CODE:

```
import pandas as pd
import math

class Node:
    def __init__(self, l):
        self.label = l
        self.branches = {}

def entropy(data):
    total_ex = len(data)
    positive_ex = len(data.ix[data[data.columns[-1]] == "Yes"])
    negative_ex = len(data.ix[data[data.columns[-1]] == "No"])
    entropy = 0

    if(positive_ex > 0):
        entropy = -positive_ex/float(total_ex) * (math.log(positive_ex,2) -
math.log(total_ex,2))
    if(negative_ex > 0):
        entropy += -negative_ex/float(total_ex) * (math.log(negative_ex,2) -
math.log(total_ex,2))
    return entropy

def gain(S, data, attribute):
    values = set(data[attribute])
    gain = S
    for val in values:
        gain -= len(data.ix[data[attribute] == val]) / float(len(data)) *
entropy(data.ix[data[attribute] == val])
    return gain

def get_attribute(data):
    entropy_s = entropy(data)
    attribute = ""
    max_gain = 0

    for col in data.columns[:-1]:
        g = gain(entropy_s, data, col)
        if (g > max_gain):
            max_gain = g
            attribute = col

    return attribute

def decision_tree(data):
    root = Node("")
    #print(data)
    #print(data[data.columns[-1]][0])
    if (entropy(data) == 0):
        #print(data)
```

```
        root.label = list(data[data.columns[-1]])[0]
        return root
    if (len(data.columns) == 1):
        return
    else:
        attrib = get_attribute(data)
        root.label = attrib
        values = set(data[attrib])
        for val in values:
            root.branches[val] = decision_tree(data.ix[data[attrib] ==
val].drop(attrib,axis = 1))
        return root

def get_rules(root, rule='', rules=[]):
    if not root.branches:
        rules.append(rule[:-1] + '=>' + root.label)
        return rules
    for i in root.branches:
        get_rules(root.branches[i], rule + root.label + '=' + i + '^', rules)

    return rules

def tree_test(tree, test_str):
    if not tree.branches:
        return tree.label
    else:
        return tree_test(tree.branches[test_str[tree.label]], test_str)

#main
data = pd.read_csv('data.csv')
#print(data)
tree = decision_tree(data)
rules = get_rules(tree)

for rule in rules:
    print(rule)

test_str = {}
print("Enter the test cases")
for i in data.columns[:-1]:
    test_str[i] = input(i+" = ")
print("\n"+tree_test(tree, test_str))
```

OUTPUT:

```
Outlook=Overcast=>Yes
Outlook=Sunny^Humidity=High=>No
Outlook=Sunny^Humidity=Normal=>Yes
Outlook=Rainy^Wind=Strong=>No
Outlook=Rainy^Wind=Weak=>Yes
Enter the test cases
Outlook = Rainy
Temperature = High
Humidity = Normal
Wind = Strong

No
```