Ασαφή Συστήματα

Επίλυση προβλήματος ταξινόμησης

με μοντέλα TSK

Υπατία Δάμη

ΑΕΜ:8606

13/10/2019

Σκοπός εργασίας:

Σκοπός αυτής της εργασίας είναι η επίλυση προβλημάτων ταξινόμησης με χρήση διαφόρων TSK μοντέλων. Στα προβλήματα ταξινόμησης παρέχεται ένα σετ δεδομένων για την εκπαίδευση και την αξιολόγηση του μοντέλου, το οποίο περιέχει τιμές(εισόδους), για όλες τις μεταβλητές του προβλήματος, αλλά και την έξοδο του υπό μελέτη συστήματος για έναν ικανοποιητικό αριθμό εισόδων. Η έξοδος αποτελείται από διακριτές τιμές που αναπαριστούν τις κλάσεις του προβλήματος ,στις οποίες ταξινομούνται οι εγγραφές σύμφωνα με τις τιμές των χαρακτηριστικών του. Οι τιμές αυτές εξαρτώνται από τον αριθμό των κλάσεων στο εκάστοτε πρόβλημα. Σκοπός είναι στο τέλος της διαδικασίας αυτής το μοντέλο να είναι ικανό να μπορεί να ταξινομεί τις νέες εγγραφές που εισάγονται στο σύστημα στις ανάλογες κλάσεις ενώ του παρέχονται πλέον τιμές μόνο των μεταβλητών(εισόδων) του συστήματος. Το μοντέλο ουσιαστικά "εκπαιδεύεται" στο να αναγνωρίζει κάποια πρότυπα συμπεριφοράς των δεδομένων και σύμφωνα με αυτά να μπορεί να προβλέψει την έξοδο του συστήματος για οποιεσδήποτε τιμές εισόδων. Στην εργασία αυτή μελετώνται δύο προβλήματα παλινδρόμησης. Στο πρώτο πρόβλημα χρησιμοποιείται το avila dataset , που περιέχει 20876 δείγματα , με 10 χαρακτηριστικά το καθένα . Λόγω του μικρού αριθμού των χαρακτηριστικών του το dataset μας επιτρέπει να ακολουθήσουμε μια απλή διαδικασία μοντελοποίησης .Αντιθέτως, στο δεύτερο πρόβλημα της εργασίας καλούμαστε να χρησιμοποιήσουμε ένα πολύ πιο σύνθετο dataset, το isolet dataset που περιλαμβάνει 7797 δείγματα, καθένα από τα οποία περιγράφεται από 618 μεταβλητές. Το μεγάλο πλήθος χαρακτηριστικών δεν επιτρέπει την επίλυση του προβλήματος με απλή μοντελοποίηση και γιαυτό καλούμαστε να χρησιμοποιήσουμε διάφορες τεχνικές και αλγορίθμους για την απλοποίηση του. Αφού ολοκληρωθεί η εκπαίδευση και η αξιολόγηση των μοντέλων, συνθέτουμε διαγράμματα που να απεικονίζουν την απόδοση και τη συμπεριφορά του εκάστοτε μοντέλου και συγκρίνουμε την απόδοση τους .

Επίλυση προβλήματος με απλό dataset:

Στο πρώτο πρόβλημα καλούμαστε να μοντελοποιήσουμε ένα πρόβλημα χρησιμοποιώντας το avila dataset της UCI repository, που περιέχει 20876 δείγματα , με 10 χαρακτηριστικά το καθένα . Για την μοντελοποίηση του συστήματος αυτού θα δημιουργήσουμε 4 TSK μοντέλα με διαφορετικό αριθμό κανόνων. Επειδή θέλουμε να αντιμετωπίσουμε το πρόβλημα ταξινόμησης χρησιμοποιώντας την ασαφή λογική , θα δημιουργήσουμε για το κάθε μοντέλο ένα Fuzzy Inference System χρησιμοποιώντας την συνάρτηση " genfis " του matlab.Ως μέθοδος για τον διαμερισμό της εισόδου χρησιμοποιηθεί η μέθοδος Subtractive Clustering, κατά την οποία οι είσοδοι του μοντέλου ομαδοποιούνται σε clusters . Ο αριθμός των clusters προκύπτει από την ακτίνα επιρροής στον χώρο εισόδου που θα επιλέξουμε για το κάθε

cluster. Όσο μικρότερη η ακτίνα των cluster τόσα περισσότερα clusters δημιουργούνται άρα και περισσότεροι κανόνες. Τα μοντέλα που θα δημιουργηθούν θα διαφέρουν στον αριθμό κανόνων , και θα εξεταστεί πως ο αριθμός κανόνων στο καθένα επηρεάζει την ακρίβειά του. Επειδή δεν ήταν δυνατόν η τιμή της ακτίνας των clusters να πάρει τιμές κάτω από 0.2, δημιουργήθηκαν 4 μοντέλα με 2,3,4 και 5 κανόνες αντίστοιχα.

Υλοποίηση στο Matlab:

Για την υλοποίηση των μοντέλων στο Matlab δημιουργήθηκαν 4 scripts, ένα για το κάθε μοντέλο: TSK_1.m, TSK_2.m, TSK_3.m, TSK_4.m. Η διαδικασία που ακολουθήθηκε στο κάθε script είναι η ίδια ,αλλάζοντας τις τιμές των παραμέτρων που θέλουμε να εξετάσουμε στο εκάστοτε μοντέλο (ακτίνα cluster) ,επομένως η περιγραφή αναφέρεται και στα 4 script.

Διαχωρισμός δεδομένων:

Αρχικά φορτώνεται στο Workspace το σετ δεδομένων και έπειτα μεταφέρεται σε έναν πίνακα για να υποστεί επεξεργασία. Επιθυμούμε να χωρίσουμε το σύνολο των δεδομένων χωρίζεται έπειτα σε 3 πίνακες , 60% training _data, 20% evaluation_data και 20% testing_data. Είναι σημαντικό πριν γίνει αυτό να βεβαιωθούμε πως το κάθε σετ θα περιέχει ίση συχνότητα εγγραφών που ανήκουν σε μία κλάση για την καλύτερη εκπαίδευση του μοντέλου. (Αν το validation set περιέχει μια τιμή κλάσης που δεν περιέχεται στο training set δεν θα την γνωρίζει και θα ταξινομήσει λανθασμένα την εγγραφή.)Επομένως χωρίζουμε τα δεδομένα σε 12 πίνακες , καθένας από τους οποίους περιέχει τις εγγραφές που ανήκουν στην ίδια κλάση .Η διαδικασία αυτή φαίνεται στον παρακάτω κώδικα:

```
val1 = data(data(:, end) == 0, :);

val2 = data(data(:, end) == 1, :);

.

.

val12 = data(data(:, end) == 12, :);
```

Έπειτα δημιουργούμε για κάθε κλάση τους κατάλληλους δείκτες που θα χωρίζουν κατάλληλα τον πίνακα με τις εγγραφές της κάθε κλάσης , δηλαδή σε 60% training, 20% evaluation και 20% testing. Η διαδικασία αυτή φαίνεται στον παρακάτω κώδικα.

```
first_split_one = round(0.6 * length(val1));
```

```
second_split_one = round(0.8 * length(val1));
```

Στη συνέχεια. Αποθηκεύουμε στον πίνακα training_data τις εγγραφές της κάθε κλάσης μεχρι τον δείκτη first_split_i ,(όπου i ο αριθμός της κλάσης) προστίθοντάς τες. Στον πίνακα validadion_data αποθηκεύουμε το άθροισμα εγγραφών κάθε κλάσης από τον δείκτη first_split_one έως τον δείκτη second_split_I . Στον πίνακα testing_data αποθηκεύουμε το άθροισμα εγγραφών κάθε κλάσης από τον δείκτη second_split_I έως το τέλος των πινάκων. Η διαδικασία αυτή φαίνεται στον παρακάτω κώδικα για τον πίνακα training_data:

```
training_data = [val1(1:first_split_one, :); val2(1:first_split_two, :); val3(1:first_split_three,:) ;val4(1:first_split_four, :); val5(1:first_split_five, :); val6(1:first_split_six, :);val7(1:first_split_seven, :); val8(1:first_split_eight, :); val9(1:first_split_nine,:);val10(1:first_split_ten, :); val11(1:first_split_eleven, :); val12(1:first_split_twelve,:)];
```

Τέλος , ανακατεύουμε χωριστά και τα 3 σετ μας, για ακόμα καλύτερη απόδοση του μοντέλου μας. Ενδεικτικά φαίνεται στον παρακάτω κώδικα η διαδικασία για τον πίνακα training_data:

```
rand_pos = randperm(length(training_data));

for k = 1 : length(training_data)

   shuffled_data(k, :) = training_data(rand_pos(k), :);

end

training_data = shuffled_data;
```

Τα δεδομένα του πίνακα training_data χρησιμοποιούνται για την εκπαίδευση του μοντέλου, του evaluation_data για την αξιολόγηση του κάθε κύκλου εκπαίδευσης και του testing_data για τον έλεγχο της εγκυρότητας του μοντέλου αφού ολοκληρωθεί η διαδικασία εκμάθησης. Τα δεδομένα του dataset είναι εξαρχής κανονικοποιημένα στο διάστημα [-1.1], επομένως δεν χρειάζονται περεταίρω κανονικοποίηση.

Εκπαίδευση του μοντέλου:

Αφού διαχωριστούν τα δεδομένα δημιουργούμε ένα αντικείμενο “genfis_opt” μέσω της συνάρτησης genfisOptions(), στο οποίο θέτουμε τα χαρακτηριστικά του fuzzy inference system .Παραδείγματος χάριν για το πρώτο μοντέλο θέτουμε την ακτίνα επιρροής των cluster 0.7 .Σε όλα τα μοντέλα θέτουμε ως μέθοδο διαμερισμού την Subtractive Clustering. Αφού κάνουμε τις απαραίτητες ρυθμίσεις δημιουργούμε το fuzzy inference system μέσω της συνάρτησης “genfis()”.Η συνάρτηση παίρνει ως ορίσματα τα δεδομένα εκπαίδευσης εισόδου, τα δεδομένα εκπαίδευσης εξόδου και τα χαρακτηριστικά

του συστήματος “genfis_opt”.Το σύστημα που δημιουργείται αναπαρίσταται από το αντικείμενο “initial_fis”.Έπειτα μέσω της συνάρτησης “plot()”, δημιουργούμε διαγράμματα των συναρτήσεων συμμετοχής για κάποια clusters , για να τις συγκρίνουμε στο τέλος με την τελική τους μορφή μετά την εκπαίδευση του μοντέλου. Στη συνέχεια δημιουργούμε ένα αντικείμενο “anfis_opt” μέσω της συνάρτησης “anfisOptions()“ και θέτουμε τα χαρακτηριστικά της εκπαίδευσης του μοντέλου. Συγκεκριμένα θέτουμε τον αριθμό των εποχών εκπαίδευσης “400” που είναι ένας ικανοποιητικός αριθμός για να εξάγουμε συμπεράσματα και να μην είναι χρονοβόρα η διαδικασία εκπαίδευσης. Θέτουμε ως μοντέλο εκμάθησης το αντικείμενο “initial_fis” και εισάγουμε τα δεδομένα αξιολόγησης. Το μοντέλο πλέον είναι έτοιμο για εκπαίδευση , η οποία ξεκινάει καλώντας την συνάρτηση “anfis()” με ορίσματα τα δεδομένα εκπαίδευσης και τις ρύθμισης που κάναμε προηγουμένως. Η συνάρτηση επιστρέφει για κάθε εποχή το σφάλμα εκμάθησης(trainError), το σφάλμα αξιολόγησης(chkError) , το μοντέλο με το μικρότερο σφάλμα αξιολόγησης (chkFIS) και το βήμα διαφόρισης των παραμέτρων εισόδου και εξόδου. Οι διαδικασίες αυτές φαίνονται στον παρακάτω κώδικα:

```
anfis_opt=anfisOptions('InitialFIS',initial_fis);

anfis_opt.ValidationData=evaluation_data;

anfis_opt.EpochNumber=400;

[train_fis,trainError,stepSize,chkFIS,chkError] = anfis(training_data,anfis_opt);
```

Το “trainError” είναι το σφάλμα εκπαίδευσης σε κάθε εποχή εκπαίδευσης .Το “chkError” είναι το σφάλμα αξιολόγησης είναι το σφάλμα που προκύπτει όταν στο τέλος της κάθε εποχής εκπαίδευσης ελέγξουμε την απόδοση του μοντέλου με τα δεδομένα “evaluation_data”. Το μοντέλο “chkFIS” είναι το μοντέλο που έδωσε το μικρότερο σφάλμα αξιολόγησης , σε μία από τις 400 εποχές , και είναι το μοντέλο που επιλέγεται από το σύστημα με τις ανάλογες παραμέτρους εισόδου και εξόδου. Για την αξιολόγηση του τελικού μοντέλου καλούμε τη συνάρτηση “evalfis()” ,με ορίσματα το τελικό βέλτιστο μοντέλο και τα δεδομένα ελέγχου , “testing_data” και έτσι προκύπτει η τελική έξοδος του συστήματος που αποθηκεύεται στο διάνυσμα “system_output”. Με τη συνάρτηση round() στρογγυλοποιούμε τυχούσες δεκαδικές τιμές στον κοντινότερο ακέραιο , διότι οι αριθμοί των κλάσεων αναπαρίστανται από αριθμούς εύρους 1-12.Όσες τιμές υπάρχουν κάτω από το 1 τις θέτουμε ίσες με 1 και όσες είναι πάνω από 12 ίσες με 12. Με τη συνάρτηση “Plot()” δημιουργούμε διαγράμματα με την τελική μορφή των συναρτήσεων συμμετοχής του κάθε Cluster , για να τις συγκρίνουμε με τις αρχικές, πριν την εκπαίδευση.

Υπολογισμός μετρικών συστήματος:

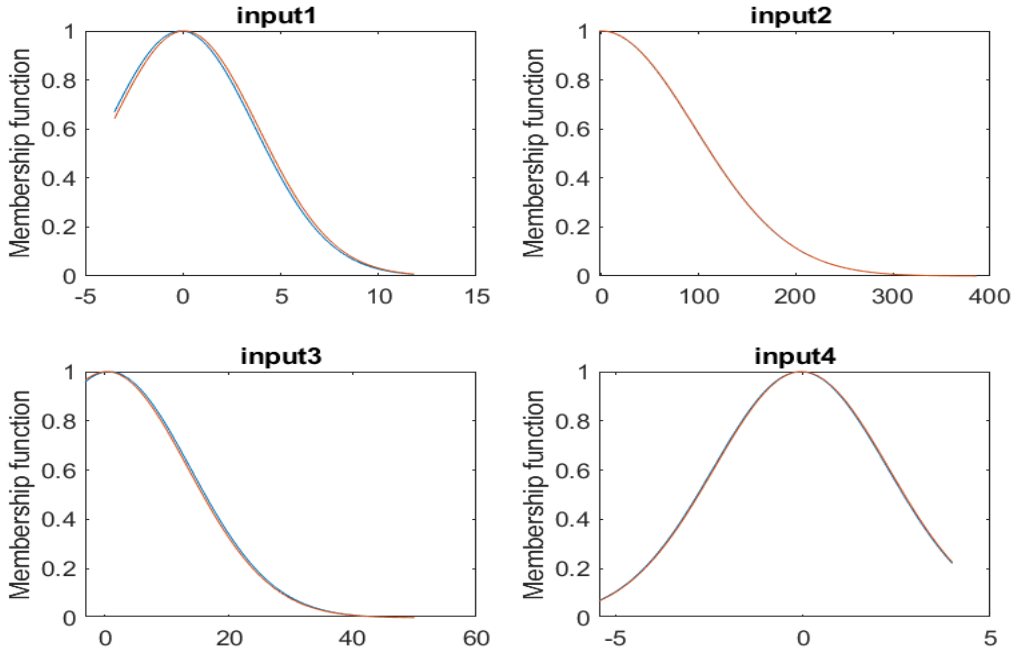
Τέλος , υπολογίζουμε ορισμένες χρήσιμες μετρικές για την αξιολόγηση του συστήματος οι οποίες είναι : ο πίνακας σφαλμάτων (error_matrix), η συνολική ακρίβεια (overal_accuracy) , και οι δείκτες producer’s accuracy(PA) και user’s accuracy(UA), και ο δείκτης K. Με τη συνάρτηση “ plot()” δημιουργούμε διαγράμματα που απεικονίζουν τα διάφορα σφάλματα, την αρχική και τελική έξοδο του συστήματος

Αποτελέσματα και σύγκριση μοντέλων :

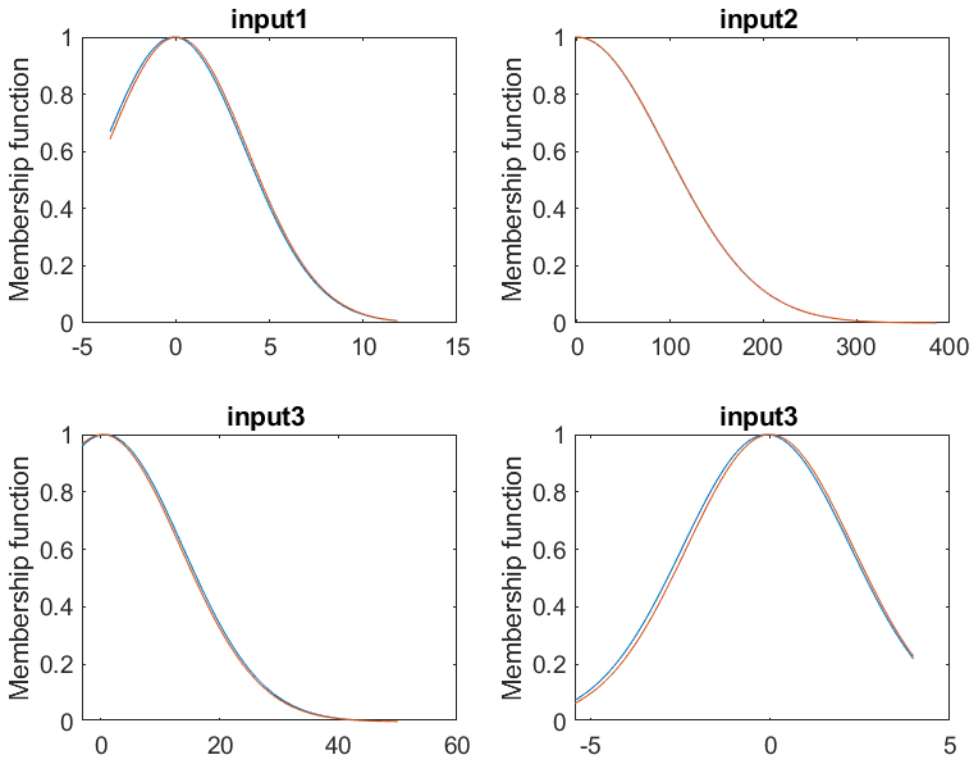
Στην ενότητα αυτή παρουσιάζονται τα διαγράμματα που προέκυψαν από την εκπαίδευση του κάθε μοντέλου , καθώς και σχολιασμός αυτών.

TSK_1 (2 rules):

Στην εικόνα 1 απεικονίζονται οι συναρτήσεις συμμετοχής για 4 από τα χαρακτηριστικά πριν από την εκπαίδευση του μοντέλου . Όπως φαίνεται ο διαμερισμός έχει γίνει με scatter partitioning με τα ασαφή σύνολα να επικαλύπτονται σχεδόν εξολοκλήρου. Αυτό συμβαίνει διότι πιθανότατα το κάθε χαρακτηριστικό κατανέμεται κυρίως σε ένα cluster . Στην εικόνα απεικονίζονται οι συναρτήσεις συμμετοχής των των χαρακτηριστικών μετά την εκπαίδευση του μοντέλου. Φαίνεται πως οι συναρτήσεις έχουν διαμορφωθεί , σύμφωνα με τις τελικές τιμές των παραμέτρων εισόδου , οι βέλτιστες των οποίων βρέθηκαν κατά τη διάρκεια της εκπαίδευσης του μοντέλου.

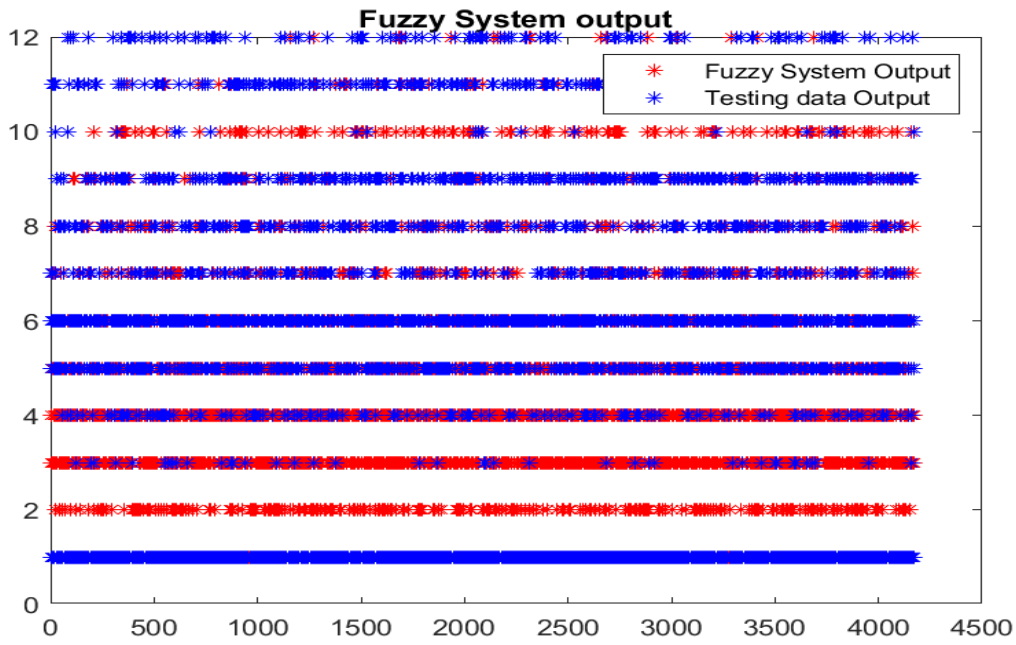


Εικόνα 1



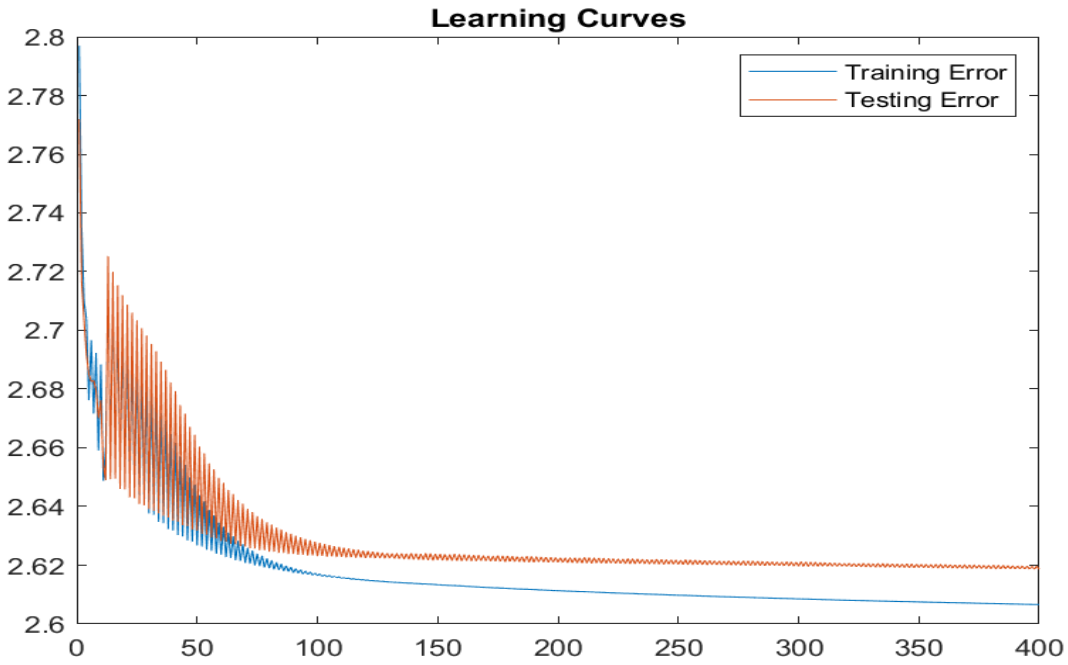
Εικόνα 2

Στην εικόνα 3 απεικονίζεται η έξοδος του ασαφούς μοντέλου (κόκκινο χρώμα) , σε σχέση με την έξοδο των δεδομένων δοκιμής . Φαίνεται πως το ασαφές μοντέλο προσεγγίζει τις τιμές της εξόδου ,κάνοντας και σφάλματα όπως να περιλαμβάνει στις προβλέψεις τιμές εντός του διαστήματος των κλάσεων, αλλά που δεν περιέχονται στην έξοδο του dataset , όπως η τιμή κλάσης '2'.



Εικόνα 3

Στην εικόνα 4 φαίνονται οι καμπύλες μάθησης του μοντέλου , με μπλε το σφάλμα εκπαίδευσης και με κόκκινο στο σφάλμα αξιολόγησης . Το σφάλμα αξιολόγησης είναι λογικά πιο μεγάλο από της εκπαίδευσης. Λόγω ότι έχουμε μόνο 2 κανόνες το σύστημά είναι πολύ γενικό , και έτσι το τελικό σφάλμα αξιολόγησης καταλήγει να είναι μικρότερο από τα υπόλοιπα μοντέλα.



Εικόνα 4

	1	2	3	4	5	6	7	8	9	10	11	12
1	117	215	566	560	207	33	5	7	3	0	1	0
2	0	0	0	0	0	0	0	0	0	0	0	0
3	2	2	3	16	10	5	2	0	0	0	0	1
4	1	5	36	53	34	8	3	1	0	0	0	0
5	12	8	29	87	156	112	29	3	2	0	0	0
6	16	50	206	331	158	15	5	0	3	0	0	1
7	0	6	28	44	77	20	3	1	0	0	0	0
8	11	6	5	34	78	58	13	0	0	2	1	0
9	1	2	4	4	15	24	27	54	91	76	29	6
10	0	0	3	6	5	3	1	0	0	0	0	0
11	1	1	2	6	10	44	50	36	19	20	9	11
12	0	1	1	3	7	6	20	18	23	19	6	3

Εικόνα 5: Error marrix

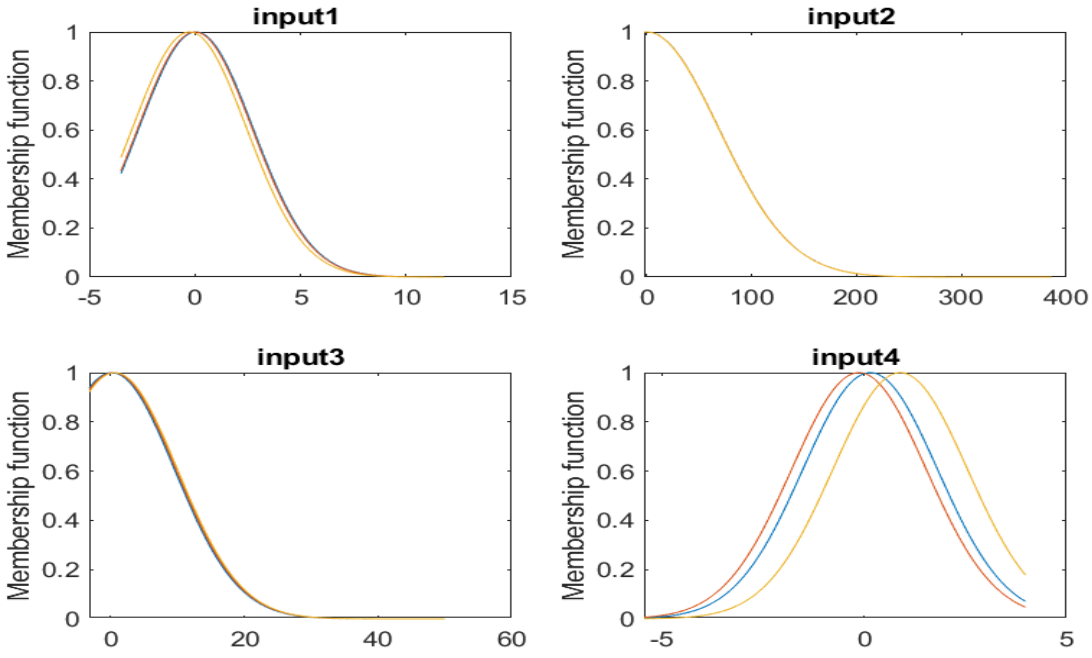
PA	0.068261	0	0.073171	0.375887	0.356164	0.019108	0.01676	0	0.273273	0	0.043062	0.028037

UA	0.726708	0	0.003398	0.046329	0.206077	0.045732	0.018987	0	0.64539	0	0.195652	0.136364

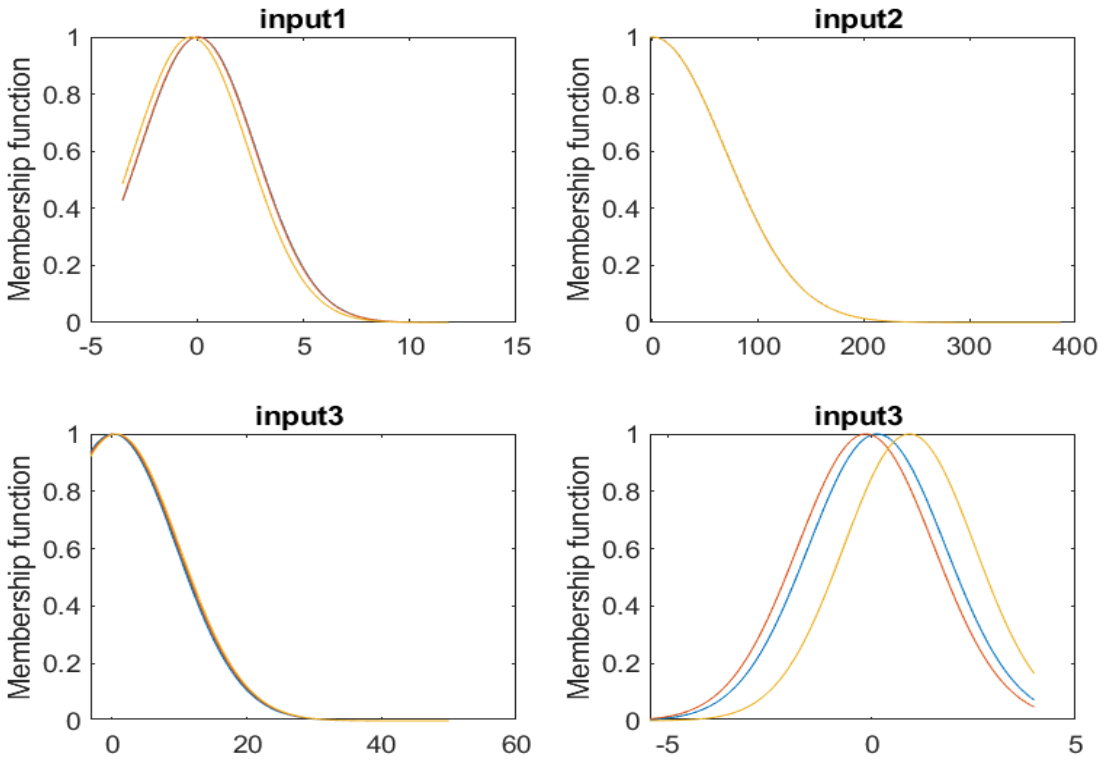
K=0.0432

TSK_2 (3 rules):

Στην εικόνα 7 απεικονίζονται οι συναρτήσεις συμμετοχής για 4 από τα χαρακτηριστικά πριν από την εκπαίδευση του μοντέλου . Όπως φαίνεται ο διαμερισμός έχει γίνει με scatter partitioning με τα ασαφή σύνολα να επικαλύπτονται σχεδόν εξολοκλήρου. Αυτό συμβαίνει διότι πιθανότατα το κάθε χαρακτηριστικό κατανέμεται κυρίως σε ένα cluster . Στην εικόνα 8 απεικονίζονται οι συναρτήσεις συμμετοχής των των χαρακτηριστικών μετά την εκπαίδευση του μοντέλου. Φαίνεται πως οι συναρτήσεις έχουν διαμορφωθεί , σύμφωνα με τις τελικές τιμές των παραμέτρων εισόδου , οι βέλτιστες των οποίων βρέθηκαν κατά τη διάρκεια της εκπαίδευσης του μοντέλου.

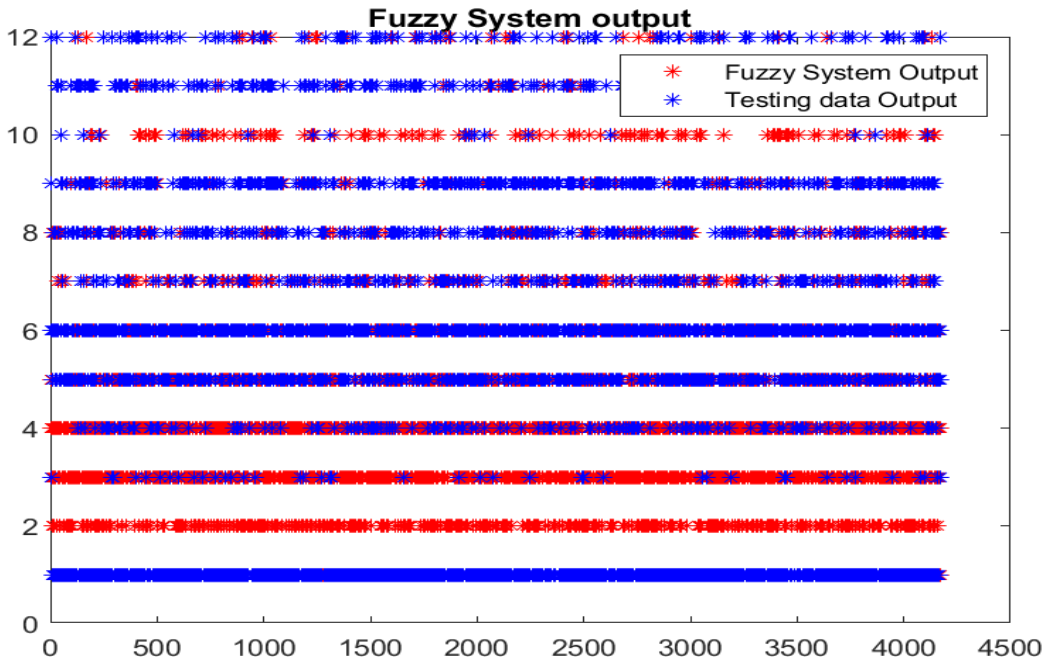


Εικόνα 7



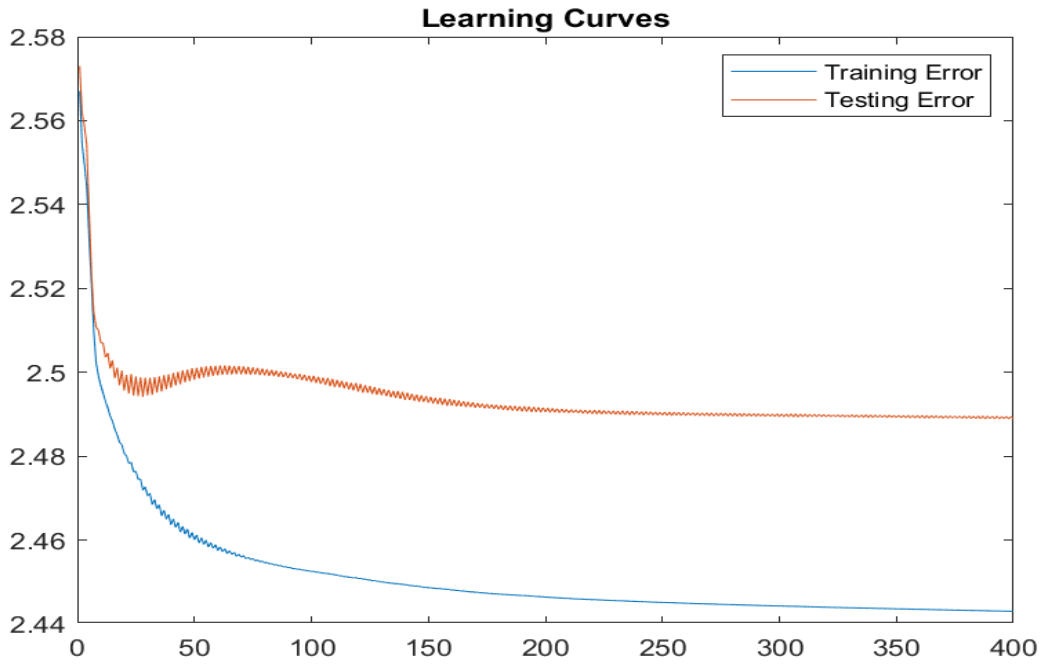
Εικόνα 8

Στην εικόνα 9 απεικονίζεται η έξοδος του ασαφούς μοντέλου (κόκκινο χρώμα) , σε σχέση με την έξοδο των δεδομένων δοκιμής . Φαίνεται πως το ασαφές μοντέλο προσεγγίζει τις τιμές της εξόδου ,κάνοντας και σφάλματα όπως να περιλαμβάνει στις προβλέψεις τιμές εντός του διαστήματος των κλάσεων, αλλά που δεν περιέχονται στην έξοδο του dataset , όπως η τιμή κλάσης ‘2’.



Εικόνα 9

Στην εικόνα 10 φαίνονται οι καμπύλες μάθησης του μοντέλου , με μπλε το σφάλμα εκπαίδευσης και με κόκκινο στο σφάλμα αξιολόγησης . Το σφάλμα αξιολόγησης είναι λογικά πιο μεγάλο από της εκπαίδευσης. Λόγω ότι έχουμε μόνο 3 κανόνες το τελικό σφάλμα αξιολόγησης καταλήγει να είναι καλύτερο από το μοντέλο με 2 κανόνες και η εκπαίδευσή του πιο ομαλή.Παρόλα αυτά και σε αυτή τη περίπτωση το σφάλμα παραμένει μεγάλο.



Εικόνα 1

	1	2	3	4	5	6	7	8	9	10	11	12
1	133	316	596	475	166	21	7	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0	0
3	1	6	7	11	10	4	1	0	0	0	0	1
4	1	7	45	42	32	12	2	0	0	0	0	0
5	4	6	21	80	149	118	47	7	5	1	0	0
6	10	61	209	311	171	14	5	2	1	1	0	0
7	0	6	24	52	70	20	5	2	0	0	0	0
8	0	4	18	37	78	63	6	2	0	0	0	0
9	1	0	1	3	14	18	29	50	116	72	24	5
10	0	4	1	4	6	3	0	0	0	0	0	0
11	0	0	0	9	18	28	32	26	24	18	17	37
12	1	0	0	1	11	5	12	22	15	22	7	11

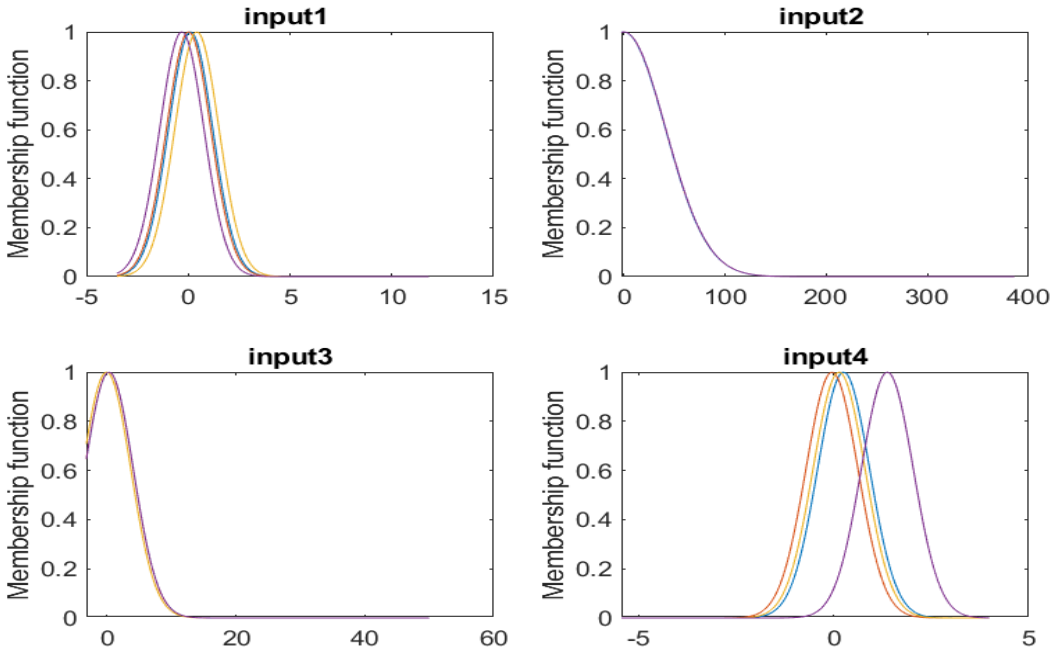
Εικόνα 11:Error matrix

PA	0.077596	0	0.170732	0.297872	0.340183	0.017834	0.027933	0.009615	0.348348	0	0.08134	0.102804
UA	0.880795	0	0.007592	0.040976	0.205517	0.045752	0.034247	0.018018	0.720497	0	0.354167	0.203704

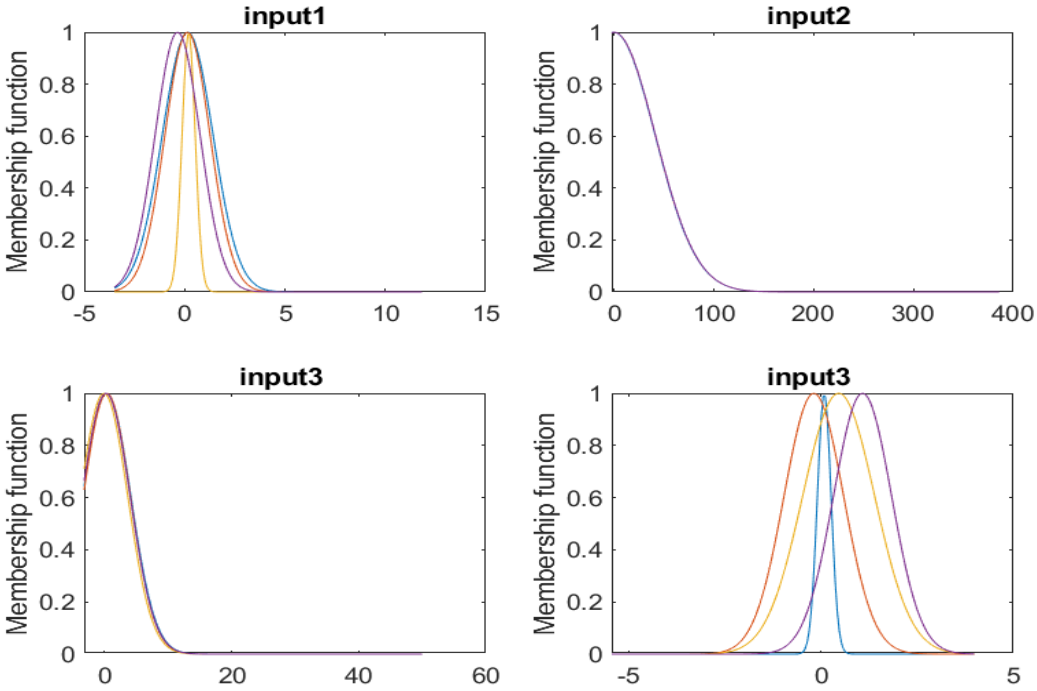
K=0.0583

TSK_3 (4 rules):

Στην εικόνα 12 απεικονίζονται οι συναρτήσεις συμμετοχής για 4 από τα χαρακτηριστικά πριν από την εκπαίδευση του μοντέλου . Όπως φαίνεται ο διαμερισμός έχει γίνει με scatter partitioning με τα ασαφή σύνολα να επικαλύπτονται σχεδόν εξολοκλήρου. Αυτό συμβαίνει διότι πιθανότατα το κάθε χαρακτηριστικό έχει μεγάλο βαθμό συμμετοχής σε ένα cluster, αν κρίνουμε από την επικάλυψη των συναρτήσεων συμμετοχής . Στην εικόνα 13 απεικονίζονται οι συναρτήσεις συμμετοχής των των χαρακτηριστικών μετά την εκπαίδευση του μοντέλου. Φαίνεται πως οι συναρτήσεις έχουν διαμορφωθεί , σύμφωνα με τις τελικές τιμές των παραμέτρων εισόδου , οι βέλτιστες των οποίων βρέθηκαν κατά τη διάρκεια της εκπαίδευσης του μοντέλου.

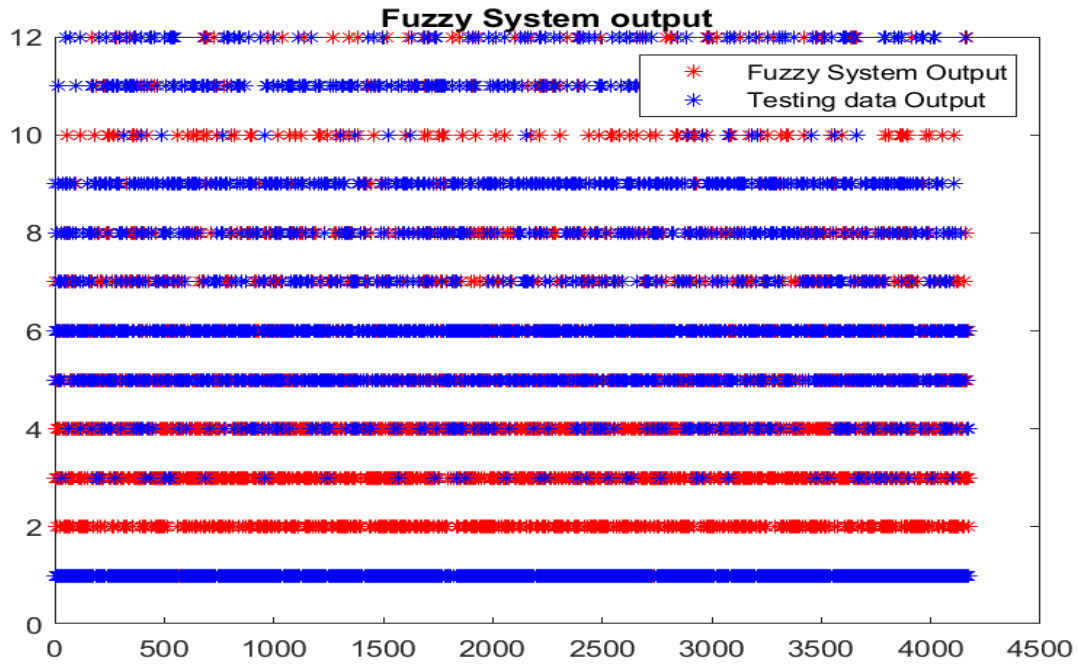


Εικόνα 12



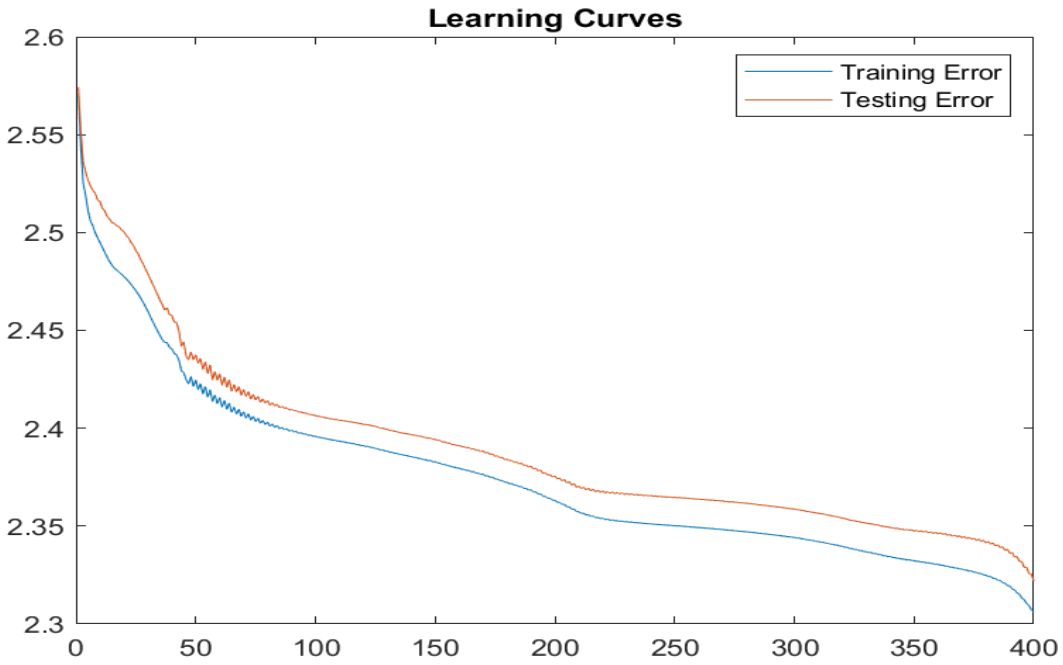
Εικόνα 13

Στην εικόνα 14 απεικονίζεται η έξοδος του ασαφούς μοντέλου (κόκκινο χρώμα) , σε σχέση με την έξοδο των δεδομένων δοκιμής . Φαίνεται πως το ασαφές μοντέλο προσεγγίζει τις τιμές της εξόδου ,κάνοντας και σφάλματα όπως να περιλαμβάνει στις προβλέψεις τιμές εντός του διαστήματος των κλάσεων, αλλά που δεν περιέχονται στην έξοδο του dataset , όπως η τιμή κλάσης '2'.



Εικόνα 14

Στην εικόνα 4 φαίνονται οι καμπύλες μάθησης του μοντέλου , με μπλε το σφάλμα εκπαίδευσης και με κόκκινο στο σφάλμα αξιολόγησης . Το σφάλμα αξιολόγησης είναι λογικά πιο μεγάλο από της εκπαίδευσης. Λόγω ότι έχουμε μόνο 4 κανόνες το σύστημα φάινεται να λειτουργεί αρκετά καλά , και έτσι το τελικό σφάλμα αξιολόγησης καταλήγει να είναι αρκετά μικρότερο από τα προηγούμενα μοντέλα.



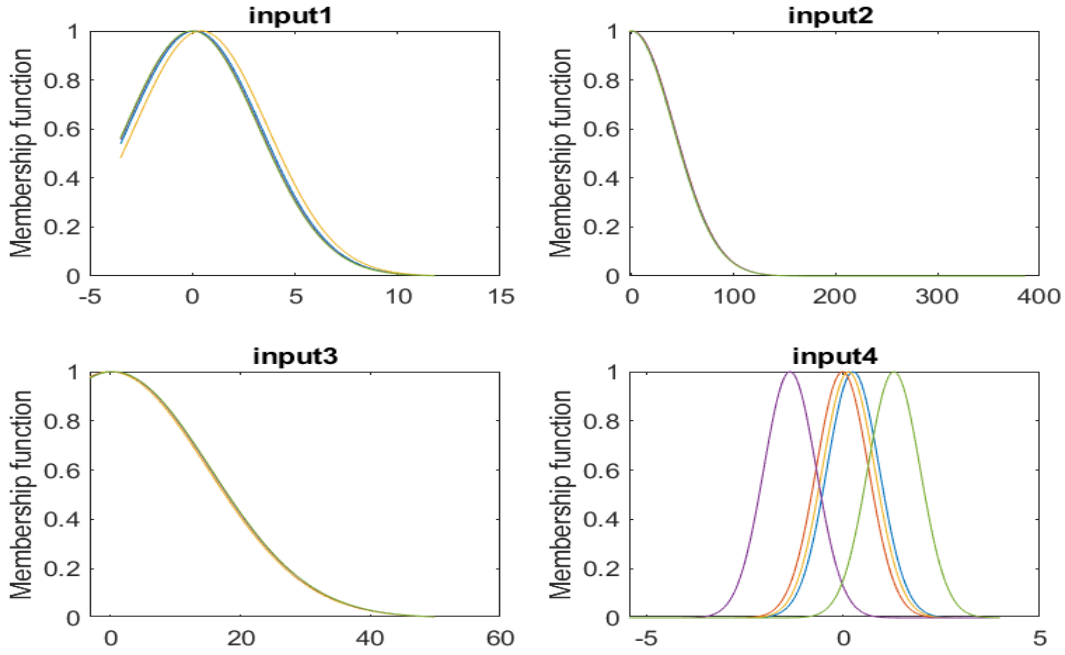
Εικόνα 15

	1	2	3	4	5	6	7	8	9	10	11	12
1	242	392	476	402	176	22	3	1	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0	0
3	4	0	9	10	12	6	0	0	0	0	0	0
4	0	9	32	57	37	6	0	0	0	0	0	0
5	3	11	27	69	138	127	49	12	2	0	0	0
6	26	72	161	246	198	77	4	0	1	0	0	0
7	3	2	15	58	61	32	7	1	0	0	0	0
8	2	0	13	46	60	63	23	1	0	0	0	0
9	0	2	2	2	10	18	41	99	89	44	15	11
10	0	1	1	6	0	2	4	3	1	0	0	0
11	0	1	4	2	14	19	31	32	22	25	21	38
12	0	0	0	1	2	2	7	6	14	16	30	29

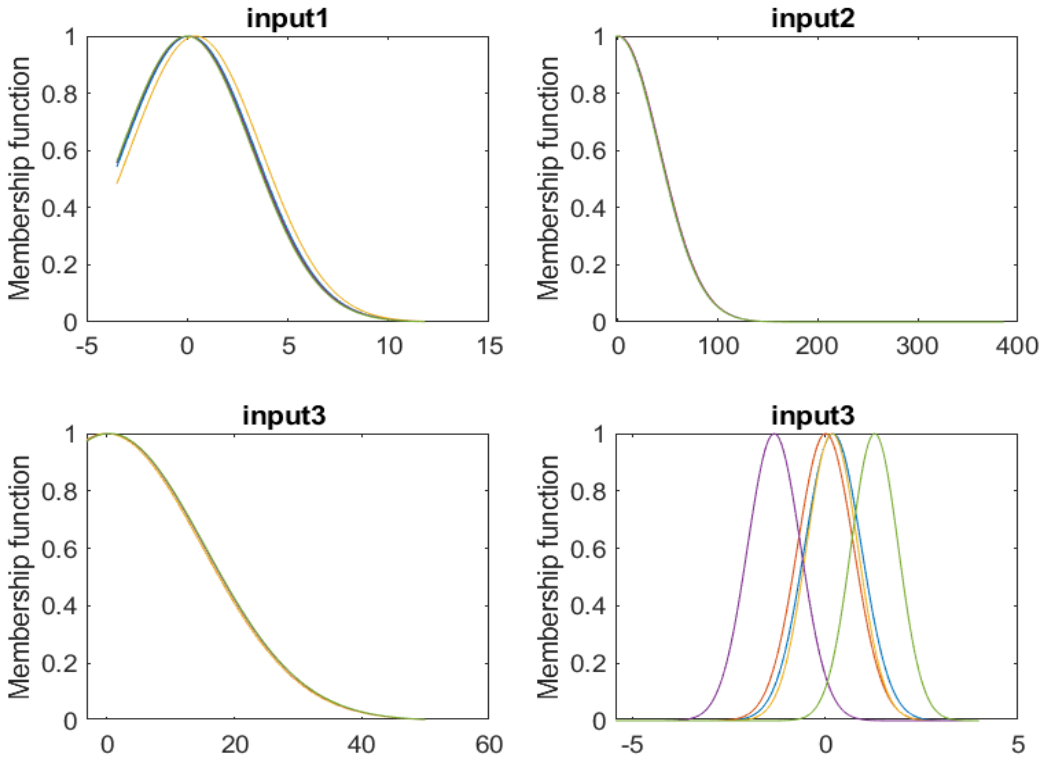
Εικόνα 16:Error matrix

TSK_4:

Στην εικόνα 18 απεικονίζονται οι συναρτήσεις συμμετοχής για 4 από τα χαρακτηριστικά πριν από την εκπαίδευση του μοντέλου . Όπως φαίνεται ο διαμερισμός έχει γίνει με scatter partitioning με κάποια από τα ασαφή σύνολα να επικαλύπτονται σχεδόν εξολοκλήρου. Αυτό συμβαίνει διότι πιθανότατα το κάθε χαρακτηριστικό κατανέμεται κυρίως σε ένα cluster ενώ το χαρακτηριστικό 4 μπορεί να ανήκει και σε κάποιο άλλο cluster ,αν κρίνουμε από τον μεγαλύτερο διασκορπισμό των συναρτήσεων συμμετοχής. Στην εικόνα 19 απεικονίζονται οι συναρτήσεις συμμετοχής των των χαρακτηριστικών μετά την εκπαίδευση του μοντέλου. Φαίνεται πως οι συναρτήσεις έχουν διαμορφωθεί , σύμφωνα με τις τελικές τιμές των παραμέτρων εισόδου , οι βέλτιστες των οποίων βρέθηκαν κατά τη διάρκεια της εκπαίδευσης του μοντέλου. Σε όλα τα μοντέλα έως τώρα παρατηρούμε πως μετά την εκπαίδευση του μοντέλου οι συναρτήσεις συμμετοχής έχουν διαμορφωθεί έτσι ώστε τα χαρακτηριστικά να ανήκουν περισσότερο ξεκάθαρα σε κάποιο cluster, κάνοντας έτσι πιο εύκολη την ταξινόμιση τους.

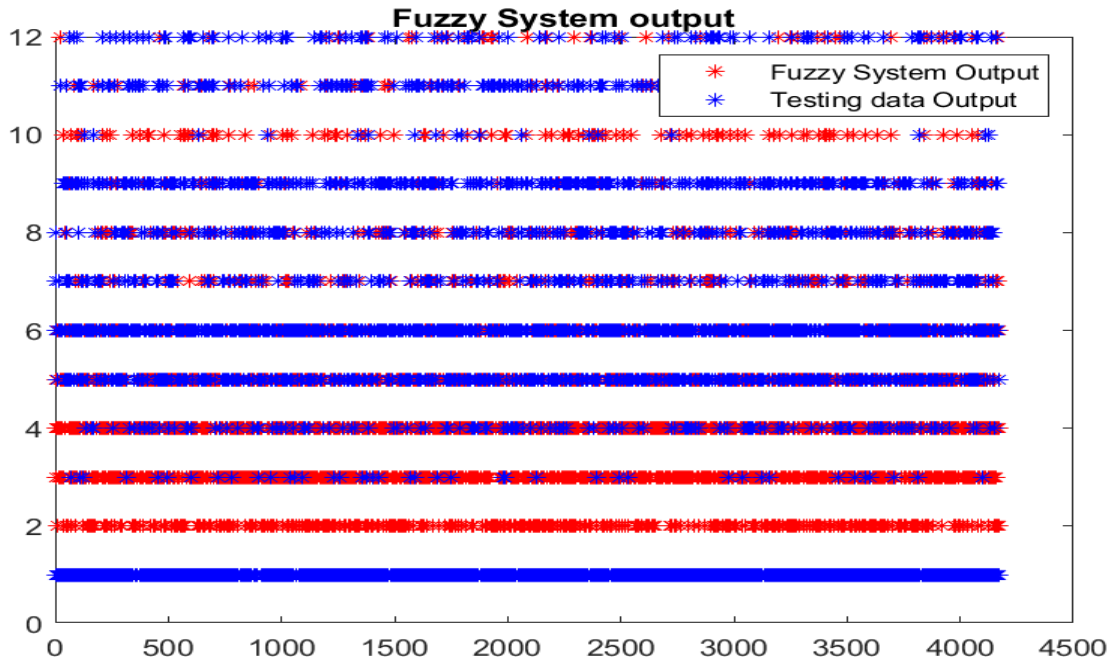


Εικόνα 17



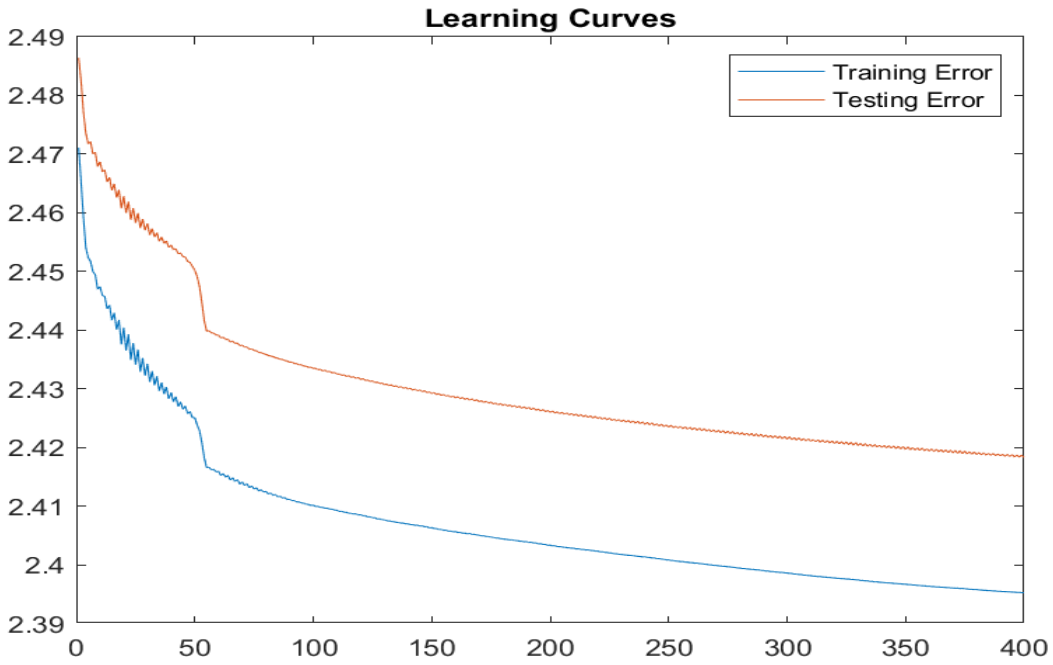
Εικόνα 18

Στην εικόνα 19 απεικονίζεται η έξοδος του ασαφούς μοντέλου (κόκκινο χρώμα) , σε σχέση με την έξοδο των δεδομένων δοκιμής . Φαίνεται πως το ασαφές μοντέλο προσεγγίζει τις τιμές της εξόδου ,κάνοντας και σφάλματα όπως να περιλαμβάνει στις προβλέψεις τιμές εντός του διαστήματος των κλάσεων, αλλά που δεν περιέχονται στην έξοδο του dataset , όπως η τιμή κλάσης '2'.



Εικόνα 19

Στην εικόνα 20 φαίνονται οι καμπύλες μάθησης του μοντέλου , με μπλε το σφάλμα εκπαίδευσης και με κόκκινο στο σφάλμα αξιολόγησης . Το σφάλμα αξιολόγησης είναι λογικά πιο μεγάλο από της εκπαίδευσης. Το σφάλμα αξιολόγισης αυτού του μοντέλου είναι μεγαλύτερο από το προηγούμενο , και αυτό ίσως να οφείλεται στο ότι έχουμε 5 κανόνες , οι οποίοι κάνουν αρκετά πιο περίπλοκη την διαδικασία της εκπαίδευσης , μην δίνοντας τόσο καλά αποτελέσματα.



Εικόνα 20

	1	2	3	4	5	6	7	8	9	10	11	12
1	175	322	614	409	162	25	5	1	1	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0	0
3	2	9	2	14	7	5	0	1	0	0	0	1
4	0	13	47	43	28	6	2	1	1	0	0	0
5	6	10	22	68	171	121	27	8	4	1	0	0
6	16	61	207	283	174	28	9	6	0	0	0	1
7	1	6	14	53	72	24	5	4	0	0	0	0
8	0	2	14	65	62	56	8	1	0	0	0	0
9	0	0	2	5	9	15	24	71	107	65	27	8
10	0	1	1	6	6	1	2	0	1	0	0	0
11	0	0	1	7	13	26	31	26	20	21	25	39
12	0	0	0	2	4	8	16	9	22	21	16	9

Εικόνα 21: Error matrix

PA	0.1021	0	0.04878	0.304965	0.390411	0.035669	0.027933	0.004808	0.321321	0	0.119617	0.084112
UA	0.875	0	0.002165	0.045026	0.241525	0.088889	0.03876	0.007813	0.685897	0	0.367647	0.155172

K=0.0718

Επίλυση προβλήματος με dataset μεγάλης διαστασιμότητας :

Στο δεύτερο πρόβλημα της εργασίας καλούμαστε να μοντελοποιήσουμε ένα πρόβλημα χρησιμοποιώντας το Isolet dataset από το UCI Repository το οποίο αποτελείται από 617 χαρακτηριστικά (εισόδους) τα οποία ταξινομούνται σε 26 κλάσεις. Η τιμή της κλάσης αποτελεί την έξοδο του συστήματος (no 618). Λόγω του μεγάλου αριθού εισόδων του μοντέλου αυξάνεται κατά πολύ ο αριθμός των κανόνων του ασαφούς συστήματος , πράγμα που κάνει πολύ δύσκολη την μοντελοποίηση του. Στην εργασία αυτή αντιμετωπίζουμε το πρόβλημα της πολυδιαστασιμότητας του σετ δεδομένων με το να επιλέξουμε κάποια από τα χαρακτηριστικά εισόδου (τα πιο σημαντικά), μέσω του αλγορίθμου του Relief , και με το να διαμερίσουμε τον χώρο εισόδου χρησιμοποιώντας τον αλγόριθμο fuzzy C-means (FCM).Με αυτόν τον τρόπο επιλέγουμε μόνο τα χαρακτηριστικά που προσφέρουν σημαντική πληροφορία για την εκπαίδευση του μοντέλου και και δημιουργούμε Clusters(ομάδες) με αυτά μειώνοντας έτσι σημαντικά τον αριθμό των κανόνων. Για να πραγματοποιήσουμε αυτό το στόχο αρκεί να ορίσουμε δύο παραμέτρους, τον αριθμό χαρακτηριστικών (NF) και τον αριθμό των κανόνων(NR) του συστήματος που ταυτίζεται με τον αριθμό των clusters που θα δημιουργηθούν από τον FCM. Πιο συγκεκριμένα για τη συγκεκριμένη εργασία η μεταβλητή NR παίρνει τιμές [3, 5 ,7 ,10] και η μεταβλητή NR παίρνει τιμές [3, 5 ,7 ,9 .12]. Οί τιμές της εκφώνησης αλλάχτηκαν σε μικρότερες λόγω της χρονοβόρας εκτέλεσης του προγράμματος για μεγάλες τιμές των NF και NR.Καλούμαστε να επιλέξουμε για αυτές τις τιμές τον συνδυασμό των παραμέτρων που θα δίνει το μικρότερο σφάλμα στο μοντέλο μας. Για ακόμα πιο έγκυρα αποτελέσματα χρησιμοποιούμε την μέθοδο Cross Validation κατά την οποία ,για έναν αριθμό επαναλήψεων,για κάθε πιθανό συνδυασμό τιμών των παραμέτρων , το dataset εκπαίδευσης χωρίζεται στα 2 σε διαφορετικά σημεία σε κάθε επανάληψη . Στο τέλος της εκπαίδευσης κρατάμε τον μέσο όρο των σφαλμάτων για όλους τους πιθανούς διαχωρισμούς του σετ .

Υλοποίηση στο Matlab:

Για την υλοποίηση της εκπαίδευσης του μοντέλου στο Matlab , δημιουργήθηκαν τα scripts class_high_dim_data.m και final_cls_model.m .Στο class_high_dim_data.m πραγματοποιείται η επιλογή δεδομένων με τον αλγόριθμο Relief, η επαναληπτική διαδικασία της εκπαίδευσης με cross Vallidation για κάθε σετ παραμέτρων RF και RL, και η εξαγωγή συμπερασμάτων για το βέλτιστο μοντέλο. Στο final_cls_model.m το βέλτιστο μοντέλο που επιλέχθηκε εκπαιδεύεται και διεξάγονται οι μετρικές και τα διαγράμματα σφάλματος του τελικού μοντέλου.

Class_High_dim_data.m:

Αρχικά φορτώνουμε τα δεδομένα από το excel data sheet τα διαμερίζουμε όπως ακριβώς και στα TSK μοντέλα για το απλό dataset, έτσι ώστε η συχνότητα εγγραφών μιας κλάσης να είναι ίδια και στα 3 σετ δεδομένων. Η μόνη διαφορά είναι ότι δημιουργούμε 26 πίνακες με τον καθένα να περιέχει τις εγγραφές που ανήκουν σε μία κλάση, διότι τώρα έχουμε 26 κλάσεις. Στη συνέχεια εφαρμόζουμε στο training set τον αλγόριθμο Reflief για την επιλογή των βέλτιστων-χρήσιμων χαρακτηριστικών. Ο αλγόριθμος επιστρέφει έναν πίνακα με τα βάρη σημαντικότητας των χαρακτηριστικών και τον δείκτη του καθενός . Ταξινομούμε τα χαρακτηριστικά σε έναν νέο πίνακα " relief_array" από το πιο σημαντικό στο πιο ασήμαντο .Η διαδικασία αυτή φαίνεται στον παρακάτω κώδικα:

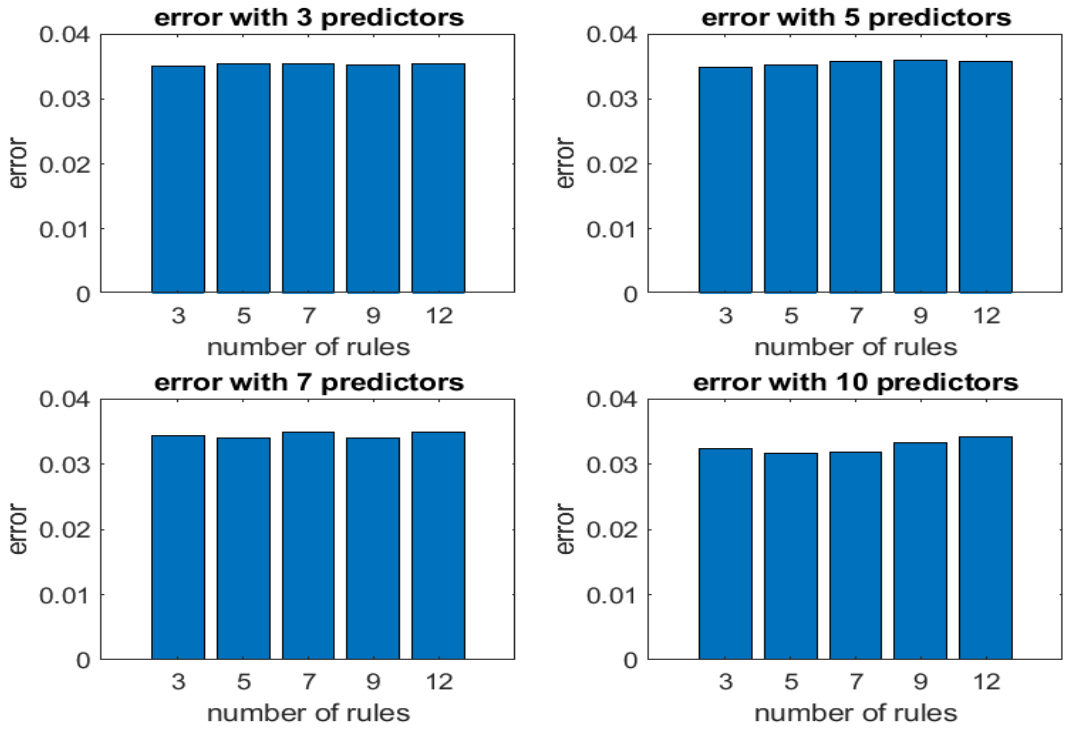
```
[idx,weights] =relieff(training_data(:,1:end-1),training_data(:,end),100);

relief_array=zeros(length(idx),2);

relief_array(:,1)=idx;

relief_array(:,2)=weights;

[relief_array,index] = sortrows(relief_array,2,'descend');
```

Έπειτα δημιουργούμε τα διανύσματα με τις τιμές των παραμέτρων NF και NR και έναν πίνακα errors_array διαστάσεων NF*NR για να αποθηκεύουμε το error του μοντέλου για κάθε ζεύγος παραμέτρων. Στη συνέχεια ξεκινάει η επαναληπτική διαδικασία, για NF*NR επαναλήψεις .Με την συνάρτηση "cvpartition()" δημιουργούμε έναν αριθμό από διαμερισμού του τεστ εκπαίδευσης δεδομένων που αποθηκεύεται στον πίνακα Sets. Για έναν αριθμό επαναλήψεων που ορίζουμε εμείς το μοντέλο εκπαιδεύεται κάθε φορά με διαφορετικά διαμελισμένο σετ δεδομένων και αποθηκεύεται για κάθε σετ το τετραγωνικό σφάλμα στον πίνακα cvsetError. Η διαδικασία φαίνεται στον παρακάτω κώδικα:

```
for k=1:Sets.NumTestSets
        training_set=Sets.training(k);
        testing_set=Sets.test(k);
        .
        .
        .
         anfis_opt = anfisOptions('InitialFIS', initial_fis, 'EpochNumber', 40, 'ValidationData', testing_input testing_output]);
```

```
[train_fis,trainError,stepSize,chkFIS,chkError] = anfis([training_input training_output],anfis_opt);

system_output = evalfis(chkFIS,testing_data(:,relief_array(1:NF(i),1)));

system_output=round(system_output);
system_output(system_output < 1) = 1;
system_output(system_output > 26) = 26;

cvSetError(k)=sum((system_output - testing_data(:, end).^2);

End
```

Στο τέλος της κάθε επανάληψης (για συγκεκριμένο σετ παραμέτρων) αποθηκεύεται ο μέσος όρος των σφαλμάτων όλων των σετ για την εκπαίδευση του μοντέλου στον πίνακα errors_array.Τέλος , με τη συνάρτηση "plot()" δημιουργούμε ένα διάγραμμα που απεικονίζει το σφάλμα του μοντέλου για κάθε συνδυασμό παραμέτρων. Το διάγραμμα φαίνεται στο παρακάτω σχήμα:



Εικόνα 22

Η ελάχιστη τιμή σφάλματος παρατηρήθηκε για NF=10 και για NR=5 .Παρατηρούμε πως για τον μεγαλύτερο αριθμό χαρακτηριστικών (10) το σφάλμα είναι μικρότερο από ότι για μικρότερες τιμές χαρακτηριστικών αλλά αυξάνει για μεγάλο αριθμό κανόνων (10). Αυτό συμβαίνει διότι για μεγάλο αριθμό κανόνων η εκπαίδευση του μοντέλου γίνεται αρκετά συγκεκριμένη και χάνει τη δυνατότητα να αντιμετωπίσει με γενικότητα άγνωστες εισόδους. Επομένως επιλέγουμε έναν επαρκή αριθμό χαρακτηριστικών και κανόνων που να μπορεί να εκπαιδεύσει σωστά το μοντέλο, χωρίς όμως υπερβολή για να αποφευχθεί η υπερεκπαίδευση. Να σημειωθεί πως το σφάλμα μεταξύ των συνδυασμών δεν παρουσιάζει μεγάλες αποκλίσεις, καθώς επιλέχθηκαν μικρές και κοντινές τιμές χαρακτηριστικών και κανόνων, για να είναι δυνατή η εκτέλεση της εκπαίδευσης σε ένα λογικό χρονικό διάστημα.

Final_model.m:

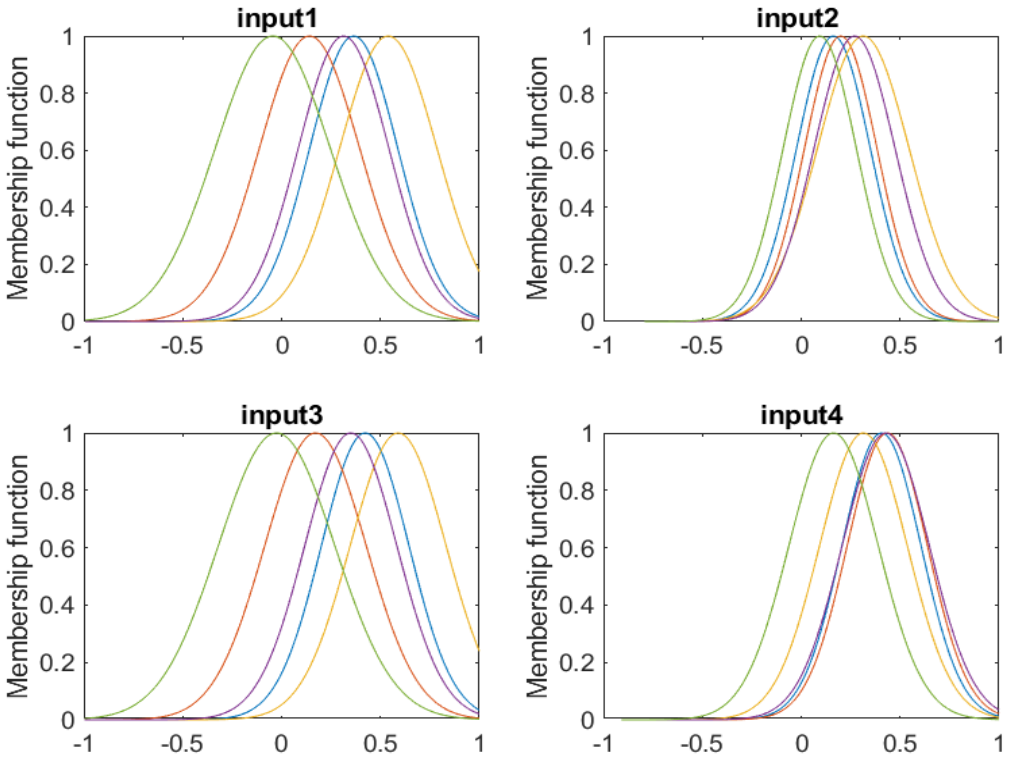
Έχοντας επιλέξει το τελικό μας μοντέλο για NF=10 και NR=5, στη συνάρτηση αυτή το εκπαιδεύουμε ακολουθώντας παρόμοια διαδικασία με τα μοντέλα TSK για το απλό dataset. Πριν το διαχωρισμό των δεδομένων σε training, evaluation και testing , ανακατεύουμε τα δεδομένα για να έχουμε καλύτερα αποτελέσματα , και εφαρμόζουμε τον αλγόριθμο του Relief όπως περιγράφηκε προηγουμένως για την επιλογή των βέλτιστων χαρακτηριστικών. Στη συνέχεια χωρίζουμε και κανονικοποιούμε τα δεδομένα , καθώς και επιλέγουμε μόνο τα χαρακτηριστικά που επέστρεψε ο αλγόριθμος Relief .Εφόσον το βέλτιστο μοντέλο έχει 10 χαρακτηριστικά επιλέγουμε τα 10 πρώτα από τον "relief_array".Η διαδικασία φαίνεται στον παρακάτω κώδικα:

```
input_training_data=training_data(:,relief_array(1:numofPred,1));
output_training_data=training_data(:,end);
```

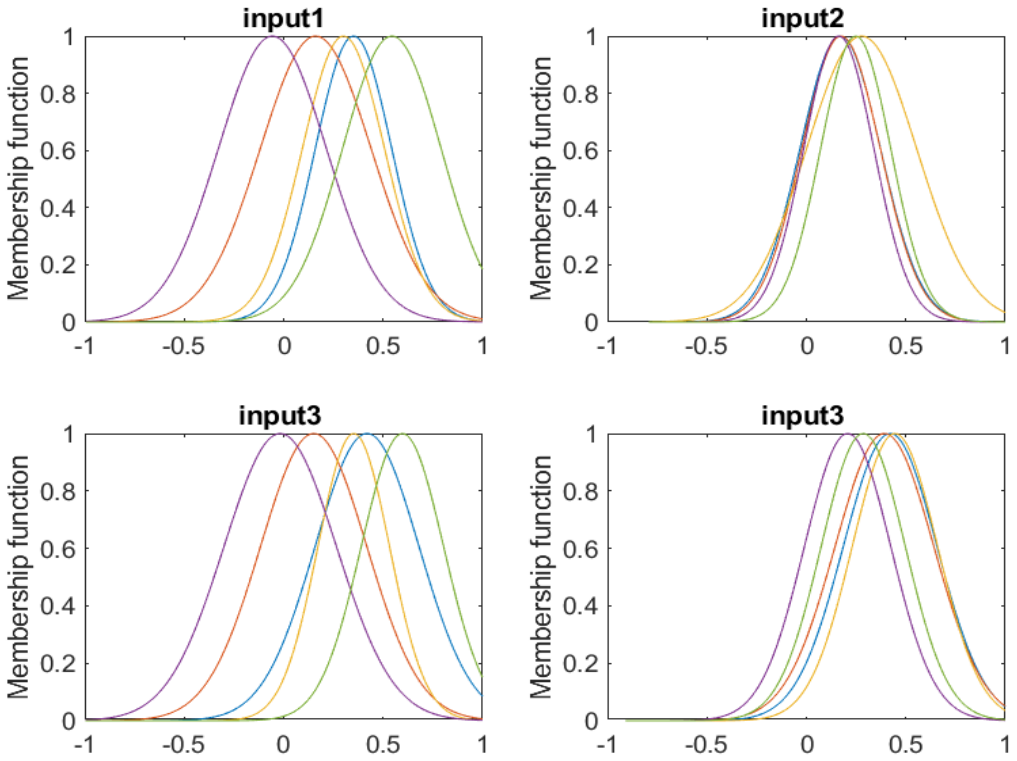
Παρόμοια εργαζόμαστε και για τα evaluation και testing data.Έπειτα επιλέγουμε τα χαρακτηριστικά της συνάρτησης genfis() για τη δημιουργία του fuzzy inference system , θέτοντας αριθμό clusters το 5 (NR=5) και ως μέθοδο διαχωρισμού του χώρου εισόδου τον FCM. Το μοντέλο εκπαιδεύεται με την συνάρτηση anfis() παρόμοια με τα προηγούμενα μοντέλα και τέλος υπολογίζουμε τις μετρικές και εξάγουμε τα διαγράμματα εξόδου συστήματος και σφαλμάτων.

Στην εικόνα 23 απεικονίζονται οι συναρτήσεις συμμετοχής για 4 από τα χαρακτηριστικά πριν από την εκπαίδευση του μοντέλου . Όπως φαίνεται ο διαμερισμός έχει γίνει με scatter partitioning με

.Τα ασαφή σύνολα σε κάποιες εισόδους επικαλύπτονται αρκετά, ενώ σε άλλες απλώνονται περισσότερο στο χώρο εισόδου, άρα έχουν βαθμο συμμετοχής ίσως και σε πολλά clusters.Αυτός ίσως να είναι ένας λόγος που το μοντέλο δεν εκπαιδεύεται τόσο καλά, επειδή σε αυτή τη περίπτωση χρειάζονται περισσότεροι κανόνες το οποίο δεν μπορεί να στηρηχθεί στην πλατφόρμα αυτή, καθώς είναι επιτρεπτός ο αριθμός έως και 5 κανόνων. Στην εικόνα 19 απεικονίζονται οι συναρτήσεις συμμετοχής των των χαρακτηριστικών μετά την εκπαίδευση του μοντέλου. Φαίνεται πως οι συναρτήσεις έχουν διαμορφωθεί , σύμφωνα με τις τελικές τιμές των παραμέτρων εισόδου , οι βέλτιστες των οποίων βρέθηκαν κατά τη διάρκεια της εκπαίδευσης του μοντέλου.

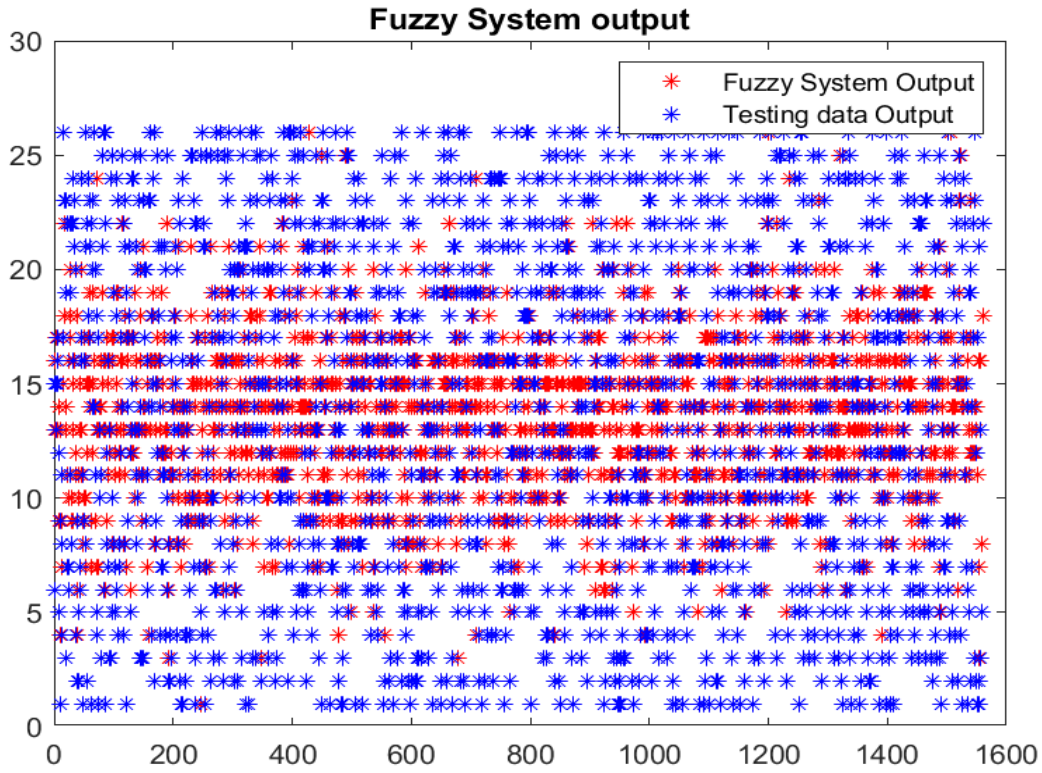


Εικόνα 23



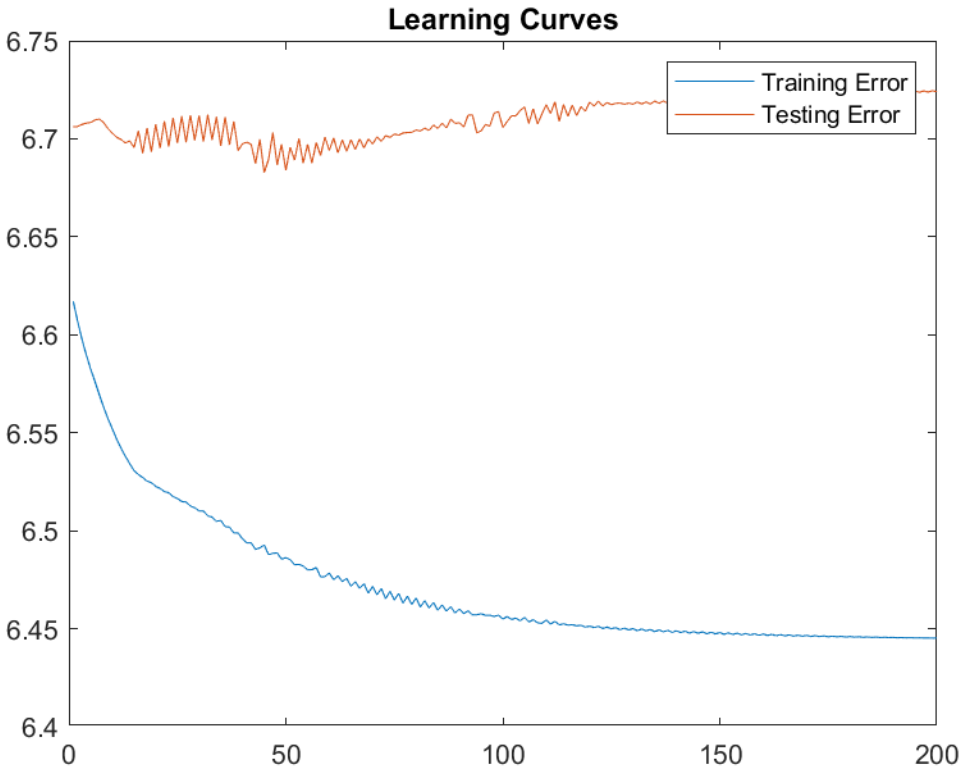
Εικόνα 24

Στην εικόνα 19 απεικονίζεται η έξοδος του ασαφούς μοντέλου (κόκκινο χρώμα) , σε σχέση με την έξοδο των δεδομένων δοκιμής . Φαίνεται πως το ασαφές μοντέλο ταξινομεί σωστα τις εγγραφές στις μεσαίες τιμές του εύρους διαστήματος, αλλα όχι τοσο καλα στις ακραίες τιμές κλάσεων.



Εικόνα 25

Στην εικόνα 26 φαίνονται οι καμπύλες μάθησης του μοντέλου , με μπλε το σφάλμα εκπαίδευσης και με κόκκινο στο σφάλμα αξιολόγησης . Το σφάλμα αξιολόγησης είναι λογικά πιο μεγάλο από της εκπαίδευσης. Το σφάλμα αξιολόγισης αυτού του μοντέλου είναι αρκετά μεγάλο και δείχνει να υπάρχει και υπερεκπαίδευση . Τα αποτελέσματα του μοντέλου δεν είναι τα επιθυμητά λόγ του όχι και τόσο καλού dataset και του μικρού αριθμού κανόνων που μας επιτρέπεται να έχουμε στο σύστημα.


Learning Curves
Training Error
Testing Error
6.75
6.7
6.65
6.6
6.55
6.5
6.45
6.4
0
50
100
150
200

Εικόνα 26