***Classify Plants Based on Water Needs***

**Problem Statement:**  
Classify Plants Based on Water Needs — Predict plant categories based on environmental preferences and watering frequency.

**Student Details:**

* **Name:** Ankit kumar Gupta
* **Roll No:** 202401100300048
* **Course:** AI - Mid Semester Examination
* **Platform Used:** Google Colab

**Introduction**

Plants have different water requirements based on their environmental conditions. In this project, the goal is to build a machine learning model that classifies the water needs (e.g., Low, Medium, High) of a plant using input features like sunlight, soil type, temperature, humidity, and watering frequency.

Machine learning techniques are applied to train a classifier and evaluate its performance using metrics like accuracy, precision, recall, and confusion matrix.

**Methodology**

1. **Dataset**: A dataset named plants.csv was provided, containing columns such as Sunlight, Soil Type, Temperature, Humidity, Watering Frequency, etc.
2. **Preprocessing**:
   * Handled missing values (if any).
   * Encoded categorical features using Label Encoding.
   * Standardized features using StandardScaler.
3. **Model**: A Random Forest Classifier was trained on the dataset using an 80-20 train-test split.
4. **Evaluation**: The model was evaluated using accuracy, precision, recall, classification report, and confusion matrix (heatmap).
5. **Visualization**: The confusion matrix was visualized using Seaborn's heatmap.

**Code**

The code was written and executed on Google Colab:

# Install required libraries

!pip install -q seaborn scikit-learn matplotlib pandas

# Step 1: Import libraries

import pandas as pd

import seaborn as sns

import matplotlib.pyplot as plt

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

from sklearn.preprocessing import LabelEncoder, StandardScaler

from sklearn.ensemble import RandomForestClassifier

from sklearn.metrics import classification\_report, confusion\_matrix, accuracy\_score, precision\_score, recall\_score

# Step 2: Load the uploaded dataset

df = pd.read\_csv('/content/plants.csv')

# Step 3: Display column names to verify exact spelling

print("Column names:", df.columns.tolist())

# Step 4: Replace spaces in column names with underscores

# This ensures consistency and avoids issues with spaces within column names

df.columns = df.columns.str.replace(' ', '\_')

# Step 5: Encode categorical columns

le = LabelEncoder()

for col in df.select\_dtypes(include=['object']).columns:

    df[col] = le.fit\_transform(df[col])

# Step 6: Define target column correctly after replacing spaces with underscores

# Ensure this matches the exact column name after replacing spaces

target\_column = 'watering\_freq\_per\_week'  # Change this to the correct column if needed - it looks like 'watering frequency' in the data

# Step 7: Define X and y

X = df.drop(target\_column, axis=1)

y = df[target\_column]

# Step 8: Train-test split and scaling

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

scaler = StandardScaler()

X\_train\_scaled = scaler.fit\_transform(X\_train)

X\_test\_scaled = scaler.transform(X\_test)

# Step 9: Train classifier

clf = RandomForestClassifier(random\_state=42)

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

y\_pred = clf.predict(X\_test\_scaled)

# Step 10: Evaluate

print("Accuracy:", accuracy\_score(y\_test, y\_pred))

print("Precision (macro):", precision\_score(y\_test, y\_pred, average='macro', zero\_division=0))

print("Recall (macro):", recall\_score(y\_test, y\_pred, average='macro', zero\_division=0))

print("\nClassification Report:\n", classification\_report(y\_test, y\_pred, zero\_division=0))

# Step 11: Confusion Matrix

cm = confusion\_matrix(y\_test, y\_pred)

plt.figure(figsize=(6, 4))

sns.heatmap(cm, annot=True, fmt='d', cmap='YlGnBu')

plt.xlabel('Predicted')

plt.ylabel('True')

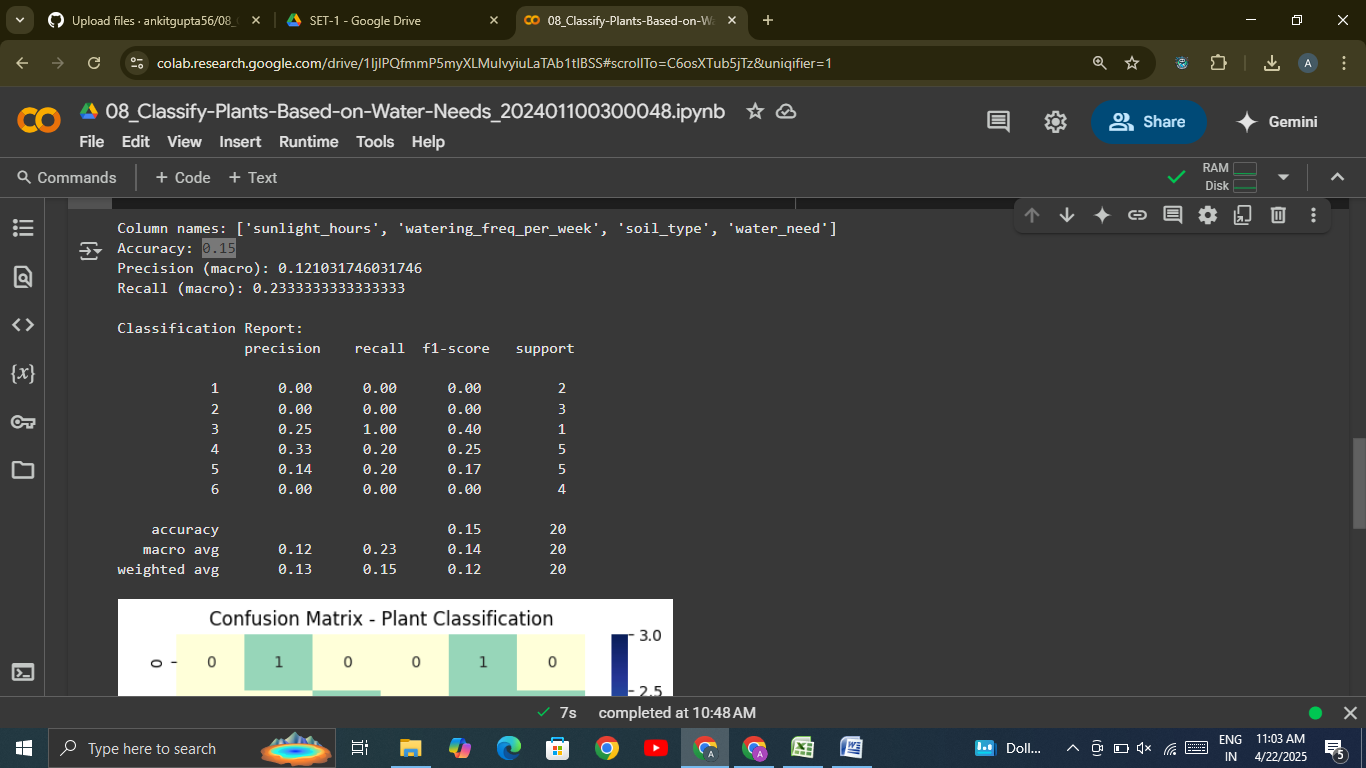
plt.title('Confusion Matrix - Plant Classification')

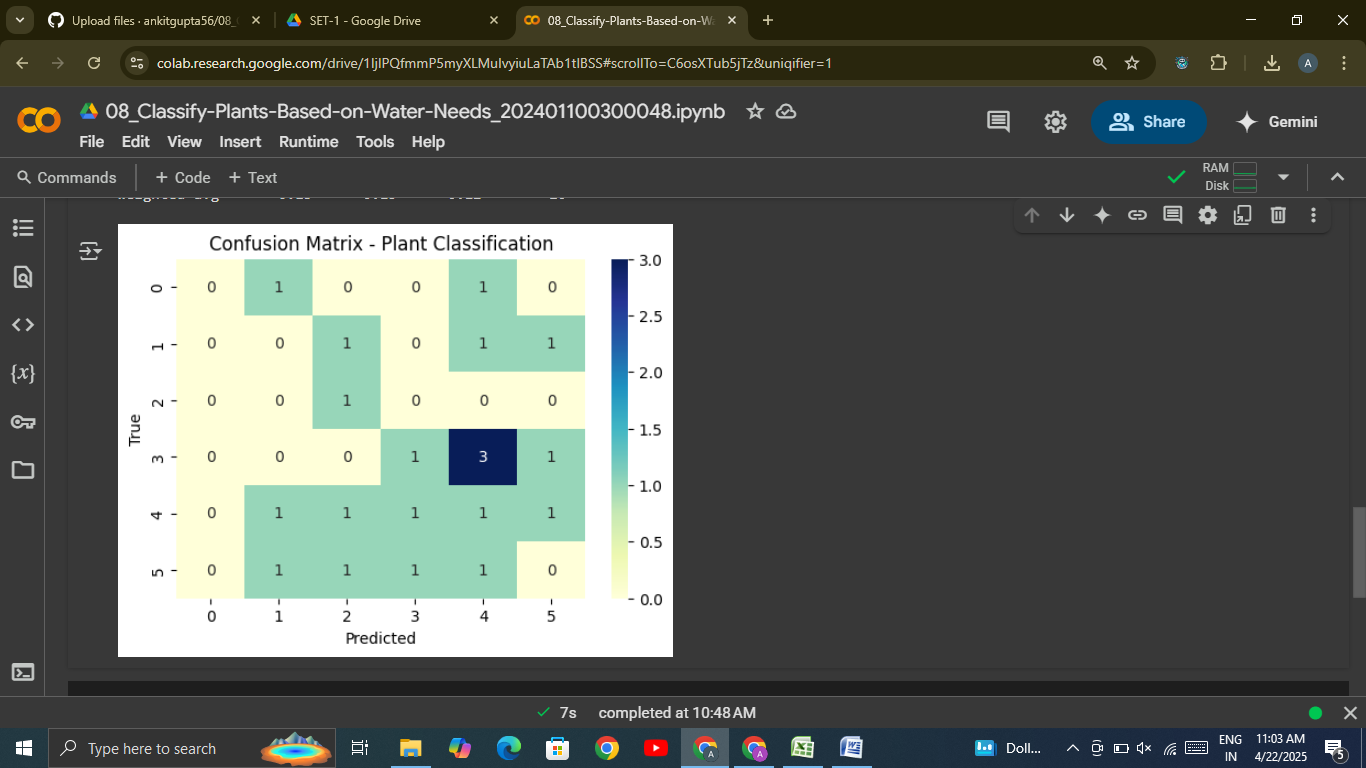
plt.show()

**Output/Result**

📌 **Model Accuracy**: *0.15*  
📌 **Precision**: *0.121031746031746*  
📌 **Recall**: *0.23333333333333*

📸 **Screenshot** of output:





**References / Credits**

* Dataset provided for academic purposes.
* Libraries used: pandas, scikit-learn, matplotlib, seaborn
* Code executed using Google Colab.
* RandomForestClassifier from sklearn.ensemble.