2023
DATA ANALYSIS
for Predictive Maintenance

Team ‘K2S2’

CONTENTS

1 프로젝트 배경 및 목적
프로젝트 주제, 개요

2 데이터 분석 & 시각화
2가지 데이터를 이용한 분석 과정

3 결론
결론과 향후 과제

4 팀소개 & 느낀점
파트 분배, 과제 후 느낀점

01

프로젝트 배경 및 목적

- 주제 선정
- 프로젝트 목적
- 프로젝트 진행 과정

기계 고장 분석 및 예측을 통한 유지보수

'디스플레이 센서 불량 요인'이란 주어진 주제에서 불량의 주 원인을 기계의 고장이라 가정하였고,
기계의 고장을 방지를 할 수 있을지 알아보고자 하였다

기계 고장에 따른 문제점

자원의 문제

고장 및 불량품에 소요된 원재료 낭비

↳ 재료 비용 증가로 수익성 약화

불량품 보관 및 폐기 문제

↳ 보관 공간 부족 및 폐기 비용 발생

공정 과정의 문제

제품의 추가 제작 필요

↳ 재료, 시간, 인력, 비용 증가

불량품 및 고장 검사 추가 및 증가

↳ 추가적인 인력과 비용 증가

고장으로 인한 후속 작업 지연

↳ 고장 발생으로 인한 생산성 저하

프로젝트 목적

고장 발생 요인 정밀 분석

최적 유지 환경 탐색

↳ 고장 발생 최소화 조건 확인

이상치 데이터 분석

↳ 다양한 형태의 고장 원인 분석

유지 / 보수 방안 제시

기계 고장 발생 방지 모델 추천

↳ 기계별 고장율 분석

부품 교체 적정 주기 예측

↳ 정상과 고장 제품의 교체 주기 비교

프로젝트 개요

주제 선정 이유

불량의 주 원인을 기계의 고장이라 가정하였고,이를 어떻게 방지를 할 수 있을지, 또한 비용 절감과 성능 이상 검출을 통한 유지 보수 방법을 동시에 얻어내기 위함.

프로젝트 진행 순서

1. 주제 선정 및 데이터 수집
2. 데이터 분석 설계
3. 데이터 분석
4. 결론 도출 및 해석

Week1 주제 선정, 데이터 수집
Week2 분석 설계, 데이터 분석
Week3 결론 도출, 시각화 자료 제작
Week4 수정 보완

Python, Google Colab, PowerBI, Github, Notion

Kaggle, Predictive Maintenance Dataset
Kaggle, Microsoft Azure Predictive Maintenance

02

데이터 분석 & 시각화

- 2가지 데이터 소개
- 데이터별 기초/심화 분석
- 데이터별 결과 해석

데이터 설명

기계 고장 데이터(1)

① 개요
- 에러(고장x) : 발생 시간(시계열 데이터),
  기계ID(1~100), 에러ID(1~4) (3919 x 3)
- 고장 교체 기록 : 발생 시간(시계열 데이터),
  기계ID, 교체 부품 (761 x 3)
- 컴포넌트 교체 기록 : 발생 시간(시계열 데이터),
  교체 부품 (3286 x 2)
- 기계 정보 : 기계 ID, 모델 종류, 사용 연식 (100 x 3)
- 시계열 정보 : 시간 (시계열 데이터),
  기계ID, 전압, 회전, 압력, 진동, (876100 x 6)

<u>전처리</u>
공통적으로 2015년 데이터만 추출하여 사용한다.

<u>데이터 파일</u>
① PdM_errors.csv
② PdM_failures.csv
③ PdM_maint.csv
④ PdM_Machines.csv
⑤ PdM_telemetry.csv

기계 고장 데이터(2)

① 데이터명 : 밀링머신 고장 예측 유지 보수 데이터
② 개요 : Row 10,000개 (기계 ID), Column 14개 (변수 등)
③ 변수 : 기온, 공정온도, 회전속도, 회전력, 공구마모시간
④ 고장 원인 : 공구마모, 열방산, 동력이상, 과부하, 랜덤
⑤ 고장으로 분류되었으나 원인이 없는 데이터 9개와
랜덤 고장 19개만 분리하여 별도 분석
⑥ 공구마모 고장은 부품 교체가 원인인 바, 변수별 분석은
제외하고 공구 마모 시간과 함께 분석

<u>전처리</u>
기온, 공정온도의 켈빈 온도를 섭씨 온도로 변경한다.

<u>데이터 파일</u>
ai4i2020.csv

How

공정 과정 중 발생하는 고장의 원인들을 탐색하며,
효과적인 유지보수 방법을 찾기 위한 분석을 진행.

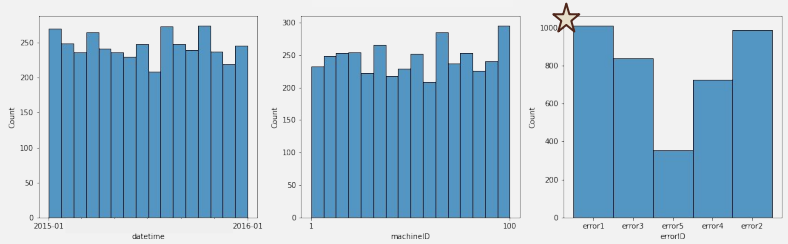
Goal

효과적인 유지보수 방법을 찾아, 비용 및 시간 절감 효과를 가져온다.

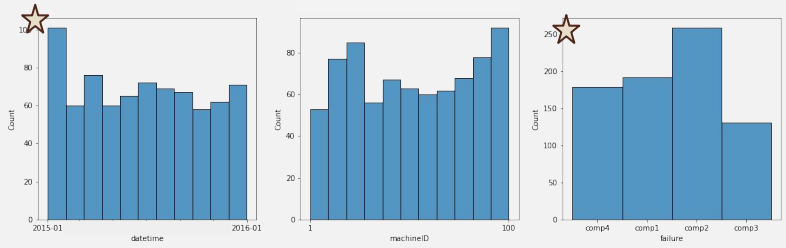
기계 고장 데이터(1)

| 기초 분석과 심화 분석 전 데이터별 시각화

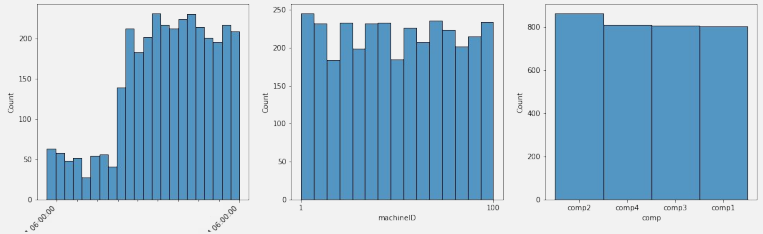
① PdM_errors.csv



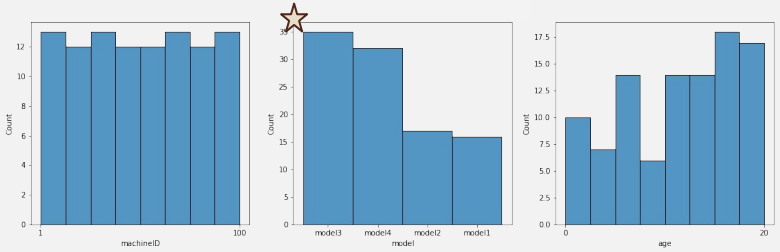
② PdM_failures.csv



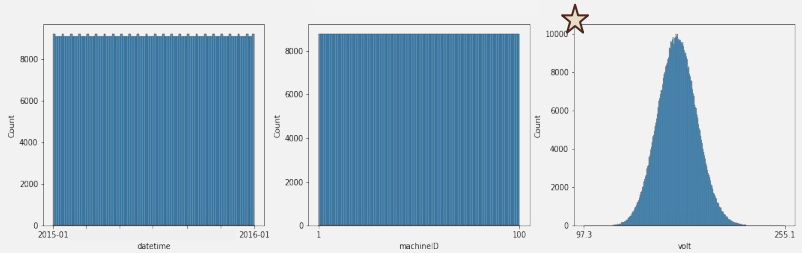
③ PdM_maint.csv

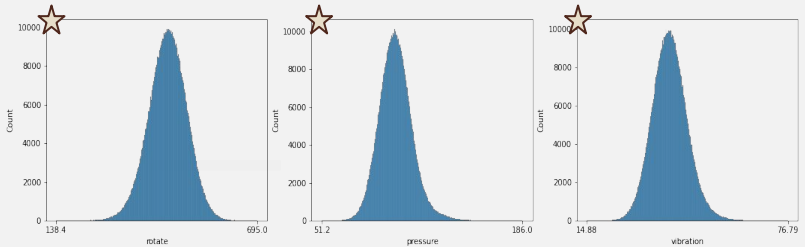


④ PdM_Machines.csv



⑤ PdM_telemetry.csv





데이터1 기초 분석

기초 통계 분석 결과

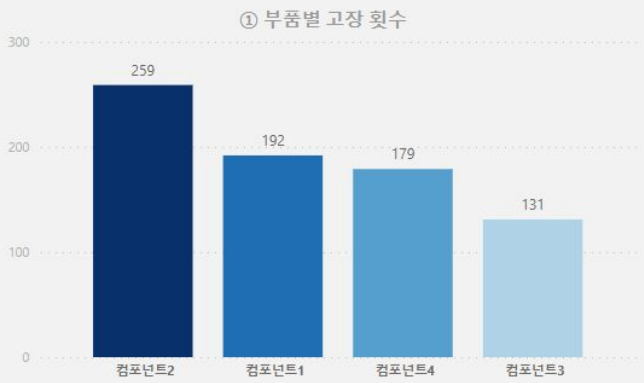
① 부품별 고장 횟수
Comp2가 고장난 횟수가 가장 많으며,
Comp2의 고장 횟수는 Comp3의 약 2배이다.

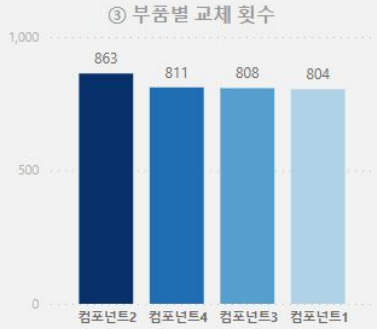
② 에러별 측정 횟수
Error 1과 Error 2가 다른 에러들에 비해 높게 측정.
Error 5가 다른 에러들에 비해 낮게 측정.

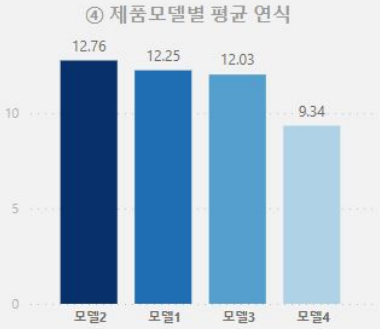
③ 부품별 교체 횟수
Comp2의 교체 횟수가 나머지에 비해 약간 많지만,
전체적으로 비슷한 수준을 보인다.

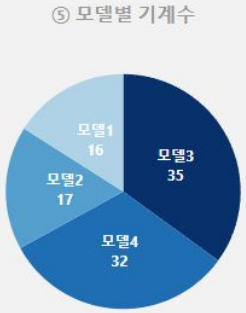
④ 제품 모델별 평균 연식
Model2의 연식이 다른 Model들에 비해 높게 측정.
Model4의 연식이 다른 Model들에 비해 낮게 측정.

⑤ 모델별 기계수
모델별로 기계수가 다르며
Model3, 4가 Model1, 2의 2배 정도 차이가 있다.









show code

데이터1 심화 분석

구체적인 분석 결과 및 과정

① 연식, 에러, 교체, 고장 간의 연관성

연식과 고장횟수 / 에러횟수와 고장횟수는 각각 약한 상관관계가 존재.
상관 관계도를 통해 특정값 이상과 이하의 평균 차이 파악.

| 전처리

각 파일에서 필요한 데이터 부분만 추출해서, 새로운 데이터프레임으로 만들었다.

```
# 필요한 데이터 추출
machines_age =
pd.DataFrame(machines_df[['machineID','age']]).set_index('machineID')
machines_errors =
pd.DataFrame(errors_df.groupby('machineID')['errorID'].count())
machines_maint =
pd.DataFrame(maint_df.groupby('machineID')['comp'].count())
machines_failures =
pd.DataFrame(failures_df.groupby('machineID')['failure'].count())
```

| 분석 과정

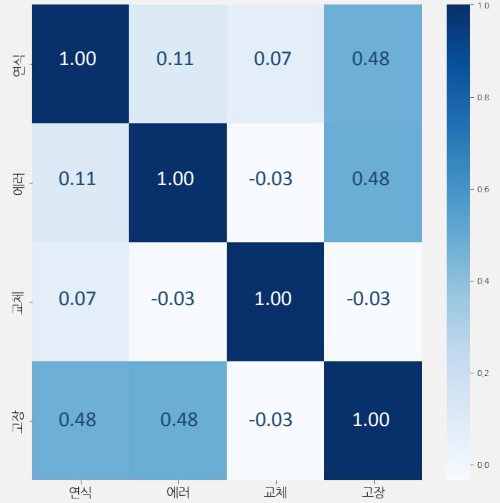
피어슨 상관계수를 이용해 연관성을 찾았다.

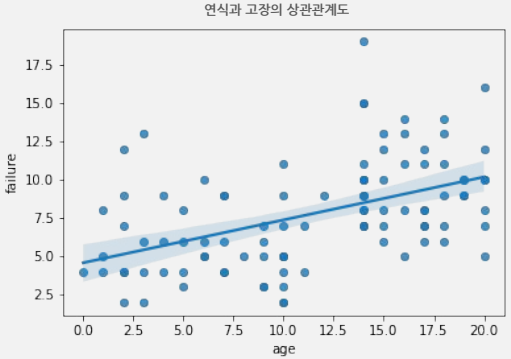
```
# 상관관계 찾기 (피어슨 상관계수 사용)
df_machines.corr(method = "pearson")

# 추출한 데이터 합치기
df_machines = pd.concat([machines_age, machines_errors,
machines_maint, machines_failures], axis=1)
df_machines.rename(columns={'errorID':'error',
'comp':'replace'},inplace=True)

# 추천 교체 연식 구하기
for문을 통해 특정 시점을 기준으로 얼마나 고장 빈도가 증가하는가 살펴보았다
```

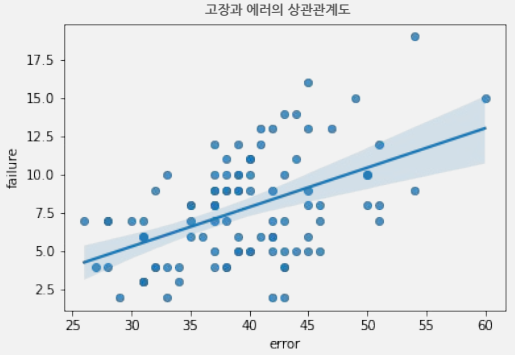
| 시각화

히트맵과 분포도 그래프를 이용해 한 눈에 연관성을 알아볼 수 있게 했다.

```
# 상관관계도를 통한 정보 추출

 상관관계도를 통해 연식과 고장, 에러와 고장 사이에는 양의 상관관계가
있음을 알 수 있다.

# 추천 교체 연식

14년 기준으로 이전과 이후의 고장 빈도를 비교했을 때, 평균 4.16회 증가로
이 때를 기점으로 기계교체를 하여 효율적으로 고장 빈도를 줄이는 것을 추천
```







show code

데이터1 심화 분석

구체적인 분석 결과 및 과정

② 요인들의 이상치와 고장 간의 연관성 *요인 : 전압, 회전, 압력, 진동

기계 고장은 압력과 진동 이상치와 연관성이 있다.

이상치의 최소값과 최대값을 파악하였다.

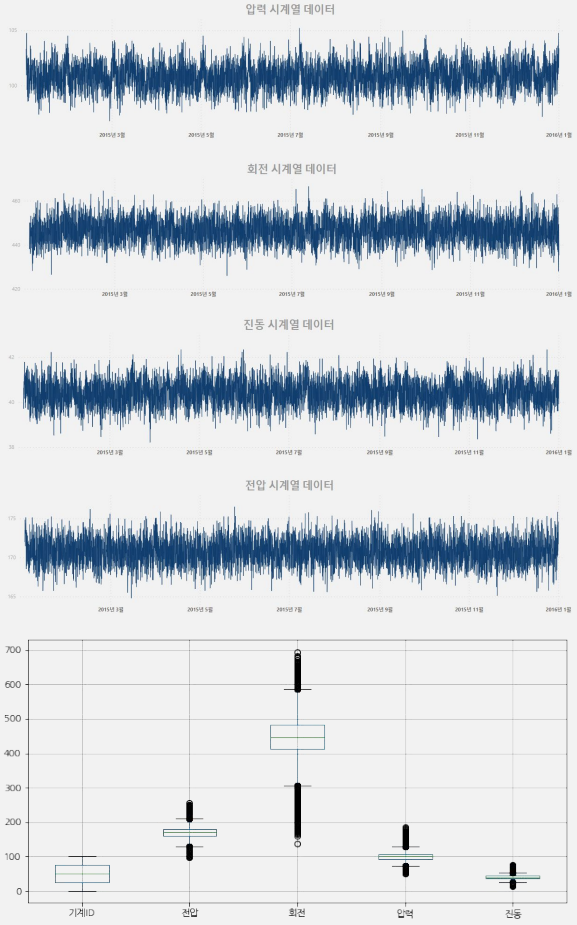
| 분석 과정

고장 데이터와 각 요인별 이상치 범위의 데이터 추출 후,

피어슨 상관계수를 이용해 연관성을 찾았다.

```
# 각 요인별 이상치 데이터 추출하는 함수 정의
def outlier(df, column):
    Q1 = np.quantile(df[column], 0.25)
    Q3 = np.quantile(df[column], 0.75)
    IQR = Q3 - Q1
    minimum = Q1 - (IQR * 1.5)
    maximum = Q3 + (IQR * 1.5)

    machineID = []
    value = []
    for i in range(len(df[column])):
        if df[column][i]>=maximum or df[column][i]<=minimum:
            machineID.append(df['machineID'][i])
            value.append(df[column][i])
    return pd.DataFrame({'machineID': machineID, column: value})

# 각 요인별 이상치 데이터 추출
volt_outlier = outlier(telemetry_df, 'volt')
# 이상치 데이터에서 기계별로 각 이상치가 몇번씩 측정되었는지 확인
volt_out_df =
pd.DataFrame(volt_outlier.groupby('machineID')['volt'].count())
# 기계별 고장횟수 & 이상치데이터 합치기
machines_outliers = pd.concat([machines_failures, volt_out_df,
rotate_out_df, pressure_out_df, vibration_out_df], axis=1)
# 기계별 고장 횟수와 이상치별 측정 횟수 상관관계 찾기 (피어슨 사용)
machines_outliers.corr(method = "pearson")
```

show code

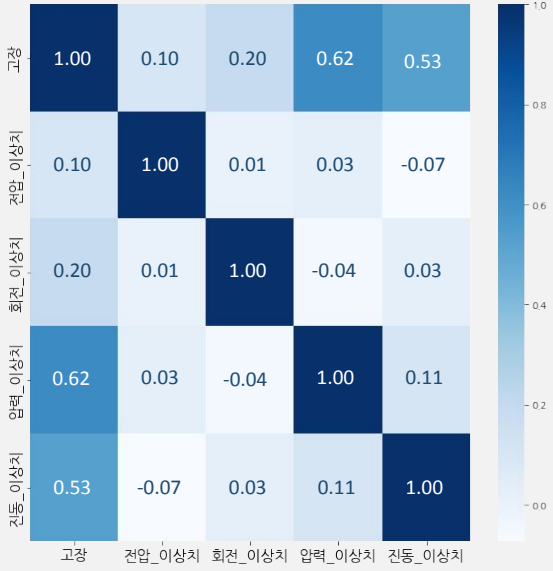


요인별 이상치 분석

	전압	회전	압력	진동
이상치(최소)	129	307	72	26.3
이상치(최대)	212	587	129	54.3

| 시각화

히트맵을 이용해 한 눈에 연관성을 알아볼 수 있도록 하였다.



데이터1 심화 분석

	고장교체주기 (일)	정기교체주기 (일)	총 평균 (일)	고장교체 (번)	정기교체 (번)	총 합계 (번)
컴포넌트1	124.04	68.172	94.32	422	381	803
컴포넌트2	136.96	55.120	85.59	517	345	862
컴포넌트3	92.89	75.642	86.96	293	514	817
컴포넌트4	117.81	66.161	89.87	431	379	810

구체적인 분석 결과 및 과정

③ 부품별 교체 주기

고장 교체 주기에 비해 정기 교체 주기가 짧다

평균 교체주기만큼 정기교체주기를 늘리는것을 추천.

| 전처리

* 기본 데이터 (주) : Maint 4806 데이터량 (ID별, 시간별 정렬)
* 기본 데이터 (보조): Failures 762 데이터량 (ID별, 시간별 정렬)
* Maint데이터와 Fail데이터를 합쳤을때 총1520데이터열 전처리
* 3,282의 고장 + 정기적 교체 주기 데이터 확보

```
# 고장으로 인한 교체 주기 추출을 위한 전처리
n = maint_df.merge(failures_df, on =
('machineID','datetime'),how = 'inner')
for i in range(len(n)):
  if n['comp'][i] == n['failure'][i]: #겹치는 데이터 (고장)
    df.loc[i] =
[n['machineID'][i],n['datetime'][i],n['failure'][i]]
  else:#겹치지않는 데이터 (정기적)
    pass
```

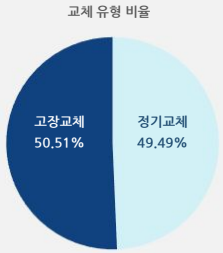
```
# 정기적 교체 주기 추출을 위한 전처리
# 전체 교체주기 - 고장 교체주기 = 일반 교체주기
list_test_positive = [] #고장 안 난 데이터
list_id_positive = []
for i in range(len(comp1_n)):
  if comp1_n['datetime'][i] not in list(df_com1['datetime']):
#겹치는 데이터 (고장)
    pass
  else:
    #겹치지 않는 데이터(정기적)
    list_test_positive.append(comp1_n['datetime'][i])
    list_id_positive.append(comp1_n['machineID'][i])
```

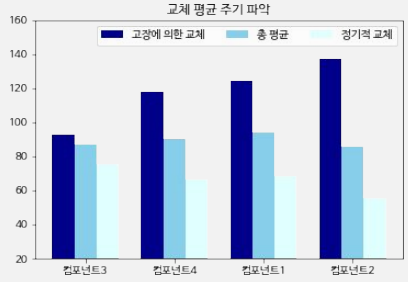
| 분석 과정

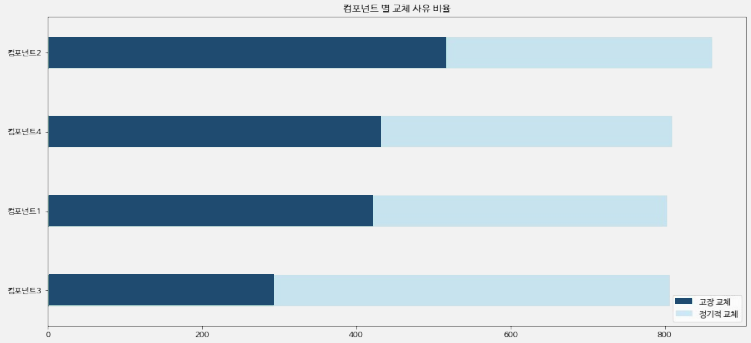
datetime을 이용해 부품별 교체 주기 확인을 위해 아래 코드로 진행, 리스트의 음수를 한개씩 있는 machineID의 time_diff값 None 처리.

```
# 교체 주기 확인 코드

comp1_re = []
n = 1
for i in range(len(df_test)-1):
  if n != range(len(df_test)):
    str_datetime1 = df_test["datetime"][n]
    str_datetime2 = df_test["datetime"][i]
    currdate1 =
datetime.strptime(str_datetime1,
'%Y-%m-%d %H:%M:%S')
    currdate2 =
datetime.strptime(str_datetime2,
'%Y-%m-%d %H:%M:%S')
    new = currdate1- currdate2
    comp1_re.append(new.days)
    n += 1
  else:
    break
```

| 시각화







show code

데이터1 심화 분석

구체적인 분석 결과 및 과정

④ 모델 종류별 고장 비율

Model 1, 2가 고장 비율이 높은 것을 알 수 있다.

| 전처리

각 파일에서 필요한 데이터 부분만 추출해서, 새로운 데이터프레임으로 만들었다.

```
# 필요한 데이터 추출
machines_failures =
pd.DataFrame(failures_df.groupby('machineID')['failure'].count())
machines_model =
pd.DataFrame(machines_df[['machineID','model']]).set_index('machineID'
)
# 기계별 모델, 고장 데이터 합치기
model_fail = pd.concat([machines_model, machines_failures], axis=1)
```

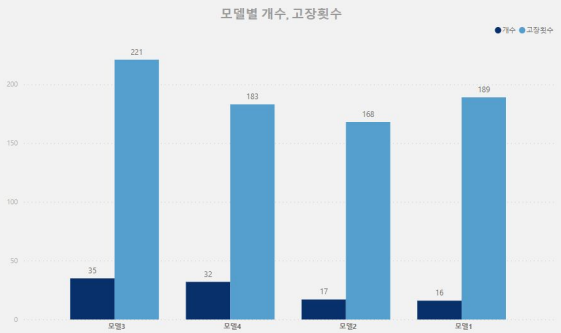
| 분석 과정

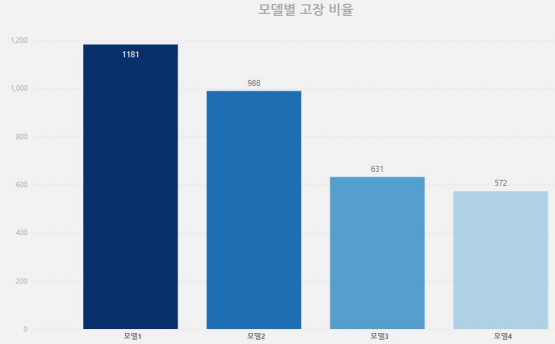
모델별 보유 개수, 고장 횟수를 추출한 후, 고장비율을 계산해 새로운 컬럼을 추가했다. *고장비율=고장횟수/보유개수*100

```
# 모델별 고장 횟수 추출
models_failures = pd.DataFrame(model_fail.groupby('model')['failure'].sum())
# 모델별 보유 개수
models_count = pd.DataFrame(machines_df.groupby('model')['machineID'].count())
# 모델별 보유 개수 & 고장 횟수 합치기
models_failures_count = pd.concat([models_count, models_failures], axis=1)
# 고장비율 컬럼 추가 (고장횟수/보유개수*100)
models_failures_count['Failure rate'] = models_failures_count['failure'] / models_failures_count['num'] *100
```

| 시각화

	개수	고장 횟수	고장 비율(%)
Model1	16	189	1181.25
Model2	17	168	988.24
Model3	35	221	631.43
Model4	32	183	571.88





show code

데이터1 결과 해석

〈유지〉

고장 발생 원인 발견

고장과 압력, 진동 이상치 연관성 有

이상치를 정상 수치 범주 안으로 조정한다면 기계 고장 발생을 줄일 수 있을 것으로 판단된다.

연식의 상관관계 有

연식에 따른 고장 발생 분석 결과, 14년 이상일 때 고장의 빈도가 높았다. 따라서 사용한 지 14년이 되었을 때 기계를 교체해주면 고장 빈도를 줄일 수 있을 것으로 판단된다.

〈보수〉

부품 교체 주기 수정

정기 교체 시기를 한 달 연장 가능

평균 교체 주기 수정을 통해 약 한 달 가량의 시간을 더 사용하여 부품 소모비와 시간을 절약 할 수 있을 것으로 판단된다.

정기 교체주기에 맞게 교체함으로써 고장으로 인한 교체를 낮출수 있을 것으로 판단된다.

〈보수〉

기계 교체 시 추천 모델

기계 교체 시, Model 3과 4 추천

4가지 모델의 개수와 고장 횟수를 비교한 결과, 모델 1과 2의 고장 비율이 높은 것을 알 수 있었다.

기계 고장이 발생하여 교체해야 할 시기가 도래한다면, 모델 3 또는 4로 교체할 것을 추천한다.

데이터2 기초 분석

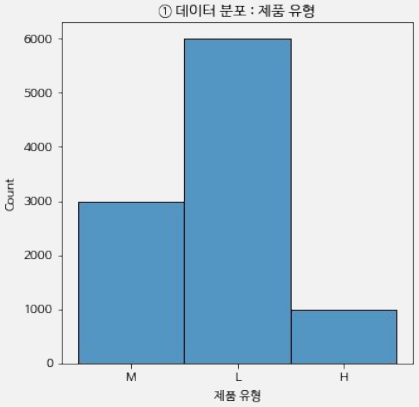
기초 통계 분석 결과

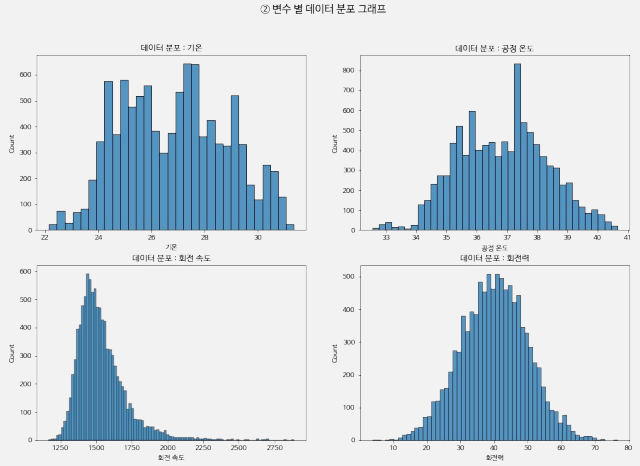
① 제품 유형
L타입 60%, M타입 30%, H타입 10%의 분포

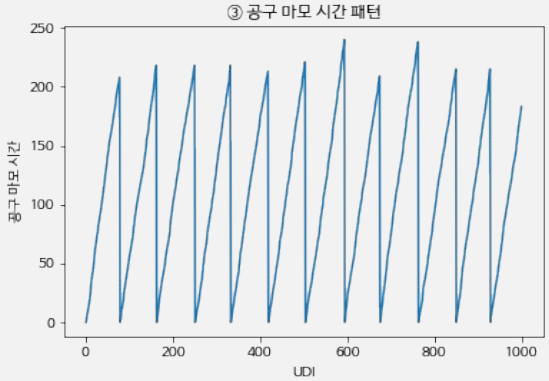
② 기온, 공정 온도, 회전력 / 회전 속도
정규 분포 / 편향 분포를 보인다.

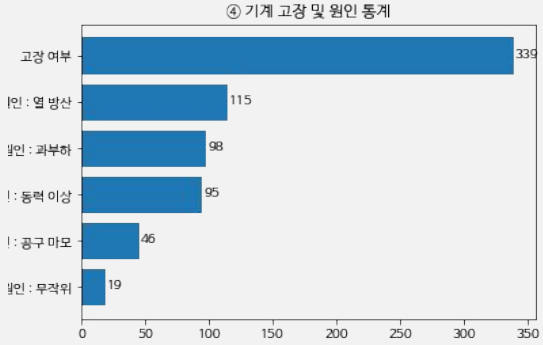
③ 공구 마모 시간
0부터 200 언저리까지 올라갔다가 교체되어 다시 0으로 돌아오는 형태 양상.

④ 기계 고장
총 10000건의 데이터 중 339 건의 기계고장 데이터가 있으며, 그 중에 열 방산 115회, 과부하 98회, 동력 이상 95회, 공구 마모 46회, 랜덤 19회의 데이터가 중복.









show code

데이터2 심화 분석

구체적인 분석 결과 및 과정

①-1 고장 원인별 환경요소와의 연관성

열 방산으로 인한 고장의 경우, 기온과의 연관성이 가장 높다.
과부하 고장의 경우, 공구마모와의 연관성이 가장 높다.
동력 이상 고장의 경우, 회전 속도와의 연관성이 가장 높다.

| 전처리

고장이 발생한 것 들로 데이터 그룹을 생성

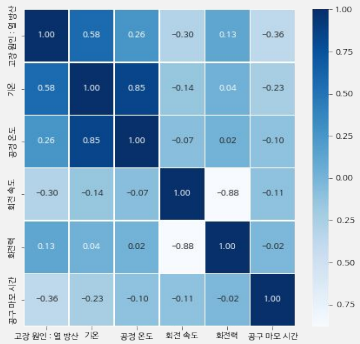
```
# 필요한 데이터 추출
df_failure =df.groupby('고장 여부')
df_fail_1=df_failure.get_group(1)
```

| 분석 과정

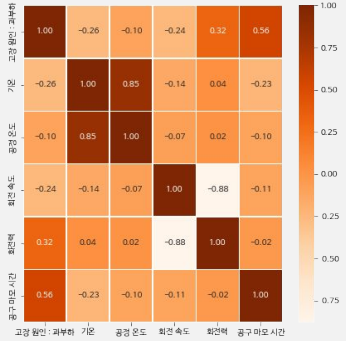
고장 원인과 5가지 변수들 간의 상관계수를 이용해
연관성 발견.

```
# 상관관계 찾기
data_HDF = df_fail_1[['고장 원인 : 열 방산','기온' , '공정온도',
'회전 속도','회전력', '공구 마모 시간']]
sns.heatmap(data = data_HDF.corr(), annot=True,
fmt = '.2f', linewidths=.5, cmap='Blues')
```

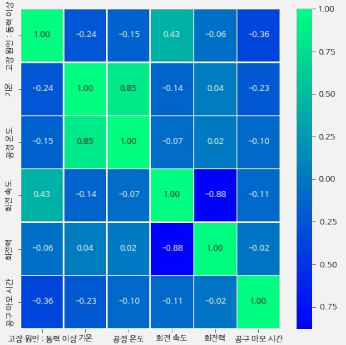
열 방산 : 변수 별 상관계수



과부하 : 변수 별 상관계수



동력 이상 : 변수 별 상관계수



열 방산 데이터 값

열방산	최소	평균	최대
기온	27.65	29.41	30.55
공정온도	36.25	37.63	39.05
회전속도	1212.00	1337.26	1379.00
회전력	41.60	53.16	68.20
공구마모시간	2.00	107.19	229.00

과부하 데이터 값

과부하	최소	평균	최대
기온	22.45	26.89	30.85
공정온도	32.95	36.92	39.95
회전속도	1181.00	1350.32	1515.00
회전력	46.30	58.37	75.40
공구마모시간	172.00	207.69	253.00

동력 이상 데이터 값

동력 이상	최소	평균	최대
기온	22.55	26.92	30.85
공정온도	33.05	36.80	40.05
회전속도	1200.00	1763.96	2886.00
회전력	3.80	48.51	76.60
공구마모시간	0.00	101.88	234.00

show code

데이터2 심화 분석

구체적인 분석 결과 및 과정

①-2 고장 원인들과 정상과의 비교

정상 그래프와 비교했을때, '열 방산'은 평균 기온이 2.58도 높고, '동력 이상' 은 평균 회전 속도가 223.7RPM 높으며, '과부하'는 평균 공구 마모 시간이 101min 길다. 또한, 모든 고장 유형에서 회전력이 정상 수치보다 높다.

| 분석 과정

변수 상황 별 3가지 고장 원인과 정상간의 비교

열방산과 정상값의 차이

열방산	최소	평균	최대
기온	+5.50	+2.58	-0.80
공정온도	+3.70	+0.79	-1.60
회전속도	+44.00	-202.99	-1316.00
회전력	+29.00	+13.53	-1.80
공구마모시간	+2.00	+0.49	-17.00

동력이상과 정상값의 차이

동력 이상	최소	평균	최대
기온	+0.4	+0.10	-0.5
공정온도	+0.5	-0.04	-0.60
회전속도	+32.00	+223.7	+191.00
회전력	-8.80	+8.88	+6.60
공구마모시간	0.00	-4.80	-12.00

정상 데이터 값

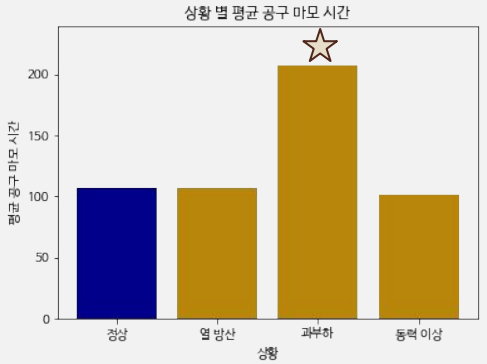
정상	최소	평균	최대
기온	22.15	26.82	31.35
공정온도	32.55	36.84	40.65
회전속도	1168.00	1540.26	2695.00
회전력	12.60	39.62	70.00
공구마모시간	0.00	106.69	246.00

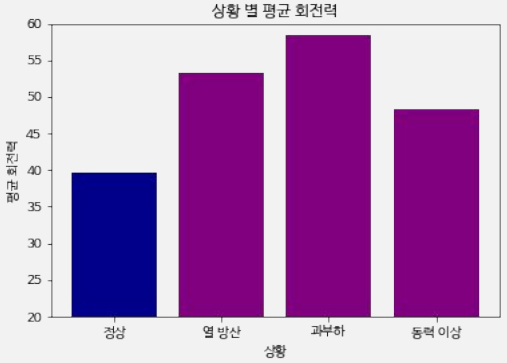
과부하와 정상값의 차이

과부하	최소	평균	최대
기온	+0.3	+0.07	-0.5
공정온도	+0.4	+0.07	-0.7
회전속도	+13.00	-189.93	-1180.00
회전력	+33.70	+18.74	+5.4
공구마모시간	+172.00	+101.00	+7.00

show code

| 시각화





데이터2 심화 분석

구체적인 분석 결과 및 과정

①-3 고장 원인의 분포별 특이점

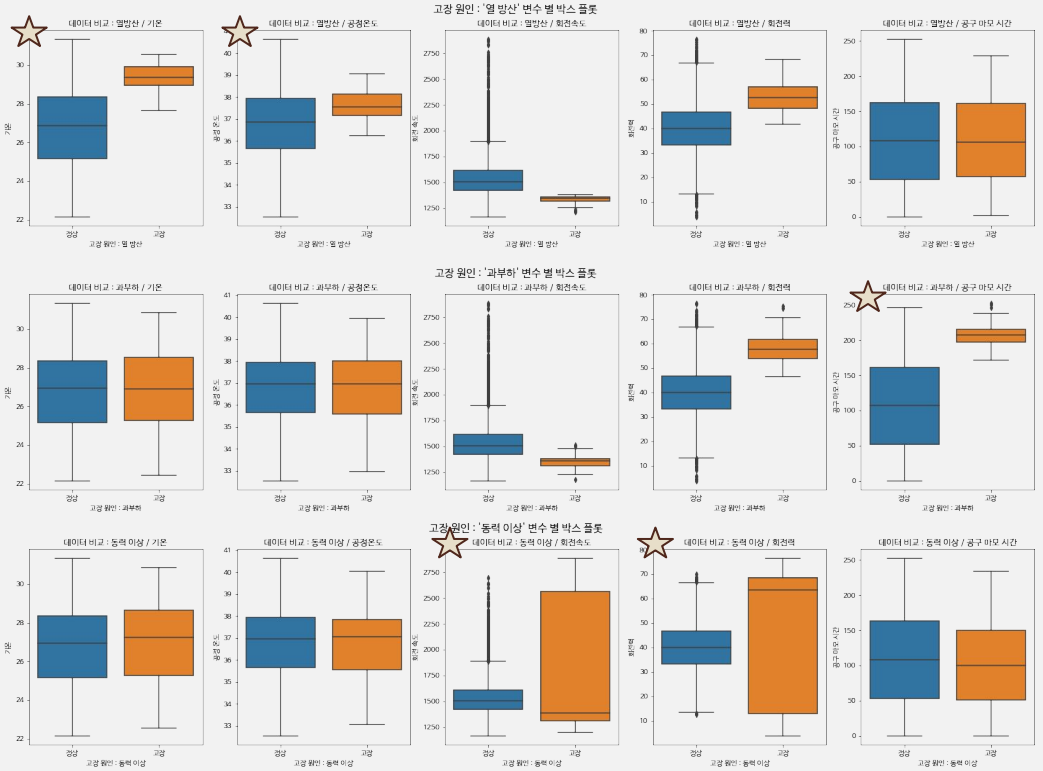
열 방산으로 인한 고장의 경우, 기온, 공정온도에서
중상위 구간에 몰려있는 분포를,
과부하 고장의 경우, 공구마모에서
상위 구간에 몰려있는 분포를,
동력 이상 고장의 경우, 회전 속도, 회전력에서
위 아래로 분산되어 있는 분포를 보인다.

| 분석 과정

박스 플롯을 이용한 고장 원인 및 변수간의 분포 집중도 비교

```
# 고장 원인 별 박스플롯 비교 분석
 ax=fig.add_subplot(1,5,1 )
 sns.boxplot(x='고장 원인 : 열방산', y='기온',
data=df, ax=ax1)
```

①-3 변수와 고장 원인의 분포별 특이점 시각화(박스플롯)



show code

데이터2 심화 분석

구체적인 분석 결과 및 과정

①-3 고장 원인의 분포별 특이점

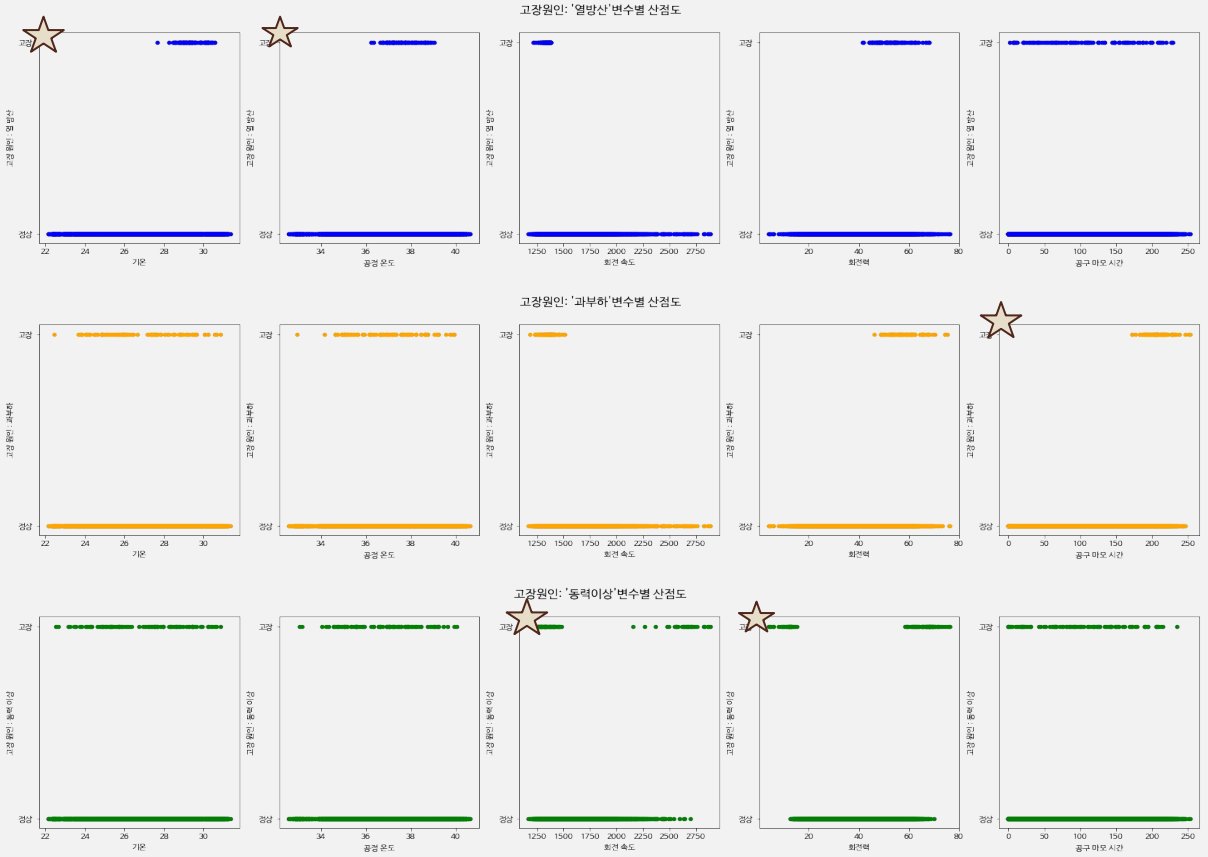
열 방산으로 인한 고장의 경우, 기온, 공정온도에서
각각 27.65도, 36.25도 이상의 구간에 몰려있는 분포를,
과부하 고장의 경우, 공구마모에서
172 min 이상의 구간에 몰려있는 분포를,
동력 이상 고장의 경우, 회전 속도, 회전력에서 각각
1479 rpm 이하와 2153 rpm 이상,
15.3 Nm 이하와 58.5 Nm 이상의 구간에
나누어져 있는 분포를 보인다.

| 분석 과정

산점도를 이용해 연관성 발견.

```
# 고장 여부 별 산점도 분석하기
 ax=fig.add_subplot(1,5,1 )
 df.plot(x='기온', y='고장 원인 : 열 방산',
kind='scatter',  c='blue', ax=ax1)
```

①-3 변수와 고장 원인의 분포별 특이점 시각화(산점도)



데이터2 심화 분석

구체적인 분석 결과 및 과정

② 고장 원인이 랜덤일때, 각각의 케이스 분석

각 케이스마다 다른 고장들과 비교분석하여 유사점과 차이점에 대해 조사했다.

이상: ▲
이하: ▼
유사: –

| 전처리

랜덤 고장 원인 19개와 고장원인을 알 수 없는 데이터 9개를 합쳐 새로운 데이터프레임을 생성했다.

```
# df_failure_add 데이터 값 수정 (Machine failure : 1 -> 0 / RNF : 0 -> 1)
df_failure_add['고장 여부'].replace(1,0, inplace = True)
df_failure_add['고장 원인 : 무작위'].replace(0,1, inplace = True)
df_failure_add

# df_RNF 데이터 값 수정 (Machine failure : 1 -> 0 / TWF : 1 -> 0)
df_RNF['고장 여부'].replace(1,0, inplace = True)
df_RNF['고장 원인 : 공구 마모'].replace(1,0, inplace = True)
df_RNF

# RNF 와 failure만 분류한 것 합친 데이터 28 (19+9)
df_RNF_plus = pd.concat([df_RNF, df_failure_add])
```

| 분석 과정

전처리 된 랜덤 고장 데이터 중 특이점이 보인 데이터를 골라 각각 타 고장 유형 데이터와 비교 분석했다.

제품ID	기온	공정 온도	회전 속도	회전력
M19544	30.5	38.7	1421	44.8
L52651	29.6	39.2	1346	61.2
L55686	25.3	36.5	1710	27.3

기온, 공정온도 최대치 –
회전속도 최대치 ▲
회전력 평균 ▼

열방산 데이터 값

열방산	최소	평균	최대
기온	27.65	29.41	30.55
공정온도	36.25	37.64	39.05
회전속도	1212.00	1337.03	1379.00
회전력	41.60	53.21	68.20

기온, 공정온도 최대치 –
회전속도, 회전력 평균 –

과부하 데이터 값

과부하	최소	평균	최대
기온	23.65	27.03	30.85
공정온도	34.15	37.04	39.95
회전속도	1181.00	1346.37	1501.00
회전력	46.30	58.51	75.40

기온, 공정온도, 회전속도 –
회전력 평균 ▼

동력이상 데이터 값

동력 이상	최소	평균	최대
기온	22.55	26.92	30.85
공정온도	33.05	36.80	40.05
회전속도	1200.00	1768.22	2886.00
회전력	3.80	48.34	76.60

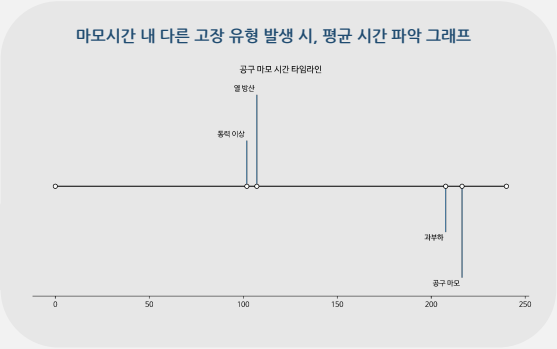
show code

데이터2 심화 분석

구체적인 분석 결과 및 과정

③ 공구마모교체 / 정기교체 사이클 패턴

고장 원인별 공구 마모 시간 평균을 볼 때, 동력 이상이 101.9min,
열방산이 107.2min, 과부하가 207.7min, 공구 마모가 216.4min 에 일어났다.
공구 교체 횟수는 정기 교체가 64회, 고장 교체가 55회로 총 119회이고,
교체 주기 평균은 정기 교체가 215.3min, 고장 교체가 216.1min 이다.

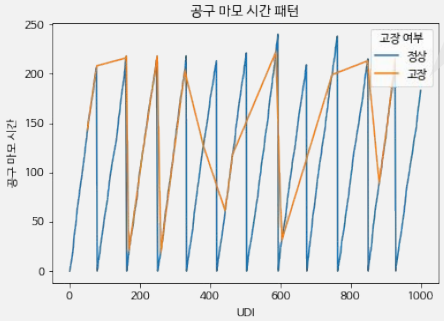


| 전처리

공구 마모 시간 패턴을 정상과 고장으로 분리하였다.

```
# 공구 교체 시기 측정
df_replace_regular = []
df_replace_error = []
for i in range(len(df) - 1):
    if df['공구 마모 시간'][i+1] > df['공구 마모 시간'][i+2]:
        if df['고장 여부'][i+1] == 0:
            df_replace_regular.append(df['공구 마모 시간'][i+1])
        else:
            df_replace_error.append(df['공구 마모 시간'][i+1])
```

| 시각화

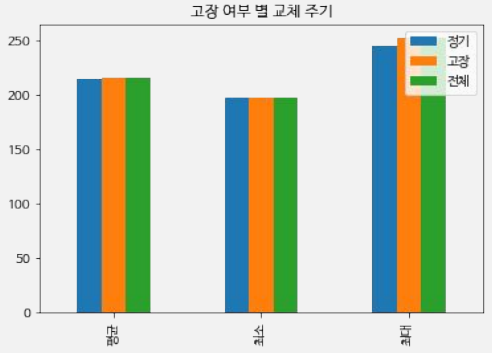


| 분석 과정

공구 마모 시간 패턴을 통해 정기적 교체와 고장에 의한 교체를
파악하였고, 이를 통하여 고장 여부별 교체주기를 구하였다.

```
# 시각화
sns.lineplot(x=x, y=y, hue=hue_data)
replace.T.plot(kind='bar', title = '고장 여부 별 교체 주기')
```

교체 주기	최소	평균	최대	교체 횟수
정기	198	215.3	246	64
고장	198	216.1	253	55



show code

데이터2 결과 해석

〈유지〉

고장 원인과 변수 관계 분석

고장 발생을 줄이기 위한 환경 추천

기온 27.65도 이하
공정온도 36.25도 이하
회전 속도 1,479 ~ 2,153 RPM 범위
회전력 15.3 ~ 58.5 Nm 범위
공구 마모 시간 172 min 이하

〈유지〉

랜덤 고장 분석 가능성

랜덤 데이터를 이용한 분석

랜덤 고장 데이터들과 다른 고장 유형 데이터의 수치를 비교 분석하여 랜덤한 데이터들의 특이점을 발견했다.

고장 데이터들을 더 학습시킨다면, 이러한 분석을 통해 고장 원인과 변수 관계를 분석하고 고장 발생을 줄일수 있을것으로 전망된다.

〈보수〉

공구 교체 시기 조절

적정한 공구 교체 시기 예측

정기적인 공구 교체시기와 고장으로 인한 교체 주기가 비슷한것으로 확인되었다.

현재 데이터 분석을 통한 결과로, 정기적 공구 교체시간을 215 min으로 유지하는 것이 가장 긍정적 결과를 얻을 수 있을 것이다.

03

결론

- 데이터 분석 결과 정리
- 결론 및 향후 과제

데이터 분석 결과 정리

모델 & 환경 추천

추후 기계 교체 시, 고장 비율이 낮은 모델3 또는 모델4로 교체하는것을 추천하며.
환경적으로는 기온, 공정온도, 회전 속도, 회전력별 적합한 수치를 유지하여, 기계 고장 발생을 줄일 수 있을 것으로 기대된다.

부품 교체 주기

고장 주기와 정기 교체 주기를 비교하여, 교체 주기를 조절한다면 소모 비용 및 시간을 절약하는데 도움이 될 것으로 해석된다.

첫번째 데이터처럼 고장 주기보다 정기 교체 주기가 빠른 경우,
정기 교체 시기를 늦춤으로써 교체 횟수를 줄임과 동시에 교체에 필요한 비용과 시간도 절약이 가능하다.

고장 발생 원인

앞으로 고장과 연관성이 높은 압력, 진동 이상치의 발생 원인을 구체적으로 탐색하여 줄이면 고장 발생을 줄일수 있을것이며, 또한 연식을 14년 기준으로 교체한다면 고장 발생을 낮출수있을것으로 기대된다. 이를 통해 적절한 공장 가동 환경 구성하여 유지 상태를 최적화할 수 있다.

랜덤 고장 케이스들에 대한 다른 데이터가 더 주어진다면, 추가적인 고장 발생 원인을 학습시켜 더 많은 분석을통해 더 정확한 고장 케이스들의 원인 해결 가능.

결론 및 문제 극복

모델 & 환경 추천

추후 기계 교체 제안과 적합한 환경 유지를 통해 기계 고장 발생률 감소시킬 수 있으며,고장을 방지하여 생산성 증가를 기대할수 있습니다. 또한 불량품 생산에 따른 추가 제작에 드는 재료, 시간, 인력, 비용을 감소시킬수있다.

부품 교체 주기

고장 주기와 정기 교체 주기를 비교 및 조절함으로써 기계 고장에 의해 소요된 부품 낭비를 줄이고, 부품 비용을 감소시켜 수익성을 증가시키며, 불량 부품의 보관 공간 낭비 및 폐기 비용을 줄일 수 있다.

고장 발생 원인

고장과 연관성이 높은 압력, 진동 및 연식의 분석을 통해 적절한 공장 가동 환경을 구성하여 유지 상태를 최적화 할 수 있다. 이를 통해 고장 발생 원인 검사 비용과 투입되는 인력 비용을 줄일 수 있다.

결론 및 향후 과제

분석 결과와 결론을 통해 기계 고장 방지를 줄이기위한 방법, 그리고 비용절감과 성능 이상 검출을 통한 유지보수 방법을 얻어낼수 있었고 이를 통해 기계 고장으로 인한 문제점들을 극복 혹은 긍정적 결과를 얻어낼 수 있을 것이다.

추가제작에 드는 재료, 시간, 인력, 비용, 수익성 등 관련 데이터를 구할 수 있다면 더 확실한 변화를 볼 수 있을 것으로 추정된다.

04

팀원 소개 및 느낀점

- 팀원 역할
- 느낀점

팀원 역할 및 느낀 점

느낀 점

데이터 수집에 한계가 있던 주제를 선택하여 아쉬움이 크다. 좀 더 구체적인 데이터가 주어졌다면, 더 구체적인 결과 해석을 할 수 있었지않을까?

데이터 수집의 한계와 제한된 주제 범위에 어려움이 있었고 이를 통해 데이터 분석을 할 때 데이터의 중요도가 얼마나 큰지 느꼈고, 데이터의 양과 제공된 정보에 따라 얻어낼 수 있는 결과물이 달라진다는 것을 가장 크게 느껴 아쉬웠다.

프로젝트를 수행함에 있어 팀원들 간의 소통이 무엇보다 중요함을 알 수 있었으며, 원활한 의견 교환과 협업이 팀 성과를 효율적으로 가져올 수 있다는 것을 배웠다,

프로젝트를 처음 시작할 땐 갈 길도 멀고 막막했는데, 팀원과 같이 협력하여 하나씩 차근차근 해나가다보니 이렇게 성과를 낼 수 있었다.

김영경

- 데이터 수집
- 데이터 설계
- 데이터1 분석 및 시각화
- 전반적인 ppt 제작

김창균

- 데이터 수집
- 데이터2 분석 및 시각화
- ppt 제작
- 데이터2 발표

송찬의

- 데이터 수집
- 데이터 설계, 의견 조율
- 데이터1 분석 및 시각화
- ppt 제작
- 전체 발표

신진섭

- 데이터 수집
- 데이터2 분석 및 시각화
- ppt 제작

Q&A

THANK YOU!

DATA ANALYSIS POWERPOINT

Team ‘K2S2’