### **Project 8: Machine Learning Model**

#### **Objective:**

Train a simple machine learning model on a dataset (e.g., the Iris dataset) using Scikit-Learn and evaluate its performance. Create visualizations to explain model predictions.

### **Instructions**

#### **Step 1: Set Up the Environment**

1. Create a new Python file called ml\_model.py.

Install the required libraries:  
Copy code  
pip install scikit-learn matplotlib pandas

#### **Step 2: Import Necessary Libraries**

Import the required libraries at the top of your script:  
python  
Copy code  
from sklearn.datasets import load\_iris

from sklearn.model\_selection import train\_test\_split

from sklearn.tree import DecisionTreeClassifier

from sklearn.metrics import accuracy\_score

import matplotlib.pyplot as plt

import pandas as pd

#### **Step 3: Load and Explore the Dataset**

Load the Iris dataset:  
python  
Copy code  
iris = load\_iris()

X = iris.data

y = iris.target

Convert it into a Pandas DataFrame for easy viewing:  
python  
Copy code  
df = pd.DataFrame(X, columns=iris.feature\_names)

df['target'] = y

print(df.head())

#### **Step 4: Split the Dataset**

Split the dataset into training and testing sets:  
python  
Copy code  
X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

#### **Step 5: Train a Decision Tree Classifier**

Initialize and train the model:  
python  
Copy code  
clf = DecisionTreeClassifier()

clf.fit(X\_train, y\_train)

#### **Step 6: Evaluate the Model**

Predict on the test set and calculate the accuracy:  
python  
Copy code  
y\_pred = clf.predict(X\_test)

accuracy = accuracy\_score(y\_test, y\_pred)

print(f"Model Accuracy: {accuracy \* 100:.2f}%")

#### **Step 7: Visualize the Decision Boundaries**

Create a scatter plot to visualize the decision boundaries:  
python  
Copy code  
plt.figure(figsize=(10, 6))

plt.scatter(X\_test[:, 0], X\_test[:, 1], c=y\_pred, cmap='viridis')

plt.title("Decision Tree Classifier Boundaries")

plt.xlabel(iris.feature\_names[0])

plt.ylabel(iris.feature\_names[1])

plt.show()

#### **Step 8: Run and Test**

* Test the model on different datasets and explore the impact of parameters on model performance.