Assessment Report

on

"Classify Plants Based on Water Needs" submitted as partial fulfilment for the award of

BACHELOR OF TECHNOLOGY

DEGREE

SESSION 2024-25 in

CSE(AI)

By

Name : Shashwat Gupta
Roll Number : 202401100300228
Section: D

Under the supervision of

"MR.BIKKI"

KIET Group of Institutions,
Ghaziabad

JULY, 2025

1. Introduction

Water is essential for plant survival, but different plants have varying water requirements. Knowing how much water a plant needs can help in water conservation and plant health. This project uses machine learning to classify plants into categories (Low, Medium, or High water needs) based on environmental inputs such as:

- Sunlight exposure
- Soil type
- Watering frequency

By training a model on labeled plant data, we aim to create a system that can make predictions about unknown plants' water needs.

---

2. Problem Statement

To predict the water requirement category of a plant (Low, Medium, High) using its environmental preferences such as sunlight exposure, soil type, and watering frequency.

---

3. Objectives

- **Automate** the classification process using a data-driven approach.
- **Assist** gardeners, farmers, and plant enthusiasts in understanding plant care needs.
- **Optimize** water usage by recommending accurate watering practices.
- **Leverage machine learning** to uncover hidden patterns in plant-environment relationships.

---

4. Methodology

1. **Data Collection**:
   A CSV file containing plant features (sunlight, soil, watering frequency) and their water needs was used.

2. **Data Preprocessing**:
   - Missing values were checked. ○ Categorical values were converted into numerical labels using Label Encoding. 3. **Model Selection**:
   - A **Random Forest Classifier** was used for classification because of its high accuracy and robustness.
4. **Training and Testing**:
   - The dataset was split into training (80%) and testing (20%) sets.
   - The model was trained on the training set and tested on the testing set.
5. **Evaluation**:
   - Accuracy and classification report were used to evaluate model performance.

---

5. Code

```
# Step 1: Import Libraries import pandas as pd import numpy as np
from sklearn.model_selection import train_test_split from
sklearn.preprocessing import LabelEncoder from sklearn.ensemble
import RandomForestClassifier from sklearn.metrics import
classification_report, accuracy_score
# Step 2: Load the Uploaded CSV File df =
pd.read_csv('/content/drive/MyDrive/plants.csv')
```

```
# Step 3: View First Few Rows print("Dataset Preview:\n",
df.head()) # Step 4: Check for Missing Values
print("\nMissing Values:\n", df.isnull().sum())
```

```
# Step 5: Encode Categorical Columns
label_encoders = {} for column in
df.columns:     if df[column].dtype
== 'object':
        le = LabelEncoder()         df[column] =
le.fit_transform(df[column])
label_encoders[column] = le # Step 6: Split
Features and Target
# Assuming the last column is the target (Plant Category)
X = df.iloc[:, :-1] y = df.iloc[:, -1]
# Step 7: Train-Test Split
X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.2, random_state=42)
# Step 8: Train a Random Forest Classifier
model = RandomForestClassifier(n_estimators=100, random_state=42)
model.fit(X_train, y_train)
```

```
# Step 9: Make Predictions y_pred
= model.predict(X_test)

# Step 10: Evaluate the Model print("\nClassification
Report:")
print(classification_report(y_test, y_pred))
 print("Accuracy Score:", accuracy_score(y_test,
y_pred))
```

6. Output

```
Dataset Preview:
    sunlight_hours  watering_freq_per_week soil_type water_need
0         7.789136                       6     sandy        low
1        11.668008                       4     loamy       high
2         9.003943                       2     loamy        low
3        11.148641                       5     loamy        low
4         9.384333                       2      clay       high
```

```
Missing Values:
 sunlight_hours            0
watering_freq_per_week     0
soil_type                  0
water_need                 0
dtype: int64
```

```
RandomForestClassifier
RandomForestClassifier(random_state=42)
```

```
Classification Report:
              precision    recall  f1-score   support

           0       0.29      0.40      0.33         5
           1       0.38      0.27      0.32        11
           2       0.20      0.25      0.22         4

    accuracy                           0.30        20
   macro avg       0.29      0.31      0.29        20
weighted avg       0.32      0.30      0.30        20

Accuracy Score: 0.3
```

7. References

- scikit-learn documentation
- pandas documentation
- Seaborn visualization library