LAB - 2

Use an appropriate dataset to build decision tree (ID3)

```
import pandas as pd
import numpy as np

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

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

def information_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(subset)

    return entropy_before - weighted_entropy

def print_entropy_and_gain(data, feature, target):
    print("Entropy and Information gain for
          each feature")
    for feature in features:
        gain = information_gain(data, feature, 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: information_gain(data, feature,
            target) for feature in features}
    best_feature = max(gains, key=gains.get)

    tree = {best_feature: {}}
    feature_values = data[best_feature].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

    return tree
```

```
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['Play Tennis']
features = ['Outlook', 'Temperature', 'Humidity', 'Windy']

print_entropy_and_gain(df, features, target)
tree = build_tree(df, target, features)
print("\nDecision tree")
print_tree(tree, indent="  ")
```

output:-

Entropy and Information gain for each feature

Feature: Outlook | Entropy: 0.9403 | Information gain: 0.241

Feature: Temperature | Entropy: 0.9403 | Information gain: 0.029

Feature: Humidity | Entropy: 0.9403 | Information gain: 0.151

Feature: Windy | Entropy: 0.9403 | Information gain: 0.048

Decision tree

Outlook:

Sunny → Humidity:

High → No

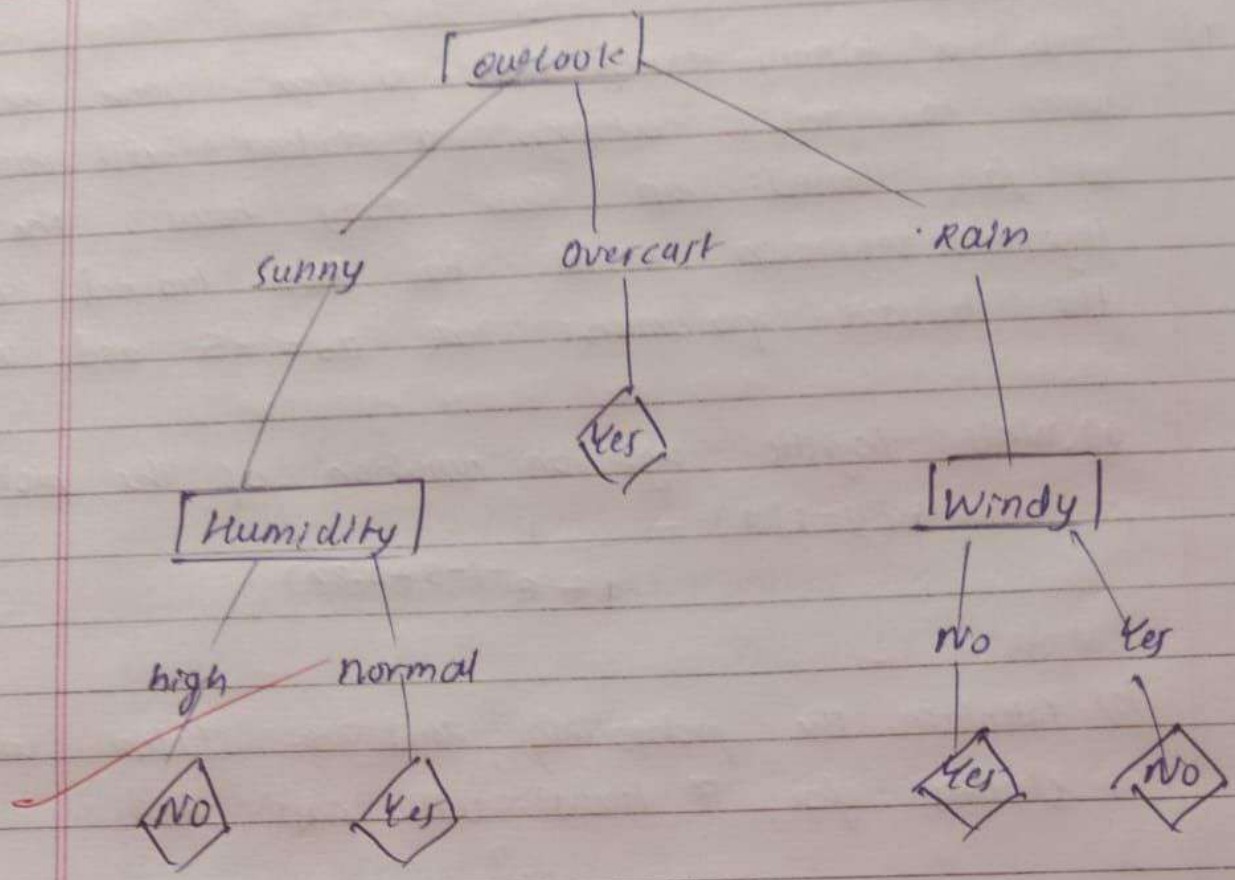
Normal → Yes

Overcast → Yes

Rain → Windy:

No → Yes

Yes → No


Outlook
Sunny
Overcast
Rain
Yes
Humidity
Windy
high
normal
No
Yes
NO
Yes
Yes
NO

12/3/25