# Scale 불량 영향 인자 도출 및 분류 모델 개발 및 성능 향상방안

청년 AI·빅데이터 아카데미 21기

BigData 종합실습2
B4 권혁준

# Contents

최근들어 선박제조에 주로 사용되는 후판 제품의 "Scale 불량 급증" 이라는 이슈가 발생했다. 그 원인을 분석해 본 결과, 압연흠, Scratch 등 다양한 불량이 발생했으나 특히 압연공장에서 Scale 불량이 급증한 것을 확인할 수 있었다. 그래서 수집된 데이터를 활용하여 다양한 분석을 통해 **불량의 근본 원인을 찾고 불량 예측 및 개선 기회를 도출**하고자 한다.

| 변수 | 변수 설명 |
|---|---|
| plate_no | Plate No |
| rolling_date | 작업시각 |
| scale | Scale불량 |
| spec_long | 제품 규격 |
| spec_country | 제품 규격 기준국 |
| steel_kind | 강종 |
| pt_thick | Plate 두께 |
| pt_width | Plate 폭 |
| pt_length | Plate 길이 |
| hsb | hsb적용여부 |
| fur_no | 가열로 호기 |
| fur_input_row | 가열로 장입열 |
| fur_heat_temp | 가열로 가열대 온도(oC) |
| fur_heat_time | 가열로 가열대 시간(분) |
| fur_soak_temp | 가열로 균열대 온도(oC) |
| fur_soak_time | 가열로 균열대 시간(분) |
| fur_total_time | 가열로 총 시간(분) |
| fur_ex_temp | 가열로 추출온도(oC) |
| rolling_method | 압연방법 |
| rolling_temp | 압연온도(oC) |
| descaling_count | 압연 중 Descaling 횟수 |
| work_group | 작업조 |

**목표변수** scale

**설명변수**

| | | |
|---|---|---|
| plate_no | pt_length | fur_soak_time |
| rolling_date | hsb | fur_total_time |
| spec_long | fur_no | fur_ex_temp |
| spec_country | fur_input_row | rolling_method |
| steel_kind | fur_heat_temp | rolling_temp |
| pt_thick | fur_heat_time | descaling_count |
| pt_width | fur_soak_temp | work_group |

**목표변수인 scale의 핵심영향인자 도출을 통해 분류 정확도가 높은 모델을 개발한다.**

### - 변수의 결측치 확인

```
plate_no          0
rolling_date      0
scale             0
spec_long         0
spec_country      0
steel_kind        0
pt_thick          0
pt_width          0
pt_length         0
hsb               0
fur_no            0
fur_input_row     0
fur_heat_temp     0
fur_heat_time     0
fur_soak_temp     0
fur_soak_time     0
fur_total_time    0
fur_ex_temp       0
rolling_method    0
rolling_temp      0
descaling_count   0
work_group        0
dtype: int64
```

### - 변수의 요약통계량 확인

| | count | mean | std | min | 25% | 50% | 75% | max |
|---|---|---|---|---|---|---|---|---|
| pt_thick | 1000.0 | 26.782 | 18.137570 | 12.0 | 15.00 | 19.0 | 34.00 | 100.0 |
| pt_width | 1000.0 | 2831.900 | 494.081478 | 1800.0 | 2500.00 | 2800.0 | 3100.00 | 4600.0 |
| pt_length | 1000.0 | 36788.200 | 13912.387116 | 7900.0 | 26650.00 | 40400.0 | 49100.00 | 54900.0 |
| fur_heat_temp | 1000.0 | 1157.245 | 21.245007 | 1103.0 | 1140.00 | 1159.0 | 1173.00 | 1206.0 |
| fur_heat_time | 1000.0 | 85.972 | 26.346297 | 55.0 | 66.00 | 75.0 | 102.25 | 158.0 |
| fur_soak_temp | 1000.0 | 1150.928 | 17.344384 | 1113.0 | 1135.75 | 1156.0 | 1164.00 | 1185.0 |
| fur_soak_time | 1000.0 | 71.720 | 20.602137 | 35.0 | 57.75 | 66.0 | 81.00 | 145.0 |
| fur_total_time | 1000.0 | 238.589 | 38.194828 | 165.0 | 210.00 | 230.0 | 263.00 | 362.0 |
| fur_ex_temp | 1000.0 | 1150.928 | 17.344384 | 1113.0 | 1135.75 | 1156.0 | 1164.00 | 1185.0 |
| rolling_temp | 1000.0 | 934.637 | 96.598015 | 0.0 | 893.75 | 948.0 | 991.00 | 1078.0 |
| descaling_count | 1000.0 | 8.557 | 1.604158 | 5.0 | 8.00 | 9.0 | 10.00 | 10.0 |

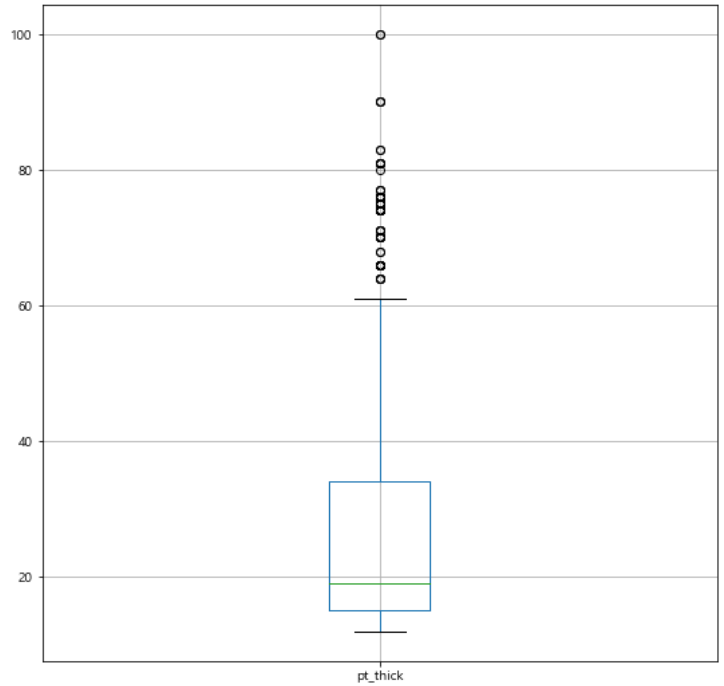
**요약 통계량 확인 결과, 압연온도 최소값이 0인것은 이상치일 확률이 높다.**

- 연속형 변수 정리 및 이상치 처리

## 변수: pt_thick, pt_width, pt_length

- spec에 따라 pt_thick, pt_width, pt_length이 달라지므로 계산후

1. 표준규격에 맞게 이상치 검사를 실시
2. 표준규격에 근거하여 두께는 100mm초과, 넓이는 4000mm초과, 길이는 IQR로 이상치 처리




pt_thick

- 연속형 변수 정리 및 이상치 처리

## 변수: pt_thick, pt_width, pt_length

- spec에 따라 pt_thick, pt_width, pt_length이 달라지므로 계산후

1. 표준규격에 맞게 이상치 검사를 실시
2. 표준규격에 근거하여 두께는 100mm초과, 넓이는 4000mm초과, 길이는 IQR로 이상치 처리

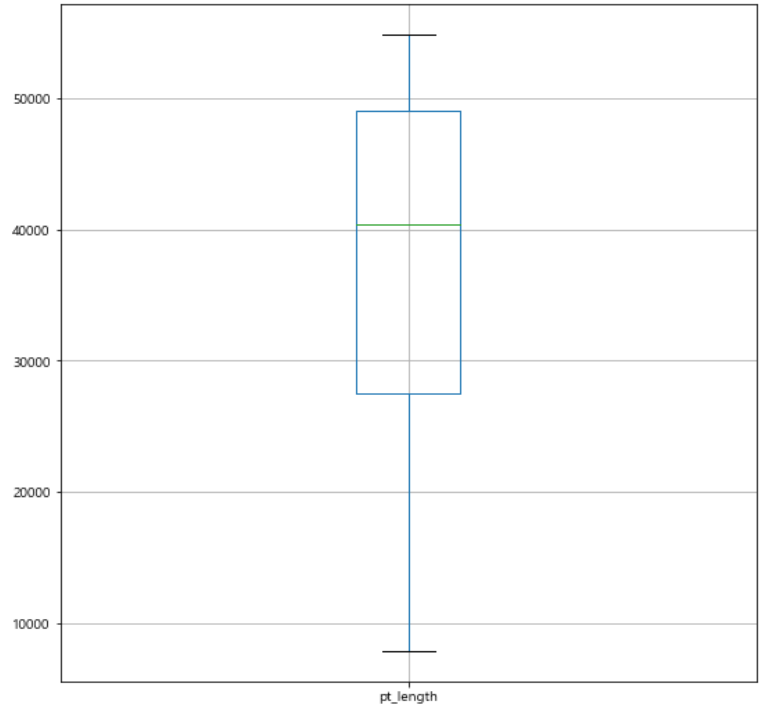

pt_width

- 연속형 변수 정리 및 이상치 처리

## 변수: pt_thick, pt_width, pt_length

- spec에 따라 pt_thick, pt_width, pt_length이 달라지므로 계산후

1. 표준규격에 맞게 이상치 검사를 실시
2. 표준규격에 근거하여 두께는 100mm초과, 넓이는 4000mm초과, 길이는 IQR로 이상치 처리



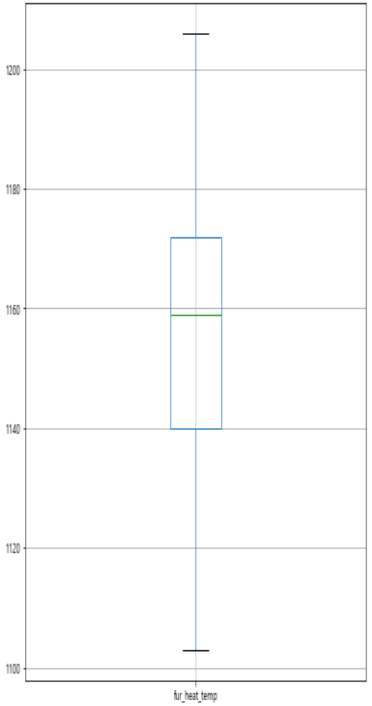

pt_ length

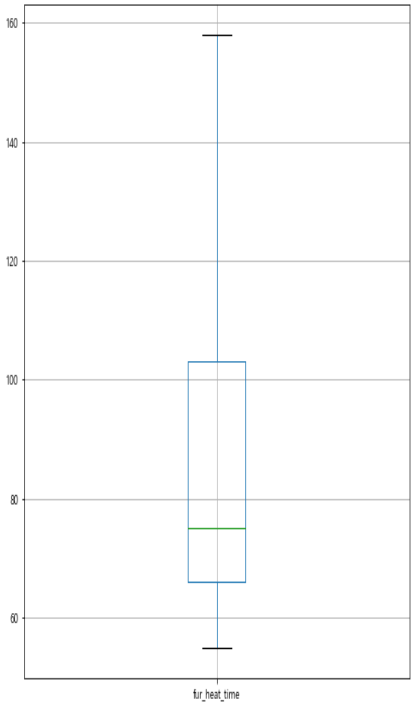
## - 연속형 변수 정리 및 이상치 처리

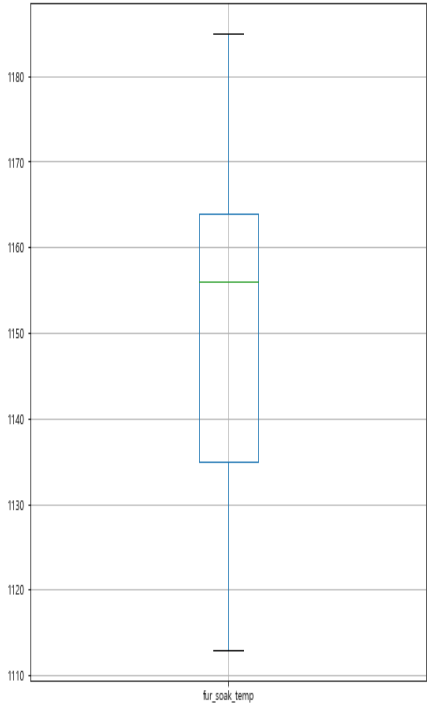
변수: fur_heat_temp, fur_heat_time, fur_soak_temp, fur_soak_time, fur_total_time, fur_ex_temp, rolling_temp

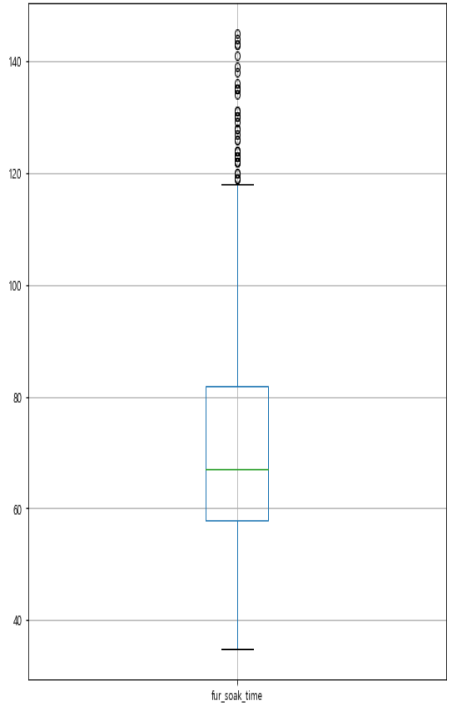
### - 온도, 시간 관련

1. IQR로 이상치 확인 후 판단



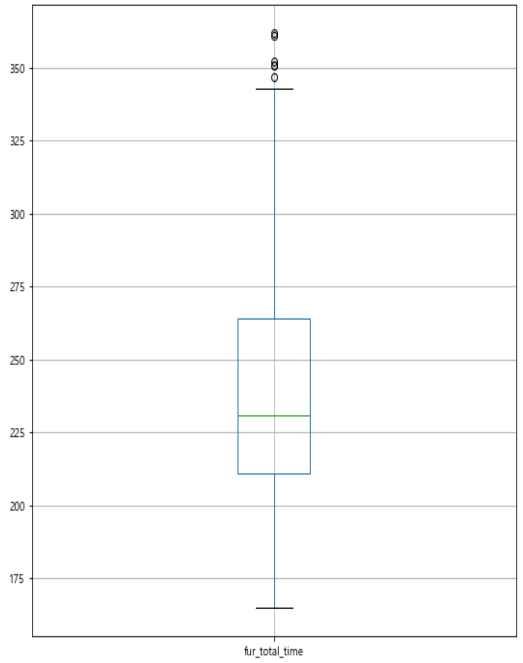

이상치가 존재하지만 연속적으로
분포하고 있기때문에 그대로 사용

## - 연속형 변수 정리 및 이상치 처리

변수: fur_heat_temp, fur_heat_time, fur_soak_temp, fur_soak_time, fur_total_time, fur_ex_temp, rolling_temp

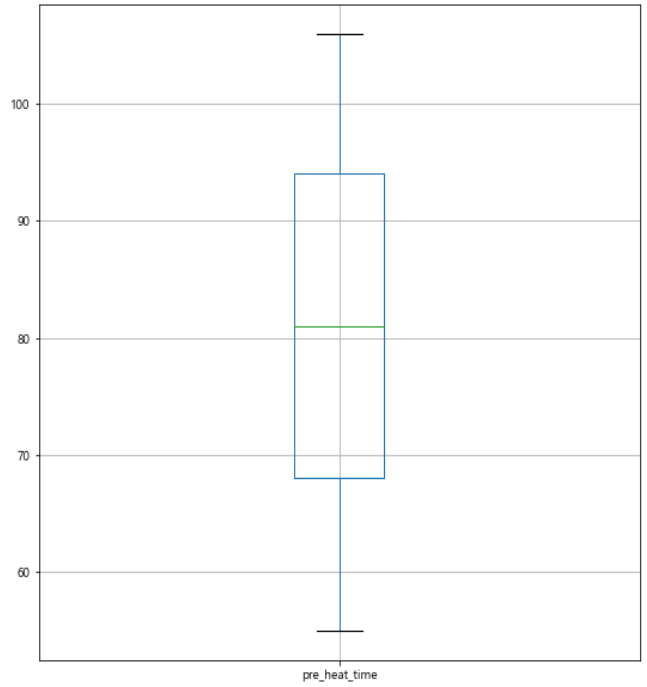
- 온도, 시간 관련

1. IQR로 이상치 확인 후 판단




fur_total_time

total은 예열대 시간 + 가열대시간 + 균열대 시간으로 종속관계라 판단하여 예열대 시간(pre_heat_time)으로 파생변수로 생성함.



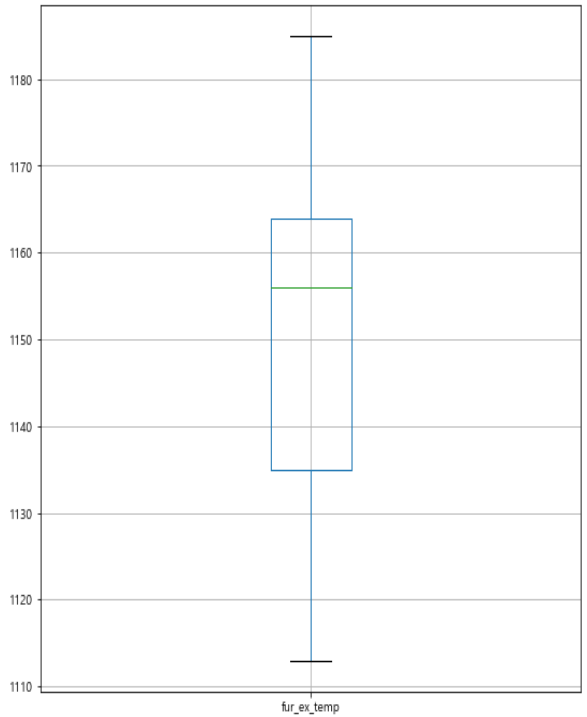

pre_heat_time

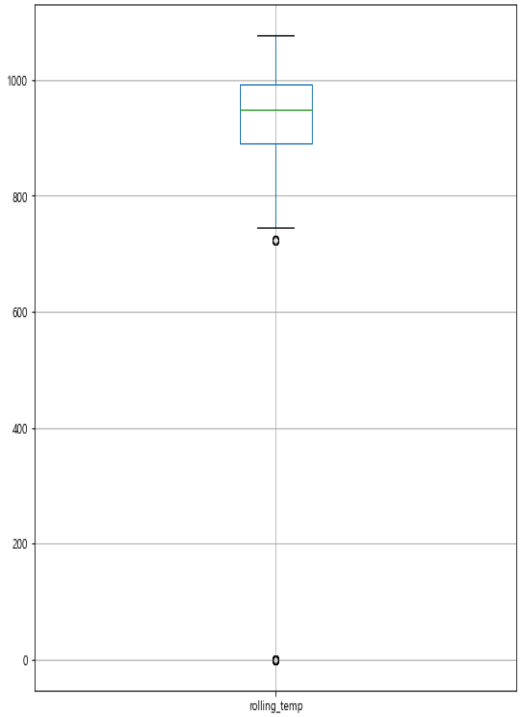
## - 연속형 변수 정리 및 이상치 처리

변수: fur_heat_temp, fur_heat_time, fur_soak_temp, fur_soak_time, fur_total_time, fur_ex_temp, rolling_temp

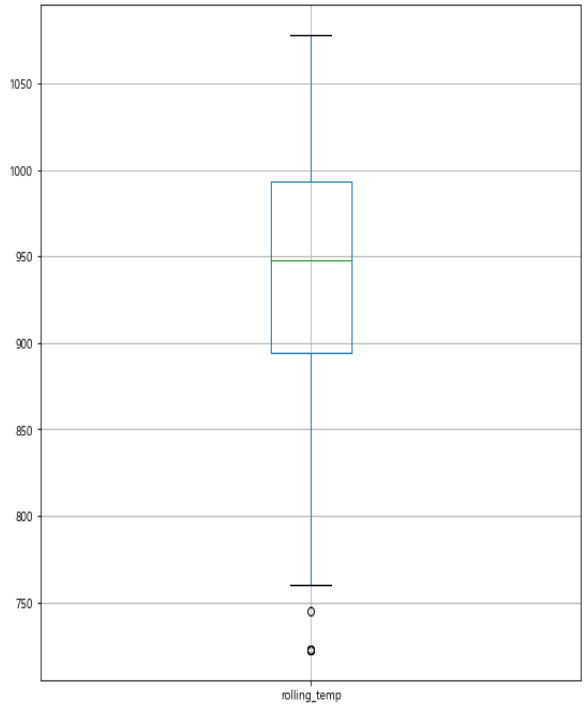
- 온도, 시간 관련

1. IQR로 이상치 확인 후 판단




rolling_temp의
온도가 1000이하 삭제

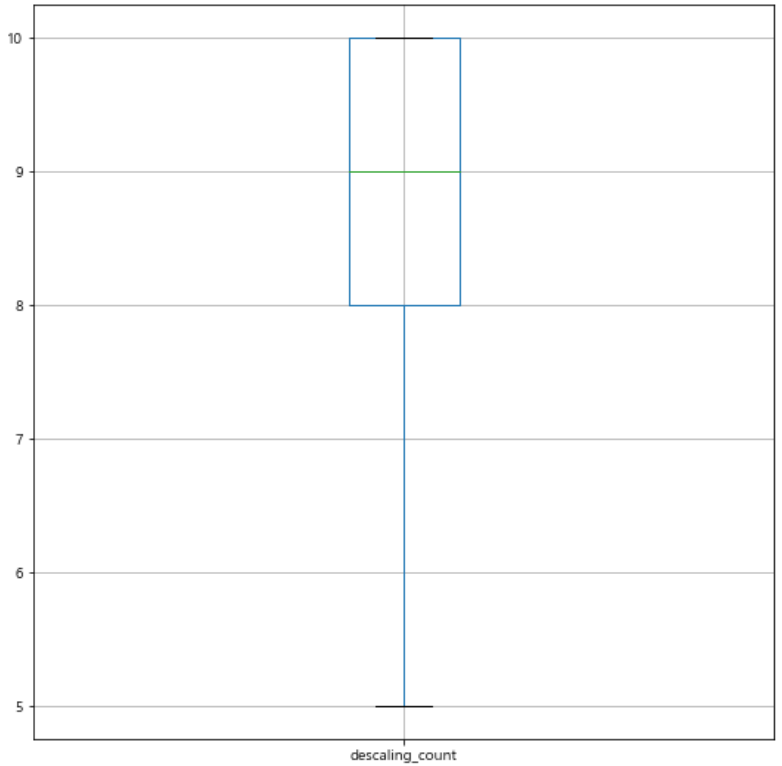



rolling_temp

## - 연속형 변수 정리 및 이상치 처리

변수: descaling_count

1. IQR로 이상치 확인 후 판단




descaling_count

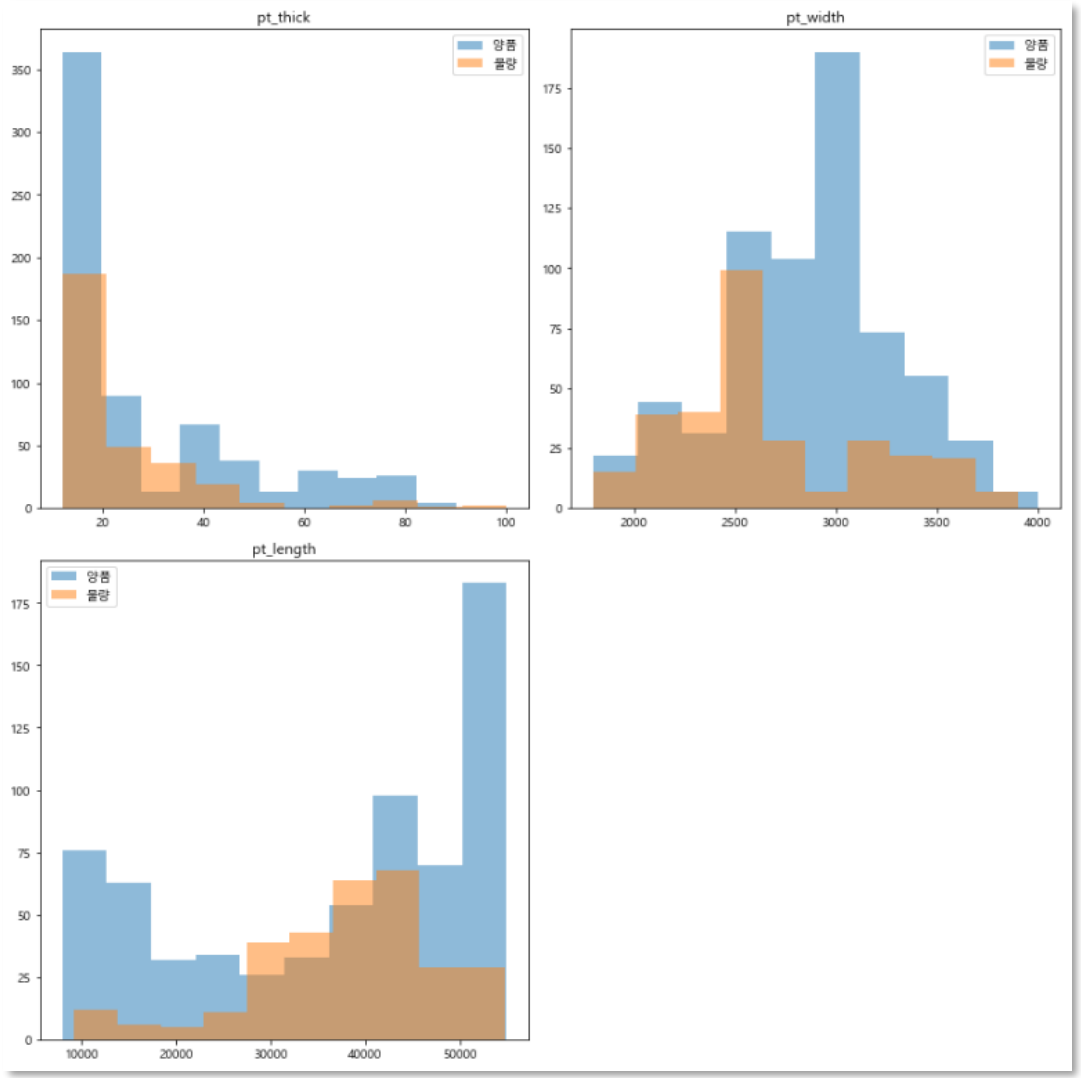
## - 연속형 변수 그래프 분석

### 히스토그램 - 양품/불량 기준(연속형 데이터)

- Plate 관련

변수: pt_thick, pt_width, pt_length

```
def fun_plot_hist(data, var):
    plt.hist(data[data["scale"] == "양품"][var], label = "양품", alpha = 0.5)
    plt.hist(data[data["scale"] == "불량"][var], label = "불량", alpha = 0.5)
    plt.title(var)
    plt.legend()
```




- scale과 폭,두께,길이의 두드러진 연관성이 없다고 판단
- 다른 변수와의 연관성을 더 조사해 볼 필요가 있다고 판단

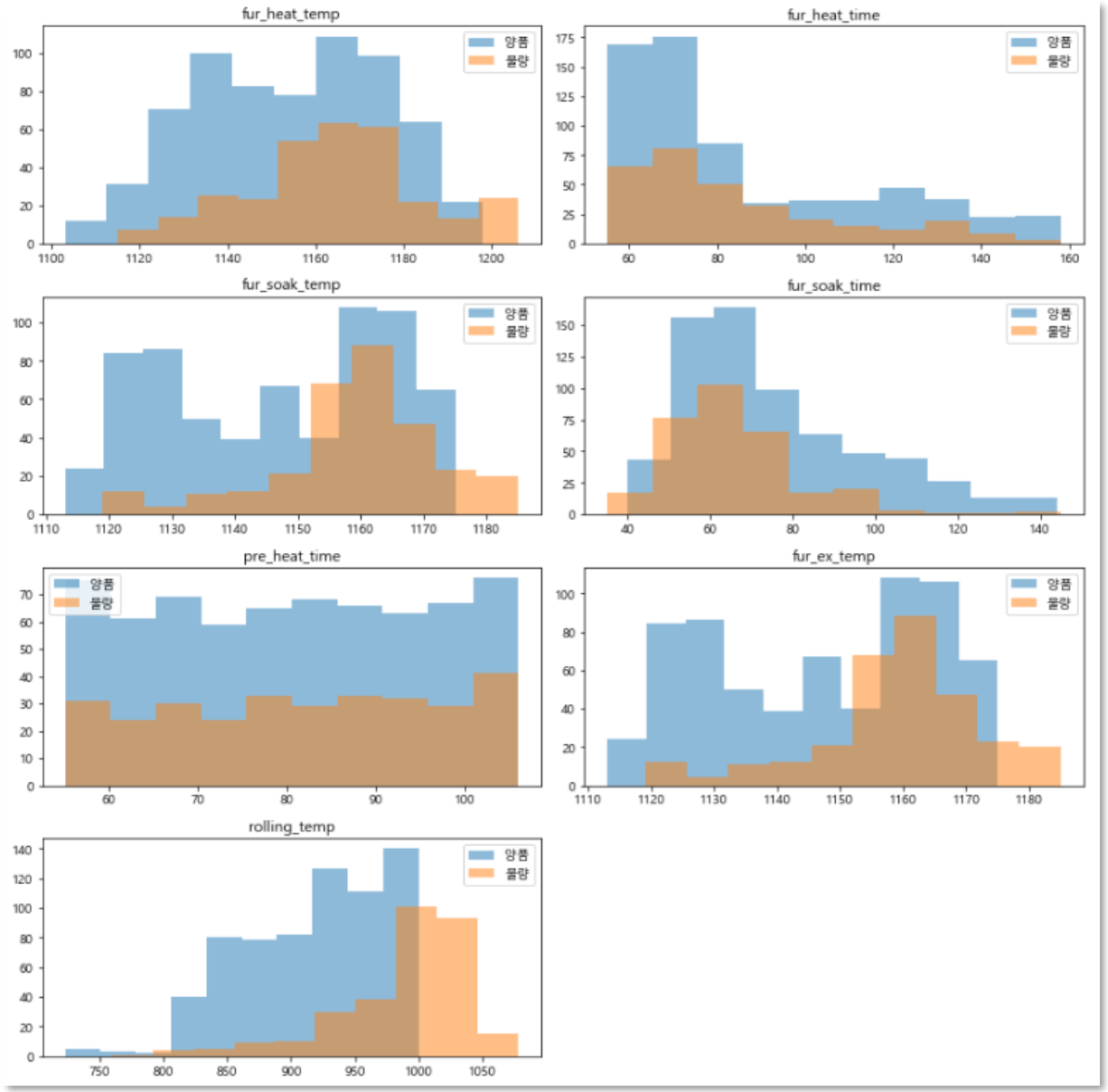
## - 연속형 변수 그래프 분석

### 히스토그램 - 양품/불량 기준(연속형 데이터)

- 가열로 시간, 온도 관련

변수: fur_heat_temp, fur_heat_time, fur_soak_temp, fur_soak_time, pre_heat_time, fur_ex_temp, rolling_temp

```
def fun_plot_hist(data, var):
    plt.hist(data[data["scale"] == "양품"][var], label = "양품", alpha = 0.5)
    plt.hist(data[data["scale"] == "불량"][var], label = "불량", alpha = 0.5)
    plt.title(var)
    plt.legend()
```




- fur_ex_temp와 rolling_temp의 경우 더 온도가 더 높을 수록 불량율이 더 높아지는 것을 확인

- fur_soak_temp와 fur_ex_temp의 모양이 똑같다고 판단하여 fur_soak_temp를 제외

- pre_heat_time의 경우 양품.불량이 고르게 분포하고 있으므로 의미가 없다고 판단하여 제외

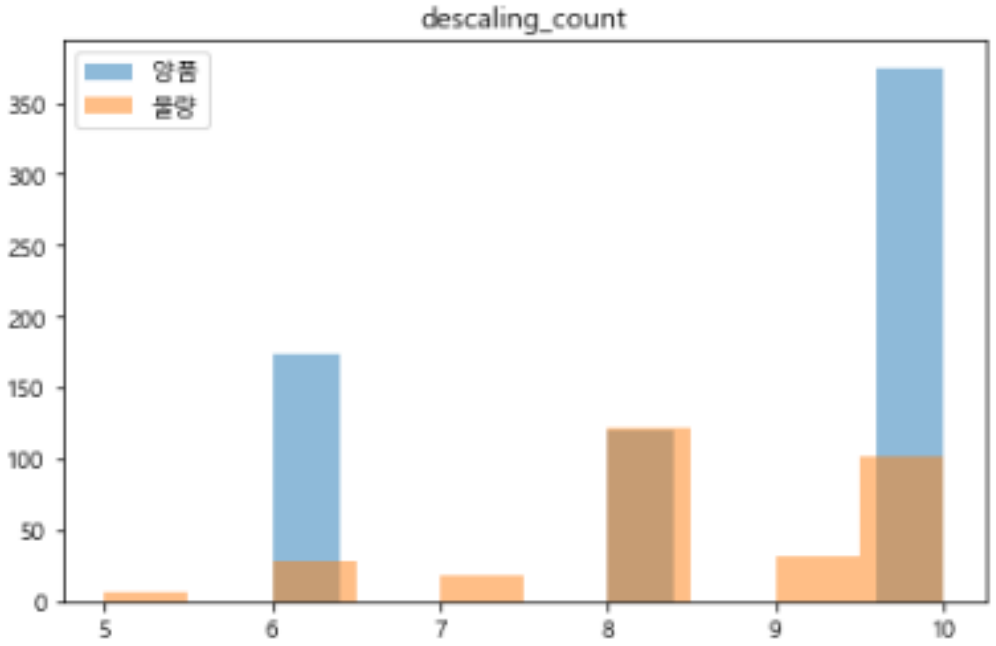
## - 연속형 변수 그래프 분석

### 히스토그램 - 양품/불량 기준(연속형 데이터)

- 압연descaling 횟수

변수: descaling_count

```
def fun_plot_hist(data, var):
    plt.hist(data[data["scale"] == "양품"][var], label = "양품", alpha = 0.5)
    plt.hist(data[data["scale"] == "불량"][var], label = "불량", alpha = 0.5)
    plt.title(var)
    plt.legend()
```




- 횟수가 많아질수록 불량율이 적어지지 않으므로 제외

## - 범주형 변수 그래프 분석

### 빈도분석 (범주형데이터)

변수: 가열로 작업순번, 가열로 호기, 작업조, 제품규격기준국

```
def fun_print_crosstab(data, var):
    return pd.crosstab(data["scale"], data[var], margins=True)
```

| fur_input_row<br>scale | 1열 | 2열 | All |
|---|---|---|---|
| 불량 | 163 | 143 | 306 |
| 양품 | 332 | 337 | 669 |
| All | 495 | 480 | 975 |

해당 범주형 변수는 큰 차이가 없다고 판단하여 제외

| fur_no<br>scale | 1호기 | 2호기 | 3호기 | All |
|---|---|---|---|---|
| 불량 | 100 | 91 | 115 | 306 |
| 양품 | 228 | 225 | 216 | 669 |
| All | 328 | 316 | 331 | 975 |

가열로 호기마다 차이가 없으므로 제외

| work_group<br>scale | 1조 | 2조 | 3조 | 4조 | All |
|---|---|---|---|---|---|
| 불량 | 108 | 82 | 66 | 50 | 306 |
| 양품 | 168 | 192 | 149 | 160 | 669 |
| All | 276 | 274 | 215 | 210 | 975 |

해당 범주형 변수는 큰 차이가 없다고 판단하여 제외

| spec_country<br>scale | 공통 | 독일 | 미국 | 영국 | 일본 | 프랑스 | 한국 | All |
|---|---|---|---|---|---|---|---|---|
| 불량 | 3 | 13 | 42 | 74 | 97 | 29 | 48 | 306 |
| 양품 | 33 | 73 | 115 | 167 | 88 | 112 | 81 | 669 |
| All | 36 | 86 | 157 | 241 | 185 | 141 | 129 | 975 |

일본산일 경우 불량이 양품보다 많음.
일본,한국,영국순으로 불량율이 높음

## - 범주형 변수 그래프 분석

### 빈도분석 (범주형데이터)

변수: 제품 규격, 강종, 작업조, hsb(hot scale braker), rolling_method

```
def fun_print_crosstab(data, var):
    return pd.crosstab(data["scale"], data[var], margins=True)
```

| spec_long<br>scale | A131-DH36TM | A283-C | A516-60 | A709-36 | AB/A | AB/AH32 | AB/B | AB/EH32-TM | AB/EH36-TM | API-2W-50T | ... | NV-A36-TM | NV-B | NV-D32-TM | NV-D36-TM | NV-E32-TM | NV-E36-TM | PILAC-BT33 | SA283-C |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 불량 | 1 | 5 | 2 | 0 | 12 | 8 | 2 | 0 | 2 | 0 | ... | 0 | 2 | 0 | 1 | 0 | 0 | 2 | 10 |
| 양품 | 0 | 1 | 1 | 1 | 38 | 23 | 6 | 2 | 30 | 2 | ... | 2 | 1 | 3 | 3 | 2 | 5 | 36 | 11 |
| All | 1 | 6 | 3 | 1 | 50 | 31 | 8 | 2 | 32 | 2 | ... | 2 | 3 | 3 | 4 | 2 | 5 | 38 | 21 |

- 총 67개 종류중에 제품마다 규격차이가 다르다고 생각이 들고, 표본이 적은 제품도 있다고 판단
- 그러므로 해당컬럼을 제외

| steel_kind<br>scale | C | T | All |
|---|---|---|---|
| 불량 | 286 | 20 | 306 |
| 양품 | 449 | 220 | 669 |
| All | 735 | 240 | 975 |

- steel_kind의 경우 C일 때 불량율이 증가함.
- 강종마다 목표변수에 유의미하다고 생각함.

| hsb<br>scale | 미적용 | 적용 | All |
|---|---|---|---|
| 불량 | 47 | 259 | 306 |
| 양품 | 0 | 669 | 669 |
| All | 47 | 928 | 975 |

- hsb 미적용일 경우 불량율 100%의 결과가 나옴
- 그러므로 적용일 때만의 데이터를 가지고 모델을 돌릴필요가 있음

| rolling_method<br>scale | CR(제어압연) | TMCP(온도제어) | All |
|---|---|---|---|
| 불량 | 293 | 13 | 306 |
| 양품 | 525 | 144 | 669 |
| All | 818 | 157 | 975 |

- TMCP인 경우 양품율이 유의미하고 높게 나옴
- 중요한 변수라고 판단

## - 분산분석

### 제품 두께 구간마다 불량율이 차이가 있을 것이다?

#### 제품 두께 구간별 불량률의 차이 검정

```
df_PT_THICK = pd.DataFrame()
df_t_test["scale"] = df['scale']
df_t_test["pt_thick"] = df['pt_thick']
df_PT_THICK_A = df[df['pt_thick'] < 16]
df_PT_THICK_A = df_PT_THICK_A[["scale", 'pt_thick']]

df_PT_THICK_B = df[(df['pt_thick'] > 16) & (df['pt_thick'] < 20)]
df_PT_THICK_B = df_PT_THICK_B[["scale", 'pt_thick']]

df_PT_THICK_C = df[(df['pt_thick'] > 20) & (df['pt_thick'] < 40)]
df_PT_THICK_C = df_PT_THICK_C[["scale", 'pt_thick']]

df_PT_THICK_D = df[df['pt_thick'] > 40]
df_PT_THICK_D = df_PT_THICK_D[["scale", 'pt_thick']]

# 정규성 검정
statistic, p = stats.shapiro(df_PT_THICK_A['scale'])
statistic, p = stats.shapiro(df_PT_THICK_B['scale'])
statistic, p = stats.shapiro(df_PT_THICK_C['scale'])
statistic, p = stats.shapiro(df_PT_THICK_D['scale'])
print("구간 A 정규성 : statistic = {0}, p-value = {1}".format(statistic, p))
print("구간 B 정규성 : statistic = {0}, p-value = {1}".format(statistic, p))
print("구간 C 정규성 : statistic = {0}, p-value = {1}".format(statistic, p))
print("구간 D 정규성 : statistic = {0}, p-value = {1}".format(statistic, p), end='₩n ₩n')
# 등분산성 검정
var_test = stats.levene(df_PT_THICK_A['scale'], df_PT_THICK_B['scale'], df_PT_THICK_C['scale'])
print(var_test,end='₩n ₩n')
# ANOVA
f_result = stats.f_oneway(df_PT_THICK_A['scale'], df_PT_THICK_B['scale'], df_PT_THICK_C['scale'])

f, p = f_result.statistic.round(3), f_result.pvalue
print("One-Way")
print("F 통계량: {}".format(f))
print("p-value: {}".format(p))
```

```
구간 A 정규성 : statistic = 0.3421415090560913, p-value = 4.009254284331164e-23
구간 B 정규성 : statistic = 0.3421415090560913, p-value = 4.009254284331164e-23
구간 C 정규성 : statistic = 0.3421415090560913, p-value = 4.009254284331164e-23
구간 D 정규성 : statistic = 0.3421415090560913, p-value = 4.009254284331164e-23

LeveneResult(statistic=40.53105945229862, pvalue=3.2604823009604536e-17)

One-Way
F 통계량: 49.742
p-value: 1.1140555711636313e-20
```

- 정규성, 등분산성의 검정결과 0.05보다 작으므로 귀무가설이 기각되어 정규성,등분산성을 띄지 않는다.
- 그래도 조건을 만족한다고 가정하여 F분석을 진행해보았다.
- 그럼에도 조건을 만족이 되지 않았기 때문에 그래프 분석의 내용을 바탕으로 모델링을 진행하였다.

## - 카이제곱

### 압연방법과 강종 종류에 따른 카이제곱 독립성 검정

```
# 압연방법과 강종 종류에 따른 카이제곱 독립성 검정
df_b = pd.crosstab(df['rolling_method'],df['steel_kind'])

chi, pval, dof, expected = stats.chi2_contingency(df_b)

print("Chisq: {}".format(chi.round(3)))
print("p_value: {}".format(pval.round(3)))
print("Degree of freedom: {}".format(dof))
print("expected value: {}".format(expected.round()))
```

```
Chisq: 568.264
p_value: 0.0
Degree of freedom: 1
expected value: [[617. 201.]
 [118.  39.]]
```

- 실행결과 p값이 0으로, 유의수준 5%에서 압연방법과 강종 종류의 차이가 있다고 말할 수 있다.
- 그러므로, 압연방법과 강종 종류에 연관성이 있다.
- 그렇지만 압연방법과, 강종 종류의 그래프 분석을 통해 둘 다 유의미한 변수라고 판단하여 남기기로 결정

### hsb적용/비적용과 강종 종류에 따른 카이제곱 독립성 검정

```
# hsb적용/비적용과 강종 종류에 따른 카이제곱 독립성 검정
df_b = pd.crosstab(df['hsb'],df['steel_kind'])

chi, pval, dof, expected = stats.chi2_contingency(df_b)

print("Chisq: {}".format(chi.round(3)))
print("p_value: {}".format(pval.round(3)))
print("Degree of freedom: {}".format(dof))
print("expected value: {}".format(expected.round()))
```

```
Chisq: 0.138
p_value: 0.711
Degree of freedom: 1
expected value: [[ 35.  12.]
 [700. 228.]]
```

- 실행결과 p값이 0.711이므로, 유의수준 5%에서 hsb과 강종 종류의 차이가 없다고 말할 수 있다.
- 그러므로, hsb과 강종 종류에 연관성이 없다.

### hsb적용/비적용과 압연 방법에 따른 카이제곱 독립성 검정

```
# hsb적용/비적용과 압연 방법에 따른 카이제곱 독립성 검정
df_b = pd.crosstab(df['hsb'],df['rolling_method'])

chi, pval, dof, expected = stats.chi2_contingency(df_b)

print("Chisq: {}".format(chi.round(3)))
print("p_value: {}".format(pval.round(3)))
print("Degree of freedom: {}".format(dof))
print("expected value: {}".format(expected.round()))
```

```
Chisq: 0.189
p_value: 0.664
Degree of freedom: 1
expected value: [[ 39.   8.]
 [779. 149.]]
```

- 실행결과 p값이 0.664로, 유의수준 5%에서 hsb과 압연 방법과 차이가 없다고 말할 수 있다.
- 그러므로, hsb와 압연방법에 연관성이 없다.

### - 전체 변수 포함/ 제외

- plate_no: 단순한 번호이므로 제외
- rolling_date: 공장이므로 동일한 환경에서 진행된 것이기 때문에 날짜와 상관없으므로 제외
- scale: 목표변수: 불량/양품
- spec_long: 표본이 적은 것도 있으므로 제외
- spec_country: 나라마다 불량율이 다름
- steel_kind: 강종의 종류 다르므로 포함
- pt_thick 포함
- pt_width 포함
- pt_length 포함
- hsb: 유의미한 차이가 있으므로 포함
- fur_no: 의미없으므로 제외
- 'fur_input_row': 의미없으므로 제외
- 'fur_heat_temp': 포함
- 'fur_heat_time': 포함
- 'fur_soak_temp': fur_ex_temp와 그래프 유형이 비슷하므로 제외
- 'fur_soak_time': 포함
- 'fur_total_time' : pre_heat_time으로 변환 -> 의미없으므로 제외
- 'fur_ex_temp' : 포함
- 'rolling_method' : 유의미하게 나타났으므로 포함
- 'rolling_temp' : 유의미하게 나타났으므로 포함
- 'descaling_count' : 의미없으므로 제외
- 'work_group': 의미없으므로 제외

### - 핵심인자 선정

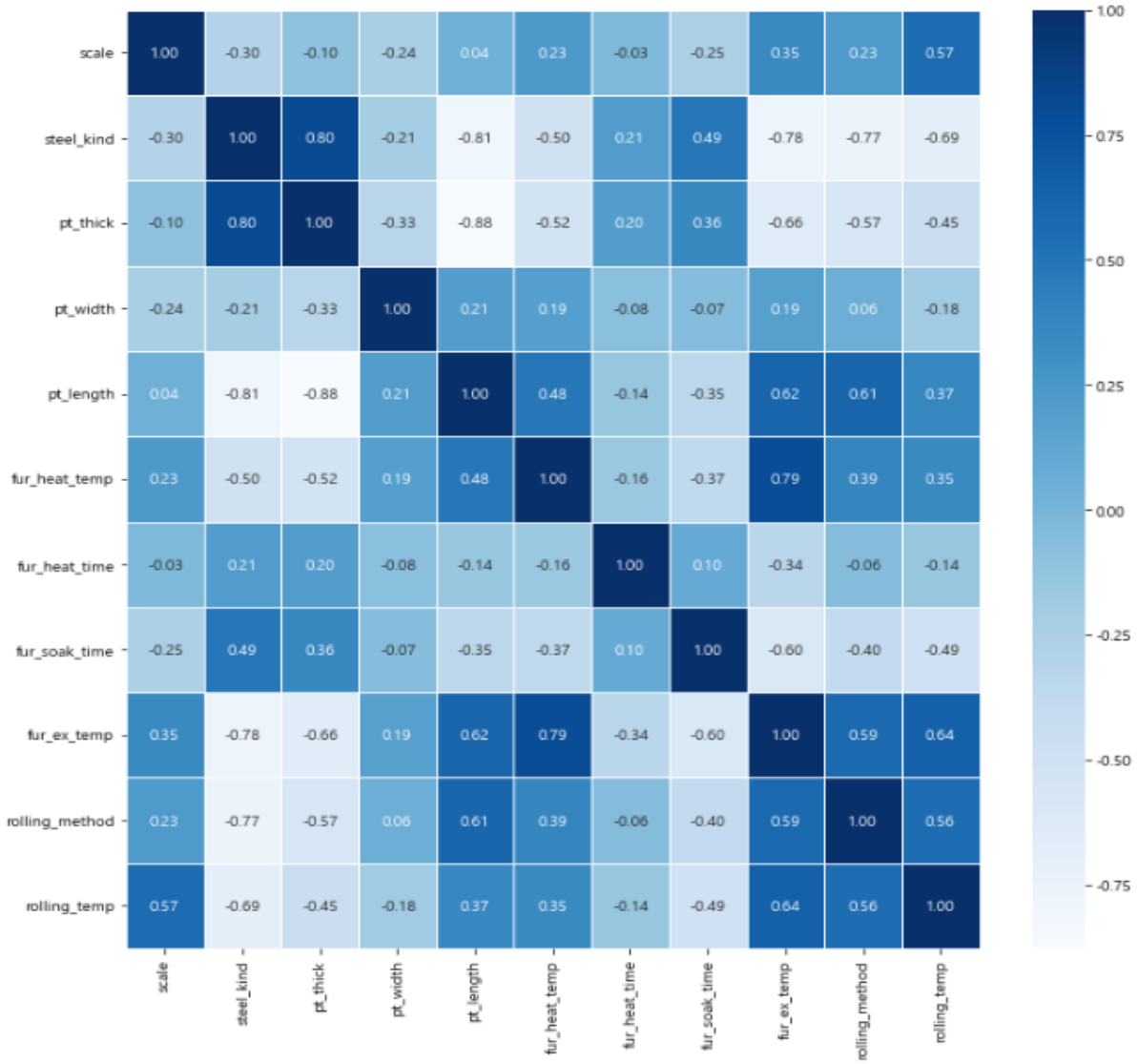
- HSB데이터를 미적용한 데이터의 경우 모두 불량으로 나왔기에 HSB를 핵심인자로 선정한 후, HSB를 적용한 데이터로만 분석을 진행하도록 하겠다.

### - 최종 변수 선정

- 목표 변수: scale
- 설명변수: spec_country, steel_kind, pt_thick, pt_width, pt_length, fur_heat_temp, fur_heat_time, fur_soak_time, fur_ex_temp, rolling_method, rolling_temp
- 설명변수 총 11개

# 05 최종변수 상관관계




scale 1.00 -0.30 -0.10 -0.24 0.04 0.23 -0.03 -0.25 0.35 0.23 0.57
steel_kind -0.30 1.00 0.80 -0.21 -0.81 -0.50 0.21 0.49 -0.78 -0.77 -0.69
pt_thick -0.10 0.80 1.00 -0.33 -0.88 -0.52 0.20 0.36 -0.66 -0.57 -0.45
pt_width -0.24 -0.21 -0.33 1.00 0.21 0.19 -0.08 -0.07 0.19 0.06 -0.18
pt_length 0.04 -0.81 -0.88 0.21 1.00 0.48 -0.14 -0.35 0.62 0.61 0.37
fur_heat_temp 0.23 -0.50 -0.52 0.19 0.48 1.00 -0.16 -0.37 0.79 0.39 0.35
fur_heat_time -0.03 0.21 0.20 -0.08 -0.14 -0.16 1.00 0.10 -0.34 -0.06 -0.14
fur_soak_time -0.25 0.49 0.36 -0.07 -0.35 -0.37 0.10 1.00 -0.60 -0.40 -0.49
fur_ex_temp 0.35 -0.78 -0.66 0.19 0.62 0.79 -0.34 -0.60 1.00 0.59 0.64
rolling_method 0.23 -0.77 -0.57 0.06 0.61 0.39 -0.06 -0.40 0.59 1.00 0.56
rolling_temp 0.57 -0.69 -0.45 -0.18 0.37 0.35 -0.14 -0.49 0.64 0.56 1.00
scale
steel_kind
pt_thick
pt_width
pt_length
fur_heat_temp
fur_heat_time
fur_soak_time
fur_ex_temp
rolling_method
rolling_temp
1.00
0.75
0.50
0.25
0.00
-0.25
-0.50
-0.75

## 로지스틱 회귀분석

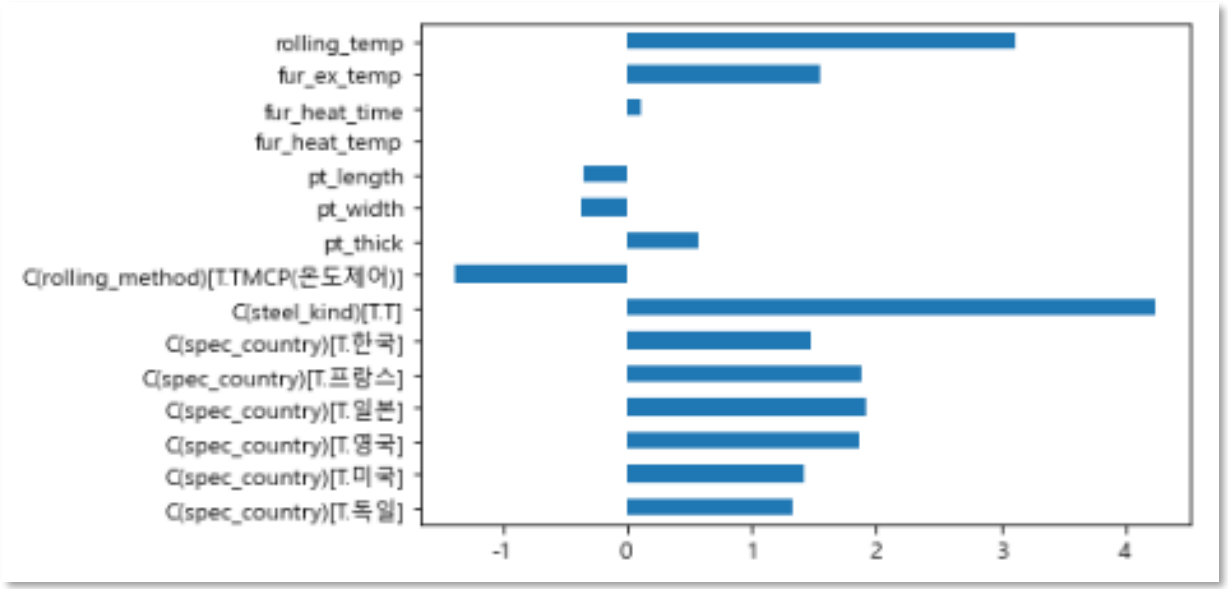
```
Train 예측/분류 결과
Accuracy: 0.899

Confusion Matrix:
[[387  20]
 [ 36 113]]
```

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.915 | 0.951 | 0.933 | 407 |
| 1 | 0.850 | 0.758 | 0.801 | 149 |
| accuracy | | | 0.899 | 556 |
| macro avg | 0.882 | 0.855 | 0.867 | 556 |
| weighted avg | 0.897 | 0.899 | 0.897 | 556 |

```
Test 예측/분류 결과
Accuracy: 0.874

Confusion Matrix:
[[248  14]
 [ 33  77]]
```

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.883 | 0.947 | 0.913 | 262 |
| 1 | 0.846 | 0.700 | 0.766 | 110 |
| accuracy | | | 0.874 | 372 |
| macro avg | 0.864 | 0.823 | 0.840 | 372 |
| weighted avg | 0.872 | 0.874 | 0.870 | 372 |

### 스케일링한 표준화를 한 coef




- 변수 중요도의 경우 stell_kind(강종의 종류), rolling_temp, 국적, fur_ex_temp 순으로 중요변수로 측정되었다.
- 강종의 종류가 T일 수록 양품일 가능성이 높아진다.
- 압연방법이 온도제어일수록 불량일 가능성이 높아진다.

## 의사결정나무

Train 예측/분류 결과
Accuracy: 0.959

Coufusion matrix:
[[407  0]
 [ 23 126]]

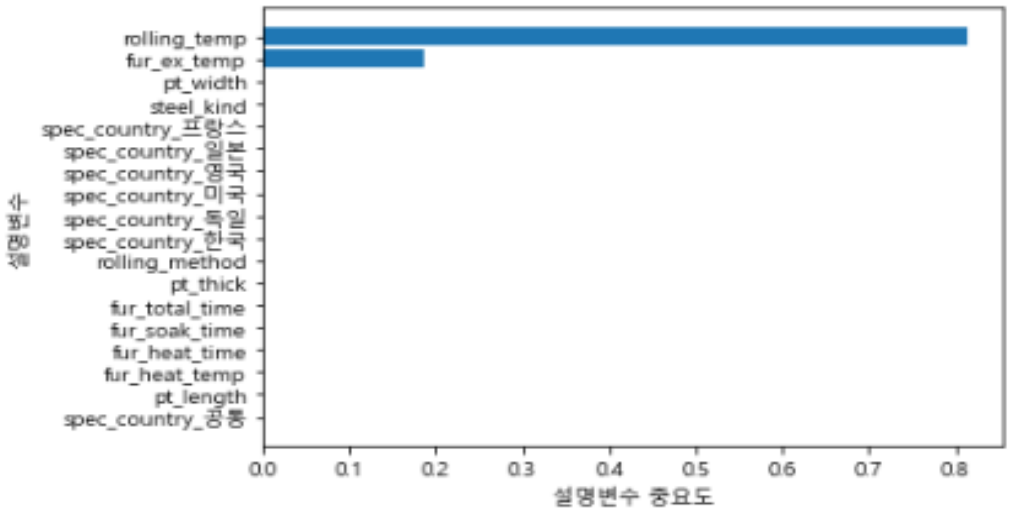
| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.947 | 1.000 | 0.973 | 407 |
| 1 | 1.000 | 0.846 | 0.916 | 149 |
| accuracy | | | 0.959 | 556 |
| macro avg | 0.973 | 0.923 | 0.944 | 556 |
| weighted avg | 0.961 | 0.959 | 0.957 | 556 |

Test 예측/분류 결과
Accuracy: 0.960

Coufusion matrix:
[[262  0]
 [ 15  95]]

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.946 | 1.000 | 0.972 | 262 |
| 1 | 1.000 | 0.864 | 0.927 | 110 |
| accuracy | | | 0.960 | 372 |
| macro avg | 0.973 | 0.932 | 0.949 | 372 |
| weighted avg | 0.962 | 0.960 | 0.959 | 372 |

coef




• gridsearchCV 튜닝 파라미터: 'max_depth': 2, 'min_samples_leaf': 2, 'min_samples_split': 10
•f1 score의 경우, 0일 경우 0.972, 1일경우 0.9270이다.
•test 데이터 기준 정확도는 0.9600이다.

- 중요 설명변수: rolling_temp,fur_ex_temp순으로 중요함

## 랜덤포레스트

Train 예측/분류 결과
Accuracy: 0.959

Coufusion matrix:
[[407 0]
[ 23 126]]

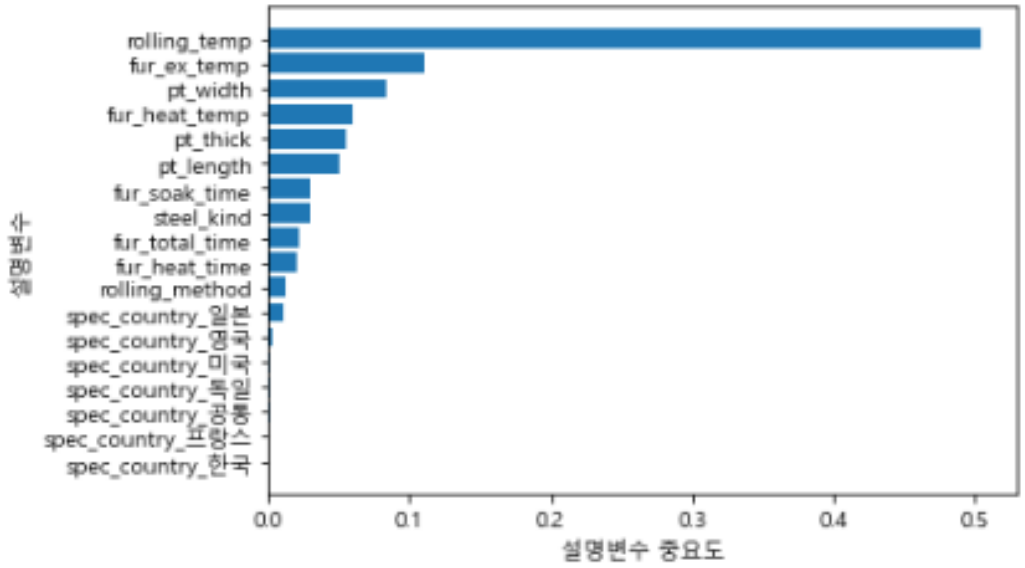
| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.947 | 1.000 | 0.973 | 407 |
| 1 | 1.000 | 0.846 | 0.916 | 149 |
| accuracy | | | 0.959 | 556 |
| macro avg | 0.973 | 0.923 | 0.944 | 556 |
| weighted avg | 0.961 | 0.959 | 0.957 | 556 |

Test 예측/분류 결과
Accuracy: 0.957

Coufusion matrix:
[[262 0]
[ 16 94]]

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.942 | 1.000 | 0.970 | 262 |
| 1 | 1.000 | 0.855 | 0.922 | 110 |
| accuracy | | | 0.957 | 372 |
| macro avg | 0.971 | 0.927 | 0.946 | 372 |
| weighted avg | 0.959 | 0.957 | 0.956 | 372 |

coef



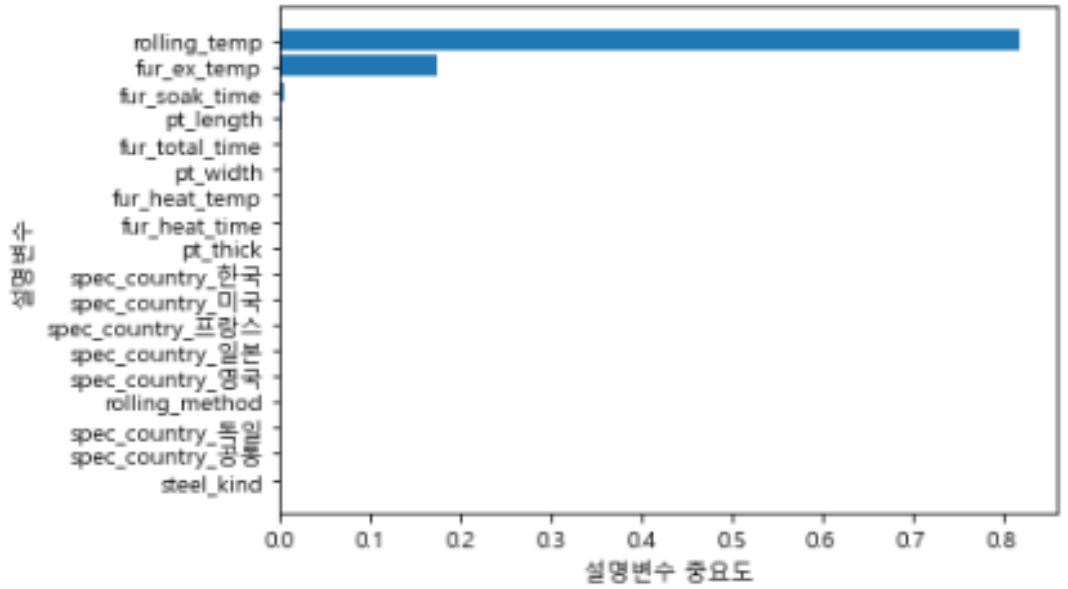

- 중요 설명변수: rolling_temp,fur_ex_temp, pt_width, fur_heat_temp순으로 중요함

• gridsearchCV 튜닝 파라미터'max_depth': 8, 'min_samples_leaf': 5, 'n_estimators'=100
•f1 score의 경우, 0일 경우 0.970, 1일경우 0.9220이다.
•test 데이터 기준 정확도는 0.9570이다

## 그래디언트 부스팅

```
Train 예측/분류 결과
Accuracy: 0.959

Coufusion matrix:
[[407   0]
 [ 23 126]]
              precision    recall  f1-score   support

           0       0.947     1.000     0.973       407
           1       1.000     0.846     0.916       149

    accuracy                          0.959       556
   macro avg       0.973     0.923     0.944       556
weighted avg       0.961     0.959     0.957       556

Test 예측/분류 결과
Accuracy: 0.960

Coufusion matrix:
[[262   0]
 [ 15  95]]
              precision    recall  f1-score   support

           0       0.946     1.000     0.972       262
           1       1.000     0.864     0.927       110

    accuracy                          0.960       372
   macro avg       0.973     0.932     0.949       372
weighted avg       0.962     0.960     0.959       372
```

coef




- 중요 설명변수: rolling_temp,fur_ex_temp, fur_soak_time, pt_length순으로 중요함

• gridsearchCV 튜닝 파라미터: max_depth= 1,min_samples_leaf= 5, n_estimators=100, learning_rate= 0.1
•f1 score의 경우, 0일 경우 0.972, 1일경우 0.927이다.
•test 데이터 기준 정확도는 0.960이다

# SVM

```
Train 예측/분류 결과
Accuracy:  0.914

Coufusion matrix:
[[390  17]
 [ 31 118]]
              precision    recall  f1-score   support

           0      0.926     0.958     0.942       407
           1      0.874     0.792     0.831       149

    accuracy                          0.914       556
   macro avg      0.900     0.875     0.887       556
weighted avg      0.912     0.914     0.912       556
```

```
Test 예측/분류 결과
Accuracy:  0.868

Coufusion matrix:
[[247  15]
 [ 34  76]]
              precision    recall  f1-score   support

           0      0.879     0.943     0.910       262
           1      0.835     0.691     0.756       110

    accuracy                          0.868       372
   macro avg      0.857     0.817     0.833       372
weighted avg      0.866     0.868     0.864       372
```

- gridsearchCV 튜닝 파라미터: gamma = 0.05, C = 1.8
- 정확도는 train: 91.4%, test: 86.8%
- tes 데이터: 1기준 f1 스코어는 75.6%로 다른 모델에 비해 낮음

# KNN

```
Train 예측/분류 결과
Accuracy:  1.000

Coufusion matrix:
[[407   0]
 [  0 149]]
              precision    recall  f1-score   support

           0      1.000     1.000     1.000       407
           1      1.000     1.000     1.000       149

    accuracy                          1.000       556
   macro avg      1.000     1.000     1.000       556
weighted avg      1.000     1.000     1.000       556
```

```
Test 예측/분류 결과
Accuracy:  0.847

Coufusion matrix:
[[244  18]
 [ 39  71]]
              precision    recall  f1-score   support

           0      0.862     0.931     0.895       262
           1      0.798     0.645     0.714       110

    accuracy                          0.847       372
   macro avg      0.830     0.788     0.804       372
weighted avg      0.843     0.847     0.842       372
```

- gridsearchCV 튜닝 파라미터: n_neighbors = 5, metric = "manhattan", weights = "distance"
- 정확도는 train: 100%, test: 84.7%
- 1기준 f1 스코어는 71.4%로 다른 모델에 비해 낮음

## 인공신경망(NN)

```
Train 예측/분류 결과
Accuracy:  0.964

Coufusion matrix:
[[402   5]
 [ 15 134]]
              precision    recall  f1-score   support

           0      0.964     0.988     0.976       407
           1      0.964     0.899     0.931       149

    accuracy                          0.964       556
   macro avg      0.964     0.944     0.953       556
weighted avg      0.964     0.964     0.964       556

Test 예측/분류 결과
Accuracy:  0.863

Coufusion matrix:
[[240  22]
 [ 29  81]]
              precision    recall  f1-score   support

           0      0.892     0.916     0.904       262
           1      0.786     0.736     0.761       110

    accuracy                          0.863       372
   macro avg      0.839     0.826     0.832       372
weighted avg      0.861     0.863     0.862       372
```

- gridsearchCV 튜닝 파라미터: batch_size=50, solver='adam', activation='relu', hidden_layer_sizes=(24, 24)
- 정확도는 train: 96.4%, test: 86.3%
- 1기준 f1 스코어는 76.1%로 다른 모델에 비해 낮음

•결과: 의사결정나무와 그래디언트 부스팅의 성능지표가 거의 동일하게 출력되었다.
- 변수중요도에서는 의사결정나무는 변수중요도가 변수 2개에 집중되어 있음
- 그래디언트 부스팅도 거의 변수 2개에 집중되어 있으나 다른 변수에 조금씩 중요도가 분산되어 있음.

그렇기 때문에 그래디언트 부스팅으로 최종모델을 정하였다

•그래디언트 부스팅
•파라미터: max_depth= 1,min_samples_leaf= 5, n_estimators=100, learning_rate= 0.1
•Test 데이터 기준: f1 score의 경우, 0일 경우 0.972, 1일경우 0.9270이다.
•Accuracy on training set: 0.959
•Accuracy on test set: 0.960

• 중요 설명변수: rolling_temp,fur_ex_temp, fur_soak_time, pt_length순으로 중요함

# 핵심 인자 도출

| 변수 | 변수 설명 | 변수 역할 | 변수 형태 | 분석 제외 사유 | 탐색적 기법 | | 모델링기법 | | | | | | | 총점 | 선정 (사유) |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | 그래프 | 검정 | 로지스틱 회귀분석 | DT | RF | GB | SVM | NN | KNN | | |
| plate_no | Plate No | ID | 범주형 | 목표변수에 영향 x | | | | | | | | | | | |
| rolling_date | 작업시각 | 날짜 | 연속형 | 관련없음 | | | | | | | | | | | |
| scale | Scale불량 | 목표변수 | 범주형 | | | | | | | | | | | | |
| spec_long | 제품 규격 | 설명변수 | 범주형 | 표본이 적어짐 | 어떤 spec의 scale이 많이 발생하는 지 확인 | | | | | | | | | | |
| spec_country | 제품 규격 기준국 | 설명변수 | 범주형 | | histogram, IQR | | | | | | | | | | |
| steel_kind | 강종 | 설명변수 | 범주형 | | 강종별 scale 차이 비교 | 압연방법, hsb에 따른 카이제곱 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | |
| pt_thick | Plate 두께 | 설명변수 | 연속형 | | histogram, IQR | scale과 분산분석 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | |
| pt_width | Plate 폭 | 설명변수 | 연속형 | | histogram, IQR | | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | |
| pt_length | Plate 길이 | 설명변수 | 연속형 | | histogram, IQR | | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | |
| hsb | hsb적용여부 | 설명변수 | 범주형 | | 빈도분석 | 압연방법, 강종 종류에 따른 카이제곱 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | 0 |
| fur_no | 가열로 호기 | 설명변수 | 범주형 | 관련없음 | 빈도분석 | | | | | | | | | | |
| fur_input_row | 가열로 장입열 | 설명변수 | 범주형 | 관련없음 | 빈도분석 | | | | | | | | | | |
| fur_heat_temp | 가열로 가열대 온도(℃) | 설명변수 | 연속형 | | histogram, IQR | | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | |
| fur_heat_time | 가열로 가열대 시간(분) | 설명변수 | 연속형 | | histogram, IQR | | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | |
| fur_soak_temp | 가열로 균열대 온도(℃) | 설명변수 | 연속형 | | histogram, IQR | | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | |
| fur_soak_time | 가열로 균열대 시간(분) | 설명변수 | 연속형 | 타 변수와 분포동일 | histogram, IQR | | | | | | | | | | |
| fur_total_time | 가열로 총 시간(분) | 설명변수 | 연속형 | 목표변수와분포일정 | histogram, IQR | | | | | | | | | | |
| fur_ex_temp | 가열로 추출온도(℃) | 설명변수 | 연속형 | | histogram, IQR | | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | 0 |
| rolling_method | 압연방법 | 설명변수 | 범주형 | | 빈도분석 | hsb, 강종 종류에 따른 카이제곱 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | |
| rolling_temp | 압연온도(℃) | 설명변수 | 연속형 | | histogram, IQR | | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | 0 |
| descaling_count | 압연 중 Descaling 횟수 | 설명변수 | 연속형 | 관련없음 | histogram, IQR | | | | | | | | | | |
| work_group | 작업조 | 설명변수 | 범주형 | 관련없음 | 빈도분석 | | | | | | | | | | |