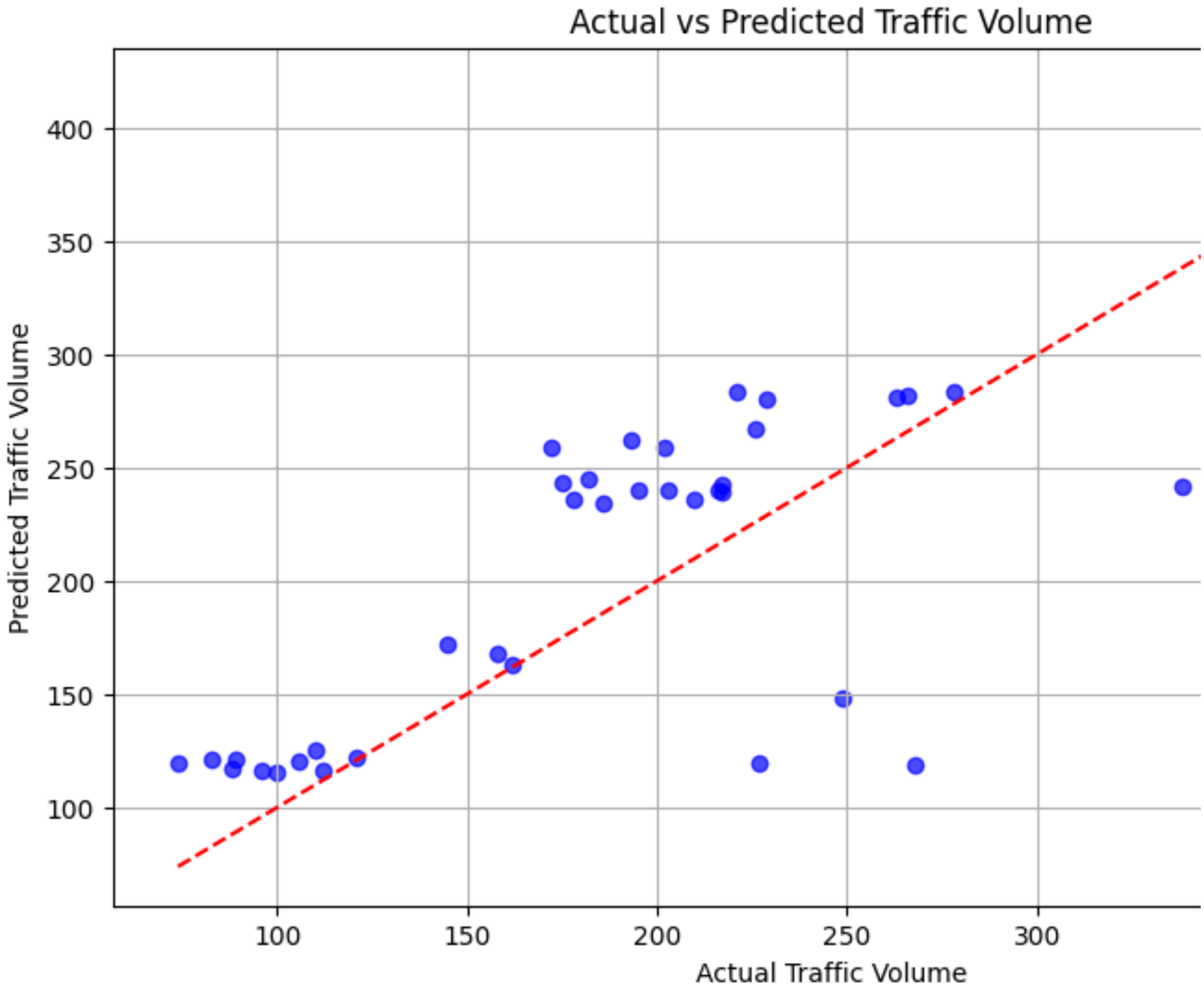
```
# Step 1: Import Libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_absolute_error, r2_score

# Step 2: Create Pseudo Dataset
np.random.seed(42)
n_samples = 200

data = pd.DataFrame({
    'hour': np.random.randint(0, 24, size=n_samples),
    'temperature_C': np.random.uniform(15, 35, size=n_samples),
    'weather': np.random.choice(['Clear', 'Cloudy', 'Rainy'], size=n_samples),
    'is_weekend': np.random.choice([0, 1], size=n_samples),
    'traffic_volume': np.nan  # to be filled based on logic
})

# Add logic to create traffic volume (simulate real-world patterns)
for i in range(n_samples):
    base = 200
    if 7 <= data.loc[i, 'hour'] <= 9 or 17 <= data.loc[i, 'hour'] <= 19:  # rush hour
        base += 150
    if data.loc[i, 'weather'] == 'Rainy':
        base += 50
    if data.loc[i, 'is_weekend'] == 1:
        base -= 100
    fluctuation = np.random.randint(-30, 30)
    data.loc[i, 'traffic_volume'] = base + fluctuation

# Step 3: Preprocess
data['weather_encoded'] = data['weather'].map({'Clear': 0, 'Cloudy': 1, 'Rainy': 2})
X = data[['hour', 'temperature_C', 'weather_encoded', 'is_weekend']]
y = data['traffic_volume']

# Step 4: Train/Test Split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Step 5: Model Training
model = LinearRegression()
model.fit(X_train, y_train)

# Step 6: Prediction & Evaluation
y_pred = model.predict(X_test)
mae = mean_absolute_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)
```

```
# Step 7: Output Results
print("Mean Absolute Error:", round(mae, 2))
print("R-squared Score:", round(r2, 2))

# Step 8: Visualization
plt.figure(figsize=(10,6))
plt.scatter(y_test, y_pred, alpha=0.7, color='blue')
plt.plot([y_test.min(), y_test.max()], [y_test.min(), y_test.max()], 'r--')
plt.xlabel('Actual Traffic Volume')
plt.ylabel('Predicted Traffic Volume')
plt.title('Actual vs Predicted Traffic Volume')
plt.grid(True)
plt.show()
```

```
Mean Absolute Error: 50.21
R-squared Score: 0.44
```


Actual vs Predicted Traffic Volume
Predicted Traffic Volume
Actual Traffic Volume