LAB-2: Implement ID3 algorithm using decision tree learning with wether dataset.

```
import pandas as pd
import numpy as np

df = pd.read_csv ("weather.csv")

def entropy(target):
    class_count = target.value_counts()
    probs = class_count/ len(target)
    return -np.sum(probs * np.log2(probs))

def info_gain(data, feature, target):
    entropy_before = entropy (target)
    feature_values = data[feature].unique()
    weighted_entropy = 0
    for value in feature_values:
        subset = target[data[feature] == value]
        weighted_entropy += (len(subset)/len(target))*entropy(su
    return entropy_before - weighted-entropy

def  print_entropy_and_gain(data, features, target):
    print(" Entropy and info Gain for each feature:")
    for feature in features:
        gain = info_gain(data, features, target)
        ent = entropy (target)
        print(f" Feature:{feature}|Entropy:{ent:4f}|
              Information Gain:{gain:4f}")
```

```
def build_tree (data , target , features):
    if len(target.unique()) == 1:
        return target.iloc[0]

    if len (features) == 0:
        return target.mode()[0]

    gains = {feature : info_gain (data, featue, target)
             for feature in feature}

    best_feature = max(gains , key=gains.get)
    tree = {best_feature : {}}
    feature_values = data[best_featra].unique()
    For value in feature_values :
        subset data = data [ data[best feature]==value]
        subset_target = target[data [best feature]==value

        remaining - features = [f for f in features if f!=
                                best_feature]

        subtree = build_tree (subset_data , subset_target
                    remaining_features)

        tree[best_feature][value] = subtree


def print_tree (tree , indent = " "):
    if isinstance (tree, dict):
        for feature , branches in tree.Items():
            print(f"{indent}{feature}:")
            for value , subtree in branches.items():
                print (f"{indent} {value} ->", end=" ")
                print_tree (subtree , indent+"    ")

    else :
        print (f"{indent}{tree} ")
```

```
target = df ['Decision']
features = ['Outlook', 'Temperature', 'Humidity', 'Wind']
print_entropy_and_gain (df, features, target)
tree = build_tree (df, target, features)
print ("Decision Tree:")
print_tree (tree, indent=" ")
```

O/P:

```
Entropy and Information Gain for each feature:
Feature: Outlook | Entropy: 0.9403 | Information Gain: 0.2
Feature: Temperature | Entropy: 0.9403 | Information Gain: 0.02
Feature: Humidity | Entropy: 0.9403 | Information Gain: 0.15
Feature: Wind | Entropy: 0.9403 | Information Gain: 0.04

Decision Tree:
    Outlook:
        Sunny  ->      Humidity:
            High  ->      No
            Normal  ->     Yes
        Overcast  ->   Yes
        Rainy  ->     Wind:
            Weak  ->     Yes
            Strong  ->    No
```

12/3/25