Lab-2

```
import pandas as pd
from collections import counter.

def entropy(data):
    labels = data['label'].tolist()
    counts = counter(labels)
    probabilities = [count / len(labels) for count
                         in counts.values()]
    entropy_value = sum(p*math.log2(p) for p
                         in probabilities if p>0)
    return entropy_value

def gain(data, feature):
    initial_entropy = entropy(data)
    feature_values = data[feature].unique()
    weighted_entropy = 0
    for value in feature.values:
        subset = data[data[feature] == value]
        weighted_entropy += (len(subset) / len(data))
                             * entropy(subset)
    return initial_entropy - weighted_entropy

def id3(data, features, target_attribute):
    if len(data['label'].unique()) == 1:
        return data['label'].iloc[0]

    if len(features) == 0:
        return data['label'].value_counts().
                                      index[0]

    bet_feature = max(features, key = lambda
        feature: gain(data, feature))
```

```
tree = {best_feature: {}}
feature = [f for f in features if f != best_feature]

for value in data[best_feature].unique():
    subset = data[data[best_feature] == value]
                .drop(columns = [best_feature])
    if len(subset) == 0:
        tree[best_feature][value] = data['label'].
                value_counts().index[0]
    else:
        tree[best_feature][value] =
                    id3(subset, features, target_attribute)
return tree.

df = pd.read_csv('/content/weather.csv')
feature = ['outlook', 'temperature', 'humidity', 'wind']
target_attribute = 'label'
decision_tree = id3(df, feature, target_attribute)
decision_tree
```

O/P:

```
{'outlook' : {'sunny' : {'humidity' : {'high' : 'no',
                                       'normal' : 'yes'}},

{ 'overcast' = 'yes',
  'rainy' : {'wind' : {'weak' : 'yes', 'strong' : 'no'}}}}
```