Source code

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

# Load your dataset
data = {
   'death': [20, 23, 13, 23, 25, 16, 17, 15, 15],
   'pm2.5': [328, 377, 309, 142, 95, 180, 277, 82, 70],
   'pm10': [421, 473, 403, 209, 132, 223, 329, 109, 107],
   'co': [4.5, 5.1, 4.5, 2.7, 1.7, 2.8, 4.0, 1.6, 1.6],
   'no2': [114, 120, 122, 87, 52, 76, 84, 53, 57],
   'so2': [328, 358, 329, 193, 80, 168, 250, 129, 99],
   'o3': [14, 14, 19, 23, 62, 27, 18, 34, 29],
   'temp': [-1.6, -1.4, 0.8, 4.2, 0.9, 1.6, -0.2, -1.9, -2.2],
   'rain': [0, 0, 0, 0, 0, 1.5, 0.5, 0, 0],
   'wind': [4.5, 3.8, 4.7, 7.4, 18.2, 3.9, 5.4, 6.2, 5.6],
   'pressure': [1026.8, 1026.4, 1023.9, 1021.4, 1024.6, 1023.7, 1023.4, 1023.1,
1024.5],
   'aqi': [379, 418, 359, 182, 125, 230, 338, 108, 95]
}

df = pd.DataFrame(data)

# ------------------ Feature Engineering & Modeling ------------------

features = ['pm2.5', 'pm10', 'co', 'no2', 'so2', 'o3', 'temp', 'rain', 'wind',
'pressure']
X = df[features]
y = df['aqi']

scaler = StandardScaler()
```

```
X_scaled = scaler.fit_transform(X)

X_train, X_test, y_train, y_test = train_test_split(X_scaled, y, test_size=0.2,
random_state=42)

model = RandomForestRegressor(random_state=42)
model.fit(X_train, y_train)

y_pred = model.predict(X_test)
rmse = mean_squared_error(y_test, y_pred) ** 0.5 # Calculate RMSE without
'squared' parameter
r2 = r2_score(y_test, y_pred)

print("\n✅ Model Evaluation:")
print("RMSE:", round(rmse, 2))
print("R² Score:", round(r2, 2))

# ------------------ Visualizations ------------------

# Heatmap (Correlation Matrix)
plt.figure(figsize=(10, 6))
sns.heatmap(df[features + ['aqi']].corr(), annot=True, cmap='coolwarm')
plt.title("Heatmap of Features and AQI")
plt.show()

# KDE Plot (KTE Diagram)
plt.figure(figsize=(6, 4))
sns.kdeplot(df['aqi'], fill=True)
plt.title("KDE of AQI")
plt.xlabel("AQI")
plt.show()

# Box Plot of AQI
plt.figure(figsize=(6, 4))
sns.boxplot(x=df['aqi'])
plt.title("Boxplot of AQI")
plt.show()
```

```
# Line Plot for AQI and Pollutants
plt.figure(figsize=(10, 5))
plt.plot(df['pm2.5'], label='PM2.5', marker='o')
plt.plot(df['pm10'], label='PM10', marker='o')
plt.plot(df['aqi'], label='AQI', marker='o')
plt.title("PM2.5, PM10, and AQI Over Days")
plt.xlabel("Day Index")
plt.ylabel("Values")
plt.legend()
plt.show()

# Prediction Output
output = pd.DataFrame({'Actual AQI': y_test.values, 'Predicted AQI': y_pred})
print("\n🎯 Prediction Output:\n")
print(output)
```