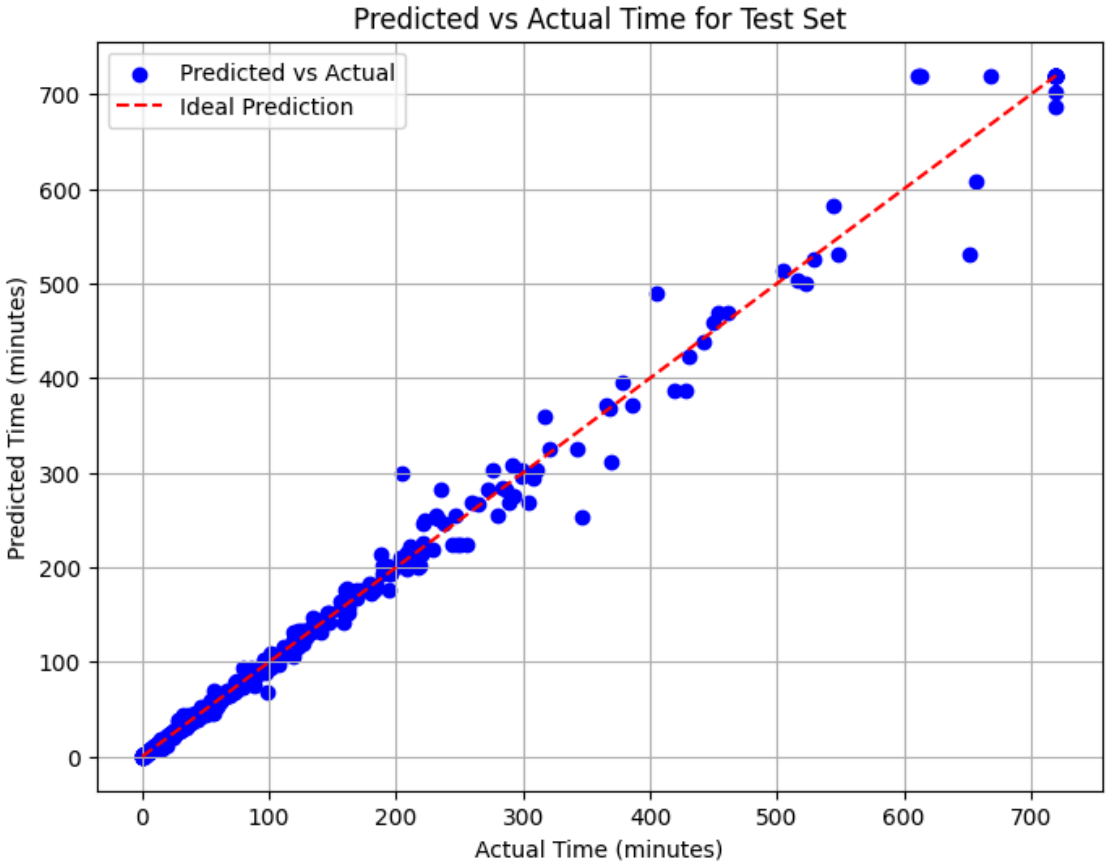
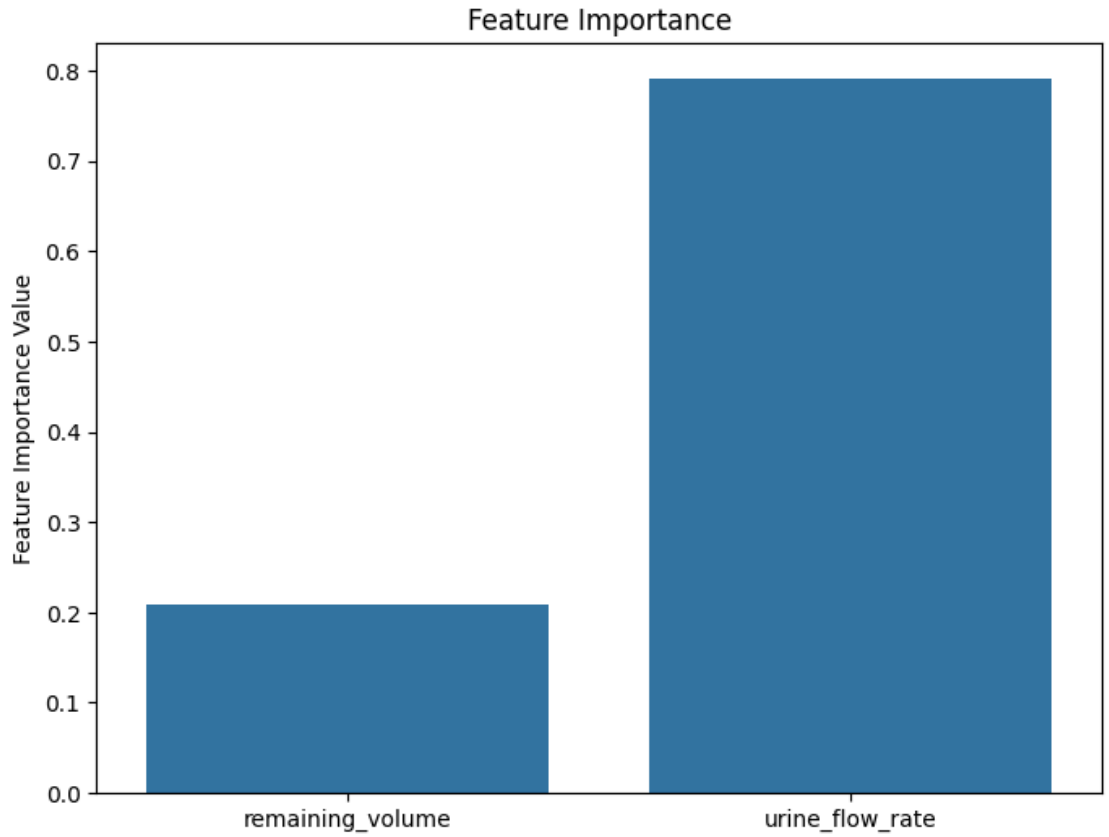
In [6]:

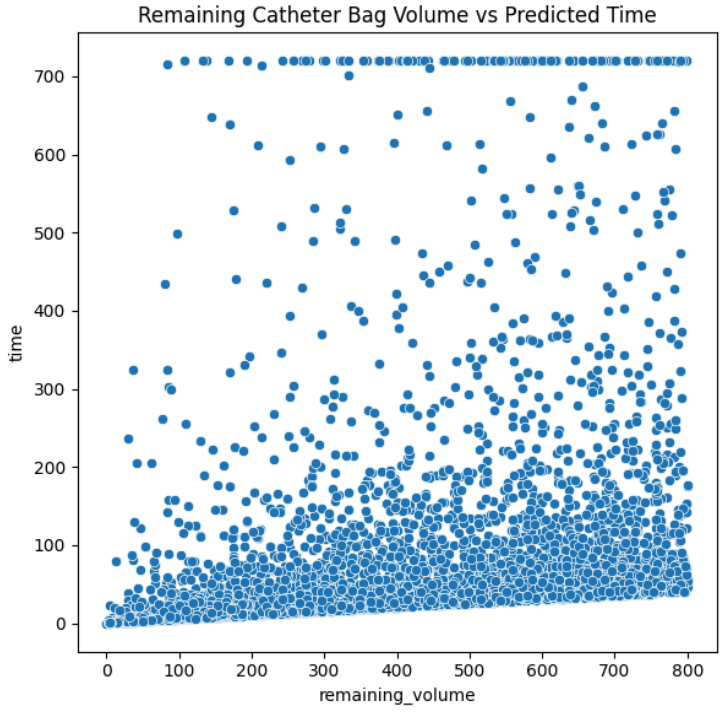
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeRegressor
from sklearn.metrics import mean_absolute_error, mean_squared_error, r2_score
from sklearn.preprocessing import StandardScaler
import joblib

df = pd.read_csv('DATASET.csv')
print(df.head())

X = df[['remaining_volume', 'urine_flow_rate']]
y = df['time']

X_train, X_temp, y_train, y_temp = train_test_split(X, y, test_size=0.4, random_state=42)
X_val, X_test, y_val, y_test = train_test_split(X_temp, y_temp, test_size=0.5, random_state=42)

scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_val_scaled = scaler.transform(X_val)
X_test_scaled = scaler.transform(X_test)

model = DecisionTreeRegressor(random_state=42)
model.fit(X_train_scaled, y_train)

y_val_pred = model.predict(X_val_scaled)
y_test_pred = model.predict(X_test_scaled)

mae_val = mean_absolute_error(y_val, y_val_pred)
mse_val = mean_squared_error(y_val, y_val_pred)
r2_val = r2_score(y_val, y_val_pred)

mae_test = mean_absolute_error(y_test, y_test_pred)
mse_test = mean_squared_error(y_test, y_test_pred)
r2_test = r2_score(y_test, y_test_pred)

print(f"Validation MAE: {mae_val}")
print(f"Validation MSE: {mse_val}")
print(f"Validation R²: {r2_val}")
print(f"Test MAE: {mae_test}")
print(f"Test MSE: {mse_test}")
print(f"Test R²: {r2_test}")

plt.figure(figsize=(8, 6))
plt.scatter(y_test, y_test_pred, color='blue', label='Predicted vs Actual')
plt.plot([0, 720], [0, 720], color='red', linestyle='--', label='Ideal Prediction')
plt.title('Predicted vs Actual Time for Test Set')
plt.xlabel('Actual Time (minutes)')
plt.ylabel('Predicted Time (minutes)')
plt.legend()
plt.grid(True)
plt.show()

plt.figure(figsize=(8, 6))
sns.barplot(x=['remaining_volume', 'urine_flow_rate'], y=model.feature_importances_)
plt.title('Feature Importance')
plt.ylabel('Feature Importance Value')
plt.show()

plt.figure(figsize=(12, 6))
plt.subplot(1, 2, 1)
sns.scatterplot(data=df, x='remaining_volume', y='time')
plt.title('Remaining Catheter Bag Volume vs Predicted Time')

plt.subplot(1, 2, 2)
sns.scatterplot(data=df, x='urine_flow_rate', y='time')
plt.title('Urine Flow Rate vs Predicted Time')

plt.tight_layout()
plt.show()
```

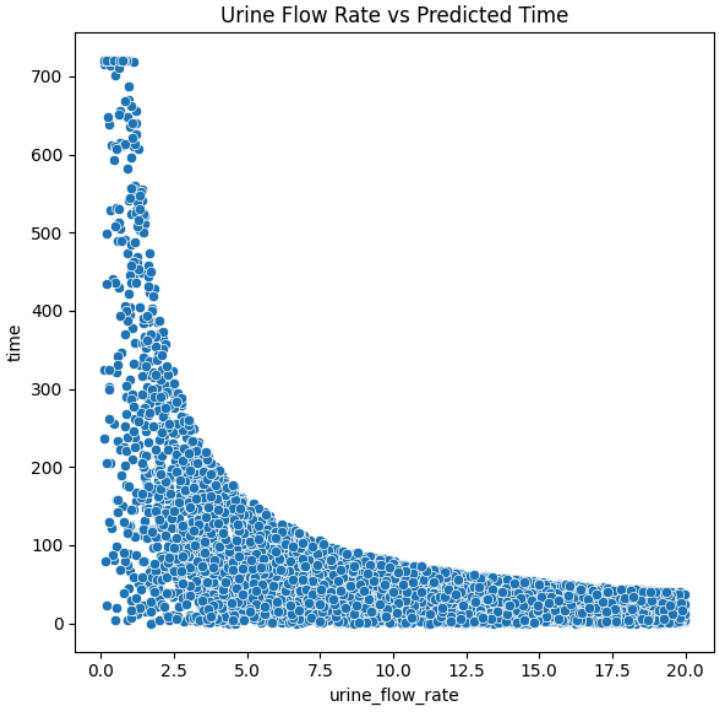
```
joblib.dump(model, 'decision_tree.pkl')
joblib.dump(scaler, 'scaler.pkl')
```

```
   remaining_volume  urine_flow_rate        time
0               386         2.511605  153.686614
1               390         2.707014  144.070191
2               497         1.134827  437.952320
3               515        12.556304   41.015253
4               207         9.497465   21.795289
Validation MAE: 2.987035859145459
Validation MSE: 80.29824142236836
Validation R²: 0.9950878396940115
Test MAE: 3.2062668623504242
Test MSE: 113.15698282641783
Test R²: 0.9947328536897645
```









Out[6]: ['scaler.pkl']

In [ ]: