
후판공정 Scale 불량 핵심 요인 분석 및 개선안 제시

문제 정의 및 목표

- Scale 불량 발생 증가라는 이슈 발생
 - 압연 공정에서 Scale 불량이 급증한 것을 확인
 - 스케일이 발생하면 제품의 품질이 저하될 수 있음
- 후판 제품 Scale 불량률의 핵심 요인 도출 필요
- 근본 원인을 찾아 개선된 프로세스와 사전 조치를 통해 불량 발생률 저하 및 품질 향상

1. 데이터 구성 및 품질 확인

데이터 구성

기간: 2023/01/03 ~ 2023/01/10(총 1000개 데이터)

목표 변수

- scale: 스케일(산화철)의 불량 여부

설명 변수

- plate_no: 제품 번호
- rolling_date: 열연작업시각
- spec_long: 제품 규격
- spec_country: 제품 규격 기준국
- steel_kind: 강종(탄소강(C), 티타늄강(T))
- pt_thick: 후판 지시 두께
- pt_width: 후판 지시 폭
- pt_length: 후판 지시 길이
- hsb: Hot Scale Braker(열간 스케일 파쇄기) 적용 여부
- fur_no: 가열로 호기
- fur_input_row: 가열로 장입열
- fur_heat_temp: 가열로의 가열대 소재온도
- fur_heat_time: 가열로의 가열대 재로시간(분)
- fur_soak_temp: 가열로의 균열대 소재온도
- fur_soak_time: 가열로의 균열대 재로시간(분)
- fur_total_time: 가열로 총 재로시간(예열대+가열대+균열대)
- fur_ex_temp: 가열로 추출온도(계산치)
- rolling_method: 압연 방법(TMCP, CR)
- rolling_temp: 압연 온도(압연 과정 동안 유지되는 주변 온도)
- descaling_count: 압연 디스케일링 횟수
- work_group: 작업조(4조 2교대 - 07시/19시 기준)

목표변수 Scale의 기술통계량

count	1000
unique	2
top	양품
freq	690

- 고유값이 두 개 존재(양품 / 불량)
- 1000개의 데이터 중 690개의 양품이 존재

결측치 확인

```
df_raw.isnull().sum()
```

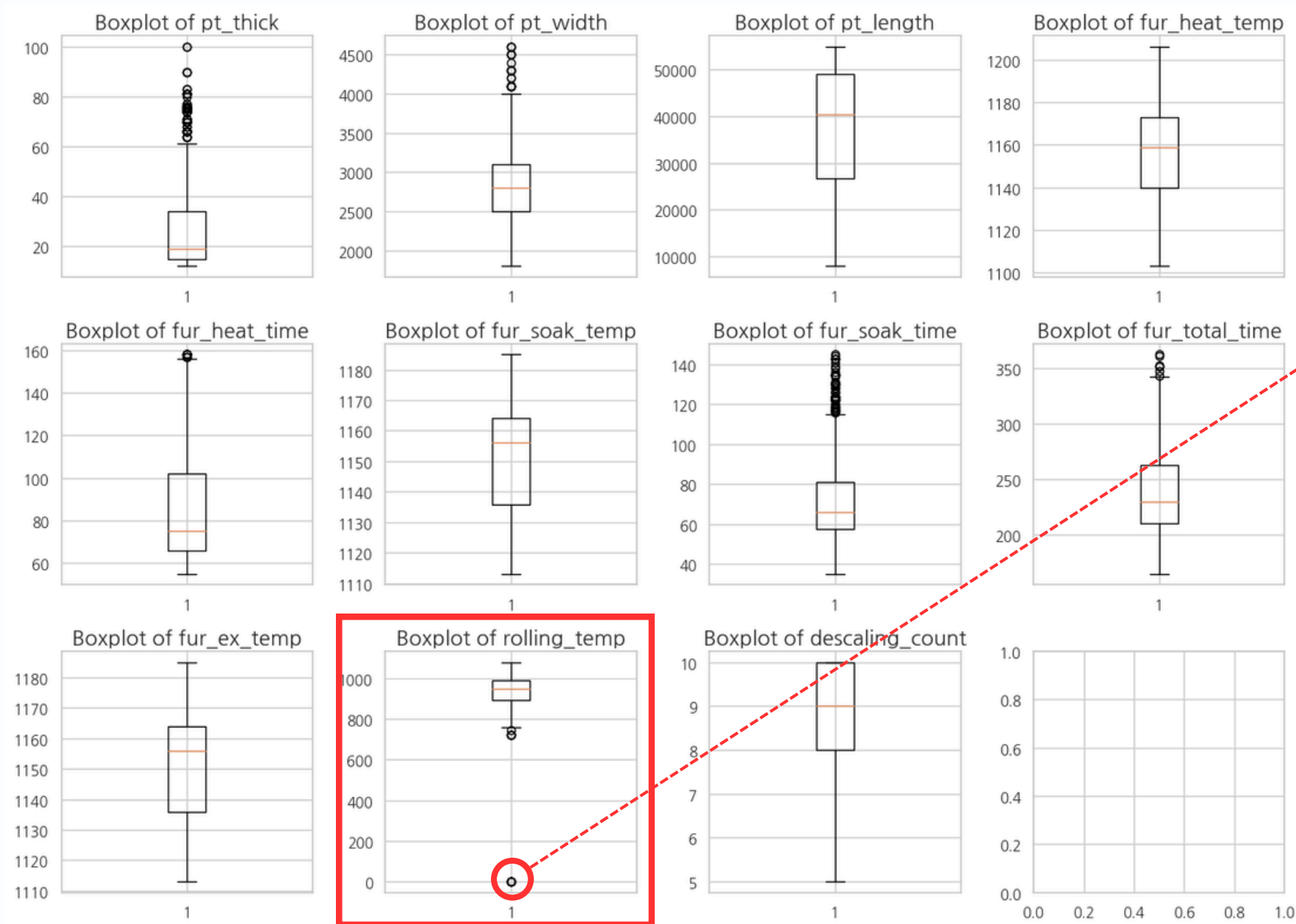
plate_no	0	fur_input_row	0
rolling_date	0	fur_heat_temp	0
scale	0	fur_heat_time	0
spec_long	0	fur_soak_temp	0
spec_country	0	fur_soak_time	0
steel_kind	0	fur_total_time	0
pt_thick	0	fur_ex_temp	0
pt_width	0	rolling_method	0
pt_length	0	rolling_temp	0
hsb	0	descaling_count	0
fur_no	0	work_group	0

- 결측값은 존재하지 않음

1. 데이터 구성 및 품질 확인

이상치 확인

- Boxplot으로 데이터를 확인한 결과, **rolling_temp**에서 이상치를 발견했다.



rolling_temp	
count	1,000.0000
mean	934.6370
std	96.5980
min	0.0000
25%	893.7500
50%	948.0000
75%	991.0000
max	1,078.0000

rolling_temp는 압연 과정 동안 유지되는 주변 온도로, 해당 온도는 0도가 될 수 없으므로 이상치로 판단

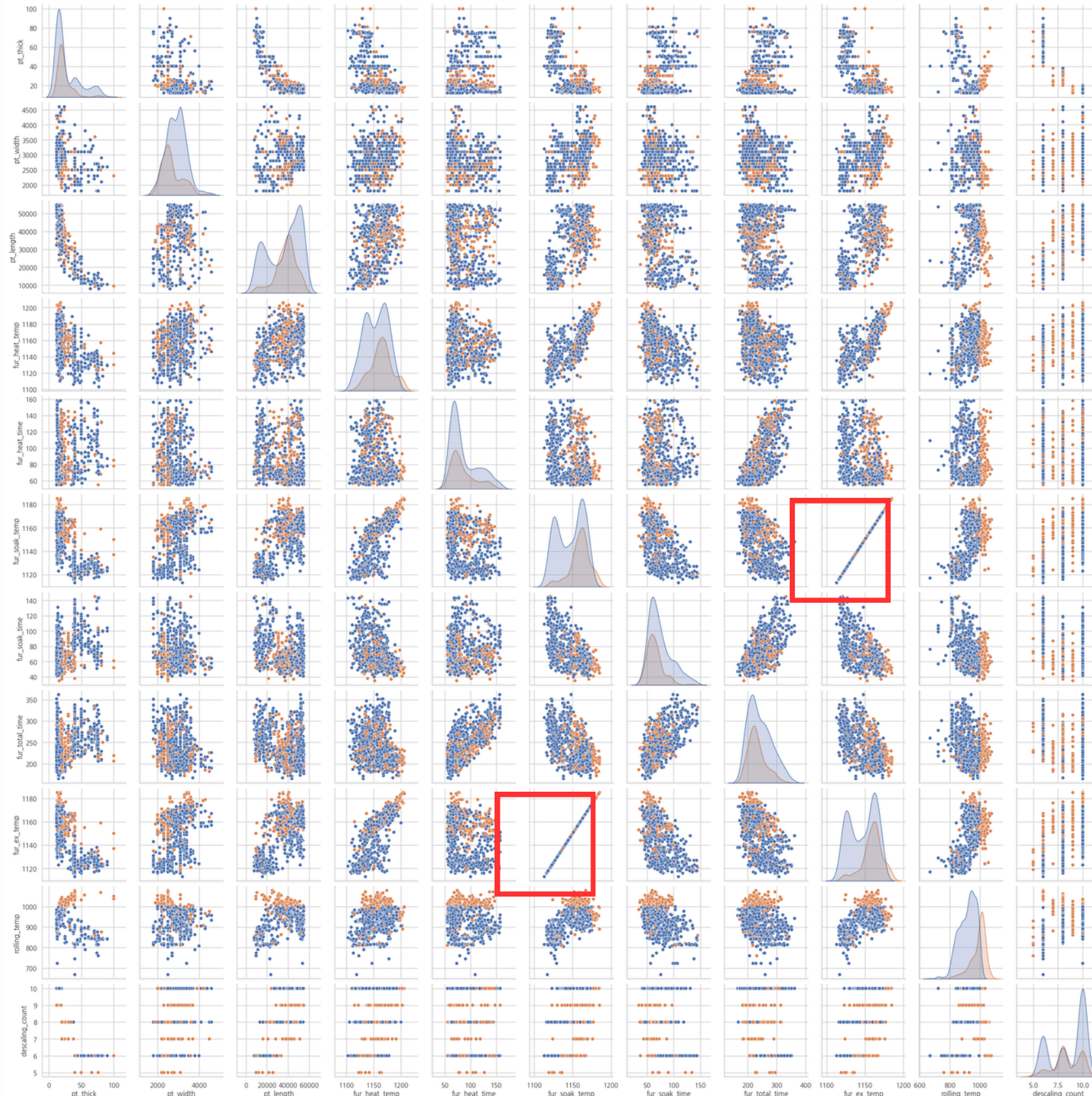
다른 데이터에 영향을 최소화하기 위해 동일한 spec_long(제품 규격)을 가진 데이터들의 rolling_temp 평균값으로 대체

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

	plate_no	rolling_date	scale	spec_long	spec_country	steel_kind	pt_thick
53	PLT_1054	03JAN2023:16:22:29	양품	GL-E36-TM	독일	T	!
221	PLT_1222	04JAN2023:19:44:09	양품	JS-SM490YB	일본	C	
222	PLT_1223	04JAN2023:19:56:39	양품	JS-SM490YB	일본	C	
598	PLT_1599	07JAN2023:11:40:51	양품	PILAC-BT33	한국	T	
599	PLT_1600	07JAN2023:11:52:00	양품	PILAC-BT33	한국	T	
600	PLT_1601	07JAN2023:11:59:10	양품	NV-D32-TM	영국	T	4

2. EDA - 변수들 간의 상관관계

- 산점도를 통해 완전한 선형 관계를 띄는 두 변수 발견
(fur_soak_temp & fur_ex_temp)



scale	1.000	-0.098	-0.190	0.052	0.222	-0.036	0.328	-0.220	-0.128	0.328	0.510	-0.040
pt_thick	-0.098	1.000	-0.320	-0.867	-0.525	0.207	-0.666	0.357	0.338	-0.666	-0.456	-0.845
pt_width	-0.190	-0.320	1.000	0.148	0.201	-0.101	0.217	-0.097	-0.133	0.217	-0.150	0.350
pt_length	0.052	-0.867	0.148	1.000	0.471	-0.138	0.603	-0.334	-0.276	0.603	0.373	0.818
fur_heat_temp	0.222	-0.525	0.201	0.471	1.000	-0.164	0.787	-0.373	-0.314	0.787	0.350	0.488
fur_heat_time	-0.036	0.207	-0.101	-0.138	-0.164	1.000	-0.349	0.115	0.736	-0.349	-0.135	-0.167
fur_soak_temp	0.328	-0.666	0.217	0.603	0.787	-0.349	1.000	-0.597	-0.561	1.000	0.633	0.628
fur_soak_time	-0.220	0.357	-0.097	-0.334	-0.373	0.115	-0.597	1.000	0.633	-0.597	-0.480	-0.399
fur_total_time	-0.128	0.338	-0.133	-0.276	-0.314	0.736	-0.561	0.633	1.000	-0.561	-0.341	-0.339
fur_ex_temp	0.328	-0.666	0.217	0.603	0.787	-0.349	1.000	-0.597	-0.561	1.000	0.633	0.628
rolling_temp	0.510	-0.456	-0.150	0.373	0.350	-0.135	0.633	-0.480	-0.341	0.633	1.000	0.323
descaling_count	-0.040	-0.845	0.350	0.818	0.488	-0.167	0.628	-0.399	-0.339	0.628	0.323	1.000

- fur_soak_temp와 fur_ex_temp의 상관 계수는 1.000로 나타남 (완전한 양의 상관관계)
- 두 변수 간의 상관계수가 높으면 모델링 시 다중공선성이 발생할 수 있음

⇒ fur_ex_temp(가열로 추출 온도)는 후판의 최종 온도이므로 scale 불량에 직접적인 영향이 덜 하다고 판단하여 제거

2. EDA - 설명변수와 범주형 변수

- 범주형 변수들만 추출해서 카이제곱 검정(유의수준 0.05)

rolling_date
chi-square statistic: 1000.000
dof: 999.000
p-value: 0.485

steel_kind
chi-square statistic: 77.710
dof: 1.000
p-value: 0.000

hsb
chi-square statistic: 106.413
dof: 1.000
p-value: 0.000

fur_no
chi-square statistic: 2.819
dof: 2.000
p-value: 0.244

fur_input_row
chi-square statistic: 0.922
dof: 1.000
p-value: 0.337

work_group
chi-square statistic: 13.846
dof: 3.000
p-value: 0.003

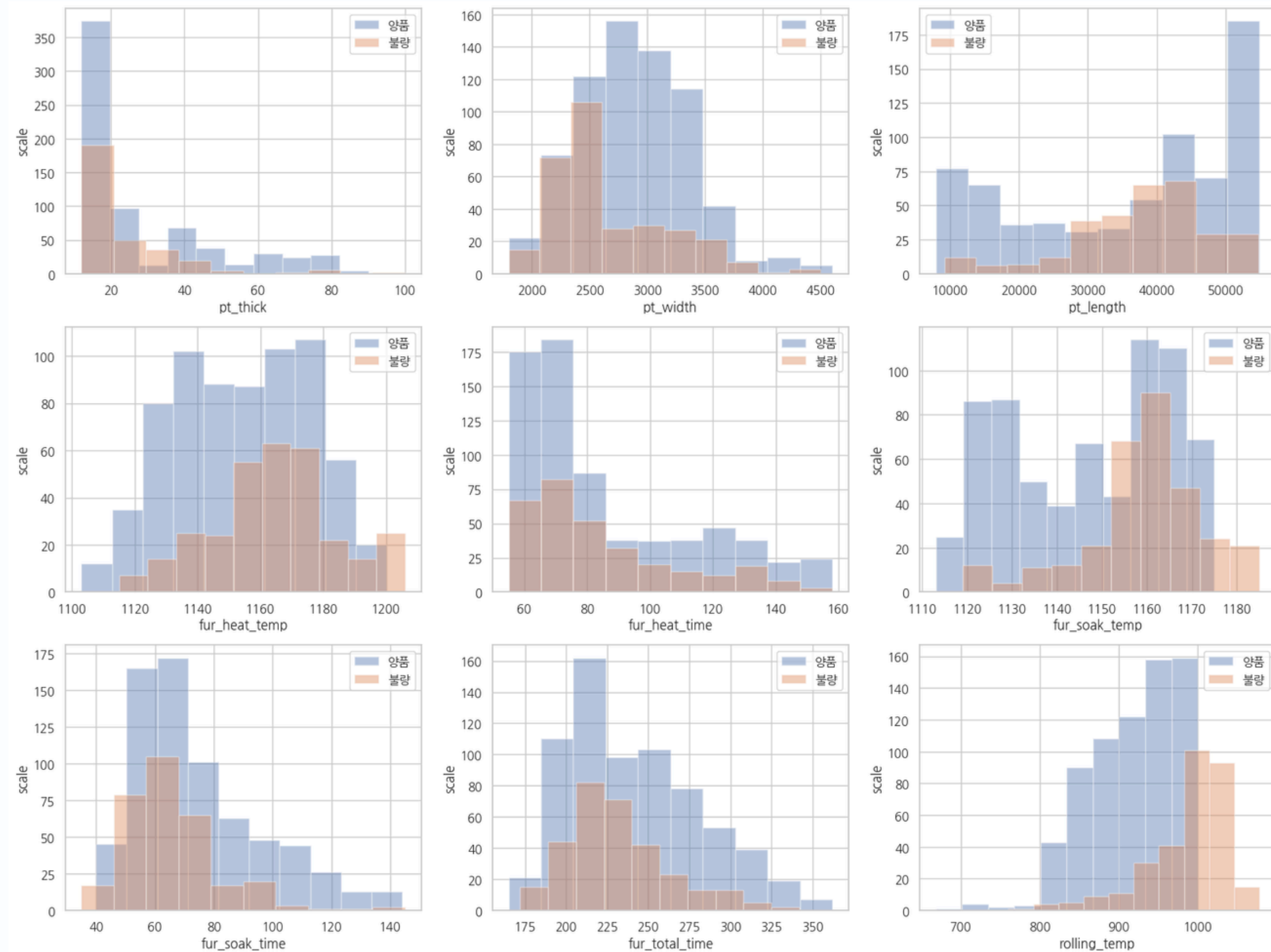
descaling_count
chi-square statistic: 216.085
dof: 5.000
p-value: 0.000

steel_kind, hsb, descaling_count는
scale에 많은 영향을 미침

rolling_date, fur_no, fur_input_row의
p-value가 0.5보다 높음
⇒ scale에 영향이 없다고 판단하여 제거

2. EDA - 설명변수와 연속형 변수

- 연속형 변수들만 추출해서 Histogram으로 표현함



불량 경향 ↑

- plate_thick(판의 두께)가 얇을수록
- plate_width(판의 폭)가 작을수록
- fur_soak_temp(가열로 균열대 온도)가 높을수록
- fur_total_time(가열로 총 재로시간)이 짧을수록
- rolling_temp(압연온도)가 높을수록

3. 모델링 & 모델 평가

[로지스틱 회귀분석]

Current function value: 0.300917
Iterations: 35
Function evaluations: 36
Gradient evaluations: 36

Logit Regression Results

Dep. Variable:	scale	No. Observations:	1000
Model:	Logit	Df Residuals:	983
Method:	MLE	Df Model:	16
Date:	Mon, 21 Oct 2024	Pseudo R-squ.:	0.5139
Time:	07:04:57	Log-Likelihood:	-300.92
converged:	False	LL-Null:	-619.10
Covariance Type:	nonrobust	LLR p-value:	4.372e-125

	coef	std err	z	P> z	[0.025	0.975]
Intercept	-1.9140	0.689	-2.779	0.005	-3.264	-0.564
C(steel_kind) [T.2]	1.0608	1.088	0.975	0.330	-1.071	3.193
C(hsb) [T.2]	14.0791	15.723	0.895	0.371	-16.738	44.896
C(rolling_method) [T.2]	-0.8540	0.756	-1.130	0.259	-2.336	0.628
C(work_group) [T.2]	0.0937	0.288	0.326	0.745	-0.470	0.658
C(work_group) [T.3]	-0.3892	0.322	-1.208	0.227	-1.020	0.242
C(work_group) [T.4]	-0.5844	0.342	-1.710	0.087	-1.254	0.085
pt_thick	-0.4918	0.371	-1.325	0.185	-1.219	0.236
pt_width	-0.5749	0.138	-4.174	0.000	-0.845	-0.305
pt_length	-0.6131	0.254	-2.417	0.016	-1.110	-0.116
fur_heat_temp	0.4254	0.256	1.664	0.096	-0.076	0.927
fur_heat_time	0.6187	0.225	2.751	0.006	0.178	1.060
fur_soak_temp	1.0411	0.417	2.497	0.013	0.224	1.858
fur_soak_time	-0.0378	0.225	-0.168	0.866	-0.478	0.402
fur_total_time	-0.2245	0.269	-0.836	0.403	-0.751	0.302
rolling_temp	1.7317	0.255	6.795	0.000	1.232	2.231
descaling_count	-1.1555	0.220	-5.257	0.000	-1.586	-0.725

모델의 설명력은 51.39%

[의사결정나무]

- GridSerachCV로 찾은 최적의 파라미터

```
param_grid=[{'min_samples_leaf': range(5,50,10),  
             'max_depth':range(3,12,3),  
             'min_samples_split': range(5,50,10)}]  
  
tree=GridSearchCV(DecisionTreeClassifier(random_state=456), param_grid)  
tree.fit(X_train, y_train)  
  
{'max_depth': 9, 'min_samples_leaf': 5, 'min_samples_split': 5}
```

- 최적의 파라미터로 생성한 모델의 정확도 확인

```
tree=DecisionTreeClassifier(max_depth=9,  
                             min_samples_leaf=5,  
                             min_samples_split=5,  
                             random_state=456)  
  
tree.fit(X_train, y_train)  
  
train accuracy: 0.9914  
f1-score: 0.9846  
test accuracy: 0.9900
```

⇒ Train은 99.14%, Test는 99%

3. 모델링 & 모델 평가

[랜덤포레스트]

- GridSearchCV로 찾은 최적의 파라미터

```
param_grid=[{'min_samples_leaf': range(5,50,10),  
             'max_depth':range(3,12,3),  
             'min_samples_split': range(5,50,10)}]  
  
rf_temp=GridSearchCV(RandomForestClassifier(random_state=456), param_grid)  
rf_temp.fit(X_train, y_train)
```

```
{'max_depth': 9, 'min_samples_leaf': 5, 'min_samples_split': 15}
```

- 최적의 파라미터로 생성한 모델의 정확도 확인

```
rf=RandomForestClassifier(max_depth=9,  
                           min_samples_leaf=5,  
                           min_samples_split=15,  
                           random_state=456)  
rf.fit(X_train, y_train)
```

```
train accuracy: 0.9643  
f1-score: 0.9355  
test accuracy: 0.9600
```

⇒ Train은 96.43%, Test는 96%

[그래디언트 부스팅]

- GridSearchCV로 찾은 최적의 파라미터

```
param_grid=[{'min_samples_leaf': range(5,50,10),  
             'max_depth':range(3,12,3),  
             'min_samples_split': range(5,50,10)}]  
  
gb_temp=GridSearchCV(GradientBoostingClassifier(random_state=456), param_grid)  
gb_temp.fit(X_train, y_train)
```

```
{'max_depth': 3, 'min_samples_leaf': 15, 'min_samples_split': 5}
```

- 최적의 파라미터로 생성한 모델의 정확도 확인

```
gb=GradientBoostingClassifier(max_depth=3,  
                              min_samples_leaf=15,  
                              min_samples_split=5,  
                              random_state=456)  
gb.fit(X_train, y_train)
```

```
train accuracy: 0.9943  
f1-score: 0.9898  
test accuracy: 0.9933
```

⇒ Train은 99.43%, Test는 99.33%

3. 모델링 & 모델 평가

[XGBoost]

- GridSearchCV로 찾은 최적의 파라미터

```
param_grid=[{'min_samples_leaf': range(5,50,10),
             'max_depth':range(3,12,3),
             'min_samples_split': range(5,50,10)}]

xgb_temp=GridSearchCV(XGBClassifier(random_state=456), param_grid)
xgb_temp.fit(X_train, y_train)

{'max_depth': 3, 'min_samples_leaf': 5, 'min_samples_split': 5}
```

- 최적의 파라미터로 생성한 모델의 정확도 확인

```
xgb=XGBClassifier(max_depth=3,
                  min_samples_leaf=5,
                  min_samples_split=5,
                  random_state=456)
xgb.fit(X_train, y_train)
```

```
train accuracy: 1.0000
f1-score: 0.9846
test accuracy: 0.9900
```

⇒ Train은 100%, Test는 99%

[SVM]

- GridSearchCV로 찾은 최적의 파라미터

```
param_grid=[{'C': [0.1, 1, 10],
             'kernel':['linear', 'poly', 'rbf', 'sigmoid'],
             'gamma': ['scale','auto',0.1, 1]}]

svm_temp=GridSearchCV(SVC(random_state=456), param_grid)
svm_temp.fit(X_train, y_train)

{'C': 1, 'gamma': 0.1, 'kernel': 'poly'}
```

- 최적의 파라미터로 생성한 모델의 정확도 확인

```
svm=SVC(C=1,
        gamma=0.1,
        kernel='poly',
        random_state=456)
svm.fit(X_train, y_train)
```

```
train accuracy: 0.9371
f1-score: 0.8191
test accuracy: 0.8867
```

⇒ Train은 93.71%, Test는 88.67%

3. 모델링 & 모델 평가

모델별 Accuracy와 F1-score 비교

	train accuracy	test accuracy	f1-score
model			
Logistic	0.8814	0.8567	0.7772
Decision Tree	0.9914	0.9900	0.9846
Random Forest	0.9643	0.9600	0.9355
Gradient Boosting	0.9943	0.9933	0.9898
XGBoost	1.0000	0.9900	0.9846
SVM	0.9371	0.8867	0.8191

⇒ Accuracy와 F1-score 모두 가장 높은 모델은 Gradient Boosting

최종 모델 생성 - Gradient Boosting

```
fin_model=GradientBoostingClassifier(max_depth=3,  
                                     min_samples_leaf=15,  
                                     min_samples_split=5,  
                                     random_state=456)  
  
fin_model.fit(X_train, y_train)  
  
y_pred=fin_model.predict(X_test)
```

[최종 모델의 confusion maxtrix 확인]

Confusion Matrix:

```
[[201  0]  
 [  2 97]]
```

Accuracy: 0.9933

Classification Report:

	precision	recall	f1-score	support
0	0.99	1.00	1.00	201
1	1.00	0.98	0.99	99
accuracy			0.99	300
macro avg	1.00	0.99	0.99	300
weighted avg	0.99	0.99	0.99	300

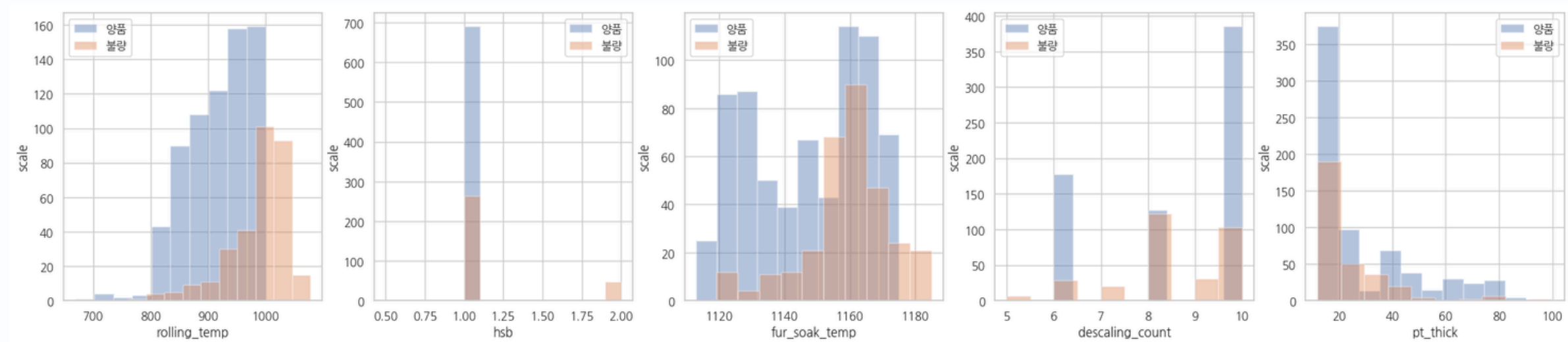
- 양품에 속하는 201개의 샘플 정확하게 예측
불량품에 속하는 2개의 샘플 오분류, 97개는 정확하게 예측
- 99.33%의 높은 정확도를 보임
Precision, Recall, F1-score 모두 1 또는 1에 가까운 값을
보이므로 데이터셋에 대한 일반화 성능이 좋다고 판단됨
⇒ 실무에서도 사용 가능

4. 핵심 요인 도출 및 기대 효과

[설명변수 중요도]

	feature	importance
8	rolling_temp	0.5500
11	hsb	0.1740
5	fur_soak_temp	0.1580
9	descaling_count	0.0790
0	pt_thick	0.0340
7	fur_total_time	0.0020
1	pt_width	0.0010
6	fur_soak_time	0.0010
2	pt_length	0.0010
3	fur_heat_temp	0.0010
4	fur_heat_time	0.0000
13	work_group	0.0000
12	rolling_method	0.0000
10	steel_kind	0.0000

[상위 5개 핵심 요인 Histogram]



1. **rolling_temp(압연 온도)**: 압연은 철강을 필요한 두께와 너비로 압착하는 과정으로 올바른 압연 온도 유지가 중요
⇒ **압연 온도 낮춤**
2. **hsb(열간 스케일 파쇄기)**: HSB는 공정 중 표면의 스케일을 고압수만으로 파쇄하는 장치로 후판의 표면 품질에 영향
⇒ **HSB 적용**
3. **fur_soak_temp(가열로 균열대 소재온도)**: 가열대는 압연할 재료를 가열하는 영역으로 온도가 가장 높음
  > 예열대 - 균열대 - 가열대 순으로 온도가 높으므로 가열대의 온도는 예열대와 균열대보다 높게 온도를 낮춰야 함
⇒ **가열로 균열대 온도를 낮춤**
4. **descaling_count**: 디스케일링은 철강 표면의 스케일을 제거하는 과정으로 적절한 디스케일링은 후판의 표면 품질에 영향
⇒ **압연 중 디스케일링 횟수 증가**
5. **pt_thick(후판 두께)**: 일반적으로 4.5mm 이상인 경우 후판으로 분류됨
⇒ **후판 두께를 얇은 제품 위주로 생산**

[기대 효과]

적절한 압연 온도 설정, 적절한 가열로 온도 설정, 스케일 제거 기술 개선을 통한 표면 처리 등
생성된 모델을 통해 불량률 예측하여 스케일 발생률을 낮추고 품질 향상을 도모할 수 있을 것이다.

THANK YOU

감사합니다