```
# ✅ Step 1: Import required libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns


from google.colab import files
uploaded = files.upload()

import pandas as pd
df = pd.read_excel("cnc_machine_dataset.xlsx")  # Update if your file name is different
print(df.head())

# Step 2: Drop missing values (if any)
df.dropna(inplace=True)
```

Choose Files CNCDATA…EANED.xlsx

- **CNCDATASETCLEANED.xlsx**(application/vnd.openxmlformats-officedocument.spreadsheetml.sheet) - 83714 bytes, last modified: 5/21/2025 - 100% done

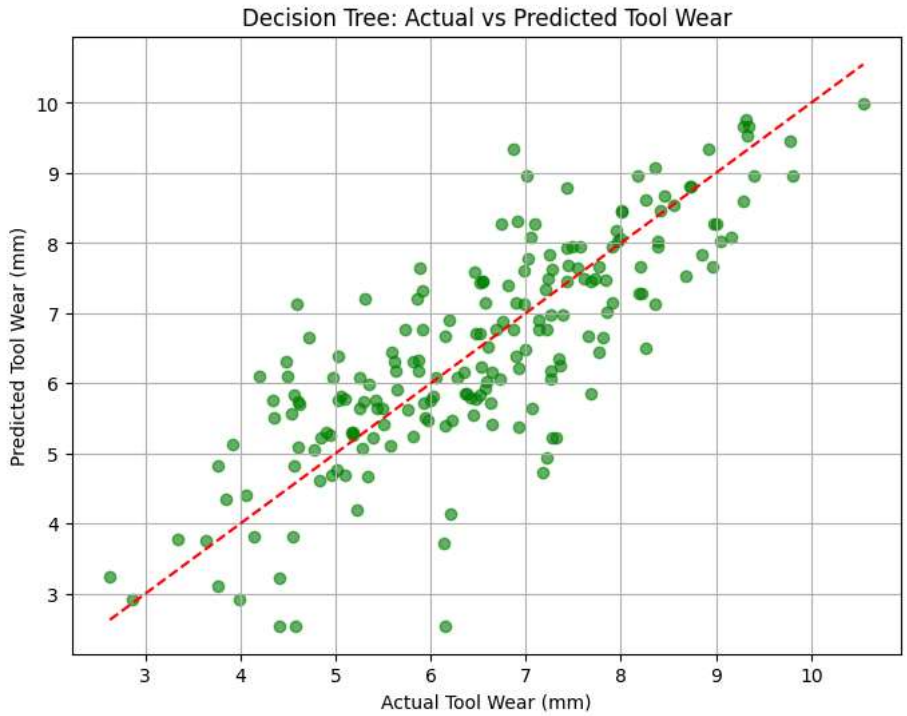
```
Saving CNCDATASETCLEANED.xlsx to CNCDATASETCLEANED.xlsx
   Spindle_Speed_RPM  Feed_Rate_mm_per_min  Cutting_Time_min  \
0        2019.865985             53.530215         23.487456
1        2383.022737            292.653737         56.035909
2        2788.425012            407.195063         43.194719
3        2645.392440            169.802729         55.766475
4        1718.035727            321.481180         48.021512

   Tool_Temperature_C Material_Type  Depth_of_Cut_mm  Coolant_Used  \
0          383.708508        Copper         4.021960             1
1           84.545206      Titanium         1.000947             0
2          791.206493      Titanium         1.826171             1
3          515.174557        Copper         0.716828             0
4          318.052034      Aluminum         2.751092             0

   Tool_Age_min  Tool_Wear_mm
0    207.855540      0.430250
1    458.142936      0.446965
2     36.999327      0.055088
3    991.951941      0.482956
4    408.862520      0.173470
```

Double-click (or enter) to edit

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.tree import DecisionTreeRegressor
from sklearn.metrics import r2_score
import matplotlib.pyplot as plt

# Step 1: Load the data
df = pd.read_excel("CNCDATASETCLEANED.xlsx")

# Step 2: Features and Target
X = df.drop(columns='Tool_Wear_mm')
y = df['Tool_Wear_mm']

# Step 3: Train-Test Split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Step 4: Models
models = {
    "Linear Regression": LinearRegression(),
    "Decision Tree": DecisionTreeRegressor(random_state=42)
}

# Step 5: Train and Evaluate
for name, model in models.items():
    model.fit(X_train, y_train)
    y_pred = model.predict(X_test)
    r2 = r2_score(y_test, y_pred)
    print(f"✅ {name} R² Score: {r2:.3f}")

# Step 6: Plot for Decision Tree
best_model = DecisionTreeRegressor(random_state=42)
best_model.fit(X_train, y_train)
```

```
y_pred = best_model.predict(X_test)

plt.figure(figsize=(8,6))
plt.scatter(y_test, y_pred, alpha=0.6, color='green')
plt.plot([y_test.min(), y_test.max()], [y_test.min(), y_test.max()], 'r--')
plt.xlabel("Actual Tool Wear (mm)")
plt.ylabel("Predicted Tool Wear (mm)")
plt.title("Decision Tree: Actual vs Predicted Tool Wear")
plt.grid(True)
plt.show()
```

```
✅ Linear Regression R² Score: 0.891
✅ Decision Tree R² Score: 0.605
```



Start coding or generate with AI.

Start coding or generate with AI.

Start coding or generate with AI.

Start coding or generate with AI.