Naïve Bayes και Classification Evaluation στην R

1 Εισαγωγή

1.1 Εισαγωγή στο Naïve Bayes

Ο Naïve Bayes είναι ένας αλγόριθμος μηχανικής μάθησης που βασίζεται στο θεώρημα του Bayes καθώς και στην υπόθεση της ανεξαρτησίας μεταξύ των χαρακτηριστικών. Δεδομένου ενός συνόλου δεδομένων με χαρακτηριστικά $x_1, x_2, ..., x_n$ και χαρακτηριστικό-κλάση c, το θεώρημα του Bayes επιτρέπει τον υπολογισμό της πιθανότητας ένα δείγμα $x = \{x_1, x_2, ..., x_n\}$ να ανήκει στο c ως το συνδυασμό των πιθανοτήτων $P(x_1 | c), P(x_2 | c), ..., P(x_n | c)$:

$$P(c | x) = \frac{P(x_1 | c) \cdot P(x_2 | c) \cdot ... \cdot P(x_n | c) \cdot P(c)}{P(x_1) \cdot P(x_2) \cdot ... \cdot P(x_n)}$$

όπου παρατηρούμε ότι η πιθανότητα του συνόλου των χαρακτηριστικών δίνεται από το γινόμενό τους, καθώς οι τιμές των χαρακτηριστικών είναι ανεξάρτητες μεταξύ τους.

1.2 Naïve Bayes στην R

Για την κατασκευή του πιθανοτικού μοντέλου Naïve Bayes στην R, μπορούμε να χρησιμοποιήσουμε τη βιβλιοθήκη e1071:

```
> library(e1071)
```

Για να κατασκευάσουμε ένα μοντέλο τρέχουμε την εντολή naiveBayes:

```
> model <- naiveBayes(Target ~ ., data = ..., laplace = ...)
```

όπου με την παράμετρο laplace επιλέγουμε τη χρήση laplace smoothing.

Έχοντας ένα μοντέλο, για να προβλέψουμε μια νέα τιμή τρέχουμε την εντολή

```
> predict(model, trvalue)
```

όπου προσθέτοντας την παράμετρο type = "raw", δίνονται επιπλέον οι εκ των υστέρων πιθανότητες

1.2 Classification Evaluation στην R

Για να αξιολογήσουμε ένα μοντέλο ταξινόμησης χρησιμοποιούμε τις παρακάτω βιβλιοθήκες:

```
> library(MLmetrics)
> library(ROCR)
```

Έχοντας το αποτέλεσμα ενός αλγορίθμου (`pred`) και τις παραγματικές τιμές μπορούμε να δούμε το confusion matrix και να υπολογίσουμε χρήσιμες μετρικές με τις παρακάτω εντολές:

```
> ConfusionMatrix(ytest, pred)
> Precision(ytest, pred)
> Recall(ytest, pred)
> F1_Score(ytest, pred)
```

Στις παραπάνω εντολές μπορούμε να επιλέξουμε την κλάση για τις οποίες υπολογίζονται οι μετρικές δίνοντάς τη ως τρίτη παράμετρο:

```
> Precision(ytest, pred, "class1")
```

Έχοντας το αποτέλεσμα ενός αλγορίθμου σε μορφή πιθανοτήτων (`pred_prob` – posterior probabilities), μπορούμε να υπολογίσουμε τα TPR και FPR με τις εντολές:

```
> pred_obj = prediction(pred_prob, ytest, label.ordering = ...)
> ROCcurve <- performance(pred_obj, "tpr", "fpr")
```

Όπου στο label.ordering θέτουμε τις κλάσεις με 2η κλάση αυτή που αφορά το αντικείμενο `pred_prob` (η βιβλιοθήκη υποστηρίζει μόνο binary classification).

Μπορούμε να σχεδιάσουμε την καμπύλη με τις παρακάτω εντολές:

```
> plot(ROCcurve, col = "blue")
> abline(0,1, col = "grey")
```

Και να βρούμε την περιοχή κάτω από την καμπύλη με την εντολή:

```
> performance(pred_obj, "auc")
```

2 Κατασκευή Μοντέλου Naïve Bayes και Κατάταξη Τιμών

Για την κατασκευή ενός μοντέλου Naïve Bayes θα χρησιμοποιήσουμε ως εφαρμογή τα δεδομένα εκπαίδευσης του παρακάτω πίνακα για ένα πρόβλημα δυαδικής ταξινόμησης.

Weather	Day	HighTraffic
Hot	Vacation	No
Cold	Work	Yes
Normal	Work	No
Cold	Weekend	Yes
Normal	Weekend	Yes
Cold	Work	No
Hot	Work	No
Hot	Vacation	Yes

Θα απαντήσουμε στα παρακάτω ερωτήματα:

α) Χρησιμοποιώντας τον Naïve Bayes Classifier, σε ποια κλάση θα κατατάσσατε μία νέα παρατήρηση με τιμές (Weather, Day) = (Hot, Vacation);

β) Χρησιμοποιώντας τον Naïve Bayes Classifier, σε ποια κλάση θα κατατάσσατε μία νέα παρατήρηση με τιμές (Weather, Day) = (Hot, Weekend);

γ) Επαναλάβετε το ερώτημα (β) χρησιμοποιώντας Laplace smoothing.

δ) Κατασκευάστε το πλήρες μοντέλο στην R και επαναλάβετε το ερώτημα (α).

ε) Κατασκευάστε το πλήρες μοντέλο στην R χρησιμοποιώντας Laplace Smoothing και επαναλάβετε το ερώτημα (β).

2.1 Εισαγωγή Δεδομένων και Βιβλιοθηκών

Αρχικά διαβάζουμε τα δεδομένα και φορτώνουμε τις απαραίτητες βιβλιοθήκες:

```
> traffic = read.csv("traffic.txt")
> library(e1071)
```

2.2 Υπολογισμός τιμής πιθανότητας

Για το ερώτημα (α) υπολογίζουμε τις πιθανότητες με τους παρακάτω τύπους:

$$P(Yes \mid Hot, Vacation) = \frac{P(Hot \mid Yes) \cdot P(Vacation \mid Yes) \cdot P(Yes)}{P(Hot) \cdot P(Vacation)} = \frac{1/4 \cdot 1/4 \cdot 1/2}{3/8 \cdot 2/8} = 1/3$$

$$P(No|Hot,Vacation)=\frac{P(Hot|No)\cdot P(Vacation|No)\cdot P(No)}{P(Hot)\cdot P(Vacation)}=\frac{2/4\cdot 1/4\cdot 1/2}{3/8\cdot 2/8}=2/3$$

Αφού *P(No|Hot,Vacation) > P(Yes|Hot,Vacation)*, ο αλγόριθμος θα κατατάξει την τιμή ως No.

Αντίστοιχα, για το ερώτημα (β):

$$P(Yes|Hot,Weekend)=\frac{P(Hot|Yes)\cdot P(Weekend|Yes)\cdot P(Yes)}{P(Hot)\cdot P(Weekend)}=\frac{1/4\cdot 2/4\cdot 1/2}{3/8\cdot 2/8}=2/3$$

$$P(No|Hot,Weekend)=\frac{P(Hot|No)\cdot P(Weekend|No)\cdot P(No)}{P(Hot)\cdot P(Weekend)}=\frac{2/4\cdot 0/4\cdot 1/2}{3/8\cdot 2/8}=0$$

Αφού *P(No|Hot,Weekend) < P(Yes|Hot,Weekend)*, ο αλγόριθμος θα κατατάξει την τιμή ως Yes.

Για το ερώτημα (γ) θα υπολογίσουμε εκ νέου τις πιθανότητες προσθέτοντας όμως τη μονάδα ως Laplaciανό συντελεστή, οπότε έχουμε τις πιθανότητες $P(Hot|Yes)=(1+1)/(4+3)=2/7$, $P(Weekend|Yes)=(2+1)/(4+3)=3/7$, $P(Hot|No)=(2+1)/(4+3)=3/7$ και την πιθανότητα $P(Weekend|No)=(0+1)/(4+3)=1/7$, άρα τελικά έχουμε τα παρακάτω:

$$P(Yes|Hot,Weekend)=\frac{P(Hot|Yes)\cdot P(Weekend|Yes)\cdot P(Yes)}{P(Hot)\cdot P(Weekend)}=\frac{2/7\cdot 3/7\cdot 1/2}{3/8\cdot 2/8}=32/49$$

$$P(No|Hot,Weekend)=\frac{P(Hot|No)\cdot P(Weekend|No)\cdot P(No)}{P(Hot)\cdot P(Weekend)}=\frac{3/7\cdot 1/7\cdot 1/2}{3/8\cdot 2/8}=16/49$$

Άρα, αφού *P(No|Hot,Weekend) < P(Yes|Hot,Weekend)*, ο αλγόριθμος θα κατατάξει την τιμή ως Yes.

2.3 Κατασκευή Μοντέλου με την R

2.3.1 Κατασκευή Απλού Μοντέλου

Για το ερώτημα (δ) εφαρμόζουμε το Naïve Bayes για το dataset:

```
> model <- naiveBayes(HighTraffic ~ ., data = traffic)
```

Εμφανίζουμε αν θέλουμε το μοντέλο στην οθόνη:

```
> print(model)
```

```
Naive Bayes Classifier for Discrete Predictors
Call: naiveBayes.default(x = X, y = Y, laplace = laplace)

A-priori probabilities:
Y
 No Yes
0.5 0.5

Conditional probabilities:
     Weather
Y     Cold  Hot Normal
  No  0.25 0.50   0.25
  Yes 0.50 0.25   0.25

     Day
Y     Vacation Weekend Work
  No      0.25    0.00 0.75
  Yes     0.25    0.50 0.25
```

Υπολογίζουμε εκ νέου το ερώτημα (α):

```
> trvalue <- data.frame(Weather = factor("Hot", levels(traffic$Weather)),
Day = factor("Vacation", levels(traffic$Day)))

> predict(model, trvalue)
```

Μπορούμε επιπλέον να πάρουμε τις πιθανότητες:

```
> predict(model, trvalue, type = "raw")
```

2.3.1 Κατασκευή Μοντέλου με Laplace Smoothing

Για το ερώτημα (ε) εφαρμόζουμε το Naïve Bayes με Laplace Smoothing για το dataset:

```
> model <- naiveBayes(HighTraffic ~ ., data = traffic, laplace = 1)
```

Υπολογίζουμε εκ νέου το ερώτημα (β):

```
> trvalue <- data.frame(Weather = factor("Hot", levels(traffic$Weather)),
Day = factor("Weekend", levels(traffic$Day)))

> predict(model, trvalue)
```

Μπορούμε επιπλέον να πάρουμε τις πιθανότητες:

```
> predict(model, trvalue, type = "raw")
```

3 Εφαρμογή με Μετρικές Αξιολόγησης και Καμπύλη ROC

Θα εφαρμόσουμε τον ταξινομητή Naïve Bayes στο dataset HouseVotes84. Αρχικά, εισάγουμε το dataset, αγνοώντας τα δείγματα για τα οποία δεν υπάρχουν οι σχετικές τιμές (missing):

```
> data(HouseVotes84, package = "mlbench")
> votes = na.omit(HouseVotes84)
```

Επίσης, εισάγουμε τις βιβλιοθήκες που θα μας χρειαστούν για να εκτελέσουμε και να αξιολογήσουμε το μοντέλο:

```
> library(e1071)
> library(MLmetrics)
> library(ROCR)
```

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

```
> trainingdata = votes[1:180,]
> testingdata = votes[181:232,]
```

Στη συνέχεια, θα απαντήσουμε στα παρακάτω ερωτήματα:

α) Κατασκευάστε ένα μοντέλο Naïve Bayes χρησιμοποιώντας τα δεδομένα εκπαίδευσης και εφαρμόστε το μοντέλο στα testdata.

β) Για τα testdata υπολογίστε precision, recall και f-measure για την κλάση democrat.

γ) Κατασκευάστε την καμπύλη ROC για το μοντέλο.

3.1 Κατασκευή και Εφαρμογή Μοντέλου Naïve Bayes

Κάνουμε training το μοντέλο με την παρακάτω εντολή (ερώτημα (α)):

```
> model <- naiveBayes(Class ~ ., data = trainingdata)
```

Μπορούμε στη συνέχεια να εφαρμόσουμε το μοντέλο στο test set:

```
> xtest = testingdata[,-1]
> ytest = testingdata[,1]
> pred = predict(model, xtest)
> predprob = predict(model, xtest, type = "raw")
```

3.2 Υπολογισμός Μετρικών Αξιολόγησης και Σχεδίαση ROC Curve

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

```
> ConfusionMatrix(pred, ytest)
```

```
> Precision(ytest, pred, "democrat")
```

```
> Recall(ytest, pred, "democrat")
```

Για την καμπύλη ROC (ερώτημα (γ)) θα χρειαστεί αρχικά να υπολογίσουμε τα TPR και FRP με τις παρακάτω εντολές:

```
> pred_obj = prediction(predprob[,1], ytest, label.ordering = c("republican", "democrat"))
```

```
> ROCcurve <- performance(pred_obj, "tpr", "fpr")
```

Αν εκτυπώσουμε το αντικείμενο ROCcurve, θα δούμε τα TPR και FRP και τα αντίστοιχα thresholds.

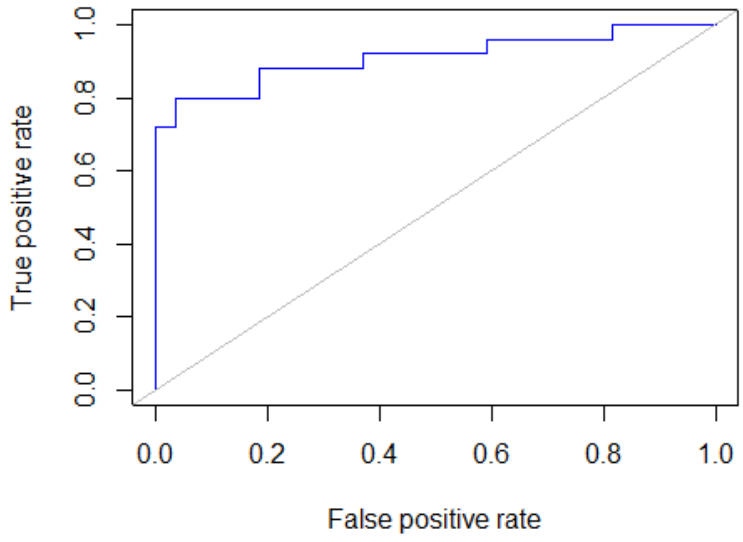
Τέλος, μπορούμε να σχεδιάσουμε την καμπύλη με τις παρακάτω εντολές:

```
> plot(ROCcurve, col = "blue")
```

```
> abline(0,1, col = "grey")
```

Και να βρούμε την περιοχή κάτω από την καμπύλη με την εντολή:

```
> performance(pred_obj, "auc")
```



4 Πρόβλημα για Εξάσκηση

Ζητείται να αξιολογηθεί η απόδοση δύο μοντέλων κατάταξης M1 και M2. Το test set που έχει επιλεγεί περιέχει 10 δείγματα με 26 δυαδικά χαρακτηριστικά τα οποία επισημαίνονται ως $x_1, x_2, \ldots, x_{26}$. Στον παρακάτω πίνακα φαίνονται οι εκ των υστέρων πιθανότητες που προκύπτουν εφαρμόζοντας τα δύο μοντέλα στο συγκεκριμένο test set. (Εμφανίζονται μόνο οι εκ των υστέρων πιθανότητες της θετικής κλάσης). Υποθέστε ότι ενδιαφερόμαστε μόνο να εντοπίσουμε δείγματα που ανήκουν στην θετική (1) κλάση.

Class	P_M1	P_M2
1	0.73	0.61
1	0.69	0.03
0	0.44	0.68
0	0.55	0.31
1	0.67	0.45
1	0.47	0.09
0	0.08	0.38
0	0.15	0.05
1	0.45	0.01
0	0.35	0.04

Μπορείτε να κατασκευάσετε τον πίνακα στην R με τις παρακάτω εντολές:

```
> Class = c(1, 1, 0, 0, 1, 1, 0, 0, 1, 0)
> P_M1 = c(0.73, 0.69, 0.44, 0.55, 0.67, 0.47, 0.08, 0.15, 0.45, 0.35)
> P_M2 = c(0.61, 0.03, 0.68, 0.31, 0.45, 0.09, 0.38, 0.05, 0.01, 0.04)
> data = data.frame(Class, P_M1, P_M2)
```

Απαντήστε στα παρακάτω ερωτήματα:

α) Υπολογίστε τα TPR και FPR για τα 2 μοντέλα

β) Να σχεδιάσετε στο ίδιο γράφημα την καμπύλη ROC και για τα δύο μοντέλα M1, M2. Ποιό μοντέλο είναι καλύτερο με βάση τη μετρική AUC;

β) Επιλέξτε ένα κατώφλι να είναι ίσο με t = 0.5. Αυτό σημαίνει ότι οποιοδήποτε δείγμα του τεστ σετ έχει εκ των υστέρων πιθανότητα μεγαλύτερη του t θα καταταχθεί στην θετική κλάση (1). Να υπολογίσετε τις μετρικές precision, recall και F-measure για τα δύο μοντέλα με το συγκεκριμένο κατώφλι. Συγκρίνετε τα F-measures των δύο μοντέλων. Ποιο μοντέλο είναι καλύτερο; Τα αποτελέσματα είναι αναμενόμενα από τις καμπύλες ROC;