multi_target_modeling

July 20, 2025

1 Model Building: Air Quality Forecasting

This notebook builds and evaluates machine learning models to forecast pollutant concentrations.

[6]:
```
# Imports
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.ensemble import RandomForestRegressor
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_squared_error, r2_score, mean_absolute_error
import joblib
%matplotlib inline
```

[7]:
```
# Load the processed data
df = pd.read_csv('E:/air_quality_forecasting/data/processed_air_quality.csv',␣
 ↪parse_dates=['datetime'])
df.set_index('datetime', inplace=True)
df.head()
```

[7]:
```
                     CO(GT)  PT08.S1(CO)  NMHC(GT)  C6H6(GT)  PT08.S2(NMHC)  \
datetime
2004-03-10 19:00:00     2.0       1292.0     112.0       9.4          955.0
2004-03-10 20:00:00     2.2       1402.0      88.0       9.0          939.0
2004-03-10 21:00:00     2.2       1376.0      80.0       9.2          948.0
2004-03-10 22:00:00     1.6       1272.0      51.0       6.5          836.0
2004-03-10 23:00:00     1.2       1197.0      38.0       4.7          750.0

                     NOx(GT)  PT08.S3(NOx)  NO2(GT)  PT08.S4(NO2)  \
datetime
2004-03-10 19:00:00    103.0        1174.0     92.0        1559.0
2004-03-10 20:00:00    131.0        1140.0    114.0        1555.0
2004-03-10 21:00:00    172.0        1092.0    122.0        1584.0
2004-03-10 22:00:00    131.0        1205.0    116.0        1490.0
2004-03-10 23:00:00     89.0        1337.0     96.0        1393.0

                     PT08.S5(O3)  …  weekday  month  CO(GT)_rolling3  \
datetime                          …
2004-03-10 19:00:00        972.0  …      2.0    3.0         2.300000
2004-03-10 20:00:00       1074.0  …      2.0    3.0         2.266667
2004-03-10 21:00:00       1203.0  …      2.0    3.0         2.133333
2004-03-10 22:00:00       1110.0  …      2.0    3.0         2.000000
2004-03-10 23:00:00        949.0  …      2.0    3.0         1.666667

                     NOx(GT)_rolling3  NO2(GT)_rolling3  C6H6(GT)_rolling3  \
datetime
2004-03-10 19:00:00        134.500000        102.500000          10.650000
2004-03-10 20:00:00        133.333333        106.333333          10.100000
2004-03-10 21:00:00        135.333333        109.333333           9.200000
2004-03-10 22:00:00        144.666667        117.333333           8.233333
2004-03-10 23:00:00        130.666667        111.333333           6.800000

                     CO(GT)_lag1  NOx(GT)_lag1  NO2(GT)_lag1  C6H6(GT)_lag1
datetime
2004-03-10 19:00:00          2.6         166.0         113.0           11.9
2004-03-10 20:00:00          2.0         103.0          92.0            9.4
2004-03-10 21:00:00          2.2         131.0         114.0            9.0
2004-03-10 22:00:00          2.2         172.0         122.0            9.2
2004-03-10 23:00:00          1.6         131.0         116.0            6.5

[5 rows x 25 columns]
```

1.1 Define Targets and Features

[8]:

```python
# Forecast multiple pollutants
targets = ['CO(GT)', 'NO2(GT)', 'NOx(GT)', 'C6H6(GT)']

# Loop over each target
results = {}
models = {}
from sklearn.ensemble import RandomForestRegressor
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_absolute_error, mean_squared_error, r2_score

for target in targets:
    print(f' Modeling target: {target}')
    df = pd.read_csv('E:/air_quality_forecasting/data/processed_air_quality.
 ↪csv', parse_dates=['datetime'])
    df.set_index('datetime', inplace=True)

    features = df.drop(columns=targets).select_dtypes(include=[np.number]).
 ↪columns.tolist()
    X = df[features]
    y = df[target]
    X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,␣
↪shuffle=False)

    model = RandomForestRegressor(n_estimators=100, random_state=42)
    model.fit(X_train, y_train)
    y_pred = model.predict(X_test)

    mae = mean_absolute_error(y_test, y_pred)
    rmse = mean_squared_error(y_test, y_pred) ** 0.5
    r2 = r2_score(y_test, y_pred)

    results[target] = {'MAE': mae, 'RMSE': rmse, 'R2': r2}
    models[target] = model

    import os
    os.makedirs('E:/air_quality_forecasting/models', exist_ok=True)
    import joblib
    joblib.dump(model, f'E:/air_quality_forecasting/models/{target.
↪replace("(GT)", "").lower()}_model.pkl')
```

```
Modeling target: CO(GT)
Modeling target: NO2(GT)
Modeling target: NOx(GT)
Modeling target: C6H6(GT)
```

2 Summary of Results

pd.DataFrame(results).T

[9]:

```
model = RandomForestRegressor(n_estimators=100, random_state=42)
model.fit(X_train, y_train)
y_pred = model.predict(X_test)
```

2.1 Evaluate Model

[10]:

```
print(f"MAE: {mean_absolute_error(y_test, y_pred):.3f}")
print(f"RMSE: {mean_squared_error(y_test, y_pred) ** 0.5:.3f}")
print(f"R²: {r2_score(y_test, y_pred):.3f}")
```

```
MAE: 0.081
RMSE: 0.195
R²: 0.999
```

2.2 Save Model

[11]:

```
import os
os.makedirs('E:/air_quality_forecasting/models', exist_ok=True)
joblib.dump(model, 'E:/air_quality_forecasting/models/co_forecast_model.pkl')
print("Model saved successfully  ")
```

```
Model saved successfully
```