ΠΑΝΕΠΙΣΤΗΜΙΟ ΜΑΚΕΔΟΝΙΑΣ

ΤΜΗΜΑ ΕΦΑΡΜΟΣΜΕΝΗΣ ΠΛΗΡΟΦΟΡΙΚΗΣ

Όνομα: Δημήτρης

Επίθετο: Μανωλάκης

Α.Μ.: it1423

ΜΗΧΑΝΙΚΗ ΜΑΘΗΣΗ

1η Εργασία

ΤΑΞΙΝΟΜΗΣΗ

Προεπεξεργασία Δεδομένων:

Αρχικά, η response variable Effort μετατράπηκε σε δυαδική μεταβλητή κάνοντας χρήση της διαμέσου (median) ως όριο ανάμεσα στις δυο κλάσεις 'High' και 'Low' που δημιουργήθηκαν. Επιπλέον το dataset χωρίστηκε σε δυο επιμέρους κομμάτια, τα training και testing datasets με αναλογία 70/30.

Μέθοδος 1: Λογιστική Παλινδρόμηση

Συνολικά, δημιουργήθηκαν τρία μοντέλα προκειμένου να εντοπιστεί αυτό με την υψηλότερη απόδοση. Το πρώτο μοντέλο περιέχει όλες τις μεταβλητές του dataset και βλέπουμε στο summary πως η μονή στατιστικά σημαντική μεταβλητή είναι η AdjustedFunctionPoints.

Εικόνα 1: Summary της Λογιστικής Παλινδρόμησης

```
Call:
glm(formula = as.factor(effort.bin) ~ ., family = binomial, data = Data,
    subset = train_ind)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.7281  -1.0462  -0.0004   0.8892   3.3804

Coefficients: (1 not defined because of singularities)
                        Estimate Std. Error z value Pr(>|z|)
(Intercept)             1.233958   0.204124   6.045 1.49e-09 ***
AdjustedFunctionPoints -0.010462   0.004677  -2.237   0.0253 *
Inputcount              0.004418   0.006805   0.649   0.5162
Outputcount             0.006189   0.006986   0.886   0.3757
Enquirycount            0.002354   0.007491   0.314   0.7534
Filecount              -0.001064   0.007492  -0.142   0.8871
Interfacecount         -0.004728   0.008201  -0.577   0.5643
Addedcount              0.003650   0.005958   0.613   0.5401
Changedcount            0.002187   0.006215   0.352   0.7249
Deletedcount                  NA         NA      NA       NA
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 490.34  on 353  degrees of freedom
Residual deviance: 386.23  on 345  degrees of freedom
AIC: 404.23

Number of Fisher Scoring iterations: 7
```

Στην παρακάτω εικόνα βλέπουμε το Confusion Matrix, που περιέχει τα αποτελέσματα της απόδοσης του μοντέλου κάνοντας χρήση των δεδομένων ελέγχου:

Εικόνα 2: Confusion Matrix

Είναι εμφανές πως η απόδοση του μοντέλου δεν είναι καλή καθώς ταξινομεί λανθασμένα 114 από

τις 152 παρατηρήσεις των δεδομένων ελέγχου. Το test error rate είναι ίσο με 0.75, δηλαδή το μοντέλο ταξινόμησε λανθασμένα το 75% των δεδομένων.

Σε μια δεύτερη προσπάθεια, δημιουργούμε μοντέλο λογιστικής παλινδρόμησης το οποίο περιέχει μόνο τη στατιστικά σημαντική μεταβλητή AdjustedFunctionPoints και βλέπουμε το test error rate να βελτιώνεται ελαφρά σε 0.737. Τέλος, προστίθεται στο μοντέλο αυτό η επόμενη πιο στατιστικά σημαντική μεταβλητή, η Outputcount, και καταλήγουμε σε μια επιπλέον βελτίωση του test error rate σε 0.71.

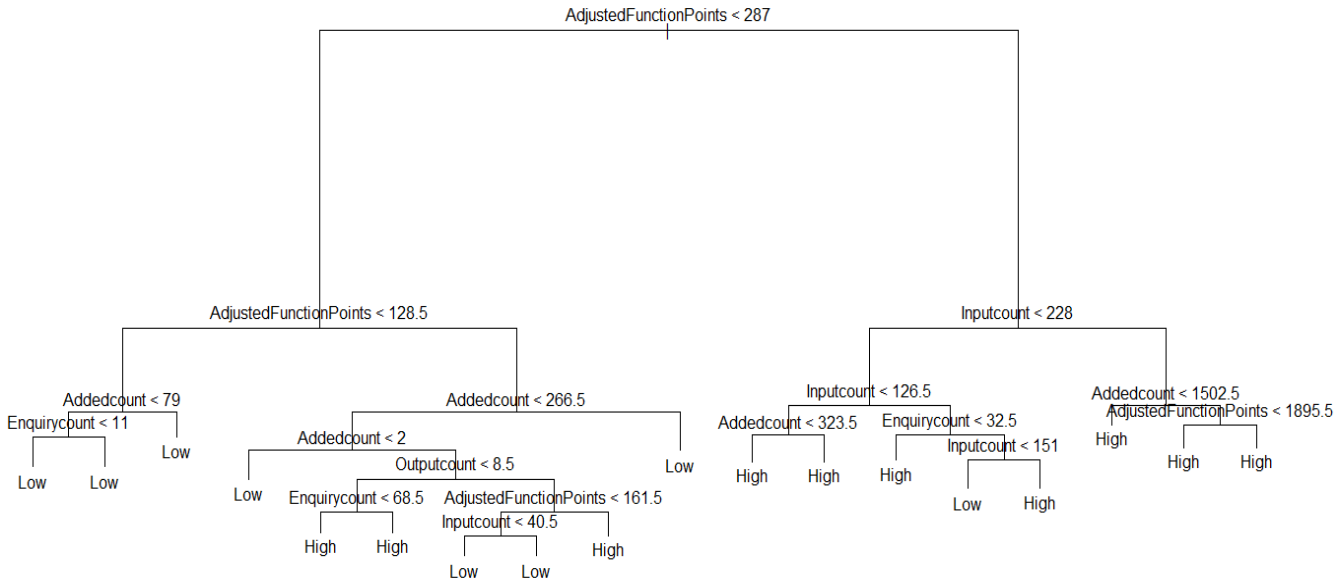
Μέσω αναδειγματοληψίας με επανάθεση των δεδομένων ελέγχου, υπολογίζουμε το διάστημα εμπιστοσύνης του test error rate του τρίτου και βέλτιστου μοντέλου λογιστικής παλινδρόμησης:

(0.6331722, 0.7878805)

Μέθοδος 2: Decision Tree

Αρχικά, δημιουργείται μοντέλο δέντρου απόφασης το οποίο οπτικοποιείται στην παρακάτω εικόνα:

Εικόνα 3: Οπτικοποίηση του Decision Tree



Βλέπουμε ότι στη ριζά του δέντρου χρησιμοποιείται η AdjustedFunctionPoints. Αναμενόμενο καθώς όλα τα προηγούμενα μοντέλα την χαρακτηρίζουν σταθερά ως στατιστικά σημαντική. Κάποιες επιπλέον πληροφορίες δίνονται από το Summary του μοντέλου:

Εικόνα 4: Summary του Decision Tree

```
Classification tree:
tree(formula = as.factor(effort.bin) ~ ., data = Data, subset = train_ind)
Variables actually used in tree construction:
[1] "AdjustedFunctionPoints" "Addedcount"             "Enquirycount"
[4] "Outputcount"            "Inputcount"
Number of terminal nodes:  18
Residual mean deviance:  0.8149 = 273.8 / 336
Misclassification error rate: 0.209 = 74 / 354
```

Οπού μπορούμε να δούμε τις μεταβλητές που εν τελεί χρησιμοποιήθηκαν στο μοντέλο, το πλήθος των terminal nodes καθώς και το training error rate.

Αναφορικά με την απόδοση του μοντέλου στα δεδομένα ελέγχου εξετάζεται το Confusion Matrix:

Εικόνα 5: Confusion Matrix

tree.pred	High	Low
High	51	23
Low	19	59

Η βελτίωση σε σχέση με την λογιστική παλίνδρομη είναι μεγάλη καθώς 110 από τις 152 παρατηρήσεις των δεδομένων ελέγχου ταξινομούνται σωστά, δίνοντας test error rate ίσο με 0.276.

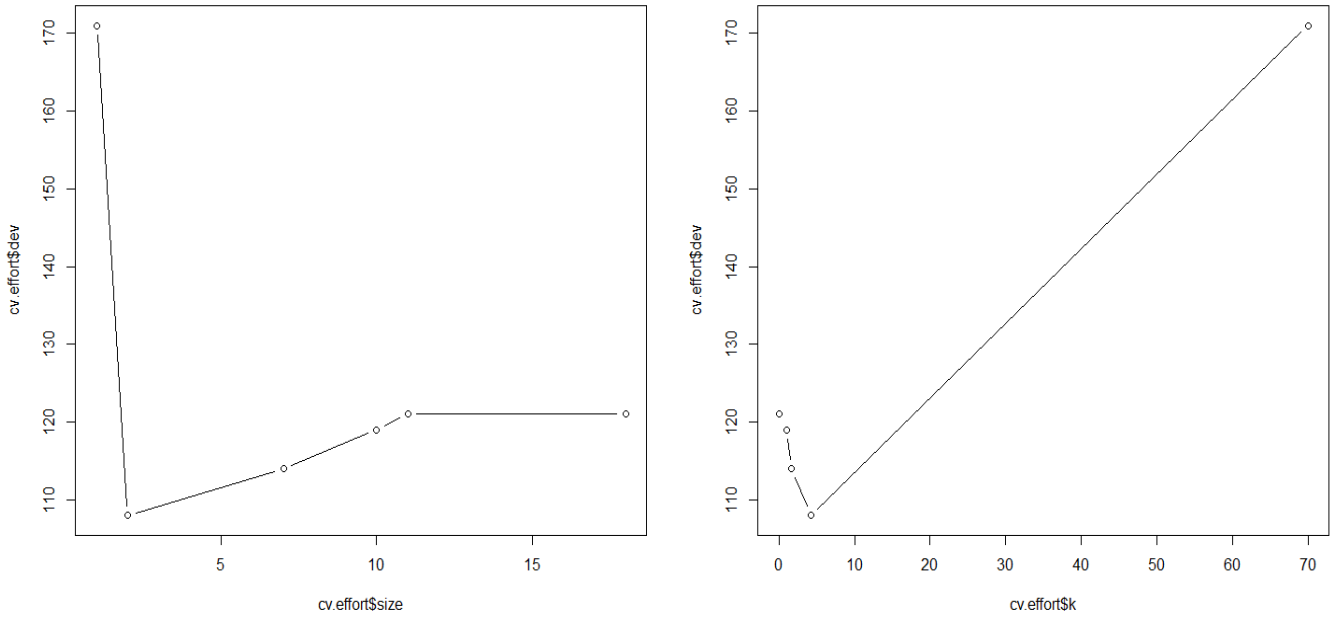
Σε επόμενο βήμα, εξετάζεται αν η πραγματοποίηση κλαδέματος (pruning) στο Decision Tree θα βελτιώσει την απόδοση του. Κάνοντας χρήση cross-validation επιλέγεται η βέλτιστη τιμή για την cost-complexity παράμετρο k, η οποία καθορίζει το κλάδεμα του δέντρου. Η παρακάτω εικόνα παρουσιάζει τα αποτελέσματα του cross-validation:

Εικόνα 6: Optimal Decision Tree pruning.

```
$size
[1] 18 11 10  7  2  1

$dev
[1] 121 121 119 114 108 171

$k
[1]      -Inf  0.000000  1.000000  1.666667  4.200000 70.000000

$method
[1] "misclass"

attr(,"class")
[1] "prune"         "tree.sequence"
```

Βλέπουμε ότι το βέλτιστο πλήθος των terminal nodes που επιλέγεται είναι ίσο μόνο με 2, καθώς διαθέτει τα λιγότερα cross-validation errors (108).

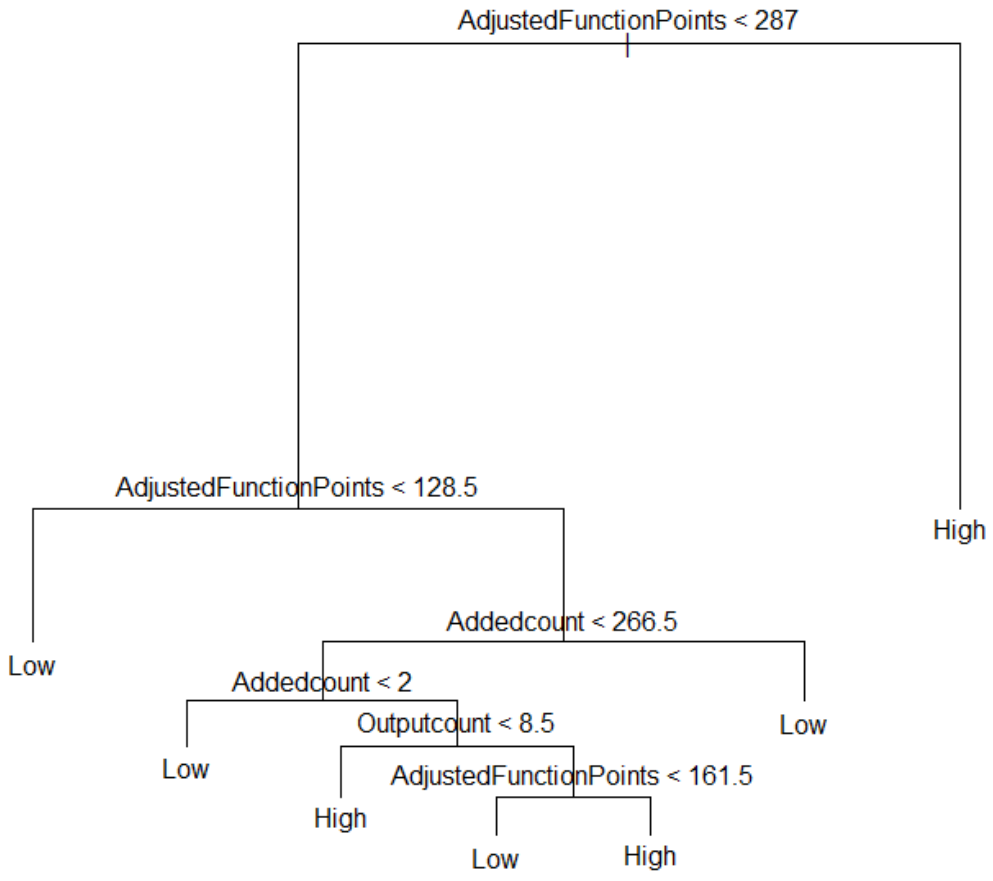
Εικόνα 7: Optimal Decision Tree pruning.



Στην παραπάνω εικόνα βλέπουμε τα plot των cross-validation errors σε σχέση με τις τιμές του πλήθους των terminal nodes και το μέγεθος του k.

Λόγω του κλαδέματος καταλήγουμε σε ένα πολύ πιο ευνόητο και ερμηνεύσιμο δέντρο όπως φαίνεται και από την οπτικοποίηση του:

Εικόνα 8: Οπτικοποίηση του pruned Decision Tree.



Το τίμημα της απλούστερης αυτής κατασκευής είναι μόνο μια ελαφρώς χειρότερη απόδοση στα

δεδομένα ελέγχου ίση με 0.283 έναντι του 0.276 που είχαμε πριν το κλάδεμα.

Κάνοντας χρήση του bootstrap υπολογίζεται για το μη κλαδεμένο Decision Tree το διάστημα εμπιστοσύνης για το test error rate:

(0.1924017, 0.3602299)

Μέθοδος 3: Support Vector Machine

Αρχικά, δημιουργούνται δυο μοντέλα οπού το πρώτο κάνει χρήση του **Polynomial** kernel και το δεύτερο κάνει χρήση του **Radial** kernel. Και στα δυο μοντέλα χρησιμοποιείται η τιμή 1 για το κόστος και την παράμετρο γ. Το δεύτερο μοντέλο έχει την καλύτερη απόδοση στα δεδομένα ελέγχου με test error rate ίσο με 0.283.

Στην συνέχεια, διατηρώντας το Radial kernel πραγματοποιείται cross-validation και επιλέγονται οι βέλτιστες τιμές για τις παραμέτρους cost και γ.

Εικόνα 9: Optimal Decision Tree pruning.

```
Parameter tuning of 'svm':

- sampling method: 10-fold cross validation

- best parameters:
 cost gamma
   10     2

- best performance: 0.2654762
```

Στην παραπάνω εικόνα βλέπουμε ότι για την παράμετρο cost επιλέχθηκε η τιμή 10 και για την γ η τιμή 2. Αξιολογούμε το μοντέλο που δημιουργείται με τις παραμέτρους αυτές στα δεδομένα ελέγχου και καταλήγουμε σε βελτιωμένο test error rate ίσο με 0.257.

Το διάστημα εμπιστοσύνης για το test error rate του μοντέλου υπολογίστηκε κάνοντας χρήση bootstrap και είναι το εξής:

(0.1773420, 0.3358159)

Κάνοντας μια σύγκριση στα αποτελέσματα είναι εύκολο να καταλήξουμε πως το SVM μοντέλο έχει την καλύτερη απόδοση καθώς διαθέτει το μικρότερο test error rate, το διάστημα εμπιστοσύνης που παράχθηκε μέσω του bootstrap είναι γύρω από την μικρότερη τιμή και είναι

μάλιστα το πιο στενό από όλα, δίνοντας μεγαλύτερη βεβαιότητα επομένως για την απόδοση αυτή. Αν στόχος μας είναι η βέλτιστη ταξινόμηση των δεδομένων είναι προφανώς η καλύτερη επιλογή. Σε περίπτωση που μας ενδιαφέρει όμως η οπτικοποίηση και η ερμηνεία του μοντέλου τα δέντρα απόφασης αποτελούν την καλύτερη επιλογή, με ένα μικρό τίμημα στην απόδοση του μοντέλου.