

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

Import Play Tennis Data

```
In [3]: import pandas as pd
from pandas import DataFrame
df_tennis = DataFrame.from_csv('C:\\Users\\Dr.Thyagaraju\\Desktop\\Data\\PlayTennis.csv')
df_tennis
```

Out[3]:

	PlayTennis	Outlook	Temperature	Humidity	Wind
0	No	Sunny	Hot	High	Weak
1	No	Sunny	Hot	High	Strong
2	Yes	Overcast	Hot	High	Weak
3	Yes	Rain	Mild	High	Weak
4	Yes	Rain	Cool	Normal	Weak
5	No	Rain	Cool	Normal	Strong
6	Yes	Overcast	Cool	Normal	Strong
7	No	Sunny	Mild	High	Weak
8	Yes	Sunny	Cool	Normal	Weak
9	Yes	Rain	Mild	Normal	Weak
10	Yes	Sunny	Mild	Normal	Strong
11	Yes	Overcast	Mild	High	Strong
12	Yes	Overcast	Hot	Normal	Weak

	PlayTennis	Outlook	Temperature	Humidity	Wind
13	No	Rain	Mild	High	Strong

Entropy of the Training Data Set

```
In [77]: def entropy(probs): # Calculate the Entropy of given probability
import math
return sum( [-prob*math.log(prob, 2) for prob in probs] )

def entropy_of_list(a_list): # Entropy calculation of list of discrete
values (YES/NO)
from collections import Counter
cnt = Counter(x for x in a_list)
print("No and Yes Classes:",a_list.name,cnt)
num_instances = len(a_list)*1.0
probs = [x / num_instances for x in cnt.values()]
return entropy(probs) # Call Entropy:

# The initial entropy of the YES/NO attribute for our dataset.
#print(df_tennis['PlayTennis'])
total_entropy = entropy_of_list(df_tennis['PlayTennis'])
print("Entropy of given PlayTennis Data Set:",total_entropy)

No and Yes Classes: PlayTennis Counter({'Yes': 9, 'No': 5})
Entropy of given PlayTennis Data Set: 0.9402859586706309
```

Information Gain of Attributes

```
In [78]: def information_gain(df, split_attribute_name, target_attribute_name, t
race=0):
print("Information Gain Calculation of ",split_attribute_name)

...
Takes a DataFrame of attributes, and quantifies the entropy of a ta
```

```

rget
    attribute after performing a split along the values of another attribute.
    ...

    # Split Data by Possible Vals of Attribute:
    df_split = df.groupby(split_attribute_name)
    #print(df_split.groups)
    for name, group in df_split:
        print(name)
        print(group)

    # Calculate Entropy for Target Attribute, as well as
    # Proportion of Obs in Each Data-Split
    nobs = len(df.index) * 1.0
    #print("NOBS", nobs)
    df_agg_ent = df_split.agg({target_attribute_name : [entropy_of_list
, lambda x: len(x)/nobs] })[target_attribute_name]
    #print("DFAGGENT", df_agg_ent)
    df_agg_ent.columns = ['Entropy', 'PropObservations']
    #if trace: # helps understand what fxn is doing:
    #    print(df_agg_ent)

    # Calculate Information Gain:
    new_entropy = sum( df_agg_ent['Entropy'] * df_agg_ent['PropObservations'] )
    old_entropy = entropy_of_list(df[target_attribute_name])
    return old_entropy - new_entropy

print('Info-gain for Outlook is :'+str( information_gain(df_tennis, 'Outlook', 'PlayTennis')),"\n")
print('\n Info-gain for Humidity is: ' + str( information_gain(df_tennis, 'Humidity', 'PlayTennis')),"\n")
print('\n Info-gain for Wind is:' + str( information_gain(df_tennis, 'Wind', 'PlayTennis')),"\n")
print('\n Info-gain for Temperature is:' + str( information_gain(df_tennis, 'Temperature', 'PlayTennis')),"\n")

```

Information Gain Calculation of Outlook

```

Overcast
  PlayTennis Outlook Temperature Humidity Wind
2      Yes Overcast      Hot      High Weak
6      Yes Overcast      Cool     Normal Strong
11     Yes Overcast      Mild     High Strong
12     Yes Overcast      Hot      Normal Weak
Rain
  PlayTennis Outlook Temperature Humidity Wind
3      Yes Rain      Mild     High Weak
4      Yes Rain      Cool     Normal Weak
5      No  Rain      Cool     Normal Strong
9      Yes Rain      Mild     Normal Weak
13     No  Rain      Mild     High Strong
Sunny
  PlayTennis Outlook Temperature Humidity Wind
0      No  Sunny      Hot      High Weak
1      No  Sunny      Hot      High Strong
7      No  Sunny      Mild     High Weak
8      Yes Sunny      Cool     Normal Weak
10     Yes Sunny      Mild     Normal Strong
No and Yes Classes: PlayTennis Counter({'Yes': 4})
No and Yes Classes: PlayTennis Counter({'Yes': 3, 'No': 2})
No and Yes Classes: PlayTennis Counter({'No': 3, 'Yes': 2})
No and Yes Classes: PlayTennis Counter({'Yes': 9, 'No': 5})
Info-gain for Outlook is :0.246749819774

```

Information Gain Calculation of Humidity

High

```

  PlayTennis Outlook Temperature Humidity Wind
0      No  Sunny      Hot      High Weak
1      No  Sunny      Hot      High Strong
2      Yes Overcast      Hot      High Weak
3      Yes Rain      Mild     High Weak
7      No  Sunny      Mild     High Weak
11     Yes Overcast      Mild     High Strong
13     No  Rain      Mild     High Strong

```

Normal

```

  PlayTennis Outlook Temperature Humidity Wind
4      Yes Rain      Cool     Normal Weak

```

	PlayTennis	Outlook	Temperature	Humidity	Wind
5	No	Rain	Cool	Normal	Strong
6	Yes	Overcast	Cool	Normal	Strong
8	Yes	Sunny	Cool	Normal	Weak
9	Yes	Rain	Mild	Normal	Weak
10	Yes	Sunny	Mild	Normal	Strong
12	Yes	Overcast	Hot	Normal	Weak

No and Yes Classes: PlayTennis Counter({'No': 4, 'Yes': 3})
 No and Yes Classes: PlayTennis Counter({'Yes': 6, 'No': 1})
 No and Yes Classes: PlayTennis Counter({'Yes': 9, 'No': 5})

Info-gain for Humidity is: 0.151835501362

Information Gain Calculation of Wind

Strong

	PlayTennis	Outlook	Temperature	Humidity	Wind
1	No	Sunny	Hot	High	Strong
5	No	Rain	Cool	Normal	Strong
6	Yes	Overcast	Cool	Normal	Strong
10	Yes	Sunny	Mild	Normal	Strong
11	Yes	Overcast	Mild	High	Strong
13	No	Rain	Mild	High	Strong

Weak

	PlayTennis	Outlook	Temperature	Humidity	Wind
0	No	Sunny	Hot	High	Weak
2	Yes	Overcast	Hot	High	Weak
3	Yes	Rain	Mild	High	Weak
4	Yes	Rain	Cool	Normal	Weak
7	No	Sunny	Mild	High	Weak
8	Yes	Sunny	Cool	Normal	Weak
9	Yes	Rain	Mild	Normal	Weak
12	Yes	Overcast	Hot	Normal	Weak

No and Yes Classes: PlayTennis Counter({'No': 3, 'Yes': 3})
 No and Yes Classes: PlayTennis Counter({'Yes': 6, 'No': 2})
 No and Yes Classes: PlayTennis Counter({'Yes': 9, 'No': 5})

Info-gain for Wind is:0.0481270304083

Information Gain Calculation of Temperature

Cool

	PlayTennis	Outlook	Temperature	Humidity	Wind
4	Yes	Rain	Cool	Normal	Weak
5	No	Rain	Cool	Normal	Strong
6	Yes	Overcast	Cool	Normal	Strong
8	Yes	Sunny	Cool	Normal	Weak

Hot

	PlayTennis	Outlook	Temperature	Humidity	Wind
0	No	Sunny	Hot	High	Weak
1	No	Sunny	Hot	High	Strong
2	Yes	Overcast	Hot	High	Weak
12	Yes	Overcast	Hot	Normal	Weak

Mild

	PlayTennis	Outlook	Temperature	Humidity	Wind
3	Yes	Rain	Mild	High	Weak
7	No	Sunny	Mild	High	Weak
9	Yes	Rain	Mild	Normal	Weak
10	Yes	Sunny	Mild	Normal	Strong
11	Yes	Overcast	Mild	High	Strong
13	No	Rain	Mild	High	Strong

No and Yes Classes: PlayTennis Counter({'Yes': 3, 'No': 1})
 No and Yes Classes: PlayTennis Counter({'No': 2, 'Yes': 2})
 No and Yes Classes: PlayTennis Counter({'Yes': 4, 'No': 2})
 No and Yes Classes: PlayTennis Counter({'Yes': 9, 'No': 5})

Info-gain for Temperature is:0.029222565659

```
from collections import Counter
cnt = Counter(x for x in df_tennis['PlayTennis'])
print(cnt)
print(cnt.values())
print(max(cnt.values()))
print(max(cnt.keys()))
print(next(iter(cnt)))
if len(cnt) == 1:
    print(cnt.keys()[0])
elif df_tennis.empty or (not attribute_names):
    print(None)
else:
    index_of_max = max(cnt.values())
    print(index_of_max)
    default_class = max(cnt.keys())
    print(default_class)
```

ID3 Algorithm

```
In [79]: def id3(df, target_attribute_name, attribute_names, default_class=None)
```

```

):

    ## Tally target attribute:
    from collections import Counter
    cnt = Counter(x for x in df[target_attribute_name]) # class of YES /
NO

    ## First check: Is this split of the dataset homogeneous?
    if len(cnt) == 1:
        return next(iter(cnt))

    ## Second check: Is this split of the dataset empty?
    # if yes, return a default value
    elif df.empty or (not attribute_names):
        return default_class

    ## Otherwise: This dataset is ready to be divvied up!
    else:
        # Get Default Value for next recursive call of this function:
        default_class = max(cnt.keys()) #[index_of_max] # most common v
alue of target attribute in dataset

        # Choose Best Attribute to split on:
        gainz = [information_gain(df, attr, target_attribute_name) for
attr in attribute_names]
        index_of_max = gainz.index(max(gainz))
        best_attr = attribute_names[index_of_max]

        # Create an empty tree, to be populated in a moment
        tree = {best_attr: {}}
        remaining_attribute_names = [i for i in attribute_names if i !=
best_attr]

        # Split dataset
        # On each split, recursively call this algorithm.
        # populate the empty tree with subtrees, which
        # are the result of the recursive call
        for attr_val, data_subset in df.groupby(best_attr):
            subtree = id3(data_subset,

```

```

        target_attribute_name,
        remaining_attribute_names,
        default_class)
    tree[best_attr][attr_val] = subtree
    return tree

```

Predicting Attributes

```

In [82]: # Get Predictor Names (all but 'class')
attribute_names = list(df_tennis.columns)
print("List of Attributes:", attribute_names)
attribute_names.remove('PlayTennis') #Remove the class attribute
print("Predicting Attributes:", attribute_names)

```

List of Attributes: ['PlayTennis', 'Outlook', 'Temperature', 'Humidity', 'Wind']
 Predicting Attributes: ['Outlook', 'Temperature', 'Humidity', 'Wind']

Tree Construction

```

In [85]: # Run Algorithm:
from pprint import pprint
tree = id3(df_tennis, 'PlayTennis', attribute_names)
print("\n\nThe Resultant Decision Tree is :\n")
pprint(tree)

```

Information Gain Calculation of Outlook

	PlayTennis	Outlook	Temperature	Humidity	Wind
Overcast					
2	Yes	Overcast	Hot	High	Weak
6	Yes	Overcast	Cool	Normal	Strong
11	Yes	Overcast	Mild	High	Strong
12	Yes	Overcast	Hot	Normal	Weak
Rain					
	PlayTennis	Outlook	Temperature	Humidity	Wind

3	Yes	Rain	Mild	High	Weak
4	Yes	Rain	Cool	Normal	Weak
5	No	Rain	Cool	Normal	Strong
9	Yes	Rain	Mild	Normal	Weak
13	No	Rain	Mild	High	Strong

Sunny

	PlayTennis	Outlook	Temperature	Humidity	Wind
0	No	Sunny	Hot	High	Weak
1	No	Sunny	Hot	High	Strong
7	No	Sunny	Mild	High	Weak
8	Yes	Sunny	Cool	Normal	Weak
10	Yes	Sunny	Mild	Normal	Strong

No and Yes Classes: PlayTennis Counter({'Yes': 4})
No and Yes Classes: PlayTennis Counter({'Yes': 3, 'No': 2})
No and Yes Classes: PlayTennis Counter({'No': 3, 'Yes': 2})
No and Yes Classes: PlayTennis Counter({'Yes': 9, 'No': 5})

Information Gain Calculation of Temperature

Cool

	PlayTennis	Outlook	Temperature	Humidity	Wind
4	Yes	Rain	Cool	Normal	Weak
5	No	Rain	Cool	Normal	Strong
6	Yes	Overcast	Cool	Normal	Strong
8	Yes	Sunny	Cool	Normal	Weak

Hot

	PlayTennis	Outlook	Temperature	Humidity	Wind
0	No	Sunny	Hot	High	Weak
1	No	Sunny	Hot	High	Strong
2	Yes	Overcast	Hot	High	Weak
12	Yes	Overcast	Hot	Normal	Weak

Mild

	PlayTennis	Outlook	Temperature	Humidity	Wind
3	Yes	Rain	Mild	High	Weak
7	No	Sunny	Mild	High	Weak
9	Yes	Rain	Mild	Normal	Weak
10	Yes	Sunny	Mild	Normal	Strong
11	Yes	Overcast	Mild	High	Strong
13	No	Rain	Mild	High	Strong

No and Yes Classes: PlayTennis Counter({'Yes': 3, 'No': 1})
No and Yes Classes: PlayTennis Counter({'No': 2, 'Yes': 2})

```

No and Yes Classes: PlayTennis Counter({'Yes': 4, 'No': 2})
No and Yes Classes: PlayTennis Counter({'Yes': 9, 'No': 5})
Information Gain Calculation of Humidity
High
  PlayTennis Outlook Temperature Humidity Wind
0      No    Sunny          Hot    High   Weak
1      No    Sunny          Hot    High  Strong
2      Yes  Overcast          Hot    High   Weak
3      Yes    Rain          Mild    High   Weak
7      No    Sunny          Mild    High   Weak
11     Yes  Overcast          Mild    High  Strong
13     No    Rain           Mild    High  Strong
Normal
  PlayTennis Outlook Temperature Humidity Wind
4      Yes    Rain           Cool   Normal  Weak
5      No     Rain           Cool   Normal  Strong
6      Yes  Overcast          Cool   Normal  Strong
8      Yes    Sunny          Cool   Normal  Weak
9      Yes    Rain           Mild   Normal  Weak
10     Yes    Sunny          Mild   Normal  Strong
12     Yes  Overcast          Hot    Normal  Weak
No and Yes Classes: PlayTennis Counter({'No': 4, 'Yes': 3})
No and Yes Classes: PlayTennis Counter({'Yes': 6, 'No': 1})
No and Yes Classes: PlayTennis Counter({'Yes': 9, 'No': 5})
Information Gain Calculation of Wind
Strong
  PlayTennis Outlook Temperature Humidity Wind
1      No    Sunny          Hot    High  Strong
5      No     Rain           Cool   Normal  Strong
6      Yes  Overcast          Cool   Normal  Strong
10     Yes    Sunny          Mild   Normal  Strong
11     Yes  Overcast          Mild    High  Strong
13     No     Rain           Mild    High  Strong
Weak
  PlayTennis Outlook Temperature Humidity Wind
0      No    Sunny          Hot    High  Weak
2      Yes  Overcast          Hot    High  Weak
3      Yes    Rain          Mild    High  Weak
4      Yes    Rain          Cool   Normal  Weak

```

```

7          No      Sunny      Mild      High  Weak
8          Yes     Sunny      Cool      Normal Weak
9          Yes     Rain       Mild      Normal Weak
12         Yes     Overcast    Hot       Normal Weak
No and Yes Classes: PlayTennis Counter({'No': 3, 'Yes': 3})
No and Yes Classes: PlayTennis Counter({'Yes': 6, 'No': 2})
No and Yes Classes: PlayTennis Counter({'Yes': 9, 'No': 5})
Information Gain Calculation of Temperature
Cool
    PlayTennis Outlook Temperature Humidity Wind
4          Yes     Rain       Cool      Normal Weak
5          No      Rain       Cool      Normal Strong
Mild
    PlayTennis Outlook Temperature Humidity Wind
3          Yes     Rain       Mild      High  Weak
9          Yes     Rain       Mild      Normal Weak
13         No      Rain       Mild      High  Strong
No and Yes Classes: PlayTennis Counter({'Yes': 1, 'No': 1})
No and Yes Classes: PlayTennis Counter({'Yes': 2, 'No': 1})
No and Yes Classes: PlayTennis Counter({'Yes': 3, 'No': 2})
Information Gain Calculation of Humidity
High
    PlayTennis Outlook Temperature Humidity Wind
3          Yes     Rain       Mild      High  Weak
13         No      Rain       Mild      High  Strong
Normal
    PlayTennis Outlook Temperature Humidity Wind
4          Yes     Rain       Cool      Normal Weak
5          No      Rain       Cool      Normal Strong
9          Yes     Rain       Mild      Normal Weak
No and Yes Classes: PlayTennis Counter({'Yes': 1, 'No': 1})
No and Yes Classes: PlayTennis Counter({'Yes': 2, 'No': 1})
No and Yes Classes: PlayTennis Counter({'Yes': 3, 'No': 2})
Information Gain Calculation of Wind
Strong
    PlayTennis Outlook Temperature Humidity Wind
5          No      Rain       Cool      Normal Strong
13         No      Rain       Mild      High  Strong
Weak

```

```

    PlayTennis Outlook Temperature Humidity Wind
3         Yes    Rain          Mild    High Weak
4         Yes    Rain          Cool    Normal Weak
9         Yes    Rain          Mild    Normal Weak
No and Yes Classes: PlayTennis Counter({'No': 2})
No and Yes Classes: PlayTennis Counter({'Yes': 3})
No and Yes Classes: PlayTennis Counter({'Yes': 3, 'No': 2})
Information Gain Calculation of Temperature
Cool
    PlayTennis Outlook Temperature Humidity Wind
8         Yes    Sunny          Cool    Normal Weak
Hot
    PlayTennis Outlook Temperature Humidity Wind
0         No    Sunny          Hot    High Weak
1         No    Sunny          Hot    High Strong
Mild
    PlayTennis Outlook Temperature Humidity Wind
7         No    Sunny          Mild    High Weak
10        Yes    Sunny          Mild    Normal Strong
No and Yes Classes: PlayTennis Counter({'Yes': 1})
No and Yes Classes: PlayTennis Counter({'No': 2})
No and Yes Classes: PlayTennis Counter({'No': 1, 'Yes': 1})
No and Yes Classes: PlayTennis Counter({'No': 3, 'Yes': 2})
Information Gain Calculation of Humidity
High
    PlayTennis Outlook Temperature Humidity Wind
0         No    Sunny          Hot    High Weak
1         No    Sunny          Hot    High Strong
7         No    Sunny          Mild    High Weak
Normal
    PlayTennis Outlook Temperature Humidity Wind
8         Yes    Sunny          Cool    Normal Weak
10        Yes    Sunny          Mild    Normal Strong
No and Yes Classes: PlayTennis Counter({'No': 3})
No and Yes Classes: PlayTennis Counter({'Yes': 2})
No and Yes Classes: PlayTennis Counter({'No': 3, 'Yes': 2})
Information Gain Calculation of Wind
Strong
    PlayTennis Outlook Temperature Humidity Wind

```

```

1          No   Sunny      Hot    High  Strong
10         Yes   Sunny      Mild   Normal Strong
Weak
   PlayTennis Outlook Temperature Humidity Wind
0          No   Sunny      Hot    High  Weak
7          No   Sunny      Mild   High  Weak
8          Yes   Sunny      Cool   Normal Weak
No and Yes Classes: PlayTennis Counter({'No': 1, 'Yes': 1})
No and Yes Classes: PlayTennis Counter({'No': 2, 'Yes': 1})
No and Yes Classes: PlayTennis Counter({'No': 3, 'Yes': 2})

```

The Resultant Decision Tree is :

```

{'Outlook': {'Overcast': 'Yes',
             'Rain': {'Wind': {'Strong': 'No', 'Weak': 'Yes'}},
             'Sunny': {'Humidity': {'High': 'No', 'Normal': 'Yes'}}}}

```

Classification Accuracy

```

In [87]: def classify(instance, tree, default=None):
          attribute = next(iter(tree))#tree.keys()[0]
          if instance[attribute] in tree[attribute].keys():
              result = tree[attribute][instance[attribute]]
              if isinstance(result, dict): # this is a tree, delve deeper
                  return classify(instance, result)
              else:
                  return result # this is a label
          else:
              return default

```

```

In [90]: df_tennis['predicted'] = df_tennis.apply(classify, axis=1, args=(tree,
          'No')) )
          # classify func allows for a default arg: when tree doesn't have an
          # answer for a particular
          # combination of attribute-values, we can use 'no' as the default g
          uess

```

```
print('Accuracy is:' + str( sum(df_tennis['PlayTennis']==df_tennis['predicted'] ) / (1.0*len(df_tennis.index)) ))

df_tennis[['PlayTennis', 'predicted']]
```

Accuracy is:1.0

Out[90]:

	PlayTennis	predicted
0	No	No
1	No	No
2	Yes	Yes
3	Yes	Yes
4	Yes	Yes
5	No	No
6	Yes	Yes
7	No	No
8	Yes	Yes
9	Yes	Yes
10	Yes	Yes
11	Yes	Yes
12	Yes	Yes
13	No	No

Classification Accuracy: Training/Testing Set

```
In [91]: training_data = df_tennis.iloc[1:-4] # all but last thousand instances
test_data = df_tennis.iloc[-4:] # just the last thousand
train_tree = id3(training_data, 'PlayTennis', attribute_names)

test_data['predicted2'] = test_data.apply(
    # <---- test_data source
                                classify,
                                axis=1,
                                args=(train_tree, 'Yes') ) # <
---- train_data tree

print ( '\n\n Accuracy is : ' + str( sum(test_data['PlayTennis']==test_data['predicted2']) / (1.0*len(test_data.index)) ) )
```

Information Gain Calculation of Outlook

Overcast

	PlayTennis	Outlook	Temperature	Humidity	Wind	predicted
2	Yes	Overcast	Hot	High	Weak	Yes
6	Yes	Overcast	Cool	Normal	Strong	Yes

Rain

	PlayTennis	Outlook	Temperature	Humidity	Wind	predicted
3	Yes	Rain	Mild	High	Weak	Yes
4	Yes	Rain	Cool	Normal	Weak	Yes
5	No	Rain	Cool	Normal	Strong	No
9	Yes	Rain	Mild	Normal	Weak	Yes

Sunny

	PlayTennis	Outlook	Temperature	Humidity	Wind	predicted
1	No	Sunny	Hot	High	Strong	No
7	No	Sunny	Mild	High	Weak	No
8	Yes	Sunny	Cool	Normal	Weak	Yes

No and Yes Classes: PlayTennis Counter({'Yes': 2})

No and Yes Classes: PlayTennis Counter({'Yes': 3, 'No': 1})

No and Yes Classes: PlayTennis Counter({'No': 2, 'Yes': 1})

No and Yes Classes: PlayTennis Counter({'Yes': 6, 'No': 3})

Information Gain Calculation of Temperature

Cool

	PlayTennis	Outlook	Temperature	Humidity	Wind	predicted
4	Yes	Rain	Cool	Normal	Weak	Yes
5	No	Rain	Cool	Normal	Strong	No

6	Yes	Overcast	Cool	Normal	Strong	Yes
8	Yes	Sunny	Cool	Normal	Weak	Yes

Hot

	PlayTennis	Outlook	Temperature	Humidity	Wind	predicted
1	No	Sunny	Hot	High	Strong	No
2	Yes	Overcast	Hot	High	Weak	Yes

Mild

	PlayTennis	Outlook	Temperature	Humidity	Wind	predicted
3	Yes	Rain	Mild	High	Weak	Yes
7	No	Sunny	Mild	High	Weak	No
9	Yes	Rain	Mild	Normal	Weak	Yes

No and Yes Classes: PlayTennis Counter({'Yes': 3, 'No': 1})
No and Yes Classes: PlayTennis Counter({'No': 1, 'Yes': 1})
No and Yes Classes: PlayTennis Counter({'Yes': 2, 'No': 1})
No and Yes Classes: PlayTennis Counter({'Yes': 6, 'No': 3})

Information Gain Calculation of Humidity

High

	PlayTennis	Outlook	Temperature	Humidity	Wind	predicted
1	No	Sunny	Hot	High	Strong	No
2	Yes	Overcast	Hot	High	Weak	Yes
3	Yes	Rain	Mild	High	Weak	Yes
7	No	Sunny	Mild	High	Weak	No

Normal

	PlayTennis	Outlook	Temperature	Humidity	Wind	predicted
4	Yes	Rain	Cool	Normal	Weak	Yes
5	No	Rain	Cool	Normal	Strong	No
6	Yes	Overcast	Cool	Normal	Strong	Yes
8	Yes	Sunny	Cool	Normal	Weak	Yes
9	Yes	Rain	Mild	Normal	Weak	Yes

No and Yes Classes: PlayTennis Counter({'No': 2, 'Yes': 2})
No and Yes Classes: PlayTennis Counter({'Yes': 4, 'No': 1})
No and Yes Classes: PlayTennis Counter({'Yes': 6, 'No': 3})

Information Gain Calculation of Wind

Strong

	PlayTennis	Outlook	Temperature	Humidity	Wind	predicted
1	No	Sunny	Hot	High	Strong	No
5	No	Rain	Cool	Normal	Strong	No
6	Yes	Overcast	Cool	Normal	Strong	Yes

Weak

	PlayTennis	Outlook	Temperature	Humidity	Wind	predicted
2	Yes	Overcast	Hot	High	Weak	Yes
3	Yes	Rain	Mild	High	Weak	Yes
4	Yes	Rain	Cool	Normal	Weak	Yes
7	No	Sunny	Mild	High	Weak	No
8	Yes	Sunny	Cool	Normal	Weak	Yes
9	Yes	Rain	Mild	Normal	Weak	Yes

No and Yes Classes: PlayTennis Counter({'No': 2, 'Yes': 1})
No and Yes Classes: PlayTennis Counter({'Yes': 5, 'No': 1})
No and Yes Classes: PlayTennis Counter({'Yes': 6, 'No': 3})

Information Gain Calculation of Temperature

Cool

	PlayTennis	Outlook	Temperature	Humidity	Wind	predicted
4	Yes	Rain	Cool	Normal	Weak	Yes
5	No	Rain	Cool	Normal	Strong	No

Mild

	PlayTennis	Outlook	Temperature	Humidity	Wind	predicted
3	Yes	Rain	Mild	High	Weak	Yes
9	Yes	Rain	Mild	Normal	Weak	Yes

No and Yes Classes: PlayTennis Counter({'Yes': 1, 'No': 1})
No and Yes Classes: PlayTennis Counter({'Yes': 2})
No and Yes Classes: PlayTennis Counter({'Yes': 3, 'No': 1})

Information Gain Calculation of Humidity

High

	PlayTennis	Outlook	Temperature	Humidity	Wind	predicted
3	Yes	Rain	Mild	High	Weak	Yes

Normal

	PlayTennis	Outlook	Temperature	Humidity	Wind	predicted
4	Yes	Rain	Cool	Normal	Weak	Yes
5	No	Rain	Cool	Normal	Strong	No
9	Yes	Rain	Mild	Normal	Weak	Yes

No and Yes Classes: PlayTennis Counter({'Yes': 1})
No and Yes Classes: PlayTennis Counter({'Yes': 2, 'No': 1})
No and Yes Classes: PlayTennis Counter({'Yes': 3, 'No': 1})

Information Gain Calculation of Wind

Strong

	PlayTennis	Outlook	Temperature	Humidity	Wind	predicted
5	No	Rain	Cool	Normal	Strong	No

Weak

```

    PlayTennis Outlook Temperature Humidity Wind predicted
3         Yes   Rain         Mild    High  Weak      Yes
4         Yes   Rain         Cool    Normal Weak      Yes
9         Yes   Rain         Mild    Normal Weak      Yes
No and Yes Classes: PlayTennis Counter({'No': 1})
No and Yes Classes: PlayTennis Counter({'Yes': 3})
No and Yes Classes: PlayTennis Counter({'Yes': 3, 'No': 1})
Information Gain Calculation of Temperature
Cool
    PlayTennis Outlook Temperature Humidity Wind predicted
8         Yes   Sunny         Cool    Normal Weak      Yes
Hot
    PlayTennis Outlook Temperature Humidity Wind predicted
1         No    Sunny         Hot     High  Strong     No
Mild
    PlayTennis Outlook Temperature Humidity Wind predicted
7         No    Sunny         Mild    High  Weak      No
No and Yes Classes: PlayTennis Counter({'Yes': 1})
No and Yes Classes: PlayTennis Counter({'No': 1})
No and Yes Classes: PlayTennis Counter({'No': 1})
No and Yes Classes: PlayTennis Counter({'No': 2, 'Yes': 1})
Information Gain Calculation of Humidity
High
    PlayTennis Outlook Temperature Humidity Wind predicted
1         No    Sunny         Hot     High  Strong     No
7         No    Sunny         Mild    High  Weak      No
Normal
    PlayTennis Outlook Temperature Humidity Wind predicted
8         Yes   Sunny         Cool    Normal Weak      Yes
No and Yes Classes: PlayTennis Counter({'No': 2})
No and Yes Classes: PlayTennis Counter({'Yes': 1})
No and Yes Classes: PlayTennis Counter({'No': 2, 'Yes': 1})
Information Gain Calculation of Wind
Strong
    PlayTennis Outlook Temperature Humidity Wind predicted
1         No    Sunny         Hot     High  Strong     No
Weak
    PlayTennis Outlook Temperature Humidity Wind predicted
7         No    Sunny         Mild    High  Weak      No

```

```
8          Yes   Sunny      Cool   Normal  Weak      Yes
No and Yes Classes: PlayTennis Counter({'No': 1})
No and Yes Classes: PlayTennis Counter({'No': 1, 'Yes': 1})
No and Yes Classes: PlayTennis Counter({'No': 2, 'Yes': 1})
```

Accuracy is : 0.75

```
C:\Users\Dr.Thyagaraju\Anaconda3\lib\site-packages\ipykernel_launcher.p
y:8: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead
```

```
See the caveats in the documentation: http://pandas.pydata.org/pandas-d
ocs/stable/indexing.html#indexing-view-versus-copy
```

**Lab Exercise : Apply above Program to classify
the new sample /new data set**

End