**ΔΕΥΤΕΡΗ ΕΡΓΑΣΙΑ: Support Vector Machines Classifiacation on MNIST**

**ΜΑΘΗΜΑ: ΝΕΥΡΩΝΙΚΑ ΔΙΚΤΥΑ - ΒΑΘΙΑ ΜΑΘΗΣΗ**

**ΓΙΑΝΝΑΚΟΠΟΥΛΟΣ ΚΩΝΣΤΑΝΤΙΝΟΣ , ΑΕΜ: 16051** (τμήμα Μαθηματικών)

**ΕΙΣΑΓΩΓΗ**

**Βάση Δεδομένων:**

Για την παρούσα εργασία, καθώς και για την επικείμενη 1η Εργασία, χρησιμοποιήθηκε η βάση δεδομένων MNIST, μια πολύ γνώστη βάση δεδομένων, υποσύνολο της μεγαλύτερης NIST. Περιέχει χειρόγραφα ψηφία αριθμών 0,1,…,9. Στην ιστοσελίδα http://yann.lecun.com/exdb/mnist/ υπάρχουν τέσσερα αρχεία διαθέσιμα: ένα training set 60,000 παρατηρήσεων και τα αντίστοιχα labels, καθώς και ένα test set 10,000 παρατηρήσεων και τα αντίστοιχα labels. Οι εικόνες που περιέχονται σε αυτά είναι διαστάσεων 28x28 σε επίπεδα του γκρι και έχει κανονικοποιηθεί το μέγεθος τους αλλά και έχουν κεντροποιηθεί.

**Γλώσσα Προγραμματισμού:**

Χρησιμοποιήθηκε η -ανοικτού κώδικα- γλώσσα προγραμματισμού R, που χρησιμοποιείται ευρέως στον τομέα της υπολογιστικής στατιστικής. Με τις κατάλληλες βιβλιοθήκες και με τη νεοσύστατη βιβλιοθήκη Keras, δίνεται η δυνατότητα υλοποίησης πολλών αλγορίθμων, όπως o kNN και ο nearest Centroid που χρησιμοποιήθηκαν εδώ, αλλά και άλλων αλγορίθμων στατιστικής μάθησης αλλά και αρχιτεκτονικών βαθιάς μάθησης.

**Preprocessing**

Στη βιβλιοθήκη Keras της R, περιέχονται μερικά από τα πιο γνωστά dataset, όπως η MNIST, Cifar100, Cifar10 κλπ. Αφού, λοιπόν, φορτώθηκαν τα δεδομένα και ονομάστηκαν ως:

x\_train: 60,000 x 28 x 28

y\_train: 60,000 (τα αντίστοιχα labels)

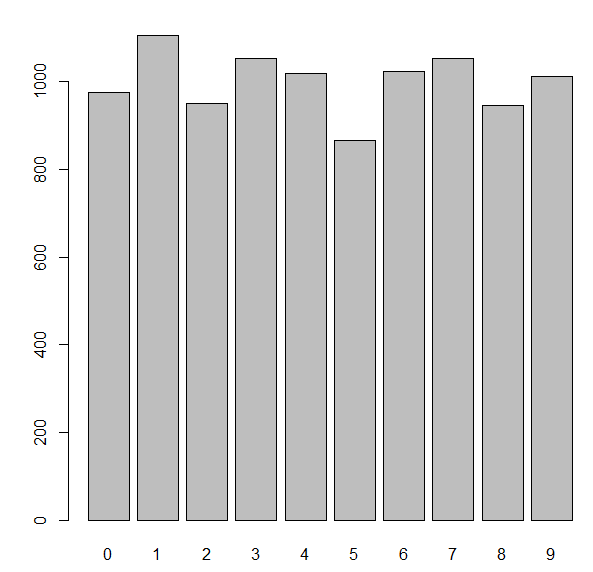
x\_test: 10,000 x 28 x 28

y\_test: 10,000 (τα αντίστοιχα labels)

**SVMs**

Στη γλώσσα R, το interface της βιβλιοθήκης libsvm της python είναι η βιβλιοθήκη e1071.

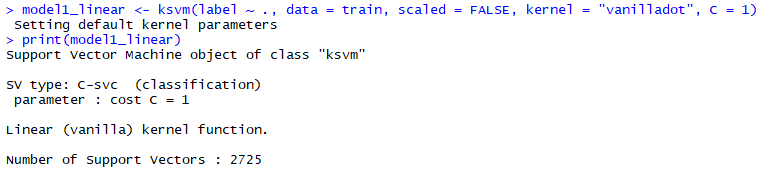
Αρχικά, δημιουργήθηκαν δύο data frames τo train και test τα οποία περιέχουν 784 μεταβλητές και τα labels. Έγινε scaling στο [0,1].

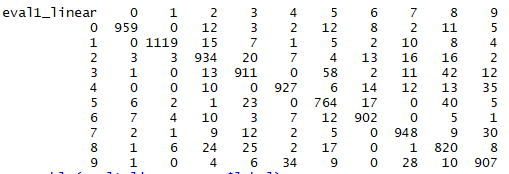
Επίσης, λόγω του υψηλού υπολογιστικού κόστους και προκειμένου να πραγματοποιηθεί grid search για να βρεθούν οι βέλτιστες παράμετροι, πάρθηκε τυχαίο δείγμα 10,000 παρατηρήσεων από το train.

*Τα labels του train στο τυχαίο δείγμα*

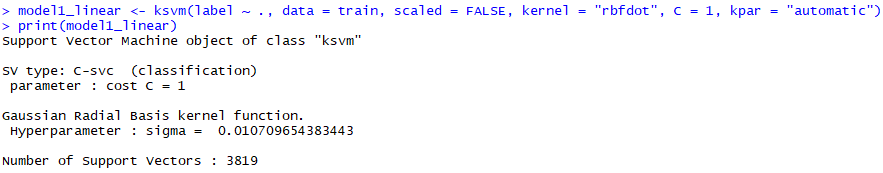
Δεδομένου ότι στην MNIST έχουμε labels με 10 classes ενώ τα SVM κάνουν binary classification, θα πρέπει να βρεθεί μια λύση στο πρόβλημα.

Η βιβλιοθήκη e1071, που χρησιμοποιεί τη libsvm, χρησιμοποιεί για k>2 classes, one-against-all approach, δηλαδή κάνει train σε k(k-1)/2 binary classifiers και η κλάση βρίσκεται με ένα απλό voting.

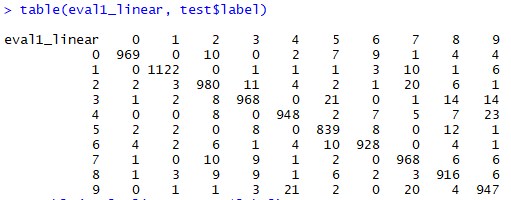
* Δοκιμάζοντας, λοιπόν, ένα SVM με γραμμικό πυρήνα και default cost=1 έχουμε:
* 

**Confusion Matrix:**

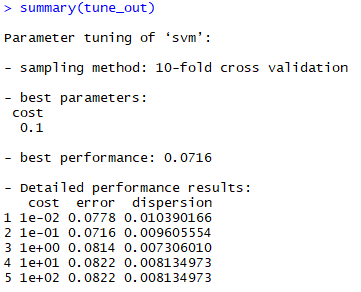
Έχουμε accuracy 91.91%.

* Στη συνέχεια έχουμε ένα SVM με radial πυρήνα και default parameters cost=1 και gamma=0.01:

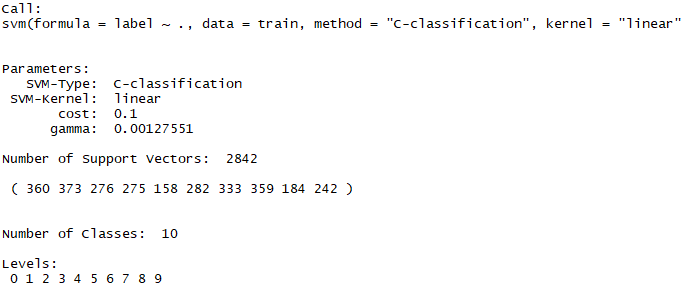
**Confusion Matrix** με accuracy 95.85%:

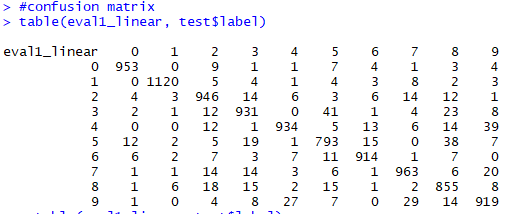


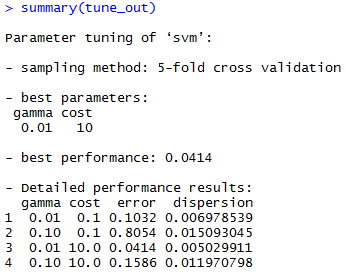
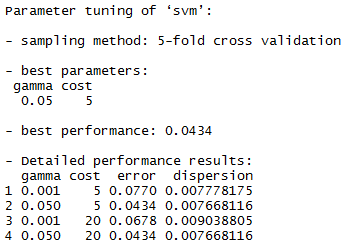
* Στη συνέχεια, πραγματοποιείται grid search, για το μοντέλο με γραμμικό πυρήνα και για τις τιμές 0.01, 0.1, 1, 10, 100 για το cost, με 10-fold cross-validation. Το μικρότερο σφάλμα φαίνεται να δίνεται για τις τιμές 0.1 και 0.01.

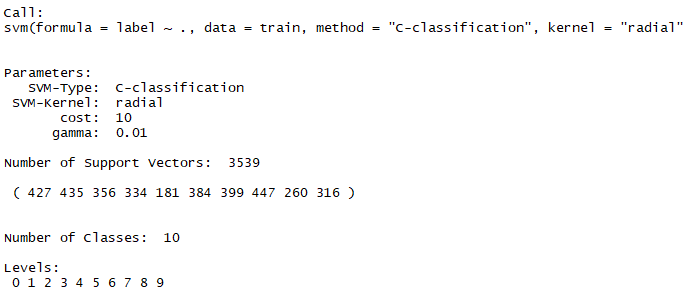


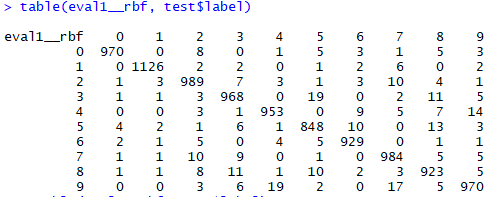
* Επομένως, για το SVM με γραμμικό πυρήνα και cost=0.1 έχουμε:

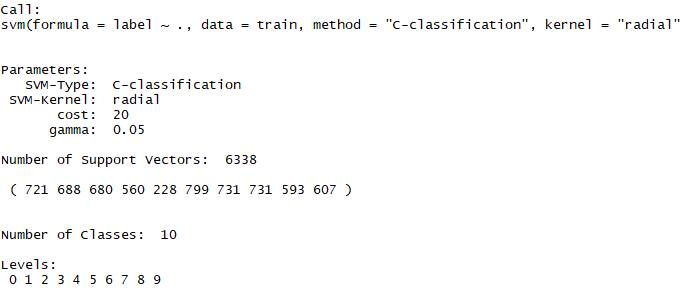


Με **confusion matrix** τον παρακάτω και accuracy 93.28%:

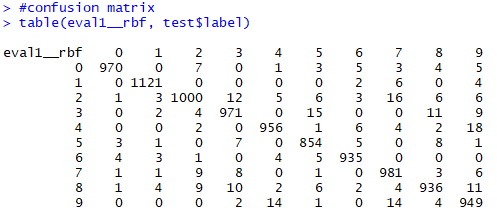
* Στη συνέχεια θα πραγματοποιήσουμε grid search, με 5-fold cross validation για τις ακόλουθες τιμές για το cost και το gamma.
* Για το gamma: 10-2 και 10
* Για το cost: 10-1 και 10
* Στη συνέχεια με νέο grid search για τις τιμές:
* Για το gamma: 0.01 και 0.05
* Για το cost: 5 και 20
* Οπότε με τη σειρά έχουμε:



Με **confusion matrix** τον παρακάτω και accuracy 96.6%:

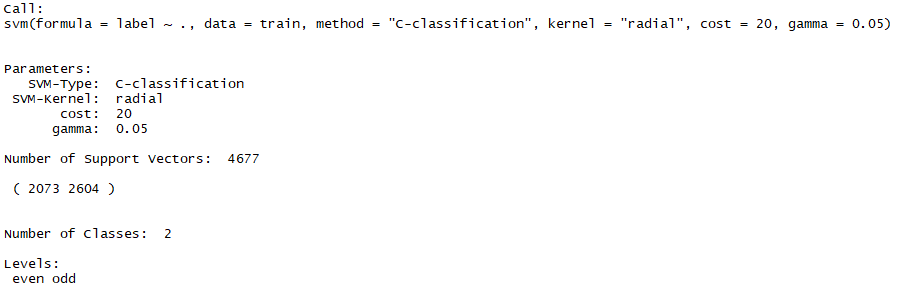


Με **confusion matrix** τον παρακάτω και accuracy 96.73%:



Σε αυτό το σημείο μπορούμε να συγκρίνουμε με τα ανάλογα αποτελέσματα των αλγορίθμων k nearest neighbours και nearest centroid που έδωσαν αποτελέσματα 96.91% (k=1), 97.08% (k=3) και 82.03% στα ίδια δεδομένα.

(Για sample 30000 παρατηρήσεων του train και παραμέτρους cost=3 και gamma=0.04 παίρνουμε accuracy 98.13%, αλλά ο χρόνος για grid search αυξάνεται κατά πολύ)

*  Τέλος, για να εξετάσουμε και ένα δυαδικό πρόβλημα, κωδικοποιούμε τα labels σε odd και even. Στη συνέχεια, δημιουργούμε ένα SVM με παραμέτρους C=20 και gamma=0.05 και έχουμε:

Και **confusion matrix** με accuracy 97.92%:

