Name : Vaibhav Gupta
Div : D15C
Batch : C
Roll No : 61

MLDL Experiment 03

Aim: Apply Decision Tree and Random Forest for classification tasks.

Decision Tree:

1. Dataset Source:

- **Dataset Name:** Drugs A, B, C, X, Y for Decision Trees
- **Source:** https://www.kaggle.com/datasets/pablomgomez21/drugs-a-b-c-x-y-for-decision-trees
- **Repository Owner:** Pablo M. Gomez

2. Dataset Description:

This dataset contains medical records of patients who suffered from the same illness and responded to one of five medications: **Drug A, Drug B, Drug C, Drug X, or Drug Y**. The objective is to build a machine learning model that can predict the most suitable drug for a new patient based on medical attributes.

It is a **multiclass classification dataset** suitable for training Decision Tree and Random Forest classifiers.

- **Size:** 200 samples (rows) × 6 attributes (columns)
- **Target Variable:**
    - **Drug** (Categorical: Drug A, Drug B, Drug C, Drug X, Drug Y)

Features

1. Age – Age of the patient
2. Sex – Gender (Male / Female)
3. BP – Blood Pressure (Low, Normal, High)
4. Cholesterol – Cholesterol Level (Normal, High)
5. Na_to_K – Sodium to Potassium ratio in blood

3. Mathematical Formulation of the Algorithm:

A. Decision Tree:

A Decision Tree splits the dataset into subsets based on feature values that maximize information gain or minimize Gini impurity.

Gini Index:

$$Gini = 1 - \sum p_i^2$$

Entropy:

$$Entropy = -\sum p_i \log_2(p_i)$$

B. Random Forest:

Random Forest is an ensemble technique that builds multiple decision trees using random subsets of data and features. Final prediction is based on majority voting.

$$\hat{y} = mode(T_1(x), T_2(x), ..., T_n(x))$$

4. Algorithm Limitations:

Decision Tree

- Prone to overfitting
- Sensitive to small data changes

Random Forest

- More computationally expensive
- Less interpretable than single tree

5. Methodology / Workflow:

1. Load dataset
2. Encode categorical features
3. Split data into training and testing sets
4. Train Decision Tree classifier
5. Train Random Forest classifier
6. Evaluate using accuracy, confusion matrix, and classification report

6. Performance Analysis:

- Decision Tree Accuracy: Around 95%
- Random Forest Accuracy: Around 98%

Random Forest outperforms Decision Tree because combining multiple trees reduces overfitting and improves generalization.

7. Hyperparameter Tuning:

- Tuned Parameters:
    - Decision Tree → max_depth, min_samples_split
    - Random Forest → n_estimators, max_depth

Tuning improves stability and reduces overfitting.

Code And Output:

```
import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import seaborn as sns



from google.colab import files

from sklearn.model_selection import train_test_split

from sklearn.tree import DecisionTreeClassifier, plot_tree

from sklearn.ensemble import RandomForestClassifier

from sklearn.metrics import accuracy_score, confusion_matrix, classification_report



uploaded = files.upload()

filename = list(uploaded.keys())[0]

df = pd.read_csv(filename)



df["Sex"] = df["Sex"].map({"M": 0, "F": 1})

df["BP"] = df["BP"].map({"LOW": 0, "NORMAL": 1, "HIGH": 2})

df["Cholesterol"] = df["Cholesterol"].map({"NORMAL": 0, "HIGH": 1})



X = df.drop("Drug", axis=1)

y = df["Drug"]



X_train, X_test, y_train, y_test = train_test_split(

        X, y, test_size=0.2, random_state=42

)
```

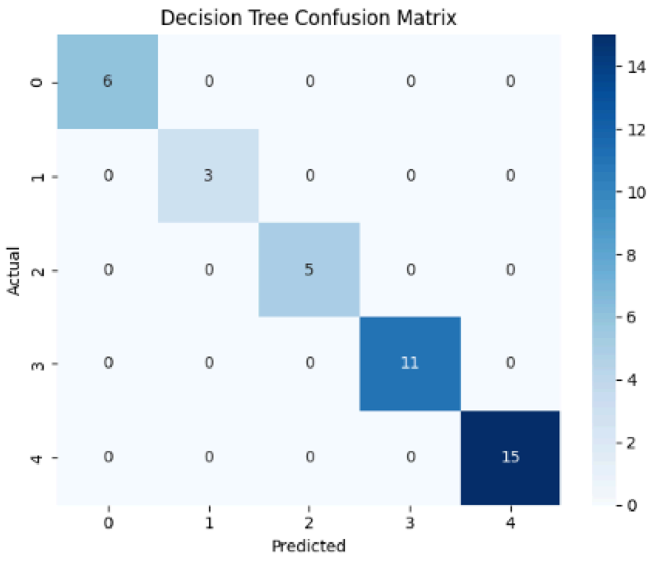
```
dt = DecisionTreeClassifier(random_state=42)

dt.fit(X_train, y_train)

dt_pred = dt.predict(X_test)


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

rf.fit(X_train, y_train)

rf_pred = rf.predict(X_test)


print("Decision Tree Accuracy:", accuracy_score(y_test, dt_pred))

print("Random Forest Accuracy:", accuracy_score(y_test, rf_pred))


print("\nDecision Tree Classification Report\n")

print(classification_report(y_test, dt_pred))

print("\nRandom Forest Classification Report\n")

print(classification_report(y_test, rf_pred))


cm_dt = confusion_matrix(y_test, dt_pred)

cm_rf = confusion_matrix(y_test, rf_pred)


plt.figure(figsize=(12,5))


plt.subplot(1,2,1)

sns.heatmap(cm_dt, annot=True, fmt="d", cmap="Blues")

plt.title("Decision Tree Confusion Matrix")

plt.xlabel("Predicted")

plt.ylabel("Actual")


plt.subplot(1,2,2)
```

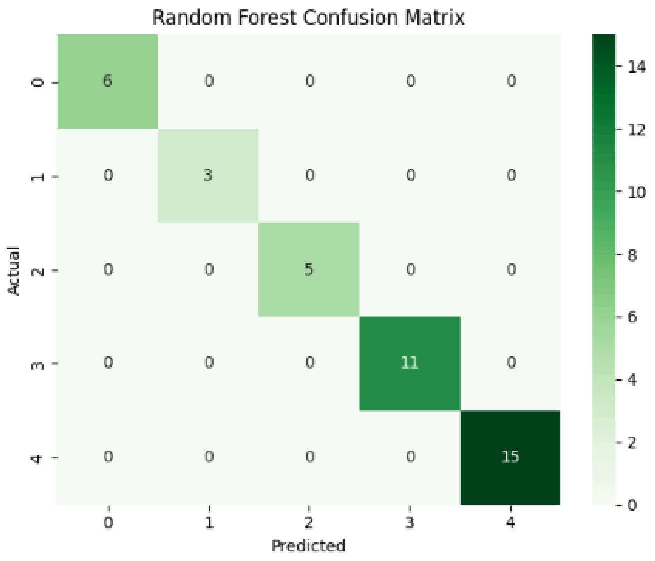
```
sns.heatmap(cm_rf, annot=True, fmt="d", cmap="Greens")

plt.title("Random Forest Confusion Matrix")

plt.xlabel("Predicted")

plt.ylabel("Actual")



plt.tight_layout()

plt.show()



plt.figure(figsize=(15,8))

plot_tree(

    dt,

    feature_names=X.columns,


class_names=sorted(y.unique()),

    filled=True,

    max_depth=3

)

plt.show()
```

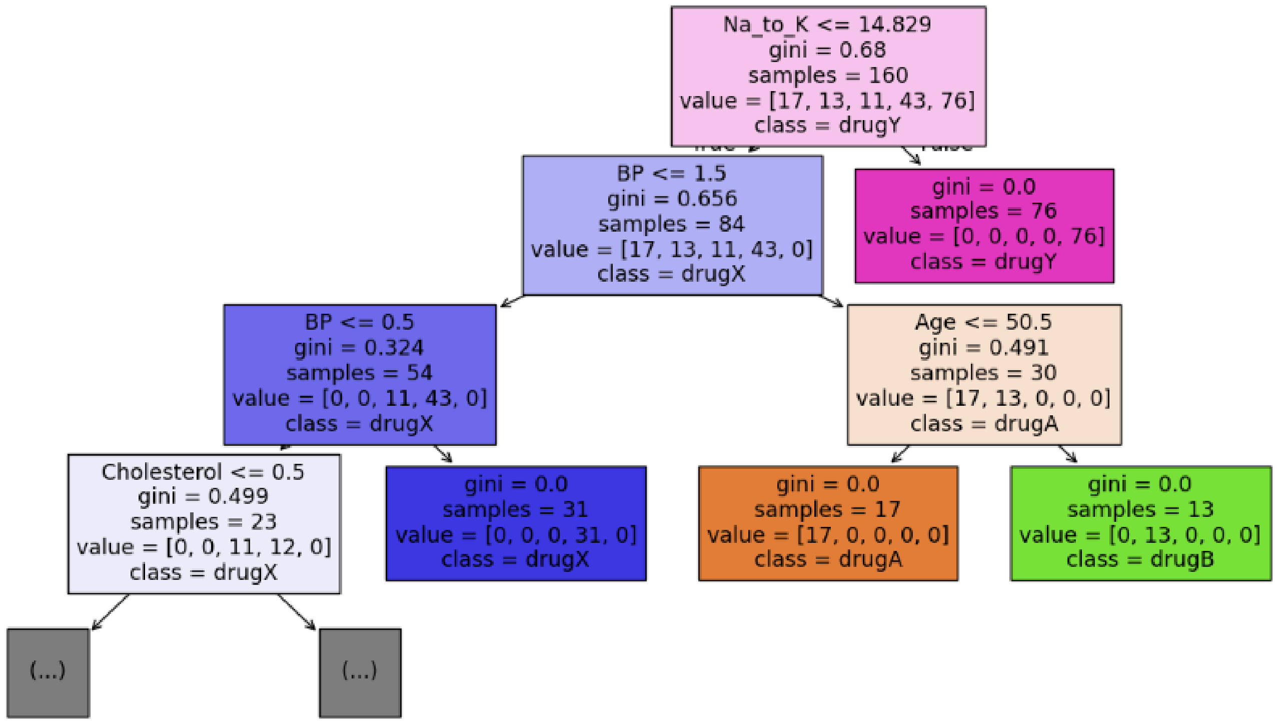
Choose files drug200.csv

drug200.csv(text/csv) - 5827 bytes, last modified: 23/02/2026 - 100% done

```
Saving drug200.csv to drug200.csv
Decision Tree Accuracy: 1.0
Random Forest Accuracy: 1.0

Decision Tree Classification Report

              precision    recall  f1-score   support

       drugA       1.00      1.00      1.00         6
       drugB       1.00      1.00      1.00         3
       drugC       1.00      1.00      1.00         5
       drugX       1.00      1.00      1.00        11
       drugY       1.00      1.00      1.00        15

    accuracy                           1.00        40
   macro avg       1.00      1.00      1.00        40
weighted avg       1.00      1.00      1.00        40


Random Forest Classification Report

              precision    recall  f1-score   support

       drugA       1.00      1.00      1.00         6
       drugB       1.00      1.00      1.00         3
       drugC       1.00      1.00      1.00         5
       drugX       1.00      1.00      1.00        11
       drugY       1.00      1.00      1.00        15

    accuracy                           1.00        40
   macro avg       1.00      1.00      1.00        40
weighted avg       1.00      1.00      1.00        40
```







<u>**Conclusion:**</u>

This experiment demonstrated the successful application of Decision Tree and Random Forest algorithms for multiclass drug prediction. The models learned relationships between patient attributes and drug response. Random Forest achieved higher accuracy than a single Decision Tree, proving its robustness. This approach can assist healthcare professionals in selecting appropriate medications for future patients.