```
import numpy as np
import pandas as pd
from collections import Counter
import json
# Function to calculate entropy
def entropy(data):
            labels = data.iloc[:, -1]
             label_counts = Counter(labels)
            total = len(labels)
            return -sum((count/total) * np.log2(count/total) for count in label counts.values())
# Function to calculate information gain
def information_gain(data, attribute):
             total_entropy = entropy(data)
            values = data[attribute].unique()
             weighted_entropy = sum((len(subset) / len(data)) * entropy(subset)
                                                                                      for value in values
                                                                                     for subset in [data[data[attribute] == value]])
            return total_entropy - weighted_entropy
# Function to choose the best attribute
def best_attribute(data):
            attributes = data.columns[:-1]
             return max(attributes, key=lambda attr: information_gain(data, attr))
# Function to build the decision tree recursively
def id3(data, features):
            labels = data.iloc[:, -1]
             if len(set(labels)) == 1:
                        return labels.iloc[0]
            if len(features) == 0:
                        return labels.mode()[0]
            best attr = best attribute(data)
            tree = {best_attr: {}}
             for value in data[best attr].unique():
                         subset = data[data[best_attr] == value].drop(columns=[best_attr])
                          tree[best_attr][value] = id3(subset, subset.columns[:-1])
             return tree
# Example dataset
data = pd.DataFrame({
            'Outlook': ['Sunny', 'Sunny', 'Overcast', 'Rain', 'Rain', 'Rain', 'Overcast', 'Sunny', 'Sunny', 'Rain', 'Sunny', 'Overcast', 'Overcas 'Temperature': ['Hot', 'Hot', 'Hot', 'Mild', 'Cool', 'Cool', 'Mild', 'Cool', 'Mild', 'Mild', 'Mild', 'Hot', 'Mild'], 'Humidity': ['High', 'High', 'High', 'High', 'Normal', 'Normal', 'High', 'Normal', 'Normal', 'Normal', 'High', 'Normal', 
             'Wind': ['Weak', 'Strong', 'Weak', 'Weak', 'Strong', 'Strong', 'Weak', 'Weak', 'Weak', 'Strong', 'Weak', 'Strong', 'Yes', 'Yes',
})
# Build the decision tree
decision_tree = id3(data, list(data.columns[:-1]))
print(decision tree)
 Type ("Outlook': {'Sunny': {'Humidity': {'High': 'No', 'Normal': 'Yes'}}, 'Overcast': 'Yes', 'Rain': {'Wind': {'Weak': 'Yes', 'Strong':
```