PAGE NO : 5
DATE : 27/10/20
EXP.NO. : 3

3 Write a program to demonstrate the working of the decision tree based ID3 algorithm. Use an appropriate dataset for building the decision tree & apply this knowledge to classify a new sample.

```
import pandas as pd
from pandas import DataFrame
df-tennis = pd.read-csv('c:/users/lenovo/
        Desktop/4MT16CS060-Prajwal/enjoysport
                .csv')
attribute_names = list(df-tennis.columns)
attribute_names.remove('Play Tennis')
print(attribute_names)

def entropy_of_list(lst):
    from collections import counter
    count = Counter(x for x in lst)
    num_instances = len(lst) * 1
    probs = [x/num_instances for x
        in count.values()]
    return entropy(probs)

def entropy(probs):
    import math
```

ARUN'S

PAGE NO : 6
DATE :
EXP.NO. : 3

```
    return sum ( [-prob * math.log(prob,2)
                  for prob in probs])
total_entropy = entropy_of_list(df_tennis
                  ['Play Tennis'])
def information_gain(df, split_attribute_name,
        target_attribute_name, trace=0):
    df_split = df.groupby(split_attribute_name)
    nobs = len(df.index) * 1
    df_agg_ent = columns = [df.
    df_agg_ent = df_split.agg({target_
        attribute_name :
            [entropy_of_list, lambda x: len(x)/nobs]})
    df_agg_ent.columns = ['Entropy'] * df_agg_ent
                  ['Probobservations'])
new_entropy = sum(df_agg_ent['Entropy']
        * df_agg_ent['probobservations'])
old_entropy = entropy_of_list(df[target_
        attribute_name])
print('Split attribute_name', 'IG:',
            old_entropy, -new_entropy)
return old_entropy - new_entropy

def id3(df, target_attribute name, attribute_
    names, default_class = None):
    from collections import Counter
```

ARUN'S

```
count = Counter ([x for x in df[target-
        attribute_names] for
    attr in attribute_names]
index_of_max = gain.index[max(gain))
best_attr = attribute_names[index_of_max]
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_attribut
_name, remaining_attribute_names,
        default_class)
    tree[best_attr][attr_val] = subtree
return tree

from pprint import pprint
tree = id3[df_tennis, 'Play Tennis',
        attribute names)
print("In The Resultant Decision Tree is:
        \n")
pprint(tree).
```

output :

['outlook', 'Temperature', 'Humidity', 'wind']

outlook IG : 0.246749819774 4391
Temperature IG : 0.029922256589546 47
Humidity IG : 0.151835501362341 36
wind IG : 0.048127030408269 27

Temperature IG : 0.019973094021974 89
Humidity IG : 0.019973094021974 89
wind IG : 0.970950599454 6686

Temperature IG : 0.570950594454 6686
Humidity IG : 0.970950594454 6686
wind. IG : 0.019973094021974 89

The Resultant Decision tree is :

{'outlook' : {'overcast' : 'yes',
'Rain' : {'wind' : {'strong' : 'NO',
'weak' : 'yes'}},
'sunny' {'Humidity' : {'High' : 'NO',
'Normal' : 'yes'}}}}