

17/3/25

Lab 3

```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
import math
import copy
```

```
dataset = pd.read_csv('content/weather.csv')
X = dataset.iloc[:, :].values
attribute = ['Outlook', 'Temp', 'Humidity', 'wind']
```

```
class Node(object):
```

```
    def __init__(self):
```

```
        self.value = None
```

```
        self.decision = None
```

```
        self.child = None
```

```
    def findEntropy(data, series):
```

```
        yes = 0
```

```
        no = 0
```

```
        ans = -1
```

```
        idx = len(data[0]) - 1
```

```
        entropy = 0
```

```
        for i in rows:
```

```
            if data[i][idx] == 'yes':
```

```
                yes = yes + 1
```

```
            else:
```

```
                no = no + 1
```

```
        n = yes / (yes + no)
```

```
        y = no / (yes + no)
```

```
        if n != 0 and y != 0:
```

$$\text{entropy} = -1 \times (\text{math.log2}(x) + y \times \text{math.log2}(y))$$

if $x == 1$:

ans = 1

if $y == 1$:

ans = 0

return entropy, ans

def findMaxGain(data, rows, columns):

maxGain = 0

actidx = -1

entropy, ans = findEntropy(data, rows)

if entropy == 0:

return MaxGain, actidx, ans

for j in columns:

mydict = {}

idx = j

for i in rows:

key = data[i][idx]

if key not in mydict:

mydict[key] = 1

else:

mydict[key] = mydict[key] + 1

gain = entropy

for key in mydict:

yes = 0

no = 0

for k in rows:

if data[k][j] == key:

if data[k][-1] == 'yes':

yes = yes + 1

else:

no = no + 1

$x = \text{yes} / (\text{yes} + \text{no})$
 $y = \text{no} / (\text{yes} + \text{no})$

if $x! = 0$ and $y! = 0$
 $\text{gain} += (\text{mydict}[\text{key}] * (\text{math.log2}(x) + \text{math.log2}(y)))$

if $\text{gain} > \text{maxGain}$:
 $\text{maxGain} = \text{gain}$
 $\text{setidx} = j$

return $\text{maxGain}, \text{setidx}, \text{ans}$

def buildTree(data, rows, columns):
 $\text{maxGain}, \text{idx}, \text{ans} = \text{findMaxGain}(X, \text{rows})$
 $\text{root} = \text{Node}()$
 $\text{root.children} = []$

if $\text{maxGain} == 0$:
 if $\text{ans} == 1$:
 $\text{root.value} = \text{'yes'}$
 else:
 $\text{root.value} = \text{'No'}$
 return root

$\text{root.value} = \text{attribute}[\text{idx}]$
 $\text{mydict} = \{\}$
 for i in rows :
 $\text{key} = \text{data}[i][\text{idx}]$
 if key not in mydict :
 $\text{mydict}[\text{key}] = 1$
 else:
 $\text{mydict}[\text{key}] += 1$

```
new_columns = copy.deepcopy(columns)
new_columns.remove(idx)
```

```
for key in mydict:
```

```
    new_rows = []
```

```
    for i in rows:
```

```
        if data[i][idx] == key:
```

```
            new_rows.append(i)
```

```
    temp = buildTree(data, new_rows, new_columns)
```

```
    temp.decision = key
```

```
    root.children.append(temp)
```

```
return root
```

```
def plotTree(root, graph=None, parent_name=None,
             node_id=0):
```

```
    if graph is None:
```

```
        graph = nx.DiGraph()
```

```
    node_name = f"Node {node_id}"
```

```
    graph.add_node(node_name, label=root.value)
```

```
    if parent_name:
```

```
        graph.add_edge(parent_name, node_name,
                       label=root.decision)
```

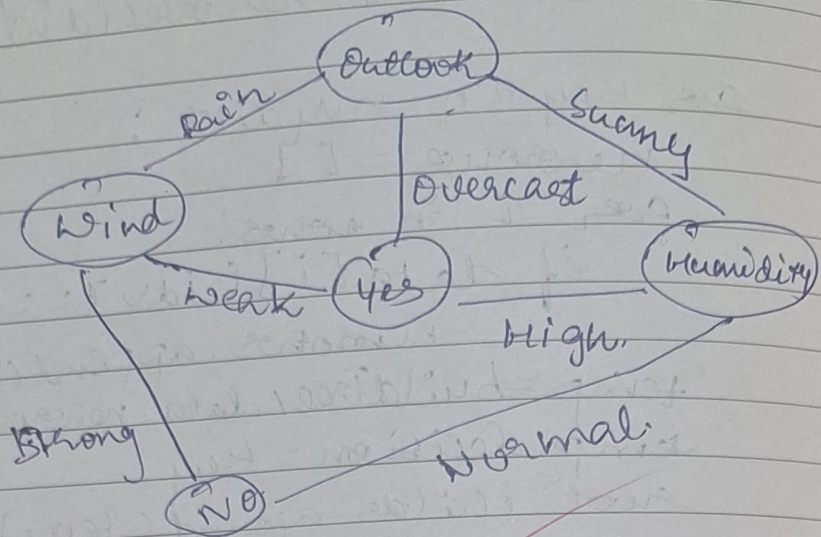
```
    for i, child in enumerate(root.children):
```

```
        node_id += 1
```

```
        plotTree(child, graph, node_name, node_id)
```

```
    return graph
```

```
plotTree(root)
```

Sreelal SB
 17/3/25