Assessment Report

on

“Classify Vegetables Based on Nutritional Content”

submitted as partial fulfillment for the award of

BACHELOR OF TECHNOLOGY DEGREE

SESSION 2024-25

in

CSE(AI&ML)

By

Name : Arnav Gupta

Roll Number : 202401100400048

section: A

Under the supervision of

BIKKI KUMAR SIR

KIET Group of Institutions, Ghaziabad

May, 2025

📚 Introduction

In this project, we aim to classify vegetables into categories like **leafy**, **fruit**, and **root** based on their **nutritional content** — specifically **Vitamin A**, **Vitamin C**, and **Fiber** levels. This type of classification is useful in nutritional planning and health-focused applications.

Below is an example dataset:

Vitamin A	Vitamin C	Fiber	Type
70.78	35.78	8.31	Root
8.17	82.82	1.15	Fruit
45.83	33.52	0.94	Leafy

We will use a **Decision Tree Classifier** to train a model and evaluate it using accuracy, precision, recall, F1-score, and a heatmap of the confusion matrix.

🛠️ Methodology

Data Loading: Read the `vegetables.csv` file containing nutritional info and type labels.

Data Splitting: Divide data into features (`vitamin_a`, `vitamin_c`, `fiber`) and labels (`type`).

Model Training: Use `DecisionTreeClassifier` from `sklearn`.

Model Evaluation:

- Classification Report: Precision, Recall, F1-score
- Confusion Matrix Heatmap

Visualization: Plot heatmap using `seaborn`.

💻 Code

The following is the code used in this project:

```
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
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score, classification_report,
confusion_matrix


# Load dataset
```

```
data = pd.read_csv('/content/vegetables.csv')


# Display sample info

print(data.head(), data.tail())

print(data.info())

print("\nMissing Values:\n", data.isnull().sum())


# Features and target

X = data[['vitamin_a', 'vitamin_c', 'fiber']]

y = data['type']


# Encode labels

y_encoder = LabelEncoder()

y_encoded = y_encoder.fit_transform(y)


# Train-test split

X_train, X_test, y_train, y_test = train_test_split(X, y_encoded,
test_size=0.2, random_state=42)


# Random Forest Classifier

model = RandomForestClassifier(n_estimators=100, random_state=42)

model.fit(X_train, y_train)


# Predictions

y_pred = model.predict(X_test)
```

```
# Evaluation

print("\nAccuracy:", accuracy_score(y_test, y_pred))

print("\nClassification Report:\n", classification_report(y_test,
y_pred, target_names=y_encoder.classes_))


# Confusion Matrix Heatmap

conf_matrix = confusion_matrix(y_test, y_pred)

plt.figure(figsize=(8, 6))

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

            xticklabels=y_encoder.classes_,
yticklabels=y_encoder.classes_)

plt.title("Confusion Matrix Heatmap")

plt.xlabel("Predicted")

plt.ylabel("Actual")

plt.show()
```

📈 Output/Results

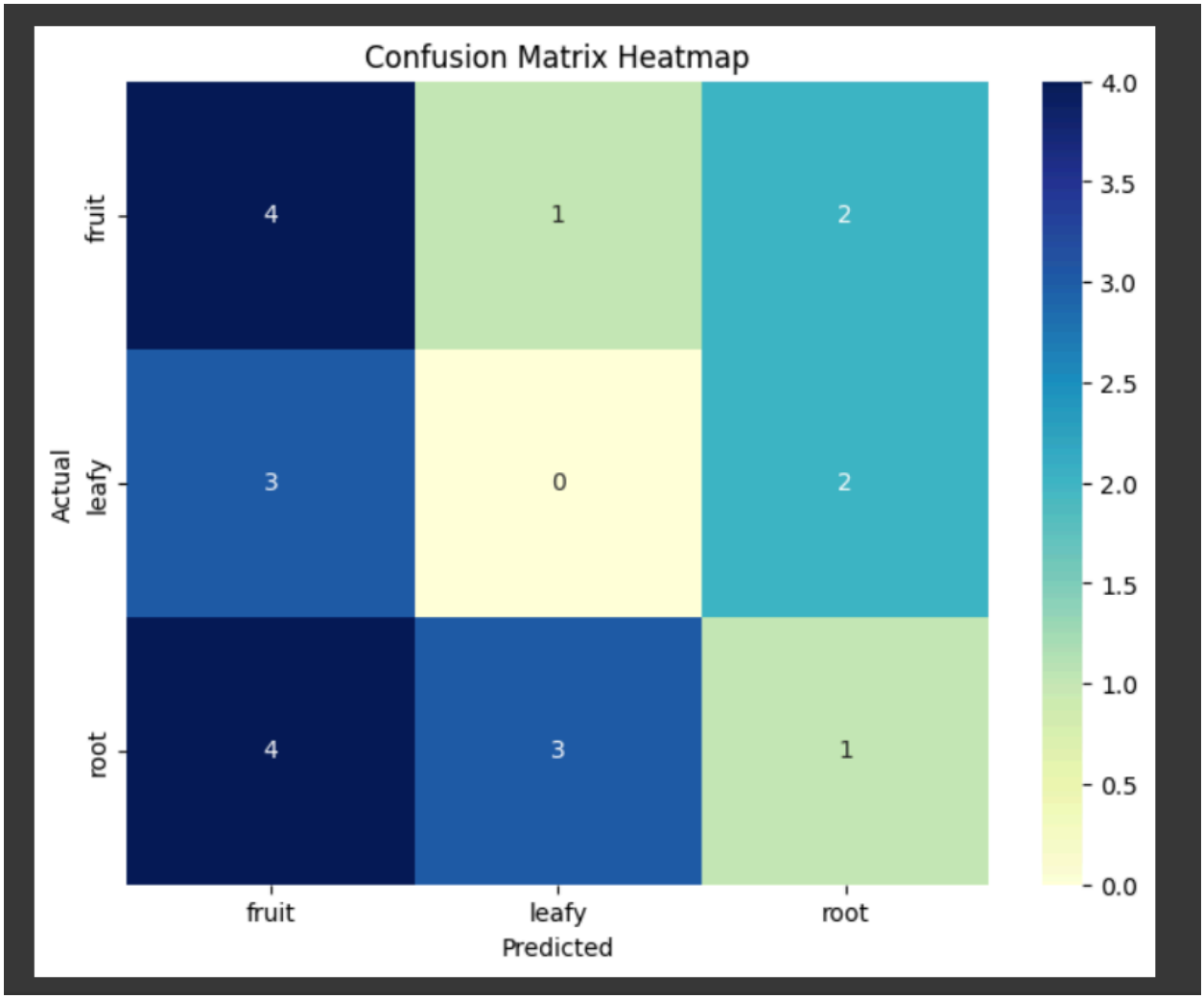
📸 **Paste the screenshot** from your Google Colab output here, including:

- Accuracy and Classification Report

```
First 5 rows:
   vitamin_a  vitamin_c     fiber   type
0  70.783510  35.779827  8.313735   root
1  54.353822  49.421245  5.989785  fruit
2   8.172535  82.824925  1.149330  fruit
3  45.830064  33.520805  0.938573  leafy
4  48.469629  17.376159  9.096268   root

Last 5 rows:
    vitamin_a  vitamin_c     fiber   type
95  71.024280  58.280684  0.801800  leafy
96  31.435011  99.543752  3.547713   root
97  47.116790  85.569609  9.417257  leafy
98  82.163690  52.144569  6.685726   root
99  45.926520   6.364091  6.786700  fruit
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 100 entries, 0 to 99
Data columns (total 4 columns):
 #   Column     Non-Null Count  Dtype
---  ------     --------------  -----
 0   vitamin_a  100 non-null    float64
 1   vitamin_c  100 non-null    float64
 2   fiber      100 non-null    float64
 3   type       100 non-null    object
dtypes: float64(3), object(1)
memory usage: 3.3+ KB

Missing Values:

vitamin_a    0
vitamin_c    0
fiber        0
```

```
 3   type        100 non-null    object
dtypes: float64(3), object(1)
memory usage: 3.3+ KB

Missing Values:

vitamin_a    0
vitamin_c    0
fiber        0
type         0
dtype: int64

Accuracy: 0.25

Classification Report:
              precision    recall  f1-score   support

       fruit       0.36      0.57      0.44         7
       leafy       0.00      0.00      0.00         5
        root       0.20      0.12      0.15         8

    accuracy                           0.25        20
   macro avg       0.19      0.23      0.20        20
weighted avg       0.21      0.25      0.22        20
```

- Confusion Matrix Heatmap



📚 References/Credits

- **Dataset**: Kaggle - Predict Employee Attrition Dataset
- **Random Forest Classifier Documentation**: scikit-learn
- **Confusion Matrix Heatmap Tutorial**: Seaborn Heatmap Documentation