

Assessment Report

On

“***Classify Plants Based on Water Needs”***

submitted as partial fulfillment for the award of

BACHELOR OF TECHNOLOGY

DEGREE

SESSION 2024-25

In

CSE(AI)

By

Name : Ananya Sharma

Roll Number : 202401100300038

Section: A

**Under the supervision of**

***“BIKKI GUPTA”***

KIET Group of Institutions, Ghaziabad

May, 2025

1. **Introduction**

This project aims to classify plants based on their water needs, using environmental factors like sunlight hours, watering frequency, and soil type. The primary objective is to predict whether a plant requires low, medium, or high water based on these features.

A machine learning model, specifically **Gradient Boosting**, is employed for classification, and class imbalance is addressed using **SMOTE** (Synthetic Minority Over-sampling Technique). The model's performance is evaluated using accuracy, precision, recall, and F1-score to ensure reliable predictions. This project is useful for optimizing plant care by accurately determining water requirements based on environmental conditions, which can improve gardening and agricultural practices.

1. **Problem Statement**

Classify plants into low, medium, or high water‑need categories using sunlight hours, watering frequency, and soil type. Balance classes with SMOTE and employ a Gradient Boosting classifier to automate accurate water‑requirement predictions for optimized plant care.

**3. Objective**

* Preprocess and encode environmental features (sunlight hours, watering frequency, soil type) for modeling.
* Balance the target classes using SMOTE to mitigate dataset imbalance.
* Train a Gradient Boosting classifier to predict plant water‑need categories (low, medium, high).
* Evaluate the model’s performance using accuracy, precision, recall, and F1‑score.
* Provide reliable, data‑driven recommendations to optimize plant watering practices.

# 4. Methodology

1. **Data Preprocessing:** The dataset is loaded and cleaned by encoding categorical variables (soil\_type and water\_need) into numeric values using Label Encoding. This step ensures the data is suitable for machine learning models.
2. **Handling Class Imbalance:** To address the class imbalance in the target variable (water\_need), SMOTE (Synthetic Minority Over-sampling Technique) is applied to generate synthetic samples for minority classes, improving model performance**.**
3. **Feature Selection:** The relevant features, including sunlight\_hours, watering\_freq\_per\_week, and soil\_type\_encoded, are selected to train the model.
4. **Model Training:** A Gradient Boosting Classifier is used to train the model on the resampled data. The model is then evaluated on an 80-20 train-test split.
5. **Evaluation:** The model's performance is assessed using accuracy, precision, recall, and F1-score to ensure reliable classification of plant water needs.

# Code

import pandas as pd

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

df.head()

df.info()

print("\n")

print("Look at unique values for each column")

for col in df.columns:

    print(f"{col}: {df[col].unique()}")

print(df['water\_need'].value\_counts())

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

from sklearn.preprocessing import LabelEncoder

from sklearn.ensemble import GradientBoostingClassifier

from imblearn.over\_sampling import SMOTE

from sklearn.metrics import accuracy\_score, classification\_report

data = pd.read\_csv('plants.csv')

soil\_encoder = LabelEncoder()

data['soil\_type\_encoded'] = soil\_encoder.fit\_transform(data['soil\_type'])

target\_encoder = LabelEncoder()

data['water\_need\_encoded'] = target\_encoder.fit\_transform(data['water\_need'])

X = data[['sunlight\_hours', 'watering\_freq\_per\_week', 'soil\_type\_encoded']]

y = data['water\_need\_encoded']

smote = SMOTE(random\_state=42)

X\_resampled, y\_resampled = smote.fit\_resample(X, y)

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

model = GradientBoostingClassifier(random\_state=42)

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

y\_pred = model.predict(X\_test)

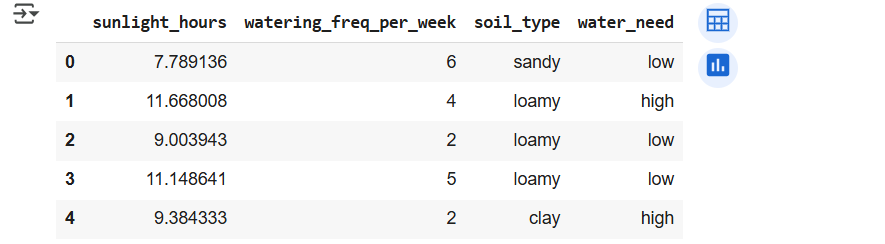
accuracy = accuracy\_score(y\_test, y\_pred)

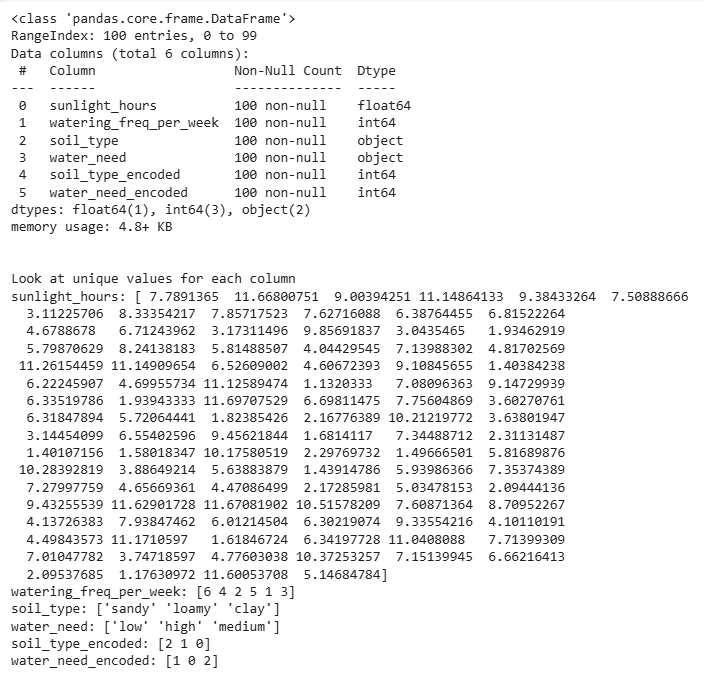
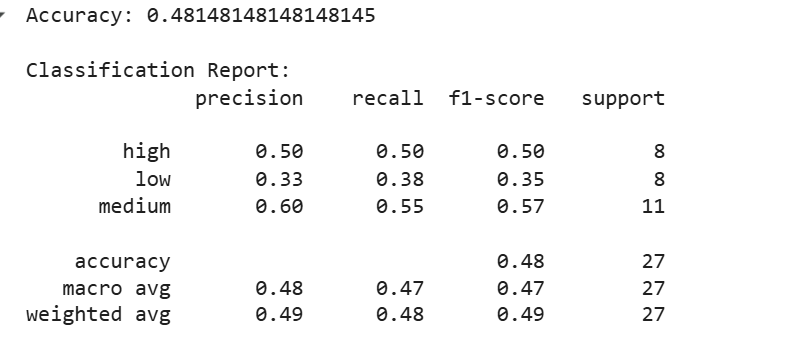
print("Accuracy:", accuracy)

print("\nClassification Report:")

print(classification\_report(y\_test, y\_pred, target\_names=target\_encoder.classes\_))

1. **OUTPUT**

****



1. **References**

* Given dataset : “plants.csv” .
* Matplotlib Documentation: [https://matplotlib.org](https://matplotlib.org/)
* Pandas Documentation: <https://pandas.pydata.org>
* Scikit documentation: <https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.GradientBoostingClassifier.html>