

In [48]:

```
import math
import csv
```

In [55]:

```
data = pd.read_csv(r"C:\Users\Admin\Downloads\3-dataset.csv")
print(data)
```

	outlook	temperature	humidity	wind	answer
0	sunny	hot	high	weak	no
1	sunny	hot	high	strong	no
2	overcast	hot	high	weak	yes
3	rain	mild	high	weak	yes
4	rain	cool	normal	weak	yes
5	rain	cool	normal	strong	no
6	overcast	cool	normal	strong	yes
7	sunny	mild	high	weak	no
8	sunny	cool	normal	weak	yes
9	rain	mild	normal	weak	yes
10	sunny	mild	normal	strong	yes
11	overcast	mild	high	strong	yes
12	overcast	hot	normal	weak	yes
13	rain	mild	high	strong	no

In [56]:

```
class Node:
    def __init__(self):
        self.children = []
        self.value = ""
        self.isLeaf = False
        self.pred = ""
```

In [57]:

```
def entropy(examples):
    pos = 0.0
    neg = 0.0
    for _, row in examples.iterrows():
        if row["answer"] == "yes":
            pos += 1
        else:
            neg += 1
    if pos == 0.0 or neg == 0.0:
        return 0.0
    else:
        p = pos / (pos + neg)
        n = neg / (pos + neg)
        return -(p * math.log(p, 2) + n * math.log(n, 2))
```

In [58]:

```
def info_gain(examples, attr):
    uniq = np.unique(examples[attr])
```

```

#print ("\n",uniq)
gain = entropy(examples)
#print ("\n",gain)
for u in uniq:
    subdata = examples[examples[attr] == u]
    #print ("\n",subdata)
    sub_e = entropy(subdata)
    gain -= (float(len(subdata)) / float(len(examples))) * sub_e
    #print ("\n",gain)
return gain

```

In [59]:

```

def ID3(examples, attrs):
    root = Node()

    max_gain = 0
    max_feat = ""
    for feature in attrs:
        #print ("\n",examples)
        gain = info_gain(examples, feature)
        if gain > max_gain:
            max_gain = gain
            max_feat = feature
    root.value = max_feat
    #print ("\nMax feature attr",max_feat)
    uniq = np.unique(examples[max_feat])
    #print ("\n",uniq)
    for u in uniq:
        #print ("\n",u)
        subdata = examples[examples[max_feat] == u]
        #print ("\n",subdata)
        if entropy(subdata) == 0.0:
            newNode = Node()
            newNode.isLeaf = True
            newNode.value = u
            newNode.pred = np.unique(subdata["answer"])
            root.children.append(newNode)
        else:
            dummyNode = Node()
            dummyNode.value = u
            new_attrs = attrs.copy()
            new_attrs.remove(max_feat)
            child = ID3(subdata, new_attrs)
            dummyNode.children.append(child)
            root.children.append(dummyNode)

    return root

```

In [60]:

```
def printTree(root: Node, depth=0):
    for i in range(depth):
        print("\t", end="")
    print(root.value, end="")
    if root.isLeaf:
        print(" -> ", root.pred)
    print()
    for child in root.children:
        printTree(child, depth + 1)
```

In [61]:

```
def classify(root: Node, new):
    for child in root.children:
        if child.value == new[root.value]:
            if child.isLeaf:
                print ("Predicted Label for new example", new," is:",
child.pred)
                exit
            else:
                classify (child.children[0], new)
```

In [62]:

```
root = ID3(data, features)
print("Decision Tree is:")
printTree(root)
print ("-----")

new = {"outlook":"sunny", "temperature":"hot", "humidity":"normal",
"wind":"strong"}
classify (root, new)
Decision Tree is:
outlook
    overcast ->  ['yes']

    rain
        wind
            strong ->  ['no']

            weak ->  ['yes']

    sunny
        humidity
            high ->  ['no']

            normal ->  ['yes']

-----
```

```
Predicted Label for new example {'outlook': 'sunny', 'temperature': 'hot',  
'humidity': 'normal', 'wind': 'strong'} is: ['yes']
```

In []: