

```
In [109... 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, cross_val_score
from sklearn.preprocessing import StandardScaler
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, mean_absolute_error, r2_score
import pickle
```

```
In [119... file_path = "final.csv"
dataset = pd.read_csv(file_path)
dataset.head
```

```
Out[119... <bound method NDFrame.head of      Unnamed: 0  urine_output  urine_flow_rate  catheter_bag_volume \
0           0      97.131536      98.872967           0.000000
1           1      99.346882     100.763500           0.266756
2           2     101.285898      96.603298           0.533511
3           3     118.937229     121.835150           0.800267
4           4      85.055654      77.605240           1.067022
...         ...         ...         ...         ...
2995        2995     119.699002     113.404649           798.932978
2996        2996      84.333825      86.112174           799.199733
2997        2997     115.705107     119.858386           799.466489
2998        2998      77.013908      76.538159           799.733244
2999        2999     120.628318     126.819734           800.000000

      remaining_catheter_bag_volume      time
0           800.000000      720.000000
1           799.839968      720.000000
2           799.679936      720.000000
3           799.519904      720.000000
4           799.359872      719.039680
...         ...         ...
2995         320.704141       0.960320
2996         320.544109       0.720240
2997         320.384077       0.480160
2998         320.224045       0.240080
2999         320.064013       0.000000

[3000 rows x 6 columns]>
```

```
In [112... scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)
```

```
In [113... X_train, X_temp, y_train, y_temp = train_test_split(X_scaled, y, test_size=0.3) # 70% training
X_val, X_test, y_val, y_test = train_test_split(X_temp, y_temp, test_size=0.5)

model = LinearRegression()
model.fit(X_train, y_train)
```

```
Out[113... LinearRegression
LinearRegression()
```

```
In [114... y_val_pred = model.predict(X_val)
```

```
In [115... val_mse = mean_squared_error(y_val, y_val_pred)
val_mae = mean_absolute_error(y_val, y_val_pred)
val_r2 = r2_score(y_val, y_val_pred)

print("\n VALIDATION RESULTS:")
print(f"Validation MSE: {val_mse}")
print(f"Validation MAE: {val_mae}")
print(f"Validation R²: {val_r2}")
```

```
VALIDATION RESULTS:
Validation MSE: 0.00012808537471389615
Validation MAE: 0.000533541033262469
Validation R²: 0.9999999970514053
```

```
In [116... y_test_pred = model.predict(X_test)
```

```

test_mse = mean_squared_error(y_test, y_test_pred)
test_mae = mean_absolute_error(y_test, y_test_pred)
test_r2 = r2_score(y_test, y_test_pred)

print("\n TEST RESULTS:")
print(f"Test MSE: {test_mse}")
print(f"Test MAE: {test_mae}")
print(f"Test R²: {test_r2}")

```

TEST RESULTS:

Test MSE: 0.0016651098831032413

Test MAE: 0.0026675853050511238

Test R²: 0.9999999608804873

```

In [117... with open("model_1.pkl", "wb") as file:
            pickle.dump(model, file)
with open("scaler.pkl", "wb") as file:
            pickle.dump(scaler, file)

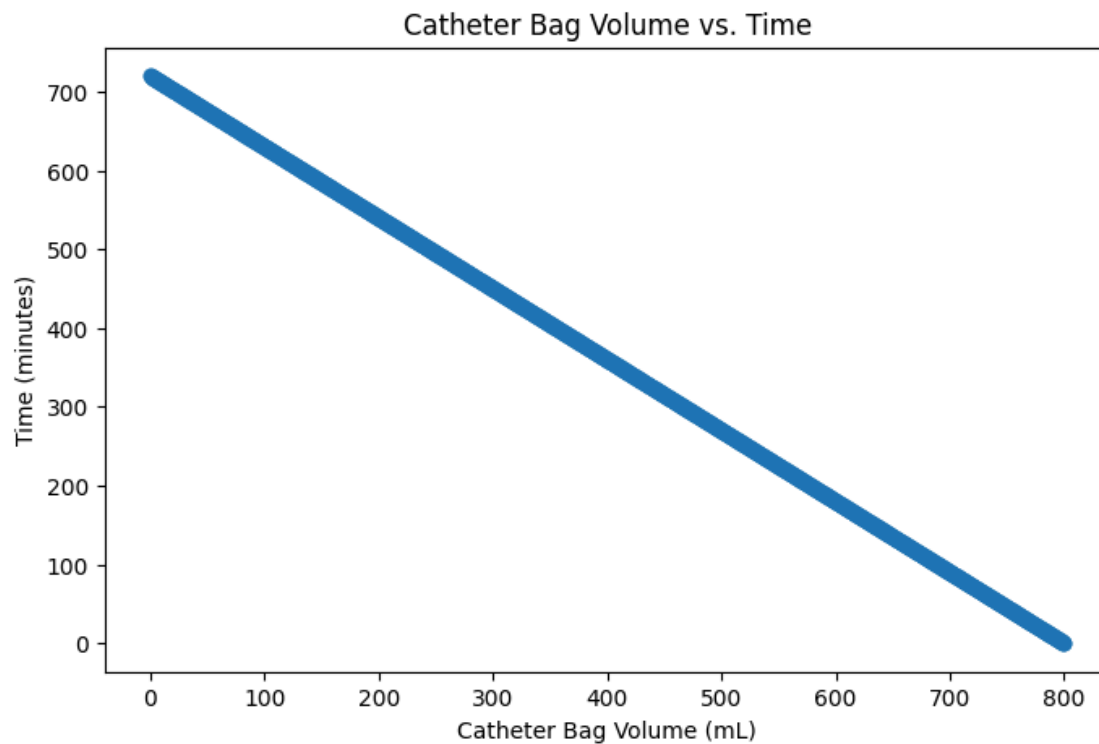
```

```

In [118... import matplotlib.pyplot as plt

plt.figure(figsize=(8,5))
plt.scatter(dataset["catheter_bag_volume"], dataset["time"], alpha=0.5)
plt.xlabel("Catheter Bag Volume (mL)")
plt.ylabel("Time (minutes)")
plt.title("Catheter Bag Volume vs. Time")
plt.show()

```



In []: