

```

import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import mean_squared_error, r2_score
from sklearn.preprocessing import StandardScaler

# Load the dataset (replace with the actual path to your data)
data = pd.read_csv('AirQualityUCI.csv', sep=';')

# Data Preprocessing
# Handle missing values (replace NaN with the mean of the column)
data = data.fillna(data.mean())

# Convert date/time columns to a single datetime object (if applicable)
try:
    data['DateTime'] = pd.to_datetime(data['Date'] + ' ' + data['Time'],
format='%d/%m/%Y %H.%M.%S')
    data = data.drop(['Date', 'Time'], axis=1)
    data = data.set_index('DateTime')
except KeyError:
    print("Date and Time columns not found in the expected format. Proceeding without
combining them.")

# Select features and target variable
# Adjust the feature list based on your chosen dataset
features = ['CO(GT)', 'C6H6(GT)', 'T', 'RH', 'AH']
target = 'NO2(GT)'

# Ensure all selected features are present in the DataFrame
for feature in features:
    if feature not in data.columns:
        print(f"Warning: Feature '{feature}' not found in the dataset.")
        features.remove(feature)
if target not in data.columns:
    print(f"Error: Target variable '{target}' not found in the dataset. Please choose a
valid target.")
else:
    X = data[features]
    y = data[target]

    # Data Scaling (important for some algorithms)
    scaler = StandardScaler()
    X_scaled = scaler.fit_transform(X)

    # Split data into training and testing sets
    X_train, X_test, y_train, y_test = train_test_split(X_scaled, y, test_size=0.2,
random_state=42)

    # Choose a machine learning model
    # Here, we'll use RandomForestRegressor as an example
    model = RandomForestRegressor(n_estimators=100, random_state=42)

    # Train the model
    model.fit(X_train, y_train)

```

```
# Make predictions on the test set
y_pred = model.predict(X_test)

# Evaluate the model
mse = mean_squared_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)

print(f'Mean Squared Error: {mse:.2f}')
print(f'R-squared: {r2:.2f}')

# Example of making a single prediction (assuming you have new data)
new_data = pd.DataFrame({
    'CO(GT)': [2.0],
    'C6H6(GT)': [10.0],
    'T': [25.0],
    'RH': [50.0],
    'AH': [1.2]
})

# Ensure the order of columns matches the training data
new_data = new_data[features]

# Scale the new data using the *same* scaler fitted on the training data
new_data_scaled = scaler.transform(new_data)

predicted_no2 = model.predict(new_data_scaled)
print(f'\nPredicted NO2 for new data: {predicted_no2[0]:.2f}')
```