

**ΕΘΝΙΚΟ ΚΑΙ ΚΑΠΟΔΙΣΤΡΙΑΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ ΤΜΗΜΑ ΟΙΚΟΝΟΜΙΚΩΝ ΕΠΙΣΤΗΜΩΝ**

ΠΡΟΓΡΑΜΜΑ ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ «ΕΦΑΡΜΟΣΜΕΝΗΣ ΟΙΚΟΝΟΜΙΚΗΣ ΚΑΙ ΧΡΗΜΑΤΟΟΙΚΟΝΟΜΙΚΗΣ»

ΚΑΤΕΥΘΥΝΣΗ

«ΔΙΟΙΚΗΣΗ, ΑΝΑΛΥΤΙΚΗ ΚΑΙ ΠΛΗΡΟΦΟΡΙΑΚΑ ΣΥΣΤΗΜΑΤΑ ΕΠΙΧΕΙΡΗΣΕΩΝ»

Master of Science in

Business Administration, Analytics and Information Systems

**Data Science (R)**

**Assignment 2**

**Κωνσταντίνος Κουτσομπίνας**

Αθήνα 21/06/2025

Περιεχόμενα

[Άσκηση 3 3](#_Toc201929034)

[a) Training και test set 3](#_Toc201929035)

[b) Εκπαίδευση και βασικά χαρακτηριστικά 3](#_Toc201929036)

[c) Περιγραφή τερματικού κόμβου 3](#_Toc201929037)

[d) Γραφική αναπαράσταση δέντρου 4](#_Toc201929038)

[e) Confusion Matrix 5](#_Toc201929039)

[f) Cross Validation 5](#_Toc201929040)

[g) Γραφική αναπαράσταση αποτελεσμάτων cross validation 5](#_Toc201929041)

[h) Βέλτιστο δέντρο 6](#_Toc201929042)

[i) Pruning 6](#_Toc201929043)

[j) Διαφορά στο train 7](#_Toc201929044)

[k) Διαφορά στο test 7](#_Toc201929045)

[Άσκηση 4 8](#_Toc201929046)

[a) Training και test set 8](#_Toc201929047)

[b) Εκπαίδευση, απεικόνιση και διερμήνευση αποτελεσμάτων 8](#_Toc201929048)

[c) Cross validation 9](#_Toc201929049)

[d) Bagging 9](#_Toc201929050)

[e) Random forests 11](#_Toc201929051)

# Άσκηση 3

## Training και test set

Χωρίζουμε τα δεδομένα σε train (800 samples) και test (270 samples)

1. set.seed(42)

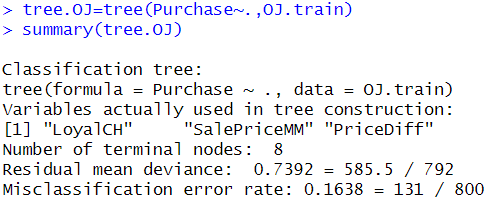
2. train=sample(1:nrow(OJ), 800)

3. OJ.train=OJ[train,]

4. OJ.test=OJ[-train,]

## Εκπαίδευση και βασικά χαρακτηριστικά

Εκπαιδεύουμε ένα δέντρο στο training set:



Βλέπουμε ότι το δέντρο έχει **8 τερματικούς κόμβους και 16.38%(131/800) misclassification error rate**

## Περιγραφή τερματικού κόμβου

Παρατηρούμε τους κόμβους του δέντρου:

A screenshot of a computer code

AI-generated content may be incorrect.

* Ένας τερματικός κόμβος είναι ο κόμβος 4.
* Στον κόμβο αυτό υπάρχουν 64 παρατηρήσεις, εκ των οποίων το 100% είναι της κλάσης ΜΜ (minute maid orange juice) συνεπώς το predicted class σε αυτόν τον κόμβο είναι MM.
* Στον κόμβο αυτό καταλήγουν οι παρατηρήσεις που έχουν LoyalCH < 0.48285 and LoyalCH < 0.064156 δηλαδή απλά **LoyalCH < 0.064156.**
* Αυτό το αποτέλεσμα είναι λογικό καθώς το loyalCH εκφράζει το brand loyalty του καταναλωτή για τη κλάση CH (Citrus Hill orange juice). Συνεπώς μικρό loyalty προς την κλάση ch είναι λογικό να σημαίνει ότι ο καταναλωτής ανήκει στην κλάση MM.
* Επιπλέον αξίζει να σημειωθεί ότι στη ρίζα του δέντρου (κόμβος 1) φαίνεται ότι οι παρατηρήσεις μας είναι **imbalanced** καθώς στις 800 παρατηρήσεις του training set έχουμε 61.5% CH και 38.5% MM.

## Γραφική αναπαράσταση δέντρου

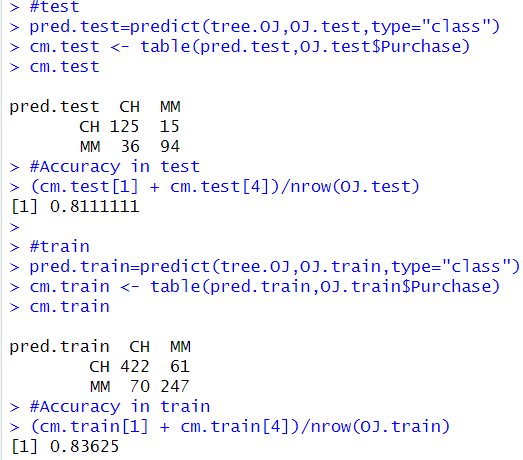
A diagram of a number of numbers

AI-generated content may be incorrect.

* Βλέπουμε τους 8 τερματικούς κόμβους που αναφέραμε και πριν.
* Το δέντρο μας δεν είναι ιδιαίτερα μεγάλο.
* Παρατηρούμε ότι μόνο 3 μεταβλητές από τις 17 (!) χρησιμοποιούνται για τα split του δέντρου.
* Επίσης φαίνεται ότι υπάρχουν 4 τερματικοί κόμβοι με προβλεπόμενη κλάση MM και 4 με CH.
* Μία ακόμα σημαντική παρατήρηση είναι ότι υπάρχουν φύλλα-αδέρφια (τα σημειωμένα στο παρακάτω διάγραμμα), τα οποία έχουν τα ίδια labels. Δηλαδή έχουν δημιουργηθεί splits τα οποία δεν αλλάζουν τη πρόβλεψη του εξεταζόμενου δείγματος, είτε ικανοποιείται η συνθήκη είτε όχι. A screenshot of a computer

  AI-generated content may be incorrect.

## Confusion Matrix



Παρατηρούμε ότι το accuracy στο test set είναι 81.1% (δηλ. error 18.9%) ενώ στο train set είναι 83.6% (δηλ. error 16.4%, όπως είδαμε και στο [ερώτημα b](#_Εκπαίδευση_και_βασικά))

Το αποτέλεσμα αυτό είναι λογικό καθώς το δέντρο εκπαιδεύτηκε στο train set, συνεπώς περιμένουμε καλύτερη απόδοση στα γνωστά δεδομένα.

Ωστόσο και οι 2 αποδόσεις είναι αρκετά καλές και αρκετά κοντά, συνεπώς το σφάλμα γενίκευσης είναι σχετικά μικρό.

## Cross Validation

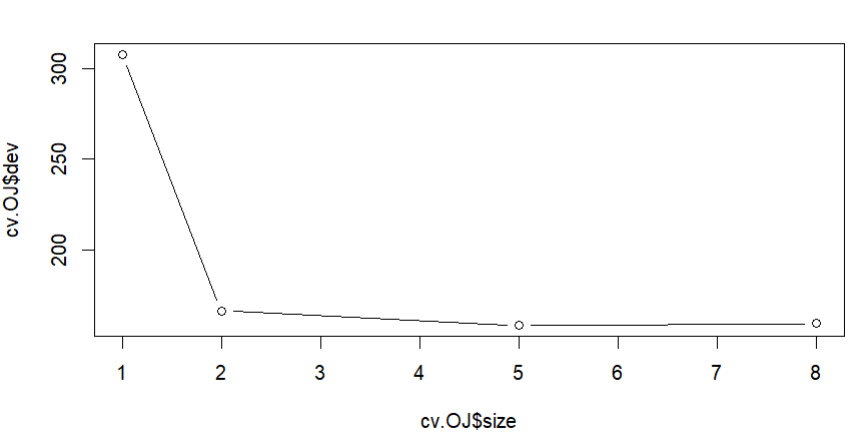
Εφαρμόζουμε cross validation

2. set.seed(42)

3. cv.OJ=cv.tree(tree.OJ,FUN=prune.misclass)

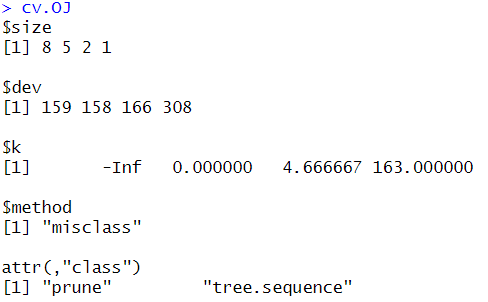
## Γραφική αναπαράσταση αποτελεσμάτων cross validation

Βλέπουμε το σφάλμα σε σχέση με το πλήθος τερματικών κόμβων του δέντρου, όπως προέκυψε από το cross validation



## Βέλτιστο δέντρο

Παρατηρούμε και στο προηγούμενο διάγραμμα, αλλά και στις λεπτομέρειες του cross validation:



ότι το μικρότερο σφάλμα (dev=158) εμφανίζεται για **πλήθος τερματικών κόμβων (size) ίσο με 5**

## Pruning

«Κλαδεύουμε» το δέντρο μας σε πλήθος τερματικών κόμβων ίσο με 5:

1. prune.OJ=prune.misclass(tree.OJ,best=5)

Ας δούμε και τη διαγραμματική απεικόνιση του νέου δέντρου

A diagram of a graph

AI-generated content may be incorrect.

Παρατηρούμε ότι τα φύλλα αδέρφια που είχαν τα ίδια labels (τα οποία σημειώσαμε στο [ερώτημα d](#_Γραφική_αναπαράσταση_δέντρου)) έχουν «κλαδευτεί».

## Διαφορά στο train

A screenshot of a computer code

AI-generated content may be incorrect.

Παρατηρούμε ότι η απόδοση στο train set δεν άλλαξε. Αυτό είναι λογικό στην περίπτωσή μας, καθώς όπως αναφέραμε, δεν έγινε κάποια σημαντική αλλαγή στο δέντρο. Τα splits που κλαδεύτηκαν δεν άλλαζαν τις προβλέψεις στους τερματικούς κόμβους, με αποτέλεσμα οι προβλέψεις στο train set να είναι ακριβώς οι ίδιες, και κατά συνέπεια και το accuracy & classification error.

## Διαφορά στο test

A screenshot of a computer code

AI-generated content may be incorrect.

Παρομοίως η απόδοση στο test set είναι ακριβώς η ίδια για τους λόγους που αναφέραμε.

Υποσημείωση:

Η σταθερή απόδοση πριν και μετά το pruning συνέβη διότι το αρχικό μας δέντρο δεν ήταν ιδιαίτερα μεγάλο. Τα όρια που υπάρχουν ήδη στις συναρτήσεις της R, προκειμένου να περιορίσουν το overfit, ήταν αποτελεσματικά στην περίπτωσή μας και δεν επέτρεψαν στο δέντρο να προσαρμοστεί στο θόρυβο του εν λόγω train data set.

Υπό άλλες συνθήκες εάν υπήρχε σημαντικό overfit στο αρχικό δέντρο, το pruning θα μας επέτρεπε να μειώσουμε το σφάλμα γενίκευσης, επιλέγοντας τις υπερπαραμέτρους που θα οδηγούσαν στην καλύτερη απόδοση στα validation sets του cross validation, και συνεπώς στην καλύτερη απόδοση σε άγνωστα δεδομένα. Σε μία τέτοια περίπτωση θα περιμέναμε η απόδοση στα γνωστά δεδομένα (train set) ενδεχομένως να μειωθεί, καθώς δεν θα επιτρέπαμε στο μοντέλο μας να προσαρμοστεί στις λεπτομέρειες και στο θόρυβο, και η ικανότητα γενίκευσης (συνεπώς και η απόδοση στο test set) να αυξηθεί.

# Άσκηση 4

## Training και test set

1. set.seed(42)

2. train=sample(1:nrow(Carseats), (80/100)\*nrow(Carseats))

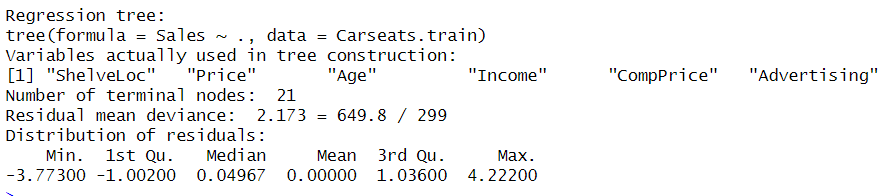
3. Carseats.train=Carseats[train,]

4. Carseats.test=Carseats[-train,]

Χωρίζουμε τα δεδομένα μας σε train (80%) και test (20%)

## Εκπαίδευση, απεικόνιση και διερμήνευση αποτελεσμάτων

Εκπαιδεύουμε το μοντέλο μας στο train set:

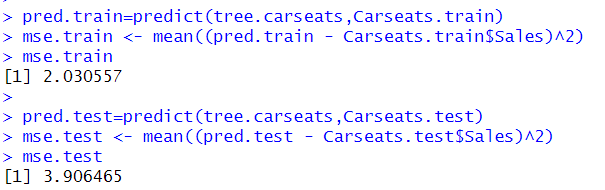


Η γραφική απεικόνιση του δέντρου:

A diagram of a company

AI-generated content may be incorrect.

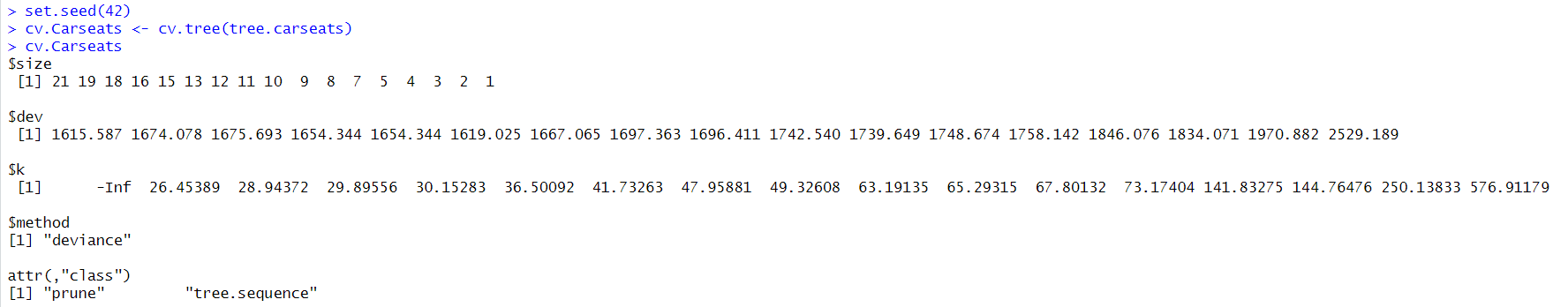
Και το MSE στο train και test



* Παρατηρούμε ότι το δέντρο μας έχει 21 τερματικούς κόμβους και χρησιμοποιεί τις 6 από τις 10 μεταβλητές ("ShelveLoc", "Price", "Age", "Income", "CompPrice", "Advertising")
* Παρατηρούμε επίσης ότι το MSE στο train και στο test είναι σχετικά μικρό (το target variable είναι το sales το οποίο εκφράζεται σε χιλιάδες και το mse έχει διαφορά περίπου 1.9) και τα 2 error είναι αρκετά κοντά μεταξύ τους. Συνεπώς, δεν φαίνεται να υπάρχει σημαντικό overfit.

## Cross validation

Εκτελούμε cross validation



A black lines on a white background

AI-generated content may be incorrect.

Βλέπουμε ότι στο cross validation δίνει ως optimum size το 21, δηλαδή το αρχικό μέγεθος του δέντρου. Συνεπώς το pruning δεν έχει νόημα (καθώς δεν θα αλλάξει τίποτα στο δέντρο), και επομένως οι αποδόσεις τόσο σε train όσο και test set παραμένουν οι ίδιες.

Επίσης αυτό επιβεβαιώνει ότι δεν υπάρχει overfit στο αρχικό μας δέντρο, καθώς φαίνεται πως η μείωση της χωρητικότητας του δέντρου δεν θα βελτιώσει την ικανότητα γενίκευσης.

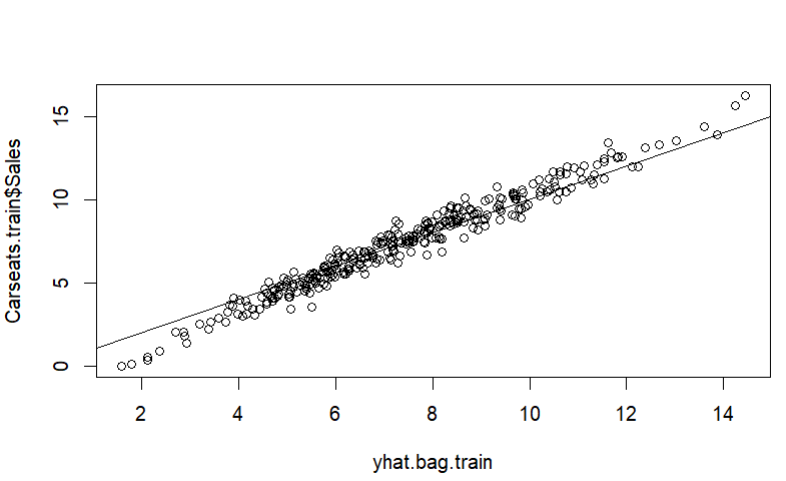
## Bagging

Κάνουμε train ένα bagging μοντέλο στα train data, με

A computer code with blue text

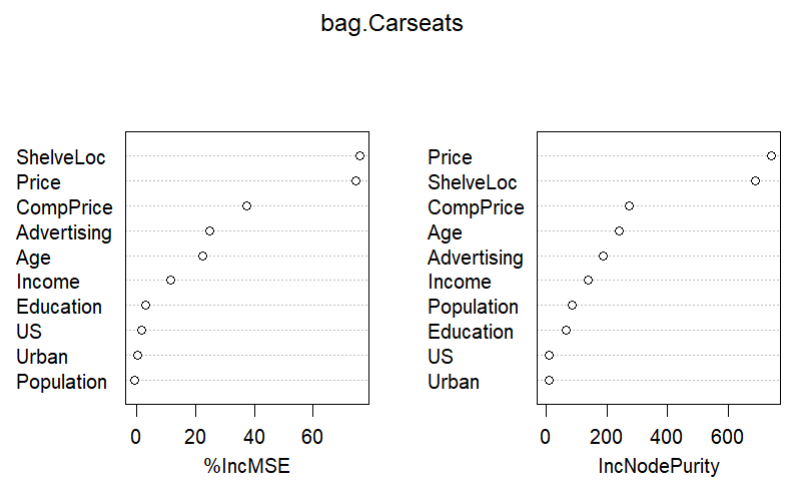
AI-generated content may be incorrect.

Βλέπουμε ότι το MSE στο train είναι 0.42512 ενώ πριν ήταν 2.030557 και το MSE στο train είναι 2.247194 ενώ πριν ήταν 3.906465. Συνεπώς υπάρχει σημαντική βελτίωση τόσο στο train όσο και στο test. Βλέπουμε και τα regression plot :

A line graph with dots and numbers

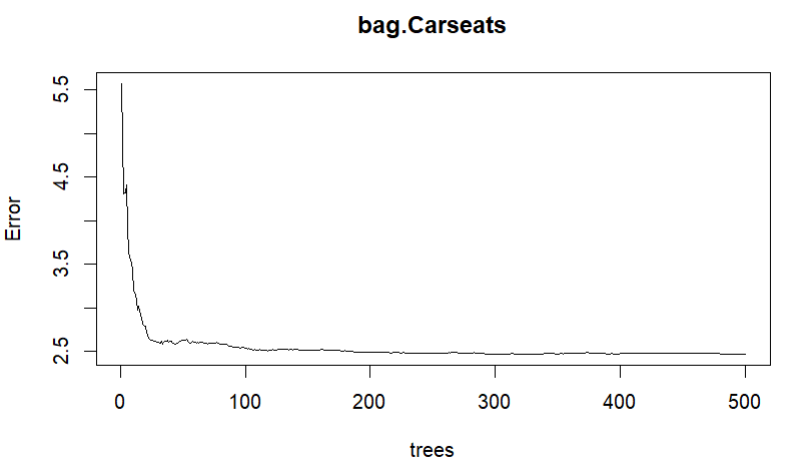
AI-generated content may be incorrect.

Επίσης παρατηρούμε το importance των μεταβλητών :



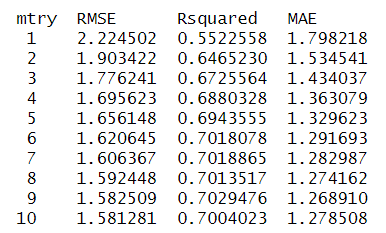
Οι μεταβλητές Shelveloc, Price φαίνονται να είναι οι πιο σημαντικές, ενώ οι μεταβλητές Education, US, Urban, Population φαίνονται να μην είναι ιδιαίτερα σημαντικές, οι οποίες αξίζει να σημειωθεί ότι είναι και αυτές που δεν χρησιμοποιήθηκαν στο αρχικό μας decision tree.

Τέλος να παρατηρήσουμε στο παρακάτω διάγραμμα, ότι τα 500 trees που χρησιμοποιήθηκαν by default, δεν είναι απαραίτητα, και το όφελος στο συγκεκριμένο παράδειγμα από τα επιπλέον δέντρα, σταματάει περίπου στα 100 δέντρα.



## Random forests

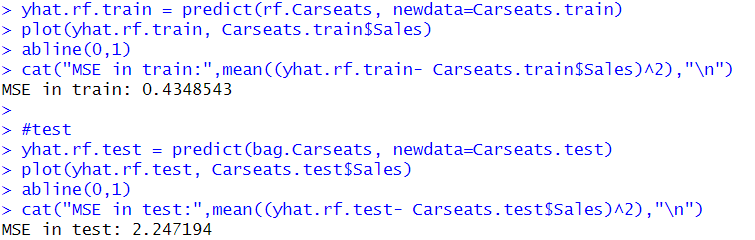
Για το ερώτημα αυτό, χρησιμοποιώντας grid search για την υπερπαράμετρο mtree, (παρότι δε ζητήθηκε) βλέπουμε τα εξής αποτελέσματα:



A graph with a line

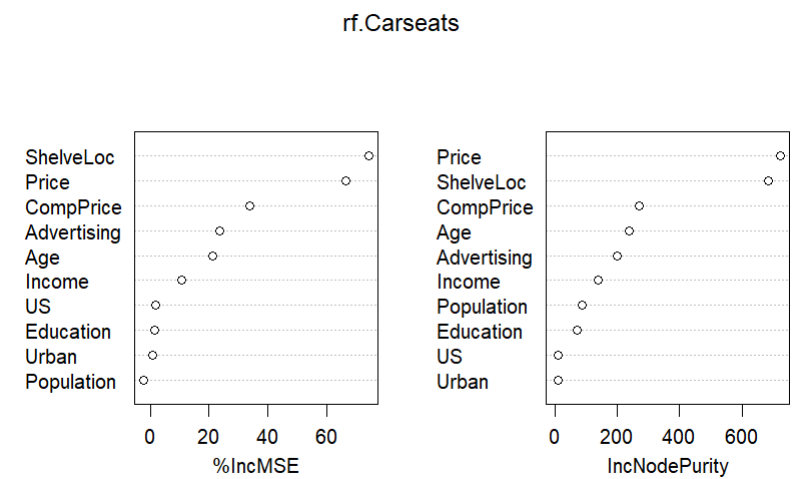
AI-generated content may be incorrect.

Παρατηρούμε ότι το καλύτερο RMSE επιτυγχάνεται με mtry = 10, δηλαδή με **bagging.** Παρόλα αυτά χρησιμοποιούμε mtry = 9 (το οποίο είχε το 2ο καλύτερο RMSE και το καλύτερο R square & MAE) για να εκπαιδεύσουμε ένα μοντέλο random forest και να συγκρίνουμε τα αποτελέσματα. Παρατηρούμε λοιπόν τα εξής:



Το MSE τόσο στο train αυξήθηκε λίγο ενώ στο test έμεινε ίδιο. Συνεπώς το μικρότερο mtry δεν βελτίωσε τα αποτελέσματα. (Με ακόμα μικρότερο mtry θα έχουμε χειρότερα αποτελέσματα με βάση το προηγούμενο διάγραμμα).

Tέλος η σημαντικότητα των μεταβλητών παραμένει σχετικά αμετάβλητη.



Στο συγκεκριμένο παράδειγμα το random forest (δηλαδή η επιλογή m<p ) δεν βελτίωσε τα αποτελέσματα. Αυτό μπορεί να οφείλεται π.χ. στα λίγα data (320 δεδομένα στο train set), στο περιορισμένο πλήθος σημαντικών μεταβλητών (με αποτέλεσμα ορισμένα δέντρα να εκπαιδεύονται σε μη σημαντικές μεταβλητές) ή στο περιορισμένο fine tuning που πραγματοποιήσαμε (ίσως με αλλαγή άλλων υπερπαραμέτρων να επιτύχουμε καλύτερες αποδόσεις).