In [ ]:
```
# #https://www.w3schools.com/python/python_ml_decision_tree.asp
# Machine Learning Lab2: Created by Jibrael Jos,PhD
# Topic: Decision Tree Explorations
# Student Name: naveen Krishna
# Roll No: 23122023
# Date: 12th March 2024
# Submission : 12th March 2024
```

Questions:

1.Upload data from a csv file
2.Upload from a text file where seperator is tab
3.Upload from an Excel Sheet

In [ ]:
```
# Uploading data from a csv file.

import pandas as pd

df = pd.read_csv('dataTree1.csv')
df
```

Out[ ]:

	Age	Experience	Rank	Nationality	Go
0	36	10	9	UK	NO
1	42	12	4	USA	NO
2	23	4	6	N	NO
3	52	4	4	USA	NO
4	43	21	8	USA	YES
5	44	14	5	UK	NO
6	66	3	7	N	YES
7	35	14	9	UK	YES
8	52	13	7	N	YES
9	35	5	9	N	YES
10	24	3	5	USA	NO
11	18	3	7	UK	YES
12	45	9	9	UK	YES

In [ ]:
```
# Upload from a text file where seperator is tab

import pandas as pd

input_file_path = 'dataTree1.csv'
output_file_path = 'dataTree1.txt'

df = pd.read_csv(input_file_path, sep=',')

df.to_csv(output_file_path, sep='\t', index=False)
```

```
df = pd.read_csv('dataTree1.txt', sep="\t")
df
```

Out[ ]:

	Age	Experience	Rank	Nationality	Go
0	36	10	9	UK	NO
1	42	12	4	USA	NO
2	23	4	6	N	NO
3	52	4	4	USA	NO
4	43	21	8	USA	YES
5	44	14	5	UK	NO
6	66	3	7	N	YES
7	35	14	9	UK	YES
8	52	13	7	N	YES
9	35	5	9	N	YES
10	24	3	5	USA	NO
11	18	3	7	UK	YES
12	45	9	9	UK	YES

In [ ]:

```
# Upload from an Excel file.

df = pd.read_excel('dataTree1.xlsx')
df
```

Out[ ]:

	Age	Experience	Rank	Nationality	Go
0	36	10	9	UK	NO
1	42	12	4	USA	NO
2	23	4	6	N	NO
3	52	4	4	USA	NO
4	43	21	8	USA	YES
5	44	14	5	UK	NO
6	66	3	7	N	YES
7	35	14	9	UK	YES
8	52	13	7	N	YES
9	35	5	9	N	YES
10	24	3	5	USA	NO
11	18	3	7	UK	YES
12	45	9	9	UK	YES

Questions:

4. Explore map function in a data frame
5. Create a map function to convert a month column to Numbers Jan-1, Feb-2 and so on
6. Create a map function to convert True to 1 and False to Zero

In [ ]:
```
# Exploring map functions and creating a map function to convert to Numbe
## Creating a new column and inputting values for months by using basic p
import random as rnd
from sklearn import tree
from sklearn.tree import DecisionTreeClassifier
import matplotlib.pyplot as plt



with open('dataTree1.csv', 'r')as file:
    lines = file.readlines()

months = ['Jan', 'Feb', 'March', 'April', 'May', 'June', 'July', 'Aug', '




# Uncomment only when you want to make modifications in your csv file.

# with open('dataTree1.csv', 'w')as file:
#     for line in lines:
#         r = rnd.randint(0,11)
#         modified_line = line.strip()+","+months[r]
#         file.write(modified_line+'\n')
#     file.close

# header = 'Age,Experience,Rank,Nationality,Go,Month\n'
# with open('dataTree1.csv', 'r')as file:
#     line = file.readlines()

# with open('dataTree1.csv','w')as file:
#     file.write(header)
#     for i in line:
#         file.write(i)




# Using of map function:
df = pd.read_csv('dataTree1.csv')
month_map = {'Jan': 0, 'Feb': 1, 'March': 2, 'April': 3, 'May': 4, 'June'
df['Month'] = df['Month'].map(month_map)

# Create a map function to convert Yes to 1 and False to 0
map_go = {'YES': 1, 'NO': 0}
df['Go'] = df['Go'].map(map_go)

# Create a map function for converting the Nationalities to numeric value
map_nationality = {'UK': 0, 'USA': 1, 'N': 2}
```

```
df['Nationality'] = df['Nationality'].map(map_nationality)
df
```

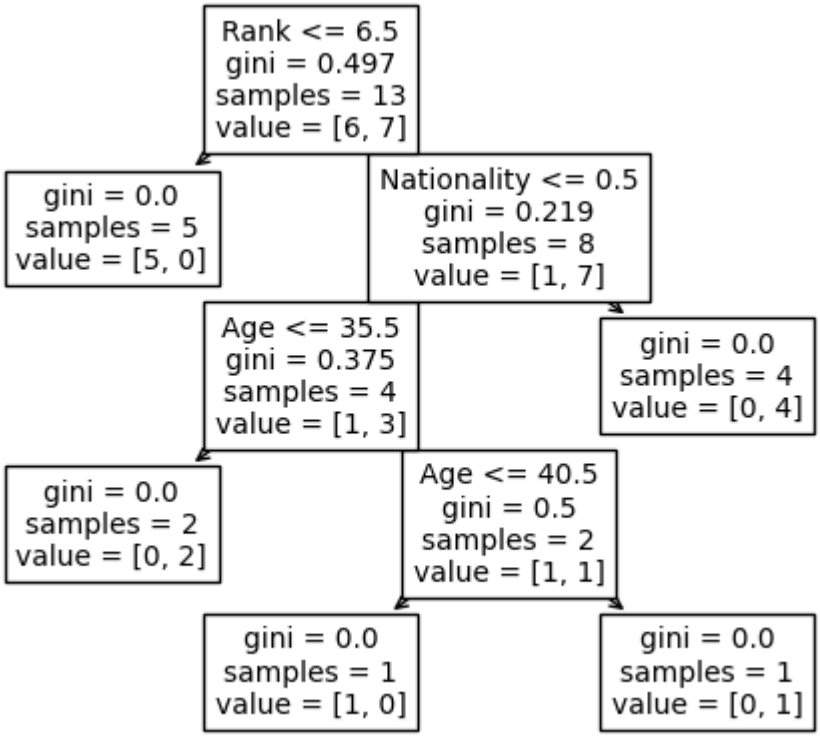
Out[ ]:

	Age	Experience	Rank	Nationality	Go	Month
0	36	10	9	0	0	5
1	42	12	4	1	0	10
2	23	4	6	2	0	8
3	52	4	4	1	0	3
4	43	21	8	1	1	9
5	44	14	5	0	0	0
6	66	3	7	2	1	5
7	35	14	9	0	1	6
8	52	13	7	2	1	0
9	35	5	9	2	1	0
10	24	3	5	1	0	9
11	18	3	7	0	1	5
12	45	9	9	0	1	4

In [ ]:
```
features = ['Age', 'Experience', 'Rank', 'Nationality']
print(features)
X = df[features]
y = df['Go']

dtree = DecisionTreeClassifier(criterion='gini')
dtree = dtree.fit(X, y)

tree.plot_tree(dtree, feature_names=features)
```

```
['Age', 'Experience', 'Rank', 'Nationality']
```

Out[ ]:
```
[Text(0.4, 0.9, 'Rank <= 6.5\ngini = 0.497\nsamples = 13\nvalue = [6,
7]'),
 Text(0.2, 0.7, 'gini = 0.0\nsamples = 5\nvalue = [5, 0]'),
 Text(0.6, 0.7, 'Nationality <= 0.5\ngini = 0.219\nsamples = 8\nvalue =
[1, 7]'),
 Text(0.4, 0.5, 'Age <= 35.5\ngini = 0.375\nsamples = 4\nvalue = [1,
3]'),
 Text(0.2, 0.3, 'gini = 0.0\nsamples = 2\nvalue = [0, 2]'),
 Text(0.6, 0.3, 'Age <= 40.5\ngini = 0.5\nsamples = 2\nvalue = [1, 1]'),
 Text(0.4, 0.1, 'gini = 0.0\nsamples = 1\nvalue = [1, 0]'),
 Text(0.8, 0.1, 'gini = 0.0\nsamples = 1\nvalue = [0, 1]'),
 Text(0.8, 0.5, 'gini = 0.0\nsamples = 4\nvalue = [0, 4]')]
```



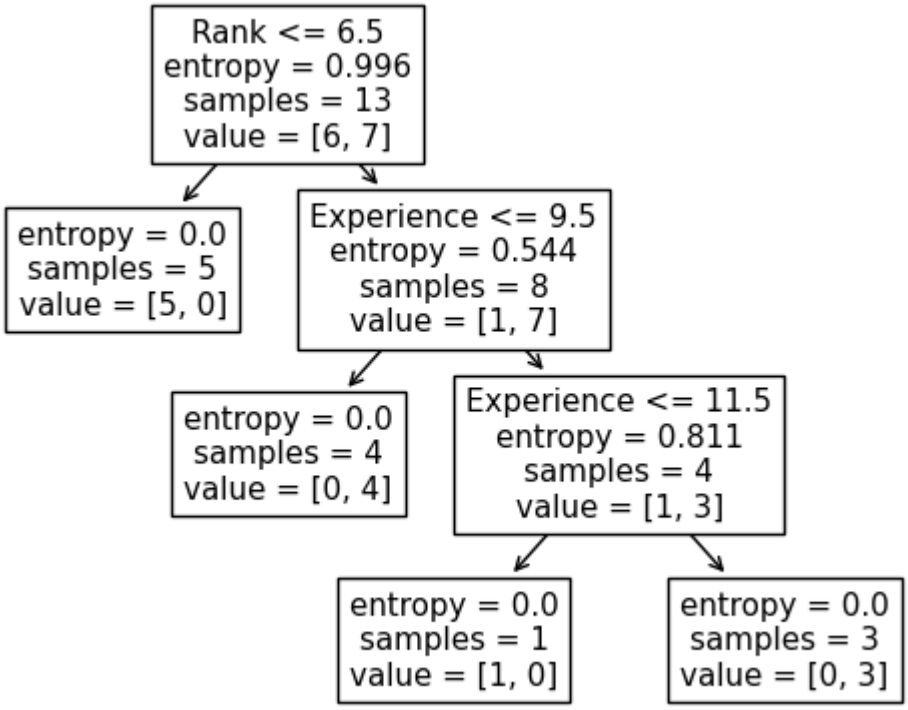
Questions:

7. Run Code above
8. Check GINI value with Maths Calculation in an Excel Workbook
9. Change Gini to Entropy and check calculation
10. Change different parameters and study the impact

In [ ]:
```
# Changing gini to entropy bu using the parameter creterion:

features = ['Age', 'Experience', 'Rank', 'Nationality']
print(features)
X = df[features]
y = df['Go']

dtree = DecisionTreeClassifier(criterion='entropy')
dtree = dtree.fit(X, y)

tree.plot_tree(dtree, feature_names=features)
```

```
['Age', 'Experience', 'Rank', 'Nationality']
```

Out[ ]:
```
[Text(0.3333333333333333, 0.875, 'Rank <= 6.5\nentropy = 0.996\nsamples
= 13\nvalue = [6, 7]'),
 Text(0.16666666666666666, 0.625, 'entropy = 0.0\nsamples = 5\nvalue =
[5, 0]'),
 Text(0.5, 0.625, 'Experience <= 9.5\nentropy = 0.544\nsamples = 8\nvalu
e = [1, 7]'),
 Text(0.3333333333333333, 0.375, 'entropy = 0.0\nsamples = 4\nvalue =
[0, 4]'),
 Text(0.6666666666666666, 0.375, 'Experience <= 11.5\nentropy = 0.811\ns
amples = 4\nvalue = [1, 3]'),
 Text(0.5, 0.125, 'entropy = 0.0\nsamples = 1\nvalue = [1, 0]'),
 Text(0.8333333333333334, 0.125, 'entropy = 0.0\nsamples = 3\nvalue =
[0, 3]')]
```



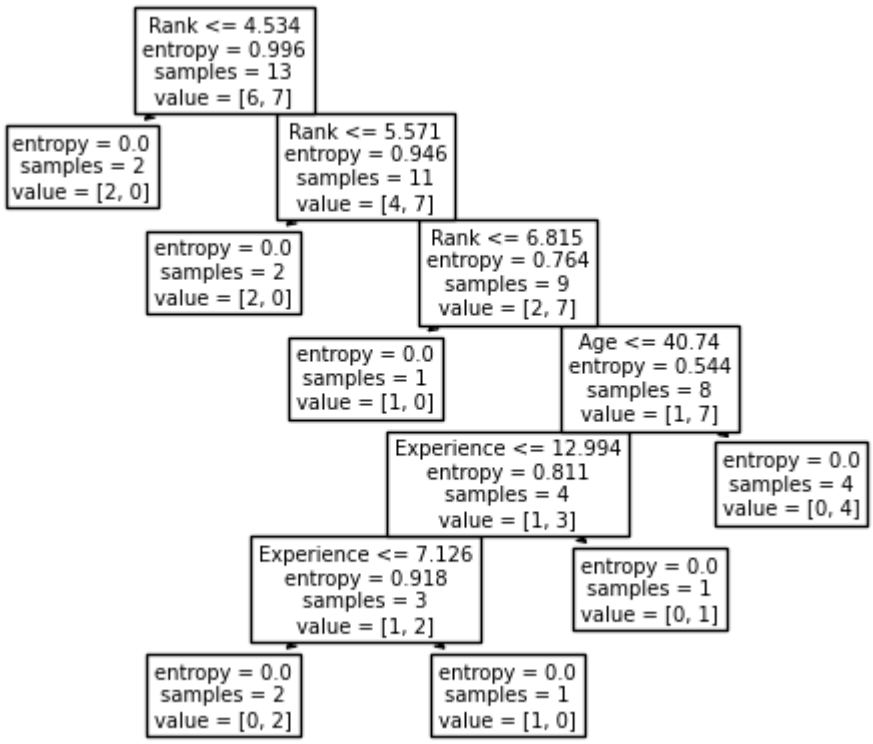
Some of the main parameters are performed below:

Splitter

In [ ]:
```
# splitter: It is one of the main paramenetr in the decision tree classif
# It is used to select the splits for each of the nodes in the dtree.
# There are 2 values for the parameter, one which will choose the 'best'
# Initially it will be set into the 'best' splitting if we are not mentio

features = ['Age', 'Experience', 'Rank', 'Nationality']
print(features)
X = df[features]
y = df['Go']

dtree = DecisionTreeClassifier(criterion='entropy', splitter="random")
dtree = dtree.fit(X, y)

tree.plot_tree(dtree, feature_names=features)
```

```
['Age', 'Experience', 'Rank', 'Nationality']
```

Out[ ]:
```
[Text(0.2857142857142857, 0.9285714285714286, 'Rank <= 4.534\nentropy =
0.996\nsamples = 13\nvalue = [6, 7]'),
 Text(0.14285714285714285, 0.7857142857142857, 'entropy = 0.0\nsamples =
2\nvalue = [2, 0]'),
 Text(0.42857142857142855, 0.7857142857142857, 'Rank <= 5.571\nentropy =
0.946\nsamples = 11\nvalue = [4, 7]'),
 Text(0.2857142857142857, 0.6428571428571429, 'entropy = 0.0\nsamples =
2\nvalue = [2, 0]'),
 Text(0.5714285714285714, 0.6428571428571429, 'Rank <= 6.815\nentropy =
0.764\nsamples = 9\nvalue = [2, 7]'),
 Text(0.42857142857142855, 0.5, 'entropy = 0.0\nsamples = 1\nvalue = [1,
0]'),
 Text(0.7142857142857143, 0.5, 'Age <= 40.74\nentropy = 0.544\nsamples =
8\nvalue = [1, 7]'),
 Text(0.5714285714285714, 0.35714285714285715, 'Experience <= 12.994\nen
tropy = 0.811\nsamples = 4\nvalue = [1, 3]'),
 Text(0.42857142857142855, 0.21428571428571427, 'Experience <= 7.126\nen
tropy = 0.918\nsamples = 3\nvalue = [1, 2]'),
 Text(0.2857142857142857, 0.07142857142857142, 'entropy = 0.0\nsamples =
2\nvalue = [0, 2]'),
 Text(0.5714285714285714, 0.07142857142857142, 'entropy = 0.0\nsamples =
1\nvalue = [1, 0]'),
 Text(0.7142857142857143, 0.21428571428571427, 'entropy = 0.0\nsamples =
1\nvalue = [0, 1]'),
 Text(0.8571428571428571, 0.35714285714285715, 'entropy = 0.0\nsamples =
4\nvalue = [0, 4]')]
```



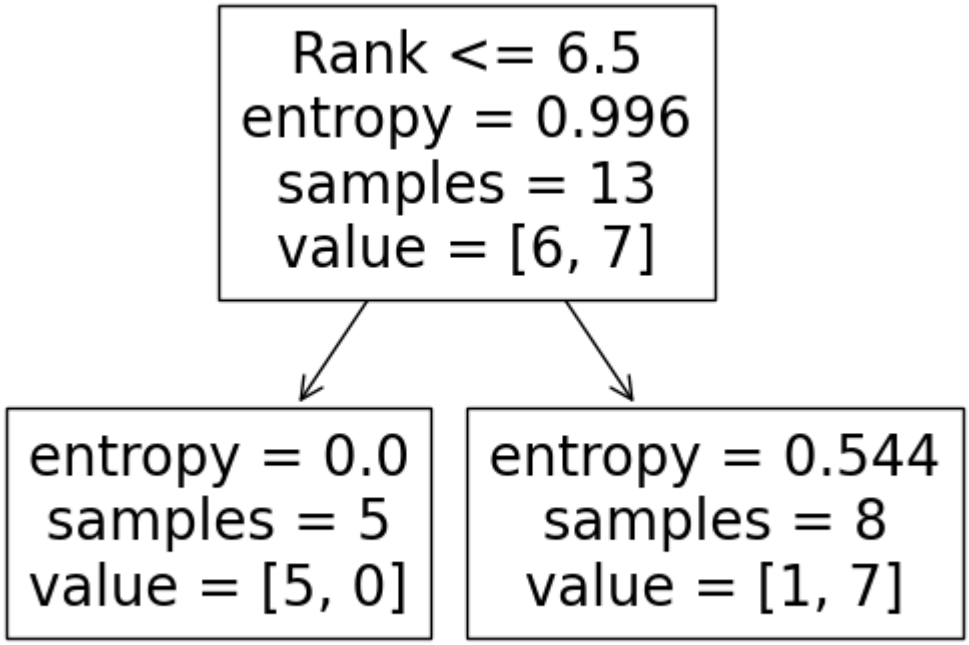
Max_depth

In [ ]:
```
#max_depth: The maximum depth of the tree.
# If not specified or None, the tree is expanded until all leaves are pur

features = ['Age', 'Experience', 'Rank', 'Nationality']
print(features)
X = df[features]
y = df['Go']
```

```
dtree = DecisionTreeClassifier(criterion='entropy', splitter="best", max_
dtree = dtree.fit(X, y)

tree.plot_tree(dtree, feature_names=features)
```

```
['Age', 'Experience', 'Rank', 'Nationality']
```

Out[ ]:
```
[Text(0.5, 0.75, 'Rank <= 6.5\nentropy = 0.996\nsamples = 13\nvalue =
[6, 7]'),
 Text(0.25, 0.25, 'entropy = 0.0\nsamples = 5\nvalue = [5, 0]'),
 Text(0.75, 0.25, 'entropy = 0.544\nsamples = 8\nvalue = [1, 7]')]
```

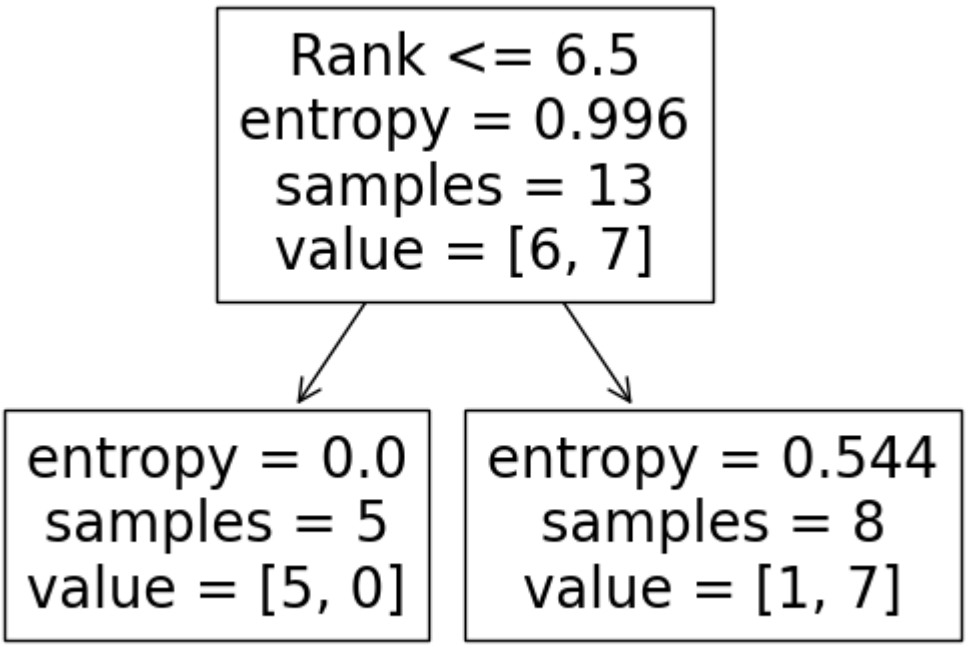


Random_state

In [ ]:
```
# random_state: This parameter allows you to control the randomness invol

features = ['Age', 'Experience', 'Rank', 'Nationality']
print(features)
X = df[features]
y = df['Go']

dtree = DecisionTreeClassifier(criterion='entropy', splitter="best", max_
dtree = dtree.fit(X, y)

tree.plot_tree(dtree, feature_names=features)
```

```
['Age', 'Experience', 'Rank', 'Nationality']
```

Out[ ]:
```
[Text(0.5, 0.75, 'Rank <= 6.5\nentropy = 0.996\nsamples = 13\nvalue =
[6, 7]'),
 Text(0.25, 0.25, 'entropy = 0.0\nsamples = 5\nvalue = [5, 0]'),
 Text(0.75, 0.25, 'entropy = 0.544\nsamples = 8\nvalue = [1, 7]')]
```



Questions:

11. Check IRIS dataset with Entropy
12. Compare method we used in Excel and in Python for IRIS data
13. Read about Decision Tree (share URL read). Identify advantages and disadvantages
14. Watch a video on Decision Tree(share URL)Share some learnings
15 Decision Tree can be viewed using plottree and graphviz .. Explore both methods

---

In [ ]:
```
# Check Iris dataset with Entropy

import pandas as pd
df = pd.read_csv("Iris.csv")
df.head()
```

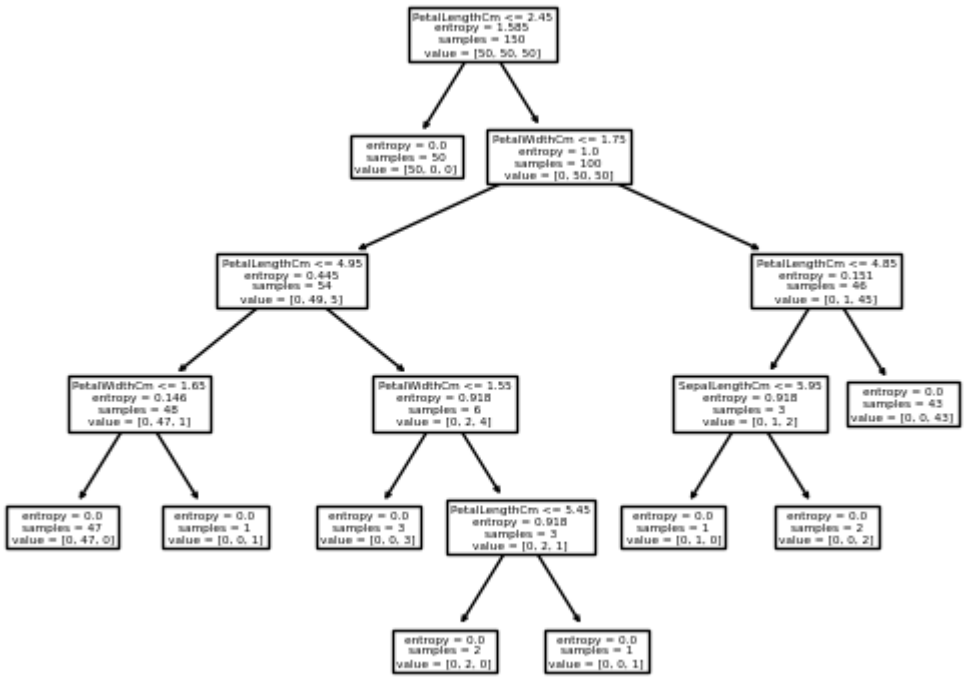
Out[ ]:

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
0	1	5.1	3.5	1.4	0.2	Iris-setosa
1	2	4.9	3.0	1.4	0.2	Iris-setosa
2	3	4.7	3.2	1.3	0.2	Iris-setosa
3	4	4.6	3.1	1.5	0.2	Iris-setosa
4	5	5.0	3.6	1.4	0.2	Iris-setosa

In [ ]:
```
# Exploring map function and changing the class names of the iris flowers

df = pd.read_csv('Iris.csv', index_col = 'Id')

iris = {'Iris-setosa': 0, 'Iris-versicolor': 1, 'Iris-virginica': 2}
df['Species'] = df['Species'].map(iris)
df['Species']

# Saving the changes by creating another csv file along with the chnages.
df.to_csv('Iris_2.csv', index=False)
```

In [ ]:
```
from sklearn.tree import DecisionTreeClassifier
from sklearn import tree
import matplotlib.pyplot as plt
df = pd.read_csv("Iris.csv")

features = ['SepalLengthCm', 'SepalWidthCm', 'PetalLengthCm', 'PetalWidth

X = df[features]
y = df['Species']

dtree = DecisionTreeClassifier(criterion='entropy')
dtree = dtree.fit(X, y)

tree.plot_tree(dtree, feature_names=features)
```

Out[ ]:
```
[Text(0.5, 0.9166666666666666, 'PetalLengthCm <= 2.45\nentropy = 1.585\nsamples = 150\nvalue = [50, 50, 50]'),
 Text(0.4230769230769231, 0.75, 'entropy = 0.0\nsamples = 50\nvalue = [50, 0, 0]'),
 Text(0.5769230769230769, 0.75, 'PetalWidthCm <= 1.75\nentropy = 1.0\nsamples = 100\nvalue = [0, 50, 50]'),
 Text(0.3076923076923077, 0.5833333333333334, 'PetalLengthCm <= 4.95\nentropy = 0.445\nsamples = 54\nvalue = [0, 49, 5]'),
 Text(0.15384615384615385, 0.4166666666666667, 'PetalWidthCm <= 1.65\nentropy = 0.146\nsamples = 48\nvalue = [0, 47, 1]'),
 Text(0.07692307692307693, 0.25, 'entropy = 0.0\nsamples = 47\nvalue = [0, 47, 0]'),
 Text(0.23076923076923078, 0.25, 'entropy = 0.0\nsamples = 1\nvalue = [0, 0, 1]'),
 Text(0.46153846153846156, 0.4166666666666667, 'PetalWidthCm <= 1.55\nentropy = 0.918\nsamples = 6\nvalue = [0, 2, 4]'),
 Text(0.38461538461538464, 0.25, 'entropy = 0.0\nsamples = 3\nvalue = [0, 0, 3]'),
 Text(0.5384615384615384, 0.25, 'PetalLengthCm <= 5.45\nentropy = 0.918\nsamples = 3\nvalue = [0, 2, 1]'),
 Text(0.46153846153846156, 0.08333333333333333, 'entropy = 0.0\nsamples = 2\nvalue = [0, 2, 0]'),
 Text(0.6153846153846154, 0.08333333333333333, 'entropy = 0.0\nsamples = 1\nvalue = [0, 0, 1]'),
 Text(0.8461538461538461, 0.5833333333333334, 'PetalLengthCm <= 4.85\nentropy = 0.151\nsamples = 46\nvalue = [0, 1, 45]'),
 Text(0.7692307692307693, 0.4166666666666667, 'SepalLengthCm <= 5.95\nentropy = 0.918\nsamples = 3\nvalue = [0, 1, 2]'),
 Text(0.6923076923076923, 0.25, 'entropy = 0.0\nsamples = 1\nvalue = [0, 1, 0]'),
 Text(0.8461538461538461, 0.25, 'entropy = 0.0\nsamples = 2\nvalue = [0, 0, 2]'),
 Text(0.9230769230769231, 0.4166666666666667, 'entropy = 0.0\nsamples = 43\nvalue = [0, 0, 43]')]
```



Advantages and Disadvantages of Decision Tree

Advantages:

- Dicision tree requires less efforts for data preparation during pre-processing
- It does not require normalization of the data.
- It doesnot require scaling of data as well.
- Missing values doesnot affect the performance of the tree
- It is very easy to explain

Disadvantages:

- A small change in the data can cause a big drastic change in the structure of the decision tree.
- Sometimes the calculations may be more complex than other algorithms.
- It often involves hight time to train the data
- Training in Decision tree may cause more expensive in terms of complexity and time taken is more.
- This algorithm is inadequate for applying regression and predicting continuous values.

Link reffered : dhirajkumarblog.medium.com

Some learnings on Decision Tree

1. Introductrion to Decision Tree

- A supervised machine learning algorithm that can be applied to regression and classification problems is the decision tree.
- Recursively dividing the dataset into subsets according to the most important feature at each node is how it operates.

2. Components of Decision Tree

1. **Root Node:** The topmost node in the Decision Tree
2. **Internal Node:** The node thatrepresent a decision.
3. **Leaf Node:** erminal nodes that provide the final output

3. Splitting Creteria

- Decision trees use various criteria to determine how to split the data at each node. Common criteria include Gini impurity and information gain (entropy).
- The goal is to create pure nodes with samples belonging to a single class.

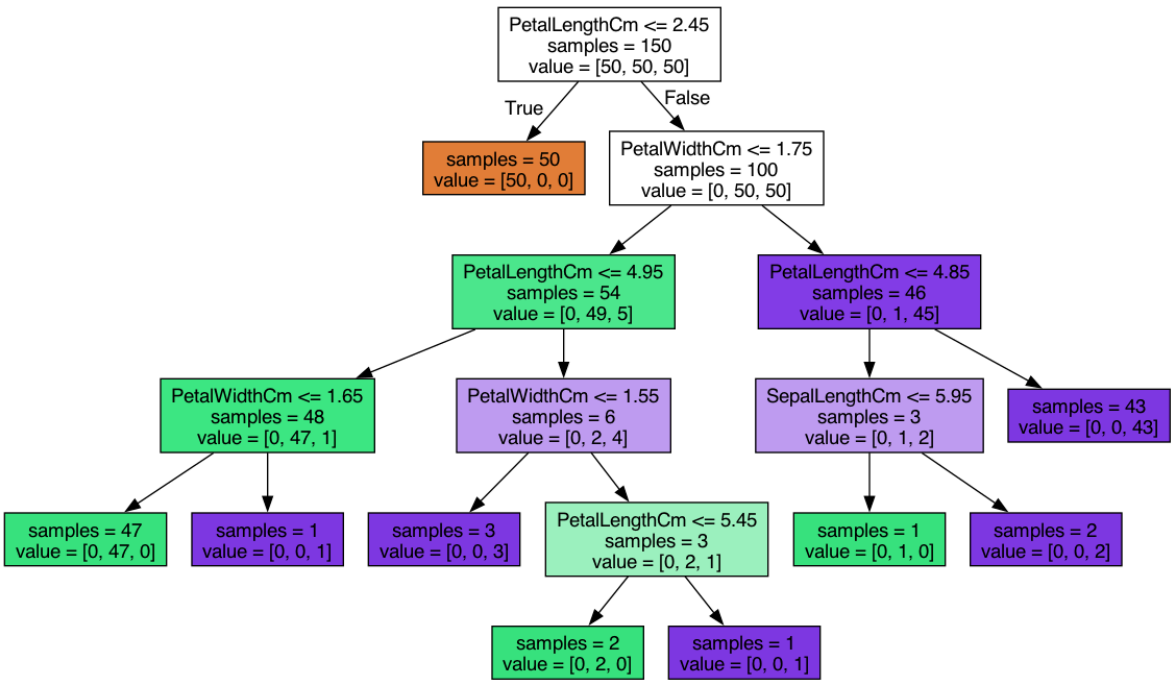
4. Some important terms

- **Entropy:** Measures the randomness or disorder in a set of samples. It is minimized when all samples belong to a single class.
- **Gini impurity:** Measures the probability of incorrectly classifying a randomly chosen element.

Links reffered: Youtube video link

Implimenting the decision tree graph using graphviz package

In [ ]:
```
from sklearn.tree import DecisionTreeClassifier
from sklearn import tree
import matplotlib.pyplot as plt
import graphviz

df = pd.read_csv("Iris.csv")

features = ['SepalLengthCm', 'SepalWidthCm', 'PetalLengthCm', 'PetalWidth

X = df[features]
y = df['Species']

dtree = DecisionTreeClassifier(criterion='entropy')
dtree = dtree.fit(X, y)
```



In [ ]:
```
```