후판 공정 불량률 개선안 도출

- 분석 배경 및 목표
- 데이터 구성
- 그래프를 활용한 탐색적 분석
- 회귀, 분류모델을 이용한 분석
- Vital Few 선정
- Solution
- 기대효과

분석 배경 및 목표

배경

OO공장에서 후판 공정상 Scale 불량 발생 증가

분석 방향

- 그래프를 이용한 탐색적 분석
- 로지스틱 회귀분석
- 의사결정나무
- 랜덤 포레스트
- 그래디언트 부스팅

⇨ Vital Few 선정

목표

Vital Few 조절을 통해 Scale 발생률 5% 개선

데이터 구성 : 데이터 불러오기

In [17]:
```
df_raw = pd.read_csv("/home/piai/HW/Statistics3/SCALE불량.csv",engine='python', encoding='cp949')
df_raw
```

Out[17]:

	PLATE_NO	ROLLING_DATE	SCALE	SPEC	STEEL_KIND	PT_THK	PT_WDTH	PT_LTH	PT_WGT	FUR_NO	...	FUR_HZ_TEMP	FUR_HZ_TIME	FUR
0	PB562774	2008-08-01:00:00:15	양품	AB/EH32-TM	T1	32.25	3707	15109	14180	1호기	...	1144	116	
1	PB562775	2008-08-01:00:00:16	양품	AB/EH32-TM	T1	32.25	3707	15109	14180	1호기	...	1144	122	
2	PB562776	2008-08-01:00:00:59	양품	NV-E36-TM	T8	33.27	3619	19181	18130	2호기	...	1129	116	
3	PB562777	2008-08-01:00:01:24	양품	NV-E36-TM	T8	33.27	3619	19181	18130	2호기	...	1152	125	
4	PB562778	2008-08-01:00:01:44	양품	BV-EH36-TM	T8	38.33	3098	13334	12430	3호기	...	1140	134	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
715	PB563502	2008-08-02:13:35:36	불량	NK-KA	C0	20.14	3580	38639	21870	3호기	...	1172	72	
716	PB563503	2008-08-02:13:35:02	양품	NV-A32	C0	15.08	3212	48233	18340	2호기	...	1150	61	
717	PB563504	2008-08-02:14:40:00	양품	NV-A32	C0	16.60	3441	43688	19590	2호기	...	1169	65	
718	PB563505	2008-08-02:13:35:19	양품	LR-A	C0	15.59	3363	48740	80240	3호기	...	1179	86	
719	PB563506	2008-08-02:14:40:53	양품	GL-A32	C0	16.09	3400	54209	69840	3호기	...	1186	82	

720 rows × 21 columns

데이터 구성 : 변수

목표변수

- PLATE_NO Plate No
- ROLLING_DATE 작업시각
- SCALE Scale불량
- SPEC 제품 규격
- STEEL_KIND 강종
- PT_THK Plate 두께
- PT_WDTH Plate 폭
- PT_LTH Plate 길이
- PT_WGT Plate 중량
- FUR_NO 가열로 호기
- FUR_NO_ROW 가열로 작업순번
- FUR_HZ_TEMP 가열로 가열대 온도
- FUR_HZ_TIME 가열로 가열대 시간
- FUR_SZ_TEMP 가열로 균열대 온도
- FUR_SZ_TIME 가열로 균열대 시간
- FUR_TIME 가열로 시간
- FUR_EXTEMP 추출온도
- ROLLING_TEMP_T5 압연온도
- HSB HSB적용(1-적용,0-미적용)
- ROLLING_DESCALING 압연 중 Descaling 횟수
- WORK_GR 작업조

Vital Few 제외 변수 : PLATE_NO , ROLLING_DATE

후판의 기계, 재료적 특성, 설비 문제, 제작 공정의 요인이 아님

데이터 구성 : 결측치 확인 및 제거

```
In [8]: df_raw.info()
```

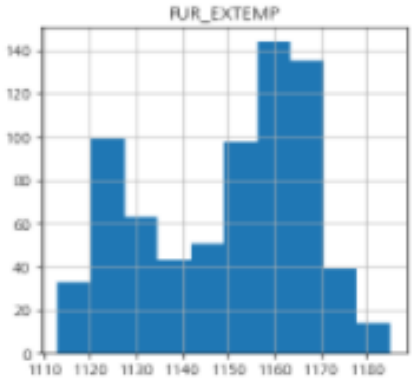
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 720 entries, 0 to 719
Data columns (total 21 columns):
 #   Column             Non-Null Count  Dtype
---  ------             --------------  -----
 0   PLATE_NO           720 non-null    object
 1   ROLLING_DATE       720 non-null    object
 2   SCALE              720 non-null    object
 3   SPEC               720 non-null    object
 4   STEEL_KIND         720 non-null    object
 5   PT_THK             720 non-null    float64
 6   PT_WDTH            720 non-null    int64
 7   PT_LTH             720 non-null    int64
 8   PT_WGT             720 non-null    int64
 9   FUR_NO             720 non-null    object
 10  FUR_NO_ROW         720 non-null    int64
 11  FUR_HZ_TEMP        720 non-null    int64
 12  FUR_HZ_TIME        720 non-null    int64
 13  FUR_SZ_TEMP        720 non-null    int64
 14  FUR_SZ_TIME        720 non-null    int64
 15  FUR_TIME           720 non-null    int64
 16  FUR_EXTEMP         720 non-null    int64
 17  ROLLING_TEMP_T5    720 non-null    int64
 18  HSB                720 non-null    object
 19  ROLLING_DESCALING  720 non-null    int64
 20  WORK_GR            720 non-null    object
dtypes: float64(1), int64(12), object(8)
memory usage: 118.2+ KB
```

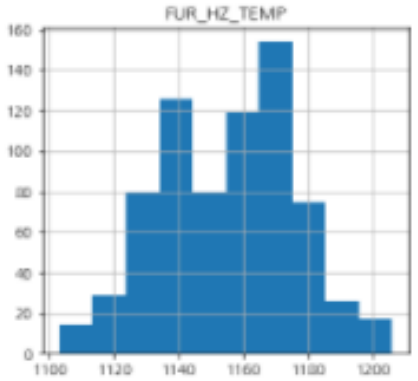
```
In [9]: df_raw  = df_raw.astype({'FUR_NO_ROW' : 'object'})
df_raw.info()
```

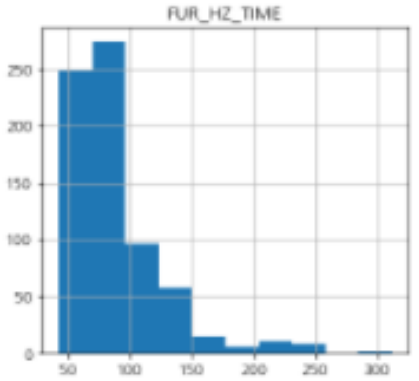
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 720 entries, 0 to 719
Data columns (total 21 columns):
 #   Column             Non-Null Count  Dtype
---  ------             --------------  -----
 0   PLATE_NO           720 non-null    object
 1   ROLLING_DATE       720 non-null    object
 2   SCALE              720 non-null    object
 3   SPEC               720 non-null    object
 4   STEEL_KIND         720 non-null    object
 5   PT_THK             720 non-null    float64
 6   PT_WDTH            720 non-null    int64
 7   PT_LTH             720 non-null    int64
 8   PT_WGT             720 non-null    int64
 9   FUR_NO             720 non-null    object
 10  FUR_NO_ROW         720 non-null    object
 11  FUR_HZ_TEMP        720 non-null    int64
 12  FUR_HZ_TIME        720 non-null    int64
 13  FUR_SZ_TEMP        720 non-null    int64
 14  FUR_SZ_TIME        720 non-null    int64
 15  FUR_TIME           720 non-null    int64
 16  FUR_EXTEMP         720 non-null    int64
 17  ROLLING_TEMP_T5    720 non-null    int64
 18  HSB                720 non-null    object
 19  ROLLING_DESCALING  720 non-null    int64
 20  WORK_GR            720 non-null    object
dtypes: float64(1), int64(11), object(9)
memory usage: 118.2+ KB
```

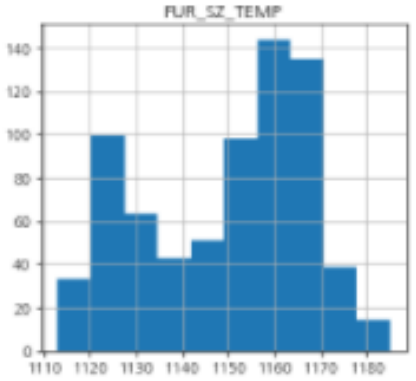
- 결측치가 없음
- 가열로 작업 순번은 범주형으로 분류

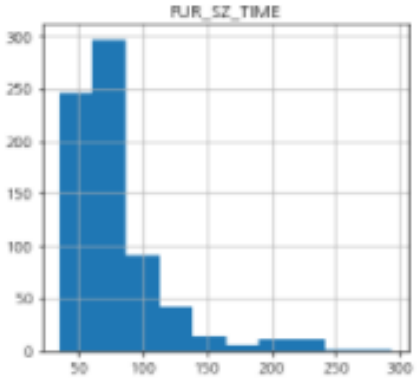
데이터 구성 : 이상치 확인 및 제거

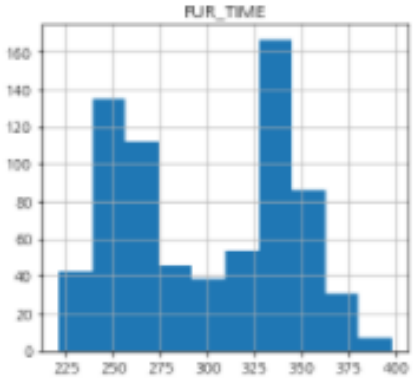




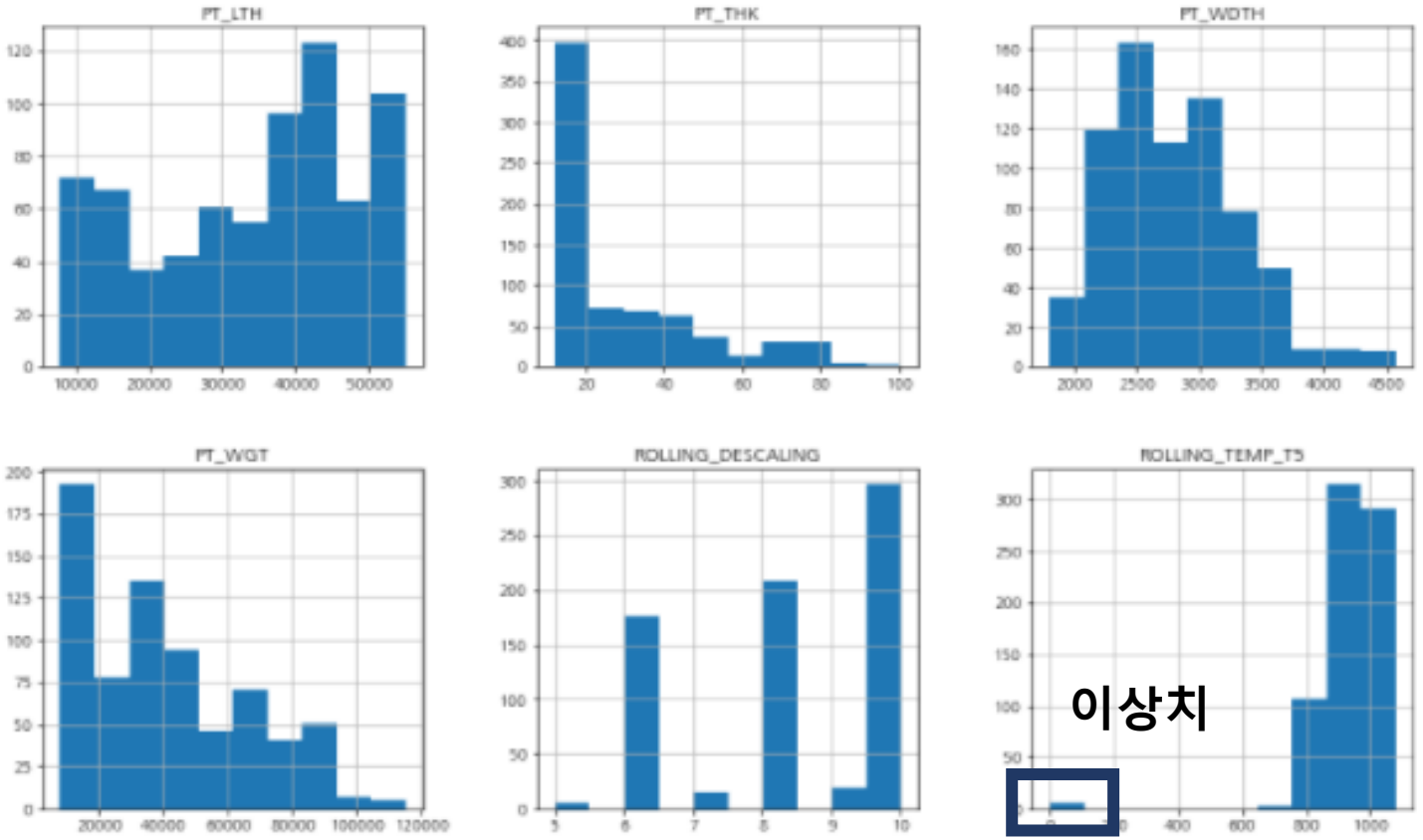






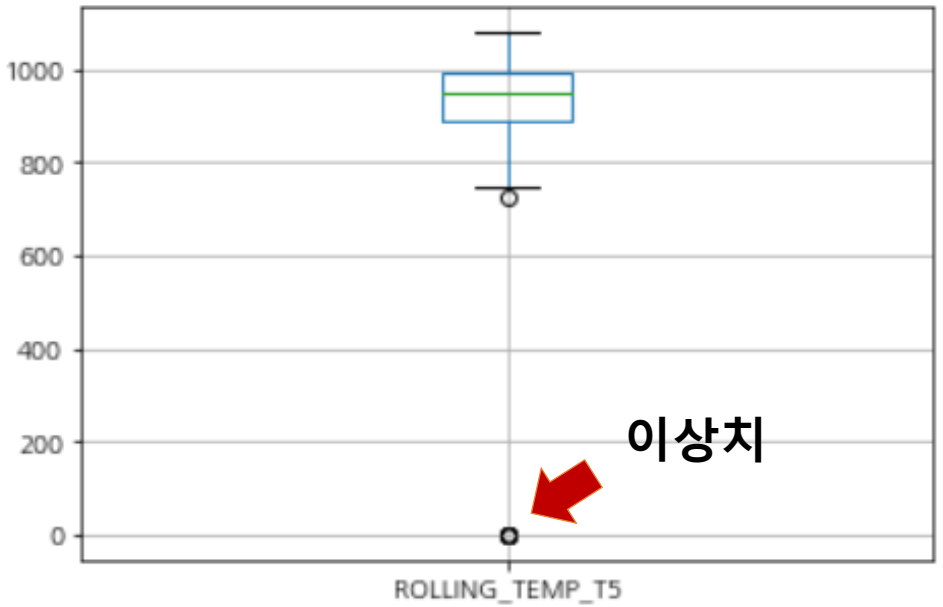


데이터 구성 : 이상치 확인 및 제거



- ROLLING_TEMP_T5 에서 이상치 존재 가능성 발견
- Box_plot으로 확인 필요

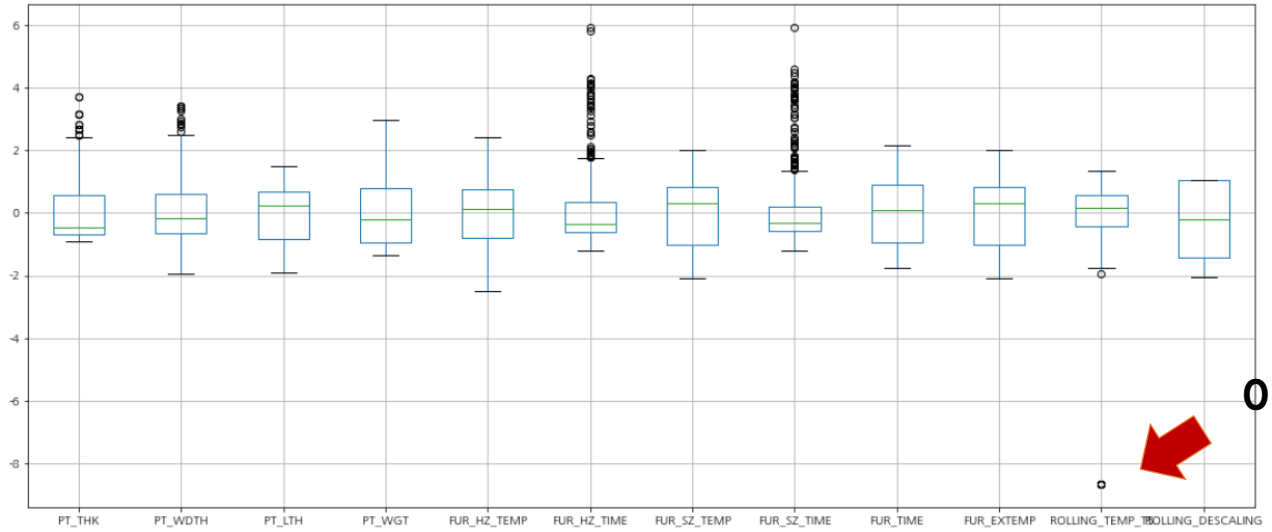
데이터 구성 : 이상치 확인 및 제거



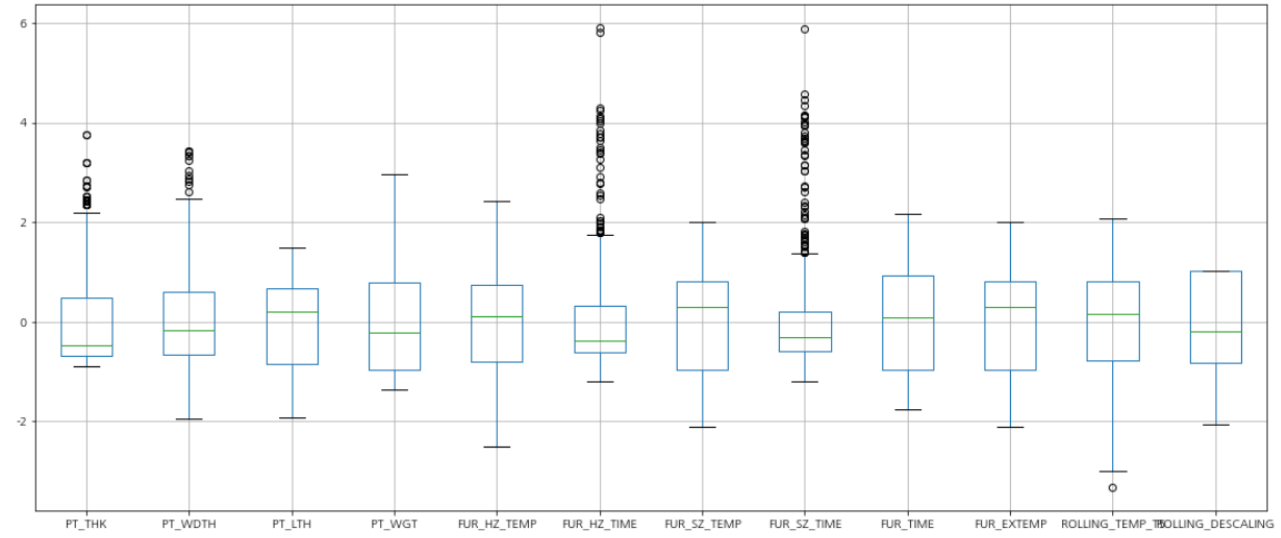

```
In [74]: df_raw[df_raw['ROLLING_TEMP_T5']<10].count()

Out[74]: PLATE_NO            6
         ROLLING_DATE        6
         SCALE               6
         SPEC                6
         STEEL_KIND          6
         PT_THK              6
         PT_WDTH             6
         PT_LTH              6
         PT_WGT              6
         FUR_NO              6
         FUR_NO_ROW          6
         FUR_HZ_TEMP         6
         FUR_HZ_TIME         6
         FUR_SZ_TEMP         6
         FUR_SZ_TIME         6
         FUR_TIME            6
         FUR_EXTEMP          6
         ROLLING_TEMP_T5     6
         HSB                 6
         ROLLING_DESCALING   6
         WORK_GR             6
         dtype: int64
```

- 이상치 존재
- 720개 중 6개의 이상치 데이터
- 이상치 처리 방법 : 삭제

데이터 구성 : 이상치 확인 및 제거

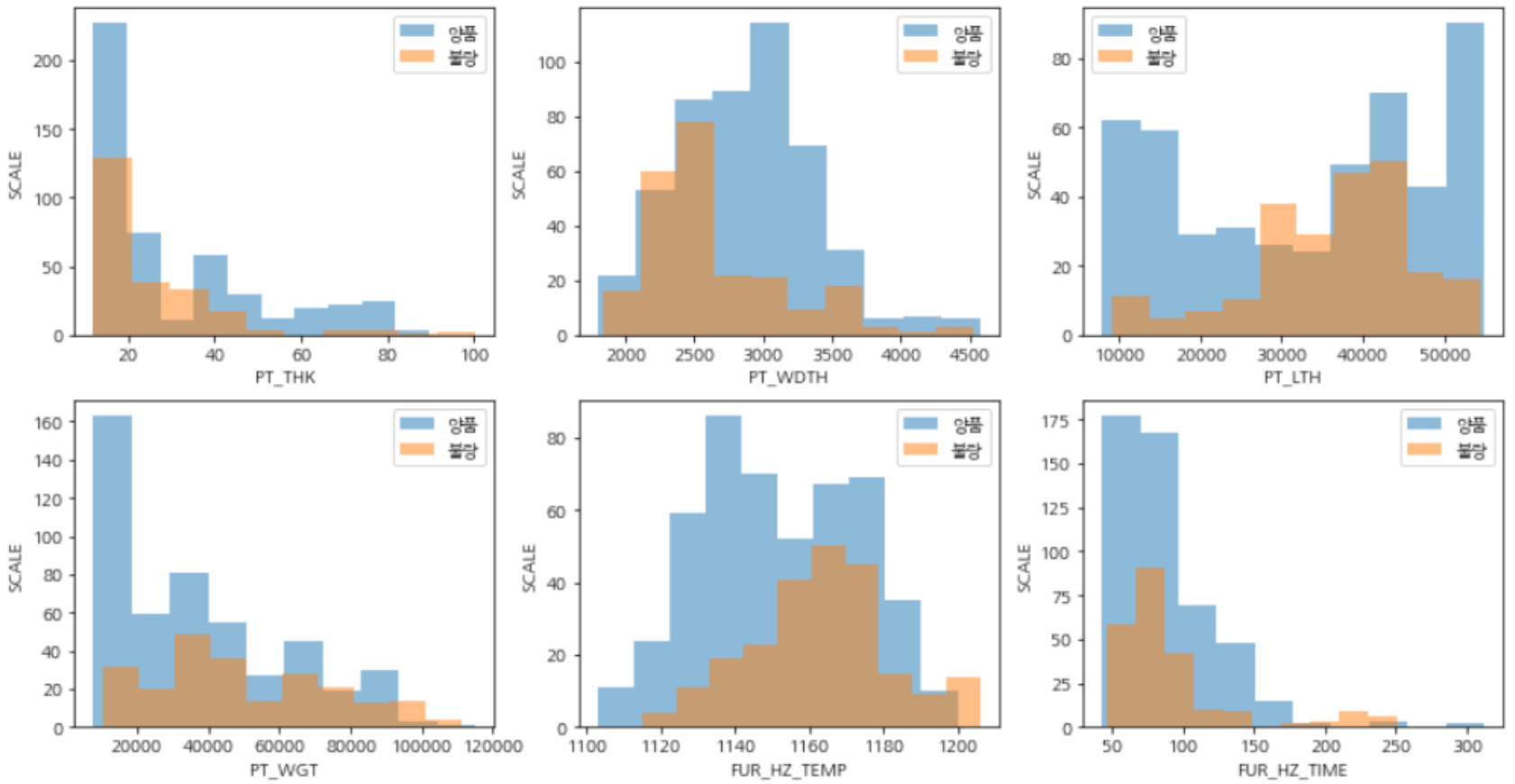


이상치



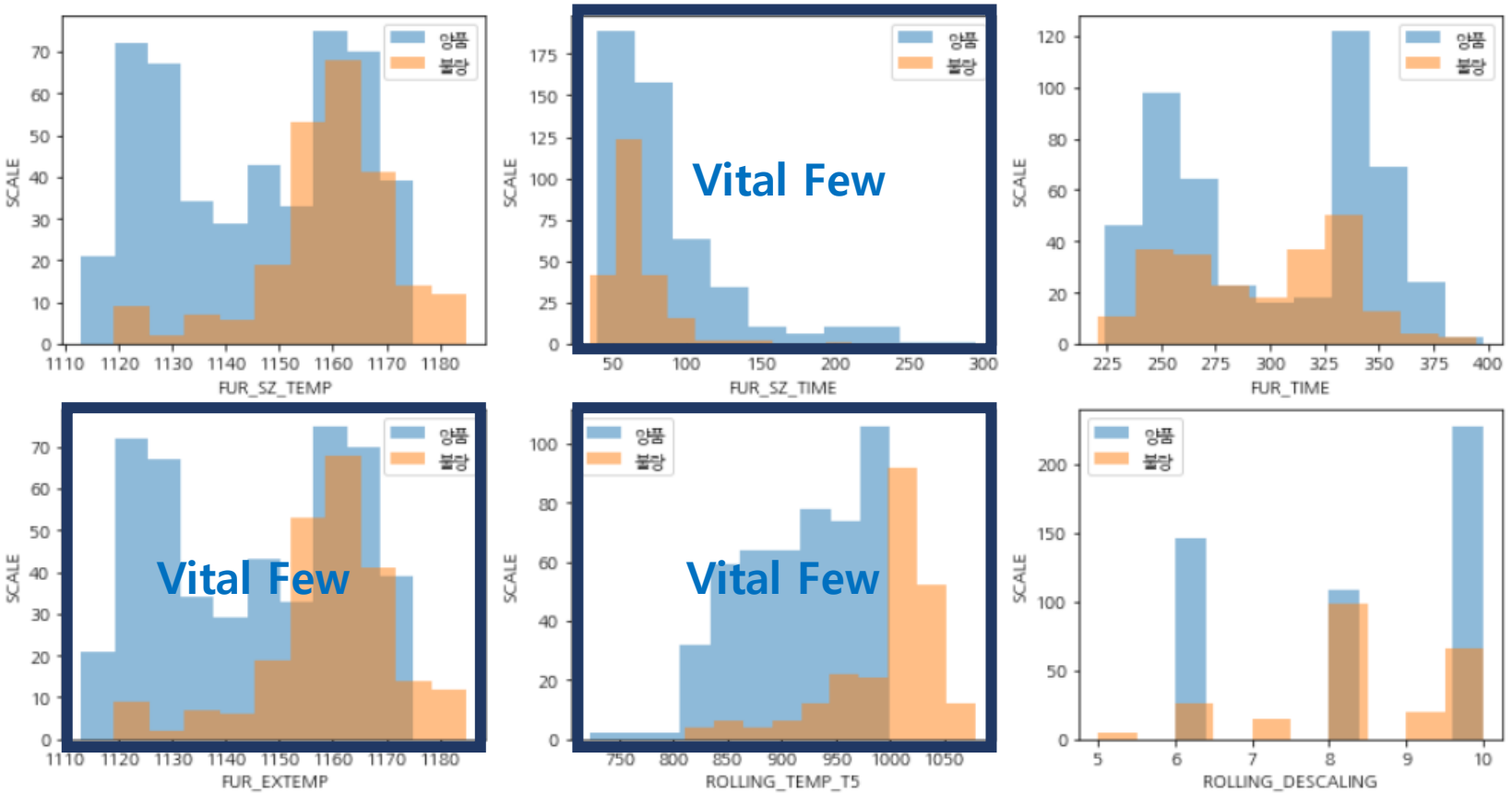
이상치 제거된 모습
(표준화한 연속형 설명변수)

그래프를 활용한 탐색적 분석 : 목표변수 vs 연속형 설명변수



- 판의 두께, 길이, 폭에 따라 불량 발생의 변화가 있지만, 고객의 요구에 따라 지정되는 값
- 즉, Vital Few 로 선정하기 어려움
- 판의 무게에 따른 발생률의 변화가 미미함
- 가열로 가열대에서의 시간, 온도에서 불량률과의 약간의 상관관계가 발견

그래프를 활용한 탐색적 분석 : 목표변수 vs 연속형 설명변수

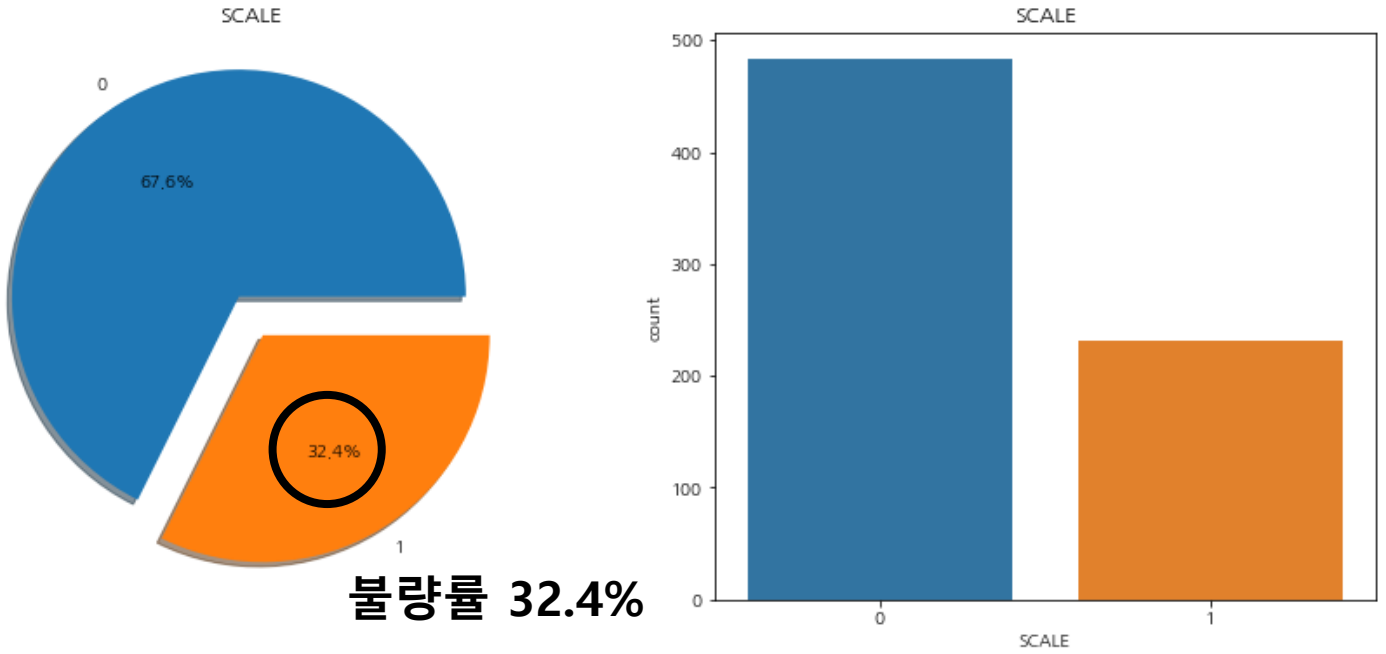


- 온도(추출, 압연, 균열대)가 높을 때, 높은 불량률
- 가열대에서의 온도가 짧으면 높은 확률로 불량 발생
- FUR_TIME, ROLLING_DESCALING에서는 비교적 적은 영향력

그래프를 활용한 탐색적 분석 : 목표변수 vs 범주형 설명변수

- SCALE 발생률 (목표변수와 범주형 설명변수의 관계를 설명할 지표)

In [82]:
```
## 독립변수에 따른 SCALE 불량률을 알기 위한 함수

def piecount(col):
    f, ax = plt.subplots(1, 2, figsize=(15, 6))
    df_raw[col].value_counts().plot.pie(explode=[0.1 for i in range(df_raw[col].nunique())], autopct='%1.1f%%', ax=ax[0], shadow=True)
    ax[0].set_title(col)
    ax[0].set_ylabel('')
    sns.countplot(col, data=df_raw, ax=ax[1])
    ax[1].set_title(col)
    plt.show()

piecount('SCALE')
```



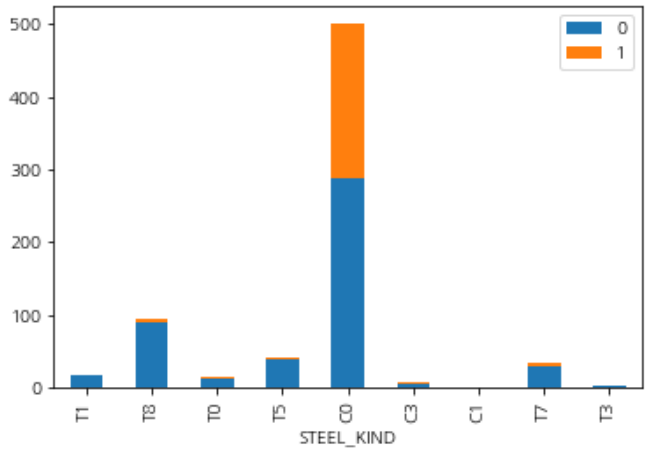
불량률 32.4%

그래프를 활용한 탐색적 분석 : 목표변수 vs 범주형 설명변수

- Scale 발생여부와 강종의 관계

In [83]:
```
aggregate('STEEL_KIND', 'SCALE', df_raw).plot(kind='bar' , stacked=True)
```

Out[83]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7fae13a0f6d0>
```



In [84]:
```
fun_print_crosstab(df_raw, 'STEEL_KIND')
```

```
STEEL_KIND   C0  C1  C3  T0  T1  T3  T5  T7  T8
SCALE
0           289   0   6  13  16   2  39  29  89
1           212   1   1   2   2   0   2   6   5
STEEL_KIND        C0   C1        C3        T0        T1   T3        T5  #
SCALE
0           0.576846  0.0  0.857143  0.866667  0.888889  1.0  0.95122
1           0.423154  1.0  0.142857  0.133333  0.111111  0.0  0.04878

STEEL_KIND        T7        T8
SCALE
0           0.828571  0.946809
1           0.171429  0.053191
```

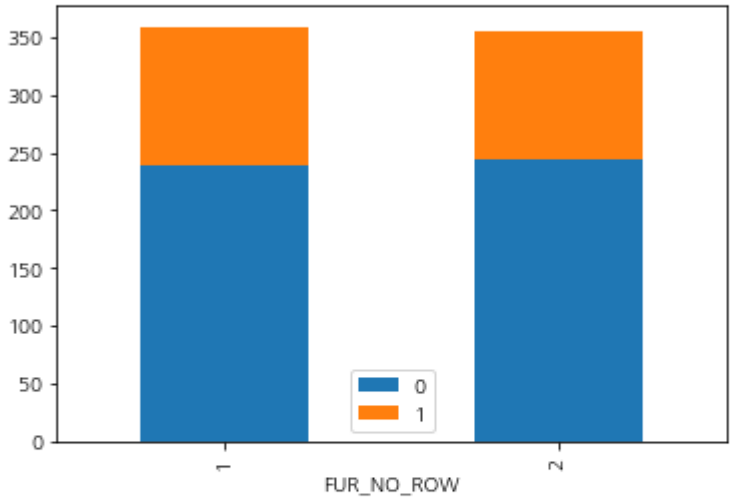
- C0 강종의 생산량이 가장 많음
- C0 불량률 42.3%
- C1 불량률 100%, 하지만 1개의 데이터
- 나머지 강종 불량률 20% 미만
- 즉, 특히 C0 강종에서만 전체 불량률인 32.4% 보다 불량이 많이 발생하기에 강종은 불량률에 충분한 영향을 끼침
- 따라서, **Vital Few**에 포함

그래프를 활용한 탐색적 분석 : 목표변수 vs 범주형 설명변수

- Scale 발생여부와 작업순서의 관계

In [85]:
```
aggregate('FUR_NO_ROW', 'SCALE', df_raw).plot(kind='bar' , stacked=True)
```

Out[85]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7fae13b9c370>
```



In [86]:
```
fun_print_crosstab(df_raw, 'FUR_NO_ROW')
```

```
FUR_NO_ROW    1    2
SCALE
0           239  244
1           120  111
FUR_NO_ROW         1         2
SCALE
0           0.665738  0.687324
1           0.334262  0.312676
```

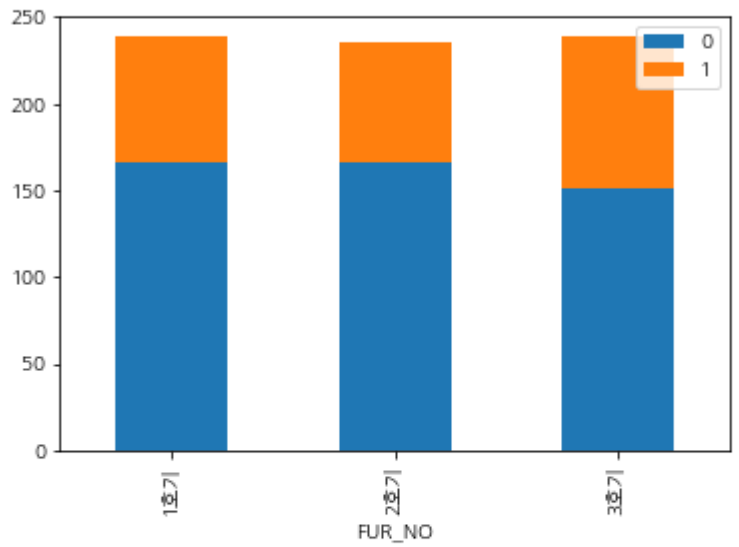
- 작업순서에 따른 불량률 = 약 32%
- 즉, 가열로 작업순서는 불량률에 영향을 끼치지 않음

그래프를 활용한 탐색적 분석 : 목표변수 vs 범주형 설명변수

- Scale 발생여부와 가열로 호기의 관계

In [87]:
```
aggregate('FUR_NO', 'SCALE', df_raw).plot(kind='bar' , stacked=True)
```

Out[87]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7fae138d9a90>
```



In [88]:
```
fun_print_crosstab(df_raw, 'FUR_NO')
```

```
FUR_NO  1호기  2호기  3호기
SCALE
0       166  166  151
1        73   70   88
FUR_NO       1호기      2호기       3호기
SCALE
0       0.694561  0.70339  0.631799
1       0.305439  0.29661  0.368201
```

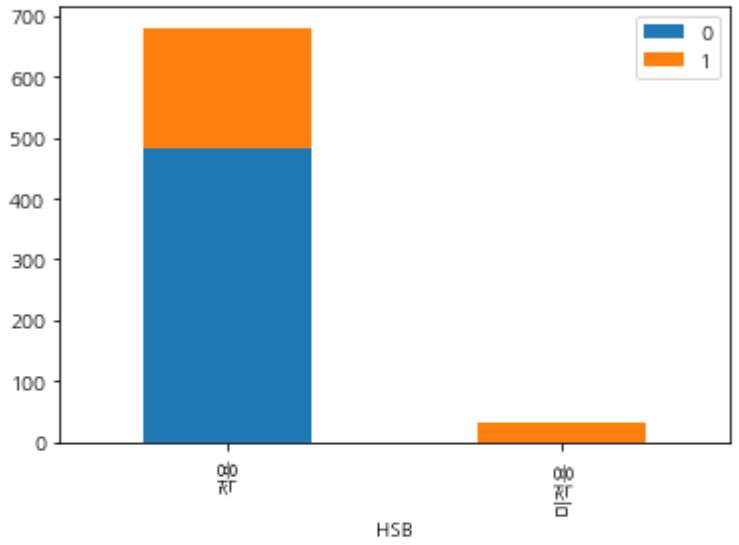
- 가열로에 따른 불량률 = 약 32%
- 즉, 가열로는 불량률에 영향을 끼치지 않음

그래프를 활용한 탐색적 분석 : 목표변수 vs 범주형 설명변수

- Scale 발생여부와 HSB 적용 여부와의 관계

In [89]:
```
aggregate('HSB', 'SCALE', df_raw).plot(kind='bar' , stacked=True)
```

Out [89]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7fae13c307c0>
```



In [90]:
```
fun_print_crosstab(df_raw, 'HSB')
```

```
HSB    미적용   적용
SCALE
0         0  483
1        33  198
HSB    미적용        적용
SCALE
0       0.0  0.709251
1       1.0  0.290749
```

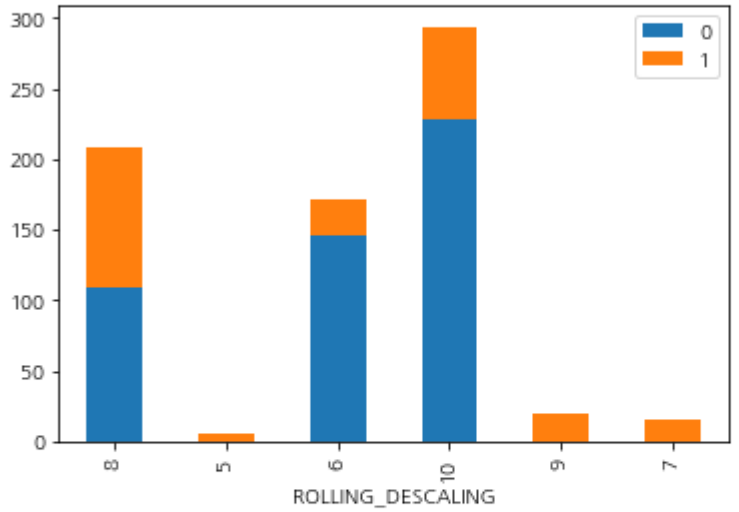
- HSB 미적용 -> 무조건 불량
- HSB 적용 -> 약 30% 불량
- HSB가 미적용된 제품에서는 100% 불량이 발생하고 적용하면 개선되기에 불량 발생에 큰 영향을 끼침
- 불량이 발생한 이유가 무조건 HSB가 미적용되었기 때문이라고 단정지을 수 없음
- 즉, HSB 변수를 제외시키고 다른 중요 변수를 찾아 이들의 영향 때문에 HSB가 적용되어도 불량이 발생한다고 말할 수 없음
- 따라서, HSB가 적용되어도 불량률이 발생한는 이유를 분석하기 위해 HSB를 **Vital Few**에 포함

그래프를 활용한 탐색적 분석 : 목표변수 vs 범주형 설명변수

- Scale 발생여부와 rolling_descaling적용 여부와의 관계

```
In [91]: aggregate('ROLLING_DESCALING', 'SCALE', df_raw).plot(kind='bar' , stacked=True)
```

```
Out[91]: <matplotlib.axes._subplots.AxesSubplot at 0x7fae13af2340>
```




```
In [92]: fun_print_crosstab(df_raw, 'ROLLING_DESCALING')
```

ROLLING_DESCALING	5	6	7	8	9	10
SCALE						
0	0	146	0	109	0	228
1	5	26	15	99	20	66

ROLLING_DESCALING	5	6	7	8	9	10
SCALE						
0	0.0	0.848837	0.0	0.524038	0.0	0.77551
1	1.0	0.151163	1.0	0.475962	1.0	0.22449

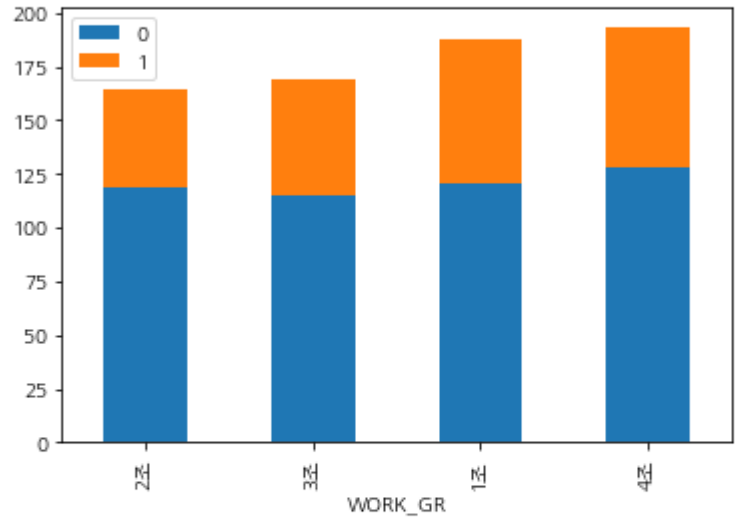
- ROLLING_DESCALING의 횟수가 많을수록 불량률이 개선되는 경향
- 따라서 **Vital Few**에 포함

그래프를 활용한 탐색적 분석 : 목표변수 vs 범주형 설명변수

- Scale 발생여부와 작업조의 관계

In [93]:
```
aggregate('WORK_GR', 'SCALE', df_raw).plot(kind='bar' , stacked=True)
```

Out [93]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7fae14040490>
```



In [94]:
```
fun_print_crosstab(df_raw, 'WORK_GR')
```

```
WORK_GR   1조   2조   3조   4조
SCALE
0        121  119  115  128
1         67   45   54   65
WORK_GR        1조        2조        3조        4조
SCALE
0        0.643617  0.72561  0.680473  0.663212
1        0.356383  0.27439  0.319527  0.336788
```

- 작업조에 따른 불량률의 차이가 미미
- 각 작업조의 불량률이 32.4%와 근소한 차이를 보임
- 즉, 작업조는 불량률에 영향을 끼치지 않음

그래프를 활용한 탐색적 분석 : 목표변수 vs 범주형 설명변수

- Scale 발생여부와 SPEC의 관계

In [95]:
```
fun_print_crosstab(df_raw, 'SPEC')
```

```
SPEC   A131-DH36TM  A283-C  A516-60  A709-36  AB/A  AB/AH32  AB/B  AB/EH32-TM  #
SCALE
0                0       1        1        1     3        4     3           2
1                1       5        1        0     4        1     3           0

SPEC   AB/EH36-TM  API-2W-50T  ...  NV-A32-TM  NV-A36-TM  NV-B  NV-D32-TM  #
SCALE                          ...
0              16           2  ...          2          2     1          3
1               1           0  ...          1          0     2          0

SPEC   NV-D36-TM  NV-E32-TM  NV-E36-TM  PILAC-BT33  SA283-C  V42JBN3
SCALE
0              4          2          5          36       11        3
1              1          0          0           2       10        1

[2 rows x 66 columns]
SPEC   A131-DH36TM    A283-C  A516-60  A709-36      AB/A  AB/AH32  AB/B  #
SCALE
0              0.0  0.166667      0.5      1.0  0.428571      0.8   0.5
1              1.0  0.833333      0.5      0.0  0.571429      0.2   0.5

SPEC   AB/EH32-TM  AB/EH36-TM  API-2W-50T  ...  NV-A32-TM  NV-A36-TM  #
SCALE                                      ...
0             1.0    0.941176         1.0  ...   0.666667        1.0
1             0.0    0.058824         0.0  ...   0.333333        0.0

SPEC       NV-B  NV-D32-TM  NV-D36-TM  NV-E32-TM  NV-E36-TM  PILAC-BT33  #
SCALE
0      0.333333        1.0        0.8        1.0        1.0    0.947368
1      0.666667        0.0        0.2        0.0        0.0    0.052632

SPEC   SA283-C  V42JBN3
SCALE
0      0.52381     0.75
1      0.47619     0.25

[2 rows x 66 columns]
```

- 66개의 SPEC의 종류
- 1개의 데이터만 포함한 SPEC 다수
- 따라서 SPEC으로 불량률 판단 불가

그래프를 활용한 탐색적 분석 : Vital Few

Vital Few

- FUR_SZ_TIME
- FUR_EXTEMP
- ROLLING_TEMP_T5
- STEEL KIND
- HSB
- ROLLING_DESCALING

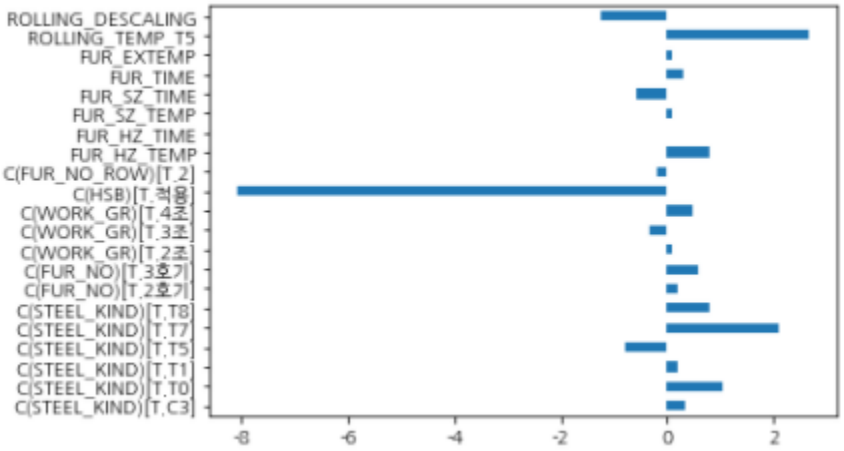
분류모델을 이용한 분석 : 중요하지 않은 변수 제거

설명변수 20개 − PLATE_NO, ROLLING_DATE, SPEC, PT_THK, PT_WGT, PT_LTH, PT_WDTH = **분류모델 사용 변수**

분류모델을 이용한 분석 : 로지스틱 회귀분석

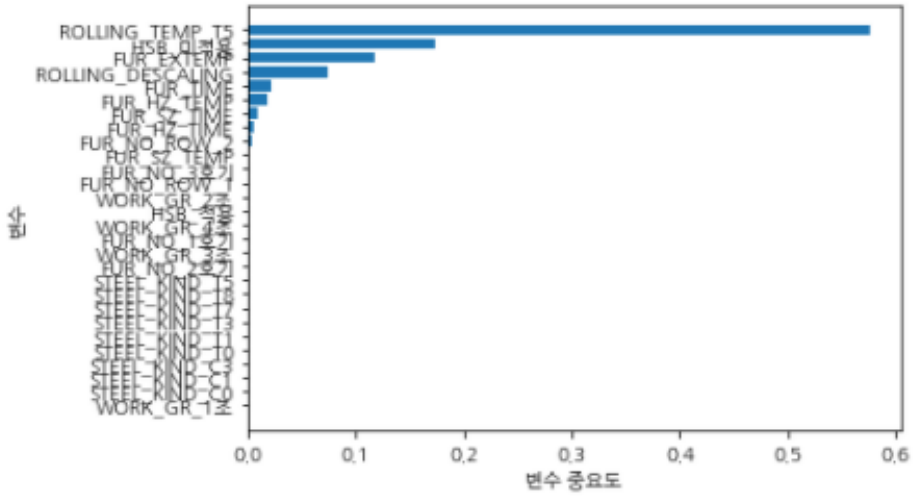
Dep. Variable:	SCALE	No. Observations:	499
Model:	Logit	Df Residuals:	478
Method:	MLE	Df Model:	20
Date:	Tue, 24 Nov 2020	Pseudo R-squ.:	0.5719
Time:	23:28:54	Log-Likelihood:	-134.02
converged:	False	LL-Null:	-313.05
Covariance Type:	nonrobust	LLR p-value:	9.715e-64

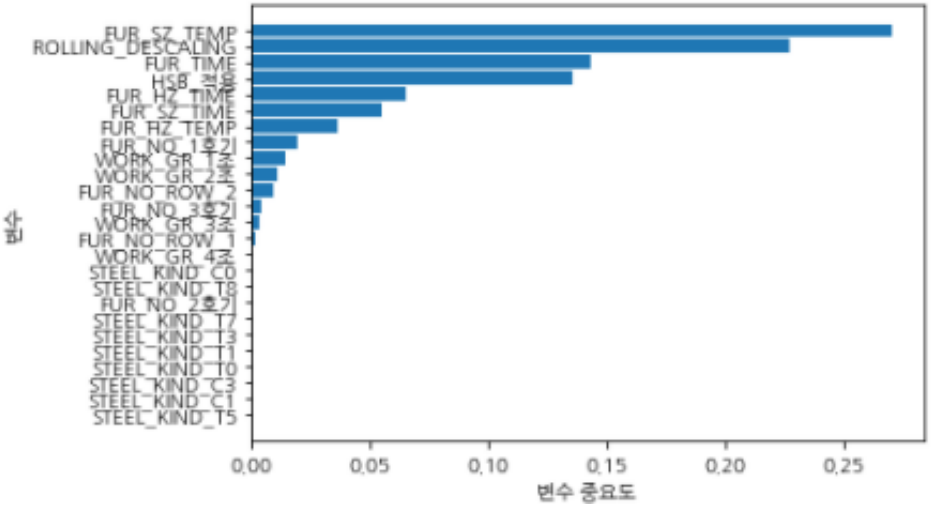
	coef	std err	z	P>|z|	[0.025	0.975]
Intercept	-0.3864	27.515	-0.014	0.989	-54.315	53.542
C(STEEL_KIND)[T.C3]	0.6343	2.804	0.226	0.821	-4.861	6.130
C(STEEL_KIND)[T.T0]	-0.1377	1.383	-0.100	0.921	-2.848	2.573
C(STEEL_KIND)[T.T1]	-2.0077	1.376	-1.459	0.145	-4.705	0.689
C(STEEL_KIND)[T.T5]	-2.7106	1.841	-1.472	0.141	-6.320	0.898
C(STEEL_KIND)[T.T7]	1.5285	1.799	0.850	0.396	-1.998	5.055
C(STEEL_KIND)[T.T8]	-0.7971	1.715	-0.465	0.642	-4.158	2.564
C(FUR_NO)[T.2호기]	-0.1602	0.389	-0.411	0.681	-0.923	0.60
C(FUR_NO)[T.3호기]	0.1953	0.396	0.493	0.622	-0.581	0.97
C(WORK_GR)[T.2조]	-1.4006	0.463	-3.028	0.002	-2.307	-0.494
C(WORK_GR)[T.3조]	-1.8785	0.518	-3.629	0.000	-2.893	-0.864
C(WORK_GR)[T.4조]	-0.8318	0.446	-1.865	0.062	-1.706	0.042
C(HSB)[T.적용]	-13.7805	9.113	-1.512	0.130	-31.641	4.08
FUR_NO_ROW[T.2]	0.0772	0.330	0.234	0.815	-0.569	0.723
FUR_HZ_TEMP	0.0644	0.020	3.157	0.002	0.024	0.104
FUR_HZ_TIME	0.0020	0.005	0.412	0.680	-0.007	0.011
FUR_SZ_TEMP	-0.0425	7.11e+04	-5.98e-07	1.000	-1.39e+05	1.39e+05
FUR_SZ_TIME	-0.0274	0.011	-2.589	0.010	-0.048	-0.007
FUR_TIME	0.0027	0.004	0.623	0.534	-0.006	0.011
FUR_EXTEMP	-0.0425	7.11e+04	-5.98e-07	1.000	-1.39e+05	1.39e+05
ROLLING_TEMP_T5	0.0465	0.007	6.980	0.000	0.033	0.060
ROLLING_DESCALING	-0.7178	0.171	-4.195	0.000	-1.053	-0.382



- 표준화 후, 로지스틱 회귀분석을 통한 변수의 중요도
- 로지스틱 회귀분석은 목표, 설명변수간의 선형관계가 보장될 때 적절한 분석모델
- 또한, 계수들의 p-value 값들이 0.05가 넘는 변수가 다수 존재, 모델의 설명력도 높지 않음
- 따라서, 이 모델로만 판단하는 것은 무리가 있음

분류모델을 이용한 분석 : 의사결정나무





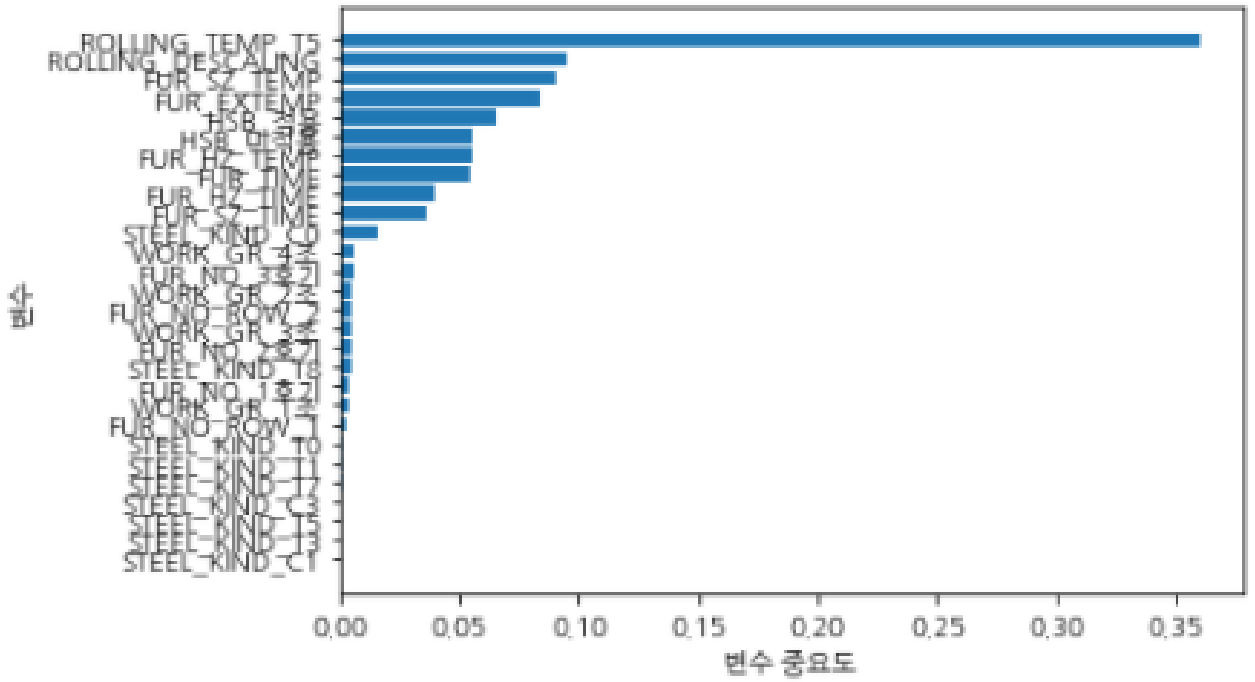
- 표준화 후, 의사결정나무 모델(default)을 이용한 변수의 중요도
- 첫번째 분석 결과, 중요 인자 3개 - ROLLING_TEMP_T5, HSB, FUR_EXTEMP
- 3개의 변수 제외 후 두번째 분석 결과, FUR_SZ_TEMP, ROLLING_DESCALING, FUR_TIME

분류모델을 이용한 분석 : 의사결정나무

Vital Few

- FUR_EXTEMP
- FUR_SZ_TEMP
- ROLLING_TEMP_T5
- FUR_TIME
- HSB
- ROLLING_DESCALING

분류모델을 이용한 분석 : 랜덤 포레스트

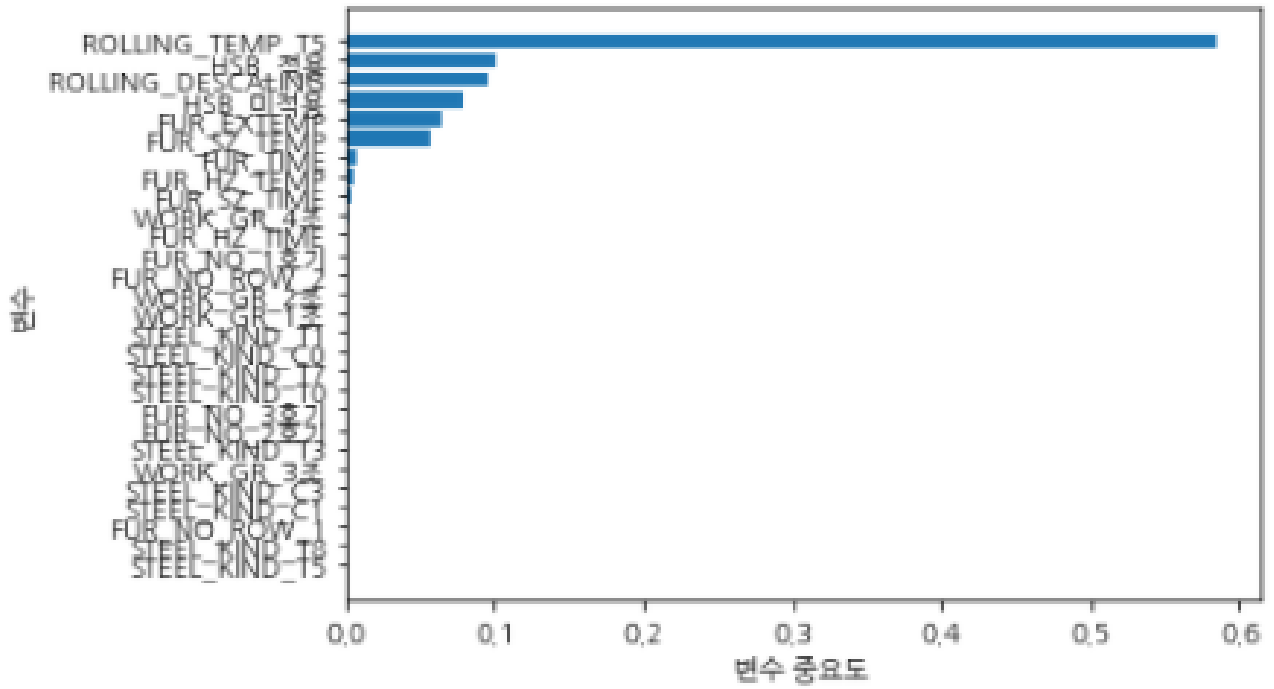


- 표준화 후, 랜덤 포레스트 모델(default)을 이용한 변수의 중요도

Vital Few

- FUR_EXTEMP
- FUR_SZ_TEMP
- ROLLING_TEMP_T5
- FUR_HZ_TEMP
- HSB
- ROLLING_DESCALING

분류모델을 이용한 분석 : 그래디언트 부스팅



- 표준화 후, 그래디언트 부스팅 모델(default)을 이용한 변수의 중요도

Vital Few

- ROLLING_TEMP_T5
- HSB
- ROLLING_DESCALING
- FUR_EXTEMP
- FUR_SZ_TEMP
- FUR_TIME

Vital Few 선정 : 분석 결과 종합

탐색적 분석

- FUR_SZ_TIME
- **FUR_EXTEMP**
- **ROLLING_TEMP_T5**
- STEEL KIND
- **HSB**
- **ROLLING_DESCALING**

의사결정나무

- **FUR_EXTEMP**
- FUR_SZ_TEMP
- **ROLLING_TEMP_T5**
- FUR_TIME
- **HSB**
- **ROLLING_DESCALING**

랜덤 포레스트

- **FUR_EXTEMP**
- FUR_SZ_TEMP
- **ROLLING_TEMP_T5**
- FUR_HZ_TEMP
- **HSB**
- **ROLLING_DESCALING**

그래디언트 부스팅

- **ROLLING_TEMP_T5**
- **HSB**
- **ROLLING_DESCALING**
- **FUR_EXTEMP**
- FUR_SZ_TEMP
- FUR_TIME

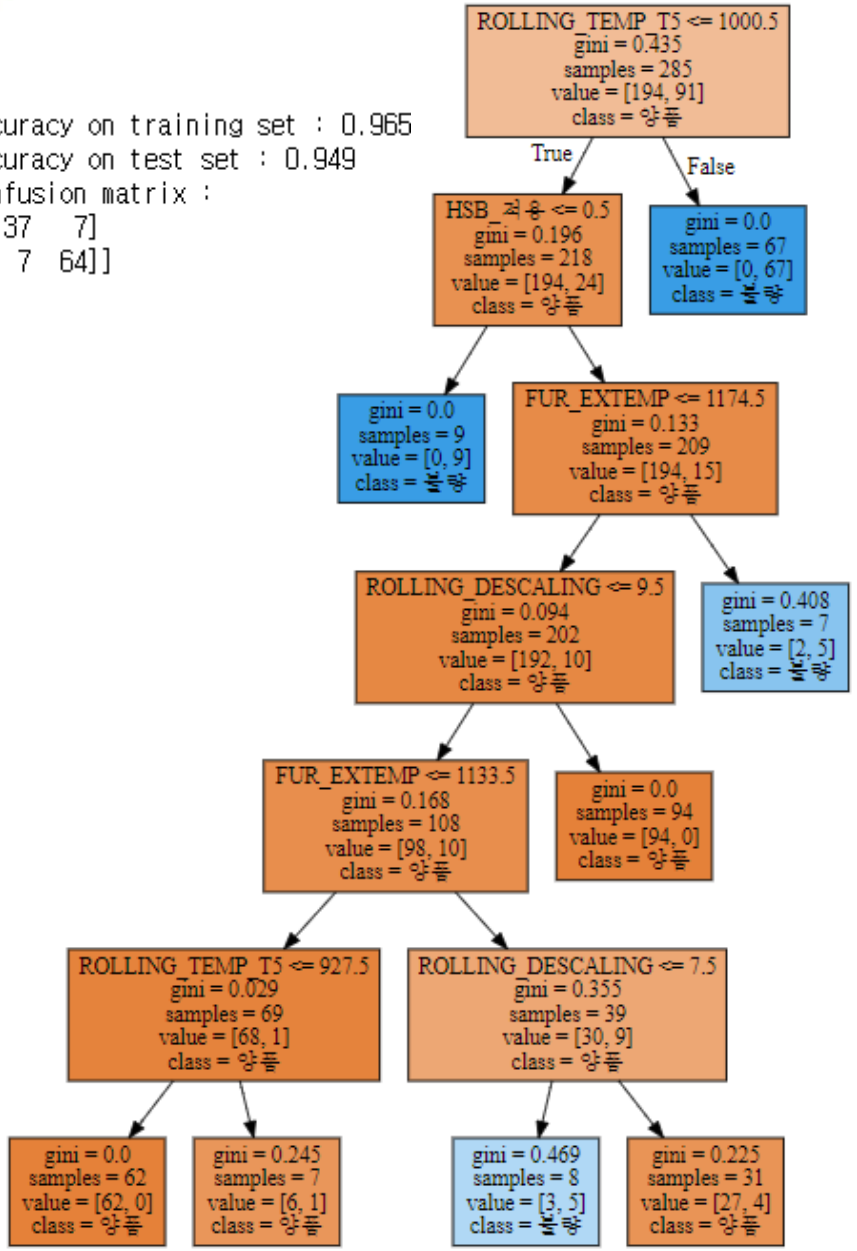
Vital Few 선정 : 최종 Vital Few

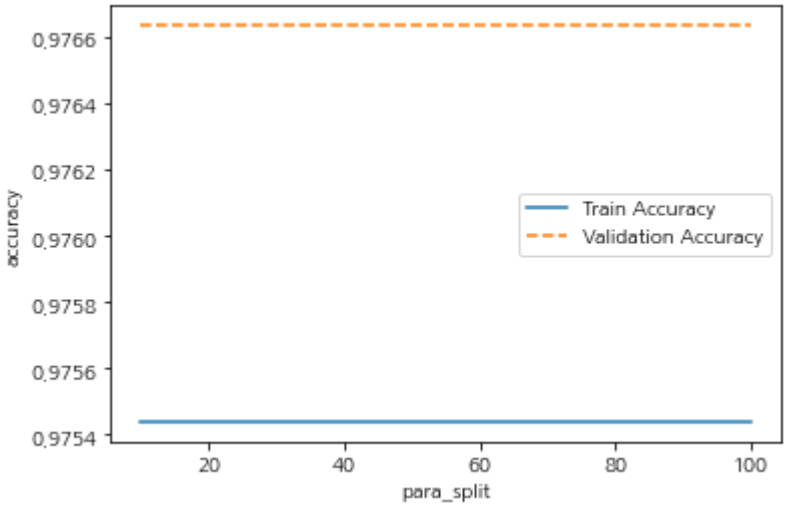
최종 Vital Few

- ROLLING_TEMP_T5
- ROLLING_DESCALING
- HSB
- FUR_EXTEMP

Solution : 최종 분류 모델 생성

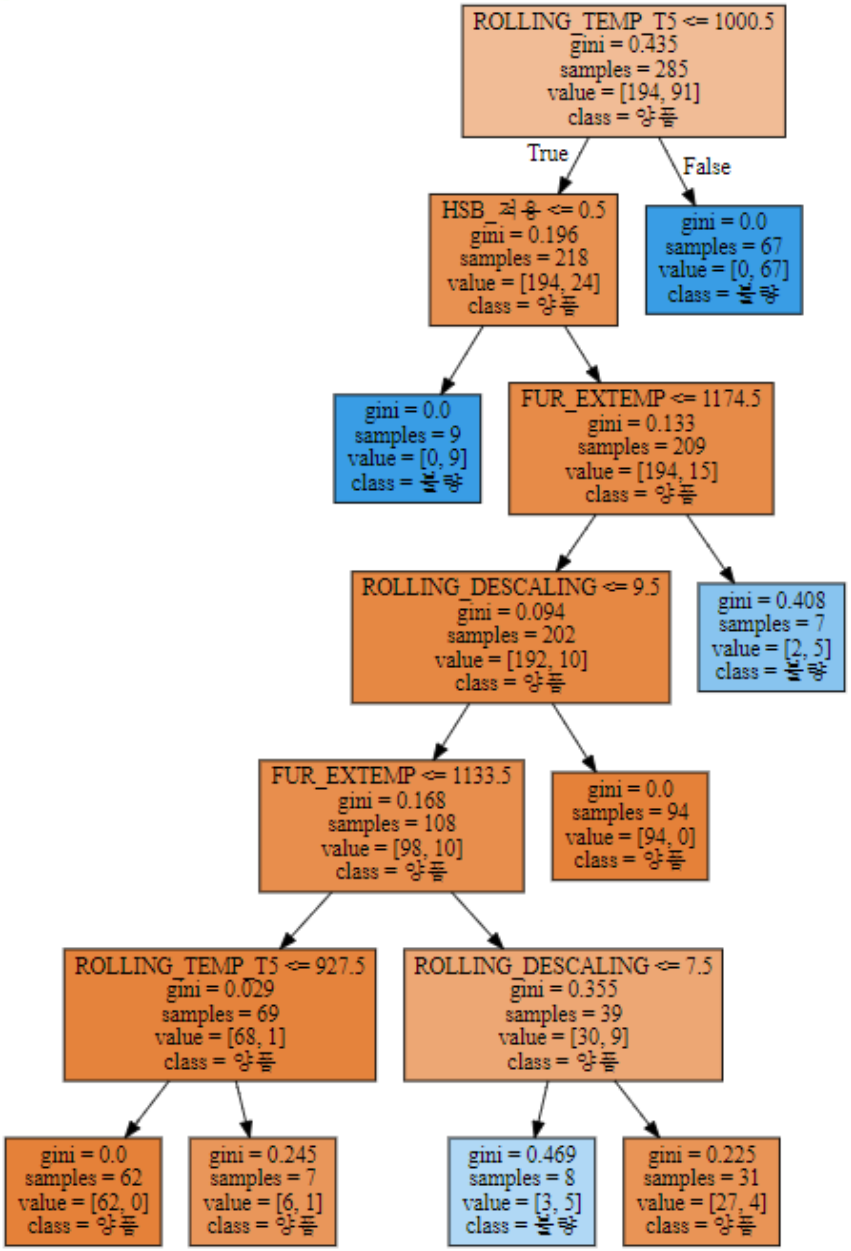
```
Accuracy on training set : 0.965
Accuracy on test set : 0.949
Confusion matrix :
[[137   7]
 [  7  64]]
```





- 최종 Vital Few로 의사결정나무 모델 생성
- Max_depth = 6
- Min_samples_leaf = 7
- Min_samples_split 은 정확도에 영향을 주지 않음
- 20개의 변수 중에서 4개만 사용했다는 점을 감안, 약 95%의 정확도는 양호하다고 생각
- 하지만, 71개의 불량품 중 7개를 잘못 판단

Solution : 개선 방향



- 압연온도 1000도 이하, 추출온도 1130도 이하
  -> 불량률 상당부분 개선될 것

- ROLLING_DESCALING 7회 이하
  -> HSB 적용해도 불량품 발생확률 증가

- HSB를 적용, ROLLING_DESCALING 8회 이상
  -> 불량률 현저히 낮아질 것

기대 효과

	Q	R	S	T	U	V	W	X
1	FUR_EXTEI	ROLLING_	HSB	ROLLING_	WORK_GR			
2	1118	897	적용	8	2조		MIN, T5	750
3	1115	930	적용	8	2조		MAX, T5	1000
2	1126	971	적용	8	3조		MIN, EX	1110
5	1110	865	적용	8	3조		MAX, EX	1130
4	1123	928	적용	8	1조			
4	1124	790	적용	8	4조			
0	1129	934	적용	8	2조			
4	1127	870	적용	8	4조			
3	1125	801	적용	8	1조			
3	1114	842	적용	8	4조			
7	1127	975	적용	8	2조			
0	1128	800	적용	8	2조			

```
In [114]: # 평가
          y_pred_2 = tree_uncustomized.predict(df_x)

In [117]: y_pred_2 = pd.DataFrame(y_pred_2)
          y_pred_2[0].value_counts()

Out[117]: 0    634
          1     80
          Name: 0, dtype: int64

In [118]: defective_rate=80/634 * 100
          defective_rate

Out[118]: 12.618296529968454
```

- 데이터 원본에서 **압연온도 750~1000도**, **추출온도 1110~1130도**로 수정한 새로운 데이터 생성
- 다른 데이터 값은 수정하지 않음, 결측치 6개 제거
- 의사결정나무 모델(default)에 적용
- 716개의 데이터 중 80개의 불량품 발생
- 불량률 : 32.4% -> **12.6%**
- 즉, 기존의 20개의 변수 중 2개의 변수만 조절해준다면 **20%의 불량률 개선 효과**를 볼 수 있음