**Στέργιος Τσάνταλης (iis21125)**

**(Α)Δέντρο απόφασης με το χέρι**

1. Το σύνολο των στοιχείων είναι **16**. Για το χαρακτηριστικό **Poisonous** συγκεκριμένα είναι **8 Yes** και **8 No**. Στη συνέχεια υπολογίζουμε το **Gini** του **Poisonous**.

Gini (Poisonous) = 1 – ((Πλήθος Yes/Σύνολο στοιχείων)2 +(Πλήθος No/Σύνολο στοιχείων)2)= 1 – ((8/16)2 +(8/16)2)= 1 – (64/256 + 64/256) = 1 – 128/256 = 128/256 = **0.5**

Κάνουμε διαχωρισμό με βάση το χαρακτηριστικό **Height**. Για το **Tall** έχουμε **4 Yes** και **5 No**. Για το **Short** έχουμε **4 Yes** και **3** **No**.Υπολογίζουμε το **Gini** των **Tall** και **Short**.

Gini(Tall) = 1 – ((Πλήθος Yes στο Tall/Σύνολο Tall)2 + (Πλήθος No στο Tall/Σύνολο Tall)2)= 1 – ((4/9)2 +(5/9)2)= 1 – (16/81 + 25/81)= 1 – 41/81 = 40/81 = **0.49**

Gini(Short) = 1 - ((Πλήθος Yes στο Short/Σύνολο Short)2 + (Πλήθος No στο Short/Σύνολο Short)2)= 1 – ((4/7)2 + (3/7)2)= 1 – (16/49 + 9/49)= 1 – 25/49 = 24/49 = **0.48**

Οπότε βρίσκουμε το συνολικό Gini παιδιών και έχουμε:

(Σύνολο Tall στοιχείων/Σύνολο στοιχείων) \* Gini(Tall) + (Σύνολο Short στοιχείων/Σύνολο στοιχείων) \* Gini(Short)= (9/16)\*0.49 + (7/16)\*0.48 = **0.48 <= Gini(Height)**

**ΕΠΟΜΕΝΩΣ** => Gain(Height) = Gini(Poisonous) – Gini(Height) = 0.5 – 0.48 = **0.02**

Κάνουμε διαχωρισμό με βάση το χαρακτηριστικό **Stripes**. Για το **Yes** έχουμε **7 Yes** και **2 No**. Για το **No** έχουμε **1 Yes** και **6** **No**.Υπολογίζουμε το **Gini** των **Yes** και **No**.

Gini(Yes) = 1 – ((Πλήθος Yes στο Yes/Σύνολο Yes)2 + (Πλήθος No στο Yes/Σύνολο Yes)2)= 1 – ((7/9)2 +(2/9)2)= 1 – (49/81 + 4/81)= 1 – 53/81 = 28/81 = **0.34**

Gini(No) = 1 - ((Πλήθος Yes στο No/Σύνολο No)2 + (Πλήθος No στο No/Σύνολο No)2)= 1 – ((1/7)2 + (6/7)2)= 1 – (1/49 + 36/49)= 1 – 37/49 = 12/49 = **0.24**

Οπότε βρίσκουμε το συνολικό Gini παιδιών και έχουμε:

(Σύνολο Yes στοιχείων/Σύνολο στοιχείων) \* Gini(Yes) + (Σύνολο No στοιχείων/Σύνολο στοιχείων) \* Gini(No)= (9/16)\*0.34 + (7/16)\*0.24 = **0.29 <= Gini(Stripes)**

**ΕΠΟΜΕΝΩΣ** => Gain(Stripes) = Gini(Poisonous) – Gini(Stripes) = 0.5 – 0.29 = **0.31**

Κάνουμε διαχωρισμό με βάση το χαρακτηριστικό **Colour**.

|  |  |  |  |
| --- | --- | --- | --- |
|  | D1 | D2 | D3 |
|  | Purple | Red | Blue |
| Yes | 4 | 0 | 4 |
| No | 2 | 4 | 2 |

Υπολογίζουμε το **Gini** των **D1,D2** και **D3**.

Gini(D1) = 1 – ((Πλήθος Yes στο D1/Σύνολο D1)2 + (Πλήθος No στο D1/Σύνολο D1)2)= 1 – ((4/6)2 +(2/6)2)= 1 – (16/36 + 4/36)= 1 – 20/36 = 16/36 = **0.44**

Gini(D2) = 1 - ((Πλήθος Yes στο D2/Σύνολο D2)2 + (Πλήθος No στο D2/Σύνολο D2)2)= 1 – ((0/4)2 + (4/4)2)= 1 – (0 + 1)= 1 – 1 = **0**

Gini(D3) = 1 – ((Πλήθος Yes στο D3/Σύνολο D3)2 + (Πλήθος No στο D3/Σύνολο D3)2)= 1 – ((4/6)2 +(2/6)2)= 1 – (16/36 + 4/36)= 1 – 20/36 = 16/36 = **0.44**

Οπότε βρίσκουμε το συνολικό Gini παιδιών και έχουμε:

(Σύνολο D1 στοιχείων/Σύνολο στοιχείων) \* Gini(D1) + (Σύνολο D2 στοιχείων/Σύνολο στοιχείων) \* Gini(D2) + (Σύνολο D3 στοιχείων/Σύνολο στοιχείων) \* Gini(D3)= (6/16)\*0.44 + 0 + (6/16)\*0.44 = **0.33 <= Gini(Colour)**

**ΕΠΟΜΕΝΩΣ** => Gain(Colour) = Gini(Poisonous) – Gini(Colour) = 0.5 – 0.33 = **0.27**

Κάνουμε διαχωρισμό με βάση το χαρακτηριστικό **Texture**.

|  |  |  |  |
| --- | --- | --- | --- |
|  | D4 | D5 | D6 |
|  | Rough | Smooth | Hairy |
| Yes | 2 | 3 | 3 |
| No | 2 | 3 | 3 |

Υπολογίζουμε το **Gini** των **D4,D5** και **D6**.

Gini(D4) = 1 – ((Πλήθος Yes στο D4/Σύνολο D4)2 + (Πλήθος No στο D4/Σύνολο D4)2)= 1 – ((2/4)2 +(2/4)2)= 1 – (4/16 + 4/16)= 1 – 8/16 = 8/16 = **0.5**

Gini(D5) = 1 - ((Πλήθος Yes στο D5/Σύνολο D5)2 + (Πλήθος No στο D5/Σύνολο D5)2)= 1 – ((3/6)2 + (3/6)2)= 1 – (9/36 + 9/36)= 1 – 18/36 = 18/36 = **0.5**

Gini(D6) = 1 – ((Πλήθος Yes στο D6/Σύνολο D6)2 + (Πλήθος No στο D6/Σύνολο D6)2)= 1 – ((3/6)2 +(3/6)2)= 1 – (9/36 + 9/36)= 1 – 18/36 = 18/36 = **0.5**

Οπότε βρίσκουμε το συνολικό Gini παιδιών και έχουμε:

(Σύνολο D4 στοιχείων/Σύνολο στοιχείων) \* Gini(D4) + (Σύνολο D5 στοιχείων/Σύνολο στοιχείων) \* Gini(D5) + (Σύνολο D6 στοιχείων/Σύνολο στοιχείων) \* Gini(D6)= (4/16)\*0.5 + (6/16)\*0.5 + (6/16)\*0.5 = **0.5 <= Gini(Texture)**

**ΕΠΟΜΕΝΩΣ** => Gain(Texture) = Gini(Poisonous) – Gini(Texture) = 0.5 – 0.5 = **0**

Ως ρίζα θα επιλέξουμε αυτή με το μεγαλύτερο Gain. Στην προκειμένη περίπτωση είναι το χαρακτηριστικό Stripes (με Gain = 0.31), στο οποίο έχουμε 13/16 σωστές εγγραφές => 81.25% ακρίβεια. Αυτό σημαίνει ότι 3/16 εγγραφές κατηγοριοποιούνται λάθος.

1. Εφαρμόζοντας τον αλγόριθμο J48 στο training dataset το Weka μας δίνει το παρακάτω output:

=== Classifier model (full training set) ===

J48 pruned tree

------------------

Stripes = Yes

| Colour = Purple: Yes (3.0)

| Colour = Red: No (2.0)

| Colour = Blue: Yes (4.0)

Stripes = No: No (7.0/1.0)

Number of Leaves : 4

Size of the tree : 6

Time taken to build model: 0 seconds

=== Evaluation on training set ===

Time taken to test model on training data: 0 seconds

=== Summary ===

Correctly Classified Instances 15 93.75 %

Incorrectly Classified Instances 1 6.25 %

Kappa statistic 0.875

Mean absolute error 0.1071

Root mean squared error 0.2315

Relative absolute error 21.4286 %

Root relative squared error 46.291 %

Total Number of Instances 16

=== Detailed Accuracy By Class ===

TP Rate FP Rate Precision Recall F-Measure MCC ROC Area PRC Area Class

0,875 0,000 1,000 0,875 0,933 0,882 0,953 0,946 Yes

1,000 0,125 0,889 1,000 0,941 0,882 0,953 0,917 No

Weighted Avg. 0,938 0,063 0,944 0,938 0,937 0,882 0,953 0,932

=== Confusion Matrix ===

a b <-- classified as

7 1 | a = Yes

0 8 | b = No

Οπότε ο αλγόριθμος J48 επιλέγει ως ρίζα του δέντρου το χαρακτηριστικό **Poisonous**.

1. Για το συγκεκριμένο test dataset και τα δύο δέντρα πετυχαίνουν ακρίβεια 50%, καθώς και τα δύο δίνουν τα σωστά αποτελέσματα για τα 2/4 του dataset.

**(Β) Μελέτη Περίπτωσης με το Weka**

Μετά από δοκιμή μερικών τιμών για το minNumObj το καλύτερο μοντέλο φαίνεται πως είναι αυτό για την τιμή 13. Έχει ακρίβεια 96.3542 % και τον παρακάτω πίνακα σύγχυσης:

=== Confusion Matrix ===

a b c d <-- classified as

1190 18 2 0 | a = unacc

10 362 10 2 | b = acc

0 10 57 2 | c = good

0 9 0 56 | d = vgood

Οι κατηγορίες που δεν καλύπτονται ικανοποιητικά είναι οι c (good) και d(vgood) και προκύπτουν από τον πίνακα σύγχυσης.

**ΚΑΝΟΝΕΣ ΓΙΑ ΚΑΘΕ ΚΑΤΗΓΟΡΙΑ**

* **Unacc**: persons <= 2
* **Acc**: persons > 2, safety > 1, buying > 24.47,

maint <= 3.76, lug\_boot > 343, maint <= 3.14,

safety > 2

* **Good**: persons > 2, safety > 1, buying <= 24.47,

maint < 2.78, safety <= 2, lug\_boot > 363,

buying <= 12.5

* **Vgood**: persons > 2, safety > 1, buying > 24.47,

maint < 2.78, safety > 2, lug\_boot > 278,

doors > 3