

**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(AIML)**

By

Name : Abhinendra kumar

Roll Number : 202401100400008

Section: A

**Under the supervision of**

“BIKKI KUMAR GUPTA”

**KIET Group of Institutions, Ghaziabad**

**May, 2025**

**1. Introduction**

In this project, we aim to classify vegetables using a supervised machine learning approach. Classification problems like this are common in agricultural technology for sorting produce, enhancing supply chains, and improving market standards. We use a labeled dataset containing various features of vegetables.

**2. Methodology**

**Dataset:** A CSV file containing vegetable features.

**Preprocessing:**

* Handled missing values (if any).
* Standardized features using StandardScaler.

**Model Used:** Random Forest Classifier.

**Evaluation Metrics:** Accuracy Score and Classification Report.

**Tools:** Python, pandas, scikit-learn.

**3.Code**

**# No need to install anything extra for basic modeling**

**import pandas as pd**

**import numpy as np**

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

**from sklearn.preprocessing import StandardScaler**

**from sklearn.ensemble import RandomForestClassifier**

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

**# Load the dataset**

**df = pd.read\_csv('/content/drive/MyDrive/vegetables.csv')  # Update path if needed**

**df.head()**

**# Features and target**

**X = df[['vitamin\_a', 'vitamin\_c', 'fiber']]**

**y = df['type']**

**# Split into train and test**

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

**# Scale the features**

**scaler = StandardScaler()**

**X\_train\_scaled = scaler.fit\_transform(X\_train)**

**X\_test\_scaled = scaler.transform(X\_test)**

**# Using Random Forest Classifier for simplicity and accuracy**

**model = RandomForestClassifier(random\_state=42)**

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

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

**print("Accuracy:", accuracy\_score(y\_test, y\_pred))**

**print("\nClassification Report:\n", classification\_report(y\_test, y\_pred))**

**# Get new data from user**

**vitamin\_a = float(input("Enter vitamin A content: "))**

**vitamin\_c = float(input("Enter vitamin C content: "))**

**fiber = float(input("Enter fiber content: "))**

**# Create an array from the user input**

**new\_data = np.array([[vitamin\_a, vitamin\_c, fiber]])**

**# Scale the new data using the same scaler**

**new\_data\_scaled = scaler.transform(new\_data)**

**# Make the prediction**

**prediction = model.predict(new\_data\_scaled)**

**# Output the prediction**

**print("Predicted vegetable type:", prediction[0])**

**4.Output**

**A computer screen shot of a computer

AI-generated content may be incorrect.**

**5. References**

* Dataset: Provided by course/instructor (or your own collected data)
* Libraries: pandas, numpy, scikit-learn
* Documentation: Scikit-learn
* Code and concept inspiration: Classroom lectures and official docs