Decision Trees are a popular supervised learning algorithm used for both classification and regression tasks. They build a tree-like model of decisions and their possible consequences based on features of the data.

Here's a step-by-step guide to implementing Decision Trees using Python and the popular machine learning library, scikit-learn:

Step 1: Import the required libraries

```
from sklearn.tree import DecisionTreeClassifier, DecisionTreeRegressor
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score, mean_squared_error
```

Step 2: Prepare the dataset

```
# Assuming X and y are NumPy arrays or pandas DataFrames
X = ...  # Input features
y = ...  # Target variable
```

Step 3: Split the dataset into training and testing sets

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

Step 4: Create and train the Decision Tree model

```
model = DecisionTreeClassifier()  # For classification tasks
# model = DecisionTreeRegressor()  # For regression tasks
model.fit(X_train, y_train)
```

Step 5: Make predictions

```
y_pred = model.predict(X_test)
```

Your URL should open in a new window. If it does not, make sure that pop ups are not blocked and reopen the link. ×

Step 6: Evaluate the model

```
accuracy = accuracy_score(y_test, y_pred)  # For classification tasks
# mse = mean_squared_error(y_test, y_pred)  # For regression tasks
```

Step 7: Print the evaluation results

```
print(f"Accuracy: {accuracy}")  # For classification tasks
# print(f"Mean Squared Error: {mse}")  # For regression tasks
```

Your URL should open in a new window. If it does not, make sure that pop ups are not blocked and reopen the link.