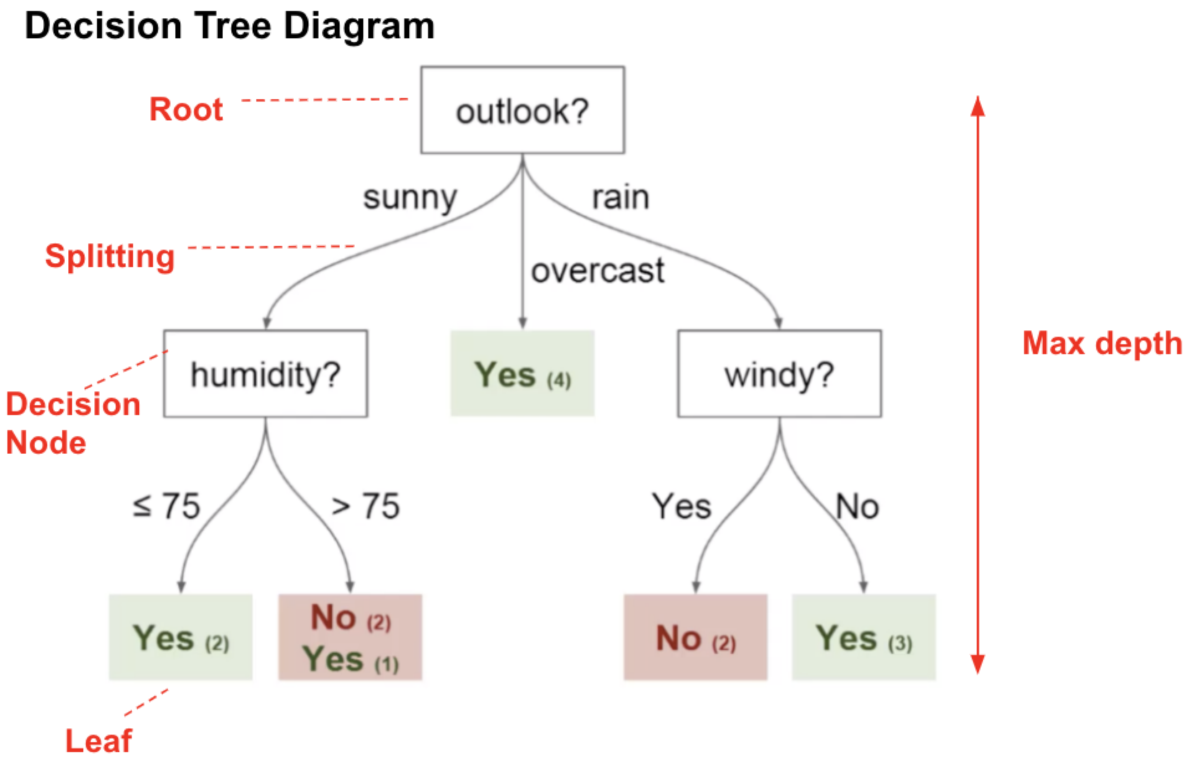
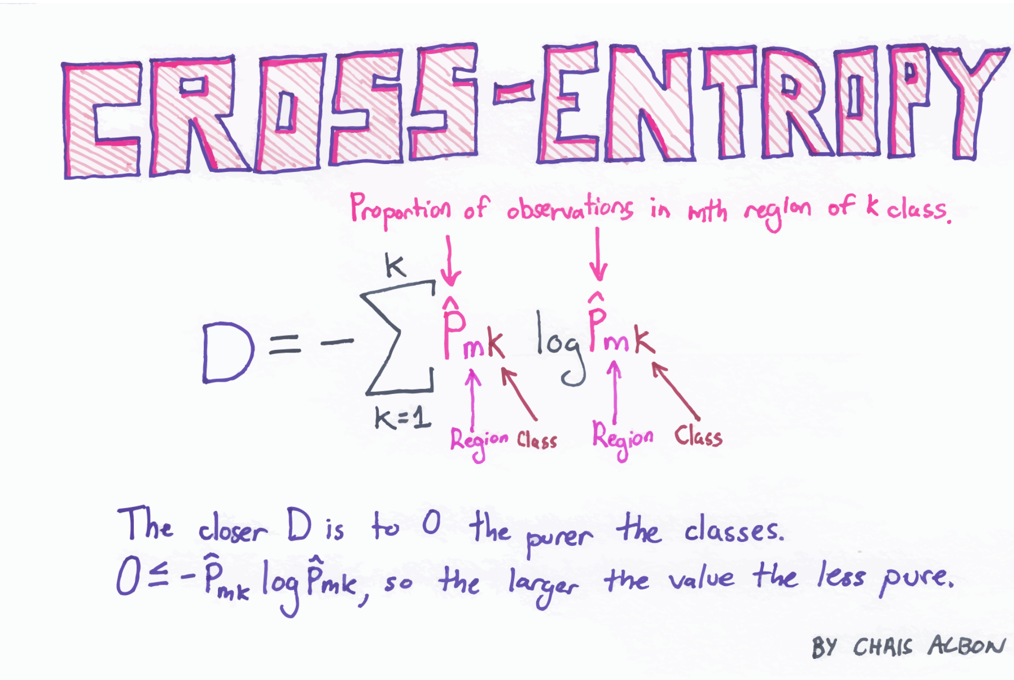
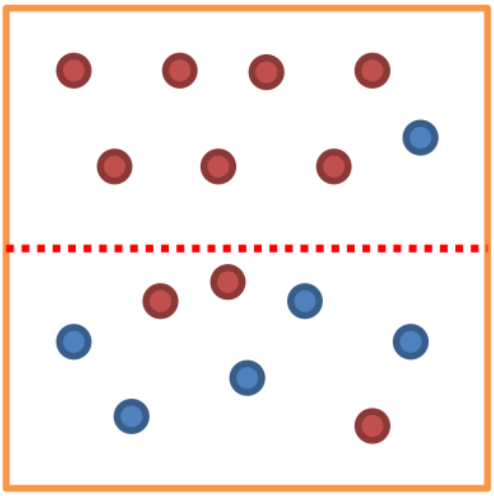

Decision Tree Diagram
Root
outlook?
sunny
rain
overcast
Splitting
humidity?
Yes (4)
windy?
Max depth
Decision Node
≤ 75
> 75
Yes
No
Yes (2)
No (2)
Yes (1)
No (2)
Yes (3)
Leaf





```python
import numpy as np
H = (-(10/16) *np.log2(10/16)) + (-(6/16) * np.log2(6/16))
H
```

```
0.9544340029249649
```

$$H(X) = -\sum_{x \in X} p(x)\log_2 p(x)$$

$$H(X \mid Y_n) = -\sum_{\substack{x \in X \\ y \in Y_n}} p(y)p(x \mid y)\log_2 p(x \mid y)$$

```
Hxy =  -0.5 * ((7/8) *np.log2(7/8) + (1/8)*np.log2(1/8))   + ((-0.5)* ((3/8)*np.log2(3/8) + (5
Hxy
```

```
0.7489992230622806
```

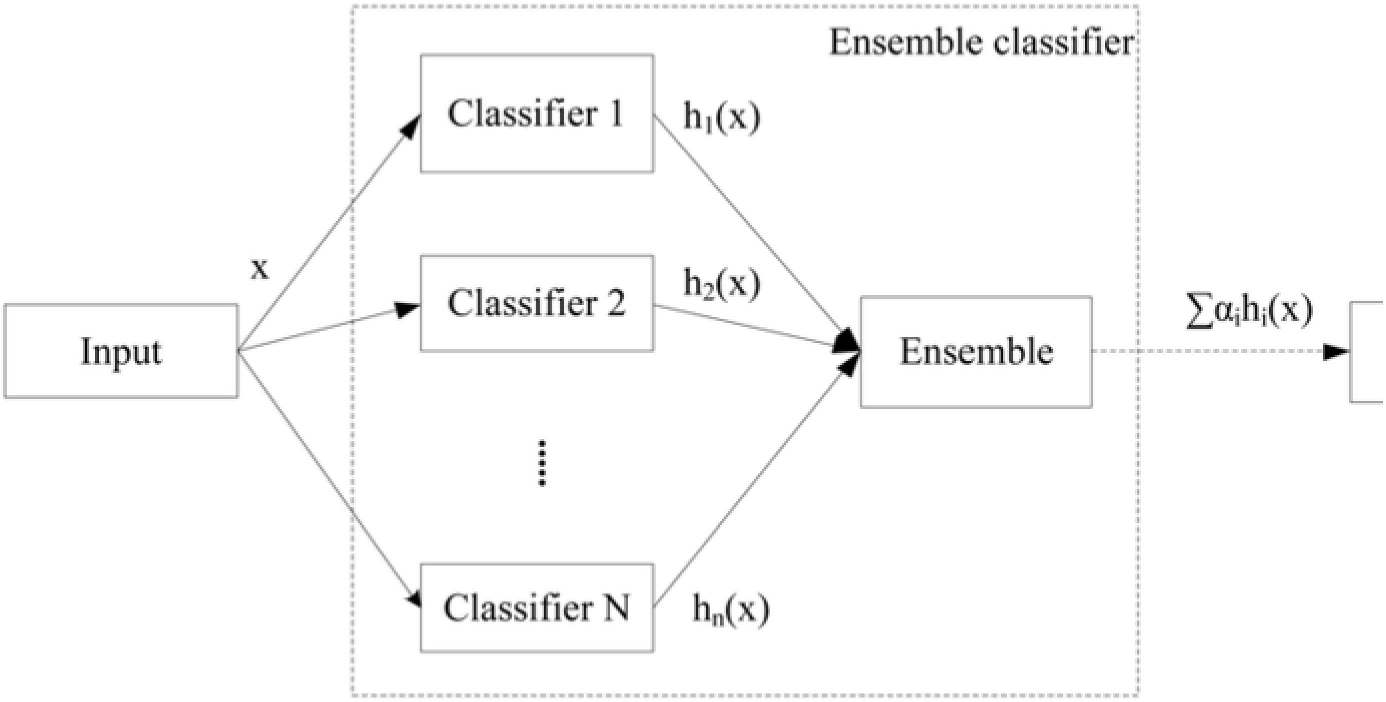
3. The formula for information gain is given by:

$$GAIN_{split} = Entropy(p) - \left(\sum_{i=1}^{k} \frac{n_i}{n} Entropy(i) \right)$$

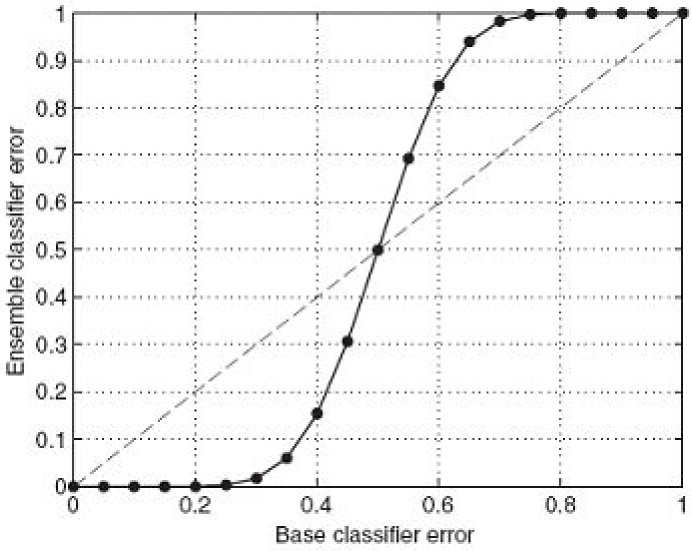
a. Using the data from question 2, calculate the information gain for splitting on Attribute1, and also for Attribute2. Which would make a better root node? why?

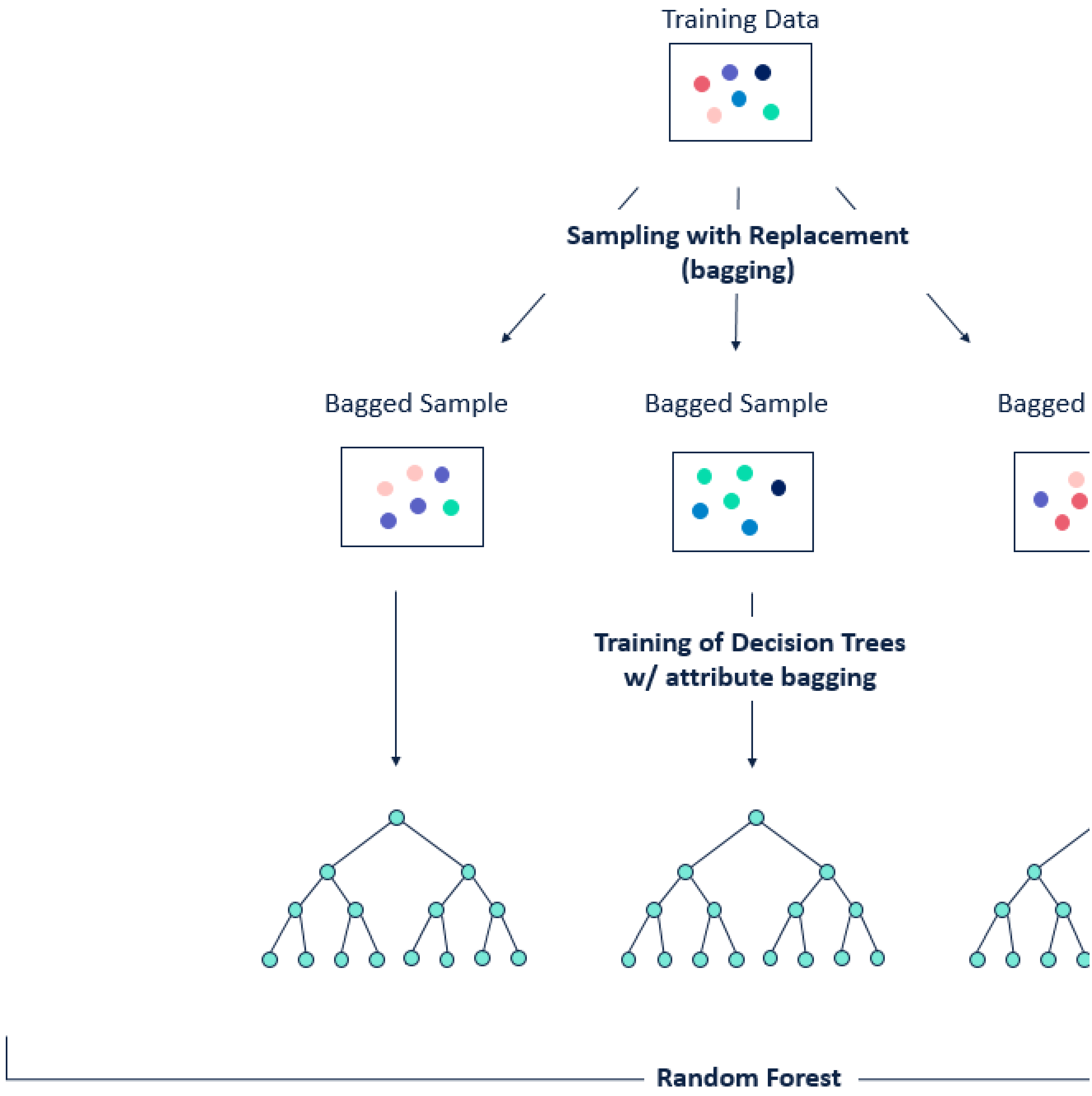
Income	Lot size	Ownership	Income	Lot size	Ownership
60.0	18.4	Owner	75.0	19.6	Non-owner
85.5	16.8	Owner	52.8	20.8	Non-owner
64.8	21.6	Owner	64.8	17.2	Non-owner
61.5	20.8	Owner	43.2	20.4	Non-owner
87.0	23.6	Owner	84.0	17.6	Non-owner
110.1	19.2	Owner	49.2	17.6	Non-owner
108.0	17.6	Owner	59.4	16.0	Non-owner
82.8	22.4	Owner	66.0	18.4	Non-owner
69.0	20.0	Owner	47.4	16.4	Non-owner
93.0	20.8	Owner	33.0	18.8	Non-owner
51.0	22.0	Owner	51.0	14.0	Non-owner
81.0	20.0	Owner	63.0	14.8	Non-owner

Income	Lot size	Ownership
51.0	14.0	Non-owner
63.0	14.8	Non-owner
59.4	16.0	Non-owner
47.4	16.4	Non-owner
85.5	16.8	Owner
64.8	17.2	Non-owner
108.0	17.6	Owner
84.0	17.6	Non-owner
49.2	17.6	Non-owner
60.0	18.4	Owner
66.0	18.4	Non-owner
33.0	18.8	Non-owner
110.1	19.2	Owner
75.0	19.6	Non-owner
69.0	20.0	Owner
81.0	20.0	Owner
43.2	20.4	Non-owner
61.5	20.8	Owner
93.0	20.8	Owner
52.8	20.8	Non-owner
64.8	21.6	Owner
51.0	22.0	Owner
82.8	22.4	Owner
87.0	23.6	Owner



$$P(X \geq 13) = \sum_{i=13}^{25} \binom{25}{i} \varepsilon^i (1-\varepsilon)^{25-i} = 0.06$$


Ensemble classifier error
Base classifier error
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
0
0.2
0.4
0.6
0.8
1


Training Data
Sampling with Replacement
(bagging)
Bagged Sample
Bagged Sample
Bagged
Training of Decision Trees
w/ attribute bagging
Random Forest

The Bagging Algorithm

Input:

Training set S

Base Learning Algorithm B

Number of bootstrap samples T

Procedure:

For i = 1 to T {

S' = bootstrap sample from S (S' is a sample with replacemen

$C_i = B(S')$ (create a new classifier from S')

}

$$C^*(x) = \underset{y \in Y}{\arg\max} \sum_{i : C_i(x) = y} 1$$ (the most often predicted label y)

Output

Classifier C*