K.I.E.T. Group of Institutions

Ghaziabad

Project title-Vegetable Classification Based on Nutritional Content

Name-Kushagra Srivastava

Roll no-202401100300143

Branch-CSE-AI

Section-B

Introduction

This project involves classifying vegetables into categories such as **leafy**, **root**, and **fruit** based on their nutritional attributes including Vitamin A, Vitamin C, and fiber content. Using machine learning classification algorithms, particularly Random Forest, the project aims to evaluate how well a model can distinguish between vegetable types based on nutrition.

Methodology

1. **Dataset**: The data contains nutritional information for various vegetables. The main features used were:

- Vitamin a
- Vitamin c
- fiber
- Target: type (root, leafy, fruit)

2. **Preprocessing**: The data was split into training and testing sets using an 80-20 split.

3. **Model**: A RandomForestClassifier from Scikit-learn was used to train the model.

4. **Evaluation**: The performance was measured using:

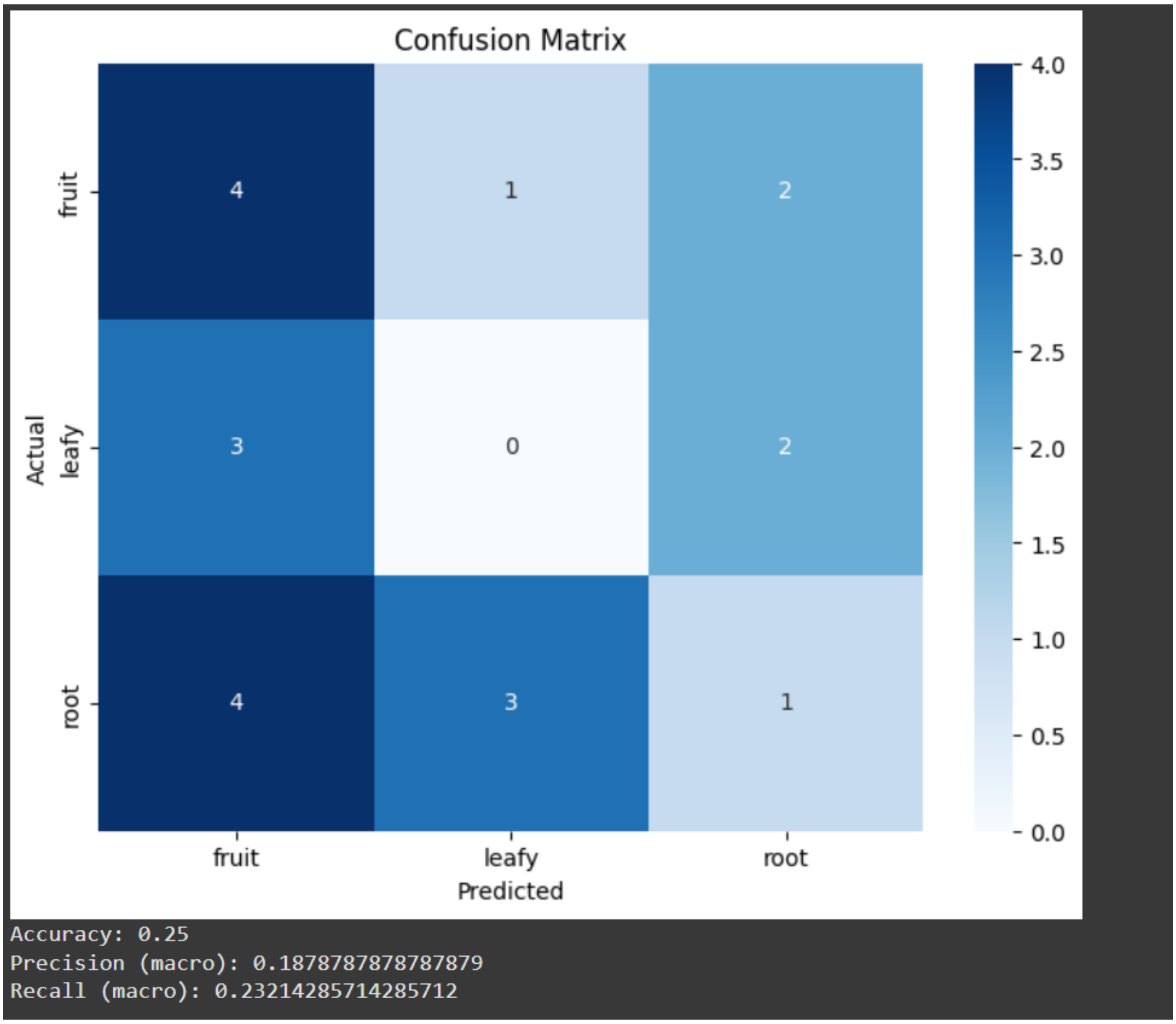
- Confusion matrix (visualized as a heatmap)
- Accuracy
- Precision
- Recall
- Classification report

Code:

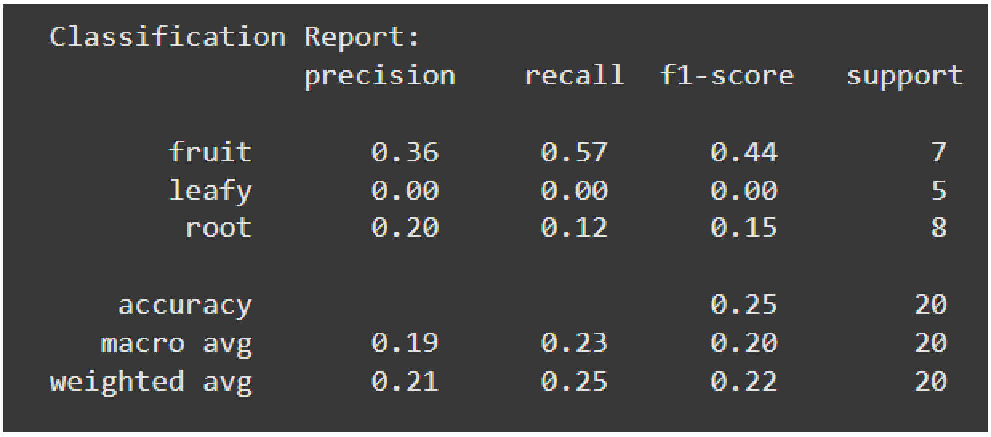
```
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import confusion_matrix, classification_report, accuracy_score, precision_score, recall_score
import seaborn as sns
import matplotlib.pyplot as plt

# Step 1: Load Dataset
df = pd.read_csv('vegetables.csv')

# Step 2: Preprocess Data
X = df.drop('type', axis=1)
y = df['type']

# Train-test split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Step 3: Train the Model
model = RandomForestClassifier(random_state=42)
model.fit(X_train, y_train)

# Step 4: Predictions
y_pred = model.predict(X_test)

# Step 5: Evaluation
# Confusion Matrix
cm = confusion_matrix(y_test, y_pred)
labels = model.classes_

# Heatmap
```

```
plt.figure(figsize=(8,6))
sns.heatmap(cm, annot=True, fmt='d', cmap='Blues', xticklabels=labels, yticklabels=labels)
plt.xlabel('Predicted')
plt.ylabel('Actual')
plt.title('Confusion Matrix')
plt.show()

# Metrics
accuracy = accuracy_score(y_test, y_pred)
precision = precision_score(y_test, y_pred, average='macro')  # macro for multiclass
recall = recall_score(y_test, y_pred, average='macro')

print("Accuracy:", accuracy)
print("Precision (macro):", precision)
print("Recall (macro):", recall)

# Detailed report
print("\nClassification Report:\n", classification_report(y_test, y_pred))
```

Output



```
Accuracy: 0.25
Precision (macro): 0.1878787878787879
Recall (macro): 0.23214285714285712
```



```
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
```

References

- **Dataset**: Provided.
- **Libraries**: Scikit-learn, Pandas, Matplotlib, Seaborn
- **Tools**: Google collab