SEOULTECH

Machine Predictive Maintenance Classification

Dataset to predict machine failure (binary) and type (multiclass)

2021.12.14

* 데이터사이언스학과 석사과정 이정언
* 데이터사이언스학과 석사과정 김재호

Final Project : Select subject of Project (※Subject : Machine Predictive Maintenance Classification)

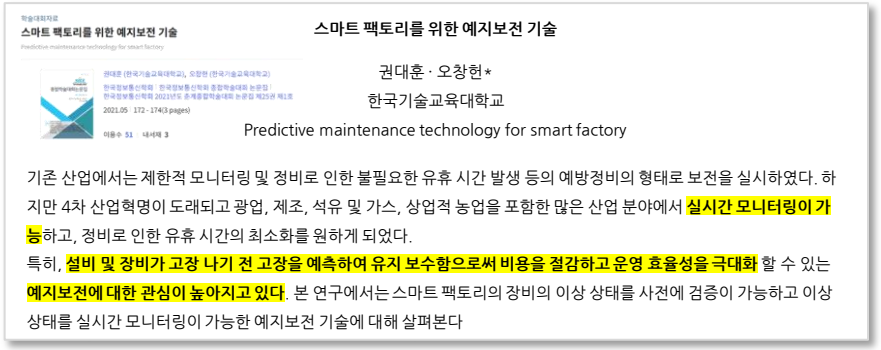
✓ 빅데이터 기반 의사결정지원 표준 아키텍쳐, 스마트팩토리, 제조 AI 적용사례들을 참고하여 'Machine Maintenance'와 관련된 주제로 캐글을 통하여 Datasets을 선택하였음

▶ 빅데이터 기반 의사결정지원 표준 아키텍쳐 : 의사결정

출처 : 한국산업기술평가관리원 KEIT PD ISSUE REPORT VOL 17-10

▶ 스마트 팩토리를 위한 예지보전 기술_권대훈,오창헌 (2021)

출처 : 한국정보통신학회 2021년도 춘계종합학술대회 논문집 제25권 제1호



스마트 팩토리를 위한 예지보전 기술

권대훈 · 오창헌*

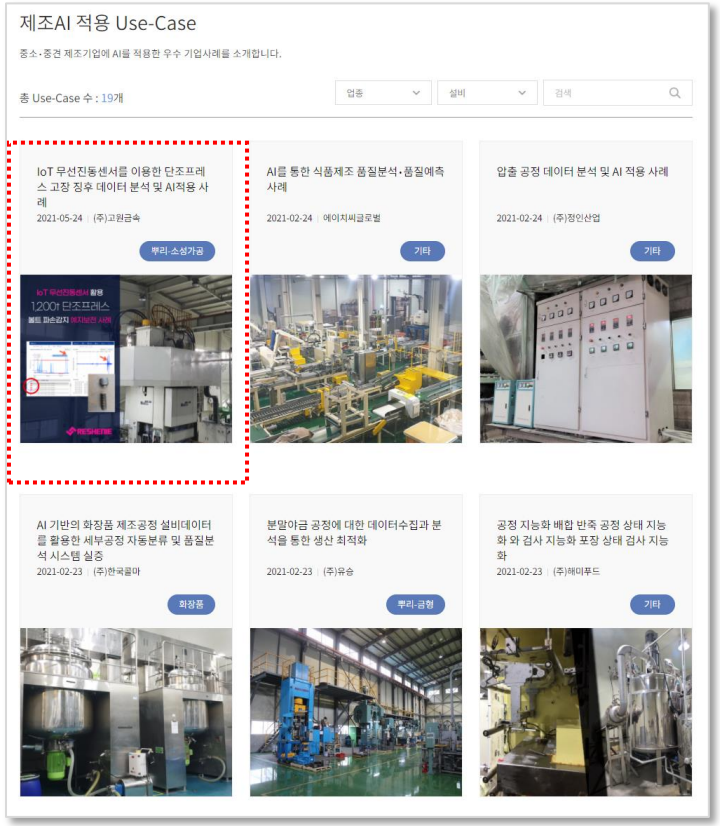
한국기술교육대학교

Predictive maintenance technology for smart factory

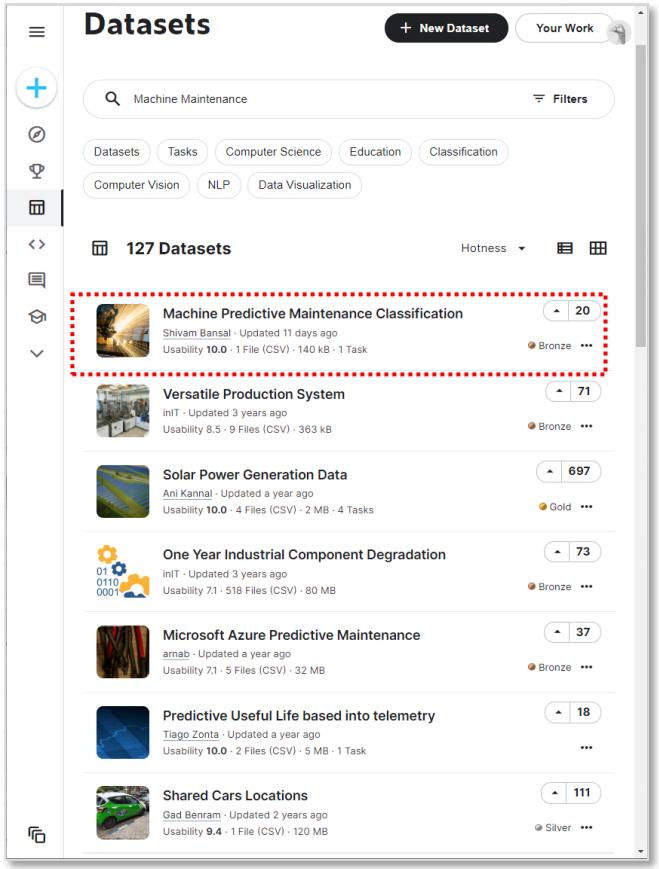
기존 산업에서는 제한적 모니터링 및 정비로 인한 불필요한 유휴 시간 발생 등의 예방정비의 형태로 보전을 실시하였다. 하지만 4차 산업혁명이 도래되고 광업, 제조, 석유 및 가스, 상업적 농업을 포함한 많은 산업 분야에서 **실시간 모니터링이 가능**하고, 정비로 인한 유휴 시간의 최소화를 원하게 되었다.

특히, **설비 및 장비가 고장 나기 전 고장을 예측하여 유지 보수함으로써 비용을 절감하고 운영 효율성을 극대화** 할 수 있는 **예지보전에 대한 관심이 높아지고 있다**. 본 연구에서는 스마트 팩토리의 장비의 이상 상태를 사전에 검증이 가능하고 이상 상태를 실시간 모니터링이 가능한 예지보전 기술에 대해 살펴본다

▶ 인공지능 중소벤쳐 제조 플렛폼 KAMP ⇒ Use-Case



▶ 'Machine Maintenance' Dataset search in Kaggle



Final Project : Overview

➢ 프로젝트명 : Machine Predictive Maintenance Classification

➢ 인 원 : 이정언 (2021-1학기 입학), 김재호 (2021-2학기 입학)

➢ 기간 : 2021.11.02 ~ 2021.12.14

➢ 목표 : Proposal presentation (11/16) & Final presentation (12/14)

➢ 산출물 : 발표자료, 발표영상

➢ 데이터셋 : Machine Predictive Maintenance Classification in Kaggle

https://www.kaggle.com/shivamb/machine-predictive-maintenance-classification

➢ 분석환경 구축

- Google Cloud
- Excel
- Python 3.7 ~
  - ✓ Pandas, numpy
  - ✓ scikit-learn
  - ✓ Matplotlib, seaborn

➢ Project R&R

이름	소속	Role & Responsibilities
김재호 (21512070)	EDM lab (지도교수 : 홍정식)	- 프로젝트 기획 (70%), 프로젝트 실행 (30%) - 파워포인트 발표자료 작성 및 발표 ★★ - 데이터 마이닝 (공통) - 데이터 시각화 (공통)
이정언 (21510097)	EDM lab (지도교수 : 홍정식)	- 프로젝트 기획 (30%) , 프로젝트 실행 (70%) - 데이터 마이닝 (공통) - 데이터 시각화 (공통) - 의사결정 트리를 활용한 데이터 분석 및 예측 모델링 ★★★

➢ 커뮤니케이션 :

- 오프라인 : 주 1회 이상, 프론티어관 EDM-Lab 307-1호

- 온라인 : KakaoTalk (메신저), Zoom (회의용)

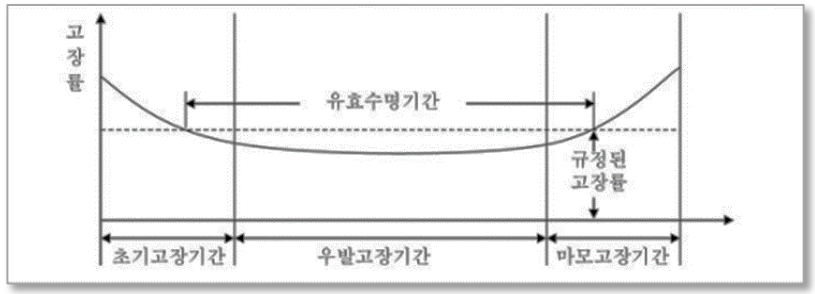
Final Project : Background of Project

✓ 설비의 고장유형은 초기고장, 우발고장, 마모고장 단계로 나누어짐

✓ 설비의 유지정비활동은 예방정비(PM), 예측정비 (PDM), 사후정비(CM) 등이 있으며, 과도한 예방정비는 유지 비용의 증가 및 생산성 저하를 불러오기 때문에 적절한 정비시점을 예측하는 예측정비가 필요함

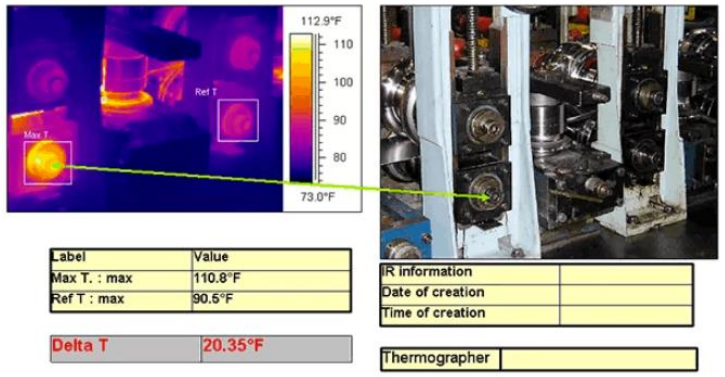
▶ 고장률 곡선

- 설비 수명특성곡선 또는 고장률 곡선이라고 함
- 욕조와 비슷하여 욕조곡선 (Bathtub Curve) 이라고도 함



▶ 설비의 고장기간별 원인과 대책

구분	고장원인	대책
초기고장	• 설계, 제작, 수리착오에 따른 고장 • 사용방법 미숙에 따른 고장	• 결함발견 수리, 부식검사 • 메이커 품질보증 의존
우발고장	• 설계한계 초과에 따른 고장 • 진동 및 충격에 의한 고장	• 설비한계의 변경 • 정상운전 실시 • 사후보전(BM) 실시
마모고장	• 마모, 피로열화, 절연열화 등 특성열화에 따른 고장	• 예방보전(PM) 실시

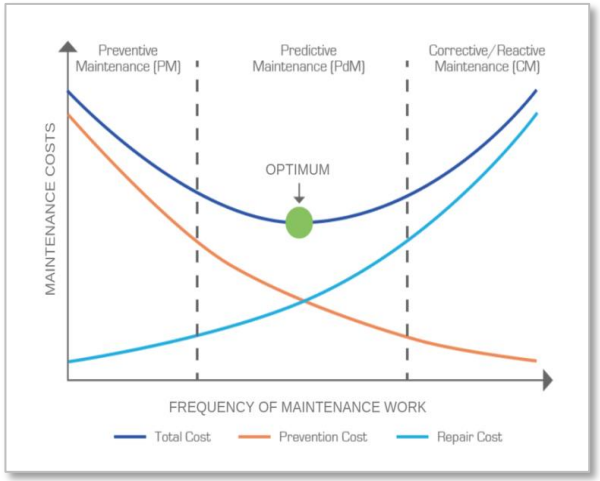


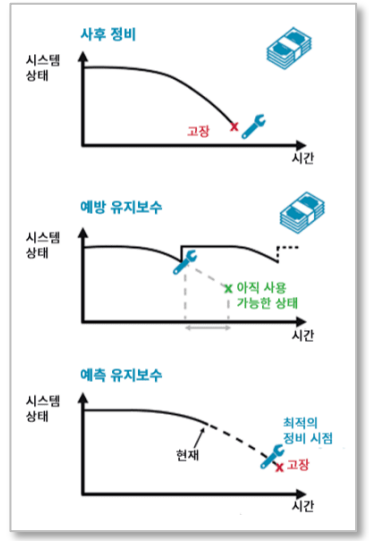
▲ PDM 정비에 대한 예시

▶ 유지정비 활동 분류

유지정비 분류	내용	예시
사후정비 (CM) (corrective maintenance)	• 대상 설비가 파괴적 고장이 난 이후 실행하는 유지정비 활동 • 주로 중요하지 않거나, 예비설비가 있는 설비에 적용	• 소형 모터, 1차 배터리, 조명시설 등
예방 또는 계획정비 (PM) (preventive or schedule-based maintenance)	• 대상 설비가 기능적 고장이 나기 전에 신뢰성 정보에 기반하여 주기적 정비 스케줄에 입각하여 진행하는 유지정비 활동 • 상대적으로 중요한 설비들이 대상이며, 신뢰성 정보를 기반으로 점검 • 수리 주기가 제공되는 설비 또는 안전 문제로 과잉정비가 필요한 설비들이 대상	• 자동차 브레이크 패드, 자동차·선박 엔진, 항공기·발전소 터빈 등
예측정비 (PDM) (predictive maintenance)	• 설비의 상태를 수시 또는 상시로 점검하여 유지정비가 필요한 시점 대비 유지정비에 필요한 소요시간(Lead time) 만큼 빠르게 유지정비 의사를 결정하는 체계 • 상태 기반 정비 대상 설비 중 유지정비에 필요한 소요시간이 상대적으로 길거나 또는 다운타임에 의한 비용 손실이 매우 큰 설비가 대상	• 해상풍력발전기, 대형 전력설비(변압기, 차단기)

▶ Maintenance Method & Cost Analysis





Final Project : Column Description of Dataset

- ✓ Predictive maintenance dataset은 컬럼이 10개, Row는 10,000열로 csv형태로 구성된 데이터이며, 총 10개의 변수로 string 변수 3개, integer 변수 4개, Decimal 변수 3개로 구성되어 있음
- ✓ (UDI, Product ID, Type, Air temperature, Process temperature, Rotational speed, Torque, Tool wear) 설명변수 8개, (Target, Failure Type) 목적변수 2개

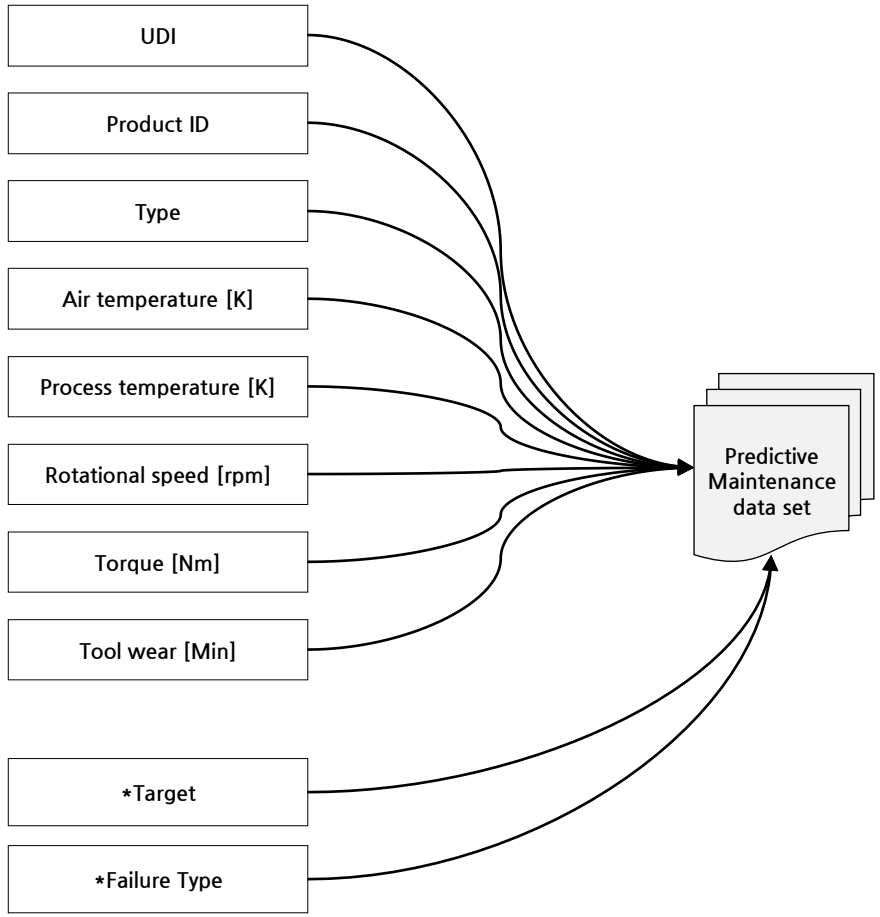
Data Table (10 column x 10,000 row, predictive_maintenance.csv, 531.01 kB)

변수	UDI	Product ID	Type	Air temperature	Process temperature	Rotational speed	Torque	Tool wear	Target	Failure Type
속성	Integer	String	String	Decimal	Decimal	Integer	Decimal	Integer	Integer	String
단위	-	-		[K]	[K]	[rpm]	[Nm]	[Min]	-	-
설명	unique identifier ranging from 1 to 10000	consisting of a letter L, M, or H for low (50% of all products), medium (30%), and high (20%) as product quality variants and a variant-specific serial number	L, M, H	generated using a random walk process later normalized to a standard deviation of 2 K around 300 K	generated using a random walk process normalized to a standard deviation of 1 K, added to the air temperature plus 10 K	calculated from a power of 2860 W, overlaid with a normally distributed noise	torque values are normally distributed around 40 Nm with an ïƒ = 10 Nm and no negative values	The quality variants H/M/L add 5/3/2 minutes of tool wear to the used tool in the process.	Failure or Not	Type of Failure
Example (6 samples)	1	M14860	M	298.1	308.6	1551	42.8	0	0	No Failure
	78	L47257	L	298.8	308.9	1455	41.3	208	1	Tool Wear Failure
	3830	H33243	H	302.3	310.9	1366	48.4	130	1	Heat Dissipation Failure
	4354	M19213	M	302	309.7	1386	62.7	142	1	Power Failure
	5489	L52668	L	302.7	312.2	1450	52.1	213	1	Overstrain Failure
	7869	H37282	H	300.4	311.9	1438	46.7	41	0	Random Failures

label	Tool Wear Failure (TWF)	Heat dissipation failure (HDF)	Power Failure (PWF)	Overstrain Failure (OSF)	Random Failures (RNF)
Failure Type	the tool will be replaced or fail at a randomly selected tool wear time between 200 - 240 mins (120 times in our dataset). At this point in time, the tool is replaced 74 times, and fails 46 times (randomly assigned).	heat dissipation causes a process failure, if the difference between air- and process temperature is below 8.6 K and the tool's rotational speed is below 1380 rpm. This is the case for 115 data points.	the product of torque and rotational speed (in rad/s) equals the power required for the process. If this power is below 3500 W or above 9000 W, the process fails, which is the case 95 times in our dataset.	if the product of tool wear and torque exceeds 11,000 minNm for the L product variant (12,000 for M, 13,000 for H), the process fails due to overstrain. This is true for 98 datapoints.	each process has a chance of 0,1 % to fail regardless of its process parameters. This is the case for 19 datapoints, more frequent than could be expected for 10,000 datapoints in our dataset.

Final Project : Project Goals

✓ Machine의 Failure Type에 따른 고장발생 징후를 탐색하고, 이를 예측할 수 있는 모델을 제시하여 PDM (Predictive maintenance)기반의 설비보전을 할 수 있도록 의사결정을 지원하고자 함



▶ 프로젝트 세부목표

- 시각화를 통해 데이터에 대한 기본 분석과 NA값이나 이상치 등의 전처리
- Decision Tree 기법을 이용해 고장 유형을 분류해서 여러 규칙들을 추출
- 공정 기계의 어떤 원리나 도메인 지식을 찾아보고 고장으로 분류한 case들을 해석하고, 예측설비를 위한 적절한 기준을 제시
- 마지막으로, 분석한 결과들로 도출한 insight를 통해 데이터에 대한 의미 있는 시각화를 해보려고 함

▶ 프로젝트 예상 이슈

- 고장이 난 경우는 전체의 3% 정도로 적은 편이라 이를 염두하고 진행
- Random Failure의 Target 변수값이 No Failure와 같은 0이므로 데이터 분석 전 전처리 방법을 결정하는 것이 필요

Target	0	1	1	1	1	0
Failure Type	No Failure	Tool Wear Failure	Heat Dissipation Failure	Power Failure	Overstrain Failure	Random Failures

Final Project : EDA - 기초 데이터셋 분석

✓ 기초 데이터 셋에 대한 information, describe, 목적변수 value count, 평균값 등의 기초 통계량 분석을 진행

① Dataset - 불러오기

```
df_org = pd.read_csv('predictive_maintenance.csv', index_col='UDI')
df_org[:10]
```

UDI	Product ID	Type	Air temperature [K]	Process temperature [K]	Rotational speed [rpm]	Torque [Nm]	Tool wear [min]	Target	Failure Type
1	M14860	M	298.1	308.6	1551	42.8	0	0	No Failure
2	L47181	L	298.2	308.7	1408	46.3	3	0	No Failure
3	L47182	L	298.1	308.5	1498	49.4	5	0	No Failure
4	L47183	L	298.2	308.6	1433	39.5	7	0	No Failure
5	L47184	L	298.2	308.7	1408	40.0	9	0	No Failure

② Dataset - Information 확인

```
df_org.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 10000 entries, 1 to 10000
Data columns (total 9 columns):
 #   Column                  Non-Null Count  Dtype
---  ------                  --------------  -----
 0   Product ID              10000 non-null  object
 1   Type                    10000 non-null  object
 2   Air temperature [K]     10000 non-null  float64
 3   Process temperature [K] 10000 non-null  float64
 4   Rotational speed [rpm]  10000 non-null  int64
 5   Torque [Nm]             10000 non-null  float64
 6   Tool wear [min]         10000 non-null  int64
 7   Target                  10000 non-null  int64
 8   Failure Type            10000 non-null  object
dtypes: float64(3), int64(3), object(3)
memory usage: 781.2+ KB
```

③ Dataset - describe 확인

```
df_org.describe()
```

	Air temperature [K]	Process temperature [K]	Rotational speed [rpm]	Torque [Nm]	Tool wear [min]	Target
count	10000.000000	10000.000000	10000.000000	10000.000000	10000.000000	10000.000000
mean	300.004930	310.005560	1538.776100	39.986910	107.951000	0.033900
std	2.000259	1.483734	179.284096	9.968934	63.654147	0.180981
min	295.300000	305.700000	1168.000000	3.800000	0.000000	0.000000
25%	298.300000	308.800000	1423.000000	33.200000	53.000000	0.000000
50%	300.100000	310.100000	1503.000000	40.100000	108.000000	0.000000
75%	301.500000	311.100000	1612.000000	46.800000	162.000000	0.000000
max	304.500000	313.800000	2886.000000	76.600000	253.000000	1.000000

④ 목적변수 - Value Count 확인

```
df_org['Failure Type'].value_counts()
```

```
No Failure                 9652
Heat Dissipation Failure    112
Power Failure                95
Overstrain Failure           78
Tool Wear Failure            45
Random Failures              18
Name: Failure Type, dtype: int64
```

⇒ 고장은 1만개 중 348개, 약 3.5%만 해당

⑤ 목적변수별 평균값 통계량 확인

```
df_group = df_org.groupby(['Failure Type'], as_index=False).mean()
failure_type_mean = df_group.drop(['Target'], axis = 1)
failure_type_mean
```

	Failure Type	Air temperature [K]	Process temperature [K]	Rotational speed [rpm]	Torque [Nm]	Tool wear [min]
0	Heat Dissipation Failure	302.567857	310.799107	1337.964286	52.778571	107.339286
1	No Failure	299.972855	309.994343	1540.324389	39.624316	106.678927
2	Overstrain Failure	299.867949	310.051282	1354.243590	56.878205	208.217949
3	Power Failure	300.075789	309.954737	1763.968421	48.514737	101.884211
4	Random Failures	300.766667	310.755556	1489.444444	43.522222	119.888889
5	Tool Wear Failure	300.288889	310.164444	1570.666667	37.226667	216.555556

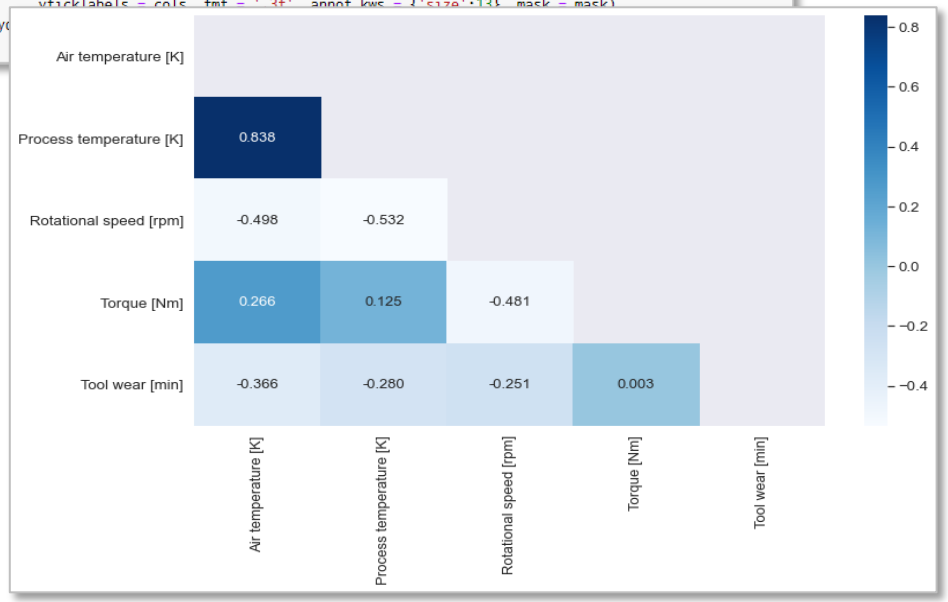
Final Project : 수치형 변수 분석 및 시각화

- ✓ 수치형 변수에 대한 관계 분석을 진행 및 시각화를 진행함
- ✓ Air Temp & Process Temp 변수는 비례하는 경향, rpm-torque 변수는 반비례 경향으로 추정

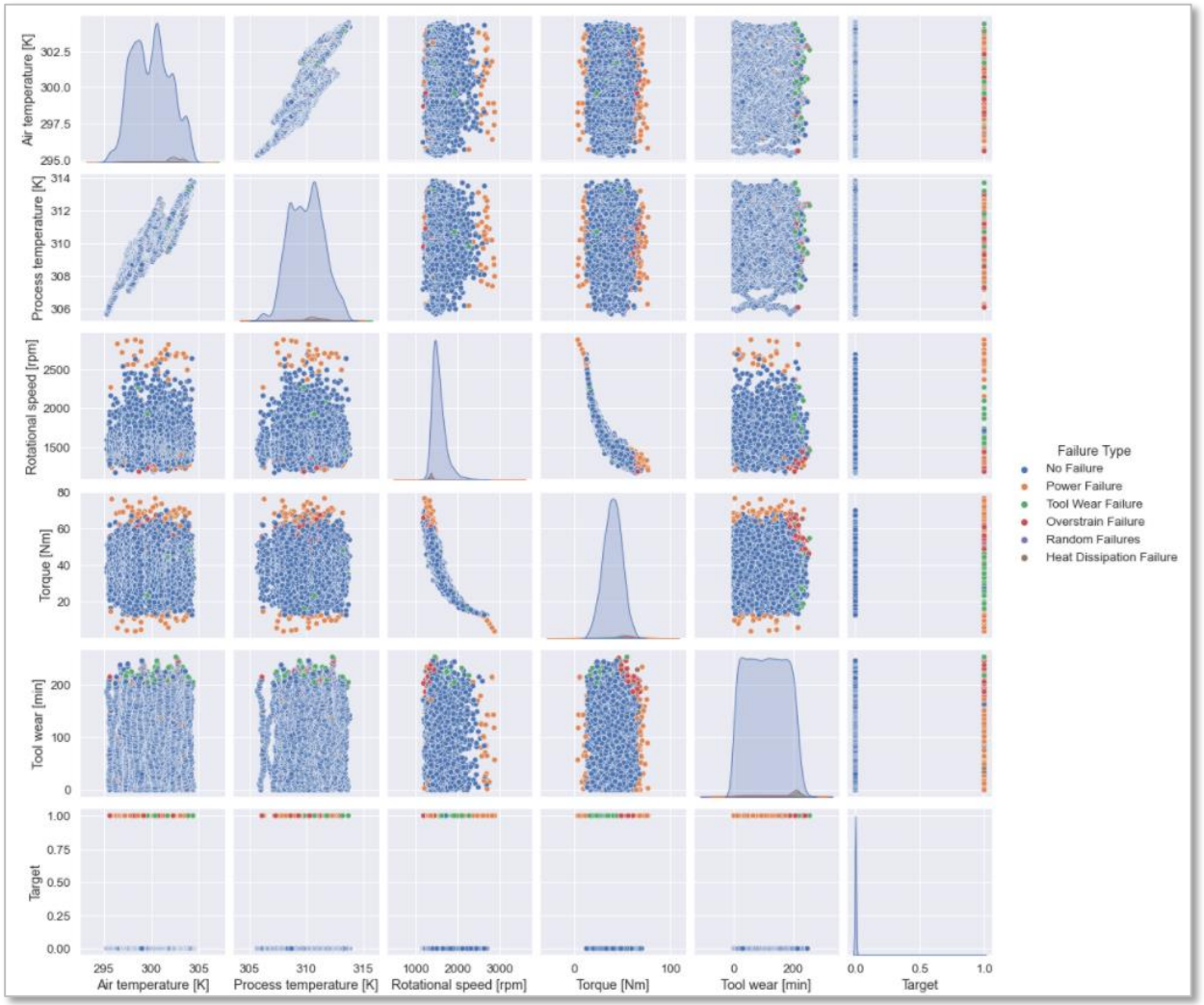
```
cols = ['Air temperature [K]', 'Process temperature [K]',
       'Rotational speed [rpm]', 'Torque [Nm]', 'Tool wear [min]']

corr = failure_type_mean[cols].corr(method = 'pearson')
mask = np.zeros_like(corr)
mask[np.triu_indices_from(mask)] = True

display(corr)
```

	Air temperature [K]	Process temperature [K]	Rotational speed [rpm]	Torque [Nm]	Tool wear [min]
Air temperature [K]	1.000000	0.838492	-0.498128	0.266246	-0.365517
Process temperature [K]	0.838492	1.000000	-0.532174	0.125058	-0.280383
Rotational speed [rpm]	-0.498128	-0.532174	1.000000	-0.481274	-0.251461
Torque [Nm]	0.266246	0.125058	-0.481274	1.000000	0.003276
Tool wear [min]	-0.365517	-0.280383	-0.251461	0.003276	1.000000

```
sns.set(rc = {'figure.figsize':(12,8)}, font_scale = 1.2)
hm = sns.heatmap(corr.values, cmap = "Blues", cbar = True, annot = True, xticklabels = cols,
                 yticklabels = cols, fmt = '.3f', annot_kws = {'size':13}, mask = mask)
plt.tight_layo
plt.show()
```




```
sns.set(font_scale=1.2)
sns.pairplot(data=df_org, hue='Failure Type', height=2.5)
```

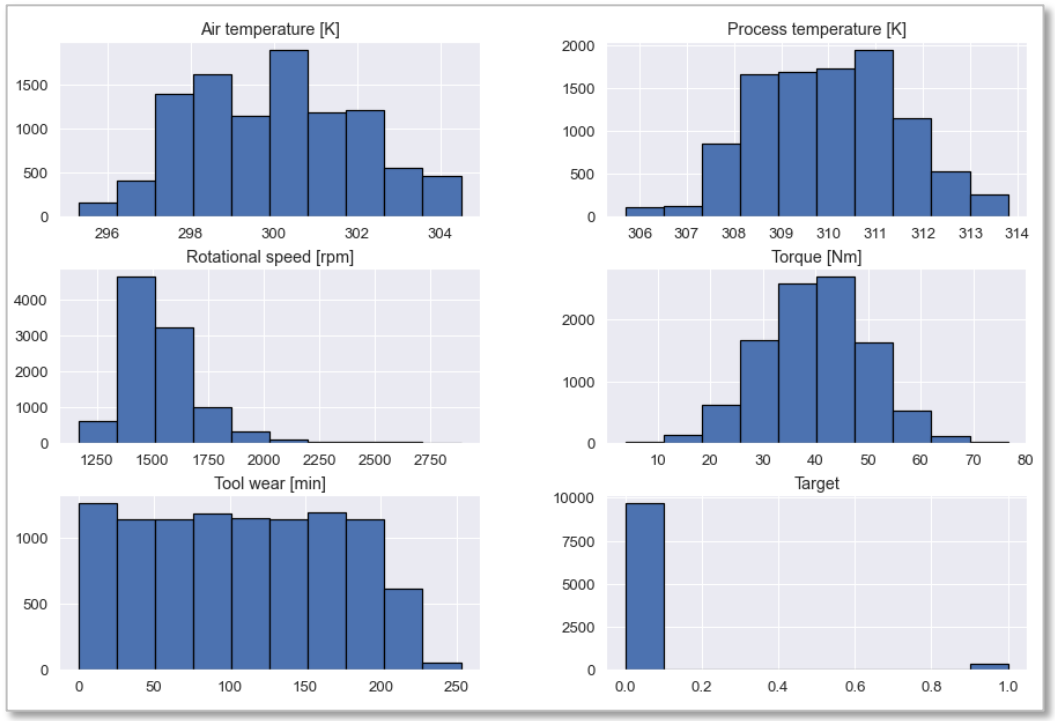


Final Project : 기초데이터셋 변수별 시각화

① 목적변수별 평균값 통계량 확인 및 시각화

```
df_group = df_org.groupby(['Failure Type'], as_index=False).mean()
failure_type_mean = df_group.drop(['Target'], axis = 1)
failure_type_mean
```

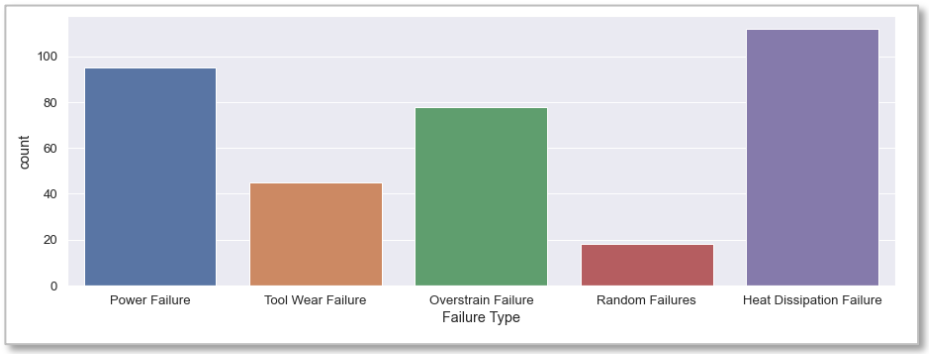
	Failure Type	Air temperature [K]	Process temperature [K]	Rotational speed [rpm]	Torque [Nm]	Tool wear [min]
0	Heat Dissipation Failure	302.567857	310.799107	1337.964286	52.778571	107.339286
1	No Failure	299.972855	309.994343	1540.324389	39.624316	106.678927
2	Overstrain Failure	299.867949	310.051282	1354.243590	56.878205	208.217949
3	Power Failure	300.075789	309.954737	1763.968421	48.514737	101.884211
4	Random Failures	300.766667	310.755556	1489.444444	43.522222	119.888889
5	Tool Wear Failure	300.288889	310.164444	1570.666667	37.226667	216.555556



② 목적변수 - Value Count 확인 및 시각화

```
df_org['Failure Type'].value_counts()
```

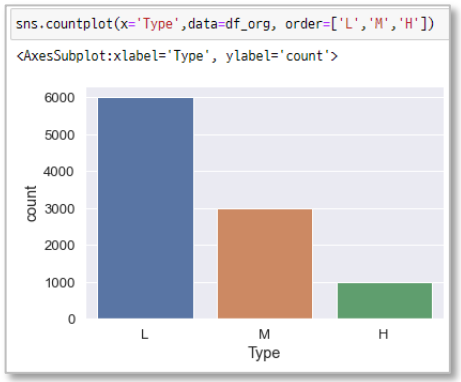
```
No Failure                9652
Heat Dissipation Failure   112
Power Failure               95
Overstrain Failure          78
Tool Wear Failure           45
Random Failures             18
Name: Failure Type, dtype: int64
```



③ 설명변수 - Type 확인 및 시각화

```
sns.countplot(x='Type',data=df_org, order=['L','M','H'])
```

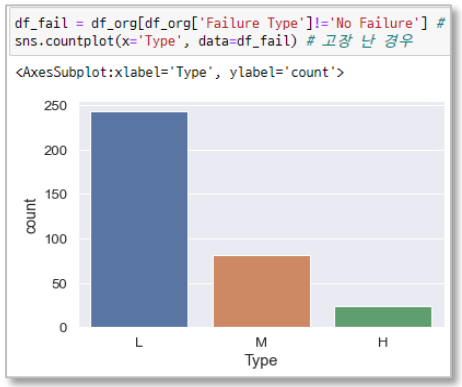
```
<AxesSubplot:xlabel='Type', ylabel='count'>
```



④ 설명변수 - Type 확인 및 시각화 (고장 경우만)

```
df_fail = df_org[df_org['Failure Type']!='No Failure'] #
sns.countplot(x='Type', data=df_fail) # 고장 난 경우
```

```
<AxesSubplot:xlabel='Type', ylabel='count'>
```

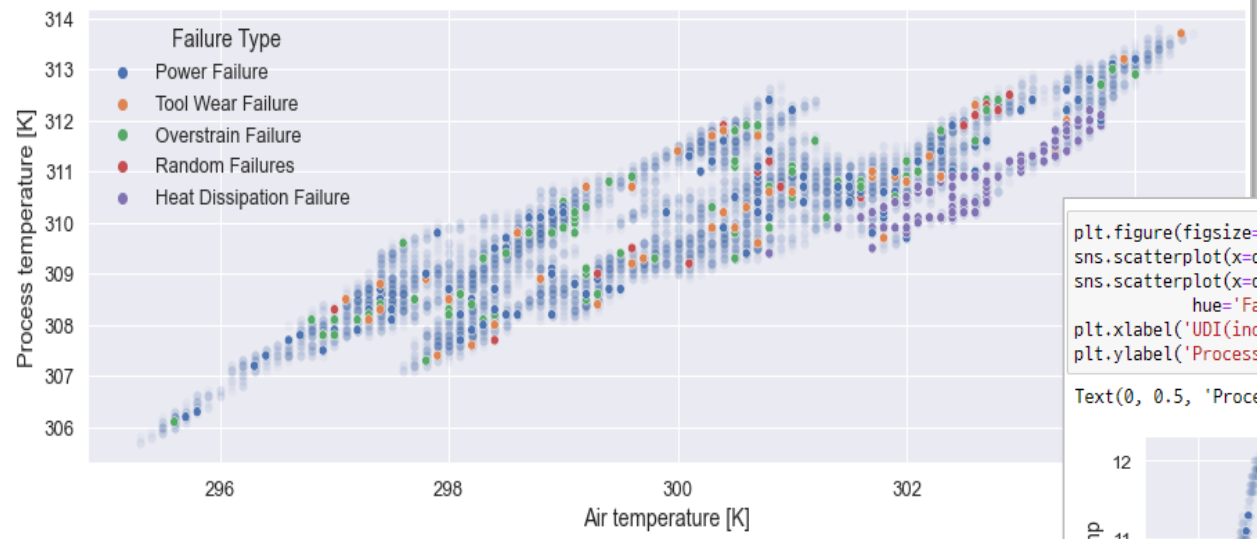


Final Project : 변수별 관계 시각화

✓ 변수별 관계 분석을 위한 시각화 진행

```
plt.figure(figsize=(15,5))
sns.scatterplot(x='Air temperature [K]', y='Process temperature [K]', alpha=0.05, data=df_org)
sns.scatterplot(x='Air temperature [K]', y='Process temperature [K]',
            hue='Failure Type', data=df_fail)
```

```
<AxesSubplot:xlabel='Air temperature [K]', ylabel='Process temperature [K]'>
```

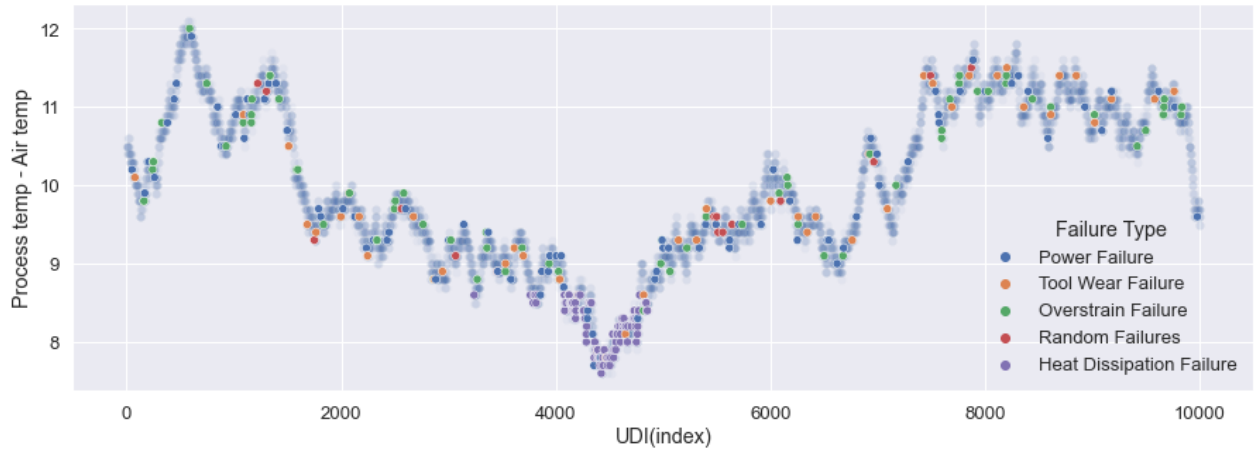


▲ Air Temperature & ProcessTemperature

▼ UDI & Process Temp - Air Temp

```
plt.figure(figsize=(15,5))
sns.scatterplot(x=df_org.index, y=df_org['Process temperature [K]']-df_org['Air temperature [K]'], alpha=0.05, data=df_org)
sns.scatterplot(x=df_fail.index, y=df_fail['Process temperature [K]']-df_fail['Air temperature [K]'],
            hue='Failure Type', data=df_fail)
plt.xlabel('UDI(index)')
plt.ylabel('Process temp - Air temp')
```

```
Text(0, 0.5, 'Process temp - Air temp')
```



Final Project： Preprocessing

- ✓ 캐글 데이터셋 Baseline과 차별화된 분석 진행을 위한 전처리 진행
- ✓ Baseline은 Target 변수(1, 0)만을 목표 변수로 사용하였으나, 본 프로젝트에서는 Failure Type을 목표 변수로 사용하여 분석을 진행함

‘product id’
변수 삭제

```
df = pd.concat([pd.get_dummies(df_org['Type']), df_org[['Air temperature [K]', 'Process temperature [K]',
       'Rotational speed [rpm]', 'Torque [Nm]', 'Tool wear [min]',
       'Failure Type']]],axis=1)
df.rename(columns = {'H' : 'Type_H', 'L' : 'Type_L','M':'Type_M'}, inplace = True)
df.head()
```

UDI	Type_H	Type_L	Type_M	Air temperature [K]	Process temperature [K]	Rotational speed [rpm]	Torque [Nm]	Tool wear [min]	Failure Type
1	0	0	1	298.1	308.6	1551	42.8	0	No Failure
2	0	1	0	298.2	308.7	1408	46.3	3	No Failure
3	0	1	0	298.1	308.5	1498	49.4	5	No Failure
4	0	1	0	298.2	308.6	1433	39.5	7	No Failure
5	0	1	0	298.2	308.7	1408	40.0	9	No Failure

‘Failure Type’
목표 변수로 설정

‘Type’
범주형 변수 더미화

‘Target’
변수 삭제

Final Project： Classification Using DT (1/3)

✓ 의사결정트리(Decision Tree)를 활용한 데이터 분석을 진행

3.1 split dataset ¶

```
from sklearn.model_selection import train_test_split
from sklearn import tree
import graphviz
from sklearn.metrics import confusion_matrix
from sklearn.metrics import accuracy_score, f1_score, recall_score
```

```
X = df.drop('Failure Type', axis=1)
y = df['Failure Type']
Xtrain, Xtest, ytrain, ytest = train_test_split(X, y, train_size=0.8, random_state=123)
```

```
ytrain.value_counts()
```

```
No Failure                 7729
Heat Dissipation Failure     96
Power Failure                75
Overstrain Failure           54
Tool Wear Failure            33
Random Failures              13
Name: Failure Type, dtype: int64
```

3.2 classification

- 디시전트리 사용

```
model = tree.DecisionTreeClassifier(criterion='entropy', min_samples_leaf=0.001, max_depth=5, random_state=1234)
model.fit(Xtrain, ytrain)
```

```
DecisionTreeClassifier(criterion='entropy', max_depth=5, min_samples_leaf=0.001,
                       random_state=1234)
```

3.3 Performance

- imbalnced data이기에 Accuracy보다는 F1 score
- 다중 클래스라서 F1 score micro를 사용
- 특히 recall : 실제 고장을 고장이 났다고 예측한 비율. 고장이 난 걸 제대로 분류하는게 중요

```
y_model = model.predict(Xtest)
```

```
accuracy_score(ytest, y_model)
```

```
0.969
```

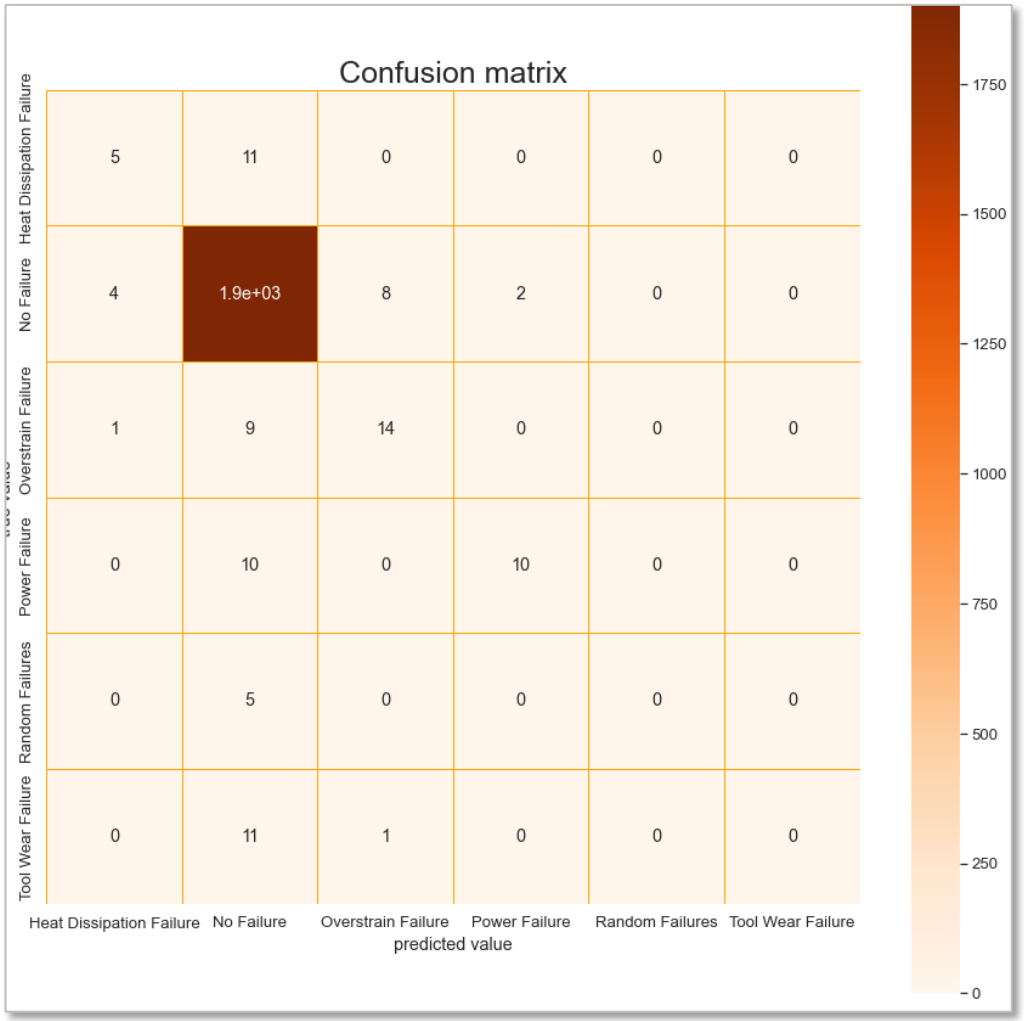
```
f1_score(ytest, y_model, average='macro')
```

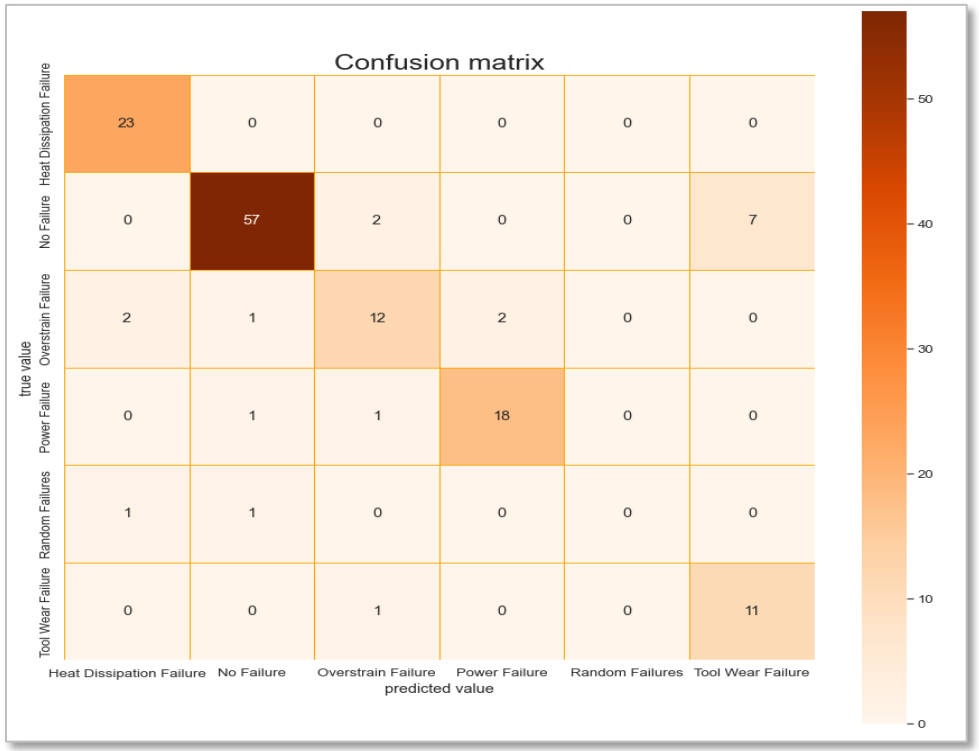
```
0.4316480287897064
```

```
plt.figure(figsize=(15,15))
cmat = confusion_matrix(ytest, model.predict(Xtest))
#cmat=cmat[[0,2,3,4,5],[[0],[2],[3],[4],[5]]]
sns.heatmap(cmat, square=True, annot=True, cmap='Oranges',cbar=True, linewidths=0.5,linecolor='orange')
plt.xticks(np.arange(0.5, len(model.classes_), 1), model.classes_)
plt.yticks(np.arange(0.5, len(model.classes_), 1), model.classes_)
plt.title('Confusion matrix', fontsize = 25)
plt.xlabel('predicted value')
plt.ylabel('true value');
```

Final Project： Classification Using DT (2/3)

- ✓ 정상작동 sample이 많은 관계로 결과가 Clsaasification이 잘 되지 않은 것으로 판단되어, Under sampling후 재분석을 진행함.
- ✓ 정상작동 sample 중 몇 개를 random하게 뽑아서 정상작동과 고장작동인 경우를 1:1이 되도록 맞추어 분석함
- ✓ Under sampling을 진행한 경우 각 고장별 특징이 조금더 두드러지게 분류되는 것을 확인할 수 있었음






```
df2=df.copy()
no_Failure_ind = df2[df2['Failure Type'] == 'No Failure'].index
failure = df2[df2['Failure Type'] != 'No Failure']
random_ind = np.random.choice(no_Failure_ind, 1*len(failure), replace=False)
failure_ind = df2[df2['Failure Type'] != 'No Failure'].index
undersample_ind = np.concatenate([failure_ind, random_ind])
undersample = df2.loc[undersample_ind]
```

Final Project： Classification Using DT (3/3)

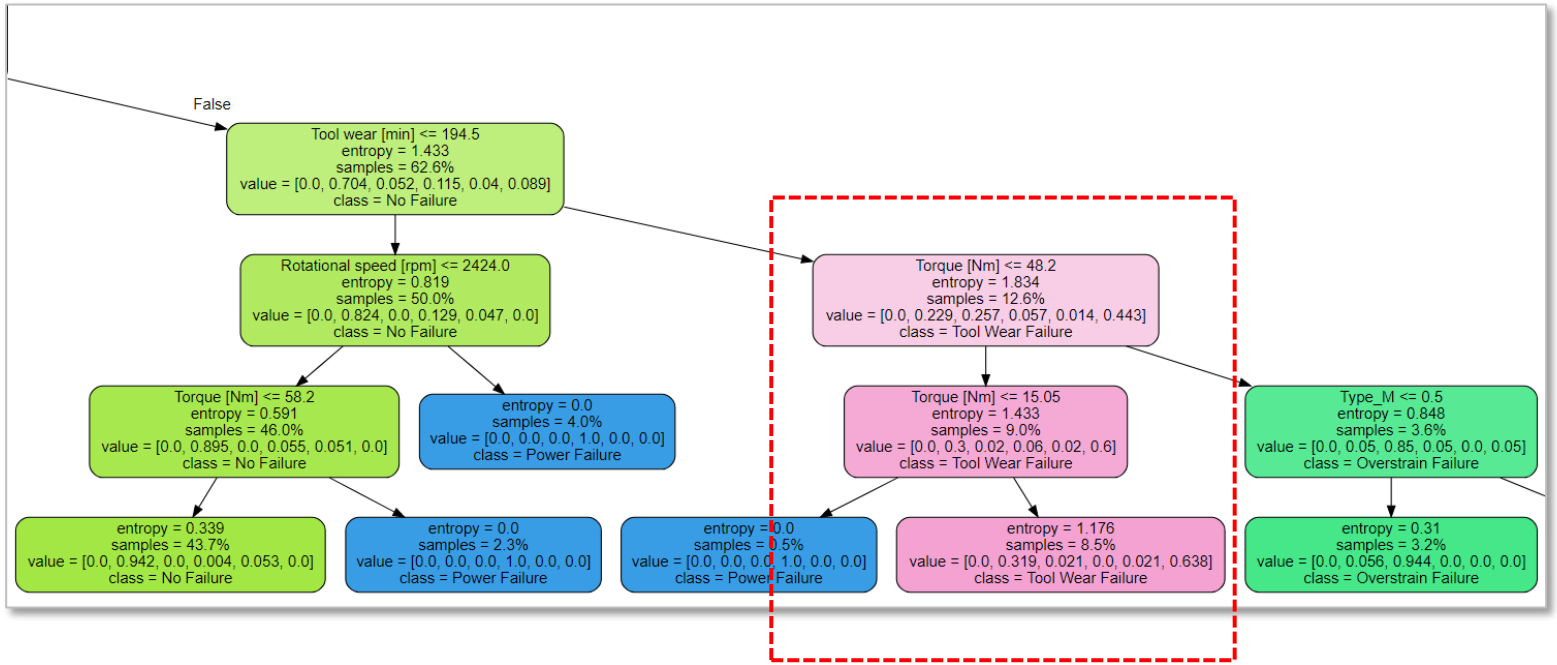
✓ 의사결정 트리를 확인하였을 때도 Under sampling을 통해 고장모드의 분류가 잘 되었음을 확인할 수 있음

```
dot_data = tree.export_graphviz(model, out_file=None, feature_names=X.columns,
                                class_names=model.classes_,
                                filled=True, rounded=True,
                                proportion=True)
graph = graphviz.Source(dot_data)
graph
```

```
<graphviz.files.Source at 0x2770f2406a0>
```

```
print(tree.export_text(model, feature_names=list(X.columns)))
```

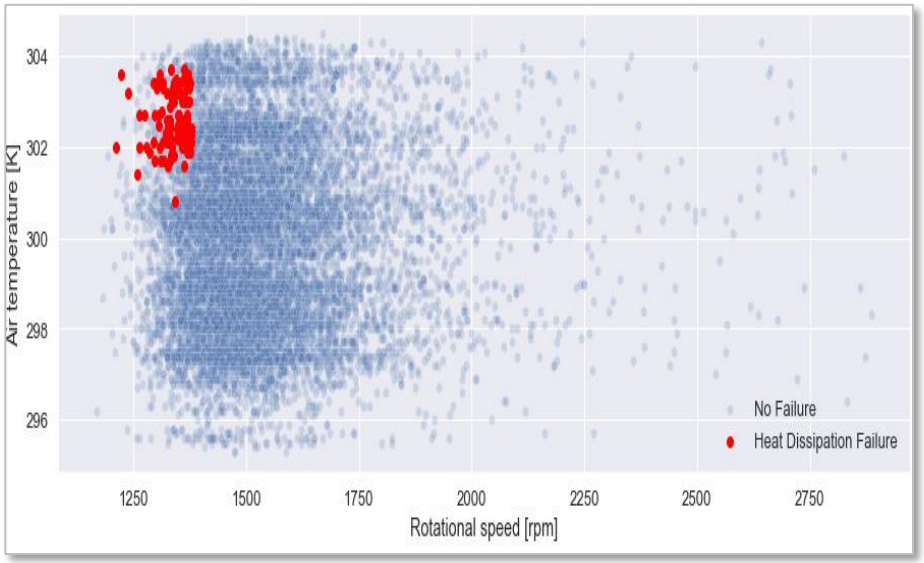
```
|--- Rotational speed [rpm] <= 1379.50
|   |--- Air temperature [K] <= 301.55
|   |   |--- Torque [Nm] <= 62.20
|   |   |   |--- Tool wear [min] <= 186.50
|   |   |   |   |--- class: No Failure
|   |   |   |--- Tool wear [min] >  186.50
|   |   |   |   |--- class: Overstrain Failure
|   |   |--- Torque [Nm] >  62.20
|   |   |   |--- Tool wear [min] <= 199.00
|   |   |   |   |--- class: Power Failure
|   |   |   |--- Tool wear [min] >  199.00
|   |   |   |   |--- class: Overstrain Failure
|   |--- Air temperature [K] >  301.55
|   |   |--- Torque [Nm] <= 65.65
|   |   |   |--- Tool wear [min] <= 181.00
|   |   |   |   |--- class: Heat Dissipation Failure
|   |   |   |--- Tool wear [min] >  181.00
|   |   |   |   |--- class: Heat Dissipation Failure
|   |   |--- Torque [Nm] >  65.65
|   |   |   |--- Rotational speed [rpm] <= 1267.00
|   |   |   |   |--- class: Heat Dissipation Failure
|   |   |   |--- Rotational speed [rpm] >  1267.00
|   |   |   |   |--- class: Power Failure
|--- Rotational speed [rpm] >  1379.50
|   |--- Tool wear [min] <= 194.50
|   |   |--- Rotational speed [rpm] <= 2403.50
|   |   |   |--- Torque [Nm] <= 58.00
|   |   |   |   |--- class: No Failure
|   |   |   |--- Torque [Nm] >  58.00
|   |   |   |   |--- class: Power Failure
|   |   |--- Rotational speed [rpm] >  2403.50
|   |   |   |--- class: Power Failure
|   |--- Tool wear [min] >  194.50
|   |   |--- Torque [Nm] <= 48.20
```

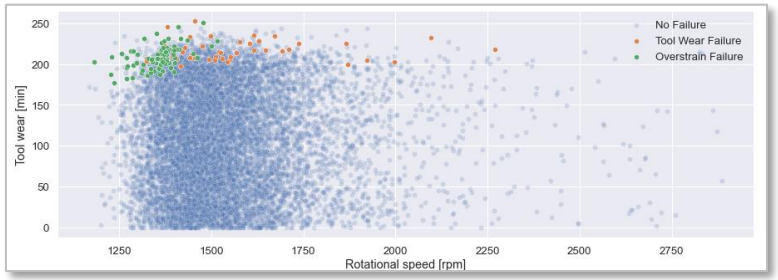


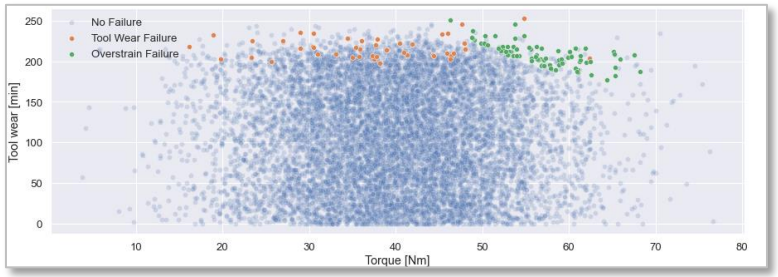
Final Project： Conclusion

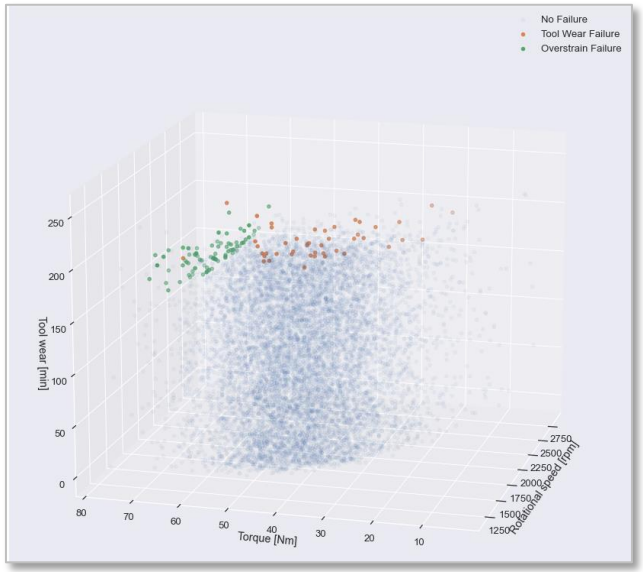
- ✓ 공구 사용시간이 180이상 이면 토크나 rpm에 따라 tool wear나 overstrain일어날 수 있으며, rpm 낮거나 overstrain 토크가 크다면 tool wear 발생 가능
- ✓ 180분 이상 사용시 공구를 교체하는 것으로 고장 예방 주기를 설정하는 것이 좋아 보인다고 판단함

label	Tool Wear Failure (TWF)	Heat dissipation failure (HDF)	Power Failure (PWF)	Overstrain Failure (OSF)	Random Failures (RNF)
Failure Type	the tool will be replaced or fail at a randomly selected tool wear time between 200 - 240 mins (120 times in our dataset). At this point in time, the tool is replaced 74 times, and fails 46 times (randomly assigned).	heat dissipation causes a process failure, if the difference between air- and process temperature is below 8.6 K and the tool's rotational speed is below 1380 rpm. This is the case for 115 data points.	the product of torque and rotational speed (in rad/s) equals the power required for the process. If this power is below 3500 W or above 9000 W, the process fails, which is the case 95 times in our dataset.	if the product of tool wear and torque exceeds 11,000 minNm for the L product variant (12,000 for M, 13,000 for H), the process fails due to overstrain. This is true for 98 datapoints.	each process has a chance of 0,1 % to fail regardless of its process parameters. This is the case for 19 datapoints, more frequent than could be expected for 10,000 datapoints in our dataset.
Result	rpm > 1379.5 and Tool wear > 194.5 and	rpm <= 1379.5 and Air temp > 301.35 and Torque <= 62.45	-	Rotational speed <= 1379.5 and Air temperature <= 301.35 and Torque <= 62.2 and Tool wear > 180	-









- 끝 -