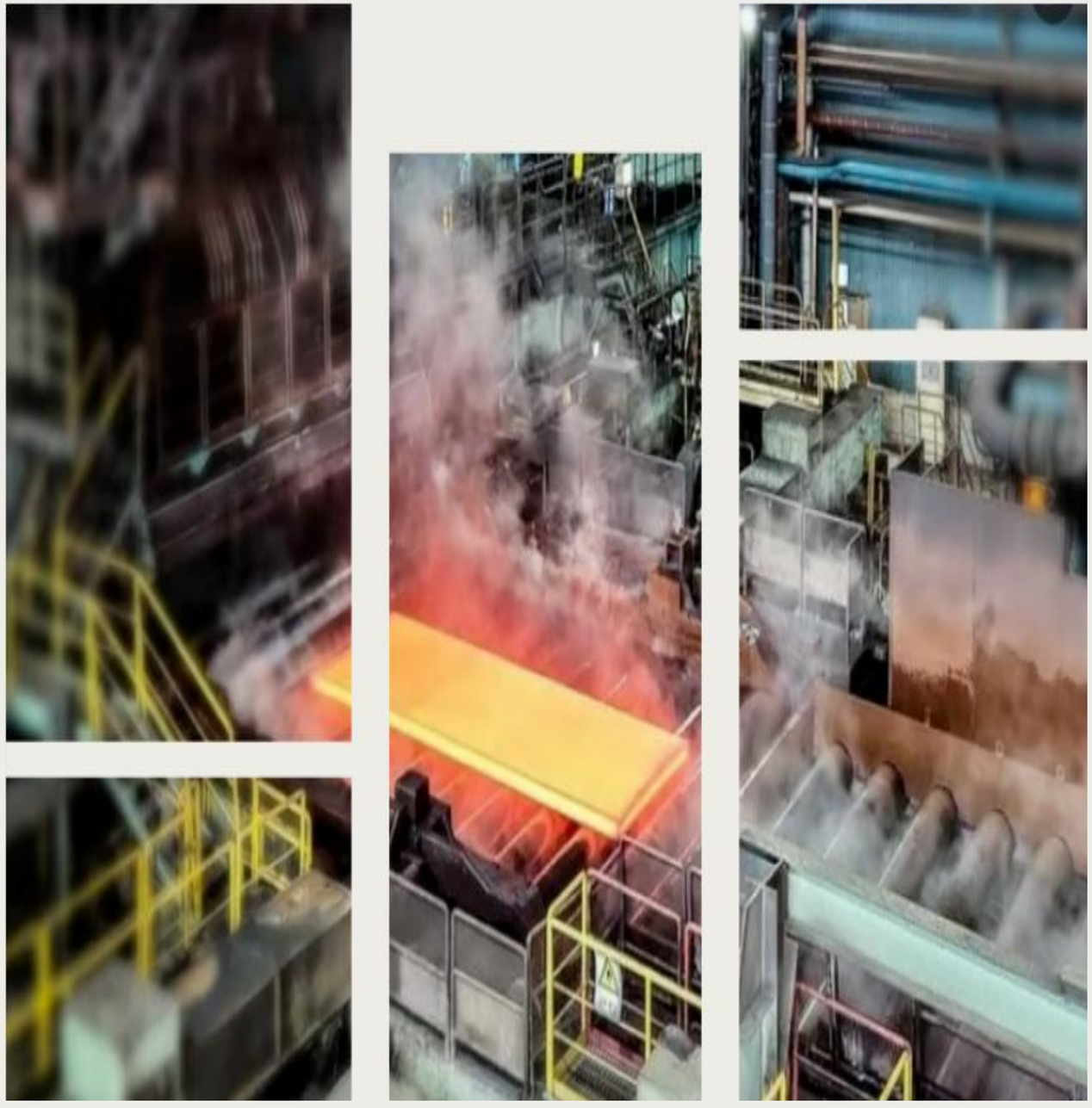
종합실습2_Scale 불량 분석

선박 후판 공정 Scale 불량 예측



Posco Big Data Academy 31기

A3조 *홍정택, 이민성, 정주영, 허유진, 윤영채, 유창우

Scale 불량 예측 모델

Table of Contents

POSCO -inc

철강 산업의 핵심 공정인 포스코 후판공정은
수십 년간 축적된 기술력과 품질관리 노하우를 바탕으로,
조선·건설·에너지 산업 전반에 고품질 후판을 공급하고 있습니다.

과제 정의

POSCO에서 고객사 외주 제품 생산 시 Scale 불량 발생 증가 이슈 발생

원인 분석 결과 압연 공정에서 Scale 불량 급증

데이터 수집 후 분석을 통한 불량 발생 근본 원인 및 개선 기회 도출

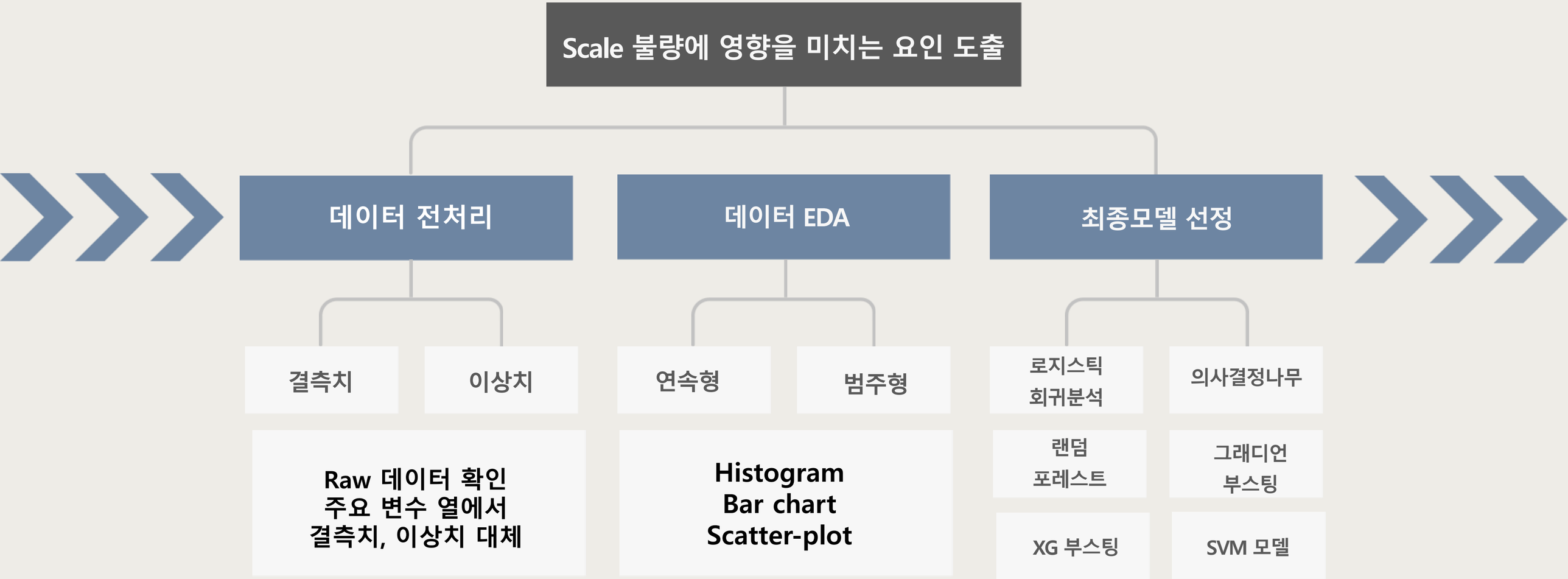
포스코그룹은 해외 투자 가속화로 철강 체질 개선을 통한 실적 성장을 이어간다는 방침이다. 포스코홀딩스 관계자는 이날 콘퍼런스콜에서 "향후 국내에선 경쟁력이 떨어지는 설비가 있으면 과감하게 셧다운을 추진할 것"이라고 말했다.

Scale 불량 요인을 분석·예측해 불량률 저감을 통한 품질 경쟁력 및 기업 신뢰도 향상이 목표

철강 사업의 지속 성장을 위해 생산 공정 내 불량률을 낮추고 제품 일관성 확보 필요

02 분석 계획

분석 계획


Scale 불량에 영향을 미치는 요인 도출
데이터 전처리
데이터 EDA
최종모델 선정
결측치
이상치
연속형
범주형
로지스틱 회귀분석
의사결정나무
랜덤 포레스트
그래디언 부스팅
XG 부스팅
SVM 모델
Raw 데이터 확인
주요 변수 열에서
결측치, 이상치 대체
Histogram
Bar chart
Scatter-plot

03 데이터 전처리

결측치 및 변수 설정

01 결측치 확인

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Data columns (total 21 columns):
 #   Column           Non-Null Count  Dtype
---  ------           --------------  -----
 0   plate_no         1000 non-null   object
 1   rolling_date     1000 non-null   object
 2   scale            1000 non-null   object
 3   spec_long        1000 non-null   object
 4   spec_country     1000 non-null   object
 5   steel_kind       1000 non-null   object
 6   pt_thick         1000 non-null   int64
 7   pt_width         1000 non-null   int64
 8   pt_length        1000 non-null   int64
 9   hsb              1000 non-null   object
 10  fur_no           1000 non-null   object
 11  fur_input_row    1000 non-null   object
 12  fur_heat_temp    1000 non-null   int64
 13  fur_heat_time    1000 non-null   int64
 14  fur_soak_temp    1000 non-null   int64
 15  fur_soak_time    1000 non-null   int64
 16  fur_total_time   1000 non-null   int64
 17  rolling_method   1000 non-null   object
 18  rolling_temp     1000 non-null   int64
 19  descaling_count  1000 non-null   int64
 20  work_group       1000 non-null   object
dtypes: int64(10), object(11)
```

02 변수 선정

- plate_no
  각 후판을 구분하기 위한 고유 식별번호
  인덱스와 같은 역할을 하므로 분석에서 제외

- rolling_date
  날짜와 시간이 모두 포함된 변수
  시간 단위의 분석을 위해 시각(hour) 정보만 추출하여 rolling_hour 파생변수 생성

- spec_long
  특정 강종과 두께 정보를 조합한 변수
  이미 변수로 보유 중인 steel_kind와 pt_thick의 정보가 포함되어 제외

- spec_country
  국가별 규격 차이로 불량률이 상이하게 나타날 수 있으나,
  이를 특정 국가의 문제로 해석하기 어렵기 때문에 분석에서 제외

03 데이터 전처리

이상치 확인

	pt_thick	pt_width	pt_length	fur_heat_temp	fur_heat_time	fur_soak_temp	fur_soak_time	fur_total_time	rolling_temp	descaling_count
count	1000.00000	1000.000000	1000.000000	1000.000000	1000.000000	1000.000000	1000.000000	1000.000000	1000.000000	1000.000000
mean	26.78200	2831.900000	36788.200000	1157.245000	85.972000	1150.928000	71.720000	238.589000	934.637000	8.557000
std	18.13757	494.081478	13912.387116	21.245007	26.346297	17.344384	20.602137	38.194828	96.598015	1.604158
min	12.00000	1800.000000	7900.000000	1103.000000	55.000000	1113.000000	35.000000	165.000000	0.000000	5.000000
25%	15.00000	2500.000000	26650.000000	1140.000000	66.000000	1135.750000	57.750000	210.000000	893.750000	8.000000
50%	19.00000	2800.000000	40400.000000	1159.000000	75.000000	1156.000000	66.000000	230.000000	948.000000	9.000000
75%	34.00000	3100.000000	49100.000000	1173.000000	102.250000	1164.000000	81.000000	263.000000	991.000000	10.000000
max	100.00000	4600.000000	54900.000000	1206.000000	158.000000	1185.000000	145.000000	362.000000	1078.000000	10.000000

```
df_raw[df_raw['rolling_temp'] == 0][['plate_no', 'rolling_temp']]
```

	plate_no	rolling_temp
53	PLT_1054	0
221	PLT_1222	0
222	PLT_1223	0
598	PLT_1599	0
599	PLT_1600	0
600	PLT_1601	0

```
target_plates = ['PLT_1054', 'PLT_1222', 'PLT_1223', 'PLT_1599', 'PLT_1600', 'PLT_1601']
df_raw.loc[df_raw['plate_no'].isin(target_plates), ['plate_no', 'rolling_temp']]
```

	plate_no	rolling_temp
53	PLT_1054	842.1
221	PLT_1222	959.6
222	PLT_1223	959.6
598	PLT_1599	885.7
599	PLT_1600	885.7
600	PLT_1601	891.0

rolling_temp = 0 인 값이 6개 임을 확인하여 제품규격이 같은 제품 중 sclae =양품인 값으로 대체

04 파생변수

파생변수

- 기존 데이터에는 온도, 시간, 치수 등 개별적인 측정값만 존재해 이들 간의 물리적·공정적 관계를 반영한 변수 부족
- 이에 따라 공정 현상과 불량 메커니즘을 보다 정확히 설명하기 위해 여러 핵심 변수를 조합하여 파생변수를 생성

shift_worker

[설명]
rolling_hours 변수를 활용하여 작업시간대를 주간(07 ~ 18)와 야간(19 ~ 06)으로 범주화

[도출배경]
주간/야간 근무 시 피로도가 근무에 영향 끼칠 것으로 예상

fur_pre_time

[설명]
가열로 총 체류 시간에서 가열대 시간과 균열대 시간을 제외해 예열대 체류 시간 생성

[도출배경]
: 예열시간이 충분하지 않으면 표면 산화층이 불안정해져 scale불량이 증가할 것으로 예상

heat_soak_diff

[설명]
가열대 온도와 균열대 온도의 차이

[도출배경]
열 온도가 급격하게 변화하는 상황이 영향을 미칠 것으로 예상

pt_area

[설명]
제품의 폭과 길이를 곱해 표면 노출 면적(단면 기준)을 산출

[도출배경]
표면 노출 규모에 따른 scale 불량에 영향이 있을 것으로 예상

pt_vol

[설명]
제품의 폭, 길이, 두께의 곱으로 열 용량 대용치인 부피 요인 생성

[도출배경]
부피가 클수록 열 침투가 더디고 온도 균일화가 어려워 scale 불량과 관계가 있을 것으로 예상

steel_usage

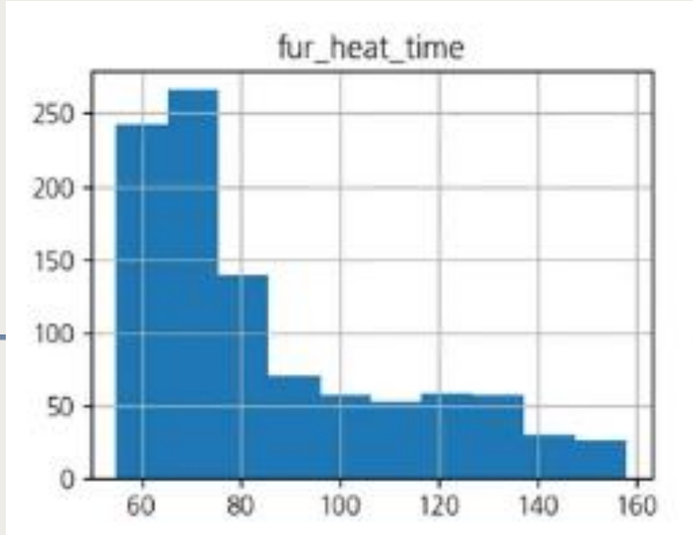
[설명]
spec_long의 강재 규격을 용도별(조선용, 구조용, 압력용기용 등)로 분류해 재질 특성을 반영한 파생변수를 생성

[도출배경]
강재의 용도에 따라 합금성분과 열처리 특성이 달라 가열, 냉각 시 Scale 불량 발생 경향에 차이가 있을 것으로 예상

05 탐색적 분석

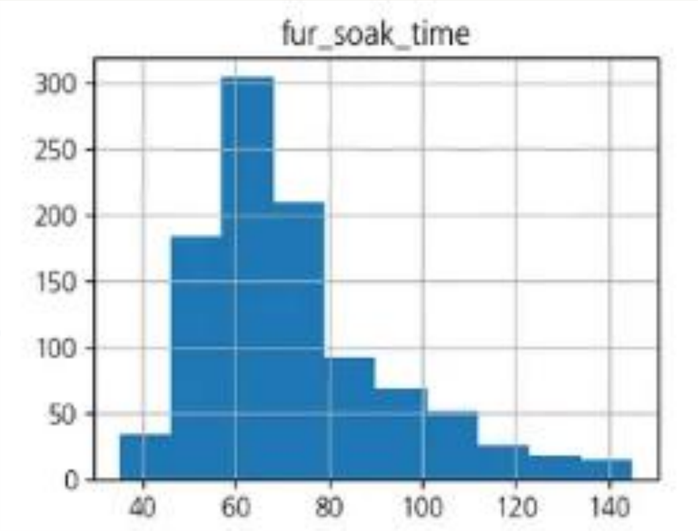
가열대 시간 구간별 비교

※ 세 그래프 모두 x축 : 재로시간, y축 : 빈도를 나타냄



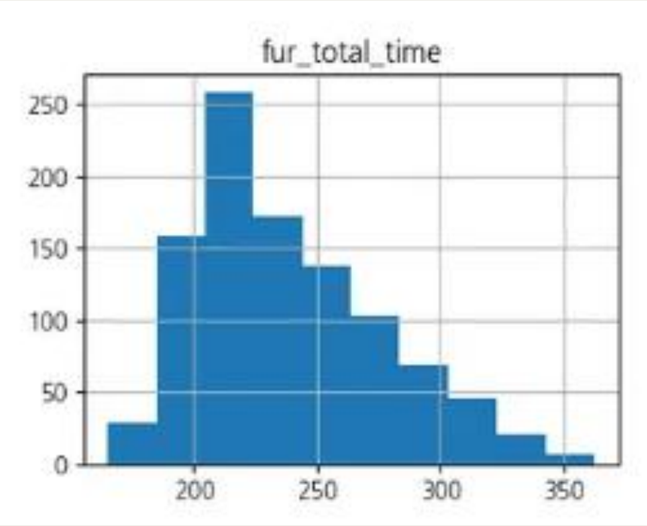
fur_heat_time
(가열로 가열대 재로시간)

- 대부분의 슬라브가 60~80분 구간에 분포하며 일부 장시간 가열 사례 존재



fur_soak_time
(가열로 균열대 재로시간)

- 대부분의 슬라브가 60~80분 구간에 분포하고, 두꺼운 슬라브의 장시간 재로 경향이 존재

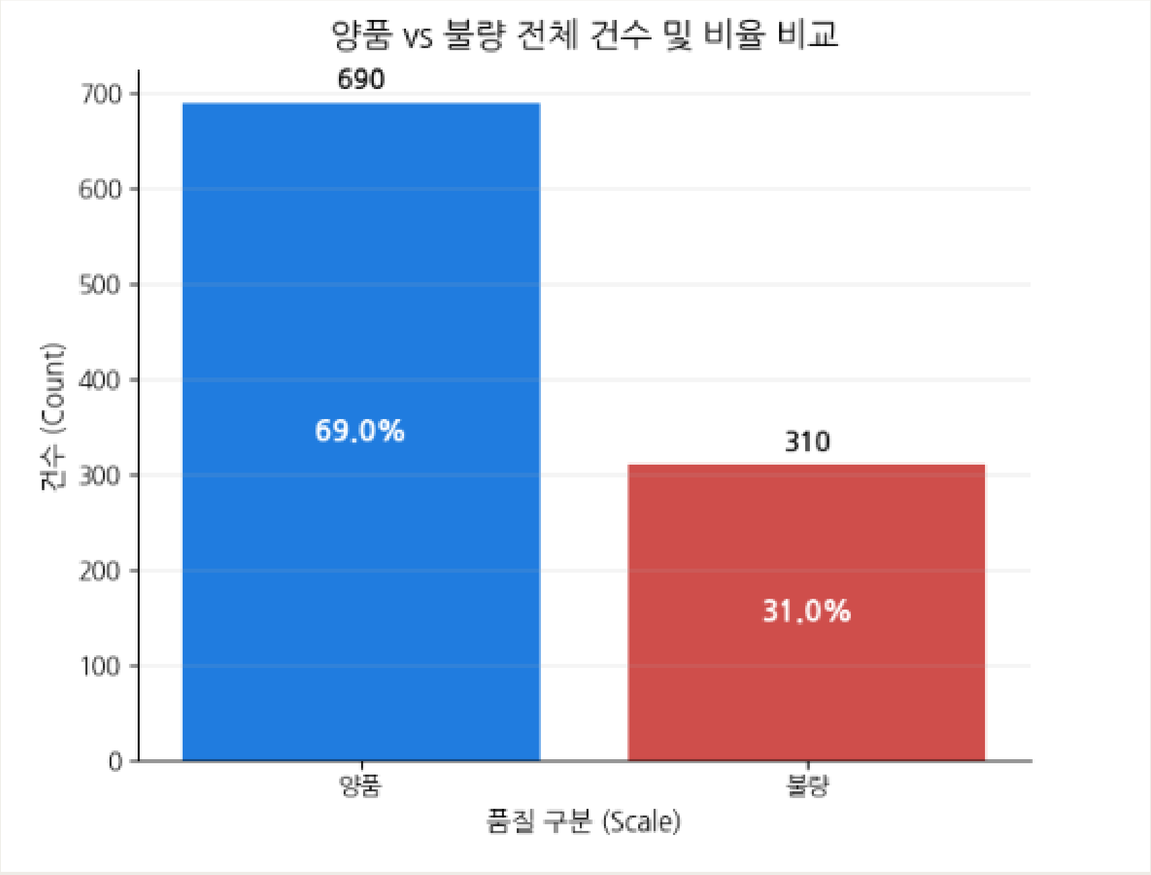


fur_total_time
(가열로 총 재로시간)

- 대부분의 슬라브가 200~250분대에 집중돼 전체 공정이 비교적 안정적으로 운영되는 경향 존재

05 탐색적 분석

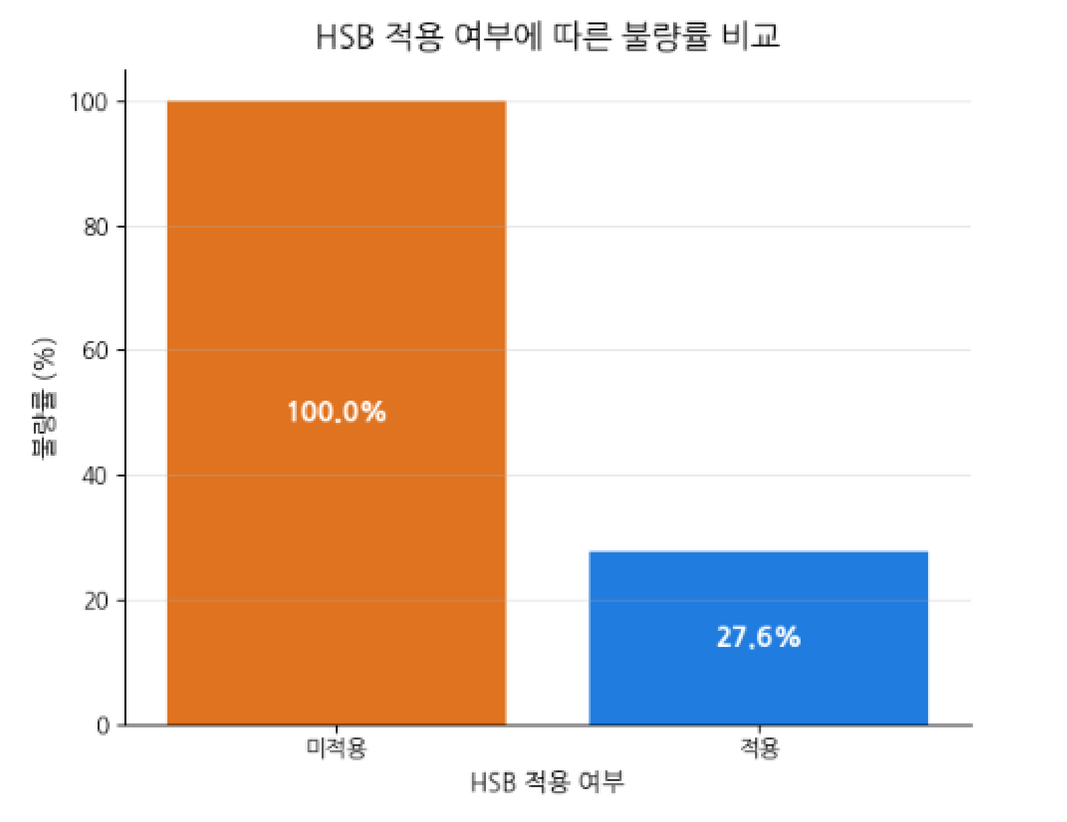
양품 vs 불량 전체 건수 및 비율 비교



- 전체 데이터 중 **양품이 69%, 불량이 31%로 약 7:3 비율**의 분포
  이는 분류 모델에서 **클래스 불균형이 심각한 수준(예: 9:1 이상)은 아니**므로, 분석 시 SMOTE와 같은 **오버샘플링 미적용**
- 다만 **불량률이 30% 수준으로 현실적인 생산 라인의 결함 비율과 유사**하다는 점에서 본 데이터는 실제 공정 상황을 충분히 반영한 학습 데이터로서 신뢰도 확보 가능
- 따라서 기본 **모델 학습 시에는 원본 데이터의 Scale 비율을 유지**하고, 이후 **모델별 성능(Recall, Precision) 편차가 클 경우에만 SMOTE 적용** 여부를 실험적 비교 대상으로 고려하기로 결정

05 탐색적 분석

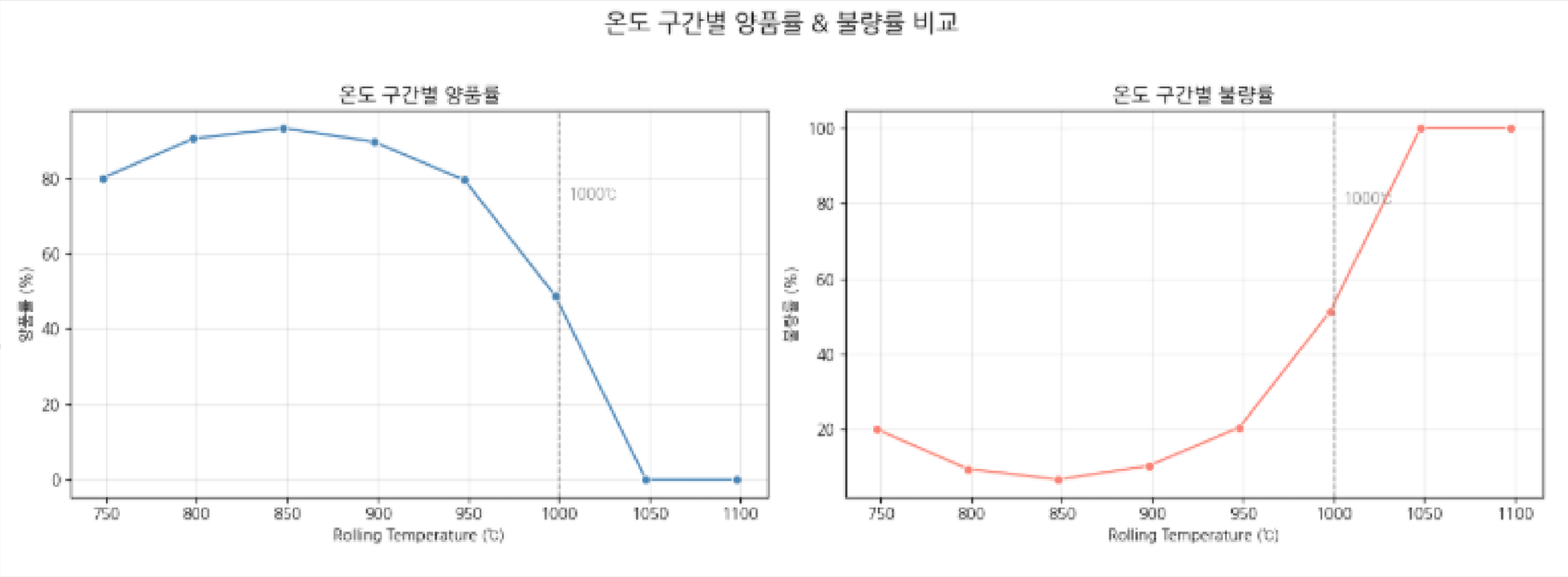
HSB 적용 여부에 따른 불량률 비교



- **hsb(Hot Scale Breaker) 미적용 시 100% 확률로 불량 발생**
- 다만, **적용 시에도** 여전히 **약 27% 수준의 불량이 남아** 있으므로, 온도 구간, 냉각 속도, 강종별 차이 등 **다른 변수와의 상호작용 분석이 추가로 필요**
- **hsb**는 주로 고온 상태의 슬라브 표면에 발생한 산화 스케일을 제거해 롤 마찰 불균형 및 균열 발생을 방지하는 역할을 수행하므로, 공정 안정성을 위해 **적용 여부가 불량 발생의 주요 결정 요인을 시사**

05 탐색적 분석

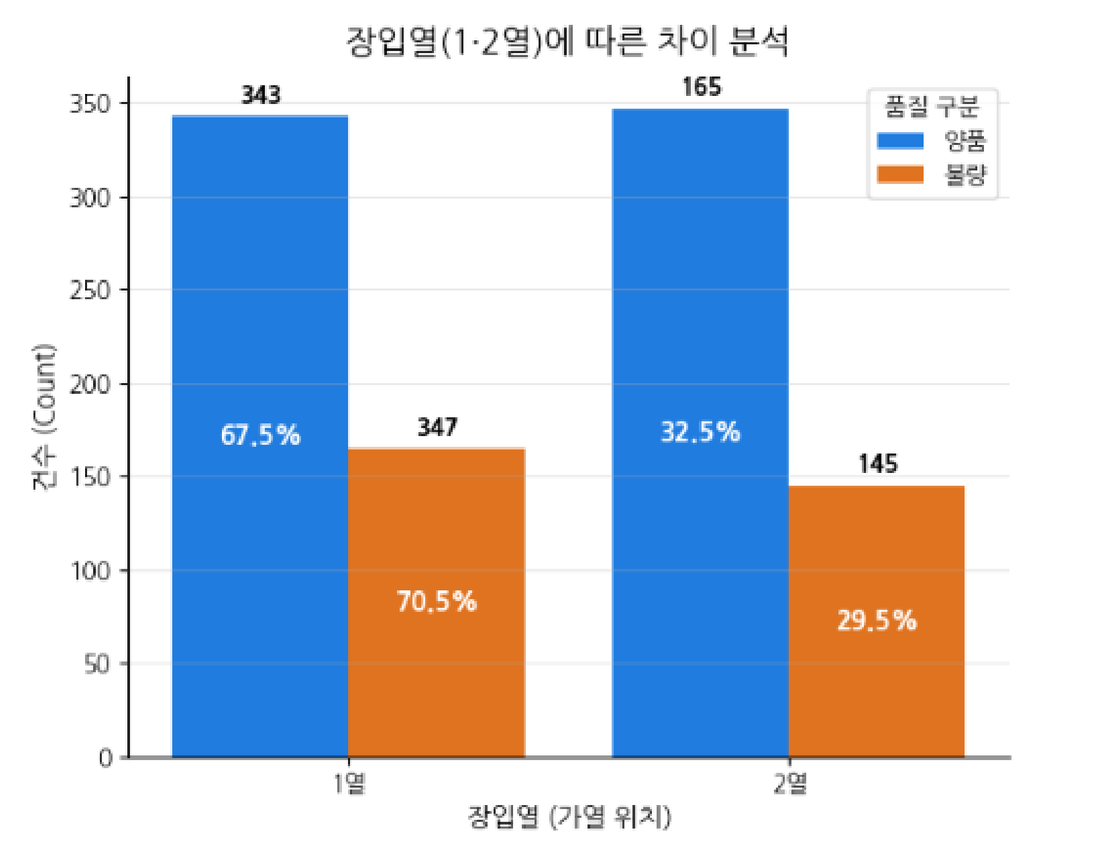
압연온도가 1000°C 일 때 양품·불량 파악



- **rolling_temp = 1000 °C 구간에서의 양·불량 분포 확인**
- 압연 온도가 **950도 이상부터 불량률**이 **기하급수적**으로 **상승**
- Scale **불량을 최소화**하기 위해서는 **압연 온도를 950°C 미만**, **특히 800~950°C 구간으로 안정적으로 관리**하는 것이 **중요**

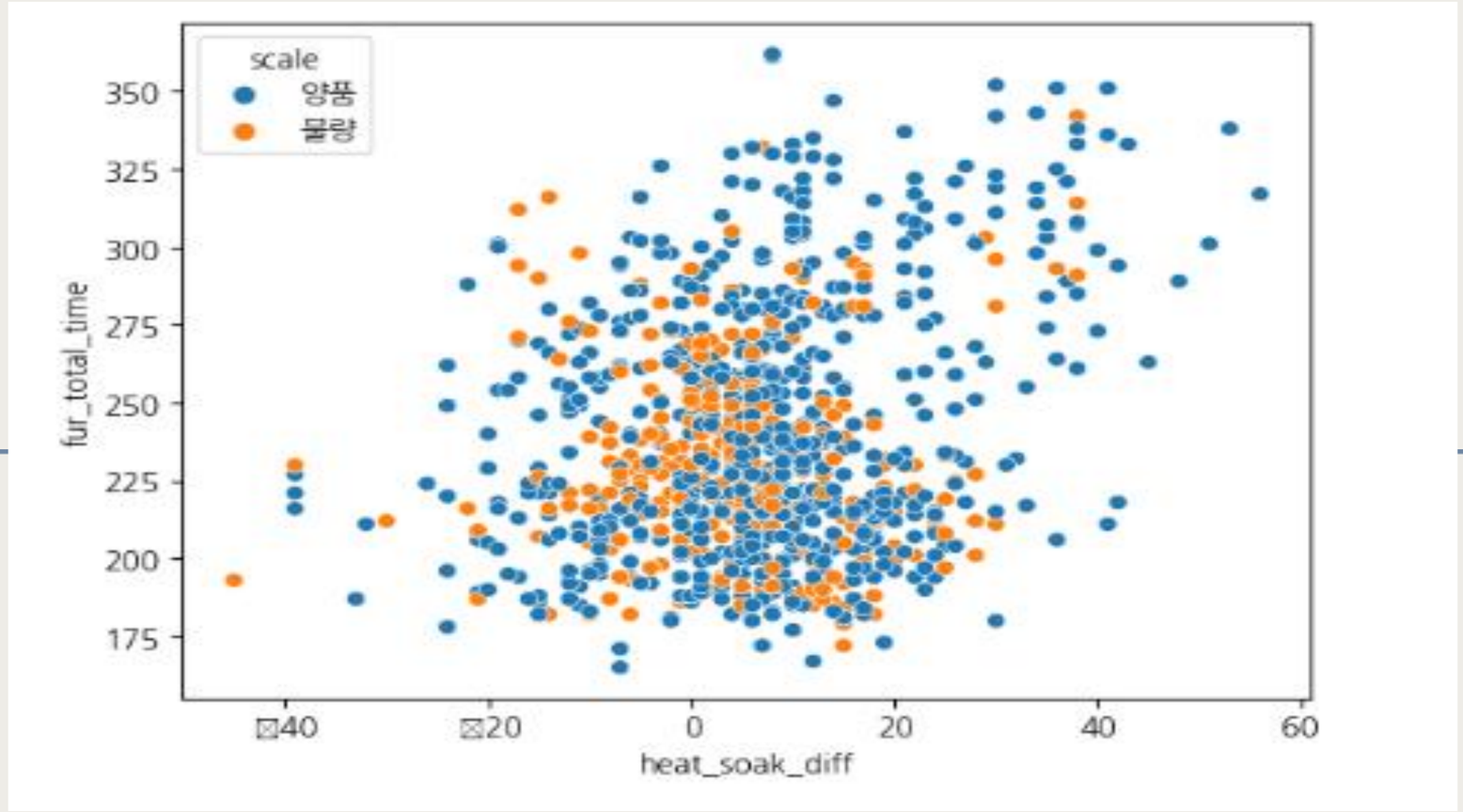
05 탐색적 분석

장입열(1·2열)에 따른 차이 분석



- 장입열(가열 위치) **1열과 2열 모두 불량과 양품의 비율이 거의 동일하게 분포**
- 따라서 **장입열(가열 위치)**은 Scale **불량 발생에 대한 주요 영향 요인으로 보기 어려움**

가열로별 △temp를 통해 가열로 성능 차이 파악



- **가열대 소재온도**와 **균열대 소재온도 차이**가 plate의 **scale 여부에 미치는 영향** 분석
- **가열로 총 재로시간**이 plate의 **scale 여부에 미치는 영향** 분석
- 두 변수만으로는 **scale 여부를 명확히 분류하기 어려울 것**으로 판단

06 모델링

로지스틱 회귀분석

HyperParameter 결정

```
Optimization terminated successfully.
         Current function value: 0.398194
         Iterations 7
                           Logit Regression Results
==============================================================================
Dep. Variable:                  scale   No. Observations:                  700
Model:                          Logit   Df Residuals:                      694
Method:                           MLE   Df Model:                            5
Date:                Tue, 04 Nov 2025   Pseudo R-squ.:                  0.3556
Time:                        00:15:02   Log-Likelihood:                -278.74
converged:                       True   LL-Null:                       -432.57
Covariance Type:            nonrobust   LLR p-value:                2.254e-64
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept     -1.3349      0.126    -10.606      0.000      -1.582      -1.088
fur_soak_temp  1.2563      0.202      6.226      0.000       0.861       1.652
fur_heat_time  0.3906      0.127      3.086      0.002       0.142       0.639
rolling_temp   1.4427      0.158      9.128      0.000       1.133       1.752
descaling_count -1.2864    0.175     -7.363      0.000      -1.629      -0.944
pt_vol        -0.2628      0.104     -2.520      0.012      -0.467      -0.058
==============================================================================
```

	Variable	VIF
0	const	1.00
5	pt_vol	1.04
2	fur_heat_time	1.17
3	rolling_temp	1.71
4	descaling_count	1.77
1	fur_soak_temp	2.84

» Scale에 따른 독립변수와의 로지스틱 회귀모델
후진제거법(>0.05)과 다중공산성(>10)
설명계수 값: 35.6% -> 설명력이 낮음

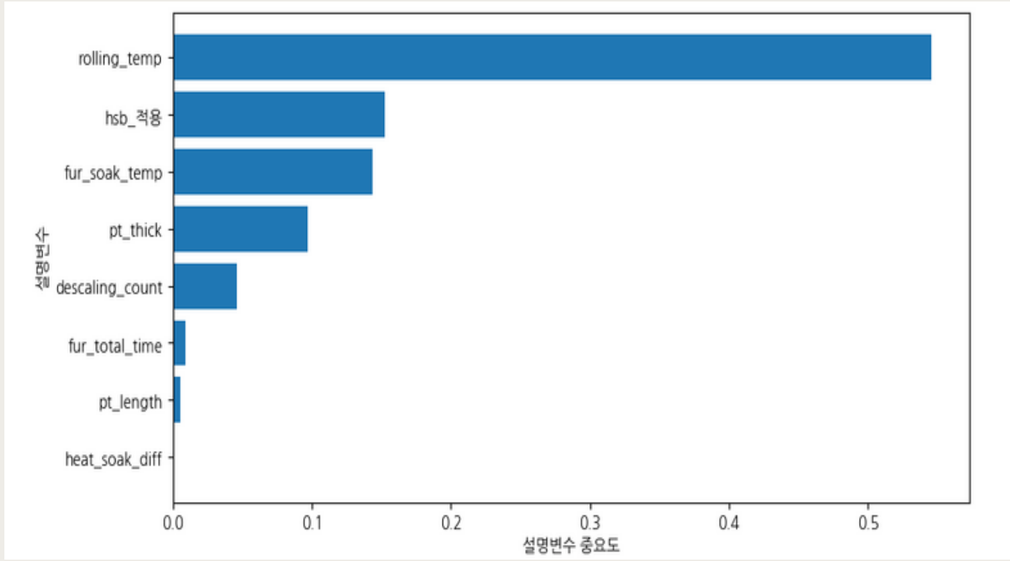
Confusion Matrix

```
Confusion Matrix:
[[189  17]
 [ 43  51]]

              precision    recall  f1-score   support

           0      0.815     0.917     0.863       206
           1      0.750     0.543     0.630        94

    accuracy                          0.800       300
   macro avg      0.782     0.730     0.746       300
weighted avg      0.794     0.800     0.790       300
```

» Recall 값이 54%,
F1-score 값이 63%로 낮음

데이터 불균형이 있는 목표변수를
분류하기에 적합하지 않은 모델로 판단

중요 변수 확인



» • rolling_temp, hsb_적용
• fur_soak_temp의 중요도 높음
• rolling_temp는 다른 변수 대비 중요도 매우 높음

06 모델링

Decision Tree

HyperParameter 결정

```
# 8. 최종모델의 성능확인
tree_final = DecisionTreeClassifier(min_samples_leaf = 6, min_samples_split = 35,\
                                    max_depth = 13, random_state=1234, )
tree_final.fit(df_train_x_DT, df_train_y_DT)
```

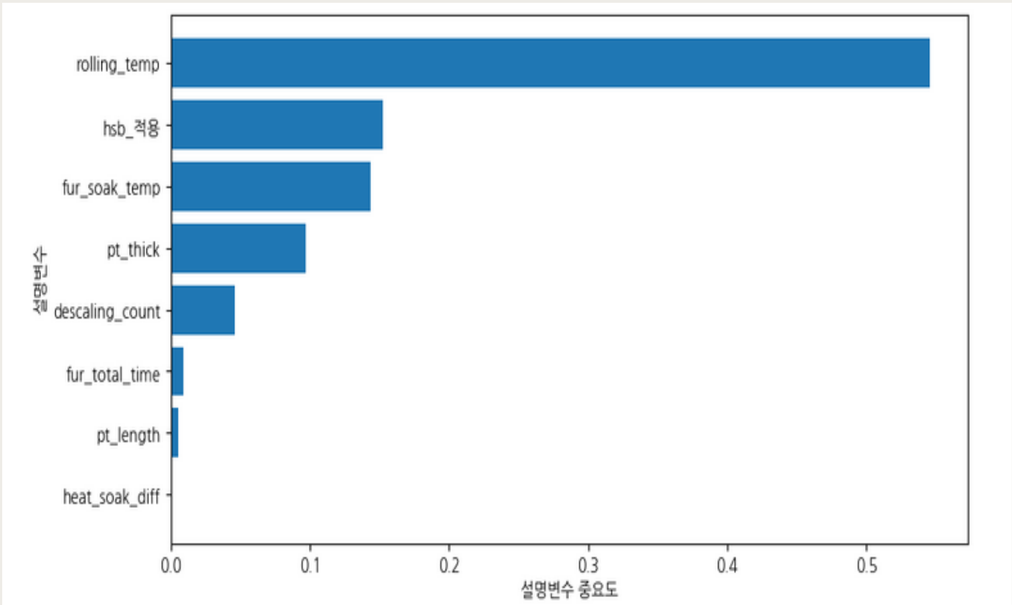
▲ Grid_search를 통해 결정한 HyperParameter 값

Confusion Matrix

```
Test Confusion matrix:
[[206   0]
 [  2  92]]
              precision    recall  f1-score   support

           0      0.990     1.000     0.995       206
           1      1.000     0.979     0.989        94

    accuracy                         0.993       300
   macro avg      0.995     0.989     0.992       300
weighted avg      0.993     0.993     0.993       300
```

- Confusion Matrix 상의 모든 지표에서 높은 성능 보임
- Scale 불량과 정상을 잘 나누는 변수가 있을 것으로 예상

중요 변수 확인



- rolling_temp, hsb_적용, fur_soak_temp의 설명변수 중요도 높음
- 특히 rolling_temp의 중요도가 매우 높음

06 모델링

Random Forest

HyperParameter 결정

```
# 7. 최종모델의 성능확인
tree_final = RandomForestClassifier(max_depth=11, min_samples_leaf=10,
                                    min_samples_split=20, n_estimators = 100, random_state=1234 )
tree_final.fit(df_train_x_RF, df_train_y_RF)
```

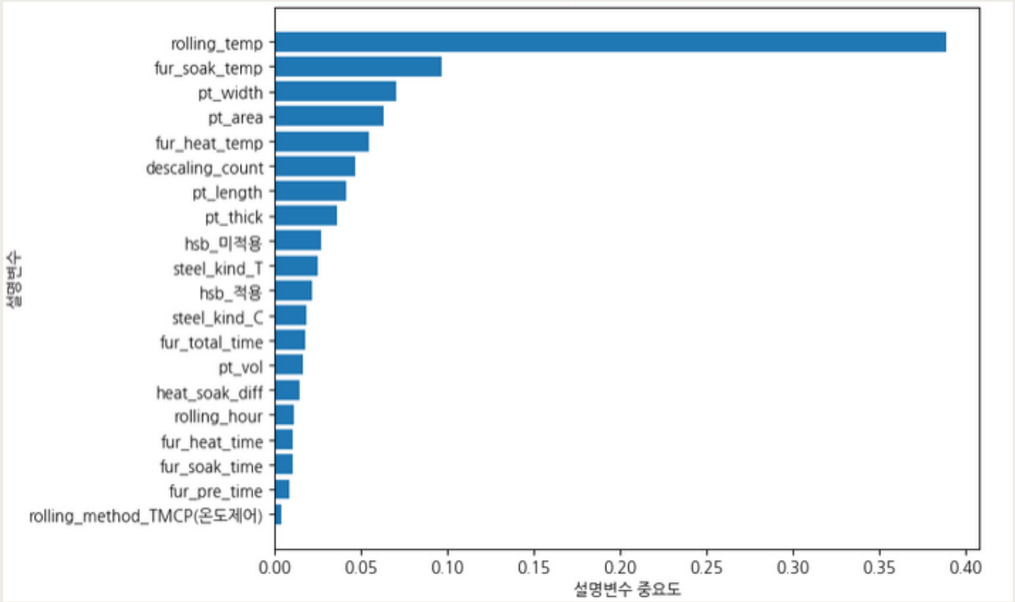
▲ Grid_search를 통해 결정한 HyperParameter 값

Confusion Matrix

```
Test Confusion matrix:
[[206   0]
 [ 26  68]]
              precision    recall  f1-score   support

           0       0.888     1.000     0.941       206
           1       1.000     0.723     0.840        94

    accuracy                          0.913       300
   macro avg       0.944     0.862     0.890       300
weighted avg       0.923     0.913     0.909       300
```

Decision Tree 모델에 비해 성능이 전체적으로 감소

중요 변수 확인



- rolling_temp, fur_soak_temp, 슬라브 규격의 중요도 높음
- 마찬가지로 rolling_temp의 중요도가 매우 높음
- 슬라브의 규격의 중요도가 높으나, 모델링 후 영향력이 적은 결과가 나타날 것으로 예상

06 모델링

Gradient Boosting

HyperParameter 결정

```
# 7. 최종모델의 성능확인
tree_final = GradientBoostingClassifier(min_samples_leaf = 10, min_samples_split = 21,\
                                        max_depth = 10, random_state=1234, learning_rate = 0.1, n_estimators = 300)
tree_final.fit(df_train_x_GB, df_train_y_GB)
```

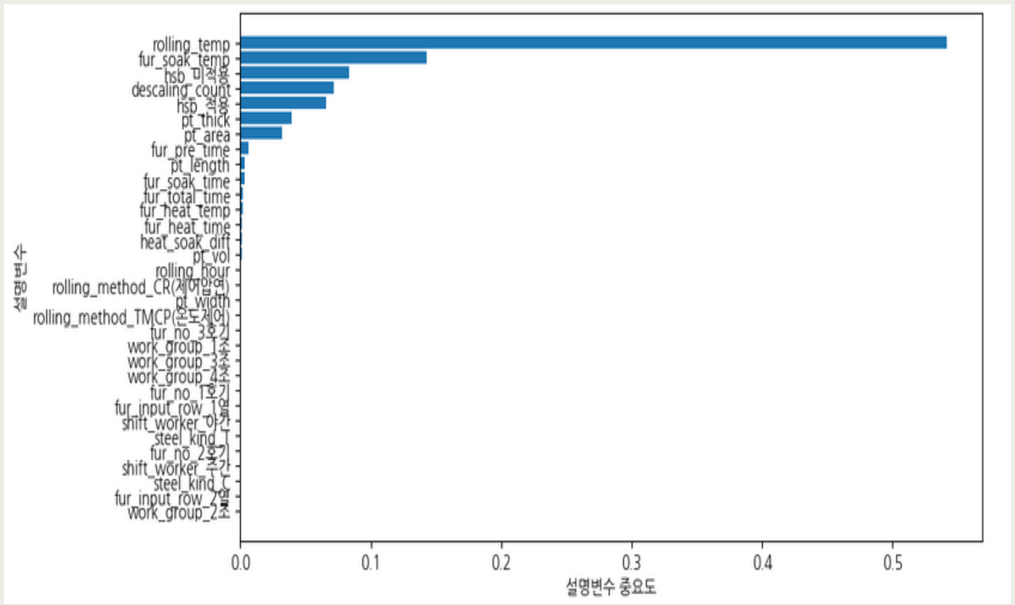
▲ Grid_search를 통해 결정한 HyperParameter 값

Confusion Matrix

```
Test Accuracy: 0.990

Test Confusion matrix:
[[ 92   2]
 [  1 205]]
              precision    recall  f1-score   support

          불량       0.989     0.979     0.984        94
          양품       0.990     0.995     0.993       206

    accuracy                          0.990       300
   macro avg       0.990     0.987     0.988       300
weighted avg       0.990     0.990     0.990       300
```

- 정확도는 Decision Tree보다 낮으나, F1-score는 높음
- 불량 데이터가 7:3으로 불균형하므로 F1-score로 평가하는 것이 적절한 접근이라고 판단

중요 변수 확인



rolling_temp, fur_soak_temp, hsb 미적용 순으로 중요도 높음

이전 모델들의 설명변수 중요도 분석 결과와 동일하게,
rolling_temp는 다른 변수들에 비해 월등히 중요도가 높음

=> 가장 큰 영향을 주는 변수로 판단

06 모델링

XGBoosting

HyperParameter 결정

```
# 7. 최종모델의 성능확인
tree_final = XGBClassifier(max_depth = 6, learning_rate = 0.1, n_estimators = 50, random_state=1234)
tree_final.fit(df_train_x_XG, df_train_y_XG)
```

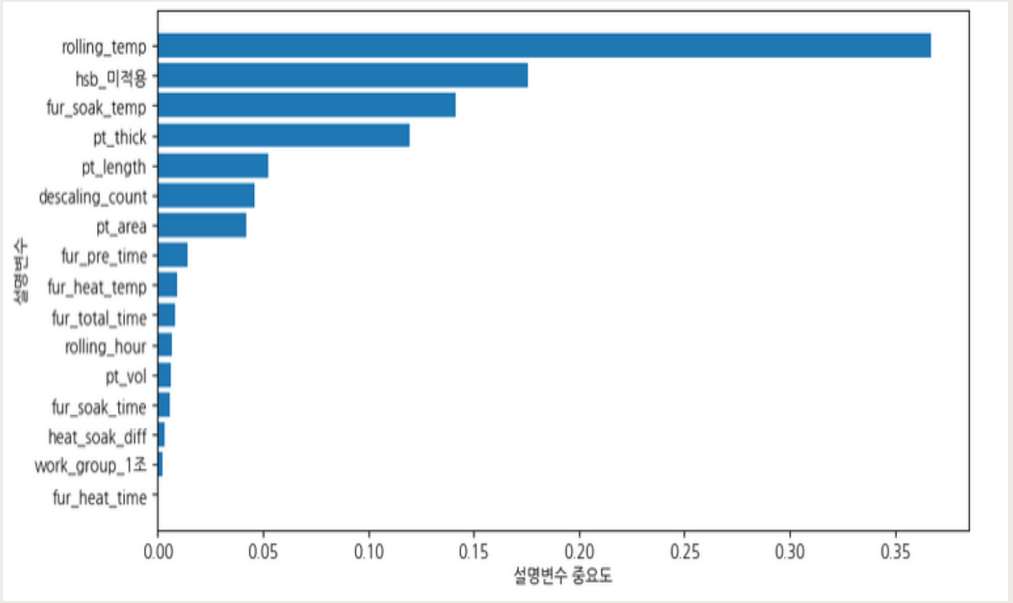
▲ Grid_search를 통해 결정한 HyperParameter 값

Confusion Matrix

```
Test Confusion matrix:
[[206   0]
 [  1  93]]
              precision    recall  f1-score   support

           0      0.995     1.000     0.998       206
           1      1.000     0.989     0.995        94

    accuracy                          0.997       300
   macro avg      0.998     0.995     0.996       300
weighted avg      0.997     0.997     0.997       300
```

- 정확도, F1-score 모두 Gradient Boosting보다 높음
- 변수들의 중요도가 달라졌을 것으로 예상

중요 변수 확인



- rolling_temp, fur_soak_temp, hsb_미적용의 중요도가 높음
- 다만 타 모델들과 달리 rolling_temp의 중요도 떨어짐

06 모델링

SVM

HyperParameter 결정

```
# 9. 최종모델의 성능확인
svc_final = SVC(gamma=0.1, C=10, random_state=1234)
svc_final.fit(df_train_x_SVM_scaled, df_train_y_SVM)
```

▲ Grid_search를 통해 결정한 HyperParameter 값

Confusion Matrix

```
Test Confusion matrix:
[[192  14]
 [ 23  71]]
              precision    recall  f1-score   support

           0      0.893     0.932     0.912       206
           1      0.835     0.755     0.793        94

    accuracy                          0.877       300
   macro avg      0.864     0.844     0.853       300
weighted avg      0.875     0.877     0.875       300
```

Tree 모델들에 비해 성능이 많이 떨어진다.

성능에 대한 생각

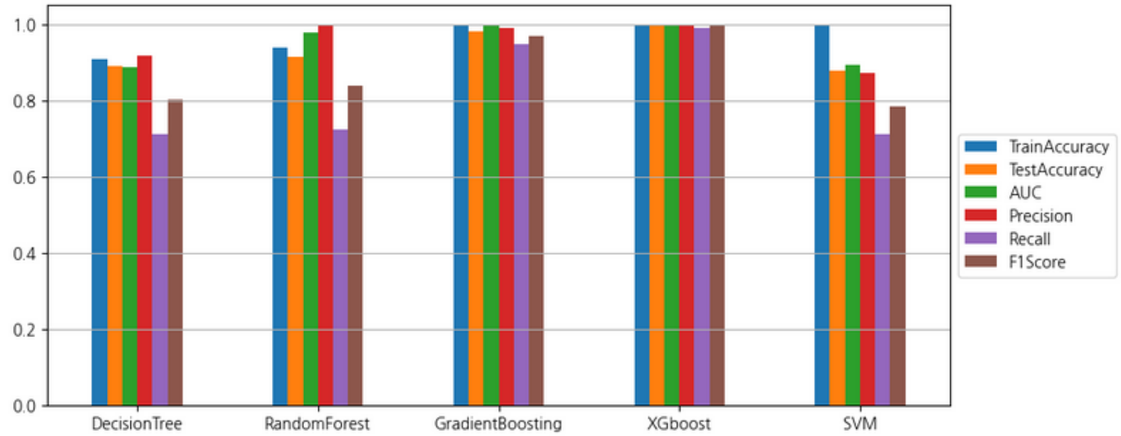
- Deep Learning이 발전하기 전까지 SVM은 우수한 성능의 모델로 평가받던 알고리즘
- 그러나 본 과제의 데이터 규모는 빅데이터로 보기에는 **다소 적은 수준의 데이터셋**

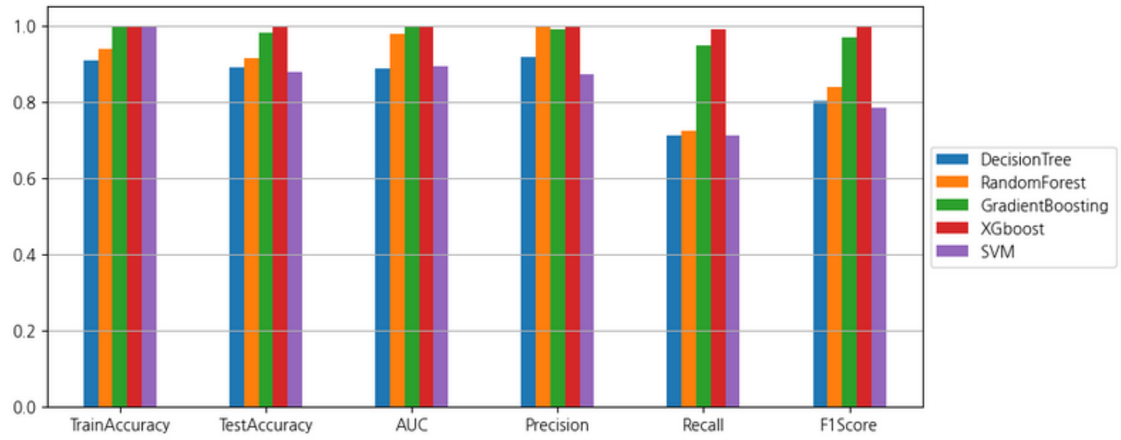
이에 따라 SVM은 **데이터 수의 한계**로 인해 **타 모델 대비 성능이 수치적으로 감소**된 것으로 판단

06 모델링

결론: Classification Model 성능 비교

그래프를 이용한 모델 비교





성능에 대한 생각

- 해당 데이터는 클래스 간 비율이 불균형한 특성 지님
- 단순 정확도(Accuracy) 지표는 다수 클래스에 치우친 예측 결과를 높게 평가할 가능성 존재
- 따라서 정밀도(Precision)와 재현율(Recall)을 동시에 고려하는 **F1-score**를 모델 선택 기준으로 사용하는 것이 타당

XGBoost 모델이 F1-Score를 **포함한 모든 성능 지표에서 가장 우수**한 결과를 보였음.
따라서 **스케일 불량 예측 및 개선을 위해서는 XGBoost 모델을 적용하는 것이 가장 적절**

중요변수 제거 후 데이터 분석

rolling_temp, hsb

설명변수 중요도가 매우 높음

데이터 분석

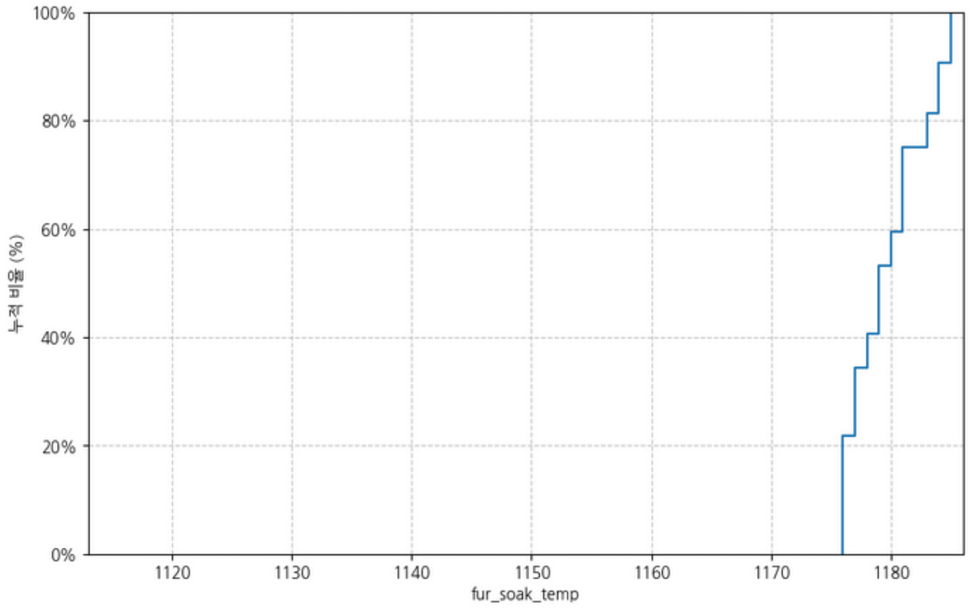
rolling_temp > 1000 인 경우 모두 ‘불량’

hsb = ‘미적용’ 인 경우 모두 ‘불량’

descaling_count = [5,7,9], 즉 홀수인 경우 모두 ‘불량’

```python
# 제거하는 변수들
descaling_list = [5, 7, 9]
condition_to_remove = (
    (df_raw['rolling_temp'] > 1000) |
    (df_raw['hsb'] == '미적용') |
    (df_raw['descaling_count'].isin(descaling_list))
```

중요 변수
제거 후 분석



불량 원인:

rolling_temp가 1000°C 초과이면 모두 불량

descaling 횟수가 홀수이면 모두 불량

hsb 미적용 시 모두 불량

세 조건에 해당하는 데이터 제거

fur_soak_temp가 높은 상위 32개만 불량!

최종 결론:

01 rolling_temp는 950°C(-5%반영) 미만이도록 Monitoring에 집중

02 Descaling 횟수는 짝수로만 진행

03 HSB 공정을 반드시 진행

위 세 조건을 필수적으로 지키고,

가열로 균열대 온도를 1175 °C 이하로 유지

하도록 하는 것이 핵심

토의사항

불량과 인과관계가 있는 3가지 변수에 대한 토의

Rolling Temp의 경우는 다루기가 어려울 수 있음

하지만, HSB 적용과 Descaling 횟수는 자동설비화 된 공장에서 Setting으로 설정 가능

그렇다면 왜 **HSB를 미적용**하고, **Descaling 횟수를 홀수로 설정**했을까?

08 결론 및 토의

토의사항

불량과 인과관계가 있는 3가지 변수에 대한 토의

1

- HSB도 결국 공정이기에 필연적으로 비용이 발생하고 공정 부하도 증가
- 따라서 Scale에 대해서 신경 쓰지 않는 제품(Ex. 2차 가공 후 만들어진 맨홀)이라면, 고객사에서 hsb를 생략하고 상대적으로 저렴한 가격에 매입이 가능할 것으로 예상
- 포스코에서는 'HSB비용과 공정부하에 드는 부수적 비용 > 정상 가격 – 할인된 가격'을 만족해야만 판매할 것

2

'이전에 쌓인 데이터를 기반으로 Shot Blasting 공정에서 Scale 제거될 수 있는 경우, HSB를 생략할 수 있을 것'

해당 방식을 채택함으로써 공정 비용이 감소할 것으로 예상

3

- '지난 10월 21일에 견학한 후판 공장에서 Descaling 횟수가 홀수인 경우는 불가능한 구조였다고 생각'
- 하지만 데이터상으로는 홀수인 경우가 존재
- 다시 공장을 가볼 수 없어서 확인할 수 없으나, 데이터 기반의 'Scale Zero'를 이루기 위해서는 필수적으로 짝수 횟수만 실행해야 할 것으로 판단

Lesson Learned

01 현상 이해 부족으로 인한 애로사항

Hsb가 미적용인 경우 모두 불량품인 것을 확인하였는데, 공정 과정의 배경 지식 부족으로 인해 단순히 데이터상의 분석밖에 진행하지 못한 점

fur_ex_temp(가열로 추출 온도)에 대한 적절한 계산식의 부재로 인해 해당 변수에 대한 분석을 진행하지 못한 점

-초반에는 제품 규격을 drop했으나, 후반에 이르러서야 해당 변수의 중요성을 제고하고 도메인 지식 확보의 필요성을 느끼게 되었음

02 종합실습1에서의 피드백

전처리 과정 통일, 체크리스트를 활용한 작업 분담, 전반적인 분석 계획의 상세 수립 등, 이전 프로젝트에서 느꼈던 분업의 어려움이 상당수 개선