Experiment No. – 1.4

Student Name:	**Milan Sharma**	**UID:**	**23MAI10003**
Branch:	**ME – CSE - AIML**	**Section/Group:**	**23MAI – 1 (A)**
Semester:	**2nd**	**Date of Performance:**	**07Feb2024**
Subject Name:	**Machine Learning Lab**	**Subject Code:**	**23 CSH 651**

Aim of the Experiment:

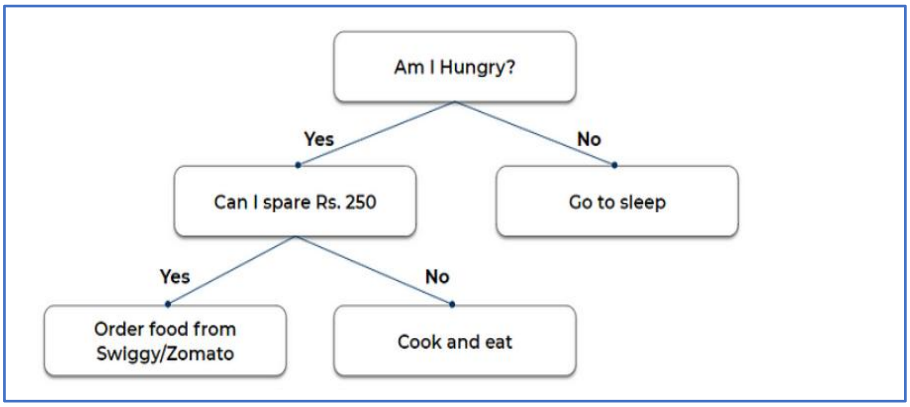
Implementing Decision Tree algorithm using Python

Theory:

A decision tree is a flowchart-like tree structure where each internal node denotes the feature, branches denote the rules and the leaf nodes denote the result of the algorithm. It is a versatile supervised machine-learning algorithm, which is used for both classification and regression problems. It is one of the very powerful algorithms. And it is also used in Random Forest to train on different subsets of training data, which makes random forest one of the most powerful algorithms in machine learning.

Decision Tree Terminologies:-

1. Root Node, Decision/Internal Node, Leaf/Terminal Node
2. Branch/Sub-Tree, Parent Node, Child Node
3. Impurity: A measurement of the target variable’s homogeneity in a subset of data. It refers to the degree of randomness or uncertainty in a set of examples. The Gini index and entropy are two commonly used impurity measurements in decision trees for classifications task
4. Information Gain: Information gain is a measure of the reduction in impurity achieved by splitting a dataset on a particular feature in a decision tree.
5. Pruning: The process of removing branches from the tree that do not provide any additional information or lead to overfitting.
6. Entropy: Entropy is the measure of the degree of randomness or uncertainty in the dataset.
7. Example(Structure) of Decision Tree



Important Formulas :-

$$\textbf{Entropy} = -\,(P_{pass}\log_2(P_{pass}) + P_{fail}\log_2(P_{fail}))$$

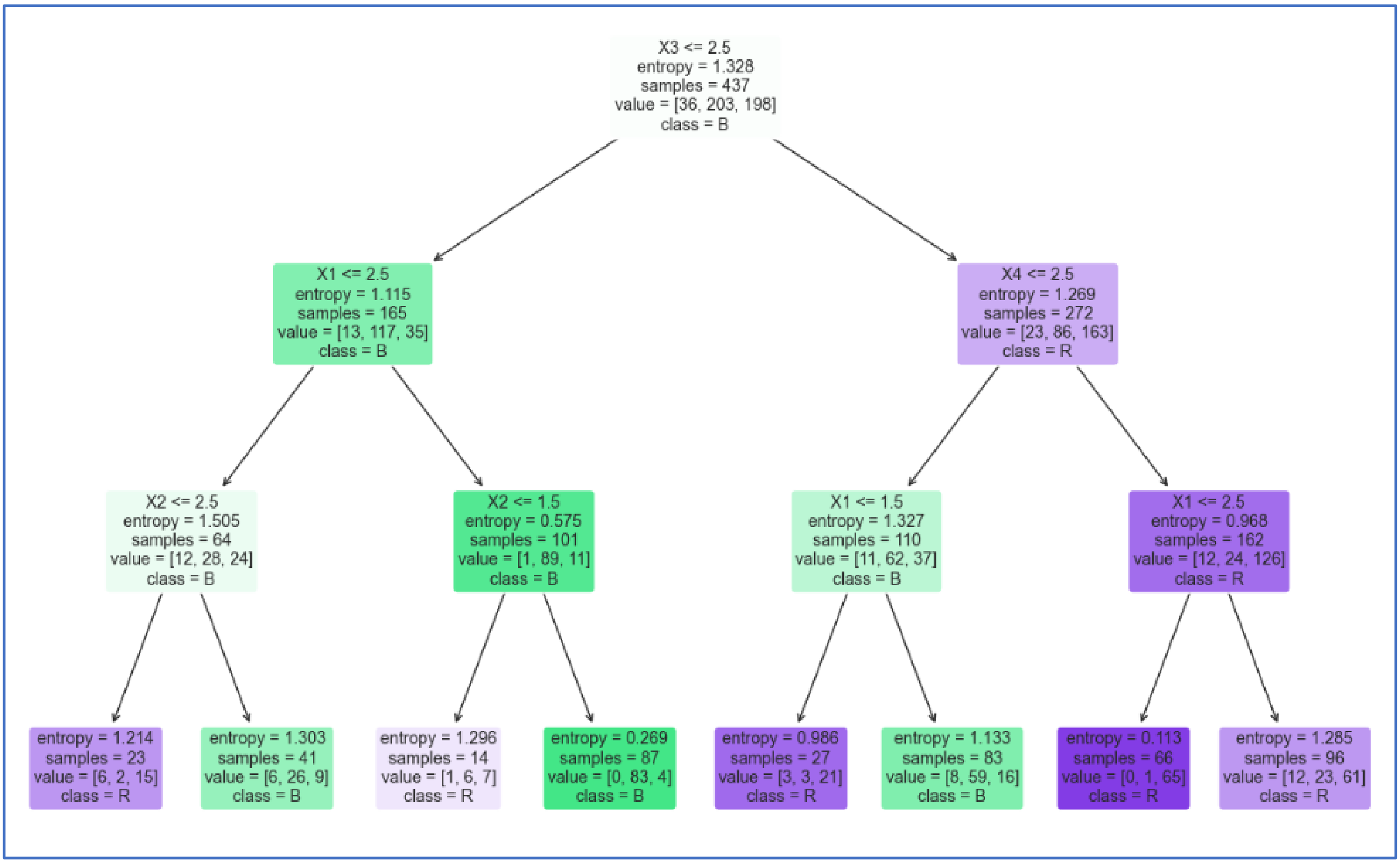
$$\textbf{Average Entropy} = [\,(n_{child\ 1}) / n\_parent\,] * E\_child1 + [\,(n_{child\ 2}) / n\_parent\,] * E\_child2$$

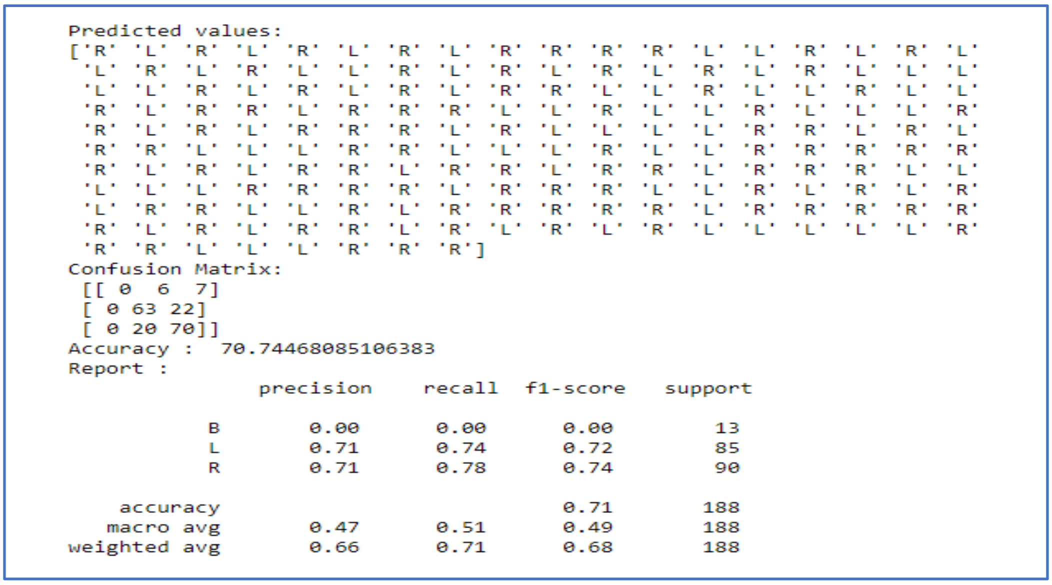
$$\textbf{INFORMATION GAIN} = \textbf{ENTROPY}_{PARENT} - \textbf{ENTROPY}_{CHILD}$$

Code:

```
# Importing the required packages
import numpy as np
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier, plot_tree
from sklearn.metrics import confusion_matrix, accuracy_score,
classification_report
import matplotlib.pyplot as plt

# Load the dataset
balance_data = pd.read_csv('C:/Users/milan\Downloads/balance-scale.csv')
print("Dataset First 5 rows: \n", balance_data.head())

X = balance_data.values[:, 1:5]
Y = balance_data.values[:, 0]
# Splitting the dataset into train and test
X_train, X_test, y_train, y_test = train_test_split(X, Y, test_size=0.3,
random_state=100)

# Decision tree with entropy
clf_entropy = DecisionTreeClassifier(criterion="entropy",
random_state=100,max_depth=3, min_samples_leaf=5)
clf_entropy.fit(X_train, y_train)

# Plotting the Decision Tree
feature_names=['X1', 'X2', 'X3', 'X4']
class_names=['L', 'B', 'R']
plt.figure(figsize=(15, 10))
plot_tree(clf_entropy, filled=True, feature_names=feature_names,
class_names=class_names, rounded=True)
plt.show()

# Result of the Decision Tree Model
y_pred_entropy = clf_entropy.predict(X_test)
print("Predicted values:")
print(y_pred_entropy)
print("Confusion Matrix: \n",confusion_matrix(y_test, y_pred_entropy))
print("Accuracy : ",accuracy_score(y_test, y_pred_entropy)*100)
print("Report : \n",classification_report(y_test, y_pred_entropy))
```

Output:





```
Predicted values:
['R' 'L' 'R' 'L' 'R' 'L' 'R' 'L' 'R' 'R' 'R' 'R' 'L' 'L' 'R' 'L' 'R' 'L'
 'L' 'R' 'L' 'R' 'L' 'L' 'R' 'L' 'R' 'L' 'R' 'L' 'R' 'L' 'R' 'L' 'L' 'L'
 'L' 'L' 'R' 'L' 'R' 'L' 'R' 'L' 'R' 'R' 'L' 'L' 'R' 'L' 'L' 'R' 'L' 'L'
 'R' 'L' 'R' 'R' 'L' 'R' 'R' 'R' 'L' 'L' 'R' 'L' 'L' 'R' 'L' 'L' 'L' 'R'
 'R' 'L' 'R' 'L' 'R' 'R' 'R' 'L' 'R' 'L' 'L' 'L' 'L' 'R' 'R' 'L' 'R' 'L'
 'R' 'R' 'L' 'L' 'L' 'R' 'R' 'L' 'L' 'L' 'R' 'L' 'L' 'R' 'R' 'R' 'R' 'R'
 'R' 'L' 'R' 'L' 'R' 'R' 'L' 'R' 'R' 'L' 'R' 'R' 'L' 'R' 'R' 'R' 'L' 'L'
 'L' 'L' 'L' 'R' 'R' 'R' 'R' 'L' 'R' 'R' 'R' 'L' 'L' 'R' 'L' 'R' 'L' 'R'
 'L' 'R' 'R' 'L' 'L' 'R' 'L' 'R' 'R' 'R' 'R' 'R' 'L' 'R' 'R' 'R' 'R' 'R'
 'R' 'L' 'R' 'L' 'R' 'R' 'L' 'R' 'L' 'R' 'L' 'R' 'L' 'L' 'L' 'L' 'L' 'R'
 'R' 'R' 'L' 'L' 'L' 'R' 'R' 'R']
Confusion Matrix:
 [[ 0  6  7]
 [ 0 63 22]
 [ 0 20 70]]
Accuracy :  70.74468085106383
Report :
               precision    recall  f1-score   support

           B       0.00      0.00      0.00        13
           L       0.71      0.74      0.72        85
           R       0.71      0.78      0.74        90

    accuracy                           0.71       188
   macro avg       0.47      0.51      0.49       188
weighted avg       0.66      0.71      0.68       188
```

Learning Outcomes:

1. **I learnt about various python libraries like pandas, sklearn.**
2. **I learnt about the concept of Decision Tree.**
3. **I learnt about how to read the dataset using pandas.**
4. **I learnt about the Entropy and Information Gain in Decision Tree.**