CSCI 6951 - Data Warehousing and Data Mining

# Lab Decision Trees ID3

## Predicting Signups

Sometimes when a high-traffic site links to a new application that offers free accounts and subscription accounts, the application will get thousands of new users. Many of these users are driven by curiosity and are not really looking for that particular type of application, so there is a low likelihood that they will become paying customers. This makes it difficult to distinguish and follow up with likely customers, so many sites report to mass-emailing everyone who has signed up, rather than using a more targeted approach.

To help with this problem, it would be useful to be able to predict the likelihood that a user will become a paying customer, and the factors that indicate if a user will become a customer. The former information can be used to guide an advertisement strategy and to make certain aspects of the site more accessible.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Referrer** | **Location** | **Read FAQ** | **Pages Viewed** | **Service Chosen** |
| 1 | Slashdot | USA | Yes | Mid | None |
| 2 | Google | France | Yes | High | Premium |
| 3 | Digg | USA | Yes | High | Basic |
| 4 | Kiwitobes | France | Yes | High | Basic |
| 5 | Google | UK | No | Mid | Premium |
| 6 | (direct) | New Zealand | No | Low | None |
| 7 | (direct) | UK | No | Mid | Basic |
| 8 | Google | USA | No | High | Premium |
| 9 | Slashdot | France | Yes | Mid | None |
| 10 | Digg | USA | No | Mid | None |
| 11 | Google | UK | No | Mid | None |
| 12 | Kiwitobes | UK | No | Mid | None |
| 13 | Digg | New Zealand | Yes | Low | Basic |
| 14 | Google | UK | Yes | Mid | Basic |
| 15 | Kiwitobes | France | Yes | Mid | Basic |

**Question 1.** In order to build a decision tree based on the data above, first we have to select one attribute for the root node by calculating the information gain for each possible attribute, as shown in the lecture. Please hand in the resulting tree including all the calculation steps.

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Referrer | | | | Location | | | | Read FAQ | | | |
|  | none | basic | pre. |  | none | basic | pre. |  | none | basic | pre. |
| Slashdot | 3 | 0 | 0 | USA | 2 | 2 | 1 | Yes | 3 | 5 | 1 |
| Google | 1 | 1 | 4 | France | 1 | 2 | 1 | No | 4 | 2 | 3 |
| Digg | 1 | 3 | 0 | UK | 2 | 2 | 2 |  |  |  |  |
| Kiwitobes | 1 | 2 | 0 | New Zealand | 2 | 1 | 0 |  |  |  |  |
| (direct) | 1 | 1 | 0 |  |  |  |  |  |  |  |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Pages Viewed | | | | Service Chosen | | |
|  | none | basic | pre. | none | basic | premium |
| Low | 1 | 2 | 0 | 7 | 7 | 4 |
| Mid | 5 | 3 | 2 |  |  |  |
| high | 1 | 2 | 2 |  |  |  |

* Referrer
  + Slashdot
    - Info ([3, 0, 0]) = entropy (3,0,0) = -(3/3) \* log2(3/3)– 0 – 0 = 0
  + Google
    - Info ([1, 1,4]) = entropy (1,1,4) = 1.2516291673878228
  + Digg
    - Info ([1, 3, 0]) = entropy (1, 3, 0) = 0.8112781244591328
  + Kiwitobes
    - Info ([1, 2, 0]) = entropy (1,2,0) = 0.9182958340544896
  + (direct)
    - Info ([1, 1, 0]) = entropy (1,1,0) = 1
    - Info ([3, 0, 0], [1, 1, 4], [1, 3, 0], [1, 2, 0], [1, 1, 0]) = 0.8616541669070521
* Location
  + USA
    - Info ([2, 2, 1]) = entropy (2, 2, 1) = 1.5219280948873621
  + France
    - Info ([1, 2, 1]) = entropy (1,2,1) = 1.5
  + UK
    - Info ([2, 2, 2]) = entropy (2, 2, 2) = 1.5849625007211563
  + New Zealand
    - Info ([2, 1,0]) = entropy (2, 1, 0) = 0.9182958340544896
    - Info ([2, 2,1], [1,2, 1], [2,2,2], [2,1,0]) = 1.4374612767181787

* Read FAQ
  + Yes
    - Info ([3, 5, 1]) = entropy (3, 5, 1) = 1.3516441151533922
  + No
    - Info ([4, 2,3]) = entropy (4, 2, 3) = 1.5304930567574826
    - Info ([3, 5, 1], [4, 2, 3]) = 1.4410685859554375
* Pages Viewed
  + Low
    - Info ([1, 2, 0]) = entropy (1, 2, 0) = 0.9182958340544896
  + Mid
    - Info ([5, 3, 2]) = entropy (5, 3, 2) = 1.4854752972273344

* + High
    - Info ([1, 2, 2]) = entropy (1, 2, 2) = 1.5219280948873621
    - Info ([9, 2], [0, 1], [0, 2]) = 0.8128103145334136
* Service Chosen
  + Info ([7, 7, 4]) = entropy (7, 7, 4) = 1.54198228
* **Gain (referrer) = 1.54 -** 0.8616541669070521 = 0.67834583309
* **Gain (location) =** 1.54 - 1.4374612767181787 = 0.10253872328
* **Gain (read FAQ) =** 1.54 - 1.4410685859554375 = 0.09893141404
* **Gain (pages viewed) = 1.54 -** 0.8128103145334136 = 0.72718968546

# The attribute selected as the splitting attribute at the root of the tree is: **Pages\_Viewed**

**Question 2.** In Python run both Id3 and CART algorithms using the training set as the test option and record Correctly Classified Instances, Incorrectly Classified Instances, Accuracy%, Kappa Statistics, a, b, c (from the confusion matrix), Time taken to build model, Number of leaf nodes. Which algorithm performs better?

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Correctly CI | Incorrectly  CI | Accuracy% | Error% | a | b | c | Time | Number of leaves |
| **Id3** | 17 | 1 | 0.9444444444444444 | 0.05555555555 | 7 | 6 | 5 | 0 ns | 12 |
| **CART** | 13 | 5 | 72.22 | 0.5833 | 5 | 6 | 7 | 0ns | 5 |

Id3 Performs better

**Question 3.** Save from Weka, the decision tree generated by **Id3** (only text mode) and the decision tree generated by the **CART**. Copy-paste the trees below. Are the two-decision tress different?

**ID3 tree:**

[Text(223.2, 201.90857142857143, 'X[0] <= 3.5\ngini = 0.648\nsamples = 18\nvalue = [7, 7, 4]'),

Text(200.88, 170.84571428571428, 'X[2] <= 0.5\ngini = 0.64\nsamples = 15\nvalue = [7, 4, 4]'),

Text(111.6, 139.78285714285715, 'X[0] <= 1.5\ngini = 0.642\nsamples = 9\nvalue = [2, 4, 3]'),

Text(44.64, 108.72, 'X[1] <= 1.0\ngini = 0.5\nsamples = 4\nvalue = [2, 2, 0]'),

Text(22.32, 77.65714285714284, 'gini = 0.0\nsamples = 1\nvalue = [0, 1, 0]'),

Text(66.96000000000001, 77.65714285714284, 'X[0] <= 0.5\ngini = 0.444\nsamples = 3\nvalue = [2, 1, 0]'),

Text(44.64, 46.59428571428572, 'gini = 0.0\nsamples = 1\nvalue = [1, 0, 0]'),

Text(89.28, 46.59428571428572, 'X[3] <= 1.5\ngini = 0.5\nsamples = 2\nvalue = [1, 1, 0]'),

Text(66.96000000000001, 15.531428571428563, 'gini = 0.0\nsamples = 1\nvalue = [1, 0, 0]'),

Text(111.6, 15.531428571428563, 'gini = 0.0\nsamples = 1\nvalue = [0, 1, 0]'),

Text(178.56, 108.72, 'X[0] <= 2.5\ngini = 0.48\nsamples = 5\nvalue = [0, 2, 3]'),

Text(156.24, 77.65714285714284, 'X[1] <= 2.5\ngini = 0.375\nsamples = 4\nvalue = [0, 1, 3]'),

Text(133.92000000000002, 46.59428571428572, 'gini = 0.444\nsamples = 3\nvalue = [0, 1, 2]'),

Text(178.56, 46.59428571428572, 'gini = 0.0\nsamples = 1\nvalue = [0, 0, 1]'),

Text(200.88, 77.65714285714284, 'gini = 0.0\nsamples = 1\nvalue = [0, 1, 0]'),

Text(290.16, 139.78285714285715, 'X[3] <= 0.5\ngini = 0.278\nsamples = 6\nvalue = [5, 0, 1]'),

Text(267.84000000000003, 108.72, 'X[0] <= 1.5\ngini = 0.444\nsamples = 3\nvalue = [2, 0, 1]'),

Text(245.52, 77.65714285714284, 'gini = 0.0\nsamples = 1\nvalue = [1, 0, 0]'),

Text(290.16, 77.65714285714284, 'X[0] <= 2.5\ngini = 0.5\nsamples = 2\nvalue = [1, 0, 1]'),

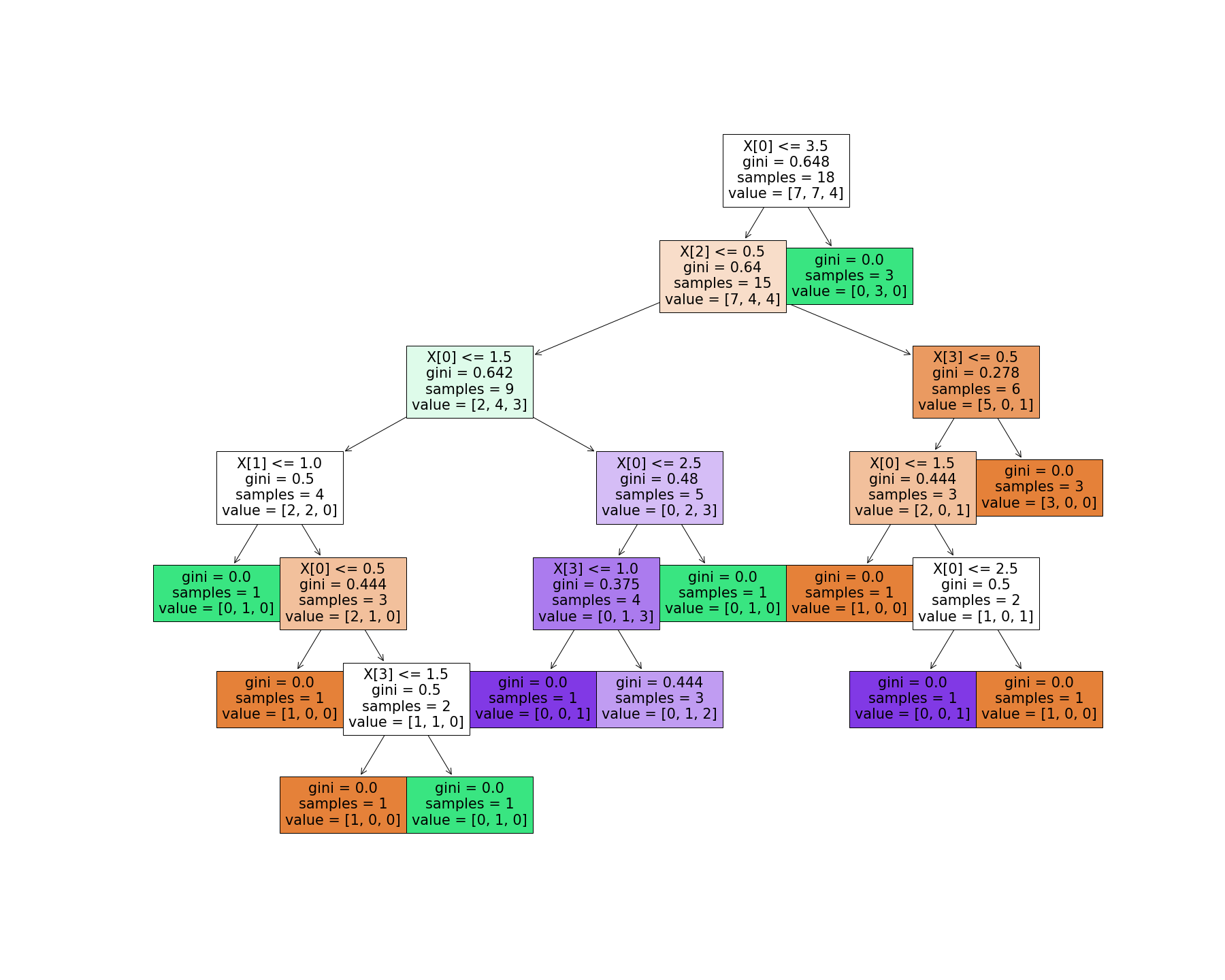
Text(267.84000000000003, 46.59428571428572, 'gini = 0.0\nsamples = 1\nvalue = [0, 0, 1]'),

Text(312.48, 46.59428571428572, 'gini = 0.0\nsamples = 1\nvalue = [1, 0, 0]'),

Text(312.48, 108.72, 'gini = 0.0\nsamples = 3\nvalue = [3, 0, 0]'),

Text(245.52, 170.84571428571428, 'gini = 0.0\nsamples = 3\nvalue = [0, 3, 0]')]

**CART tree:**



No, both the decision trees are similar.

**Question 4.** Classify the following test instances using first the **Id3** decision tree and second the **CART** decision tree. Record in the following tables the decision obtained.

**Id3:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **referrer** | **location** | **Read FAQ** | **Pages viewed** | **Service chosen** | **service – ID3** |
| Google | UK | no | mid | **premium** |  |
| Digg | USA | no | low | basic |  |
| Slashdot | New Zealand | yes | high | none |  |

**CART:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **referrer** | **location** | **Read FAQ** | **Pages viewed** | **Service chosen** | **service – CART** |
| Google | UK | no | mid | **premium** |  |
| Digg | USA | no | low | basic |  |
| Slashdot | New Zealand | yes | high | none |  |

**Question 5.** In Weka run both **Id3** and **CART** algorithms using the cross-validation test option and record Correctly Classified Instances, Incorrectly Classified Instances, Accuracy%, Kappa Statistics, a, b, c (from the confusion matrix), Time taken to build model, Number of leaf nodes. Which algorithm performs better?

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Correctly CI | Incorrectly  CI | Accuracy% | Error% | a | b | c | Time | Number of leaves |
| **Id3** | 8  44.4444 % | 9  50% | 44.44 | 55.56 | 6 | 3 | 8 | 0 seconds |  |
| **CART** | 9 50% | 9 50% | 50 | 50 | 6 | 6 | 6 | 0.01 seconds | 4 |

**ID3 Kappa statistic: 0.1818**

**CART Kappa statistic: 0.2394**

**UnClassified Instances (ID3): 1 (5.5556 %)**

**Cross- validation fold set to 10**