

# 이산화탄소 예측 모델

학번: 1818073

이름: 이동혁

Github address:

[https://github.com/1818073leedonghyeok/computer\\_programming\\_homework](https://github.com/1818073leedonghyeok/computer_programming_homework)

## 1. 안전 관련 머신러닝 모델 개발의 목적

- a. 이 모델은 다양한 독립변수인 MQ1, MQ2, MQ3, MQ4, MQ5, MQ6 등을 활용하여 CO2 농도를 예측한다. 이 모델을 통해 환경 중의 안전 문제를 조기에 감지하고 예방할 수 있다. 주요 활용 대상은 CO2 농도를 측정하는 가스감지 센서를 사용하는 공간에서 사용할 수 있게끔 만든 것이다. 근로자들의 생명을 지키면서 큰 사고로 이어지지 않게 하는 것에서 가치를 창출한다.

## 2. 안전 관련 머신러닝 모델의 네이밍의 의미

- a. 이산화탄소 예측 모델은 데이터셋을 이용하여 머신러닝을 학습해서 4 가지 목표상황으로 결론이 도출된다. 1 단계 – 깨끗한 공기, 자고있거나 공부하거나 쉬고 있는 공간. 2 단계 – 고기나 파스타 요리, 야채 튀김, 방에 한 명 또는 두 명 있는 공간. 3 단계 – 창문과 문이 닫힌 방에서 짧은 시간 동안 종이와 나무를 태우는 공간. 4 단계 – 강제 공기 순환을 활성화하거나 비활성화해야 하는 공간.

## 3. 개발 계획

- a. 데이터에 대한 요약 정리 및 시각화  
데이터는 “./data/dataset.csv” 파일에서 불러온다.  
컬럼은 MQ1, MQ2, MQ3, MQ4, MQ5, MQ6, CO2 로 구성된다.
- b. 데이터 전처리 계획  
데이터에 결측치 또는 이상치가 있는지 확인하고 처리한다. 필요한 경우, 정규화를 수행한다. 범주형 데이터가 있다면 인코딩을 수행한다.
- c. 어떠한 머신러닝 모델을 사용할 것인지 (해당 머신러닝 모델의 이론 추가)

머신러닝 모델로는 RandomForestClassifier 를 선택했고 RandomForest 는 의사결정트리의 앙상블 모델로 여러 결정 트리를 조합하여 안정적이고 강력한 예측 성능을 제공한다.

d. 머신러닝 모델 예측 결과가 어떠할 지

CO2 농도를 예측하는 모델로, 각 MQ 센서 값의 조합이 주어졌을 때 CO2 농도를 예측한다.

e. 사용할 성능 지표

모델의 분류 성능을 평가하기 위해 정확도를 사용한다.

f. 성능 검증 방법 계획 등

데이터를 학습 및 테스트 세트로 분할하고, 학습 데이터로 모델을 훈련한 후 테스트 데이터를 사용하여 모델의 성능을 평가한다. SHAP 을 사용하여 모델의 해석 가능성을 향상시킬 계획이다.

#### 4. 개발 과정

라이브러리 및 모듈

```
import numpy as np
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
import shap
import matplotlib.pyplot as plt
```

데이터 파일 읽기

```
filename = "./data/dataset.csv"
column_names = ['MQ1', 'MQ2', 'MQ3', 'MQ4', 'MQ5', 'MQ6', 'CO2']
data = pd.read_csv(filename, names=column_names)
```

데이터 분리

```
Y = data['CO2']
X = data.drop('CO2', axis=1)

X_train, X_test, y_train, y_test = train_test_split(X, y, train_size=0.9, shuffle=True, random_state=1)
```

RandomForestClassifier 학습

```
model = RandomForestClassifier(random_state=1)
model.fit(X_train, y_train)
```

모델 평가

```
acc = model.score(X_test, y_test)
print("Accuracy: {:.2f}%".format(acc * 100))
```

SHAP 을 이용한 모델 해석

```
explainer = shap.TreeExplainer(model)
shap_values = explainer.shap_values(X_test)
```

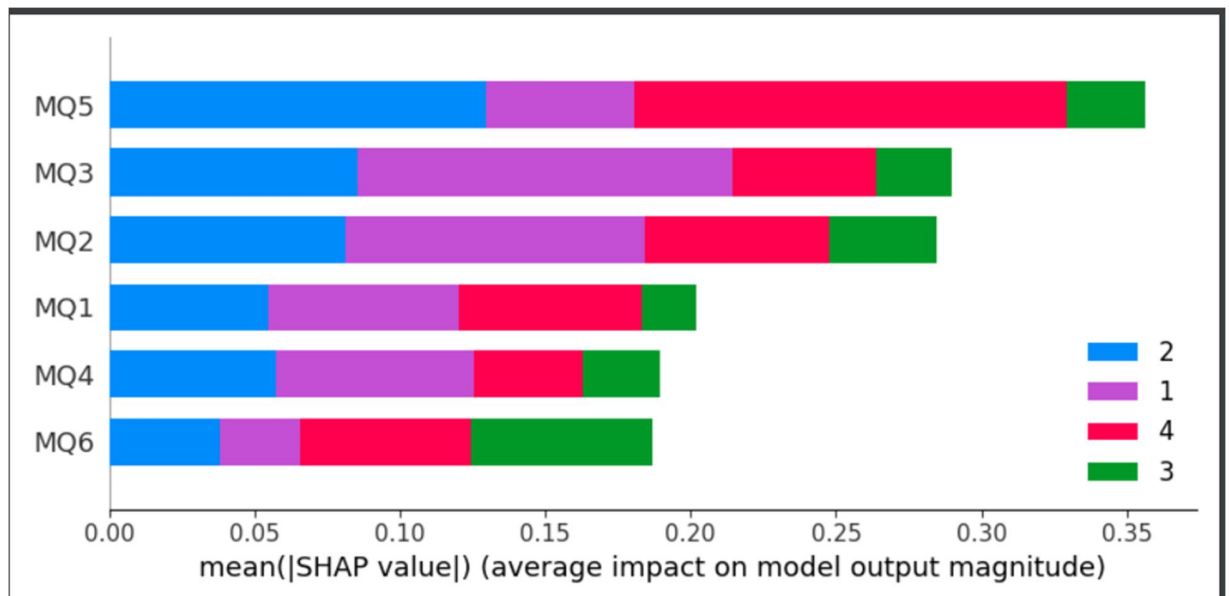
SHAP summary plot 생성

```
shap.summary_plot(shap_values, X_test, plot_type="bar", class_names=model.classes_, show=False)
```

결과 출력 및 저장

```
plt.show()
plt.savefig("./result/bar_plot.png")
```

결과 시각화



## 5. 개발 후기

머신러닝은 데이터셋의 품질과 양에 크게 좌우된다는 것을 느꼈다. SHAP 해석 도구를 사용하여 모델의 예측을 해석 가능하게 만들면서, 모델의 동작 방식에 대한 이해를 향상시킬 수 있었다. 해석 가능한 모델은 실제 응용에서 신뢰성이 높다. 결과를 시각화 함으로써 데이터의 특성 및 모델의 동작을 더 잘 이해할 수 있었다. 시각화에 대한 중요성을 느꼈다. 모델을 실제 환경에 적용할 때에는 다양한 요소들을 고려해야 한다는 것을 깨달았다. 코드에 적절히 주석을 달아 놓으며 문서화하면 다른 사람이나 나중에 내가 사용할 때에 편리할 것 같다.

