+ Code + Text

Name : Ayush Makwana

Roll No : 18BCE107

Practical : 4

```
import pandas as pd
import numpy as np
from google.colab import drive
drive.mount('/content/drive')
path='/content/drive/My Drive/1_DM/'
```

```
Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mour
```

```
df = pd.read_csv(path+'DT.csv')
df
```

	Age	Income	Student	Credit Rating	Buy Car
0	Young	High	No	Fair	No
1	Young	High	No	Good	No
2	Middle	High	No	Fair	Yes
3	Old	Medium	No	Fair	Yes
4	Old	Low	Yes	Fair	Yes
5	Old	Low	Yes	Good	No
6	Middle	Low	Yes	Good	Yes
7	Young	Medium	No	Fair	No
8	Young	Low	Yes	Fair	Yes
9	Old	Medium	Yes	Fair	Yes
10	Young	Medium	Yes	Good	Yes
11	Middle	Medium	No	Good	Yes
12	Middle	High	Yes	Fair	Yes
13	Old	Medium	No	Good	No

```
def calculate_entropy(df_label):
  classes,class_counts = np.unique(df_label,return_counts = True)
  entropy_value = np.sum([(-class_counts[i]/np.sum(class_counts))*np.log2(class_counts[i]/np
```

```
    entropy_value = np.sum([(-class_counts[i]/np.sum(class_counts))*np.log2(class_counts[i]/np.
                        for i in range(len(classes))])
    return entropy_value



def calculate_information_gain(dataset,feature,label):
    dataset_entropy = calculate_entropy(dataset[label])
    values,feat_counts= np.unique(dataset[feature],return_counts=True)

    weighted_feature_entropy = np.sum([(feat_counts[i]/np.sum(feat_counts))*calculate_entropy
                               ==values[i]).dropna()[label]) for i in range(len(values))])
    feature_info_gain = dataset_entropy - weighted_feature_entropy
    print(feature_info_gain,"\n")
    return feature_info_gain


def myEncoder(arr = []):
    d = {}
    res = []
    count = 0
    for i in arr:
        if i in d:
            res.append(d[i])
        else:
            d[i] = count
            count += 1
            res.append(d[i])
    return res



list1 = df.keys()

for i in range(len(list1)-1):
  df[list1[i]] = pd.Series(myEncoder(df[list1[i]].tolist()))
df
```

	Age	Income	Student	Credit Rating	Buy Car
0	0	0	0	0	No
1	0	0	0	1	No
2	1	0	0	0	Yes
3	2	1	0	0	Yes
4	2	2	1	0	Yes

```
# Set the features and label
features = df.columns[:-1]
label = 'Buy Car'
parent=None
features
```

```
Index(['Age', 'Income', 'Student', 'Credit Rating'], dtype='object')
```

```
dataset_entropy = calculate_entropy(df[label])
print("Info of D :",dataset_entropy,"\n")
list2 = []
for i in features:
  print("Gain of",i,":")
  list2.append(calculate_information_gain(df,i,label))
list2

print("\nRoot node is :",features[list2.index(max(list2))])
```

```
Info of D : 0.9402859586706309

Gain of Age :
0.2467498197744391

Gain of Income :
0.029222565658954647

Gain of Student :
0.15183550136234136

Gain of Credit Rating :
0.04812703040826927


Root node is : Age
```

Example

```
df = pd.read_csv(path+'laptops.csv')
df
```

	TypeName	ScreenResolution	Cpu	Ram	Memory	Gpu	OpSys	Company
0	Ultrabook	IPS Panel Retina Display 2560x1600	Intel Core i5 2.3GHz	8GB	128GB SSD	Intel Iris Plus Graphics 640	macOS	Apple
1	Ultrabook	1440x900	Intel Core i5 1.8GHz	8GB	128GB Flash Storage	Intel HD Graphics 6000	macOS	Apple
2	Notebook	Full HD 1920x1080	Intel Core i5 7200U 2.5GHz	8GB	256GB SSD	Intel HD Graphics 620	No OS	HP
3	Ultrabook	IPS Panel Retina Display 2880x1800	Intel Core i7 2.7GHz	16GB	512GB SSD	AMD Radeon Pro 455	macOS	Apple
4	Ultrabook	IPS Panel Retina Display 2560x1600	Intel Core i5 3.1GHz	8GB	256GB SSD	Intel Iris Plus Graphics 650	macOS	Apple
...	...	...	...	...	...	...	...	...
1298	2 in 1 Convertible	IPS Panel Full HD / Touchscreen 1920x1080	Intel Core i7 6500U 2.5GHz	4GB	128GB SSD	Intel HD Graphics 520	Windows 10	Lenovo
		IPS Panel Quad	Intel			Intel HD		

```
list3 = df.keys()

for i in range(len(list3)-1):
  df[list3[i]] = pd.Series(myEncoder(df[list3[i]].tolist()))
df
```

	TypeName	ScreenResolution	Cpu	Ram	Memory	Gpu	OpSys	Company
0	0	0	0	0	0	0	0	Apple
1	0	1	1	0	1	1	0	Apple
2	1	2	2	0	2	2	1	HP

```
# Set the features and label
features = df.columns[:-1]
label = 'Company'
parent=None
features
```

```
Index(['TypeName', 'ScreenResolution', 'Cpu', 'Ram', 'Memory', 'Gpu', 'OpSys'], dtype='o
```

```
dataset_entropy = calculate_entropy(df[label])
print("Info of D :",dataset_entropy,"\n")
list4 = []
for i in features:
  print("Gain of",i,":")
  list4.append(calculate_information_gain(df,i,label))
list4

print("\nRoot node is :",features[list4.index(max(list4))])
```

```
Info of D : 2.878461222011528

Gain of TypeName :
0.3185216211078834

Gain of ScreenResolution :
0.49644778052221383

Gain of Cpu :
0.863791742851753

Gain of Ram :
0.1680245346696907

Gain of Memory :
0.4810965069036035

Gain of Gpu :
0.9642764265959092

Gain of OpSys :
0.3470134071655391


Root node is : Gpu
```

CONCLUSION

Decision trees assist analysts in evaluating upcoming choices. The tree creates a visual representation of all possible outcomes, rewards and follow-up decisions in one document.