

Practical 6

Write A Program In Python To Implement ID3.

In [3]:

```
import numpy as np
import pandas as pd
eps = np.finfo(float).eps
from numpy import log2 as log

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

Out[3]:

	Outlook	Temperature	Humidity	Windy	PlayTennis
0	Sunny	Hot	High	Weak	No
1	Sunny	Hot	High	Strong	No
2	Overcast	Hot	High	Weak	Yes
3	Rainy	Mild	High	Weak	Yes
4	Rainy	Cold	Normal	Weak	Yes
5	Rainy	Cold	Normal	Strong	No
6	Overcast	Cold	Normal	Strong	Yes
7	Sunny	Mild	High	Weak	No
8	Sunny	Cold	Normal	Weak	Yes
9	Rainy	Mild	Normal	Weak	Yes
10	Sunny	Mild	Normal	Strong	Yes
11	Overcast	Mild	High	Strong	Yes
12	Overcast	Hot	Normal	Weak	Yes
13	Rainy	Mild	High	Strong	No

In [4]:

```
def find_entropy(df):
    Class = df.keys()[-1]
    entropy = 0
    values = df[Class].unique()
    for value in values:
        fraction = df[Class].value_counts()[value]/len(df[Class])
        entropy += -fraction*np.log2(fraction)
    return entropy
```

In [5]:

```
def find_entropy_attribute(df,attribute):
    Class = df.keys()[-1]
    target_variables = df[Class].unique()
    variables = df[attribute].unique()
    entropy2 = 0
    for variable in variables:
        entropy = 0
        for target_variable in target_variables:
            num = len(df[attribute][df[attribute]==variable][df[Class] ==target_variable])
            den = len(df[attribute][df[attribute]==variable])
            fraction = num/(den+eps)
            entropy += -fraction*log(fraction+eps)
        fraction2 = den/len(df)
        entropy2 += -fraction2*entropy
    return abs(entropy2)
```

In [6]:

```
def find_winner(df):
    Entropy_att = []
    IG = []
    for key in df.keys()[:-1]:
        IG.append(find_entropy(df)-find_entropy_attribute(df,key))
    return df.keys()[:-1][np.argmax(IG)]
```

In [7]:

```
def get_subtable(df, node,value):
    return df[df[node] == value].reset_index(drop=True)
```

In [8]:

```
def buildTree(df,tree=None):
    Class = df.keys()[ -1]
    node = find_winner(df)

    attValue = np.unique(df[node])
    if tree is None:
        tree={}
        tree[node] = {}
    for value in attValue:

        subtable = get_subtable(df,node,value)
        clValue,counts = np.unique(subtable['PlayTennis'],return_counts=True)

        if len(counts)==1:
            tree[node][value] = clValue[0]
        else:
            tree[node][value] = buildTree(subtable)
    return tree
```

In [15]:

```
t = buildTree(df)

for i,c in t.items():
    print(i)
    for j in c.items():
        print(j)
```

Outlook

```
('Overcast', 'Yes')
('Rainy', {'Windy': {'Strong': 'No', 'Weak': 'Yes'}})
('Sunny', {'Humidity': {'High': 'No', 'Normal': 'Yes'}})
```