



종합 실습 2

POSCO: Scale 불량 분석

A반 1조
최우혁

주어진 데이터의 설명변수는 제품의 종류를 나타내는 SPEC과 공정 날짜를 나타내는 ROLLING DATE를 포함하여 총 13개가 있음.

SPEC을 더미 변수로 변환하여 전체 데이터를 그룹으로 나눌 수도 있겠지만 다른 변수들이 SPEC이 제공하는 정보를 좀 더 자세히 제공하며, 아래 표에서도 볼 수 있듯이 FUR_SZ_TEMP의 미세한 차이를 SPEC이 분별하지 못하기 때문에 SPEC 변수를 제외.

ROLLING DATE에 따른 차이는 데이터가 커질수록 상쇄 가능하다고 생각되며, 설명 변수로써 사용하더라도 조절을 할 수가 없어서 변수로 사용할 이유가 없다고 생각되어 ROLLING DATE 역시 제외.

	SCALE	SPEC	STEEL_KIND	PT_THK	PT_WIDTH	PT_LTH	PT_WGT	FUR_NO	FUR_NO_ROW	FUR_HZ_TEMP	FUR_HZ_TIME	FUR_SZ_TEMP	FUR_SZ_TIME	FUR_TIME	FUR_EXTEMP	ROLLING_TEMP_T5	HSB	ROLLING_DESCALING	WORK_GR
0	양품	AB/EH32-TM	T1	32.25	3707	15109	14180	1호기	1	1144	116	1133	59	282	1133	934	적용	8	2조
1	양품	AB/EH32-TM	T1	32.25	3707	15109	14180	1호기	2	1144	122	1135	53	283	1135	937	적용	8	2조
2	양품	NV-E36-TM	T8	33.27	3619	19181	18130	2호기	1	1129	116	1121	55	282	1121	889	적용	8	3조
3	양품	NV-E36-TM	T8	33.27	3619	19181	18130	2호기	2	1152	125	1127	68	316	1127	885	적용	8	3조
4	양품	BV-EH36-TM	T8	38.33	3098	13334	12430	3호기	1	1140	134	1128	48	314	1128	873	적용	8	1조

SPEC과 ROLLING DATE를 제외한 raw 데이터

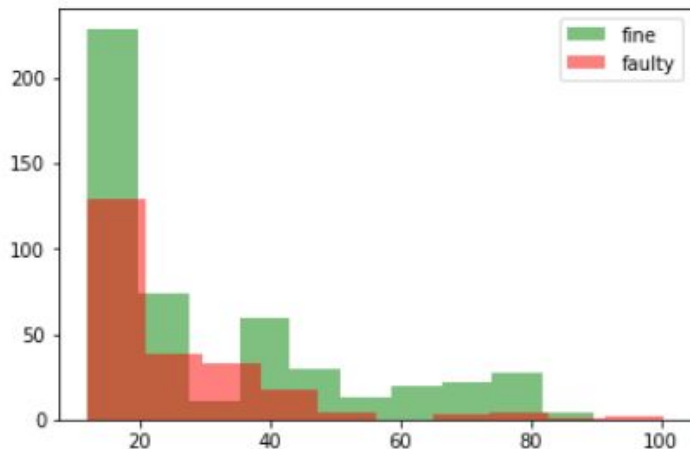
1	df.describe()												
	PT_THK	PT_WIDTH	PT_LTH	PT_WGT	FUR_NO_ROW	FUR_HZ_TEMP	FUR_HZ_TIME	FUR_SZ_TEMP	FUR_SZ_TIME	FUR_TIME	FUR_EXTEMP	ROLLING_TEMP_T5	ROLLING_DESCALING
count	720.000000	720.000000	720.000000	720.000000	720.000000	720.000000	720.000000	720.000000	720.000000	720.000000	720.000000	720.000000	720.000000
mean	29.267857	2807.127778	34323.269444	41481.777778	1.497222	1155.338889	88.527778	1149.666667	78.802778	300.563889	1149.666667	933.920833	8.319444
std	19.152651	517.839690	13828.434837	24751.322015	0.500340	20.866398	37.660799	17.571788	36.421004	45.390303	17.571788	107.863887	1.623252
min	12.030000	1800.000000	7893.000000	7820.000000	1.000000	1103.000000	43.000000	1113.000000	35.000000	221.000000	1113.000000	0.000000	5.000000
25%	16.090000	2466.000000	22663.000000	17757.500000	1.000000	1139.000000	65.000000	1132.000000	57.750000	257.000000	1132.000000	889.750000	6.000000
50%	20.130000	2725.000000	37380.000000	36120.000000	1.000000	1158.000000	75.000000	1155.000000	67.500000	304.000000	1155.000000	951.000000	8.000000
75%	40.000000	3125.000000	43735.000000	61120.000000	2.000000	1171.000000	101.000000	1164.000000	86.000000	342.000000	1164.000000	994.250000	10.000000
max	100.340000	4575.000000	54917.000000	115080.000000	2.000000	1206.000000	312.000000	1185.000000	294.000000	398.000000	1185.000000	1078.000000	10.000000

1	df.isnull().sum()	
SCALE		0
SPEC		0
STEEL_KIND		0
PT_THK		0
PT_WIDTH		0
PT_LTH		0
PT_WGT		0
FUR_NO		0
FUR_NO_ROW		0
FUR_HZ_TEMP		0
FUR_HZ_TIME		0
FUR_SZ_TEMP		0
FUR_SZ_TIME		0
FUR_TIME		0
FUR_EXTEMP		0
ROLLING_TEMP_T5		0
HSB		0
ROLLING_DESCALING		0
WORK_GR		0

데이터의 기본 통계를 확인했을 때, 특이사항은 발견되지 않으며 결측치도 없는것으로 확인됨.

양품과 불량품 데이터는 각 489개, 231개로 약 2:1의 비율임.

각 설명 변수에 대해서 '양품'과 '불량품'의 경향과 이상치를 함께 검사하기 위해서 히스토그램과 각 성분에 대한 기초 통계를 사용하여 데이터를 확인한다



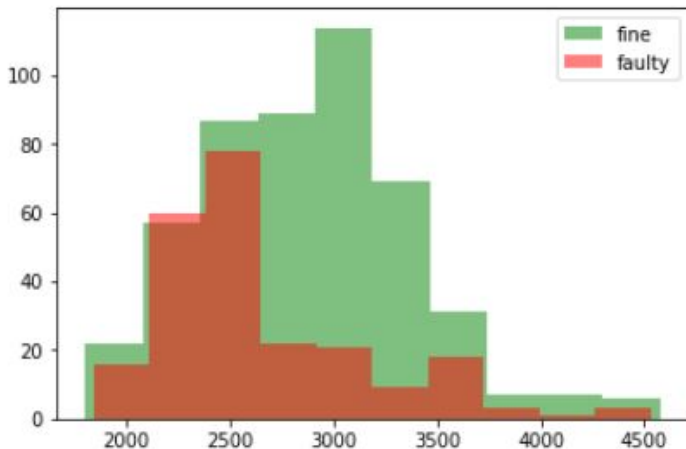
```
1 df[df['SCALE'] == '양품']['PT_THK'].describe()
count    489.000000
mean      30.910321
std       20.831402
min       12.030000
25%       15.080000
50%       20.130000
75%       40.350000
max       89.630000
Name: PT_THK, dtype: float64
```

```
1 df[df['SCALE'] != '양품']['PT_THK'].describe()
count    231.000000
mean      25.790952
std       14.426501
min       12.050000
25%       18.110000
50%       20.130000
75%       30.130000
max      100.340000
Name: PT_THK, dtype: float64
```

PT_THK

양품의 분포가 더 고르게 넓어보이지만 양품과 불량품의 비율 차이로 생길 수 있다고 생각하며, 양품과 불량품이 대체로 비슷한 분포를 보이고 있다는 점에서 PT_THK를 유의미한 파라미터로 놓기 어려울 것으로 보임.

각 설명 변수에 대해서 '양품'과 '불량품'의 경향과 이상치를 함께 검사하기 위해서 히스토그램과 각 성분에 대한 기초 통계를 사용하여 데이터를 확인한다



```
1 df[df['SCALE'] == '양품']['PT_WIDTH'].describe()
```

```
count    489.000000
mean     2880.402863
std       508.477359
min      1800.000000
25%      2511.000000
50%      2875.000000
75%      3151.000000
max      4575.000000
Name: PT_WIDTH, dtype: float64
```

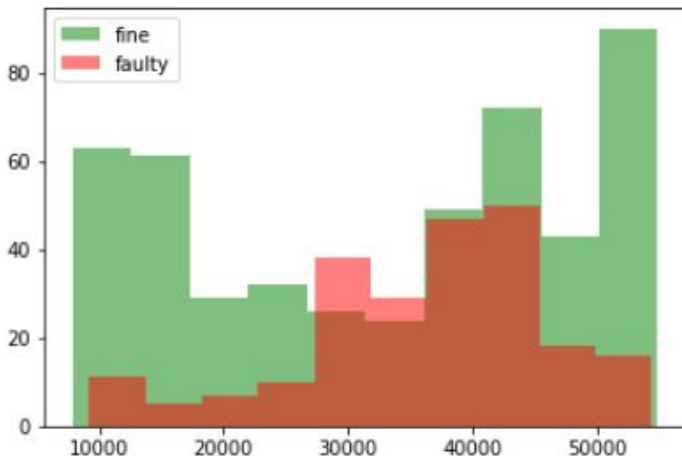
```
1 df[df['SCALE'] != '양품']['PT_WIDTH'].describe()
```

```
count    231.000000
mean     2652.012987
std       504.113449
min      1840.000000
25%      2288.000000
50%      2516.000000
75%      2792.000000
max      4528.000000
Name: PT_WIDTH, dtype: float64
```

PT_WIDTH

75% 이상의 불량품은 양품 PT_WIDTH의 평균값보다 아래에 분포하고 있음. 이로 인해 양품과 불량품의 평균은 200 이상, 약 10%의 차이를 보인다. PT_WIDTH는 유의미한 분류 기준으로 생각됨.

각 설명 변수에 대해서 '양품'과 '불량품'의 경향과 이상치를 함께 검사하기 위해서 히스토그램과 각 성분에 대한 기초 통계를 사용하여 데이터를 확인한다



```
1 df[df['SCALE'] == '양품']['PT_LTH'].describe()
```

```
count    489.000000
mean     33368.981595
std      15232.024856
min       7893.000000
25%      16740.000000
50%      37198.000000
75%      47067.000000
max      54917.000000
Name: PT_LTH, dtype: float64
```

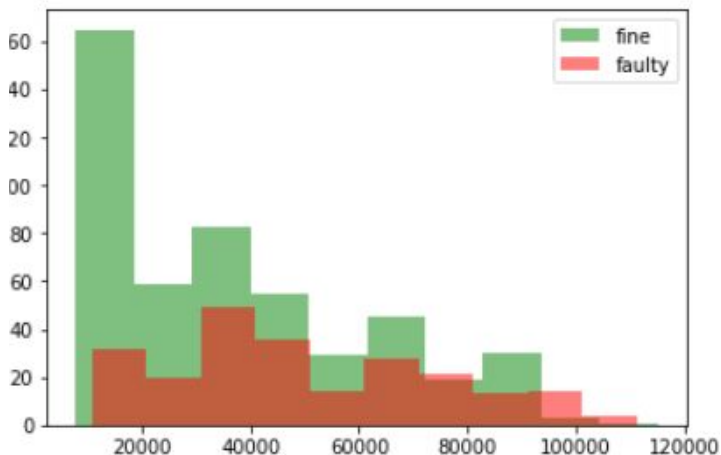
```
1 df[df['SCALE'] != '양품']['PT_LTH'].describe()
```

```
count     231.000000
mean     36343.385281
std       9973.901193
min       9175.000000
25%      30730.000000
50%      37440.000000
75%      42824.500000
max      54458.000000
Name: PT_LTH, dtype: float64
```

PT_LTH

LTH가 30000~45000을 중심으로 불량품이 분포해있으며, 양품은 이와 반대로 해당 구간에서 상대적으로 적은 분포를 보인다. PT_LTH 역시 유의미할 것으로 예상.

각 설명 변수에 대해서 '양품'과 '불량품'의 경향과 이상치를 함께 검사하기 위해서 히스토그램과 각 성분에 대한 기초 통계를 사용하여 데이터를 확인한다



```
1 df[df['SCALE'] == '양품']['PT_WGT'].describe()

count      489.000000
mean       37763.190184
std        24036.486107
min         7820.000000
25%        15790.000000
50%        32840.000000
75%        51760.000000
max        115080.000000
Name: PT_WGT, dtype: float64
```

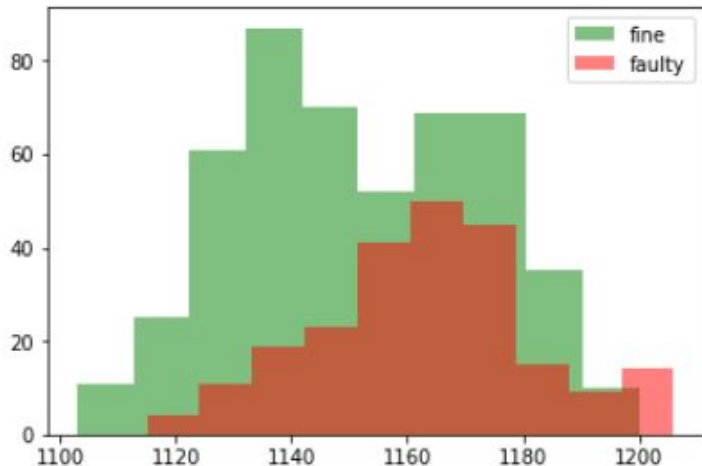
```
1 df[df['SCALE'] != '양품']['PT_WGT'].describe()

count      231.000000
mean       49353.593074
std        24446.954529
min        10670.000000
25%        33060.000000
50%        42720.000000
75%        66250.000000
max        111200.000000
Name: PT_WGT, dtype: float64
```

PT_WGT

양품의 경우 WGT가 낮을수록 빈도가 높아지지만, 불량품의 경우 상대적으로 골고루 분포해있어서 WGT를 이용하여 이 둘을 분간하기는 어려울 것으로 생각됨. 따라서 PT_WGT를 제외.

각 설명 변수에 대해서 '양품'과 '불량품'의 경향과 이상치를 함께 검사하기 위해서 히스토그램과 각 성분에 대한 기초 통계를 사용하여 데이터를 확인한다



```
df[df['SCALE'] == '양품']['FUR_HZ_TEMP'].describe
```

```
unt    489.000000
an     1151.789366
d       21.075358
n     1103.000000
%     1135.000000
%     1151.000000
%     1169.000000
x     1200.000000
me: FUR_HZ_TEMP, dtype: float64
```

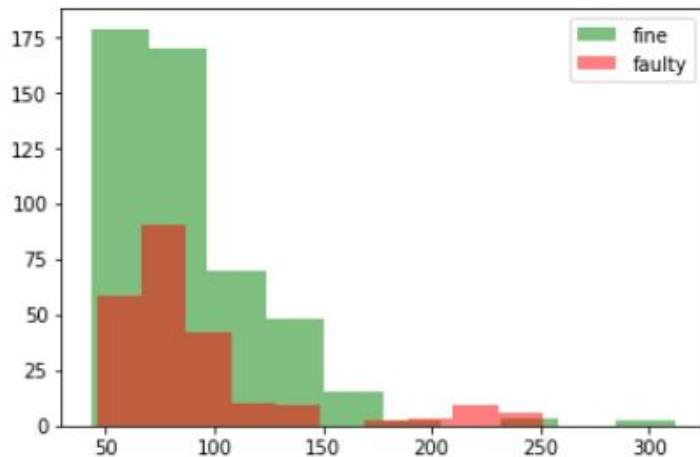
```
df[df['SCALE'] != '양품']['FUR_HZ_TEMP'].describe
```

```
unt     231.000000
an     1162.852814
d       18.308824
n     1115.000000
%     1152.500000
%     1163.000000
%     1172.500000
x     1206.000000
me: FUR_HZ_TEMP, dtype: float64
```

FUR_HZ_TEMP

그래프로 볼 때 약간의 차이는 있지만, 분석값을 보았을 때, 평균값 1% 미만 차이가 있으며, 표준편차 역시 큰 차이가 없어서 해당 파라미터에 대한 유의미한 분석은 어려울 것으로 보임.

각 설명 변수에 대해서 '양품'과 '불량품'의 경향과 이상치를 함께 검사하기 위해서 히스토그램과 각 성분에 대한 기초 통계를 사용하여 데이터를 확인한다



```
1 df[df['SCALE'] == '양품']['FUR_HZ_TIME'].describe()

count    489.000000
mean      87.075665
std       34.110242
min        43.000000
q1        64.000000
q2        74.000000
q3       106.000000
max       312.000000
Name: FUR_HZ_TIME, dtype: float64
```

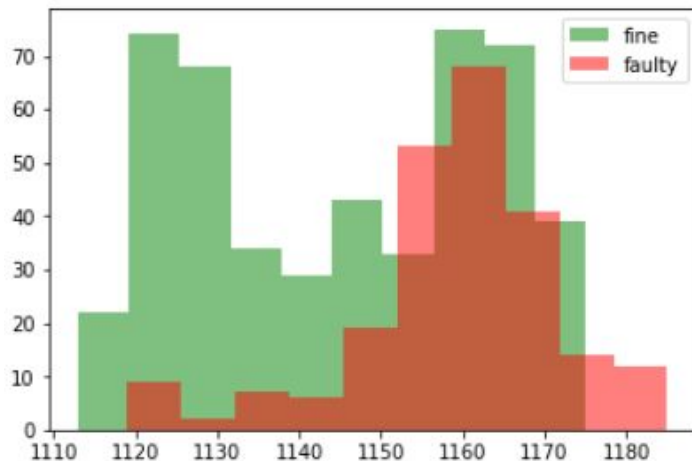
```
1 df[df['SCALE'] != '양품']['FUR_HZ_TIME'].describe()

count    231.000000
mean      91.601732
std       44.172456
min        46.000000
q1        66.000000
q2        77.000000
q3        96.000000
max       251.000000
Name: FUR_HZ_TIME, dtype: float64
```

FUR_HZ_TIME

그래프로 볼 때 약간의 차이는 있지만, 분석값을 보았을 때, 평균값 1% 미만 차이가 있으며, 표준편차 역시 큰 차이가 없어서 해당 파라미터에 대한 유의미한 분석은 어려울 것으로 보임.

각 설명 변수에 대해서 '양품'과 '불량품'의 경향과 이상치를 함께 검사하기 위해서 히스토그램과 각 성분에 대한 기초 통계를 사용하여 데이터를 확인한다



```
1 df[df['SCALE'] == '양품']['FUR_SZ_TEMP'].describe()

count    489.000000
mean     1145.143149
std       17.772237
min       1113.000000
max       1127.000000
%ile     1147.000000
%ile     1162.000000
%ile     1175.000000
name: FUR_SZ_TEMP, dtype: float64
```

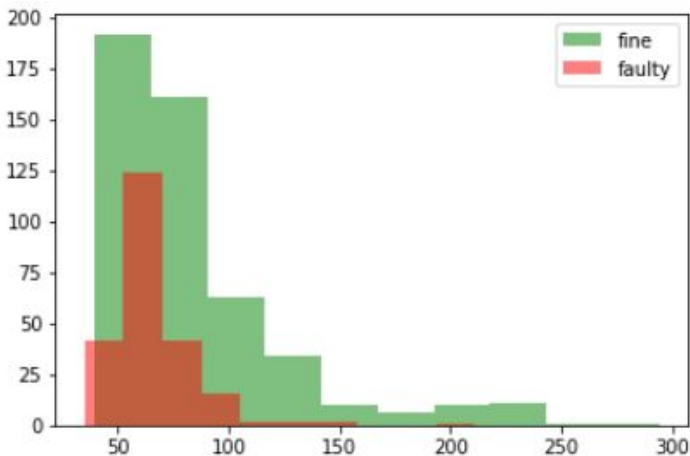
```
1 df[df['SCALE'] != '양품']['FUR_SZ_TEMP'].describe()

count    231.000000
mean     1159.242424
std       12.628543
min       1119.000000
max       1154.000000
%ile     1161.000000
%ile     1166.000000
%ile     1185.000000
name: FUR_SZ_TEMP, dtype: float64
```

FUR_SZ_TEMP

FUR_SZ_TEMP에 의하면 불량품은 낮은~중간 범위에서 낮은 빈도를 보이며, 양품은 낮은, 높은 두 범위에서 높은 빈도를 보임. 수치로는 큰 차이를 알 수 없지만, 회귀분석에 분석할 시 유의미한 차이를 찾을 수 있을 것 같으므로, VITAL FEW로 추가.

각 설명 변수에 대해서 '양품'과 '불량품'의 경향과 이상치를 함께 검사하기 위해서 히스토그램과 각 성분에 대한 기초 통계를 사용하여 데이터를 확인한다



```
df[df['SCALE'] == '양품']['FUR_SZ_TIME'].describe()
```

```
count    489.000000
mean     84.809816
std      40.769075
min      40.000000
max      40.000000
%25      59.000000
%50      71.000000
%75      95.000000
%100     294.000000
Name: FUR_SZ_TIME, dtype: float64
```

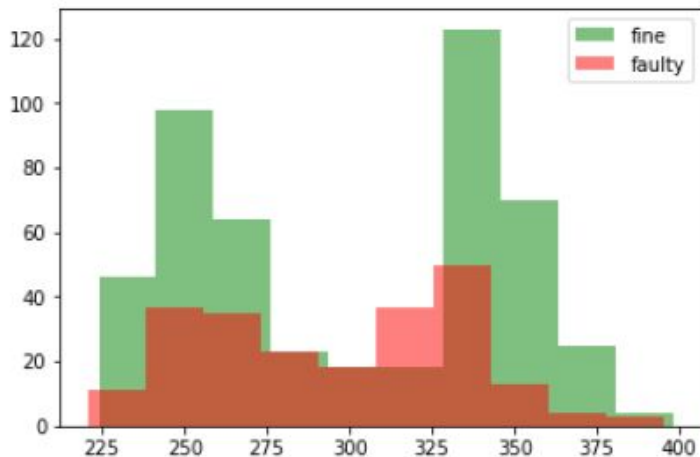
```
df[df['SCALE'] != '양품']['FUR_SZ_TIME'].describe()
```

```
count    231.000000
mean     66.086580
std      19.519697
min      35.000000
max      55.000000
%25      62.000000
%50      72.000000
%75     211.000000
%100     211.000000
Name: FUR_SZ_TIME, dtype: float64
```

FUR_SZ_TIME

FUR_SZ_TIME의 경우 평균과 표준편차에서 상대적으로 큰 차이를 보이지만, 분포의 생김새는 양품과 불량품이 흡사한 것으로 보임. 분포가 닮았다면 지금의 차이는 샘플의 데이터 부족으로 설명할 수 있으며, 따라서 임의로 유의미한 분석이 어려울 것으로 생각됨.

각 설명 변수에 대해서 '양품'과 '불량품'의 경향과 이상치를 함께 검사하기 위해서 히스토그램과 각 성분에 대한 기초 통계를 사용하여 데이터를 확인한다



```
1 df[df['SCALE'] == '양품']['FUR_TIME'].describe()
```

```
count    489.000000
mean     302.098160
std       47.382886
min      224.000000
25%      256.000000
50%      308.000000
75%      342.000000
max      398.000000
Name: FUR_TIME, dtype: float64
```

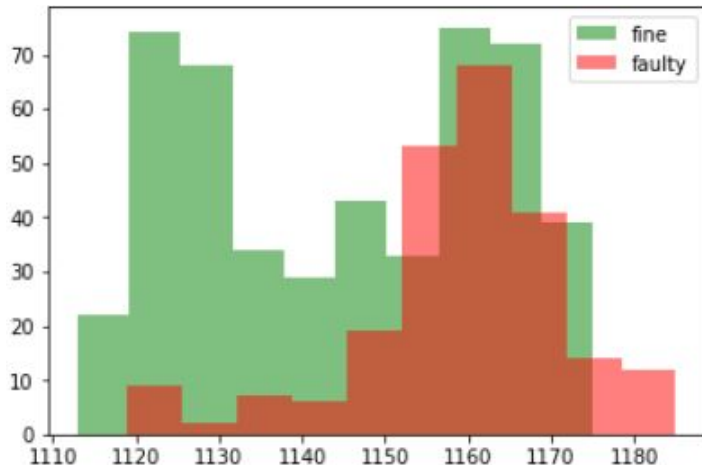
```
1 df[df['SCALE'] != '양품']['FUR_TIME'].describe()
```

```
count    231.000000
mean     297.316017
std       40.760484
min      221.000000
5%       259.500000
0%       299.000000
5%       342.000000
max      395.000000
ame: FUR TIME, dtype: float64
```

FUR_TIME

FUR_TIME 역시 양품과 불량품이 유사한 그래프 분포를 띈다. 양품이 중간 범위에서 상대적으로 적은 분포를 보여주지만 유의미한 분석은 상대적으로 어려울 것으로 생각되므로 FUR_TIME 제외

각 설명 변수에 대해서 '양품'과 '불량품'의 경향과 이상치를 함께 검사하기 위해서 히스토그램과 각 성분에 대한 기초 통계를 사용하여 데이터를 확인한다



```
1 df[df['SCALE'] == '양품']['FUR_EXTEMP'].describe
```

```
count    489.000000
mean     1145.143149
std       17.772237
min      1113.000000
5%       1127.000000
10%      1147.000000
50%      1162.000000
90%      1175.000000
max      1175.000000
Name: FUR_EXTEMP, dtype: float64
```

```
1 df[df['SCALE'] != '양품']['FUR_EXTEMP'].describe
```

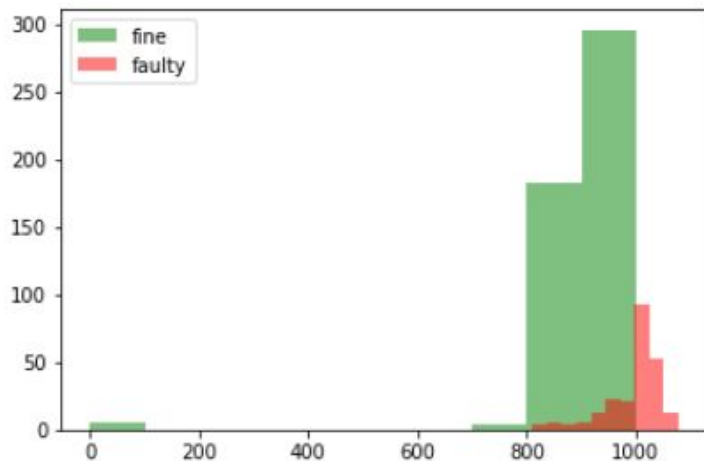
```
count     231.000000
mean     1159.242424
std       12.628543
min      1119.000000
5%       1154.000000
10%      1161.000000
50%      1166.000000
90%      1185.000000
max      1185.000000
Name: FUR_EXTEMP, dtype: float64
```

FUR_EXTEMP

평균, 표준편차 모두 큰 차이가 나지 않지만, Q1, Q2에서 큰 차이를 보임. 분포 자체에 차이가 있으므로 유의미한 파라미터로 보임

+ 비슷한 형태의 분포가 반복해서 등장함, 다중공선성이 있을 것

각 설명 변수에 대해서 '양품'과 '불량품'의 경향과 이상치를 함께 검사하기 위해서 히스토그램과 각 성분에 대한 기초 통계를 사용하여 데이터를 확인한다



```
df[df['SCALE'] == '양품']['ROLLING_TEMP_T5'].describe
```

```
t      489.000000
      905.400818
      115.678978
         0.000000
      873.000000
      925.000000
      964.000000
      1000.000000
: ROLLING_TEMP_T5, dtype: float64
```

```
df[df['SCALE'] != '양품']['ROLLING_TEMP_T5'].describe
```

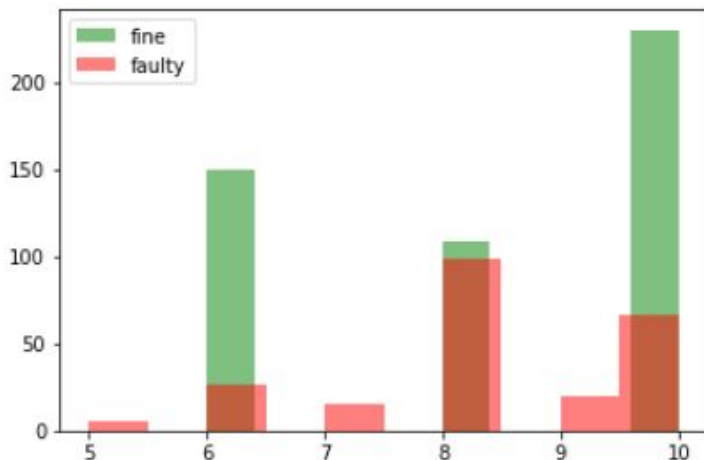
```
t      231.000000
      994.294372
         50.875893
      811.000000
      973.000000
      1008.000000
      1026.000000
      1078.000000
: ROLLING_TEMP_T5, dtype: float64
```

ROLLING_TEMP_T5

- + ROLLING_TEMP_T5 <= 100에서 이상치 의심 샘플이 관측됨.
- + ROLLING_TEMP_T5 >= 1000에서 불량품만 관측됨.

불량품의 경우 평균이 이상치를 제거한 양품의 평균보다 10%이상 높은 수치를 보임. 유의미한 관찰 대상으로 생각됨

각 설명 변수에 대해서 '양품'과 '불량품'의 경향과 이상치를 함께 검사하기 위해서 히스토그램과 각 성분에 대한 기초 통계를 사용하여 데이터를 확인한다



```
if(df['SCALE'] == '양품')['ROLLING_DESCALING'].desc
```

```
489.000000
8.327198
1.734208
6.000000
6.000000
8.000000
10.000000
10.000000
```

ROLLING_DESCALING, dtype: float64

```
if(df['SCALE'] != '양품')['ROLLING_DESCALING'].desc
```

```
231.000000
8.303030
1.362204
5.000000
8.000000
8.000000
10.000000
10.000000
```

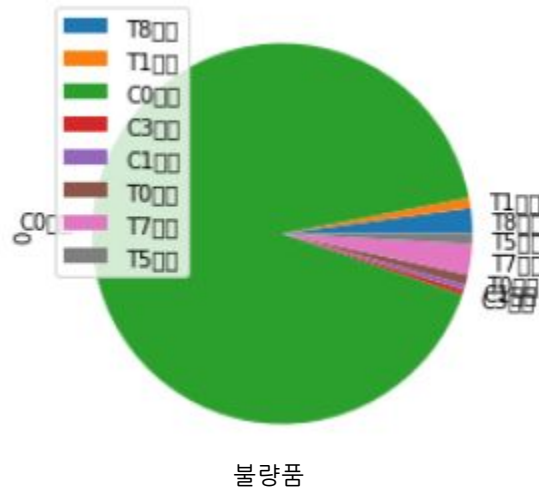
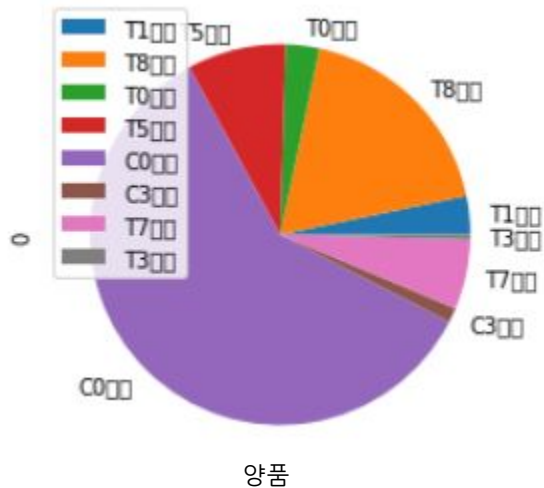
ROLLING_DESCALING, dtype: float64

ROLLING_DESCALING

+ ROLLING_DESCALING이 홀수일 때는 불량품만 관측됨.

하지만 불량품이어서 DESCALING을 더 했을 수도 있으므로 (인과 관계 반대) 해당 insight는 사용 불가 이외에는 큰 차이는 찾을 수 없으므로 ROLLING_DESCALING도 제외.

각 설명 변수에 대해서 '양품'과 '불량품'의 경향과 이상치를 함께 검사하기 위해서 히스토그램과 각 성분에 대한 기초 통계를 사용하여 데이터를 확인한다

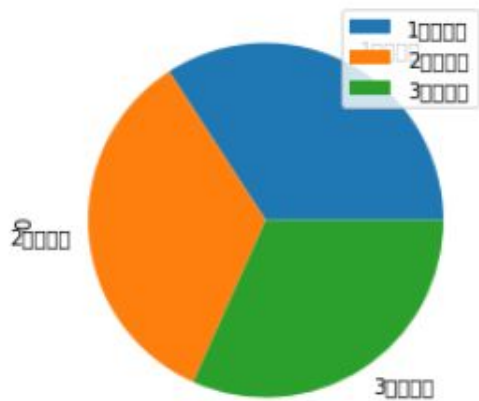


STEEL_KIND

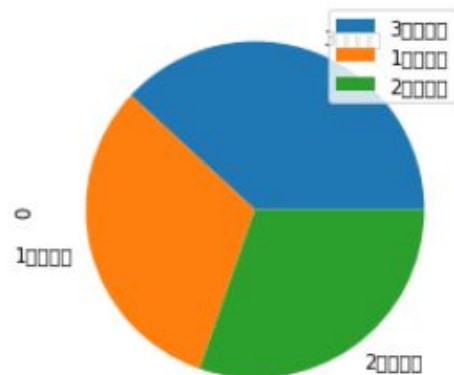
불량품의 대부분을 차지하는 것은 C0로 확인됨. 하지만 주문생산품에 대한 조절은 불가능하므로, 해당 파라미터는 VITAL FEW에서 제외.

종합 실습 2

각 설명 변수에 대해서 '양품'과 '불량품'의 경향과 이상치를 함께 검사하기 위해서 히스토그램과 각 성분에 대한 기초 통계를 사용하여 데이터를 확인한다



양품



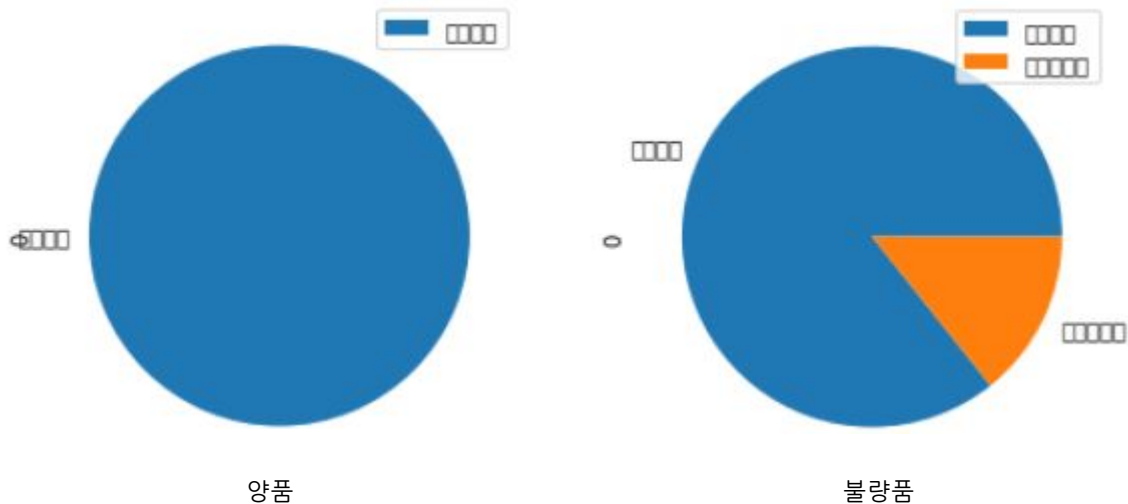
불량품

FUR_NO_ROW

양품과 불량에 대한 유의미한 차이가 파이 플롯에서 보이지 않아서 FUR_NO_ROW 생략.

종합 실습 2

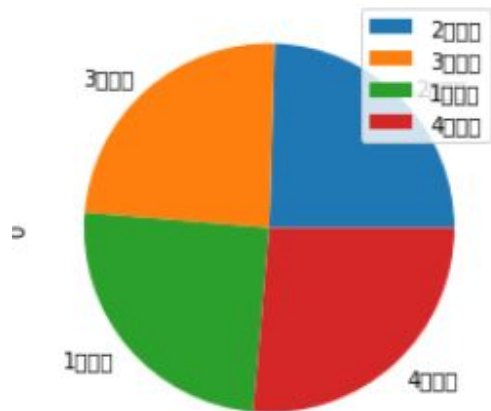
각 설명 변수에 대해서 '양품'과 '불량품'의 경향과 이상치를 함께 검사하기 위해서 히스토그램과 각 성분에 대한 기초 통계를 사용하여 데이터를 확인한다



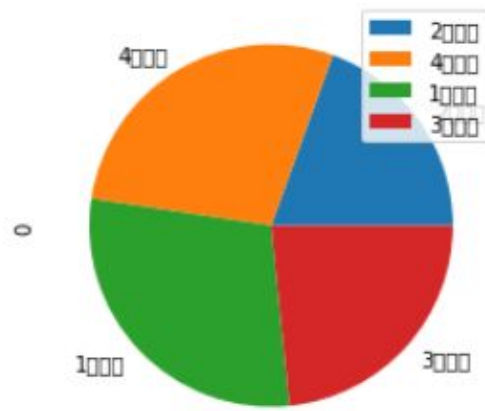
HSB

HSB 미적용 샘플은 모두 불량품에서만 발견됨. HSB를 파라미터로서 채택.

각 설명 변수에 대해서 '양품'과 '불량품'의 경향과 이상치를 함께 검사하기 위해서 히스토그램과 각 성분에 대한 기초 통계를 사용하여 데이터를 확인한다



양품



불량품

WORK_GR

4번 그룹에서 타 그룹 대비 불량품의 비율이 타 그룹 대비 양품의 비율보다 조금 증가하지만,

1. 그 차이가 작으며,
2. 도메인 전문가가 워크 그룹 간의 차이에 대해 설명하지 않음

위의 두 이유로 해당 파라미터는 생략함

VITAL FEW

- PT_WIDTH
- PT_LENGTH
- FUR_SZ_TEMP
- FUR_EXTEMP
- ROLLING_TEMP_T5
- HSB

이상치 처리

- $ROLLING_TEMP_T5 < 100$

위의 데이터 전 처리 과정 후, 정확한 모델을 얻기 위해, 그리고 스케일을 필요로 하는 모델을 사용하기 위해서 데이터를 스케일 해준다.

	PT_WDTH	PT_LTH	FUR_SZ_TEMP	FUR_EXTEMP	ROLLING_TEMP_T5	HSB_미적용	HSB_적용
0	1.738825	-1.399249	-0.958762	-0.958762	-0.118132	-0.220132	0.220132
1	1.738825	-1.399249	-0.844450	-0.844450	-0.072514	-0.220132	0.220132
2	1.568260	-1.103953	-1.644633	-1.644633	-0.802387	-0.220132	0.220132
3	1.568260	-1.103953	-1.301698	-1.301698	-0.863210	-0.220132	0.220132
4	0.558436	-1.527970	-1.244542	-1.244542	-1.045678	-0.220132	0.220132

로지스틱 회귀분석

설명력

0.524

유의수준 0.05에서 유의한 변수

PT_LTH, PT_WDTH, ROLLING_TEMP_T5

Test Set에 대한 정확도: 0.829

Confusion Matrix:

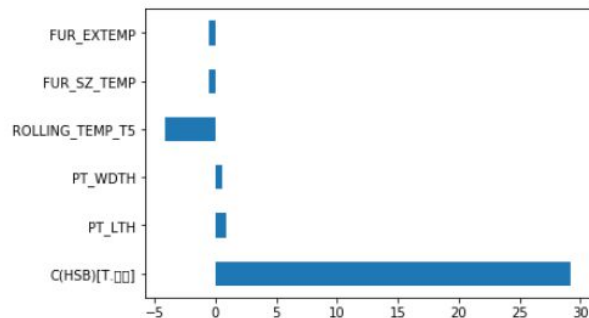
62	9
28	117

기타 모델들보다 정확성이 떨어짐

Logit Regression Results

Dep. Variable:	N_SCALE	No. Observations:	504
Model:	Logit	Df Residuals:	498
Method:	MLE	Df Model:	5
Date:	Sun, 12 May 2019	Pseudo R-squ.:	0.5244
Time:	02:50:51	Log-Likelihood:	-149.81
converged:	False	LL-Null:	-314.97
		LLR p-value:	3.028e-69

	coef	std err	z	P> z	[0.025	0.975]
Intercept	-26.7952	7.03e+04	-0.000	1.000	-1.38e+05	1.38e+05
C(HSB)[T.적용]	29.1819	7.03e+04	0.000	1.000	-1.38e+05	1.38e+05
PT_LTH	0.8447	0.254	3.324	0.001	0.347	1.343
PT_WDTH	0.5451	0.182	2.996	0.003	0.188	0.902
ROLLING_TEMP_T5	-4.1072	0.574	-7.157	0.000	-5.232	-2.982
FUR_SZ_TEMP	-0.5436	nan	nan	nan	nan	nan
FUR_EXTEMP	-0.5436	nan	nan	nan	nan	nan



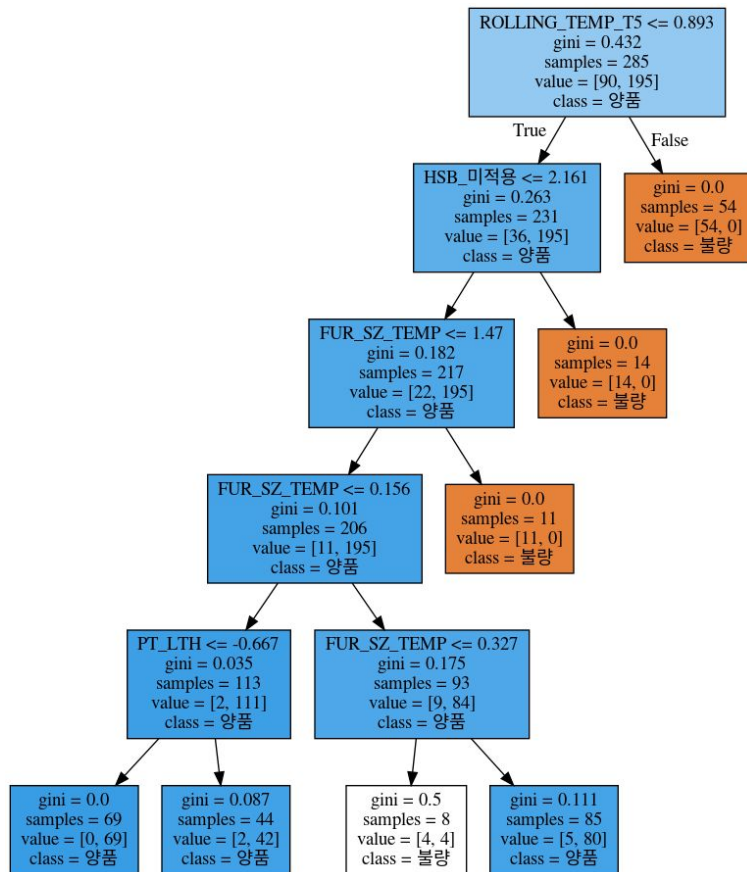
Decision Tree 모델

max_depth=5

min_samples_leaf=8

Training Set에 대한 정확도: 0.961

Test Set에 대한 정확도: 0.944



Random Forest 모델

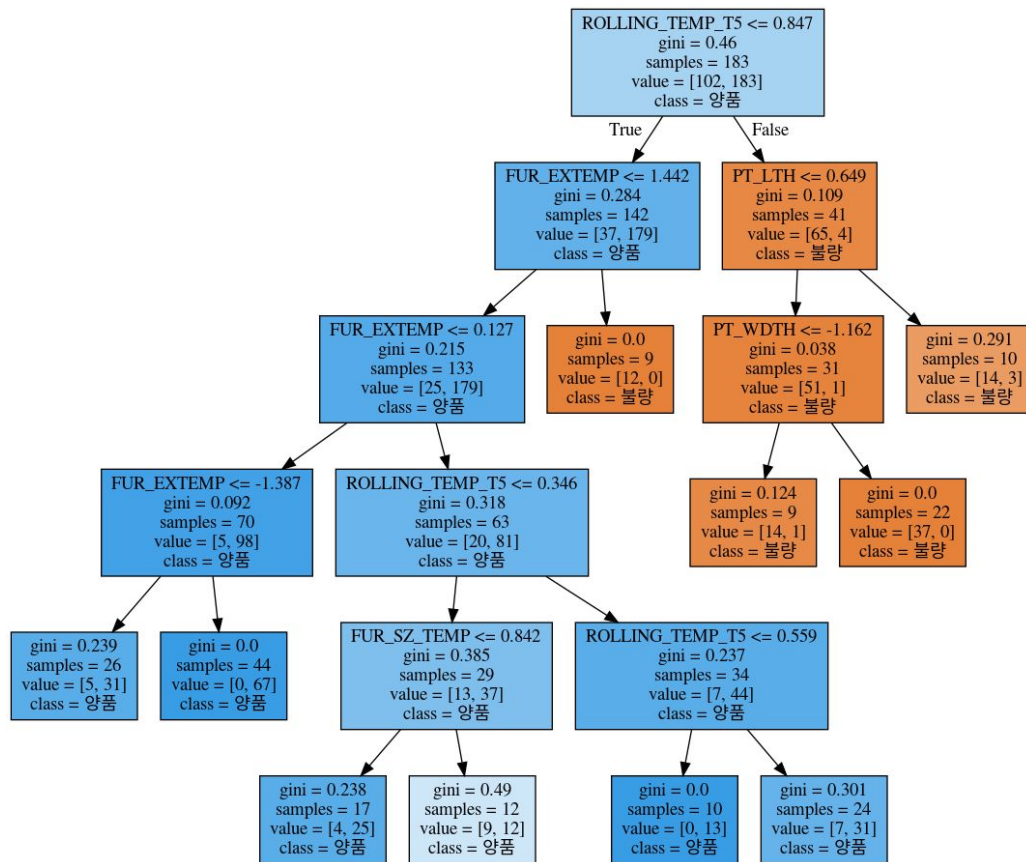
max_depth=5

min_samples_leaf=8

n_estimators=30

Training Set에 대한 정확도: 0.937

Test Set에 대한 정확도: 0.930



Gradient Boosting 모델

max_depth=5

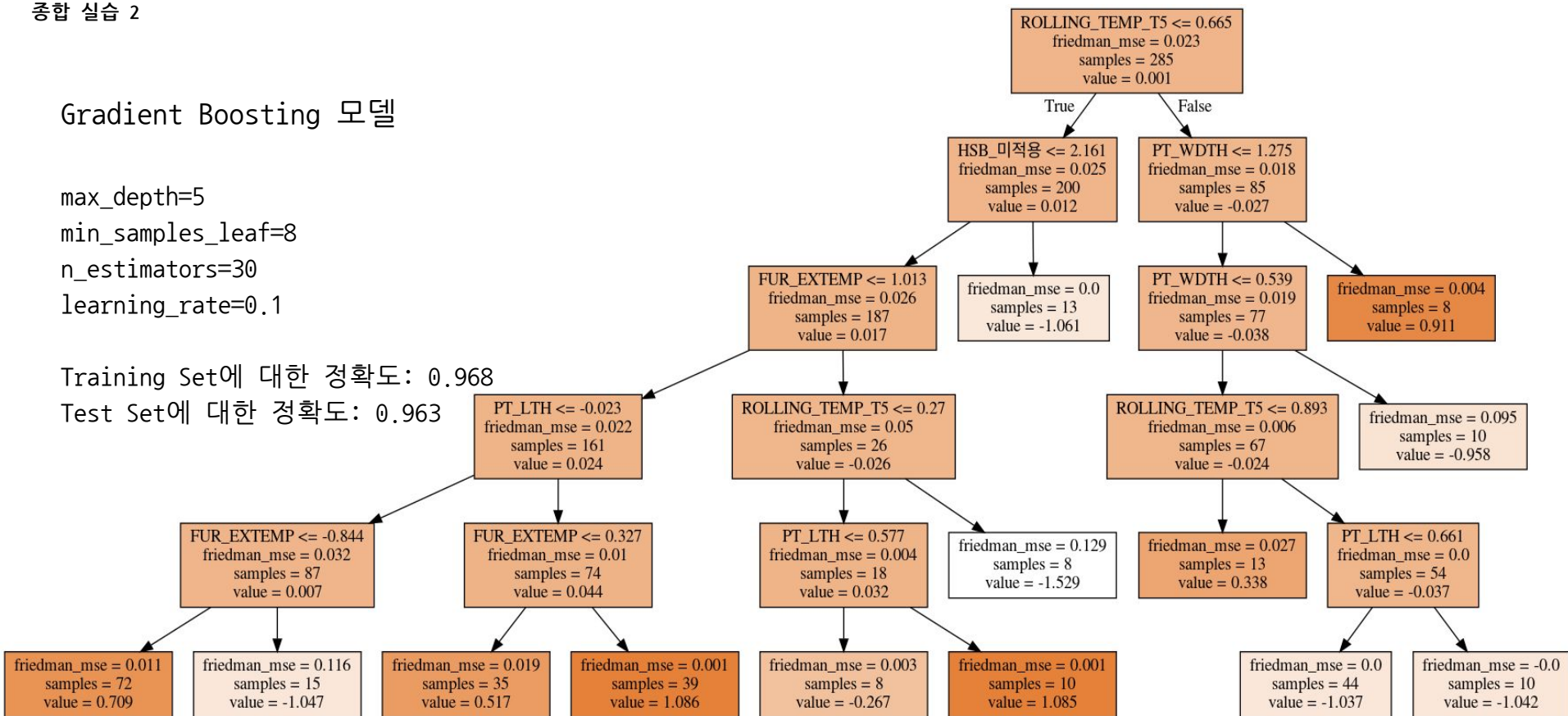
min_samples_leaf=8

n_estimators=30

learning_rate=0.1

Training Set에 대한 정확도: 0.968

Test Set에 대한 정확도: 0.963



SVC 모델

해당 데이터를 사용하여 모델을 만들었을 때, 스케일을 했음에도 불구하고 c 와 γ 값에 모델의 정확도가 변하지 않음.

Confusion Matrix

53	14
6	142

Training Set에 대한 정확도: 0.923

Test Set에 대한 정확도: 0.907

KNN 모델

해당 데이터를 사용하여 모델을 만들었을 때, 스케일을 했음에도 불구하고 $weight$ 와 $n_neighbor$ 값에 모델의 정확도가 변하지 않음.

Confusion Matrix

52	15
11	137

Training Set에 대한 정확도: 0.944

Test Set에 대한 정확도: 0.879

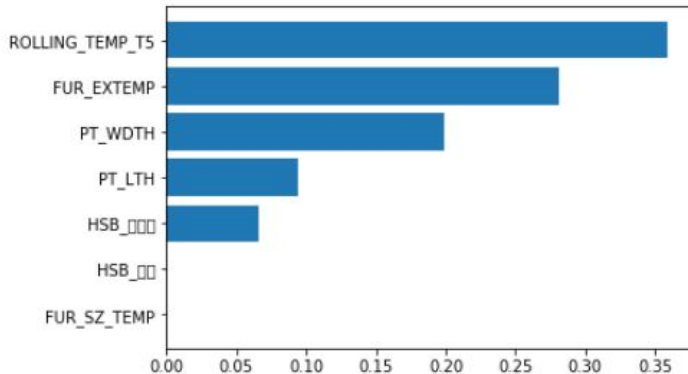
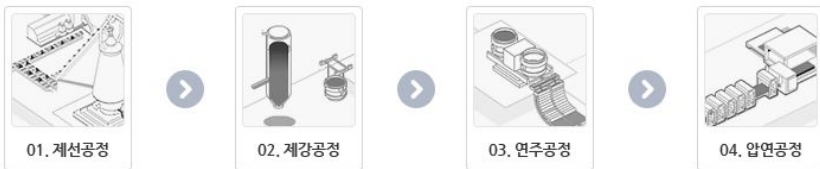
모델별로 약간의 차이가 있지만, 대체적으로 보았을 때 가장 중요한 파라미터는 ROLLING_TEMP_T5로 보임

- Decision Tree와 Ensemble 모델들에서 가장 상위의 파라미터가 ROLLING_TEMP_T5

Gradient Boosting 모델이 0.963으로 가장 높은 분류 정확성을 보여줌.

- 이 때의 파라미터 중요도 순위는 오른쪽 그래프와 같음

절반 이상의 중요도를 차지하는 ROLLING_TEMP_T5와
FUR_EXTEMP는 각각 가열대 온도와 압연 온도로



포스코 철강 생산 공정에서 연주 공정과 압연 공정, 마지막
두 과정에 의해 SCALE이 발생하는 경우가 가장 많은 것으로 판단됨.

- SCALE을 줄이기 위해서는 ROLLING_TEMP_T5와 FUR_EXTEMP를 일정 수준 이하로 유지해야함