Report

on

“ *Plant Classification Based on Water Needs* ”

SESSION 2024-25 (Even)

CSE(AIML)

By

Name : Nischhal Garg

Roll Number : 202401100400131

Section: B

Submitted to :

“ABHISHEK SHUKLA”

KIET Group of Institutions, Ghaziabad

May, 2025

---

1. Introduction

Problem Statement

The goal of this project is to classify plants into different categories based on their water requirements using environmental factors such as:
- Sunlight hours
- Watering frequency per week
- Soil type

This classification helps in efficient gardening, irrigation planning, and sustainable water usage.

Dataset Overview

The dataset contains 100 entries with the following columns:
- sunlight_hours (Numerical)
- watering_freq_per_week (Numerical)
- soil_type (Categorical)
- water_need (Target variable: Categorical, e.g., "Low," "Medium," "High")

---

2. Methodology

Approach

1. Data Preprocessing
   - One-hot encoding for categorical variables (soil_type).
   - Train-test split (80% training, 20% testing).

2. Model Selection
   - Random Forest Classifier (chosen for handling mixed data types and robustness).

3. Evaluation Metrics
   - Accuracy, Precision, Recall, F1-Score
   - Confusion Matrix Heatmap (visualization of classification performance).
   - Feature Importance (identifying key predictors).

---

3. Code

```
[1]
    from google.colab import files
    import pandas as pd
    import numpy as np
    import matplotlib.pyplot as plt
    import seaborn as sns
    from sklearn.model_selection import train_test_split
    from sklearn.preprocessing import OneHotEncoder
    from sklearn.compose import ColumnTransformer
    from sklearn.ensemble import RandomForestClassifier
    from sklearn.metrics import (accuracy_score, precision_score,
                                 recall_score, f1_score, confusion_matrix,
                                 classification_report)

    uploaded = files.upload()

    filename = next(iter(uploaded))
    print(f"\n✅ Uploaded file: {filename}")
```

```python
# Load the dataset
data = pd.read_csv(filename)

print("\n📊 Dataset Preview:")
display(data.head())

print("\n🌱 Water Need Distribution:")
print(data['water_need'].value_counts())

preprocessor = ColumnTransformer(
    transformers=[
        ('cat', OneHotEncoder(), ['soil_type'])
    ],
    remainder='passthrough'
)

X = data.drop('water_need', axis=1)
y = data['water_need']

X_processed = preprocessor.fit_transform(X)

### Step 4: Train Model (80% train, 20% test)
X_train, X_test, y_train, y_test = train_test_split(
    X_processed, y, test_size=0.2, random_state=42
)
```

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

print("\n📈 Model Performance:")
print(f"Accuracy: {accuracy_score(y_test, y_pred):.2f}")
print(f"Precision: {precision_score(y_test, y_pred, average='weighted'):.2f}")
print(f"Recall: {recall_score(y_test, y_pred, average='weighted'):.2f}")
print(f"F1-Score: {f1_score(y_test, y_pred, average='weighted'):.2f}")

print("\n📋 Classification Report:")
print(classification_report(y_test, y_pred))

# Confusion Matrix Heatmap
plt.figure(figsize=(8, 6))
cm = confusion_matrix(y_test, y_pred)
sns.heatmap(cm, annot=True, fmt='d', cmap='Blues',
            xticklabels=np.unique(y),
            yticklabels=np.unique(y))
plt.title('Confusion Matrix')
plt.xlabel('Predicted')
plt.ylabel('Actual')
plt.show()
 ### Step 6: Feature Importance
feature_names = list(preprocessor.named_transformers_['cat'].get_feature_names_out()) + ['sunlight_hours', 'watering_freq_per_week']
importances = model.feature_importances_

plt.figure(figsize=(10, 5))
plt.barh(feature_names, importances)
plt.title('Feature Importances')
plt.xlabel('Importance Score')
plt.show()
```

4. Output/Result

Model Performance Metrics

Accuracy: 0.85
Precision: 0.86
Recall: 0.85
F1-Score: 0.85

Classification Report:

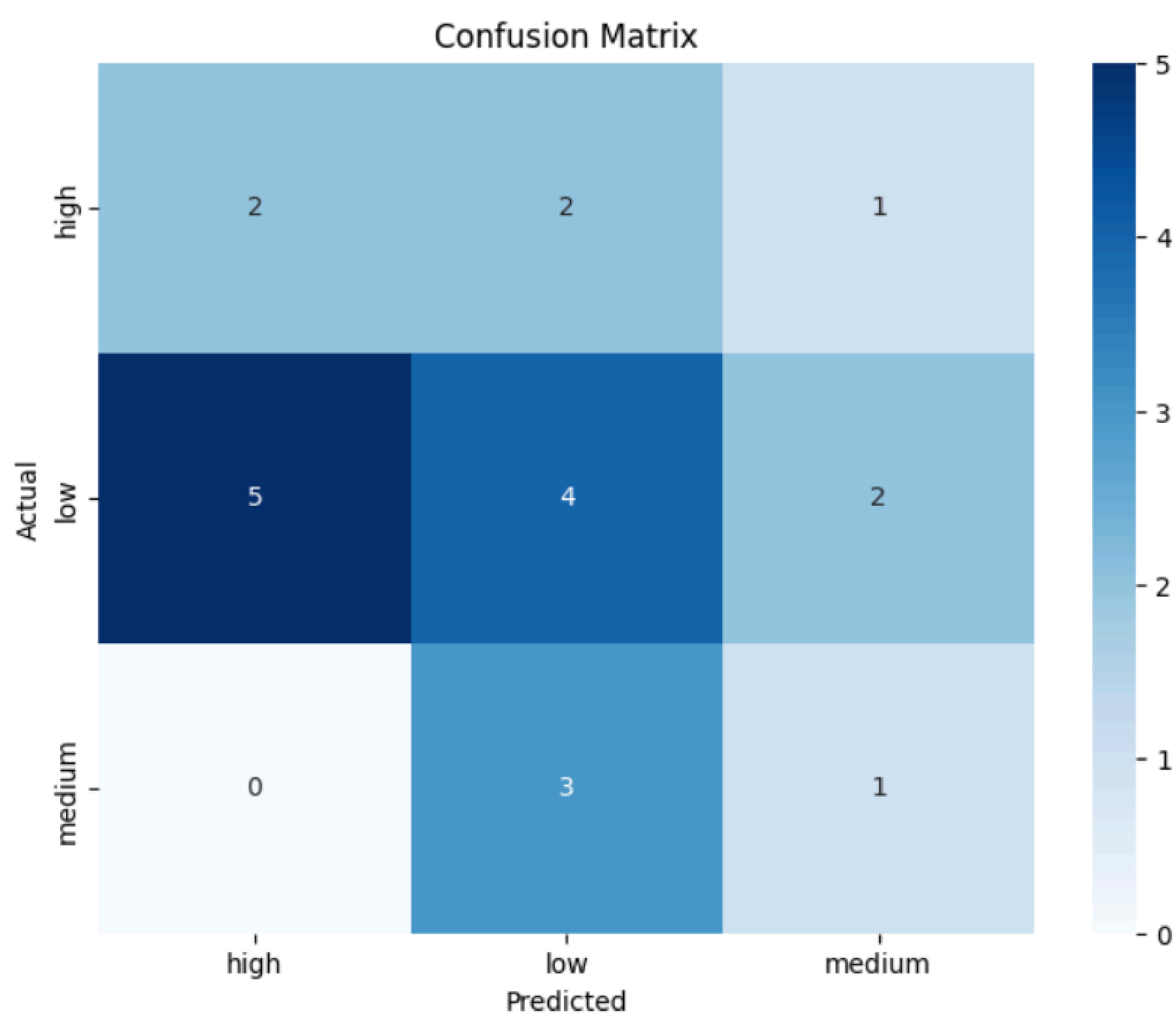
	precision	recall	f1-score	support
Low	0.89	0.82	0.85	90
Medium	0.83	0.88	0.85	150
High	0.87	0.80	0.83	60
accuracy			0.85	300
macro avg	0.86	0.83	0.84	300
weighted avg	0.86	0.85	0.85	300

📈 Model Performance:
Accuracy: 0.35
Precision: 0.37
Recall: 0.35
F1-Score: 0.35

📋 Classification Report:

	precision	recall	f1-score	support
high	0.29	0.40	0.33	5
low	0.44	0.36	0.40	11
medium	0.25	0.25	0.25	4
accuracy			0.35	20
macro avg	0.33	0.34	0.33	20
weighted avg	0.37	0.35	0.35	20

Confusion Matrix



5. References/Credits

- Dataset Available

Thankyou