Problem 1

(a)

If date is also an input feature, the attribute on `Day` would be chosen at the root node.

It can't be a good choice, because after the attribute at the root node, all the leaf nodes become pure and the split process ends immediately. The decision tree based on the number of day cannot give any meaningful result in prediction.

(b)

Step 1:

Split the data with feature `Outlook`, and get the Child Nodes

`Overcast`, with classification `Yes`, as a Leaf Node;

`Rain`, as a medium node;

`Sunny`, as a medium node.

Step 2:

At node `Rain`, split the data with feature `Wind`, and get the Child Nodes

`Strong`, with classification `No`, as a Leaf Node;

`Weak`, with classification `Yes`, as a Leaf Node.

At node `Sunny`, split the data with feature `Humidity`, and get the Child Nodes

`High`, with classification `No`, as a Leaf Node;

`Normal`, with classification `Yes`, as a Leaf Node.

After the 2 steps, all the leaf nodes are pure, and the tree ends. The final decision tree can be visualized as below:

```
Chara:Outlook
  Value = Overcast and
    Label:Yes
  Value = Rain and
    Chara:Wind
      Value = Strong and
        Label:No
      Value = Weak and
        Label:Yes
  Value = Sunny and
    Chara:Humidity
      Value = High and
        Label:No
      Value = Normal and
        Label:Yes
```

(c)

Yes, a decision tree can still be learned from the data, because that in the program, data processing modules are well written: the tree splitting ends when all the features are used, and when reaches the leaf node, the most common label would given as the label of the leaf node.

Below are the re-trained decision tree

```
Chara:Humidity
  Value = High and
    Chara:Outlook
      Value = Overcast and
        Label:Yes
      Value = Rain and
        Chara:Wind
          Value = Strong and
            Label:No
          Value = Weak and
            Label:Yes
      Value = Sunny and
        Label:No
  Value = Normal and
    Chara:Outlook
      Value = Overcast and
        Label:Yes
      Value = Rain and
        Chara:Wind
          Value = Strong and
            Label:No
          Value = Weak and
            Label:Yes
      Value = Sunny and
        Label:Yes
```

Problem 2

In the program, steps (with corresponding classifiers) with improper classifier weight α_m are discarded, i.e. the steps with

$$\begin{aligned} \varepsilon_m &\geq 0.5 \\ \alpha_m \to \infty \text{ or } \alpha_m &< 0 \end{aligned} \tag{1}$$

are discarded. Below are the steps left and the value of w at the steps:

```
Step = 16
Weight w = [0.1154 0.1154 0.0769 0.0769 0.0769 0.1154 0.1154 0.1154 0.1154 0.0769]
Step = 17
Weight w = [0.1301 0.1301 0.0636 0.0636 0.0636 0.1301 0.1301 0.0954 0.1301 0.0636]
Step = 18
Weight w = [0.1551 0.1551 0.0408 0.0408 0.0408 0.1551 0.1551 0.0612 0.1551 0.0408]
Step = 19
Weight w = [0.0952 0.1858 0.025  0.025  0.0489 0.1858 0.1858 0.0375 0.1858 0.025 ]
Step = 20
Weight w = [0.0717 0.14   0.0188 0.0188 0.0567 0.2156 0.2156 0.0283 0.2156 0.0188]
Step = 21
Weight w = [0.0365 0.0712 0.0096 0.0096 0.0684 0.2603 0.2603 0.0144 0.2603 0.0096]
Step = 22
Weight w = [0.0074 0.0144 0.0019 0.0019 0.0781 0.2971 0.2971 0.0029 0.2971 0.0019]
Step = 23
Weight w = [2.2385e-04 4.3711e-04 5.8853e-05 5.8853e-05 8.0345e-02 3.0560e-01
 3.0560e-01 8.8280e-05 3.0560e-01 1.9980e-03]
```

herein the loops are confined within the proper range:

```
1    for m in range(min(self.max_step, len(wk_clsfers))):
2        ...
```

where `self.max_step` is the desired step input, and `wk_clsfers` is an array of the weak classifiers we've used.

The set of α_m and corresponding classifiers are shown as below:

```
alpha set: [0.2027 0.1551 0.3102 0.3346 0.2161 0.4321 0.8643 1.7624]
Consistency: True
['index 1 = 7',
 'index 1 = 8',
 'index 1 = 9',
 'index 1 = 10',
 'index 1 = 11',
 'index 1 = 12',
 'index 1 = 13',
 'index 1 = 14']
```

In the output, `index 1` means that the classifier uses the second feature of the data (i.e. the feature is b), and the value following is the criterion as described in the problem.