K-인공지능 제조데이터 분석 경진대회 보고서

프로젝트명	이상탐지 기반 공정 품질 예측 모델 개발	
팀명	Next AI LAB	
내용요약	1. 분석 배경	용해 공정에서의 균일함이 보장되지 않아 품질 관리에 어려움을 겪고 있다.
	2. 분석목표	용해 공정 중의 문제점을 해결하기 위해, PLC를 통해 수집된 데이터셋 항목을 중심으로 분석모델을 개발하여 해당 과정에서 완제품 품질을 예측할 수 있도록 하였다.
	3. 제조데이터 소개 및 전처리	제조데이터의 각 칼럼을 분석하고 특성에 맞춰서 전처리를 진행하였다.
	4. 분석 모델	데이터 특성에 맞는 모델 설계 및 학습 진행
	5. 모델 평가 및 결과해석	평가지표를 이용한 분석 모델 학습 성능 해석
	6. 파급효과	식품 제조현장 및 타 분야에서의 분석모델 적용 가능성

상기 본인(팀)은 위의 내용과 같이 K-인공지능 제조데이터 분석 경진대회 결과 보고서를 제출합니다.

2022 년 11 월 9 일

팀장 : 김예진 (서명)
팀원 : 김선중 (서명)
팀원 : 양희성 (서명)

한국과학기술원장 귀중

□ 문제정의

1. 분석 배경

1) 공정(설비) 개요

- 공정(설비) 정의 및 특징
  우리가 다루는 용해 탱크 데이터셋은 식품 제조업, 그중에서도 분말 유크림 제조 과정 중 전처리 단계인 용해 공정을 바탕으로 수집된 데이터이다. 이번 용해 공정은 분무건조(Spray Drying) 생산설비에서 분말 원재료를 액상 원재료에 녹이는 공정으로 원료 전처리 작업의 가장 첫 번째 단계이다.

  최종 생산물의 품질을 보장하기 위해, 용해 공정에서 균일함이 보장된 혼합물이 생산되어야 한다. 용해 공정 혼합물이 잘 용해가 이루어졌는지 여부는 용질과 용매의 상대적 용량, 용해 온도, 물리적 힘 등에 영향을 받는다.

  원료 용해 과정에서 크게 온도, 교반속도가 미치는 영향을 살펴볼 수 있다. 온도가 지나치게 낮고, 교반속도가 너무 느리면 용해가 잘 이루어지지 않을 수 있다. 반대로 온도가 지나치게 높고, 교반속도가 과도하면 용해 중의 화학반응 등의 작용으로 원하는 품질에 미치지 않을 수 있으며, 기계 설비에 영향을 줄 수도 있다. 또한 온도, 교반속도 외 다양한 원인이 용해 과정에 영향을 주는 원인이 되기도 한다.

2) 이슈사항(Pain Point)

원재료 전처리의 첫 번째 단계인 용해 공정에서 용해가 잘 이루어지지 않으면, 이후의 공정들과 최종 생산물의 품질에 큰 영향을 줄 수밖에 없다.

특히 분말 유크림의 작업 과정은 분무건조 공법을 이용하는데, 용해 공정을 거쳐 후공정에서 다시 건조과정을 통해 분말을 생산하게 된다. 용해 공정에서 문제가 생기면 분말의 품질이 보장되기 어려워진다. 따라서 용해 공정에서 용해 품질을 유지하는 것은 중요한 문제이다.

특히 원료를 여러 종류로 투입하거나, 대량의 제품을 생산할 때는 원료를 시차를 두고 순차적으로 투입하게 된다. 투입할 때마다 내용물의 온도와 점도, 탱크 교반속도 등에 변화가 발생하게 된다. 이때 설비 운영 값을 계속해서 조정해줄 필요가 있는데, 정해진 가이드라인이 없어 특정 작업자의

경험과 노하우에 의존하고, 인력 공백이 발생하면 품질 유지에 어려움을 겪는 경우가 많다.

이 밖에도 계절/날씨 등의 변화로 습도와 온도에 의해 원료나 용해 탱크에 영향을 주는 등 용해 공정에 영향을 미치는 요소들은 이 밖에도 많이 존재하지만, 모두 제어할 수 없다는 문제점이 있다.

3) 해결하고자 하는 문제

용해 공정이 진행되는 과정에서 용해 상태를 측정할 수 있는 절대적인 방법이 아직까지 부재하다는 것이 가장 큰 문제이다.

공정 중간중간에 품질검사를 진행하거나 작업자가 직접 확인하는 방식으로 용해 상태를 확인하여 작업을 해야 한다. 그 밖에, 공정 중간에 품질을 확인하지 못하고 후공정에 가서야 확인하여 빠른 품질확인이 어려운 경우도 많다.

공정 진행 중에 품질확인을 위해 특정 작업자에 의존하는 등 객관적인 품질 측정이 어렵거나, 공정 진행 중에 전혀 품질확인을 하지 못하고 후공정에 가서야 품질확인이 이루어지는 등 현재와 같은 품질확인 시스템으로는 용해 공정 생산물의 품질이 보장되지 못한다는 문제가 있다.

따라서 공정 중 실시간으로 변화하는 온도, 교반속도, 내용량 등의 설비운영값과 그 결괏값을 모델링하여 이를 바탕으로 생산품 질을 예측하는 시스템을 개발할 필요가 있다.

2. 분석 목표

1) 분석 목표

앞서 살펴보았던 제조업 공정 과정, 특히 용해 과정 중의 문제점을 해결하기 위해, PLC를 통해 수집된 데이터셋 항목을 중심으로 분석모델을 개발하여 해당 과정에서 완제품 품질을 예측할 수 있도록 하였다.

2) 제조데이터 분석 목적

주어진 문제를 제조업 분야의 이상 탐지(Anomaly Detection) 문제로 판단하여, 라벨 값으로 주어진 불량여부(TAG) 컬럼을 정답으로 두고 지도 학습을 시행하였다.

공정 과정에서 다양한 원인이 품질에 영향을 미친다는 특성이 있다. 또한 데이터셋에 나와 있는 변수들은 특히 영향력이 높은 변수인 만큼, 주어진 데이터셋에 나와 있는 주요 변수들을 모두 활용하여 품질을 예측하는 모델을 만들고자 하였다.

제조업 산업 특성상 라벨 컬럼에서 불량(NG)의 비율이 적은 불균형 데이터가 주어졌기 때문에, 학습을 균형 있게 진행하고자 K-nearest neighbors 기반의 SMOTE 기법을 이용하여 균등한 학습을 통해 모델을 구축하였다.

□ 제조데이터 정의 및 처리과정

3. 제조데이터 소개

1) 제조데이터 수집방법

주어진 데이터셋은 분무건조공법을 이용한 분말유크림 제조 분야에서 수집된 데이터로, 용해혼합 공정에서 PLC 및 DBMS를 통해 얻어진 것이다. 사이클 주기를 설정하여, 6초마다 데이터가 수집되도록 하였다. 총 수집기간은 2020년 3월 4일부터 2020년 4월 30일까지로 약 2개월에 해당한다.

2) 제조데이터 유형, 구조

- 데이터 속성정의 표

컬럼 명	설명	데이터 타입
STD_DT	날짜 & 시간	object
NUM	인덱스	int64
MELT_TEMP	용해 온도	int64
MOTORSPEED	용해 교반속도	int64
MELT_WEIGHT	용해탱크 내용량(중량)	int64
INSP	생산품의 수분함유량(%)	float64
TAG	불량여부	object

데이터셋의 컬럼은 데이터 수집일시(STD_DT), 데이터 인덱스(NUM), 용해온도(MELT_TEMP), 용해 교반속도(MOTORSPEED), 용해 탱크 내용량(MELT_WEIGHT), 수분함유량(INSP), 불량 여부(TAG) 등 총 7개이며, 835,200개의 관측치로 이루어져 있다. 수치를 살펴보면 데이터 수집일시, 인덱스, 불량 여부를 제외하고는 모두 연속형 수치를 띤다.

교반속도(MOTORSPEED)가 0인 경우는 공정 완료, 원료 추가 투입, 설비 이상, 작업자 휴식 등의 다양한 이유로 설비를 중지/정지한 경우이며, 내용량(MELT_WEIGHT)이 0인 경우는 용해 탱크에 아직 원료가 투입되기 전이거나, 공정 완료 후 용액이 다음 공정으로 넘어가서 탱크가 빈 경우이다. 불량 여부(TAG) 값 OK는 '양품', NG는 '불량'을 의미한다.

용해 온도와 교반속도 데이터는 소수점 1자리가 생략되어 있어 값 nnn은 실제로 nn.n을 의미한다.

다른 변수에 영향을 받지 않는 용해 온도, 교반속도, 내용량, 수분함유량 등을 독립변수로 두고, 각 변수의 전체적인 영향을 확인하기 위해 불량 여부를 종속변수로 두었다.

- 전체 데이터의 통계수치 확인

	NUM	MELT_TEMP	MOTORSPEED	MELT_WEIGHT	INSP
count	835200.000000	835200.000000	835200.000000	835200.000000	835200.000000
mean	417599.500000	509.200623	459.782865	582.962125	3.194853
std	241101.616751	128.277519	639.436413	1217.604433	0.011822
min	0.000000	308.000000	0.000000	0.000000	3.170000
25%	208799.750000	430.000000	119.000000	186.000000	3.190000
50%	417599.500000	469.000000	168.000000	383.000000	3.190000
75%	626399.250000	502.000000	218.000000	583.000000	3.200000
max	835199.000000	832.000000	1804.000000	55252.000000	3.230000

그림 4 : df의 통계수치

- 결측치 확인

  전체 데이터의 결측치를 확인한 결과 결측치는 확인되지 않았다.

```
[ ]  # 결측치 확인
     df.isnull().sum()
```

```
STD_DT          0
NUM             0
MELT_TEMP       0
MOTORSPEED      0
MELT_WEIGHT     0
INSP            0
TAG             0
dtype: int64
```

그림 2 : 결측치 확인

3) 데이터 전처리 및 시각화

- 상관분석
  - 라벨 값인 불량여부(TAG)와의 상관관계를 살펴보았다. (그림 3)
  - 추후 모델 학습을 위해 라벨 컬럼인 TAG의 OK, NG 값을 sklearn의 labelencoder를 사용하여 각각 1과 0으로 변환하였다.
  - 데이터수집일시(STD_DT)는 월, 일, 시, 분, 초로 분리하여 각각의 항목에 대해 상관성을 보고자 했다.
  - 온도(MELT-TEMP), 수분함유량(INSP), 교반속도(MOTORSPEED) 순으로

상관성이 높은 것으로 나타났으며, 수집일시의 월(MONTH), 일(DAY)도 상관성이 있는 것으로 보인다.

- 주어진 요소 이외에 다양한 원인들이 품질에 영향을 주는데도 불구하고 품질 예측에 반영이 어렵다. 또한 단일 요소와 라벨 값 간의 상관성이 떨어지는 것처럼 보여도 여러 요소가 복합적으로 작용하여 품질에 영향을 줄 수 있다. 따라서 미미한 상관성이 있어도 해당 데이터셋의 컬럼들은 모델 학습에 모두 반영하기로 하였다.

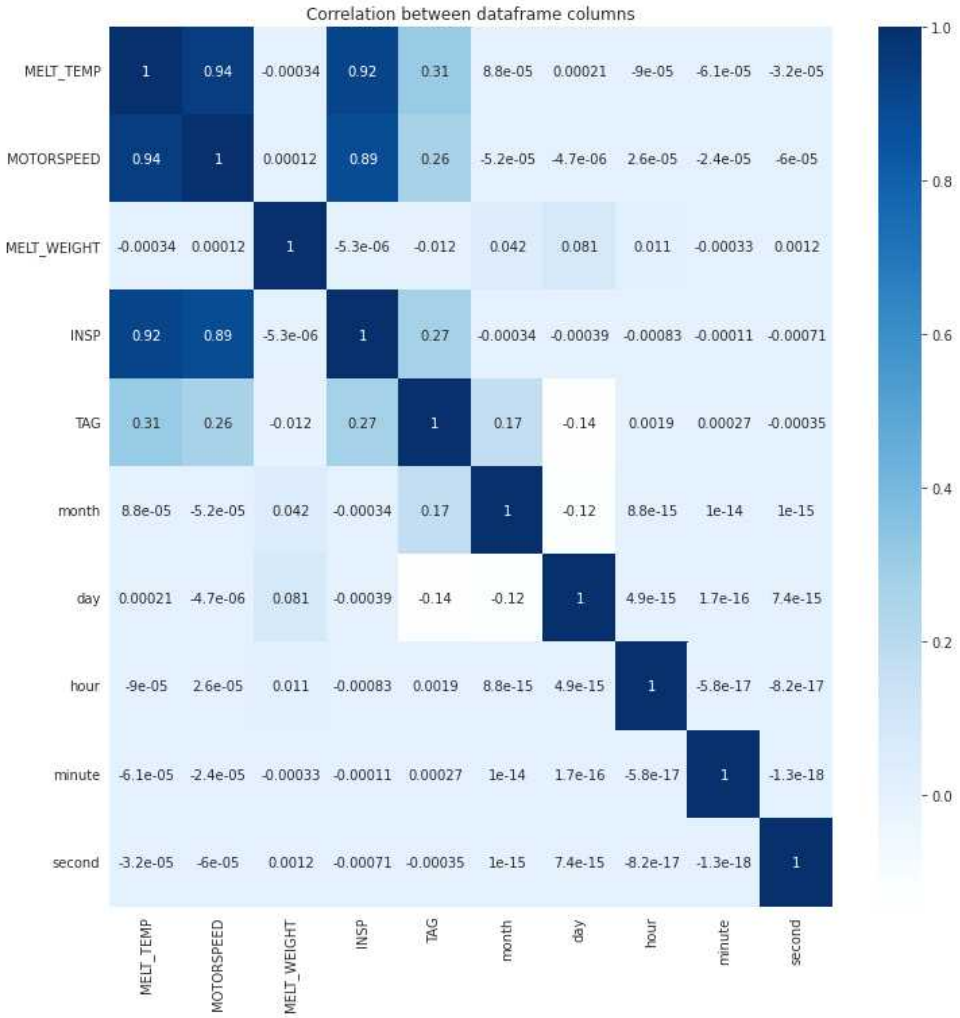


그림 3 : 컬럼 간의 상관관계(히트맵)

- 컬럼별 데이터 전처리 및 시각화
  - 데이터 수집일시(STD_DT)
    - ▸ 날짜/시간이라는 속성 자체가 불량 여부에 직접적인 영향을 주는 것은 아니다. 그러나 해당 데이터셋에 나와 있는 독립변수들 외에도 태그 값(불량 여부)에 영향을 주는 요인들은 산재한다.
    - ▸ 개별 날짜에는 여러 변수가 혼재하여 불량 여부에 영향을 끼친다. 따라서 날

짜 컬럼이 의미하는 것이 단순한 날짜 값이 아니라 주어진 컬럼 이외에 불량 여부에 영향을 줄 수 있는 여러 독립 변수들의 대푯값으로 판단하여 반영하기로 하고, 전처리를 진행하였다.

▸ 날짜 및 시간 데이터를 datetime 형식으로 바꿔주었다.

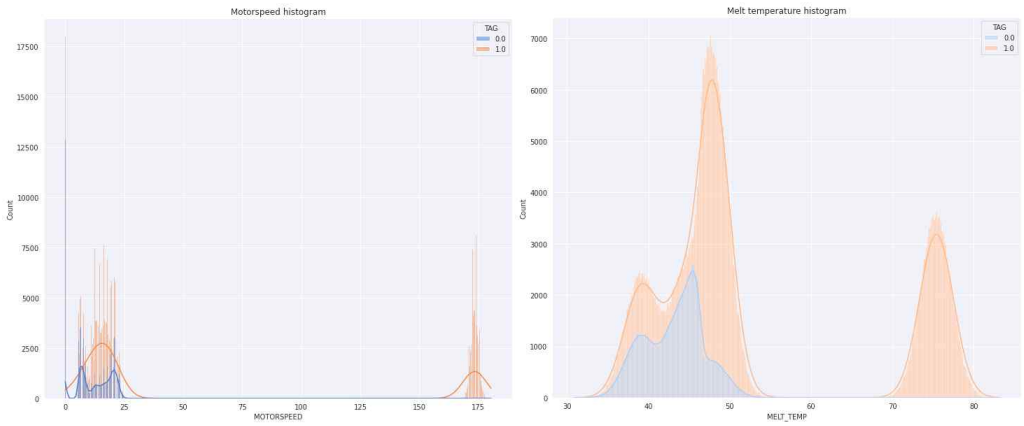
▸ 또한 날짜 및 시간을 그대로 사용할 예정이기에 월, 일, 시, 분, 초 컬럼으로 나누어 준 후, 기존의 STD_DT 컬럼은 삭제하였다.

	MELT_TEMP	MOTORSPEED	MELT_WEIGHT	INSP	TAG	month	day	hour	minute	second
0	48.9	11.6	631	3.19	1.0	3	4	0	0	0
1	43.3	7.8	609	3.19	1.0	3	4	0	0	6
2	46.4	15.4	608	3.19	1.0	3	4	0	0	12
3	37.9	21.2	606	3.19	1.0	3	4	0	0	18
4	79.8	173.6	604	3.21	1.0	3	4	0	0	24

- 용해온도 및 교반속도

▸ 용해온도(MELT_TEMP)와 교반속도(MOTORSPEED)는 원래 nn.n의 값이지만 nnn형태로 나타나 있기에, 이를 다시 원래 값으로 되돌리는 작업을 진행한다.

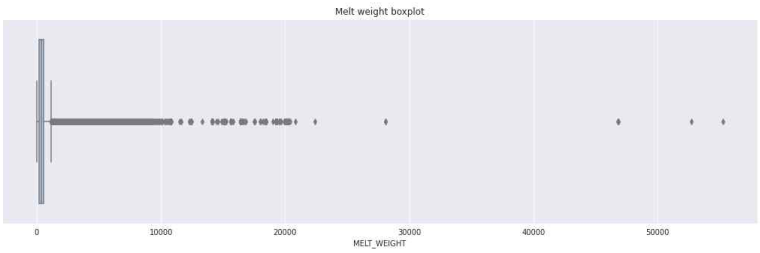
▸ 각 컬럼의 값들에 / 10의 연산을 해주어 전처리를 진행하였다.



- 용해탱크 내용량

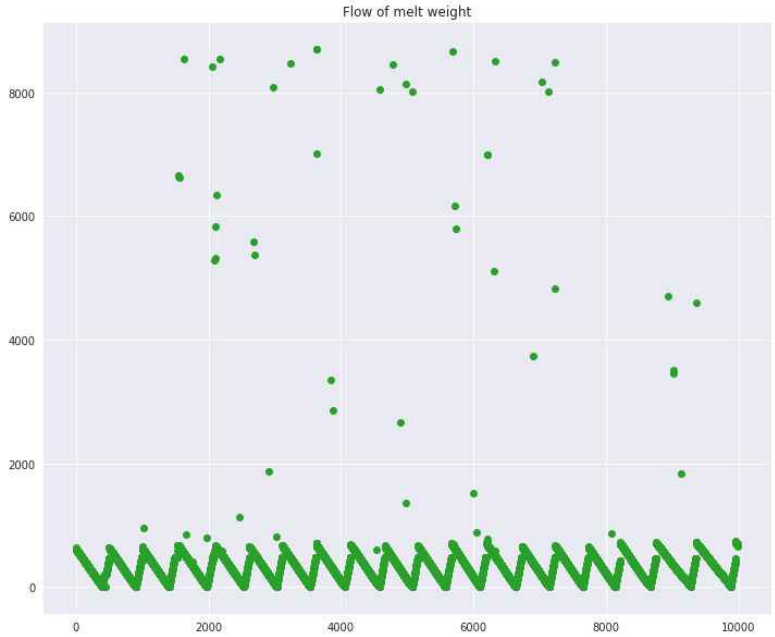
▸ 내용량(MELT_WEIGHT)의 경우 최솟값이 0, 최댓값이 55252인데 평균(mean)은 약 582에 불과하다. 값의 분포가 고르지 못하다고 판단하여, 따라서 이상치를 먼저 확인하였다.

▸ boxplot을 통해 확인한 결과 이상치의 양이 상당히 많았다.



▸ scatter를 통해 전체 분포를 확인한 결과 일정한 사이클을 그리며 값이 변화하

는 것을 확인할 수 있었다.

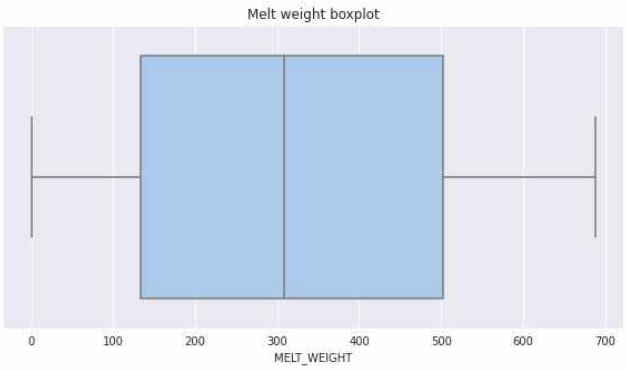


- 위의 사이클을 벗어나는 각 이상치를 확인해보니 그 값이 근방에 비해 10배(또는 100배)의 값을 가지는 것을 확인하였다. (예시 : 655, 654, 6656, 654, 655, 기계의 오작동으로 656이 6656으로 나타난 것으로 보인다.)
- 내용량은 대부분 세 자리 숫자이고 최빈값이 688이라는 점을 고려하여, 688보다 높은 수치를 이상치로 판단하여 세 자릿수로 변환하여 전처리를 진행하였다. 내용량 값의 자릿수에 따라 다른 방식으로 변환하였다.

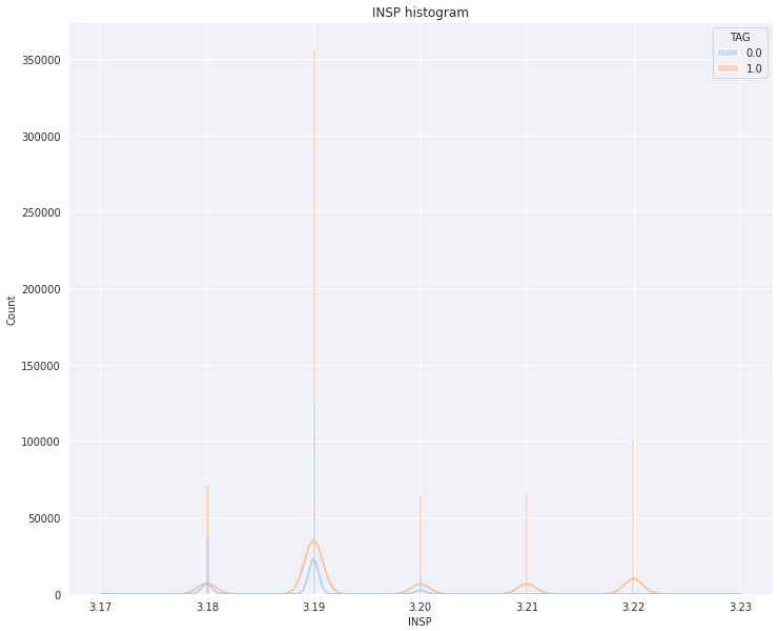
순번	내용량	전처리 작업	비고
1	$1000 \leq$ 내용량 < 55252	100으로 나눈다.	다섯 자릿 수를 세 자릿 수로 변환
2	$100 \leq$ 내용량 < 7000	10으로 나눈다.	네 자릿 수를 세 자릿 수로 변환
3	$7000 \leq$ 내용량 < 10000	1000으로 나눈 나머지를 구한다.	네 자릿 수를 세 자릿 수로 변환
4	$688 \leq$ 내용량 < 1000	10으로 나눈다.	세 자릿 수를 두 자릿 수로 변환

* 내용량의 최댓값은 55252이다.

- boxplot으로 재시각화하여 확인한 결과 이상치가 잘 처리되었음을 확인할 수 있다.



- 수분함유량(INSP)
  - ▸수분함유량은 제품 품질검사를 수행하여 획득 가능한 값으로, 3.17 ~ 3.23 사이의 값을 보인다.
  - ▸연속적인 값이 아닌 이산적인 값의 분포를 보이기 때문에, 범주형 데이터로 판단하여 원-핫-인코딩(one-hot-encoding)을 통해 가공하였다.



- NG 라벨 값
  - ▸NG값들의 분포를 확인하기 위해 그래프를 그려보았다. NG값들이 3월 중순부터 4월 중순까지 분포되어있는 것을 확인할 수 있었다. (그림 4)

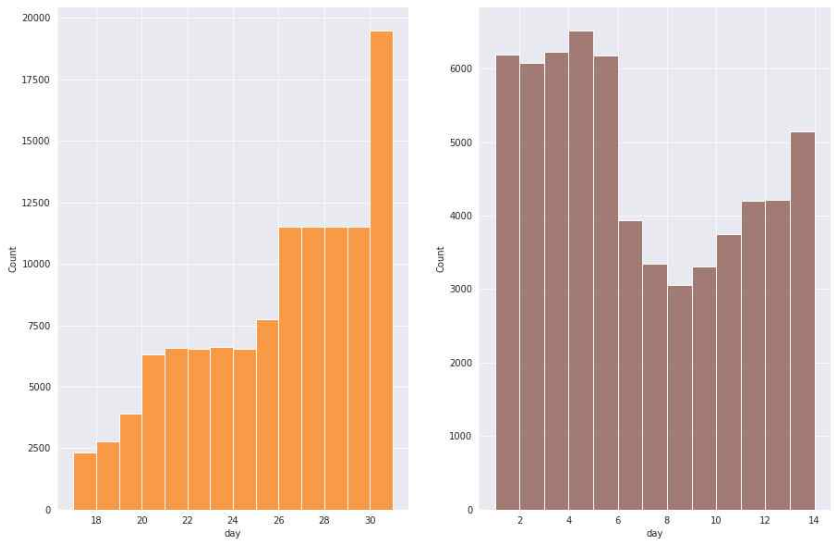


그림 4 : 월별 NG 분포 (왼쪽 : 3월, 오른쪽 : 4월)

- 전체 데이터 시각화

- 데이터 값들의 조정을 마치고 피쳐 전체를 한눈에 시각화하여 살펴보았다. (그림 5, 6)

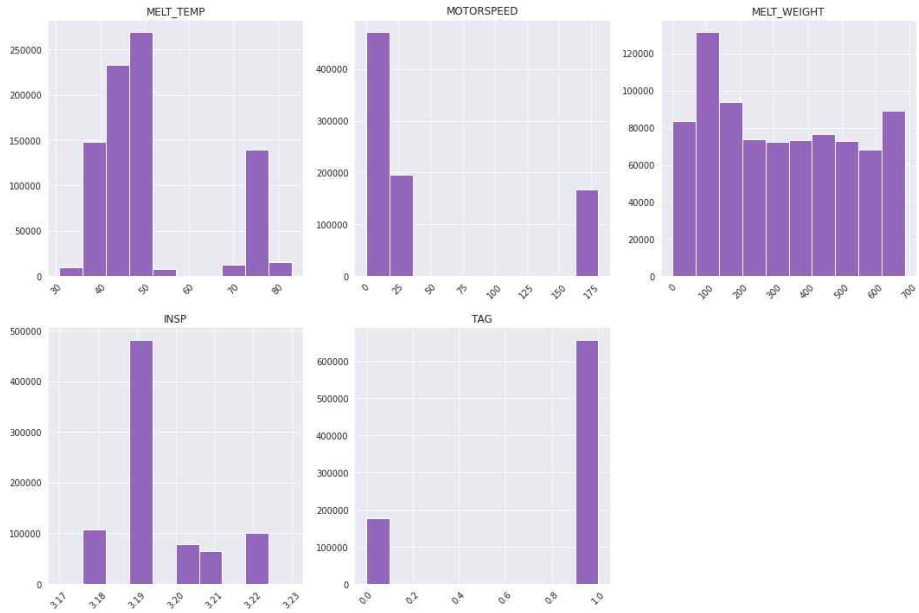


그림 5 : 각 피쳐 별 히스토그램 (전체)

그림 6 : 각 피쳐별 산점도

- 피쳐 별 데이터의 흐름을 100개 단위로 살펴보았다. 앞서 살펴보았던 내용량

(MELT_WEIGHT)과 함께 다른 요소들(MELT_TEMP, MOTORSPEED, INSP)도 그래프의 모양이 비슷한 주기를 가지며 반복되는 것을 알 수 있다. (그림 7)

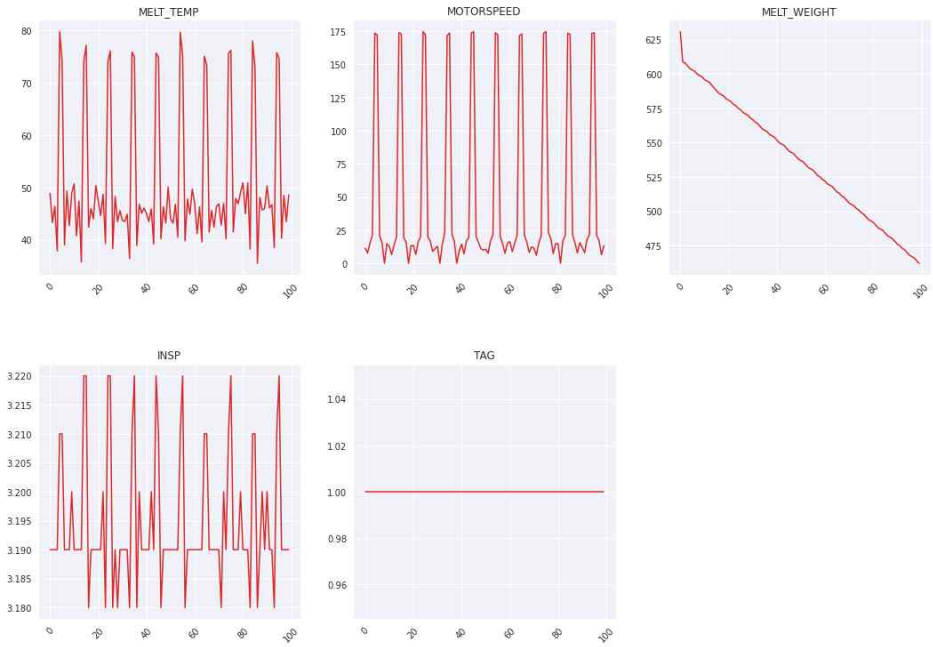
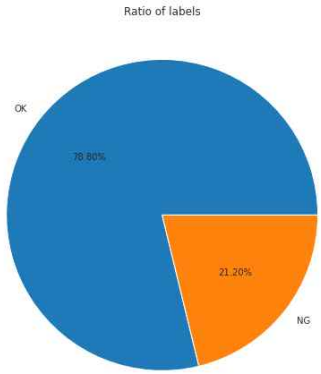


그림 7 : 각 피쳐 별 그래프 (100개 단위)

- 파이 차트를 통해 TAG값, OK와 NG의 비율을 살펴보면 OK의 양은 658133, NG의 양은 177067로, 약 4:1의 분포를 보인다.



4) 훈련/테스트 데이터 분리 및 불균형 데이터 문제 해결 (SMOTE)

- 성공적인 모델 학습을 위해 훈련데이터와 검증데이터, 테스트데이터로 데이터를 분리하는 과정을 진행하였다.

- 훈련/테스트 데이터 분리

  - 일반적인 모델 학습에서는 훈련데이터와 테스트데이터의 라벨값의 비율을 맞춰준다. 편중된 데이터로 학습했을 경우 모델이 제대로 작동하기 어렵기 때문이

다. 그러나 제조업의 이상 탐지 분야에서는 라벨값의 임의 조정이 어렵다. 따라서 본 연구에서는 훈련데이터에서 균등하게 학습을 시키는 것에 초점을 맞추고 테스트데이터를 통해 모델이 잘 작동되는지 확인하였다.

- 존재하는 데이터 내에서 테스트할 수 있는 분포를 최대한 균등하게 맞추기 위해 다음과 같은 방법을 사용하였다.

- NG가 데이터 셋 중간에만 분포가 되어 있기 때문에, 테스트데이터를 뒤에서만 나눠주면 NG값의 비율이 현저히 적어진다. 그래서 테스트데이터를 전체 데이터 중간쯤부터 나눠서 할당해 줌으로써 OK와 NG의 비율을 학습데이터 비율과 비슷하게 맞춰주었다.

- 클래스 불균형 문제 해결 (SMOTE)

  - 데이터의 NG 수가 OK에 비해 현저히 적어서 값을 맞춰주기 위해 imblearn의 SMOTE를 이용하여 학습 데이터의 NG 수를 인위적으로 OK 수에 맞춰주었다.

- 훈련/검증 데이터 분리

  - 모델에 데이터들을 입력하기 전 마지막 단계로 학습데이터와 검증데이터를 나누어준다. sklearn에서 제공하는 train_test_split 함수를 사용하여, 기존의 train_feature와 train_label을 7:3으로 다시 나누어 학습데이터 X_train과 y_train, 검증데이터 X_valid와 y_valid를 분리한다. 전처리과정에서 테스트 데이터를 30%로 떼어냈으므로, 훈련데이터, 검증데이터, 테스트데이터의 비율은 49:21:30이 된다. (표 1)

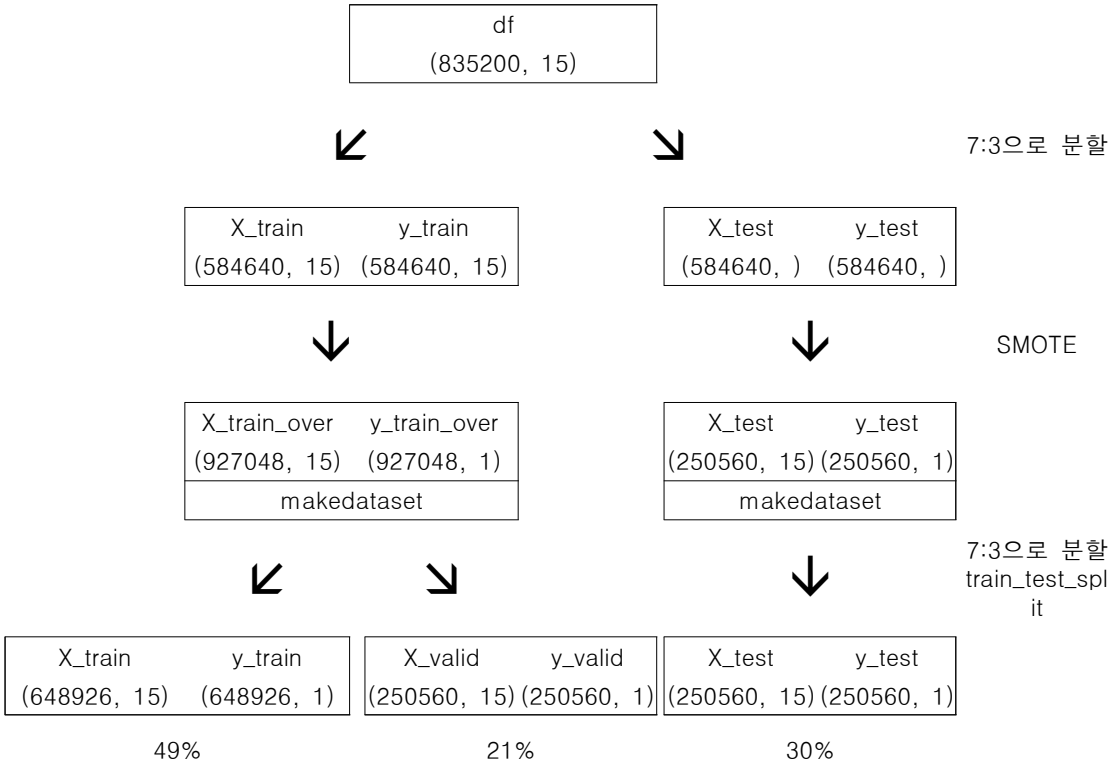


표 1 : 데이터 분리 및 전처리 과정

5) 라벨 분리 및 Min Max 정규화

- 라벨 컬럼을 새로운 변수 y에 할당 후, 데이터셋에 있는 라벨 컬럼을 삭제하였다.
- 다음으로 sklearn의 MinMaxScaler를 통해 정규화를 진행하여 데이터를 0과 1사이의 값으로 변환하였다.

6) Window 정의 및 데이터셋 생성

- RNN을 이용한 시계열데이터의 예측 문제에서 많이 사용되는 방법은 슬라이딩 윈도우(sliding window)를 사용하는 방법이다. 인터벌(window size)은 10으로 잡았는데, 그 이유는 교반속도와 온도가 10을 주기로 일정한 주기성을 가지기 때문이다.
- 한 시점 t에서의 feature를 벡터 x_t로, label을 y_t로 표현하자. x_t는 15차원의 벡터이고 y_t는 0 또는 1의 값이다. windows size가 10인 sliding window 방식은 다음과 같이 설명된다. 과거의 feature vector 10개 x_{t-10}, x_{t-9}, x_{t-8}, …, x_{t-1}를 가지고 현재의 label y_t를 예측한다.

- 따라서, 본 실험에서 사용하는 RNN은 10개의 15차원 벡터 x_{t-10}, x_{t-9}, x_{t-8}, …, x_{t-1}를 입력값으로 받고, 마지막 timestep에서 하나의 실수값 z_t을 출력한다. 이 실수값은 다시 시그모이드(sigmoid) 함수를 거쳐 0과 1 사이의 값 $\hat{y}_t = \sigma(z_t)$으로 변환된다. 우리는 이 예측값 $\hat{y}_t$을 과거 10개의 timestep을 가지고 얻은, 시각 t에서의 제품의 품질이 양품(label : 1)일 확률로 생각할 수 있다. 구체적인 모델 구조는 아래에 그림으로 나타내었다.

- sliding window를 적용하기 전 학습데이터는 X_train_over와y_train_over로, 테스트 데이터는 X_test와 y_test로 할당했었다. X_train_over는 15차원의 벡터들이 927048개 쌓여있는 것으로, 모양이 (927048, 15)인 행렬이다. y_train_over는 0 또는 1의 값이 927048개 쌓여있는 것으로 927048차원의 벡터이다. y_train_over는 SMOTE를 거치고 난 상태이므로 0(NG)과 1(OK)의 비율이 1:1로 맞춰져있다. 다시 말해, y_train_over에서 0의 개수와 1의 개수는 모두 463524개로 동일하다. X_test는 15차원의 벡터들이 250560개 쌓여있는 것으로 모양이 (250560, 15)인 행렬이다. y_test는 0 또는 1의 값이 250560개 쌓여있는 것으로 250560 차원의 벡터이다.

- sliding window는 makedataset이라는 함수에 X_train_over와 y_train_over (또는 X_test와 y_test), 그리고 window size인 10을 인자로 넣음으로써 구현된다. 그러면 train_feature와 train_label (또는 test_feature와 test_label)이 반환된다. train_feature는 X_train_over를 열 개 행씩 잘라 하나의 샘플을 만든 결과이므로, 모양이 (927038, 10, 15)인 텐서가 된다. 다시 말해, $t = 10, 11, \cdots, 927047$에 대하여 t번째 샘플을 행렬 $[x_{t-10}\ x_{t-10}\ \cdots\ x_{t-1}]^T$이라고 두는 것이다. 반면 y_train_over는 각 t에 대하여 y_t만을 기록하므로 927038차원의 벡터가 된다. 다만 코드 상에서의 y_train_over는 모델의 입력값으로 넣기 위하여 열벡터, 즉 (927038, 1) 모양의 행렬로 두었다. 마찬가지로 test_feature는 (250550, 10, 15) 모양의 텐서이고, test_label은 (250550, 1) 모양의 행렬이다.

□ 분석모델 개발

4. 분석 모델

1) AI 분석모델 : GRU

- RNN과 LSTM

  RNN은 순차데이터(sequential data)를 처리하는 데 있어서 아주 효과적인 모델이지만, 기본적인 RNN 모델(vanilla RNN) 보다는 LSTM(Long Short Term Memory, hochreiter & schmidhuber 1997)이 훨씬 더 나은 성능을 보이는 것으로 알려져있다. LSTM은 RNN의 고질적인 문제였던 장기 의존성(long term dependency) 문제를 해결하여 RNN의 표준적인 알고리즘으로 확립된 바 있다. (그림 8)

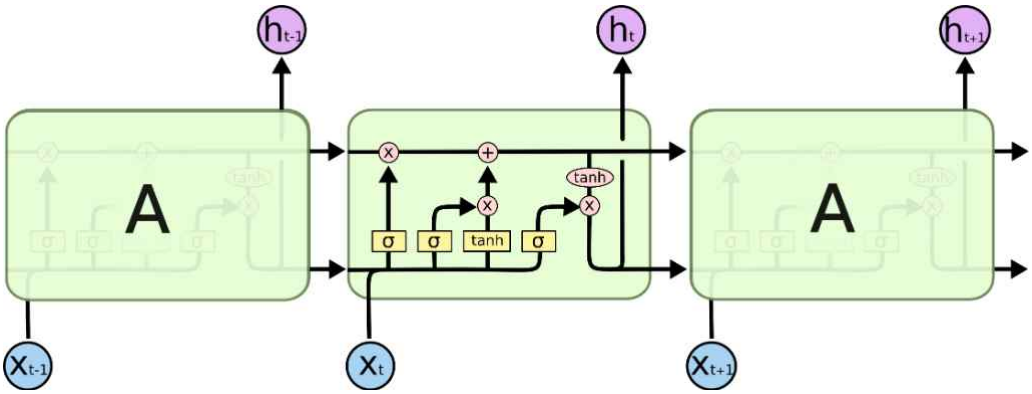
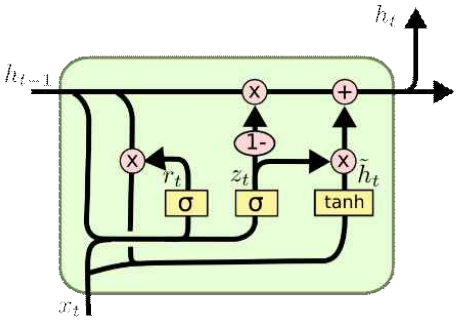


그림 8 : LSTM 의 구조(http://colah.github.io/posts/2015-08-Understanding-LSTMs)

- GRU

  한편, GRU(Gated Recurrent Unit, Cho et al., 2014)는 LSTM을 간소화하여 만든 구조이다. 아래 그림은 GRU의 구조를 나타낸 것으로, LSTM에서는 3개였던 게이트 수가 GRU에서는 2개로 줄어들었고, LSTM에서는 은닉상태벡터(hidden state vector)와 셀상태벡터(cell state vector)를 모두를 사용했다면 GRU에서는 은닉상태벡터(hidden state vector)만을 사용하고 있다. (그림 9)



$$z_t = \sigma\left(W_z \cdot [h_{t-1}, x_t]\right)$$
$$r_t = \sigma\left(W_r \cdot [h_{t-1}, x_t]\right)$$
$$\tilde{h}_t = \tanh\left(W \cdot [r_t * h_{t-1}, x_t]\right)$$
$$h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t$$

그림 9: GRU의 구조(http://colah.github.io/posts/2015-08-Understanding-LSTMs)

- 알고리즘(GRU) 선정 이유

  GRU는 LSTM보다 간단하게 구성되어 있음에도 불구하고 LSTM과 비슷한 성능을 내는 것으로 알려져 있다. 따라서 본 실험에서 GRU와 LSTM 모두를 사용하였고, 그 결과 GRU가 LSTM보다 조금 더 낮거나 비슷한 성능을 보이는 것으로 관찰되었다.

  LSTM과 GRU의 성능에 관해서는 아직도 많은 연구가 이루어지고 있는 중이기 때문에, 정확히 어떤 이유에서 GRU가 더 나은 성능을 보이는지 이야기하는 것은 쉽지 않다. 한편, LSTM은 GRU에 비해 먼 과거를 효율적으로 포착하는 것으로 알려져 있다. 하지만, 이번 문제에서는 주요 특성(feature)들이 10 정도의 주기를 가지고 있으므로, 아주 먼 과거를 포착하는 것이 큰 의미는 없을 것이다. 또한, 복잡하지 않고 간단한 수열(sequence)에 대한 문제에서는 GRU가 LSTM보다 더 나은 성능을 보인다는 연구[1] 등을 통해 보면, 원래 데이터의 컬럼 수가 적은 이번 문제에 GRU가 적합하리라는 결론을 조심스럽게 내릴 수 있다.

- 적용하고자 하는 AI 분석 방법론(알고리즘)의 구체적 소개

  GRU 레이어는 한 개만을 사용할 수도 있지만(single layer GRU), 조금 더 정교한 모델 구조를 위해 여러 개의 GRU 레이어를 쌓기도 한다(stacked GRU). 이번 실험에서는 1 layer GRU, 2 layer GRU, 3 layer GRU 등을 고려했지만, 결과적으로 3 layer GRU가 가장 좋은 성능을 보였으므로, 3 layer GRU를 최종 모델로 선정했다.

  활성화함수(activation function)로는 RNN에서 표준적으로 쓰이는 하이퍼볼릭탄젠트(hyperbolic tangent) 함수를 사용했다. 옵티마이저(optimizer)는 Adam을 사용하였다.

1) Cahuantzi, Roberto, Xinye Chen, and Stefan Güttel. "A comparison of LSTM and GRU networks for learning symbolic sequences." (2021).

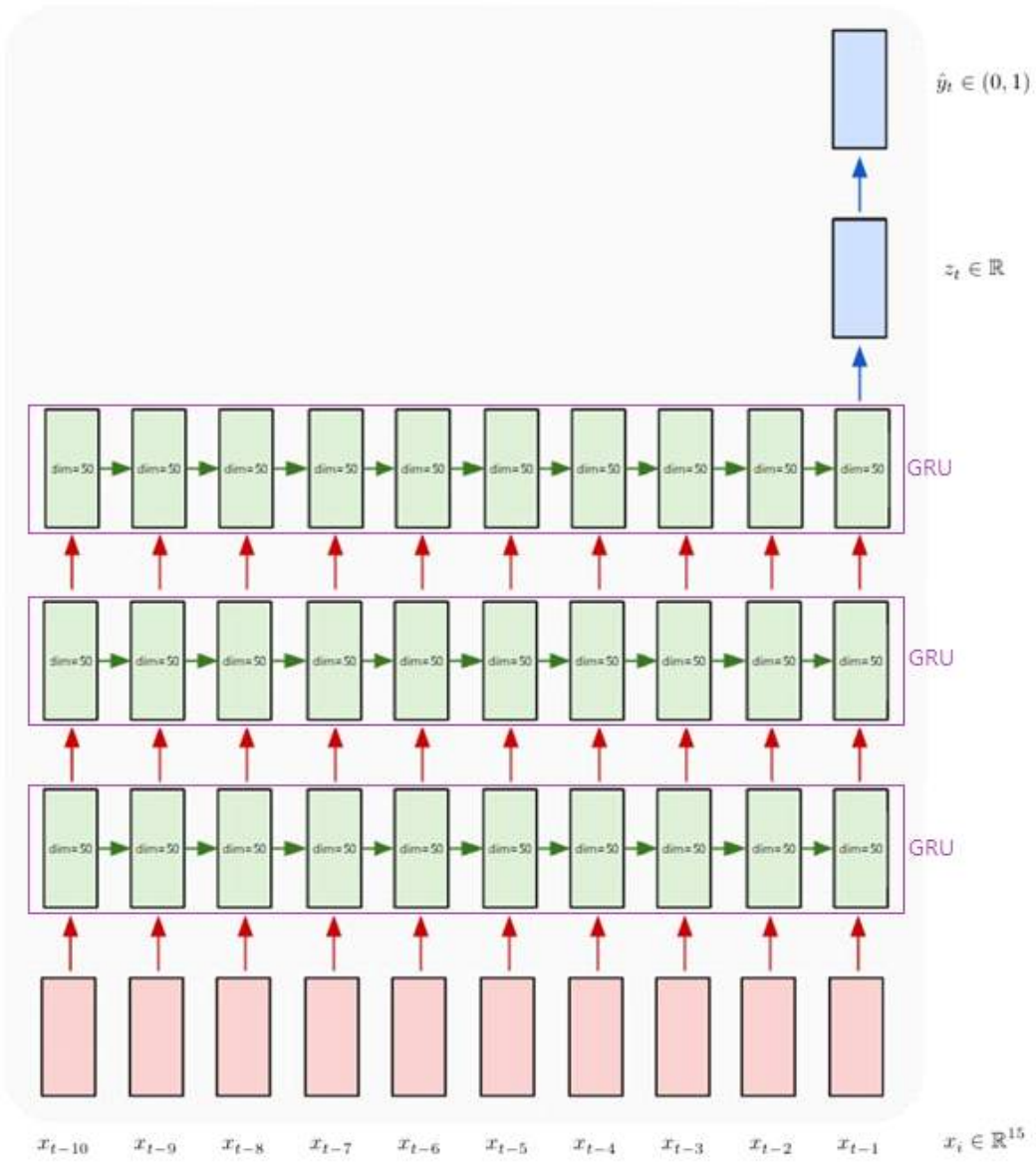


그림 10 : 모델 구조

2) 모델 구축 및 훈련

- GRU 모델 구축

  모델의 구성, 학습, 평가 등은 keras의 sequential API를 사용하여 구현하였다. model.add를 통해 세 개의 GRU 레이어를 쌓았다. 처음 두 레이어에서는 10개의 timestep 모두에서 은닉상태벡터가 출력되어야 하므로 return_sequences=True로 두었고, 마지막 레이어에서는 마지막 timestep에서만 결과가 출력되어야 하므로 return_sequences=False로 두었다. 그리고 출력된 결과를 0과 1 사이의 값으로 변환시켜주기 위하여 시그모이드 함수를 활성화함수로 하는 선형 레이어를 하나 더 쌓아주었다. 또한, 모델의 과적합(overfitting)을 미연에 방지하기 위해 조기 종료(early stopping) 옵션을

사용하였다. 그리고 ModelCheckpoint를 통해 모델이 가장 잘 학습되었을 때의 가중치들을 파일로 저장하여 나중에 불러올 수 있게 했다.

- 모델 훈련

  X_train과 y_train을 model.fit 함수에 넣어 학습을 진행하였다. epoch 수는 200으로 설정하였지만, 앞서 말한 조기종료 때문에 20 epoch 전후에 학습은 종료된다.

□ 분석결과 및 시사점

5. 모델 평가 및 결과해석

1) 손실함수 그래프

- 손실함수에 대하여 그래프를 그려봄으로써 모델이 잘 학습되었는지 확인하였다. 아래 그림은 epoch 수에 따른 학습데이터의 검증데이터의 손실 값에 대한 그래프이다. (그림 11)

- 그래프에 따르면 손실함수는 대체로 감소하는 경향을 보였다. epoch 13 이후에는 손실함수의 값이 한번씩 증가하는 모습을 보이는데, 이것은 과적합(overfitting)이 일어난 것으로 보았다. 이 실험에서는 조기 종료(early stopping)을 통해 과적합이 일어나는 것을 사전에 방지했다.

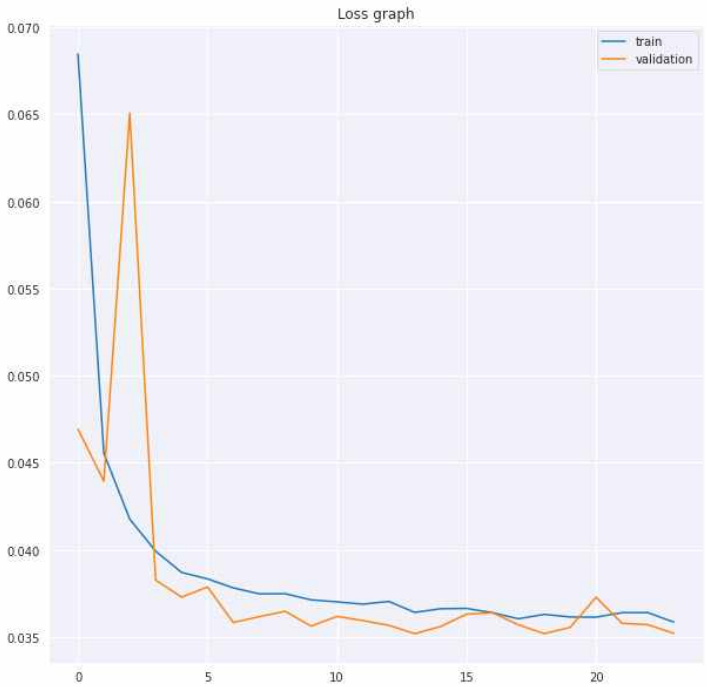


그림 11 : 손실함수 그래프 (학습데이터, 검증데이터)

2) 평가 지표 소개

모델 학습 후 예측 결과를 통한 평가지표로 confusion matrix, accuracy, F1 score, roc-auc score를 사용하였다.

- 혼동행렬(Confusion Matrix)

혼동행렬은 모델의 예측 결과와 실제 결과의 비교를 행렬로 나타낸 결과이

다. 행은 실제 결괏값을 표현하며 열은 모델의 예측 결과값을 표현한다. 앞에서 언급하였듯이, 우리가 다루고 있는 훈련데이터는 불량(NG)보다 양품(OK)가 훨씬 많이 분포된 불균형 데이터이기 때문에, 종합적으로 예측의 결과를 확인할 수 있어야 한다. 그것의 가장 기본적인 단계로 혼동행렬(confusion matrix)을 관찰하였다.

모델을 훈련한 후 훈련 샘플들을 모델에 넣어 예측값을 얻으면 0과 1 사이의 실숫값을 얻는다. 훈련 샘플의 개수가 250560개이므로, 250560차원의 벡터를 얻는 셈이다. 이것은 코드상에서 pred라는 객체로 저장되어 있다. 이 벡터는 그 성분이 0과 1 사이의 실숫값이므로, 0.5를 임계값(threshold)으로 하여 0.5보다 작은 값들은 0으로, 0.5보다 크거나 같은 값은 1로 변환시켜 pred_df라는 객체로 저장하였다. 이 pred_df를 실제 데이터의 label인 test_label과 비교하여 혼동행렬을 만들 수 있다.

- 정확도(Accuracy)

  정확도는 전체 훈련 샘플들 중 옳게 예측한 샘플들(TP + TN)의 비율을 나타낸다. 분류문제에서 생각해볼 수 있는 가장 기본적인 평가 지표이다.

- F1 score

  이진분류기(binary classifier)의 성능을 측정하기 위해서는 정밀도(precision)과 재현율(recall)을 모두 고려해야 한다. 정밀도는 OK로 예측한 샘플들 중 실제로 품질이 OK인 샘플들의 비율로, type 1 error(FP)와 연관되어 있다. 재현율은 실제로 품질이 OK인 샘플들 중 OK로 예측한 샘플들의 비율로, type 2 error(FN)와 연관되어 있다.

  불균형한 데이터를 사용할시, 일반적으로 정밀도와 재현율의 지표를 사용하는데, 단독으로 하나만 선택해서 사용할 수는 없다. 정밀도와 재현율은 서로 상충하는 관계를 띄기 때문에 둘 중 하나만의 지표를 사용하여 평가한다면 분류기의 예측값이 극단적으로 몰릴 수 있기 때문이다. 그래서 불균형한 데이터에 대한 정확한 분석을 위해 두 지표를 모두 사용하는 것이 바람직하다. 이 경우에는, FP와 FN의 평균이라는 의미에서 정확도와 재현율의 조화평균을 계산한 f1-score를 주요 지표 중 하나로 선택하였다.

- AUC(area under the ROC curve)

  AUC를 설명하기에 앞서, ROC Curve에 대해 살펴보아야 한다. ROC

(Receiver Operating Characteristic) Curve은 TPR(TP Rate, Sensitivity)을 y축으로, FPR(FP Rate, 1-Specificity, NG 샘플에 대한 recall)을 x축으로 하는 그래프로 모든 임계값에서 분류 모델의 성능을 보여준다. AUC는 ROC 곡선 아래 영역을 의미한다. AUC의 값은 0.5와 1 사이의 값으로 나타나며 AUC의 값이 클수록 모델의 성능이 좋다고 말할 수 있다.

AUC는 가이드북에는 설명되어 있지 않은 평가 지표다. 하지만, 이번 실험에서 중요하게 쓰인 SMOTE에 관한 논문[2]을 읽어보니, AUC가 주요한 평가 지표로서 쓰이고 있음을 고려해보았을 때, AUC도 마찬가지로 이번 실험의 평가 지표로 넣는 것이 바람직하다고 보았다.

3) 분석 결과

- 혼동행렬(Confusion Matrix)

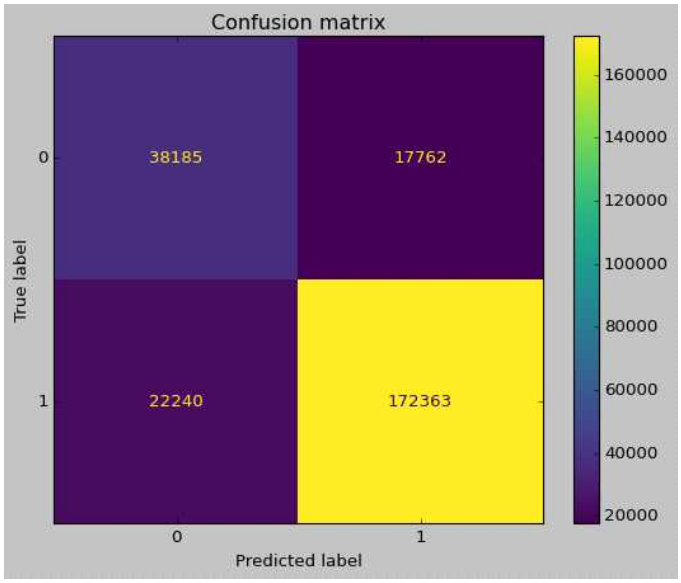


그림 12 : 혼동행렬

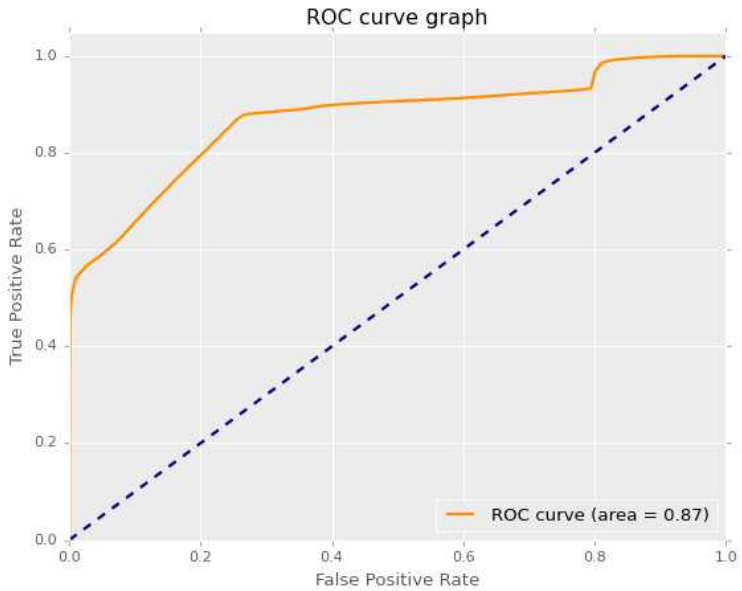
- 왼쪽 위 (TN, 38185) : 실제 품질 NG, 모델 예측 NG인 샘플의 개수
- 오른쪽 위 (FP, 17762) : 실제 품질 NG, 모델 예측 OK인 샘플의 개수
- 왼쪽 아래 (FN, 22240) : 실제 품질 OK, 모델 예측 NG인 샘플의 개수
- 오른쪽 아래 (TP, 172363) : 실제 품질 OK, 모델 예측 OK인 샘플의 개수

- 평가지표 (표2)

2) Chawla, Nitesh V., et al. "SMOTE: synthetic minority over-sampling technique." Journal of artificial intelligence research 16 (2002): 321-357.

평가지표	
accuracy	0.8403
f1-score	0.8960
AUC	0.8710

- ROC curve



□ 중소제조기업에 미치는 파급효과

6. 파급효과

1) 적용 가능한 제조 현장

현재 제조 현장에서는 공정 중 내용물 샘플을 채취하는 것 자체에 애로사항이 많다. 만약 채취한다고 해도 공정 운영자의 노하우와 경험에만 의존하여 설비 운영값을 변경하는 방법밖에는 없다. 이 말은 운영자/숙련자의 부재시, 용해 공정을 실질적으로 관리 가능한 사람이 없다는 뜻이고 그동안은 설비 운영값을 제대로 변경하지 못하여 제품이 불량으로 나올 확률이 크다는 뜻이다.

만약 현장에 PLC 또는 DMBS(RDB)를 이용하여 수집한 데이터가 있다면 이 분석 모델을 각 분야 및 공정에 맞게 가중치를 정해주고 실시간 시계열 데이터를 입력하여 실시간 품질 예측이 가능해진다. 또한 숙련자의 부재시, 현장 실무자가 데이터를 보고 설비 운영값을 조절해 주는 것이 가능하다. 그리고 더 나아가 스마트 공장 구축이 된다면, 실시간으로 모델이 데이터를 수집하여 분석 및 결과 예측을 통해 업무 자동화가 가능해진다.

2) 타 공정 및 타 분야로의 확장 가능성

해당 분석모델로는 각 제품 및 투입원료에 따라 결과가 달라질 수밖에 없다. 이를 해결하기 위해 원하는 제품의 데이터로 재분석 및 모델 학습을 통해 값들을 맞춰줘야 한다.

실제 현장에서는 용해공정 데이터만 있는 것이 아닌 다양한 외부요인/변수가 존재한다. 다양한 데이터의 수집이 원활하다면 더욱 체계적인 분석 및 정확한 모델의 설계가 가능해진다. 따라서, 현장 전문가와 의견을 취합하여 해당 데이터의 수집 및 적용여부를 결정해야 한다.