

제 2회 K-인공지능 제조데이터 분석 경진대회

---

# 시계열 분석 기반 용해물 품질 예측 AI 모델 개발

제일

김민식, 유광열, 고아름

1

## 공모배경

- 1) 문제 사항
- 2) 분석 목표

2

## 데이터 분석

- 1) 데이터 정의
- 2) 데이터 전처리

3

## 분석 모델

- 1) 이상 탐지 모델
- 2) 머신 러닝 모델
- 3) 순환 신경망 모델

4

## 결과

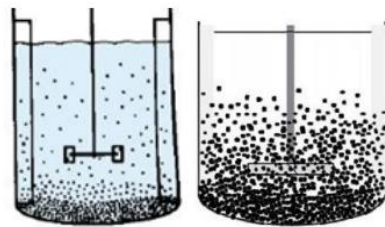
- 1) 분석 결과
- 2) 개선 방향
- 3) 기대 효과

## [공모 배경]

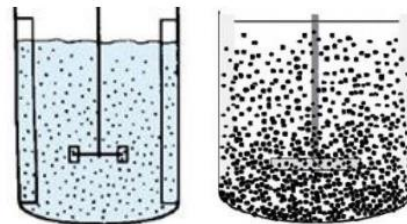
- ▶ 용해공정은 원재료의 전처리를 수행하는 첫 번째 단계이니 본 공정의 품질이 후공정 및 완제품의 품질에 미치는 영향이 큼
- ▶ 용해 품질에 영향을 미치는 많은 요인들이 존재하며, 현장 작업자는 경험과 노하우에 의존하여 대처할 수 밖에 없음
- ▶ 설비 또는 현장상황으로 인해 중간에 내용물을 확인하는 것이 불가능하여 용해상태를 확인 불가, 후공정까지 진행되어야 품질 확인이 가능한 경우도 있음
- ▶ 생산단계가 상당히 많이 진행되기까지 품질이 보장되지 못함

## [분석 목표]

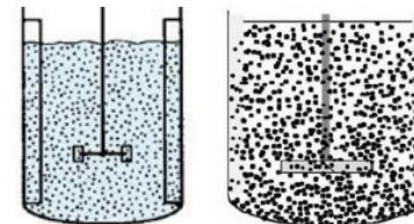
- ▶ 공정 중 실시간으로 변화하는 설비 운영값(온도, 교반 속도, 내용량)과 주요 품질검사항목의 결과값을 모델링
- ▶ 생산 중에도 생산품질 예측 및 공정 제어 간 필요 요소 분석



(a) partial suspension  
( $80\% \leq \text{RSD}$ )



(b) complete suspension  
( $20\% \leq \text{RSD} \leq 80\%$ )



(c) homogeneous suspension  
( $20\% \leq \text{RSD}$ )



» 식품 가공(분말요 크림 제조) 중 용해물 혼합 과정의 설비 및 품질 데이터

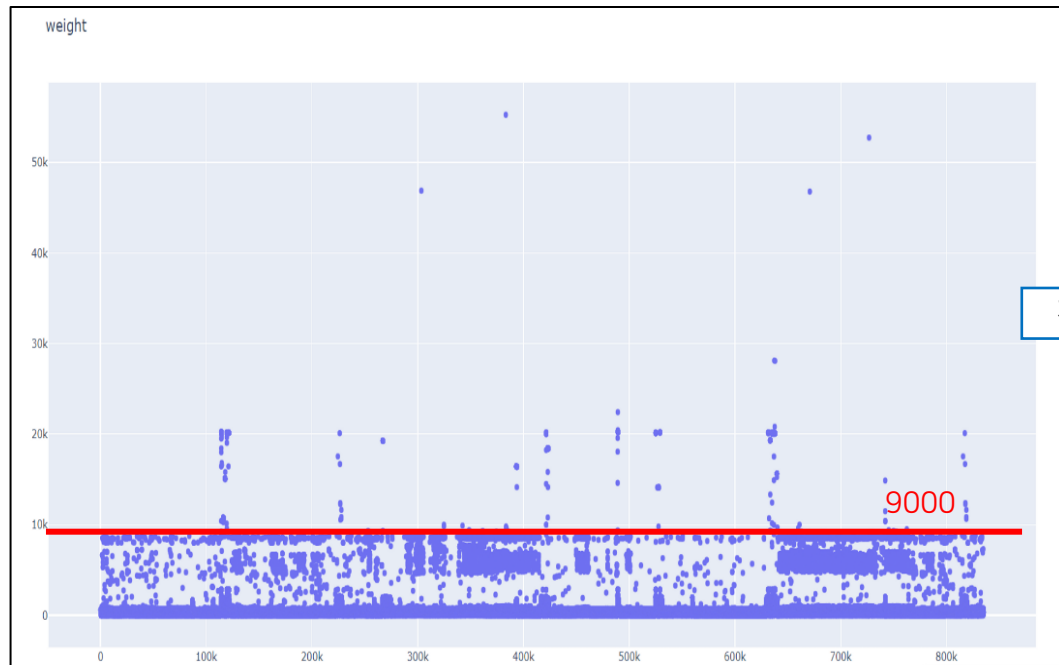
» 수집 주기 : 약 6초

» 수집 기간 : 2020년 3월 4일 ~ 2020년 4월 30일(약 2개월)

변수 조건		내용
독립 변수	용해 온도	가공 과정의 <b>용해물을 용해</b> 하기 위한 온도(범위: 0 ~ 80℃)
	교반 속도	가공 과정의 용해물을 섞기 위한 <b>모터의 속도</b>
	내용량	용해물 혼합 과정의 <b>용해물 양</b>
	수분 함유량	용해물 혼합 과정의 <b>수분 정도</b> , 일종의 <b>범주형 변수</b> 로 구성
종속 변수	불량 여부	가공 후 식품에 대한 '양품/불량' 여부 데이터 기간 중 <b>29일</b> (3월 17일 ~ 4월 14일)에서 <b>불량 발생</b>

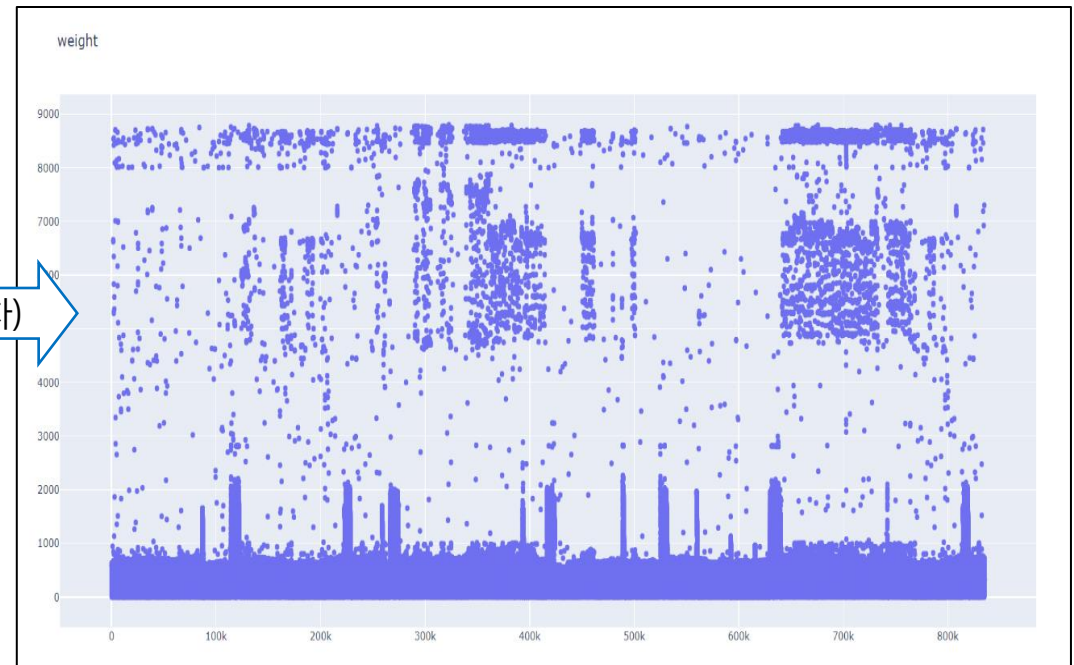


변수명	데이터 전처리	수정 예시	
		수정 전	수정 후
날짜 시간	- 6초 주기로 수집된 데이터로, 날짜 시간 정보 초 단위 입력	2020-03-04 00:00	2020-03-04 00:00:06
용해 온도, 교반 속도	- 생략된 소수점 표시를 위해 10으로 나눔	489	48.9
내용량	- 데이터 분포 파악 시 9000 이상은 이상치로 판단 → 보간법 대체 처리 - 2차 전처리 이후 최종 전처리 진행	10900	8500(주변 값 보간)



(그림) 내용량 기존 분포

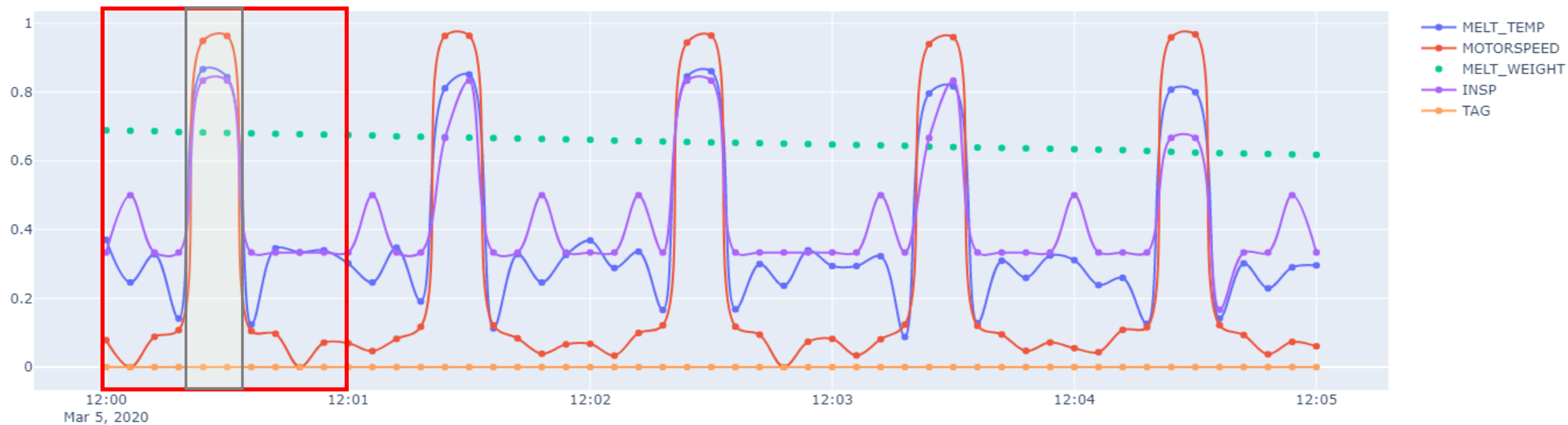
전처리(1차)



[그림] 내용량 분포 세부 확인 (9000 초과 보간법 적용 후)

[가정 1] 용해 온도 및 교반 속도의 주기는 1분을 기준으로 증가 및 반복

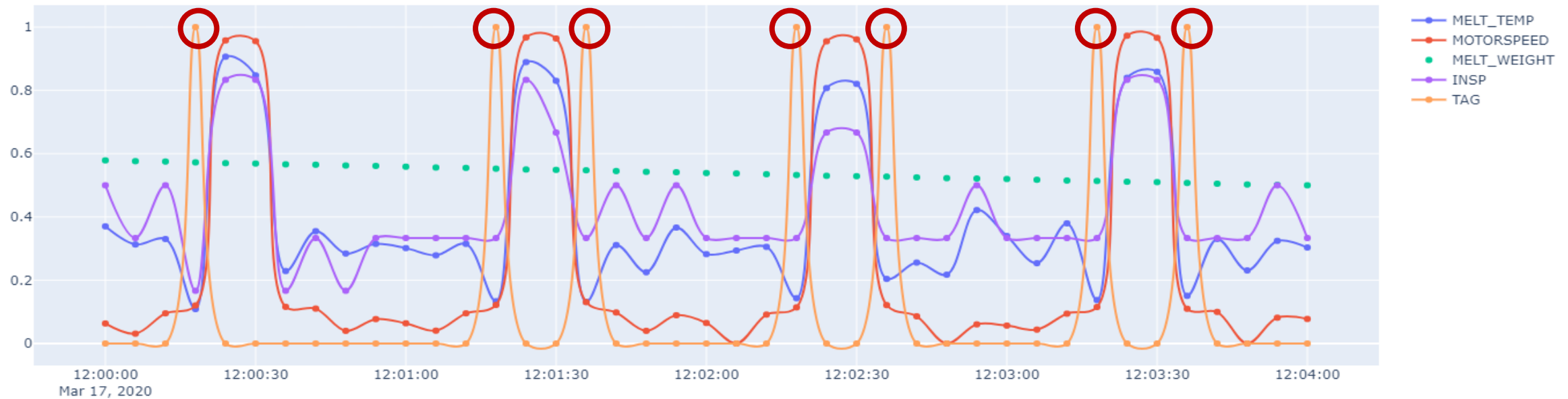
2020-03-05(정상일자)



[그림] 양품 시 분포 파악

[가정 2] 품질이 불량일 때, 온도&교반 속도가 증가 후 감소할 때 순간적으로 “정상”에서 “불량”으로 변하는 현상

2020-03-17(불량일자)

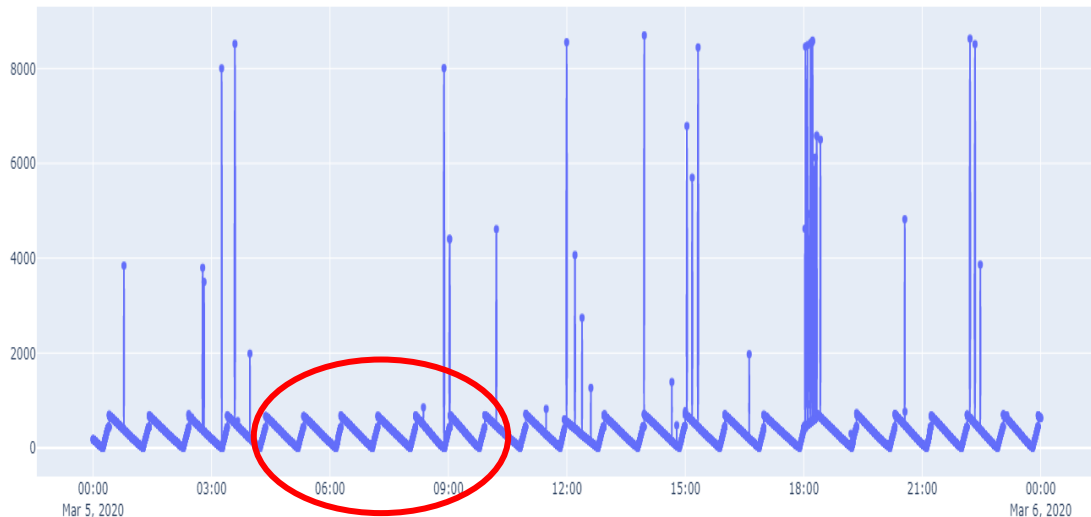


[그림] 불량일 때 분포 파악

## [가정 3] 용해 탱크 내용량 최대치 : 680 , 원재료 투입 ~ 토출 주기 추정

용해탱크 내용량(중량)

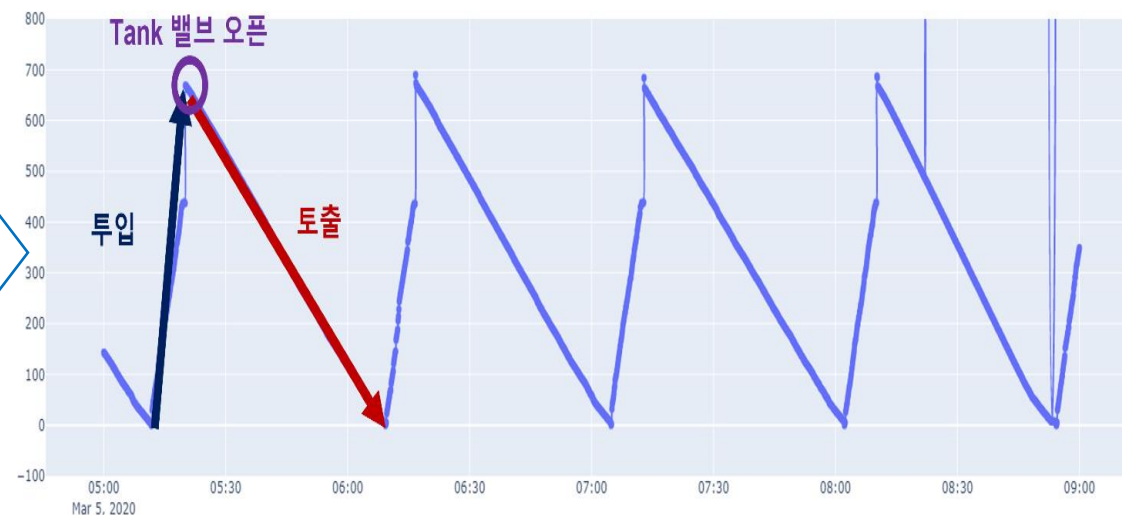
(그림) 내용량 주기 확인



확대

용해탱크 내용량(중량)

(그림) 용해탱크 내용량 기준 토출 사이클 시각화



### » 용해 탱크의 내용량은 지그재그 패턴을 보임

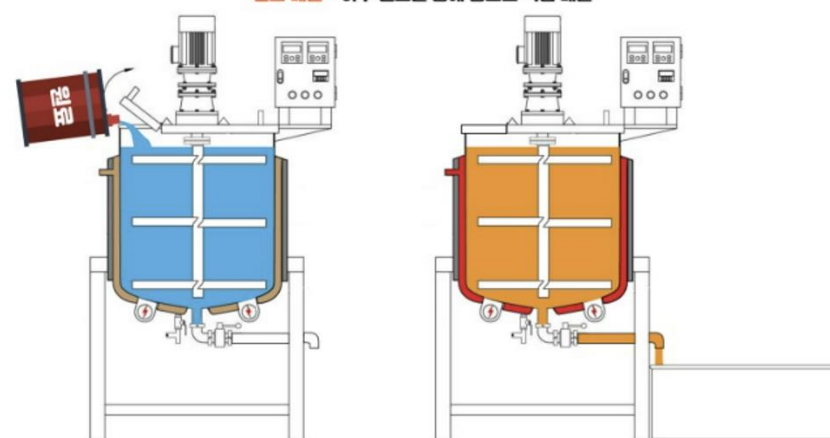
[용해 탱크 내용량의 최대는 680, 패턴에 벗어난 값들은 센서 데이터 이상치 가정]

### » '용해 탱크 투입&배출 원리'를 통해서 내용량 투입 ~ 토출 기간인 "투입 주기" 추론

[최대치 680기준, 투입 및 토출 반복하는 주기 가정]

### » '내용량의 투입 및 토출 되는 기간'을 별도로 '투입 주기' 파생변수 생성

원료 투입 - 상부 두껍을 열어 직접 투입  
원료 배출 - 하부 밸브를 통해 물으로 직접 배출



[그림] 용해 탱크 원료 투입 및 배출 원리 예시 사진





## [가정 4] 용해 탱크의 내용량의 투입 주기에 대한 별도 전처리 진행 필요

	MELT_WEIGHT	datetime	del_time
2020-03-04 00:44:30	0.0	2020-03-04 00:44:30	0 days 00:03:54
2020-03-04 01:30:48	0.0	2020-03-04 01:30:48	0 days 00:46:18
2020-03-04 02:20:36	0.0	2020-03-04 02:20:36	0 days 00:49:48
2020-03-04 02:20:48	0.0	2020-03-04 02:20:48	0 days 00:00:12
2020-03-04 04:14:00	0.0	2020-03-04 04:14:00	0 days 01:53:12
2020-03-04 05:03:24	0.0	2020-03-04 05:03:24	0 days 00:49:24
2020-03-04 05:53:48	0.0	2020-03-04 05:53:48	0 days 00:50:24
2020-03-04 06:46:12	0.0	2020-03-04 06:46:12	0 days 00:52:24

[그림] 용해탱크 내용량이 0이 되는 주기 (투입주기 파악)

» 용해 탱크의 내용량의 주기를 정확하게 파악의 필요성이 대두

[용해 탱크 내용량의 0이 되는 시점을 기준으로 파악]

» 투입 주기의 경우 다양하게 나타나는 것을 확인

[전반적인 주기: 약 50분 정도로 추정]

## 내용량 특수 경우(1) - 투입주기 짧은 경우

MELT\_WEIGHT 주기 찾기



[그림] 투입주기가 짧은 경우의 내용량 EDA

» 투입 주기가 짧은 경우를 구체적으로 EDA

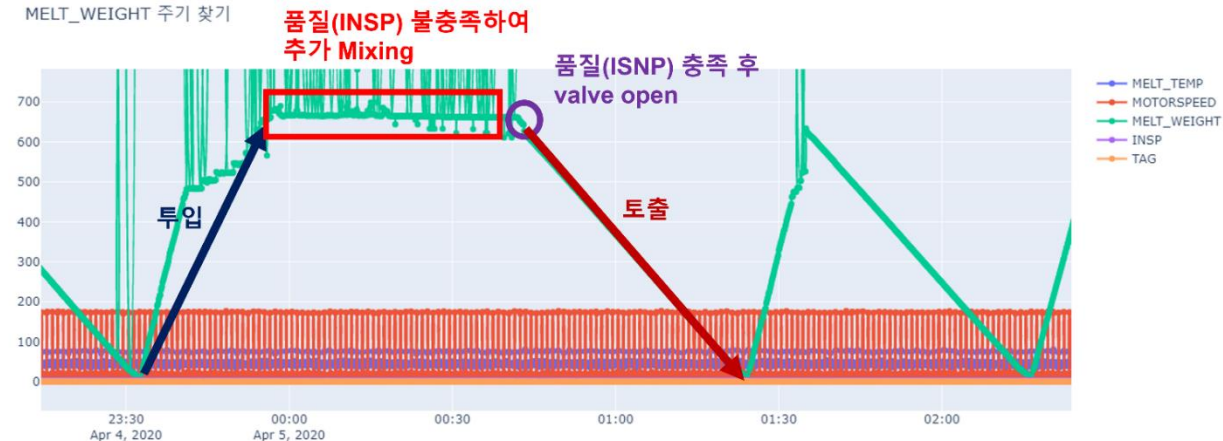
(투입 과정에서 용해 탱크의 내용량이 0이 되는 시점이 발생)

» 이는 계측 오류로 생긴 이상치로 추정



## 내용량의 특수 경우(2) - 투입 주기가 긴 경우

MELT\_WEIGHT 주기 찾기



(그림) 투입주기가 긴 경우의 내용량 EDA

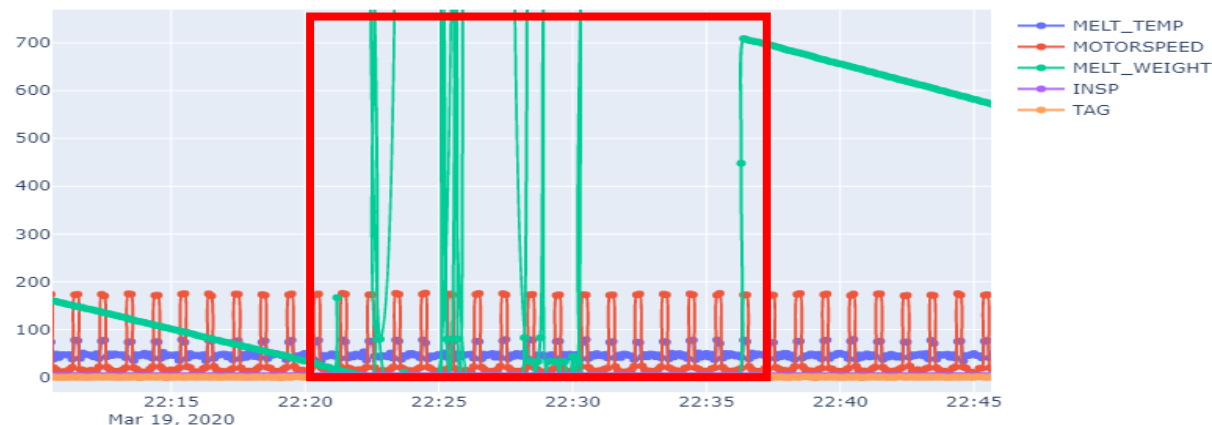
» 용해 탱크의 내용량이 최대치로 유지 후 토출하는 형태를 가짐

[가정]

- 품질을 체크해 수분량(INSP)이 불량 조건으로 추정될 경우, 혼합시간을 더 길게 가지고 밸브 개방하는 품질 체크 밸브 제어시스템 존재 가정

## 내용량의 특수 경우(3) 토출 ~ 투입되는 과정에서의 이상치

MELT\_WEIGHT 주기 찾기



(그림) 토출 및 투입 시기 발생에 대한 이상치

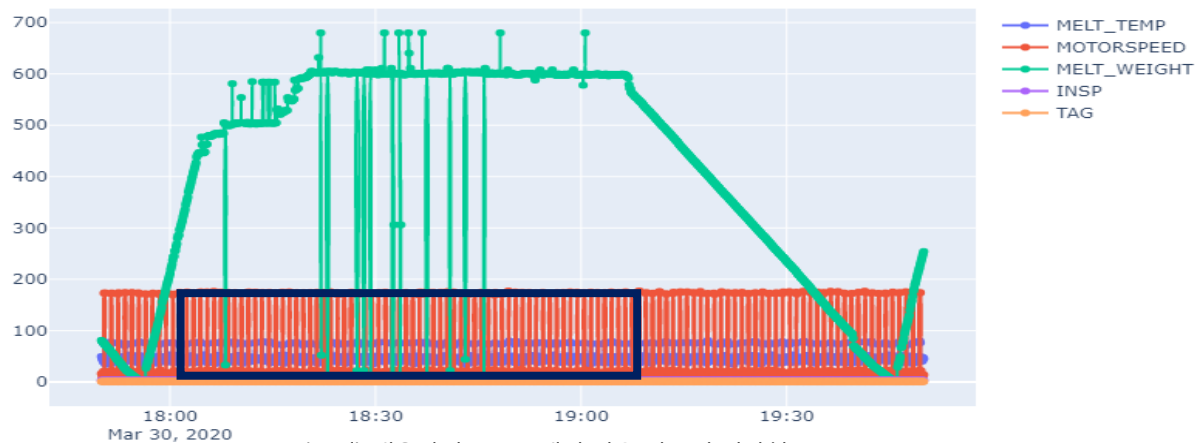
» 원재료가 토출되고 투입되는 상태에서 발생하는 이상치

» 예) 22:20 ~ 22:35 사이는 투입 과정에서의 계속 오류로 추정



## 내용량의 특수 경우(4) - 투입 ~ 토출 과정에서 주변 흐름과 맞지 않는 이상치

MELT\_WEIGHT 주기 찾기



» 내용량이 0 이외의 값(작은 값)들이 발생한 경우를 시각화 EDA

» 내용량의 투입 토출 과정에서의 **계측 오류 추정**

## [2차 전처리 요약]

### 1. 내용량(MELT\_WEIGHT) 전처리 (선형 보간 진행)

- (1) 내용량이 **680을 초과한 경우**
- (2) 내용량이 **0 부근에서 값이 튀는 값**
- (3) 내용량이 **주기에서 연속적이지 않는 값**

### 2. 계측 오류 여부, 투입 주기 파생 변수 추가

계측오류, 투입주기 추가

	NUM	MELT_TEMP	MOTORSPEED	MELT_WEIGHT	INSP	Measurement_error	MW_Period_Second	TAG
STD_DT								
2020-03-04 00:00:00	0	48.9	11.6	631.0	3.19	0	912	0
2020-03-04 00:00:06	1	43.3	7.8	609.0	3.19	0	912	0
2020-03-04 00:00:12	2	46.4	15.4	608.0	3.19	0	912	0

(그림) 최종 데이터프레임



1

## 이상 탐지 모델

### » 데이터 불균형

정상 비율 > 불량 비율

### » 비지도 학습을 활용한 이상 탐지

2

## 머신러닝 모델

### » Auto ML 용해물 품질 예측 모델 개발

### » Mixing 주기(1분) 기준 예측 진행

3

## 순환 신경망 모델

### » 시계열 딥러닝 모델(LSTM)

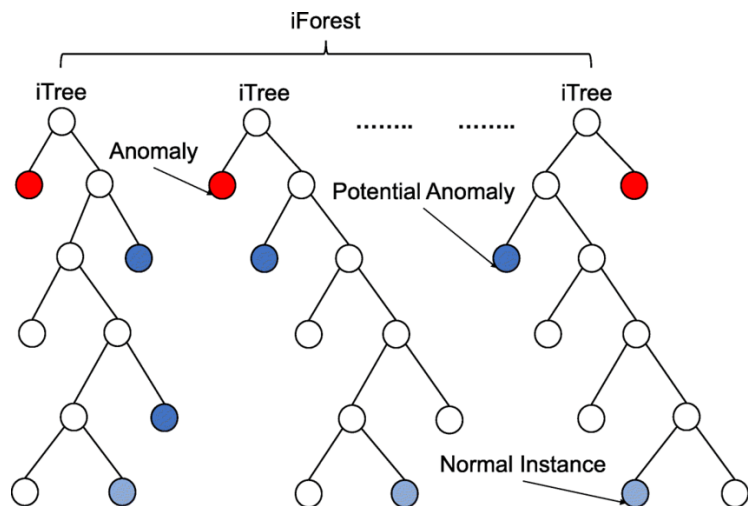
### » 용해물 품질 예측 과정에서 특징점을 잘 학습하기 위한 모델(CNN-LSTM)

### » 다양한 실험을 통한 최적 결과 도출

» 데이터 '불량 여부' 불균형 : 정상(OK) 78.8% vs 불량(NO) 21.2%

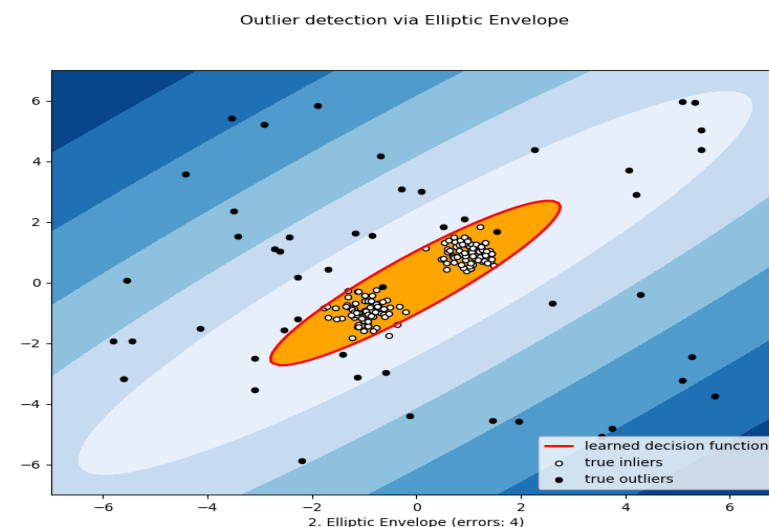
» 불량률의 비율이 적고 이를 이상치로 추정하여 분석 진행, 대표적 2가지 이상 탐지 모델 활용

## [Isolation Forest]



- Tree 기반 비지도 학습 이상치 탐지 모델
- Train 데이터에 Model 학습하여 Tree 기준을 생성 후, Tree에 벗어나는 값을 이상치로 판단

## [Elliptic Envelope]



- 데이터의 정규 분포 활용해, 데이터 분포를 타원으로 그린 후 벗어난 경우를 이상치로 판단하는 비지도 학습 모델
- LGBM 모델을 활용해 아래와 같은 추가 방법 진행

[1] Elliptic Envelope Model 이상치 라벨 생성 후, LGBM 학습 진행

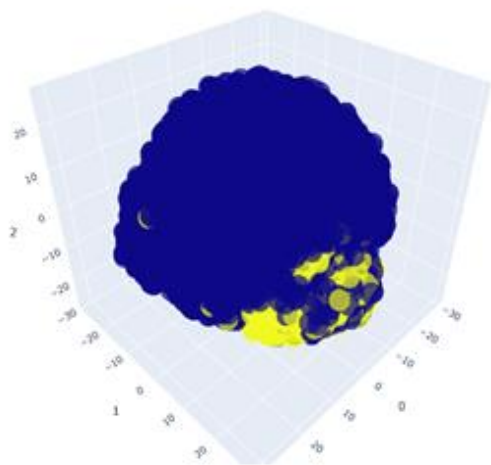
[2] Elliptic Envelope Model과 LGBM 예측 값에 대한 OR 양상불 진행



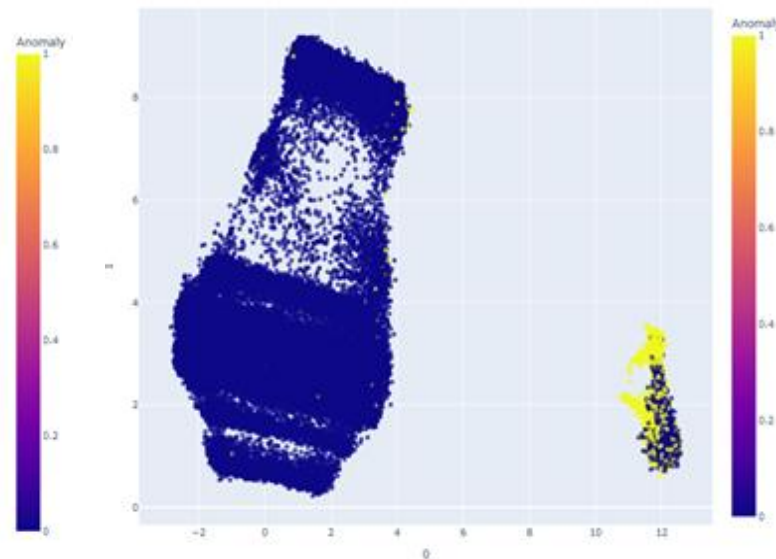
## [이상탐지 모델 결과]

	Accuracy	F1 score	Precision	Recall
Isolation Forest	49.13%	19.72%	17.78%	22.15%
Elliptic envelope	53.03%	29.82%	25.77%	35.37%
Elliptic envelope + LGBM	53.03%	29.82%	25.77%	35.37%
OR Ensemble	50.86%	29.64%	24.87%	36.69%

3d TSNE Plot for Outliers



uMAP Plot for Outliers



(그림 Isolation Forest Model를 U-map과 T-sne 활용한 시각화

- » 이상 탐지 모델의 성능(F1)이 매우 낮음
- » 이상치 탐지 모델을 시각화 해본 결과 이상치로 존재하는 값들은 존재하나, 실제 NG값과는 거리가 먼 것을 확인함

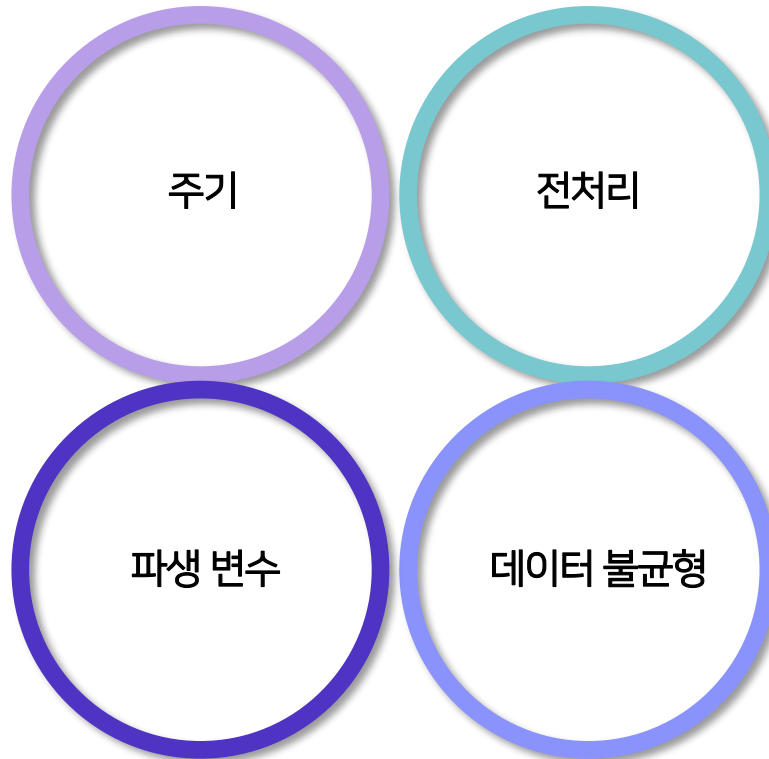
### [결론]

비지도 학습을 활용한 이상치 탐지 모델은 데이터는 불균형으로 이상치를 보이나, **데이터의 분포와 특징이 이상치를 이루고 있지 않아 성능이 낮음**



- » 데이터 측정 주기: 6초
- » 교반 주기: 1분
- » 교반 주기 기준 이상치 탐지 실험 진행

- » 파생변수(내용량 계측 오류 여부, 투입 주기) 추가 여부 성능 향상 확인 진행



- » 내용량의 계측 오류를 확인할 수 있음
- » 보간법 전처리 적합여부 판단을 위한 비교 진행
- » '정상'과 '불량' 데이터 불균형
- » 모델의 데이터 불균형 해결 기법(SMOTE) 활용 시 성능 비교



» 머신러닝 모델을 활용한 품질 예측 모델 개발 가능성 및 품질 예측에 주요 Feature 확인

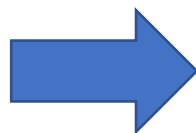
» 교반 주기 1분 기반 기존 6초 주기 데이터 → 1분 주기 데이터 변경 분석 진행

[1] 1분 내에서 각 6초 단위의 측정 정보를 Column 값으로 변경

[2] 품질 이상 여부는 1분 내에서 한 번의 “NG”이 발생할 경우 NG로 추정하여 전처리

	STD_DT	MELT_TEMP	MOTORSPEED	MELT_WEIGHT	INSP	Measurement_error	MW_Period_Second	TAG
NUM								
0	2020-03-04 00:00:00	48.9	11.6	631.0	3.19	0	912	0
1	2020-03-04 00:00:06	43.3	7.8	609.0	3.19	0	912	0
2	2020-03-04 00:00:12	46.4	15.4	608.0	3.19	0	912	0
3	2020-03-04 00:00:18	37.9	21.2	606.0	3.19	0	912	0
4	2020-03-04 00:00:24	79.8	173.6	604.0	3.21	0	912	0

[그림] 기존 데이터



	STD_DT	MELT_TEMP_00	MOTORSPEED_00	MELT_WEIGHT_00	INSP_00	Measurement_error_54	MW_Period_Second_54	TAG
0	2020-03-04 00:00:00	48.9	11.6	631.0	3.19	0.0	912.0	0
1	2020-03-04 00:01:00	50.7	12.8	598.0	3.19	0.0	912.0	0
2	2020-03-04 00:02:00	47.4	13.5	581.0	3.19	0.0	912.0	0
3	2020-03-04 00:03:00	43.7	12.9	587.0	3.19	0.0	912.0	0
4	2020-03-04 00:04:00	45.0	14.5	552.0	3.19	0.0	912.0	0

[그림] 머신러닝을 활용한 1분 주기 예측 데이터(변경)





## [Pycaret]

- » Auto ML Library, 다양한 모델에 대해 '전처리, 모델 생성 및 튜닝' 등의 작업을 쉽게 수행 및 개발
- » Pycaret classification, Regression 분석 진행
  - 성능 지표 기준 상위 3개의 (단일) Model를 선정
  - 선정 Model를 'Tunning/ Bagging / Boosting / Blending' 로 학습 및 성능 확인
  - 가장 좋은 성능을 보인 3개의 Model를 최종 선정하여 앙상블 진행

## [실험 방식]

### [1] Pycaret 생성된 모델과 별도의 Cat-boost 성능 비교 진행

- » Pycaret의 성능 신뢰 여부를 확인하기 위해서 진행

### [2] 주기(6초/1분)에 따른 성능 비교

- » 교반주기 1분과 데이터 측정 6초 주기 결과 비교

### [3] 파생변수 여부, 학습 종류에 따른 성능 비교

- » 내용량 계측 오류 여부, 투입 주기 파생변수 생성에 따른 성능 향상 여부 확인, Classification/Regression 학습 방법에 따른 성능 향상여부 확인



[머신러닝 모델 결과]

모델 내용		Pycaret(최종 앙상블 모델)		Catboost(성능 비교)	
		Accuracy	F1 score	Accuracy	F1 score
1분 주기	Classification	29.48%	38.29%	28.26%	38.96%
	Regression	30.47%	11.23%	28.58%	1.97%
	Classification + 파생 변수 추가	37.24%	36.41%	33.12%	41.97%
	Regression + 파생 변수 추가	36.85%	37.94%	35.40%	38.61%

[1분 주기]

» 학습(검증) 데이터의 경우 F1 score는 75~78%의 성능을 보였지만, 테스트 데이터의 성능은 45% 미만의 **저조한, 과적합 결과**

» 성능이 낮은 이유를 다음과 같이 판단

[1] 시계열 주기 정보를 Column으로 주는 것으로는 **시계열 데이터 특성 반영 어려움**

[2] 품질 주기 단위를 교반 주기(1분)로 맞추는 과정에서 발생하는 **정보 손실**



[머신러닝 모델 결과: 6초 단위]

모델 내용				Pycaret(최종 앙상블 모델)		Catboost(성능 비교)	
주기	회귀/분류	파생변수 여부	시계열 정보 (lag)	Accuracy	F1 score	Accuracy	F1 score
기존 주기 (6초 단위)	Classification	기본(x)	없음	69.08%	80.80%	69.64%	80.23%
	Regression	기본(x)	없음	72.32%	80.42%	66.92%	79.12%
	Classification	파생 변수(o)	없음	69.56%	81.18%	69.37%	81.04%
	Regression	파생 변수(o)	없음	69.43%	81.09%	69.20%	80.93%
	Classification	파생 변수(o)	1분	69.24%	80.95%	68.98%	80.75%
	Classification	파생 변수(o)	1 ~ 5분 전체	69.36%	20.04%	68.98%	20.20%

[6초 주기]

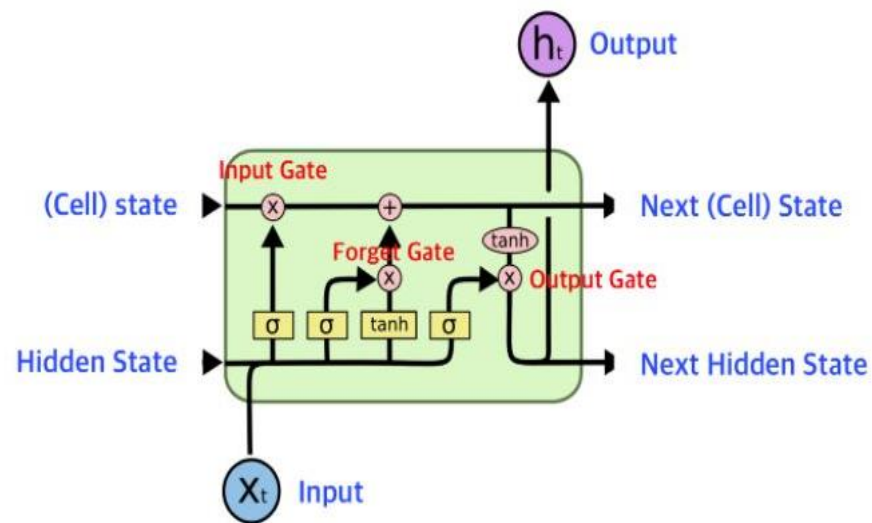
- » F1 score가 80% 높은 수치를 보이거나, 정확도는 상대적으로 낮은 것을 확인할 수 있음
- » 시계열적 요소를 반영하기 위해 Lag 정보(이전 시간의 정보)를 제공하였으나 성능 개선이 되지 않음
- » 머신 러닝 모델이 시계열적 요소를 반영을 잘하지 못한다고 판단

» 머신러닝 모델의 성능이 좋지 않아, **시계열적 요소를 고려하기 위해 순환 신경망 모델을 사용**

» LSTM, CNN-LSTM 활용

## [LSTM]

» 기존 RNN 장기 의존성 단점을 보완하고, **장/단기 기억** 가능한 신경망

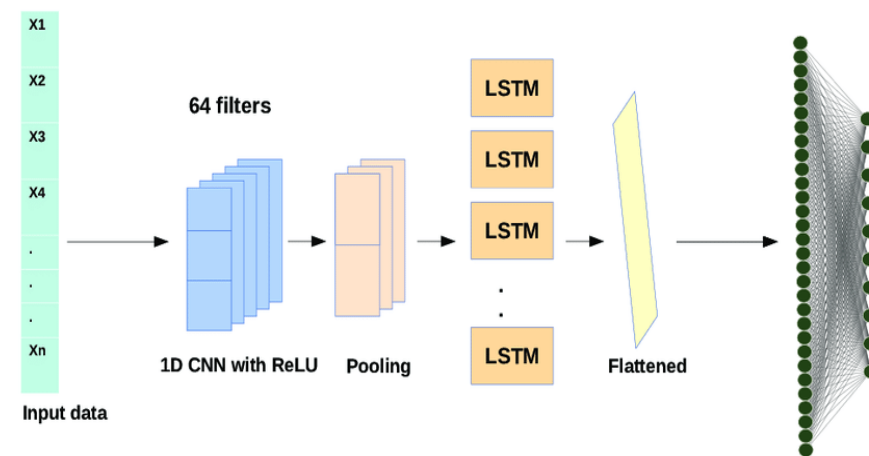


[그림] LSTM 구조

## [CNN LSTM]

» LSTM의 경우 시간의 맥락 파악의 초점이 되어, 특징을 잘 파악하지 못하는 한계

» 이미지 형태를 추출하는 CNN과 시퀀스의 시간적 맥락을 파악하는 LSTM 결합하여, **데이터의 시간적 특성 + 특징점까지 잘 파악** 한다는 장점



[그림] CNN-LSTM 구조



## [실험 내용]

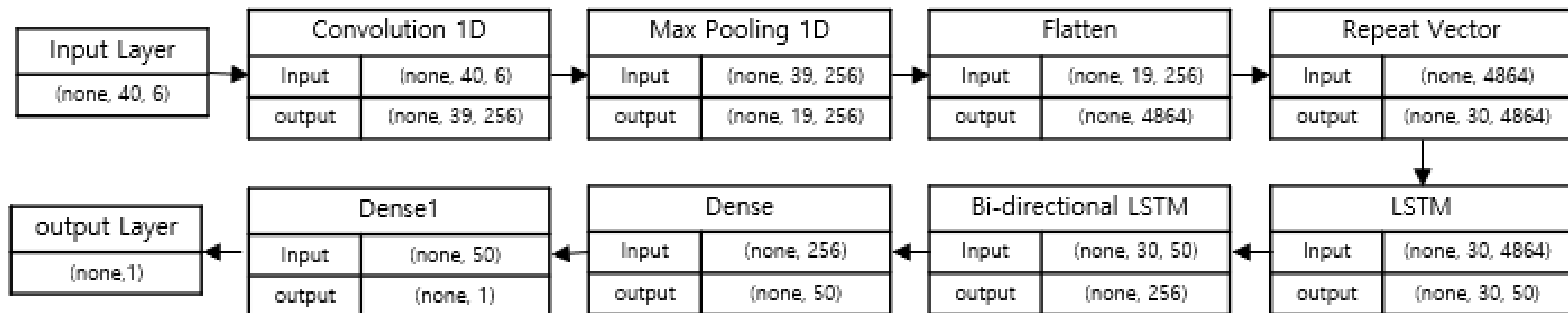
### [1] 조건별 성능실험

- 가. 시계열 자체 주기: 6초, 1분
- 나. 파생변수 추가 유무, 내용량 전처리 적용 여부
- 다. SMOTE 적용 여부: 데이터가 불균형한 점을 해소하기 위한 방안으로 적용 실험

### [2] CNN-LSTM, LSTM 성능 비교

- 가. LSTM의 경우 머신러닝 모델과 성능 차이가 크지 않아 추가적으로 CNN-LSTM 적용
- 나. Widow Size 조정: Model이 학습하는 과정에서 시계열 정보(Window size)를 분 단위로 1 ~ 10분 실험 진행

### [3] 최적의 CNN LSTM 기반 SHAP 활용 "용해물 품질 이상 탐지(진단) 영향성" 분석 진행



[그림] CNN LSTM 구조 시각화



### [실험 내용 1. 성능 실험]

#### » LSTM의 모델 활용하여 다양한 조건 적용

[1] 시계열 주기 : 6초, 1분

[2] 파생변수(계측오류, 투입주기) 추가 여부, 내용량 전처리 적용여부

[3] 데이터 불균형 해소하기 위한 SMOTE 적용 여부

#### » CNN LSTM의 경우 파생변수 추가/전처리 적용/ Window size =20(2분)으로 진행

		Accuracy	F1 score	Precision	Recall
Base(기본 변수)	LSTM	71.87%	83.24%	93.13%	75.25%
파생 변수 & 내용량 전처리	LSTM	76.36%	86.41%	92.45%	80.63%
	LSTM (주기: 1분)	41.84%	31.31%	33.34%	29.51%
	LSTM(SMOTE)	41.75%	32.90%	34.08%	31.79%
	CNN-LSTM	81.18%	89.56%	92.30%	86.89%



## [실험 내용 2. CNN LSTM 모델 최적의 Window size]

» CNN LSTM 모델에서 용해물 품질 예측 과정에서의 최적의 Window size를 찾기 위한 실험

» 기본 정보: 파생변수 추가 및 내용량 전처리(보간법) / SMOTE 미적용 / 주기는 원본 주기(6초) 진행

	Accuracy	F1 score	Precision	Recall
1분(window = 10)	77.20%	86.94%	92.90%	81.69%
2분(window = 20)	81.18%	89.56%	92.30%	86.89%
3분(window = 30)	69.69%	81.88%	92.02%	73.75%
4분(window = 40)	71.36%	82.98%	92.58%	75.19%
<b>5분(window = 50)</b>	<b>90.35%</b>	<b>94.94%</b>	<b>92.80%</b>	<b>97.15%</b>
6분(window = 60)	76.49%	86.75%	92.50%	80.54%
10분(window = 100)	72.96%	84.20%	92.03%	77.59%

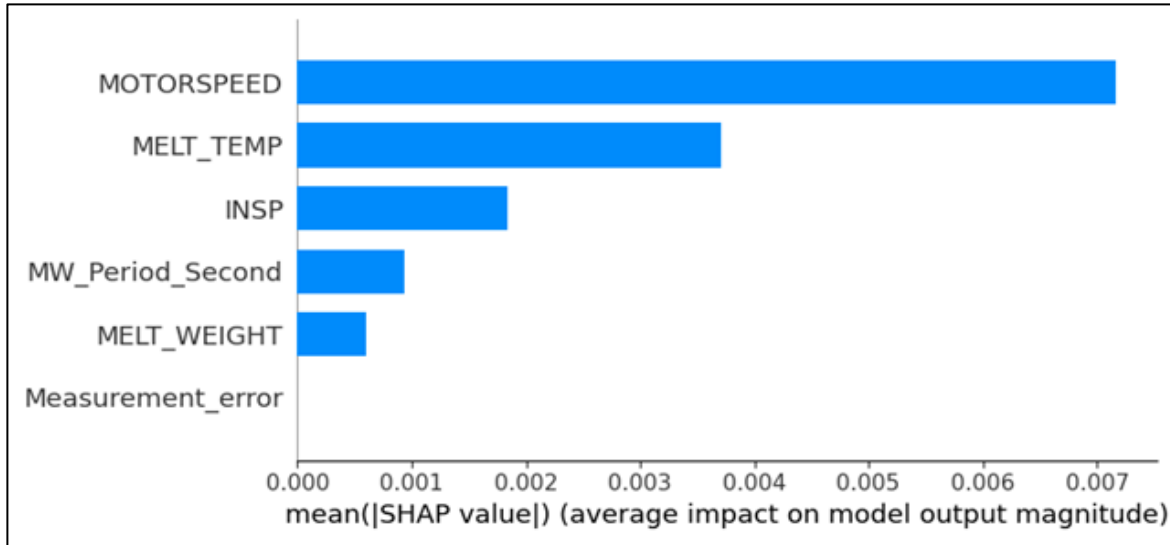
» Window size의 크기가 50일 경우 전체 성능이 가장 좋음

(Window size를 10의 단위, 1분으로 한 것은 앞선 데이터 분석 과정에서 교반 주기가 1분이라는 점을 착안)



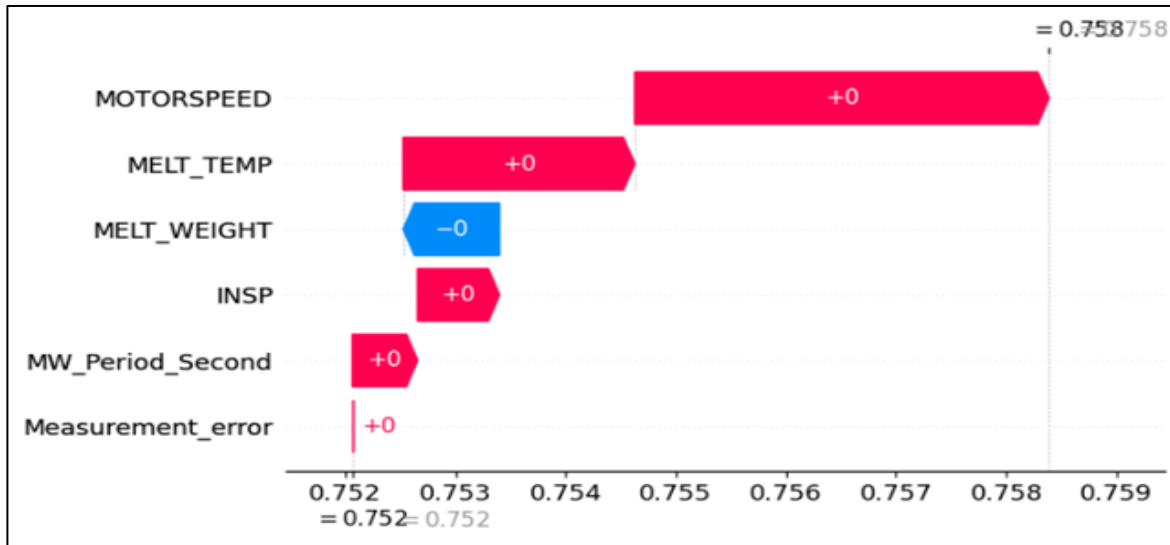
	이상치 탐지		머신러닝	순환 신경망 모델	
	Isolation Forest	Elliptic Envelope + LGBM	Auto ML	LSTM	CNN_LSTM
모델 설명	- 비지도 학습을 통한 이상탐지 방법론		- Pycaret 모듈을 활용한 ML 방법 - 분류/회귀 모델 병행 진행	- RNN 계열 모델 (시계열적 특성 반영)	- RNN 계열 모델 (시계열적 특성 반영) - 기존 LSTM에서 CNN을 활용해 특징까지 용이하게 파악
성능 (결과)	- F1 score가 20 ~ 30%로 낮으며 지나친 과적합		- 6초 단위: F1 score가 약 60 ~ 70% - 1분 단위: F1 score가 약 10 ~ 40%	- F1 Score가 약 80 ~ 83% 나옴	- F1 Score가 약 85 ~ 95% 나옴
특징/ 활용 가능성	- 이상치로 존재하는 값들을 확인하나 실제 이상치(불량)과는 거리가 멀 - 데이터의 분포와 특징이 이상치를 이루고 있지 않아 성능이 낮음 - <b>용해물 품질 예측에 적합하지 않는 방법</b>		- 다른 도메인 주기(1분) 기준으로 예측하면 안됨 - 과적합이 심하여 F1 score가 낮아 <b>용해물 품질 예측에 적합하지 않음</b>	- Precision은 높아, <b>실시간 품질 예측은 적합</b> - 정확도, F1 score가 낮아 <b>다른 Task 적용하기 어려움(최적화 등)</b>	- 전반적인 성능이 좋아 <b>용해물 품질 관련 Task에 매우 적합</b>





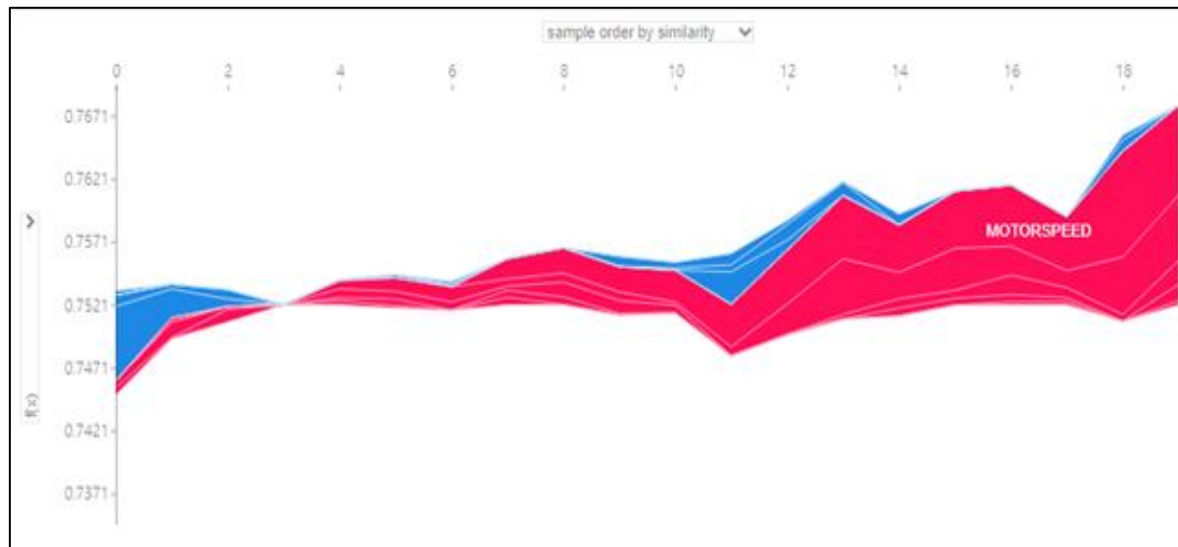
(그림) SHAP Feature Importance 시각화

- » SHAP을 활용하여 해당 예측 모델을 분석함
- » 용해물 품질 이상 탐지 과정에서 가장 중요한 요소는 '교반 속도와 용해물 온도'이다.
- » 이는 용해물 공정 과정에 직접적인 영향을 주기 때문이라 추정
- » 이외에도 추가적인 파생변수 일부도 영향을 미치는 것을 확인



(그림) 데이터 객체 시각화

- » 각 변수별로 해당 값에 어떻게 영향을 미치는지를 파악 할 수 있음
- » 붉은색은 양의 방향으로, 푸른색은 음의 방향으로 영향을 미침
- » 해당 그래프는 하나의 결과값에 대해 변수들이 어떻게 영향을 미쳤는지 나타냄
- » MELT\_WEIGHT는 음의 방향으로 나머지는 양의 방향으로 영향을 미침



(그림) SHAP Force Plot 시각화

- » 각 독립 변수별로 전체 예측 값에 영향력 정도를 시각화한 결과
- » MOTORSPEED가 양의 영향력을 강하게 미치고 있음을 알 수 있음
- » 해당 분석들을 통해 공정 최적화와 같은 다양한 분석에 사용 가능
- » 용해물 품질 공정 이외에도 다른 제조 데이터에서도 본 분석을 동일하게 활용하여서 각 독립 변수별로 영향력 확인 및 활용 가능

## [결론]

- » 앞선 CNN-LSTM SHAP 시각화를 통해서 모터의 교반 속도와 용해물의 용해 온도가 매우 중요한 영향임을 확인
- » 이를 통해서 용해물 품질 예측 과정에서 교반 속도와 용해 온도의 특이점이 발생할 경우 사전 경보를 통해서 품질 유지 방향으로 발전 가능
- » 교반 속도와 용해 온도에 따른 조건부 모델 생성을 통한 고도화 가능



## 분석모델의 특징

모델이 가벼움

짧은 training, inference 시간

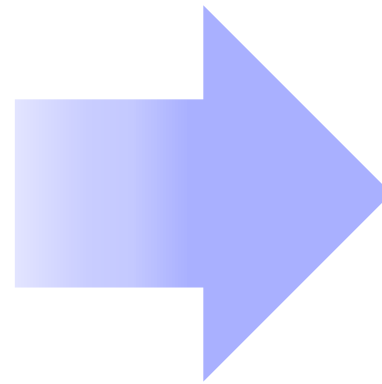
간단한 전처리

적은 계산량 비용

실시간 예측 가능

하드웨어적으로 가벼움  
(학습, 예측 둘다 하드웨어 성능 중요 X)

각 피쳐들의 결과값에 미치는 영향성 파악 가능

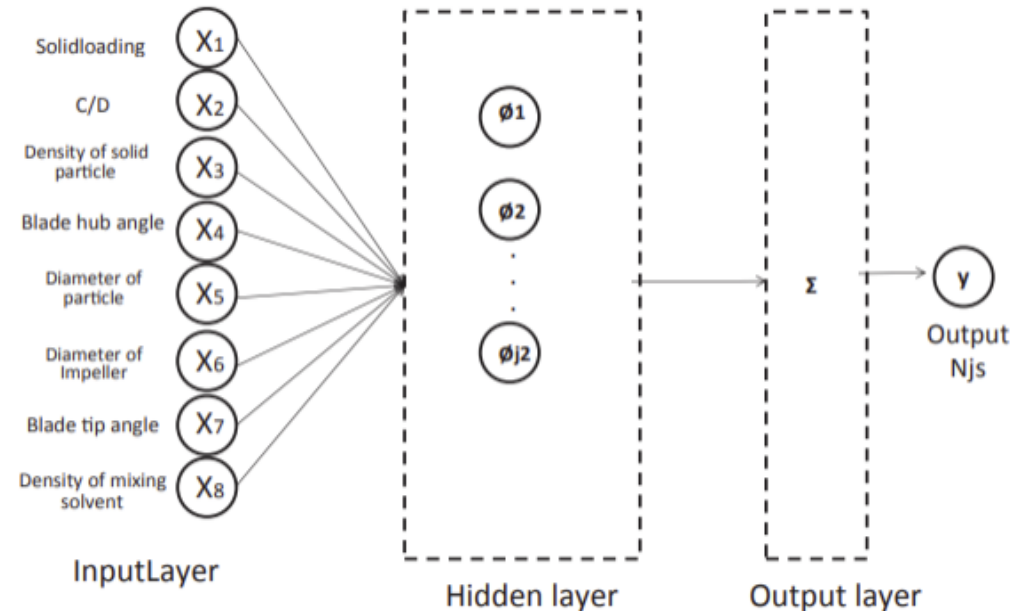


타업종  
적용가능성

업종별  
공정 최적화

## [개선방향]

- ▶ 교반 탱크의 용해물 품질 예측을 위해 CNN\_LSTM 모델을 활용하였고 어떤 Feature가 큰 영향력을 가지는지는 파악하였으나 품질이상을 발생시키는 주요 원인 인자와 범주를 파악하기에는 한계가 있음
- ▶ 선행연구에 따르면 solid-liquid mixing system의 교반 속도를 예측하기 위해 고체 원료 투입량, 탱크 직경, 밀도, 임펠러 직경, 블레이드 수 등을 고려하는 것을 알 수 있음
- ▶ 본 연구의 개선사항으로서 유체역학 거동을 유추해볼 수 있는 밀도, 부피 등을 고려한 레이놀즈 수 등을 넣으면 정확도가 높아질 것으로 기대
- ▶ 또는, 공정설비 내부 온도 등도 용해물이 반응하는데 영향을 미칠 것으로 판단되어 주변 기상정보 활용도 고려됨



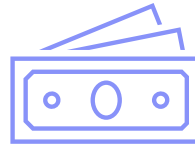


## 기술적 효과



- 현장작업자의 경험과 노하우 등 암묵지에 의존하여 대처  
→ 품질예측 모델을 기반 제품 품질 향상
- 용해물을 생산하는 중소제조기업 뿐만 아니라 교반 탱크 제작중소기업에도 **품질 보증할 수 있는 제어시스템**을 개발할 수 있을 것으로 기대

## 경제적 효과



- 경제적 여건이 되지 않은 중소제조기업의 연구개발비용 개선
- 용해물의 품질을 향상시켜 매출증대
- 식품, 제약, 화장품, 석유공업, 발효 공업, 섬유공업, 고무공업, 수처리 공업 등의 다양한 활용처가 존재  
→ 중소제조기업 경제적 파급효과 큼

## 환경적 효과



- 품질증대로 인해 불량품의 비율이 줄어들어 폐기물을 줄일 수 있을 것으로 기대
- 물품생산을 위해 소비되는 전력 감소  
→ 탄소저감에도 기여

감사합니다