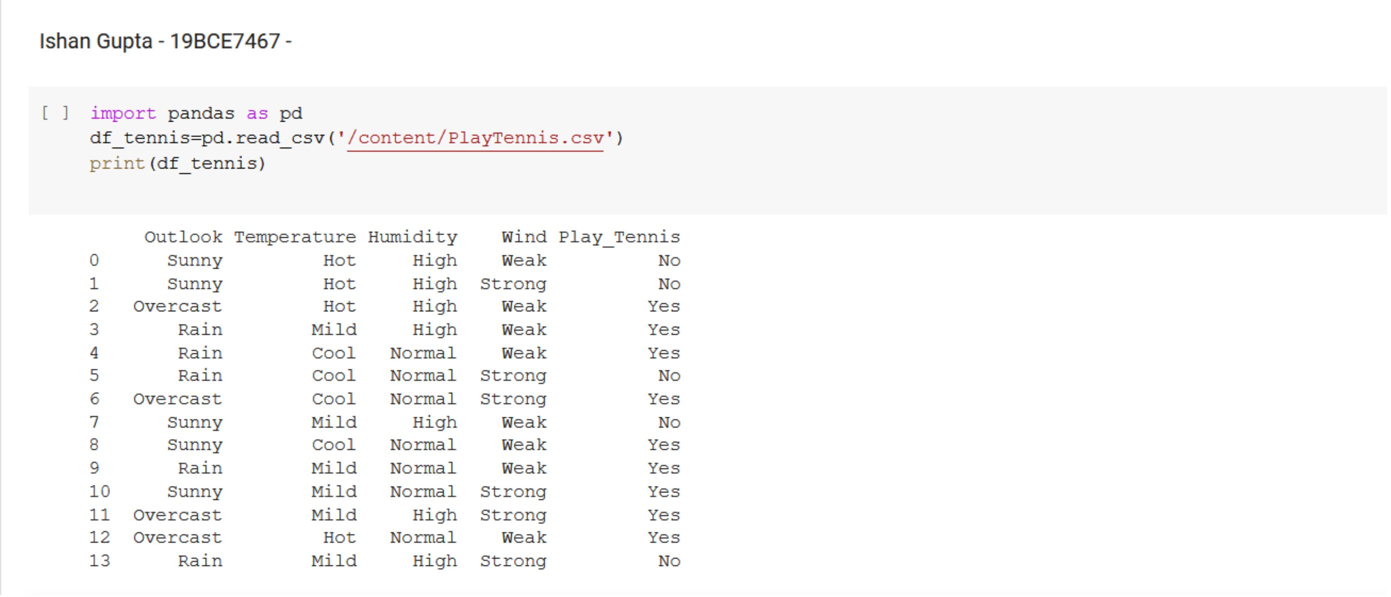
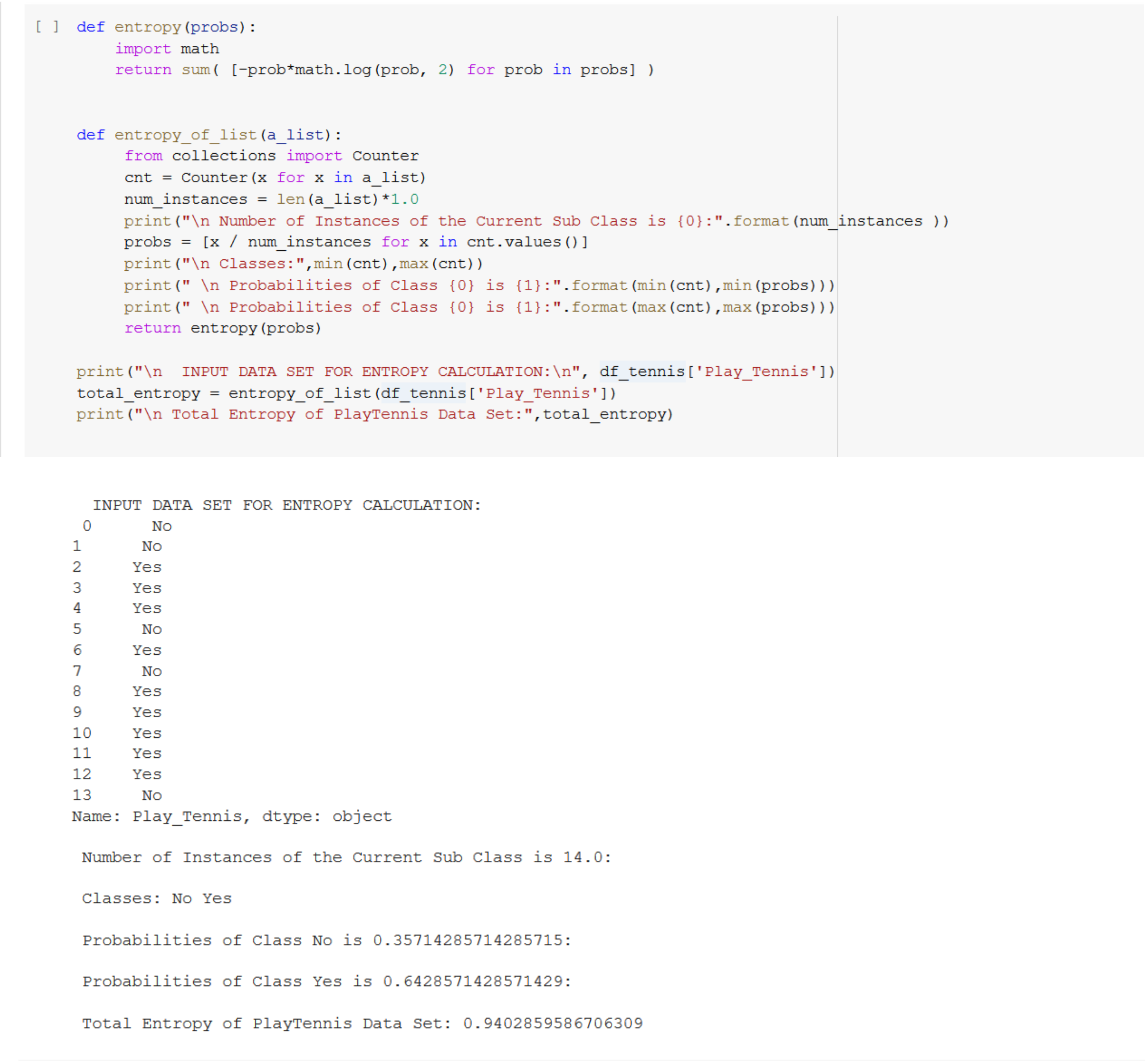


Ishan Gupta - 19BCE7467 -

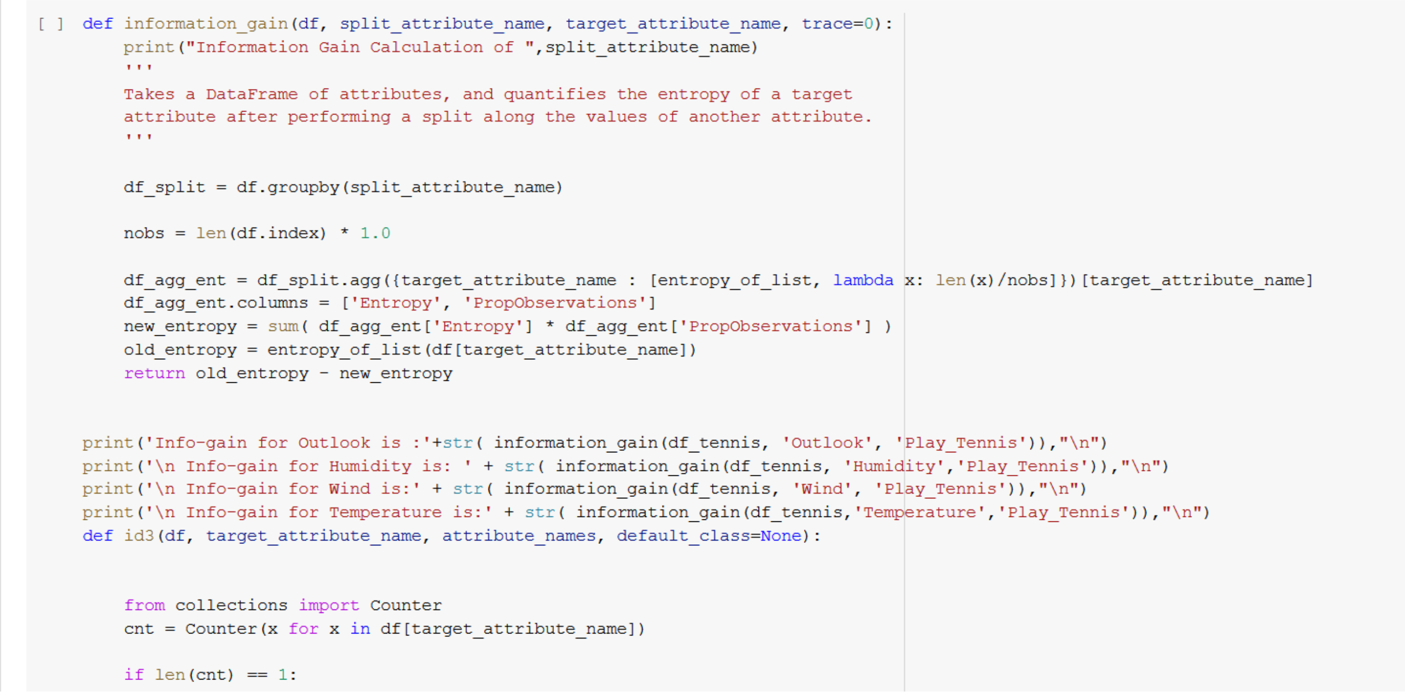
```
[ ]  import pandas as pd
     df_tennis=pd.read_csv('/content/PlayTennis.csv')
     print(df_tennis)
```

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

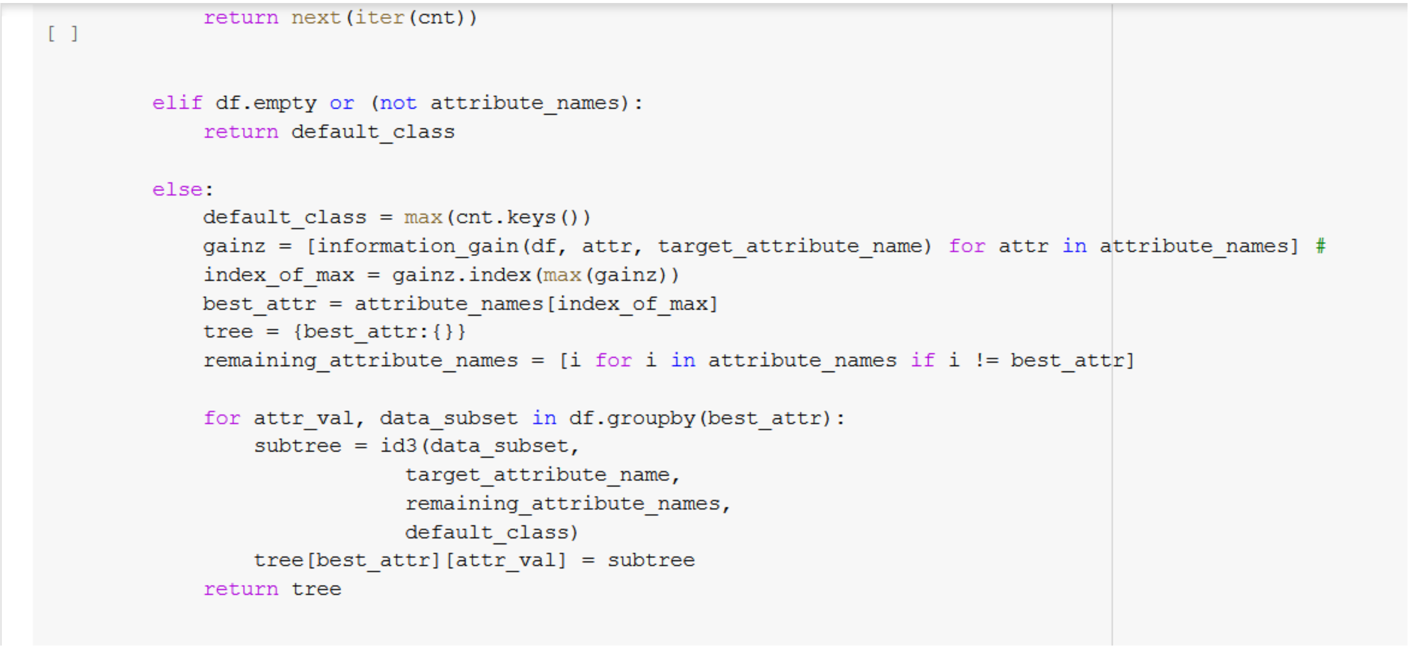


```
[ ]  def entropy(probs):
         import math
         return sum( [-prob*math.log(prob, 2) for prob in probs] )


     def entropy_of_list(a_list):
          from collections import Counter
          cnt = Counter(x for x in a_list)
          num_instances = len(a_list)*1.0
          print("\n Number of Instances of the Current Sub Class is {0}:".format(num_instances ))
          probs = [x / num_instances for x in cnt.values()]
          print("\n Classes:",min(cnt),max(cnt))
          print(" \n Probabilities of Class {0} is {1}:".format(min(cnt),min(probs)))
          print(" \n Probabilities of Class {0} is {1}:".format(max(cnt),max(probs)))
          return entropy(probs)

     print("\n  INPUT DATA SET FOR ENTROPY CALCULATION:\n", df_tennis['Play_Tennis'])
     total_entropy = entropy_of_list(df_tennis['Play_Tennis'])
     print("\n Total Entropy of PlayTennis Data Set:",total_entropy)
```

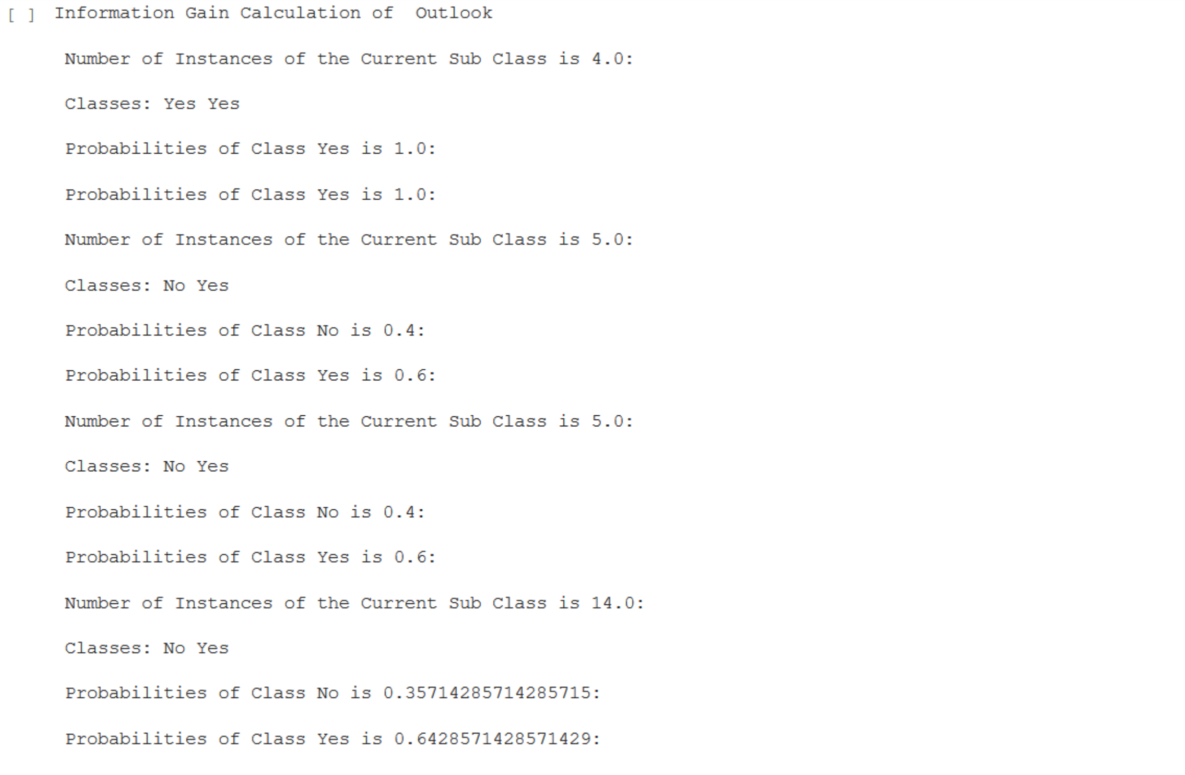
```
  INPUT DATA SET FOR ENTROPY CALCULATION:
 0      No
1      No
2     Yes
3     Yes
4     Yes
5      No
6     Yes
7      No
8     Yes
9     Yes
10    Yes
11    Yes
12    Yes
13     No
Name: Play_Tennis, dtype: object

 Number of Instances of the Current Sub Class is 14.0:

 Classes: No Yes

 Probabilities of Class No is 0.35714285714285715:

 Probabilities of Class Yes is 0.6428571428571429:

 Total Entropy of PlayTennis Data Set: 0.9402859586706309
```



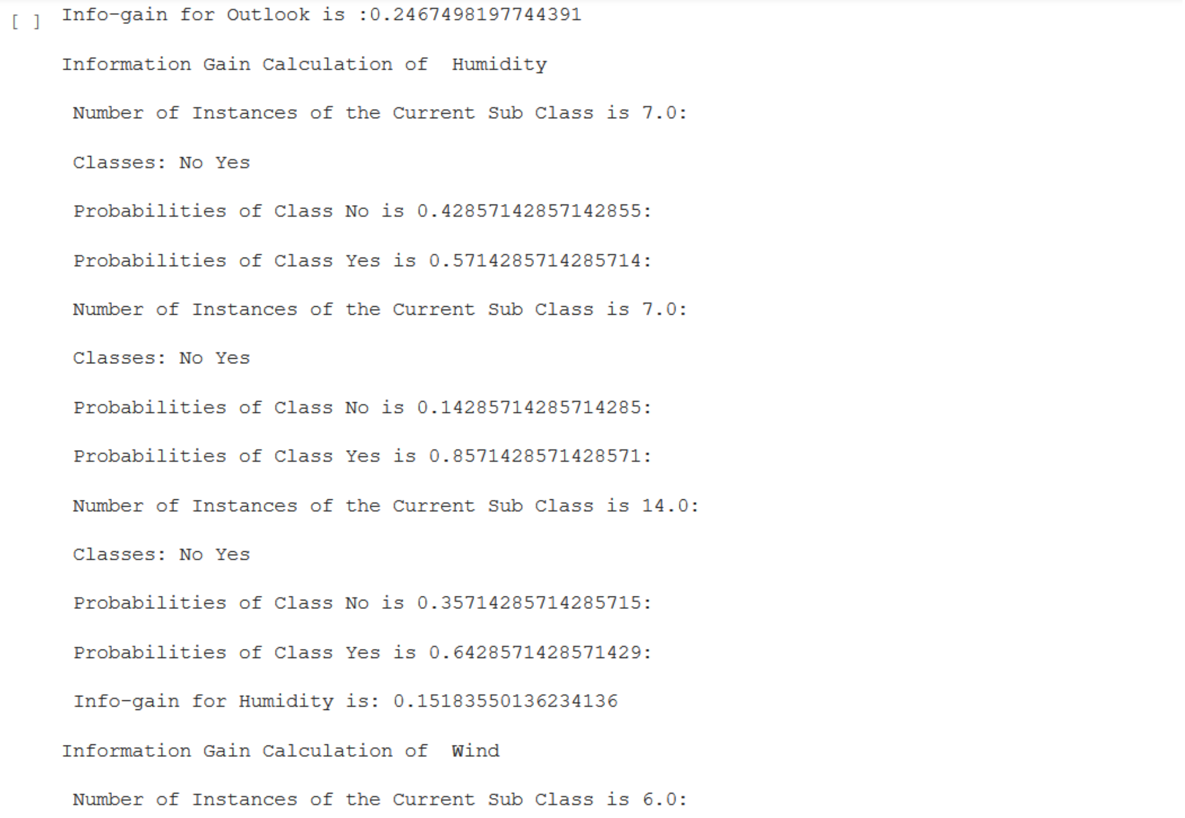
```
[ ] def information_gain(df, split_attribute_name, target_attribute_name, trace=0):
        print("Information Gain Calculation of ",split_attribute_name)
        '''
        Takes a DataFrame of attributes, and quantifies the entropy of a target
        attribute after performing a split along the values of another attribute.
        '''

        df_split = df.groupby(split_attribute_name)

        nobs = len(df.index) * 1.0

        df_agg_ent = df_split.agg({target_attribute_name : [entropy_of_list, lambda x: len(x)/nobs]})[target_attribute_name]
        df_agg_ent.columns = ['Entropy', 'PropObservations']
        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', 'Play_Tennis')),"\n")
    print('\n Info-gain for Humidity is: ' + str( information_gain(df_tennis, 'Humidity','Play_Tennis')),"\n")
    print('\n Info-gain for Wind is:' + str( information_gain(df_tennis, 'Wind', 'Play_Tennis')),"\n")
    print('\n Info-gain for Temperature is:' + str( information_gain(df_tennis,'Temperature','Play_Tennis')),"\n")
    def id3(df, target_attribute_name, attribute_names, default_class=None):


        from collections import Counter
        cnt = Counter(x for x in df[target_attribute_name])

        if len(cnt) == 1:
```



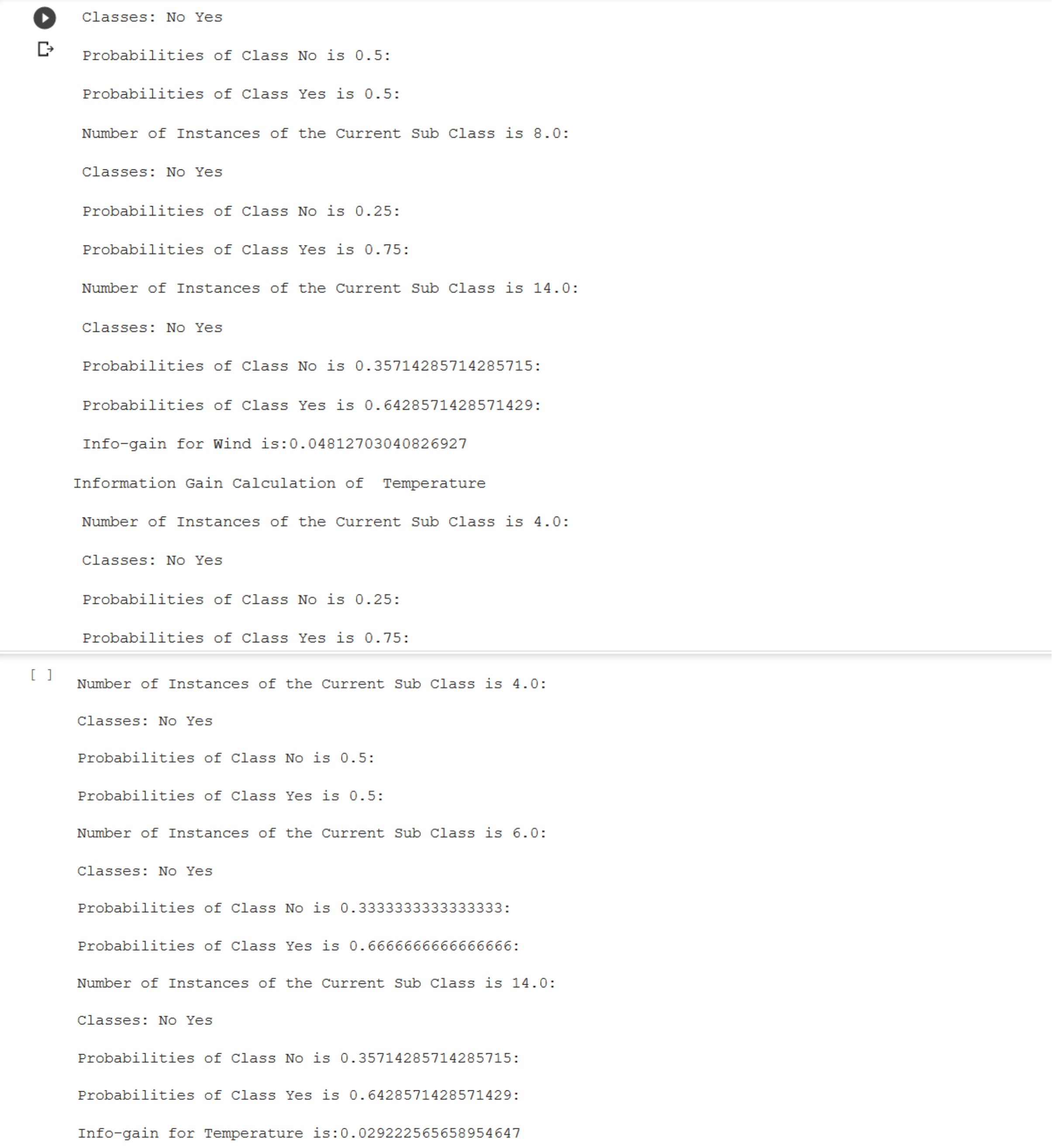
```
[ ]         return next(iter(cnt))


        elif df.empty or (not attribute_names):
            return default_class

        else:
            default_class = max(cnt.keys())
            gainz = [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]
            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_name,
                        remaining_attribute_names,
                        default_class)
                tree[best_attr][attr_val] = subtree
            return tree
```



```
[ ]  Information Gain Calculation of  Outlook

      Number of Instances of the Current Sub Class is 4.0:

      Classes: Yes Yes

      Probabilities of Class Yes is 1.0:

      Probabilities of Class Yes is 1.0:

      Number of Instances of the Current Sub Class is 5.0:

      Classes: No Yes

      Probabilities of Class No is 0.4:

      Probabilities of Class Yes is 0.6:

      Number of Instances of the Current Sub Class is 5.0:

      Classes: No Yes

      Probabilities of Class No is 0.4:

      Probabilities of Class Yes is 0.6:

      Number of Instances of the Current Sub Class is 14.0:

      Classes: No Yes

      Probabilities of Class No is 0.35714285714285715:

      Probabilities of Class Yes is 0.6428571428571429:
```



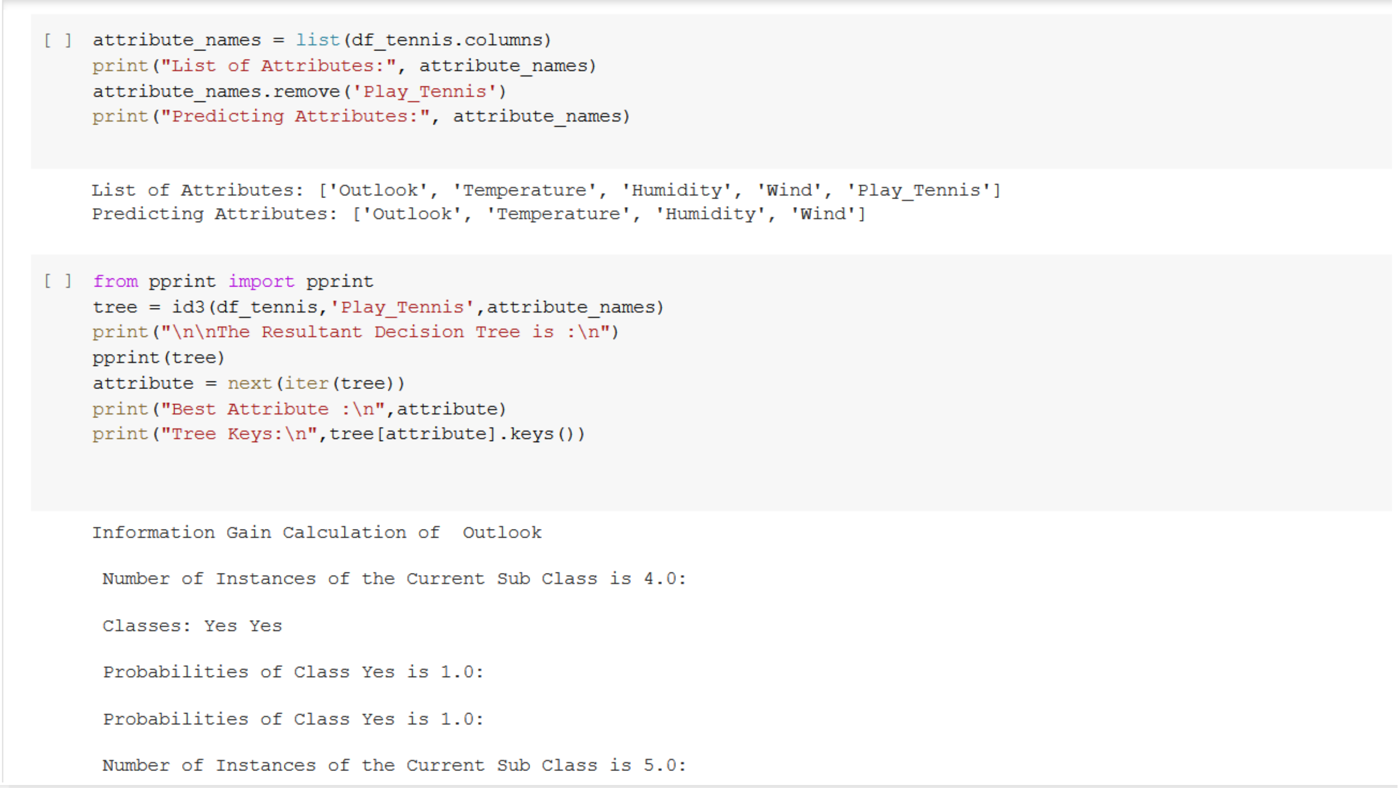
```
[ ]  Info-gain for Outlook is :0.2467498197744391

     Information Gain Calculation of  Humidity

      Number of Instances of the Current Sub Class is 7.0:

      Classes: No Yes

      Probabilities of Class No is 0.42857142857142855:

      Probabilities of Class Yes is 0.5714285714285714:

      Number of Instances of the Current Sub Class is 7.0:

      Classes: No Yes

      Probabilities of Class No is 0.14285714285714285:

      Probabilities of Class Yes is 0.8571428571428571:

      Number of Instances of the Current Sub Class is 14.0:

      Classes: No Yes

      Probabilities of Class No is 0.35714285714285715:

      Probabilities of Class Yes is 0.6428571428571429:

      Info-gain for Humidity is: 0.15183550136234136

     Information Gain Calculation of  Wind

      Number of Instances of the Current Sub Class is 6.0:
```



```
 Classes: No Yes

 Probabilities of Class No is 0.5:

 Probabilities of Class Yes is 0.5:

 Number of Instances of the Current Sub Class is 8.0:

 Classes: No Yes

 Probabilities of Class No is 0.25:

 Probabilities of Class Yes is 0.75:

 Number of Instances of the Current Sub Class is 14.0:

 Classes: No Yes

 Probabilities of Class No is 0.35714285714285715:

 Probabilities of Class Yes is 0.6428571428571429:

 Info-gain for Wind is:0.04812703040826927

Information Gain Calculation of  Temperature

 Number of Instances of the Current Sub Class is 4.0:

 Classes: No Yes

 Probabilities of Class No is 0.25:

 Probabilities of Class Yes is 0.75:
```

[ ]

```
Number of Instances of the Current Sub Class is 4.0:

Classes: No Yes

Probabilities of Class No is 0.5:

Probabilities of Class Yes is 0.5:

Number of Instances of the Current Sub Class is 6.0:

Classes: No Yes

Probabilities of Class No is 0.3333333333333333:

Probabilities of Class Yes is 0.6666666666666666:

Number of Instances of the Current Sub Class is 14.0:

Classes: No Yes

Probabilities of Class No is 0.35714285714285715:

Probabilities of Class Yes is 0.6428571428571429:

Info-gain for Temperature is:0.029222565658954647
```

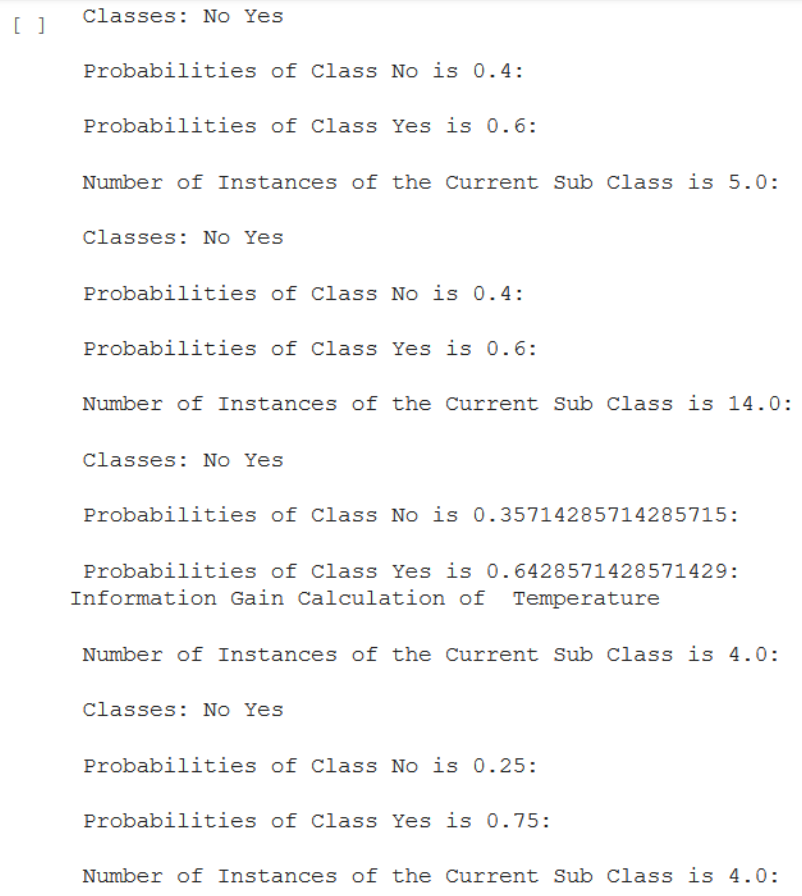


```
[ ]  attribute_names = list(df_tennis.columns)
     print("List of Attributes:", attribute_names)
     attribute_names.remove('Play_Tennis')
     print("Predicting Attributes:", attribute_names)
```

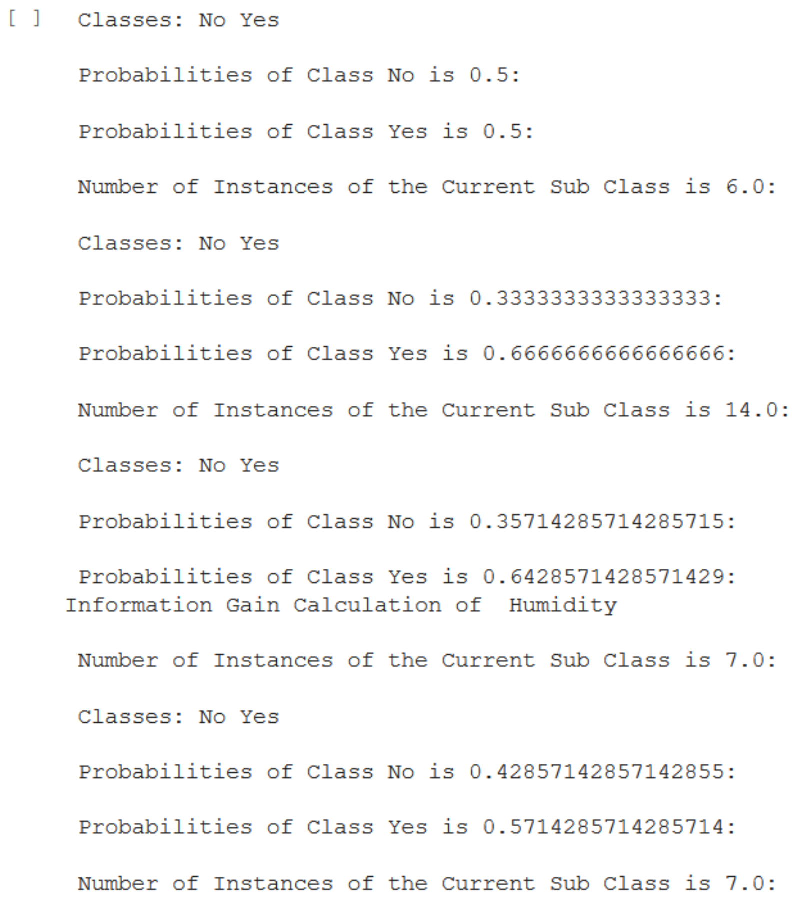
```
     List of Attributes: ['Outlook', 'Temperature', 'Humidity', 'Wind', 'Play_Tennis']
     Predicting Attributes: ['Outlook', 'Temperature', 'Humidity', 'Wind']
```

```
[ ]  from pprint import pprint
     tree = id3(df_tennis,'Play_Tennis',attribute_names)
     print("\n\nThe Resultant Decision Tree is :\n")
     pprint(tree)
     attribute = next(iter(tree))
     print("Best Attribute :\n",attribute)
     print("Tree Keys:\n",tree[attribute].keys())
```

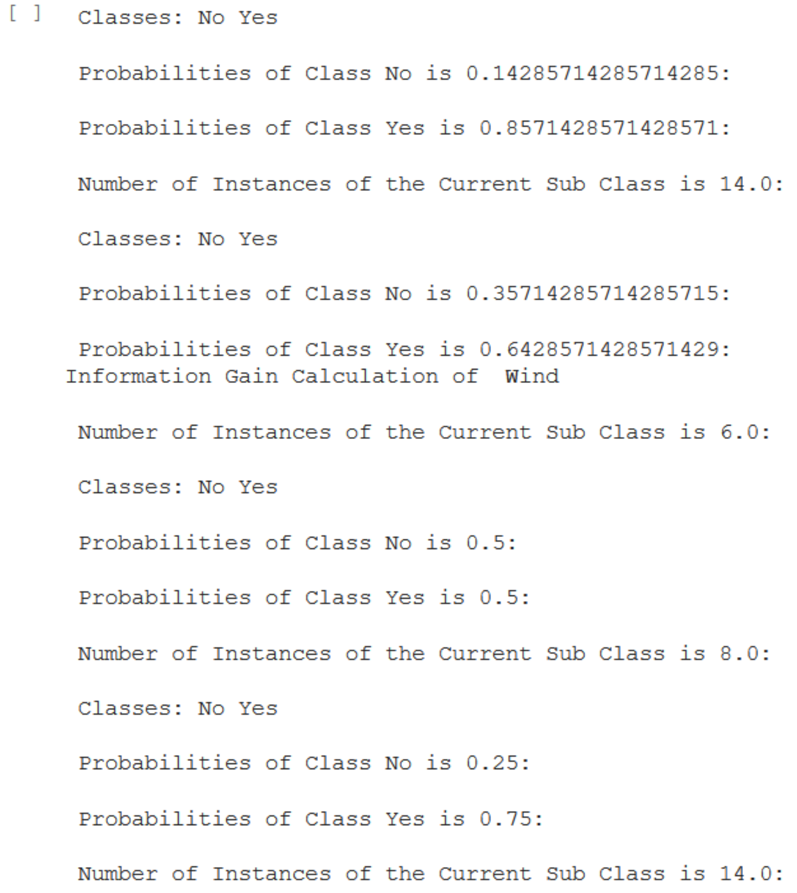
```
     Information Gain Calculation of  Outlook

      Number of Instances of the Current Sub Class is 4.0:

      Classes: Yes Yes

      Probabilities of Class Yes is 1.0:

      Probabilities of Class Yes is 1.0:

      Number of Instances of the Current Sub Class is 5.0:
```



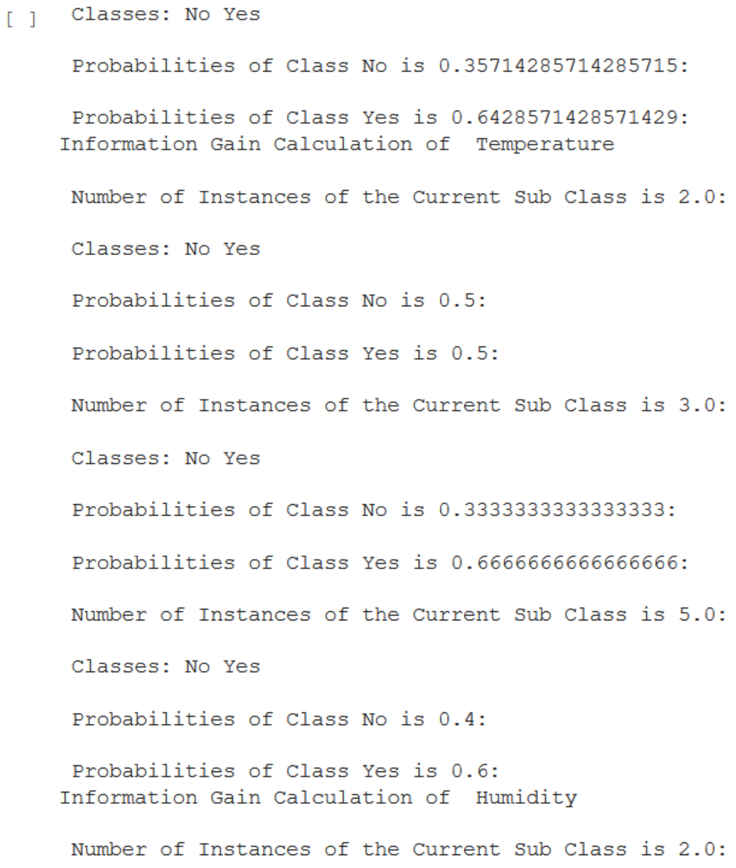
```
[ ]  Classes: No Yes

     Probabilities of Class No is 0.4:

     Probabilities of Class Yes is 0.6:

     Number of Instances of the Current Sub Class is 5.0:

     Classes: No Yes

     Probabilities of Class No is 0.4:

     Probabilities of Class Yes is 0.6:

     Number of Instances of the Current Sub Class is 14.0:

     Classes: No Yes

     Probabilities of Class No is 0.35714285714285715:

     Probabilities of Class Yes is 0.6428571428571429:
    Information Gain Calculation of  Temperature

     Number of Instances of the Current Sub Class is 4.0:

     Classes: No Yes

     Probabilities of Class No is 0.25:

     Probabilities of Class Yes is 0.75:

     Number of Instances of the Current Sub Class is 4.0:
```



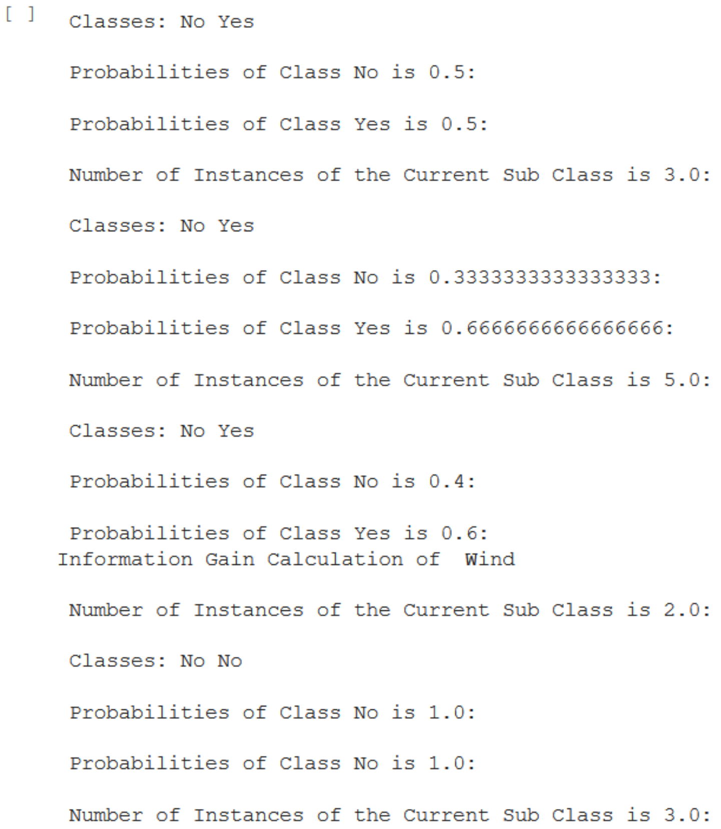
```
[ ]  Classes: No Yes

     Probabilities of Class No is 0.5:

     Probabilities of Class Yes is 0.5:

     Number of Instances of the Current Sub Class is 6.0:

     Classes: No Yes

     Probabilities of Class No is 0.3333333333333333:

     Probabilities of Class Yes is 0.6666666666666666:

     Number of Instances of the Current Sub Class is 14.0:

     Classes: No Yes

     Probabilities of Class No is 0.35714285714285715:

     Probabilities of Class Yes is 0.6428571428571429:
    Information Gain Calculation of  Humidity

     Number of Instances of the Current Sub Class is 7.0:

     Classes: No Yes

     Probabilities of Class No is 0.42857142857142855:

     Probabilities of Class Yes is 0.5714285714285714:

     Number of Instances of the Current Sub Class is 7.0:
```



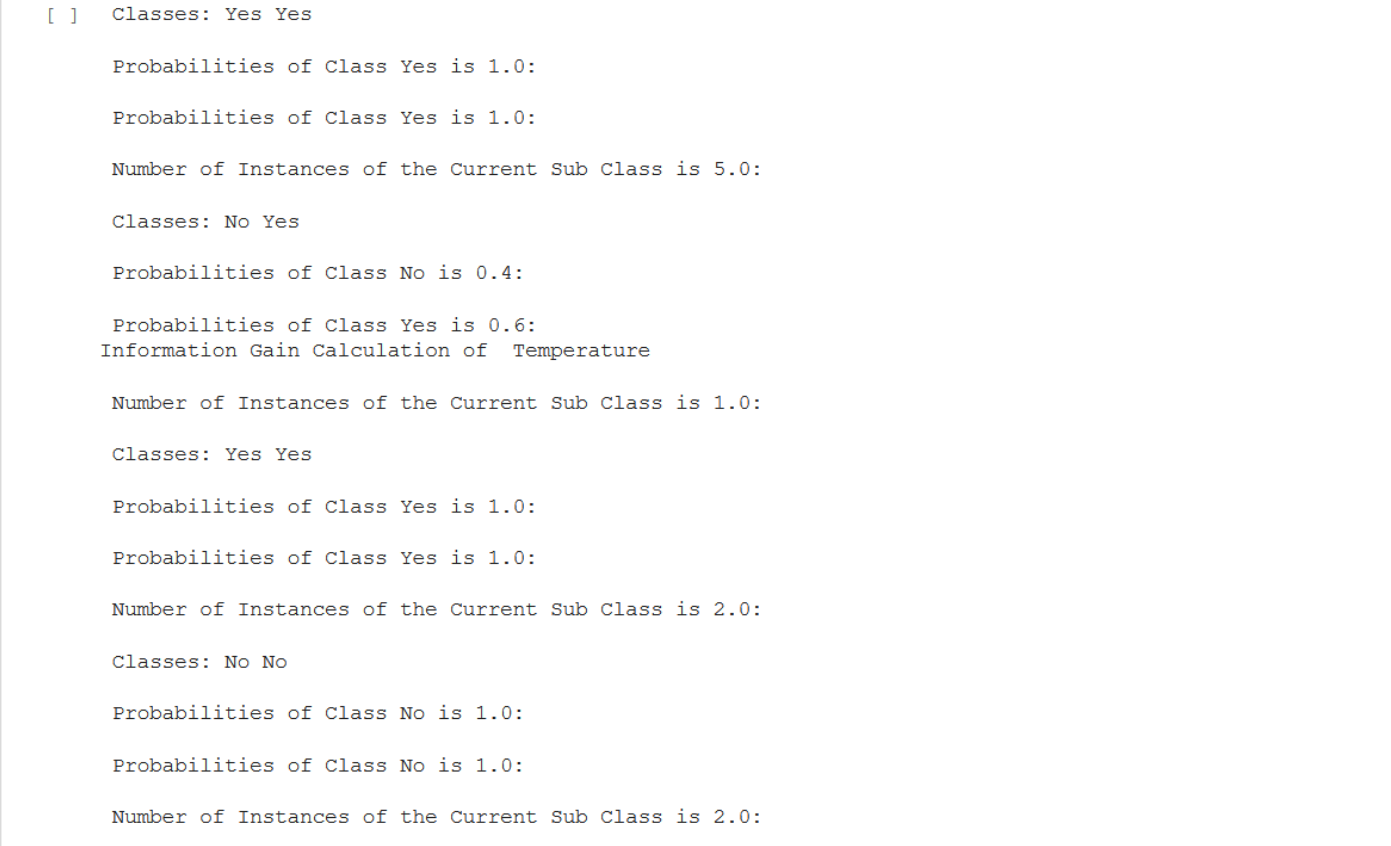
```
[ ]  Classes: No Yes

     Probabilities of Class No is 0.14285714285714285:

     Probabilities of Class Yes is 0.8571428571428571:

     Number of Instances of the Current Sub Class is 14.0:

     Classes: No Yes

     Probabilities of Class No is 0.35714285714285715:

     Probabilities of Class Yes is 0.6428571428571429:
    Information Gain Calculation of  Wind

     Number of Instances of the Current Sub Class is 6.0:

     Classes: No Yes

     Probabilities of Class No is 0.5:

     Probabilities of Class Yes is 0.5:

     Number of Instances of the Current Sub Class is 8.0:

     Classes: No Yes

     Probabilities of Class No is 0.25:

     Probabilities of Class Yes is 0.75:

     Number of Instances of the Current Sub Class is 14.0:
```



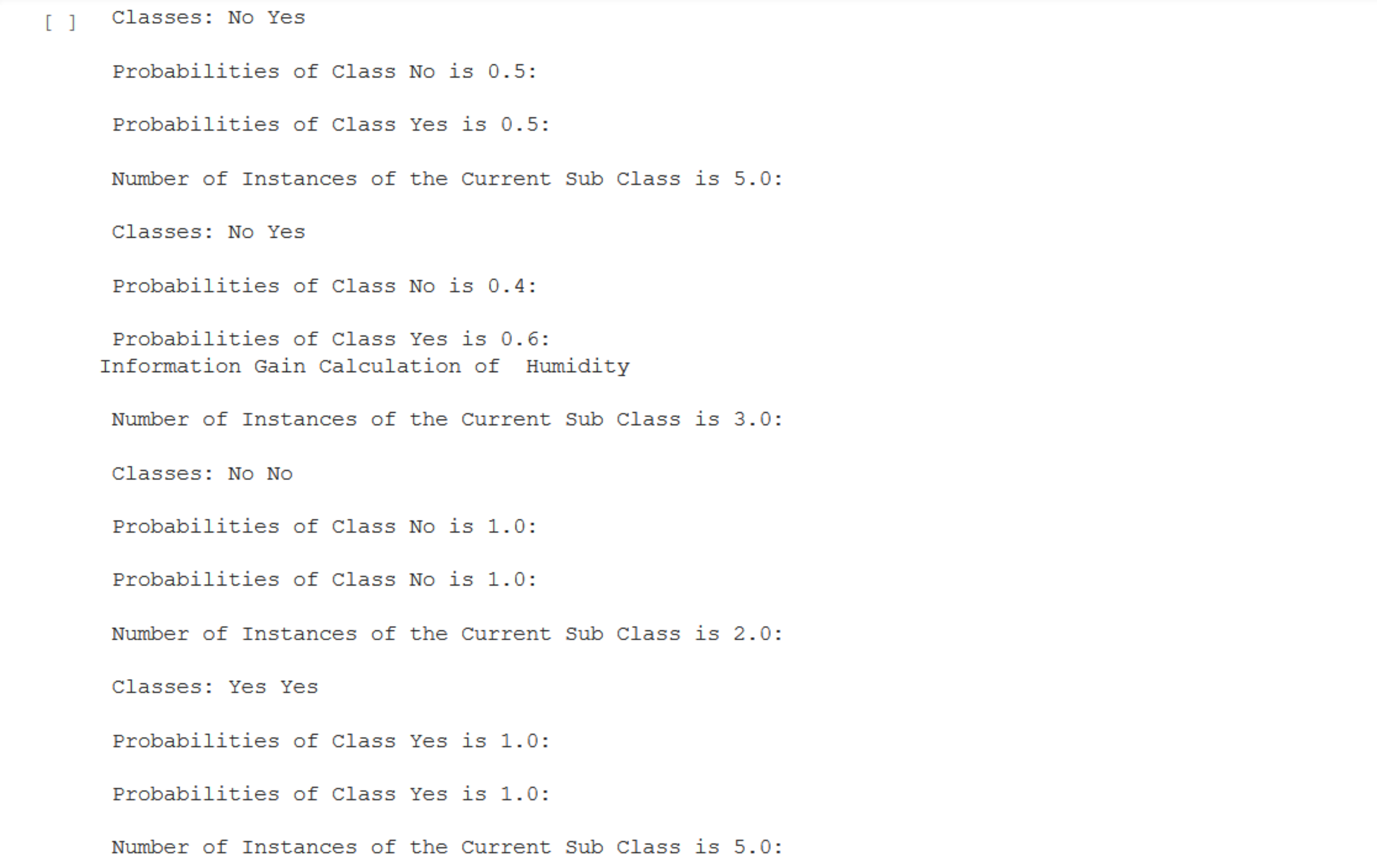
```
[ ]  Classes: No Yes

     Probabilities of Class No is 0.35714285714285715:

     Probabilities of Class Yes is 0.6428571428571429:
    Information Gain Calculation of  Temperature

     Number of Instances of the Current Sub Class is 2.0:

     Classes: No Yes

     Probabilities of Class No is 0.5:

     Probabilities of Class Yes is 0.5:

     Number of Instances of the Current Sub Class is 3.0:

     Classes: No Yes

     Probabilities of Class No is 0.3333333333333333:

     Probabilities of Class Yes is 0.6666666666666666:

     Number of Instances of the Current Sub Class is 5.0:

     Classes: No Yes

     Probabilities of Class No is 0.4:

     Probabilities of Class Yes is 0.6:
    Information Gain Calculation of  Humidity

     Number of Instances of the Current Sub Class is 2.0:
```



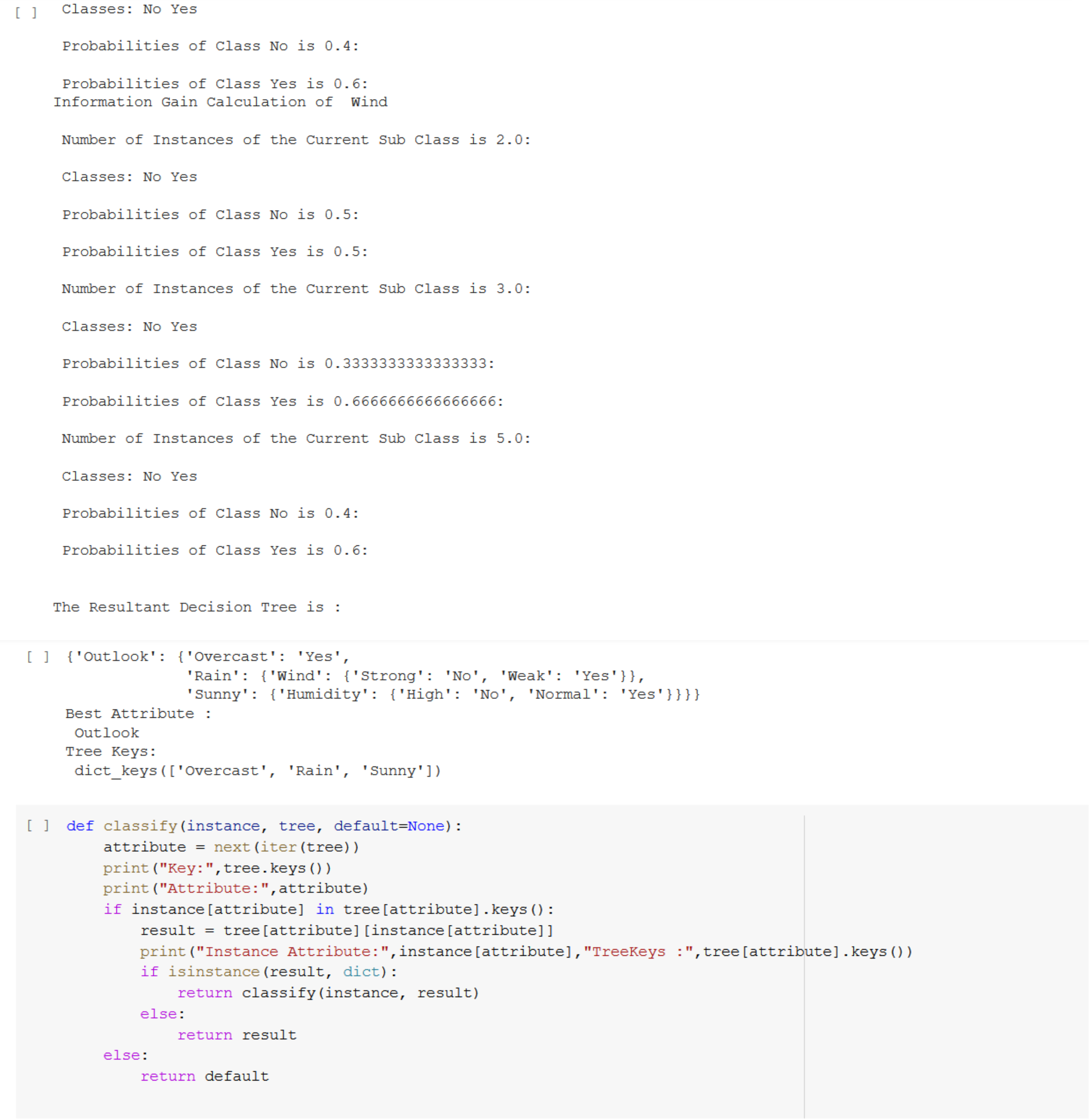
```
[ ]  Classes: No Yes

     Probabilities of Class No is 0.5:

     Probabilities of Class Yes is 0.5:

     Number of Instances of the Current Sub Class is 3.0:

     Classes: No Yes

     Probabilities of Class No is 0.3333333333333333:

     Probabilities of Class Yes is 0.6666666666666666:

     Number of Instances of the Current Sub Class is 5.0:

     Classes: No Yes

     Probabilities of Class No is 0.4:

     Probabilities of Class Yes is 0.6:
    Information Gain Calculation of  Wind

     Number of Instances of the Current Sub Class is 2.0:

     Classes: No No

     Probabilities of Class No is 1.0:

     Probabilities of Class No is 1.0:

     Number of Instances of the Current Sub Class is 3.0:
```



```
[ ]  Classes: Yes Yes

     Probabilities of Class Yes is 1.0:

     Probabilities of Class Yes is 1.0:

     Number of Instances of the Current Sub Class is 5.0:

     Classes: No Yes

     Probabilities of Class No is 0.4:

     Probabilities of Class Yes is 0.6:
    Information Gain Calculation of  Temperature

     Number of Instances of the Current Sub Class is 1.0:

     Classes: Yes Yes

     Probabilities of Class Yes is 1.0:

     Probabilities of Class Yes is 1.0:

     Number of Instances of the Current Sub Class is 2.0:

     Classes: No No

     Probabilities of Class No is 1.0:

     Probabilities of Class No is 1.0:

     Number of Instances of the Current Sub Class is 2.0:
```



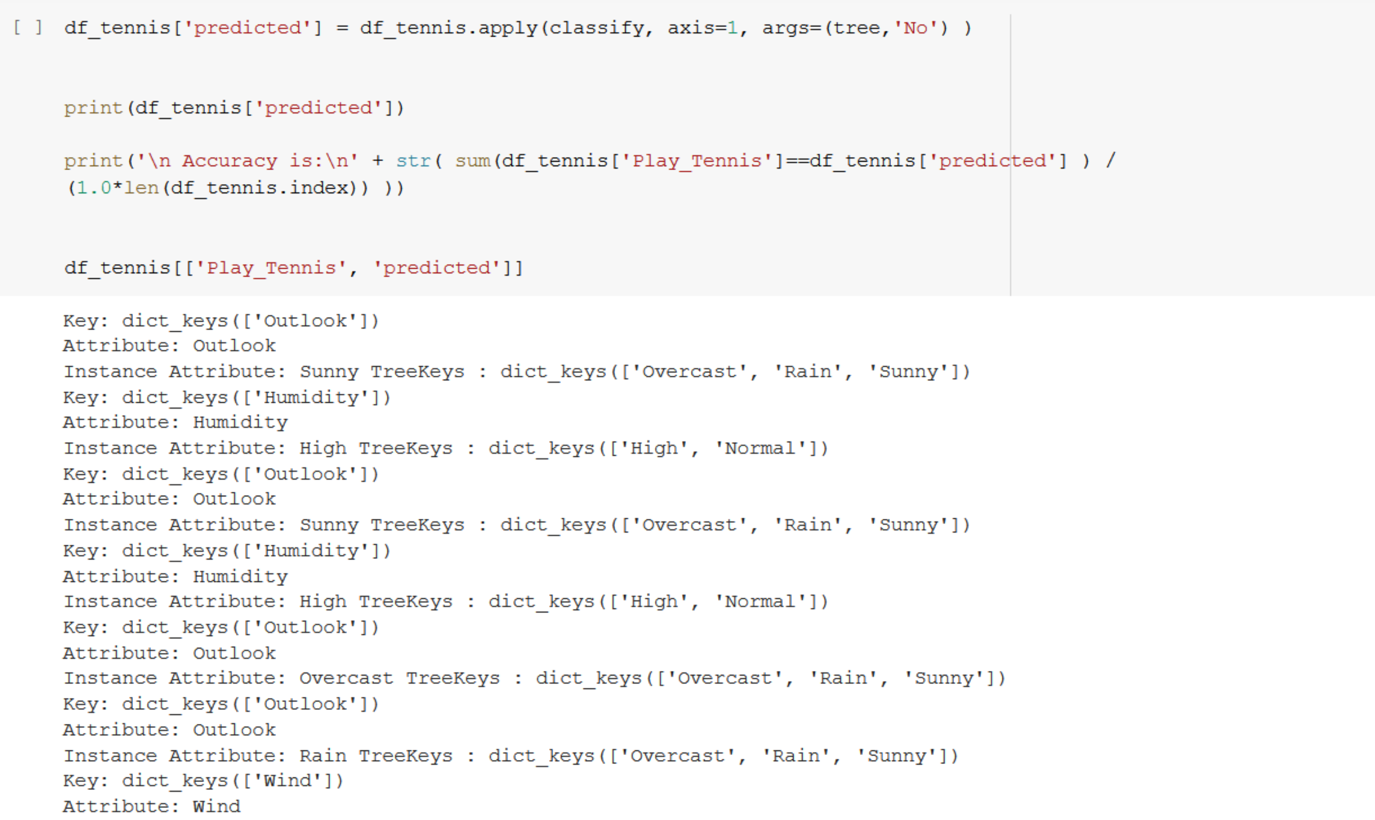
```
[ ]  Classes: No Yes

     Probabilities of Class No is 0.5:

     Probabilities of Class Yes is 0.5:

     Number of Instances of the Current Sub Class is 5.0:

     Classes: No Yes

     Probabilities of Class No is 0.4:

     Probabilities of Class Yes is 0.6:
    Information Gain Calculation of  Humidity

     Number of Instances of the Current Sub Class is 3.0:

     Classes: No No

     Probabilities of Class No is 1.0:

     Probabilities of Class No is 1.0:

     Number of Instances of the Current Sub Class is 2.0:

     Classes: Yes Yes

     Probabilities of Class Yes is 1.0:

     Probabilities of Class Yes is 1.0:

     Number of Instances of the Current Sub Class is 5.0:
```



```
[ ]  Classes: No Yes

     Probabilities of Class No is 0.4:

     Probabilities of Class Yes is 0.6:
    Information Gain Calculation of  Wind

     Number of Instances of the Current Sub Class is 2.0:

     Classes: No Yes

     Probabilities of Class No is 0.5:

     Probabilities of Class Yes is 0.5:

     Number of Instances of the Current Sub Class is 3.0:

     Classes: No Yes

     Probabilities of Class No is 0.3333333333333333:

     Probabilities of Class Yes is 0.6666666666666666:

     Number of Instances of the Current Sub Class is 5.0:

     Classes: No Yes

     Probabilities of Class No is 0.4:

     Probabilities of Class Yes is 0.6:


    The Resultant Decision Tree is :
```

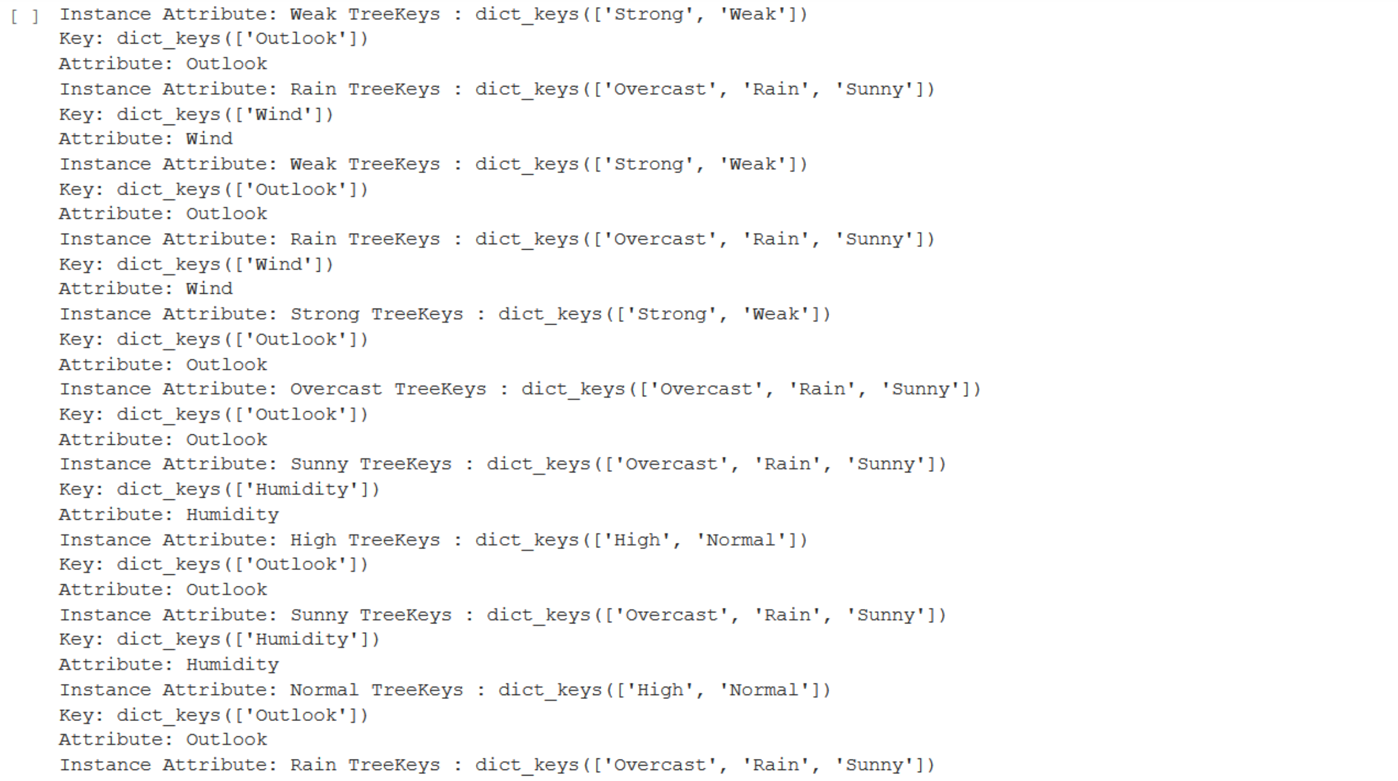
```
[ ]  {'Outlook': {'Overcast': 'Yes',
                  'Rain': {'Wind': {'Strong': 'No', 'Weak': 'Yes'}},
                  'Sunny': {'Humidity': {'High': 'No', 'Normal': 'Yes'}}}}
     Best Attribute :
      Outlook
     Tree Keys:
      dict_keys(['Overcast', 'Rain', 'Sunny'])
```

```
[ ]  def classify(instance, tree, default=None):
         attribute = next(iter(tree))
         print("Key:",tree.keys())
         print("Attribute:",attribute)
         if instance[attribute] in tree[attribute].keys():
             result = tree[attribute][instance[attribute]]
             print("Instance Attribute:",instance[attribute],"TreeKeys :",tree[attribute].keys())
             if isinstance(result, dict):
                 return classify(instance, result)
             else:
                 return result
         else:
             return default
```

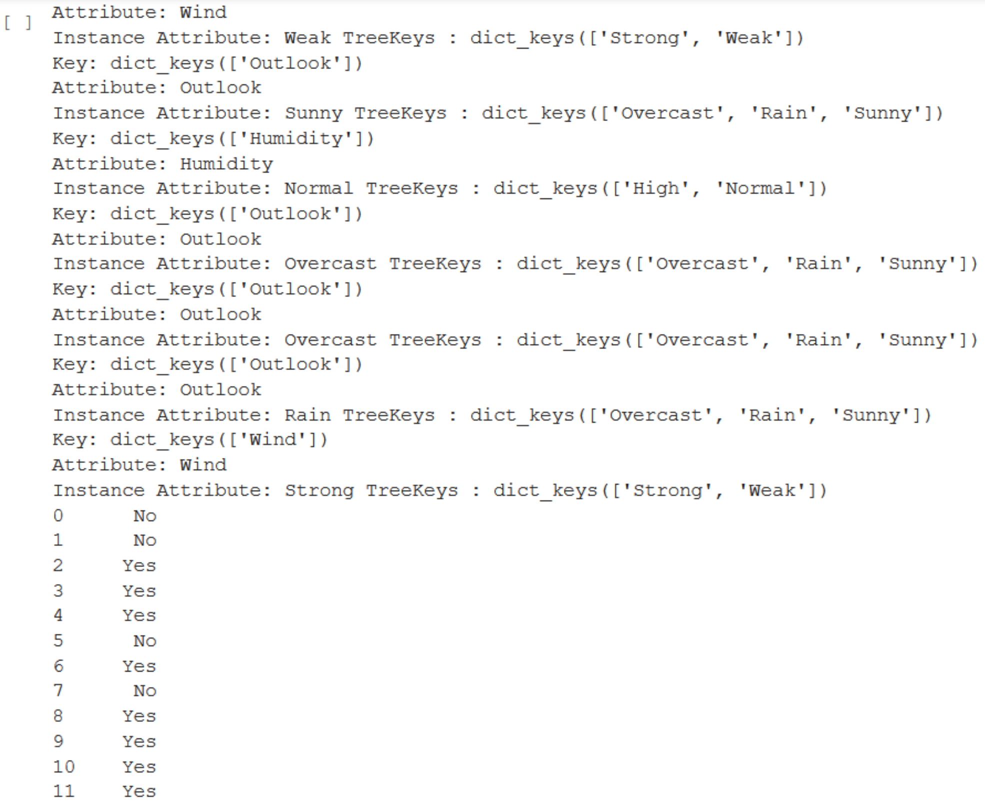


```
[ ] df_tennis['predicted'] = df_tennis.apply(classify, axis=1, args=(tree,'No') )


    print(df_tennis['predicted'])

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


    df_tennis[['Play_Tennis', 'predicted']]
```

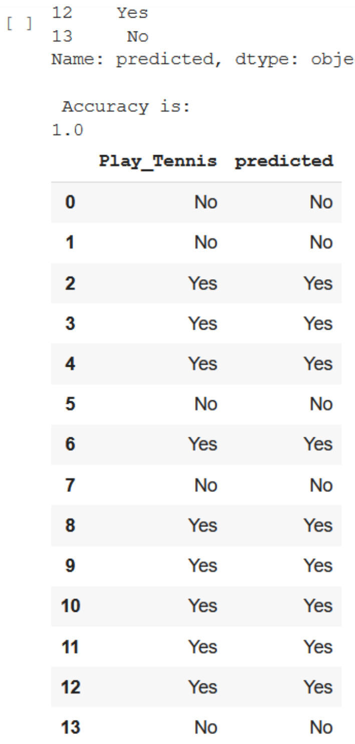
```
Key: dict_keys(['Outlook'])
Attribute: Outlook
Instance Attribute: Sunny TreeKeys : dict_keys(['Overcast', 'Rain', 'Sunny'])
Key: dict_keys(['Humidity'])
Attribute: Humidity
Instance Attribute: High TreeKeys : dict_keys(['High', 'Normal'])
Key: dict_keys(['Outlook'])
Attribute: Outlook
Instance Attribute: Sunny TreeKeys : dict_keys(['Overcast', 'Rain', 'Sunny'])
Key: dict_keys(['Humidity'])
Attribute: Humidity
Instance Attribute: High TreeKeys : dict_keys(['High', 'Normal'])
Key: dict_keys(['Outlook'])
Attribute: Outlook
Instance Attribute: Overcast TreeKeys : dict_keys(['Overcast', 'Rain', 'Sunny'])
Key: dict_keys(['Outlook'])
Attribute: Outlook
Instance Attribute: Rain TreeKeys : dict_keys(['Overcast', 'Rain', 'Sunny'])
Key: dict_keys(['Wind'])
Attribute: Wind
```



```
[ ] Instance Attribute: Weak TreeKeys : dict_keys(['Strong', 'Weak'])
    Key: dict_keys(['Outlook'])
    Attribute: Outlook
    Instance Attribute: Rain TreeKeys : dict_keys(['Overcast', 'Rain', 'Sunny'])
    Key: dict_keys(['Wind'])
    Attribute: Wind
    Instance Attribute: Weak TreeKeys : dict_keys(['Strong', 'Weak'])
    Key: dict_keys(['Outlook'])
    Attribute: Outlook
    Instance Attribute: Rain TreeKeys : dict_keys(['Overcast', 'Rain', 'Sunny'])
    Key: dict_keys(['Wind'])
    Attribute: Wind
    Instance Attribute: Strong TreeKeys : dict_keys(['Strong', 'Weak'])
    Key: dict_keys(['Outlook'])
    Attribute: Outlook
    Instance Attribute: Overcast TreeKeys : dict_keys(['Overcast', 'Rain', 'Sunny'])
    Key: dict_keys(['Outlook'])
    Attribute: Outlook
    Instance Attribute: Sunny TreeKeys : dict_keys(['Overcast', 'Rain', 'Sunny'])
    Key: dict_keys(['Humidity'])
    Attribute: Humidity
    Instance Attribute: High TreeKeys : dict_keys(['High', 'Normal'])
    Key: dict_keys(['Outlook'])
    Attribute: Outlook
    Instance Attribute: Sunny TreeKeys : dict_keys(['Overcast', 'Rain', 'Sunny'])
    Key: dict_keys(['Humidity'])
    Attribute: Humidity
    Instance Attribute: Normal TreeKeys : dict_keys(['High', 'Normal'])
    Key: dict_keys(['Outlook'])
    Attribute: Outlook
    Instance Attribute: Rain TreeKeys : dict_keys(['Overcast', 'Rain', 'Sunny'])
```



```
[ ] Attribute: Wind
    Instance Attribute: Weak TreeKeys : dict_keys(['Strong', 'Weak'])
    Key: dict_keys(['Outlook'])
    Attribute: Outlook
    Instance Attribute: Sunny TreeKeys : dict_keys(['Overcast', 'Rain', 'Sunny'])
    Key: dict_keys(['Humidity'])
    Attribute: Humidity
    Instance Attribute: Normal TreeKeys : dict_keys(['High', 'Normal'])
    Key: dict_keys(['Outlook'])
    Attribute: Outlook
    Instance Attribute: Overcast TreeKeys : dict_keys(['Overcast', 'Rain', 'Sunny'])
    Key: dict_keys(['Outlook'])
    Attribute: Outlook
    Instance Attribute: Overcast TreeKeys : dict_keys(['Overcast', 'Rain', 'Sunny'])
    Key: dict_keys(['Outlook'])
    Attribute: Outlook
    Instance Attribute: Rain TreeKeys : dict_keys(['Overcast', 'Rain', 'Sunny'])
    Key: dict_keys(['Wind'])
    Attribute: Wind
    Instance Attribute: Strong TreeKeys : dict_keys(['Strong', 'Weak'])
    0      No
    1      No
    2     Yes
    3     Yes
    4     Yes
    5      No
    6     Yes
    7      No
    8     Yes
    9     Yes
    10    Yes
    11    Yes
```



```
[ ] 12    Yes
    13     No
    Name: predicted, dtype: object

     Accuracy is:
    1.0
```

	Play_Tennis	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