋의 시각화 결과는 아래와 같으며, 평균 생산중량은 500 ~ 700, 불량수는 140여개, 생산시간은 670여개 (1시간) 정도로 확인하였음
- 생산시간 10분 이상 or 생산량이100 이상을 만족하면서, 생산량이 최대 1,000을 넘지 않는 로트를 필터링하여, 대표적인 생산 조건들로 분석을 진행함



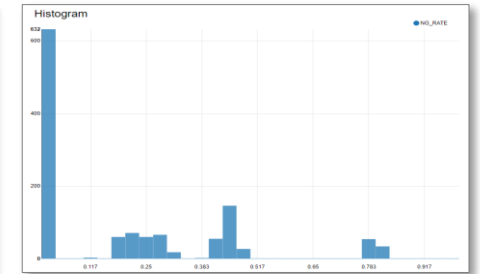
로트별 생산량 (최대값)



로트별 불량수



로트별 생산시간



로트별 불량률

5.4 생산로트 데이터셋의 생산성&품질 분류 모델링을 위한 TARGET 변수 라벨링

- 생산성과 품질은 트레이드 오프 관계에 있어서 두 가지를 모두 만족하는 것이 쉽지 않음
- 생산성, 품질을 모두 만족하는 로트를 구분하여, 분류 모델로 분석하고 이들의 차이의 해석 필요

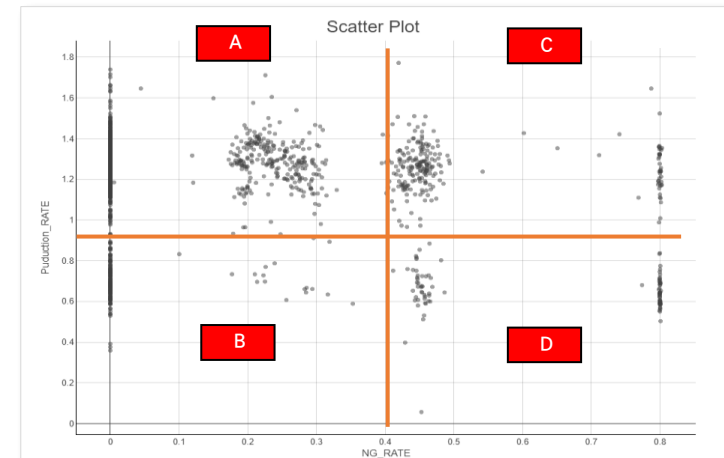
※ 생산성 & 품질 분류 기준은 다음과 같음

A Group (품질우수 & 생산성 우수) : NG_RATE < 0.4 and Production_Rate > 0.9

B Group (품질우수 & 생산성 미흡) : NG_RATE < 0.4 and Production_Rate <= 0.9

C Group (품질미흡 & 생산성 우수) : NG_RATE >= 0.4 and Production_Rate > 0.9

D Group (품질미흡 & 생산성 미흡) : NG_RATE >= 0.4 and Production_Rate <= 0.9



5.5 생산성 & 품질 그룹에 대한 Decision Tree 모델링 분석결과 (Min Samples Node = 30개, Quality Index = Gini)

Scorer View

Confusion Matrix

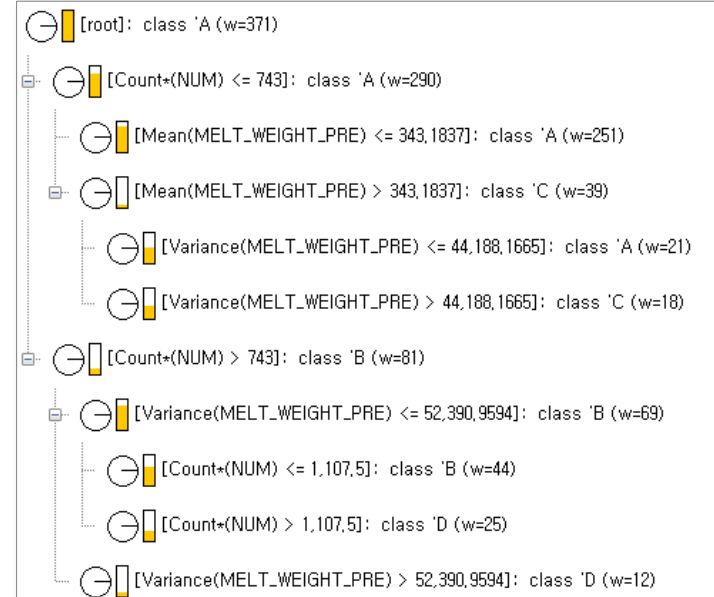
Rows Number : 371	A (Predicted)	B (Predicted)	C (Predicted)	D (Predicted)	
A (Actual)	212	2	10	4	92.98%
B (Actual)	1	31	0	14	67.39%
C (Actual)	59	1	8	1	11.59%
D (Actual)	0	10	0	18	64.29%
	77.94%	70.45%	44.44%	48.65%	

Class Statistics


Class	True Positives	False Positives	True Negatives	False Negatives	Recall	Precision	Sensitivity	Specificity	F-measure
A	212	60	83	16	92.98%	77.94%	92.98%	58.04%	84.80%
B	31	13	312	15	67.39%	70.45%	67.39%	96.00%	68.89%
C	8	10	292	61	11.59%	44.44%	11.59%	96.69%	18.39%
D	18	19	324	10	64.29%	48.65%	64.29%	94.46%	55.38%

Overall Statistics

Overall Accuracy	Overall Error	Cohen's kappa (κ)	Correctly Classified	Incorrectly Classified
72.51%	27.49%	0.469	269	102



- ✓ A 클래스의 Rule이 Count(num) <= 743 & MELT_WEIGHT <= 343으로 생산시간이 길거나 생산량이 많다고 생산성이 좋은 것이 아님으로 확인
- ✓ B 클래스의 Rule을 확인한 결과 MELT_WEIGHT의 Variance도 생산성과 품질에 영향을 주는 것으로 확인하였으며 이를 관리하는 것도 검토가 필요
- ✓ 로트별 공정변수가 분기에 사용되지 않은 것을 확인할 때 생산성과 품질에 영향을 주는 변수로 작업시간과 생산량이 더 중요한 변수임을 확인하였고, 공정별 작업 조건들은 큰 영향력이 없었던 것으로 확인
- ✓ 결과적으로 작업량이 늘어나는 경우와 작업시간이 늘어나는 경우 품질과 생산성에 영향이 있을 수 있으므로 적절한 공정 스케줄 관리가 필요함

	용해공정의 로트별 부적합 분석 및 생산성 향상 지침서	문서번호	PAS-I-G-001(0)
		제,개정일	2022. 11. 09
	Ⅶ. 결론 (분석결과 및 시사점)	작성부서	서울과학기술대학팀
		페이지	17 / 17

6.1 결론

- 불량 발생 비율 구간 및 불량률 기준에 따른 RISK를 레벨화 하여 LSTM 시계열 데이터 분석모델을 분석함
- Risk를 Level 단위로 분류하여 관리를 하는 것도 비교적 좋은 성능을 나타냄
- 생산로트별 공정조건 및 생산조건을 확인하여 품질과 생산성 모두 우수한 조건을 만족하는 분류 모델을 찾고 해석을 실시하고자 하였음
- 작업량이 늘어나는 경우와 작업시간이 늘어나는 경우 품질과 생산성에 영향이 있을 수 있으므로 적절한 공정 스케줄 관리가 필요함

6.2 중소제조기업에 미치는 파급효과

- 전통적인 통계적 품질관리 관점에서는 불량 트렌드를 예측하는데 슈하르트 3시그마 관리도를 사용하는데 딥러닝 시계열 모델을 사용하여 보다 신속하고 정확하게 불량을 예측 및 관리하는 것이 가능함
- 생산성, 품질을 모두 만족하는 로트를 구분하여, 분류 모델로 분석하고 이들의 차이의 해석하여 공정조건을 최적화 하는 것이 필요하며, 불량 또는 생산성 등 단일 기준으로만 의사결정을 하는 것이 아닌 복합적인 차원에서 전략적인 의사결정을 내릴 수 있어야 함
- 해당 분석 지침서를 통해 중소제조업체도 본 분석에서 적용된 관점으로 데이터를 분석해보고 해석한다면 생산성과 품질을 모두 만족하는 공정조건을 찾는 데 도움이 될 것이라고 판단함