



## **Πτυχιακή εργασία**

**Τμήμα:** Πανεπιστήμιο Θεσσαλίας

Σχολή Θετικών Επιστημών

Τμήμα Πληροφορικής και Τηλεπικοινωνιών

**Θέμα:** Μεθοδολογίες Ασαφούς Μοντελοποίησης για εξαγωγή γνώσης για την αυτόματη εξαγωγή ερμηνεύσιμων ασαφών κανόνων από δεδομένα και νευρωνικών δικτύων για επεξεργασία σπινθηρογραφικών εικόνων.

**Φοιτήτρια:** Άννα Φελέκη

**Επιβλέπων καθηγητής:** Σταμούλης Γεώργιος

**Συνεπιβλέπουσα καθηγήτρια:** Ελπινίκη Παπαγεωργίου

Λαμία 2019

## Ευχαριστίες

Σε αυτό το σημείο θέλω να ευχαριστήσω την καθηγήτρια μου κα. Ελπινίκη Παπαγεωργίου, Αναπληρώτρια Καθηγήτρια του Πανεπιστημίου Θεσσαλίας, για την επιστημονική της καθοδήγηση στις τεχνικές επεξεργασίας και ανάλυσης δεδομένων και ιατρικών εικόνων, για τη βοήθεια της καθ' όλη τη διάρκεια της πτυχιακής με ουσιαστικές συμβουλές καθώς και για το τόσο ενδιαφέρον θέμα που επέλεξε και που διάλεξε εμένα να το αναπτύξω. Επίσης να ευχαριστήσω τον Καθηγητή του Τμήματος κ. Γεώργιο Σταμούλη, για την βοήθεια και το ενδιαφέρον του στα πλαίσια της παρούσας διπλωματικής εργασίας.

Επίσης θα ήθελα να ευχαριστήσω τον κ. Παπανδριανό Νικόλαο, Πυρηνικό Ιατρό και Επίκουρο Καθηγητή του Τμήματος Νοσηλευτικής του Πανεπιστημίου Θεσσαλίας, για τη συλλογή και διάθεση των σπινθηρογραφικών εικόνων, για την πολύτιμη βοήθειά του στην προεπεξεργασία τους και στη διάγνωσή τους, καθώς και για την αξιολόγηση των αποτελεσμάτων που προέκυψαν.

Τέλος, ευχαριστώ την οικογένεια μου που ήταν πάντα δίπλα μου για την απόκτηση του πτυχίου, αλλά και γενικά στη ζωή μου και για την στήριξη, που πάντα μου προσφέρουν.

«Με ατομική μου ευθύνη και γνωρίζοντας τις κυρώσεις (1), που προβλέπονται από της διατάξεις της παρ. 6 του άρθρου 22 του Ν. 1599/1986, δηλώνω ότι:

1. Δεν παραθέτω κομμάτια βιβλίων ή άρθρων ή εργασιών άλλων αυτολεξεί χωρίς να τα περικλείω σε εισαγωγικά και χωρίς να αναφέρω το συγγραφέα, τη χρονολογία, τη σελίδα. Η αυτολεξεί παράθεση χωρίς εισαγωγικά χωρίς αναφορά στην πηγή, είναι λογοκλοπή. Πέραν της αυτολεξεί παράθεσης, λογοκλοπή θεωρείται και η παράφραση εδαφίων από έργα άλλων, συμπεριλαμβανομένων και έργων συμφοιτητών μου, καθώς και η παράθεση στοιχείων που άλλοι συνέλεξαν ή επεξεργάστηκαν, χωρίς αναφορά στην πηγή. Αναφέρω πάντοτε με πληρότητα την πηγή κάτω από τον πίνακα ή σχέδιο, όπως στα παραθέματα.

2. Δέχομαι ότι η αυτολεξεί παράθεση χωρίς εισαγωγικά, ακόμα κι αν συνοδεύεται από αναφορά στην πηγή σε κάποιο άλλο σημείο του κειμένου ή στο τέλος του, είναι αντιγραφή. Η αναφορά στην πηγή στο τέλος π.χ. μιας παραγράφου ή μιας σελίδας, δεν δικαιολογεί συρραφή εδαφίων έργου άλλου συγγραφέα, έστω και παραφρασμένων, και παρουσίασή τους ως δική μου εργασία.

3. Δέχομαι ότι υπάρχει επίσης περιορισμός στο μέγεθος και στη συχνότητα των παραθεμάτων που μπορώ να εντάξω στην εργασία μου εντός εισαγωγικών. Κάθε μεγάλο παράθεμα (π.χ. σε πίνακα ή πλαίσιο, κλπ), προϋποθέτει ειδικές ρυθμίσεις, και όταν δημοσιεύεται προϋποθέτει την άδεια του συγγραφέα ή του εκδότη. Το ίδιο και οι πίνακες και τα σχέδια

4. Δέχομαι όλες τις συνέπειες σε περίπτωση λογοκλοπής ή αντιγραφής.

(1) «Όποιος εν γνώσει του δηλώνει ψευδή γεγονότα ή αρνείται ή αποκρύπτει τα αληθινά με έγγραφη υπεύθυνη δήλωση του άρθρου 8 παρ. 4 Ν. 1599/1986 τιμωρείται με φυλάκιση τουλάχιστον τριών μηνών. Εάν ο υπαίτιος αυτών των πράξεων σκόπευε να προσπορίσει στον εαυτόν του ή σε άλλον περιουσιακό όφελος βλάπτοντας τρίτον ή σκόπευε να βλάψει άλλον, τιμωρείται με κάθειρξη μέχρι 10 ετών.»

Περίληψη με λέξεις κλειδιά

Στα πλαίσια της παρούσας διπλωματικής εργασίας, μελετήθηκαν διάφορες μεθοδολογίες ασαφούς λογικής για την εξαγωγή γνώσης από δεδομένα σε μορφή ερμηνεύσιμων ασαφών κανόνων, καθώς και μεθοδολογίες συνελκτικών νευρωνικών δικτύων για ανάλυση και επεξεργασία σπινθηρογραφικών εικόνων, whole-body.

Πιο συγκεκριμένα, στο πρώτο μέρος της πτυχιακής παρουσιάστηκαν οι μεθοδολογίες προεπεξεργασίας και ανάλυσης δεδομένων για ένα πολύ ενδιαφέρον αλλά ταυτόχρονα δύσκολο πρόβλημα ταξινόμησης που αφορά την οστεοαρθρίτιδα. Για την ταξινόμηση των δεδομένων σε δυο και τρεις κατηγορίες αντίστοιχα, χρησιμοποιήθηκε το εργαλείο Keel, το οποίο περιέχει έτοιμους αλγορίθμους προεπεξεργασίας και ανάλυσης δεδομένων με ασαφείς τεχνικές. Στόχος της ανάλυσης αυτής είναι η εξαγωγή ασαφών κανόνων καθώς και η ακρίβεια της ταξινόμησης των δεδομένων οστεοαρθρίτιδας που χρησιμοποιήθηκαν.

Στο δεύτερο μέρος συλλέχτηκαν από πυρηνικό ιατρό και παρουσιάστηκαν δεδομένα ιατρικών εικόνων που αφορούν σπινθηρογραφήματα, όπου αναπαριστάται η ακτινογραφία ανθρώπων, που τους έχει χορηγηθεί ραδιενεργό υγρό για τη διαδικασία της σπινθηρογραφικής απεικόνισης του σκελετού. Χρησιμοποιήθηκαν συνελκτικά νευρωνικά δίκτυα για την κατηγοριοποίηση τους, καθώς και το εργαλείο Gimp για την προεπεξεργασία των σπινθηρογραφικών εικόνων. Ο σκοπός της έρευνας ερευνητικής μελέτης που έγινε στα πλαίσια αυτής της εργασίας είναι να δημιουργηθεί ένα ικανό μοντέλο, το οποίο να είναι σε θέση να έχει καλή απόδοση.

Λέξεις-κλειδιά: Οστεοαρθρίτιδα, σπινθηρογράφημα, κατηγοριοποίηση

Σελίδα μονή με **περίληψη στην αγγλική γλώσσα** (Abstract) και λέξεις κλειδιά

Within the framework of the present thesis many different methodologies have been studied about fuzzy logic for mining information from data in the form of fuzzy rule and also methodologies of convolutional neural networks for analysis and preprocessing scintigraphy whole body images.

Specifically, at the first part are presented preprocessing methodologies and data analysis for a very interesting and difficult classification problem about osteoarthritis. Keel was used for classification of data in two and three respectively, which has built-in preprocessing and analyzing algorithms with fuzzy techniques. The aim of this analysis is fuzzy rule mining and the accuracy of classification of osteoarthritis data.

In the second part, medical image were gathered by nuclear doctor, where human actinography is presented where they have been injected with radioactive liquid for the procedure of skeletal scintigraphy. Convolutional neural networks used for classification and Gimp for preprocessing. The aim of the research is to build a capable model, which has good accuracy.

Keywords: Osteoarthritis, scintigraphy, classification

## Περιεχόμενα

1. Εισαγωγή	
1.1. Τί είναι η κατηγοριοποίηση (classification) και πώς λειτουργεί.....	8
1.2. Μεθοδολογίες Κατηγοριοποίησης.....	9
1.3. Αξιολόγηση Κατηγοριοποίησης.....	12
1.3.1. Απόδοση.....	12
1.3.2. Πίνακας Σύγκρισης.....	12
1.3.3. Ταχύτητα.....	13
1.3.4. Επεκτασιμότητα.....	13
1.3.5. Ανθεκτικότητα.....	14
1.4. Ασαφής Λογική και Ασαφής Κατηγοριοποίηση.....	14
1.5. Χρήση της ασαφούς κατηγοριοποίησης (fuzzy classification) σε μεγάλα δεδομένα.....	15
2. Δεδομένα οστεοαρθρίτιδας	
2.1. Περιγραφή δεδομένων.....	17
2.2. Τί είναι τα μη-ισορροπημένα δεδομένα.....	18
2.3. Προ-Επεξεργασία δεδομένων.....	19
3. Εργαλείο Ταξινόμησης Keel	
3.1. Περιγραφή του Keel.....	20
3.2. Πώς κάνουμε εισαγωγή τα δεδομένα οστεοαρθρίτιδας.....	20
4. Υλοποίηση αλγορίθμων ταξινόμησης	
4.1. Αποτελέσματα.....	32
4.2. Σύγκριση αποτελεσμάτων.....	38
4.3. Παρουσίαση με έτοιμα δεδομένα από το Keel και εξαγωγή ασαφών κανόνων.....	66
5. Ταξινόμηση ιατρικών εικόνων με χρήση Convolutional Neural Nets	
5.1. Δεδομένα εικόνας.....	84
5.2. Τί είναι η προ-επεξεργασία εικόνας (image preprocessing).....	86
5.3. Τί είναι η τμηματοποίηση εικόνας και εφαρμογή της (image segmentation η εξαγωγή (feature extraction) και επιλογή (feature selection) χαρακτηριστικών από εικόνα.....	90
6. Βαθιά μάθηση (Deep Learning)	

6.1. Τί είναι η Βαθιά Μάθηση.....	99
6.2. Τί είναι τα convolutional neural networks.....	101
6.3. Εφαρμογή τους σε ιατρικές εικόνες.....	105
6.4. Αποτελέσματα.....	108
7. Σύνοψη-Συμπεράσματα.....	113
8. Βιβλιογραφία.....	115
9. Παράρτημα.....	118

## 1.Εισαγωγή

### 1.1. Τί είναι η κατηγοριοποίηση (classification) και πώς λειτουργεί

Η έννοια κατηγοριοποίηση συναντάται συνήθως στην μηχανική μάθηση και αναφέρεται στο πρόβλημα στο να προσδιοριστεί μια νέα παρατήρηση σε πιο σύνολο κατηγοριών ανήκει, με βάση ένα σετ δεδομένων. Το μοντέλο κατηγοριοποίησης έχει εκπαιδευτεί στο να εντοπίζει μοτίβα στα δεδομένα για να μπορεί να ταξινομήσει τα νέα δείγματα, όταν χρειαστεί. Τέτοια παραδείγματα βλέπουμε καθημερινά όπως ο διαχωρισμός των emails με βάση το περιεχόμενο τους ή τον τίτλο, κατηγοριοποίηση πελατών, ανάλογα αν δικαιούνται δάνειο ή όχι με βάση το ιστορικό τους.

Η μέθοδος της κατηγοριοποίησης ανήκει στην επιβλεπόμενη μάθηση, δηλαδή έχει δημιουργηθεί ένα μοντέλο, το οποίο στηρίχτηκε σε δεδομένα ολοκληρωμένα, που τους έχει αποδοθεί ετικέτα, από ένα πεπερασμένο διακριτό σύνολο. Υπάρχει και μέθοδος χωρίς επίβλεψη, η οποία είναι ομαδοποίηση (clustering), όπου εκεί το σετ δεδομένων δεν έχει εξόδους, απλά ομαδοποιούνται τα δεδομένα που έχουν κοινά χαρακτηριστικά.

Ο αλγόριθμος, που υλοποιεί τη μέθοδο αυτή ονομάζεται κατηγοριοποιητής και αναφέρεται στην μαθηματική συνάρτηση, που χαρτογραφεί την εισαγωγή δεδομένων σε μια κατηγορία.

Η κατηγοριοποίηση λειτουργεί ως εξής:

Το πρώτο βήμα της κατηγοριοποίησης είναι να δημιουργηθεί το μοντέλο οπότε σύμφωνα με τα δεδομένα που έχουμε που είναι προκατηγοριοποιημένα αποφασίζει ο προγραμματιστής πόσα θα χρησιμοποιηθούν για δεδομένα εκπαίδευσης (training data) και πόσα για δοκιμαστικά δεδομένα (testing data), συνήθως το ποσοστό είναι 70% τα δεδομένα εκπαίδευσης και 30% τα δοκιμαστικά. Τα δεδομένα γενικά είναι αρκετά δείγματα τα οποία περιλαμβάνουν τα χαρακτηριστικά τους όπου είναι η είσοδος του μοντέλου και σαν έξοδο έχουν την κλάση από τις καθορισμένες του προβλήματος που ανήκουν. Καθώς το μοντέλο αυτό περιέχει τα δεδομένα αυτά και μας βοηθούν στο να κατηγοριοποιήσουμε νέα δεδομένα βρισκόμαστε στην μέθοδο επιβλεπόμενης μάθησης (supervised learning). Στη συνέχεια ο αλγόριθμος διασχίζει αυτά τα δεδομένα για να μάθει το μοτίβο που κυριαρχεί και συγκεκριμένα να καταλάβει τις συσχετίσεις των χαρακτηριστικών και κατά πόσο βοηθούν στην απόφαση για την έξοδο. Το μοντέλο, που λέγεται και αλλιώς κατηγοριοποιητής



(classifier), αναπαρίσταται με τη μορφή κανόνων κατηγοριοποίησης (classification rules), δέντρων απόφασης (decision trees) ή μαθηματικών τύπων.

Για να δούμε όμως την απόδοση του μοντέλου, το οποίο πηγάζει από το πόσο σωστά και σχετικά είναι τα δεδομένα εκπαίδευσης χρειάζεται να χρησιμοποιήσουμε τα δοκιμαστικά δεδομένα (test data). Χρησιμοποιώντας τα δεδομένα αυτά τα περνάμε μέσα από το μοντέλο χωρίς την έξοδο που τη γνωρίζουμε και αφήνουμε το μοντέλο να προβλέψει μια έξοδο για το κάθε δείγμα. Έχοντας τα αποτελέσματα του μοντέλου τα συγκρίνουμε με τα πραγματικά και βλέπουμε την απόδοση του. Αν η απόδοση είναι αρκετά υψηλή τότε είναι έτοιμο να κατηγοριοποιήσει νέα δεδομένα που δεν ξέρουμε την έξοδο (unsupervised learning).

## 1.2.Μεθοδολογίες Κατηγοριοποίησης

Πριν προχωρήσουμε στις μεθοδολογίες κατηγοριοποίησης πρέπει να αναφερθεί ότι στην επιβλεπόμενη μάθηση ανήκουν εκτός από την κατηγοριοποίηση και η παλινδρόμηση (Regression). Η κατηγοριοποίηση και η παλινδρόμηση έχουν πολλές ομοιότητες, καθώς και στις δύο περιπτώσεις ο στόχος είναι να προβλεφθεί όσο πιο σωστά γίνεται η έξοδος. Επιπλέον, και στις δύο περιπτώσεις χρησιμοποιείται ένα σύνολο δεδομένων, το οποίο είναι η πηγή για την κατασκευή μοντέλου. Η διαφορά ανάμεσα στην κατηγοριοποίηση και παλινδρόμηση είναι στον τύπο εξόδου, δηλαδή της εξαρτημένης μεταβλητής. Συγκριμένα ο τύπος αυτός είναι αριθμητικές τιμές συνεχόμενες, ενώ η κατηγοριοποίηση έχει διακριτές τιμές.

Οι μεθοδολογίες της κατηγοριοποίησης στη Μηχανική Μάθηση είναι:

- ✓ Linear Classifiers:
  - 1) Logistic Regression
  - 2) Naïve Bayes Classifier
- ✓ Support Vector Machines
- ✓ Decision Trees
- ✓ Random Forest
- ✓ Neural network
- ✓ Nearest Neighbor

Αναλυτικά για την κάθε μέθοδο έχουμε:

## Linear Classifiers

### (a) Logistic Regression

Η μέθοδος της Λογιστικής Παλινδρόμησης είναι μια μέθοδος, η οποία χρησιμοποιείται στην στατιστική και μας βοηθάει στην ανάλυση δεδομένων, όπου υπάρχουν περισσότερες από μια ανεξάρτητες μεταβλητές, τα χαρακτηριστικά δηλαδή, που καθορίζουν το αποτέλεσμα. Η έξοδος ανήκει σε σύνολο δύο τιμών και ο στόχος του αλγορίθμου αυτού είναι να βρει το καλύτερο μοντέλο που ταιριάζει, έτσι ώστε να περιγράψει την σχέση ανάμεσα στις εξαρτημένες και ανεξάρτητες μεταβλητές.

### (b) Naïve Bayes Classifiers

Με λίγα λόγια ο κατηγοριοποιητής Bayes υποθέτει ότι η παρουσία ενός συγκεκριμένου χαρακτηριστικού σε μια κλάση δεν έχει σχέση με την παρουσία των υπολοίπων χαρακτηριστικών που είναι παρόντα. Ακόμη και αν αυτά τα χαρακτηριστικά βασίζονται το ένα στο άλλο, όλες αυτές οι ιδιότητες συμμετέχουν ανεξάρτητα. Ο αλγόριθμος αυτός είναι εύκολο στην κατασκευή και ιδιαίτερα χρήσιμος για μεγάλα σετ δεδομένων. Επίσης είναι γνωστό ότι ξεπερνάει αρκετές σοφιστικές μεθόδους.

## Decision Trees

Τα δέντρα αποφάσεως δημιουργούν μοντέλα κατηγοριοποίησης ή οπισθοδρόμησης (classification or regression) σε μορφή δέντρου. Διασπάει τα δεδομένα σε μικρότερα υπο-σετ δεδομένων και με αυτό τον τρόπο δημιουργείται το δέντρο. Το τελικό αποτέλεσμα είναι ένα δέντρο με κόμβους απόφασης και φύλλα. Ο κόμβος απόφασης έχει δύο ή παραπάνω διακλαδώσεις και τα φύλλα αναπαριστούν την απόφαση του μοντέλου. Ο αλγόριθμος αυτός διαχειρίζεται αριθμητικά και κατηγορικά δεδομένα.

## Random Forest

Ο αλγόριθμος αυτός είναι μέθοδος που εφαρμόζεται σε όλες τις κατηγορίες προβλημάτων (classification, regression κ.ά.), το οποίο γίνεται κατασκευάζοντας ένα πλήθος από δέντρα απόφασης, οπότε δημιουργείται ένα δάσος, κατά την ώρα της εκπαίδευσης και βγάζουν σαν έξοδο την κλάση.

## Neural Network

Ένα νευρωνικό δίκτυο αποτελείται από μονάδες (νευρώνες) στοιχισμένες σε επίπεδα (layers), τα οποία μετατρέπουν μια είσοδο, η οποία βρίσκεται σε μορφή πίνακα σε κάποια έξοδο. Κάθε νευρώνας παίρνει μια είσοδο εφαρμόζει (συνήθως μη γραμμική) συνάρτηση σε αυτόν και περνάει την έξοδο στο επόμενο επίπεδο. Γενικά τα δίκτυα είναι μπροστινής κατεύθυνσης: ο νευρώνας τροφοδοτεί την έξοδο του σε όλους τους νευρώνες του επόμενου επιπέδου, αλλά δεν υπάρχει καμία ανατροφοδότηση για το προηγούμενο επίπεδο. Τα βάρη εφαρμόζονται στα σήματα που δημιουργούνται ανάμεσα στα επίπεδα και αυτά τα βάρη προσαρμόζονται στην φάση της εκπαίδευσης, έτσι ώστε να προσαρμοστεί το νευρωνικό δίκτυο στο συγκεκριμένο πρόβλημα.

## Nearest Neighbor

Ο αλγόριθμος του κ-κοντινότερου γείτονα είναι ένα πρόβλημα κατηγοριοποίησης και είναι με επίβλεψη (Supervised). Παίρνει πολλά σημεία με ετικέτα (labeled points) και τα χρησιμοποιεί έτσι ώστε να δοθεί ετικέτα στα άλλα σημεία. Για να γίνει αυτό κοιτάει στα ήδη σημεία με ετικέτα, τα οποία βρίσκονται κοντά στο νέο σημείο τους γείτονες του δηλαδή, και βάζει τους γείτονες να αποφασίσουν ποια ετικέτα ταιριάζει. Το κ- είναι ο αριθμός των γειτόνων που γίνεται ο έλεγχος.

## 1.3. Αξιολόγηση κατηγοριοποίησης

### 1.3.1. Απόδοση

Η απόδοση είναι μια μετρική για να αξιολογούμε τα μοντέλα κατηγοριοποίησης. Με λίγα λόγια είναι ο παράγοντας προβλέψεων που πέτυχε σωστά το μοντέλο μας. Ο ορισμός είναι:

$$\text{Απόδοση} = \frac{\text{Αριθμός σωστών προβλέψεων}}{\text{Συνολικός αριθμός προβλέψεων}}$$

### 1.3.2. Πίνακας σύγχυσης (Confusion matrix)

Ο πίνακας σύγχυσης είναι ένας πίνακας που επιτρέπει την εμφάνιση της απόδοσης του αλγορίθμου συνήθως σε εποπτική μάθηση (supervised learning) και στην μη εποπτική μάθηση (unsupervised learning) λέγεται πίνακας αντιστοίχισης (matching matrix).

Κάθε γραμμή του πίνακα αναπαριστά τις περιπτώσεις της προβλεπόμενης κλάσης, ενώ κάθε στήλη αναπαριστά τις περιπτώσεις της σωστής κλάσης (ή το αντίθετο). Το όνομα προέρχεται, από το γεγονός ότι είναι ευκρινές να δούμε αν ο αλγόριθμος «συγχύζεται» ανάμεσα στις δύο κλάσεις.

$$\text{Απόδοση} = \frac{TP + TN}{TP + TN + FP + FN}$$

Όπου το TP= True Positive, TN = True Negatives, FP = False Positives and FN = False Negatives

Η αναπαράσταση είναι η εξής με ένα παράδειγμα:

	Actual Class			
Predicted Class		Cat	Dog	Rabbit
	Cat	5	2	0
	Dog	3	3	2
	Rat	0	1	11

*Πίνακας 1*

Εάν ένα σύστημα κατηγοριοποίησης έχει εκπαιδευτεί να διαχωρίσει ανάμεσα σε σκύλο, γάτα και λαγό, ο πίνακας σύγχυσης συγκεντρώνει τα αποτελέσματα του έλεγχου του αλγορίθμου. Ας υποθέσουμε ότι έχουμε 27 δείγματα, 8 γάτες, 6 σκύλους και 13 λαγούς. Σε αυτό το παράδειγμα 8 από τις γάτες, το σύστημα τις πρόβλεψε σαν γάτες, 2 σαν σκύλους και 0 σαν λαγούς και ούτε καθεξής. Βλέπουμε το πρόβλημα του συστήματος κατηγοριοποίησης είναι στους σκύλους και τις γάτες. Οι σωστές προβλέψεις βρίσκονται στη διαγώνιο γραμμή με έντονα γράμματα.

### 1.3.3. Ταχύτητα

Σχετίζεται με την πολυπλοκότητα της μεθόδου και το υπολογιστικό κόστος που αυτή συνεπάγεται. Η εκτέλεση περίπλοκων αλγορίθμων, οι οποίοι απαιτούν εκτεταμένους υπολογισμούς, προκαλούν καθυστερήσεις. Οι καθυστερήσεις αυτές μπορεί να υπάρχουν στη διαδικασία κατασκευής, αλλά και στη χρήση των μοντέλων, στην εφαρμογή τους δηλαδή για την κατηγοριοποίηση μιας νέας παρατήρησης.

### 1.3.4. Επεκτασιμότητα

Η έννοια επεκτασιμότητα αναφέρεται στην ικανότητα των μεθόδων να χειριστούν πολύ μεγάλα σύνολα δεδομένων. Η μηχανική μάθηση προσφέρει αρκετές μεθόδους κατηγοριοποίησης, αλλά αρκετές από αυτές δεν είναι εύκολα διαχειρίσιμες στην επεξεργασία δεδομένων μεγάλου όγκου. Δημιουργούν προβλήματα στην υπολογιστική πολυπλοκότητα και τα ζητήματα αυτά θέτουν όρια στη δυνατότητα εφαρμογής των μεθόδων. Στη σημερινή εποχή η παραγωγή και καταγραφή δεδομένων είναι μαζικότερη και για αυτό το λόγο οι

εφαρμογές εξόρυξης και επεξεργασίας δεδομένων πρέπει να ανταποκρίνονται στο ανάλογο μέγεθος δεδομένων.

### **1.3.5. Ανθεκτικότητα**

Αναφέρεται στην ικανότητα των μεθόδων κατηγοριοποίησης να κάνουν τις απαραίτητες προβλέψεις, όταν τα δεδομένα περιέχουν θόρυβο και χαμένες τιμές.

## **1.4. Ασαφής Λογική και Ασαφής Κατηγοριοποίηση**

Η ασαφής λογική είναι μια μορφή λογικής, η οποία παίρνει πολλές τιμές και ο διαχωρισμός κλάσεων περιέχει σύνολα αριθμών. Περιέχει μη αυστηρούς μηχανισμούς εξαγωγής συμπερασμάτων και πλησιάζει την έννοια της ανθρώπινης λογικής, καθώς οι άνθρωποι παίρνουν αποφάσεις ανακριβείς και με αβεβαιότητα κάποιες φορές και δεν βασίζονται συνήθως σε αριθμητικά δεδομένα. Ένα παράδειγμα είναι σχετικά με το ύψος. Δεν μπορούμε να καθορίσουμε από ποιο ύψος και πάνω καθορίζεται ο ψηλός με τον κοντό. Αν αυτό το όριο ήταν 1.70 δεν μπορούμε να πούμε ότι ο άνθρωπος με ύψος 1.69 είναι κοντός, ενώ αυτός που είναι 1.71 θεωρείται ψηλός, ενώ η διαφορά τους είναι μόλις 2 εκατοστά. Το πλεονέκτημα της ασαφούς λογικής είναι ότι μπορεί να επεξεργαστεί και να αναλύσει συστήματα, που είναι αρκετά πολύπλοκα.

Τώρα ας περάσουμε στην έννοια της Ασαφής κατηγοριοποίησης.

Όπως έχει αναφερθεί και παραπάνω οι συνήθεις μέθοδοι κατηγοριοποίησης προσπαθούν να μειώσουν τη διάσταση ενός περίπλοκου και σει δεδομένων ομαδοποιώντας τα δεδομένα σε διακριτές κλάσεις, δηλαδή θα ανήκουν στην κλάση ή όχι.

Η ασαφής κατηγοριοποίηση λειτουργεί κάπως διαφορετικά, καθώς κάθε δείγμα από τα δεδομένα μπορεί να ανήκει σε πολλές κλάσεις, αλλά με διαφορετικό ποσοστό στην καθεμία. Το συνολικό ποσοστό όμως για κάθε δείγμα ισούται με 100. Τα δεδομένα που εφάπτονται στην έννοια της ασαφής κατηγοριοποίησης είναι συνεχόμενα.

## 1.5. Χρήση της ασαφούς κατηγοριοποίησης (fuzzy classification) σε μεγάλα δεδομένα

Αξίζει να σημειωθεί ότι τα Μεγάλα Δεδομένα (Big Data) αποτελούν δεδομένα, όπου η πολυπλοκότητα τους και ο όγκος τους που έχουν χρειάζονται ειδικές τεχνικές και μεθόδους για να εξαχθεί χρήσιμη πληροφορία. Με τους συνήθεις αλγόριθμους δεν θα προκύψει αξιόπιστο αποτέλεσμα. Μπορούμε να πούμε ότι ένα σετ δεδομένων περιέχεται στην κατηγορία Big Data, από τη στιγμή που αναπαριστά πρόκληση για να το επεξεργαστούμε με παραδοσιακές εφαρμογές. Τα Μεγάλα Δεδομένα ορίζονται από τα 5 V

Volume (Όγκος)

Variety (Ποικιλία)

Velocity (Ταχύτητα)

Veracity (Εγκυρότητα)

Value (Τιμή)

Ένα νέο προγραμματιστικό μοντέλο για τα μεγάλα δεδομένα στα καταναμημένα συστήματα είναι το MapReduce. Ας αναφέρουμε λίγα λόγια για την τεχνική αυτή. Είναι ένα προγραμματιστικό μοντέλο για επεξεργασία μεγάλων δεδομένων με παράλληλο και καταναμημένο αλγόριθμο. Διασπάται σε δύο φάσεις, η πρώτη είναι map, δηλαδή ο σχεδιασμός και η ταξινόμηση των δεδομένων. Η δεύτερη είναι το reduce, η ελάττωση κόμβων. Το σύστημα MapReduce οργανώνει τη διαδικασία ταξινόμησης των καταναμημένων servers, τρέχει διάφορα έργα παράλληλα, διαχειρίζεται όλες τις επικοινωνίες και τα δεδομένα μετακινούνται στα διάφορα συστήματα και παρέχει ανοχή σε σφάλματα. Οι συναρτήσεις και οι λειτουργίες βρίσκονται σε όλους τους κόμβους. Κάθε κόμβος που κάνει υπολογισμό δουλεύει με τα δεδομένα που είναι αποθηκευμένα σε αυτόν. Μόνο τα αναγκαία δεδομένα μετακινούνται στο δίκτυο. Μερικές πλατφόρμες που ανήκουν στην έννοια MapReduce είναι το Hadoop, το οποίο είναι ανοικτό λογισμικό γραμμένο σε Java, καθώς και οι Spark και Flink βιβλιοθήκες.

Τα συστήματα, τα οποία είναι βασισμένα σε ασαφείς κανόνες (Fuzzy Rule Based Systems) έχουν έναν σημαντικό ρόλο, λόγω της ερμηνείας που δίνουν στις πληροφορίες και την παροχή γνώσης που δίνουν στην μορφή γλωσσικών κανόνων ή κατανοητά και απλά ασαφή

μοντέλα. Για αυτό τον λόγο πρέπει να είναι παρόντα τα συστήματα αυτά στην Επιστήμη των Δεδομένων, επειδή έχουν καλή απόδοση και είναι εύκολα στην κατανόηση.

Συγκεκριμένα η μέθοδος Chi-FRBCS παράγει κανόνες της μορφής

Rule R<sub>j</sub>: IF x<sub>1</sub> IS A<sub>1j</sub> AND ... AND x<sub>n</sub> IS A<sub>nj</sub> THEN Class = C<sub>j</sub> with RW<sub>j</sub>

Παράγει ασαφή διαχωρισμό για τις κλάσεις δημιουργώντας ασαφή κανόνα για κάθε δείγμα. Σε περίπτωση που παραχθεί ο ίδιος κανόνας διαγράφεται και από αυτά που απομένουν υπερισχύει ο κανόνας με το μεγαλύτερο βάρος.

Ένα άλλο ενδιαφέρον παράδειγμα μοντέλου είναι οι τεχνικές του πλησιέστερου γείτονα με ασαφή λογική. Η μέθοδος αυτή λειτουργεί κάνοντας ανάθεση κλάσης στα σημεία υπολογίζοντας την απόσταση στον k-πλησιέστερο γείτονα. Σε αυτό το σημείο αξίζει να σημειωθεί ότι ο αλγόριθμος k<sub>nn</sub> έχει επεκταθεί FkNN υπολογίζοντας τις ασαφείς αποστάσεις ανάμεσα στα σημεία, τα οποία ορίζουν την ασάφεια της κάθε πλειάδας. Οι αποστάσεις έχουν ασαφή βάρη.

Ένα πρόβλημα που συναντάμε συχνά στα μεγάλα δεδομένα είναι ότι υπάρχει έλλειψη δειγμάτων για τη διαδικασία της εκπαίδευσης και είναι δύσκολο για έναν αλγόριθμο να καθορίσει ένα καλό μοντέλο που θα έχει καλή απόδοση. Τότε έχουμε την περίπτωση των μη ισορροπημένων δεδομένων, που θα αναλυθεί παρακάτω και κάποιες φορές τα κατηγοριοποιημένα δεδομένα μιας κλάσης είναι τόσο λίγα που μπορεί να θεωρηθούν θόρυβος ανάμεσα στα άλλα.

Οπότε γεννάται η αβεβαιότητα, επειδή έχουμε διαφορετικές πηγές, ποικιλία στα δεδομένα, μη ολοκληρωμένα. Όμως τα FRBCS μπορούν να διαχειριστούν όλα αυτά τα προβλήματα, ακόμα και την έλλειψη δεδομένων.

Το σχέδιο MapReduce βασίζεται στον αλγόριθμο Chi, όπου η πρώτη φάση είναι η κατασκευή του ασαφούς κανόνα (Fuzzy Rule Base) και η δεύτερη φάση κάνει μια εκτίμηση για την κλάση που ανήκουν, ανάλογα με τα δεδομένα. Στην φάση του Reduce υπάρχουν δύο περιπτώσεις Chi-FRBCS-BigData-Max και Chi-FRBCS-BigData-Average, τα οποία ευθύνονται για τον υπολογισμό των βαρών. Τα ομοιογενή ασαφή τμήματα μοιράζονται από όλα τα καθήκοντα της Map φάσης. Οι κανόνες που επαναλαμβάνονται συμπύσσονται και τα βάρη επίσης. Συνήθως, η απόδοση βελτιώνεται όταν υπάρχουν λιγότερα maps, καθώς και η αύξηση των κανόνων.



## 2. Δεδομένα οστεοαρθρίτιδας

### 2.1. Περιγραφή δεδομένων

Τα δεδομένα που θα δουλέψουμε είναι σχετικά με την οστεοαρθρίτιδα του γονάτου. Η Οστεοαρθρίτιδα (ΟΑ), επίσης γνωστή ως εκφυλιστική αρθρίτιδα ή εκφυλιστική νόσος των αρθρώσεων είναι μια ομάδα μηχανικών ανωμαλιών που συνεπάγονται υποβάθμιση της αρθρώσεως, συμπεριλαμβανομένων του αρθρικού χόνδρου και υποχόνδριου οστού. Τα συμπτώματα μπορεί να περιλαμβάνουν πόνο στις αρθρώσεις, ευαισθησία, δυσκαμψία, κλείδωμα, και μερικές φορές μία διάχυση υγρού στην περιοχή της άρθρωσης. Μια ποικιλία αιτιών-κληρονομικά, αναπτυξιακά, μεταβολικά και μηχανικά- μπορεί να κινήσουν τις διαδικασίες που οδηγούν σε απώλεια του χόνδρου. Όταν οι επιφάνειες των οστών είναι λιγότερο καλά προστατευμένες από χόνδρους, τα οστά είναι δυνατόν να εκτεθούν και να καταστραφούν. Ως αποτέλεσμα της μειωμένης κίνησης λόγω πόνου, οι περιφερειακοί μύες μπορεί να ατροφήσουν και οι σύνδεσμοι μπορεί να γίνουν πιο χαλαροί.

Ας αναφέρουμε ότι τα δεδομένα αυτά είναι σε μορφή Excel (.xls).

Έχουμε 4 κατηγορίες των δεδομένων.

1. Dataset\_full: Πρόκειται για ένα dataset που περιέχει δεδομένα για την οστεοαρθρίτιδα με 4796 καταχωρήσεις και 191 χαρακτηριστικά. (Έχει και missing data, όπου θα επιλυθούν στη συνέχεια)
2. Dataset1: Είναι υπό-dataset του παραπάνω με 1122 καταχωρήσεις και 138 χαρακτηριστικά.
3. Dataset2: Απομονώνουμε επίσης τις 1727 καταχωρήσεις που πάσχουν από παχυσαρκία.
4. Dataset3: Και επίσης οι υπόλοιπες 3026 καταχωρήσεις, όπου περιέχονται οι ασθενείς που δεν πάσχουν από παχυσαρκία.

#### Έξοδος Δεδομένων:

Τα παραπάνω dataset έχουν 3 εξόδους : Progression που σημαίνει ότι υπάρχει πρόβλημα στο γόνατο, η δεύτερη έξοδος είναι Incidence, δηλαδή υπάρχει κίνδυνος να εμφανιστεί οστεοαρθρίτιδα στο γόνατο κατά τη διάρκεια της έρευνας και η τρίτη έξοδος είναι Non-exposed control group, δηλαδή ότι δεν έχουν εκτεθεί.

## 2.2. Τί είναι τα μη-ισορροπημένα δεδομένα

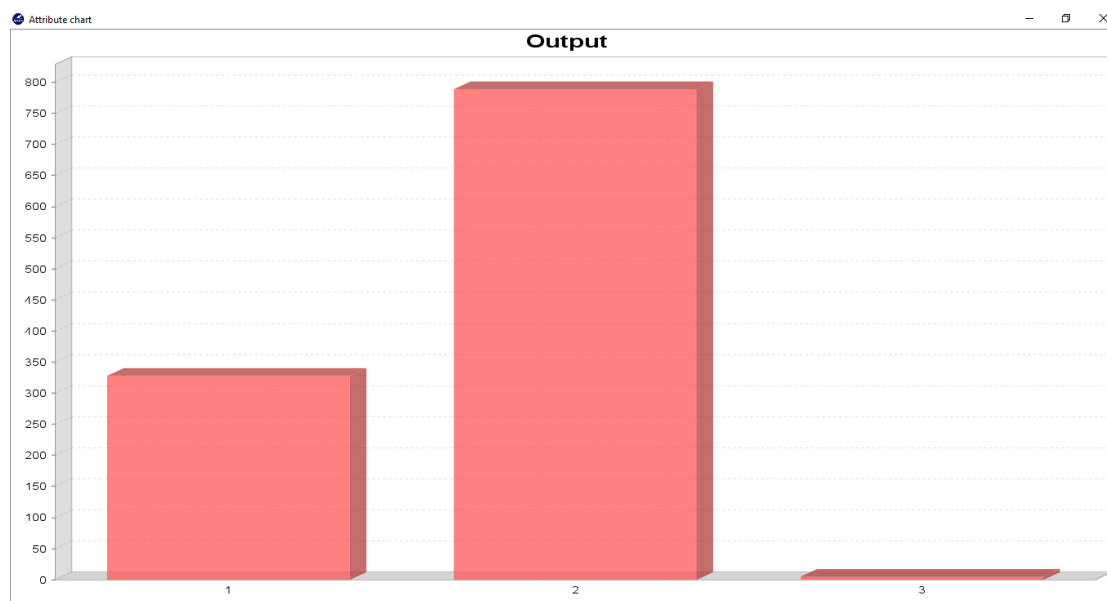
Τα μη-ισορροπημένα δεδομένα (imbalanced datasets) είναι μια ειδική περίπτωση στα προβλήματα κατηγοριοποίησης, όπου η κατανομή της κάθε κλάσης δεν είναι ομοιόμορφη ανάμεσα τους. Γενικά αυτά τα δεδομένα αποτελούνται από 2 είδη κλάσεων: η κλάση που υπερिशύει σε αριθμό δειγμάτων είναι η αρνητική (majority) και αυτή με τα υπόλοιπα ελάχιστα η θετική κλάση (minority).

Βλέπουμε ότι έχουμε περισσότερα δεδομένα για την πρώτη και δεύτερη κλάση, ενώ για την τρίτη δεν έχουμε σχεδόν καθόλου. Συγκεκριμένα είναι :

Πρώτη :328

Δεύτερη :789

Τρίτη :5



Σχήμα 1

Αυτοί οι τύποι δεδομένων είναι μια πρόκληση για την εξαγωγή γνώσης, μιας και οι αλγόριθμοι κατηγοριοποίησης δεν ισχύουν κα ως αποτέλεσμα θα δουλέψουν λανθασμένα,

καθώς θεωρούν ότι τα δεδομένα είναι ισορροπημένα και δουλεύουν μόνο με την κλάση που υπερισχύει.

### **2.3. Προ-Επεξεργασία δεδομένων**

Όπως είναι γνωστό κανένα από τα δεδομένα που θα χειριζόμαστε δεν είναι έτοιμα για εφαρμογή στους αλγορίθμους, πάντα χρειάζονται προ-επεξεργασία.

Το `dataset_full` έχει Missing Data, τα οποία επιλύθηκαν προγραμματιστικά με την γλώσσα Python και χρησιμοποιώντας την βιβλιοθήκη Pandas Dataframe. Τα δεδομένα είναι σε μορφή Excel και έγιναν εισαγωγή στο περιβάλλον της Python, όπου μετατράπηκαν σε dataframe με τη βοήθεια της βιβλιοθήκης και έγινε η απαραίτητη επεξεργασία, δηλαδή όπου υπάρχει NaN τιμή αντικαταστάθηκε από το mean της στήλης και αποθηκεύτηκε πίσω σε Excel μορφή. Στη συνέχεια έτρεξα αλγορίθμους με το dataset, αφού πρώτα το έκανα εισαγωγή στο Keel Software.

### **3.Εργαλείο Ταξινόμησης Keel**

#### **3.1. Περιγραφή του Keel**

Το εργαλείο ταξινόμησης Keel (Knowledge Extraction based on Evolutionary Learning) είναι ένα δωρεάν εργαλείο, το οποίο δίνει την πρόσβαση στον χρήστη σε εξελικτικούς αλγορίθμους (evolutionary algorithms) και σε τεχνικές για διάφορα προβλήματα εξαγωγής πληροφορίας (classification, regression, clustering, pattern mining). Τα κύρια χαρακτηριστικά του είναι ότι περιέχει ένα μεγάλο φάσμα αλγορίθμων για μοντέλα πρόβλεψης, μεθόδους προεπεξεργασίας (data transformation, discretization, instance and feature selection, noise filtering) και μεταγενέστερη επεξεργασία. Εκτός αυτού έχει μεθόδους για διαφορετικές περιοχές του τομέα εξαγωγής γνώσης, όπως δέντρα απόφασης, διακριτή εκμάθηση (crisp rule learning) και ασαφείς κανόνες. Επίσης διαθέτει στατιστική βιβλιοθήκη για να αναλύσει τα αποτελέσματα του αλγορίθμου και πρόσβαση στην αλλαγή παραμέτρων για τους αλγορίθμους, έτσι ώστε να μπορεί να τροποποιηθεί η απόδοση.

Η πλατφόρμα έχει στόχο να δημιουργήσει πειράματα περιέχοντας πολλαπλά δεδομένα και αλγορίθμους, έτσι ώστε να προκύψουν τα αποτελέσματα.

#### **3.2. Πώς κάνουμε εισαγωγή τα δεδομένα οστεοαρθρίτιδας**

Παρακάτω ακολουθεί περιγραφή πως μπορούμε να κάνουμε εισαγωγή ένα dataset στο Keel και να βγάλουμε τα αποτελέσματά μας.

Αρχικά βλέπουμε το dataset, το οποίο έχει 138 στήλες και 1122 καταχωρήσεις, το οποίο είναι στην μορφή του Excel. Βλέπουμε προς το τέλος τις στήλες για να παρατηρήσουμε ότι οι έξοδοι χρειάζονται προεπεξεργασία.

	DS	DT	DU	DV	DW	DX	DY	DZ	EA	EB	EC	ED	EE	EF	EG	EH	EI				
1	VarName125	VarName126	VarName127	VarName128	VarName129	VarName130	VarName131	VarName132	VarName133	VarName134	VarName135	VarName136	VarName137	VarName138	VarName139	VarName140	VarName141	VarName142	VarName143	VarName144	VarName1
2	0	0	0	0	0	0	0	20.9	20	0	1	3	4	3	15	0	1	Progression			
3	0	0	0	0	0	14	33	0	0	0	1	4	4	21	0	0	0	Incidence			
4	0	0	0	0	0	0	3	0	0	0	1	4	4	0	0	0	0	Incidence			
5	0	0	0	0	0	0	0	0	0	0	1	4	4	2	0	0	0	Incidence			
6	4	0	0	0	0	8	10	0	0	0	5	3	3	0	0	1	0	Incidence			
7	0	0	0	0	0	1.1	2.1	0	0	0	0	4	4	0	0	0	0	Incidence			
8	4	0	0	0	0	0	0	0	0	0	3	1	4	0	0	1	0	Incidence			
9	0	0	0	0	0	0	0	0	0	0	1	4	4	0	0	0	0	Incidence			
10	2	0	0	0	0	0	0	1	4	0	2	4	4	0	0	1	0	Incidence			
11	0	0	0	0	0	0	0	0	0	0	2	4	4	0	0	0	0	Incidence			
12	0	0	0	0	0	0	0	0	0	0	0	4	4	0	0	0	0	Incidence			
13	0	0	0	0	0	0	0	0	0	0	0	4	4	0	0	0	0	Incidence			
14	6	0	0	0	0	15	3	0	0	0	4	3	4	30	1	0	0	Progression			
15	0	0	0	0	0	0	1	0	0	0	1	4	4	0	0	0	0	Incidence			
16	5	0	0	0	0	21.6	18.6	0	0	0	2	4	4	1	1	0	0	Incidence			
17	4	0	0	0	0	17.5	33.4	10	0	1	5	1	3	10	1	1	0	Progression			
18	3	0	0	0	0	2	7	0	0	0	2	4	4	0	0	1	0	Incidence			
19	0	0	0	0	0	0	0	0	0	0	0	4	4	0	0	0	0	Incidence			
20	0	0	0	0	0	0	0	0	0	0	1	4	4	0	0	0	0	Incidence			
21	2	0	0	0	0	14.6	0	0	0	0	1	4	4	0	0	1	0	Incidence			
22	0	0	0	0	0	2.1	9.2	0	0	0	2	4	4	2	5	0	0	Incidence			
23	2	0	0	0	0	1	1	0	0	0	2	4	4	0	0	1	0	Incidence			
24	0	0	0	0	0	2.1	16.7	0	0	0	1	4	4	2	0	0	0	Incidence			
25	0	0	0	0	0	10	9	0	0	0	1	4	4	0	0	0	0	Incidence			
26	2	0	0	0	0	4	0	0	0	0	3	3	4	30	1	0	0	Progression			
27	0	0	0	0	0	0	0	0	0	0	0	4	4	0	0	0	0	Incidence			
28	6	0	0	0	0	12	13	0	0	0	5	3	3	0	1	0	0	Progression			

Σχήμα 2

Όπως καταλαβαίνουμε η έξοδος δεν μας βοηθάει για να κάνουμε το dataset Import σε classifier, οπότε άλλαξα τις Output classes σε 1,2,3

	DQ	DP	DQ	DR	DS	DT	DU	DV	DW	DX	DY	DZ	EA	EB	EC	ED	EE	EF	EG	EH	EI	EJ	EK
1	VarName12	VarName12	VarName12	VarName12	VarName12	VarName12	VarName12	VarName12	VarName12	VarName12	VarName12	VarName12	VarName12	VarName12	VarName12	VarName12	VarName12	VarName12	VarName12	VarName12	VarName12	VarName12	VarName12
2	1	3	0	2	0	0	0	0	0	0	20.9	20	0	1	3	4	3	15	0	1	1		
3	1	5	0	0	0	0	0	0	0	14	33	0	0	0	1	4	4	21	0	0	2		
4	0	3	0	2	0	0	0	0	0	0	3	0	0	0	1	4	4	0	0	1	2		
5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	4	2	0	0	0	2		
6	0	4	4	4	4	0	0	0	0	8	10	0	0	0	5	3	3	0	1	1	2		
7	0	0	0	0	0	0	0	0	0	0	1.1	2.1	0	0	0	4	4	0	0	0	2		
8	0	0	0	0	4	0	0	0	0	0	0	0	0	0	3	1	4	0	1	0	2		
9	0	0	0	3	0	0	0	0	0	0	0	0	0	0	1	4	4	0	0	1	2		
10	0	1	0	0	2	0	0	0	0	0	0	1	4	0	2	4	4	0	1	0	2		
11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	4	4	0	0	0	2		
12	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4	4	0	0	0	2		
13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4	4	0	0	0	2		
14	1	1	7	1	6	0	0	0	0	15	3	0	0	0	4	3	4	30	1	1	1		
15	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	4	4	0	0	0	2		
16	1	3	6	5	5	0	0	0	0	21.6	18.6	0	0	0	2	4	4	1	1	1	2		
17	1	5	2	8	4	0	0	0	0	17.5	33.4	10	0	1	5	1	3	10	1	1	1		
18	0	3	2	4	3	0	0	0	0	2	7	0	0	0	2	4	4	0	1	1	2		
19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4	4	0	0	0	1	2		
20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	4	4	0	0	0	2		
21	0	0	4	0	2	0	0	0	0	14.6	0	0	0	0	1	4	4	0	1	0	2		
22	1	2	0	6	0	0	0	0	0	2.1	9.2	0	0	0	2	4	2	5	0	1	2		
23	0	1	0	2	2	0	0	0	0	1	1	0	0	0	2	4	4	0	1	1	2		
24	0	3	0	4	0	0	0	0	0	2.1	16.7	0	0	0	1	4	4	0	0	1	2		
25	0	1	0	6	0	0	0	0	0	10	9	0	0	0	1	4	4	0	0	1	2		
26	1	0	1	0	2	0	0	0	0	4	0	0	0	0	3	3	4	30	1	0	1		
27	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4	4	0	0	0	0	2		
28	0	7	7	6	6	0	0	0	0	12	13	0	0	0	5	3	3	0	1	1	1		
29	0	1	2	0	0	1	0	2	0	18.9	16.5	0	0	0	1	4	4	0	0	0	2		
30	0	2	0	0	0	0	0	0	0	0	1	0	0	0	0	4	4	0	0	0	2		
31	0	4	0	4	0	0	0	0	0	11	0	0	0	0	1	4	4	0	0	1	2		
32	1	0	2.479236	0	0	0	0	0	0	15	14	0	0	0	5	1	3	30	1	1	2		
33	0	3	0	4	0	0	0	0	0	0	3	0	0	0	1	4	4	0	0	1	2		
34	1	0	8	0	5	0	0	0	0	7	8	0	0	0	3	1	4	16	1	0	2		
35	1	0	0	0	0	0	0	0	0	0	1	0	0	0	1	4	4	0	0	0	2		
36	0	5	5	6	6	0	0	0	0	10.9	15.2	0	0	0	5	3	3	0	1	1	1		
37	0	4	0	2.546246	0	0	0	0	0	0	1	0	0	0	1	4	4	0	0	0.6544	2		

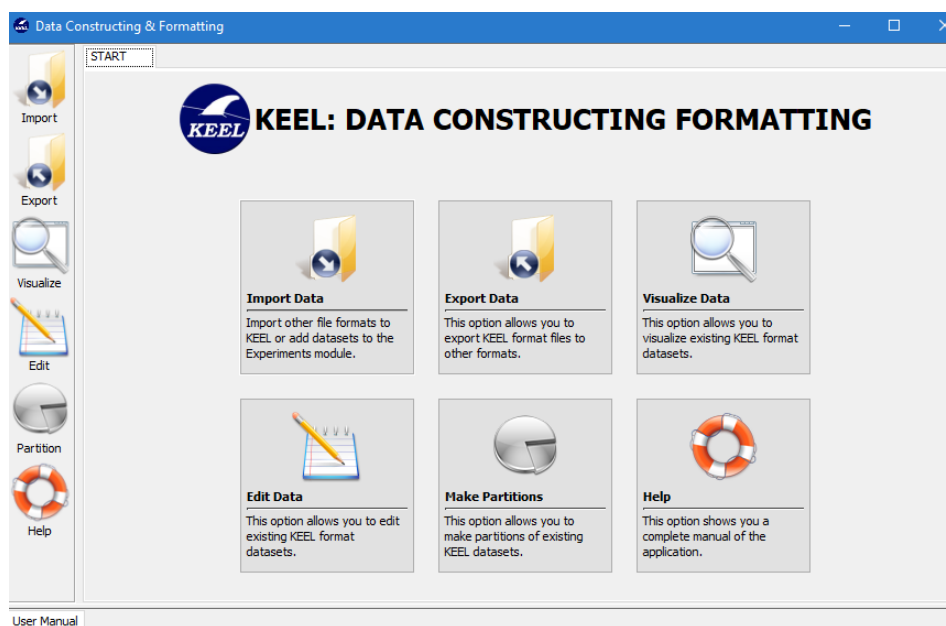
Σχήμα 3

Μετά είναι η σειρά να χρησιμοποιήσουμε το keel.

Αφού το κατεβάσουμε από την σελίδα και κάνουμε τα απαραίτητα βήματα αυτό είναι το πρώτο μενού που βλέπουμε. Και επιλέγουμε το Data Management και μετά το Import Data καθώς θέλουμε να εισάγουμε τα δεδομένα, τα οποία βρίσκονται σε excel μορφή.

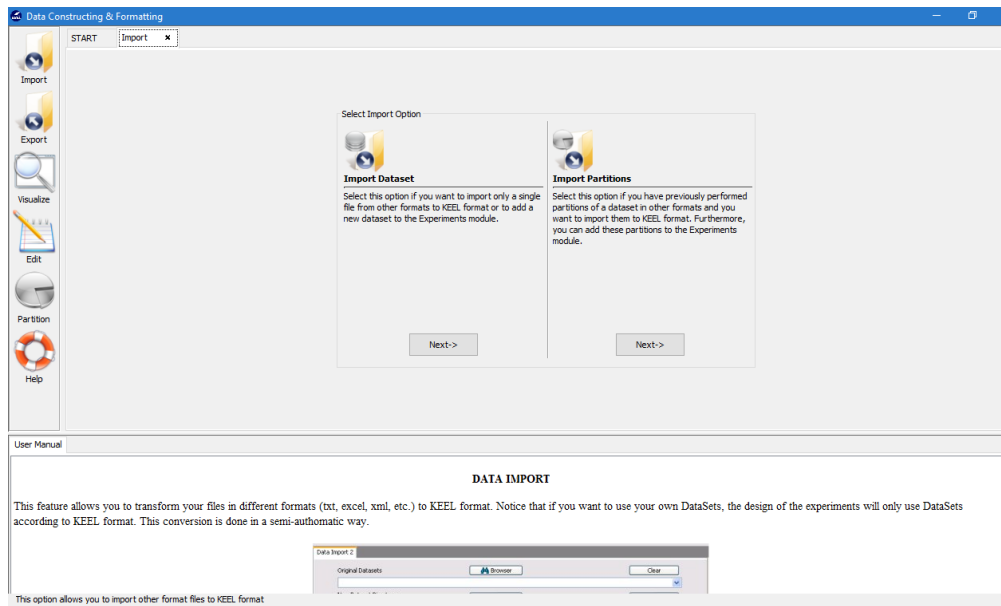


Σχήμα 4



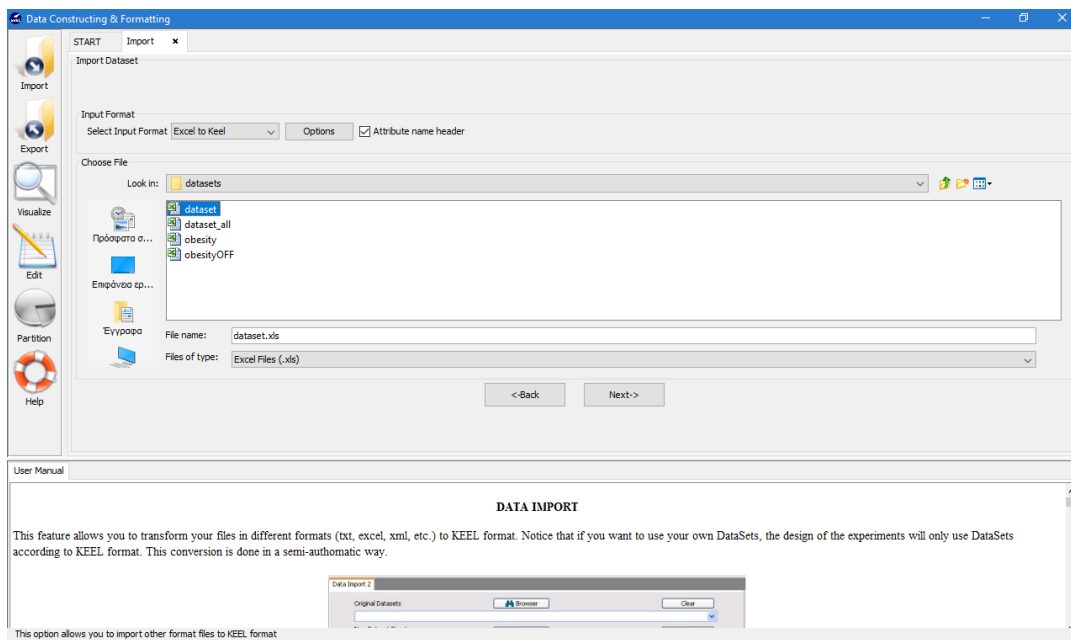
Σχήμα 5

Και στη συνέχεια κάνουμε Import Dataset, γιατί το dataset μας είναι σε ενιαία Excel μορφή, ενώ θα επιλέγαμε Import partitions αν ήταν χωρισμένο και σε keel μορφή.



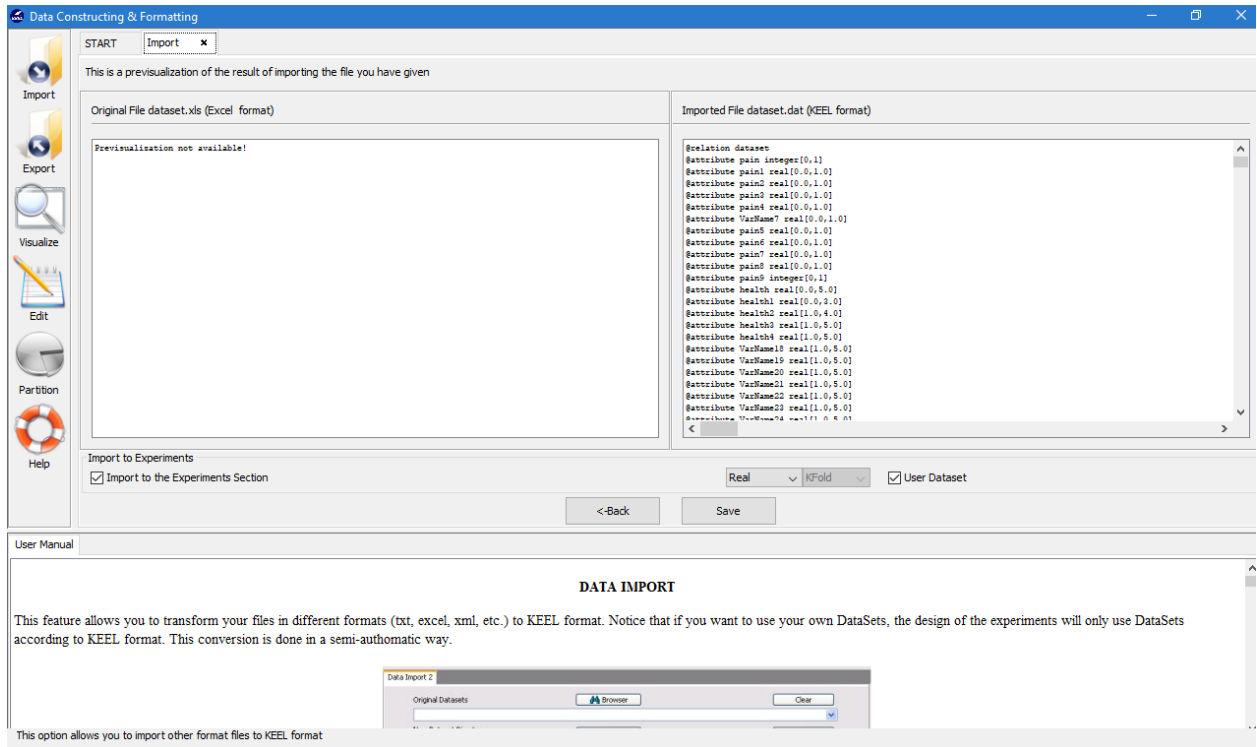
Σχήμα 6

Κάνουμε import το dataset με την μορφή excel to keel και το attribute name header clicked γιατί οι στήλες περιέχουν το όνομα των χαρακτηριστικών. Το βρίσκουμε όπου είναι αποθηκευμένο στον υπολογιστή μας και το επιλέγουμε και προχωράμε στο επόμενο βήμα.



Σχήμα 7

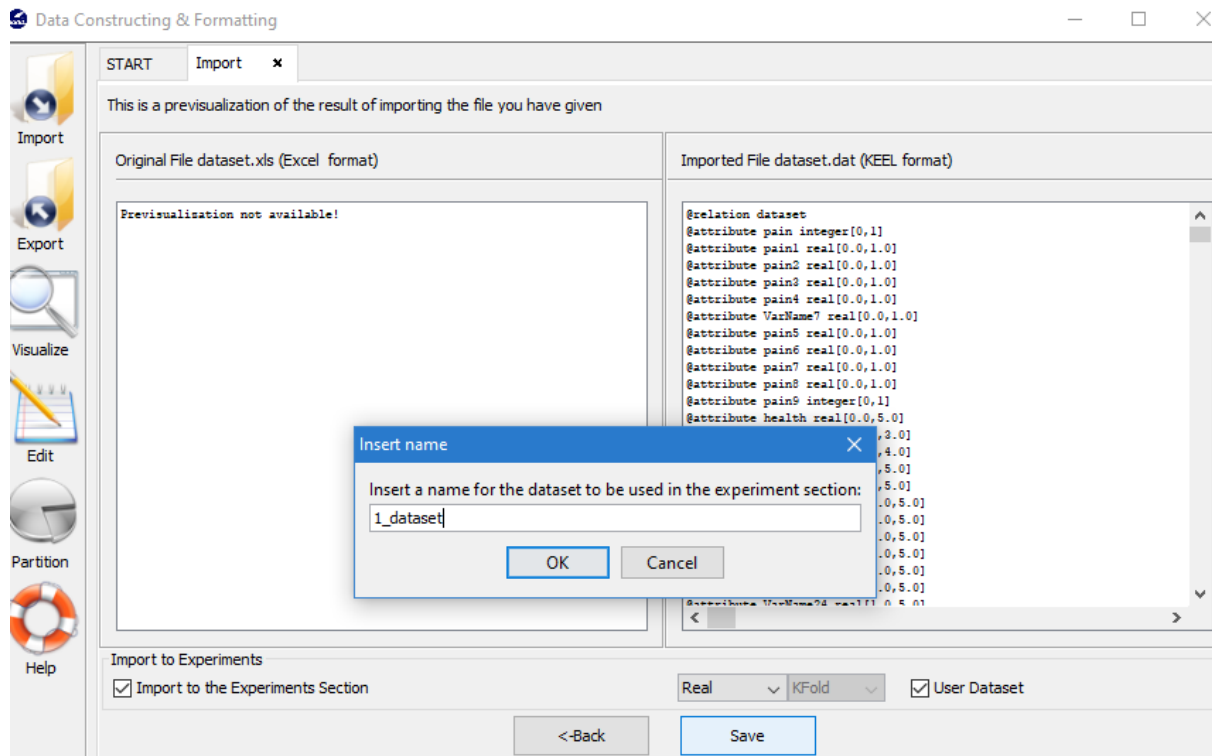
Στη συνέχεια μας δίνεται η επιλογή να το κάνουμε Import to the Experiments Section, την οποία την αφήνουμε επιλεγμένη και πατάμε Save.



Σχήμα 8



Μετά μας ζητάει να δώσουμε όνομα στο dataset.



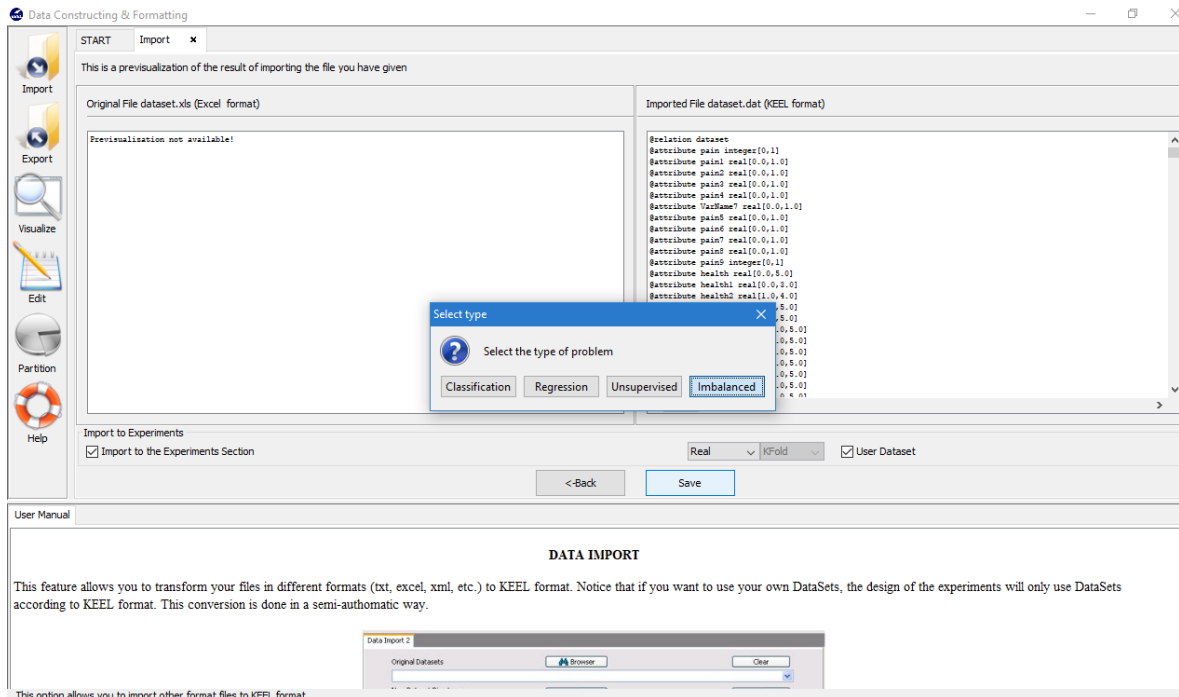
Σχήμα 9

Έπειτα έχουμε την επιλογή να διαλέξουμε τι κατηγορία πρόβλημα έχουμε. Το συγκεκριμένο είναι imbalanced και μπορούμε να το δούμε από το visualize data (αφού το έχουμε κάνει Import πρώτα για να είναι σε keel format). Επίσης μπορούμε να παρατηρήσουμε ότι είναι Imbalanced από την Excel μορφή, πριν δηλαδή το κάνουμε Import στο Keel Software με την εντολή =COUNTIF(EI:EI;3), όπου EI είναι η στήλη που περιέχει το Output. Και έχουμε ότι για κάθε περίπτωση έχουμε τόσες εμφανίσεις.

1: 328

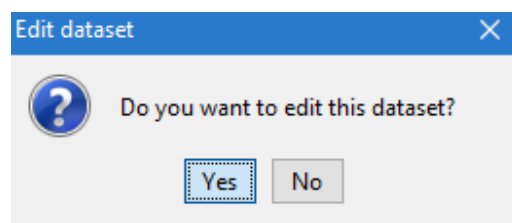
2: 789

3: 5



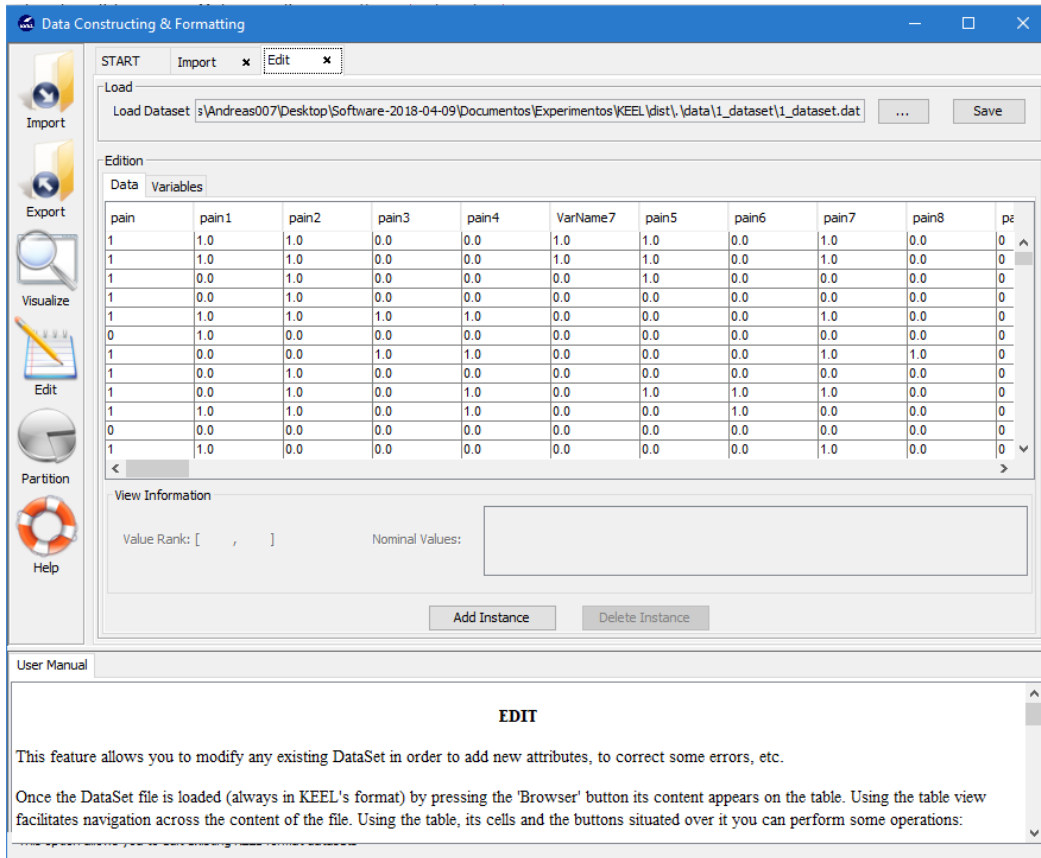
Σχήμα 10

Μετά θα μας ρωτήσει αν θέλουμε να κάνουμε edit το dataset και θα επιλέξουμε ναι για να αλλάξουμε τον τύπο της εξόδου, γιατί μετά την μετατροπή από το Excel, που αλλάξαμε κάθε string σε αριθμό( Progression -> 1, Incidence ->2, Non-exposed -> 3) θα νομίζει ότι είναι Regression πρόβλημα, ενώ εμείς θέλουμε να φαίνονται Nominal τιμές.



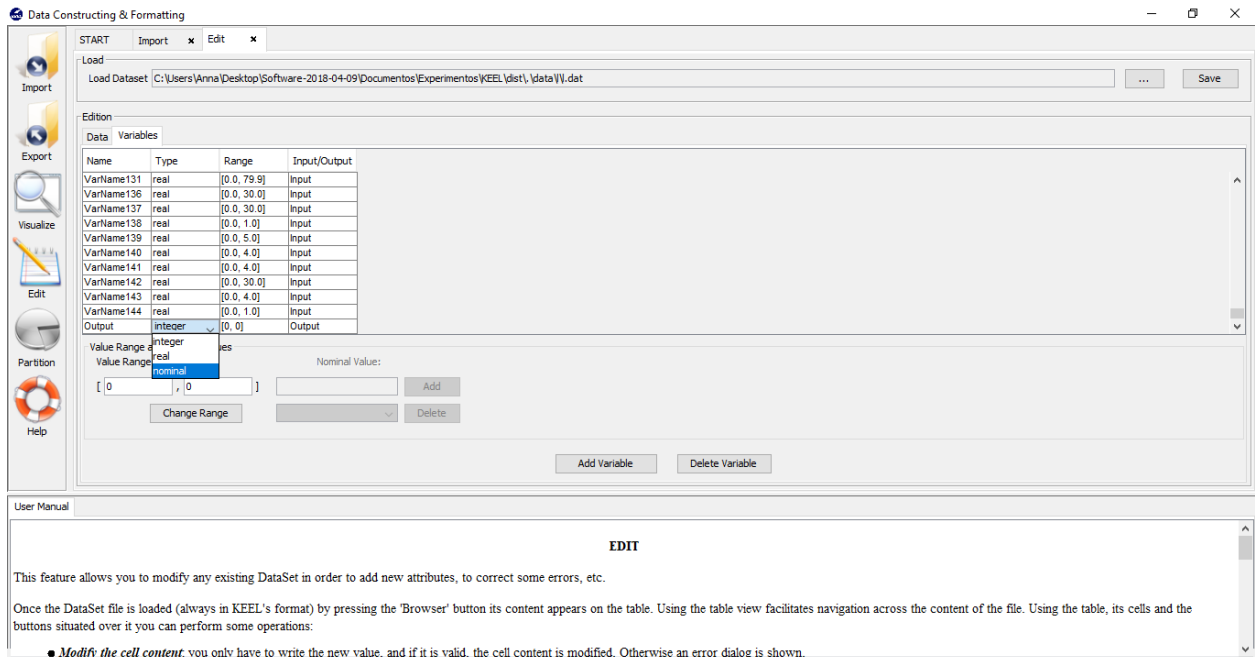
Σχήμα 11

Αυτό που μας εμφανίζεται στη συνέχεια είναι το παρακάτω και πάμε στην επιλογή Variables για να αλλάξουμε την μορφή.



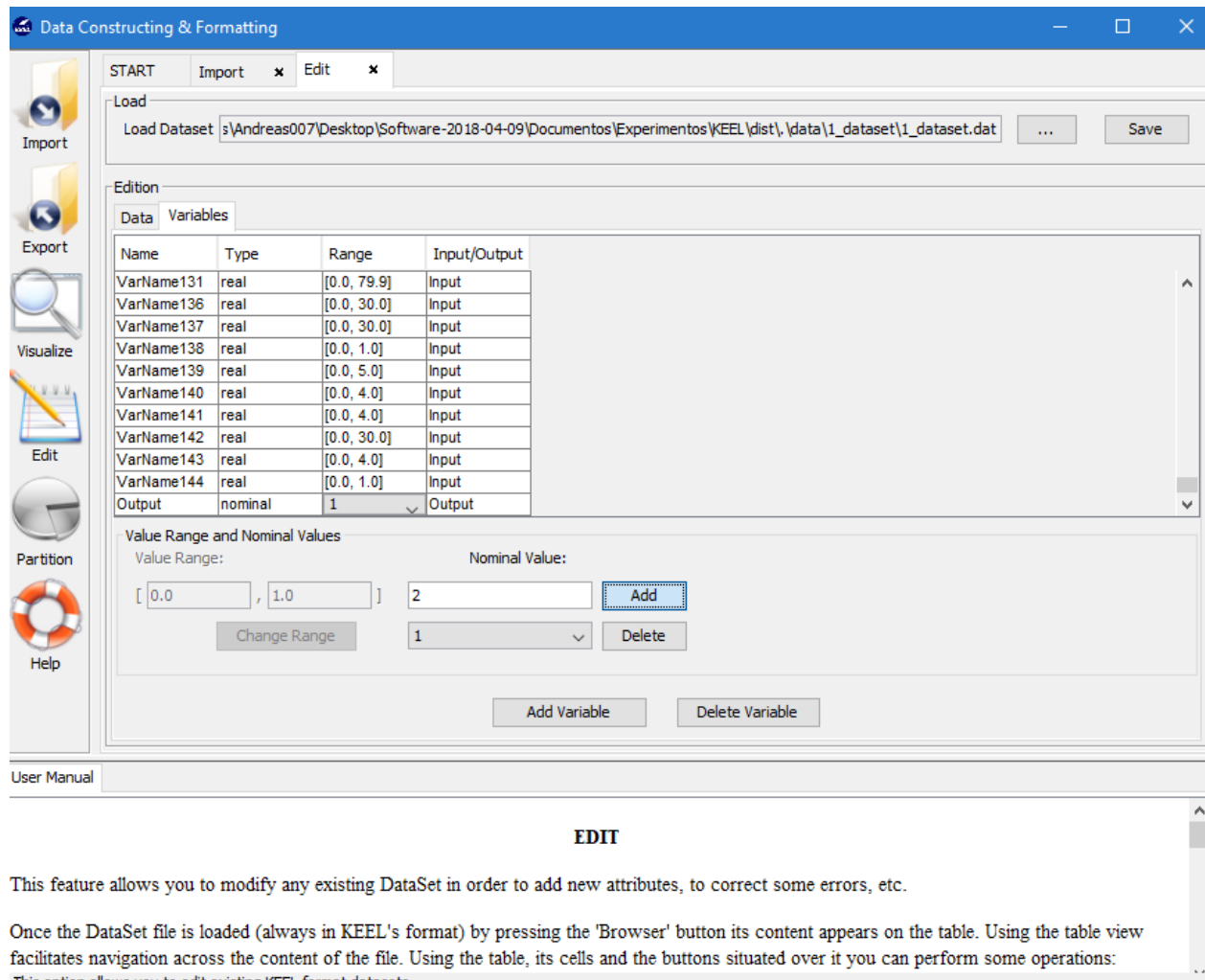
Σχήμα 12

Και όπως βλέπουμε παρακάτω αλλάζουμε την μορφή από Integer σε Nominal.



Σχήμα 13

Επειδή όμως μια τιμή χάνονταν με αυτόν τον τρόπο την έκανα add για να είναι σωστό το dataset.



The screenshot shows the 'Data Constructing & Formatting' application window. The 'Edit' tab is active, displaying a table of variables and options to modify their ranges and nominal values.

Name	Type	Range	Input/Output
VarName131	real	[0.0, 79.9]	Input
VarName136	real	[0.0, 30.0]	Input
VarName137	real	[0.0, 30.0]	Input
VarName138	real	[0.0, 1.0]	Input
VarName139	real	[0.0, 5.0]	Input
VarName140	real	[0.0, 4.0]	Input
VarName141	real	[0.0, 4.0]	Input
VarName142	real	[0.0, 30.0]	Input
VarName143	real	[0.0, 4.0]	Input
VarName144	real	[0.0, 1.0]	Input
Output	nominal	1	Output

Value Range and Nominal Values

Value Range: [ 0.0 , 1.0 ]

Nominal Value: 2

Buttons: Add, Change Range, Delete, Add Variable, Delete Variable

User Manual

### EDIT

This feature allows you to modify any existing DataSet in order to add new attributes, to correct some errors, etc.

Once the DataSet file is loaded (always in KEEL's format) by pressing the 'Browser' button its content appears on the table. Using the table view facilitates navigation across the content of the file. Using the table, its cells and the buttons situated over it you can perform some operations:

This option allows you to edit existing KEEL format datasets

Σχήμα 14

Και κάνουμε save τις αλλαγές και επιλέγουμε που θέλουν να αποθηκευτούν στον υπολογιστή μας και θα μας ρωτήσει αν πρέπει να το αντικαταστήσει με αυτό που ήδη υπάρχει και θα επιλέξουμε Ναι.

Μετά θα μας ρωτήσει αν θέλουμε να δημιουργήσουμε partitions και θα επιλέξουμε Ναι γιατί το Imbalanced Module δουλεύει με 5 fold.

The screenshot shows the 'Data Constructing & Formatting' window. The 'Load' section shows the dataset path: 's:\Andreas007\Desktop\Software-2018-04-09\Documents\Experimentos\KEEL\dist\, \data\1\_dataset\1\_dataset.dat'. The 'Edition' section has a 'Data' tab with a table of variables:

Name	Type	Range	Input/Output
VarName131	real	[0.0, 79.9]	Input
VarName136	real	[0.0, 30.0]	Input
VarName137	real	[0.0, 30.0]	Input
VarName138	real	[0.0, 1.0]	Input
VarName139	real	[0.0, 5.0]	Input
VarName140	real	[0.0, 4.0]	Input
VarName141	real	[0.0, 4.0]	Input
VarName142	real	[0.0, 30.0]	Input
VarName143	real	[0.0, 4.0]	Input
VarName144	real	[0.0, 1.0]	Input
Output	nominal	1	

A dialog box titled 'Make partitions' is open, asking 'Do you want to make partitions for this dataset?' with 'Yes' and 'No' buttons. The 'Yes' button is highlighted. Below the table, there are fields for 'Value Range and Nominal Values' and buttons for 'Add Variable' and 'Delete Variable'.

**EDIT**

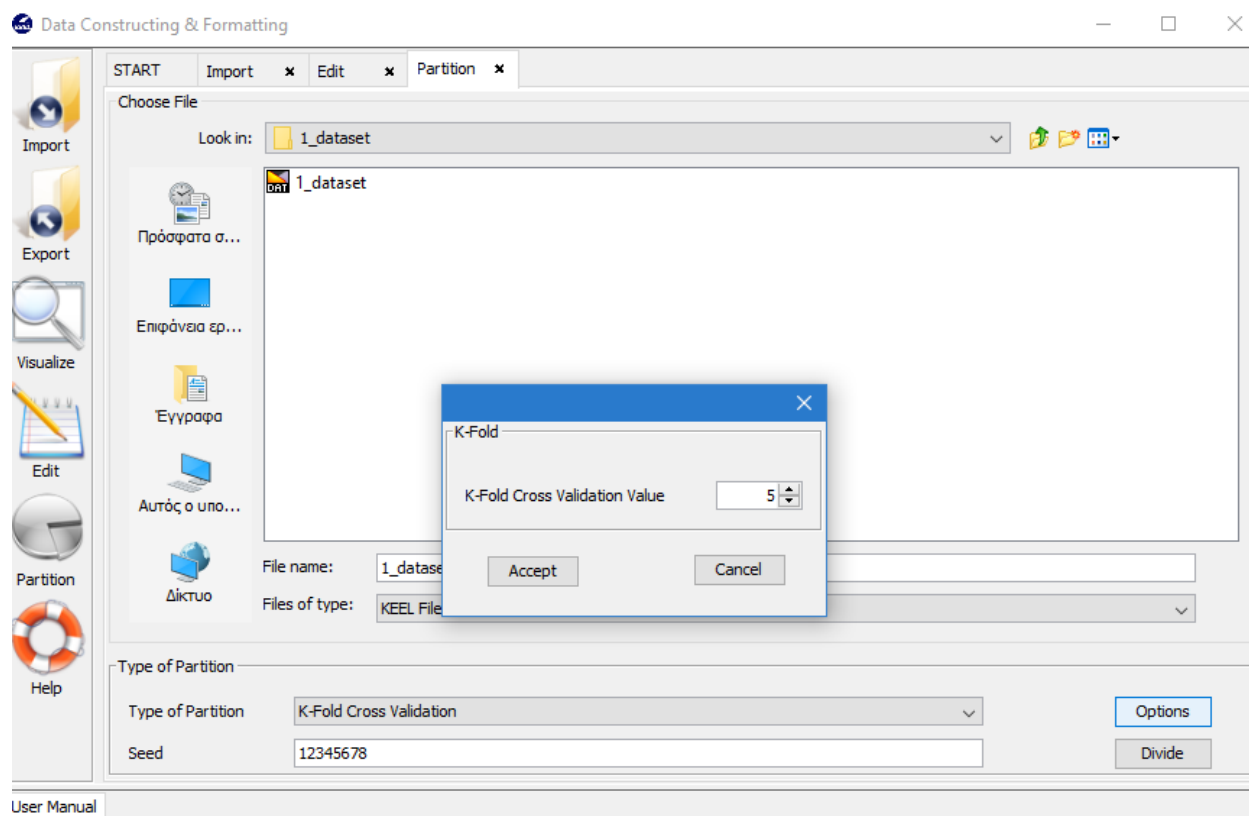
This feature allows you to modify any existing DataSet in order to add new attributes, to correct some errors, etc.

Once the DataSet file is loaded (always in KEEL's format) by pressing the 'Browser' button its content appears on the table. Using the table view facilitates navigation across the content of the file. Using the table, its cells and the buttons situated over it you can perform some operations:

This option allows you to edit existing KEEL format datasets

Σχήμα 15

Πάμε στο options και επιλέγουμε τον αριθμό 5 και πατάμε accept και μετά επιλέγουμε το divide.



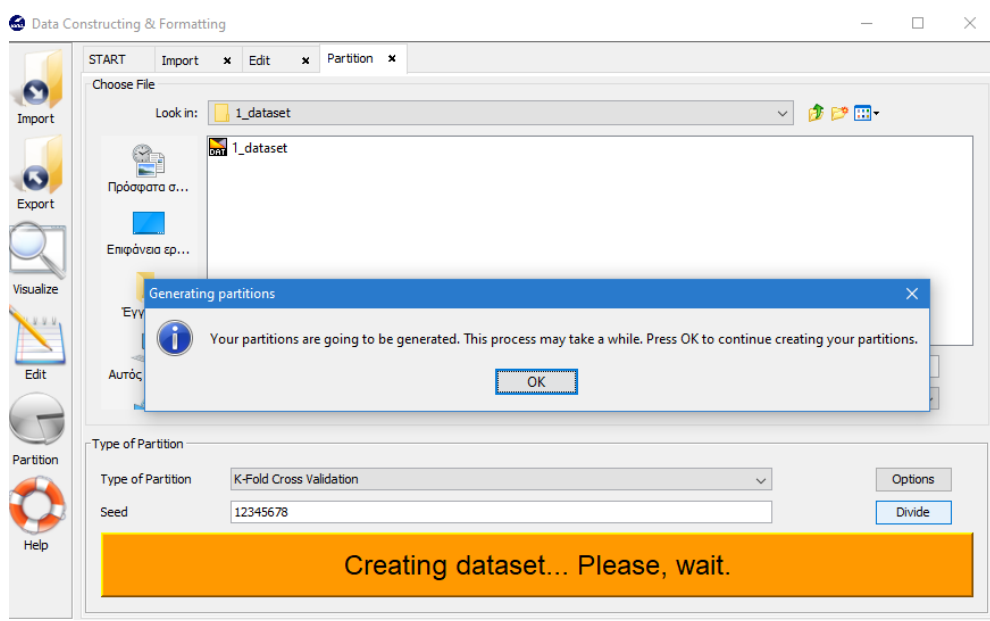
## PARTITION

All the DataSets created by the user or transformed from UCI files, are called "Complete DataSets" and cannot be used in experiments. First it's necessary to divide the whole file in pairs of training and test files.

To divide a file, you must select the "Complete DataSet" and indicate the root name of a directory that will contain all the files generated during the

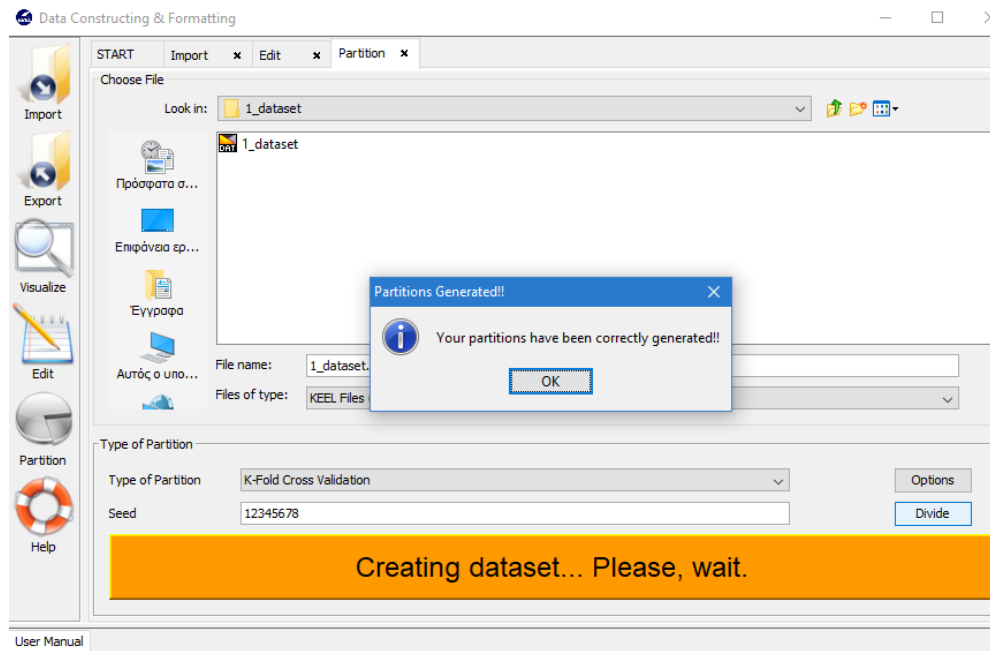
Σχήμα 16

Και θα μας ρωτήσει που θέλουμε να το αποθηκεύσουμε στον υπολογιστή μας και εμφανίζεται ότι δημιουργούνται τα partitions μετά.



Σχήμα 17

Και μετά μας ειδοποιεί ότι δημιουργήθηκαν σωστά τα partitions.



Σχήμα 18

## 4. Υλοποίηση αλγορίθμων ταξινόμησης

### 4.1. Αποτελέσματα

Αυτή την στιγμή έχει δημιουργηθεί το dataset, το οποίο έχει επεξεργαστεί και έχει χωριστεί σε 5 partitions και είναι έτοιμο να τρέξουμε αλγορίθμους με τα δεδομένα αυτά. Οπότε γυρνάμε πίσω στο αρχικό μενού.



Σχήμα 19

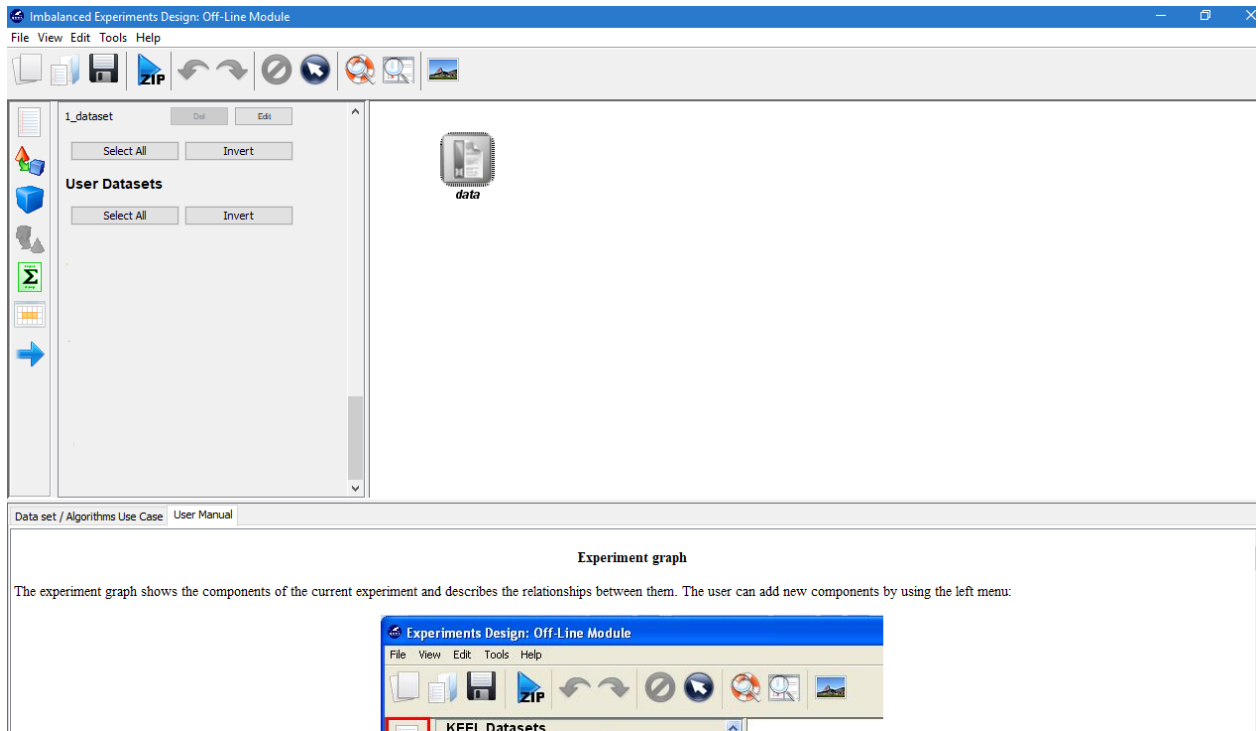


Και επιλέγουμε το Modules και μετά το Imbalanced Learning μιας και κατηγοριοποιήσαμε το dataset μας εκεί.




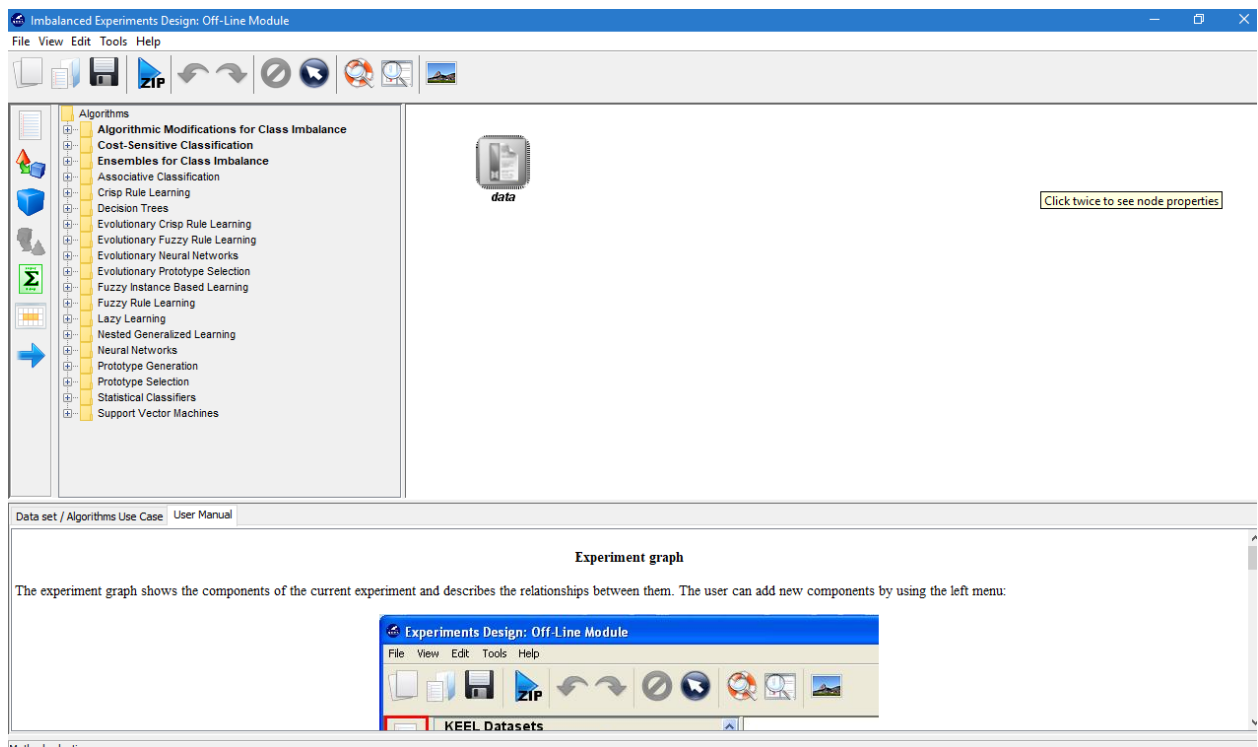
Σχήμα 20

Βρίσκουμε το dataset μας που το έχουμε ονομάσει 1\_Dataset, το επιλέγουμε και κάνουμε δεξί κλικ στην επιφάνεια για να εισαχθεί στο πείραμα.



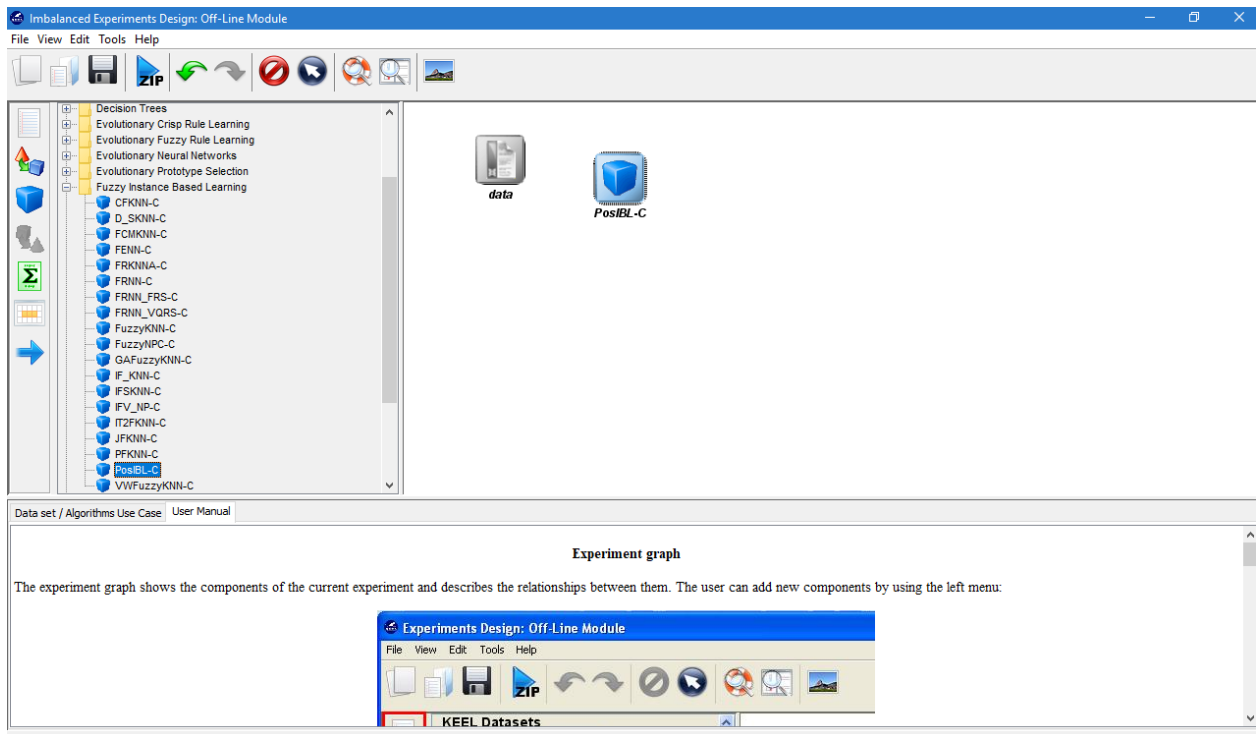
Σχήμα 21

Επιλέγοντας αυτό το εικονίδιο  (Σχήμα 22) έχουμε πρόσβαση στους αλγορίθμους που μπορούμε να τρέξουμε, οι οποίοι φαίνονται και παρακάτω.




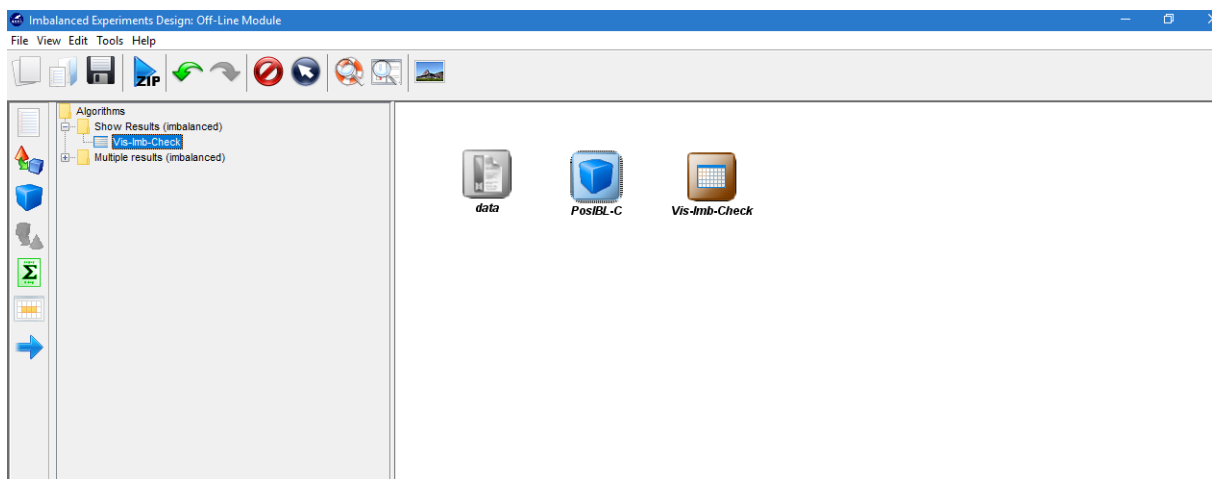
Σχήμα 23

Ας τρέξουμε τον αλγόριθμο PosIBL-C από Fuzzy Instance Based Learning που έβγαλε τα καλύτερα αποτελέσματα. Ανοίγουμε τις επιλογές από το Fuzzy Instance Based Learning και επιλέγουμε τον αλγόριθμο και κάνουμε δεξί κλικ στην επιφάνεια και με αυτό τον τρόπο έχει εισαχθεί στο πείραμα.



Σχήμα 24

Θα χρειαστούμε όμως να δούμε και τα αποτελέσματα από το παραπάνω πείραμα, οπότε από την κάθετη μπάρα αριστερά, επιλέγουμε το εικονίδιο  (Σχήμα 25) και από το Show Results εισάγουμε το Vis- Imb- Check

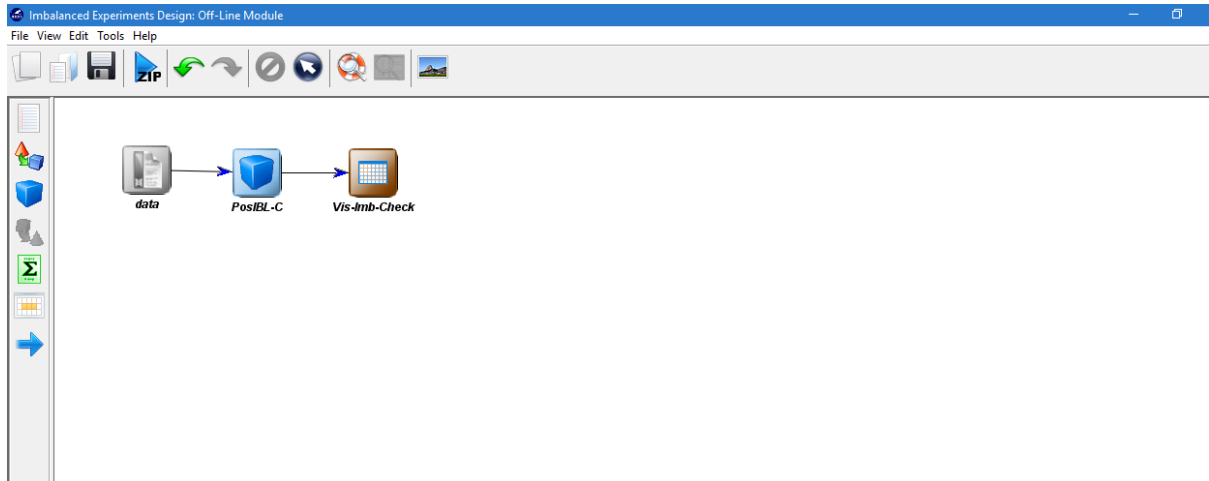


Σχήμα 26

Χρειάζονται και σύνδεση τα παραπάνω το οποίο θα γίνει επιλέγοντας το  
και τα συνδέουμε με τον εξής τρόπο.



(Σχήμα 27)

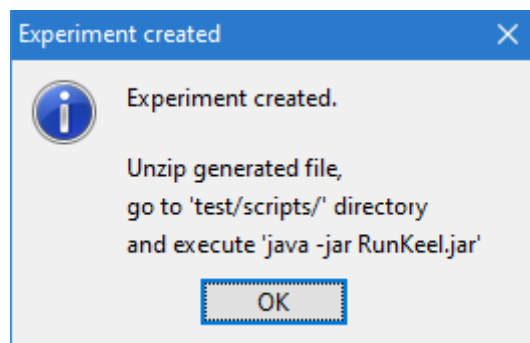


Σχήμα 28

Το πείραμα μας είναι έτοιμο, το μόνο που χρειάζεται είναι να του δώσουμε ένα όνομα και περιοχή που θα αποθηκευτεί και θα αναζητήσουμε τα αποτελέσματα μας.



Για αυτό το λόγο επιλέγουμε το εικονίδιο ZIP (Σχήμα 29) και ονομάζουμε το πείραμα μας έστω test και εμφανίζεται το παρακάτω μήνυμα.



Σχήμα 30

## 4.2. Σύγκριση αποτελεσμάτων

Εκτελούμε τα βήματα που μας ζητά το Keel και όταν τελειώσει το πείραμα, πάμε στο φάκελο test\test\results\Vis-Imb-Check\TSTPosIBL-C και ανοίγουμε το αρχείο result0s0 που έχει δημιουργηθεί για να δούμε το accuracy.

Το έτρεξα πρώτα σαν κατηγορία classification. Από την κατηγορία Decision Trees έτρεξα το C45-C και είχα 82% απόδοση, χωρίς imbalanced features.

Παρακάτω βλέπουμε την απόδοση και το confusion matrix

```
-----  
Global Average and Variance  
-----  
  
TEST RESULTS  
  
Dataset,C45-C,C45-C,C45-C  
,Average (Correctly Classified),Variance (Correctly Classified),Not Classified  
distaaset_firstaset_first ,0.8199510114,0.0023254075,0.0000000000  
  
TRAIN RESULTS
```

Σχήμα 31

```
CONFUSION MATRIX. ALGORITHM: C45-C  
-----  
  
TEST RESULTS  
,1,3,2  
1,225,0,103  
3,0,5,0  
2,99,0,690
```

Σχήμα 32

Με τα εργαλεία όμως του Keel διαπίστωσα ότι όλα τα παραπάνω dataset είναι Imbalanced, δηλαδή από τις τρεις εξόδους υπερσχύουν η πρώτη και η δεύτερη. Οπότε ξαναέτρεξα το dataset αλλά το άλλαξα κατηγορία.

Με αυτόν τον τρόπο έγινε το παρακάτω πινακάκι, όπου έχουν προηγηθεί αρκετά τρεξίματα με μια μεγάλη ποικιλία αλγορίθμων.

Datasets	Global Classification
<b>Dataset1(1122)</b>	
<b>Default</b> (Neural- EvRBFN-C)	50%
Generations 100->300	41%
Mutator rate 0.1->0.3	36%
Generations 100->20	56%
Validation rate 0.15->0.3	42%
Neurons rate 0.1->0.5	80%****
Replacement rate 0.1->0.6	39%
<b>Default</b> (Fuzzy Instance-PosIBL-C)	90%***
<b>Default</b> (Fuzzy Instance. FRNN-FRS-C)	80%***
<b>Dataset_All (4796)</b>	
<b>Default</b> (Fuzzy Instance-PosIBL-C)	40%
<b>Default</b> (Fuzzy Instance. FRNN-FRS-C)	50%
<b>Default</b> (Crisp_Rule_Learning-C45Rules-C)	98%***
<b>Dataset2(1727)</b>	
<b>Default</b> (Ensemble-Adaboost-NC-I)	64%
Number of Classifiers 10->2	73%****
Number of Classifiers 10->30	62%
Instances per Leaf 2->7	73%****
<b>Default</b> (Ensemble-EasyBoost)	55%
Instances per Leaf 2->7	56%
Classifiers 10->2	55%
<b>Default</b> (Fuzzy Instance. FRNN-FRS-C)	40%
K-value 3 to 15	40%

<b>Default</b> (Fuzzy Instance-PosIBL-C)	20%
<b>Default</b> (Fuzzy Instance-FuzzyKnn-C)	40%
Initialization(Crisp->Class-Mean)	99%***
<b>Dataset3(3026)</b>	
<b>Default</b> (Ensemble-Adaboost-NC-I)	59%
Pruned True->False	53%
Instanced Per Leaf 2->4	46%
Lambda 2->1	55%
<b>Default</b> (Ensemble-Ada-C2-I)	41%
<b>Default</b> (Fuzzy Instance. FRNN-FRS-C)	93%****
K-value 3 to 15	40%

*Πίνακας 2*

Παρακάτω βλέπουμε τους fuzzy rules που παράχθηκαν από τα παραπάνω dataset χρησιμοποιώντας τον αλγόριθμο FURIA-C από την κατηγορία Fuzzy Rule Learning.

Θα παρουσιαστούν fuzzy rules από όλα τα 5 folds στα παρακάτω dataset επειδή είχαμε 100% απόδοση στο dataset\_full και dataset1. Στο dataset2 και 3 έχουμε 0% και στα 5 folds.

Αρχικά παρατηρούμε ότι στο Dataset\_Full που έχει 4796 καταχωρήσεις και τα 5 folds βγάζουν 100% απόδοση, το οποίο είναι εν μέρει λογικό, καθώς έχουμε να κάνουμε με μεγάλο όγκο δεδομένων.

Παρατηρούμε ότι στο 0 fold έχουν δημιουργηθεί 11 ασαφείς κανόνες.



```

FURIA rules:
=====
=> V00COHORT=1 (CF = 0.0)
(P02KPN <= 0(-> 1)) => V00COHORT=3 (CF = 0.19)
(P02KPN <= 0(-> 1)) => V00COHORT=3 (CF = 0.19)
(P01SXKOA <= 0.583(-> 1)) => V00COHORT=2 (CF = 0.96)
(P01SXKOA <= 0(-> 1) and (V00KOOSFX2 >= 0.978(-> 0)) => V00COHORT=2 (CF = 0.99)
(P01SXKOA <= 0(-> 1)) => V00COHORT=2 (CF = 0.96)
(P01SXKOA <= 0(-> 1)) => V00COHORT=2 (CF = 0.96)
(P01SXKOA <= 0(-> 1)) => V00COHORT=2 (CF = 0.96)
(P01SXKOA <= 0(-> 1)) => V00COHORT=2 (CF = 0.96)
(P01SXKOA <= 0(-> 1)) => V00COHORT=2 (CF = 0.96)
(P01SXKOA <= 0(-> 1)) => V00COHORT=2 (CF = 0.96)

Reglas Buenas
Inside_m_Ruleset
(P01SXKOA >= 1(-> 0.583)) => V00COHORT=1 (CF = 1.0)
(P02KPN <= 0(-> 1) and (V00KOOSFX3 <= 0(-> 0.894) and (V00HSPSS >= 55.33(-> 52.4) and (P01KSX <= 0(-> 2) and (V00KSXRKN2 >= 1(-> 0)) =>
V00COHORT=3 (CF = 0.76)
(P02KPN <= 0(-> 1) and (V00KOOSFX3 <= 0(-> 0.894) and (V00KOOSQOL >= 100(-> 87.5) and (V00HSPSS >= 55.33(-> 55.09) and (P01KPNREV <= 0(-> 1))
and (P01KSX <= 0(-> 1) and (V00KOOSYML >= 100(-> 96.4)) => V00COHORT=3 (CF = 0.57)
(P01SXKOA <= 0.583(-> 1) and (P02KPN >= 0.873(-> 0)) => V00COHORT=2 (CF = 1.0)
(P01SXKOA <= 0(-> 1) and (V00KOOSFX2 >= 0.978(-> 0) and (V00SF1 >= 2(-> 1)) => V00COHORT=2 (CF = 1.0)
(P01SXKOA <= 0(-> 1) and (V00HSPSS <= 55.28(-> 55.33)) => V00COHORT=2 (CF = 0.99)
(P01SXKOA <= 0(-> 1) and (P01KSX >= 1(-> 0)) => V00COHORT=2 (CF = 1.0)
(P01SXKOA <= 0(-> 1) and (P01HPNR12 >= 0.436(-> 0) and (V00SF9 <= 2(-> 3) and (V00SF10 >= 2(-> 1)) => V00COHORT=2 (CF = 1.0)
(P01SXKOA <= 0(-> 1) and (P01KPNLEV >= 1(-> 0)) => V00COHORT=2 (CF = 1.0)
(P01SXKOA <= 0(-> 1) and (V00KOOSYML <= 96.4(-> 100) and (V00KSXRKN2 <= 0(-> 1)) => V00COHORT=2 (CF = 0.99)
(P01SXKOA <= 0(-> 1) and (P01KPNREV >= 1(-> 0)) => V00COHORT=2 (CF = 1.0)

Number of Rules : 11

REGLAS = 11

```

Σχήμα 33

Στο 1 fold έχουμε συν ένα κανόνα παραπάνω , καθώς δημιουργήθηκαν 12 κανόνες.

```

FURIA rules:
=====
=> V00COHORT=1 (CF = 0.0)
(P02KPN <= 0(-> 1) and (V00HSPSS >= 56.13(-> 53.46) and (V00KOOSFX3 <= 0(-> 0.894) and (P01KSX <= 0(-> 1)) => V00COHORT=3 (CF = 0.47)
(P02KPN <= 0(-> 1) and (V00HSPSS >= 56.13(-> 54.52) and (V00KOOSQOL >= 100(-> 93.8) and (V00KOOSFX2 <= 0(-> 0.978) and (P01BP30 <= 0(-> 1) and (
P01HPNL12 <= 0(-> 1)) => V00COHORT=3 (CF = 0.56)
(P02KPN <= 0(-> 1)) => V00COHORT=3 (CF = 0.2)
(P01SXKOA <= 0.583(-> 1)) => V00COHORT=2 (CF = 0.96)
(P01SXKOA <= 0(-> 1)) => V00COHORT=2 (CF = 0.96)
(P01SXKOA <= 0(-> 1)) => V00COHORT=2 (CF = 0.96)
(P01SXKOA <= 0(-> 1)) => V00COHORT=2 (CF = 0.96)
(P01SXKOA <= 0(-> 1)) => V00COHORT=2 (CF = 0.96)
(P01SXKOA <= 0(-> 1)) => V00COHORT=2 (CF = 0.96)
(P01SXKOA <= 0(-> 1)) => V00COHORT=2 (CF = 0.96)
(P01SXKOA <= 0(-> 1)) => V00COHORT=2 (CF = 0.96)

Reglas Buenas
Inside_m_Ruleset
(P01SXKOA >= 1(-> 0.583)) => V00COHORT=1 (CF = 1.0)
(P02KPN <= 0(-> 1) and (V00HSPSS >= 56.13(-> 53.46) and (V00KOOSFX3 <= 0(-> 0.894) and (P01KSX <= 0(-> 1) and (V00KSXRKN2 >= 1(-> 0)) =>
V00COHORT=3 (CF = 0.72)
(P02KPN <= 0(-> 1) and (V00HSPSS >= 56.13(-> 54.52) and (V00KOOSQOL >= 100(-> 93.8) and (V00KOOSFX2 <= 0(-> 0.978) and (P01BP30 <= 0(-> 1) and (
P01HPNL12 <= 0(-> 1) and (V00SF1 <= 1(-> 2) and (P01KPNREV <= 0(-> 1)) => V00COHORT=3 (CF = 0.64)
(P02KPN <= 0(-> 1) and (V00HSPSS >= 57.28(-> 57.22) and (P01KSX <= 0(-> 1) and (V00KOOSYML >= 100(-> 96.4) and (P01KPNREV <= 0(-> 1) and (
P01BP30 <= 0(-> 1) and (V00HSHSS >= 49.12(-> 49.09)) => V00COHORT=3 (CF = 0.74)
(P01SXKOA <= 0.583(-> 1) and (P02KPN >= 0.873(-> 0)) => V00COHORT=2 (CF = 1.0)
(P01SXKOA <= 0(-> 1) and (V00HSPSS <= 56.42(-> 56.44)) => V00COHORT=2 (CF = 0.99)
(P01SXKOA <= 0(-> 1) and (V00KOOSQOL <= 93.8(-> 100) and (P01KSX >= 1(-> 0)) => V00COHORT=2 (CF = 1.0)
(P01SXKOA <= 0(-> 1) and (V00KOOSFX3 >= 0.894(-> 0)) => V00COHORT=2 (CF = 0.99)
(P01SXKOA <= 0(-> 1) and (P01KPNREV >= 1(-> 0)) => V00COHORT=2 (CF = 1.0)
(P01SXKOA <= 0(-> 1) and (P01BP30 >= 1(-> 0)) => V00COHORT=2 (CF = 0.98)
(P01SXKOA <= 0(-> 1) and (V00KOOSYML <= 96.4(-> 100)) => V00COHORT=2 (CF = 0.99)
(P01SXKOA <= 0(-> 1) and (P01HPNL12 >= 1(-> 0) and (P01HPR12CV <= 0(-> 1)) => V00COHORT=2 (CF = 0.99)

Number of Rules : 12

REGLAS = 12

```

Σχήμα 34

Στο 2 fold δημιουργήθηκαν 14 ασαφείς κανόνες.

```
FURIA rules:
=====

=> V00COHORT=1 (CF = 0.0)
(P02KPN <= 0(-> 1)) and (V00KOOSFX2 <= 0(-> 0.978)) and (V00HSPSS >= 55.58(-> 54.53)) => V00COHORT=3 (CF = 0.41)
(P02KPN <= 0(-> 1)) => V00COHORT=3 (CF = 0.2)
(P02KPN <= 0(-> 1)) and (V00WOMTSR <= 0(-> 1)) => V00COHORT=3 (CF = 0.27)
(P01SXKOA <= 0.583(-> 1)) => V00COHORT=2 (CF = 0.96)
(P01SXKOA <= 0(-> 1)) => V00COHORT=2 (CF = 0.96)
(P01SXKOA <= 0(-> 1)) => V00COHORT=2 (CF = 0.96)
(P01SXKOA <= 0(-> 1)) => V00COHORT=2 (CF = 0.96)
(P01SXKOA <= 0(-> 1)) => V00COHORT=2 (CF = 0.96)
(P01SXKOA <= 0(-> 1)) and (V00HSMSS >= 57.16(-> 57.06)) => V00COHORT=2 (CF = 0.97)
(P01SXKOA <= 0(-> 1)) => V00COHORT=2 (CF = 0.96)
(P01SXKOA <= 0(-> 1)) => V00COHORT=2 (CF = 0.96)
(P01SXKOA <= 0(-> 1)) => V00COHORT=2 (CF = 0.96)
(P01SXKOA <= 0(-> 1)) and (V00HSMSS >= 54.74(-> 54.2)) => V00COHORT=2 (CF = 0.96)

Reglas Buenas
Inside m_Ruleset
(P01SXKOA >= 1(-> 0.583)) => V00COHORT=1 (CF = 1.0)
(P02KPN <= 0(-> 1)) and (V00KOOSFX2 <= 0(-> 0.978)) and (V00HSPSS >= 55.58(-> 54.53)) and (V00KSXRKN2 >= 1(-> 0)) => V00COHORT=3 (CF = 0.6)
(P02KPN <= 0(-> 1)) and (V00SF1 <= 1(-> 2)) and (V00KOOSFSR >= 100(-> 95)) and (V00HSMSS <= 56.47(-> 57.06)) => V00COHORT=3 (CF = 0.59)
(P02KPN <= 0(-> 1)) and (V00WOMTSR <= 0(-> 1)) and (V00SF1 <= 1(-> 2)) and (P01BP30 <= 0(-> 1)) and (V00KOOSVML >= 100(-> 96.4)) and (P01HPNL12 <= 0(-> 1)) => V00COHORT=3 (CF = 0.63)
(P01SXKOA <= 0.583(-> 1)) and (P02KPN >= 1(-> 0)) => V00COHORT=2 (CF = 1.0)
(P01SXKOA <= 0(-> 1)) and (V00KOOSFX2 >= 0.978(-> 0)) and (V00SF1 >= 2(-> 1)) => V00COHORT=2 (CF = 0.99)
(P01SXKOA <= 0(-> 1)) and (V00KOOSQDL <= 93.8(-> 100)) and (V00KOOSFSR <= 95(-> 100)) and (V00KOOSQDL >= 75(-> 68.8)) => V00COHORT=2 (CF = 1.0)
(P01SXKOA <= 0(-> 1)) and (P01KXS >= 1(-> 0)) => V00COHORT=2 (CF = 1.0)
(P01SXKOA <= 0(-> 1)) and (V00HSPSS <= 55.86(-> 56.07)) => V00COHORT=2 (CF = 0.99)
(P01SXKOA <= 0(-> 1)) and (V00HSMSS >= 57.16(-> 57.06)) and (P01HPNR12 >= 1(-> 0)) and (P01TMJE300 >= 8.98(-> 5)) => V00COHORT=2 (CF = 0.99)
(P01SXKOA <= 0(-> 1)) and (P01BP30 >= 1(-> 0)) and (V00HSMSS >= 51.83(-> 51.24)) and (V00HSPSS >= 56.71(-> 15.39)) => V00COHORT=2 (CF = 0.98)
(P01SXKOA <= 0(-> 1)) and (V00HSMSS <= 48.53(-> 48.61)) and (V00HSPSS <= 59.43(-> 61.04)) => V00COHORT=2 (CF = 0.99)
(P01SXKOA <= 0(-> 1)) and (P01KPNLEV >= 1(-> 0)) => V00COHORT=2 (CF = 1.0)
(P01SXKOA <= 0(-> 1)) and (V00HSMSS >= 54.74(-> 54.2)) and (V00KOOSFX2 >= 0.978(-> 0)) and (V00KOOSFSR >= 100(-> 0)) => V00COHORT=2 (CF = 0.98)

Number of Rules : 14

REGLAS = 14
```

Σχήμα 35

Στη επόμενο fold to 3 έχουμε 13 κανόνες.

```
FURIA rules:
=====
=> V00COHORT=1 (CF = 0.0)
(P02KPN <= 0(-> 1)) and (V00KOOSFX3 <= 0(-> 0.894)) and (V00HSPSS >= 55.33(-> 55.28)) => V00COHORT=3 (CF = 0.45)
(P02KPN <= 0(-> 1)) and (V005F1 <= 1(-> 2)) => V00COHORT=3 (CF = 0.35)
(P01SXKOA <= 0(-> 1)) => V00COHORT=2 (CF = 0.96)
(P01SXKOA <= 0(-> 1)) => V00COHORT=2 (CF = 0.96)
(P01SXKOA <= 0(-> 1)) => V00COHORT=2 (CF = 0.96)
(P01SXKOA <= 0(-> 1)) => V00COHORT=2 (CF = 0.96)
(P01SXKOA <= 0(-> 1)) => V00COHORT=2 (CF = 0.96)
(P01SXKOA <= 0(-> 1)) => V00COHORT=2 (CF = 0.96)
(P01SXKOA <= 0(-> 1)) and (P01HPNR12 >= 1(-> 0)) => V00COHORT=2 (CF = 0.99)
(P01SXKOA <= 0(-> 1)) => V00COHORT=2 (CF = 0.96)
(P01SXKOA <= 0(-> 1)) => V00COHORT=2 (CF = 0.96)
(P01SXKOA <= 0(-> 1)) => V00COHORT=2 (CF = 0.96)

Reglas Buenas
Inside m_Ruleset
(P01SXKOA >= 1(-> 0)) => V00COHORT=1 (CF = 1.0)
(P02KPN <= 0(-> 1)) and (V00KOOSFX3 <= 0(-> 0.894)) and (V00HSPSS >= 55.33(-> 55.28)) and (V00HSMSS <= 54.69(-> 54.74)) and (P01KXS <= 0(-> 1)) => V00COHORT=3 (CF = 0.63)
(P02KPN <= 0(-> 1)) and (V005F1 <= 1(-> 2)) and (P01HPNR12 <= 0(-> 1)) and (V00KOOSYML >= 100(-> 96.4)) and (P01BP30 <= 0(-> 1)) and (V00HSPSS >= 57.47(-> 56.71)) and (V00HSMSS >= 54.1(-> 51.14)) => V00COHORT=3 (CF = 0.69)
(P01SXKOA <= 0(-> 1)) and (P02KPN >= 0.873(-> 0)) => V00COHORT=2 (CF = 1.0)
(P01SXKOA <= 0(-> 1)) and (V00KOOSFX3 >= 0.894(-> 0)) and (V00HSPSS <= 52.5(-> 53.18)) and (P01BPBAD >= 1.559(-> 1)) => V00COHORT=2 (CF = 1.0)
(P01SXKOA <= 0(-> 1)) and (P01KXS >= 1(-> 0)) => V00COHORT=2 (CF = 1.0)
(P01SXKOA <= 0(-> 1)) and (V00HSPSS <= 55.86(-> 56.07)) => V00COHORT=2 (CF = 0.99)
(P01SXKOA <= 0(-> 1)) and (V00HSPSS <= 57.22(-> 57.28)) => V00COHORT=2 (CF = 0.98)
(P01SXKOA <= 0(-> 1)) and (V00KOOSYML <= 96.4(-> 100)) => V00COHORT=2 (CF = 0.99)
(P01SXKOA <= 0(-> 1)) and (P01HPNR12 >= 1(-> 0)) and (V00KOOSFX2 >= 0.978(-> 0)) => V00COHORT=2 (CF = 1.0)
(P01SXKOA <= 0(-> 1)) and (V00HSMSS <= 51.24(-> 51.37)) and (V00KOOSFX2 >= 0.978(-> 0)) => V00COHORT=2 (CF = 1.0)
(P01SXKOA <= 0(-> 1)) and (P01BP30 >= 1(-> 0)) and (V005F1 >= 5(-> 1)) => V00COHORT=2 (CF = 0.98)
(P01SXKOA <= 0(-> 1)) and (P01HPNL12 >= 1(-> 0)) => V00COHORT=2 (CF = 0.99)

Number of Rules : 13

REGLAS = 13
```

Σχήμα 36

Στο τελευταίο fold το 4 έχουμε 15 κανόνες. Βλέπουμε ότι μαζί με τα fold αυξήθηκαν κατά ένα και οι κανόνες.

```

FURIA rules:
=====

=> V00COHORT=1 (CF = 0.0)
(P02KPN <= 0(-> 1)) and (V00KOOSFX2 <= 0(-> 0.978)) => V00COHORT=3 (CF = 0.32)
(P02KPN <= 0(-> 1)) and (V00KOOSFX5 <= 0(-> 1)) => V00COHORT=3 (CF = 0.28)
(P01SXKOA <= 0.583(-> 1)) => V00COHORT=2 (CF = 0.96)
(P01SXKOA <= 0(-> 1)) => V00COHORT=2 (CF = 0.96)
(P01SXKOA <= 0(-> 1)) => V00COHORT=2 (CF = 0.96)
(P01SXKOA <= 0(-> 1)) => V00COHORT=2 (CF = 0.96)
(P01SXKOA <= 0(-> 1)) => V00COHORT=2 (CF = 0.96)
(P01SXKOA <= 0(-> 1)) => V00COHORT=2 (CF = 0.96)
(P01SXKOA <= 0(-> 1)) and (V00HSPSS <= 56.44(-> 56.71)) => V00COHORT=2 (CF = 0.98)
(P01SXKOA <= 0(-> 1)) => V00COHORT=2 (CF = 0.96)
(P01SXKOA <= 0(-> 1)) => V00COHORT=2 (CF = 0.96)
(P01SXKOA <= 0(-> 1)) => V00COHORT=2 (CF = 0.96)
(P01SXKOA <= 0(-> 1)) => V00COHORT=2 (CF = 0.96)
(P01SXKOA <= 0(-> 1)) and (V00KSXRKN2 <= 0(-> 4)) => V00COHORT=2 (CF = 0.96)
(P01SXKOA <= 0(-> 1)) => V00COHORT=2 (CF = 0.96)

Reglas Buenas
Inside m Ruleset
(P01SXKOA >= 1(-> 0.583)) => V00COHORT=1 (CF = 1.0)
(P02KPN <= 0(-> 1)) and (V00KOOSFX2 <= 0(-> 0.978)) and (V00HSPSS >= 56.66(-> 56.44)) and (P01BP30 <= 0(-> 1)) and (V00KOOSYML >= 100(-> 96.4)) =>
V00COHORT=3 (CF = 0.57)
(P02KPN <= 0(-> 1)) and (V00KOOSFX5 <= 0(-> 1)) and (V00HSPSS >= 57.78(-> 57.76)) and (V00KSXRKN2 >= 1(-> 0)) => V00COHORT=3 (CF = 0.7)
(P01SXKOA <= 0.583(-> 1)) and (P02KPN >= 0.873(-> 0)) => V00COHORT=2 (CF = 1.0)
(P01SXKOA <= 0(-> 1)) and (V00KOOSFX3 >= 0.894(-> 0)) and (V00WOMTSR >= 1(-> 0)) and (V00HSPSS <= 53.46(-> 53.52)) => V00COHORT=2 (CF = 1.0)
(P01SXKOA <= 0(-> 1)) and (V00KOOSFX2 >= 0.978(-> 0)) and (V00SF1 >= 2(-> 1)) and (V00KQOL1 >= 1(-> 0)) and (V00P7RKFR <= 2(-> 3)) => V00COHORT=2 (
CF = 1.0)
(P01SXKOA <= 0(-> 1)) and (P01HPNR12 >= 1(-> 0)) and (V00SF11 <= 4.322(-> 5)) => V00COHORT=2 (CF = 0.99)
(P01SXKOA <= 0(-> 1)) and (P01KSX >= 1(-> 0)) => V00COHORT=2 (CF = 1.0)
(P01SXKOA <= 0(-> 1)) and (V00HSPSS <= 52.11(-> 52.15)) => V00COHORT=2 (CF = 0.99)
(P01SXKOA <= 0(-> 1)) and (V00HSPSS <= 56.44(-> 56.71)) and (V00HSMSS >= 57.16(-> 56.74)) => V00COHORT=2 (CF = 0.98)
(P01SXKOA <= 0(-> 1)) and (V00KOOSFSR <= 93.8(-> 95)) and (V00HSMSS <= 54.79(-> 57.06)) => V00COHORT=2 (CF = 0.99)
(P01SXKOA <= 0(-> 1)) and (P01KPNREV >= 1(-> 0)) => V00COHORT=2 (CF = 1.0)
(P01SXKOA <= 0(-> 1)) and (V00HSMSS >= 57.06(-> 55.48)) and (V00KOOSYML <= 96.4(-> 100)) => V00COHORT=2 (CF = 1.0)
(P01SXKOA <= 0(-> 1)) and (V00KSXRKN2 <= 0(-> 4)) and (P01BP30 >= 1(-> 0)) and (V00HSPSS >= 56.15(-> 15.67)) => V00COHORT=2 (CF = 0.98)
(P01SXKOA <= 0(-> 1)) and (V00HSMSS <= 49.04(-> 49.12)) => V00COHORT=2 (CF = 0.98)

Number of Rules : 15
REGLAS = 15

```

Σχήμα 37

Στο Dataset1 με 1122 καταχωρήσεις έχουμε επίσης 100% απόδοση και στα 5 folds.  
Πιο αναλυτικά τα βλέπουμε παρακάτω.

Στο 0 fold, δημιουργήθηκαν 26 ασαφείς κανόνες.

```
FURIA rules:|
=====

(VarName139 >= 3(-> 2)) and (VarName91 >= 1.215(-> 1)) => Output=1 (CF = 0.76)
(VarName139 >= 3(-> 2)) and (VarName91 >= 1.215(-> 1)) and (VarName101 >= 0.481(-> 0)) and (VarName83 <= 0(-> 1)) =>
Output=1 (CF = 0.9)
(VarName139 >= 3(-> 2)) => Output=1 (CF = 0.67)
(VarName139 >= 3(-> 2)) and (VarName117 <= 56.3(-> 93.8)) => Output=1 (CF = 0.71)
(VarName139 >= 3(-> 2)) => Output=1 (CF = 0.67)
(VarName139 >= 3(-> 2)) and (VarName106 >= 59.89(-> 59.43)) => Output=1 (CF = 0.75)
(VarName139 >= 3(-> 2)) => Output=1 (CF = 0.67)
(VarName139 >= 3(-> 2)) => Output=1 (CF = 0.67)
(VarName139 >= 3(-> 2)) => Output=1 (CF = 0.67)
(VarName139 >= 3(-> 2)) => Output=1 (CF = 0.67)
(VarName139 >= 3(-> 2)) and (VarName112 <= 3(-> 4)) => Output=1 (CF = 0.65)
(VarName119 >= 0.475(-> 0)) => Output=1 (CF = 0.69)
(VarName139 >= 3(-> 2)) => Output=1 (CF = 0.67)
(VarName139 >= 3(-> 2)) => Output=1 (CF = 0.67)
(VarName139 >= 3(-> 2)) and (VarName98 >= 0.376(-> 0)) and (VarName112 <= 3(-> 4)) and (VarName69 >= 1(-> 0)) =>
Output=1 (CF = 0.83)
(VarName139 >= 3(-> 2)) => Output=1 (CF = 0.67)
(VarName119 >= 0.475(-> 0)) => Output=1 (CF = 0.69)
=> Output=3 (CF = 0.0)
=> Output=2 (CF = 0.0)
(VarName108 >= 84.4(-> 83.3)) => Output=2 (CF = 0.85)
=> Output=2 (CF = 0.0)
=> Output=2 (CF = 0.0)
=> Output=2 (CF = 0.0)
(VarName93 <= 1(-> 2)) and (VarName130 <= 10.5(-> 10.9)) => Output=2 (CF = 0.88)
=> Output=2 (CF = 0.0)
=> Output=2 (CF = 0.0)
```

Σχήμα 38

```

Reglas Buenas
Inside_m_Ruleset
(VarName139 >= 3(-> 2)) and (VarName91 >= 1.215(-> 1)) and (VarName109 >= 4(-> 3)) and (VarName117 <= 43.8(-> 50)) => Output=1 (CF = 0.96)
(VarName139 >= 3(-> 2)) and (VarName91 >= 1.215(-> 1)) and (VarName101 >= 0.481(-> 0)) and (VarName83 <= 0(-> 1)) and (VarName37 <= 1(-> 2)) =>
Output=1 (CF = 0.96)
(VarName139 >= 3(-> 2)) and (VarName131 >= 20.9(-> 20)) and (VarName38 >= 1(-> 0)) and (VarName71 <= 1(-> 2)) => Output=1 (CF = 0.97)
(VarName139 >= 3(-> 2)) and (VarName117 <= 56.3(-> 93.8)) and (pain6 <= 0(-> 1)) and (VarName96 >= 4(-> 3)) and (VarName62 <= 1(-> 2)) => Output=1 (
CF = 0.96)
(VarName139 >= 3(-> 2)) and (VarName115 <= 75(-> 78.6)) and (VarName85 <= 1(-> 2)) and (VarName65 >= 2(-> 1)) and (VarName80 >= 1(-> 0)) => Output=1
(CF = 0.97)
(VarName139 >= 3(-> 2)) and (VarName106 >= 59.89(-> 59.43)) and (health <= 2(-> 3)) and (VarName124 >= 3(-> 2)) and (VarName42 >= 1(-> 0)) =>
Output=1 (CF = 0.96)
(VarName139 >= 3(-> 2)) and (VarName107 >= 5(-> 4)) and (VarName116 >= 14(-> 13.8)) and (VarName116 <= 25.5(-> 26)) and (VarName39 <= 2(-> 3)) =>
Output=1 (CF = 0.96)
(VarName139 >= 3(-> 2)) and (VarName45 <= 0.56(-> 1)) and (VarName93 >= 2(-> 1)) and (VarName70 >= 1(-> 0)) and (VarName53 <= 1(-> 2)) => Output=1 (
CF = 0.96)
(VarName139 >= 3(-> 2)) and (VarName90 <= 1(-> 2)) and (VarName106 >= 55.54(-> 52.35)) and (VarName105 >= 42.66(-> 41.34)) and (VarName41 >= 1(->
0)) and (health3 <= 4(-> 5)) => Output=1 (CF = 0.93)
(VarName139 >= 3(-> 2)) and (VarName111 <= 3.4(-> 4)) and (VarName105 >= 52.64(-> 52.34)) and (VarName63 >= 0.526(-> 0)) and (VarName68 <= 1(-> 2))
=> Output=1 (CF = 0.91)
(VarName139 >= 3(-> 2)) and (VarName112 <= 3(-> 4)) and (VarName112 >= 3(-> 2)) and (VarName141 <= 3(-> 4)) => Output=1 (CF = 0.94)
(VarName119 >= 0.475(-> 0)) and (VarName18 <= 3(-> 4)) and (VarName30 >= 2(-> 1)) => Output=1 (CF = 0.94)
(VarName139 >= 3(-> 2)) and (VarName90 <= 0.762(-> 1)) and (VarName21 <= 1(-> 2)) and (VarName116 <= 3.4(-> 7.9)) => Output=1 (CF = 0.84)
(VarName139 >= 3(-> 2)) and (VarName117 >= 93.8(-> 87.5)) and (VarName124 >= 3(-> 1)) => Output=1 (CF = 0.81)
(VarName139 >= 3(-> 2)) and (VarName98 >= 0.376(-> 0)) and (VarName112 <= 3(-> 4)) and (VarName69 >= 1(-> 0)) and (VarName26 <= 1(-> 2)) => Output=1
(CF = 0.9)
(VarName139 >= 3(-> 2)) and (VarName90 <= 0.762(-> 1)) and (VarName68 >= 1(-> 0)) and (pain7 <= 0(-> 1)) and (VarName63 <= 0.526(-> 1)) => Output=1
CF = 0.92)
(VarName119 >= 0.475(-> 0)) and (VarName59 >= 2(-> 1)) and (VarName24 >= 5(-> 4)) and (VarName123 >= 4(-> 3)) => Output=1 (CF = 0.97)
(health <= 0(-> 1)) => Output=3 (CF = 0.67)
(VarName139 <= 2(-> 3)) and (VarName119 <= 0(-> 0.475)) => Output=2 (CF = 1.0)
(VarName108 >= 84.4(-> 83.3)) and (VarName32 >= 1(-> 0)) => Output=2 (CF = 0.95)
(VarName117 >= 62.5(-> 56.3)) and (VarName124 <= 2(-> 3)) and (VarName115 >= 78.6(-> 71.4)) and (VarName105 >= 45.09(-> 44.98)) and (VarName93 >=
1(-> 0)) => Output=2 (CF = 0.99)
(VarName93 <= 0(-> 1)) and (VarName107 <= 4(-> 5)) and (VarName23 <= 4(-> 5)) and (VarName33 >= 1.078(-> 1)) => Output=2 (CF = 0.92)
(VarName91 <= 0(-> 1)) and (pain7 >= 1(-> 0)) and (VarName138 <= 0(-> 1)) and (VarName106 <= 57.16(-> 58.05)) => Output=2 (CF = 0.99)
(VarName93 <= 1(-> 2)) and (VarName130 <= 10.5(-> 10.9)) and (VarName130 >= 9(-> 8.7)) => Output=2 (CF = 0.94)
(VarName26 <= 0(-> 1)) and (VarName94 <= 0(-> 1)) and (VarName111 >= 2(-> 1.1)) and (VarName37 <= 1(-> 2)) => Output=2 (CF = 0.96)
(VarName141 >= 4(-> 3.314)) and (VarName49 >= 1(-> 0)) and (VarName106 >= 54.9(-> 54.36)) => Output=2 (CF = 0.98)

Number of Rules : 26
REGLAS = 26

```

Σχήμα 39

Στο 1 fold έπεσε ο αριθμός των ασαφών κανόνων σε 21.

```

FURIA rules:
=====
(VarName139 >= 3(-> 2)) and (VarName93 >= 2(-> 1)) => Output=1 (CF = 0.78)
(VarName139 >= 3(-> 2)) and (VarName91 >= 1.215(-> 1)) => Output=1 (CF = 0.74)
(VarName139 >= 3(-> 2)) => Output=1 (CF = 0.66)
(VarName139 >= 3(-> 2)) => Output=1 (CF = 0.66)
(VarName139 >= 3(-> 2)) and (VarName50 <= 0(-> 1)) and (VarName94 >= 1(-> 0)) => Output=1 (CF = 0.74)
(VarName139 >= 3(-> 2)) and (VarName105 <= 51.85(-> 52.34)) and (pain6 <= 0(-> 1)) => Output=1 (CF =
0.76)
(VarName139 >= 3(-> 2)) and (VarName91 >= 1.215(-> 1)) and (VarName122 <= 5(-> 6)) => Output=1 (CF =
0.74)
(VarName139 >= 3(-> 2)) => Output=1 (CF = 0.66)
(VarName139 >= 3(-> 2)) => Output=1 (CF = 0.66)
(VarName139 >= 3(-> 2)) => Output=1 (CF = 0.66)
(VarName139 >= 3(-> 2)) => Output=1 (CF = 0.66)
(VarName139 >= 3(-> 0)) => Output=1 (CF = 0.47)
(VarName139 >= 3(-> 2)) => Output=1 (CF = 0.66)
(VarName139 >= 3(-> 2)) => Output=1 (CF = 0.66)
(VarName139 >= 3(-> 2)) => Output=1 (CF = 0.66)
(VarName100 >= 0.307(-> 0)) => Output=1 (CF = 0.66)
=> Output=3 (CF = 0.0)
=> Output=2 (CF = 0.0)
=> Output=2 (CF = 0.0)
=> Output=2 (CF = 0.0)
(VarName108 >= 72.2(-> 69.4)) => Output=2 (CF = 0.79)
(VarName111 <= 12(-> 15)) and (VarName111 >= 5(-> 4)) => Output=2 (CF = 0.68)

```

Σχήμα 40

```

Reglas Buenas
Inside m Ruleset
(VarName139 >= 3(-> 2)) and (VarName93 >= 2(-> 1)) and (VarName108 <= 78.1(-> 80.6)) and (VarName44 <= 0(-> 1)) => Output=1 (CF = 0.96)
(VarName139 >= 3(-> 2)) and (VarName91 >= 1.215(-> 1)) and (VarName38 >= 0.669(-> 0)) and (VarName115 >= 83.3(-> 82.1)) => Output=1 (CF = 0.96)
(VarName139 >= 3(-> 2)) and (VarName113 <= 87.5(-> 100)) and (VarName108 <= 61.1(-> 63.9)) and (VarName78 <= 2(-> 3)) and (VarName142 <= 15(-> 20))
=> Output=1 (CF = 0.97)
(VarName139 >= 3(-> 2)) and (VarName115 <= 75(-> 78.6)) and (VarName73 >= 2(-> 1)) and (VarName97 >= 1(-> 0)) => Output=1 (CF = 0.95)
(VarName139 >= 3(-> 2)) and (VarName50 <= 0(-> 1)) and (VarName94 >= 1(-> 0)) and (VarName42 <= 0(-> 1)) and (VarName105 >= 53.17(-> 53.01)) and (
VarName120 >= 1(-> 0)) => Output=1 (CF = 0.93)
(VarName139 >= 3(-> 2)) and (VarName105 <= 51.85(-> 52.34)) and (pain6 <= 0(-> 1)) and (VarName58 >= 1(-> 0)) and (VarName82 <= 0(-> 1)) and (
VarName70 <= 2(-> 3)) => Output=1 (CF = 0.95)
(VarName139 >= 3(-> 2)) and (VarName91 >= 1.215(-> 1)) and (VarName122 <= 5(-> 6)) and (VarName53 <= 0.353(-> 1)) and (VarName47 >= 2(-> 1)) and (
VarName43 >= 1(-> 0)) => Output=1 (CF = 0.96)
(VarName139 >= 3(-> 2)) and (VarName90 <= 1(-> 2)) and (VarName116 >= 4.9(-> 4.5)) and (VarName103 <= 0(-> 1)) and (health <= 2(-> 3)) and (
VarName91 >= 1(-> 0)) => Output=1 (CF = 0.97)
(VarName139 >= 3(-> 2)) and (VarName111 <= 1.1(-> 2)) and (VarName102 >= 1(-> 0)) and (VarName20 <= 3(-> 4)) => Output=1 (CF = 0.9)
(VarName139 >= 3(-> 2)) and (VarName37 >= 0.682(-> 0)) and (VarName125 <= 0(-> 1)) and (pain5 <= 0(-> 1)) and (VarName105 <= 53.85(-> 54.8)) =>
Output=1 (CF = 0.93)
(VarName139 >= 3(-> 2)) and (VarName115 <= 75(-> 78.6)) and (VarName64 >= 2(-> 1)) and (VarName85 <= 2(-> 3)) and (VarName28 <= 0(-> 1)) => Output=
(CF = 0.94)
(VarName139 >= 3(-> 0)) and (VarName90 <= 0.762(-> 1)) and (VarName141 <= 1(-> 2)) and (VarName92 <= 1(-> 1.689)) => Output=1 (CF = 0.84)
(VarName139 >= 3(-> 2)) and (VarName98 >= 0.376(-> 0)) and (VarName45 >= 3(-> 2)) and (VarName23 >= 3(-> 2)) => Output=1 (CF = 0.95)
(VarName139 >= 3(-> 2)) and (VarName98 >= 0.376(-> 0)) and (VarName130 <= 18.7(-> 18.8)) and (VarName89 >= 0.492(-> 0)) and (VarName27 <= 0(-> 1))
=> Output=1 (CF = 0.93)
(VarName139 >= 3(-> 2)) and (VarName90 <= 0.762(-> 1)) and (VarName75 >= 1(-> 0)) and (VarName113 >= 87.5(-> 86.1)) and (VarName92 >= 2(-> 1)) =>
Output=1 (CF = 0.91)
(VarName100 >= 0.307(-> 0)) and (VarName139 <= 1.413(-> 2)) => Output=1 (CF = 0.76)
(health <= 0(-> 1)) => Output=3 (CF = 0.67)
(VarName139 <= 2(-> 3)) and (VarName119 <= 0(-> 0.475)) => Output=2 (CF = 1.0)
(VarName117 >= 68.8(-> 62.5)) and (VarName91 <= 1.215(-> 2)) and (VarName117 <= 91.7(-> 100)) => Output=2 (CF = 0.9)
(VarName93 <= 1(-> 2)) and (VarName70 >= 3(-> 2)) and (VarName33 >= 3(-> 2)) => Output=2 (CF = 0.95)
(VarName108 >= 72.2(-> 69.4)) and (VarName83 >= 2(-> 1)) and (VarName115 >= 53.6(-> 50)) and (VarName75 >= 2(-> 1)) => Output=2 (CF = 0.94)
(VarName111 <= 12(-> 15)) and (VarName111 >= 5(-> 4)) and (VarName103 >= 1(-> 0)) and (VarName51 <= 0(-> 1)) => Output=2 (CF = 0.93)

Number of Rules : 22

REGLAS = 22

```

### Σχήμα 41

Στο επόμενο fold το 2 είναι αρκετοί οι κανόνες, καθώς έφτασαν τους 36.

```

FURIA rules:
=====

(VarName139 >= 3(-> 2)) => Output=1 (CF = 0.65)
(VarName139 >= 3(-> 2)) => Output=1 (CF = 0.65)
(VarName139 >= 3(-> 2)) => Output=1 (CF = 0.65)
(VarName139 >= 3(-> 2)) and (VarName69 >= 1(-> 0)) and (VarName24 >= 4.46(-> 3)) and (VarName61 <= 0(-> 1)) => Output=1 (CF = 0.85)
(VarName139 >= 3(-> 2)) => Output=1 (CF = 0.65)
(VarName139 >= 3(-> 2)) and (VarName111 <= 1.1(-> 2)) => Output=1 (CF = 0.65)
(VarName139 >= 3(-> 2)) => Output=1 (CF = 0.65)
(VarName139 >= 3(-> 2)) => Output=1 (CF = 0.65)
=> Output=1 (CF = 0.0)
(VarName139 >= 3(-> 2)) and (VarName131 <= 1(-> 3.1)) => Output=1 (CF = 0.63)
(VarName139 >= 3(-> 2)) => Output=1 (CF = 0.65)
(VarName139 >= 3(-> 2)) and (VarName108 >= 97.2(-> 93.8)) => Output=1 (CF = 0.6)
(VarName139 >= 3(-> 2)) and (VarName106 <= 58.89(-> 59.19)) and (VarName83 <= 0(-> 1)) => Output=1 (CF = 0.65)
=> Output=1 (CF = 0.0)
=> Output=1 (CF = 0.0)
(VarName139 >= 3(-> 2)) and (health <= 1.138(-> 2)) and (VarName125 <= 2(-> 3)) => Output=1 (CF = 0.73)
(VarName139 >= 3(-> 2)) => Output=1 (CF = 0.65)
(VarName139 >= 3(-> 2)) => Output=1 (CF = 0.65)
=> Output=3 (CF = 0.0)
=> Output=2 (CF = 0.0)
(VarName117 >= 68.8(-> 62.5)) and (VarName107 <= 2(-> 3)) => Output=2 (CF = 0.91)
(VarName108 >= 75(-> 69.4)) => Output=2 (CF = 0.8)
(VarName93 <= 1(-> 3)) and (VarName110 >= 85.7(-> 82.1)) and (VarName113 >= 86.1(-> 83.3)) and (VarName99 <= 0(-> 0.34)) => Output=2 (CF = 0.98)
(VarName93 <= 1(-> 2)) and (VarName110 >= 85.7(-> 82.1)) => Output=2 (CF = 0.85)
(VarName93 <= 1(-> 2)) and (VarName91 <= 1.215(-> 2)) and (VarName141 >= 4(-> 3.314)) => Output=2 (CF = 0.97)
(VarName93 <= 1(-> 2)) and (VarName67 <= 0(-> 1)) and (VarName110 >= 100(-> 96.4)) => Output=2 (CF = 0.94)
=> Output=2 (CF = 0.0)
=> Output=2 (CF = 0.0)
(VarName141 >= 4(-> 3.314)) => Output=2 (CF = 0.9)
(VarName70 <= 0.669(-> 1)) and (VarName25 <= 0(-> 1)) => Output=2 (CF = 0.87)
(VarName112 <= 1(-> 8)) and (VarName37 <= 1(-> 2)) and (VarName35 <= 0(-> 1)) => Output=2 (CF = 0.89)
(VarName58 <= 0(-> 1)) and (VarName105 >= 47.38(-> 42.48)) => Output=2 (CF = 0.87)
=> Output=2 (CF = 0.0)
=> Output=2 (CF = 0.0)
(VarName95 <= 0(-> 1)) => Output=2 (CF = 0.93)
=> Output=2 (CF = 0.0)

```

### Σχήμα 42



```

Reglas Buenas
inside #ruleset
(VarName139 >= 3(-> 2)) and (VarName108 <= 61.1(-> 62.5)) => Output=1 (CF = 0.87)
(VarName139 >= 3(-> 2)) and (VarName117 <= 62.5(-> 68.8)) and (VarName90 <= 1(-> 2)) and (VarName24 >= 4.46(-> 4)) and (pain6 <= 0(-> 1)) and (
VarName94 >= 1(-> 0)) => Output=1 (CF = 0.97)
(VarName139 >= 3(-> 2)) and (VarName101 >= 0.481(-> 0)) and (VarName45 >= 3(-> 2)) => Output=1 (CF = 0.9)
(VarName139 >= 3(-> 2)) and (VarName69 >= 1(-> 0)) and (VarName24 >= 4.46(-> 3)) and (VarName61 <= 0(-> 1)) and (VarName62 >= 1(-> 0)) => Output=1 (
CF = 0.94)
(VarName139 >= 3(-> 2)) and (VarName101 >= 0.481(-> 0)) and (VarName97 >= 1(-> 0)) and (VarName26 <= 1(-> 2)) and (VarName56 >= 0.512(-> 0)) and (
VarName140 >= 2(-> 1)) => Output=1 (CF = 0.94)
(VarName139 >= 3(-> 2)) and (VarName111 <= 1.1(-> 2)) and (VarName110 <= 96.4(-> 100)) and (VarName58 >= 1(-> 0)) => Output=1 (CF = 0.9)
(VarName139 >= 3(-> 2)) and (VarName91 >= 2(-> 1.215)) and (VarName106 >= 61.27(-> 60.03)) and (VarName53 <= 0.353(-> 1)) and (pain1 >= 1(-> 0.667))
=> Output=1 (CF = 0.95)
(VarName139 >= 3(-> 2)) and (VarName108 <= 75(-> 77.8)) and (VarName48 <= 0(-> 0.539)) and (VarName112 >= 1(-> 0)) => Output=1 (CF = 0.95)
(VarName139 >= 4(-> 3)) and (VarName111 <= 1.1(-> 2)) and (VarName123 <= 0(-> 2)) => Output=1 (CF = 0.81)
(VarName139 >= 3(-> 2)) and (VarName131 <= 1(-> 3.1)) and (VarName106 >= 63.09(-> 62.99)) => Output=1 (CF = 0.87)
(VarName139 >= 3(-> 2)) and (VarName122 >= 5(-> 4)) and (health1 >= 3(-> 2)) and (VarName108 >= 66.7(-> 64.3)) and (VarName124 <= 5(-> 6)) =>
Output=1 (CF = 0.93)
(VarName139 >= 3(-> 2)) and (VarName108 >= 97.2(-> 93.8)) and (VarName144 >= 1(-> 0)) and (VarName78 >= 0.54(-> 0)) and (health3 >= 4(-> 3)) =>
Output=1 (CF = 0.91)
(VarName139 >= 3(-> 2)) and (VarName106 <= 58.89(-> 59.19)) and (VarName83 <= 0(-> 1)) and (VarName130 >= 4(-> 3)) and (VarName106 >= 51.83(->
49.11)) and (VarName35 <= 0.799(-> 1)) => Output=1 (CF = 0.92)
(VarName119 >= 0.475(-> 0)) and (VarName18 <= 3(-> 4)) and (VarName66 <= 1(-> 2)) => Output=1 (CF = 0.92)
(VarName119 >= 0.475(-> 0)) and (VarName119 <= 0.475(-> 1)) and (VarName25 >= 1(-> 0)) => Output=1 (CF = 0.82)
(VarName139 >= 3(-> 2)) and (health <= 1.138(-> 2)) and (VarName125 <= 2(-> 3)) and (VarName18 <= 4.357(-> 5)) => Output=1 (CF = 0.8)
(VarName139 >= 3(-> 2)) and (VarName85 >= 1(-> 0)) and (VarName130 <= 15.1(-> 16)) and (VarName106 <= 57.06(-> 57.16)) => Output=1 (CF = 0.9)
(VarName139 >= 3(-> 2)) and (VarName88 >= 2(-> 1)) and (VarName105 <= 36.46(-> 39.32)) and (health >= 3(-> 2)) => Output=1 (CF = 0.86)
(health <= 0(-> 1)) => Output=3 (CF = 0.67)
(VarName139 <= 2(-> 3)) and (VarName119 <= 0(-> 0.475)) => Output=2 (CF = 1.0)
(VarName117 >= 68.8(-> 62.5)) and (VarName107 <= 2(-> 3)) and (VarName108 <= 96.4(-> 97.2)) and (VarName106 <= 55.09(-> 57.16)) and (VarName84 <=
0(-> 1)) => Output=2 (CF = 0.99)
(VarName108 >= 75(-> 69.4)) and (VarName69 <= 0.546(-> 1)) and (VarName37 <= 0(-> 1)) and (VarName83 >= 1(-> 0)) and (VarName130 >= 24(-> 23)) =>
Output=2 (CF = 0.97)
(VarName93 <= 1(-> 3)) and (VarName110 >= 85.7(-> 82.1)) and (VarName113 >= 86.1(-> 83.3)) and (VarName99 <= 0(-> 0.34)) and (VarName109 >= 1(-> 0))
=> Output=2 (CF = 1.0)
(VarName93 <= 1(-> 2)) and (VarName110 >= 85.7(-> 82.1)) and (VarName111 >= 7(-> 6)) and (VarName59 <= 0(-> 1)) => Output=2 (CF = 0.98)
(VarName93 <= 1(-> 2)) and (VarName91 <= 1.215(-> 2)) and (VarName141 >= 4(-> 3.314)) and (VarName83 >= 1(-> 0)) and (health >= 2(-> 1)) => Output=2
(CF = 0.99)
(VarName93 <= 1(-> 2)) and (VarName67 <= 0(-> 1)) and (VarName110 >= 100(-> 96.4)) and (VarName106 >= 59.6(-> 58.72)) => Output=2 (CF = 0.99)
(VarName90 >= 2(-> 1)) and (VarName25 <= 1(-> 2)) and (VarName61 >= 1(-> 0)) and (VarName106 <= 55.19(-> 56.08)) and (VarName23 <= 4(-> 5)) =>
Output=2 (CF = 0.97)
(VarName70 >= 3(-> 2)) and (VarName110 <= 85.7(-> 89.3)) and (VarName93 <= 1(-> 2)) and (VarName29 <= 1(-> 2)) => Output=2 (CF = 0.96)
(VarName141 >= 4(-> 3.314)) and (VarName47 >= 2(-> 1)) => Output=2 (CF = 0.96)
(VarName70 <= 0.669(-> 1)) and (VarName25 <= 0(-> 1)) and (VarName90 >= 2(-> 1)) and (VarName45 <= 0.56(-> 1)) => Output=2 (CF = 0.98)

```

### Σχήμα 43

```

(VarName112 <= 1(-> 8)) and (VarName37 <= 1(-> 2)) and (VarName35 <= 0(-> 1)) and (VarName124 >= 4(-> 3)) and (pain3 <= 0(-> 1)) => Output=2 (CF =
0.98)
(VarName58 <= 0(-> 1)) and (VarName105 >= 47.38(-> 42.48)) and (VarName100 >= 1(-> 0)) and (VarName75 <= 0(-> 1)) and (pain1 <= 0(-> 1)) => Output=2
(CF = 0.96)
(VarName52 >= 2(-> 1)) and (VarName90 <= 0(-> 0.762)) => Output=2 (CF = 0.93)
(VarName57 >= 2(-> 1)) and (VarName22 <= 2(-> 3)) and (VarName122 >= 6(-> 5)) and (VarName40 <= 3(-> 4)) => Output=2 (CF = 0.93)
(VarName95 <= 0(-> 1)) and (health >= 3(-> 2)) and (VarName124 <= 2(-> 3)) => Output=2 (CF = 0.99)
(VarName130 >= 63(-> 62.5)) and (VarName75 <= 3(-> 4)) => Output=2 (CF = 0.85)

Number of Rules : 36
REGLAS = 36

```

### Σχήμα 44



Στο 3 fold οι κανόνες πήγαν στους 30.

```
FURIA rules:
=====
(VarName139 >= 3(-> 2)) => Output=1 (CF = 0.65)
(VarName139 >= 3(-> 2)) => Output=1 (CF = 0.65)
(VarName139 >= 3(-> 2)) => Output=1 (CF = 0.65)
(VarName139 >= 3(-> 2)) => Output=1 (CF = 0.65)
(VarName139 >= 3(-> 2)) => Output=1 (CF = 0.65)
(VarName139 >= 3(-> 2)) => Output=1 (CF = 0.65)
(VarName139 >= 3(-> 2) and (VarName90 <= 0.762(-> 1)) => Output=1 (CF = 0.69)
(VarName139 >= 3(-> 2) and (VarName50 <= 0(-> 1)) => Output=1 (CF = 0.65)
(VarName139 >= 3(-> 2)) => Output=1 (CF = 0.65)
(VarName139 >= 3(-> 2)) => Output=1 (CF = 0.65)
(VarName139 >= 3(-> 2)) => Output=1 (CF = 0.65)
(VarName139 >= 3(-> 2) and (VarName88 <= 0.526(-> 1)) => Output=1 (CF = 0.64)
(VarName139 >= 4(-> 3)) => Output=1 (CF = 0.67)
(VarName139 >= 3(-> 2)) => Output=1 (CF = 0.65)
(VarName139 >= 3(-> 2) and (VarName101 <= 0(-> 1)) and (VarName106 >= 55.12(-> 52.4)) and (VarName95 >= 2(-> 1)) => Output=1 (CF = 0.74)
(VarName100 >= 0.307(-> 0)) and (VarName30 >= 0.565(-> 0)) and (pain6 <= 0(-> 1)) => Output=1 (CF = 0.83)
(VarName141 <= 3.314(-> 4)) and (VarName100 >= 0.307(-> 0)) => Output=1 (CF = 0.69)
(VarName139 >= 3(-> 2) and (VarName105 >= 50.6(-> 50.22)) and (VarName106 >= 57.34(-> 57.16)) and (VarName90 <= 0(-> 0.762)) =>
Output=1 (CF = 0.7)
=> Output=3 (CF = 0.0)
=> Output=2 (CF = 0.0)
(VarName117 >= 68.8(-> 62.5)) => Output=2 (CF = 0.88)
=> Output=2 (CF = 0.0)
=> Output=2 (CF = 0.0)
(VarName117 >= 68.8(-> 62.5)) => Output=2 (CF = 0.88)
(VarName81 <= 0(-> 0.517)) and (VarName105 >= 49.81(-> 48.21)) => Output=2 (CF = 0.87)
=> Output=2 (CF = 0.0)
=> Output=2 (CF = 0.0)
=> Output=2 (CF = 0.0)
=> Output=2 (CF = 0.0)
=> Output=2 (CF = 0.0)
```

Σχήμα 45

```

Reglas Buenas
Inside m_Ruleset
(VarName139 >= 3(-> 2)) and (VarName117 <= 62.5(-> 68.8)) and (VarName32 >= 2(-> 1)) and (VarName61 <= 0(-> 1)) and (VarName25 >= 1(-> 0)) => Output=1 (CF = 0.96)
(VarName139 >= 3(-> 2)) and (VarName101 >= 0.481(-> 0)) and (VarName62 >= 1(-> 0)) and (VarName130 <= 24(-> 24.5)) and (VarName24 >= 4(-> 3)) => Output=1 (CF = 0.95)
(VarName139 >= 3(-> 2)) and (VarName117 <= 62.5(-> 68.8)) and (VarName55 >= 2(-> 1)) and (pain5 <= 0(-> 1)) => Output=1 (CF = 0.96)
(VarName139 >= 3(-> 2)) and (VarName115 <= 71.4(-> 75)) and (VarName58 >= 2(-> 1)) and (VarName96 <= 6(-> 7)) and (VarName36 <= 0(-> 1)) => Output=1 (CF = 0.95)
(VarName139 >= 3(-> 2)) and (VarName37 >= 0.682(-> 0)) and (VarName106 >= 54.45(-> 54.42)) and (VarName56 >= 1(-> 0.512)) and (VarName31 >= 1(-> 0)) and (VarName38 <= 1(-> 2)) => Output=1 (CF = 0.95)
(VarName139 >= 3(-> 2)) and (VarName101 >= 1(-> 0)) and (VarName83 <= 1(-> 2)) and (VarName53 >= 0.353(-> 0)) and (VarName94 <= 1(-> 2)) and (VarName37 <= 1(-> 2)) => Output=1 (CF = 0.95)
(VarName139 >= 3(-> 2)) and (VarName90 <= 0.762(-> 1)) and (VarName37 >= 0.682(-> 0)) and (VarName33 <= 2(-> 3)) and (VarName112 <= 8(-> 9)) and (health3 >= 3(-> 2)) => Output=1 (CF = 0.94)
(VarName139 >= 3(-> 2)) and (VarName50 <= 0(-> 1)) and (VarName117 <= 56.3(-> 62.5)) and (VarName110 <= 85.7(-> 89.3)) and (VarName74 <= 0(-> 1)) => Output=1 (CF = 0.94)
(VarName139 >= 3(-> 2)) and (VarName111 <= 1.1(-> 2)) and (VarName102 >= 1(-> 0)) and (health3 >= 3(-> 2)) => Output=1 (CF = 0.88)
(VarName139 >= 3(-> 2)) and (VarName94 >= 3(-> 2)) and (VarName85 <= 1(-> 2)) => Output=1 (CF = 0.95)
(VarName139 >= 3(-> 2)) and (VarName38 >= 1(-> 0)) and (VarName106 >= 60.55(-> 60.03)) and (VarName71 >= 2(-> 1)) => Output=1 (CF = 0.87)
(VarName139 >= 3(-> 2)) and (VarName88 <= 0.526(-> 1)) and (VarName91 >= 2(-> 1.215)) and (VarName96 <= 1(-> 2)) and (VarName115 >= 85.7(-> 78.6)) => Output=1 (CF = 0.92)
(VarName139 >= 4(-> 3)) and (VarName131 <= 2.1(-> 3)) and (VarName23 <= 4(-> 5)) => Output=1 (CF = 0.91)
(VarName139 >= 3(-> 2)) and (VarName19 <= 4.568(-> 5)) and (VarName110 <= 82.1(-> 85.7)) and (VarName96 <= 5(-> 6)) and (health >= 3(-> 2)) => Output=1 (CF = 0.95)
(VarName139 >= 3(-> 2)) and (VarName101 <= 0(-> 1)) and (VarName106 >= 55.12(-> 52.4)) and (VarName95 >= 2(-> 1)) and (VarName35 <= 0.799(-> 1)) and (VarName30 <= 1(-> 2)) => Output=1 (CF = 0.92)
(VarName100 >= 0.307(-> 0)) and (VarName30 >= 0.565(-> 0)) and (pain6 <= 0(-> 1)) and (VarName105 <= 47.35(-> 47.77)) and (VarName70 <= 1(-> 2)) => Output=1 (CF = 0.96)
(VarName141 <= 3.314(-> 4)) and (VarName100 >= 0.307(-> 0)) and (VarName119 <= 0.475(-> 1)) and (VarName26 >= 2(-> 1)) => Output=1 (CF = 0.86)
(VarName139 >= 3(-> 2)) and (VarName105 >= 50.6(-> 50.22)) and (VarName106 >= 57.34(-> 57.16)) and (VarName90 <= 0(-> 0.762)) and (VarName122 <= 1(-> 2)) => Output=1 (CF = 0.86)
(health <= 0(-> 1)) => Output=3 (CF = 0.67)
(VarName139 <= 2(-> 3)) and (VarName119 <= 0(-> 0.475)) => Output=2 (CF = 1.0)

```

Σχήμα 46

```

VarName122 <= 1(-> 2)) => Output=1 (CF = 0.86)
(health <= 0(-> 1)) => Output=3 (CF = 0.67)
(VarName139 <= 2(-> 3)) and (VarName119 <= 0(-> 0.475)) => Output=2 (CF = 1.0)
(VarName117 >= 68.8(-> 62.5)) and (VarName112 <= 2(-> 6)) and (VarName131 >= 3(-> 2.3)) and (VarName106 <= 55.09(-> 55.19)) and (VarName66 <= 2(-> 3)) => Output=2 (CF = 0.99)
(VarName108 >= 72.2(-> 71.9)) and (VarName115 >= 78.6(-> 75)) and (VarName88 >= 1(-> 0.526)) and (VarName111 >= 7(-> 6.8)) and (VarName45 <= 1(-> 2)) and (VarName26 >= 1(-> 0)) => Output=2 (CF = 0.98)
(VarName25 <= 0(-> 2)) and (VarName70 <= 0.669(-> 1)) and (VarName139 <= 3(-> 4)) and (VarName123 >= 5(-> 4)) => Output=2 (CF = 0.95)
(VarName117 >= 68.8(-> 62.5)) and (VarName108 >= 83.3(-> 75)) and (VarName105 <= 46.99(-> 47.06)) and (VarName76 >= 1(-> 0)) => Output=2 (CF = 0.95)
(VarName81 <= 0(-> 0.517)) and (VarName105 >= 49.81(-> 48.21)) and (VarName78 >= 0.54(-> 0)) and (VarName91 <= 1.215(-> 2)) and (VarName45 >= 0.56(-> 0)) => Output=2 (CF = 0.98)
(VarName58 <= 0(-> 1)) and (VarName42 <= 0(-> 1)) and (VarName106 >= 49.11(-> 45.38)) and (VarName125 >= 3(-> 2)) and (VarName80 <= 0(-> 1)) => Output=2 (CF = 0.99)
(VarName125 <= 0(-> 1)) and (VarName37 <= 0(-> 2)) and (health >= 2(-> 1)) and (VarName32 <= 0(-> 1)) and (VarName93 >= 1(-> 0)) => Output=2 (CF = 0.99)
(VarName54 >= 1(-> 0)) and (VarName108 >= 69.4(-> 68.8)) and (VarName105 >= 46.83(-> 43.18)) and (VarName45 >= 1(-> 0)) => Output=2 (CF = 0.98)
(VarName125 <= 1(-> 2)) and (VarName131 <= 5(-> 6)) and (health >= 3(-> 2)) => Output=2 (CF = 0.99)
(VarName141 >= 3.314(-> 3)) and (VarName93 <= 0(-> 1)) and (VarName106 <= 57.06(-> 57.33)) and (health1 >= 3(-> 2)) and (VarName22 <= 3(-> 4)) => Output=2 (CF = 1.0)
Number of Rules : 30

```

Σχήμα 47

Και τέλος στο 4 fold δημιουργήθηκαν 27 ασαφείς κανόνες.

```
FURIA rules:
=====

(VarName139 >= 3(-> 2)) and (VarName117 <= 62.5(-> 68.8)) => Output=1 (CF = 0.75)
(VarName139 >= 3(-> 2)) and (VarName117 <= 62.5(-> 68.8)) => Output=1 (CF = 0.75)
(VarName139 >= 3(-> 2)) => Output=1 (CF = 0.65)
(VarName139 >= 3(-> 2)) and (VarName117 <= 62.5(-> 68.8)) => Output=1 (CF = 0.75)
(VarName139 >= 3(-> 2)) and (VarName106 >= 55.12(-> 55.07)) => Output=1 (CF = 0.69)
(VarName139 >= 3(-> 2)) => Output=1 (CF = 0.65)
(VarName139 >= 3(-> 2)) => Output=1 (CF = 0.65)
(VarName139 >= 3(-> 2)) and (VarName7 >= 1(-> 0)) => Output=1 (CF = 0.75)
(VarName139 >= 3(-> 2)) => Output=1 (CF = 0.65)
(VarName139 >= 3(-> 2)) => Output=1 (CF = 0.65)
(VarName139 >= 3(-> 2)) => Output=1 (CF = 0.65)
(VarName139 >= 3(-> 2)) => Output=1 (CF = 0.65)
(VarName139 >= 3(-> 2)) => Output=1 (CF = 0.65)
(VarName139 >= 3(-> 2)) => Output=1 (CF = 0.65)
(VarName119 >= 0.475(-> 0)) => Output=1 (CF = 0.66)
(VarName139 >= 3(-> 2)) and (VarName106 >= 57.16(-> 56.92)) and (VarName105 >= 42.66(-> 42.51)) and (VarName117 <= 56.3(-> 62.5))
=> Output=1 (CF = 0.88)
(VarName141 <= 3.314(-> 4)) => Output=1 (CF = 0.56)
(VarName119 >= 0.475(-> 0)) => Output=1 (CF = 0.66)
(VarName100 >= 1(-> 0)) and (pain6 <= 0(-> 1)) and (VarName97 >= 1(-> 0)) => Output=1 (CF = 0.78)
=> Output=3 (CF = 0.0)
=> Output=2 (CF = 0.0)
=> Output=2 (CF = 0.0)
=> Output=2 (CF = 0.0)
(VarName25 <= 0(-> 1)) and (VarName98 <= 0(-> 1)) => Output=2 (CF = 0.91)
=> Output=2 (CF = 0.0)
(VarName125 <= 2(-> 3)) and (VarName36 <= 0(-> 0.601)) => Output=2 (CF = 0.89)
(VarName105 >= 43.7(-> 36.46)) => Output=2 (CF = 0.77)
```

Σχήμα 48

```
Reglas Buenas
Inside m_Ruleset
(VarName139 >= 3(-> 2)) and (VarName117 <= 62.5(-> 68.8)) and (VarName55 >= 2(-> 1)) and (VarName70 <= 0(-> 1)) => Output=1
(CF = 0.96)
(VarName139 >= 3(-> 2)) and (VarName117 <= 62.5(-> 68.8)) and (VarName70 >= 2(-> 1)) and (VarName110 >= 89.3(-> 85.7)) and (
VarName58 >= 1(-> 0)) and (VarName94 >= 1(-> 0)) => Output=1 (CF = 0.96)
(VarName139 >= 3(-> 2)) and (VarName110 <= 80(-> 82.1)) and (VarName91 >= 2(-> 1.215)) and (VarName54 <= 1(-> 2)) and (
VarName125 >= 4(-> 3)) and (VarName70 <= 3(-> 4)) => Output=1 (CF = 0.97)
(VarName139 >= 3(-> 2)) and (VarName117 <= 62.5(-> 68.8)) and (VarName19 <= 3(-> 4)) and (VarName49 <= 1(-> 2)) => Output=1
(CF = 0.96)
(VarName139 >= 3(-> 2)) and (VarName106 >= 55.12(-> 55.07)) and (VarName92 >= 4(-> 3)) and (VarName68 <= 1(-> 2)) =>
Output=1 (CF = 0.95)
(VarName139 >= 3(-> 2)) and (VarName23 >= 5(-> 4)) and (VarName91 >= 1.215(-> 1)) and (VarName90 <= 1(-> 2)) and (VarName66
>= 2(-> 1)) and (VarName24 >= 5(-> 4)) => Output=1 (CF = 0.97)
(VarName139 >= 3(-> 2)) and (VarName105 <= 30.23(-> 31.72)) and (VarName142 <= 6(-> 15)) => Output=1 (CF = 0.92)
(VarName139 >= 3(-> 2)) and (VarName7 >= 1(-> 0)) and (VarName32 >= 2(-> 1)) and (VarName115 >= 92.9(-> 89.3)) => Output=1 (
CF = 0.92)
(VarName139 >= 3(-> 2)) and (VarName65 >= 2(-> 1)) and (health1 >= 2.458(-> 2)) and (VarName70 >= 2(-> 1)) => Output=1 (CF
= 0.94)
(VarName139 >= 3(-> 2)) and (VarName88 <= 0.526(-> 1)) and (VarName106 >= 56.64(-> 56.18)) and (VarName115 <= 92.9(->
96.4)) and (VarName67 <= 1(-> 2)) and (VarName108 <= 97.2(-> 100)) => Output=1 (CF = 0.95)
(VarName139 >= 3(-> 2)) and (VarName131 <= 2.3(-> 3)) and (VarName131 >= 1.1(-> 1)) and (VarName26 <= 0.684(-> 1)) =>
Output=1 (CF = 0.88)
(VarName139 >= 3(-> 2)) and (VarName110 <= 80(-> 85.7)) and (VarName73 >= 2(-> 1)) and (pain6 <= 0(-> 1)) => Output=1 (CF =
0.94)
(VarName139 >= 3(-> 2)) and (VarName125 >= 4(-> 3)) and (VarName111 <= 2(-> 2.1)) and (VarName96 >= 1(-> 0)) and (
VarName105 >= 44.98(-> 42.11)) => Output=1 (CF = 0.92)
(VarName139 >= 3(-> 2)) and (VarName110 <= 87.5(-> 89.3)) and (VarName70 <= 0(-> 0.669)) and (health4 <= 4(-> 5)) and (
VarName54 <= 0(-> 1)) and (VarName32 >= 1(-> 0)) => Output=1 (CF = 0.95)
(VarName119 >= 0.475(-> 0)) and (VarName89 >= 0.492(-> 0)) and (VarName113 <= 61.1(-> 63.9)) and (VarName21 <= 3(-> 4)) =>
Output=1 (CF = 0.96)
(VarName139 >= 3(-> 2)) and (VarName106 >= 57.16(-> 56.92)) and (VarName105 >= 42.66(-> 42.51)) and (VarName117 <= 56.3(->
62.5)) and (VarName117 >= 50(-> 43.8)) and (health >= 2(-> 1)) => Output=1 (CF = 0.94)
```

Σχήμα 49

```

(VarName141 <= 3.314(-> 4)) and (VarName93 <= 0(-> 1)) and (pain3 >= 1(-> 0)) and (VarName27 >= 1(-> 0)) and (VarName24 >=
5(-> 4)) => Output=1 (CF = 0.92)
(VarName119 >= 0.475(-> 0)) and (VarName106 >= 61.04(-> 59.6)) and (VarName99 <= 0(-> 1)) and (pain6 >= 1(-> 0)) =>
Output=1 (CF = 0.82)
(VarName100 >= 1(-> 0)) and (pain6 <= 0(-> 1)) and (VarName97 >= 1(-> 0)) and (VarName117 >= 62.5(-> 56.3)) and (health2 <=
2(-> 3)) and (VarName34 >= 1(-> 0)) => Output=1 (CF = 0.92)
(health <= 0(-> 1)) => Output=3 (CF = 0.67)
(VarName139 <= 2(-> 3)) and (VarName119 <= 0(-> 0.475)) => Output=2 (CF = 1.0)
(VarName117 >= 68.8(-> 62.5)) and (VarName91 <= 1.215(-> 2)) and (VarName117 <= 91.7(-> 92.9)) and (VarName106 <= 56.18(->
56.23)) and (VarName26 >= 0.684(-> 0)) => Output=2 (CF = 0.99)
(VarName117 >= 62.5(-> 56.3)) and (VarName110 >= 92.9(-> 89.3)) and (VarName112 <= 2(-> 3)) and (VarName131 >= 3(-> 2.3))
=> Output=2 (CF = 0.96)
(VarName25 <= 0(-> 1)) and (VarName98 <= 0(-> 1)) and (VarName22 >= 3(-> 2.365)) and (VarName106 >= 50.45(-> 49.14)) =>
Output=2 (CF = 0.97)
(VarName93 <= 1(-> 2)) and (VarName141 >= 4(-> 3.314)) and (VarName58 <= 0(-> 1)) and (VarName95 >= 1(-> 0)) => Output=2 (
CF = 1.0)
(VarName125 <= 2(-> 3)) and (VarName36 <= 0(-> 0.601)) and (VarName24 <= 4(-> 5)) and (VarName91 >= 2(-> 1.215)) =>
Output=2 (CF = 0.97)
(VarName105 >= 43.7(-> 36.46)) and (VarName102 >= 1(-> 0)) and (VarName114 >= 3(-> 2)) and (VarName69 <= 0.546(-> 1)) =>
Output=2 (CF = 0.9)

Number of Rules : 27

REGLAS = 27

```

Σχήμα 50

Στο dataset2 και dataset3 είχαμε 0% απόδοση και στα 5 folds, το οποίο μάλλον συνέβη καθώς ξεπεράσαμε τον αριθμό δεδομένων, που μπορεί το Keel να διαχειριστεί.

Τώρα ακολουθεί το Dataset2 με 1727 καταχωρήσεις.

Στο 0 fold προκύπτουν 27 ασαφείς κανόνες.

```

FURIA rules:
=====
(VarName142 >= 2.25(-> 2)) => Output=1 (CF = 0.66)
(VarName142 >= 2.25(-> 2)) => Output=1 (CF = 0.66)
(VarName142 >= 2.25(-> 2)) => Output=1 (CF = 0.66)
(VarName142 >= 2.25(-> 2)) => Output=1 (CF = 0.66)
(VarName142 >= 2.25(-> 2)) => Output=1 (CF = 0.66)
(VarName142 >= 2.25(-> 2)) => Output=1 (CF = 0.66)
(VarName142 >= 2.25(-> 2)) => Output=1 (CF = 0.66)
(VarName142 >= 2.25(-> 2)) and (VarName138 >= 2(-> 1)) and (VarName75 <= 1.25(-> 2)) => Output=1 (CF = 0.7)
(VarName142 >= 2.25(-> 2)) => Output=1 (CF = 0.66)
(VarName142 >= 2.25(-> 2)) => Output=1 (CF = 0.66)
(VarName142 >= 2.25(-> 2)) => Output=1 (CF = 0.66)
(VarName142 >= 2.25(-> 2)) and (VarName106 >= 49.27(-> 47.65)) => Output=1 (CF = 0.63)
(VarName142 >= 2.25(-> 2)) => Output=1 (CF = 0.66)
(VarName142 >= 2.25(-> 2)) => Output=1 (CF = 0.66)
(VarName142 >= 2.25(-> 2)) and (VarName59 <= 0.25(-> 1)) and (health1 >= 2.75(-> 2)) => Output=1 (CF = 0.63)
(VarName142 >= 2.25(-> 2)) => Output=1 (CF = 0.66)
(VarName142 >= 2.25(-> 2)) => Output=1 (CF = 0.66)
(VarName142 >= 2.25(-> 2)) and (VarName124 <= 2(-> 2.75)) => Output=1 (CF = 0.65)
(VarName142 >= 2.25(-> 2)) => Output=1 (CF = 0.66)
(VarName142 >= 2.25(-> 2)) => Output=1 (CF = 0.66)
(VarName142 >= 3(-> 2)) and (VarName45 <= 0.5(-> 1)) => Output=1 (CF = 0.65)
(VarName142 >= 3(-> 2)) and (VarName75 <= 0(-> 1)) => Output=1 (CF = 0.65)
=> Output=2 (CF = 0.0)
(VarName111 >= 85.7(-> 82.1)) and (VarName69 <= 0.75(-> 1)) => Output=2 (CF = 0.8)
(VarName119 >= 75(-> 68.8)) and (VarName94 >= 1(-> 0)) => Output=2 (CF = 0.93)
(VarName142 <= 4.75(-> 5)) => Output=2 (CF = 0.76)

```

Σχήμα 51

```

Reglas Buenas
Inside m_Ruleset
(VarName142 >= 2.25(-> 2)) and (VarName111 <= 83.3(-> 85.7)) and (VarName101 >= 0.5(-> 0)) and (VarName91 <= 1.667(-> 1.75)) and (VarName104 <= 0(-> 1)) and (VarName59 >= 2(-> 1)) => Output=1 (CF = 0.96)
(VarName142 >= 2.25(-> 2)) and (VarName96 >= 2(-> 1)) and (VarName31 >= 0.75(-> 0)) and (VarName30 <= 1.333(-> 2)) and (pain5 >= 1(-> 0)) and (VarName119 >= 37.5(-> 31.3)) => Output=1 (CF = 0.97)
(VarName142 >= 2.25(-> 2)) and (VarName125 >= 6(-> 5)) and (VarName31 >= 2(-> 1)) and (VarName93 >= 3.333(-> 3)) => Output=1 (CF = 0.92)
(VarName142 >= 2.25(-> 2)) and (pain7 <= 0(-> 1)) and (VarName136 >= 11(-> 10.4)) and (VarName52 <= 0(-> 1)) and (VarName126 >= 3(-> 2)) and (VarName145 <= 7(-> 10)) => Output=1 (CF = 0.97)
(VarName142 >= 2.25(-> 2)) and (VarName119 <= 68.8(-> 75)) and (VarName20 <= 2(-> 3)) and (health2 <= 2.5(-> 3)) and (VarName104 <= 0(-> 1)) and (VarName78 <= 1.75(-> 2)) and (VarName109 >= 75(-> 72.2)) and (VarName92 >= 0.667(-> 0.25)) => Output=1 (CF = 0.97)
(VarName142 >= 2.25(-> 2)) and (VarName97 >= 4(-> 3)) and (VarName107 >= 53.16(-> 52.98)) and (VarName46 >= 2(-> 1)) and (VarName94 <= 3(-> 4)) => Output=1 (CF = 0.96)
(VarName142 >= 2.25(-> 2)) and (VarName64 >= 0.25(-> 0)) and (VarName117 <= 21.9(-> 31)) and (VarName102 >= 1(-> 0)) and (VarName83 <= 0.5(-> 1)) and (VarName79 <= 1(-> 2)) => Output=1 (CF = 0.97)
(VarName142 >= 2.25(-> 2)) and (VarName86 >= 0.333(-> 0)) and (VarName42 <= 1(-> 2)) and (VarName31 >= 1(-> 0)) and (VarName55 >= 2(-> 1)) => Output=1 (CF = 0.95)
(VarName142 >= 2.25(-> 2)) and (VarName138 >= 2(-> 1)) and (VarName75 <= 1.25(-> 2)) and (VarName86 >= 0.333(-> 0.25)) and (VarName110 <= 2(-> 3)) and (VarName53 >= 0.25(-> 0)) and (VarName21 <= 2(-> 3)) => Output=1 (CF = 0.97)
(VarName142 >= 2.25(-> 2)) and (VarName124 >= 6(-> 5)) and (VarName80 <= 2(-> 3)) and (VarName145 <= 4(-> 5)) and (VarName74 >= 2(-> 1)) and (VarName95 <= 1(-> 2)) => Output=1 (CF = 0.95)
(VarName142 >= 2.25(-> 2)) and (VarName145 >= 8(-> 7)) and (VarName24 >= 4.75(-> 4)) and (pain5 <= 0(-> 1)) and (VarName100 <= 0(-> 0.75)) => Output=1 (CF = 0.95)
(VarName142 >= 2.25(-> 2)) and (VarName100 >= 1(-> 0)) and (health3 <= 2(-> 3)) and (health4 <= 1(-> 2)) => Output=1 (CF = 0.95)
(VarName142 >= 2.25(-> 2)) and (VarName106 >= 49.27(-> 47.65)) and (VarName20 >= 2(-> 1)) and (VarName107 >= 55.44(-> 55.36)) and (VarName78 <= 0(-> 1)) => Output=1 (CF = 0.9)
(VarName142 >= 2.25(-> 2)) and (VarName119 <= 50(-> 56.3)) and (VarName53 <= 0.25(-> 0.333)) and (VarName32 >= 1(-> 0)) and (VarName45 <= 1(-> 2)) => Output=1 (CF = 0.95)
(VarName142 >= 2.25(-> 2)) and (VarName31 >= 1(-> 0)) and (VarName111 <= 78.6(-> 79.2)) and (VarName71 <= 0(-> 1)) and (VarName97 <= 5(-> 6)) and (VarName26 >= 3(-> 2)) and (VarName70 <= 3(-> 4)) => Output=1 (CF = 0.96)
(VarName142 >= 2.25(-> 2)) and (VarName59 <= 0.25(-> 1)) and (health1 >= 2.75(-> 2)) and (VarName119 <= 68.8(-> 75)) and (VarName91 >= 0.25(-> 0)) and (VarName123 <= 3(-> 4)) and (VarName54 <= 0(-> 1)) => Output=1 (CF = 0.96)
(VarName142 >= 2.25(-> 2)) and (VarName114 <= 63.9(-> 66.7)) and (VarName117 <= 25.5(-> 27)) and (VarName92 >= 2.667(-> 2)) and (VarName93 <= 3(-> 4)) => Output=1 (CF = 0.95)
(VarName142 >= 2.25(-> 2)) and (VarName31 >= 1(-> 0)) and (VarName72 >= 2(-> 1)) and (VarName24 >= 4(-> 3)) => Output=1 (CF = 0.95)
(VarName142 >= 2.25(-> 2)) and (VarName124 <= 2(-> 2.75)) and (VarName63 >= 0.5(-> 0)) and (VarName54 <= 0(-> 1)) and (VarName63 <= 1(-> 2)) => Output=1 (CF = 0.97)
(VarName142 >= 2.25(-> 2)) and (VarName24 <= 4(-> 4.5)) and (VarName31 >= 1(-> 0)) and (VarName47 <= 1(-> 2)) and (VarName21 <= 2(-> 3)) => Output=1 (CF = 0.92)
(VarName142 >= 2.25(-> 2)) and (VarName24 <= 3.5(-> 4)) and (VarName41 >= 2(-> 1)) and (VarName86 <= 1.25(-> 2)) and (VarName49 >= 1(-> 0.25)) => Output=1 (CF = 0.96)
(VarName142 >= 3(-> 2)) and (VarName45 <= 0.5(-> 1)) and (VarName69 >= 1(-> 0.75)) and (VarName73 <= 1(-> 2)) => Output=1 (CF = 0.89)
(VarName142 >= 3(-> 2)) and (VarName75 <= 0(-> 1)) and (VarName66 >= 2(-> 1.75)) and (VarName59 >= 1(-> 0)) and (VarName110 <= 3(-> 4)) and (health1 >= 2(-> 1)) => Output=1 (CF = 0.96)

```

Σχήμα 52

```

(VarName142 <= 2(-> 2.25)) => Output=2 (CF = 0.99)
(VarName111 >= 85.7(-> 82.1)) and (VarName69 <= 0.75(-> 1)) and (health >= 3(-> 2)) and (pain7 >= 1(-> 0)) and (VarName78 >= 2(-> 1.75)) and (VarName29 <= 1(-> 2)) => Output=2 (CF = 0.97)
(VarName119 >= 75(-> 68.8)) and (VarName94 >= 1(-> 0)) and (VarName123 >= 1(-> 0)) and (VarName109 >= 80.6(-> 75)) => Output=2 (CF = 0.99)
(VarName142 <= 4.75(-> 5)) and (VarName104 >= 1(-> 0)) and (VarName97 <= 3(-> 5)) and (VarName100 <= 0(-> 1)) and (VarName70 <= 0(-> 1)) => Output=2 (CF = 0.99)

Number of Rules : 27

REGLAS = 27

```

Σχήμα 53



Στο 1 fold δημιουργούνται 36 ασαφείς κανόνες.

```
FURIA rules:
=====
(VarName142 >= 2.75(-> 2)) => Output=1 (CF = 0.67)
(VarName142 >= 2.75(-> 1)) => Output=1 (CF = 0.56)
(VarName142 >= 2.75(-> 2)) => Output=1 (CF = 0.67)
(VarName142 >= 2.75(-> 2)) => Output=1 (CF = 0.67)
(VarName142 >= 2.75(-> 2)) => Output=1 (CF = 0.67)
(VarName142 >= 2.75(-> 2)) => Output=1 (CF = 0.67)
(VarName142 >= 2.75(-> 2)) => Output=1 (CF = 0.67)
(VarName142 >= 2.25(-> 2)) and (health <= 2.75(-> 3)) => Output=1 (CF = 0.69)
(VarName142 >= 2.25(-> 2)) => Output=1 (CF = 0.67)
(VarName142 >= 2.25(-> 2)) => Output=1 (CF = 0.67)
(VarName142 >= 2.25(-> 2)) => Output=1 (CF = 0.67)
(VarName142 >= 2.25(-> 2)) => Output=1 (CF = 0.67)
(VarName142 >= 2.75(-> 2)) => Output=1 (CF = 0.67)
(VarName142 >= 2.75(-> 2.25)) and (VarName83 <= 0.75(-> 1)) => Output=1 (CF = 0.66)
(VarName142 >= 2.75(-> 2.5)) and (VarName75 <= 0.75(-> 1)) and (VarName120 <= 0(-> 1)) => Output=1 (CF = 0.71)
(VarName142 >= 2.75(-> 2)) and (health <= 2.25(-> 3)) and (VarName89 <= 0.5(-> 1)) => Output=1 (CF = 0.67)
(VarName142 >= 2.75(-> 2.25)) and (VarName124 <= 2(-> 3)) => Output=1 (CF = 0.64)
(VarName142 >= 2.75(-> 2)) => Output=1 (CF = 0.67)
(VarName142 >= 2.75(-> 2)) and (VarName67 <= 0(-> 1)) => Output=1 (CF = 0.64)
=> Output=1 (CF = 0.0)
(VarName142 >= 2.75(-> 2)) => Output=1 (CF = 0.67)
(VarName142 >= 2.75(-> 2)) and (VarName107 >= 55.52(-> 55.27)) and (VarName115 <= 1(-> 2)) and (VarName107 <= 58.15(-> 58.3)) and (VarName145 <= 1(-> 2)) and (VarName89 <= 0(-> 0.333)) => Output=1 (CF = 0.86)
(VarName142 >= 2.75(-> 0)) => Output=1 (CF = 0.51)
(VarName142 >= 2.75(-> 2)) and (VarName123 <= 1(-> 2)) => Output=1 (CF = 0.62)
(VarName142 >= 2.75(-> 2)) and (VarName107 >= 52.82(-> 51.9)) => Output=1 (CF = 0.68)
(VarName142 >= 3(-> 2)) and (VarName107 >= 58.73(-> 55.19)) and (pain8 >= 1(-> 0)) => Output=1 (CF = 0.76)
=> Output=2 (CF = 0.0)
(VarName136 <= 13(-> 14.6)) and (VarName116 >= 92.9(-> 89.3)) and (VarName125 <= 1(-> 2)) => Output=2 (CF = 0.93)
(VarName109 >= 86.1(-> 84.4)) and (VarName86 <= 0.25(-> 0.333)) and (VarName119 >= 68.8(-> 62.5)) => Output=2 (CF = 0.9)
(VarName56 <= 0(-> 0.75)) and (VarName114 >= 81.3(-> 80.6)) and (health2 >= 3(-> 2)) => Output=2 (CF = 0.88)
(VarName62 <= 0(-> 1)) and (VarName40 <= 0(-> 1)) => Output=2 (CF = 0.79)
(VarName66 <= 1(-> 2)) => Output=2 (CF = 0.79)
=> Output=2 (CF = 0.0)
(VarName30 <= 0(-> 0.25)) => Output=2 (CF = 0.76)
(VarName93 <= 1(-> 2)) => Output=2 (CF = 0.91)
(VarName119 >= 50(-> 43.8)) and (VarName136 <= 13(-> 14)) => Output=2 (CF = 0.82)
```

Σχήμα 54

```
Reglas Buenas
Inside m_Ruleset
(VarName142 >= 2.75(-> 2)) and (VarName136 >= 13.3(-> 9)) and (VarName73 >= 2(-> 1)) and (VarName126 <= 8(-> 9)) and (
VarName126 >= 5(-> 4)) and (VarName107 <= 63.56(-> 63.66)) => Output=1 (CF = 0.98)
(VarName142 >= 2.75(-> 1)) and (VarName109 <= 82.1(-> 86.1)) and (VarName107 >= 53.16(-> 50.57)) and (VarName91 <= 1.667(->
1.75)) and (VarName145 >= 10(-> 7)) and (VarName91 >= 0.333(-> 0)) => Output=1 (CF = 0.97)
(VarName142 >= 2.75(-> 2)) and (VarName109 <= 82.1(-> 83.3)) and (VarName97 <= 2(-> 3)) and (VarName30 >= 2(-> 1.75)) and (
VarName125 >= 6(-> 5)) => Output=1 (CF = 0.96)
(VarName142 >= 2.75(-> 2)) and (VarName86 >= 0.25(-> 0)) and (VarName107 >= 53.05(-> 49.89)) and (health3 <= 3(-> 4)) and (
VarName36 >= 1(-> 0)) and (VarName106 <= 44.02(-> 46.83)) => Output=1 (CF = 0.98)
(VarName142 >= 2.75(-> 2)) and (VarName32 >= 0.5(-> 0)) and (VarName93 >= 4(-> 3)) and (VarName112 >= 28.3(-> 28)) and (
VarName41 <= 2(-> 3)) => Output=1 (CF = 0.96)
(VarName142 >= 2.75(-> 2)) and (VarName66 >= 2(-> 1.75)) and (VarName109 <= 84.4(-> 86.1)) and (VarName26 <= 1.5(-> 2)) and (
VarName65 >= 1.5(-> 1)) and (pain7 >= 1(-> 0)) => Output=1 (CF = 0.95)
(VarName142 >= 2.75(-> 2)) and (VarName66 >= 2(-> 1.75)) and (pain7 <= 0(-> 1)) and (VarName31 >= 1(-> 0)) and (VarName107
<= 57.16(-> 58.72)) and (VarName33 <= 3(-> 4)) => Output=1 (CF = 0.97)
(VarName142 >= 2.25(-> 2)) and (health <= 2.75(-> 3)) and (VarName86 >= 0.5(-> 0.25)) and (VarName117 <= 21(-> 22)) and (
VarName92 >= 1.5(-> 1.333)) and (VarName116 >= 75(-> 71.4)) => Output=1 (CF = 0.97)
(VarName142 >= 2.25(-> 2)) and (VarName69 >= 0.5(-> 0.25)) and (VarName117 <= 21.3(-> 22)) and (VarName111 >= 83.3(->
82.1)) and (VarName7 <= 0(-> 1)) and (VarName30 <= 1(-> 1.333)) => Output=1 (CF = 0.96)
(VarName142 >= 2.25(-> 2)) and (VarName56 >= 0.25(-> 0)) and (VarName124 <= 2(-> 3)) and (VarName112 <= 19(-> 19.1)) and (
VarName136 >= 13.4(-> 13)) and (pain4 >= 1(-> 0)) => Output=1 (CF = 0.96)
(VarName142 >= 2.25(-> 2)) and (VarName68 >= 2(-> 1)) and (VarName91 <= 1.667(-> 1.75)) and (VarName126 <= 7(-> 9)) and (
VarName94 >= 2(-> 1)) => Output=1 (CF = 0.96)
(VarName142 >= 2.25(-> 2)) and (VarName48 >= 2(-> 1)) and (VarName53 <= 1(-> 1.5)) and (VarName110 >= 4(-> 3)) and (
VarName63 <= 2(-> 3)) => Output=1 (CF = 0.97)
(VarName142 >= 2.75(-> 2)) and (VarName124 >= 7(-> 6)) and (VarName91 >= 4(-> 3)) and (VarName68 <= 3(-> 4)) => Output=1 (
CF = 0.96)
(VarName142 >= 2.75(-> 2.25)) and (VarName83 <= 0.75(-> 1)) and (VarName86 >= 0.25(-> 0)) and (VarName120 <= 0.25(-> 1))
and (VarName96 >= 1(-> 0)) => Output=1 (CF = 0.96)
(VarName142 >= 2.75(-> 2.5)) and (VarName75 <= 0.75(-> 1)) and (VarName120 <= 0(-> 1)) and (VarName100 >= 1(-> 0.75)) =>
Output=1 (CF = 0.9)
```

Σχήμα 55

```

(VarName142 >= 2.75(-> 2.5)) and (VarName75 <= 0.75(-> 1)) and (VarName120 <= 0(-> 1)) and (VarName100 >= 1(-> 0.75)) => Output=1 (CF = 0.9)
(VarName142 >= 2.75(-> 2)) and (health <= 2.25(-> 3)) and (VarName89 <= 0.5(-> 1)) and (VarName124 >= 3(-> 2)) and (health2 <= 2.75(-> 3)) and (
VarName91 <= 3(-> 4)) => Output=1 (CF = 0.95)
(VarName142 >= 2.75(-> 2.25)) and (VarName124 <= 2(-> 3)) and (VarName56 >= 0.25(-> 0)) and (VarName42 <= 0(-> 1)) => Output=1 (CF = 0.94)
(VarName142 >= 2.75(-> 2)) and (VarName87 >= 0.75(-> 0)) and (VarName72 <= 0(-> 1)) and (VarName64 >= 2(-> 1)) and (VarName106 >= 44.11(-> 43.34))
=> Output=1 (CF = 0.94)
(VarName142 >= 2.75(-> 2)) and (VarName67 <= 0(-> 1)) and (pain7 <= 0(-> 1)) and (VarName50 >= 1(-> 0)) and (VarName91 <= 2(-> 2.333)) => Output=1 (
CF = 0.93)
(VarName142 >= 5(-> 4)) and (VarName94 <= 0(-> 1)) and (VarName117 >= 10.6(-> 10)) and (VarName41 >= 1(-> 0)) => Output=1 (CF = 0.95)
(VarName142 >= 2.75(-> 2)) and (VarName49 >= 2(-> 1)) and (VarName30 <= 1(-> 2)) and (VarName125 <= 5(-> 6)) and (VarName33 <= 2(-> 3)) => Output=1 (
CF = 0.94)
(VarName142 >= 2.75(-> 2)) and (VarName107 >= 55.52(-> 55.27)) and (VarName115 <= 1(-> 2)) and (VarName107 <= 58.15(-> 58.3)) and (VarName145 <=
1(-> 2)) and (VarName89 <= 0(-> 0.333)) and (VarName109 <= 97.2(-> 100)) => Output=1 (CF = 0.94)
(VarName142 >= 2.75(-> 0)) and (VarName97 >= 5(-> 4)) and (VarName131 <= 12(-> 13)) and (VarName116 <= 96.4(-> 100)) and (VarName126 <= 5(-> 6)) =>
Output=1 (CF = 0.95)
(VarName142 >= 2.75(-> 2)) and (VarName123 <= 1(-> 2)) and (VarName107 <= 48.76(-> 49.04)) and (VarName60 <= 1(-> 2)) and (VarName106 <= 57.08(->
58.26)) and (VarName126 <= 7(-> 8)) => Output=1 (CF = 0.93)
(VarName142 >= 2.75(-> 2)) and (VarName107 >= 52.82(-> 51.9)) and (VarName117 >= 15(-> 14)) and (VarName84 <= 1(-> 2)) and (VarName42 <= 1.25(-> 2))
and (VarName47 >= 1(-> 0)) => Output=1 (CF = 0.97)
(VarName142 >= 3(-> 2)) and (VarName107 >= 58.73(-> 55.19)) and (pain8 >= 1(-> 0)) and (VarName78 >= 2(-> 1)) => Output=1 (CF = 0.89)
(VarName142 <= 2.5(-> 2.75)) => Output=2 (CF = 0.99)
(VarName136 <= 13(-> 14.6)) and (VarName116 >= 92.9(-> 89.3)) and (VarName125 <= 1(-> 2)) and (VarName26 >= 1(-> 0)) => Output=2 (CF = 0.99)
(VarName199 >= 86.1(-> 84.4)) and (VarName86 <= 0.25(-> 0.333)) and (VarName119 >= 68.8(-> 62.5)) and (VarName112 >= 4(-> 3)) => Output=2 (CF = 0.94)
(VarName56 <= 0(-> 0.75)) and (VarName114 >= 81.3(-> 80.6)) and (health2 >= 3(-> 2)) and (VarName23 <= 4(-> 4.25)) and (VarName23 >= 4(-> 3)) and (
VarName40 <= 3(-> 4)) => Output=2 (CF = 0.99)
(VarName62 <= 0(-> 1)) and (VarName40 <= 0(-> 1)) and (VarName113 >= 2(-> 1)) and (VarName107 >= 54.97(-> 54.92)) and (VarName53 <= 0(-> 0.25)) and (
VarName33 >= 1(-> 0)) => Output=2 (CF = 0.93)
(VarName66 <= 1(-> 2)) and (VarName136 >= 25.1(-> 25)) and (VarName111 >= 64.3(-> 60.7)) and (VarName106 <= 40.77(-> 42.24)) => Output=2 (CF = 0.83)
(VarName85 >= 1(-> 0)) and (VarName115 <= 3(-> 4)) and (VarName121 <= 0(-> 1)) and (VarName89 >= 1.667(-> 1)) => Output=2 (CF = 0.91)
(VarName30 <= 0(-> 0.25)) and (VarName145 >= 2(-> 0)) and (VarName98 <= 0(-> 1)) and (health3 <= 3(-> 4)) and (VarName119 <= 37.5(-> 43.8)) =>
Output=2 (CF = 0.95)
(VarName93 <= 1(-> 2)) and (VarName38 >= 1(-> 0)) => Output=2 (CF = 0.96)
(VarName119 >= 50(-> 43.8)) and (VarName136 <= 13(-> 14)) and (VarName145 >= 2(-> 1)) and (VarName23 >= 5(-> 4)) and (VarName81 <= 0.25(-> 1)) and (
VarName145 <= 21(-> 30)) => Output=2 (CF = 0.95)
Number of Rules : 36

REGLAS = 36

```

Σχήμα 56

Στο 2 fold έχουμε 27 κανόνες.

```

FURIA rules:
=====
(VarName142 >= 2.75(-> 2)) => Output=1 (CF = 0.67)
(VarName142 >= 2.75(-> 2)) => Output=1 (CF = 0.67)
(VarName142 >= 2.75(-> 2)) => Output=1 (CF = 0.67)
(VarName142 >= 2.75(-> 2)) => Output=1 (CF = 0.67)
(VarName142 >= 2.75(-> 2)) => Output=1 (CF = 0.67)
(VarName142 >= 2.25(-> 2)) => Output=1 (CF = 0.67)
(VarName142 >= 2.25(-> 2)) and (VarName124 <= 2(-> 3)) => Output=1 (CF = 0.67)
(VarName142 >= 2.25(-> 2)) and (pain7 <= 0(-> 1)) => Output=1 (CF = 0.71)
(VarName142 >= 2.25(-> 2)) => Output=1 (CF = 0.67)
(VarName142 >= 2.25(-> 2)) => Output=1 (CF = 0.67)
(VarName142 >= 2.25(-> 2)) and (VarName75 <= 0(-> 1)) => Output=1 (CF = 0.67)
(VarName142 >= 2.25(-> 2)) and (VarName21 <= 1(-> 2)) => Output=1 (CF = 0.76)
(VarName142 >= 2.25(-> 2)) => Output=1 (CF = 0.67)
(VarName142 >= 2.25(-> 2)) => Output=1 (CF = 0.67)
(VarName142 >= 2.25(-> 2)) => Output=1 (CF = 0.67)
(VarName142 >= 2.25(-> 2)) and (VarName83 <= 0(-> 3)) => Output=1 (CF = 0.67)
(VarName142 >= 2.25(-> 2)) and (VarName44 <= 0(-> 1)) => Output=1 (CF = 0.63)
(VarName142 >= 2.25(-> 1)) => Output=1 (CF = 0.54)
=> Output=2 (CF = 0.0)
(VarName111 >= 78.6(-> 75)) and (VarName86 <= 0(-> 0.25)) and (VarName107 >= 54.91(-> 54.77)) => Output=2 (CF = 0.84)
=> Output=2 (CF = 0.0)
(VarName111 >= 85.7(-> 83.3)) => Output=2 (CF = 0.76)
(VarName126 <= 3(-> 4)) => Output=2 (CF = 0.75)
(VarName43 <= 1(-> 2)) and (VarName69 <= 0.75(-> 1)) => Output=2 (CF = 0.76)
(VarName124 <= 0(-> 1)) => Output=2 (CF = 0.79)
=> Output=2 (CF = 0.0)
=> Output=2 (CF = 0.0)

```

Σχήμα 57

```

Reglas Buenas
Inside m_ruleset
(VarName142 >= 2.75(-> 2)) and (VarName111 <= 75(-> 82.1)) and (VarName109 <= 65.6(-> 66.7)) and (VarName53 <= 1(-> 1.333)) and (VarName21 <= 2(-> 2.5)) and (VarName96 >= 2(-> 1)) and (VarName91 >= 0.5(-> 0)) => Output=1 (CF = 0.98)
(VarName142 >= 2.75(-> 2)) and (VarName43 >= 2(-> 1)) and (VarName44 >= 2(-> 1)) and (VarName43 <= 2(-> 3)) and (health >= 3(-> 2)) and (health <= 4(-> 5)) => Output=1 (CF = 0.98)
(VarName142 >= 2.75(-> 2)) and (VarName86 >= 0.25(-> 0)) and (VarName107 >= 53.835(-> 52.42)) and (VarName47 >= 1(-> 2)) and (VarName83 <= 1(-> 2)) and (VarName97 >= 2(-> 1)) and (VarName107 <= 60.2(-> 60.29)) => Output=1 (CF = 0.97)
(VarName142 >= 2.75(-> 2)) and (VarName111 <= 78.6(-> 82.1)) and (VarName23 <= 4(-> 4.5)) and (VarName66 >= 3(-> 2)) and (VarName37 <= 2(-> 3)) and (VarName140 <= 0(-> 2)) => Output=1 (CF = 0.98)
(VarName142 >= 2.75(-> 2)) and (VarName40 >= 0.75(-> 0)) and (VarName112 <= 19(-> 19.1)) and (VarName43 >= 0.75(-> 0)) and (VarName93 >= 3(-> 2)) => Output=1 (CF = 0.98)
(VarName142 >= 2.25(-> 2)) and (VarName37 >= 3(-> 2)) and (health4 <= 2(-> 3)) and (VarName36 >= 1(-> 0)) => Output=1 (CF = 0.96)
(VarName142 >= 2.25(-> 2)) and (VarName124 <= 2(-> 3)) and (VarName63 >= 0.25(-> 0)) and (VarName26 <= 1(-> 2)) => Output=1 (CF = 0.93)
(VarName142 >= 2.25(-> 2)) and (pain7 <= 0(-> 1)) and (VarName136 >= 12(-> 11)) and (VarName42 <= 1(-> 2)) and (VarName107 <= 50.51(-> 51.53)) => Output=1 (CF = 0.95)
(VarName142 >= 2.25(-> 2)) and (VarName126 >= 5(-> 3)) and (VarName21 <= 2(-> 2.5)) and (VarName31 >= 1(-> 0)) and (VarName38 <= 1(-> 2)) and (VarName55 >= 1(-> 0)) and (VarName95 <= 3(-> 4)) => Output=1 (CF = 0.98)
(VarName142 >= 2.25(-> 2)) and (VarName45 <= 0.667(-> 1)) and (VarName86 >= 0.5(-> 0.333)) and (VarName67 <= 1(-> 2)) and (VarName50 <= 0(-> 1)) and (VarName24 >= 3.5(-> 3)) => Output=1 (CF = 0.97)
(VarName142 >= 2.25(-> 2)) and (VarName75 <= 0(-> 1)) and (VarName126 >= 3(-> 2)) and (VarName7 <= 0(-> 1)) and (VarName144 <= 3(-> 3.25)) and (VarName107 <= 60.29(-> 60.3)) => Output=1 (CF = 0.97)
(VarName142 >= 2.25(-> 2)) and (VarName21 <= 1(-> 2)) and (VarName34 >= 1(-> 0)) and (VarName94 <= 1(-> 2)) => Output=1 (CF = 0.93)
(VarName142 >= 2.25(-> 2)) and (VarName107 <= 42.98(-> 43.01)) and (VarName107 >= 41.19(-> 40.9)) and (VarName73 <= 1(-> 2)) => Output=1 (CF = 0.94)
(VarName142 >= 2.25(-> 2)) and (VarName71 >= 2(-> 1)) and (VarName48 <= 0.25(-> 1)) and (VarName33 >= 2(-> 1)) => Output=1 (CF = 0.87)
(VarName142 >= 2.25(-> 2)) and (VarName49 >= 2(-> 1)) and (VarName116 <= 64.3(-> 67.9)) and (VarName124 <= 7(-> 8)) => Output=1 (CF = 0.96)
(VarName142 >= 2.25(-> 2)) and (VarName83 <= 0(-> 3)) and (VarName69 >= 0.5(-> 0)) and (VarName35 <= 0(-> 1)) and (VarName119 >= 62.5(-> 56.3)) => Output=1 (CF = 0.94)
(VarName142 >= 2.25(-> 2)) and (VarName44 <= 0(-> 1)) and (VarName56 >= 0.5(-> 0)) and (pain3 <= 0.75(-> 1)) => Output=1 (CF = 0.87)
(VarName142 >= 2.25(-> 1)) and (VarName97 >= 7(-> 6)) and (VarName136 >= 38(-> 37.4)) => Output=1 (CF = 0.87)
(VarName142 <= 2.5(-> 2.75)) => Output=2 (CF = 0.99)
(VarName111 >= 78.6(-> 75)) and (VarName86 <= 0(-> 0.25)) and (VarName107 >= 54.91(-> 54.77)) and (VarName107 <= 57.21(-> 57.73)) and (VarName94 >= 1(-> 0)) and (VarName96 <= 1(-> 2)) => Output=2 (CF = 0.99)
(VarName43 <= 0(-> 1)) and (VarName136 >= 5(-> 4.2)) and (VarName74 >= 1(-> 0)) and (VarName109 >= 86.1(-> 83.3)) => Output=2 (CF = 0.9)
(VarName111 >= 85.7(-> 83.3)) and (VarName104 >= 1(-> 0.75)) and (VarName7 >= 1(-> 0)) => Output=2 (CF = 0.76)
(VarName126 <= 3(-> 4)) and (VarName124 >= 4(-> 3)) and (VarName107 >= 44.7(-> 42.97)) and (VarName69 <= 0(-> 1)) => Output=2 (CF = 0.88)
(VarName43 <= 1(-> 2)) and (VarName69 <= 0.75(-> 1)) and (VarName59 >= 2(-> 1)) and (VarName124 <= 5(-> 6)) and (health1 >= 3(-> 2.5)) => Output=2 (CF = 0.91)
(VarName124 <= 0(-> 1)) and (VarName107 <= 54.2(-> 54.43)) and (VarName24 >= 4.75(-> 4)) => Output=2 (CF = 0.88)
(VarName107 <= 53.61(-> 53.708)) and (VarName105 >= 16(-> 15)) and (VarName107 >= 46.76(-> 46.21)) => Output=2 (CF = 0.81)
(VarName23 <= 3.5(-> 4)) and (VarName91 >= 2(-> 1.75)) and (VarName126 >= 7(-> 6)) and (VarName52 <= 0(-> 1)) => Output=2 (CF = 0.88)

Number of Rules : 27
REGLAS = 27

```

Σχήμα 58

To 3 fold έχει 26 ασαφείς κανόνες.

```

FURIA rules:
=====
(VarName142 >= 2.75(-> 2)) => Output=1 (CF = 0.67)
(VarName142 >= 3(-> 2)) => Output=1 (CF = 0.67)
(VarName142 >= 2.75(-> 2)) => Output=1 (CF = 0.67)
(VarName142 >= 2.75(-> 2)) => Output=1 (CF = 0.67)
(VarName142 >= 2.75(-> 2)) => Output=1 (CF = 0.67)
=> Output=1 (CF = 0.0)
(VarName142 >= 2.75(-> 2)) and (VarName59 <= 1(-> 2)) => Output=1 (CF = 0.66)
(VarName142 >= 2.75(-> 2)) and (VarName21 <= 2.25(-> 3)) => Output=1 (CF = 0.69)
(VarName142 >= 2.75(-> 2)) => Output=1 (CF = 0.67)
(VarName142 >= 2.75(-> 2)) and (VarName120 <= 0.25(-> 1)) and (VarName124 <= 2(-> 3)) => Output=1 (CF = 0.78)
(VarName142 >= 2.75(-> 2)) => Output=1 (CF = 0.67)
(VarName142 >= 3(-> 2)) => Output=1 (CF = 0.67)
(VarName142 >= 3(-> 2)) and (VarName26 <= 1(-> 2)) => Output=1 (CF = 0.64)
(VarName142 >= 3(-> 2)) and (VarName106 >= 49.11(-> 49.04)) => Output=1 (CF = 0.64)
(VarName142 >= 3(-> 2)) and (health <= 2.75(-> 3)) and (VarName111 >= 87.5(-> 85.7)) => Output=1 (CF = 0.66)
(VarName142 >= 3(-> 2)) and (VarName107 >= 53.58(-> 53.16)) and (VarName117 <= 1(-> 2)) and (VarName107 <= 58.05(-> 58.72)) => Output=1 (CF = 0.71)
(VarName142 >= 3(-> 2)) => Output=1 (CF = 0.67)
(VarName142 >= 3(-> 2)) => Output=1 (CF = 0.67)
(VarName142 >= 3(-> 2)) and (health <= 2.75(-> 3)) => Output=1 (CF = 0.69)
(VarName142 >= 3(-> 2)) and (VarName124 <= 1(-> 2)) => Output=1 (CF = 0.63)
(VarName142 >= 3(-> 2)) => Output=1 (CF = 0.67)
=> Output=2 (CF = 0.0)
(VarName43 <= 1(-> 2)) and (VarName119 >= 75(-> 68.8)) and (VarName94 >= 1(-> 0)) => Output=2 (CF = 0.95)
(VarName43 <= 0(-> 1)) => Output=2 (CF = 0.77)
(VarName125 <= 1(-> 2)) => Output=2 (CF = 0.81)
(VarName43 <= 1(-> 2)) and (VarName116 >= 82.1(-> 78.6)) => Output=2 (CF = 0.78)

```

Σχήμα 59



```

Reglas Buenas
Inside m Ruleset
(VarName142 >= 2.75(-> 2)) and (VarName43 >= 2(-> 1)) and (VarName73 >= 2(-> 1)) and (VarName19 >= 4(-> 3)) => Output=1 (CF = 0.97)
(VarName142 >= 3(-> 2)) and (VarName43 >= 2(-> 1)) and (VarName43 <= 2(-> 3)) and (VarName36 >= 0.25(-> 0)) and (VarName94 <= 3(-> 4)) => Output=1 (CF = 0.9)
(VarName142 >= 2.75(-> 2)) and (VarName43 >= 2(-> 1)) and (VarName91 <= 1.75(-> 2)) and (VarName74 >= 2(-> 1)) => Output=1 (CF = 0.95)
(VarName142 >= 2.75(-> 2)) and (VarName119 <= 62.5(-> 68.8)) and (VarName83 <= 0(-> 0.25)) and (VarName89 >= 0.75(-> 0.5)) and (VarName74 <= 1(-> 2)) and (health4 <= 4(-> 5)) => Output=1 (CF = 0.97)
(VarName142 >= 2.75(-> 2)) and (VarName69 >= 2(-> 1)) and (VarName117 <= 21.3(-> 22.7)) and (VarName65 >= 0.25(-> 0)) and (VarName23 >= 4(-> 3)) => Output=1 (CF = 0.97)
(VarName142 >= 5(-> 4)) and (VarName91 >= 4(-> 3)) => Output=1 (CF = 0.95)
(VarName142 >= 2.75(-> 2)) and (VarName59 <= 1(-> 2)) and (VarName119 <= 68.8(-> 75)) and (VarName54 <= 0(-> 1)) and (VarName126 >= 3(-> 2)) and (VarName66 <= 1.25(-> 2)) => Output=1 (CF = 0.95)
(VarName142 >= 2.75(-> 2)) and (VarName21 <= 2.25(-> 3)) and (VarName112 >= 10.6(-> 9)) and (VarName125 <= 4(-> 5)) and (VarName95 >= 2(-> 1)) and (VarName89 <= 1(-> 2)) => Output=1 (CF = 0.96)
(VarName142 >= 2.75(-> 2)) and (VarName66 >= 2(-> 1)) and (VarName89 <= 0.333(-> 0.5)) and (VarName109 <= 86.1(-> 87.5)) and (VarName81 <= 0(-> 1)) and (VarName111 <= 92.9(-> 96.4)) => Output=1 (CF = 0.97)
(VarName142 >= 2.75(-> 2)) and (VarName120 <= 0.25(-> 1)) and (VarName124 <= 2(-> 3)) and (VarName63 >= 0.25(-> 0)) => Output=1 (CF = 0.95)
(VarName142 >= 2.75(-> 2)) and (VarName114 <= 56.3(-> 58.3)) and (VarName107 >= 53.31(-> 53.032)) and (VarName111 <= 87.5(-> 89.3)) and (VarName81 >= 2(-> 1)) => Output=1 (CF = 0.96)
(VarName142 >= 3(-> 2)) and (VarName145 >= 8(-> 5)) and (VarName84 <= 0.5(-> 1)) and (VarName86 >= 0.5(-> 0)) and (health2 <= 2.75(-> 3)) => Output=1 (CF = 0.97)
(VarName142 >= 3(-> 2)) and (VarName26 <= 1(-> 2)) and (VarName125 >= 2(-> 0)) and (VarName38 <= 0(-> 1)) and (VarName109 <= 81.3(-> 83.3)) and (VarName106 <= 42.21(-> 42.25)) => Output=1 (CF = 0.95)
(VarName142 >= 3(-> 2)) and (VarName106 >= 49.11(-> 49.04)) and (VarName24 <= 4(-> 4.5)) and (VarName136 >= 12(-> 11.5)) => Output=1 (CF = 0.88)
(VarName142 >= 3(-> 2)) and (health <= 2.75(-> 3)) and (VarName111 >= 87.5(-> 85.7)) and (VarName91 >= 0.667(-> 0.5)) and (VarName124 <= 3.5(-> 4)) and (VarName65 <= 0(-> 1)) => Output=1 (CF = 0.84)
(VarName142 >= 3(-> 2)) and (VarName107 >= 53.58(-> 53.16)) and (VarName117 <= 1(-> 2)) and (VarName107 <= 58.05(-> 58.72)) and (VarName98 >= 1(-> 0)) => Output=1 (CF = 0.88)
(VarName142 >= 3(-> 2)) and (VarName125 >= 6(-> 5)) and (VarName145 >= 10(-> 8.5)) and (VarName30 <= 0.5(-> 0.667)) and (VarName24 >= 3(-> 2)) => Output=1 (CF = 0.93)
(VarName142 >= 3(-> 2)) and (VarName71 >= 2(-> 1)) and (VarName75 <= 1(-> 2)) and (VarName32 <= 0(-> 1)) => Output=1 (CF = 0.84)
(VarName142 >= 3(-> 2)) and (health <= 2.75(-> 3)) and (VarName117 >= 16(-> 15.9)) and (VarName125 <= 3(-> 5)) => Output=1 (CF = 0.83)
(VarName142 >= 3(-> 2)) and (VarName124 <= 1(-> 2)) and (VarName23 <= 3(-> 3.5)) and (VarName27 <= 0(-> 1)) and (VarName106 <= 57.08(-> 59.63)) => Output=1 (CF = 0.86)
(VarName142 >= 3(-> 2)) and (VarName125 >= 6(-> 5.75)) and (VarName119 >= 43.8(-> 37.5)) and (VarName107 >= 55.81(-> 54.91)) and (VarName80 <= 0(-> 1)) => Output=1 (CF = 0.94)
(VarName142 <= 2.5(-> 2.75)) => Output=2 (CF = 0.99)
(VarName43 <= 1(-> 2)) and (VarName119 >= 75(-> 68.8)) and (VarName94 >= 1(-> 0)) and (VarName136 >= 2(-> 1)) => Output=2 (CF = 0.98)
(VarName43 <= 0(-> 1)) and (VarName138 <= 1(-> 2)) and (VarName100 <= 0.75(-> 1)) => Output=2 (CF = 0.98)
(VarName125 <= 1(-> 2)) and (VarName83 >= 2(-> 1)) and (health >= 2.25(-> 2)) and (VarName67 >= 3(-> 2)) => Output=2 (CF = 0.95)
(VarName43 <= 1(-> 2)) and (VarName116 >= 82.1(-> 78.6)) and (VarName75 >= 1(-> 0)) and (VarName124 >= 4(-> 3)) and (VarName94 <= 1(-> 2)) and (VarName106 <= 54.33(-> 54.41)) and (VarName143 >= 2(-> 1)) => Output=2 (CF = 0.98)
Number of Rules : 26

```

Σχήμα 60

Τέλος το 4 fold έχει 24 κανόνες.

```
FURIA rules:
=====
(VarName142 >= 2.25(-> 2)) and (VarName43 >= 2(-> 1)) => Output=1 (CF = 0.78)
(VarName142 >= 2.25(-> 2)) and (VarName111 <= 75(-> 82.1)) and (VarName109 <= 65.6(-> 66.7)) and (VarName85 <= 0.75(-> 1))
=> Output=1 (CF = 0.88)
(VarName142 >= 3(-> 2)) => Output=1 (CF = 0.67)
(VarName142 >= 3(-> 2)) and (VarName56 >= 0.333(-> 0)) => Output=1 (CF = 0.72)
(VarName142 >= 3(-> 2)) and (VarName56 >= 0.333(-> 0)) => Output=1 (CF = 0.72)
(VarName142 >= 3(-> 2)) => Output=1 (CF = 0.67)
(VarName142 >= 3(-> 2)) and (VarName43 >= 1(-> 0)) => Output=1 (CF = 0.73)
(VarName142 >= 3(-> 2)) and (VarName56 >= 0.333(-> 0)) => Output=1 (CF = 0.72)
(VarName142 >= 3(-> 2)) => Output=1 (CF = 0.67)
(VarName142 >= 3(-> 2.25)) and (VarName45 >= 1.25(-> 1)) => Output=1 (CF = 0.73)
(VarName142 >= 3(-> 2)) => Output=1 (CF = 0.67)
(VarName142 >= 3(-> 2)) and (VarName19 >= 4.75(-> 4)) and (VarName59 <= 1(-> 2)) => Output=1 (CF = 0.69)
(VarName142 >= 3(-> 2)) => Output=1 (CF = 0.67)
(VarName142 >= 3(-> 2)) and (VarName43 >= 1(-> 0)) => Output=1 (CF = 0.73)
(VarName142 >= 3(-> 2.25)) and (VarName55 >= 2(-> 1)) => Output=1 (CF = 0.77)
(VarName142 >= 5(-> 4)) => Output=1 (CF = 0.71)
(VarName142 >= 3(-> 2)) and (VarName65 <= 1(-> 2)) => Output=1 (CF = 0.65)
(VarName142 >= 3(-> 2)) => Output=1 (CF = 0.67)
(VarName142 >= 3(-> 2)) and (VarName136 >= 33(-> 31)) => Output=1 (CF = 0.76)
(VarName142 >= 3(-> 2)) => Output=1 (CF = 0.67)
=> Output=2 (CF = 0.0)
=> Output=2 (CF = 0.0)
=> Output=2 (CF = 0.0)
=> Output=2 (CF = 0.0)
```

Σχήμα 61

```
Reglas Buenas
Inside_m_Ruleset
(VarName142 >= 2.25(-> 2)) and (VarName43 >= 2(-> 1)) and (VarName91 <= 1.667(-> 1.75)) and (VarName91 >= 1(-> 0.75)) and (VarName42 >= 1(-> 0)) => Output=1 (CF
= 0.98)
(VarName142 >= 2.25(-> 2)) and (VarName111 <= 75(-> 82.1)) and (VarName109 <= 65.6(-> 66.7)) and (VarName85 <= 0.75(-> 1)) and (VarName53 <= 1(-> 1.5)) and (
VarName126 <= 8(-> 10)) => Output=1 (CF = 0.98)
(VarName142 >= 3(-> 2)) and (VarName86 >= 0.5(-> 0.333)) and (VarName21 <= 2.25(-> 3)) and (VarName45 <= 0.75(-> 1)) and (VarName125 >= 3(-> 2)) and (VarName103
<= 0(-> 1)) => Output=1 (CF = 0.96)
(VarName142 >= 3(-> 2)) and (VarName56 >= 0.333(-> 0)) and (VarName126 >= 4(-> 0)) and (VarName107 >= 53.31(-> 52.42)) and (VarName73 >= 2(-> 1)) and (VarName76
<= 2(-> 3)) => Output=1 (CF = 0.96)
(VarName142 >= 3(-> 2)) and (VarName56 >= 0.333(-> 0)) and (VarName126 >= 4(-> 3)) and (VarName91 >= 4(-> 3)) and (VarName34 >= 1(-> 0)) and (VarName85 <= 3(->
4)) => Output=1 (CF = 0.97)
(VarName142 >= 3(-> 2)) and (pain7 <= 0(-> 1)) and (health4 <= 4.5(-> 5)) and (VarName90 >= 0.5(-> 0)) and (VarName45 <= 1(-> 1.75)) and (VarName104 <= 0(-> 1))
=> Output=1 (CF = 0.97)
(VarName142 >= 3(-> 2)) and (VarName43 >= 1(-> 0)) and (VarName37 >= 1(-> 0)) and (pain7 <= 0(-> 1)) and (VarName99 <= 0(-> 1)) and (VarName109 <= 83.3(->
86.1)) => Output=1 (CF = 0.97)
(VarName142 >= 3(-> 2)) and (VarName56 >= 0.333(-> 0)) and (VarName108 <= 4(-> 5)) and (VarName117 <= 4(-> 4.3)) and (VarName31 >= 1(-> 0)) and (VarName45 >=
1(-> 0)) => Output=1 (CF = 0.94)
(VarName142 >= 3(-> 2)) and (VarName69 >= 2(-> 1)) and (VarName91 <= 1.333(-> 1.5)) and (VarName87 <= 1(-> 2)) and (VarName36 <= 3(-> 4)) => Output=1 (CF = 0.97)
(VarName142 >= 3(-> 2.25)) and (VarName45 >= 1.25(-> 1)) and (VarName145 >= 14(-> 7)) and (VarName71 <= 0(-> 1)) and (VarName97 >= 5(-> 4)) => Output=1 (CF =
0.96)
(VarName142 >= 3(-> 2)) and (VarName66 >= 2.25(-> 2)) and (VarName75 <= 1(-> 2)) and (VarName24 <= 4.75(-> 5)) and (pain5 >= 1(-> 0)) => Output=1 (CF = 0.94)
(VarName142 >= 3(-> 2)) and (VarName19 >= 4.75(-> 4)) and (VarName59 <= 1(-> 2)) and (VarName63 >= 0.25(-> 0)) and (health <= 2.75(-> 3)) and (VarName76 <= 0(->
1)) => Output=1 (CF = 0.94)
(VarName142 >= 3(-> 2)) and (VarName71 >= 2(-> 1)) and (VarName106 >= 39.67(-> 37.78)) and (VarName91 >= 2.5(-> 2)) and (VarName21 <= 3(-> 4)) and (VarName65 >=
2(-> 1)) => Output=1 (CF = 0.95)
(VarName142 >= 3(-> 2)) and (VarName43 >= 1(-> 0)) and (VarName107 >= 55.44(-> 55.36)) and (VarName123 <= 2(-> 3)) and (VarName116 >= 82.1(-> 78.6)) and (
VarName96 >= 1(-> 0)) => Output=1 (CF = 0.95)
(VarName142 >= 3(-> 2.25)) and (VarName55 >= 2(-> 1)) and (VarName108 >= 13(-> 12)) => Output=1 (CF = 0.96)
(VarName142 >= 5(-> 4)) and (VarName110 <= 2(-> 3)) and (VarName43 >= 1(-> 0)) and (VarName65 >= 1.5(-> 1)) and (VarName93 >= 2(-> 1)) => Output=1 (CF = 0.97)
(VarName142 >= 3(-> 2)) and (VarName65 <= 1(-> 2)) and (VarName126 >= 3(-> 2)) and (VarName107 >= 55.44(-> 55.19)) and (VarName125 >= 6(-> 5)) and (health3 >=
4(-> 3)) and (VarName18 >= 5(-> 4)) => Output=1 (CF = 0.96)
(VarName142 >= 3(-> 2)) and (VarName136 <= 4.1(-> 5)) and (VarName113 <= 1(-> 2)) and (VarName124 >= 1(-> 0)) and (VarName106 <= 57.76(-> 58.28)) => Output=1 (
CF = 0.95)
(VarName142 >= 3(-> 2)) and (VarName136 >= 33(-> 31)) and (VarName42 <= 1(-> 2)) and (VarName108 <= 8(-> 9)) and (health1 >= 2(-> 1)) => Output=1 (CF = 0.95)
(VarName142 >= 3(-> 2)) and (VarName45 <= 0.25(-> 1)) and (VarName24 <= 4(-> 5)) and (VarName81 <= 0(-> 0.25)) and (VarName119 <= 68.8(-> 75)) and (VarName42 <=
1(-> 2)) => Output=1 (CF = 0.95)
(VarName142 <= 2(-> 2.25)) => Output=2 (CF = 0.99)
(VarName43 <= 1(-> 2)) and (VarName119 >= 75(-> 68.8)) and (pain6 >= 1(-> 0)) and (VarName107 <= 51.33(-> 51.94)) => Output=2 (CF = 0.98)
(VarName111 >= 83.3(-> 82.1)) and (VarName69 <= 0(-> 1)) and (VarName124 >= 4(-> 3)) and (VarName124 >= 5(-> 6)) and (pain6 <= 0(-> 1)) and (VarName22 <= 4(->
5)) => Output=2 (CF = 0.98)
(VarName43 <= 1(-> 2)) and (VarName21 >= 2.5(-> 2.25)) and (VarName140 >= 1(-> 0)) and (VarName91 >= 1(-> 0.667)) => Output=2 (CF = 0.95)

Number of Rules : 24
REGLAS = 24
```

Σχήμα 62

Το Dataset3 έχει 3026 καταχωρήσεις, όπου όπως αναφέρθηκε και τα 5 folds έχουν 0% απόδοση. Για αυτόν το λόγο βλέπουμε ότι δημιουργούνται τόσοι πολλοί ασαφείς κανόνες, όπως στο 0 fold, που δημιουργούνται 29 ασαφείς κανόνες.

```

FURIA rules:
=====

(VarName142 >= 2.5(-> 2)) and (VarName119 <= 62.5(-> 68.8)) => Output=1 (CF = 0.59)
(VarName142 >= 2.5(-> 2)) and (VarName119 <= 62.5(-> 75)) => Output=1 (CF = 0.57)
(VarName142 >= 2.5(-> 2.25)) and (VarName119 <= 62.5(-> 68.8)) => Output=1 (CF = 0.59)
(VarName142 >= 2.5(-> 2)) and (VarName119 <= 62.5(-> 81.3)) => Output=1 (CF = 0.56)
(VarName142 >= 2.5(-> 2)) and (VarName63 >= 1.25(-> 1)) => Output=1 (CF = 0.65)
(VarName142 >= 2.5(-> 2)) => Output=1 (CF = 0.5)
(VarName142 >= 2.5(-> 2)) and (VarName119 <= 62.5(-> 68.8)) and (VarName45 <= 0.75(-> 1)) and (VarName34 >= 1(-> 0)) => Output=1 (CF = 0.68)
(VarName142 >= 2.5(-> 2)) and (VarName119 <= 75(-> 81.3)) and (pain8 <= 0.25(-> 1)) => Output=1 (CF = 0.56)
(VarName142 >= 2.5(-> 2)) and (VarName119 <= 75(-> 81.3)) => Output=1 (CF = 0.54)
(VarName142 >= 2.5(-> 2)) and (VarName119 <= 75(-> 81.3)) => Output=1 (CF = 0.54)
(VarName142 >= 2.5(-> 2.25)) => Output=1 (CF = 0.5)
(VarName142 >= 2.5(-> 1)) => Output=1 (CF = 0.38)
(VarName142 >= 2.5(-> 2)) => Output=1 (CF = 0.5)
(VarName142 >= 2.5(-> 2)) and (VarName78 <= 0.75(-> 1)) => Output=1 (CF = 0.47)
(VarName142 >= 2.5(-> 2)) => Output=1 (CF = 0.5)
(VarName142 >= 2.5(-> 2)) => Output=1 (CF = 0.5)
(VarName142 >= 2.75(-> 2)) and (VarName32 <= 0(-> 1)) and (VarName114 <= 81.3(-> 83.3)) => Output=1 (CF = 0.56)
(VarName142 >= 4(-> 3)) => Output=1 (CF = 0.53)
(VarName142 >= 2.5(-> 2)) and (VarName91 <= 0.667(-> 0.75)) => Output=1 (CF = 0.47)
(VarName142 >= 2.75(-> 2)) => Output=1 (CF = 0.5)
(VarName142 >= 2.75(-> 2)) => Output=1 (CF = 0.5)
=> Output=3 (CF = 0.0)
=> Output=3 (CF = 0.0)
=> Output=3 (CF = 0.0)
=> Output=3 (CF = 0.0)
=> Output=2 (CF = 0.0)
=> Output=2 (CF = 0.0)
(VarName136 <= 16.6(-> 17)) and (VarName116 >= 64.3(-> 62.5)) => Output=2 (CF = 0.81)
=> Output=2 (CF = 0.0)

```

Σχήμα 63

```

Reglas Buenas
Inside m Ruleset
(VarName142 >= 2.5(-> 2)) and (VarName119 <= 62.5(-> 68.8)) and (VarName111 <= 67.9(-> 70.8)) and (VarName27 <= 0(-> 1)) and (VarName131 <= 12.5(-> 13)) => Output=1 (CF = 0.88)
(VarName142 >= 2.5(-> 2)) and (VarName119 <= 62.5(-> 75)) and (VarName116 <= 62.5(-> 64.3)) and (VarName82 <= 1(-> 2)) and (VarName144 >= 3(-> 2)) => Output=1 (CF = 0.95)
(VarName142 >= 2.5(-> 2.25)) and (VarName119 <= 62.5(-> 68.8)) and (VarName112 >= 12.5(-> 11.7)) and (VarName107 >= 51.08(-> 49.05)) and (VarName106 <= 43.52(-> 44.14)) and (VarName112 <= 29.8(-> 30)) and (VarName100 >= 0.25(-> 0)) and (pain6 <= 0(-> 1)) => Output=1 (CF = 0.97)
(VarName142 >= 2.5(-> 2)) and (VarName119 <= 62.5(-> 81.3)) and (VarName36 >= 0.25(-> 0)) and (VarName106 <= 50.78(-> 51.29)) and (health4 >= 3.5(-> 3)) and (VarName126 <= 2(-> 3)) => Output=1 (CF = 0.94)
(VarName142 >= 2.5(-> 2)) and (VarName63 >= 1.25(-> 1)) and (VarName25 >= 1(-> 0)) and (VarName24 >= 5(-> 4)) and (VarName35 >= 2(-> 1)) and (VarName112 >= 18(-> 17)) => Output=1 (CF = 0.95)
(VarName142 >= 2.5(-> 2)) and (VarName119 <= 43.8(-> 50)) and (VarName99 >= 0.25(-> 0)) and (pain6 <= 0(-> 1)) and (VarName43 >= 1(-> 0)) and (VarName52 >= 1(-> 0)) => Output=1 (CF = 0.94)
(VarName142 >= 2.5(-> 2)) and (VarName119 <= 62.5(-> 68.8)) and (VarName45 <= 0.75(-> 1)) and (VarName34 >= 1(-> 0)) and (VarName126 >= 4(-> 3)) and (VarName106 <= 51.29(-> 52.08)) and (VarName88 <= 1(-> 2)) => Output=1 (CF = 0.94)
(VarName142 >= 2.5(-> 2)) and (VarName119 <= 75(-> 81.3)) and (pain8 <= 0.25(-> 1)) and (VarName69 >= 2(-> 1)) and (VarName114 <= 47.2(-> 50)) and (VarName86 <= 2(-> 3)) => Output=1 (CF = 0.91)
(VarName142 >= 2.5(-> 2)) and (VarName119 <= 75(-> 81.3)) and (VarName83 <= 0(-> 1)) and (VarName99 >= 0.25(-> 0)) and (VarName56 <= 0(-> 1)) and (VarName131 >= 6(-> 5)) and (VarName117 <= 5(-> 6)) and (VarName92 >= 1(-> 0)) => Output=1 (CF = 0.93)
(VarName142 >= 2.5(-> 2)) and (VarName119 <= 75(-> 81.3)) and (VarName61 <= 0(-> 1)) and (VarName30 >= 0.667(-> 0)) and (VarName65 >= 1(-> 0)) and (VarName99 >= 1(-> 0)) and (VarName142 <= 4(-> 5)) => Output=1 (CF = 0.94)
(VarName142 >= 2.5(-> 2.25)) and (VarName100 >= 0.5(-> 0)) and (health3 >= 5(-> 4.75)) and (VarName93 >= 3(-> 2)) and (VarName43 <= 0(-> 1)) and (VarName34 >= 1(-> 0)) and (pain5 <= 0(-> 1)) => Output=1 (CF = 0.88)
(VarName142 >= 2.5(-> 1)) and (VarName92 >= 1.25(-> 1)) and (VarName106 >= 53.56(-> 53.18)) and (VarName45 <= 0.25(-> 1)) and (VarName77 >= 1(-> 0)) and (VarName56 <= 0.25(-> 0.667)) => Output=1 (CF = 0.88)
(VarName142 >= 2.5(-> 2)) and (VarName100 >= 1(-> 0)) and (VarName36 >= 0.25(-> 0)) and (VarName52 <= 0(-> 1)) and (VarName78 >= 1(-> 0.75)) and (VarName20 <= 2(-> 3)) => Output=1 (CF = 0.96)
(VarName142 >= 2.5(-> 2)) and (VarName78 <= 0.75(-> 1)) and (VarName109 <= 78.1(-> 83.3)) and (VarName86 >= 0.333(-> 0.25)) and (VarName121 <= 0(-> 1)) and (VarName28 <= 2(-> 3)) => Output=1 (CF = 0.93)
(VarName142 >= 2.5(-> 2)) and (VarName145 >= 12(-> 8)) and (VarName72 >= 1(-> 0)) and (VarName60 <= 0(-> 1)) => Output=1 (CF = 0.92)

```

Σχήμα 64

```

(VarName142 >= 2.5(-> 2)) and (VarName25 >= 2(-> 1)) and (VarName50 <= 0(-> 1)) and (VarName30 >= 2(-> 1.25)) and (VarName111 <=
96.4(-> 100)) => Output=1 (CF = 0.95)
(VarName142 >= 2.75(-> 2)) and (VarName32 <= 0(-> 1)) and (VarName114 <= 81.3(-> 83.3)) and (VarName89 <= 0.25(-> 0.5)) and (
VarName85 <= 0(-> 1)) => Output=1 (CF = 0.91)
(VarName142 >= 4(-> 3)) and (VarName106 >= 56.11(-> 55.87)) and (VarName91 >= 1.75(-> 1)) and (VarName115 <= 2(-> 3)) => Output=1 (
CF = 0.82)
(VarName142 >= 2.5(-> 2)) and (VarName91 <= 0.667(-> 0.75)) and (VarName91 >= 0.333(-> 0.25)) => Output=1 (CF = 0.88)
(VarName142 >= 2.75(-> 2)) and (VarName40 >= 2(-> 1)) and (VarName124 >= 3(-> 2)) and (VarName88 <= 1(-> 2)) => Output=1 (CF =
0.87)
(VarName142 >= 2.75(-> 2)) and (VarName99 <= 0(-> 1)) and (VarName86 >= 1.5(-> 1.333)) and (VarName106 <= 49.95(-> 51)) =>
Output=1 (CF = 0.92)
(pain <= 0(-> 1)) and (health <= 1(-> 2)) and (pain5 <= 0(-> 1)) and (VarName116 >= 100(-> 96.4)) and (pain7 <= 0(-> 1)) =>
Output=3 (CF = 0.66)
(pain <= 0(-> 1)) and (VarName119 >= 100(-> 93.8)) and (VarName23 <= 4(-> 5)) and (VarName106 <= 58.84(-> 58.91)) and (health3 >=
5(-> 4)) and (pain1 <= 0(-> 1)) => Output=3 (CF = 0.61)
(pain <= 0(-> 1)) and (VarName142 <= 0(-> 1)) and (VarName92 <= 0(-> 1)) and (health3 <= 4(-> 5)) and (VarName106 >= 52.15(->
51.84)) and (VarName107 >= 48.23(-> 47.88)) => Output=3 (CF = 0.7)
(pain <= 0(-> 1)) and (VarName106 >= 60.47(-> 59.08)) and (VarName37 >= 1(-> 0)) and (health <= 2(-> 3)) => Output=3 (CF = 0.75)
(VarName142 <= 2.25(-> 2.5)) => Output=2 (CF = 0.93)
(VarName119 >= 66.7(-> 62.5)) => Output=2 (CF = 0.84)
(VarName136 <= 16.6(-> 17)) and (VarName116 >= 64.3(-> 62.5)) and (VarName82 >= 1(-> 0)) => Output=2 (CF = 0.8)
(pain6 >= 1(-> 0)) and (VarName63 <= 1(-> 1.333)) and (VarName58 >= 1(-> 0)) and (VarName91 <= 2(-> 2.25)) => Output=2 (CF = 0.83)

Number of Rules : 29

REGLAS = 29

```

Σχήμα 65

To 1 fold δημιουργεί 23 ασαφείς κανόνες.

```

FURIA rules:
=====
(VarName142 >= 2.5(-> 2)) => Output=1 (CF = 0.49)
(VarName142 >= 2.5(-> 2)) => Output=1 (CF = 0.49)
(VarName142 >= 2.5(-> 2)) => Output=1 (CF = 0.49)
(VarName142 >= 2.5(-> 2)) and (VarName111 <= 87.5(-> 89.3)) and (pain6 <= 0(-> 1)) and (VarName94 >= 2(-> 1)) => Output=1 (CF =
0.72)
(VarName142 >= 2.5(-> 2)) and (VarName110 >= 2(-> 1)) and (VarName24 >= 4.75(-> 4)) => Output=1 (CF = 0.61)
(VarName142 >= 2.5(-> 2)) and (VarName110 >= 2(-> 1)) and (VarName24 >= 4.75(-> 4)) => Output=1 (CF = 0.61)
(VarName142 >= 2.5(-> 2)) => Output=1 (CF = 0.49)
(VarName142 >= 2.5(-> 2)) and (VarName100 >= 0.25(-> 0)) and (VarName35 >= 1(-> 0)) and (pain6 <= 0(-> 1)) => Output=1 (CF = 0.65)
(VarName142 >= 2.5(-> 2)) and (VarName93 >= 3(-> 1)) => Output=1 (CF = 0.53)
(VarName142 >= 2.5(-> 2)) and (VarName100 >= 0.25(-> 0)) and (VarName24 >= 4.75(-> 4.5)) and (VarName35 >= 1(-> 0)) and (
VarName114 >= 100(-> 97.2)) and (VarName95 >= 1(-> 0)) and (VarName43 <= 1(-> 2)) => Output=1 (CF = 0.87)
(VarName142 >= 2.5(-> 2)) and (VarName22 <= 2(-> 3)) and (VarName111 <= 85.7(-> 89.3)) and (pain8 <= 0.25(-> 1)) and (VarName100
>= 0.25(-> 0)) => Output=1 (CF = 0.71)
(VarName142 >= 2.5(-> 2)) and (VarName83 <= 0.25(-> 1)) and (VarName93 >= 3(-> 2)) => Output=1 (CF = 0.54)
(VarName142 >= 2.5(-> 2)) and (VarName63 <= 0(-> 1)) and (VarName107 >= 46.8(-> 46.53)) => Output=1 (CF = 0.45)
(VarName142 >= 2.5(-> 2)) and (VarName43 >= 1(-> 0)) => Output=1 (CF = 0.55)
(VarName142 >= 2.75(-> 2)) => Output=1 (CF = 0.49)
(VarName142 >= 2.75(-> 2)) => Output=1 (CF = 0.49)
(VarName142 >= 2.75(-> 2)) => Output=1 (CF = 0.49)
=> Output=3 (CF = 0.0)
=> Output=3 (CF = 0.0)
=> Output=2 (CF = 0.0)
=> Output=2 (CF = 0.0)
=> Output=2 (CF = 0.0)
(VarName106 >= 43.62(-> 41.82)) => Output=2 (CF = 0.76)

```

Σχήμα 66

```

Reglas Buenas
Inside m Ruleset
(VarName142 >= 2.5(-> 2)) and (VarName116 <= 53.6(-> 57.1)) and (VarName73 >= 3(-> 2)) and (VarName70 >= 2(-> 1)) => Output=1 (CF = 0.97)
(VarName142 >= 2.5(-> 2)) and (VarName111 <= 64.3(-> 67.9)) and (VarName145 >= 12(-> 18)) and (VarName145 <= 28(-> 38)) => Output=1 (CF = 0.94)
(VarName142 >= 2.5(-> 2)) and (VarName111 <= 64.3(-> 67.9)) and (VarName36 >= 1.5(-> 1)) and (VarName106 <= 48.62(-> 51.02)) and (VarName55 <= 1(-> 2)) and (
pain9 <= 0(-> 1)) => Output=1 (CF = 0.96)
(VarName142 >= 2.5(-> 2)) and (VarName111 <= 87.5(-> 89.3)) and (pain6 <= 0(-> 1)) and (VarName94 >= 2(-> 1)) and (VarName124 >= 4(-> 3)) and (VarName75 <= 2(->
3)) and (VarName105 <= 0(-> 4)) and (VarName77 <= 2(-> 3)) => Output=1 (CF = 0.96)
(VarName142 >= 2.5(-> 2)) and (VarName110 >= 2(-> 1)) and (VarName24 >= 4.75(-> 4)) and (VarName46 >= 1(-> 0)) and (VarName109 <= 55.6(-> 56.3)) and (VarName97
<= 4(-> 5)) => Output=1 (CF = 0.95)
(VarName142 >= 2.5(-> 2)) and (VarName110 >= 2(-> 1)) and (VarName24 >= 4.75(-> 4)) and (VarName36 >= 0.75(-> 0)) and (health4 <= 4(-> 5)) and (VarName109 <=
75(-> 77.8)) and (VarName94 >= 1(-> 0)) => Output=1 (CF = 0.96)
(VarName142 >= 2.5(-> 2)) and (VarName116 <= 75(-> 78.6)) and (VarName78 <= 1.25(-> 2)) and (VarName21 <= 2.25(-> 3)) and (health3 <= 4(-> 5)) and (VarName106
>= 46.03(-> 46.02)) => Output=1 (CF = 0.94)
(VarName142 >= 2.5(-> 2)) and (VarName100 >= 0.25(-> 0)) and (VarName35 >= 1(-> 0)) and (pain6 <= 0(-> 1)) and (VarName99 >= 0.25(-> 0)) and (health1 <= 2(->
2.75)) and (VarName43 <= 1(-> 2)) and (VarName107 <= 62.43(-> 64.8)) => Output=1 (CF = 0.94)
(VarName142 >= 2.5(-> 2)) and (VarName93 >= 3(-> 1)) and (VarName73 >= 2(-> 1)) and (VarName95 <= 1(-> 2)) and (VarName69 >= 2(-> 1)) and (VarName25 <= 1(-> 2))
=> Output=1 (CF = 0.94)
(VarName142 >= 2.5(-> 2)) and (VarName100 >= 0.25(-> 0)) and (VarName24 >= 4.75(-> 4.5)) and (VarName35 >= 1(-> 0)) and (VarName114 >= 100(-> 97.2)) and (
VarName95 >= 1(-> 0)) and (VarName43 <= 1(-> 2)) and (health3 >= 5(-> 4.75)) => Output=1 (CF = 0.93)
(VarName142 >= 2.5(-> 2)) and (VarName22 <= 2(-> 3)) and (VarName111 <= 85.7(-> 89.3)) and (pain8 <= 0.25(-> 1)) and (VarName100 >= 0.25(-> 0)) and (VarName117
>= 16(-> 14.9)) and (VarName24 >= 5(-> 4)) and (VarName78 >= 0.75(-> 0)) => Output=1 (CF = 0.96)
(VarName142 >= 2.5(-> 2)) and (VarName83 <= 0.25(-> 1)) and (VarName93 >= 3(-> 2)) and (VarName56 <= 0.75(-> 1)) and (VarName107 >= 60.6(-> 60.3)) and (
VarName145 <= 4(-> 5)) and (VarName125 <= 3(-> 4)) => Output=1 (CF = 0.93)
(VarName142 >= 2.5(-> 2)) and (VarName63 <= 0(-> 1)) and (VarName107 >= 46.8(-> 46.53)) and (VarName107 <= 48.81(-> 49.02)) and (VarName136 <= 6(-> 8)) =>
Output=1 (CF = 0.87)
(VarName142 >= 2.5(-> 2)) and (VarName43 >= 1(-> 0)) and (VarName74 >= 2(-> 1)) and (VarName71 <= 0(-> 1)) and (VarName106 <= 43.33(-> 43.85)) and (VarName113
<= 8.8(-> 9)) => Output=1 (CF = 0.94)
(VarName142 >= 2.75(-> 2)) and (VarName106 >= 53.2(-> 52.91)) and (VarName73 >= 2(-> 1)) and (VarName7 <= 0(-> 1)) => Output=1 (CF = 0.93)
(VarName142 >= 2.75(-> 2)) and (VarName37 >= 2(-> 1)) and (VarName66 <= 0(-> 1)) and (VarName136 >= 15(-> 12)) and (VarName43 <= 1(-> 2)) and (VarName106 >=
46.29(-> 46.27)) => Output=1 (CF = 0.93)
(VarName142 >= 2.75(-> 2)) and (VarName45 <= 0.75(-> 1)) and (VarName107 <= 59.14(-> 60.12)) and (VarName65 >= 1(-> 0)) and (VarName125 >= 4(-> 3)) and (
VarName106 >= 52.91(-> 52.1)) => Output=1 (CF = 0.89)
(pain <= 0(-> 1)) and (VarName106 >= 57.28(-> 56.97)) and (VarName142 <= 0(-> 1)) and (VarName116 >= 100(-> 96.4)) and (pain1 <= 0(-> 1)) and (pain7 <= 0(-> 1))
=> Output=3 (CF = 0.69)
(pain <= 0(-> 1)) and (VarName142 <= 0(-> 1)) and (VarName92 <= 0(-> 1)) and (VarName37 >= 1(-> 0)) and (VarName107 <= 55.34(-> 56.33)) => Output=3 (CF = 0.71)
(VarName142 <= 2(-> 2.5)) => Output=2 (CF = 0.93)
(VarName119 >= 66.7(-> 62.5)) => Output=2 (CF = 0.83)
(VarName111 >= 85.7(-> 82.1)) and (VarName114 >= 88.9(-> 87.5)) => Output=2 (CF = 0.85)
(VarName106 >= 43.62(-> 41.82)) and (VarName127 >= 1(-> 0.25)) => Output=2 (CF = 0.84)

Number of Rules : 23
REGLAS = 23

```

Σχήμα 67

Το 2 fold επίσης δημιουργεί 23 ασαφείς κανόνες, αλλά έχει διαφορετικές τιμές μέσα στους κανόνες.

```

FURIA rules:
=====

(VarName142 >= 2.5(-> 2)) and (VarName119 <= 62.5(-> 68.8)) => Output=1 (CF = 0.58)
(VarName142 >= 2.75(-> 2)) and (VarName119 <= 62.5(-> 68.8)) and (VarName107 >= 54.843(-> 52.97)) and (VarName136 >= 8.5(-> 8.4))
and (pain6 <= 0(-> 1)) => Output=1 (CF = 0.75)
(VarName142 >= 2.5(-> 2)) and (VarName119 <= 62.5(-> 68.8)) => Output=1 (CF = 0.58)
(VarName142 >= 2.5(-> 2)) and (VarName119 <= 62.5(-> 68.8)) => Output=1 (CF = 0.58)
(VarName142 >= 2.5(-> 2)) and (VarName119 <= 62.5(-> 68.8)) and (VarName24 >= 5(-> 4)) => Output=1 (CF = 0.62)
(VarName142 >= 2.5(-> 2)) and (VarName93 >= 3(-> 2)) => Output=1 (CF = 0.56)
(VarName142 >= 2.5(-> 2)) and (VarName96 >= 2(-> 1)) and (VarName81 <= 0.75(-> 1)) and (VarName26 <= 1(-> 2)) => Output=1 (CF =
0.66)
(VarName142 >= 2.5(-> 2)) => Output=1 (CF = 0.5)
(VarName142 >= 2.5(-> 2)) => Output=1 (CF = 0.5)
(VarName142 >= 2.5(-> 2)) and (VarName92 >= 1.25(-> 1)) => Output=1 (CF = 0.57)
(VarName142 >= 2.5(-> 2)) and (VarName93 >= 3(-> 2.25)) => Output=1 (CF = 0.56)
(VarName142 >= 2.5(-> 2)) => Output=1 (CF = 0.5)
(VarName142 >= 2.5(-> 2)) and (VarName109 <= 78.1(-> 80.6)) and (VarName51 <= 0(-> 1)) => Output=1 (CF = 0.55)
(VarName142 >= 2.75(-> 2)) and (VarName83 <= 0(-> 1)) and (VarName34 >= 1(-> 0)) and (VarName106 >= 48.33(-> 48.02)) => Output=1 (
CF = 0.5)
(VarName142 >= 2.5(-> 2)) => Output=1 (CF = 0.5)
(VarName142 >= 2.75(-> 2)) and (pain6 <= 0(-> 1)) and (VarName94 >= 2(-> 1)) and (VarName107 >= 54.67(-> 54.1)) => Output=1 (CF =
0.75)
(VarName142 >= 2.75(-> 2.5)) and (VarName100 >= 0.25(-> 0)) => Output=1 (CF = 0.55)
(VarName142 >= 2.75(-> 2)) and (pain6 <= 0(-> 1)) and (VarName100 >= 0.25(-> 0)) => Output=1 (CF = 0.59)
=> Output=3 (CF = 0.0)
=> Output=2 (CF = 0.0)
=> Output=2 (CF = 0.0)
(VarName116 >= 66.7(-> 64.3)) => Output=2 (CF = 0.76)
(VarName116 >= 67.9(-> 64.3)) => Output=2 (CF = 0.76)

```

Σχήμα 68



```

Reglas Buenas
Inside = Ruleset
(VarName142 >= 2.5(-> 2)) and (VarName119 <= 62.5(-> 68.8)) and (VarName111 <= 66.7(-> 67.9)) and (VarName50 <= 1(-> 2)) and (VarName73 >= 1(-> 0)) => Output=1 (CF = 0.92)
(VarName142 >= 2.75(-> 2)) and (VarName119 <= 62.5(-> 68.8)) and (VarName107 >= 54.843(-> 52.97)) and (VarName136 >= 8.5(-> 8.4)) and (pain6 <= 0(-> 1)) and (VarName106 <= 49.75(-> 50.73)) and (VarName24 >= 5(-> 4)) => Output=1 (CF = 0.86)
(VarName142 >= 2.5(-> 2)) and (VarName119 <= 62.5(-> 68.8)) and (VarName116 <= 64.3(-> 67.9)) and (VarName69 >= 3(-> 2)) and (VarName145 <= 28(-> 30)) and (VarName21 >= 2(-> 1.5)) => Output=1 (CF = 0.96)
(VarName142 >= 2.5(-> 2)) and (VarName119 <= 62.5(-> 68.8)) and (VarName145 >= 21(-> 20)) and (VarName136 >= 19(-> 16.6)) and (VarName110 <= 3(-> 4)) => Output=1 (CF = 0.95)
(VarName142 >= 2.5(-> 2)) and (VarName119 <= 62.5(-> 68.8)) and (VarName24 >= 5(-> 4)) and (VarName59 >= 2(-> 1)) and (VarName25 >= 1(-> 0)) and (VarName109 >= 72.2(-> 69.4)) and (VarName25 <= 1(-> 2)) => Output=1 (CF = 0.95)
(VarName142 >= 2.5(-> 2)) and (VarName93 >= 3(-> 2)) and (VarName30 >= 2(-> 1.25)) and (VarName37 >= 3(-> 2)) and (VarName109 >= 38.9(-> 36.1)) and (VarName51 >= 1(-> 0)) => Output=1 (CF = 0.93)
(VarName142 >= 2.5(-> 2)) and (VarName96 >= 2(-> 1)) and (VarName81 <= 0.75(-> 1)) and (VarName26 <= 1(-> 2)) and (VarName97 <= 2(-> 3)) => Output=1 (CF = 0.76)
(VarName142 >= 2.5(-> 2)) and (VarName114 <= 81.3(-> 83.3)) and (VarName131 <= 18(-> 19.2)) and (VarName36 >= 0.25(-> 0)) => Output=1 (CF = 0.85)
(VarName142 >= 2.5(-> 2)) and (VarName73 <= 4(-> 3)) and (VarName86 <= 0.25(-> 0.333)) and (pain >= 1(-> 0)) => Output=1 (CF = 0.94)
(VarName142 >= 2.5(-> 2)) and (VarName92 >= 1.25(-> 1)) and (VarName86 >= 0.25(-> 0)) and (VarName97 <= 2(-> 3)) and (VarName109 <= 75(-> 77.8)) and (VarName123 <= 5(-> 6)) => Output=1 (CF = 0.93)
(VarName142 >= 2.5(-> 2)) and (VarName93 >= 3(-> 2.25)) and (VarName43 <= 0(-> 1)) and (VarName53 >= 0.25(-> 0)) and (VarName107 >= 56.86(-> 56.82)) => Output=1 (CF = 0.7)
(VarName142 >= 2.5(-> 2)) and (VarName58 >= 1(-> 0)) and (VarName112 <= 6.4(-> 7)) and (VarName111 <= 96.4(-> 100)) and (VarName74 >= 0.25(-> 0)) and (VarName97 <= 3(-> 4)) and (VarName89 <= 1(-> 2)) => Output=1 (CF = 0.78)
(VarName142 >= 2.5(-> 2)) and (VarName109 <= 78.1(-> 80.6)) and (VarName51 <= 0(-> 1)) and (VarName106 <= 45.76(-> 46.27)) and (VarName123 >= 6(-> 5)) and (VarName136 >= 15.7(-> 15)) => Output=1 (CF = 0.9)
(VarName142 >= 2.75(-> 2)) and (VarName83 <= 0(-> 1)) and (VarName34 >= 1(-> 0)) and (VarName106 >= 48.33(-> 48.02)) and (VarName106 <= 51.61(-> 51.65)) and (VarName37 >= 2(-> 1)) => Output=1 (CF = 0.83)
(VarName142 >= 2.5(-> 2)) and (VarName112 <= 1(-> 2)) and (VarName75 >= 0.25(-> 0)) and (VarName107 <= 56.79(-> 57.06)) => Output=1 (CF = 0.7)
(VarName142 >= 2.75(-> 2)) and (pain6 <= 0(-> 1)) and (VarName94 >= 2(-> 1)) and (VarName107 >= 54.67(-> 54.1)) and (VarName92 <= 0.667(-> 0.75)) and (VarName93 >= 2(-> 1)) => Output=1 (CF = 0.81)
(VarName142 >= 2.75(-> 2.5)) and (VarName100 >= 0.25(-> 0)) and (VarName105 >= 7(-> 6)) and (pain3 >= 1(-> 0)) and (VarName108 >= 4(-> 3)) and (VarName64 <= 1(-> 2)) => Output=1 (CF = 0.86)
(VarName142 >= 2.75(-> 2)) and (pain6 <= 0(-> 1)) and (VarName100 >= 0.25(-> 0)) and (VarName115 >= 2(-> 1)) and (VarName20 <= 1(-> 2)) and (VarName83 <= 0(-> 1)) and (VarName21 <= 3(-> 4)) => Output=1 (CF = 0.92)
(pain <= 0(-> 1)) and (VarName142 <= 0(-> 1)) and (VarName106 >= 55.33(-> 55.09)) and (VarName119 >= 91.7(-> 87.5)) => Output=3 (CF = 0.51)
(VarName142 <= 2.25(-> 2.5)) => Output=2 (CF = 0.93)
(VarName119 >= 66.7(-> 62.5)) => Output=2 (CF = 0.84)
(VarName116 >= 67.7(-> 64.3)) and (VarName107 <= 52.48(-> 52.69)) and (VarName124 <= 2.75(-> 3)) and (VarName40 <= 0(-> 1)) => Output=2 (CF = 0.84)
(VarName116 >= 67.9(-> 64.3)) and (VarName85 >= 2(-> 1)) and (VarName33 <= 2(-> 3)) => Output=2 (CF = 0.9)

Number of Rules : 23
REGLAS = 23

```

Σχήμα 69

Παρατηρούμε επίσης ότι το 3 παράγει 23 ασαφείς κανόνες, αλλά πάλι με διαφορετικές τιμές.

```

FURIA rules:
=====
(VarName142 >= 2.5(-> 2)) and (VarName119 <= 62.5(-> 68.8)) => Output=1 (CF = 0.58)
(VarName142 >= 2.5(-> 2)) and (VarName119 <= 62.5(-> 68.775)) => Output=1 (CF = 0.58)
(VarName142 >= 2.5(-> 2)) and (VarName119 <= 62.5(-> 68.8)) and (pain6 <= 0(-> 1)) and (VarName58 >= 1(-> 0)) => Output=1 (CF = 0.73)
(VarName142 >= 2.5(-> 2)) and (VarName119 <= 62.5(-> 68.8)) and (pain6 <= 0(-> 1)) and (VarName58 >= 1(-> 0)) and (VarName106 <= 53.85(-> 54.51)) and (VarName55 <= 0(-> 1)) => Output=1 (CF = 0.85)
(VarName142 >= 2.5(-> 2.25)) => Output=1 (CF = 0.5)
(VarName142 >= 2.5(-> 2.25)) => Output=1 (CF = 0.5)
(VarName142 >= 2.5(-> 2.25)) => Output=1 (CF = 0.5)
(VarName142 >= 2.5(-> 2)) => Output=1 (CF = 0.5)
(VarName142 >= 2.75(-> 2.5)) => Output=1 (CF = 0.5)
(VarName142 >= 2.75(-> 2.5)) and (VarName119 <= 62.5(-> 68.8)) and (VarName21 <= 2(-> 3)) and (VarName19 <= 4.5(-> 5)) and (VarName101 >= 1(-> 0)) => Output=1 (CF = 0.8)
(VarName142 >= 2.75(-> 2)) and (VarName119 <= 62.5(-> 68.8)) and (VarName91 <= 1.25(-> 1.333)) => Output=1 (CF = 0.59)
(VarName142 >= 2.75(-> 2)) => Output=1 (CF = 0.5)
(VarName142 >= 2.75(-> 2)) => Output=1 (CF = 0.5)
(VarName142 >= 2.75(-> 2)) and (VarName119 <= 62.5(-> 68.8)) => Output=1 (CF = 0.58)
(VarName142 >= 2.75(-> 2)) and (VarName50 <= 0(-> 1)) => Output=1 (CF = 0.47)
(VarName142 >= 2.75(-> 2)) => Output=1 (CF = 0.5)
(VarName142 >= 2.75(-> 1)) => Output=1 (CF = 0.4)
(VarName142 >= 2.75(-> 2.5)) => Output=1 (CF = 0.5)
=> Output=3 (CF = 0.0)
=> Output=3 (CF = 0.0)
=> Output=2 (CF = 0.0)
=> Output=2 (CF = 0.0)
(VarName111 >= 82.1(-> 78.6)) => Output=2 (CF = 0.79)

```

Σχήμα 70

```

Reglas Buenas
Inside m_Ruleset
(VarName142 >= 2.5(-> 2)) and (VarName119 <= 62.5(-> 68.8)) and (VarName111 <= 66.7(-> 67.9)) and (VarName46 <= 1(-> 2)) and (VarName50 <= 1(-> 2)) and (
VarName127 <= 0(-> 1)) and (health >= 2(-> 1)) => Output=1 (CF = 0.97)
(VarName142 >= 2.5(-> 2)) and (VarName119 <= 62.5(-> 68.775)) and (VarName116 <= 64.3(-> 67.9)) and (VarName94 >= 2(-> 1)) and (VarName103 >= 1(-> 0)) and (
VarName84 <= 2(-> 3)) => Output=1 (CF = 0.95)
(VarName142 >= 2.5(-> 2)) and (VarName119 <= 62.5(-> 68.8)) and (pain6 <= 0(-> 1)) and (VarName58 >= 1(-> 0)) and (VarName107 >= 55.18(-> 54.94)) and (VarName22
>= 2.25(-> 2)) and (VarName34 <= 3(-> 4)) => Output=1 (CF = 0.95)
(VarName142 >= 2.5(-> 2)) and (VarName119 <= 62.5(-> 68.8)) and (pain6 <= 0(-> 1)) and (VarName58 >= 1(-> 0)) and (VarName106 <= 53.85(-> 54.51)) and (VarName55
<= 0(-> 1)) and (VarName60 <= 1(-> 2)) and (VarName107 <= 63.66(-> 64.38)) => Output=1 (CF = 0.97)
(VarName142 >= 2.5(-> 2.25)) and (VarName100 >= 0.25(-> 0)) and (VarName36 >= 0.25(-> 0)) and (VarName24 >= 5(-> 4)) and (VarName106 <= 51(-> 51.02)) and (
VarName42 <= 1(-> 2)) and (VarName59 <= 0(-> 1)) => Output=1 (CF = 0.95)
(VarName142 >= 2.5(-> 2.25)) and (VarName100 >= 0.25(-> 0)) and (VarName24 >= 4.5(-> 4)) and (VarName65 >= 0.25(-> 0)) and (VarName125 >= 6(-> 5)) and (
VarName123 >= 5(-> 4)) and (VarName22 <= 3(-> 4)) => Output=1 (CF = 0.96)
(VarName142 >= 2.5(-> 2.25)) and (VarName100 >= 0.25(-> 0)) and (VarName36 >= 1.5(-> 1)) and (pain5 <= 0(-> 0.75)) and (VarName94 >= 1(-> 0)) and (VarName37 <=
2(-> 3)) => Output=1 (CF = 0.96)
(VarName142 >= 2.5(-> 2)) and (VarName73 >= 2(-> 1)) and (VarName145 <= 1.25(-> 2)) and (VarName70 >= 1(-> 0)) and (pain6 <= 0(-> 1)) => Output=1 (CF = 0.95)
(VarName142 >= 2.75(-> 2.5)) and (VarName100 >= 0.25(-> 0)) and (VarName22 <= 2(-> 3)) and (VarName65 >= 0.25(-> 0)) and (VarName106 >= 54.8(-> 54.26)) and (
VarName87 <= 1(-> 2)) and (VarName117 >= 1(-> 0)) => Output=1 (CF = 0.94)
(VarName142 >= 2.75(-> 2.5)) and (VarName119 <= 62.5(-> 68.8)) and (VarName21 <= 2(-> 3)) and (VarName19 <= 4.5(-> 5)) and (VarName101 >= 1(-> 0)) and (pain7 <=
0(-> 1)) and (VarName67 <= 2(-> 4)) => Output=1 (CF = 0.94)
(VarName142 >= 2.75(-> 2)) and (VarName119 <= 62.5(-> 68.8)) and (VarName91 <= 1.25(-> 1.333)) and (VarName112 >= 12.8(-> 12)) and (VarName51 <= 0(-> 1)) =>
Output=1 (CF = 0.88)
(VarName142 >= 2.75(-> 2)) and (VarName107 >= 57.16(-> 56.74)) and (VarName91 <= 1.25(-> 1.333)) and (VarName114 <= 86.1(-> 87.5)) and (VarName117 <= 8(-> 8.5))
=> Output=1 (CF = 0.83)
(VarName142 >= 2.75(-> 2)) and (VarName145 >= 2(-> 0)) and (VarName107 >= 59.47(-> 59.29)) and (VarName144 <= 2(-> 3)) and (VarName20 <= 3(-> 4)) => Output=1 (
CF = 0.88)
(VarName142 >= 2.75(-> 2)) and (VarName119 <= 62.5(-> 68.8)) and (VarName89 <= 0.25(-> 0.333)) and (VarName42 <= 1(-> 2)) and (VarName112 >= 6(-> 5.3)) and (
VarName31 <= 0(-> 1)) => Output=1 (CF = 0.88)
(VarName142 >= 2.75(-> 2)) and (VarName50 <= 0(-> 1)) and (VarName110 >= 2(-> 1)) and (VarName109 >= 81.3(-> 80.6)) and (VarName95 >= 1(-> 0)) and (VarName78 >=
0.667(-> 0)) and (VarName92 <= 2(-> 3)) and (VarName45 <= 1.75(-> 2)) => Output=1 (CF = 0.94)
(VarName142 >= 2.75(-> 2)) and (VarName106 >= 53.24(-> 52.52)) and (VarName112 <= 1(-> 1.1)) and (VarName97 >= 2(-> 1)) and (VarName126 >= 4(-> 3)) => Output=1 (
CF = 0.91)
(VarName142 >= 2.75(-> 1)) and (VarName73 >= 3(-> 2)) and (VarName125 <= 1(-> 3)) and (VarName56 >= 1(-> 0)) => Output=1 (CF = 0.9)
(VarName142 >= 2.75(-> 2.5)) and (VarName63 <= 0(-> 0.25)) and (VarName145 >= 14(-> 10)) and (VarName25 >= 1(-> 0)) and (VarName143 <= 3(-> 3.5)) and (VarName65
<= 0(-> 1)) => Output=1 (CF = 0.93)
(pain <= 0(-> 1)) and (health <= 1(-> 2)) and (pain7 <= 0(-> 1)) and (VarName116 >= 100(-> 96.4)) => Output=3 (CF = 0.68)
(pain <= 0(-> 1)) and (VarName119 >= 100(-> 93.8)) and (VarName106 >= 55.33(-> 45.34)) and (VarName107 >= 49.12(-> 49.09)) and (VarName107 <= 54.2(-> 54.35))
and (pain7 <= 0(-> 1)) and (pain5 <= 0(-> 1)) and (health <= 2(-> 3)) => Output=3 (CF = 0.87)
(VarName142 <= 2.25(-> 2.5)) => Output=2 (CF = 0.93)
(VarName119 >= 68.8(-> 62.5)) => Output=2 (CF = 0.84)
(VarName111 >= 82.1(-> 78.6)) and (VarName107 <= 52.48(-> 52.5)) => Output=2 (CF = 0.81)

Number of Rules : 23
REGLAS = 23

```

## Σχήμα 71

Και τέλος το 4 fold παράγει επίσης 28 κανόνες.

```
FURIA rules:
=====

(VarName142 >= 2.75(-> 2)) => Output=1 (CF = 0.5)
(VarName142 >= 2.75(-> 2)) => Output=1 (CF = 0.5)
(VarName142 >= 2.75(-> 2)) and (VarName119 <= 62.5(-> 75)) and (VarName136 >= 17.5(-> 14.4)) and (VarName24 >= 5(-> 4)) and (
VarName106 <= 43.46(-> 43.7)) => Output=1 (CF = 0.85)
(VarName142 >= 2.75(-> 2)) and (VarName119 <= 62.5(-> 75)) => Output=1 (CF = 0.57)
(VarName142 >= 2.75(-> 2)) => Output=1 (CF = 0.5)
(VarName142 >= 2.75(-> 2)) => Output=1 (CF = 0.5)
(VarName142 >= 2.75(-> 2.25)) and (VarName100 >= 0.25(-> 0)) => Output=1 (CF = 0.56)
(VarName142 >= 2.75(-> 2)) => Output=1 (CF = 0.5)
(VarName142 >= 2.75(-> 2)) and (VarName119 <= 68.8(-> 75)) and (pain6 <= 0(-> 1)) and (VarName35 >= 1(-> 0)) => Output=1 (CF =
0.67)
(VarName142 >= 2.75(-> 2)) and (VarName111 <= 87.5(-> 89.3)) and (pain8 <= 0.25(-> 1)) => Output=1 (CF = 0.62)
(VarName142 >= 2.75(-> 2)) and (VarName145 >= 12(-> 10)) => Output=1 (CF = 0.6)
(VarName142 >= 2.75(-> 2)) => Output=1 (CF = 0.5)
(VarName142 >= 2.75(-> 2)) => Output=1 (CF = 0.5)
=> Output=1 (CF = 0.0)
(VarName142 >= 2.75(-> 2)) and (VarName22 <= 2.25(-> 3)) => Output=1 (CF = 0.51)
(VarName142 >= 2.75(-> 2)) => Output=1 (CF = 0.5)
(VarName142 >= 2.75(-> 2.25)) and (VarName100 >= 0.25(-> 0)) => Output=1 (CF = 0.56)
(VarName142 >= 2.75(-> 2)) => Output=1 (CF = 0.5)
(VarName142 >= 2.75(-> 2)) => Output=1 (CF = 0.5)
(VarName142 >= 2.75(-> 2)) => Output=1 (CF = 0.5)
(VarName142 >= 2.75(-> 2)) => Output=1 (CF = 0.5)
(VarName142 >= 3(-> 2)) and (VarName22 <= 2.25(-> 4)) => Output=1 (CF = 0.5)
(VarName142 >= 3(-> 2)) => Output=1 (CF = 0.5)
=> Output=3 (CF = 0.0)
=> Output=3 (CF = 0.0)
=> Output=2 (CF = 0.0)
=> Output=2 (CF = 0.0)
(VarName116 >= 67.9(-> 64.3)) => Output=2 (CF = 0.76)
```

Σχήμα 72



```

Reglas Buenas
Inside m Ruleset
(VarName142 >= 2.75(-> 2)) and (VarName96 >= 2(-> 1)) and (VarName111 <= 66.7(-> 67.9)) and (VarName54 <= 1(-> 2)) and (
VarName21 <= 2(-> 3)) and (VarName37 >= 3(-> 2)) => Output=1 (CF = 0.96)
(VarName142 >= 2.75(-> 2)) and (VarName96 >= 2(-> 1)) and (VarName116 <= 64.3(-> 67.9)) and (VarName69 >= 3(-> 2)) and (
VarName145 <= 28(-> 30)) and (VarName106 <= 55.6(-> 57.76)) => Output=1 (CF = 0.96)
(VarName142 >= 2.75(-> 2)) and (VarName119 <= 62.5(-> 75)) and (VarName136 >= 17.5(-> 14.4)) and (VarName24 >= 5(-> 4)) and
(VarName106 <= 43.46(-> 43.7)) and (VarName107 <= 61.86(-> 62.27)) and (VarName115 <= 4(-> 5)) => Output=1 (CF = 0.96)
(VarName142 >= 2.75(-> 2)) and (VarName119 <= 62.5(-> 75)) and (VarName111 <= 64.3(-> 67.9)) and (VarName43 <= 1(-> 2)) and
(VarName37 <= 3(-> 4)) and (VarName97 <= 7(-> 8)) => Output=1 (CF = 0.97)
(VarName142 >= 2.75(-> 2)) and (VarName114 <= 83.3(-> 88.9)) and (VarName125 >= 4(-> 3)) and (VarName36 >= 0.25(-> 0)) and (
VarName91 <= 1.5(-> 1.667)) and (VarName104 <= 0(-> 1)) => Output=1 (CF = 0.94)
(VarName142 >= 2.75(-> 2)) and (VarName96 >= 2(-> 1)) and (VarName63 >= 1.667(-> 1)) and (VarName107 <= 54.2(-> 54.61)) and
(VarName97 <= 5(-> 6)) and (VarName73 >= 2(-> 1)) => Output=1 (CF = 0.95)
(VarName142 >= 2.75(-> 2.25)) and (VarName100 >= 0.25(-> 0)) and (VarName74 >= 2(-> 1)) and (VarName107 >= 59.53(-> 59.44))
and (pain1 >= 1(-> 0.25)) and (VarName119 <= 68.8(-> 75)) => Output=1 (CF = 0.95)
(VarName142 >= 2.75(-> 2)) and (VarName30 >= 1.25(-> 1)) and (VarName24 >= 4.5(-> 4)) and (VarName108 >= 4(-> 3)) and (
VarName50 <= 1(-> 2)) and (VarName106 <= 51.28(-> 51.29)) and (VarName38 <= 1(-> 2)) and (VarName70 <= 1(-> 2)) => Output=1
(CF = 0.96)
(VarName142 >= 2.75(-> 2)) and (VarName119 <= 68.8(-> 75)) and (pain6 <= 0(-> 1)) and (VarName35 >= 1(-> 0)) and (
VarName126 >= 5(-> 4)) and (VarName136 <= 25(-> 26)) and (VarName106 <= 54.8(-> 55.09)) and (VarName117 >= 3(-> 2)) =>
Output=1 (CF = 0.97)
(VarName142 >= 2.75(-> 2)) and (VarName111 <= 87.5(-> 89.3)) and (pain8 <= 0.25(-> 1)) and (VarName106 >= 56.11(-> 55.87))
and (VarName112 >= 11(-> 10.6)) and (VarName93 <= 3(-> 4)) => Output=1 (CF = 0.94)
(VarName142 >= 2.75(-> 2)) and (VarName145 >= 12(-> 10)) and (VarName40 >= 2(-> 1)) and (VarName36 >= 0.5(-> 0)) =>
Output=1 (CF = 0.93)
(VarName142 >= 2.75(-> 2)) and (VarName32 <= 0.5(-> 1)) and (pain7 <= 0(-> 1)) and (VarName109 <= 94.4(-> 96.4)) and (
VarName67 >= 1(-> 0)) and (VarName91 <= 1.333(-> 1.5)) and (VarName95 <= 1(-> 2)) => Output=1 (CF = 0.92)
(VarName142 >= 2.75(-> 2)) and (VarName58 >= 1(-> 0)) and (pain6 <= 0(-> 1)) and (VarName35 >= 2(-> 1)) and (VarName107 >=
46.49(-> 42.22)) and (VarName125 <= 5(-> 6)) => Output=1 (CF = 0.95)
(VarName142 >= 5(-> 4)) and (VarName32 <= 0.5(-> 1)) and (VarName119 <= 68.8(-> 75)) and (VarName136 <= 14(-> 15)) and (
VarName36 >= 1(-> 0)) => Output=1 (CF = 0.91)
(VarName142 >= 2.75(-> 2)) and (VarName22 <= 2.25(-> 3)) and (VarName106 >= 57.76(-> 57.49)) and (VarName34 >= 1(-> 0)) and
(VarName108 >= 2(-> 1.3)) => Output=1 (CF = 0.92)

```

Σχήμα 73

```

(VarName142 >= 2.75(-> 2)) and (VarName32 <= 0.5(-> 1)) and (VarName94 >= 1(-> 0)) and (health <= 2(-> 2.75)) and (
VarName112 <= 6(-> 12)) and (VarName114 <= 77.8(-> 84.4)) and (VarName117 <= 14(-> 14.6)) and (VarName58 >= 1(-> 0)) =>
Output=1 (CF = 0.94)
(VarName142 >= 2.75(-> 2.25)) and (VarName100 >= 0.25(-> 0)) and (VarName97 <= 1(-> 2)) and (VarName114 >= 100(-> 97.2))
and (VarName119 <= 68.8(-> 75)) and (health3 >= 5(-> 4)) and (VarName117 <= 1.1(-> 2)) => Output=1 (CF = 0.92)
(VarName142 >= 2.75(-> 2)) and (VarName58 >= 2(-> 1)) and (VarName91 <= 1.333(-> 1.5)) and (VarName131 <= 27(-> 27.1)) =>
Output=1 (CF = 0.86)
(VarName142 >= 2.75(-> 2)) and (VarName73 >= 4(-> 3)) and (VarName68 >= 3(-> 2)) => Output=1 (CF = 0.91)
(VarName142 >= 2.75(-> 2)) and (VarName30 >= 1.25(-> 1)) and (VarName74 >= 2(-> 1)) and (VarName113 <= 4(-> 5)) and (
VarName86 >= 0.333(-> 0.25)) => Output=1 (CF = 0.92)
(VarName142 >= 2.75(-> 2)) and (VarName109 <= 56.3(-> 58.3)) and (VarName47 <= 2(-> 3)) and (VarName91 <= 1.5(-> 2)) =>
Output=1 (CF = 0.94)
(VarName142 >= 3(-> 2)) and (VarName22 <= 2.25(-> 4)) and (VarName106 >= 56.71(-> 56.54)) and (VarName91 >= 2(-> 1.5)) and (
VarName112 <= 1.1(-> 2)) => Output=1 (CF = 0.88)
(VarName142 >= 3(-> 2)) and (VarName91 >= 2.333(-> 2)) and (VarName126 >= 5(-> 4)) and (VarName63 <= 2(-> 3)) and (
VarName28 >= 1(-> 0)) and (VarName39 <= 0(-> 1)) => Output=1 (CF = 0.92)
(pain <= 0(-> 1)) and (VarName106 >= 56.44(-> 56.15)) and (VarName119 >= 100(-> 93.8)) and (pain7 <= 0(-> 1)) and (
VarName107 >= 49.12(-> 49.09)) and (VarName107 <= 54.69(-> 54.74)) => Output=3 (CF = 0.85)
(pain <= 0(-> 1)) and (VarName142 <= 0(-> 1)) and (health <= 1(-> 2)) and (VarName23 >= 5(-> 4)) and (pain7 <= 0(-> 1)) and
(pain6 <= 0(-> 1)) => Output=3 (CF = 0.62)
(VarName142 <= 2.25(-> 2.75)) => Output=2 (CF = 0.93)
(VarName109 >= 80.6(-> 78.1)) and (VarName96 <= 1(-> 2)) => Output=2 (CF = 0.84)
(VarName116 >= 67.9(-> 64.3)) and (VarName127 >= 1(-> 0.25)) => Output=2 (CF = 0.85)

Number of Rules : 28

REGLAS = 28

```

Σχήμα 74

### 4.3. Παρουσίαση με έτοιμα δεδομένα από το Keel και εξαγωγή ασαφών κανόνων

- ❖ Τα datasets προέρχονται από το UCI Machine Learning Repository.
- ❖ Παρακάτω βλέπουμε τους fuzzy rules που παράχθηκαν από τα παραπάνω dataset χρησιμοποιώντας τον αλγόριθμο FURIA-C από την κατηγορία Fuzzy Rule Learning.
- ❖ Στην παρακάτω περίπτωση εκτελέστηκαν τα dataset στην κατηγορία classification και για αυτό το λόγο δημιουργούνται 10 fold για κάθε dataset, όπου για κάθε fold έχουμε και fuzzy rules, όμως εγώ κράτησα τους fuzzy rules από το fold με την καλύτερη απόδοση.

Έπειτα με την εφαρμογή του Keel παράχθηκαν τα εξής αποτελέσματα:

Ας σημειωθεί ότι ο αριθμός των 10 folds που ακολουθεί είναι από 0-9.

## Bupa

Το dataset αυτό αναλύει κάποιες ανωμαλίες που παρουσιάζονται στο συκώτι, από υπερβολική χρήση αλκοόλ. Η έξοδος είναι αν ο ασθενής πάσχει ή όχι, από αλκοολισμό.

Για το σετ δεδομένων του Bupa, το οποίο έχει 345 καταχωρήσεις έχουμε ότι από τα 10 το καλύτερο είναι το 9 με απόδοση 80% και παρακάτω βλέπουμε τους ασαφείς κανόνες που έχουν παραχθεί από το Keel.

```
FURIA rules:
=====

=> selector=1 (CF = 0.0)
=> selector=1 (CF = 0.0)
=> selector=2 (CF = 0.0)
=> selector=2 (CF = 0.0)

Reglas Buenas
Inside m_Ruleset
(gammagt <= 20(-> 21)) and (sgpt >= 20(-> 19)) => selector=1 (CF = 0.72)
(gammagt <= 7(-> 8)) => selector=1 (CF = 0.89)
(alkphos <= 65(-> 66)) and (sgpt <= 21(-> 22)) => selector=2 (CF = 0.81)
(gammagt >= 36(-> 35)) => selector=2 (CF = 0.71)

Number of Rules : 4

REGLAS = 4
```

Σχήμα 75

## Cleveland

Τα δεδομένα που χρησιμοποιούνται σε αυτό το dataset είναι μέρος από το Heart Disease Data Set. Ο στόχος είναι να ανιχνευθεί κατά πόσο έχει καρδιακό πρόβλημα ο ασθενής και η έξοδος κυμαίνεται από 0 έως 4.

Περιέχει 303 καταχωρήσεις και στο συγκεκριμένο τα καλύτερα fold είναι τα 6 και 7 με απόδοση 61%.

Πρώτα βλέπουμε το 6 fold.

```

FURIA rules:
=====
=> num=0 (CF = 0.0)
=> num=0 (CF = 0.0)
(thal >= 7(-> 6)) and (thalach <= 132(-> 136)) => num=3 (CF = 0.42)
(oldpeak >= 2(-> 1.6)) => num=4 (CF = 0.13)
=> num=4 (CF = 0.0)
=> num=4 (CF = 0.0)

Reglas Buenas
Inside m_Ruleset
(thal <= 3(-> 6)) => num=0 (CF = 0.78)
(ca <= 0(-> 1)) and (cp <= 3(-> 4)) => num=0 (CF = 0.86)
(thal >= 7(-> 6)) and (thalach <= 132(-> 136)) and (age <= 58(-> 60)) => num=3 (CF = 0.55)
(oldpeak >= 2(-> 1.6)) and (chol <= 212(-> 226)) and (age >= 60(-> 58)) and (thalach <= 132(-> 134)) => num=4 (CF = 0.68)
(ca >= 3(-> 2)) and (thalach >= 154(-> 147)) => num=4 (CF = 0.52)
(oldpeak >= 3.8(-> 3.6)) and (trestbps <= 120(-> 140)) => num=4 (CF = 0.52)

Number of Rules : 6

REGLAS = 6

```

Σχήμα 76

Και στη συνέχεια το 7 μαζί με τους ασαφείς κανόνες.

```

FURIA rules:
=====
=> num=0 (CF = 0.0)
=> num=0 (CF = 0.0)
=> num=0 (CF = 0.0)
=> num=2 (CF = 0.0)
(ca >= 2(-> 1)) => num=3 (CF = 0.32)
(thalach <= 132(-> 133)) and (trestbps >= 138(-> 136)) => num=4 (CF = 0.24)

Reglas Buenas
Inside m_Ruleset
(cp <= 3(-> 4)) and (age <= 56(-> 57)) and (thal <= 3(-> 6)) and (chol >= 157(-> 149)) => num=0 (CF = 0.99)
(ca <= 0(-> 1)) and (exang <= 0(-> 1)) => num=0 (CF = 0.87)
(cp <= 3(-> 4)) and (sex <= 0(-> 1)) => num=0 (CF = 0.94)
(oldpeak >= 1.9(-> 1.8)) and (thalach <= 150(-> 154)) and (thalach >= 134(-> 133)) => num=2 (CF = 0.58)
(ca >= 2(-> 1)) and (age <= 59(-> 63)) and (oldpeak >= 3(-> 2.8)) => num=3 (CF = 0.76)
(thalach <= 132(-> 133)) and (trestbps >= 138(-> 136)) and (thalach >= 124(-> 123)) => num=4 (CF = 0.56)

Number of Rules : 6

REGLAS = 6

```

Σχήμα 77

## Ecoli

Το αντικείμενο του προβλήματος αυτού είναι να προβλέψουμε αν υπάρχουν κάποιες πρωτεΐνες, το οποίο ανιχνεύεται χρησιμοποιώντας κάποιες μετρήσεις του κυττάρου (κυτόπλασμα, εσωτερική μεμβράνη, περίπλασμα, εξωτερική μεμβράνη, λιποπρωτεΐνη της εξωτερικής και εσωτερικής μεμβράνης, αποσπασίμη αλληλουχία σήματος).

Έχει συνολικά 336 καταχωρήσεις και το καλύτερο fold είναι το 5 με απόδοση 85%.

Η εικόνα μας δείχνει τους ασαφείς κανόνες που παράχθηκαν.

```
FURIA rules:
=====
(a6 <= 38(-> 39)) => class=cp (CF = 0.78)
(a6 <= 52(-> 64)) => class=cp (CF = 0.63)
(a6 <= 47(-> 49)) and (a2 <= 57(-> 59)) => class=cp (CF = 0.84)
(a6 <= 38(-> 39)) => class=cp (CF = 0.78)
=> class=im (CF = 0.0)
=> class=im (CF = 0.0)
=> class=im (CF = 0.0)
=> class=imL (CF = 0.0)
=> class=imU (CF = 0.0)
=> class=imU (CF = 0.0)
=> class=imU (CF = 0.0)
=> class=imU (CF = 0.0)
=> class=imU (CF = 0.0)
=> class=om (CF = 0.0)
=> class=om (CF = 0.0)
=> class=omL (CF = 0.0)
=> class=pp (CF = 0.0)
=> class=pp (CF = 0.0)
=> class=pp (CF = 0.0)
Reglas Buenas
Inside m Ruleset
(a6 <= 38(-> 39)) and (a1 <= 47(-> 48)) => class=cp (CF = 0.95)
(a6 <= 52(-> 64)) and (a1 <= 47(-> 48)) and (a2 <= 49(-> 51)) => class=cp (CF = 0.97)
(a6 <= 47(-> 49)) and (a2 <= 57(-> 59)) and (a1 <= 59(-> 61)) and (a7 <= 46(-> 48)) => class=cp (CF = 0.96)
(a6 <= 38(-> 39)) and (a6 >= 35(-> 31)) and (a2 <= 55(-> 57)) => class=cp (CF = 0.93)
(a6 >= 61(-> 59)) and (a1 <= 46(-> 47)) => class=im (CF = 0.9)
(a7 >= 78(-> 77)) and (a1 <= 58(-> 62)) => class=im (CF = 0.86)
(a6 >= 66(-> 65)) and (a1 <= 73(-> 76)) => class=im (CF = 0.78)
(a3 <= 1(-> 48)) and (a5 <= 51(-> 52)) => class=imL (CF = 0.4)
(a7 >= 62(-> 33)) and (a1 >= 75(-> 74)) and (a1 <= 81(-> 83)) => class=imU (CF = 0.85)
(a6 >= 61(-> 29)) and (a1 >= 58(-> 49)) and (a2 <= 43(-> 44)) and (a2 >= 42(-> 39)) => class=imU (CF = 0.78)
(a6 >= 76(-> 72)) and (a1 >= 79(-> 74)) => class=imU (CF = 0.85)
(a7 >= 66(-> 49)) and (a6 <= 75(-> 76)) and (a1 >= 58(-> 52)) and (a1 <= 67(-> 68)) and (a2 <= 51(-> 52)) => class=imU (CF = 0.78)
(a5 >= 72(-> 71)) and (a1 >= 52(-> 31)) => class=om (CF = 0.87)
(a5 >= 68(-> 64)) and (a2 >= 71(-> 67)) => class=om (CF = 0.84)
(a3 <= 1(-> 48)) and (a1 >= 66(-> 49)) and (a4 >= 5(-> 1)) => class=omL (CF = 0.72)
(a2 >= 59(-> 58)) and (a5 <= 49(-> 51)) and (a6 <= 65(-> 71)) => class=pp (CF = 0.91)
(a2 >= 57(-> 55)) and (a6 <= 52(-> 67)) and (a5 <= 58(-> 63)) and (a5 >= 49(-> 45)) => class=pp (CF = 0.8)
(a2 <= 8(-> 36)) and (a1 >= 61(-> 51)) and (a5 <= 46(-> 49)) => class=pp (CF = 0.76)
Number of Rules : 18
REGLAS = 18
```

Σχήμα 78

## Glass

Η μελέτη για την κατηγορία γυαλιού είχε κίνητρο από την εγκληματολογική έρευνα. Στην σκηνή του εγκλήματος το γυαλί που μένει πίσω μπορεί να χρησιμοποιηθεί σαν στοιχείο. Η έξοδος είναι η κατηγορία.

Περιλαμβάνει 213 καταχωρήσεις και ανάμεσα στα 10 folds το καλύτερο προέκυψε το 7 με απόδοση 85%. Ακολουθούν οι ασαφείς κανόνες.

```
FURIA rules:
=====
(Al <= 1.41029(-> 1.42955)) and (Mg >= 3.43036(-> 3.39893)) => typeGlass=1 (CF = 0.65)
(Al <= 1.3493(-> 1.35893)) => typeGlass=1 (CF = 0.55)
(Al <= 1.29152(-> 1.30115)) => typeGlass=1 (CF = 0.55)
(Mg >= 3.34954(-> 3.25974)) and (RI >= 1.517209(-> 1.517118)) => typeGlass=1 (CF = 0.66)
(Ca <= 8.32444(-> 8.3352)) and (Mg >= 3.43934(-> 3.39893)) => typeGlass=2 (CF = 0.74)
(Na <= 13.25035(-> 13.8688)) and (RI <= 1.517301(-> 1.521196)) => typeGlass=2 (CF = 0.46)
(Fe >= 0.0408(-> 0)) => typeGlass=2 (CF = 0.48)
=> typeGlass=2 (CF = 0.0)
(K >= 0.34776(-> 0.22977)) => typeGlass=2 (CF = 0.45)
=> typeGlass=2 (CF = 0.0)
=> typeGlass=5 (CF = 0.0)
=> typeGlass=5 (CF = 0.0)
=> typeGlass=6 (CF = 0.0)
=> typeGlass=7 (CF = 0.0)

Reglas Buenas
Inside m Ruleset
(Al <= 1.41029(-> 1.42955)) and (Mg >= 3.43036(-> 3.39893)) and (Si >= 72.9516(-> 72.89)) and (Al >= 1.12139(-> 1.07966)) => typeGlass=1 (CF = 0.94)
(Al <= 1.3493(-> 1.35893)) and (Si <= 72.2012(-> 72.2796)) and (RI <= 1.522996(-> 1.523656)) => typeGlass=1 (CF = 0.78)
(Al <= 1.29152(-> 1.30115)) and (Al >= 1.27868(-> 1.26905)) => typeGlass=1 (CF = 0.79)
(Mg >= 3.34954(-> 3.25974)) and (RI >= 1.517209(-> 1.517118)) and (Mg <= 3.61894(-> 3.62792)) and (Ca <= 8.59344(-> 8.6042)) => typeGlass=1 (CF = 0.94)
(Ca <= 8.32444(-> 8.3352)) and (Mg >= 3.43934(-> 3.39893)) and (RI <= 1.517118(-> 1.517551)) and (Si <= 72.9572(-> 72.9684)) => typeGlass=2 (CF = 0.94)
(Na <= 13.25035(-> 13.8688)) and (RI <= 1.517301(-> 1.521196)) and (Si >= 73.0972(-> 73.0916)) and (Na >= 12.8979(-> 12.87795)) => typeGlass=2 (CF = 0.87)
(Fe >= 0.0408(-> 0)) and (RI >= 1.517893(-> 1.517756)) and (Ca <= 8.67952(-> 8.75484)) => typeGlass=2 (CF = 0.88)
(Ca >= 10.56252(-> 10.2452)) and (RI >= 1.521765(-> 1.52172)) => typeGlass=2 (CF = 0.89)
(K >= 0.34776(-> 0.22977)) and (RI >= 1.518121(-> 1.517847)) and (Mg >= 3.61894(-> 3.57853)) => typeGlass=2 (CF = 0.92)
(Na <= 12.991(-> 13.1373)) and (RI <= 1.517301(-> 1.517346)) and (RI >= 1.516549(-> 1.516458)) and (Mg >= 2.72094(-> 0)) => typeGlass=2 (CF = 0.89)
(Mg <= 2.68953(-> 2.81674)) and (Na <= 13.3767(-> 13.5496)) and (Na >= 12.63855(-> 12.6186)) => typeGlass=5 (CF = 0.83)
(Na <= 11.56125(-> 11.94693)) and (RI <= 1.52172(-> 1.52475)) => typeGlass=5 (CF = 0.53)
(K <= 0(-> 0.03105)) and (Ba <= 0(-> 0.53865)) and (RI <= 1.519693(-> 1.521264)) and (Na >= 13.789(-> 13.7225)) => typeGlass=6 (CF = 0.81)
(Ba >= 0.40005(-> 0.2709)) => typeGlass=7 (CF = 0.8)

Number of Rules : 14
```

## Σχήμα 79

## Haberman

Το dataset αυτό περιέχει περιπτώσεις από μια μελέτη, που έγινε μεταξύ του 1958 και 1970 στο Πανεπιστήμιο του Σικάγο, σχετικά με τους ασθενείς που έκαναν επέμβαση για καρκίνο στο στήθος και επεβίωσαν. Η έξοδος εξαρτάται από τον ασθενή αν έζησε παραπάνω από 5 χρόνια μετά την επέμβαση θεωρείται θετική, ενώ αν πέθανε πριν περάσουν 5 χρόνια θεωρείται αρνητική.

Αποτελείται από 306 δείγματα και το καλύτερο fold είναι το 2 με απόδοση 77%. Οι ασαφείς κανόνες φαίνονται παρακάτω.

```
FURIA rules:
=====

=> class=positive (CF = 0.0)
=> class=negative (CF = 0.0)
=> class=negative (CF = 0.0)
=> class=negative (CF = 0.0)

Reglas Buenas
Inside m_Ruleset
(Positive >= 5(-> 4)) and (Age >= 44(-> 43)) => class=positive (CF = 0.58)
(Positive <= 1(-> 2)) => class=negative (CF = 0.85)
(Positive <= 2(-> 3)) => class=negative (CF = 0.84)
(Age <= 43(-> 44)) => class=negative (CF = 0.84)

Number of Rules : 4

REGLAS = 4
```

Σχήμα 80

## Iris

Το συγκεκριμένο dataset θεωρείται το πιο γνωστό για αναγνώριση προτύπων. Περιέχει 3 κλάσεις με 50 δείγματα το καθένα, όπου η κάθε κλάση αναφέρεται στην κατηγορία του φυτού. Η πρώτη κλάση είναι γραμμικά διαχωρισμένη από τις άλλες 2, ενώ η τελευταία δεν είναι με την δεύτερη.

Στο Iris dataset με 150 καταχωρήσεις προέκυψαν καλύτερα fold να είναι το 4, 7, 8 και 9 με απόδοση 100%.

Οι ασαφείς κανόνες ακολουθούν για όλα τα folds.

Πρώτα για το 4.

```

FURIA rules:
=====

(petalLength <= 1.9(-> 3.3)) => class=Iris-setosa (CF = 0.97)
(petalLength >= 3.3(-> 1.9)) and (petalWidth <= 1.6(-> 1.7)) and (petalLength <= 4.9(-> 5)) =>
class=Iris-versicolor (CF = 0.97)
(petalWidth >= 1.7(-> 1.6)) => class=Iris-virginica (CF = 0.93)
(petalLength >= 5(-> 4.9)) => class=Iris-virginica (CF = 0.92)

Reglas Buenas
Inside m_Ruleset
(petalLength <= 1.9(-> 3.3)) => class=Iris-setosa (CF = 0.97)
(petalLength >= 3.3(-> 1.9)) and (petalWidth <= 1.6(-> 1.7)) and (petalLength <= 4.9(-> 5)) =>
class=Iris-versicolor (CF = 0.97)
(petalWidth >= 1.7(-> 1.6)) => class=Iris-virginica (CF = 0.93)
(petalLength >= 5(-> 4.9)) => class=Iris-virginica (CF = 0.92)

Number of Rules : 4

REGLAS = 4

```

Σχήμα 81

Στη συνέχεια για το 7 το οποίο βλέπουμε παράχθηκε ένας παραπάνω κανόνας.

```

FURIA rules:
=====

=> class=Iris-setosa (CF = 0.0)
(petalLength >= 3(-> 1.9)) => class=Iris-versicolor (CF = 0.5)
=> class=Iris-virginica (CF = 0.0)
=> class=Iris-virginica (CF = 0.0)
(petalLength >= 5(-> 4.9)) => class=Iris-virginica (CF = 0.92)

Reglas Buenas
Inside m_Ruleset
(petalLength <= 1.9(-> 3)) => class=Iris-setosa (CF = 0.97)
(petalLength >= 3(-> 1.9)) and (petalLength <= 4.8(-> 5)) and (petalWidth <= 1.6(-> 1.7)) =>
class=Iris-versicolor (CF = 0.97)
(petalWidth >= 1.8(-> 1.7)) => class=Iris-virginica (CF = 0.95)
(petalLength >= 5.5(-> 5.1)) => class=Iris-virginica (CF = 0.95)
(petalLength >= 5(-> 4.9)) and (petalWidth <= 1.5(-> 1.6)) => class=Iris-virginica (CF = 0.73)

Number of Rules : 5

REGLAS = 5

```

Σχήμα 82

Μετά το 8 και το 9 που έβγαλαν τους ίδιους κανόνες.



```
FURIA rules:
=====

(petalLength <= 1.9(-> 3)) => class=Iris-setosa (CF = 0.97)
(petalLength >= 3(-> 1.9)) and (petalWidth <= 1.7(-> 1.8)) and (petalLength <= 5(-> 5.1)) =>
class=Iris-versicolor (CF = 0.93)
(petalWidth >= 1.8(-> 1.7)) => class=Iris-virginica (CF = 0.95)
(petalLength >= 5(-> 4.9)) => class=Iris-virginica (CF = 0.92)

Reglas Buenas
Inside m_Ruleset
(petalLength <= 1.9(-> 3)) => class=Iris-setosa (CF = 0.97)
(petalLength >= 3(-> 1.9)) and (petalWidth <= 1.7(-> 1.8)) and (petalLength <= 5(-> 5.1)) =>
class=Iris-versicolor (CF = 0.93)
(petalWidth >= 1.8(-> 1.7)) => class=Iris-virginica (CF = 0.95)
(petalLength >= 5(-> 4.9)) => class=Iris-virginica (CF = 0.92)

Number of Rules : 4

REGLAS = 4
```

Σχήμα 83

## Monk-2

Το MONK-2 αποτελεί ένα από τα 3 προβλήματα, σχετικά με τεχνητή νοημοσύνη προβλήματα, το οποίο περιέχει με 6 διακριτά χαρακτηριστικά. Κάθε πρόβλημα περιλαμβάνει την εκμάθηση μιας δυαδικής συνάρτησης σε σχέση με την κάθε διακριτή περιοχή κάθε φορά, από ένα δείγμα παραδειγμάτων για εκπαίδευση, που ανήκουν στην κλάση 0 ή 1.

Το dataset Monk-2 με 432 καταχωρήσεις έχει αρκετά folds που πέτυχαν 100% απόδοση. Τα folds είναι 0, 1, 2, 3, 4, 5, 6, 7, 8, 9

Αρχικά βλέπουμε τους ασαφείς κανόνες για το 0 fold.

```

FURIA rules:
=====

(a2 >= 3(-> 2)) and (a4 >= 2(-> 1)) => class=0 (CF = 0.99)
(a5 >= 4(-> 3)) => class=0 (CF = 0.99)
(a2 >= 3(-> 2)) and (a5 <= 2(-> 3)) => class=0 (CF = 0.99)
(a2 <= 2(-> 3)) and (a5 <= 3(-> 4)) => class=1 (CF = 1.0)
(a4 <= 1(-> 2)) and (a5 <= 3(-> 4)) and (a5 >= 3(-> 2)) => class=1 (CF = 0.97)

Reglas Buenas
Inside m_Ruleset
(a2 >= 3(-> 2)) and (a4 >= 2(-> 1)) => class=0 (CF = 0.99)
(a5 >= 4(-> 3)) => class=0 (CF = 0.99)
(a2 >= 3(-> 2)) and (a5 <= 2(-> 3)) => class=0 (CF = 0.99)
(a2 <= 2(-> 3)) and (a5 <= 3(-> 4)) => class=1 (CF = 1.0)
(a4 <= 1(-> 2)) and (a5 <= 3(-> 4)) and (a5 >= 3(-> 2)) => class=1 (CF = 0.97)

Number of Rules : 5

REGLAS = 5

```

Σχήμα 84

Για το 1, 3, 4, 5, 6, 7, 8 fold έχουμε τους παρακάτω ασαφείς κανόνες.

```

FURIA rules:
=====

(a2 >= 3(-> 2)) and (a4 >= 2(-> 1)) => class=0 (CF = 0.99)
(a5 >= 4(-> 3)) => class=0 (CF = 0.99)
(a2 >= 3(-> 2)) and (a5 <= 2(-> 3)) => class=0 (CF = 0.98)
(a2 <= 2(-> 3)) and (a5 <= 3(-> 4)) => class=1 (CF = 1.0)
(a4 <= 1(-> 2)) and (a5 <= 3(-> 4)) and (a5 >= 3(-> 2)) => class=1 (CF = 0.97)

Reglas Buenas
Inside m_Ruleset
(a2 >= 3(-> 2)) and (a4 >= 2(-> 1)) => class=0 (CF = 0.99)
(a5 >= 4(-> 3)) => class=0 (CF = 0.99)
(a2 >= 3(-> 2)) and (a5 <= 2(-> 3)) => class=0 (CF = 0.98)
(a2 <= 2(-> 3)) and (a5 <= 3(-> 4)) => class=1 (CF = 1.0)
(a4 <= 1(-> 2)) and (a5 <= 3(-> 4)) and (a5 >= 3(-> 2)) => class=1 (CF = 0.97)

Number of Rules : 5

REGLAS = 5

```

Σχήμα 85

Για το 2 fold προκύπτουν τα εξής, τα οποία δεν συμβαδίζουν με τα παραπάνω που έβγαλαν κανόνες ακριβώς ίδιους.

```
FURIA rules:
=====

(a2 >= 3(-> 2)) and (a4 >= 2(-> 1)) => class=0 (CF = 0.99)
(a5 >= 4(-> 3)) => class=0 (CF = 0.99)
(a2 >= 3(-> 2)) and (a5 <= 2(-> 3)) => class=0 (CF = 0.98)
(a2 <= 2(-> 3)) and (a5 <= 3(-> 4)) => class=1 (CF = 0.99)
(a4 <= 1(-> 2)) and (a5 <= 3(-> 4)) and (a5 >= 3(-> 2)) => class=1 (CF = 0.97)

Reglas Buenas
Inside m_Ruleset
(a2 >= 3(-> 2)) and (a4 >= 2(-> 1)) => class=0 (CF = 0.99)
(a5 >= 4(-> 3)) => class=0 (CF = 0.99)
(a2 >= 3(-> 2)) and (a5 <= 2(-> 3)) => class=0 (CF = 0.98)
(a2 <= 2(-> 3)) and (a5 <= 3(-> 4)) => class=1 (CF = 0.99)
(a4 <= 1(-> 2)) and (a5 <= 3(-> 4)) and (a5 >= 3(-> 2)) => class=1 (CF = 0.97)

Number of Rules : 5

REGLAS = 5
```

Σχήμα 86

Και επίσης το 9 fold προέκυψε με διαφορετικούς κανόνες.

```
FURIA rules:
=====

(a2 >= 3(-> 2)) and (a4 >= 2(-> 1)) => class=0 (CF = 0.99)
(a5 >= 4(-> 3)) => class=0 (CF = 0.99)
(a2 >= 3(-> 2)) and (a5 <= 2(-> 3)) => class=0 (CF = 0.99)
(a2 <= 2(-> 3)) and (a5 <= 3(-> 4)) => class=1 (CF = 1.0)
(a4 <= 1(-> 2)) and (a5 <= 3(-> 4)) and (a5 >= 3(-> 2)) => class=1 (CF = 0.97)

Reglas Buenas
Inside m_Ruleset
(a2 >= 3(-> 2)) and (a4 >= 2(-> 1)) => class=0 (CF = 0.99)
(a5 >= 4(-> 3)) => class=0 (CF = 0.99)
(a2 >= 3(-> 2)) and (a5 <= 2(-> 3)) => class=0 (CF = 0.99)
(a2 <= 2(-> 3)) and (a5 <= 3(-> 4)) => class=1 (CF = 1.0)
(a4 <= 1(-> 2)) and (a5 <= 3(-> 4)) and (a5 >= 3(-> 2)) => class=1 (CF = 0.97)

Number of Rules : 5

REGLAS = 5
```

Σχήμα 87

## New-Thyroid

Το dataset αυτό έχει σαν στόχο την ανίχνευση της κατάστασης του ασθενή αν πάσχει από υπερθυρεοειδισμό ή υποθυρεοειδισμός ή αν είναι φυσιολογικός. Έχει 215 καταχωρήσεις, όπου προέκυψε ότι τα καλύτερα fold είναι το 1 και 3 με απόδοση 95%.

Για το fold 1 προκύπτουν 7 ασαφείς κανόνες.

```
FURIA rules:
=====
(thyroxin <= 12.4(-> 12.9)) and (thyroxin >= 5.7(-> 5.1)) and (TSH_value <= 7.7(-> 11.5)) =>
class=normal (CF = 0.97)
(triiodothyronine <= 2.5(-> 3)) and (thyroxin >= 7.3(-> 6.8)) and (thyroxin <= 13.8(-> 15.2)) =>
class=normal (CF = 0.97)
(thyroidstimulating <= 1.6(-> 1.7)) => class=normal (CF = 0.78)
=> class=hyper (CF = 0.0)
=> class=hyper (CF = 0.0)
=> class=hypo (CF = 0.0)
=> class=hypo (CF = 0.0)
Reglas Buenas
Inside m_Ruleset
(thyroxin <= 12.4(-> 12.9)) and (thyroxin >= 5.7(-> 5.1)) and (TSH_value <= 7.7(-> 11.5)) and (TSH_value
>= 0.1(-> 0)) => class=normal (CF = 0.99)
(triiodothyronine <= 2.5(-> 3)) and (thyroxin >= 7.3(-> 6.8)) and (thyroxin <= 13.8(-> 15.2)) and (
T3resin >= 98(-> 95)) => class=normal (CF = 0.99)
(thyroidstimulating <= 1.6(-> 1.7)) and (thyroidstimulating >= 1.3(-> 1.2)) and (T3resin >= 103(-> 97))
and (T3resin <= 116(-> 144)) => class=normal (CF = 0.99)
(thyroxin >= 12.9(-> 11.3)) and (TSH_value <= 0.4(-> 0.5)) => class=hyper (CF = 0.94)
(triiodothyronine >= 2.7(-> 2.5)) and (thyroxin >= 11.1(-> 9.5)) and (TSH_value <= 0.6(-> 3.5)) =>
class=hyper (CF = 0.94)
(thyroxin <= 5.6(-> 5.7)) => class=hypo (CF = 0.87)
(TSH_value >= 11.5(-> 7)) and (thyroxin <= 6.8(-> 8.1)) => class=hypo (CF = 0.91)

Number of Rules : 7
REGLAS = 7
```

Σχήμα 88

Και για το 3 fold επίσης παράγονται 7 ασαφείς κανόνες αλλά με διαφορετικές τιμές.

```
FURIA rules:
=====
(thyroxin <= 12.4(-> 12.9)) and (thyroxin >= 5.7(-> 5.3)) and (TSH_value <= 7(-> 11.5)) => class=normal (
CF = 0.98)
(triiodothyronine <= 2.5(-> 3)) and (thyroxin >= 7.3(-> 6.8)) and (thyroxin <= 13.8(-> 16)) =>
class=normal (CF = 0.98)
=> class=normal (CF = 0.0)
=> class=hyper (CF = 0.0)
=> class=hyper (CF = 0.0)
=> class=hypo (CF = 0.0)
=> class=hypo (CF = 0.0)
Reglas Buenas
Inside m_Ruleset
(thyroxin <= 12.4(-> 12.9)) and (thyroxin >= 5.7(-> 5.3)) and (TSH_value <= 7(-> 11.5)) and (TSH_value
>= 0.1(-> 0)) => class=normal (CF = 0.99)
(triiodothyronine <= 2.5(-> 3)) and (thyroxin >= 7.3(-> 6.8)) and (thyroxin <= 13.8(-> 16)) and (T3resin
>= 98(-> 97)) => class=normal (CF = 0.99)
(TSH_value >= 1.4(-> 0.6)) and (TSH_value <= 2(-> 5)) and (thyroxin >= 4.2(-> 3.5)) => class=normal (CF
= 0.99)
(thyroxin >= 12.9(-> 11.3)) and (TSH_value <= 0.4(-> 0.5)) => class=hyper (CF = 0.94)
(triiodothyronine >= 3.3(-> 2.9)) => class=hyper (CF = 0.93)
(thyroxin <= 5.6(-> 5.7)) => class=hypo (CF = 0.87)
(TSH_value >= 11.5(-> 7)) => class=hypo (CF = 0.85)

Number of Rules : 7
```

Σχήμα 89

## Pima

Σε αυτή την περίπτωση η έξοδος καθορίζει αν ο ασθενής έχει διαβήτη ή όχι.

Στο dataset Pima με 768 καταχωρήσεις προέκυψε ότι το καλύτερο fold είναι το 4 με απόδοση 79% και δημιουργήθηκαν 6 ασαφείς κανόνες.

```
FURIA rules:
=====

('plas' <= 127(-> 128)) => 'class'=tested_negative (CF = 0.8)
=> 'class'=tested_negative (CF = 0.0)
=> 'class'=tested_negative (CF = 0.0)
('plas' <= 154(-> 155)) => 'class'=tested_negative (CF = 0.73)
=> 'class'=tested_positive (CF = 0.0)
('age' >= 31(-> 30)) => 'class'=tested_positive (CF = 0.51)

Reglas Buenas
Inside m_Ruleset
('plas' <= 127(-> 128)) and ('age' <= 29(-> 30)) => 'class'=tested_negative (CF = 0.9)
('mass' <= 27.7(-> 27.9)) => 'class'=tested_negative (CF = 0.87)
('plas' <= 106(-> 107)) => 'class'=tested_negative (CF = 0.87)
('plas' <= 154(-> 155)) and ('age' <= 30(-> 31)) and ('pres' >= 74(-> 72)) => 'class'=tested_negative (
CF = 0.85)
('plas' >= 155(-> 154)) => 'class'=tested_positive (CF = 0.81)
('age' >= 31(-> 30)) and ('mass' >= 30.3(-> 29.9)) and ('plas' >= 108(-> 107)) =>
'class'=tested_positive (CF = 0.73)

Number of Rules : 6

REGLAS = 6
```

Σχήμα 90

## Vehicle

Ο στόχος αυτού του dataset είναι δοθέντος μιας σιλουέτας αυτοκινήτου να κατηγοριοποιηθεί σε ποια κλάση ανήκει από τις τέσσερις, σύμφωνα με τα χαρακτηριστικά που εξάγονται από την σιλουέτα, η οποία πρέπει να περιστραφεί για να αποκτήσουμε παραπάνω χαρακτηριστικά.

Το dataset Vehicle έχει 846 καταχωρήσεις, όπου το καλύτερο fold είναι το 5 με απόδοση 77% και προέκυψαν 29 ασαφείς κανόνες, οι οποίοι διαιρούνται σε δύο εικόνες, λόγω της έκτασης των κανόνων.

```

FURIA rules:
=====

=> class=van (CF = 0.0)
=> class=van (CF = 0.0)
=> class=van (CF = 0.0)
=> class=van (CF = 0.0)
=> class=van (CF = 0.0)
=> class=van (CF = 0.0)
=> class=van (CF = 0.0)
=> class=van (CF = 0.0)
=> class=van (CF = 0.0)
(SCATTERRATIO >= 168(-> 167)) => class=saab (CF = 0.42)
(SCATTERRATIO >= 168(-> 167)) => class=saab (CF = 0.42)
=> class=saab (CF = 0.0)
=> class=saab (CF = 0.0)
=> class=saab (CF = 0.0)
=> class=saab (CF = 0.0)
=> class=bus (CF = 0.0)
=> class=bus (CF = 0.0)
=> class=bus (CF = 0.0)
=> class=bus (CF = 0.0)
=> class=bus (CF = 0.0)
=> class=bus (CF = 0.0)
=> class=bus (CF = 0.0)
=> class=bus (CF = 0.0)
(DISTANCECIRCULARITY >= 91(-> 90)) => class=opel (CF = 0.43)
(MINORVARIANCE >= 589(-> 587)) and (MAXLENGTHASPECTRATIO >= 8(-> 7)) => class=opel (CF = 0.51)
(MINORVARIANCE >= 388(-> 367)) => class=opel (CF = 0.38)
=> class=opel (CF = 0.0)
(MAJORVARIANCE >= 181(-> 180)) and (MAXLENGTHASPECTRATIO >= 8(-> 7)) and (HOLLOWSRATIO <= 198(-> 199)) => class=opel (CF = 0.59)
=> class=opel (CF = 0.0)

```

Σχήμα 91

```

Reglas Buenas
Inside m Ruleset
(MAJORVARIANCE <= 176(-> 180)) and (MAXLENGTHASPECTRATIO >= 9(-> 8)) and (LENGTHRECTANGULAR >= 140(-> 139)) => class=van (CF = 0.98)
(MINORVARIANCE <= 311(-> 312)) and (LENGTHRECTANGULAR >= 139(-> 138)) and (SCATTERRATIO <= 139(-> 140)) => class=van (CF = 0.96)
(SCATTERRATIO <= 146(-> 147)) and (MAXLENGTHASPECTRATIO >= 8(-> 7)) and (PRAXISASPECTRATIO >= 62(-> 61)) => class=van (CF = 0.96)
(SCATTERRATIO <= 133(-> 140)) and (LENGTHRECTANGULAR >= 128(-> 127)) and (GYRATIONRADIUS <= 131(-> 132)) => class=van (CF = 0.94)
(MINORVARIANCE <= 223(-> 227)) and (CIRCULARITY >= 37(-> 36)) and (MINORSKEWNESS <= 17(-> 20)) => class=van (CF = 0.9)
(ELONGATEDNESS >= 42(-> 38)) and (LENGTHRECTANGULAR >= 155(-> 153)) => class=van (CF = 0.97)
(PRAXISRECTANGULAR <= 18(-> 19)) and (SCATTERRATIO <= 119(-> 120)) and (ELONGATEDNESS <= 57(-> 58)) => class=van (CF = 0.89)
(PRAXISRECTANGULAR <= 18(-> 19)) and (MAJORVARIANCE >= 166(-> 165)) and (COMPACTNESS >= 86(-> 84)) => class=van (CF = 0.82)
(ELONGATEDNESS >= 49(-> 46)) and (LENGTHRECTANGULAR >= 133(-> 131)) and (RADIUSRATIO <= 124(-> 127)) => class=van (CF = 0.92)
(SCATTERRATIO >= 168(-> 167)) and (HOLLOWSRATIO >= 197(-> 196)) and (COMPACTNESS >= 107(-> 106)) and (MAJORSKEWNESS >= 70(-> 69)) => class=saab (CF = 0.94)
(SCATTERRATIO >= 168(-> 167)) and (HOLLOWSRATIO >= 199(-> 198)) and (MAJORVARIANCE >= 223(-> 222)) => class=saab (CF = 0.83)
(MAJORSKEWNESS <= 68(-> 69)) and (MAJORVARIANCE >= 177(-> 175)) and (PRAXISASPECTRATIO <= 61(-> 62)) and (HOLLOWSRATIO >= 205(-> 204)) => class=saab (CF = 0.75)
(LENGTHRECTANGULAR <= 136(-> 137)) and (GYRATIONRADIUS >= 149(-> 148)) => class=saab (CF = 0.64)
(LENGTHRECTANGULAR <= 132(-> 134)) and (MAJORSKEWNESS <= 61(-> 62)) => class=saab (CF = 0.75)
(LENGTHRECTANGULAR <= 130(-> 132)) and (RADIUSRATIO <= 136(-> 140)) and (RADIUSRATIO >= 130(-> 129)) and (CIRCULARITY <= 37(-> 38)) => class=saab (CF = 0.85)
(MAXLENGTHASPECTRATIO <= 7(-> 8)) and (MINORVARIANCE >= 317(-> 306)) and (PRAXISASPECTRATIO >= 67(-> 66)) => class=bus (CF = 0.98)
(HOLLOWSRATIO <= 189(-> 192)) and (CIRCULARITY >= 42(-> 41)) and (MAJORSKEWNESS >= 80(-> 79)) => class=bus (CF = 0.98)
(MAXLENGTHASPECTRATIO <= 7(-> 8)) and (PRAXISASPECTRATIO >= 63(-> 62)) and (COMPACTNESS <= 88(-> 89)) and (CIRCULARITY >= 39(-> 37)) => class=bus (CF = 0.96)
(MAJORSKEWNESS >= 75(-> 74)) and (PRAXISASPECTRATIO >= 61(-> 59)) and (HOLLOWSRATIO <= 192(-> 193)) and (MAJORVARIANCE >= 169(-> 165)) and (RADIUSRATIO <= 208(-> 246)) => class=bus (CF = 0.96)
(MAXLENGTHASPECTRATIO <= 8(-> 9)) and (CIRCULARITY >= 42(-> 41)) and (MAJORSKEWNESS >= 74(-> 73)) and (SCATTERRATIO >= 147(-> 144)) => class=bus (CF = 0.97)
(DISTANCECIRCULARITY <= 75(-> 77)) and (PRAXISASPECTRATIO >= 64(-> 63)) and (MINORVARIANCE >= 310(-> 307)) => class=bus (CF = 0.95)
(MAXLENGTHASPECTRATIO <= 7(-> 8)) and (ELONGATEDNESS <= 37(-> 38)) and (COMPACTNESS >= 93(-> 90)) => class=bus (CF = 0.97)
(DISTANCECIRCULARITY <= 70(-> 71)) and (MINORVARIANCE >= 299(-> 296)) and (DISTANCECIRCULARITY >= 70(-> 69)) => class=bus (CF = 0.95)
(DISTANCECIRCULARITY >= 91(-> 90)) and (COMPACTNESS <= 103(-> 104)) and (LENGTHRECTANGULAR >= 173(-> 171)) => class=opel (CF = 0.92)
(MINORVARIANCE >= 589(-> 587)) and (MAXLENGTHASPECTRATIO >= 8(-> 7)) and (COMPACTNESS <= 103(-> 104)) and (MINORSKEWNESS >= 4(-> 3)) and (MINORSKEWNESS <= 7(-> 8)) => class=opel (CF = 0.94)
(MINORVARIANCE >= 388(-> 367)) and (MAJORSKEWNESS <= 67(-> 68)) and (HOLLOWSRATIO <= 203(-> 204)) and (MINORKURTOSIS >= 9(-> 8)) => class=opel (CF = 0.75)
(LENGTHRECTANGULAR <= 135(-> 137)) and (COMPACTNESS <= 84(-> 85)) and (CIRCULARITY >= 38(-> 37)) => class=opel (CF = 0.77)
(MAJORVARIANCE >= 181(-> 180)) and (MAXLENGTHASPECTRATIO >= 8(-> 7)) and (HOLLOWSRATIO <= 198(-> 199)) and (MINORVARIANCE >= 706(-> 705)) => class=opel (CF = 0.87)
(LENGTHRECTANGULAR <= 126(-> 127)) and (MAXLENGTHASPECTRATIO >= 7(-> 6)) and (MINORSKEWNESS >= 2(-> 1)) and (MINORSKEWNESS <= 6(-> 8)) => class=opel (CF = 0.85)
Number of Rules : 29

```

Σχήμα 92

## Wine

Τα δεδομένα αυτά προέκυψαν ύστερα από την ανάλυση κρασιών τα οποία παράχθηκαν στην ίδια περιοχή στην Ιταλία, αλλά συμπληρώθηκαν από τρεις διαφορετικές καλλιέργειες. Η ανάλυση καθόρισε τις ποσότητες των 13 συστατικών ξεχωριστά στους 3 τύπους κρασιών.

Έπονται οι κανόνες από το Wine dataset, που έχει 178 καταχωρήσεις, όπου τα καλύτερα fold με 100% είναι 0,1, 2, 3, 8

Στο fold 0 προέκυψαν οι ακόλουθοι 6 ασαφείς κανόνες.

```
FURIA rules:
=====
=> class=1 (CF = 0.0)
(at7 >= 2.37(-> 2.27)) => class=1 (CF = 0.73)
=> class=2 (CF = 0.0)
=> class=2 (CF = 0.0)
=> class=2 (CF = 0.0)
=> class=3 (CF = 0.0)

Reglas Buenas
Inside m_Ruleset
(at13 >= 845(-> 840)) and (at1 >= 13.05(-> 12.99)) => class=1 (CF = 0.97)
(at7 >= 2.37(-> 2.27)) and (at13 >= 735(-> 714)) and (at4 <= 25(-> 30)) => class=1 (CF = 0.98)
(at10 <= 3.8(-> 3.85)) and (at1 <= 12.99(-> 13.05)) => class=2 (CF = 0.98)
(at2 <= 1.53(-> 1.6)) and (at10 <= 4.68(-> 5)) and (at5 <= 110(-> 122)) => class=2 (CF = 0.96)
(at13 <= 465(-> 470)) and (at1 <= 13.05(-> 13.11)) => class=2 (CF = 0.96)
(at7 <= 1.39(-> 1.41)) and (at10 >= 3.85(-> 3.6)) => class=3 (CF = 0.97)

Number of Rules : 6

REGLAS = 6
```

Σχήμα 93

Στο επόμενο fold 1 προέκυψαν 7 ασαφείς κανόνες.

```

FURIA rules:
=====

=> class=1 (CF = 0.0)
(at1 >= 13.24(-> 13.17)) => class=1 (CF = 0.64)
=> class=2 (CF = 0.0)
=> class=2 (CF = 0.0)
=> class=2 (CF = 0.0)
=> class=2 (CF = 0.0)
=> class=3 (CF = 0.0)

Reglas Buenas
Inside m_Ruleset
(at13 >= 760(-> 714)) and (at7 >= 2.33(-> 2.27)) and (at4 <= 25(-> 30)) => class=1 (CF = 0.97)
(at1 >= 13.24(-> 13.17)) and (at1 <= 13.3(-> 13.32)) and (at2 <= 3.98(-> 4.28)) => class=1 (CF = 0.79)
(at10 <= 3.8(-> 3.85)) and (at13 <= 750(-> 830)) => class=2 (CF = 0.98)
(at1 <= 12.72(-> 12.85)) and (at7 >= 1.41(-> 1.36)) => class=2 (CF = 0.98)
(at10 <= 3.35(-> 3.52)) => class=2 (CF = 0.98)
(at5 <= 85(-> 90)) and (at7 >= 1.59(-> 1.1)) => class=2 (CF = 0.96)
(at7 <= 1.39(-> 1.41)) and (at10 >= 3.85(-> 3.6)) => class=3 (CF = 0.97)

Number of Rules : 7

REGLAS = 7

```

Σχήμα 94

Στο 2 fold δημιουργήθηκαν 6 ασαφείς κανόνες.

```

FURIA rules:
=====

=> class=1 (CF = 0.0)
=> class=2 (CF = 0.0)
=> class=2 (CF = 0.0)
=> class=2 (CF = 0.0)
=> class=2 (CF = 0.0)
=> class=3 (CF = 0.0)

Reglas Buenas
Inside m_Ruleset
(at13 >= 760(-> 714)) and (at7 >= 2.19(-> 2.14)) => class=1 (CF = 0.96)
(at10 <= 3.8(-> 3.85)) and (at13 <= 750(-> 830)) => class=2 (CF = 0.98)
(at1 <= 12.72(-> 12.85)) and (at7 >= 1.41(-> 1.36)) => class=2 (CF = 0.98)
(at5 <= 86(-> 88)) and (at1 >= 13.03(-> 12.87)) => class=2 (CF = 0.84)
(at10 <= 3.35(-> 3.52)) => class=2 (CF = 0.98)
(at7 <= 1.39(-> 1.41)) and (at10 >= 3.85(-> 3.6)) => class=3 (CF = 0.97)

Number of Rules : 6

REGLAS = 6

```

Σχήμα 95



Στο 3 fold έχουμε επίσης 6 ασαφείς κανόνες.

```
FURIA rules:
=====

(at13 >= 760(-> 714)) and (at7 >= 2.33(-> 2.27)) => class=1 (CF = 0.96)
=> class=1 (CF = 0.0)
(at10 <= 3.94(-> 4.1)) => class=2 (CF = 0.87)
=> class=2 (CF = 0.0)
=> class=2 (CF = 0.0)
=> class=3 (CF = 0.0)

Reglas Buenas
Inside m_Ruleset
(at13 >= 760(-> 714)) and (at7 >= 2.33(-> 2.27)) and (at4 <= 25(-> 30)) => class=1 (CF = 0.97)
(at1 >= 13.24(-> 13.23)) and (at1 <= 13.3(-> 13.32)) and (at2 <= 3.98(-> 4.28)) => class=1 (CF = 0.81)
(at10 <= 3.94(-> 4.1)) and (at13 <= 718(-> 720)) => class=2 (CF = 0.98)
(at2 <= 1.21(-> 1.24)) => class=2 (CF = 0.93)
(at1 <= 12.64(-> 12.85)) and (at7 >= 1.41(-> 1.36)) => class=2 (CF = 0.97)
(at7 <= 1.39(-> 1.41)) and (at10 >= 3.85(-> 3.6)) => class=3 (CF = 0.97)

Number of Rules : 6

REGLAS = 6
```

Σχήμα 96

Στο 8 fold έχουμε 7 ασαφείς κανόνες.

```
FURIA rules:
=====

(at13 >= 760(-> 714)) => class=1 (CF = 0.77)
=> class=2 (CF = 0.0)
=> class=2 (CF = 0.0)
=> class=2 (CF = 0.0)
=> class=2 (CF = 0.0)
=> class=3 (CF = 0.0)
=> class=3 (CF = 0.0)

Reglas Buenas
Inside m_Ruleset
(at13 >= 760(-> 714)) and (at7 >= 2.37(-> 2.27)) => class=1 (CF = 0.96)
(at10 <= 3.8(-> 3.85)) and (at1 <= 12.99(-> 13.07)) => class=2 (CF = 0.98)
(at5 <= 87(-> 89)) and (at7 >= 1.59(-> 0.92)) => class=2 (CF = 0.96)
(at2 <= 1.25(-> 1.29)) => class=2 (CF = 0.94)
(at13 <= 465(-> 510)) and (at1 <= 12.64(-> 12.77)) => class=2 (CF = 0.96)
(at12 <= 2.11(-> 2.12)) and (at11 <= 0.82(-> 0.84)) => class=3 (CF = 0.96)
(at7 <= 0.92(-> 1.3)) and (at1 >= 12.85(-> 12.77)) => class=3 (CF = 0.95)

Number of Rules : 7

REGLAS = 7
```

Σχήμα 97

## Wisconsin

Το dataset αυτό περιέχει περιπτώσεις, από μια μελέτη, η οποία έγινε στο Πανεπιστήμιο του Wisconsin, σχετικά με ασθενείς που έκαναν επέμβαση για καρκίνο του στήθους. Η έξοδος του dataset είναι να καθοριστεί αν ο όγκος είναι καλοήθης ή κακοήθης.

Τα δεδομένα του Wisconsin με τις 669 καταχωρήσεις προκύπτουν τα καλύτερα fold να είναι το 2 και 6 με απόδοση 98%.

Στο 2 fold προκύπτουν 17 ασαφείς κανόνες, που είναι λογικό αν αναλογιστούμε το πλήθος των δεδομένων.

```
FURIA rules:
=====
=> class=2 (CF = 0.0)
=> class=2 (CF = 0.0)
=> class=2 (CF = 0.0)
(cellSize <= 4(-> 5)) => class=2 (CF = 0.86)
=> class=2 (CF = 0.0)
(blandChromatin <= 3(-> 4)) and (clumpThickness <= 6(-> 8)) => class=2 (CF = 0.96)
(cellSize <= 4(-> 10)) => class=2 (CF = 0.77)
(cellSize <= 4(-> 5)) => class=2 (CF = 0.86)
=> class=4 (CF = 0.0)
(bareNuclei >= 8(-> 7)) => class=4 (CF = 0.97)
=> class=4 (CF = 0.0)
(cellSize >= 3(-> 2)) and (marginalAdhesion >= 4(-> 3)) => class=4 (CF = 0.94)
=> class=4 (CF = 0.0)
(cellSize >= 3(-> 2)) and (marginalAdhesion >= 3(-> 2)) => class=4 (CF = 0.93)
(epithelialSize >= 3(-> 2)) and (blandChromatin >= 3(-> 1)) => class=4 (CF = 0.88)
(bareNuclei >= 6(-> 5)) => class=4 (CF = 0.97)
(bareNuclei >= 3(-> 2)) => class=4 (CF = 0.86)
```

Σχήμα 98

```

Reglas Buenas
Inside m_Ruleset
(cellSize <= 2(-> 3)) and (bareNuclei <= 2(-> 4)) and (clumpThickness <= 7(-> 8)) => class=2 (CF = 1.0)
(cellShape <= 2(-> 3)) and (clumpThickness <= 5(-> 6)) and (marginalAdhesion <= 2(-> 3)) => class=2 (CF = 1.0)
(bareNuclei <= 4(-> 5)) and (cellSize <= 4(-> 6)) and (clumpThickness <= 5(-> 6)) and (normalNucleoli <= 2(-> 3)) =>
class=2 (CF = 1.0)
(cellSize <= 4(-> 5)) and (bareNuclei <= 7(-> 8)) and (epithelialSize >= 4(-> 3)) and (bareNuclei >= 5(-> 3)) and (
normalNucleoli <= 8(-> 10)) => class=2 (CF = 0.89)
(bareNuclei <= 0(-> 1)) and (cellSize <= 6(-> 7)) and (clumpThickness <= 6(-> 8)) => class=2 (CF = 0.95)
(blandChromatin <= 3(-> 4)) and (clumpThickness <= 6(-> 8)) and (marginalAdhesion <= 1(-> 2)) and (cellSize >= 4(->
3)) => class=2 (CF = 0.9)
(cellSize <= 4(-> 10)) and (bareNuclei <= 3(-> 4)) and (clumpThickness <= 4(-> 5)) and (marginalAdhesion <= 6(->
10)) => class=2 (CF = 1.0)
(cellSize <= 4(-> 5)) and (bareNuclei <= 2(-> 3)) and (normalNucleoli <= 2(-> 3)) => class=2 (CF = 1.0)
(cellSize >= 5(-> 4)) and (clumpThickness >= 7(-> 6)) => class=4 (CF = 0.99)
(bareNuclei >= 8(-> 7)) and (blandChromatin >= 4(-> 3)) => class=4 (CF = 0.99)
(normalNucleoli >= 9(-> 8)) and (bareNuclei >= 2(-> 0)) => class=4 (CF = 0.98)
(cellSize >= 3(-> 2)) and (marginalAdhesion >= 4(-> 3)) and (cellShape >= 5(-> 4)) and (bareNuclei >= 1(-> 0)) =>
class=4 (CF = 0.99)
(normalNucleoli >= 3(-> 2)) and (clumpThickness >= 7(-> 6)) and (marginalAdhesion <= 1(-> 3)) => class=4 (CF = 0.95)
(cellSize >= 3(-> 2)) and (marginalAdhesion >= 3(-> 2)) and (epithelialSize <= 2(-> 3)) and (cellShape >= 3(-> 2))
=> class=4 (CF = 0.91)
(epithelialSize >= 3(-> 2)) and (blandChromatin >= 3(-> 1)) and (mitoses >= 3(-> 1)) and (bareNuclei <= 3(-> 7)) =>
class=4 (CF = 0.94)
(bareNuclei >= 6(-> 5)) and (marginalAdhesion <= 3(-> 5)) and (cellSize <= 6(-> 7)) => class=4 (CF = 0.97)
(bareNuclei >= 3(-> 2)) and (clumpThickness >= 6(-> 5)) and (normalNucleoli <= 3(-> 5)) => class=4 (CF = 0.97)

Number of Rules : 17

REGLAS = 17

```

### Σχήμα 99

Στο 6 fold οι ασαφείς κανόνες φτάνουν τον αριθμό 13.

```

FURIA rules:
=====
=> class=2 (CF = 0.0)
(bareNuclei <= 4(-> 5)) and (cellSize <= 4(-> 6)) => class=2 (CF = 0.96)
=> class=2 (CF = 0.0)
=> class=2 (CF = 0.0)
=> class=2 (CF = 0.0)
(cellSize <= 4(-> 5)) => class=2 (CF = 0.87)
=> class=4 (CF = 0.0)
=> class=4 (CF = 0.0)
=> class=4 (CF = 0.0)
=> class=4 (CF = 0.0)
=> class=4 (CF = 0.0)
=> class=4 (CF = 0.0)
=> class=4 (CF = 0.0)
=> class=4 (CF = 0.0)
Reglas Buenas
Inside m_Ruleset
(cellShape <= 2(-> 3)) and (clumpThickness <= 6(-> 7)) => class=2 (CF = 0.99)
(bareNuclei <= 4(-> 5)) and (cellSize <= 4(-> 6)) and (normalNucleoli <= 2(-> 3)) and (clumpThickness <= 5(-> 6)) => class=2 (CF =
1.0)
(cellShape <= 3(-> 4)) and (bareNuclei <= 2(-> 3)) => class=2 (CF = 1.0)
(bareNuclei <= 0(-> 1)) and (clumpThickness <= 6(-> 8)) => class=2 (CF = 0.96)
(blandChromatin <= 4(-> 5)) and (clumpThickness <= 6(-> 7)) and (clumpThickness >= 6(-> 5)) and (bareNuclei >= 4(-> 2)) => class=2
(CF = 0.72)
(cellSize <= 4(-> 5)) and (epithelialSize >= 7(-> 6)) and (cellSize >= 4(-> 3)) and (cellShape <= 5(-> 6)) => class=2 (CF = 0.86)
(cellSize >= 5(-> 4)) and (blandChromatin >= 5(-> 4)) => class=4 (CF = 0.98)
(bareNuclei >= 6(-> 5)) and (marginalAdhesion >= 6(-> 5)) => class=4 (CF = 0.98)
(cellShape >= 5(-> 4)) and (blandChromatin >= 5(-> 4)) => class=4 (CF = 0.98)
(cellSize >= 4(-> 3)) and (clumpThickness >= 7(-> 6)) => class=4 (CF = 0.97)
(bareNuclei >= 3(-> 2)) and (cellShape >= 3(-> 2)) and (cellSize <= 3(-> 10)) => class=4 (CF = 0.91)
(bareNuclei >= 4(-> 3)) and (blandChromatin >= 4(-> 3)) => class=4 (CF = 0.95)
(cellSize >= 4(-> 3)) and (marginalAdhesion >= 6(-> 5)) and (bareNuclei >= 1(-> 0)) => class=4 (CF = 0.99)
Number of Rules : 13
REGLAS = 13

```

### Σχήμα 100

## 5. Ταξινόμηση ιατρικών εικόνων με χρήση Convolutional Neural Nets

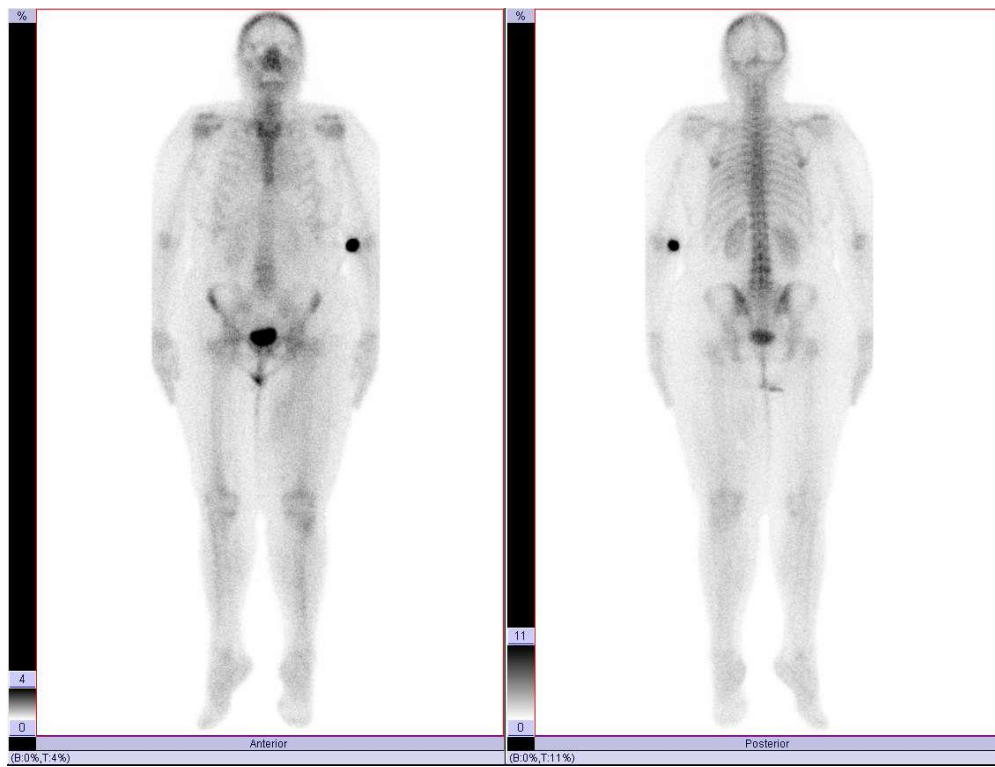
### 5.1. Δεδομένα εικόνας

Τώρα περνάμε στο δεύτερο μέρος της πτυχιακής, στο οποίο ακολουθεί η έρευνα των ακτινογραφιών, όπου έχει προηγηθεί σπινθηρογράφημα και θέλουμε να γίνει εξέταση αν οι ασθενείς πάσχουν από καρκίνο του προστάτη . Το σπινθηρογράφημα είναι μια διαγνωστική εξέταση κατά την οποία χορηγείται ενδοφλεβίως ένα ραδιενεργό φάρμακο και στη συνέχεια η ραδιενέργεια αυτή διαχέεται στο σώμα. Η κατανομή της ραδιενεργού ουσίας στο εξεταζόμενο όργανο δίνει διαγνωστικά στοιχεία στον ακτινολόγο ή στον πυρηνικό ιατρό. Κατόπιν, με τη βοήθεια ηλεκτρονικού υπολογιστή, γίνεται η κατάλληλη επεξεργασία και βγαίνει το τελικό αποτέλεσμα της εξέτασης.

Σύμφωνα με την εργασία του Johnny Dang “Classification in Bone Scintigraphy Images Using Convolutional Neural Networks”, η οποία δημοσιεύτηκε τον Ιούνιο του 2016, ο στόχος είναι η δημιουργία ενός συνελκτικού νευρωνικού δικτύου ώστε να κατηγοριοποιήσει αν τα hotspots, δηλαδή τα έντονα σημεία στις σκελετικές εικόνες, όπου έχει χορηγηθεί ραδιενεργό υλικό αναπαριστούν μετάσταση του καρκίνου του προστάτη, ή παθολογικές άλλες αιτίες. Το δίκτυο που δημιουργήθηκε εκπαιδεύτηκε με είσοδο την μπροστινή και την πίσω αναπαράσταση του ασθενή και χρησιμοποιήθηκε η βιβλιοθήκη Keras.

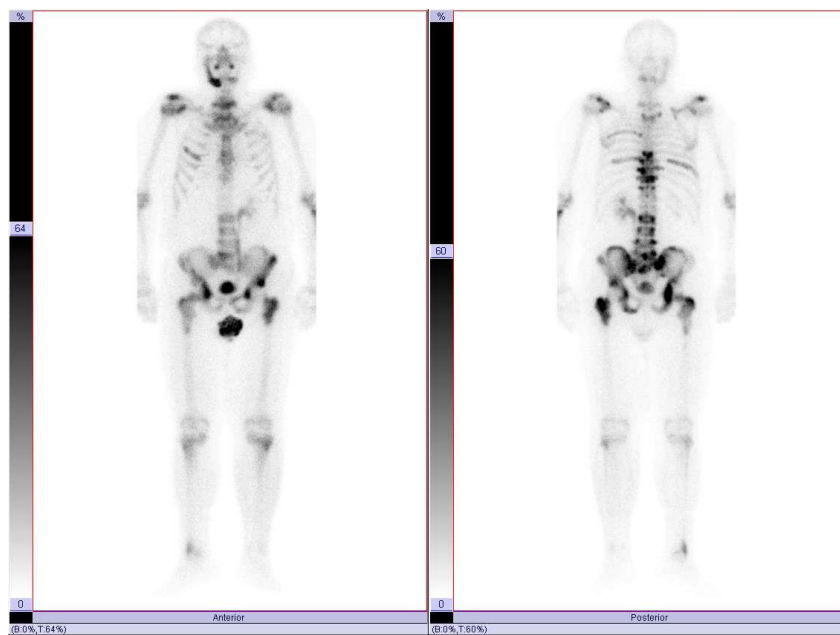
Θα εξετάσουμε δύο κατηγορίες: Φυσιολογικές και Παθολογικές, όπου διαθέτουμε 78 φυσιολογικά και 159 παθολογικά περιστατικά.

Η μορφή των εικόνων είναι εξής. Αυτή είναι η μορφή των Φυσιολογικών



*Εικόνα 1*

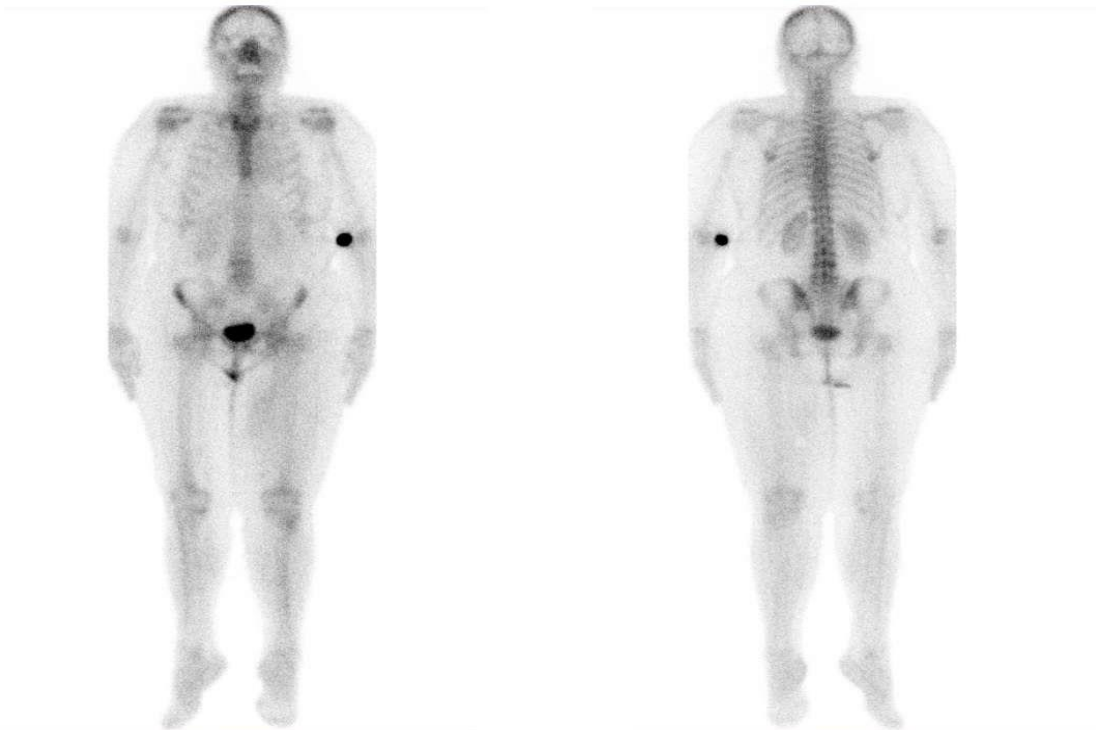
Και η παρακάτω των παθολογικών



*Εικόνα 2*

## 5.2. Τί είναι η προ-επεξεργασία εικόνας (image preprocessing)

Δε γίνεται να εισάγουμε τις φωτογραφίες με την μορφή που βλέπουμε παραπάνω, καθώς περιέχουν τμήματα που όχι μόνο δεν θα επωφελήσουν τον αλγόριθμο, αλλά θα τον αποπροσανατολίσουν. Για παράδειγμα παρατηρούμε ότι περιέχονται στοιχεία όπως ημερομηνία της εξέτασης, στοιχεία δηλαδή που είναι χρήσιμα για την καταγραφή αρχείου, αλλά όχι για κατάλληλα για είσοδο στον αλγόριθμο. Οπότε το περίγραμμα της κάθε φωτογραφίας πρέπει να αφαιρεθεί και παρακάτω βλέπουμε τα αποτελέσματα ύστερα από τις περικοπές. Το εργαλείο που χρησιμοποιήθηκε είναι η Ζωγραφική, το οποίο βρίσκεται στο λειτουργικό σύστημα των Windows.



*Εικόνα 3*

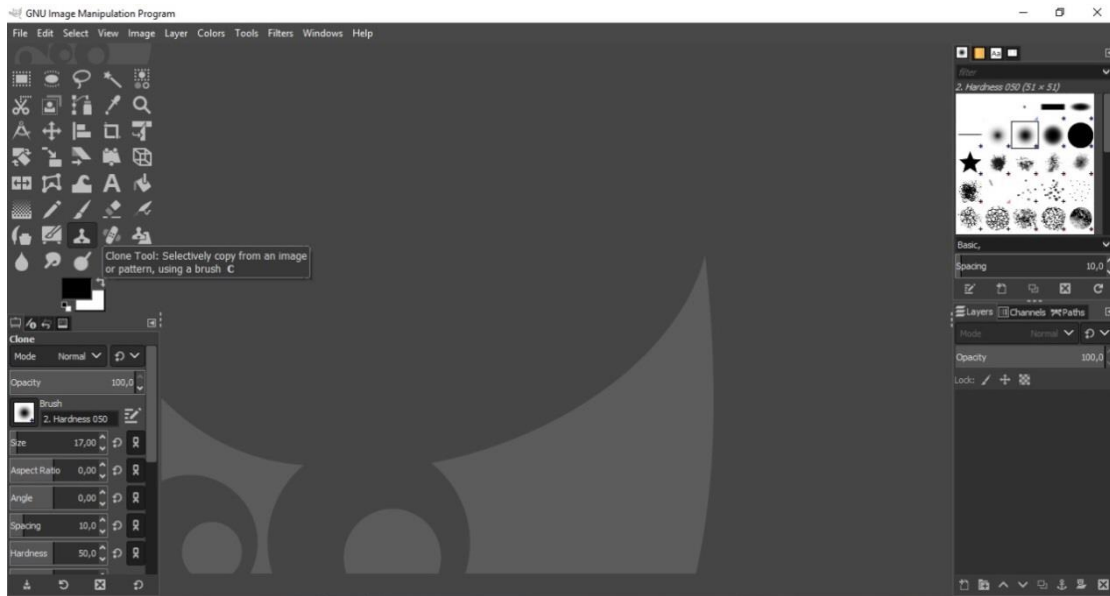


*Εικόνα 4*

Πρέπει να σημειωθεί ότι ύστερα από συνεννόηση με τον γιατρό αποφασίστηκε ότι πρέπει να αφαιρεθεί το σημείο της ουρήθρας, καθώς είναι το υγρό που έχει μαζευτεί στα ούρα και δεν έχει κάποιο όφελος στον αλγόριθμο, καθώς όλοι όσοι εξετάστηκαν θα έχουν μαύρη εκείνη την περιοχή. Το ίδιο πρέπει να συμβεί σε περίπτωση που υπάρχει έντονο σημείο στο χέρι λόγω καθετήρα, όπως φαίνεται στην φυσιολογική περίπτωση. Και το ίδιο αν έχει μαζευτεί υγρό πιο κάτω στο ανδρικό μόριο, η μπορεί και στο εσώρουχο.

Το συγκεκριμένο σημείο της προεπεξεργασίας χρειάζεται ιδιαίτερη προσοχή, καθώς θέλουμε να αφαιρεθούν τα τμήματα που αναφέρθηκαν, χωρίς να διαστρεβλώσουμε την φωτογραφία, οπότε χρειαζόμαστε ένα εργαλείο, το οποίο να παίρνει υπάρχων τμήμα και να το αντικαθιστούμε στο σημείο που θέλουμε να εξαφανίσουμε. Το εργαλείο που χρησιμοποιήθηκε είναι το Gimp 2.10.8.

Η πλατφόρμα του φαίνεται παρακάτω και φαίνεται η περιγραφή του συγκεκριμένου στοιχείου που θα χρησιμοποιήσουμε.



Σχήμα 101

Το επόμενο βήμα είναι να εισάγουμε την φωτογραφία στην πλατφόρμα.



Σχήμα 102



Αντιγράφουμε το σημείο με το χρώμα που θέλουμε και σβήνουμε τα κομμάτια που θέλουμε.



Σχήμα 103

Το τελικό αποτέλεσμα είναι το εξής



Εικόνα 5



*Εικόνα 6*

### **5.3. Τί είναι η τμηματοποίηση εικόνας και εφαρμογή της (image segmentation) εξαγωγή (feature extraction) και επιλογή (feature selection) χαρακτηριστικών από εικόνα**

Αφού έχει προηγηθεί η παραπάνω διαδικασία σειρά έχει η εισαγωγή στον αλγόριθμο και στο τέλος εισαγωγή στο συνελκτικό νευρωνικό δίκτυο. Για να εισαχθεί όμως στο μοντέλο πρέπει να γίνει προεπεξεργασία αλγοριθμικά αυτή τη φορά.

Διασχίζουμε όλες τις φωτογραφίες και περνάμε όλες από την παρακάτω διαδικασία.

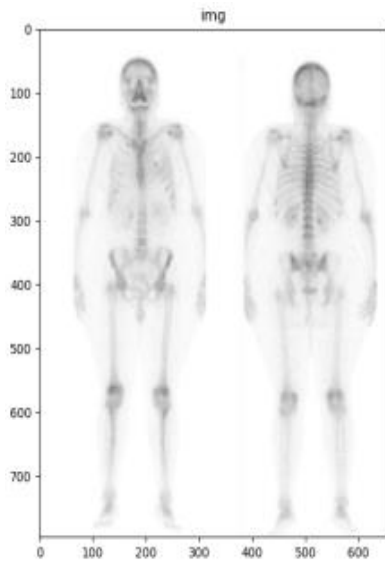
Όλες οι μέθοδοι που περιγράφονται παρακάτω έχουν βρεθεί στη βιβλιοθήκη της OpenCV της Python.

Αφού γίνει εισαγωγή, στη συνέχεια γίνεται μετατροπή της φωτογραφίας σε γκρι.

Μπορεί στο ανθρώπινο μάτι δεν φαίνεται διαφορά, αλλά αλγοριθμικά έχει διαφοροποιηθεί ο αριθμός των καναλιών.

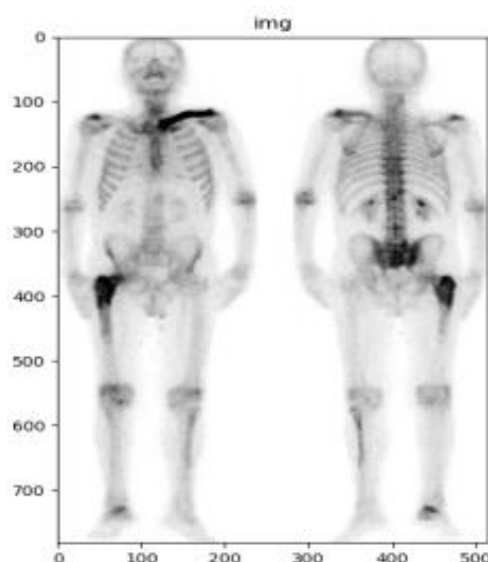
Για τις παρακάτω μεθόδους θα ακολουθήσει και παράδειγμα με φυσιολογική και παθολογική περίπτωση Αρχικά οι φωτογραφίες έχουν την παρακάτω μορφή, αφού έχει προηγηθεί βέβαια η προεπεξεργασία που αναφέρθηκε.

Φυσιολογική



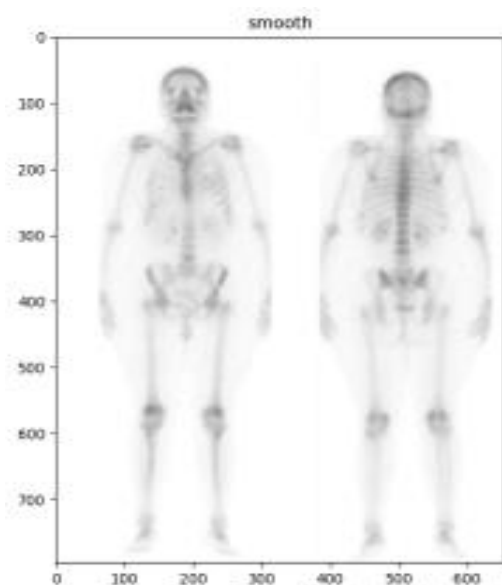
Εικόνα 7

Παθολογική

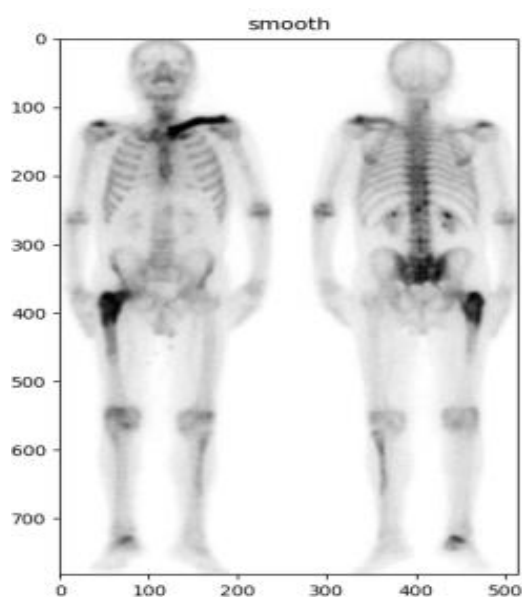


Εικόνα 8

Έχουμε τη διαδικασία blur, θολώνει δηλαδή η φωτογραφία και πρέπει να προσέξουμε σε αυτό το βήμα να αφαιρεθεί θόρυβος, αλλά όχι να χαθεί σημαντική πληροφορία. Για αυτό το λόγο επιλέγουμε μικρό πίνακα 4x4 που να διασχίζει τη φωτογραφία και να τη θολώνει.



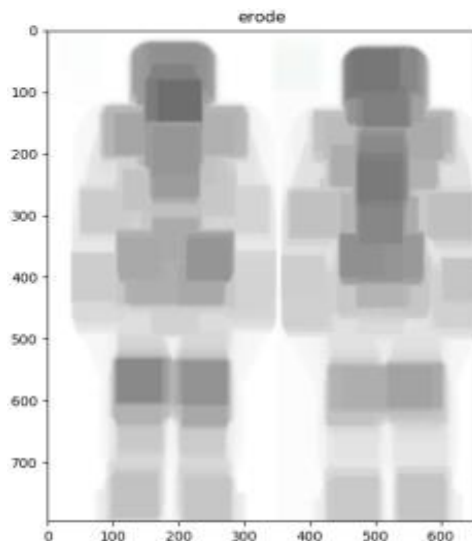
Εικόνα 9



Εικόνα 10

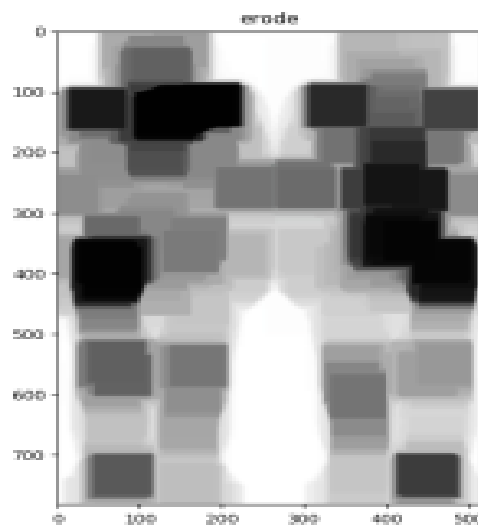
Έπειτα ακολουθεί η μέθοδος erosion, δηλαδή διάβρωση. Η λειτουργία της είναι ότι υπολογίζει το τοπικό ελάχιστο, αντίστοιχο του kernel, που διασχίζει την φωτογραφία, που είναι πίνακας από πίξελ. Αφού υπολογιστεί αντικαθίσταται η τιμή του πίξελ κάτω από τον kernel, όπου οι λευκές περιοχές ελαχιστοποιούνται και οι σκούρες μεγιστοποιούνται. Δηλαδή ξεκαθαρίζουν τα έντονα σημεία της φωτογραφίας. Τα ορίσματα που παίρνει είναι φυσικά η είσοδος, έπειτα είναι το μέγεθος του kernel, το οποίο θα διασχίζει την εικόνα και θα τη διαβρώνει. Τρίτη παράμετρος είναι οι επαναλήψεις που πρέπει να κάνει για να σιγουρευτεί ότι δεν παρέλειψε κάποιο σημείο. Παράκατω βλέπουμε πως έχουν διαμορφωθεί οι εικόνες σύμφωνα με αυτή την μέθοδο.

Φυσιολογική



Εικόνα 11

Παθολογική



Εικόνα 12

Προτελευταία μέθοδος προεπεξεργασίας είναι η thresholding, όπου επιλέχθηκε ανάμεσα στις ακόλουθες

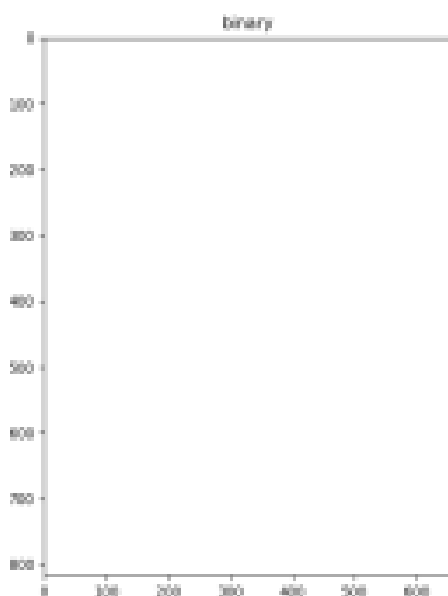
- cv.THRESH\_BINARY
- cv.THRESH\_BINARY\_INV
- cv.THRESH\_TRUNC
- cv.THRESH\_TOZERO
- cv.THRESH\_TOZERO\_INV

η πρώτη, καθώς φαίνεται ότι ταιριάζει περισσότερο και ότι αναδεικνύει καλύτερα τα σημεία που θέλουμε να τονιστούν. Όλες παίρνουν 4 ορίσματα: το πρώτο είναι η είσοδος, το δεύτερο είναι το όριο στα pixels, το οποίο στη συγκεκριμένη περίπτωση είναι 40 και σημαίνει ότι οτιδήποτε πάνω από 40 να το κάνει 255 που είναι η τρίτη παράμετρος που δέχεται. Και η τέταρτη παράμετρος που δέχεται ποια μέθοδο thresholding να χρησιμοποιηθεί.

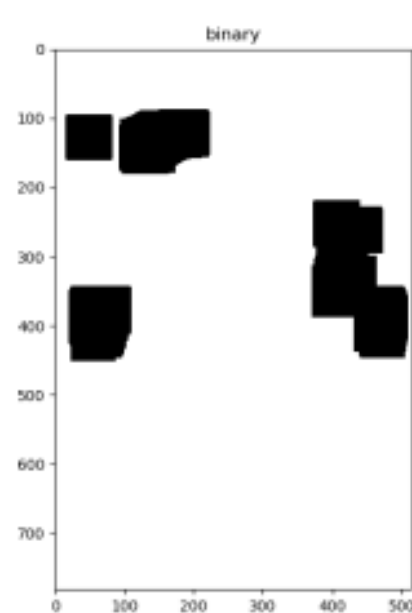
Ακολουθούν και οι πέντε περιπτώσεις για κάθε τύπο φωτογραφίας.

#### cv.THRESH\_BINARY

Στην πρώτη περίπτωση έχουμε: μιας και μπήκε το όριο στο 40 για την τιμή των pixels, βλέπουμε ότι στην περίπτωση των φυσιολογικών δεν έχει τονιστεί κάποιο σημείο έντονα. Αντίθετα, στην περίπτωση των παθολογικών έχουν τονιστεί αρκετά σημεία



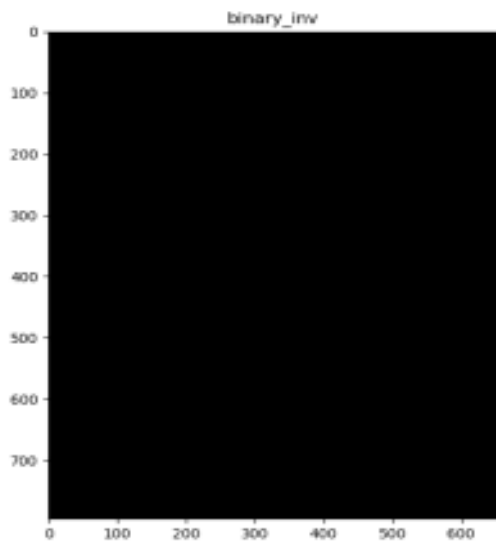
Εικόνα 13



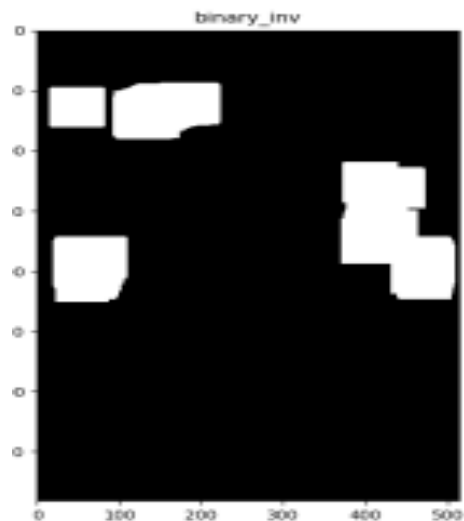
Εικόνα 14

#### cv.THRESH\_BINARY\_INV

Στη δεύτερη περίπτωση έχουμε ότι ίσχυε και παραπάνω, αλλά με την αντιστροφή χρωμάτων, από άσπρο σε μαύρο και αντίθετα.



Εικόνα 15



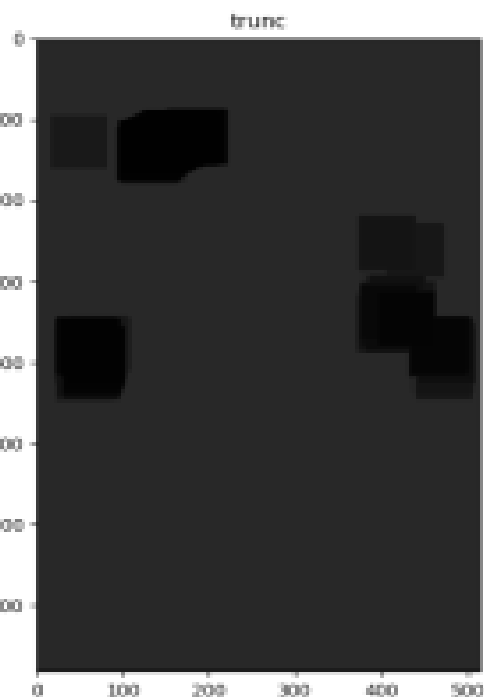
Εικόνα 16

### cv.THRESH\_TRUNC

Στην τρίτη περίπτωση thresholding δεν είναι ξεκάθαρη για τον αλγόριθμο η διαφορά, καθώς η τιμή των πίξελ είναι πολύ κοντά για να μπορέσει ο αλγόριθμος να τα ξεχωρίσει. Οπότε στο συγκεκριμένο πρόβλημα δεν προτιμάμε αυτή την τεχνική.



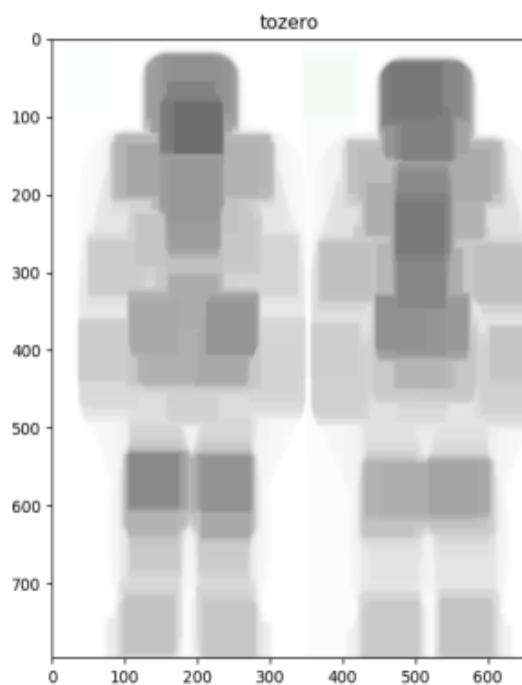
Εικόνα 17



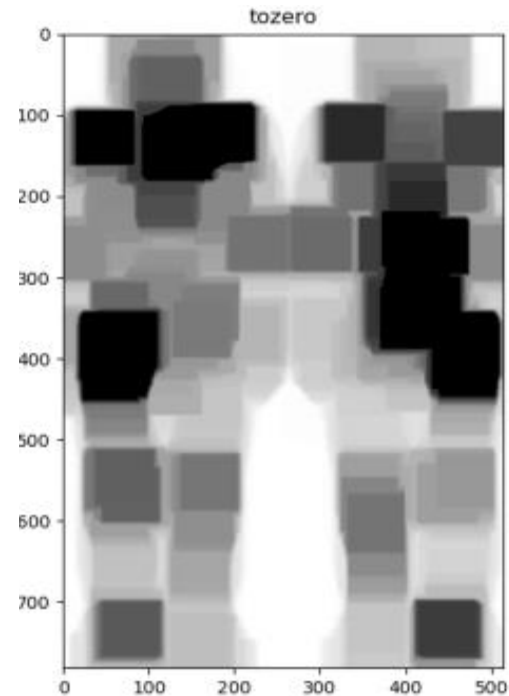
Εικόνα 18

## cv.THRESH\_TOZERO

Η επόμενη συνάρτηση thresholding δεν μας βοηθάει καθόλου καθώς είναι σαν την έξοδο, από τη μέθοδο erode, που μόλις προηγήθηκε.



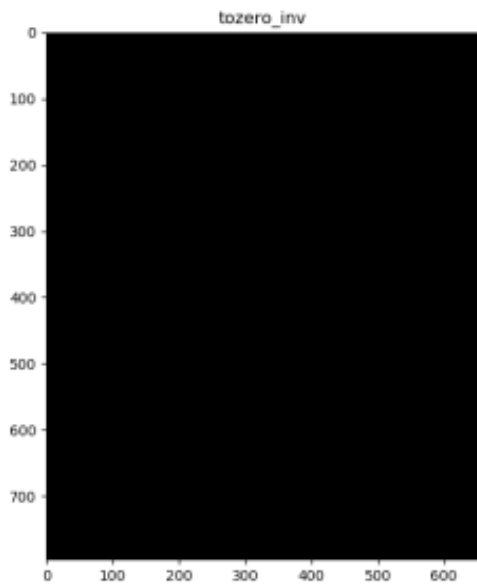
Εικόνα 19



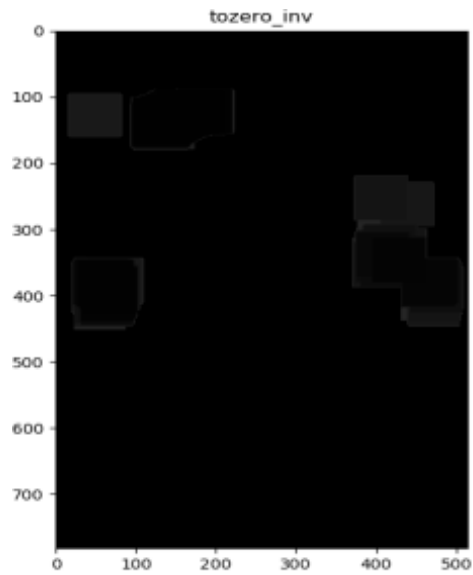
Εικόνα 20

## cv.THRESH\_TOZERO\_INV

Η λέξη inv επισημαίνει την αντιστροφή των χρωμάτων, από άσπρο σε μαύρο και αντίθετα, το οποίο δε μας βοηθάει.



Εικόνα 21



Εικόνα 22

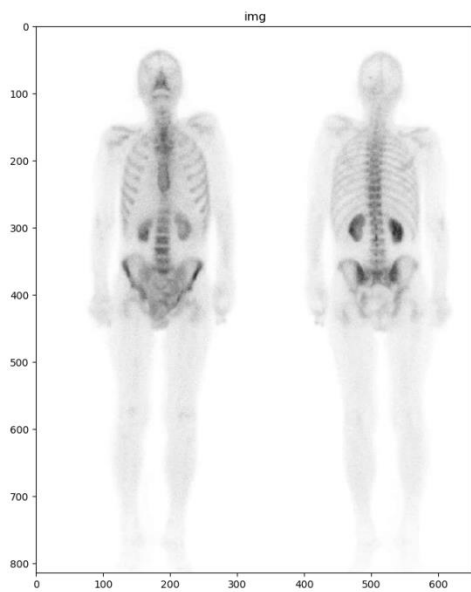
Από ότι βλέπουμε από τη μέθοδο thresholding καλύτερα είναι η binary.

Σε αυτές τις περιπτώσεις φαίνεται ξεκάθαρα η διαφορά ανάμεσα σε φυσιολογική και παθολογική, όμως υπάρχουν περιπτώσεις, όπου στις φυσιολογικές περιπτώσεις να υπάρχει πρόβλημα αρθρικό και με τη μέθοδο erosion, που γίνεται διάβρωση να τονιστεί έντονα και να παρερμηνεύσει λάθος ο αλγόριθμος το μοτίβο αυτό. Επίσης το ίδιο μπορεί να συμβεί και σε μια παθολογική κατάσταση, όπου ο ασθενής να μη βρίσκεται σε προχωρημένη κατάσταση, όποτε να μη φανούν έντονα τα παθολογικά στίγματα.

Παρακάτω βλέπουμε ένα τέτοιο παράδειγμα για κάθε περίπτωση.

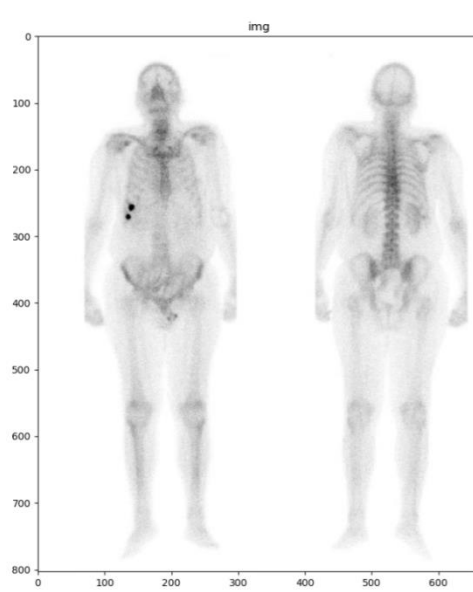


Φυσιολογική



Εικόνα 23

Παθολογική

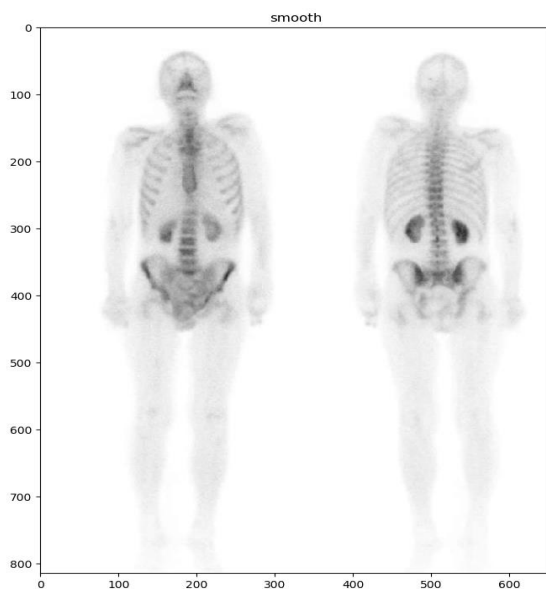


Εικόνα 24

Όπως φαίνεται η φυσιολογική έχει πιο πολλά έντονα σημεία από την παθολογική.

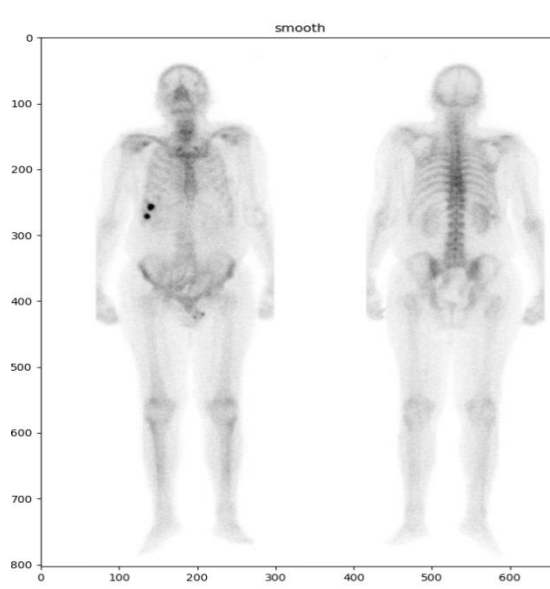
Το επόμενο βήμα της προεπεξεργασίας είναι η θόλωση.

Φυσιολογική



Εικόνα 25

Παθολογική

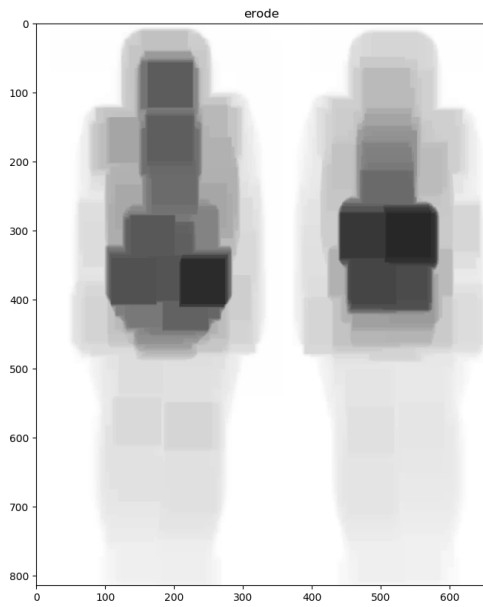


Εικόνα 26

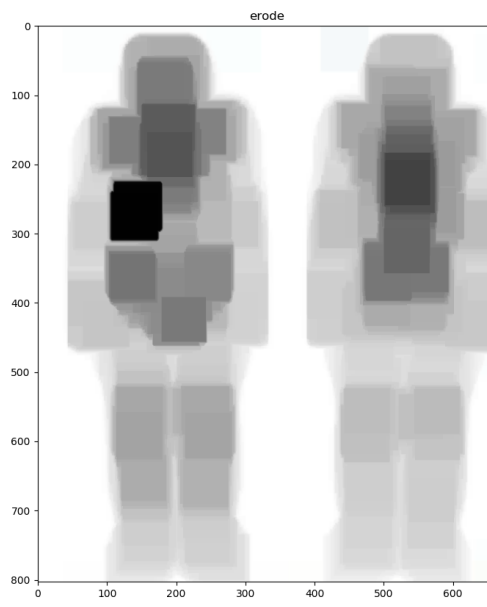
Έπειτα ακολουθεί η μέθοδος erosion, η οποία τονίζει τα σημεία.

Φυσιολογική

Παθολογική



Εικόνα 27

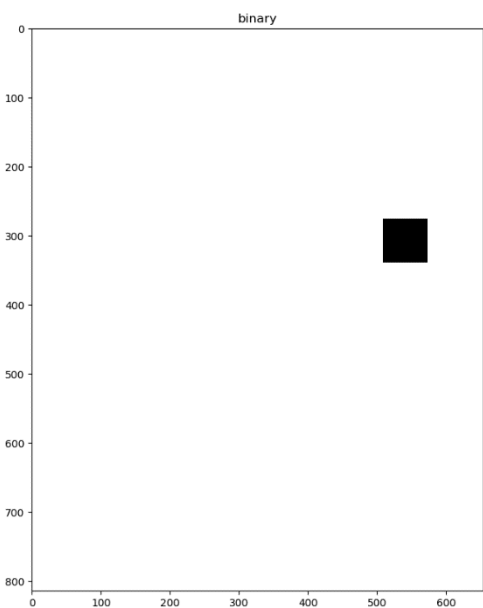


Εικόνα 28

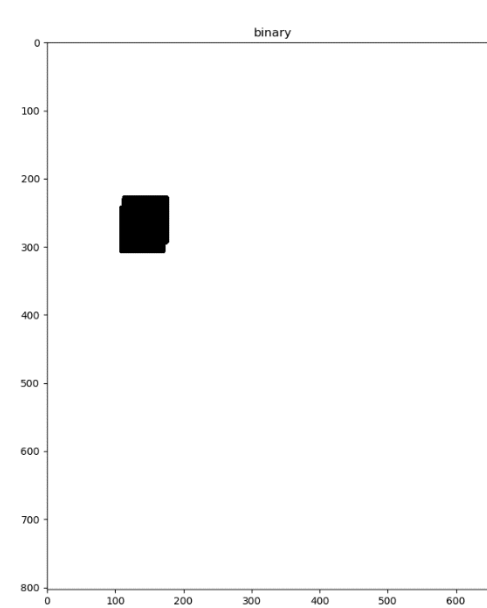
Και τέλος έχουμε τη σημαντικότερη μέθοδο, που ξεκαθαρίζει το αποτέλεσμα, που είναι το thresholding. Βλέπουμε ότι και στις δύο περιπτώσεις έχουμε σχεδόν το ίδιο αποτέλεσμα, όποτε είναι αυτά τα παραδείγματα, που μπορεί να μπερδεύουν τον αλγόριθμο

Φυσιολογική

Παθολογική



Εικόνα 29



Εικόνα 30

## 6. Βαθιά μάθηση (Deep Learning)

### 6.1. Τί είναι η Βαθιά Μάθηση

Η βαθιά μάθηση είναι μια τεχνική μηχανικής μάθησης και μαθαίνει στους υπολογιστές να λειτουργούν σύμφωνα με τον ανθρώπινο τρόπο, Βρίσκεται πίσω από τα αυτόματα αυτοκίνητα και τους επιτρέπει να ξεχωρίζουν έναν πεζό με ένα σήμα και με ένα αυτοκίνητο. Είναι γνωστό, ότι πλέον επιτυγχάνει αποτελέσματα, που παλιότερα δεν ήταν εφικτό.

Οι αρχιτεκτονικές της βαθιάς μάθησης όπως βαθιά νευρωνικά δίκτυα (deep neural networks), αναδρομικά νευρωνικά δίκτυα (recurrent neural networks), και συνελκτικά νευρωνικά δίκτυα (convolutional neural networks) εφαρμόζονται σε τομείς όπως υπολογιστική όραση (computer vision), αναγνώριση φωνής (speech recognition), επεξεργασία φυσικής γλώσσας (natural language processing), αναγνώριση βίντεο (audio recognition), μετάφραση μηχανής (machine translation), ανάλυση ιατρικής εικόνας (medical image analysis).

Η έννοια βαθιά αναφέρεται στο γεγονός ότι υπάρχουν αρκετά κρυμμένα στρώματα στο δίκτυο. Τα κλασσικά δίκτυα περιέχουν 2 με 3 κρυμμένα στρώματα, ενώ τα βαθιά δίκτυα μπορεί να έχουν και 150.

Τα μοντέλα που λειτουργούν με βαθιά μάθηση χρησιμοποιούν αρχιτεκτονικές νευρωνικών δικτύων, τα οποία περιέχουν πολλά επίπεδα.

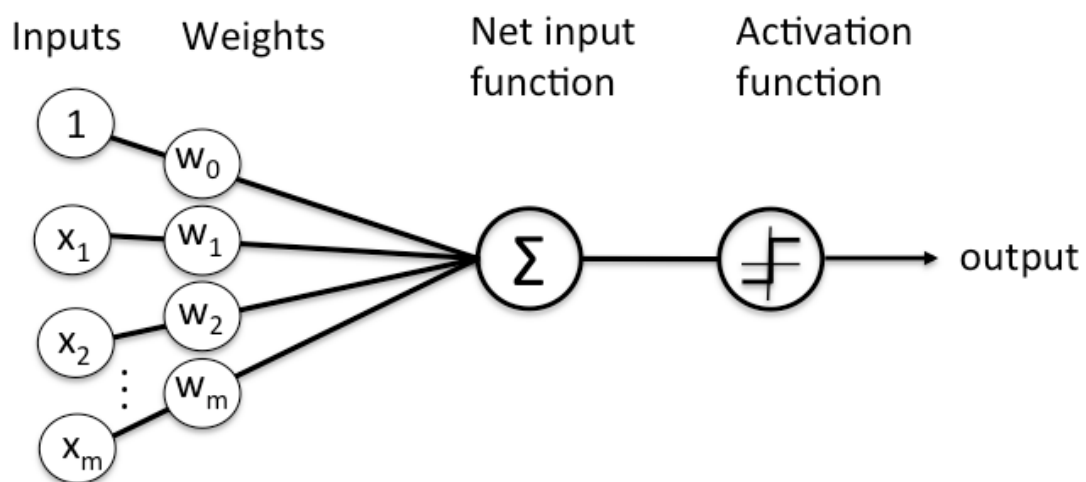
Τα νευρωνικά δίκτυα είναι ένα σει αλγορίθμων διαμορφωμένα σαν τις συνάψεις που βρίσκονται στον ανθρώπινο εγκέφαλο, τα οποία σχεδιάστηκαν για να αναγνωρίζουν μοτίβα. Τα μοτίβα αυτά είναι αριθμητικά, τα οποία περιέχονται σε πίνακες από πίξελ.

Έπειτα τα μοτίβα αυτά μας βοηθάνε στο να ομαδοποιηθούνε δεδομένα χωρίς ετικέτα, σύμφωνα με αυτά που ανιχνεύθηκαν στην είσοδο.

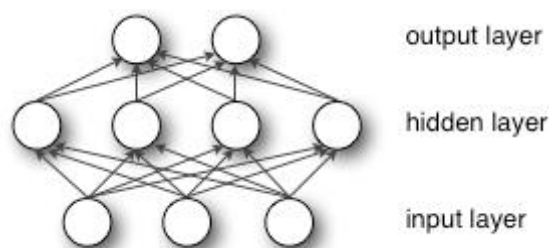
Τα επίπεδα των νευρωνικών δικτύων αποτελούνται από κόμβους . Ο κόμβος είναι το σημείο όπου συμβαίνει ο υπολογισμός, όπως συμβαίνει στους νευρώνες στον ανθρώπινο εγκέφαλο, οι οποίοι ενεργοποιούνται όταν συναντήσουν μια διέγερση. Ο κόμβος συνδυάζει την είσοδο, μαζί με κάποιες παραμέτρους ή βάρη, τα οποία ενισχύουν ή μειώνουν την ροή του δεδομένου αυτού μέσα στο νευρωνικό, καθορίζουν δηλαδή τη βαρύτητα που έχουν στον καθορισμό της ετικέτας. Με λίγα λόγια προσδίδουν αξία στις εισόδους, οι οποίες θα βοηθήσουν τον αλγόριθμο να μάθει τα μοτίβα, ποια είσοδο δηλαδή είναι πιο πιθανό να κατηγοριοποιήσει τα

δεδομένα, χωρίς σφάλμα. Το άθροισμα των εισόδων, αφού έχουν πολλαπλασιαστεί με τα βάρη περνάει μέσα από μια συνάρτηση ενεργοποίησης (activation function) , το οποίο γίνεται για κάθε κόμβο ξεχωριστά και η συνάρτηση αυτή καθορίζει αν ο κόμβος και τα δεδομένα του θα προχωρήσουν.

Παρακάτω βλέπουμε την εξής φωτογραφία για να καταλάβουμε πως είναι αρχιτεκτονικά.



Σχήμα 104



Σχήμα 105

Ο τρόπος που λειτουργούν είναι ο εξής: Κάθε στρώμα χρησιμοποιεί την έξοδο από το προηγούμενο στρώμα σαν είσοδο. Κάθε επίπεδο μαθαίνει να μετατρέπει την είσοδο σε λίγο πιο αφηρημένη και σύνθετη αναπαράσταση. Στην αναγνώριση εικόνας τα ακατέργαστα δεδομένα που είναι η είσοδος μπορεί να είναι ένας πίνακας από πίξελς. Το πρώτο επίπεδο (layer) πιο πιθανά θα κάνει τα πίξελ πιο αφαιρετικά και να κωδικοποιήσει τις γωνίες. Το δεύτερο επίπεδο μπορεί να συνθέσει και να κωδικοποιήσει τις διατάξεις των άκρων. Το τρίτο επίπεδο μπορεί να διαχωρίσει την μύτη και τα μάτια και το τέταρτο επίπεδο μπορεί να είναι σε θέση να αναγνωρίσει το πρόσωπο. Το πιο σημαντικό που πρέπει να κρατήσουμε από αυτό είναι ότι η διαδικασία της βαθιάς μάθησης μπορεί να μάθει ποια χαρακτηριστικά κατά το ιδανικό αρμόζουν σε κάθε επίπεδο. Αυτό φυσικά δε γίνεται χειροκίνητα, αλλά αλλάζοντας τις παραμέτρους όπως τον αριθμό των επιπέδων, το μήκος τους.

Οπότε βλέπουμε ότι και στις δύο τεχνικές, δηλαδή μηχανική και βαθιά μάθηση, ο σκοπός του αλγορίθμου είναι να ανιχνεύσει μοτίβα, με τη μόνη διαφορά ότι στη μηχανική τα εξάγει από τα χαρακτηριστικά, που βρίσκονται σε στήλες, ενώ στη βαθιά μάθηση, από αριθμούς πινάκων.

## 6.2. Τί είναι τα convolutional neural networks

Μια γνωστή κατηγορία, που ανήκει στα βαθιά νευρωνικά δίκτυα είναι τα συνελκτικά.

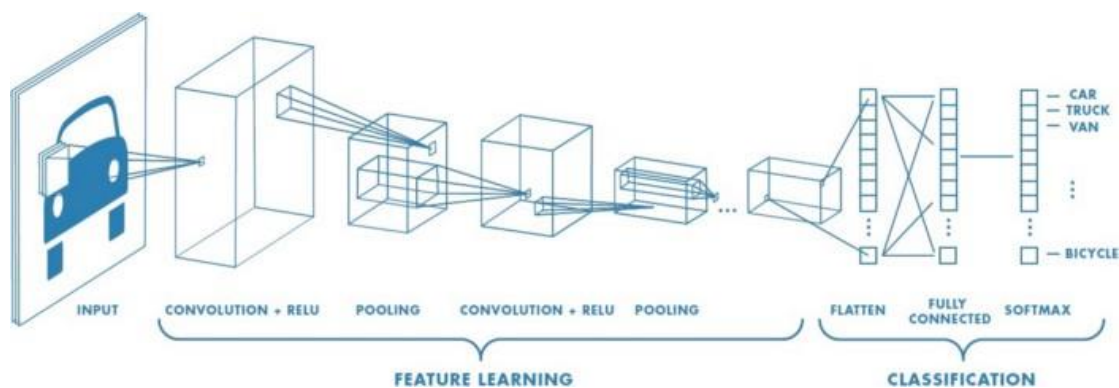
Στα νευρωνικά δίκτυα, τα συνελκτικά νευρωνικά δίκτυα αποτελούν μία από τις κύριες κατηγορίες που χρησιμοποιούνται στην αναγνώριση εικόνας, προσώπου και ανίχνευση αντικειμένου.

Τα convolutional neural networks παίρνουν σαν είσοδο μια εικόνα την επεξεργάζονται και την κατηγοριοποιούν. Ο αλγόριθμος βλέπει την εικόνα σαν έναν πίνακα από πίξελ και ανάλογα τη μορφή της την ερμηνεύει διαφορετικά. Όλες έχουν την μορφή  $h \times w \times d$  (όπου  $h$  είναι το ύψος,  $w$  είναι το πλάτος και  $d$  η διάσταση), οπότε καταλαβαίνουμε ότι μια έγχρωμη θα είναι  $64 \times 64 \times 3$ , όπου 3 είναι ο αριθμός των καναλιών (RGB), ενώ μια ασπρόμαυρη θα είναι  $4 \times 4 \times 1$ . Γενικά τα δίκτυα αυτής της μορφής περνάνε την κάθε εικόνα, μέσα από συνελκτικά

επίπεδα με φίλτρα, Pooling, dense, τα οποία συνοδεύονται με μια συνάρτηση ενεργοποίησης και στο τέλος παράγεται το αποτέλεσμα.

Τα συνελκτικά δίκτυα κάνουνε μόνα τους εξαγωγή πληροφορίας το μόνο που χρειάζεται σαν προεπεξεργασία είναι μετατροπή σε ασπρόμαυρη φωτογραφία, όπως επίσης αφαίρεση θορύβου ή και κάποια μορφολογική επεξεργασία, αν βέβαια καλυτερεύουν τον αλγόριθμο. Δεν χρειάζεται από τον επεξεργαστή να τονίσει τα σημεία.

Μια αντιπροσωπευτική φωτογραφία φαίνεται παρακάτω για την διαδικασία, που αναφέρθηκε.

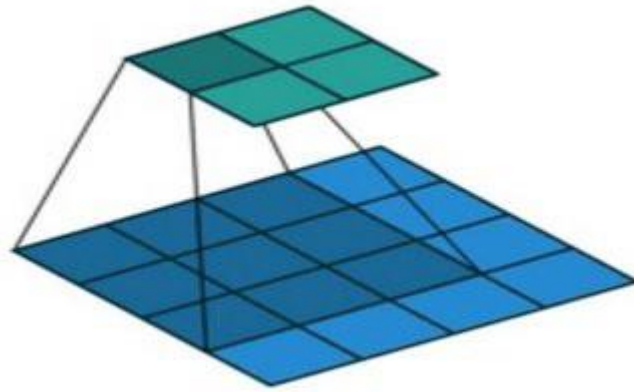


Σχήμα 106

### Συνελκτικά επίπεδα

Η συνέλιξη είναι το πρώτο επίπεδο, το οποίο εξάγει χαρακτηριστικά, από την εικόνα. Παίρνει δύο εισόδους την φωτογραφία και ένα φίλτρο. Συγκεκριμένα, η συνέλιξη πολλαπλασιάζει τον πίνακα από πίξελ, που είναι η ίδια η εικόνα με ένα πίνακα με φίλτρα ή πυρήνας (Kernel) και προσθέτει όλες τις τιμές του πολλαπλασιασμού. Στη συνέχεια η συνέλιξη μετακινείται κατά ένα βήμα, που έχει καθοριστεί και επαναλαμβάνει την ίδια διαδικασία, μέχρι να καλύψει όλα τα πίξελ.

Στα πρώτα συνελκτικά επίπεδα μπορούν να ανιχνεύσουν γραμμές και γωνίες, αλλά καθώς περνάμε αυτά τα μοτίβα μέσα στο νευρωνικό δίκτυο ξεκινάει η διαδικασία αναγνώρισης περισσότερο περίπλοκων χαρακτηριστικών.

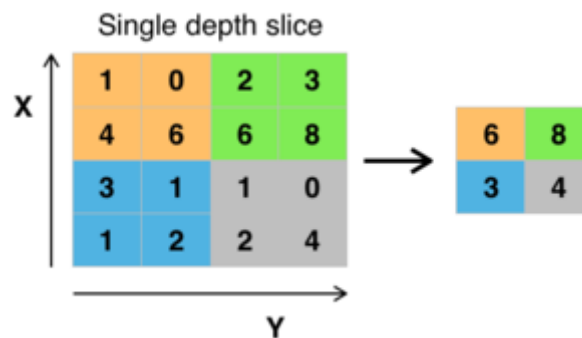


Σχήμα 107

Τώρα ας περάσουμε στις κατηγορίες των hidden layers

### Pooling

Τα convolutional δίκτυα περιλαμβάνουν τοπικά ή καθολικά pooling layers, τα οποία μειώνουν τις διαστάσεις των δεδομένων, έτσι ώστε να ελεγχθεί η υπερφόρτωση (overfitting). Αυτό γίνεται συνδυάζοντας τις εξόδους των νευρώνων σε έναν νευρώνα για το επόμενο επίπεδο. Το τοπικό pooling συνδυάζει μικρά συμπλέγματα, συνήθως 2 x 2 και με βηματισμό 2 (stride), το οποίο αφαιρεί περίπου 75% των δεδομένων. Το καθολικό ενεργεί σε όλους τους νευρώνες του συνελκτικού επιπέδου. Αυτή η πράξη μπορεί να υπολογίζει το μέγιστο ή το μέσο όρο. Το μέγιστο χρησιμοποιεί τη μέγιστη τιμή, από κάθε σύμπλεγμα των νευρώνων, και το μέσο όρο το αντίστοιχο.



Σχήμα 108

## **Fully Connected / Dense**

Τα πλήρως συνδεδεμένα επίπεδα (fully connected layers) συνδέουν κάθε νευρώνα, από το ένα επίπεδο με κάθε νευρώνα, στο επόμενο επίπεδο. Αυτή η μορφή σύνδεσης ακολουθείται αφού έχουν προηγηθεί αρκετά συνελκτικά (convolutional) και μέγιστα συγκεντρωτικά (max pooling) επίπεδα.

## **Receptive field**

Στα νευρωνικά δίκτυα, κάθε νευρώνας λαμβάνει είσοδο, από κάποιες περιοχές του προηγούμενου επιπέδου. Όπως είδαμε παραπάνω σε ένα πλήρως συνδεδεμένο επίπεδο (fully connected layer) οι περιοχές αυτές είναι όλοι οι νευρώνες, που ακολούθησαν στο προηγούμενο επίπεδο. Σε αντίθεση έρχονται τα συνελκτικά επίπεδα, όπου το πεδίο (receptive field) που λαμβάνουν την είσοδο είναι μια περιορισμένη περιοχή, από το προηγούμενο επίπεδο.

## **Συνάρτηση ενεργοποίησης**

Στο δικό μας μοντέλο συναντάμε την Relu και την sigmoid. Μετά τους συνελκτικούς υπολογισμούς προκύπτουν αρνητικές τιμές στα πίξελ, οπότε με τη βοήθεια της Relu διατηρούμε το εύρος τιμών, από 0 και πάνω για να είναι όλα θετικά. Δηλαδή, όποια αρνητική τιμή την κάνει 0. Η σιγμοειδής χρησιμοποιείται στο τελευταίο επίπεδο του νευρωνικού δικτύου και χρησιμοποιείται στα δυαδικά προβλήματα. Δίνει την πιθανότητα που ισχύει για κάθε έξοδο και ανάλογα ποια υπερισχύει προκύπτει το αποτέλεσμα.

## **Weights**

Κάθε νευρώνας σε ένα νευρωνικό δίκτυο υπολογίζει την τιμή εξόδου εφαρμόζοντας κάποια συνάρτηση στις τιμές εισόδου, οι οποίες περιέχονται μέσα στο επιτρεπτό πεδίο (receptive field) του προηγούμενου επιπέδου. Η συνάρτηση αυτή καθορίζεται από έναν πίνακα από βάρη και από πραγματικούς αριθμούς. Τα βάρη δηλαδή καθορίζουν κατά πόσο ένας νευρώνας ή αλλιώς κόμβος θα έχει βαρύτητα στον υπολογισμό της εξόδου.

Γενικότερα τα συνελκτικά νευρωνικά δίκτυα δεν είναι τόσο διαφορετικά, από τους ποικίλους αλγορίθμους, που προσπαθούν να βρουν μοτίβα στα δεδομένα. Η διαφορά είναι ότι σε αυτή την περίπτωση προσπαθούμε να βρούμε κατάλληλα βάρη για τον kernel, έτσι ώστε να καταφέρει να βρει όσο πιο κατάλληλο μοτίβο γίνεται.



### 6.3. Εφαρμογή τους σε ιατρικές εικόνες

Το μοντέλο που χρησιμοποιήσαμε είναι το Sequential, το οποίο είναι το πιο κλασσικό στα συνελκτικά νευρωνικά δίκτυα. Είναι το πιο εύκολο διαχειρίσιμο, καθώς μας επιτρέπει να φτιάξουμε το μοντέλο με μια γραμμική στοίβα από επίπεδα (layers). Χρειάζεται να γνωρίζει το μέγεθος της εισόδου και για αυτό το λόγο πρέπει το πρώτο επίπεδο να καθορίσουμε το μέγεθος. Οπότε το πρώτο επίπεδο είναι ένα διδιάστατο συνελκτικό επίπεδο και στο σύνολο χρησιμοποιούμε τέσσερα ίδιας ιδιότητας επίπεδα. Το πρώτο έχει 32 κόμβους και ο πυρήνας, ο πίνακας δηλαδή που θα φιλτράρει την εικόνα είναι 3 x 3. Για συνάρτηση ενεργοποίησης χρησιμοποιούμε την Relu και στη συνέχεια καθορίζουμε το μέγεθος που θα έχουν οι εικόνες.

Ακολουθεί το δεύτερο επίπεδο, που είναι της μορφής Pooling και συγκεκριμένα max pooling. Στο δικό μας μοντέλο χρησιμοποιήθηκαν 4 τέτοια επίπεδα με kernel 2 x 2.

Μετά από κάθε max pooling επίπεδο, ακολουθεί ένα επίπεδο drop out, το οποίο δέχεται μια παράμετρο την τιμή drop rate, η οποία έχει την εξής λειτουργία. Μέσα στο νευρωνικό δίκτυο κάποιιο κόμβοι μπορεί να πάψουν να λειτουργούν και η τιμή του drop rate καθορίζει τη βαρύτητα των βαρών και ανάλογα την τιμή κανονικοποιούνται οι τιμές τους.

Οπότε μέχρι στιγμής έχουμε προσθέσει στο δίκτυο ένα συνελκτικό, μετά ένα Pooling και μετά ένα Drop out. Αυτή την ακολουθία θα ακολουθήσουμε για άλλες τρεις φορές με τη διαφορά, ότι το επόμενο συνελκτικό θα έχει 64 κόμβους και τα άλλα δύο 128. Μετά από τα 12 επίπεδα ακολουθεί ένα Flatten layer, το οποίο αποτελεί τον συνδετικό κρίκο ανάμεσα στα συνελκτικά και στο Dense επίπεδο. Όπως γίνεται κατανοητό ακολουθεί ένα Dense επίπεδο, το οποίο αποτελείται από 512 κόμβους και συνδέει όλους τους κόμβους από το προηγούμενο επίπεδο σε αυτό με συνάρτηση ενεργοποίησης Relu. Στη συνέχεια ακολουθεί ένα dropout για τον λόγο ότι με τη χρήση τους αποφεύγουμε το overfitting. Μετά από αυτό ακολουθεί το τελευταίο Layer, το οποίο είναι Dense με έναν κόμβο μιας και η έξοδος είναι 0 ή 1, δηλαδή φυσιολογική ή παθολογική περίπτωση, με συνάρτηση ενεργοποίησης τη σιγμοειδή, η οποία είναι η κατάλληλη για έξοδο από νευρωνικό δίκτυο.

Ακολουθεί η μέθοδος compile, η οποία καθορίζει την διαδικασία της μάθησης. Σαν παραμέτρους δέχεται ένα βελτιστοποιητή (optimizer), στην προκειμένη περίπτωση επιλέχθηκε ο adam. Επόμενη παράμετρος είναι η συνάρτηση απώλειας, όπου διαλέχθηκε η binary crossentropy, όπου πρέπει να επιλεγθεί η κατάλληλη, καθώς είναι αυτή που

προσπαθεί να μειώσει το σφάλμα του μοντέλου. Τέλος, επιλέγουμε τη μέθοδο μετρικής και για προβλήματα κατηγοριοποίησης επιλέγεται η accuracy.

Για τη διαδικασία της εκπαίδευσης κανονικά έπρεπε να επιλέξουμε τη μέθοδο fit, αλλά επειδή έχουμε μικρό dataset θα χρησιμοποιήσουμε τη μέθοδο ImageDataGenerator και ανάλογα το fit\_generator. Παράγει δέσμη από τεμάχια (batch) από δεδομένα εικόνας, τα οποία αυξάνονται και ο αλγόριθμος δουλεύει με παραπάνω, από αυτά που του εισάγουμε. Ο τρόπος που αυξάνονται είναι να αλλάζουμε το μέγεθος της εικόνας, να την περιστρέφουμε, να γίνεται ζουμ, μετατόπιση στο ύψος και στο πλάτος. Οι παραπάνω μετατροπές γίνονται μόνο στα δεδομένα, που προορίζονται για εκπαίδευση του μοντέλου, ενώ αυτά που είναι για έλεγχο γίνονται μόνο ανακατανομή του μέγεθος. Για εκπαίδευση πρέπει να επιλέξουμε το μέγεθος της δέσμης των δεδομένων, που θα περνάει από τον αλγόριθμο (batch size). Επιπλέον είναι αναγκαίο να διαλέξουμε τον αριθμό των εποχών (epochs), πόσες φορές δηλαδή θα τρέξει ο αλγόριθμος. Στη συνέχεια χρειάζεται να επιλέξουμε τον αριθμό βημάτων για εκπαίδευση και για έλεγχο και είναι καλύτερο να προσαρμόζεται ανάλογα με τον αριθμό του batch size και τον αριθμό των δεδομένων που χωρίστηκαν ανάλογα στην κάθε περίπτωση. Δηλαδή για εκπαίδευση θα χρειαστούν βήματα, που προκύπτει από το υπόλοιπο της διαίρεσης των δεδομένων της εκπαίδευσης και του Batch size, ανάλογα και για τα δεδομένα ελέγχου.

Βλέπουμε ένα παράδειγμα πως τρέχει ο αλγόριθμος

```
Epoch 2/30
11/11 [=====] - 4s 373ms/step - loss: 0.6549 - acc: 0.6908 - val_loss: 0.6789 - val_acc: 0.6364
Epoch 3/30
11/11 [=====] - 4s 358ms/step - loss: 0.6057 - acc: 0.7137 - val_loss: 0.6768 - val_acc: 0.5909
Epoch 4/30
11/11 [=====] - 4s 352ms/step - loss: 0.6503 - acc: 0.6241 - val_loss: 0.6547 - val_acc: 0.6818
Epoch 5/30
11/11 [=====] - 4s 348ms/step - loss: 0.5518 - acc: 0.7194 - val_loss: 0.5310 - val_acc: 0.6458
Epoch 6/30
11/11 [=====] - 4s 349ms/step - loss: 0.4469 - acc: 0.7137 - val_loss: 0.4129 - val_acc: 0.6818
Epoch 7/30
11/11 [=====] - 4s 349ms/step - loss: 0.4887 - acc: 0.7595 - val_loss: 0.3778 - val_acc: 0.8409
Epoch 8/30
11/11 [=====] - 4s 371ms/step - loss: 0.3546 - acc: 0.8626 - val_loss: 0.4608 - val_acc: 0.7500
Epoch 9/30
11/11 [=====] - 4s 385ms/step - loss: 0.3097 - acc: 0.8969 - val_loss: 0.3785 - val_acc: 0.8125
Epoch 10/30
11/11 [=====] - 4s 383ms/step - loss: 0.4494 - acc: 0.7996 - val_loss: 0.3350 - val_acc: 0.8182
Epoch 11/30
11/11 [=====] - 4s 386ms/step - loss: 0.2936 - acc: 0.9027 - val_loss: 0.3645 - val_acc: 0.7955
Epoch 12/30
11/11 [=====] - 4s 390ms/step - loss: 0.3145 - acc: 0.8626 - val_loss: 0.2799 - val_acc: 0.8864
Epoch 13/30
11/11 [=====] - 4s 405ms/step - loss: 0.3580 - acc: 0.8466 - val_loss: 0.2995 - val_acc: 0.8333
Epoch 14/30
11/11 [=====] - 4s 379ms/step - loss: 0.6362 - acc: 0.6814 - val_loss: 0.4555 - val_acc: 0.7955
Epoch 15/30
11/11 [=====] - 4s 382ms/step - loss: 0.4168 - acc: 0.8339 - val_loss: 0.3757 - val_acc: 0.7955
Epoch 16/30
11/11 [=====] - 4s 385ms/step - loss: 0.3522 - acc: 0.8969 - val_loss: 0.3517 - val_acc: 0.9545
Epoch 17/30
11/11 [=====] - 4s 391ms/step - loss: 0.3695 - acc: 0.8912 - val_loss: 0.3323 - val_acc: 0.8333
Epoch 18/30
11/11 [=====] - 4s 384ms/step - loss: 0.3595 - acc: 0.8225 - val_loss: 0.2105 - val_acc: 0.9773
Epoch 19/30
11/11 [=====] - 4s 391ms/step - loss: 0.2615 - acc: 0.8855 - val_loss: 0.3203 - val_acc: 0.8409
Epoch 20/30
11/11 [=====] - 4s 388ms/step - loss: 0.3301 - acc: 0.8569 - val_loss: 0.1883 - val_acc: 0.9545
Epoch 21/30
11/11 [=====] - 4s 396ms/step - loss: 0.2283 - acc: 0.8969 - val_loss: 0.2093 - val_acc: 0.8750
Epoch 22/30
11/11 [=====] - 4s 395ms/step - loss: 0.3510 - acc: 0.8225 - val_loss: 0.2980 - val_acc: 0.8182
Epoch 23/30
11/11 [=====] - 4s 382ms/step - loss: 0.2750 - acc: 0.9027 - val_loss: 0.2022 - val_acc: 0.9318
Epoch 24/30
11/11 [=====] - 4s 385ms/step - loss: 0.3206 - acc: 0.8511 - val_loss: 0.2368 - val_acc: 0.8864
Epoch 25/30
11/11 [=====] - 5s 426ms/step - loss: 0.3476 - acc: 0.8580 - val_loss: 0.2104 - val_acc: 0.9375
Epoch 26/30
11/11 [=====] - 4s 385ms/step - loss: 0.3485 - acc: 0.8683 - val_loss: 0.1952 - val_acc: 0.9318
Epoch 27/30
11/11 [=====] - 4s 390ms/step - loss: 0.2316 - acc: 0.9027 - val_loss: 0.2384 - val_acc: 0.9091
Epoch 28/30
11/11 [=====] - 4s 384ms/step - loss: 0.2817 - acc: 0.8740 - val_loss: 0.1961 - val_acc: 0.9545
Epoch 29/30
11/11 [=====] - 4s 384ms/step - loss: 0.2606 - acc: 0.8569 - val_loss: 0.1784 - val_acc: 0.9375
Epoch 30/30
11/11 [=====] - 4s 397ms/step - loss: 0.4258 - acc: 0.7787 - val_loss: 0.2440 - val_acc: 0.9318
```

Σχήμα 109

Τελευταίο βήμα, αλλά πιο σημαντικό είναι να δούμε την απόδοση του μοντέλου. Δημιουργούμε μια καινούργια μεταβλητή, όπου περιέχει τις προβλεπόμενες εξόδους από τα δεδομένα ελέγχου (testing data). Έπειτα, δημιουργούμε τον πίνακα σύγκρισης και την απόδοση, συγκρίνοντας τα δεδομένα που προβλέφθηκαν με τα πραγματικά. Παρακάτω βλέπουμε ένα παράδειγμα, όπου φαίνεται πρώτα ο πίνακας και στη συνέχεια φαίνεται η απόδοση.

```
[[18 3]
 [ 3 36]]
98.8
```

Σχήμα 110

## 6.4 Αποτελέσματα

Σύμφωνα με τα παραπάνω δοκιμάστηκαν αρκετές παραλλαγές και προέκυψαν τα παρακάτω αποτελέσματα, όπου έχουν εξαχθεί πρώτα τα χαρακτηριστικά με τη διαδικασία που αναφέρθηκε παραπάνω με τις βιβλιοθήκες της OpenCV και στη συνέχεια περάστηκαν μέσα από το συνελκτικό δίκτυο. Βλέπουμε ότι έχουμε 3 διαφορετικά πινακάκια στην αρχή, τα οποία έχουν διαφορετικές τιμές στις παραμέτρους στην κάθε στήλη. Αυτό που αλλάζει όμως στους διαφορετικούς πίνακες είναι η τιμή του drop rate. Στο πρώτο πινακάκι η τιμή είναι 0.2, στο δεύτερο είναι 0.5 και στο τρίτο είναι 0.7 Επίσης βλέπουμε ότι έχουμε πέντε στήλες, όπου διαφοροποιούνται οι παράμετροι του batch size, τα pixels και η τιμή των epochs. Όπως φαίνεται καλύτερα αποτελέσματα βγάζει η πρώτη στήλη, καθώς φτάνει σχεδόν στο 85%.

	pixel 150,150 Drop-rate= 0.2 Epochs=25 75-25	Pixel 200,200 Drop- rate=0.2	Pixel 300,300 Drop- out=0.2	Pixels 150, 150 Batch- size=32 Drop- out=0.2	pixel 150,150 Drop-rate= 0.2 Epochs=30 75-25
Run 1	90%	88.3%	80%	85%	61.6%
Run 2	90%	81.6%	63%	61.6%	73.3
Run 3	85%	63.3%	86.6%	86.6%	90%
Run 4	91.6%	88.3%	66.66%	78.33%	90%
Run 5	83.3%	63.3%	71.6%	60%	90%
Run 6	83.3%	83.3%	36.6%	88.3%	86.6%
Run 7	88.3%	86.6%	85%	48.3%	88.3%
Run 8	60%	65%	83.3%	70%	90%
Run 9	90%	85%	91.6%	70%	90%
Run 10	88.3%	90%	93.3%	65%	40%
<b>Average Accuracy</b>	<b>84.98%</b>	<b>79.47%</b>	<b>75.76%</b>	<b>71.3%</b>	<b>79.98%</b>

Πίνακας 3

Στη συνέχεια έχουμε τα ίδια τρεξίματα, εκτός από epochs=30, για drop rate-0.5 και παρατηρούμε ότι έχουν πέσει οι αποδόσεις σε όλες τις περιπτώσεις.

	Pixel 150,150 Drop-out= 0.5	Pixel 200,200 Drop-out=0.5	Pixel 300,300 Drop- out=0.5	Batch-size=32 Drop-out=0.5
Run 1	50%	73.3%	63.3%	75%
Run 2	86.6%	61.6%	70%	93.3%
Run 3	51.6%	56.6%	66.6%	76.6%
Run 4	35%	55%	66.6%	71.6%
Run 5	61.6%	66.6%	71.6%	85%
Run 6	33.3%	31.6%	65%	86.6%
Run 7	53.3%	88.3%	66.6%	36.6%
Run 8	63.3%	61.6%	58.3%	76.6%
Run 9	48.3%	36.6%	63.3%	71.6%
Run 10	63.3%	65%	58.3%	38.3%
Average Accuracy	<b>54.63%</b>	<b>59.62%</b>	<b>64.96%</b>	<b>71.12%</b>

Πίνακας 4

Τελευταίος πίνακας, όπου έχει γίνει χειροκίνητα η εξαγωγή χαρακτηριστικών φαίνεται παρακάτω, όπου η τιμή του drop rate είναι 0.7. Παρατηρούμε ότι στις τρεις πρώτες στήλες έχουμε μέτρια αύξηση της απόδοσης.

	Pixel 150,150 Drop-out= 0.7	Pixel 200,200 Drop-out=0.7	Pixel 300,300 Drop-out=0.7	Batch-size=32 Drop-out=0.7
Run 1	26.6%	75%	66.6%	81.6%
Run 2	31.6%	71.6%	60%	46.6%
Run 3	68.3%	66.6%	70%	38.3%
Run 4	68.3%	38.3%	71.6%	66.6%
Run 5	61.6%	66.6%	68.3%	56.6%
Run 6	26.6%	70%	65%	63.3%
Run 7	63.3%	91.6%	75%	73.3%
Run 8	83.3%	68.3%	61.6%	63.3%
Run 9	83.3%	65%	66.6%	71.6%
Run 10	65%	70%	68.3%	66.6%
Average Accuracy	<b>57.79%</b>	<b>68.3%</b>	<b>67.3%</b>	<b>62.8%</b>

*Πίνακας 5*

Αν στηριχτούμε όμως στην ικανότητα των συνελκτικών δικτύων, που εξάγουν μόνα τους χαρακτηριστικά, προκύπτουν τα παρακάτω αποτελέσματα

Παρακάτω βλέπουμε τα αποτελέσματα που έχουν προκύψει, μόνο με μετατροπή της εικόνας σε γκρι. Έχουμε πολύ καλά αποτελέσματα στην πρώτη στήλη το μοντέλο έφτασε 87.5%

Pixels: (150,150) epochs:30, drop=0.2, batch size=16 75-25 <b>Convert to gray</b>	Pixels: (150,150) epochs:30, drop=0.2, batch size=16 70-30 <b>Convert to gray</b>	Pixels: (150,150) epochs:30, drop=0.5, batch size=16 75-25 <b>Convert to gray</b>	Pixels: (150,150) epochs:30, drop=0.7, batch size=16 75-25 <b>Convert to gray</b>
96%	91.6%	31.6%	65%
86.6%	93%	95%	70%
89.3%	84.72%	90%	25%
86.6%	68%	80%	31.6%
91.6%	91.6%	86.6%	70%
81.6%	90.27%	63.3%	56.6%
90%	90.27%	61.6%	70%
93.3%	88.88%	21.6%	36.6%
71.6%	79.16%	63.3%	63.3%
88.3%	65.27%	28.3%	80%
<b>Average:87.49%</b>	<b>Average:84.42%</b>	<b>Average:62.13%</b>	<b>Average:56.81%</b>

Πίνακας 6

Τέλος, βλέπουμε τα αποτελέσματα που προέκυψαν όταν μετά από μετατροπή σε γκρι κάναμε και blur, με kernel 2x2. Μόνο στην τελευταία στήλη προέκυψαν καλά αποτελέσματα, το οποίο μπορεί να οφείλεται στην τιμή του drop rate, που είναι 0.2, καθώς και από τα προηγούμενα αποτελέσματα βλέπουμε ότι είναι το καλύτερο. Επίσης ο αριθμός των epochs είναι κατά 5 μικρότερη, οπότε πιθανόν με 30 epochs να γίνεται overfitting.

Pixels: (150,150) epochs:30, drop=0.5, batch size=16 75-25 <b>blur</b>	Pixels: (150,150) epochs:30, drop=0.7, batch size=16 75-25 <b>blur</b>	Pixels: (150,150) epochs:25, drop=0.2, batch size=16 75-25 <b>blur</b>
35%	26.6%	86.6%
23.3%	61.6%	43%
30%	56.6%	58.3%
31.6%	75%	91.6%
30%	66.6%	90%
63.3%	60%	83.3%
33.3%	66.6%	81.6%
45%	66.6%	90%
38.3%	65%	96.6%
66.6%	60%	91.6%
<b>Average:39.64%</b>	<b>Average:60.46%</b>	<b>Average:81.26%</b>

Πίνακας 7



## 7.Σύνοψη-Συμπεράσματα

Σύμφωνα με την παραπάνω διερεύνηση μεθοδολογιών σε πραγματικά δεδομένα προέκυψαν τα εξής συμπεράσματα:

Για το πρώτο μέρος της πτυχιακής είχαμε τέσσερα διαφορετικά δεδομένα, τα οποία αφορούσαν ιατρικά δεδομένα οστεοαρθρίτιδας. Το πρώτο σετ δεδομένων είχε 1122 καταχωρήσεις και η πιο υψηλή απόδοση όσον αφορά την ακρίβεια ταξινόμησης, που φτάσαμε είναι 90% χρησιμοποιώντας τον αλγόριθμο, από την κατηγορία Fuzzy Instance τον PosIBL-C. Δεύτερη καλύτερη απόδοση ήταν 80% επίσης από την κατηγορία Fuzzy Instance με τον αλγόριθμο FRNN-FRS-C και έφτασε η ίδια απόδοση με νευρωνικά δίκτυα, που αλλάξαμε την παράμετρο neurons rate σε 0.5

Όσον αφορά το δεύτερο dataset, το ολόκληρο με τις 4796 καταχωρήσεις η απόδοση κατάφερε να φτάσει το ποσοστό 98% με τον αλγόριθμο C45-Rules από την κατηγορία Crisp Rule Learning και παρατηρούμε ότι οι ασαφείς αλγόριθμοι δεν έδωσαν ιδιαίτερα ικανοποιητικά αποτελέσματα σε τόσο μεγάλο όγκο δεδομένων.

Στη συνέχεια έχουμε το τρίτο dataset, το οποίο έχει 1727 καταχωρήσεις και καταφέραμε με το keel να επιτύχουμε 99% απόδοση με τον αλγόριθμο class-mean της crisp κατηγορίας. Το επόμενο καλύτερο ποσοστό είναι 73% με τους αλγορίθμους easyboost και adaboost, από την κατηγορία ensemble.

Το τελευταίο dataset της οστεοαρθρίτιδας με 3026 καταχωρήσεις έδωσε ποσοτό ταξινόμησης 93% με τον Frnn-FRS-C, από fuzzy instance.

Όπως καταλαβαίνουμε οι ασαφείς αλγόριθμοι έβγαλαν καλά αποτελέσματα, δηλαδή μπόρεσαν να διαχειριστούν τα μη ισορροπημένα δεδομένα στα δύο από τα τέσσερα διαφορετικά σετ δεδομένων.

Περνάμε στο δεύτερο μέρος της πτυχιακής, όπου ασχοληθήκαμε με την ανάλυση σπινθηρογραφικών εικόνων, από ασθενείς, που υποβλήθηκαν σε σπινθηρογράφημα οστού. Με τη χειροκίνητη εξαγωγή χαρακτηριστικών με τη βοήθεια της Python και σύμφωνα με κάποιες τιμές στις παραμέτρους φτάσαμε 85% υψηλότερη απόδοση στο συνελκτικό μοντέλο. Όμως με διαφοροποίηση των τιμών στις παραμέτρους πέφτει αρκετά η απόδοση.

Αντιθέτως, η θεωρία των συνελκτικών νευρωνικών δικτύων αναφέρει ότι είναι σωστό να κάνουν μόνα τους εξαγωγή χαρακτηριστικών, καθώς για αυτό έχουν προγραμματιστεί.

Όποτε δοκιμάσαμε και αυτή την περίπτωση με τη μόνη προεπεξεργασία, μετατροπή σε γκρι την φωτογραφία και έπειτα ακολούθησε η μέθοδο του θολώματος σε διαφορετική περίπτωση για αφαίρεση θορύβου, για να δούμε τα αποτελέσματα.

Παρατηρήσαμε ότι μόνο με μετατροπή σε γκρι είχε πολύ ικανοποιητικά αποτελέσματα όσον αφορά την ακρίβεια της ταξινόμησης, αποτελέσματα στις περισσότερες περιπτώσεις που έδωσαν 87,5% απόδοση. Με διαφοροποίηση στο χώρισμα των δεδομένων για εκπαίδευση και έλεγχο η απόδοση μειώθηκε ελάχιστα και έφτασε στο 84%. Επίσης όσον αφορά τη μετέπειτα προεπεξεργασία με τη θόλωση καταφέραμε και εκεί απόδοση 81% με τη σωστή διαχείριση των παραμέτρων στη διαδικασία της εκπαίδευσης.

Οπότε συνολικά η καλύτερη απόδοση προέκυψε όταν για προεπεξεργασία ακολουθήσαμε μόνο τη μέθοδο της μετατροπής σε γκρι, αλλά παράλληλα όμως είδαμε ότι με την κατάλληλη προσαρμογή των παραμέτρων σε ένα συνελκτικό νευρωνικό δίκτυο μπορούμε να πλησιάσουμε την τόσο υψηλή απόδοση, που κατάφεραν να επιτύχουν μόνα τους.

Η ερευνητική προσπάθεια για ανάλυση σπινθηρογραφικών εικόνων με Convolutional Neural Networks, συνεχίζεται με στόχο τη διερεύνηση νέων παραμέτρων και τεχνικών για την αύξηση των ποσοστών ταξινόμησης και διάγνωσης.

## 8.Βιβλιογραφία

1. Johnny Dang “Master Thesis about classification in Bone Scintigraphy Images Using Convolutional Neural Networks”, June 2016
2. Eli Gibson, Wenqi Li, Carole Sudre, Lucas Fidon, Dzhoshkun Shakir, Guotai Wang, Zach Eaton-Rosen, Robert Gray, Tom Doel, Yipeng Hu, Tom Whyntie, Parashkev Nachev, Marc Modat, Dean Barratt, Sébastien Ourselin, M. Jorge Cardoso, Tom Vercautere “NiftyNet: a deep-learning platform for medical imaging”, January 2018
3. Geert Litjens, Thijs Kooi, Babak Ehteshami, Bejnordi Arnaud, Arindra Adiyoso, Setio Francesco, Ciompi, Mohsen Ghafourian, Jeroen A.W.M. van der Laak, Bram van Ginneken, Clara I. Sánchez “A survey on deep learning in medical image analysis”, July 2017
4. Luka Sajn, Igor Kononenko, Metka Milcinski “Computerized segmentation and diagnostics of whole-body scintigrams”, June 2007
5. Konrad Gjertsson “Segmentation in Skeletal Scintigraphy Images using Convolutional Neural Networks”, June 2017
6. Renato Campanini, Markus Wenzel “A deep learning approach to bone segmentation in CT scans”, 2016
7. <https://el.wikipedia.org/wiki/%CE%9A%CE%B1%CF%84%CE%B7%CE%B3%CE%B%CF%81%CE%B9%CE%BF%CF%80%CE%BF%CE%AF%CE%B7%CF%83%CE%B7>
8. [https://repository.kallipos.gr/bitstream/11419/1236/2/Kef.\\_9.pdf](https://repository.kallipos.gr/bitstream/11419/1236/2/Kef._9.pdf)
9. <https://towardsdatascience.com/supervised-machine-learning-classification-5e685fe18a6d>
10. [https://el.wikipedia.org/wiki/%CE%A3%CF%84%CE%B1%CF%84%CE%B9%CF%83%CF%84%CE%B9%CE%BA%CE%AE\\_%CF%84%CE%B1%CE%BE%CE%B9%CE%BD%CF%8C%CE%BC%CE%B7%CF%83%CE%B7](https://el.wikipedia.org/wiki/%CE%A3%CF%84%CE%B1%CF%84%CE%B9%CF%83%CF%84%CE%B9%CE%BA%CE%AE_%CF%84%CE%B1%CE%BE%CE%B9%CE%BD%CF%8C%CE%BC%CE%B7%CF%83%CE%B7)
11. <https://medium.com/@Mandysidana/machine-learning-types-of-classification-9497bd4f2e14>
12. <https://developers.google.com/machine-learning/crash-course/classification/accuracy>
13. [https://en.wikipedia.org/wiki/Confusion\\_matrix](https://en.wikipedia.org/wiki/Confusion_matrix)
14. <http://ai.uom.gr/Courses/AdvancedNeuralNetworks/Material/FuzzyLogic.pdf>
15. [https://en.wikipedia.org/wiki/Fuzzy\\_logic](https://en.wikipedia.org/wiki/Fuzzy_logic)
16. [https://science.fandom.com/el/wiki/%CE%91%CF%83%CE%B1%CF%86%CE%AE%CF%82\\_%CE%9B%CE%BF%CE%B3%CE%B9%CE%BA%CE%AE](https://science.fandom.com/el/wiki/%CE%91%CF%83%CE%B1%CF%86%CE%AE%CF%82_%CE%9B%CE%BF%CE%B3%CE%B9%CE%BA%CE%AE)

17. [https://www.biomedware.com/files/documentation/boundaryseer/Fuzzy\\_classification/About\\_fuzzy\\_classification.htm](https://www.biomedware.com/files/documentation/boundaryseer/Fuzzy_classification/About_fuzzy_classification.htm)
18. <https://en.wikipedia.org/wiki/MapReduce>
19. <https://eventos.citius.usc.es/evia2017/presentations/EVIA2017%20-%20Thursday%20-%202003%20-%20Francisco%20Herrera%20-%20Fuzzy%20Systems%20in%20Data%20Science%20and%20Big%20Data.pdf>
20. <https://pdfs.semanticscholar.org/3d9d/f28adacbcd7b090ea087d67871acb726d206.pdf>
21. <https://el.wikipedia.org/wiki/%CE%9F%CF%83%CF%84%CE%B5%CE%BF%CE%B1%CF%81%CE%B8%CF%81%CE%AF%CF%84%CE%B9%CE%B4%CE%B1>
22. <https://oai.epi-ucsf.org/datarelease/docs/presentations/acr102008/MNACR2008.pdf>
23. <https://machinelearningmastery.com/handle-missing-data-python/>
24. <https://sci2s.ugr.es/keel/imbanced.php>
25. <https://sci2s.ugr.es/keel/development.php>
26. <http://www.keel.es/download.php>
27. <https://sci2s.ugr.es/keel/datasets.php>
28. <https://archive.ics.uci.edu/ml/index.php>
29. <https://blog.vrisko.gr/ab-ygeias/spinthirografima>
30. [https://en.wikipedia.org/wiki/Deep\\_learning](https://en.wikipedia.org/wiki/Deep_learning)
31. <https://www.mathworks.com/discovery/deep-learning.html>
32. <https://skymind.ai/wiki/neural-network>
33. [https://en.wikipedia.org/wiki/Convolutional\\_neural\\_network](https://en.wikipedia.org/wiki/Convolutional_neural_network)
34. <https://medium.com/@RaghavPrabhu/understanding-of-convolutional-neural-network-cnn-deep-learning-99760835f148>
35. <https://towardsdatascience.com/activation-functions-neural-networks-1cbd9f8d91d6>
36. <https://becominghuman.ai/what-exactly-does-cnn-see-4d436d8e6e52>
37. <https://keras.io/getting-started/sequential-model-guide/>
38. <https://keras.io/preprocessing/image/>
39. <https://towardsdatascience.com/building-a-convolutional-neural-network-cnn-in-keras-329fbbadc5f5>
40. <https://machinelearningmastery.com/dropout-for-regularizing-deep-neural-networks/>
41. <https://www.youtube.com/watch?v=bqeUmLCgsVw>
42. [https://github.com/risenW/medium\\_tutorial\\_notebooks/blob/master/dogs-vs-cats-keras-implementation.ipynb](https://github.com/risenW/medium_tutorial_notebooks/blob/master/dogs-vs-cats-keras-implementation.ipynb)

43. [https://docs.opencv.org/2.4/doc/tutorials/imgproc/erosion\\_dilatation/erosion\\_dilatation.html](https://docs.opencv.org/2.4/doc/tutorials/imgproc/erosion_dilatation/erosion_dilatation.html)
44. [https://docs.opencv.org/3.4.3/d7/d4d/tutorial\\_py\\_thresholding.html](https://docs.opencv.org/3.4.3/d7/d4d/tutorial_py_thresholding.html)
45. [https://docs.opencv.org/3.1.0/d4/d13/tutorial\\_py\\_filtering.html](https://docs.opencv.org/3.1.0/d4/d13/tutorial_py_filtering.html)
46. <https://towardsdatascience.com/applied-deep-learning-part-4-convolutional-neural-networks-584bc134c1e2>

## 9. Παράρτημα

Πρώτα βλέπουμε τον κώδικα που μας βοήθησε να αντιμετωπίσουμε τις τιμές που έλειπαν για τα δεδομένα της οστεοαρθρίτιδας.

Κάνουμε εισαγωγή το excel αρχείο και στο τέλος του προγράμματος δημιουργείται ένα άλλο excel αρχείο, στο οποίο έχουν αντικατασταθεί οι τιμές που λείπουν με το μέσο όρο της στήλης.

```
import numpy as np

import pandas as pd

dataset=pd.read_excel('dataset_all.xls')

y = dataset.iloc[:, 191].values

from sklearn.preprocessing import LabelEncoder

labelencoder_y = LabelEncoder()

y = labelencoder_y.fit_transform(y)

dataset=dataset.where(pd.notna(dataset), dataset.mean(), axis='columns')

#write new excel file

dataset.to_excel('dataset_al.xls')
```

Παρακάτω βλέπουμε τον κώδικα που χρησιμοποιήθηκε στο δεύτερο μέρος της πτυχιακής.

```
import os

import cv2

import numpy as np

import matplotlib.pyplot as plt

import random

import glob

path = '*.jpg'
```

```

# read addresses and labels from the 'train' folder
addrs = glob.glob(path)

y = [0 if 'normal' in addr else 1 for addr in addrs]
# to shuffle data
c = list(zip(addrs, y))
random.shuffle(c)
addrs, y = zip(*c)

nrows = 150
ncolumns = 150
channels = 1 # if you want to use grayscale image
kernel = np.ones((10,10), np.uint8)
def read_and_process_image(list_of_images):
    X=[]
    for img in addrs:
        #print(img)
        image = cv2.imread(img)

        """
        plt.imshow(image)
        plt.title("img")
        plt.show()
        """

        gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)

        #cv2.imwrite('gray_image.png',gray_image)
        #cv2.imshow('color_image',image)
        #cv2.imshow('gray_image',gray_image)
        #cv2.waitKey(0)
        #cv2.destroyAllWindows()

```

```

blur = cv2.blur(gray,(2,2))
"""

plt.imshow(blur)
plt.title("smooth")
plt.show()
"""

img_erosion = cv2.erode(blur, kernel, iterations=7)
"""

plt.imshow(img_erosion)
plt.title("erode")
plt.show()
"""

ret,thresh= cv2.threshold(img_erosion,40,255,cv2.THRESH_BINARY)
"""

plt.imshow(thresh)
plt.title("binary")
plt.show()
"""

X.append(cv2.resize(thresh,(nrows,ncolumns),
interpolation=cv2.INTER_CUBIC))
y = [0 if 'normal' in addr else 1 for addr in addrs]

return X,y

```

```
X,y= read_and_process_image(addrs)
```

```
X = np.array(X)
```

```
y = np.array(y)
```

```
from sklearn.model_selection import train_test_split
```



```

X_train, X_val, y_train, y_val = train_test_split(X, y, test_size=0.25)

#get the length of the train and validation data
ntrain = len(X_train)
nval = len(X_val)

#Add one dimension to X_train, X_val
X_train = np.expand_dims(X_train,axis=1)
X_val = np.expand_dims(X_val,axis=1)

#Move the dimension to the right position
X_train = np.moveaxis(X_train,1,3)
X_val = np.moveaxis(X_val,1,3)
#We will use a batch size of 16. Note: batch size should be a factor of
2.***4,8,16,32,64...***
batch_size = 16

# Set the dropout regularization
drop_rate = 0.2

from keras import layers
from keras.models import Sequential
from keras.layers import Conv2D, MaxPooling2D, Flatten, Dense, Dropout

model = Sequential()
model.add(layers.Conv2D(32, (3, 3), activation='relu',input_shape=(150,150,1)))
model.add(layers.MaxPooling2D((2, 2)))
model.add(layers.Dropout(drop_rate))
model.add(layers.Conv2D(64, (3, 3), activation='relu'))
model.add(layers.MaxPooling2D((2, 2)))
model.add(layers.Dropout(drop_rate))

```

```

model.add(layers.Conv2D(128, (3, 3), activation='relu'))
model.add(layers.MaxPooling2D((2, 2)))
model.add(layers.Dropout(drop_rate))
model.add(layers.Conv2D(128, (3, 3), activation='relu'))
model.add(layers.MaxPooling2D((2, 2)))
model.add(layers.Dropout(drop_rate))
model.add(layers.Flatten())
model.add(layers.Dense(512, activation='relu'))
model.add(layers.Dropout(0.5)) #Dropout for regularization
model.add(layers.Dense(1, activation='sigmoid'))
model.summary()

#We'll use binary_crossentropy loss because its a binary classification
model.compile(loss='binary_crossentropy', optimizer='adam', metrics=['accuracy'])

#model.fit(X_train, y_train, batch_size=2, epochs=24)
#score = model.evaluate(X_val, y_val, batch_size=2)
#print(score)

from keras.preprocessing.image import ImageDataGenerator
#Lets create the augmentation configuration
#This helps prevent overfitting, since we are using a small dataset
train_datagen = ImageDataGenerator(rescale=1./255, #Scale the image between 0 and 1
                                   rotation_range=40,
                                   width_shift_range=0.2,
                                   height_shift_range=0.2,
                                   shear_range=0.2,
                                   zoom_range=0.2,
                                   horizontal_flip=True)

val_datagen = ImageDataGenerator(rescale=1./255) #We do not augment validation data. we
only perform rescale

```

```

#Create the image generators
train_generator = train_datagen.flow(X_train, y_train, batch_size=batch_size)
val_generator = val_datagen.flow(X_val, y_val, batch_size=batch_size)

#The training part
history = model.fit_generator(train_generator,
                              steps_per_epoch=ntrain // batch_size,
                              epochs=30,
                              validation_data=val_generator,
                              validation_steps=nval // batch_size)

# Predicting the Test set results
y_pred = model.predict(X_val)
#y_pred = (y_pred > 0.5)

# Making the Confusion Matrix
from sklearn.metrics import confusion_matrix
cm = confusion_matrix(y_val, y_pred.astype(int))
print(cm)

# accuracy score
from sklearn.metrics import accuracy_score
a=accuracy_score(y_val, y_pred.astype(int))
print(a*100)
"""

# classification report
from sklearn.metrics import classification_report
cr = classification_report(y_val, y_pred.astype(int))
print(cr)
"""

```