

Lab - 2

ID3 (Iterative Dichotomiser 3) using Decision Tree

PAGE NO:
 DATE: 12/3/25

Lab - 2 (ID3)
(Iterative Dichotomiser 3)

```
import pandas as pd
import numpy as np

def entropy(target_col):
    cl, count = np.unique(target_col, return_counts=True)
    entropy = np.sum([(count[i] / np.sum(counts)) * np.log2(counts[i] / np.sum(counts))])

def InfoGain(data, Split_attr, target = "class"):
    total = entropy(data[target])
    vals, count = np.unique(data[Split_attr], return_counts=True)
    weighted = np.sum([(count[i] / np.sum(counts)) * entropy(data.where(data[Split_attr] == vals[i]).dropna()[target])])
    info = total - weighted
    return info

def ID3():
    if len(np.unique(data[target])) <= 1:
        return np.unique(data[target])[0]

    elif len(data) == 0:
        return np.unique(original[target])
        [np.argmax(np.unique(original[target], return_counts=True)[1])]

    elif len(features) == 0:
        return parent
```

else :

~~parent~~ :

for value in np.unique(data[best]):

value = value

Sub_data = data.where(data[best] == value)

dropna()

tree[best - feature][value] = Subtree

data = pd.read_csv("Content/Weather.csv")

features = data.columns[:-1]

tree = D3(data, data, features)

tree

→ { 'Outlook' : { 'overcast' : 'yes',
 'rainy' : { 'windy' : { 'false' : yes, 'true' : no },
 'Sunny' : { 'humidity' : { 'high' : 'no', 'normal' :
 'yes' } } }

~~6/6~~
 12-3-25