In [7]:

```python
import pandas as pd
import math
df = pd.read_csv('PlayTennis.csv')
print("\n Input Data Set is:\n", df)
t = df.keys()[-1]
print('Target Attribute is: ', t)
attribute_names = list(df.keys())
attribute_names.remove(t)
print('Predicting Attributes: ', attribute_names)
def entropy(probs):
    return sum( [-prob*math.log(prob, 2) for prob in probs])
def entropy_of_list(ls,value):
    from collections import Counter
    cnt = Counter(x for x in ls)
    print('Target attribute class count(Yes/No)=',dict(cnt))
    total_instances = len(ls)
    print("Total no of instances/records associated with {0} is: {1}".format(value,tota
    probs = [x / total_instances for x in cnt.values()]
    print("Probability of Class {0} is: {1:.4f}".format(min(cnt),min(probs)))
    print("Probability of Class {0} is: {1:.4f}".format(max(cnt),max(probs)))
    return entropy(probs)
def information_gain(df, split_attribute, target_attribute,battr):
    print("\n\n-----Information Gain Calculation of ",split_attribute, " --------")
    df_split = df.groupby(split_attribute)
    glist=[]
    for gname,group in df_split:
        print('Grouped Attribute Values \n',group)
        glist.append(gname)
    glist.reverse()
    nobs = len(df.index) * 1.0
    df_agg1=df_split.agg({target_attribute:lambda x:entropy_of_list(x, glist.pop())})
    df_agg2=df_split.agg({target_attribute :lambda x:len(x)/nobs})
    df_agg1.columns=['Entropy']
    df_agg2.columns=['Proportion']
    new_entropy = sum( df_agg1['Entropy'] * df_agg2['Proportion'])
    if battr !='S':
        old_entropy = entropy_of_list(df[target_attribute],'S-'+df.iloc[0][df.columns.
    else:
        old_entropy = entropy_of_list(df[target_attribute],battr)
    return old_entropy - new_entropy
def id3(df, target_attribute, attribute_names, default_class=None,default_attr='S'):
    from collections import Counter
    cnt = Counter(x for x in df[target_attribute])
    if len(cnt) == 1:
        return next(iter(cnt))
    elif df.empty or (not attribute_names):
        return default_class
    else:
        default_class=max(cnt.keys())
        gainz=[]
        for attr in attribute_names:
            ig=information_gain(df,attr,target_attribute,default_attr)
            gainz.append(ig)
            print('information gain of',attr,'is:',ig)
        index_of_max = gainz.index(max(gainz))
        best_attr = attribute_names[index_of_max]
        print("\nAttribute with the maximum gain is: ", best_attr)
        tree = {best_attr:{}}
        remaining_attribute_names =[i for i in attribute_names if i != best_attr]
        for attr_val, data_subset in df.groupby(best_attr):
            subtree = id3(data_subset,target_attribute,remaining_attribute_names,defaul
            tree[best_attr][attr_val] = subtree
        return tree
    from pprint import pprint
tree = id3(df,t,attribute_names)
print("\nThe Resultant Decision Tree is:")
print(tree)
def classify(instance, tree,default=None):
    attribute = next(iter(tree))
    if instance[attribute] in tree[attribute].keys():
        result = tree[attribute][instance[attribute]]
        if isinstance(result, dict):
            return classify(instance, result)
        else:
            return result
    else:
        return default
df_new=pd.read_csv('PlayTennisTest.csv')
df_new['predicted'] = df_new.apply(classify, axis=1, args=(tree,'?'))
print(df_new)
```

```
Probability of Class no is: 0.4000
Probability of Class yes is: 0.6000
Target attribute class count(Yes/No)= {'no': 5, 'yes': 9}
Total no of instances/records associated with S is: 14
Probability of Class no is: 0.3571
Probability of Class yes is: 0.6429
information gain of outlook is: 0.2467498197744391


-----Information Gain Calculation of  temperature  --------
Grouped Attribute Values
     outlook temperature humiduty    wind playTennis
4      rain        cool    normal    weak        yes
5      rain        cool    normal  strong         no
6  overcast        cool    normal  strong        yes
8     sunny        cool    normal    weak        yes
Grouped Attribute Values
      outlook temperature humiduty    wind playTennis
0      sunny         hot     high    weak          no
1      sunny         hot     high  strong          no
```

In [ ]:

In [ ]: