

```
In [1]: 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 [2]: file_path = "intellicathdefense.csv"
dataset = pd.read_csv(file_path)
```

```
In [3]: X = dataset[["urine_output", "urine_flow_rate", "catheter_bag_volume", "remaining_catheter_bag_volume"]]
y = dataset["time"]
```

```
In [4]: dataset.loc[(dataset["urine_output"] == 0) & (dataset["catheter_bag_volume"] == 0), "time"] = 720
dataset.loc[dataset["catheter_bag_volume"] >= 800, "time"] = 0 # 00:00
```

```
In [5]: scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)
```

```
In [6]: 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[6]: ▼ LinearRegression ⓘ ⓘ
LinearRegression()
```

```
In [7]: y_val_pred = model.predict(X_val)
```

```
In [8]: 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: 0.08984752168072684
Validation MAE: 0.2630999156603548
Validation R²: 0.9999978833595717
```

```
In [9]: y_test_pred = model.predict(X_test)

# 🚀 Compute Test Metrics
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: 0.08435391722923574
Test MAE: 0.254210969770018
Test R²: 0.9999979312377328
```

```
In [10]: with open("model_1.pkl", "wb") as file:
    pickle.dump(model, file)
with open("scaler.pkl", "wb") as file:
    pickle.dump(scaler, file)
```

```
In [ ]:
```