# Homework 4

Your Name: \_\_\_\_\_Tsucheng Lu\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Student ID:\_\_\_A20525040\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**1. Build a decision Tree [50]. Note: solve this problem by manual calculations**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Color** | **Size** | **Act** | **Age** | **Inflated** |
| YELLOW | SMALL | STRETCH | Young | T |
| YELLOW | SMALL | STRETCH | Old | T |
| YELLOW | SMALL | STRETCH | Old | T |
| YELLOW | SMALL | DIP | Kid | F |
| YELLOW | SMALL | DIP | Kid | F |
| YELLOW | LARGE | STRETCH | Old | T |
| YELLOW | LARGE | STRETCH | Old | T |
| YELLOW | LARGE | DIP | Young | F |
| YELLOW | LARGE | DIP | Young | T |
| YELLOW | LARGE | DIP | Young | F |
| PURPLE | SMALL | STRETCH | Young | F |
| PURPLE | SMALL | STRETCH | Old | T |
| PURPLE | SMALL | STRETCH | Old | F |
| PURPLE | SMALL | DIP | Kid | T |
| PURPLE | SMALL | DIP | Kid | F |
| Blue | LARGE | STRETCH | Kid | T |
| Blue | LARGE | DIP | Young | T |
| Blue | LARGE | DIP | Young | F |
| Blue | LARGE | DIP | Old | T |
| Blue | LARGE | DIP | Young | T |

1). The data above is used to classify whether a balloon is inflated or not. [30]

In this question, you need to build a decision tree, but only build the 1st (root node) and 2nd levels – in other words, you need to figure out the features to be filled in the first two levels, but you do not need to find the features to be filled in the 3rd level. On the 3rd level, you need to represent the branches by using a leaf node.

You should show the process and calculations about how to build the tree.

Note: if two variables have the same largest information gain, you should choose the one with less number of the values in the variable (i.e., fewer branches). For example, if you find that act and age have the same information gain value, and the value is the largest one, then you choose act instead of age, since there are only two values in the variable act

S = [12+, 8-] entropy = -12/20log2(12/20) – 8/20log2(8/20) = 0.971

S(Yellow) = [6+, 4-] entropy = -6/10log2(6/10) – 4/10log2(4/10) = 0.971

S(purple) = [2+, 3-] entropy = -2/5log2(2/5) – 3/5log2(3/5) = 0.971

S(blue) = [4+, 1-] entropy = -4/5log2(4/5) – 1/5log2(1/5) = 0.72192

Gain (S, color) = entropy(S) – 10/20Entropy(Yellow) – 5/20Entropy(purple) – 5/20 Entropy(blue) => 0.971-0.4855-0.24275-0.18048= 0.06227

S = [12+, 8-] entropy = -12/20log2(12/20) – 8/20log2(8/20) = 0.971

S(small) = [5+, 5-] entropy = -5/10log2(5/10) – 5/10log2(5/10) = 1.0

S(large) = [7+, 3-] entropy = -7/10log2(7/10) – 3/10log2(3/10) = 0.88132

Gain (S, size) = 0.971 – 0.5 – 0.44066 = 0.03034

S = [12+, 8-] entropy = -12/20log2(12/20) – 8/20log2(8/20) = 0.971

S(stretch) = [7+, 2-] entropy = -7/9log2(7/9) – 2/9log2(2/9) = 0.764244

S(Dip) = [5+, 6-] entropy = -5/11log2(5/11) – 6/11log2(6/11) = 0.994045

Gain (S, act) = 0.971 – 0.34389 – 0.5467 = 0.08041

S = [12+, 8-] entropy = -12/20log2(12/20) – 8/20log2(8/20) = 0.971

S(young) = [4+, 4-] entropy = -4/8log2(4/8) – 4/8log2(4/8) = 1.0

S(old) = [6+, 1-] entropy = -6/7log2(6/7) – 1/7log2(1/7) = 0.5916

S(kid) = [2+, 3-] entropy = -2/5log2(2/5) – 3/5log2(3/5) = 0.971

Gain (S, age) = 0.971 – 0.4 – 0.20709 – 0.24275 = 0.12116

Gain (S, color) = 0.06227

Gain (S, size) = 0.971 – 0.5 – 0.44066 = 0.03034

Gain (S, act) = 0.971 – 0.34389 – 0.5467 = 0.08041

**Gain (S, age) = 0.971 – 0.4 – 0.20709 – 0.24275 = 0.12116**

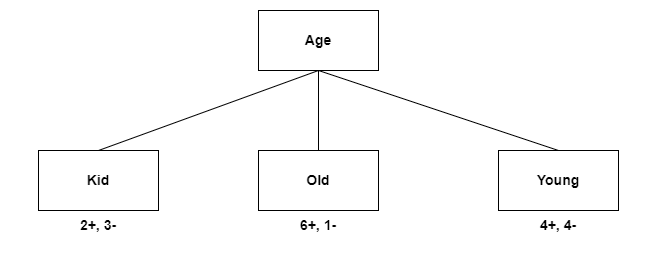


Table (Kid)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | color | size | Act | Inflated |
| 4 | Yellow | Small | Dip | F |
| 5 | Yellow | Small | Dip | F |
| 14 | Purple | Small | Dip | T |
| 15 | Purple | Small | Dip | F |
| 16 | Blue | Large | Stretch | T |

S [2+, 3-] = -2/5log2(2/5) – 3/5log2(3/5) =0.97

S(Yellow) [0+, 2-] = 0

S(Purple) [1+, 1-] = 1.0

S(Blue) [1+, 0] = 0

Gain (S, color) = 0.97 – 0 – 0.4 – 0 = 0.57

S [2+, 3-] = -2/5log2(2/5) – 3/5log2(3/5) = 0.97

S(small) [1+, 3-] = -1/4log2(1/4) – 3/4log2(3/4) =0.81125

S(large) [1+, 0] = 0

Gain (S, size) = 0.97 – 0.649 – 0 = 0.321

S [2+, 3-] = -2/5log2(2/5) – 3/5log2(3/5) = 0.97

S(Dip) [1+, 3-] = 0.81125

S(Stretch) [1+, 0] = 0

Gain (S, act) = 0.321

**Gain (S, color) = 0.97 – 0 – 0.4 – 0 = 0.57**

Gain (S, size) = 0.97 – 0.649 – 0 = 0.321

Gain (S, act) = 0.321

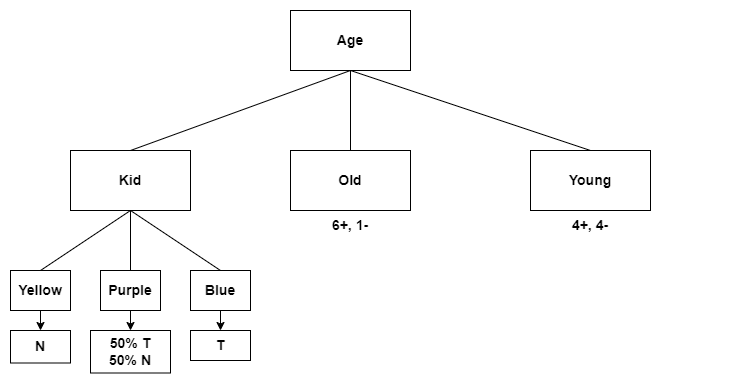


Table (Old)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Color | Size | Act | Inflated |
| 2 | Yellow | Small | Stretch | T |
| 3 | Yellow | Small | Stretch | T |
| 6 | Yellow | Large | Stretch | T |
| 7 | Yellow | Large | Stretch | T |
| 12 | Purple | Small | Stretch | T |
| 13 | Purple | Small | Stretch | F |
| 19 | Blue | Large | Dip | T |

S [6+, 1-] = -6/7log2(6/7) – 1/7log2(1/7) = 0.59168571

S(Yellow) [4+, 0-] = 0

S(Purple) [1+, 1-] = -1/2log2(1/2) – 1/2log2(1/2) = 1.0

S(Blue) [1+, 0] = 0

**Gain (S, color) = 0.59168571 – 0 – 0.28571429 – 0 = 0.30597142**

S [6+, 1-] = -6/7log2(6/7) – 1/7log2(1/7) = 0.59168571

S(Small) [3+, 1-] = -3/4log2(3/4) – 1/4log2(1/4) = 0.81125

S(Large) [3+, 0] = 0

Gain (S, size) = 0.59168571 – 0.46357143 – 0 = 0.12811428

S [6+, 1-] = -6/7log2(6/7) – 1/7log2(1/7) = 0.59168571

S(stretch) [5+, 1-] = -5/6log2(5/6) – 1/6log2(1/6) = 0.650025

S(Dip) [1+, 0] = 0

Gain (S, Act) = 0.59168571 – 0.55716429 – 0 = 0.03452142

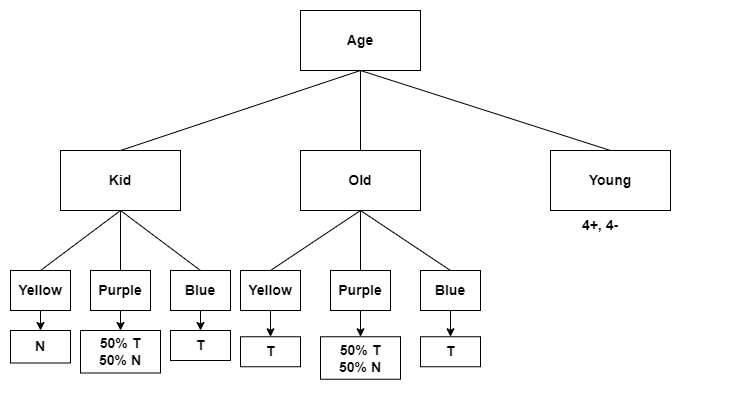


Table (Young)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Color | Size | Act | Inflated |
| 1 | Yellow | Small | Stretch | T |
| 8 | Yellow | Large | Dip | F |
| 9 | Yellow | Large | Dip | T |
| 10 | Yellow | Large | Dip | F |
| 11 | Purple | Small | Stretch | F |
| 17 | Blue | Large | Dip | T |
| 18 | Blue | Large | Dip | F |
| 20 | Blue | Large | Dip | T |

S [4+, 4-] = -4/8log2(4/8) -4/8log2(4/8) = 1.0

S(Yellow) [2+, 2-] = -2/4log2(2/4) -2/4log2(2/4) = 1.0

S(Purple) [0+, 1-] = 0

S(Blue) [2+, 1-] = 2/3log2(2/3) – 1/3log2(1/3) = 0.9183

Gain(color) = 1.0 – 0.5 – 0 – 0.3443 = 0.1556

S [4+, 4-] = -4/8log2(4/8) -4/8log2(4/8) = 1.0

S(small) [1+, 1-] = 1.0

S(large) [3+, 3-] = 1.0

Gain(size) = 1.0 – 2/8 – 6/8 = 0

S [4+, 4-] = -4/8log2(4/8) -4/8log2(4/8) = 1.0

S(Stretch) [1+, 1-] = 1.0

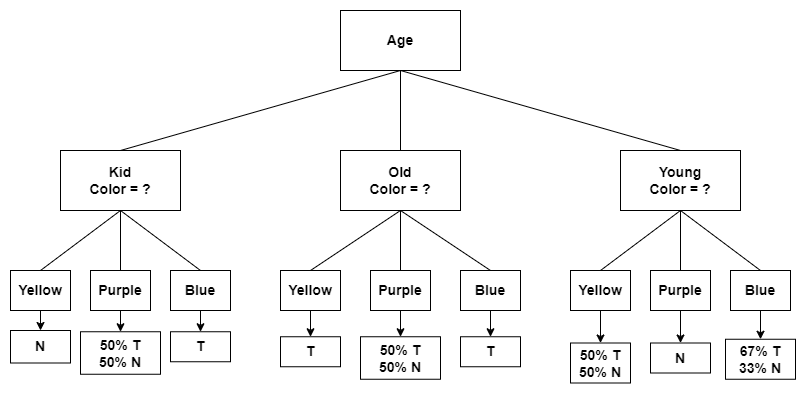
S(Dip) [3+, 3-] = 1.0

Gain(act) = 1.0 – 2/8 – 6/8 = 0

**Gain(color) = 1.0 – 0.5 – 0 – 0.3443 = 0.1556**

Gain(size) = 1.0 – 2/8 – 6/8 = 0

Gain(act) = 1.0 – 2/8 – 6/8 = 0



2). On the 3rd level of the tree, represent them by using leaf nodes (i.e., labels) on this level – you can simply use the majority of the label in each group as the predicted label. Given a new data “YELLOW, SMALL, DIP, YOUNG”, make your prediction [10]

If the new data is YELLOW, SMALL, DIP, YOUNG, then the label will be the 50% T or 50% N.

3). List at least three solutions to alleviate overfitting in decision tree [10]

1. early stopping

First of the solutions is that implement early stopping by monitoring the models during training. And if the performance degrades, we can stop training to prevent further overfitting.

2. Feature selection

good feature selection can help decrease overfitting. Considering feature carefully can reduce complexity of the decision tree and can lead to a more accurate model. Additionally, we can create new features that capture important information in the data.

3. Increasing Sample Size

Sometimes overfitting occurs because of a limited amount of training data. Generating more data can help the decision tree model generalize better and reduce overfitting.

**2. (50 points) Python practice for Decision Tree classification**

**Run decision tree techniques to find the best parameters and performance**

* Use Malware\_MultiClass.csv data by using 10-fold cross validation
* Build decision tree models

Note:

* You need to change different/multiple parameters to find the best model.
* You should try multiple models you have built, with the parameters you set
* Find the best model by using AUC score as the metric
* Finally, visualize the best tree model, Hint: you need to select a model from the 10-fold cross validation. See the example coding in the class.
* You can find data sets from “slide & data” on blackboard system

Submission

* The ipynb and saved html files

In conclusion, model [3] has a highest value in all of the model. So, model 3 has the best model.

