

소음 및 진동 예방

학번: 2218069

이름: 안서연

Github address: <https://github.com/2218069/Noise-vibration-prevention>

1. 소음 및 진동 예방 머신러닝 모델 개발의 목적

학습 모델 활용 대상: 제조업 등 기계를 이용하는 산업 현장 작업자를 대상으로 작업 환경의 온도나 습도, 작업자의 근무 시간 데이터, 기계의 마모 정도나 기계 개수 데이터의 독립 변수를 이용하여 소음과 진동의 수준인 종속 변수를 예측한다. 소음 및 진동은 여러 곳에 영향을 끼칠 수 있다. 우선, 기계를 이용하는 산업현장에서 안전성이 매우 높아진다. 머신러닝 모델을 통해 소음 및 진동 수준을 예측함으로써 작업자의 안전을 높일 수 있고 유해하거나 위험한 소음 및 진동이 감지되면 즉각적으로 경고나 조치를 통하여 사고 발생 가능성을 줄일 수 있다. 또한 작업환경을 개선함으로써 작업자의 건강을 보호할 수 있다. 이에 따라 사고나 질병으로 인한 비용을 감소 시킬 수도 있다.

2. 소음 및 진동 예방 머신러닝 모델의 네이밍의 의미

- a. 안전 보건에 관련하여 소음 및 진동은 작업자들의 건강과 밀접한 관련이 있고 기계 및 설비의 안전성을 유지하는데 중요하기 때문에 이를 예방한다는 의미를 담아 소음 및 진동 예방이라고 이름을 짓게 되었다.

3. 개발 계획

먼저 함수를 사용하여 데이터의 기본 정보를 확인한다. 이후 데이터의 값이 너무 크기 때문에 head() 함수를 이용하여 처음부터 5 행까지만 출력한다. 히스토그램이나 산점도를 통해 시각적으로 데이터를 이해한다. 결측치가 존재할 경우 dropna() 함수를 사용하여 행을 지운다. 이후 표준화를 통하여 데이터를 정규화 한다. 머신러닝 모델은 성형 회귀 모델을 선택한다. 선형 회귀란 종속 변수와 독립 변수 간의 선형 관계를 모델링하는 데 사용된다. 테스트 데이터로 먼저 선형 회귀 모델을 학습하고 예측한다. 이후

결과를 확인하여 머신러닝 모델을 이해한다. 사용할 성능 지표는 평균 제곱 오차이다. 평균 제곱 오차는 예측값과 실제값 간의 제곱 오차의 평균이다. k-fold cross-validation 을 이용하여 데이터를 여러 부분으로 나누고 검증한다. 결과를 해석하여 모델을 개선한다.

4. 개발 과정

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error
```

필요한 라이브러리를 모두 임포트 한다.

```
file_path = r'C:\Users\User\Desktop\vibration.csv'
data = pd.read_csv(file_path)
```

바탕화면에 있는 csv 파일을 가져온 후 데이터를 불러온다.

```
print("데이터 요약 정보:")
print(data.info())
```

데이터 요약 정보:

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5351426 entries, 0 to 5351425
Data columns (total 6 columns):
#   Column      Dtype
---  -
0   log_id      float64
1   section_id  float64
2   device_id   object
3   device_ty   object
4   log_dt      object
5   vbrt_lvl_num float64
dtypes: float64(3), object(3)
memory usage: 245.0+ MB
None
```

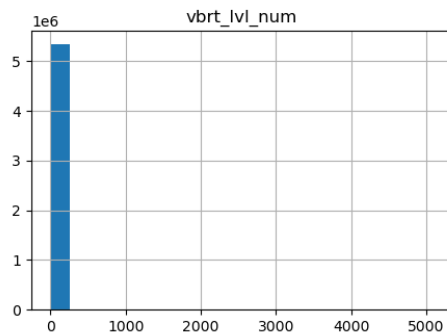
데이터의 요약된 정보를 출력한다.

```
print("데이터의 처음 5행:")
print(data.head())
```

| | log_id | ... | vbrt_lvl_num |
|---|--------|-----|--------------|
| 0 | NaN | ... | 38.374342 |
| 1 | NaN | ... | 52.125054 |
| 2 | NaN | ... | 48.495530 |
| 3 | NaN | ... | 52.235662 |
| 4 | NaN | ... | 47.875295 |

데이터의 값이 광범위하기 때문에 처음의 5 행만 출력할 수 있도록 한다.

```
data.hist(bins=20, figsize=(12, 8))
plt.show()
```



데이터를 시각화하기 위해 히스토그램을 사용하여 데이터 분포 및 특성을 파악한다.

```
data.dropna(inplace=True)
```

결측치를 처리하기 위해 dropna() 함수를 이용한다.

```
X = data[['IndependentVariable1', 'IndependentVariable2']]
y = data['DependentVariable']
```

선형 회귀 모델을 선택했기 때문에 독립 변수와 종속 변수를 x, y 로 나타낼 수 있도록 선택한다.

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

데이터를 분할한다.

```
model = LinearRegression()
model.fit(X_train, y_train)
```

모델 초기화를 먼저 진행한 후 학습을 시킨다.

```
y_pred = model.predict(X_test)
```

테스트로 먼저 데이터를 예측해본다.

```
mse = mean_squared_error(y_test, y_pred)
print("\n평균 제곱 오차 (MSE):", mse)
```

평균 제곱 오차 (mse)를 계산한다.

```
kfold = KFold(n_splits=5, shuffle=True, random_state=42)
```

```
cv_results = cross_val_score(model, X_train, y_train, cv=kfold, scoring='neg_mean_squared_error')
print("Cross-Validation MSE Scores:", -cv_results)
print("Mean MSE:", -cv_results.mean())
print("Standard Deviation of MSE:", cv_results.std())
```

k-fold 객체 생성을 하고 교차 검증을 수행한 후 결과를 출력한다.

5. 개발 후기

머신러닝 모델을 개발하는 과정이 생각보다 더 복잡하고 여러가지를 고려해야 하는 것을 알게 되었다. 실제로 산업 현장에 적용하려면 이번에 사용한 데이터셋 뿐만 아니라 기계의 품질, 유지 보수 등 고려해야 할 점이 많았다. 제일 중요한 건 데이터의 전처리인 것 같다. 모델의 예측력을 향상시킬 수 있기 때문이다. 이 머신러닝 모델은 산업 현장에서의 작업자들을 대상으로 한 모델이기 때문에 실제로 유용하게 사용될 수 있도록 개발하는 것이 가장 중요했다. 이 모델을 적용하였을 때 예상되는 결과는 작업자들의 안전을 강화하고, 환경을 개선하여 작업자들의 건강을 향상시키는 긍정적인 영향이 있을 거라고 생각한다. 앞으로 머신러닝이 산업현장에서 어떻게 쓰일 지 궁금하다.