

Classification

1. Εφαρμογή σε απλό dataset

Το τμήμα της εργασίας που αφορά σε αυτό το dataset έγινε με το *MATLAB R2019a*.

Στα δεδομένα εφαρμόστηκε κανονικοποίηση στο διάστημα $[0,1]$.

Και τα 4 μοντέλα εκπαιδεύτηκαν για 500 εποχές, με $\text{step size} = 0,001$, $\text{decrease rate} = 0,9$, $\text{increase rate} = 1,1$.

Έγινε χρήση της συνάρτησης *cvpartition* του Matlab, για τον stratified διαχωρισμό των train, validation και test συνόλων.

Στην περίπτωση της ομαδοποίησης (clustering) χωρίς το διαχωρισμό των κλάσεων, χρησιμοποιήθηκαν οι ακτίνες $r=0,2$ και $r=0,8$, ενώ στην περίπτωση της ομαδοποίησης με διαχωρισμό των κλάσεων οι ακτίνες $r=0,5$ και $r=1$.

Έγιναν χειροκίνητα οι αλλαγές των τύπων των συναρτήσεων εξόδου από *linear* σε *singleton*. Προκειμένου να γίνει αυτό, κατά την εκτέλεση της εργασίας, και χάριν ευκολίας, τα scripts εκτελέστηκαν τοποθετώντας ένα log αμέσως μετά την εντολή `mfedit(fis)`, και πατώντας *continue* μετά την τροποποίηση των συναρτήσεων εξόδου στο GUI, για την εκτέλεση του υπολοίπου.

Μετά την εκτέλεση της ταξινόμησης από το εκάστοτε μοντέλο, και πριν την εκτίμηση της απόδοσής του, έγινε στρογγυλοποίηση των τιμών που δίνει ως έξοδο, καθώς και περιορισμός τους στο διάστημα $[0, 12]$ των φυσικών αριθμών, καθώς πρόκειται για ετικέτες κλάσεων.

TSK_model_1: class independent clustering , radius= 0.2

10 fuzzy set inputs, 1 output, 7 rules

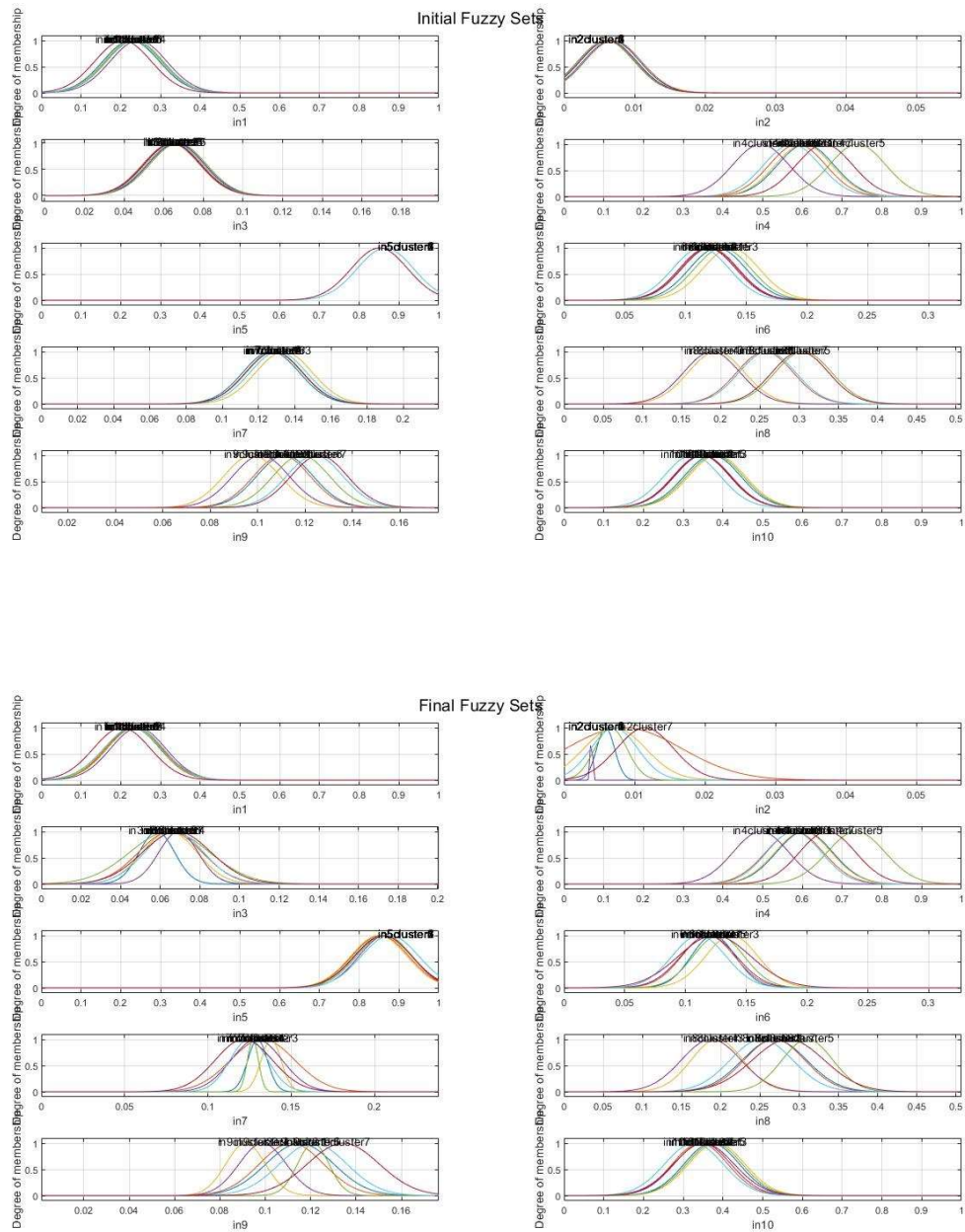


Figure 1: Διάγραμμα μάθησης

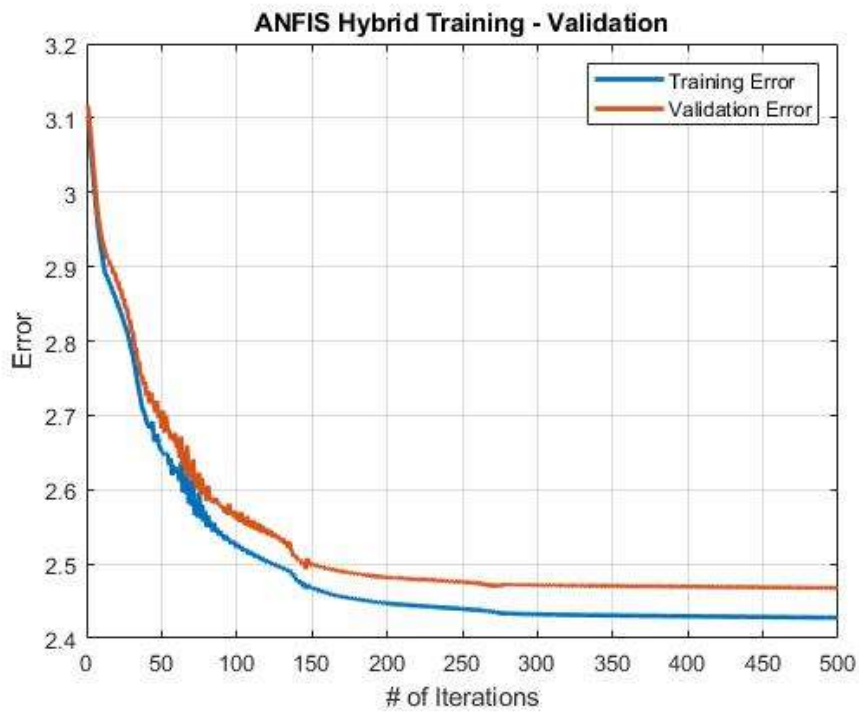


Figure 2: Πίνακας σφαλμάτων ταξινόμησης

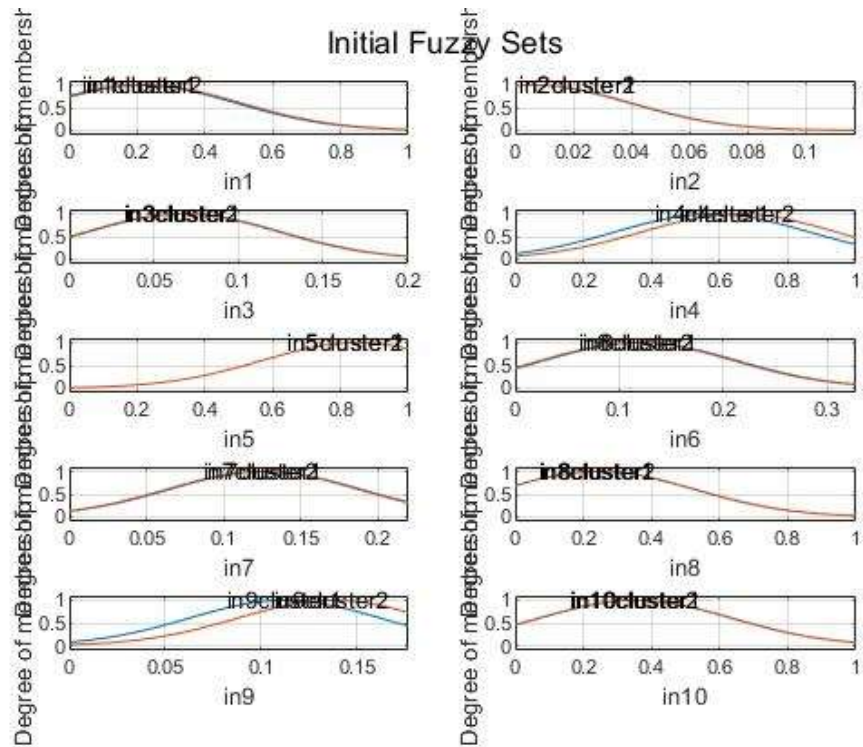
Error Matrix

	1	2	3	4	5	6	7	8	9	10	11	12
1	70	394	743	336	114	29	12	4	8	3	1	
2		3										
3		4	8	5	10	11	1	2				
4		6	68	24	19	15	5	4	1			
5	2	16	74	99	105	73	40	15	5	7	2	
6	6	96	291	201	130	53	5	3				
7		3	23	42	56	39	14		1	1		
8		4	22	33	48	42	44	13	2			
9		4	3	8	12	22	23	67	107	61	25	
10		4	3	2				1	1	3	3	
11		2	4	5	7	6	19	20	15	41	87	2
12			2	4	5	4	16	10	14	17	34	
	1	2	3	4	5	6	7	8	9	10	11	12

Predicted Class

TSK_model_2: class independent clustering , radius= 0.8

10 fuzzy set inputs, 1 output, 2 rules



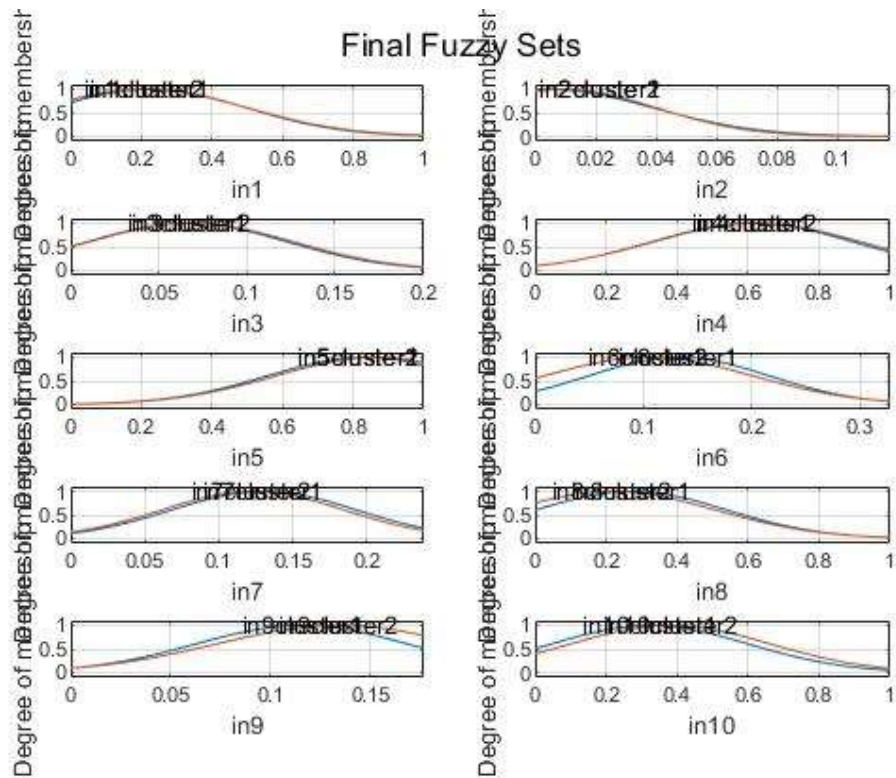


Figure 3: Διάγραμμα μάθησης

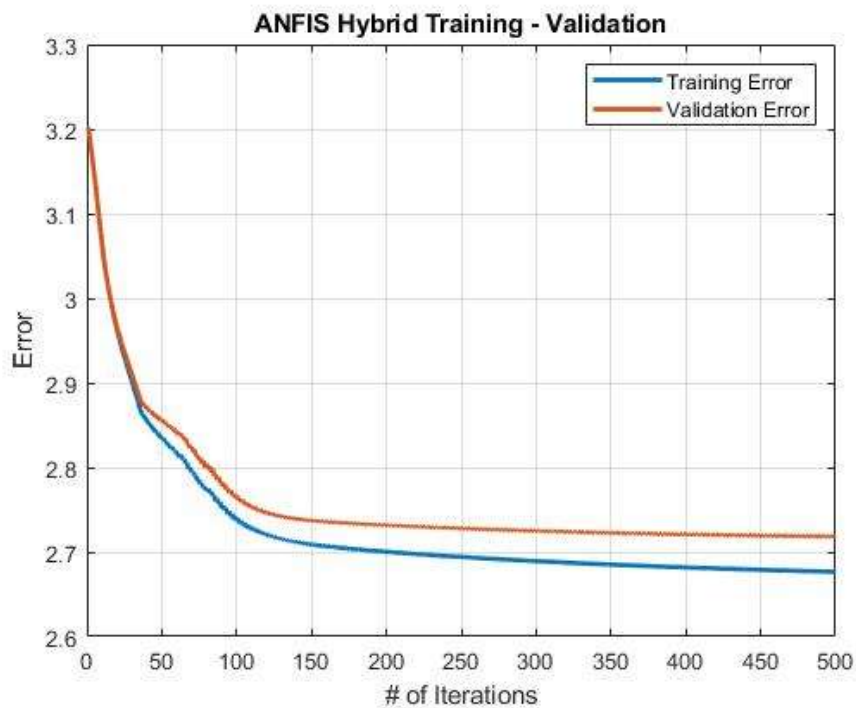


Figure 4: Πίνακας σφαλμάτων ταξινόμησης

Error Matrix

1	53	173	574	692	178	26	17					1
2				1			1					
3		4	6	11	13	6	1					
4	4	9	35	53	30	5	1	1	1	1	2	
5	5	13	33	105	153	106	18	4		1		
6	35	57	223	401	61	4	3	1				
7	1	5	18	75	65	10	3	1				
8	1	3	13	35	86	55	13	2				
9		1	1	2	10	26	33	69	74	62	35	19
10			9	4	2	2		1				
11		1	5	8	10	33	34	40	34	19	10	14
12		2		1	9	8	25	26	22	5	4	5
	1	2	3	4	5	6	7	8	9	10	11	12

Predicted Class

TSK_model_3: class Dependent clustering , radius= 0.5

10 inputs, 1 output, 41 rules

Figure 5: Αρχικά fuzzy σύνολα

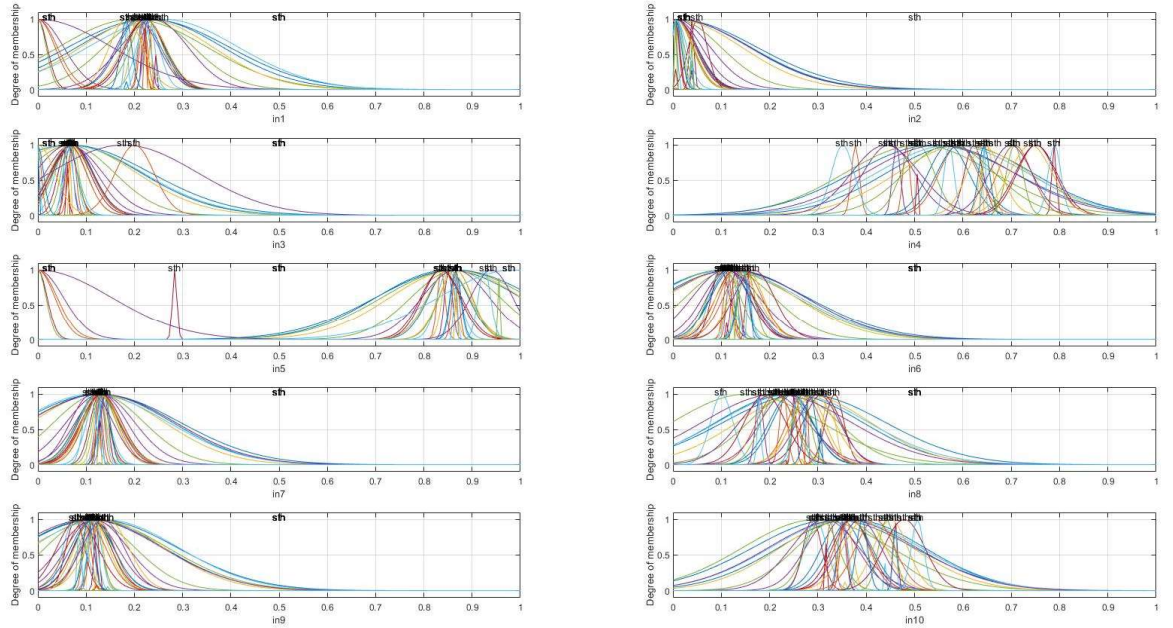


Figure 6: Τελικά fuzzy σύνολα

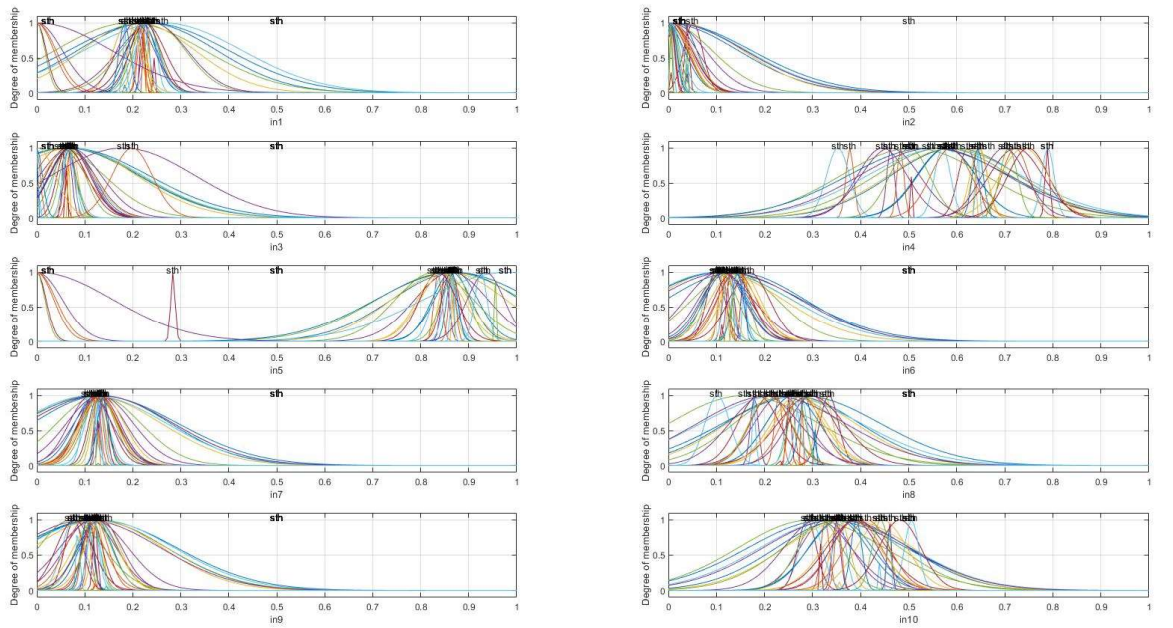


Figure 7: Διάγραμμα μάθησης

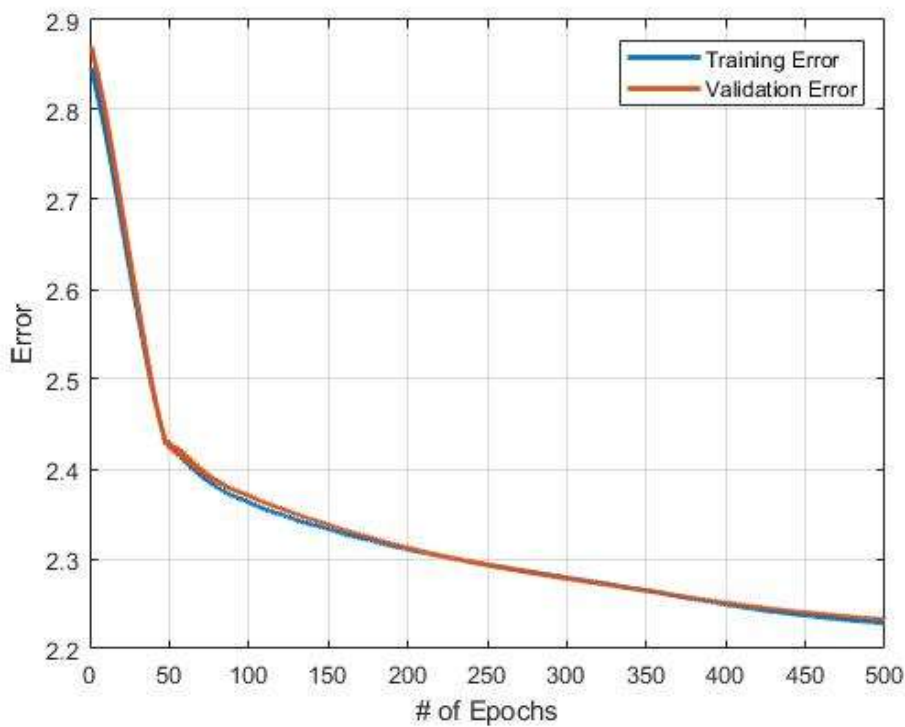


Figure 8: Πίνακας σφαλμάτων ταξινόμησης

[illegible]

TSK_model_3: class Dependent clustering , radius= 1

10 inputs, 1 output, 20 rules

Figure 9: Αρχικά fuzzy σύνολα

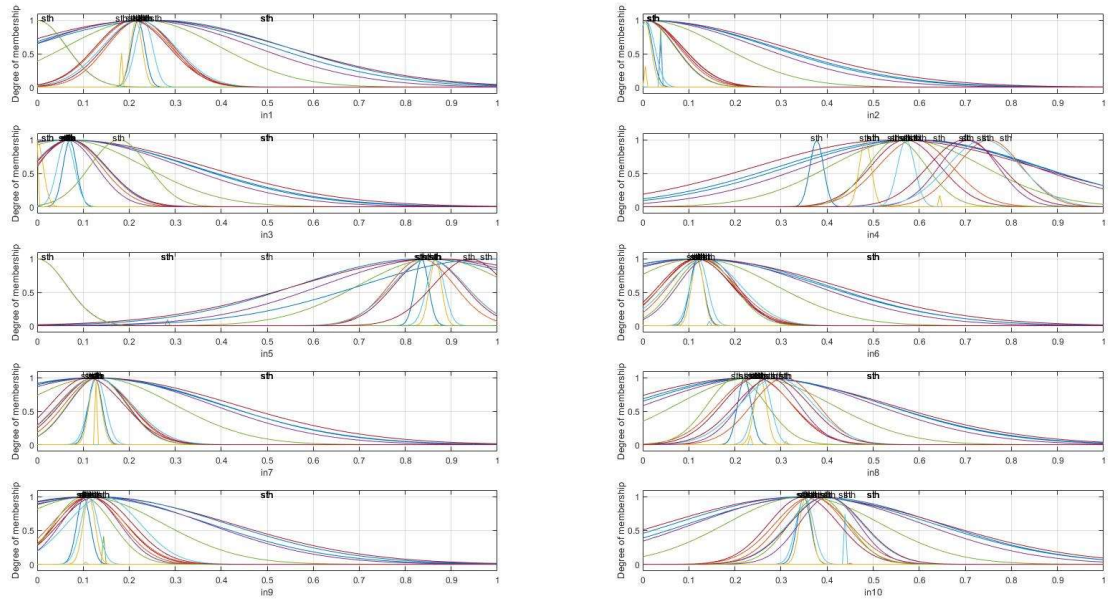


Figure 10: Τελικά fuzzy σύνολα

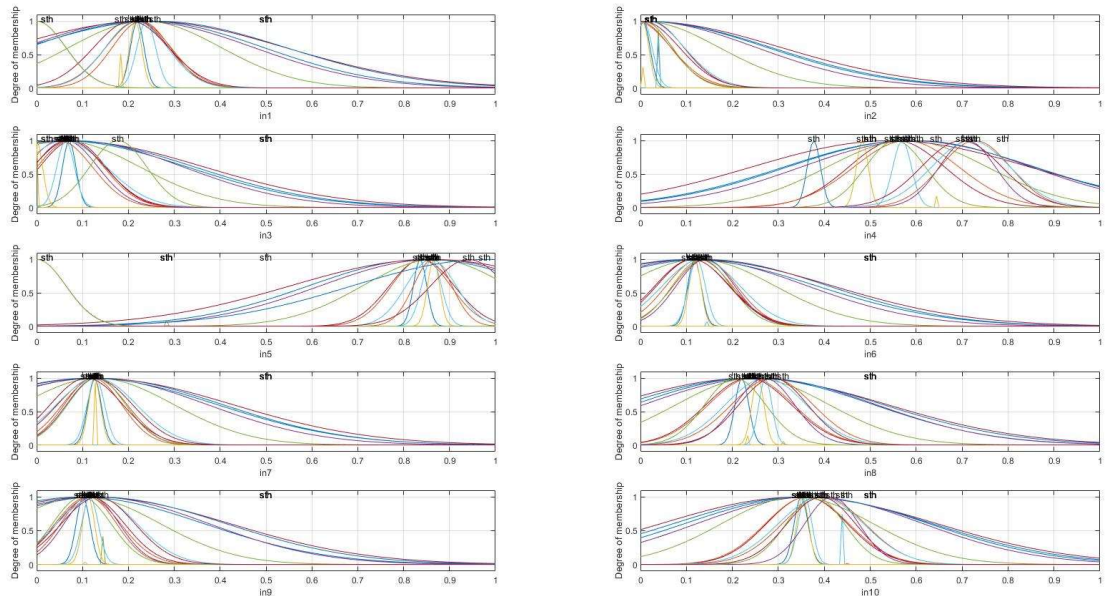


Figure 11: Διάγραμμα μάθησης

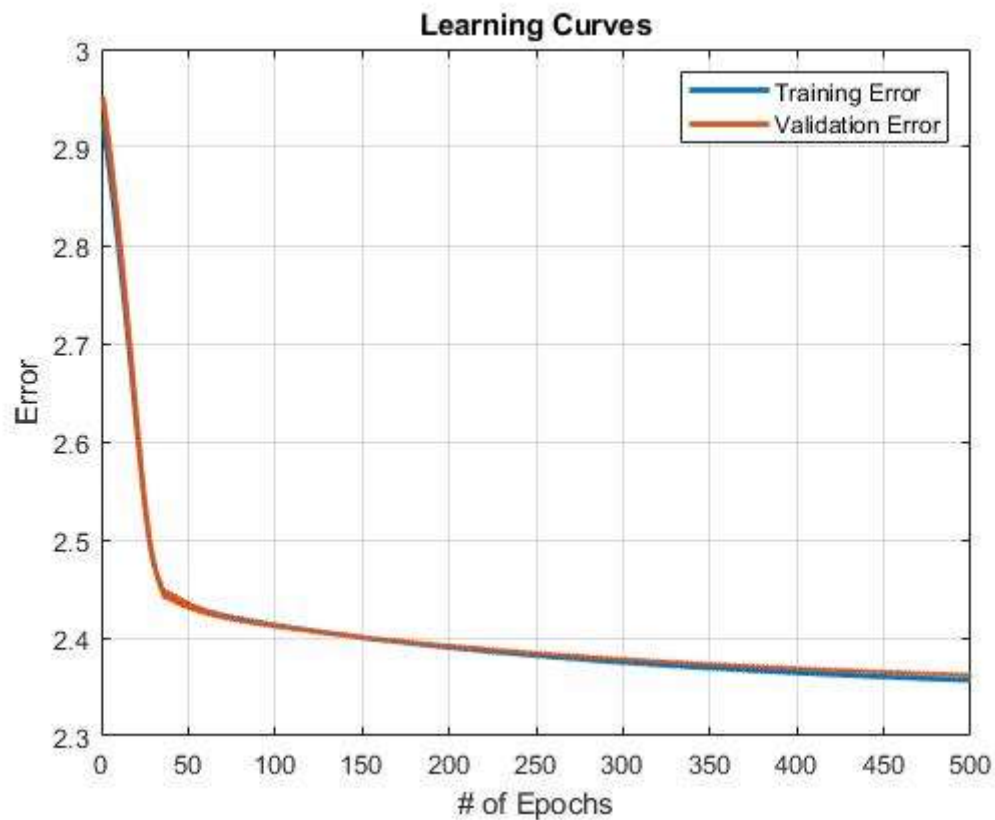


Figure 12: Πίνακας σφαλμάτων ταξινόμησης

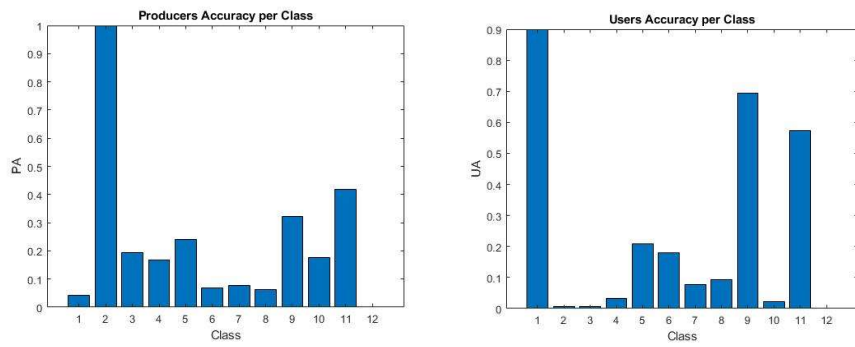
Error Matrix

1	182	388	532	423	134	46	7	2				
2			3									
3		10	3	13	8	4	2	1				
4	4	20	49	31	29	7	2					
5	1	9	37	90	129	112	55	4	1	1		
6	12	65	246	258	142	52	9					
7			20	39	62	42	12	3				
8		3	12	39	67	66	19	2				
9	1		3	2	10	11	19	47	119	115	5	
10		1		8	3	5	1					
11			2	9	9	19	18	29	42	21	13	46
12				7	4	2	6	18	12	27	23	7
	1	2	3	4	5	6	7	8	9	10	11	12

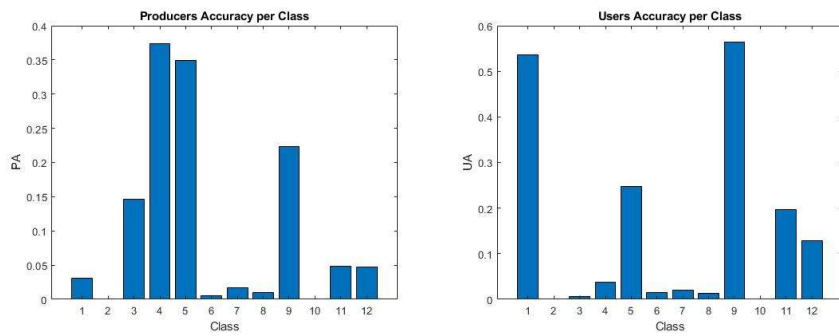
True Class

Predicted Class

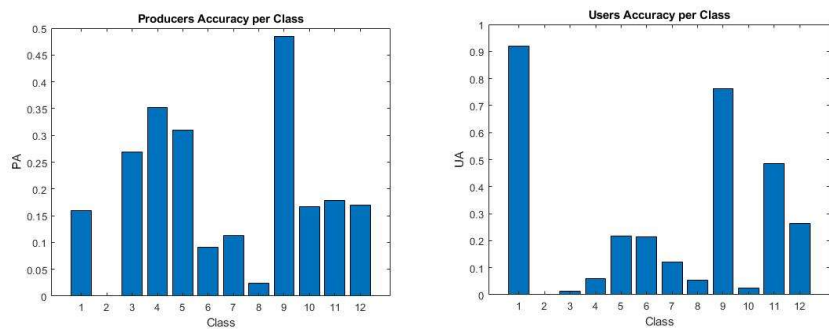
PA και UA για το TSK_model_1



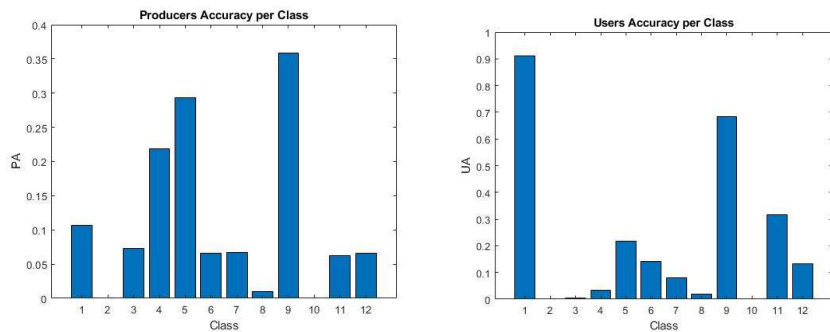
PA και UA για το TSK_model_2



PA και UA για το TSK_model_3



PA και UA για το TSK_model_4



	<i>TSK_model_1</i> Class Independent radius = 0.2	<i>TSK_model_2</i> Class Independent radius = 0.8	<i>TSK_model_3</i> Class Dependent, radius = 0.5	<i>TSK_model_4</i> Class Dependent, radius = 1
rules	7	2	41	20
OA	0.1167	0.0870	0.1884	0.1704
k	0.0690	0.0306	0.1202	0.1040

Οι αποδόσεις και για τα 4 μοντέλα είναι πολύ χαμηλές.

Βάσει των αποτελεσμάτων μας, φαίνεται πως η απόδοση του ταξινομητή αυξάνεται όσο αυξάνεται το πλήθος των κανόνων του. Ο ταξινομητής με τους περισσότερους κανόνες, το *TSK_model_3* με 41 κανόνες, έχει την καλύτερη απόδοση. Ο ταξινομητής με τους λιγότερους, το *TSK_model_2* με 2 κανόνες, τη χειρότερη. Και για τους άλλους δύο ταξινομητές ισχύουν ακριβώς τα ανάλογα.

Ως προς το μέγεθος της ακτίνας : τα *TSK_model_1* και *TSK_model_3*, στα οποία τίθεται μικρότερο μέγεθος ακτίνας cluster, πετυχαίνουν καλύτερες αποδόσεις από τα *TSK_model_2* και *TSK_model_4*, αντίστοιχα. Αυτό οφείλεται στο γεγονός ότι μικρότερη ακτίνας συνεπάγεται περισσότερα clusters, και συνεπώς μεγαλύτερη εκφραστικότητα του μοντέλου που θα δομηθεί. Περισσότερα clusters συνεπάγονται και περισσότερους παραγόμενους κανόνες.

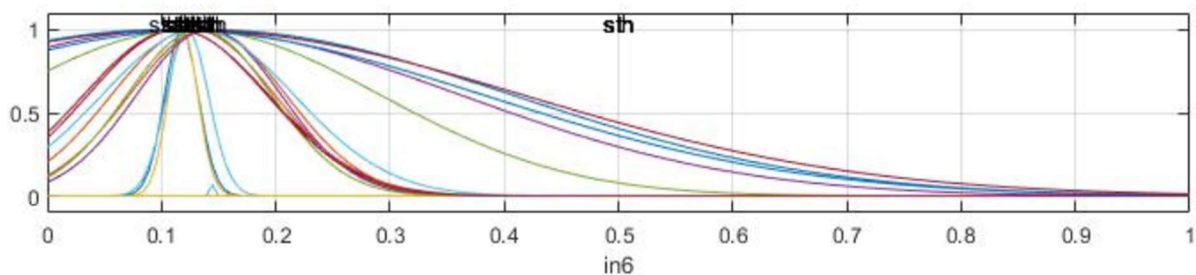
Ως προς τη μορφή της ομαδοποίησης : Τα *TSK_model_3* και *TSK_model_4*, για τα οποία εφαρμόστηκε clustering στα δεδομένα της κάθε κλάσης ξεχωριστά, πετυχαίνουν καλύτερες αποδόσεις από τα *TSK_model_1* και *TSK_model_2*, στα οποία εφαρμόστηκε «τυφλό» clustering, αντίστοιχα. Το αποτέλεσμα αυτό είναι αναμενόμενο καθώς προγραμματίζοντας το μοντέλο να κάνει clustering σε κάθε κλάση ξεχωριστά, ενσωματώνουμε στο μοντέλο την γνώση που ήδη έχουμε για την έξοδο.

Η επικάλυψη των προβολών των ασαφών συνόλων, και συγκεκριμένα στις περιπτώσεις όπου υπερβαίνει μεγάλα ποσοστά, και

Στην περίπτωση μας έχουμε και ποσοστά επικάλυψης πολύ μεγαλύτερα του 50%, ακόμα και 100% ή κοντά στο 100%, δείχνει πολύ μεγάλη ασάφεια.

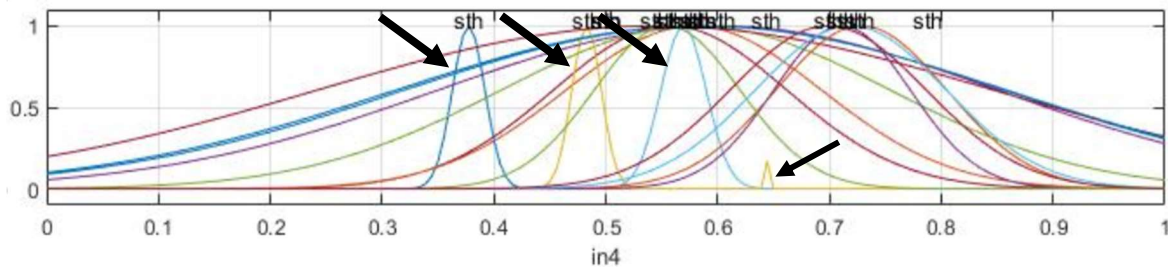
Υπάρχει πλεόνασμα των ασαφών συνόλων που χρησιμοποιούνται για να γίνει η διαμέριση της εισόδου.

Figure 13: Μεγάλες επικαλύψεις ανάμεσα στα σύνολα, μικρή χρησιμότητα για τους κανόνες, μεγάλη ασάφεια (fuzziness), δυσκολία εξαγωγής συμπερασμάτων



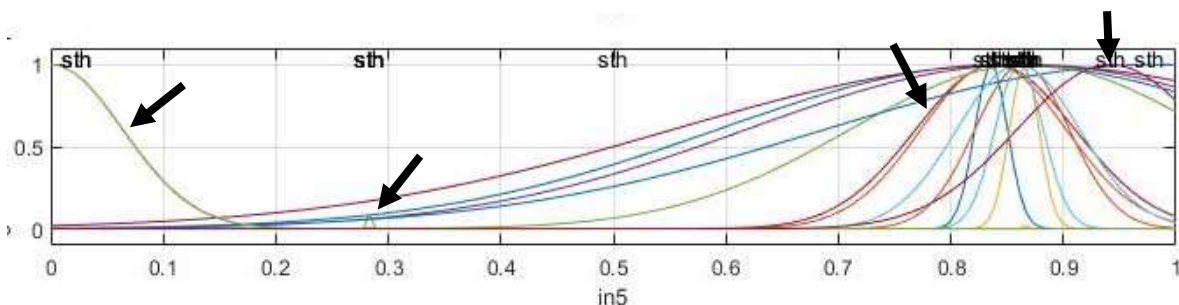
Σε περιπτώσεις όπως η παραπάνω όπου ένα ασαφές σύνολο υπερκαλύπτει όλα τα υπόλοιπα, υπάρχει πλεονασμός στη χρήση των συνόλων. Πρέπει να μειωθούν, μειώνοντας έτσι το πλήθος των κανόνων και την πολυπλοκότητα αλλά διατηρώντας την ικανότητα εξαγωγής συμπερασμάτων.

Figure 14: Κάποια σύνολα δεν έχουν καθόλου επικάλυψη μεταξύ τους



Ασαφή σύνολα που «ξεχωρίζουν» είναι επιθυμητά καθώς φανερώνουν την ύπαρξη «ισχυρού» κανόνα που διαμορφώνει δραστικά το αποτέλεσμα. Ωστόσο αν όλα τα σύνολα έχουν μηδενική επικάλυψη μεταξύ τους, ούτε αυτό εξυπηρετεί την ασαφή συλλογιστική.

Figure 15: Για τη βελτίωση της απόδοσης του συστήματος, πρέπει να διατηρηθούν τα σύνολα που συμβάλλουν περισσότερο στην εξαγωγή συμπερασμάτων και να αφαιρεθούν εκείνα που κυρίως προσθέτουν ασάφεια. Τα βελάκια δείχνουν κάποια ενδεικτικά σημεία ενδιαφέροντος.



Για τη βελτίωση της σχεδίασης του τμήματος της υπόθεσης θα μπορούσε να εφαρμοστεί μείωση των ασαφών συνόλων που θα χρησιμοποιηθούν σε κάθε είσοδο, με κριτήριο τη μεγιστοποίηση της χρησιμότητας τους στην εξαγωγή συμπεράσματος. Θα προτείναμε επιλογή των ασαφών συνόλων που θα χρησιμοποιηθούν με βάση α) πληροφορία/ γνώση των ειδικών του τομέα αναφορικά με τα χαρακτηριστικά που χρησιμοποιούνται ως είσοδοι, τη σημασία αυτών και τον τρόπο που επηρεάζουν την έξοδο, σε συνδυασμό με β) την αξιοποίηση διαγραμμάτων όπως τα παραπάνω για τον προσδιορισμό των πιο χρήσιμων συνόλων.

Από πλευράς μαθηματικής πλευράς μπορεί να γίνει δοκιμή και άλλων μορφών συναρτήσεων συμμετοχής καθώς και άλλων μεθόδων μηχανικής μάθησης για την προσαρμογή τους.

2. Εφαρμογή σε dataset με υψηλή διαστασιμότητα

Το τμήμα της εργασίας που αφορά σε αυτό το dataset έγινε με το **MATLAB R2018a**.

Το Superconductivity dataset περιλαμβάνει 7797 δείγματα με 618 μεταβλητές/χαρακτηριστικά.

Όπως αναφέρεται στην εκφώνηση, με 618 αν μεταβλητές αν διαμερίζαμε το χώρο εισόδου κάθε μεταβλητής με δύο ασαφή σύνολα, θα προέκυπταν 2^{618} κανόνες.

Υλοποίηση

Προεπεξεργασία δεδομένων: Ελέγχθηκε ότι το dataset Isolet δεν περιέχει NaN τιμές. Στα δεδομένα εφαρμόστηκε κανονικοποίηση στο διάστημα [0,1].

Μέσω του αλγορίθμου *relieff* του Matlab, οι 618 μεταβλητές ταξινομήθηκαν με βάση το βαθμό σημαντικότητάς τους, και αποθηκεύτηκαν στη μεταβλητή *rankClass30*. Οι υπολογισμοί του *relieff* έγιναν με βάση τους 30 πλησιέστερους γείτονες.

Το αρχικό dataset χωρίστηκε σε train set και test set, με αναλογία 80/20.

Αρχικοποιήθηκαν οι εξής μεταβλητές:

- Πλήθος εποχών εκπαίδευσης, epochs = 500
- Εύρος πλήθους χαρακτηριστικών που επιλεχθούν για την εκπαίδευση, $G1 = [5 - 20]$
- Εύρος τιμών της ακτίνας που θα χρησιμοποιηθεί για την ομαδοποίηση (clustering), $G2 = [0.3, 0.4, 0.5, 0.6, 0.7, 0.8]$

Η αναζήτηση πλέγματος (grid-search) υλοποιήθηκε στο $G1 \times G2$.

Για κάθε ζεύγος του $G1 \times G2$, έτρεξε 5-πτυχη διασταυρωμένη επικύρωση (5-fold cross validation).

Σε κάθε γύρο του cross validation,

1. το train set, χωρίζεται σε **trn set** και **val set** με αναλογία 80/20,
2. αρχικοποιείται μοντέλο FIS (*fis* του Matlab), με εισόδους το **trn set** (X & Y) και παραμέτρους 'SubtractiveClustering' και την ακτίνα που αντιστοιχεί στο τρέχον ζεύγος του $G1 \times G2$,
3. το μοντέλο εκπαιδεύεται με εισόδους τα **trn set** και **val set** για 500 εποχές, με step size = 0.01, decrease rate = 0.9, increase rate = 1.1
4. για το τρέχον μοντέλο αποθηκεύεται το ελάχιστο σφάλμα επικύρωσης, όπως αυτό επιστρέφεται από τη συνάρτηση *anfis*. Το σφάλμα αυτό αποθηκεύεται μαζί με το αντίστοιχο ζεύγος του $G1 \times G2$.

Αφού εκτελεστεί το grid search για όλα τα ζεύγη του $G1 \times G2$, επιλέγεται ως βέλτιστο το ζεύγος που αντιστοιχεί στο ελάχιστο σφάλμα.

Από το σύνολο train set (το 80% του αρχικού συνόλου), επιλέγονται τόσα χαρακτηριστικά όσα υποδεικνύει η τιμή του $G1$ από το ζεύγος που επιλέχθηκε ως το βέλτιστο. Τα χαρακτηριστικά αυτά επιλέγονται με βάση το πως αξιολογήθηκαν από τον αλγόριθμο *relieff* (να σημειωθεί ότι το ίδιο γίνεται και στο βήμα του cross-validation).

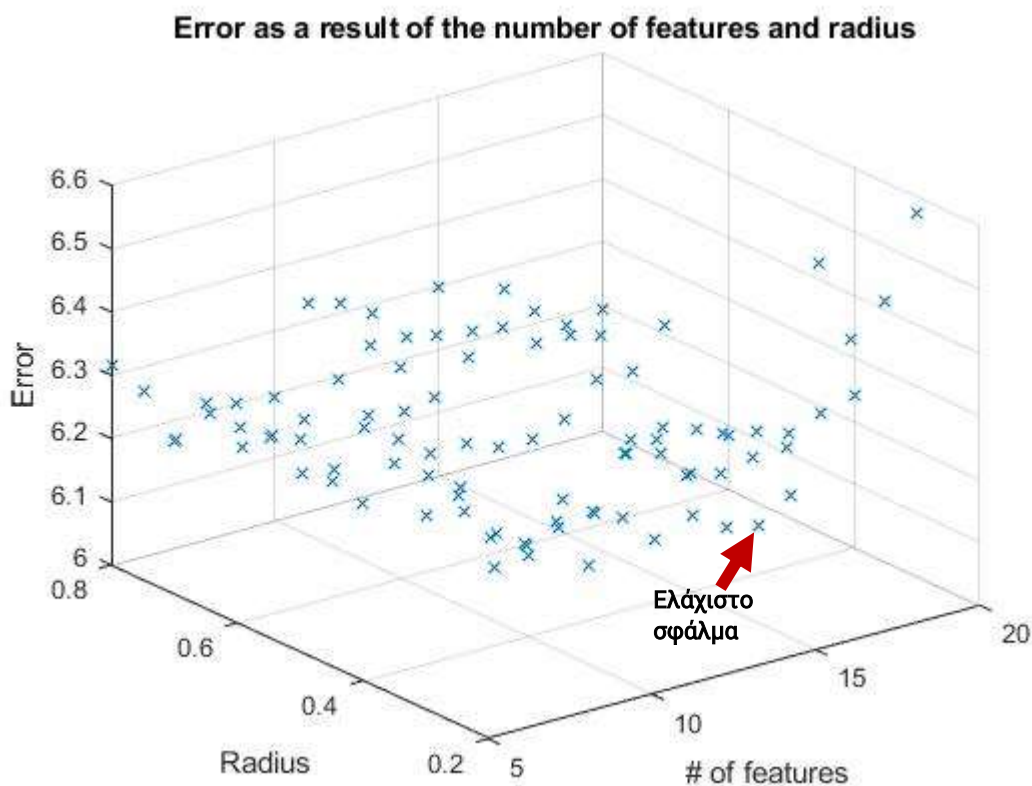
Ακολουθεί εκπαίδευση μοντέλου με είσοδο το παραπάνω σύνολο με τα βέλτιστα G1 και G2, και με ακριβώς τις ίδιες παραμέτρους όπως και στο βήμα του cross-validation (500 εποχές, 0.001 step size, κλπ).

Τέλος με βάση το εκπαιδευμένο αυτό μοντέλο γίνεται πρόβλεψη στο test set (20% του αρχικού συνόλου).

Τα αποτελέσματα της αναζήτησης grid-search αποθηκεύτηκαν στο αρχείο *Results.xls* το οποίο περιλαμβάνεται στα αρχεία της εργασίας.

Ως βέλτιστες παράμετροι από το δοσμένο εύρος επιλέχθηκαν:

- 19 μεταβλητές/χαρακτηριστικά,
- ακτίνα ομαδοποίησης ίση με 0,5
- ελάχιστο σφάλμα 6.0043



Ωστόσο όπως μπορεί να παρατηρηθεί και από το παραπάνω διάγραμμα και από το αρχείο των αποτελεσμάτων *Results.xls*, το σφάλμα κινείται σε πολύ μικρό εύρος [6.0043 - 6.5727], δηλαδή οι ελεύθερες μεταβλητές, στα διαστήματα που εξετάστηκαν, δεν επηρεάζουν ουσιαστικά. Το αποτέλεσμα αυτό ερμηνεύεται λαμβάνοντας υπόψη ότι από τις 618 μεταβλητές, χρησιμοποιήθηκαν το πολύ 20 στο grid-search που εκτελέστηκε, λόγω απαιτήσεων σε υπολογιστική ισχύ. Προφανώς μιλάμε για πολύ μικρό ποσοστό και για να μπορεί να γίνει λόγος για αντιπροσωπευτικά αποτελέσματα η αναζήτηση πλέγματος θα έπρεπε να έχει γίνει για πολύ μεγαλύτερο αριθμό μεταβλητών.

Το τελικό μοντέλο περιέχει 19 εισόδους, 1 έξοδο και 4 κανόνες.

Με 19 εισόδους, grid partitioning και 2 ασαφή σύνολα ανά είσοδο, θα είχαμε $2^{19} = 524288$ κανόνες, με 3 ασαφή σύνολα ανά είσοδο θα είχαμε 3^{19} κανόνες. Προφανώς σε αυτές τις περιπτώσεις το πλήθος των κανόνων είναι απαγορευτικό.

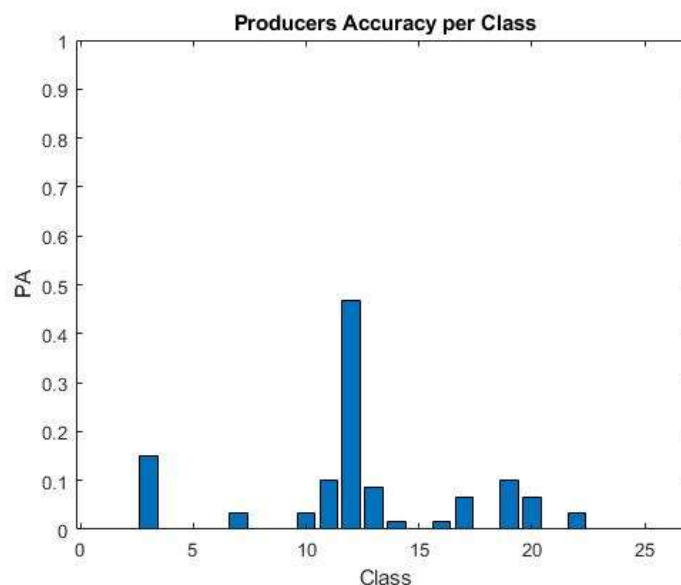
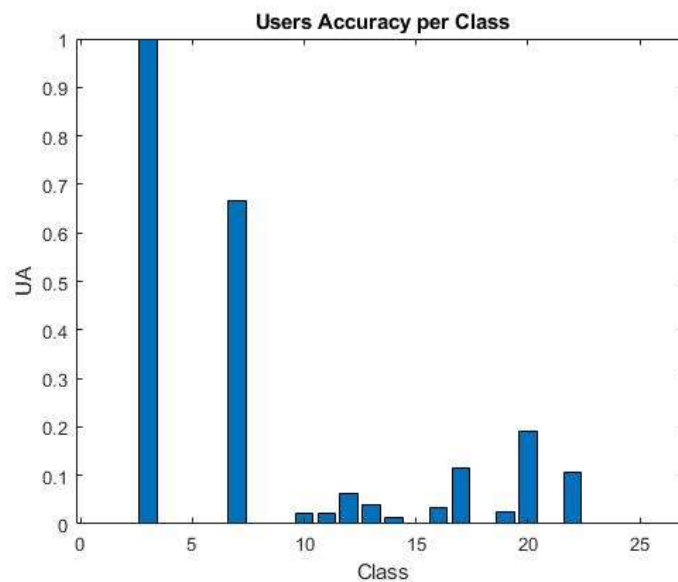
Αποτελέσματα - Διαγράμματα

Η συνολική ακρίβεια του μοντέλου είναι πολύ χαμηλή, OA = 4,49%.

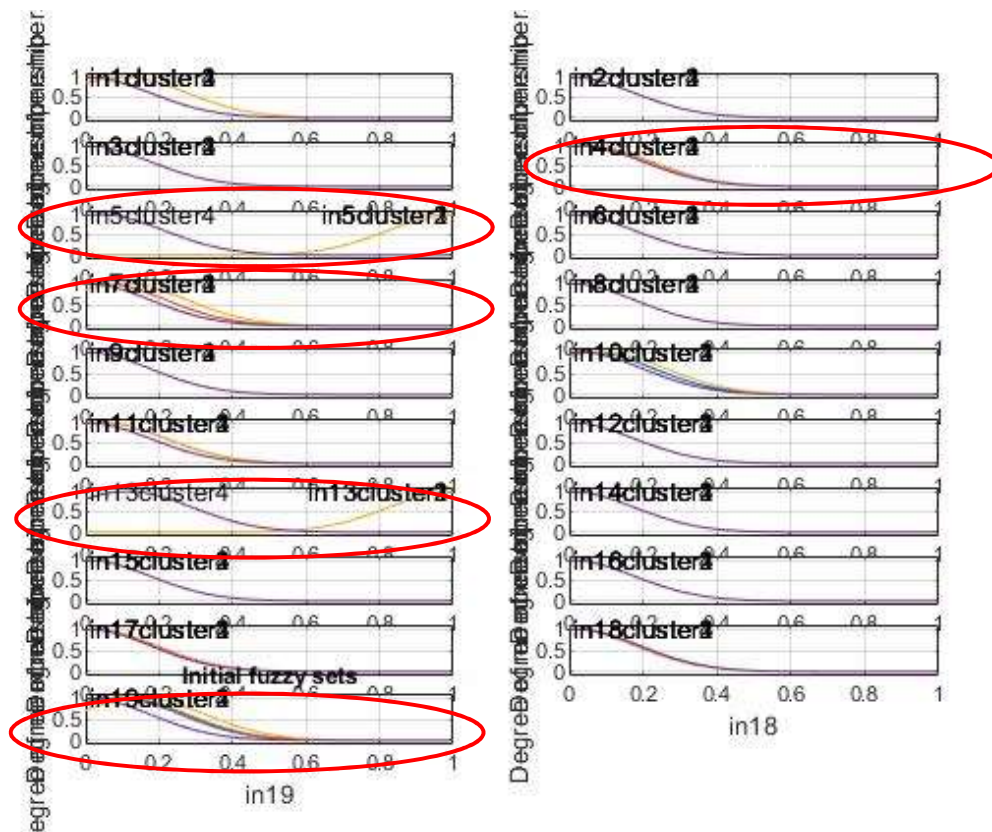
Ο συντελεστής καρρα επίσης, $k = 0.0067$.

Φαίνεται πως οι 4 κανόνες στους οποίους κατέληξε ο ταξινομητής δεν επαρκούν, ποσοτικά και ποιοτικά, για την ικανοποιητική ταξινόμηση 26 κλάσεων.

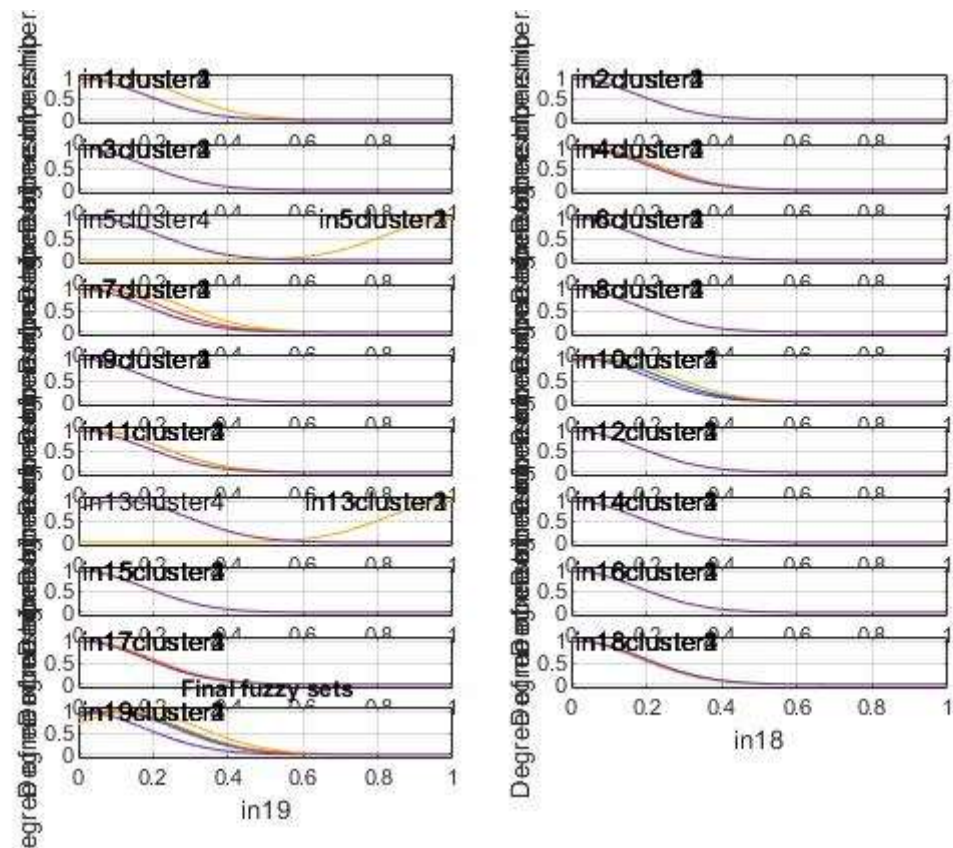
Σημαντικό UA έχουμε μόνο για τις κλάσεις 4 (UA = 1) και 8 (UA = 0,67). Το μεγαλύτερο PA είναι για την κλάση 13 (PA = 0.46). Πίνακες με όλες τις τιμές έχουν αποθηκευτεί στα αρχεία UA.x/sx και PA.x/sx, αντίστοιχα.



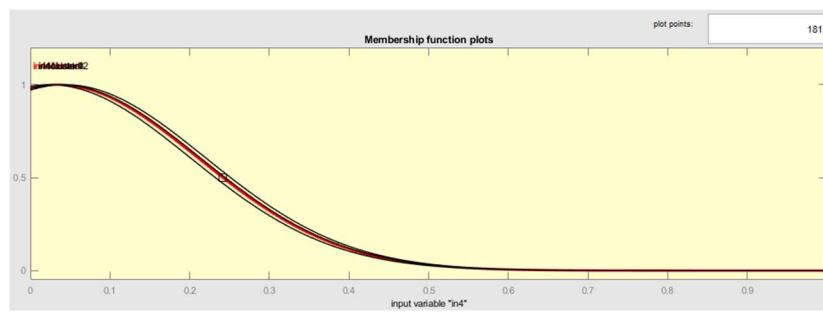
Εικόνα 1: Αρχικά fuzzy σύνολα



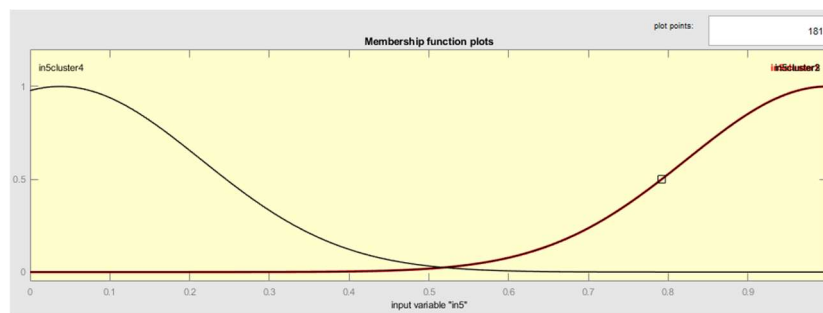
Εικόνα 2: Τελικά fuzzy σύνολα



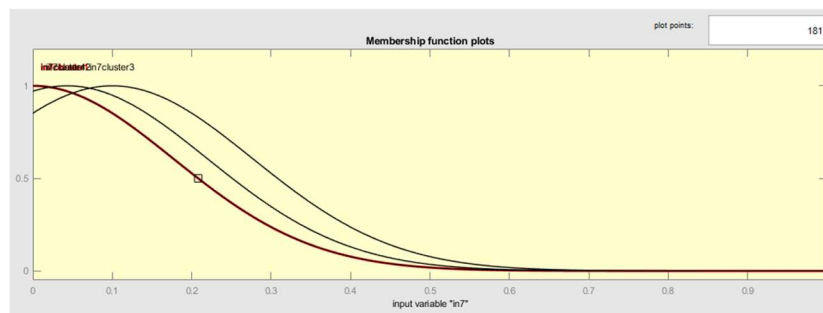
Εικόνα 3: Input 4 final



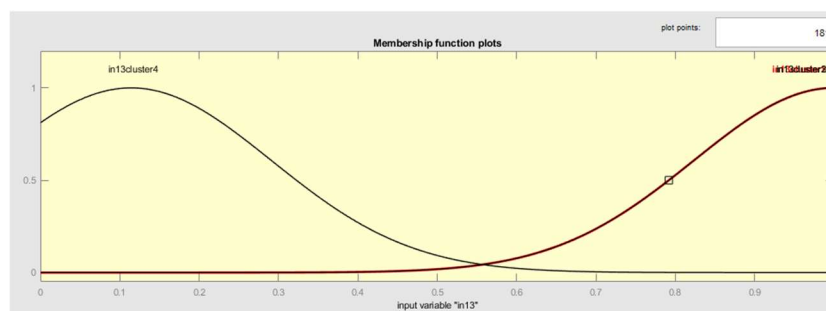
Εικόνα 4: Input 5 final



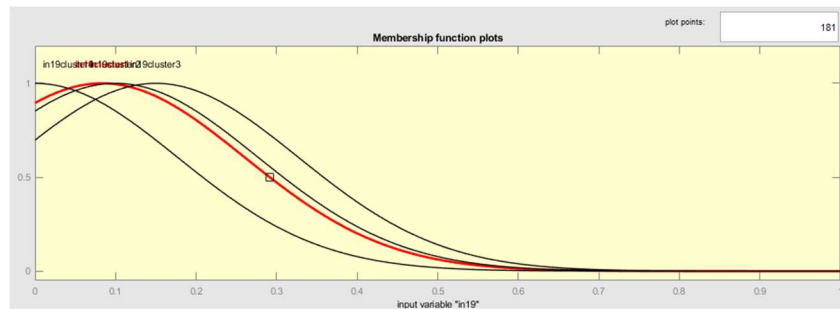
Εικόνα 5: Input 7 final



Εικόνα 6: Input 13 final

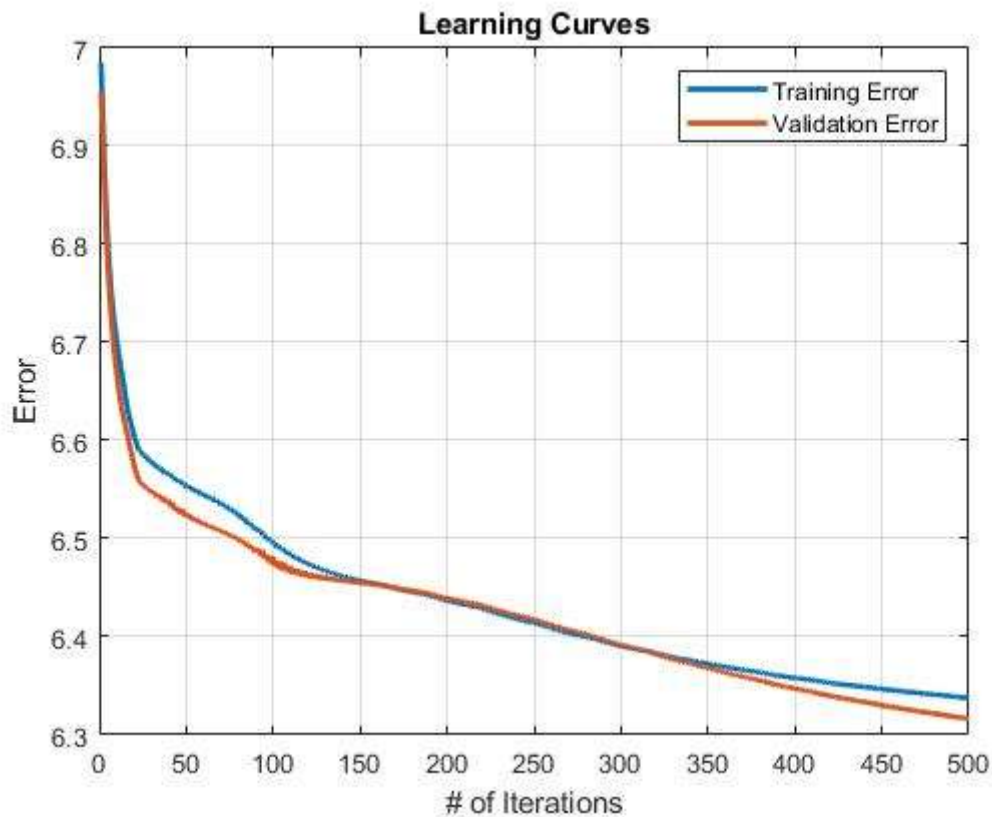


Εικόνα 7: Input 19 final



Σχετικά με τη μορφή και το πλήθος το κανόνων φαίνεται ότι αδυνατούν να περιγράψουν ικανοποιητικά το παρόν σύστημα. Πέρα του πλήθους των κανόνων που όπως προαναφέρθηκε είναι μικρό, από τα διαγράμματα που αντιστοιχούν στη διαμέριση των 19 εισόδων, και η πλειοψηφία των οποίων (17/19) παρουσιάζει μεγάλη ομοιότητα μπορεί να παρατηρηθεί πολύ μεγάλη επικάλυψη των ασαφών συνόλων, μη ικανοποιητική μοντελοποίηση της ασάφειας και συνεπώς πολύ χαμηλές δυνατότητες για τον ταξινομητή ασαφούς λογικής.

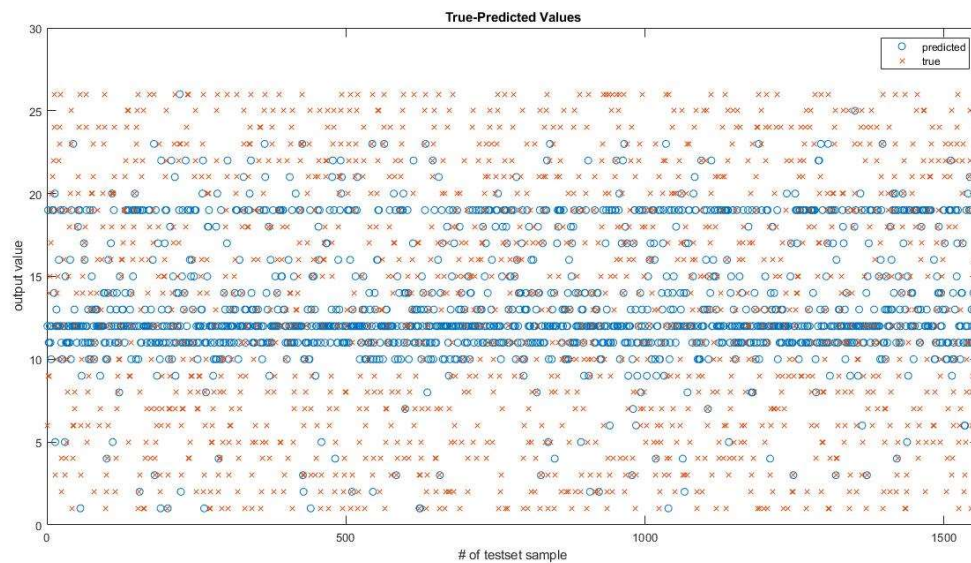
Εικόνα 8: Διαγράμματα μάθησης



Εικόνα 9: Πίνακας Σφαλμάτων Ταξινόμησης

1											5	16	26	6	3	1				3									
2											7	20	31	1						1									
3	8	8	9	8	7	5			3	4	2	1	2	2					1										
4									1	4	15	19	11	7		2	1												
5											2	15	24	4		1	3	2		9									
6										1	4	10	25	5	7	1	1		1	2				1	1				
7								2	1	6	9	10	5	11	5	5	1	1	1	3									
8								1			4	17	23	7	3	1				4									
9											4	12	31	2	1	3	1		1	5									
10									3	5	2	14	20	6	5	1	2	1	1										
11									1	2	6	7	21	11	7	1		1	1	1	1	1							
12										3	17	28	4		1	1	1		2	2			1						
13										6	15	23	5	2	1	2			5										
14										2	13	32	4	1	1		1	1	3		1	1							
15										1	10	31	8	3				1	1	5									
16										3	10	15	10	4	6	1	4	1	6										
17										2	4	6	4	12	6	4	4	5	7	2	2	2							
18										3	11	24	6		1	1	1		12										
19										4	17	27	1	3		1			6	1									
20										2	3	6	4	7	4	4	9	9	8	4									
21										1	1	1	5	5	1		3		43										
22										3	4		3		3	4	4	4	30	1	1	2		1					
23								1			9	9	20		1	1	1	1	7	4	1	3						1	1
24												17	36	1	2			1	3										
25											1		1					1	56	1									
26															1				27	5	5	8	13						
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26			

Εικόνα 10: Πρόβλεψη κλάσης - πραγματική κλάση



Από τα παραπάνω πίνακα και διάγραμμα μπορεί να παρατηρηθεί ότι ο ταξινομητής κατατάσσει με μεγαλύτερη συχνότητα τα δείγματα ελέγχου στις κλάσεις 10 – 14 και 19 από ότι στις υπόλοιπες κλάσεις, είναι δηλαδή πολωμένος προς τιμές αυτές.