2025-06-14

2025254008 곽소영

IoT 플랫폼과 IoT 장치 설계
공기질 데이터기반 다변량 시계열 예측

충북대학교 산업인공지능학과

CONTENTS

목차.

코드 설명 기존코드

```python
# 1. 라이브러리 임포트
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

✓ 3.5s

```python
# 2. 데이터 불러오기
df = pd.read_csv("AirQualityUCI.csv", sep=';', decimal=',')
```

✓ 0.1s

```python
# 3. 마지막 빈 열 제거 + 결측 행 제거
df = df.dropna(how='all', axis=1)
df = df.dropna()
```

✓ 0.0s

```python
# 4. 날짜-시간 합치기
df["Time"] = df["Time"].astype(str).str.replace(".", ":", regex=False)
df["Datetime"] = pd.to_datetime(df["Date"] + " " + df["Time"], dayfirst=True, errors='coerce')
df = df.dropna(subset=["Datetime"])  # 날짜 파싱 실패한 행 제거
df = df.set_index("Datetime")
df = df.drop(columns=["Date", "Time"])
```

✓ 0.0s

- 날짜와 시간 데이터
  - 날짜와 시간이 별도로 저장되어 있으며, 이를 하나의 Datetime 열로 합침

```python
# 5. 정수 변환 및 결측치(-200) 제거
df = df.apply(pd.to_numeric, errors='coerce')
df[df == -200] = pd.NA
df = df.dropna()
```

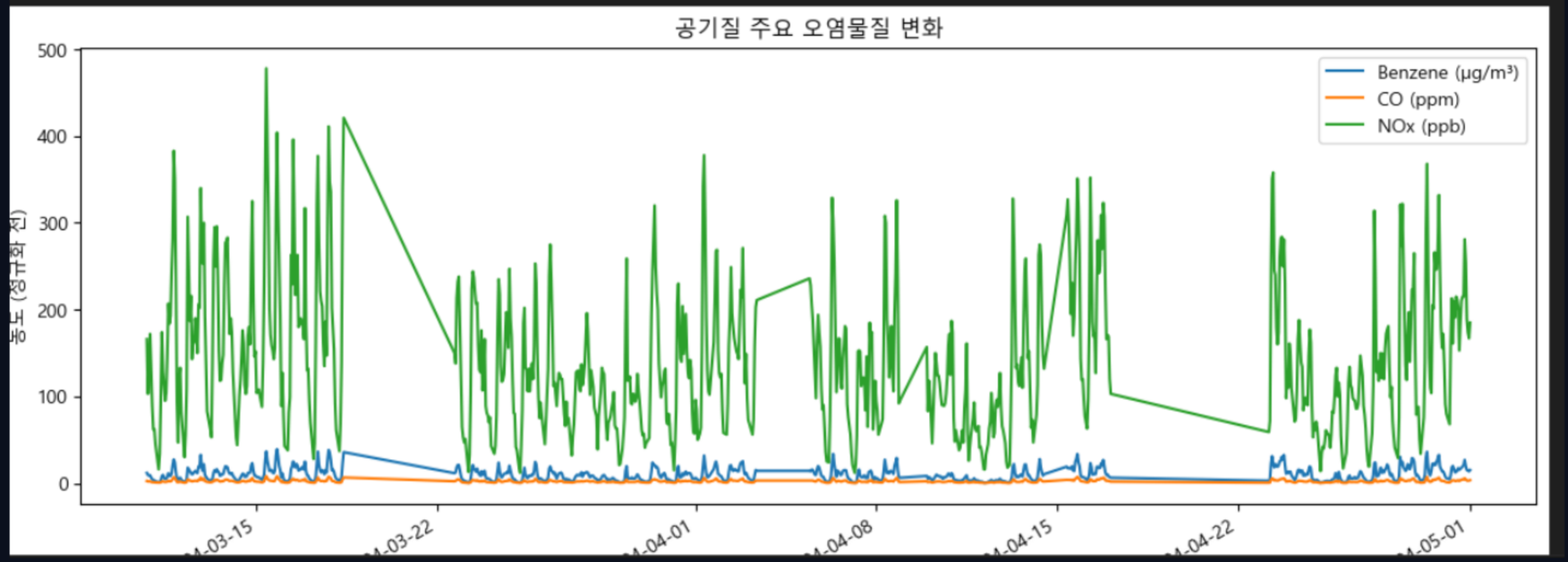
✓ 0.0s

- 결측치 처리
  - 데이터셋에는 결측치가 많으며, 일부 열은 완전히 비어 있거나 값이 -200으로 표시된 결측치가 포함되어 있음
  - 결측치를 제거한 후 데이터 정리

코드 설명 기존코드

```
plt.rcParams['font.family'] = 'Malgun Gothic'  # 윈도우 기본 한글 폰트
plt.rcParams['axes.unicode_minus'] = False  # 마이너스 부호 깨짐 방지

# 7. 시간 흐름에 따른 주요 센서 및 실제 값 시각화
plt.figure(figsize=(14, 5))
df["C6H6(GT)"].plot(label="Benzene (µg/m³)")
df["CO(GT)"].plot(label="CO (ppm)")
df["NOx(GT)"].plot(label="NOx (ppb)")
plt.legend()
plt.title("공기질 주요 오염물질 변화")
plt.xlabel("시간")
plt.ylabel("농도 (정규화 전)")
plt.show()
```

0.4s

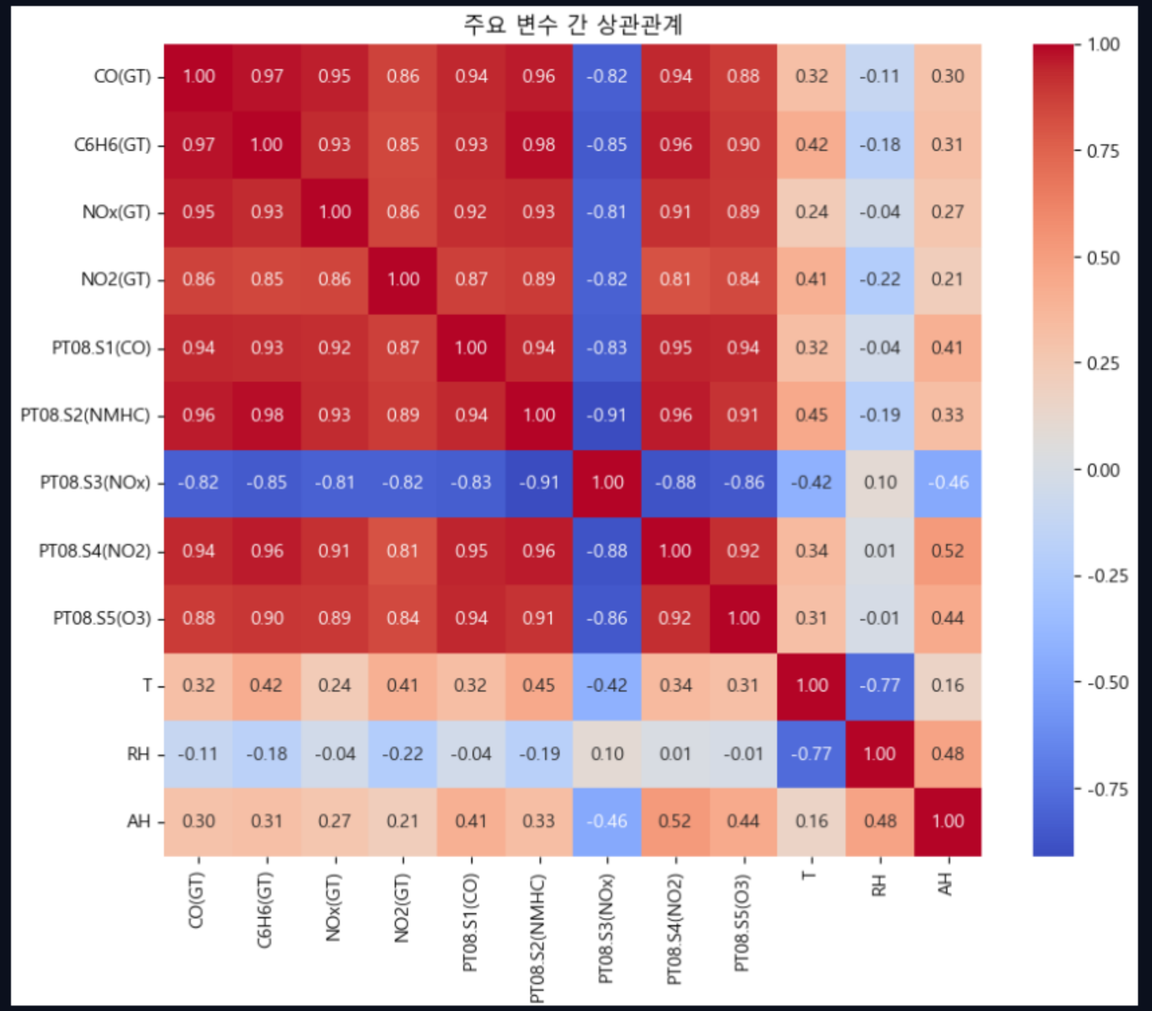


- 센서데이터
  - 주요 센서 데이터는 공기 중 오염물질 농도를 측정
    - 예를 들어, Benzene(C6H6(GT)), CO(CO(GT)), NOx(NOx(GT)) 등이 포함
  - 센서 데이터는 시간 흐름에 따라 변화하며, 이를 시각화하여 오염물질 농도의 변화를 확인

코드 설명 기존코드

```python
# 8. 주요 변수들 간의 상관관계 행렬
features_of_interest = [
    "CO(GT)", "C6H6(GT)", "NOx(GT)", "NO2(GT)",
    "PT08.S1(CO)", "PT08.S2(NMHC)", "PT08.S3(NOx)", "PT08.S4(NO2)", "PT08.S5(O3)",
    "T", "RH", "AH"
]

corr = df[features_of_interest].corr()

plt.figure(figsize=(10, 8))
sns.heatmap(corr, annot=True, cmap="coolwarm", fmt=".2f")
plt.title("주요 변수 간 상관관계")
plt.show()
```

- 상관관계
  - 주요 변수들 간의 상관관계를 계산한 결과, 일부 변수들 간에는 높은 상관관계가 존재함
    - Benzene(C6H6(GT))과 NMHC
  - 온도(T), 습도(RH), 절대 습도(AH)는 상대적으로 낮은 상관관계를 보임



코드 설명

예측 모델 개발 코드

```
# 9. 상관관계 높은 변수와 낮은 변수 그룹 분리
high_corr_features = ["C6H6(GT)", "PT08.S2(NMHC)", "PT08.S5(O3)"]
low_corr_features = ["T", "RH", "AH"]
```

✓ 0.0s

- 상관관계에 따라 상관관계가 높은 그룹, 낮은 그룹으로 분리

```
# 10. 데이터셋 구성
X_high = df[high_corr_features]
X_low = df[low_corr_features]
y = df["CO(GT)"]  # 예측 대상 변수
```

- 데이터셋 구성
  - 입력데이터로 사용될 데이터셋 구성
  - 종속변수 = 예측 대상으로 사용될 데이터 설정

```
# 11. 데이터 정규화
from sklearn.preprocessing import MinMaxScaler

scaler = MinMaxScaler()
X_high = scaler.fit_transform(X_high)
X_low = scaler.fit_transform(X_low)
y = scaler.fit_transform(y.values.reshape(-1, 1))
```

0.1s

- 데이터 정규화
  - MinMaxScaler : 데이터를 0과 1사이의 값으로 정규화
  - fit_transform 함수를 사용하여 데이터를 정규화

```
# 13. 데이터 분할``
from sklearn.model_selection import train_test_split

X_high_train, X_high_test, y_train, y_test = train_test_split(X_high, y, test_size=0.2, random_state=42)
X_low_train, X_low_test, _, _ = train_test_split(X_low, y, test_size=0.2, random_state=42)
```

0.0s

- 데이터 분할
  - train_test_split을 사용하여 학습 데이터와 테스트 데이터 분리
    - test_size = 0.2로 테스트 데이터 비율을 20%로 설정
    - random_state=42로 결과를 재현가능하도록 랜덤 시드를 설정

코드 설명

예측 모델 개발 코드

```python
# 14. 딥러닝 모델 정의 및 학습
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, LSTM, GRU, Conv1D, Flatten

def build_model(model_type, input_shape):
    model = Sequential()
    if model_type == "LSTM":
        model.add(LSTM(64, input_shape=input_shape))
    elif model_type == "GRU":
        model.add(GRU(64, input_shape=input_shape))
    elif model_type == "1D-CNN":
        model.add(Conv1D(64, kernel_size=2, activation='relu', input_shape=input_shape))
        model.add(Flatten())
    model.add(Dense(32, activation='relu'))
    model.add(Dense(1))
    model.compile(optimizer='adam', loss='mse')
    return model

models = ["GRU","LSTM", "1D-CNN"]
results = {}

for model_type in models:
    for corr_type, X_train, X_test in [("Low Corr", X_low_train, X_low_test), ("High Corr", X_high_train, X_high_test)]:
        model = build_model(model_type, (X_train.shape[1], 1))
        model.fit(X_train.reshape(-1, X_train.shape[1], 1), y_train, epochs=10, batch_size=32, verbose=0)
        mse = model.evaluate(X_test.reshape(-1, X_test.shape[1], 1), y_test, verbose=0)
        results[f"{model_type} - {corr_type}"] = mse
```

- 딥러닝 모델 정의 및 학습
  - Sequential 함수를 사용하여 딥러닝 모델을 순차적으로 구성
  - LSTM, GRU, Conv1D 모델 정의
  - Dense: 완전 연결 레이어를 추가함
  - compile: 모델 컴파일
  - fit: 모델을 학습
  - evaluate: 테스트 데이터를 사용해 모델 성능을 평가함
    - 결과는 mse 기준으로 평가됨

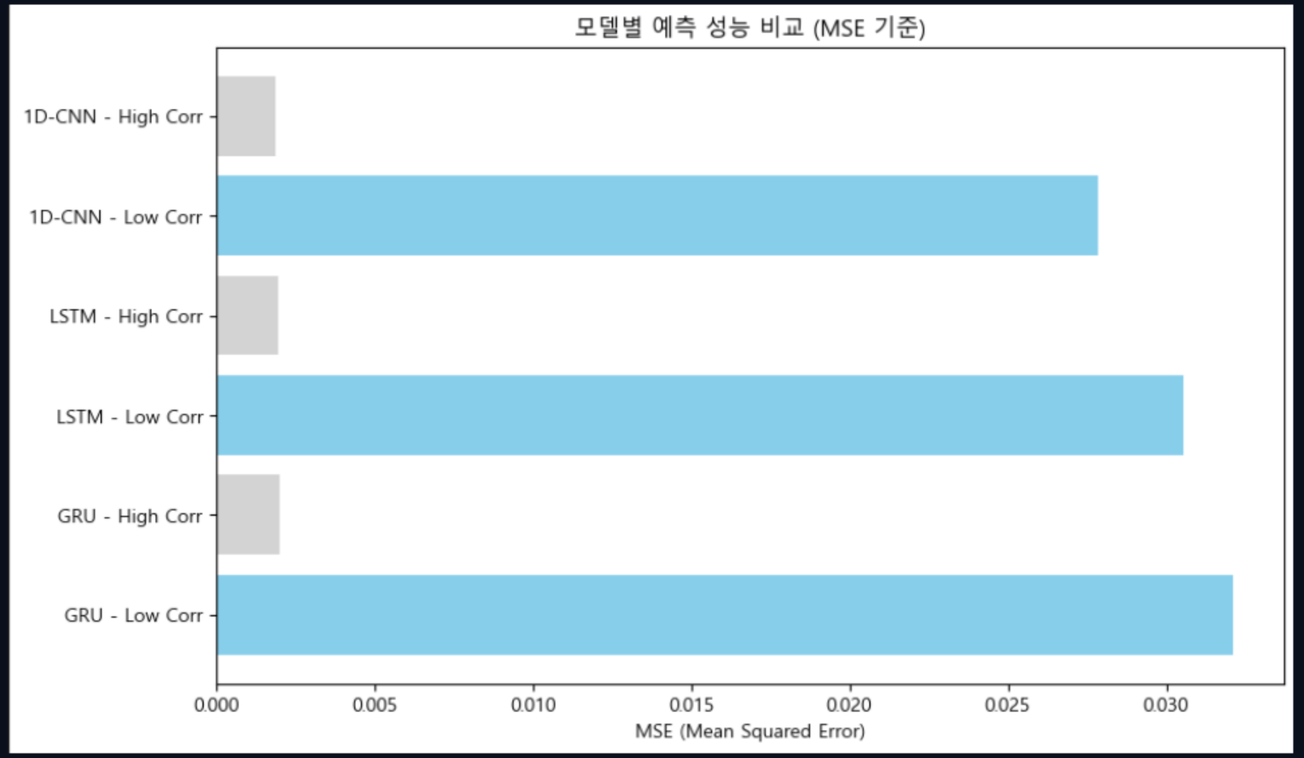
```python
# 15. 결과 시각화

plt.figure(figsize=(10, 6))
plt.barh(list(results.keys()), list(results.values()), color=['skyblue', 'lightgray'] * len(models))
plt.xlabel("MSE (Mean Squared Error)")
plt.title("모델별 예측 성능 비교 (MSE 기준)")
plt.show()
```

- 결과 시각화
  - Sequential 함수를 사용하여 딥러닝 모델을 순차적으로 구성
  - LSTM, GRU, Conv1D 모델 정의
  - Dense: 완전 연결 레이어를 추가함
  - compile: 모델 컴파일
  - fit: 모델을 학습
  - evaluate: 테스트 데이터를 사용해 모델 성능을 평가함

모델 성능 비교 및 분석

모델별 성능

- High Corr
  - GRU 〉 LSTM 〉 1D-CNN
- Row Corr
  - 1D-CNN 〉 LSTM 〉 GRU



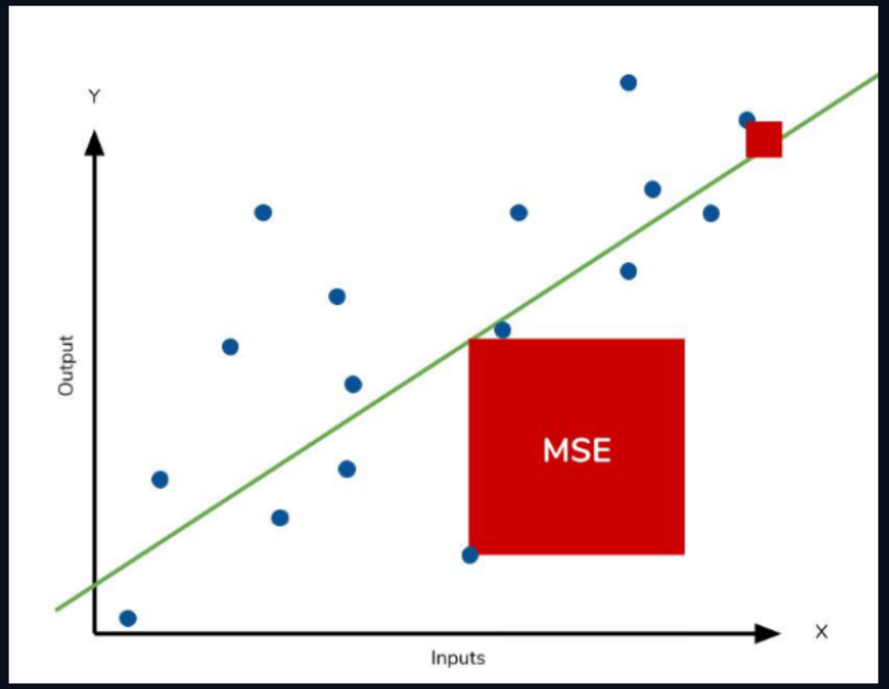
- GRU
  - GRU는 순환 신경망(RNN)의 한 종류로, 시계열 데이터 처리에 적합
  - High Corr 데이터에서 상대적으로 낮은 MSE를 기록하며, 상관관계가 높은 변수 그룹에서 더 좋은 성능을 보임
  - Low Corr 데이터에서는 성능이 다소 떨어짐

- LSTM
  - GRU와 유사하지만, 장기 의존성을 더 잘 처리할 수 있는 구조를 가짐
  - High Corr 데이터에서 GRU와 비슷한 성능을 보였으며, 시계열 데이터 예측에 강점을 보임
  - Low Corr 데이터에서도 안정적인 성능을 유지했지만, High Corr 데이터보다 성능이 약간 낮음

- 1D-CNN
  - 시계열 데이터를 처리하기 위해 컨볼루션 레이어를 사용하는 모델
  - High Corr 데이터에서 GRU와 LSTM에 비해 성능이 약간 떨어짐
  - Low Corr 데이터에서는 가장 높은 MSE를 기록하며, 시계열 데이터 처리에는 상대적으로 덜 적합한 것으로 보입니다.

모델 성능 비교 및 분석

결과값 분석



$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}(\hat{Y}_i - Y_i)^2$$

- MSE(Mean Squared Error):
  - 예측값과 실제값 간의 차이를 제곱하여 평균을 낸 값
  - 값이 작을수록 모델의 예측이 정확하다는 것을 의미함

  - High Corr 데이터에서 모든 모델이 Low Corr 데이터보다 낮은 MSE를 기록.
    -〉 상관관계가 높은 변수들이 예측에 더 유용하다는 것을 보여줌

GRU와 LSTM은 시계열 데이터 처리에 강점을 가진 모델로, High Corr 데이터에서 가장 낮은 MSE를 기록함

모델 성능 비교 및 분석

상관관계에 따른 결론

상관관계가 높은 변수 그룹(High Corr):

GRU와 LSTM이 가장 좋은 성능을 보였으며, 시계열 데이터 예측에 적합한 모델임을 확인함
1D-CNN은 상대적으로 성능이 떨어졌지만, 여전히 유용한 결과를 제공

상관관계가 낮은 변수 그룹(Low Corr):

모든 모델에서 MSE가 증가했으며, 이는 상관관계가 낮은 변수들이 예측에 덜 유용하다는 것을 나타냄
GRU와 LSTM이 여전히 상대적으로 좋은 성능을 유지했지만, 1D-CNN은 성능이 크게 떨어짐

GRU와 LSTM은 시계열 데이터 예측에 적합하며, 특히 상관관계가 높은 변수 그룹을 사용할 때 더 좋은 성능을 보였음
1D-CNN은 시계열 데이터보다는 다른 유형의 데이터에 더 적합할 수 있으므로, 데이터 특성에 따라 모델을 선택해야 함
상관관계 분석을 통해 예측에 유용한 변수 그룹을 선택하는 것이 모델 성능 향상에 중요함을 보여주는 분석임