```python
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import mean_squared_error, r2_scor
# Load Dataset
print("\nTrafficTelligence - Advanced Traffic Volume Estimation ")
df = pd.read_csv("traffic volume.csv")
df['datetime'] = pd.to_datetime(df['date'] + ' ' + df['Time'], dayfirst=True)
df['hour'] = df['datetime'].dt.hour
df['day_of_week'] = df['datetime'].dt.dayofweek
df['is_weekend'] = df['day_of_week'].apply(lambda x: 1 if x >= 5 else 0)
df['holiday'] = df['holiday'].astype('category').cat.codes
df['weather'] = df['weather'].astype('category').cat.codes
#Feature and Target Selection
features = ['holiday', 'temp', 'rain', 'snow', 'weather', 'hour', 'day_of_week',
'is_weekend']
target = 'traffic_volume'
X = df[features]
y = df[target]
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
random_state=42)
# Train Model
model = RandomForestRegressor(n_estimators=100, random_state=42)
model.fit(X_train, y_train)
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"\nRMSE: {rmse:.2f}")
print(f"R² Score: {r2:.2f}")
importances = model.feature_importances_
feature_importance = pd.DataFrame({
    'Feature': features,
    'Importance': importances
}).sort_values(by='Importance', ascending=False)
print("\nFeature Importance:\n", feature_importance)
plt.figure(figsize=(10, 6))
sns.barplot(data=feature_importance, x='Importance', y='Feature',
palette='viridis')
plt.title("Feature Importance in Traffic Volume Prediction")
plt.tight_layout()
```

```
plt.savefig("feature_importance.png")
plt.show()
```