



ΑΣΑΦΗ ΣΥΣΤΗΜΑΤΑ

Classification

Κιλάρογλου Ελευθέριος (8501)

Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών
Αριστοτέλειο Πανεπιστήμιο Θεσσαλονίκης

Θεσσαλονίκη 2021

Εισαγωγή

Στόχος της εργασίας αυτής είναι να διερευνηθεί η ικανότητα των μοντέλων TSK (Tagaki-Sugeno-Kang) στην επίλυση προβλημάτων ταξινόμησης (classification). Στο πρώτο κομμάτι της εργασίας χρησιμοποιείται ένα σχετικά μικρό dataset με λίγες διαστάσεις έτσι ώστε να επιφέρει μια απλή διερεύνηση της διαδικασίας εκπαίδευσης και αξιολόγησης μοντέλων αυτού του είδους αλλά και να επιδείξει τον τρόπο ανάλυσης και ερμηνείας των αποτελεσμάτων αυτών. Αντίθετα στο δεύτερο κομμάτι της εργασίας χρησιμοποιείται ένα πολυπλοκότερο dataset με μεγάλο αριθμό διαστάσεων αλλά ταυτόχρονα και μεγάλο σε μέγεθος. Έτσι η διαδικασία εκπαίδευσης και αξιολόγησης των μοντέλων δεν είναι απλή με αποτέλεσμα την ανάγκη να γίνει μια πιο συστηματική προσέγγιση στο πρόβλημα της μοντελοποίησης. Συγκεκριμένα θα χρησιμοποιούνται προεπεξεργαστικά βήματα όπως η επιλογή χαρακτηριστικών (feature selection) για την μείωση των διαστάσεων, καθώς και μεθόδους βελτιστοποίησης των μοντέλων μέσω της διασταυρωμένης επικύρωσης (cross validation) έτσι ώστε να πετύχουμε ένα μοντέλο που να μπορεί να περιγράψει αρκετά καλά το σύστημα δεδομένου των μειωμένων χαρακτηριστικών.

Μέρος 1^ο

Περιγραφή

Το dataset το οποίο επιλέγεται είναι το Haberman's Survival dataset από το UCI repository το οποίο περιλαμβάνει 306 δείγματα και 3 χαρακτηριστικά. Αρχικά το σύνολο δεδομένων διαχωρίζεται σε σύνολα εκπαίδευσης (training), επικύρωσης (validation) και ελέγχου (checking). Πιο συγκεκριμένα χρησιμοποιείται το 60% των δειγμάτων για το σύνολο εκπαίδευσης ενώ από 20% των δειγμάτων σε κάθε ένα από τα εναπομείναντα υποσύνολα. Ο διαχωρισμός του χώρου εισόδου γίνεται με τη μέθοδο Subtractive Clustering και τη συνέχεια εκπαιδεύονται 4 TSK. Στα πρώτα δύο μοντέλα (πρώτη περίπτωση) το subtractive clustering εκτελείται σε όλα τα δεδομένα του συνόλου εκπαίδευσης (class independent) ενώ στα επόμενα δύο μοντέλα (δεύτερη περίπτωση) ο διαμερισμός του χώρου εισόδου γίνεται εφαρμόζοντας clustering στα δεδομένα του συνόλου εκπαίδευσης που ανήκουν στην εκάστοτε κλάση ξεχωριστά (class dependent). Και στις δύο περιπτώσεις τα δυο μοντέλα που προκύπτουν διαφέρουν ως προς την παράμετρο που καθορίζει τον αριθμό των κανόνων. Δηλαδή την ακτίνα randii που χρησιμοποιείται στο subtractive clustering. Τα μοντέλα αυτά περιλαμβάνουν:

- 1) Randii 0.2 Class Independent
- 2) Randii 0.8 Class Independent
- 3) Randii 0.2 Class Dependent
- 4) Randii 0.8 Class Dependent

Και τα 4 μοντέλα εκπαιδεύονται με την υβριδική μέθοδο και τέλος γίνεται η αξιολόγηση των μοντέλων με βάση τους δείκτες απόδοσης *Error – matrix*, *Overall accuracy (OA)*, *Producer's accuracy (PA)*, *User's accuracy (UA)*, \hat{K} .

Αρχεία Matlab

- **main_part_1.m:** Περιέχει το πρόγραμμα το οποίο με την χρήση των βοηθητικών συναρτήσεων που περιγράφονται παρακάτω, εκπαιδεύει και αξιολογεί τα 4 TSK μοντέλα.
- **TSK_Classification.m:** Παίρνοντας κατάλληλες παραμέτρους ως εισόδους εκπαιδεύει ένα TSK μοντέλο και επιστρέφει τις τιμές των δεικτών αξιολόγησης ενώ ταυτόχρονα παράγει τα ζητούμενα διαγράμματα.
- **plotErrors.m:** Παράγει τις καμπύλες εκμάθησης στις οποίες φαίνεται η μείωση του σφάλματος σε σχέση με τις εποχές.
- **plotMFs.m:** Παράγει τα διαγράμματα των συναρτήσεων συμμετοχής των εισόδων ενός TSK μοντέλου.
- **plotPrediction.m:** Παράγει τα διαγράμματα στα οποία απεικονίζονται τα σφάλματα πρόβλεψης.
- **split_scale.m:** Χωρίζει το σύνολο δεδομένων σε σύνολα εκπαίδευσης-επικύρωσης-ελέγχου ενώ ταυτόχρονα κανονικοποιεί τις μεταβλητές.
- **Generate_FIS.m:** Δημιουργεί ένα Fuzzy Interference System (FIS) με τη μέθοδο του Subtractive Clustering είτε για class dependent μοντέλο είτε για class independent.
- **Generate_Fis_Explation.m:** Εξηγεί βήμα βήμα τον κώδικα του **Generate_FIS.m**.
- **GetClasses.m:** Επιστρέφει τις κλάσεις του συνόλου δεδομένων.
- **GetSameFrequencySamples.m:** Επιστρέφει training, validation και checking data τα οποία να έχουν μέγιστη διαφορά συχνότητας εμφάνισης δειγμάτων της κάθε κλάσης κάτω από ένα κατώφλι που δέχεται ως input.

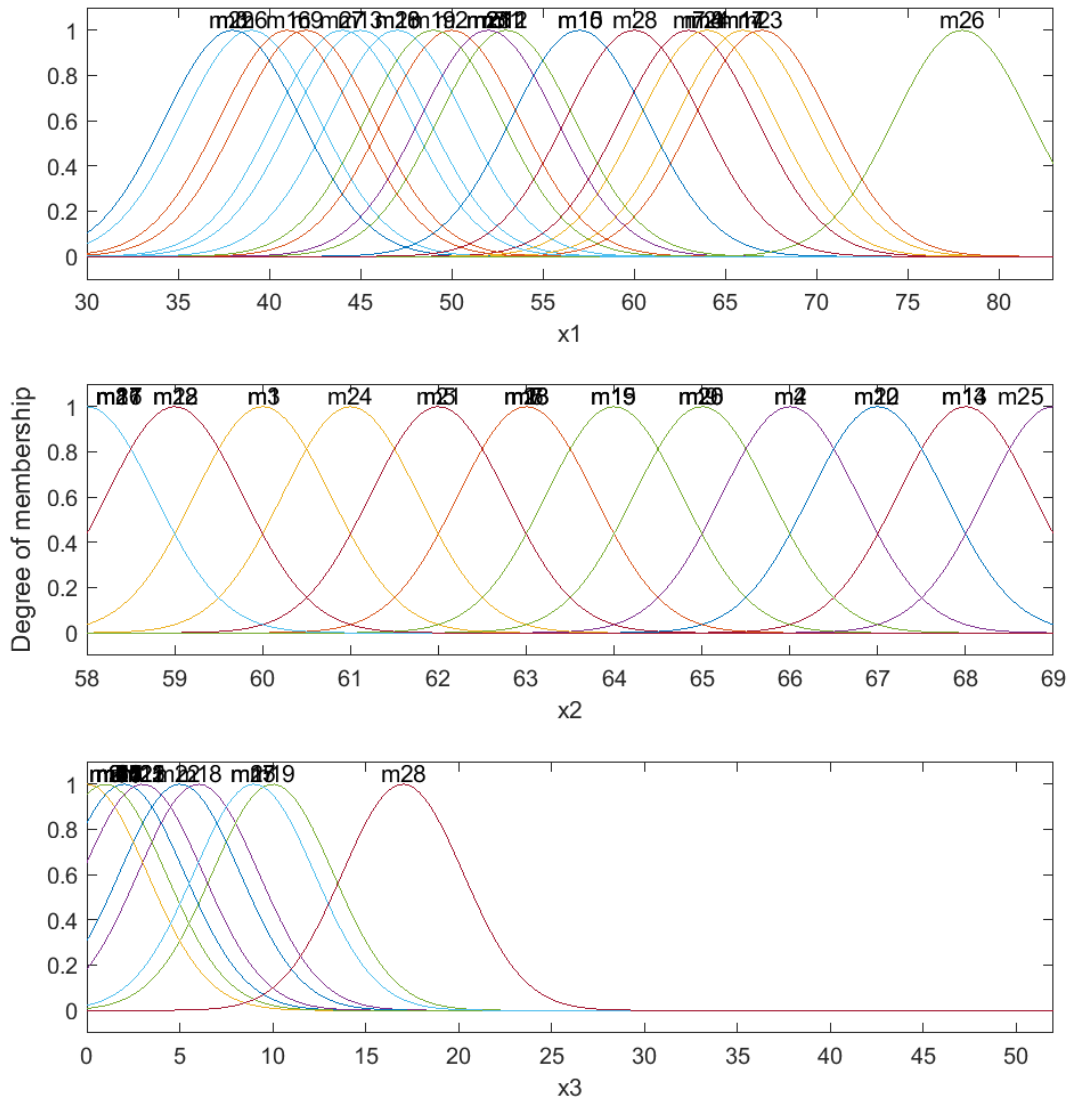
Σε αυτό το σημείο να τονιστεί ότι δε χρησιμοποιήθηκε η GetSameFrequencySamples.m καθώς η επιλογή για stratification της συνάρτησης cvpartition έδινε καλύτερα αποτελέσματα:

Διαγράμματα

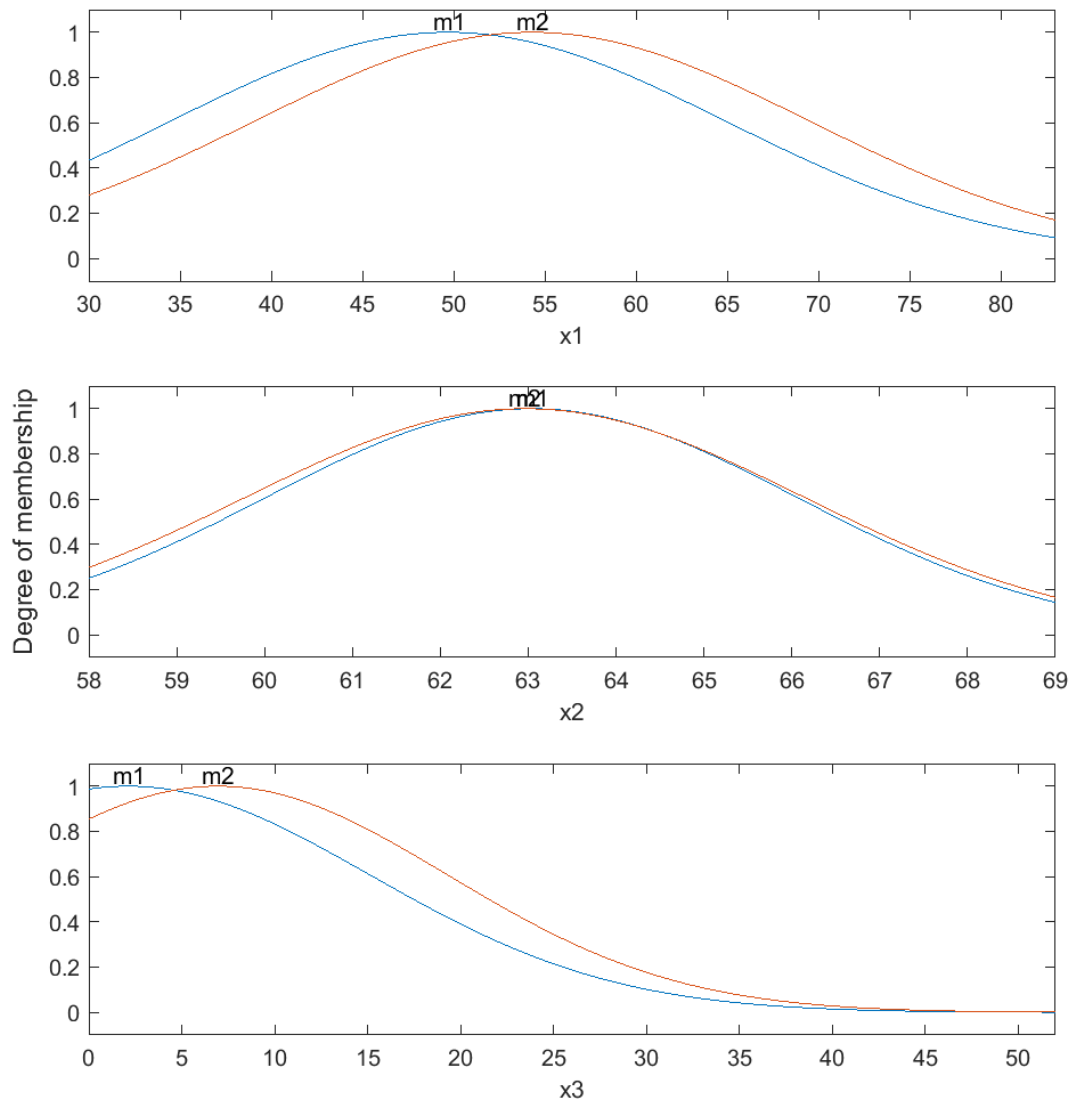
Παρακάτω εμφανίζονται 3 ειδών διαγράμματα για κάθε μια από τις 4 περιπτώσεις.

- i) Ασαφή σύνολα - Διαγράμματα στα οποία απεικονίζονται οι τελικές μορφές των ασαφών συνόλων που προέκυψαν μέσω της διαδικασίας εκπαίδευσης:
- ii) Διαγράμματα εκμάθησης – Φαίνονται οι καμπύλες εκμάθησης στις οποίες φαίνεται η μείωση του σφάλματος RMSE σε σχέση με τις εποχές εκπαίδευσης.
- iii) Διαγράμματα πρόβλεψης - Εδώ με μπλέ χρώμα βλέπουμε τις τιμές πρόβλεψης ενώ με κόκκινο χρώμα τις πραγματικές τιμές. Στον άξονα y έχουμε την τιμή της εξόδου ενώ στον άξονα x είναι το index του δείγματος που εξετάζουμε

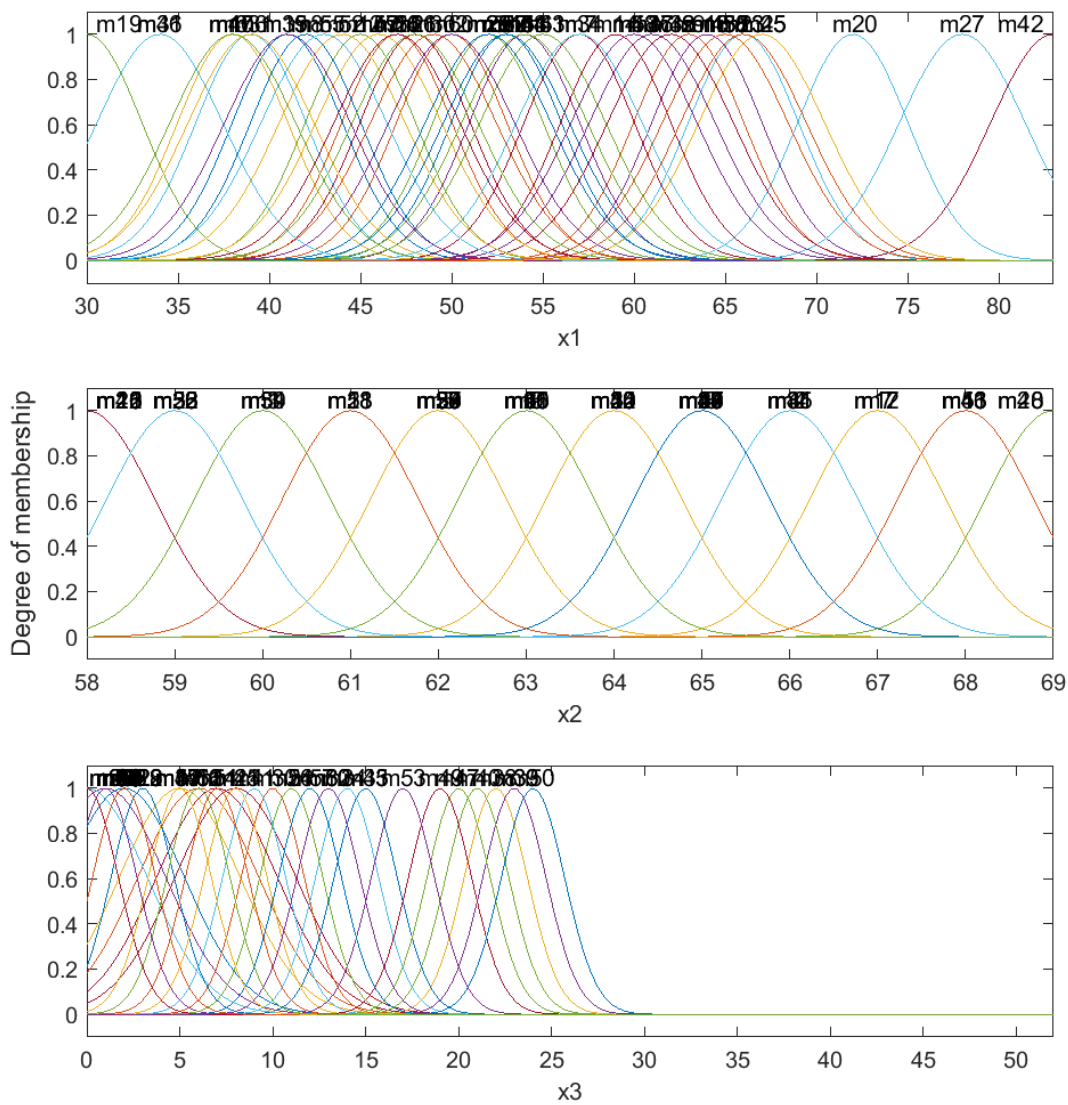
Ασαφή σύνολα:



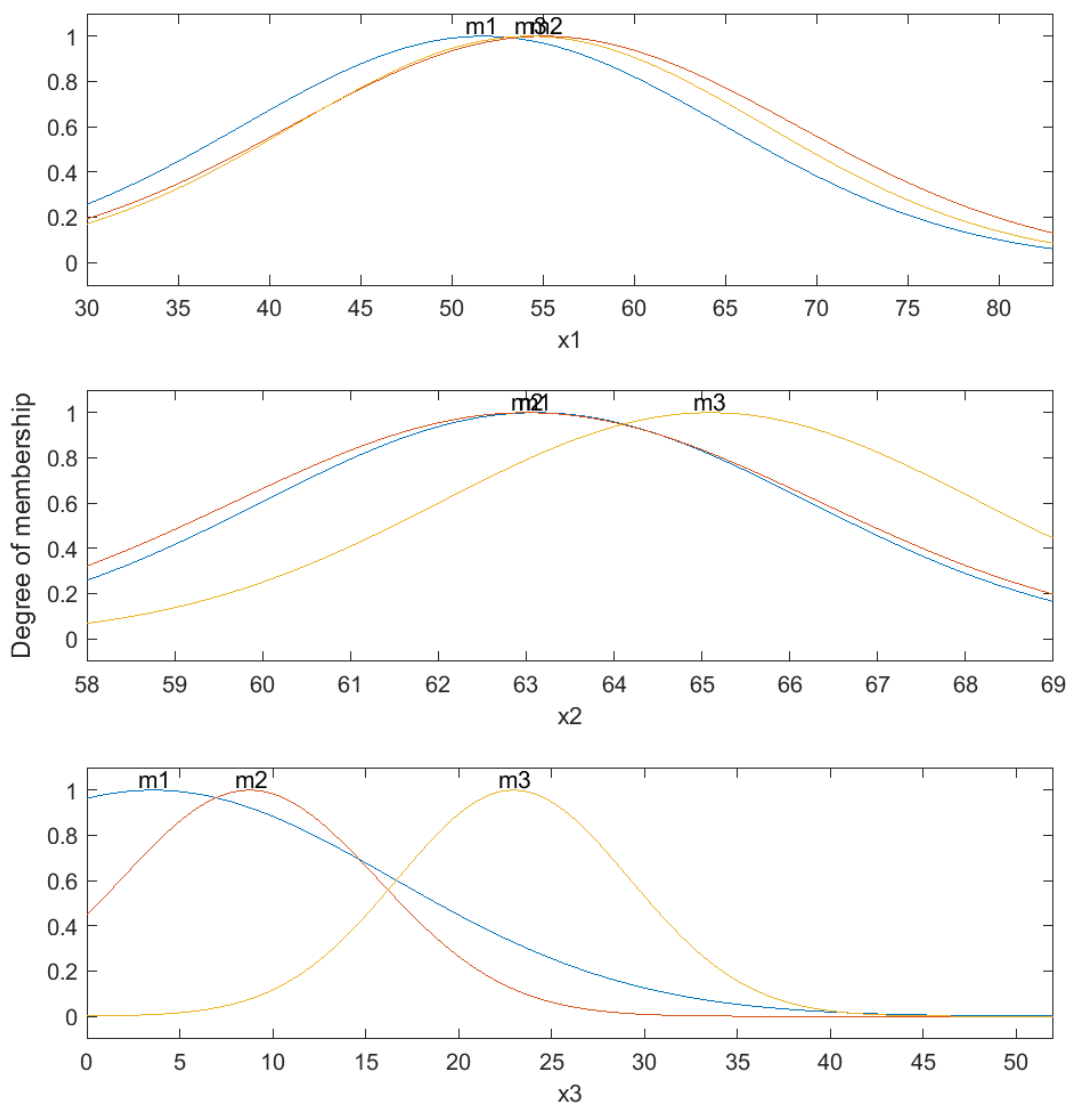
Διάγραμμα 1 - Randii 0.2 / Class Independent



Διάγραμμα 2 - Randii 0.8 / Class Independent

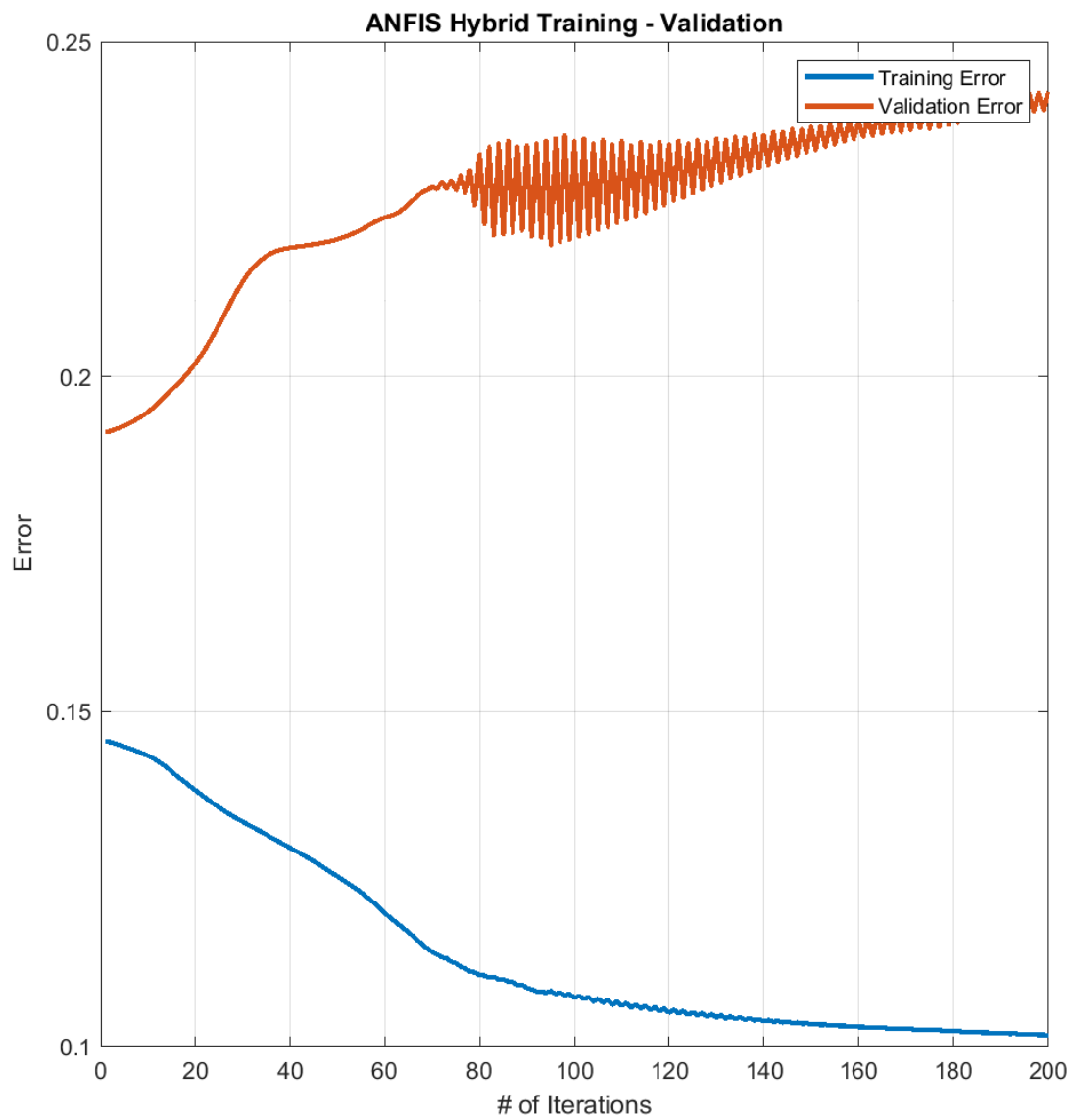


Διάγραμμα 3 - Randii 0.2 / Class Dependent

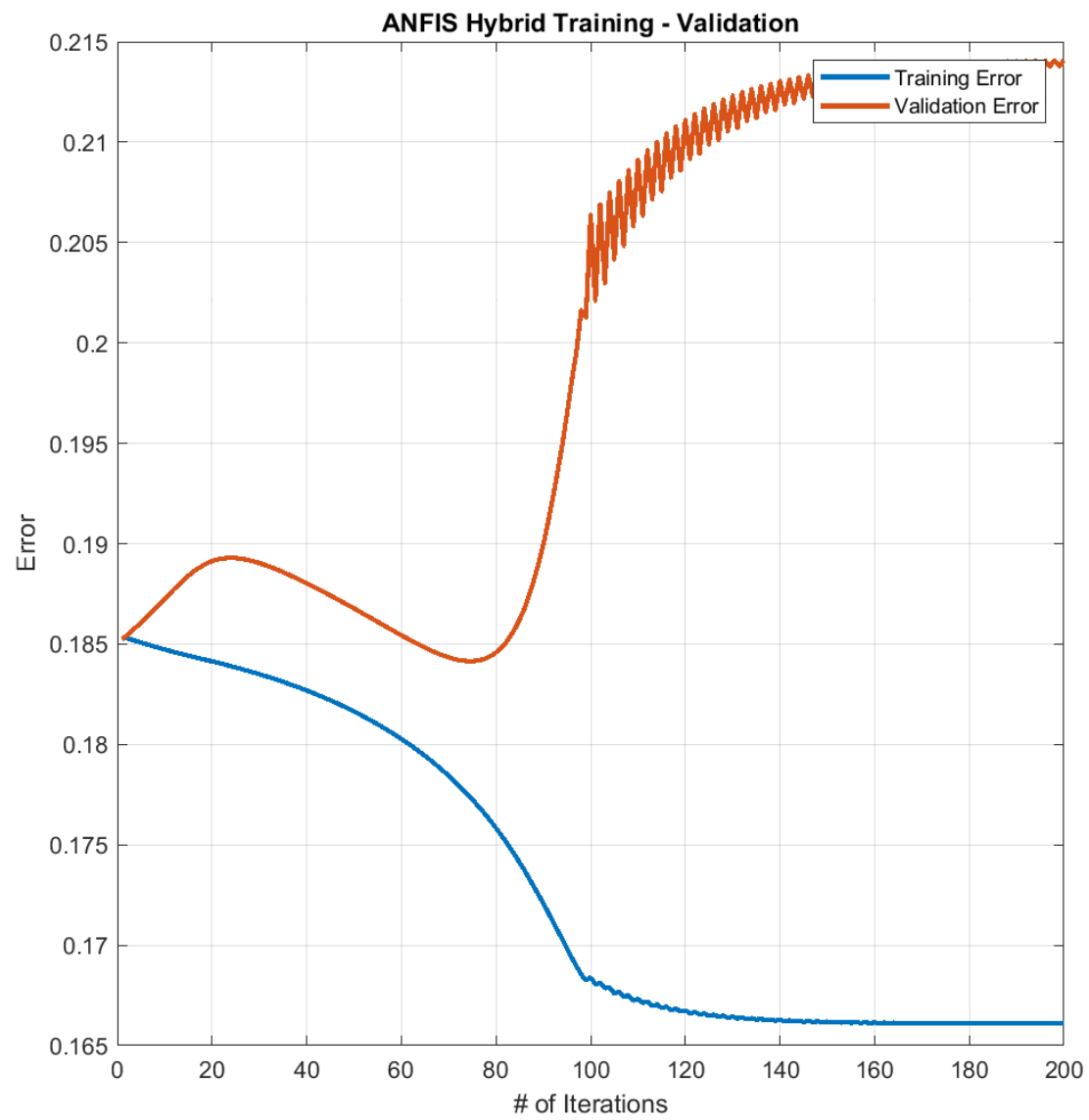


Διάγραμμα 4 - Randii 0.8 / Class Dependent

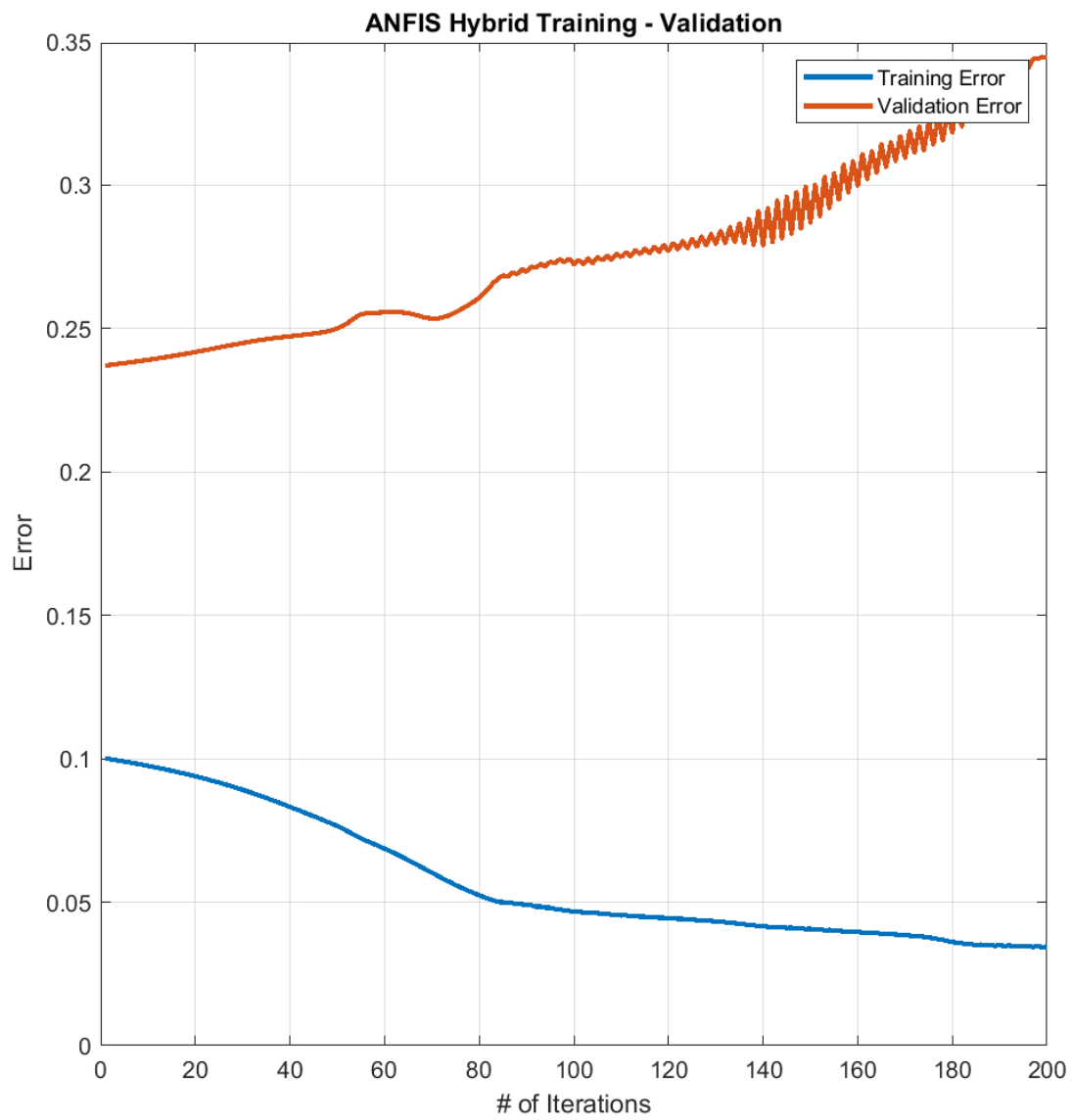
Διαγράμματα μάθησης:



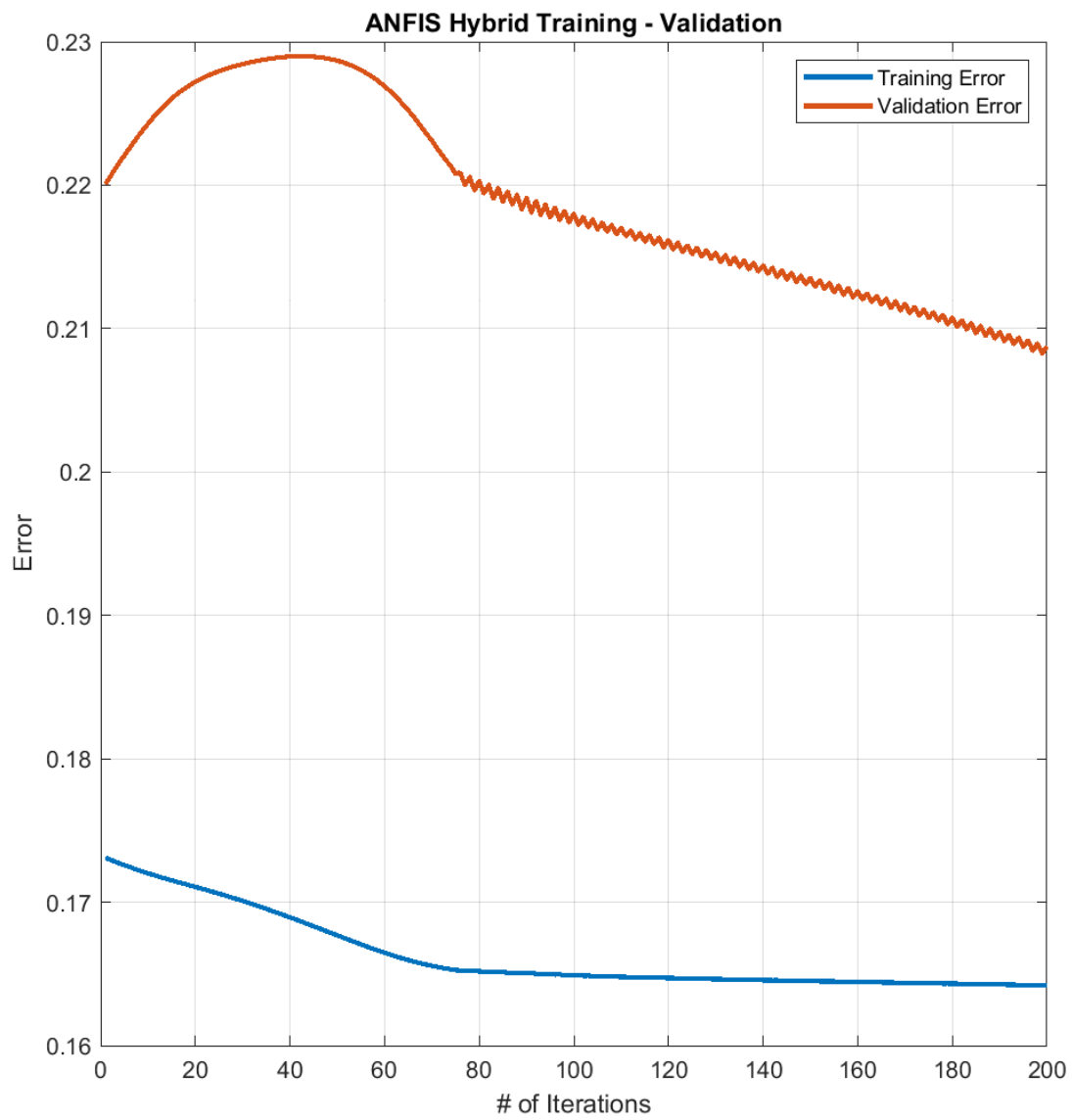
Διάγραμμα 5 - Εκμάθηση με $randii$ 0.2 / Class Independent



Διάγραμμα 6 - Εκμάθηση με $Randii$ 0.8 / Class Independent

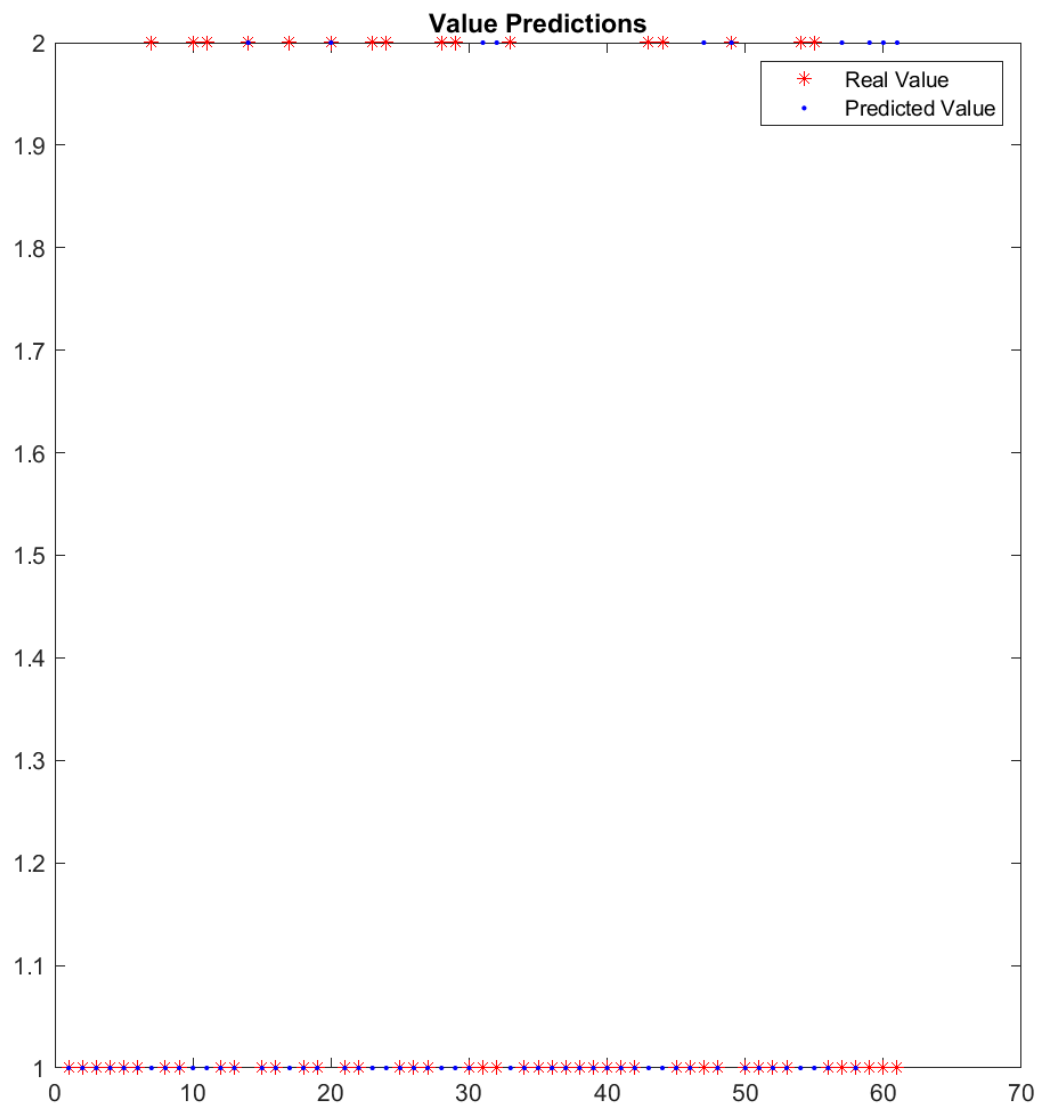


Διάγραμμα 7 - Εκμάθηση με $Randii$ 0.2 / Class Dependent

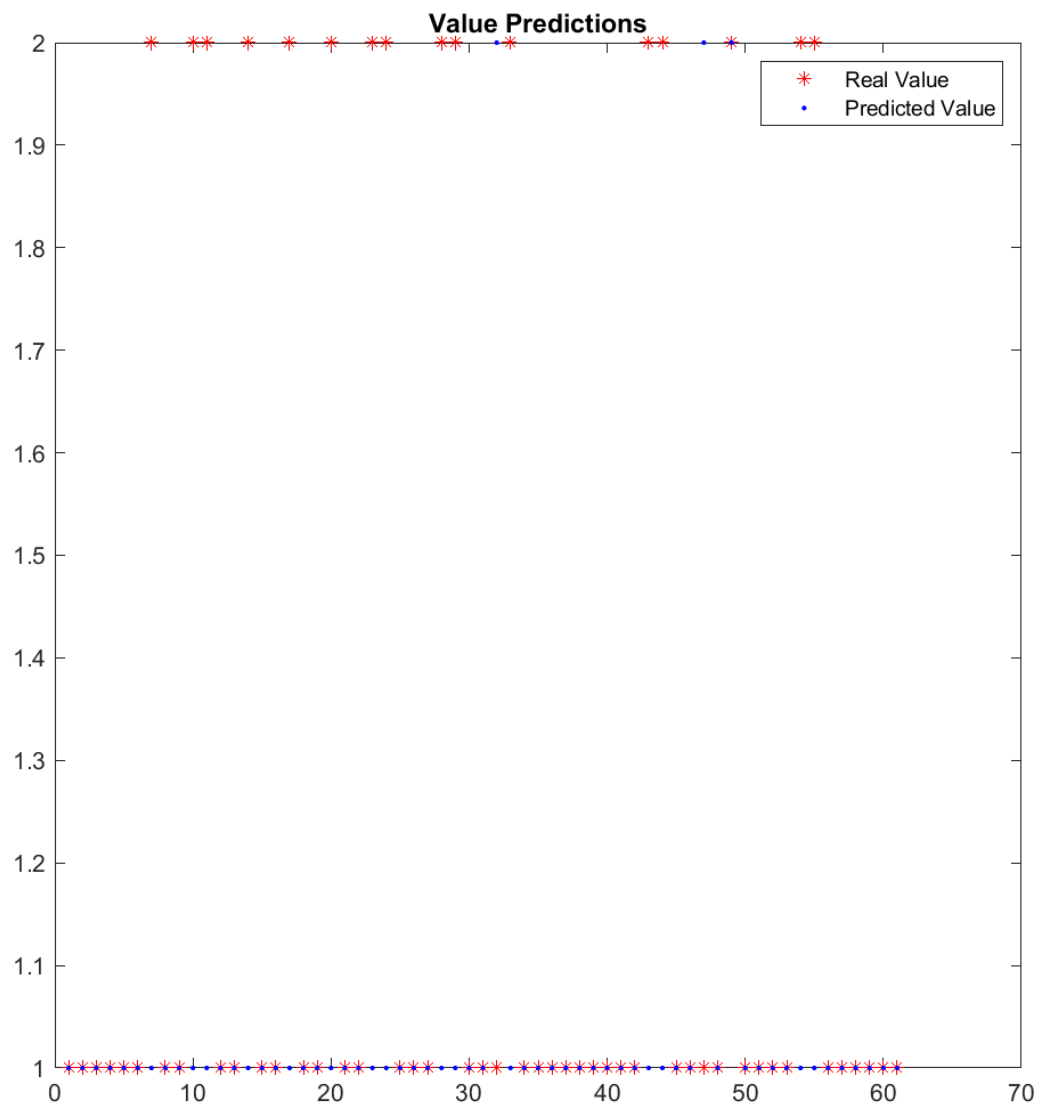


Διάγραμμα 8 - Εκμάθηση με $Randii$ 0.8 / Class Dependent

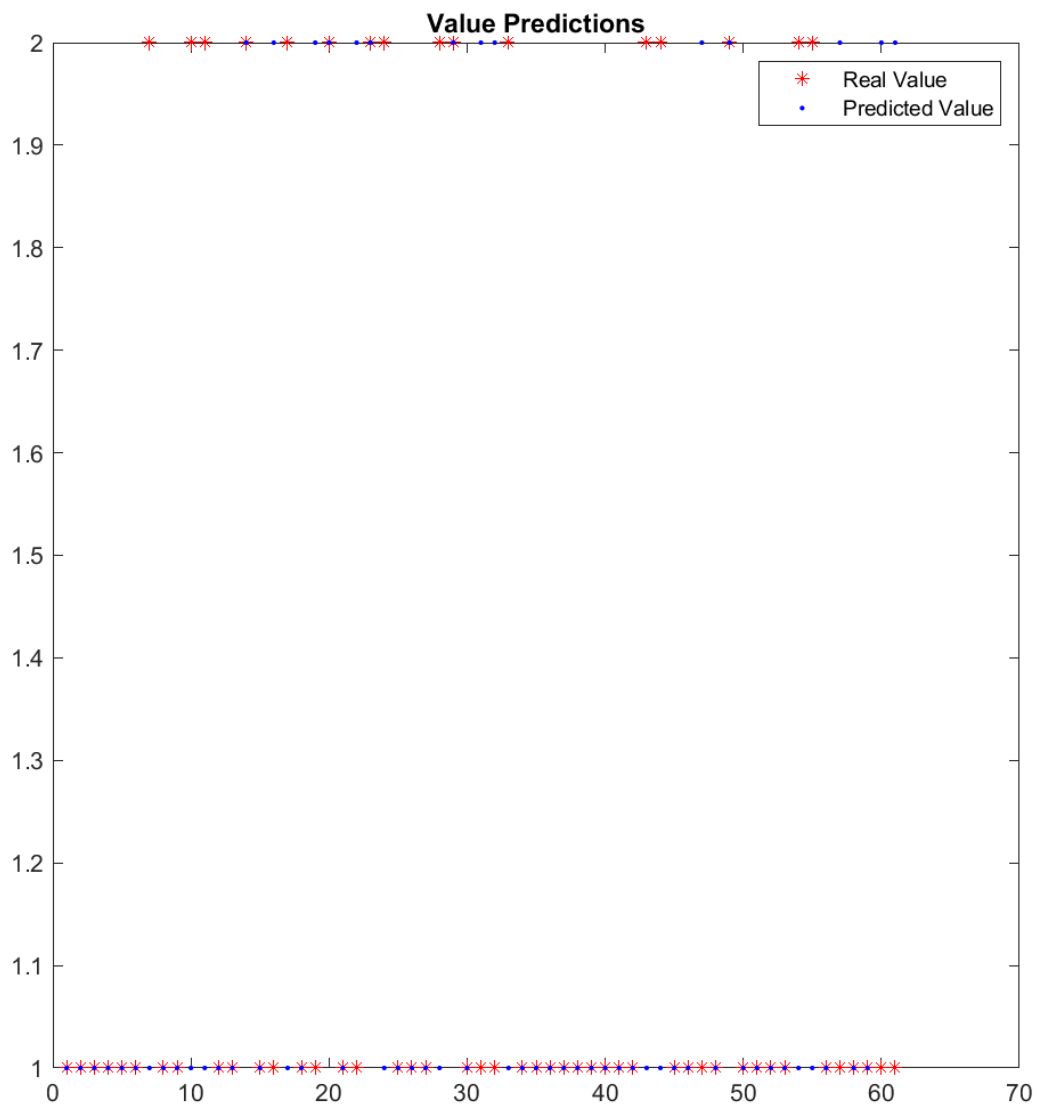
Διαγράμματα πρόβλεψης:



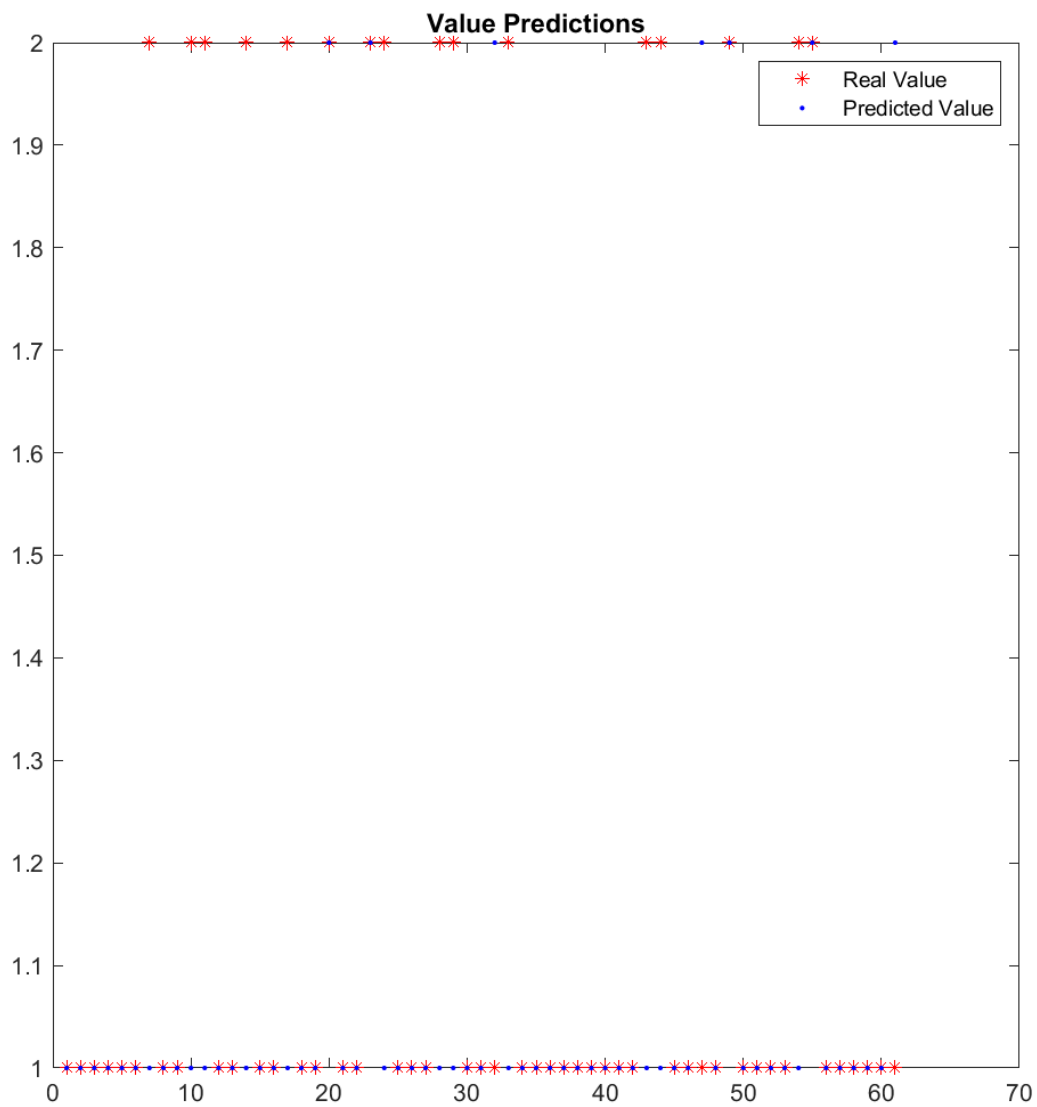
Διάγραμμα 9 - Πρόβλεψη με Randii 0.2 / Class Independent



Διάγραμμα 10 - Πρόβλεψη με Randii 0.8 / Class Independent



Διάγραμμα 11 - Πρόβλεψη με Randii 0.2 / Class Dependent



Διάγραμμα 12 - Πρόβλεψη με Randii 0.8 / Class Dependent

Πίνακες σφάλματος

Πίνακας 1 - Randii 0.2 / Class Independent

		Real Values	
		1	2
Predicted Values	1	38	13
	2	7	3

Πίνακας 2 - Randii 0.8 / Class Independent

		Real Values	
		1	2
Predicted Values	1	43	15
	2	2	1

Πίνακας 3 - Randii 0.2 / Class Dependent

		Real Values	
		1	2
Predicted Values	1	36	11
	2	9	5

Πίνακας 4 - Randii 0.8 / Class Dependent

		Real Values	
		1	2
Predicted Values	1	42	12
	2	3	4

Πίνακας 5 - Δείκτες Απόδοσης

Randii – Class Dependency	OA	User's Accuracy		Producer's Accuracy		\hat{K}
		1	2	1	2	
0.2 – Class Independent	0.67312	0.7451	0.3000	0.8444	0.1875	0.036335
0.8 – Class Independent	0.72131	0.7414	0.3333	0.9556	0.0625	0.024459
0.2 – Class Dependent	0.67312	0.7660	0.3571	0.8000	0.3125	0.11722
0.8 – Class Dependent	0.7541	0.7778	0.5714	0.9333	0.2500	0.22392

Παρατηρήσεις

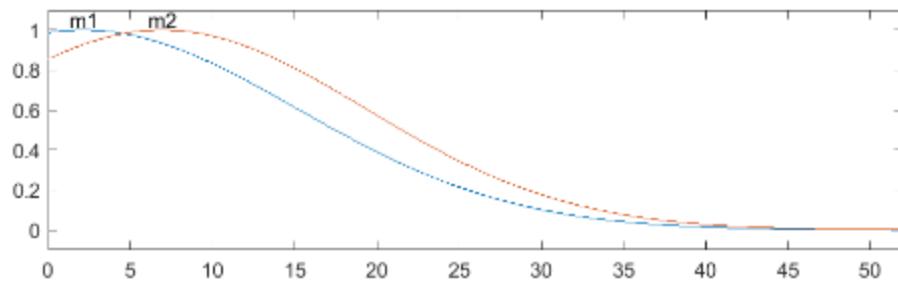
Παρατηρώντας τον παραπάνω πίνακα φαίνεται ότι στην περίπτωση των class dependent μοντέλων παρόλο που η συνολική ακρίβεια της πρόβλεψης (Overall Accuracy) παραμένει σχεδόν ίδια, το \bar{K} παίρνει μεγαλύτερες τιμές. Το \bar{K} περιέχει την πληροφορία που μας παρέχουν και οι 3 άλλες μετρικές και ουσιαστικά δείχνει το κατά πόσο συμφωνούν οι προβλέψεις με τις πραγματικές τιμές. Αυτό σημαίνει ότι στην περίπτωση των class dependent μοντέλων οι προβλέψεις συμφωνούν περισσότερο με τις πραγματικές τιμές. Αυτό συμβαίνει επειδή στην περίπτωση αυτή το subtractive clustering γίνεται ξεχωριστά σε κάθε κλάση με αποτέλεσμα να έχουμε «καθαρότερα» clusters και κατά συνέπεια και περισσότερους κανόνες. Έτσι η ερμηνευσιμότητα του κάθε κανόνα/cluster αυξάνεται με αποτέλεσμα να προκύπτουν «καθαρότερα» συμπεράσματα.

Αντίθετα εξετάζοντας την περίπτωση των 0.2 – Class Independent και 0.8 – Class Independent μοντέλων, παρόλο που όπως βλέπουμε στα Διαγράμματα 1 και 2 ο αριθμός των κανόνων παρουσιάζει μεγάλη διαφορά, η επίδοση του μοντέλου (δείκτης απόδοσης \bar{K}) αυξάνεται λίγο μόνο. Αυτό συμβαίνει επειδή περισσότεροι κανόνες δε σημαίνει απαραίτητα καθαρότερα clusters ή καλύτερη διαμέριση του χώρου της εισόδου.

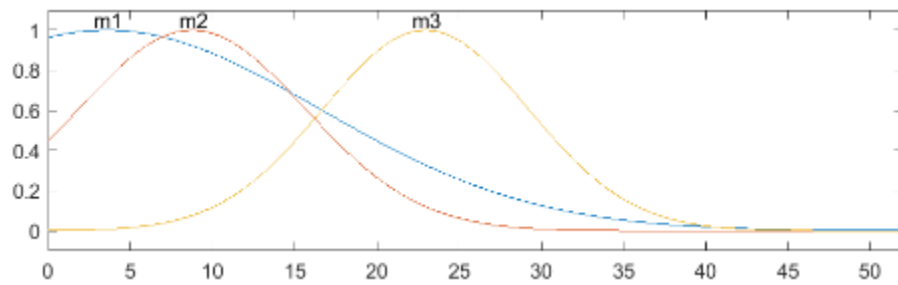
Τέλος εξετάζοντας την περίπτωση των 0.2 – Class Dependent και 0.8 – Class Dependent μοντέλων παρατηρούμε πως το δεύτερο παρουσιάζει καλύτερη επίδοση από το πρώτο παρόλο που με βάση την προηγούμενη ανάλυση δε θα έπρεπε να συμβαίνει αυτό. Αυτό συμβαίνει επειδή δεν έχει πάρα πολλούς κανόνες (όπως τα μοντέλα στα οποία το subtractive clustering εφαρμόστηκε με ακτίνα 0.2) κάτι το οποίο οδηγεί σε overfitting του μοντέλου στο σύνολο εκπαίδευσης. Αυτό μπορεί να φανεί και από τις καμπύλες εκμάθησης που φαίνονται παραπάνω. Για την περίπτωση των 0.2 – Class Independent και 0.2 – Class Dependent μοντέλων οι κανόνες είναι πάρα πολλοί και το φαινόμενο του overfitting φαίνεται από τις πρώτες κιόλας εποχές κάτι το οποίο δεν παρατηρείται στην περίπτωση των άλλων μοντέλων.

Σε αυτό το σημείο να αναφερθεί πως ακόμη και στην καλύτερη περίπτωση οι μετρικές δεν ήταν καλές. Αυτό σημαίνει ότι το μοντέλο δεν αποδίδει επαρκώς στο συγκεκριμένο dataset. Αυτό μπορεί να είναι αποτέλεσμα ενός θορυβώδους και μικρού dataset η ποιότητα του οποίου δεν επιτρέπει στο μοντέλο να προσαρμοστεί κατάλληλα και να επιλύσει το πρόβλημα της ταξινόμησης.

Σε επόμενο βήμα σχετικά με την επικάλυψη των ασαφών συνόλων όπως αναφέρθηκε και προηγουμένως η αύξηση των κανόνων δε σημαίνει απαραίτητα καλύτερη διαμέριση του χώρου εισόδου. Για παράδειγμα εξετάζοντας την απλουστευμένη περίπτωση των 0.8 – Class Independent και 0.8 – Class Dependent μοντέλων παρατηρούμε ότι για την τρίτη είσοδο η αύξηση των κανόνων επιφέρει καλύτερη διαμέριση του χώρου της:

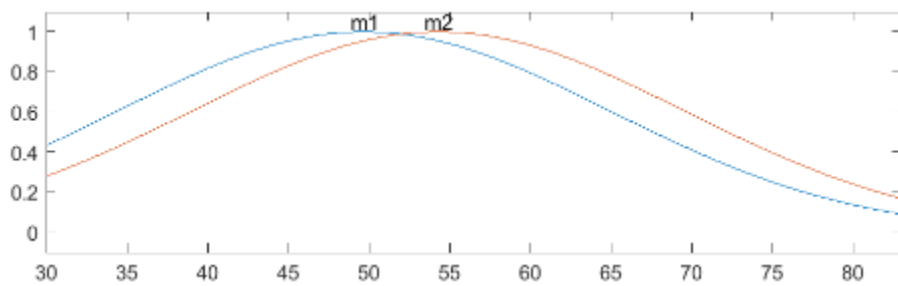


Διάγραμμα 13 - 3η εισόδος - Class Independent - 2 κανόνες

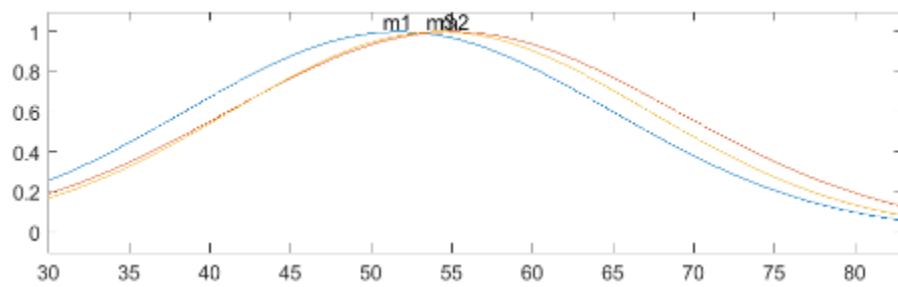


Διάγραμμα 14 - 3η εισόδος - Class Dependent - 3 κανόνες

Ενώ αντίθετα η αύξηση των κανόνων της πρώτης εισόδου δεν επιφέρει κάποια βελτίωση:

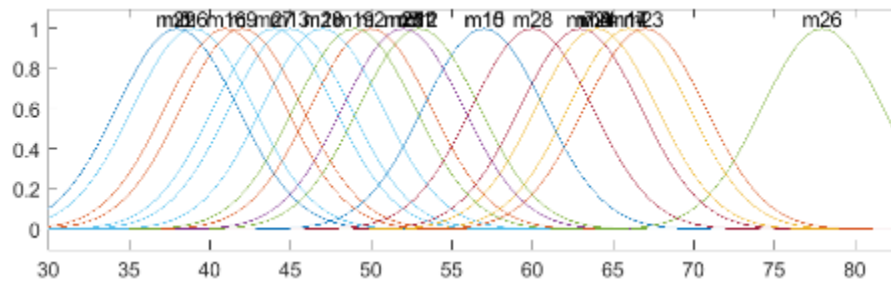


Διάγραμμα 15 - 1η εισόδος - Class Independent - 2 κανόνες

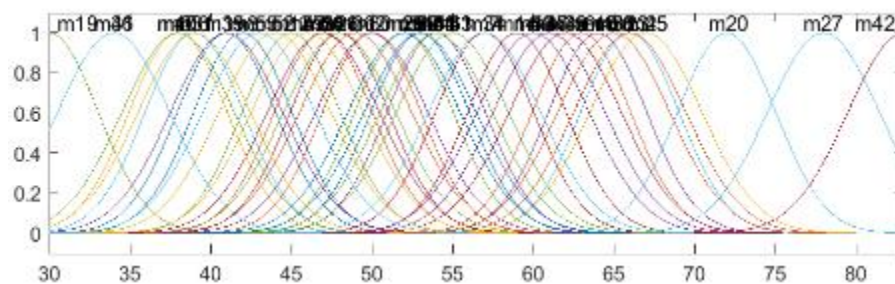


Διάγραμμα 16 - 1η εισόδος - Class Dependent - 3 κανόνες

Εξετάζοντας τώρα μια άλλη περίπτωση, συγκεκριμένα την πρώτη είσοδο των 2 πολυπλοκότερων μοντέλων:



Διάγραμμα 17 – 1η είσοδος λίγοι κανόνες



Διάγραμμα 18 - 1η είσοδος πολλοί κανόνες

παρατηρούμε ότι η αύξηση των κανόνων επιφέρει «καθαρότερα» σύνολα και κατά συνέπεια καλύτερη διαμερίση του χώρου εισόδου στις αρχικές αλλά και τις τελικές τιμές χώρου αυτού. Αντίθετα στις υπόλοιπες τιμές η αύξηση των κανόνων επιφέρει «λιγότερο καθαρά» σύνολα με αποτέλεσμα την αύξηση του θορύβου και δυσμενέστερα αποτελέσματα. Αφαιρώντας λοιπόν τα επιπλέον σύνολα (και κατά συνέπεια τους επιπλέον κανόνες) που προστέθηκαν στην περιοχή αυτή, η βελτίωση του τμήματος υπόθεσης θα ήταν εμφανής.

Οι παραπάνω παρατηρήσεις λοιπόν μπορούν να χρησιμοποιηθούν για τη βελτίωση της σχεδίασης του τμήματος υπόθεσης. Αυτό μπορεί να επιτευχθεί με τη χρήση διαφορετικού αριθμού ασαφών συνόλων για κάθε χαρακτηριστικό αλλά και τη χρήση «καθαρότερων» clusters. Έτσι τα αποτελέσματα θα περιέχουν αισθητά λιγότερο θόρυβο, ενώ το μοντέλο θα παράγει ακριβέστερες προβλέψεις.

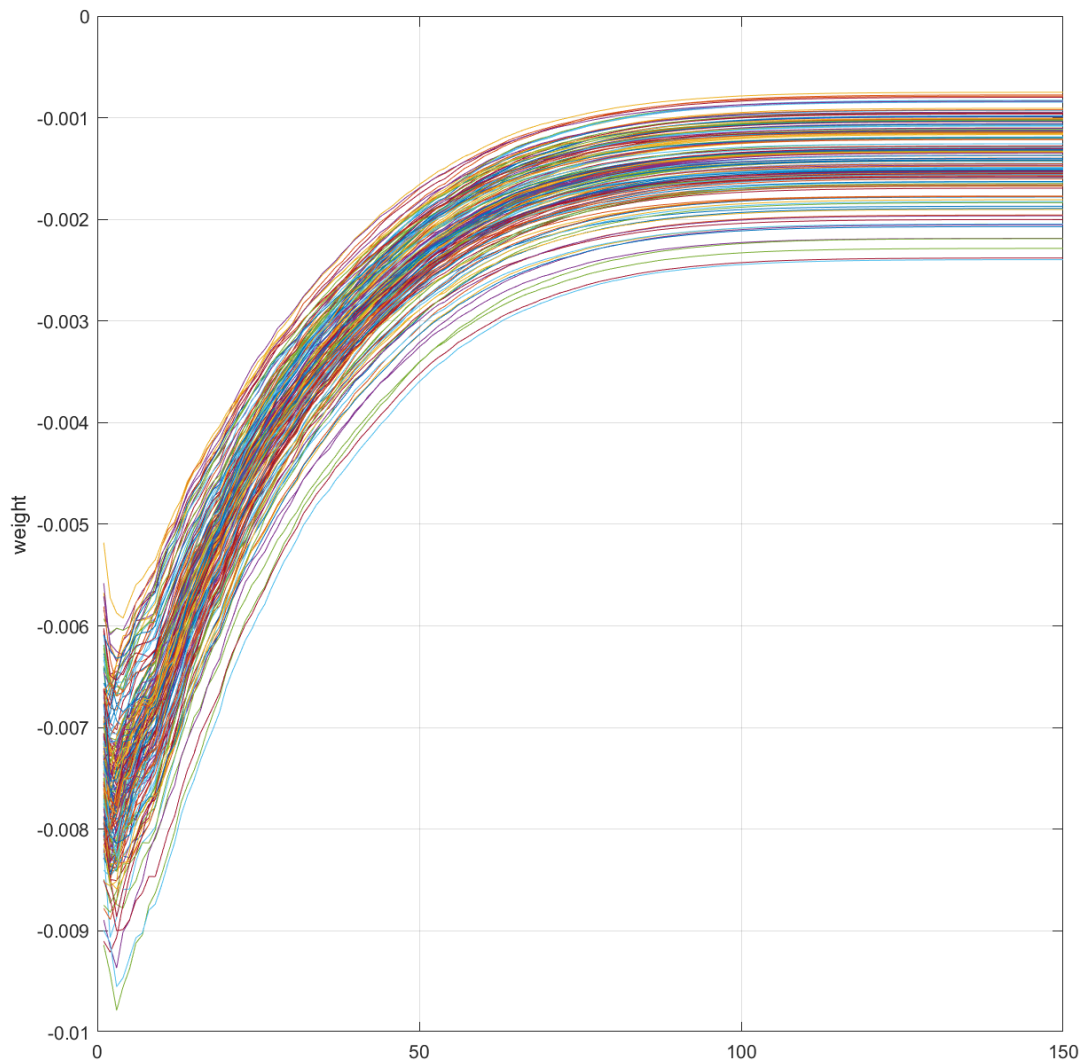
Μέρος 2°

Περιγραφή

Το dataset το οποίο επιλέγεται σε αυτήν την περίπτωση είναι το Epileptic Seizure Recognition dataset από το UCI repository το οποίο περιλαμβάνει 11500 δείγματα και 179 χαρακτηριστικά. Αρχικά το σύνολο δεδομένων διαχωρίζεται όπως και στην προηγούμενη περίπτωση σε σύνολα εκπαίδευσης (training), επικύρωσης (validation) και ελέγχου (checking). Πιο συγκεκριμένα χρησιμοποιείται το 60% των δειγμάτων για το σύνολο εκπαίδευσης ενώ από 20% των δειγμάτων σε κάθε ένα από τα εναπομείναντα υποσύνολα. Το μέγεθος όμως του dataset καθιστά απαγορευτική μια απλή εφαρμογή ενός TSK μοντέλου όπως στο πρώτο μέρος. Για τον σκοπό αυτό χρησιμοποιούνται μέθοδοι για την μείωση των χαρακτηριστικών (feature selection) αλλά και ομαδοποίησης δεδομένων (subtractive clustering) ενώ ταυτόχρονα χρησιμοποιούνται και τεχνικές βελτιστοποίησης για την βελτιστοποίηση των παραμέτρων που μας επιφέρουν οι 2 παραπάνω μέθοδοι. Όλες αυτές οι μέθοδοι εφαρμόζονται στα δεδομένα εκπαίδευσης καθώς αυτά είναι τα μόνα τα οποία θέλουμε να δώσουν πληροφορίες στην δημιουργία του μοντέλου μας.

Αρχικά για την μείωση των χαρακτηριστικών χρειαζόμαστε μια μέθοδο αξιολόγησης αυτών με σκοπό μια καλύτερη επιλογή χαρακτηριστικών. Η μέθοδος η οποία χρησιμοποιείται είναι ο αλγόριθμος relief. Ο αλγόριθμος αυτός αρχικά θέτει όλα τα βάρη των χαρακτηριστικών στο 0. Στη συνέχεια επαναληπτικά επιλέγει μια τυχαία παρατήρηση (τυχαίο δείγμα) των δεδομένων και βρίσκει τις k κοντινότερες παρατηρήσεις. Έτσι σε κάθε επανάληψη «τιμωρεί» τα χαρακτηριστικά που έχουν διαφορετικές τιμές σε ίδιες τιμές απόκρισης ενώ «επιβραβεύει» τα χαρακτηριστικά που έχουν διαφορετικές τιμές σε διαφορετικές τιμές της απόκρισης. Τελικά μετά από έναν αριθμό επαναλήψεων και έχοντας τα τελικά βάρη των χαρακτηριστικών τα κατατάσσει με βάση αυτά. Η επιλογή του k όμως δεν είναι πάντοτε απλή υπόθεση. Μια μικρή τιμή του k μπορεί να δώσει αναξιόπιστα αποτελέσματα σε θορυβώδες δεδομένα (noisy data) ενώ με μια τιμή του k ανάλογη με τον αριθμό των δειγμάτων μπορεί η relief να αποτύχει να βρεί τα σημαντικά χαρακτηριστικά.

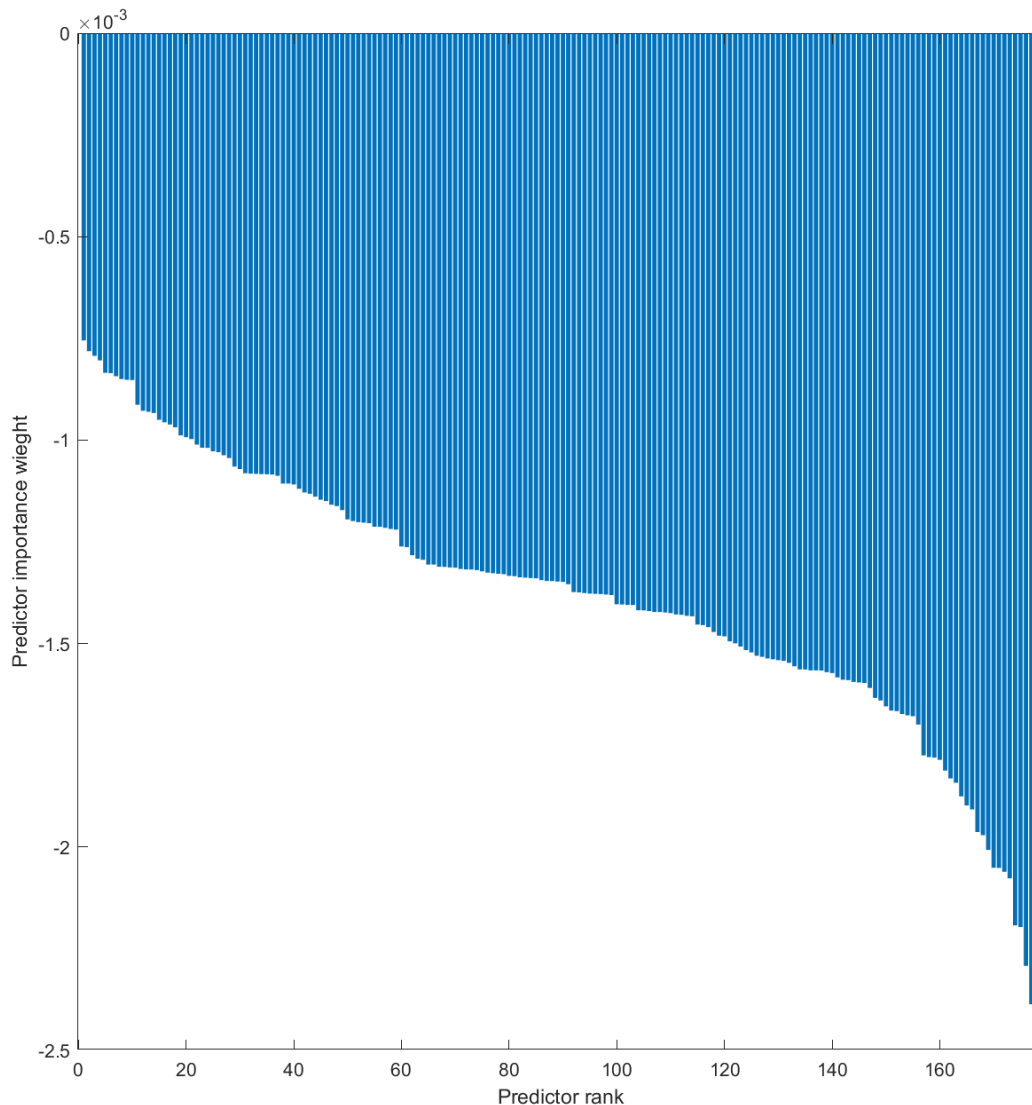
Έτσι για να βρούμε μια καλή τιμή του k φτιάξαμε τις γραφικές παραστάσεις των βαρών των χαρακτηριστικών συναρτήσει του k .



Διάγραμμα 19 - Μεταβολή βαρών των χαρακτηριστικών για διάφορες τιμές του k

Όπως παρατηρούμε μετά περίπου την τιμή $k = 120$ τα βάρη δεν αλλάζουν άλλο. Αυτό σημαίνει πως σε κάθε επανάληψη του αλγόριθμου relief, παίρνοντας υπόψη περισσότερες από 120 κοντινότερες παρατηρήσεις δεν επιφέρει αλλαγή στα βάρη των χαρακτηριστικών. Επιπλέον το 120 δεν είναι τιμή συγκρίσιμη με το μέγεθος των δεδομένων να ανησυχούμε για το φαινόμενο που αναφέρθηκε παραπάνω. Κατα συνέπεια το 120 είναι μια καλή τιμή του k και αυτή η τιμή χρησιμοποιήθηκε για την εύρεση των βέλτιστων χαρακτηριστικών με τον αλγόριθμο relief.

Έχοντας κατατάξει λοιπόν τα χαρακτηριστικά με βάση την σημαντικότητά τους αυτό που μας μένει είναι να βρούμε πόσα χαρακτηριστικά θα επιλέξουμε. Το διάγραμμα των βαρών των χαρακτηριστικών για τιμή $k = 120$ φαίνεται παρακάτω.

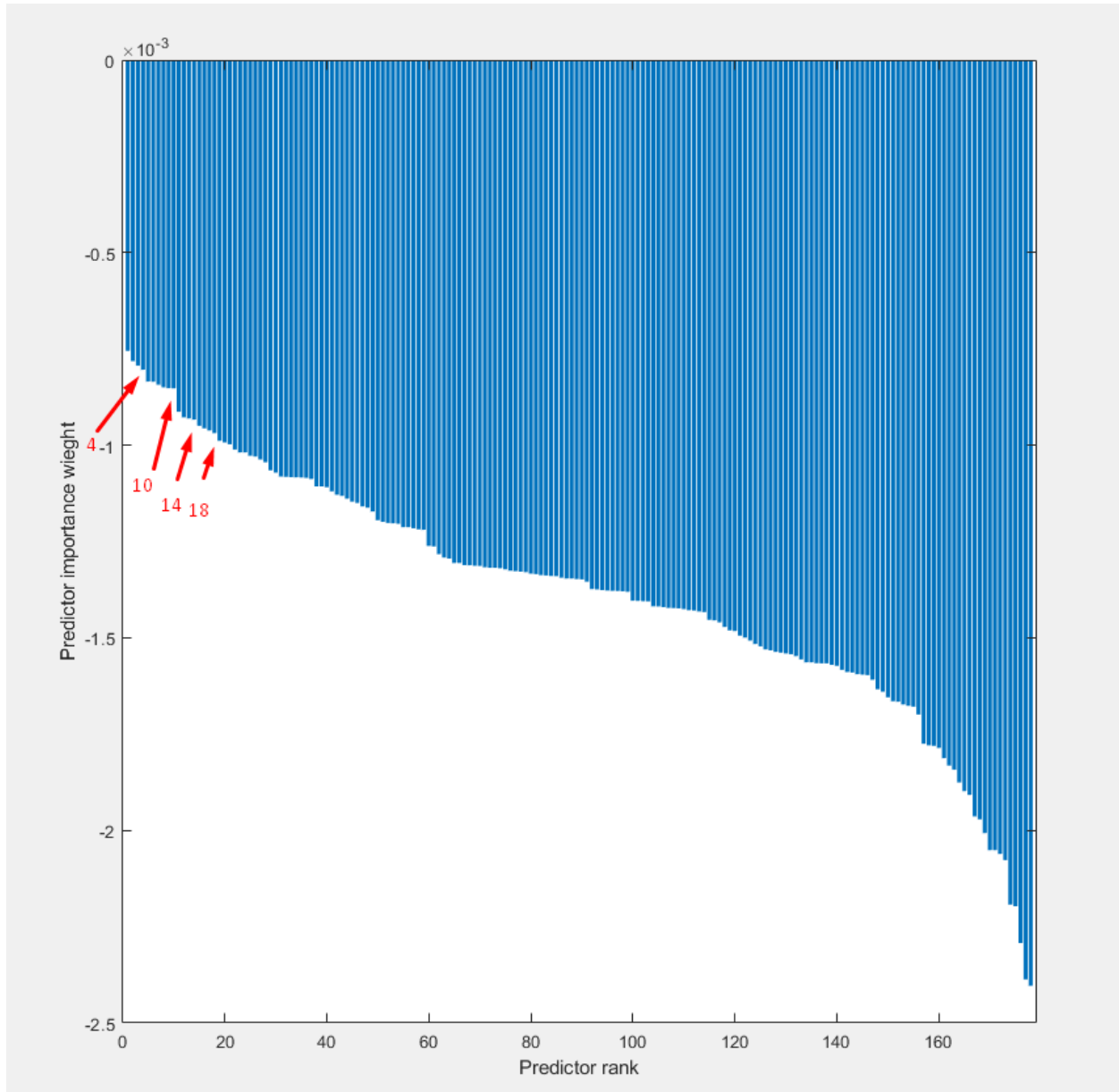


Διάγραμμα 20 - Βάρη των χαρακτηριστικών για $k = 120$

Όπως παρατηρούμε σε κάποια σημεία οι διαφορές των βαρών των χαρακτηριστικών είναι πολύ μικρή. Πολύ μικρή διαφορά βάρους μεταξύ δύο χαρακτηριστικών σημαίνει ίδια σημαντικότητα των 2 χαρακτηριστικών. Για τον λόγο αυτό εάν επιλέξουμε ένα από τα δυο χαρακτηριστικά των οποίων τα βάρη διαφέρουν λίγο μεταξύ τους αλλά δεν επιλέξουμε και το άλλο τότε θα υπήρχε σύγκρουση προς την επιλογή αυτή καθώς θα ήταν τυχαία (τυχαία επιλογή μεταξύ των δυο αυτών χαρακτηριστικών στο τελικό σύνολο των χαρακτηριστικών). Έτσι για μια πιο συστηματική

προσέγγιση επιλέγουμε έναν αριθμό χαρακτηριστικών για τα οποία τα βάρη είναι πάνω από μια συγκεκριμένη τιμή. Αυτό σημαίνει πως στο παραπάνω σχήμα επιλέγουμε χαρακτηριστικά ώσπου να επιλέξουμε μια «μεγάλη» αλλαγή μεταξύ ενός χαρακτηριστικού και του επόμενου.

Ενδεικτικά η διαδικασία αυτή φαίνεται στο παρακάτω σχήμα:



Διάγραμμα 21 - Ενδεικτικές τιμές για επιλογή αριθμού χαρακτηριστικών

Έτσι ο αριθμός των χαρακτηριστικών που θα επιλέξουμε θα είναι μια από αυτές τις τιμές. Για την ακρίβεια στο πρόβλημα αυτό χρησιμοποιήθηκαν οι τιμές 4, 6, 10, 14. Εδώ πέρα να σημειωθεί ότι η χρήση περισσότερων χαρακτηριστικών οδηγούσε σε χρονοβόρες διαδικασίες και γι' αυτό αποφεύχθηκε.

Η άλλη μέθοδος η οποία χρησιμοποιήθηκε για την απλούστευση του προβλήματος είναι όπως αναφέρθηκε παραπάνω η ομαδοποίηση των δεδομένων η οποία έγινε μέσω του αλγόριθμου subtractive clustering. Η μέθοδος αυτή έχει σαν κύρια ελεύθερη παράμετρο την ακτίνα των ομάδων *randii*. Ο αλγόριθμος αυτός περιγράφεται πολύ συνοπτικά παρακάτω. Αρχικά όλα τα δεδομένα κατατάσσονται στον πολυδιάστατο χώρο. Στη συνέχεια βρίσκει το βέλτιστο κέντρο μιας ομάδας (ενός cluster) το οποίο είναι ένα σημείο από το σύνολο των δεδομένων. Για τον σκοπό αυτό χρησιμοποιεί έναν δείκτη (potential) για να αξιολογήσει το κάθε σημείο του συνόλου δεδομένων. Σημεία που έχουν περισσότερα σημεία κοντά τους έχουν μεγαλύτερο potential. Τα κοντινά αυτά σημεία που βρίσκονται εντός της ακτίνας *randii* επηρεάζουν πολύ το potential ενώ τα σημεία εκτός της ακτίνας αυτής το επηρεάζουν λίγο. Επίσης σημεία που βρίσκονται μακριά από άλλα κέντρα άλλων ομάδων έχουν μεγαλύτερο potential έτσι ώστε να αποφυγεί η αλληλοεπικάλυψη των cluster. Τελικά το σημείο με το μεγαλύτερο potential επιλέγεται ως κέντρο του cluster αυτού και το cluster αυτό έχει ακτίνα *randii*. Η διαδικασία αυτή συνεχίζεται ώσπου όλα τα δεδομένα να καταταχθούν σε μια ομάδα. Έτσι τελικά τα δεδομένα ομαδοποιούνται και το πρόβλημα απλουστεύεται καθώς καταλήγουμε σε λιγότερους κανόνες στο νέο ασαφή σύστημα που προκύπτει (1 κανόνας για κάθε cluster).

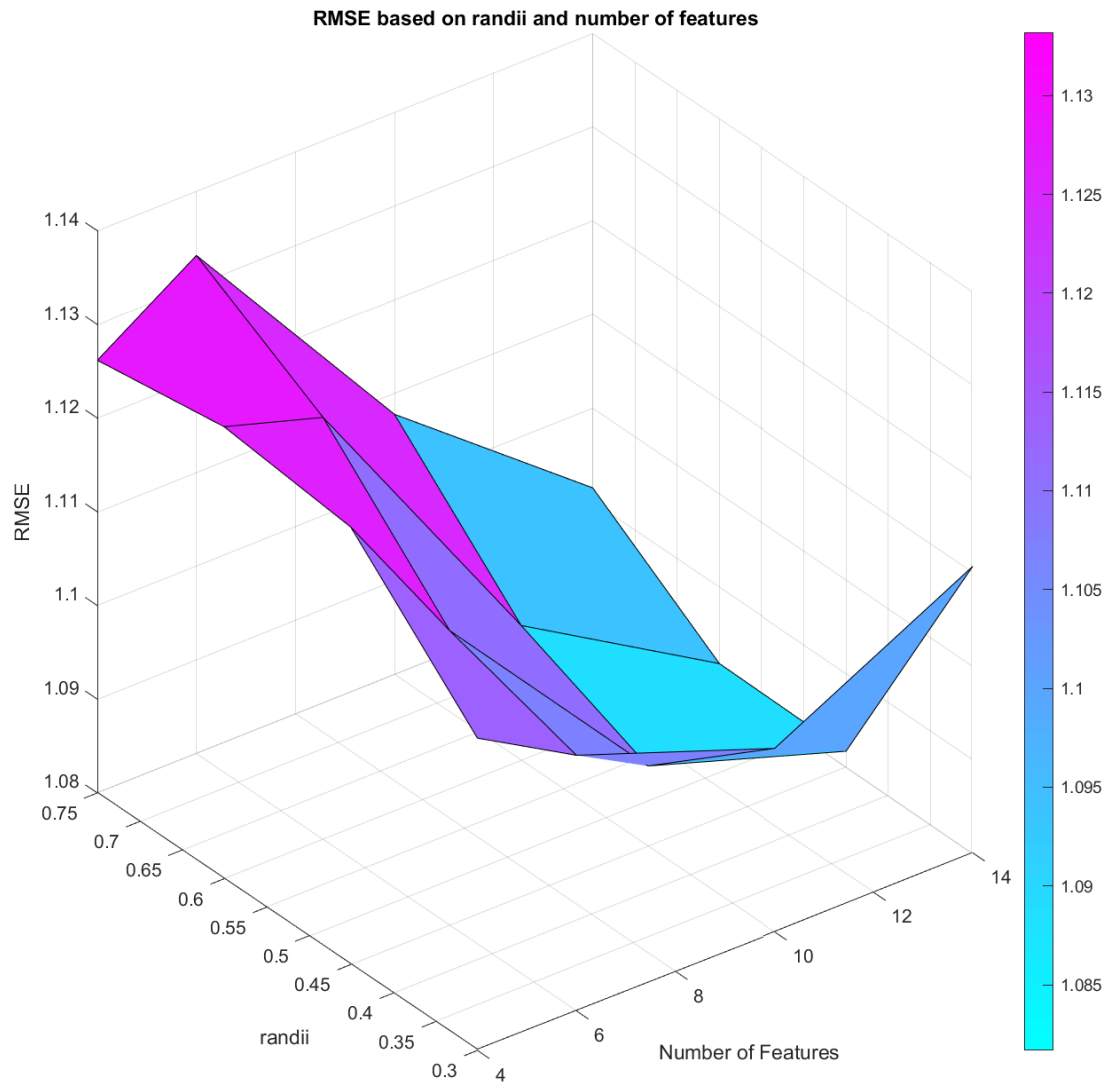
Στο πρόβλημα αυτό η επιλογή του *randii* έγινε από το σύνολο τιμών 0.30, 0.45, 0.60, 0.75.

Αφού λοιπόν έχοντας ορίσει τα σύνολα τιμών των δύο ελεύθερων παραμέτρων του προβλήματος, χρησιμοποιείται η αναζήτηση πλέγματος (grid search) για την αξιολόγηση των τιμών αυτών. Σύμφωνα με την τεχνική αυτή δημιουργούμε ένα διδιάστατο πλέγμα με τις τιμές των δύο ελεύθερων παραμέτρων μας. Κάθε σημείο του πλέγματος αυτού αντιστοιχεί σε μια επιλογή των δυο αυτών παραμέτρων. Για την αξιολόγηση του σημείου αυτού χρησιμοποιήθηκε η μέθοδος της διασταυρωμένης επικύρωσης (cross validation). Για την ακρίβεια χρησιμοποιήθηκε 5-fold cross validation. Σύμφωνα με τη μέθοδο αυτή το σύνολο εκπαίδευσης χωρίζεται στα 5. Το ένα κομμάτι αυτών των 5 συνόλων το οποίο αποτελείται από το 20% των δεδομένων εκπαίδευσης θα χρησιμοποιηθεί για επικύρωση ενώ τα υπόλοιπα κομμάτια (80%) για το σύνολο εκπαίδευσης του καινούριου συστήματος. Επειδή όμως θέλουμε όσο πιο γενικά αποτελέσματα γίνεται, αντί να χρησιμοποιήσουμε τυχαία ένα κομμάτι για το σύνολο επικύρωσης, χρησιμοποιούμε μια φορά όλα τα κομμάτια (ενώ το υπόλοιπο 80% πάντα λαμβάνεται ως το σύνολο εκπαίδευσης) και στο τέλος παίρνουμε τον μέσο όρο των 5 αυτών περιπτώσεων.

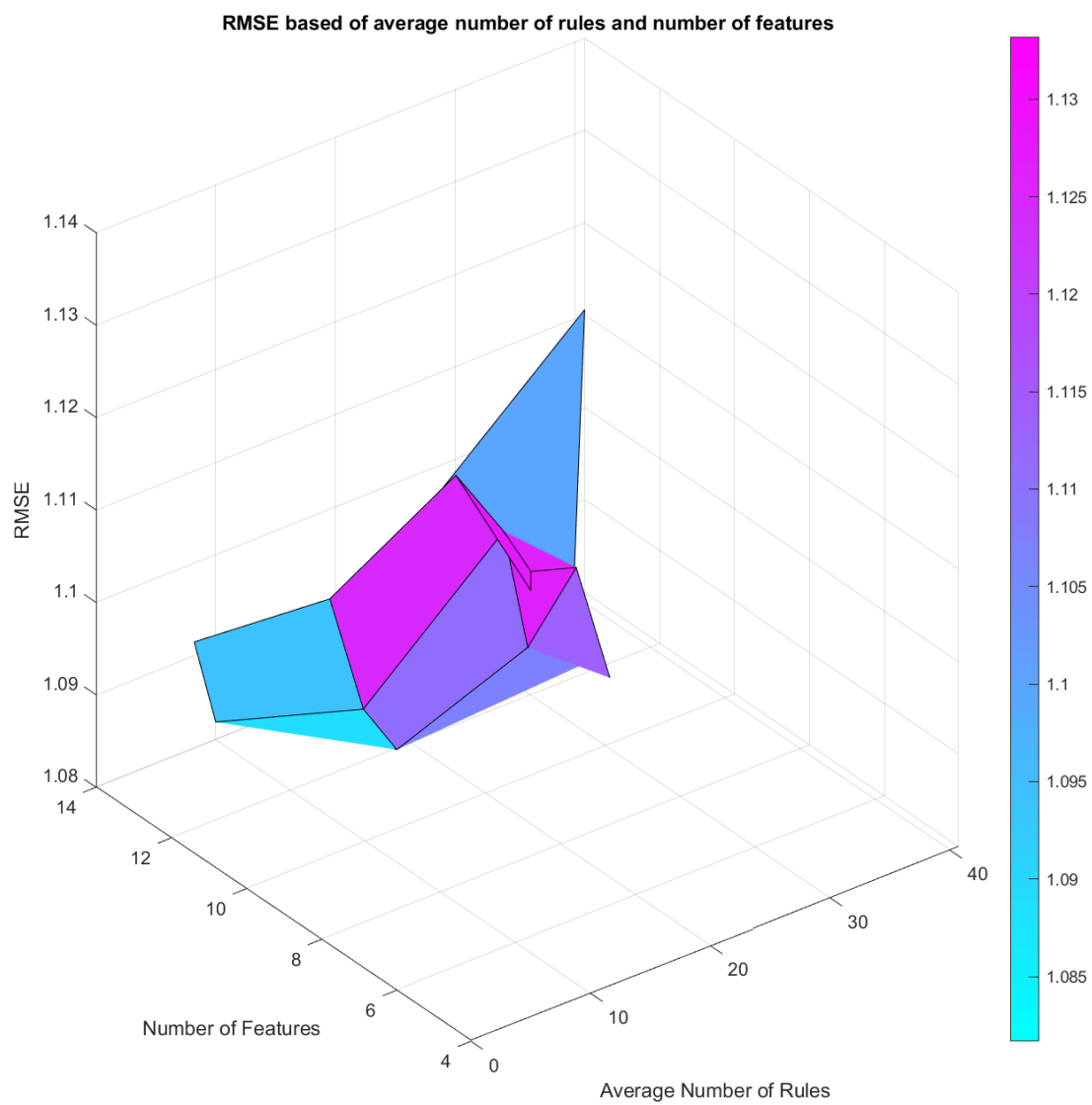
Έτσι σε κάθε μια περίπτωση αφού εκπαιδεύσουμε το μοντέλο και βρούμε το σφάλμα του, το κρατάμε και βρίσκουμε το μέσο όρο του. Αυτό γίνεται για κάθε σημείο του πλέγματος που έχουμε δημιουργήσει και τέλος επιλέγουμε την δυάδα παραμέτρων η οποία επιφέρει το μικρότερο μέσο σφάλμα.

Τα αποτελέσματα της διαδικασίας αυτής φαίνονται παρακάτω:

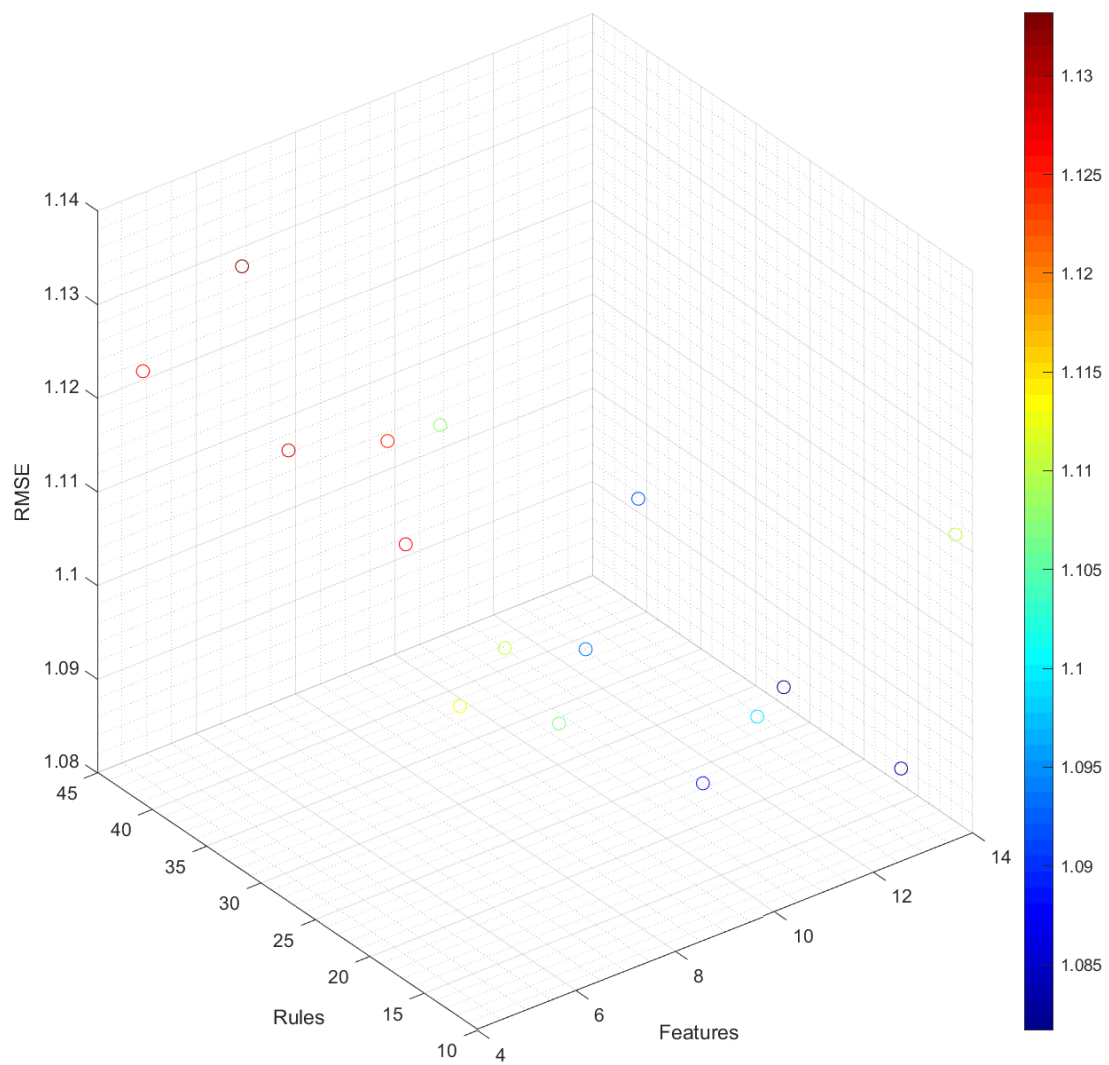
Στο πρώτο διάγραμμα παρουσιάζεται το μέσο σφάλμα RMSE συναρτήσει των 2 παραμέτρων ενώ στο δεύτερο διάγραμμα παρουσιάζεται το μέσο σφάλμα RMSE συναρτήσει του μέσου αριθμού των κανόνων και του αριθμού των επιλεγθέντων χαρακτηριστικών. Τέλος στο τρίτο διάγραμμα φαίνεται μια άλλη οπτική του δεύτερου διαγράμματος:



Διάγραμμα 22 - RMSE συναρτήσει του αριθμού των χαρακτηριστικών και της ακτίνας των cluster



Διάγραμμα 23 - RMSE συναρτήσει του αριθμού των χαρακτηριστικών και του μέσου αριθμού των κανόνων (1)



Διάγραμμα 24 - RMSE συναρτήσει του αριθμού των χαρακτηριστικών και του μέσου αριθμού των κανόνων (2)

Παρατηρούμε όπως ήταν αναμενόμενο πως κατά κανόνα όσο αυξάνεται ο αριθμός των χαρακτηριστικών αλλά και όσο μειώνεται η ακτίνα των clusters randii (και παράλληλα ο αριθμός των κανόνων) τόσο μειώνεται και το σφάλμα. Η βέλτιστη λύση βρέθηκε για αριθμό χαρακτηριστικών ίσο με 14 και ακτίνα randii ίση με 0.45

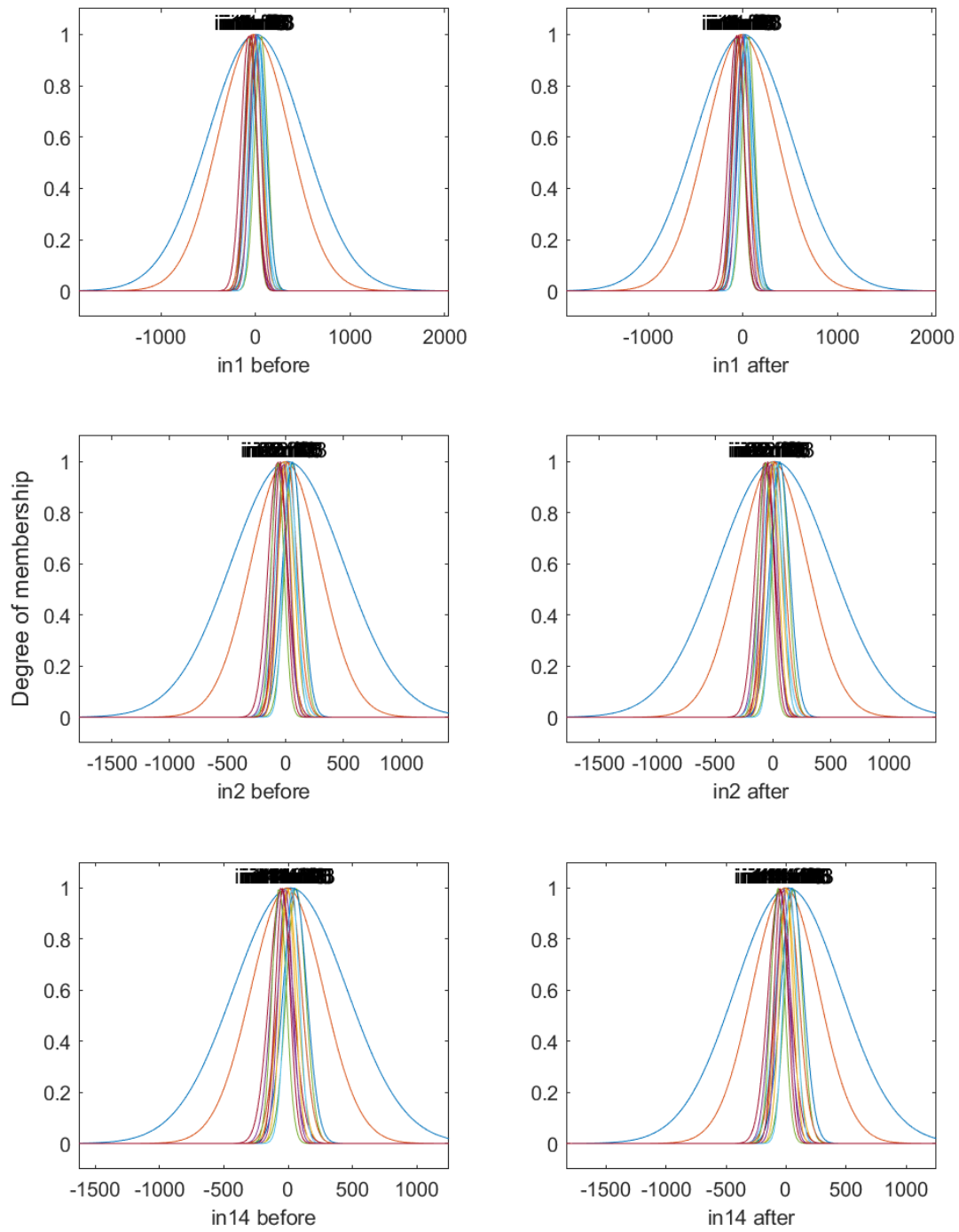
Αρχεία Matlab

- **main_part_2.m**: Περιέχει το πρόγραμμα το οποίο με την χρήση των βοηθητικών συναρτήσεων εφαρμόζει τη διαδικασία που περιγράφηκε παραπάνω και τελικά εκπαιδεύει και αξιολογεί το τελικό TSK μοντέλο.
- **plotErrors.m**: Παράγει τις καμπύλες εκμάθησης στις οποίες φαίνεται η μείωση του σφάλματος σε σχέση με τις εποχές.
- **plotMFs.m**: Παράγει τα διαγράμματα των συναρτήσεων συμμετοχής των εισόδων ενός TSK μοντέλου.
- **plotPrediction.m**: Παράγει τα διαγράμματα στα οποία απεικονίζονται τα σφάλματα πρόβλεψης.
- **split_scale.m**: Χωρίζει το σύνολο δεδομένων σε σύνολα εκπαίδευσης-επικύρωσης-ελέγχου ενώ ταυτόχρονα κανονικοποιεί τις μεταβλητές.
- **Generate_FIS.m**: Δημιουργεί ένα Fuzzy Interference System (FIS) με τη μέθοδο του Subtractive Clustering είτε για class dependent μοντέλο είτε για class independent.
- **Generate_Fis_Explanation.m**: Εξηγεί βήμα βήμα τον κώδικα του **Generate_FIS.m**.
- **GetClasses.m**: Επιστρέφει τις κλάσεις του συνόλου δεδομένων.
- **GetSameFrequencySamples.m**: Επιστρέφει training, validation και checking data τα οποία να έχουν μέγιστη διαφορά συχνότητας εμφάνισης δειγμάτων της κάθε κλάσης κάτω από ένα κατώφλι που δέχεται ως input.
- **Plot_RMSE_surf.m**: Παράγει τα διαγράμματα τα οποία απεικονίζουν τα αποτελέσματα της αναζήτησης πλέγματος με τη χρήση της συνάρτησης surf.
- **Plot_RMSE_scatter3.m**: Παράγει τα διαγράμματα τα οποία απεικονίζουν τα αποτελέσματα της αναζήτησης πλέγματος με τη χρήση της συνάρτησης scatter3.
- **Best_K.m**: Υπολογίζει τα βάρη των χαρακτηριστικών με τη συνάρτηση relief συναρτήσει του k και παράγει τα ανάλογα διαγράμματα χρησιμοποιώντας την **plots_K.m**.
- **MF_before_after.m**: Παράγει ένα διάγραμμα στο οποίο απεικονίζονται οι αρχικές αλλά και οι τελικές μορφές των ασαφών συνόλων μερικών χαρακτηριστικών.

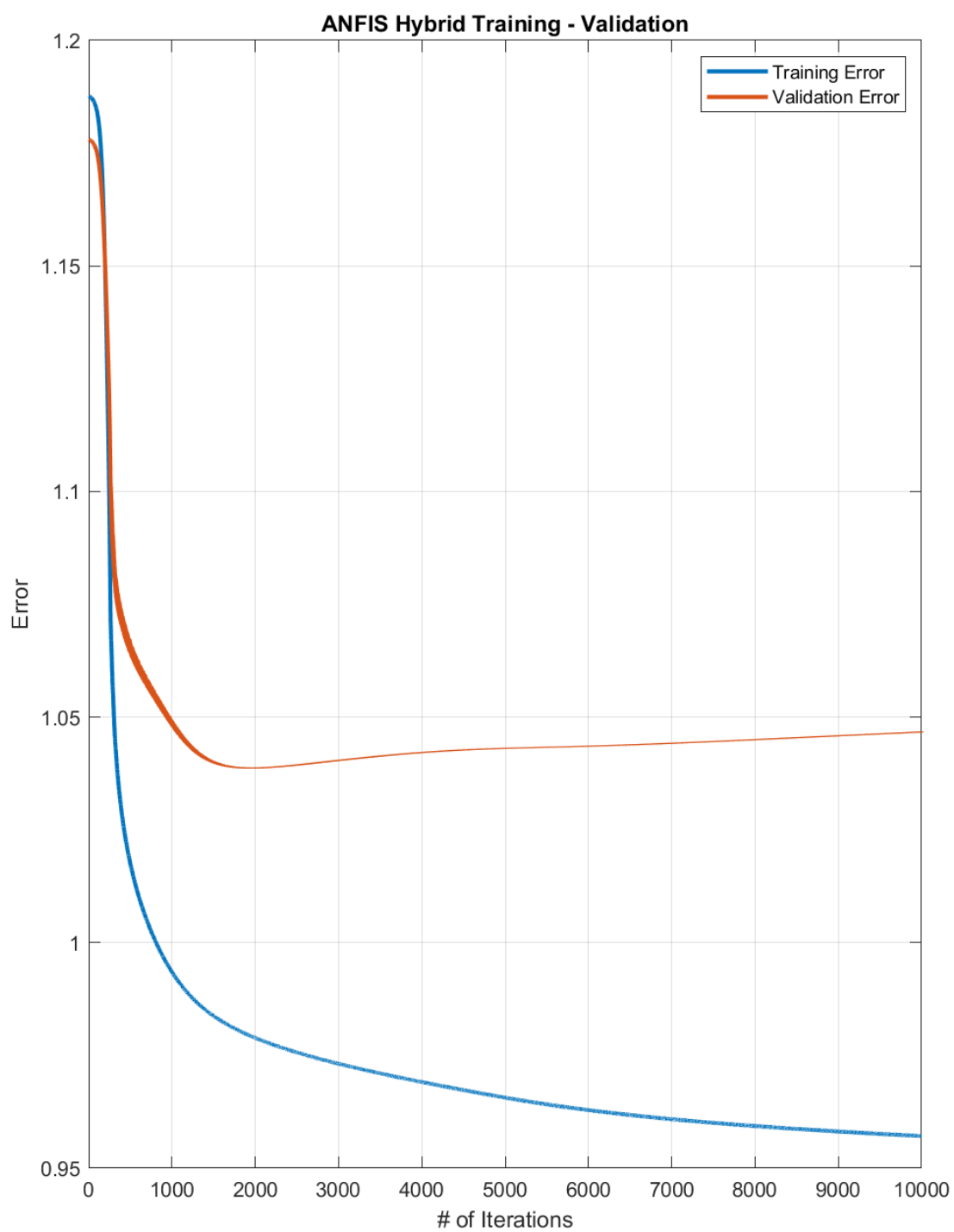
Διαγράμματα

Όπως και στο πρώτο μέρος παίρνουμε τα παρακάτω διαγράμματα τα οποία για ευκολία περιγράφονται ξανά:

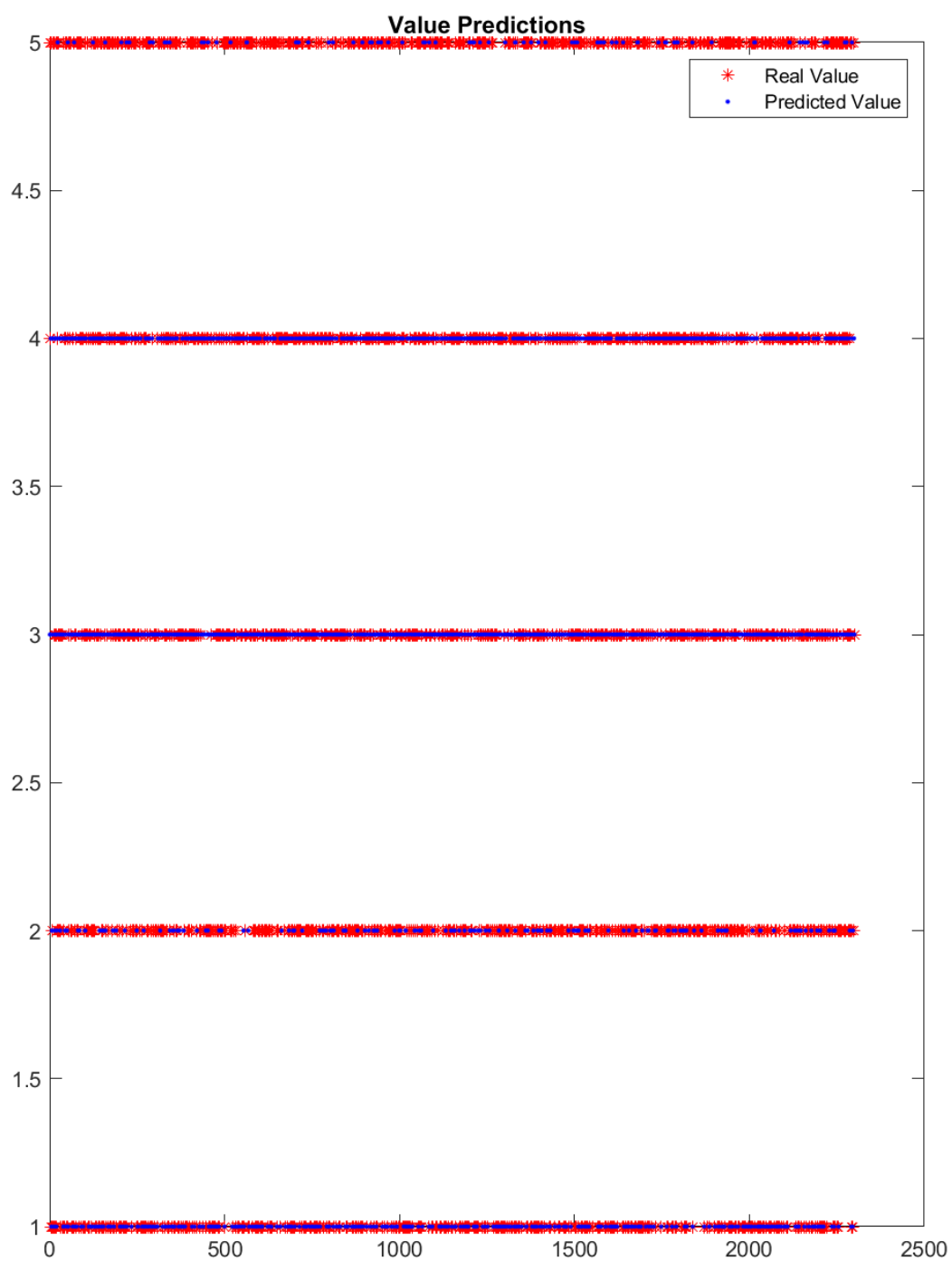
- i) Ασαφή σύνολα – Ενδεικτικά διαγράμματα στα οποία απεικονίζονται οι αρχικές αλλά και οι τελικές μορφές των ασαφών συνόλων που προέκυψαν μέσω της διαδικασίας εκπαίδευσης:
- ii) Διάγραμμα εκμάθησης – Φαίνονται οι καμπύλες εκμάθησης στις οποίες φαίνεται η μείωση του σφάλματος RMSE σε σχέση με τις εποχές εκπαίδευσης.
- iii) Διάγραμμα πρόβλεψης - Εδώ με μπλέ χρώμα βλέπουμε τις τιμές πρόβλεψης ενώ με κόκκινο χρώμα τις πραγματικές τιμές. Στον άξονα y έχουμε την τιμή της εξόδου ενώ στον άξονα x είναι το index του δείγματος που εξετάζουμε.



Διάγραμμα 25 - Ενδεικτικά ασαφή σύνολα πριν και μετά την εκπαίδευση



Διάγραμμα 26 - Διάγραμμα εκμάθησης μοντέλου μεγάλης διασασμότητας



Διάγραμμα 27 - Διάγραμμα πρόβλεψης μοντέλου μεγάλης διαστασιμότητας

		<i>Real Values</i>				
		1	2	3	4	5
<i>Predicted Values</i>	1	333	19	1	2	1
	2	68	29	18	20	7
	3	48	331	328	184	157
	4	11	79	109	219	252
	5	0	2	4	35	43

<i>OA</i>		0.4139
<i>K</i>		0.2674
<i>User's Accuracy</i>	1	0.9354
	2	0.2042
	3	0.3130
	4	0.3269
	5	0.5119
<i>Producer's Accuracy</i>	1	0.7239
	2	0.0630
	3	0.7130
	4	0.4761
	5	0.0935

Παρατηρήσεις

Όπως αναφέρθηκε και προηγουμένως το αποτέλεσμα που προέκυψε από το grid search ως την περίπτωση που επιφέρει τα καλύτερα αποτελέσματα είναι ο συνδιασμός

- Αριθμός χαρακτηριστικών : 14
- Randii : 0.45

Αυτό έχει ως συνέπεια τη δημιουργία 14 cluster και κατά συνέπεια την ύπαρξη 14 συνολικών κανόνων. Εδώ να αναφερθεί πως εάν για τον ίδιο αριθμό χαρακτηριστικών είχαμε επιλέξει grid partitioning έναντι του subtractive clustering, τότε ακόμη και με την πιο απλή περίπτωση της διαμέρισης του χώρου εισόδου σε 2 μόνο ασαφή σύνολα για κάθε μεταβλητή, το ασαφές σύστημα συμπερασμού θα είχε σύνολο $2^{14} = 16384$ κανόνες. Δηλαδή περίπου 1170 φορές περισσότερους από τώρα!

Τα σφάλματα που παρουσιάζονται στον παραπάνω πίνακα φαίνονται αρκετά μεγάλα αλλά είναι αποδεκτά δεδομένου την επιλογή μόνο των 14 από τα 179 συνολικά χαρακτηριστικά αλλά και τη δραστική μείωση των κανόνων που μας επέφερε η ομαδοποίηση των δεδομένων με τη χρήση του subtractive clustering.

Τέλος να παρατηρηθεί η επικάλυψη των ασαφών συνόλων όπου κάποια ασαφή σύνολα μάλιστα επικαλύπτονται πλήρως από κάποια άλλα. Αυτό είναι άμεση συνέπεια των κέντρων και των τιμών sigma που προέκυψαν από το subtractive clustering στην κάθε κλάση ξεχωριστά και δεν αποτελεί σφάλμα αλλά είναι ο τρόπος που λειτουργεί το εν λόγω μοντέλο.