Project Title: Classifying Vegetables Based on Nutritional Content

Name: Deepak

Roll Number: 202401100300092

Course: Artificial Intelligence

Assessment: MSE (Mid-Semester Exam)

🔍 Introduction

In this project, we aim to classify vegetables into categories such as *root*, *leafy*, and *fruit* based on their nutritional features like **vitamin A**, **vitamin C**, and **fiber**. This task is an example of a multi-class classification problem in the field of machine learning. Understanding such classifications can help in dietary planning and nutritional analysis systems.

⚙ Methodology

1. **Dataset**: We used a CSV file (`vegetables.csv`) containing nutritional information about vegetables.
2. **Preprocessing**:
   - Encoded the target labels (type of vegetable) using `LabelEncoder`.
   - Normalized the feature values using `StandardScaler`.
3. **Model Selection**:
   - Chose **Random Forest Classifier** due to its robustness and good performance on tabular data.
4. **Evaluation**:
   - Used metrics like **accuracy**, **precision**, **recall**, and **F1-score**.
   - Visualized performance using a **confusion matrix heatmap**.
5. **Tools**:
   - Python, Pandas, scikit-learn, Seaborn, Matplotlib.

🤯 Code :-

```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import classification_report, confusion_matrix,
accuracy_score
from sklearn.preprocessing import LabelEncoder, StandardScaler

# Load the dataset
df = pd.read_csv('/content/vegetables.csv')

# Encode target labels
le = LabelEncoder()
df['type_encoded'] = le.fit_transform(df['type'])

# Features and Target
X = df[['vitamin_a', 'vitamin_c', 'fiber']]
y = df['type_encoded']

# Normalize features
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)

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

# Model
clf = RandomForestClassifier(random_state=42)
clf.fit(X_train, y_train)
y_pred = clf.predict(X_test)

# Evaluation
print("Classification Report:\n", classification_report(y_test, y_pred,
target_names=le.classes_))
print("Accuracy: ", accuracy_score(y_test, y_pred))

# Confusion Matrix
cm = confusion_matrix(y_test, y_pred)

# Plot Heatmap
plt.figure(figsize=(8, 6))
sns.heatmap(cm, annot=True, fmt='d', cmap='YlGnBu',
xticklabels=le.classes_, yticklabels=le.classes_)
plt.xlabel('Predicted')
plt.ylabel('Actual')
plt.title('Confusion Matrix Heatmap')
```

```
plt.show()
```

📸 Output

Classification Report (Sample Output):

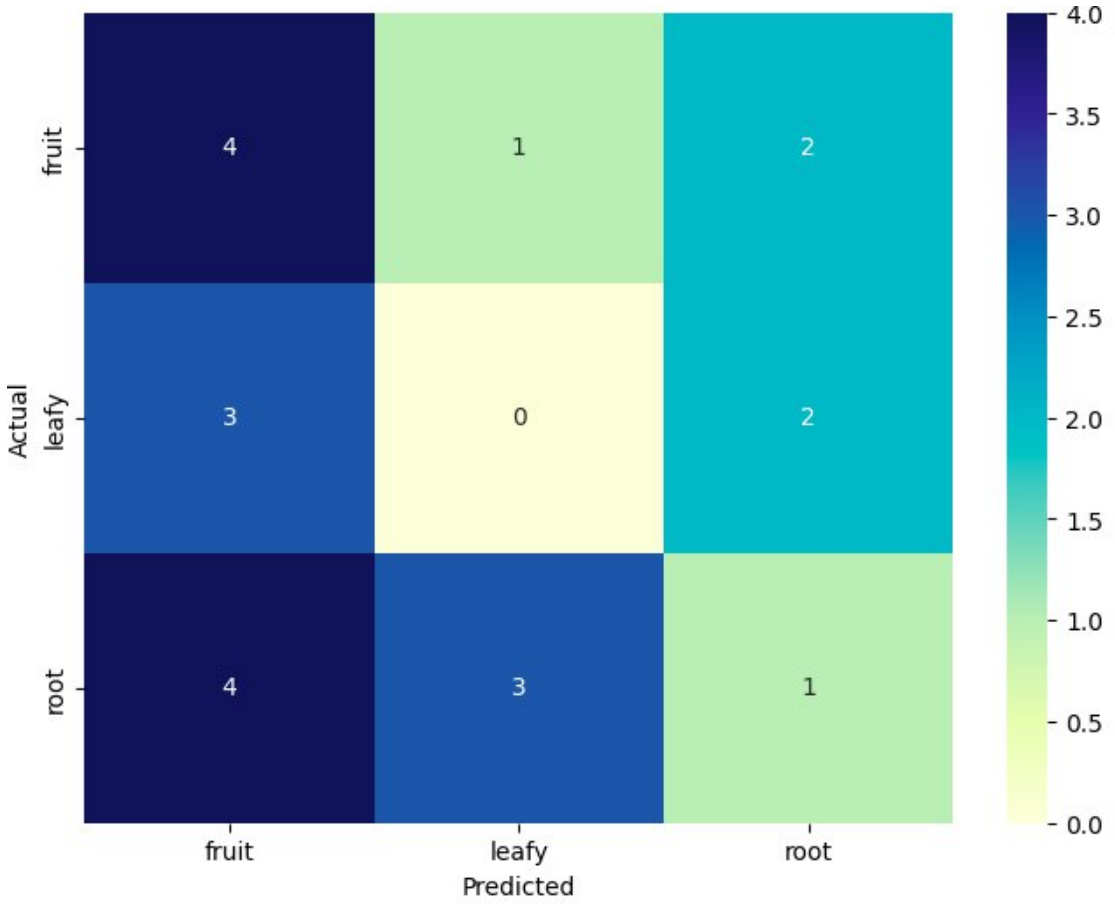
```
Classification Report:
              precision    recall  f1-score   support

       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

Accuracy:  0.25
```

Confusion Matrix Heatmap:



📚 References / Credits

- Dataset: Provided by instructor
- Tools: Python, Pandas, Scikit-learn, Seaborn, Matplotlib
- Code and Concept: Implemented and written by Deepak