

```

# air_quality_prediction.py

import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import mean_squared_error, r2_score
import matplotlib.pyplot as plt
import seaborn as sns

# Suppress warnings for clean output
import warnings
warnings.filterwarnings("ignore")

# Load dataset
def load_data(path):
    df = pd.read_csv(path, sep=';')
    df = df.replace(-200, np.nan) # Replace missing values indicator
    df = df.dropna()             # Drop rows with missing values
    return df

# Preprocess dataset
def preprocess_data(df):
    # Drop columns not needed
    df = df.drop(columns=['Date', 'Time'])

    # Define features and target
    X = df.drop(columns=['C6H6(GT)']) # Predict Benzene concentration
    y = df['C6H6(GT)']

    return train_test_split(X, y, test_size=0.2, random_state=42)

# Train model
def train_model(X_train, y_train):
    model = RandomForestRegressor(n_estimators=100, random_state=42)
    model.fit(X_train, y_train)
    return model


# Evaluate model
def evaluate_model(model, X_test, y_test):
    y_pred = model.predict(X_test)
    rmse = np.sqrt(mean_squared_error(y_test, y_pred))
    r2 = r2_score(y_test, y_pred)
    print(f"Root Mean Squared Error: {rmse:.2f}")
    print(f"R² Score: {r2:.2f}")
    return y_pred

# Plot results
def plot_results(y_test, y_pred):
    plt.figure(figsize=(8, 5))
    sns.scatterplot(x=y_test, y=y_pred)
    plt.xlabel('Actual Benzene Concentration')
    plt.ylabel('Predicted Benzene Concentration')
    plt.title('Actual vs Predicted Air Quality (C6H6)')
    plt.grid(True)
    plt.show()

# Main execution
if __name__ == "__main__":
    file_path = "AirQualityUCI.csv" # Replace with your dataset path

    try:
        data = load_data(file_path)
        X_train, X_test, y_train, y_test = preprocess_data(data)
        model = train_model(X_train, y_train)
        y_pred = evaluate_model(model, X_test, y_test)
        plot_results(y_test, y_pred)
    except FileNotFoundError:
        print(f"File not found: {file_path}")
    except Exception as e:
        print(f"An error occurred: {str(e)}")

```

 File not found: AirQualityUCI.csv

