Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών
Αριστοτέλειο Πανεπιστήμιο Θεσσαλονίκης
Τομέας Ηλεκτρονικής και Υπολογιστών

9ο Εξάμηνο
Φθινόπωρο 2019

Δένδρα Απόφασης στην R

1 Εισαγωγή

1.1 Εισαγωγή στα Δένδρα Απόφασης

Το δένδρο απόφασης είναι ένας αλγόριθμος μηχανικής μάθησης που χρησιμοποιείται κυρίως σε προβλήματα ταξινόμησης. Εφαρμόζεται σε κατηγορικά και συνεχή δεδομένα. Η βασική ιδέα του αλγορίθμου είναι ο διαδοχικός διαχωρισμός των δεδομένων με βάση κανόνες που κατασκευάζονται σύμφωνα με κάποιο κριτήριο διαχωρισμού.

Τα δένδρα απόφασης είναι υπολογιστικά εύκολα στην κατασκευή και πολύ γρήγορα στην ταξινόμηση νέων εγγραφών. Επιπλέον είναι αρκετά κατανοητά για δένδρα μικρού μεγέθους και για αυτό χρησιμοποιούνται πολλές φορές για εξερεύνηση των δεδομένων. Ένα βασικό μειονέκτημα είναι ότι ενδέχεται να παρουσιαστεί overfitting, το οποίο αντιμετωπίζεται κάνοντας pruning ή γενικότερα θέτοντας περιορισμούς κατά την κατασκευή του δένδρου.

1.2 Δένδρα Απόφασης στην R

Για την κατασκευή δένδρων απόφασης στην R, μπορούμε να χρησιμοποιήσουμε τη βιβλιοθήκη rpart:

```
> library(rpart)
```

Επιπλέον, χρησιμοποιούμε και τη βιβλιοθήκη rpart.plot με την οποία μπορούμε να σχεδιάσουμε καλύτερα τα παραγόμενα δένδρα:

```
> library(rpart.plot)
```

Για να κατασκευάσουμε ένα δένδρο απόφασης τρέχουμε την εντολή rpart:

```
> model <- rpart(Target ~ ., method = "class", data = ..., minsplit = ..., minbucket = ..., cp = ...)
```

Για να κάνουμε plot το δένδρο μπορούμε να τρέξουμε την `plot(model)` και την `text(model, use.n = TRUE)`. Εναλλακτικά, για ένα δένδρο με περισσότερες πληροφορίες μπορούμε να τρέξουμε την παρακάτω εντολή:

```
> rpart.plot(model, extra = 104, nn = TRUE)
```

Για να δούμε τις παραμέτρους της rpart, εκτός από την εντολή `?rpart`, είναι χρήσιμο να τρέξουμε την εντολή `?rpart.control` ώστε να δούμε τις παρακάτω βασικές παραμέτρους:

`minsplit`: το ελάχιστο πλήθος παρατηρήσεων που πρέπει να έχει ένας κόμβος ώστε να διαχωριστεί

`minbucket`: το ελάχιστο επιτρεπτό πλήθος παρατηρήσεων σε κάθε φύλλο του δένδρου

`maxdepth`: το μέγιστο βάθος του δένδρου

`cp`: παράμετρος που ελέγχει αν το complexity για κάποιο διαχωρισμό είναι επιτρεπτό και τίθεται εμπειρικά (όσο μεγαλώνει η τιμή της, τόσο περισσότερο γίνεται pruning στο δένδρο)

2 Κριτήρια Διαχωρισμού και Κατασκευή Δένδρων Απόφασης

Για τον υπολογισμό κριτηρίων διαχωρισμού και την κατασκευή δένδρου απόφασης θα χρησιμοποιήσουμε ως εφαρμογή τα δεδομένα εκπαίδευσης του παρακάτω πίνακα για ένα πρόβλημα δυαδικής ταξινόμησης.

Outlook	Temperature	Humidity	Play
Sunny	Hot	High	No
Sunny	Hot	Low	No
Rainy	Hot	Low	Yes
Rainy	Cool	High	Yes
Rainy	Cool	Low	Yes
Rainy	Hot	Low	No
Rainy	Cool	Low	Yes
Sunny	Hot	High	No
Sunny	Cool	Low	Yes
Rainy	Hot	Low	Yes
Sunny	Cool	Low	Yes
Rainy	Hot	High	Yes
Rainy	Cool	Low	Yes
Sunny	Cool	High	No

Στο παραπάνω dataset επιθυμούμε να εφαρμόσουμε έναν αλγόριθμο δένδρου απόφασης. Θα απαντήσουμε στα παρακάτω ερωτήματα:

α) Ποιό χαρακτηριστικό από τα Outlook, Temperature, Humidity είναι καλύτερο να χρησιμοποιηθεί στον πρώτο διαχωρισμό, με βάση το δείκτη Gini;

β) Ποιό χαρακτηριστικό από τα Outlook, Temperature, Humidity είναι καλύτερο να χρησιμοποιηθεί στον πρώτο διαχωρισμό, με βάση το δείκτη κέρδους πληροφορίας;

γ) Κατασκευάστε και κάντε plot το πλήρες δένδρο απόφασης για το παραπάνω πρόβλημα με βάση το δείκτη Gini.

2.1 Εισαγωγή Δεδομένων και Βιβλιοθηκών

Αρχικά διαβάζουμε τα δεδομένα και φορτώνουμε τις απαραίτητες βιβλιοθήκες:

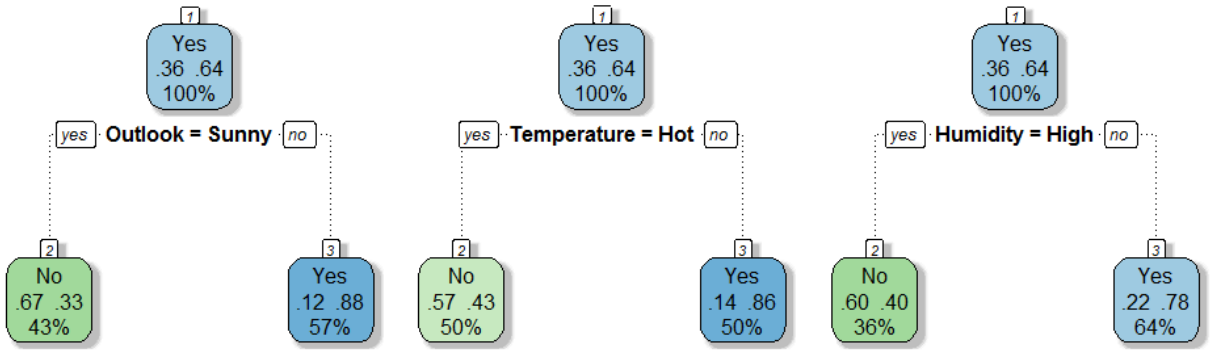
```
> weather = read.csv("weather.txt")
> library(rpart)
> library(rpart.plot)
```

2.2 Κριτήρια Διαχωρισμού

Για να δούμε το διαχωρισμό για τη μεταβλητή Outlook τρέχουμε την παρακάτω εντολή:

```
> model <- rpart(Play ~ Outlook, method = "class", data = weather, minsplit = 1)
```

Αντίστοιχα, αν τρέξουμε και για τις Temperature και Humidity, και κάνουμε plot την καθεμία (με την `rpart.plot(model, extra = 104, nn = TRUE)`), προκύπτουν τα σχήματα:



Εμπειρικά, μπορούμε ίσως ήδη να καταλάβουμε ποιος από τους τρεις διαχωρισμούς πρέπει να επιλεγεί για το πρώτο επίπεδο;

2.2.1 Gini Index

Υπολογίζουμε το Gini index για το Outlook με τους παρακάτω τύπους:

$$GINI(Sunny) = 1 - Freq(Play = No \mid Outlook = Sunny)^2 - Freq(Play = Yes \mid Outlook = Sunny)^2$$
$$= 1 - (4/6)^2 - (2/6)^2 = 0.444$$

$$GINI(Rainy) = 1 - Freq(Play = No \mid Outlook = Rainy)^2 - Freq(Play = Yes \mid Outlook = Rainy)^2$$
$$= 1 - (1/8)^2 - (7/8)^2 = 0.219$$

$$GINI_{Outlook} = Freq(Outlook = Sunny) \cdot GINI(Sunny) + Freq(Outlook = Rainy) \cdot GINI(Rainy)$$
$$= (6/14) \cdot 0.444 + (8/14) \cdot 0.219 = 0.315$$

Αντίστοιχα, για τα Temperature και Humidity, το GINI είναι $GINI_{Temperature} = 0.367$ και $GINI_{Humidity} = 0.394$. Άρα, απαντώντας στο ερώτημα (α), ο βέλτιστος πρώτος διαχωρισμός σύμφωνα με το Gini index είναι στο Outlook.

Μπορούμε επίσης να κάνουμε τους υπολογισμούς χρησιμοποιώντας την R. Για το Outlook, κατασκευάζουμε τους παρακάτω πίνακες συχνοτήτων:

```
> absfreq = table(weather[, c(1, 4)])
> freq = prop.table(absfreq, 1)
> freqSum = rowSums(prop.table(absfreq))
```

Υπολογίζουμε το Gini index για το Sunny και το Rainy:

```
> GINI_Sunny = 1 - freq["Sunny", "No"]^2 - freq["Sunny", "Yes"]^2
> GINI_Rainy = 1 - freq["Rainy", "No"]^2 - freq["Rainy", "Yes"]^2
```

Το συνολικό Gini για το Outlook υπολογίζεται με την εντολή:

```
> GINI_Outlook = freqSum["Sunny"] * GINI_Sunny
               + freqSum["Rainy"] * GINI_Rainy
```

2.2.2 Information Gain

Υπολογίζουμε το Information Gain για το Outlook με τους παρακάτω τύπους:

$$\begin{aligned} Entropy(All) &= -Freq(No)\cdot \lg(Freq(No)) - Freq(Yes)\cdot \lg(Freq(Yes)) \\ &= -(5/14)\cdot \lg(5/14) - (9/14)\cdot \lg(9/14) = 0.940 \end{aligned}$$

$$\begin{aligned} Entropy(Sunny) &= -Freq(No\,|\,Sunny)\cdot \lg(Freq(No\,|\,Sunny)) - Freq(Yes\,|\,Sunny)\cdot \lg(Freq(Yes\,|\,Sunny)) \\ &= -(4/6)\cdot \lg(4/6) - (2/6)\cdot \lg(2/6) = 0.918 \end{aligned}$$

$$\begin{aligned} Entropy(Rainy) &= -Freq(No\,|\,Rainy)\cdot \lg(Freq(No\,|\,Rainy)) - Freq(Yes\,|\,Rainy)\cdot \lg(Freq(Yes\,|\,Rainy)) \\ &= -(1/8)\cdot \lg(1/8) - (7/8)\cdot \lg(7/8) = 0.544 \end{aligned}$$

$$\begin{aligned} GAIN_{Outlook} &= Entropy(All) - Freq(Sunny)\cdot Entropy(Sunny) - Freq(Rainy)\cdot Entropy(Rainy) \\ &= 0.940 - (6/14)\cdot 0.918 + (8/14)\cdot 0.544 = 0.236 \end{aligned}$$

Αντίστοιχα, για τα Temperature και Humidity, το GAIN είναι $GAIN_{Temperature} = 0.152$ και $GAIN_{Humidity} = 0.102$. Άρα, απαντώντας στο ερώτημα (β), ο βέλτιστος πρώτος διαχωρισμός σύμφωνα με το κέρδος πληροφορίας είναι στο Outlook.

Μπορούμε επίσης να κάνουμε τους υπολογισμούς χρησιμοποιώντας την R. Αρχικά υπολογίζουμε την εντροπία για το σύνολο των δεδομένων:

```
> freq = prop.table(table(weather[, c(4)]))

> Entropy_All = - freq["No"] * log2(freq["No"]) - freq["Yes"] * log2(freq["Yes"])
```

Κατόπιν, για το Outlook κατασκευάζουμε τους παρακάτω πίνακες συχνοτήτων:

```
> absfreq = table(weather[, c(1, 4)])

> freq = prop.table(absfreq, 1)

> freqSum = rowSums(prop.table(absfreq))
```

Υπολογίζουμε την εντροπία για το Sunny και το Rainy:

```
> Entropy_Sunny = - freq["Sunny", "No"] * log2(freq["Sunny", "No"])
                  - freq["Sunny", "Yes"] * log2(freq["Sunny", "Yes"])

> Entropy_Rainy = - freq["Rainy", "No"] * log2(freq["Rainy", "No"])
                  - freq["Rainy", "Yes"] * log2(freq["Rainy", "Yes"])
```

Το συνολικό κέρδος πληροφορίας για το Outlook υπολογίζεται με την εντολή:

```
> GAIN_Outlook = Entropy_All - freqSum["Sunny"] * Entropy_Sunny
                             - freqSum["Rainy"] * Entropy_Rainy
```

2.3 Κατασκευή Δένδρου Απόφασης

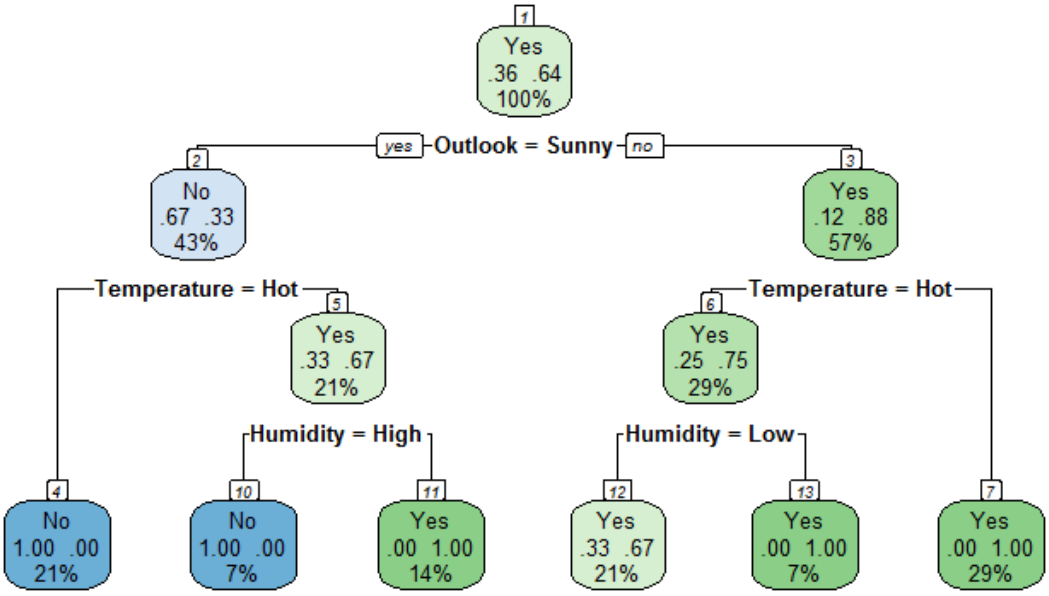
Για να κατασκευάσουμε ένα πλήρες δένδρο απόφασης (ερώτημα (γ)) τρέχουμε την rpart:

```
> model <- rpart(Play ~ Outlook + Temperature + Humidity, method =
"class", data = weather, minsplit = 1, minbucket = 1, cp = -1)
```

Επίσης, για να κάνουμε plot το δένδρο μπορούμε να τρέξουμε την παρακάτω εντολή:

```
> rpart.plot(model, extra = 104, nn = TRUE)
```

Τελικά προκύπτει το παρακάτω δένδρο:



3 Εφαρμογή με Pruning και Μετρικές Αξιολόγησης

Ένας ταξινομητής δένδρο απόφασης μπορεί να εφαρμοστεί επίσης σε datasets με συνεχείς μεταβλητές, όπως το iris dataset που θα χρησιμοποιήσουμε ως εφαρμογή. Αρχικά, εισάγουμε το dataset, επιλέγοντας ωστόσο μόνο τις 2 πρώτες στήλες και αλλάζοντας τις τελευταίες 50 τιμές του Species ώστε να κάνουμε το πρόβλημα binary classification:

```
> iris2 = iris[, c(1, 2, 5)]
> iris2$Species[c(101:150)] = iris2$Species[c(21:70)]
> iris2$Species = factor(iris2$Species)
```

Κατόπιν, κάνουμε split το dataset σε training και testing data:

```
> trainingdata = iris2[c(1:40, 51:90, 101:140),]
> testdata = iris2[c(41:50, 91:100, 141:150),]
```

Στη συνέχεια, θα απαντήσουμε στα παρακάτω ερωτήματα:

α) Κατασκευάστε το δένδρο απόφασης χρησιμοποιώντας τα δεδομένα εκπαίδευσης (με την παράμετρο minsplit της rpart ίση με 20 που είναι το default).

β) Εφαρμόστε το μοντέλο στα testdata και υπολογίστε precision, recall και f-measure για τις δύο κλάσεις.

γ) Κατασκευάστε τα δένδρα για minsplit ίσο με 10 και για minsplit ίσο με 30 και υπολογίστε precision, recall και f-measure για τις δύο κλάσεις αφού τα εφαρμόσετε στα testdata.

δ) Συγκρίνετε τα τρία μοντέλα ως προς το f-measure για την κλάση versicolor.

3.1 Κατασκευή και Εφαρμογή Δένδρου Απόφασης

Κάνουμε training το δένδρο και το σχεδιάζουμε με τις παρακάτω εντολές (ερώτημα (α), (γ)):

```
> model <- rpart(Species ~ ., method = "class", data = trainingdata, minsplit = 20)

> rpart.plot(model, extra = 104, nn = TRUE)
```

Μπορούμε στη συνέχεια να εκτελέσουμε το δένδρο στο test set:

```
> xtest = testdata[,1:2]

> ytest = testdata[,3]

> pred = predict(model, xtest, type="class")
```

3.2 Υπολογισμός Μετρικών Αξιολόγησης

Μπορούμε να δούμε το confusion matrix και να υπολογίσουμε χρήσιμες μετρικές με τις παρακάτω εντολές (ερώτημα (β), (γ)):

```
> cm = as.matrix(table(Actual = ytest, Predicted = pred))

> accuracy = sum(diag(cm)) / sum(cm)

> precision = diag(cm) / colSums(cm)

> recall = diag(cm) / rowSums(cm)

> f1 = 2 * precision * recall / (precision + recall)

> data.frame(precision, recall, f1)
```

(Εναλλακτικά μπορούμε να υπολογίσουμε τα TP, FP, TN, FN και να υπολογίσουμε το precision ως TP/(TP + FP) και το recall ως TP/(TP + FN))

Τέλος, το f-measure για τα 3 μοντέλα φαίνεται στον πίνακα (ερώτημα (δ)):

Decision Tree	F-Measure
minsplit = 10	0.833
minsplit = 20	0.947
minsplit = 30	0.857

4 Πρόβλημα για Εξάσκηση

Θεωρείστε τα δεδομένα εκπαίδευσης του παρακάτω πίνακα για ένα πρόβλημα δυαδικής ταξινόμησης.

CustomerID	Sex	CarType	Budget	Insurance
1	M	Family	Low	No
2	M	Sport	Medium	No
3	M	Sport	Medium	No
4	M	Sport	High	No
5	M	Sport	VeryHigh	No
6	M	Sport	VeryHigh	No
7	F	Sport	Low	No
8	F	Sport	Low	No
9	F	Sport	Medium	No
10	F	Sedan	High	No
11	M	Family	High	Yes
12	M	Family	VeryHigh	Yes
13	M	Family	Medium	Yes
14	M	Sedan	VeryHigh	Yes
15	F	Sedan	Low	Yes
16	F	Sedan	Low	Yes
17	F	Sedan	Medium	Yes
18	F	Sedan	Medium	Yes
19	F	Sedan	Medium	Yes
20	F	Sedan	High	Yes

Να υπολογίσετε τον δείκτη Gini και για:

α) το σύνολο των δειγμάτων εκπαίδευσης

β) το χαρακτηριστικό CustomerID

γ) το χαρακτηριστικό Sex

δ) το χαρακτηριστικό CarType χρησιμοποιώντας multiway split

ε) το χαρακτηριστικό Budget χρησιμοποιώντας multiway split

Ποιο χαρακτηριστικό πρέπει να χρησιμοποιηθεί στον πρώτο διαχωρισμό σύμφωνα με το GINI index;

Υπόδειξη: Εξηγείστε γιατί το CustomerID δεν θα μπορούσε να χρησιμοποιηθεί σαν χαρακτηριστικό διαχωρισμού αν κι έχει τον χαμηλότερο δείκτη Gini.