



ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ

ΤΜΗΜΑ: ΨΗΦΙΑΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

ΚΑΤΕΥΘΥΝΣΗ: ΨΗΦΙΑΚΕΣ ΕΠΙΚΟΙΝΩΝΙΕΣ ΚΑΙ ΔΙΚΤΥΑ

ΤΙΤΛΟΣ: ΒΕΛΤΙΣΤΟΠΟΙΗΣΗ ΔΙΑΧΕΙΡΙΣΗΣ ΔΙΚΤΥΩΝ ΜΕ ΧΡΗΣΗ ΤΕΧΝΙΚΩΝ ΜΗΧΑΝΙΚΗΣ ΜΑΘΗΣΗΣ

ΟΝΟΜΑΤΕΠΩΝΥΜΟ: ΔΕΛΙΤΖΙΑΣ ΙΩΑΝΝΗΣ

Α.Μ: ΜΕ14037



ΕΥΧΑΡΙΣΤΙΕΣ

Στο σημείο αυτό θα ήθελα να ευχαριστήσω τους επιβλέποντες καθηγητές κύριο Άγγελο Ρούσκα και κύριο Κώστα Τσαγκάρη για την βοήθεια που μου παρείχαν, καθ' όλη τη διάρκεια εκπόνησης της εργασίας, καθώς και για την υποστήριξη και άψογη συνεργασία που είχαμε.

Επίσης θα ήθελα να ευχαριστήσω και την οικογένεια μου, για τη στήριξη και την κατανόηση σε όλα τα χρόνια των σπουδών μου.

Ακόμη θα ήθελα να ευχαριστήσω και τους συμφοιτητές μου για την άριστη συνεργασία, που είχαμε κατά τη διάρκεια του μεταπτυχιακού προγράμματος σπουδών.

ΠΕΡΙΛΗΨΗ

Η παρούσα εργασία αποτελείται ουσιαστικά από τρία επιμέρους κομμάτια. Πιο αναλυτικά :

Το κεφάλαιο 1 περιλαμβάνει τον ορισμό του mobility prediction που είναι και το ζητούμενο της εργασίας. Ακόμη δίνονται κάποια εισαγωγικά στοιχεία για τη μηχανική μάθηση και γενικότερα για τα συστήματα διαχείρισης δικτύων.

Τα κεφάλαια 2 έως 5 αποτελούν ένα tutorial του περιβάλλοντος που χρησιμοποιήθηκε για την εκπόνηση της εργασίας. Περιγράφονται όλες οι σημαντικότερες λειτουργίες, οι αλγόριθμοι και μέθοδοι, καθώς και το καθένα ξεχωριστά από τα τέσσερα περιβάλλοντα του WEKA.

Το κεφάλαιο 6 περιλαμβάνει την αντιμετώπιση του προβλήματος όπως ορίστηκε στο πρώτο κεφάλαιο με τη βοήθεια του WEKA.

Το κεφάλαιο 7 περιλαμβάνει όλα τα συμπεράσματα που προέκυψαν κατά την εκπόνηση της εργασίας, καθώς και διαπιστώσεις που προέκυψαν, την αντιμετώπιση του προβλήματος.

Τέλος το κεφάλαιο 8 περιλαμβάνει τη βιβλιογραφία που χρησιμοποιήθηκε, επιστημονικά κείμενα και πηγές του διαδικτύου.

ΠΕΡΙΕΧΟΜΕΝΑ

1. Εισαγωγή στη μηχανική μάθηση(machine learning).....	6-
1.1 Εισαγωγή στο WEKA.....	8-
1.2 Επεξεργασία και συλλογή δεδομένων.....	9-
1.3 Διαχείριση δικτύου, συστήματα και απαιτήσεις.....	11-
1.4 Mobility prediction, ορισμός και χαρακτηριστικά...	15-
2. WEKA Explorer.....	18-
2.1 Preprocess.....	19-
2.2 Classify.....	22-
2.3 Cluster.....	26-
2.4 Associate.....	28-
2.5 Select.....	29-
2.6 Visualize.....	31-
3. WEKA Experimenter.....	32-
3.1 Simple Mode.....	32-
3.2 Advanced Mode.....	38-
4. WEKA Knowledge Flow.....	39-
4.1 Classifiers και Features.....	39-
4.2 Knowledge Flow Components.....	40-
4.3 Παράδειγμα χρήσης.....	43-

5. WEKA Command Line (CLI).....	-45-
5.1 Classify με τη χρήση εντολών.....	-47-
6. Markov based prediction model.....	-49-
6.1 Εισαγωγή στις αλυσίδες Markov.....	-49-
6.2 Χρήση του WEKA για την κατασκευή του Bayesian Based prediction model.....	-52-
7. Συμπεράσματα.....	-62-
8. Βιβλιογραφία-πηγές.....	-66-

1. Εισαγωγή στη μηχανική μάθηση(machine learning)

Η μηχανική μάθηση διερευνά τη μελέτη και την κατασκευή αλγορίθμων, που μπορούν να μαθαίνουν από τα δεδομένα και να κάνουν προβλέψεις σχετικά με αυτά. Τέτοιοι αλγόριθμοι λειτουργούν κατασκευάζοντας μοντέλα από πειραματικά δεδομένα, προκειμένου να κάνουν προβλέψεις βασιζόμενες στα δεδομένα, ή να εξάγουν αποφάσεις που εκφράζονται ως το αποτέλεσμα.

Στο πεδίο της ανάλυσης δεδομένων, η μηχανική μάθηση είναι μια μέθοδος που χρησιμοποιείται για την επινόηση πολύπλοκων μοντέλων και αλγορίθμων που οδηγούν στην πρόβλεψη. Τα αναλυτικά μοντέλα επιτρέπουν στους ερευνητές, τους επιστήμονες δεδομένων, τους μηχανικούς και τους αναλυτές να παράγουν αξιόπιστες αποφάσεις και αποτελέσματα και να αναδείξουν αλληλοσυσχετίσεις, μέσω της μάθησης από ιστορικές σχέσεις και τάσεις στα δεδομένα.

Οι εργασίες μηχανικής μάθησης συνήθως ταξινομούνται σε τρεις μεγάλες κατηγορίες, ανάλογα με τη φύση του εκπαιδευτικού σήματος, ή την ανατροφοδότηση, που είναι διαθέσιμη σε ένα σύστημα εκμάθησης. Αυτές είναι:

- **Επιτηρούμενη μάθηση** (επιβλεπόμενη μάθηση ή μάθηση με επίβλεψη, **supervised learning**): Το υπολογιστικό πρόγραμμα δέχεται τις παραδειγματικές εισόδους καθώς και τα επιθυμητά αποτελέσματα από έναν δάσκαλο και ο στόχος είναι να μάθει έναν γενικό κανόνα, προκειμένου να αντιστοιχίσει τις εισόδους με τα αποτελέσματα. Για παράδειγμα πρόβλεψη τιμής ενός σπιτιού προς πώληση, με βάση τις τιμές πωλήσεων γειτονικών σπιτιών.
- **Μη επιτηρούμενη μάθηση**(μάθηση χωρίς επίβλεψη, **unsupervised learning**): Χωρίς να παρέχεται κάποια εμπειρία στον αλγόριθμο μάθησης, πρέπει να βρει την δομή των δεδομένων εισόδου. Η μη επιτηρούμενη μάθηση μπορεί να είναι αυτοσκοπός, ανακαλύπτοντας κρυμμένα μοτίβα σε

δεδομένα) ή μέσο για ένα τέλος, χαρακτηριστικό της μάθησης. Για παράδειγμα η πρόβλεψη των δημογραφικών στοιχείων των κατοίκων μιας περιοχής

- **Ενισχυτική μάθηση:** ένα πρόγραμμα υπολογιστή αλληλεπιδρά με ένα δυναμικό περιβάλλον στο οποίο πρέπει να επιτευχθεί ένας συγκεκριμένος στόχος, όπως η οδήγηση ενός οχήματος, χωρίς κάποιος δάσκαλος να του λέει ρητά αν έχει φτάσει κοντά στο στόχο του. Ένα άλλο παράδειγμα είναι να μάθει να παίζει ένα παιχνίδι εναντίον κάποιου αντιπάλου.

Μια άλλη κατηγοριοποίηση των προβλημάτων μηχανικής μάθησης προκύπτει όταν κάποιος θεωρήσει το επιθυμητό αποτέλεσμα του συστήματος μηχανικής μάθησης.

- Στην **ταξινόμηση(classify)**, τα δεδομένα εισόδου χωρίζονται σε δύο ή περισσότερες κλάσεις, και η μηχανή πρέπει να κατασκευάσει ένα μοντέλο, το οποίο θα αντιστοιχίζει τα δεδομένα σε μία ή περισσότερες κλάσεις. Αυτό συνήθως εμπίπτει στην επιτηρούμενη μάθηση. Τα φίλτρα spam είναι ένα παράδειγμα ταξινόμησης, όπου οι εισοδοί είναι τα e-mails ή άλλα μηνύματα και οι κλάσεις είναι spam και όχι spam.
- Στην **παλινδρόμηση(regression)**, επίσης πρόβλημα επιτηρούμενης μάθησης, τα αποτελέσματα είναι συνεχή και όχι διακριτά.
- Στη **συσταδοποίηση(clustering)** ένα σύνολο εισόδων πρόκειται να χωριστεί σε ομάδες. Σε αντίθεση με την ταξινόμηση, οι ομάδες δεν είναι γνωστές εκ των προτέρων, καθιστώντας αυτόν τον διαχωρισμό τυπική εργασία μη επιτηρούμενης μάθησης.
- Στην εκτίμηση πυκνότητας βρίσκει την κατανομή των δεδομένων εισόδου σε κάποιο χώρο.

Η μηχανική μάθηση βρίσκει εφαρμογές σε διάφορους τομείς και επιστήμες. Χαρακτηριστικά παραδείγματα είναι η βελτιστοποίηση, η ανάκτηση πληροφορίας, το μάρκετινγκ, η οικονομία, οι μηχανές αναζήτησης, τα ηλεκτρονικά παιχνίδια, η διαχείριση δικτύων και άλλα.

1.1 Εισαγωγή στο WEKA

Το **WEKA (Waikato Environment for Knowledge Analysis)** είναι μια δημοφιλής πλατφόρμα μηχανικής μάθησης, γραμμένη σε γλώσσα προγραμματισμού Java. Αναπτύχθηκε στο πανεπιστήμιο του Waikato στην Νέα Ζηλανδία το 1993. Τα χαρακτηριστικά που το κάνουν τόσο δημοφιλές στους χρήστες του σε σχέση με τα αντίστοιχα λογισμικά μηχανικής μάθησης, είναι ότι βρίσκεται δωρεάν στο διαδίκτυο, τρέχει σε όλα τα σημερινά λειτουργικά συστήματα, παρέχει μια ευρεία γκάμα επιλογών για επεξεργασία και κατασκευή μοντέλων μάθησης και τέλος, παρέχει 4 περιβάλλοντα εργασίας που προσφέρουν εύκολη χρήση και εξοικείωση στους χρήστες.

Συγκεκριμένα το WEKA, παρέχει μια μεγάλη συλλογή από εργαλεία προβολής αποτελεσμάτων και διάφορα είδη αλγορίθμων. Τα εργαλεία χρησιμοποιούνται σε όλες τις βασικές λειτουργίες του, όπως συλλογή και επεξεργασία δεδομένων, ομαδοποίηση, ταξινόμηση, προβολή των αποτελεσμάτων με τρόπο κατανοητό στους χρήστες και επιλογή των χαρακτηριστικών που τους ενδιαφέρουν.

Οι βασικές λειτουργίες του WEKA ικανοποιούνται πλήρως από τα 4 περιβάλλοντα εργασίας του, τα οποία αναφέρονται παρακάτω:

- **WEKA Explorer:** Το περιβάλλον εργασίας που χρησιμοποιούν περισσότερο οι χρήστες, γιατί εκτελούν όλες τις βασικές λειτουργίες βήμα-βήμα. Συνίσταται για μεμονωμένα πειράματα.
- **WEKA Experimenter:** Χρησιμοποιείται από τους χρήστες, για την σύγκριση διάφορων μοντέλων μηχανικής μάθησης, ώστε να επεξεργαστούν με αποδοτικότερο τρόπο τα δεδομένα τους.
- **WEKA Knowledge Flow:** Χρησιμοποιείται από τους χρήστες, για την αναπαράσταση μέσω γραφημάτων, της ροής της γνώσης, που εξάγουν μέσω των αλγορίθμων που εφαρμόζουν στα δεδομένα τους.

- **WEKA Command Line:** Παρέχει την ίδια λειτουργικότητα με τον Explorer και τον Experimenter, αλλά χρησιμοποιείται λιγότερο, γιατί δεν είναι εύκολη η διαχείριση προγραμμάτων, που απαιτούν εφαρμογή πολλών αλγορίθμων και σύγκριση πολλών μοντέλων μηχανικής μάθησης.

1.2 Επεξεργασία και συλλογή δεδομένων

Τα δεδομένα που χρησιμοποιήθηκαν στην εργασία είναι της μορφής SNMP, που συλλέχθηκαν με τη χρήση αισθητήρων και αφορούν τιμές του wi-fi δικτύου, ενός αεροδρομίου. Πιο συγκεκριμένα χρησιμοποιήθηκε το πρωτόκολλο 802.11 καλύπτοντας πλήρως όλους τους clients, που μπορεί να είναι της μορφής (802.11b, 802.11g, 802.11n).

Χρησιμοποιήθηκε επίσης το λογισμικό Mib-Browser για την επεξεργασία των δεδομένων, ώστε να οριστεί η σημασία τους και να μετατραπούν από raw data, σε tabular data, δηλαδή δεδομένα που εμφανίζουν ακριβή πληροφορία και μπορούν να χρησιμοποιηθούν για την κατασκευή μοντέλων μηχανικής μάθησης.

Για τη συλλογή των δεδομένων χρησιμοποιήθηκαν οι εντολές της SNMP. Χαρακτηριστικά οι εντολές είναι SNMP GET και SNMP GET-NEXT. Ειδικότερα έγινε αντιστοίχιση των τιμών που συλλέχθηκαν με τις μεταβλητές που παρέχει η Mib. Τα δεδομένα αφορούν τις τιμές του γκρουπ interfaces από τη λίστα των αντικειμένων που παρέχει η Mib. Ακολουθούν αναλυτικότερα ορισμοί των μεταβλητών που αφορούν τα δεδομένα που χρησιμοποιήθηκαν στην παρούσα εργασία.

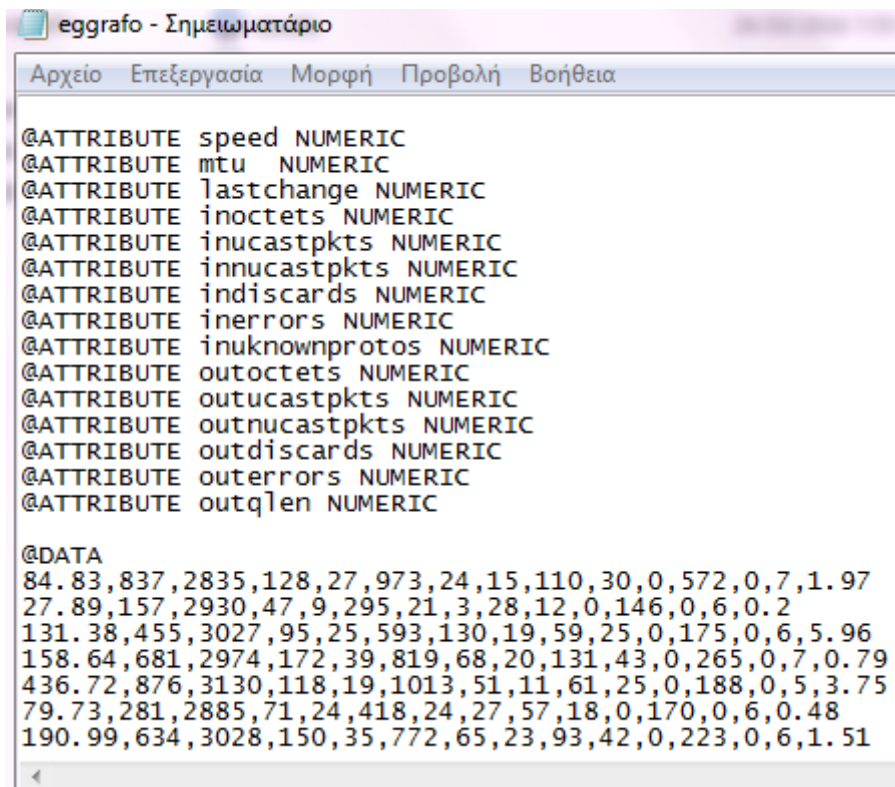
- **Mtu:** το μέγεθος του μεγαλύτερου διαγράμματος IP διευθύνσεων που μπορεί να εξυπηρετηθεί από τη συγκεκριμένη διεπαφή. Μετριέται σε οκτάδες από bit, τα λεγόμενα octets.
- **Speed:** Υπολογισμός του bandwidth της συγκεκριμένης διεπαφής. Μετριέται σε bits/sec.
- **LastChange:** Υπολογισμός του χρόνου που πέρασε, μέχρι η διεπαφή να μπει σε κατάσταση ενεργής λειτουργίας. Μετριέται σε seconds.

- **InOctets:** Ο αριθμός των octets, που λαμβάνει η συγκεκριμένη διεπαφή, συμπεριλαμβανομένου και δεδομένων σε μορφή πακέτου. Μετριέται σε octets.
- **InUcastPkts:** Ο αριθμός των unicast πακέτων που μεταφέρονται σε ένα πρωτόκολλο υψηλότερου επιπέδου. Μετριέται σε octets.
- **InNUcastPkts:** Ο αριθμός των non-unicast πακέτων(δηλαδή broadcast ή multicast πακέτων) που αποστέλλονται σε ένα πρωτόκολλο υψηλότερου επιπέδου. Μετριέται σε octets.
- **InDiscards:** Ο αριθμός των εισερχόμενων πακέτων που είναι πιθανό να απορριφθούν, παρά το γεγονός ότι δεν έχουν διαγνωστεί ως λανθασμένα. Μετριέται σε octets.
- **InErrors:** Ο αριθμός των εισερχόμενων πακέτων, που έχει διαπιστωθεί ότι είναι λανθασμένα και έτσι εμποδίζεται η λήψη τους. Μετριέται σε octets.
- **InUnknownProtos:** Ο αριθμός των πακέτων που απορρίφθηκαν από τη διεπαφή, λόγω ενός αγνώστου πρωτοκόλλου μεταφοράς, ή ενός που δεν μπορεί να υποστηριχθεί, από τη συγκεκριμένη διεπαφή. Μετριέται σε octets.
- **OutOctets:** Ο αριθμός των octets, που στέλνει η συγκεκριμένη διεπαφή, συμπεριλαμβανομένου και δεδομένων σε μορφή πακέτου. Μετριέται σε octets.
- **OutUcastPkts:** Ο αριθμός των unicast πακέτων, που ζητούν τα πρωτόκολλα υψηλότερου επιπέδου. Περιλαμβάνουν και πακέτα που πρόκειται να απορριφθούν, ή δεν έχουν σταλεί ακόμη. Μετριέται σε octets.
- **OutNUcastPkts:** Ο αριθμός των non-unicast πακέτων, που ζητούν τα πρωτόκολλα υψηλότερου επιπέδου. Περιλαμβάνουν και πακέτα που πρόκειται να απορριφθούν, ή δεν έχουν σταλεί ακόμη. Μετριέται σε octets.
- **OutDiscards:** Ο αριθμός των εξερχόμενων πακέτων που πρόκειται να απορριφθούν, παρά το γεγονός ότι δεν έχουν διαγνωστεί ως

λανθασμένα. Γίνεται κυρίως για την ελάφρυνση του χώρου αναμονής των πακέτων προς λήψη. Μετριέται σε octets.

- **OutErrors:** Ο αριθμός των εξερχόμενων πακέτων, που έχει διαπιστωθεί ότι είναι λανθασμένα και έτσι δεν γίνεται δυνατή η αποστολή τους. Μετριέται σε octets.
- **OutQLen:** Το μήκος της ουράς των εξερχόμενων πακέτων, προς αποστολή. Μετριέται σε octets.

Η εικόνα που ακολουθεί περιγράφει το αρχείο που δημιουργήθηκε για να εξυπηρετήσει τις ανάγκες της εργασίας.



```
eggrafo - Σημειωματάριο
Αρχείο Επεξεργασία Μορφή Προβολή Βοήθεια
@ATTRIBUTE speed NUMERIC
@ATTRIBUTE mtu NUMERIC
@ATTRIBUTE lastchange NUMERIC
@ATTRIBUTE inoctets NUMERIC
@ATTRIBUTE inucastpkts NUMERIC
@ATTRIBUTE innucastpkts NUMERIC
@ATTRIBUTE indiscards NUMERIC
@ATTRIBUTE inerrors NUMERIC
@ATTRIBUTE inunknownprotos NUMERIC
@ATTRIBUTE outoctets NUMERIC
@ATTRIBUTE outucastpkts NUMERIC
@ATTRIBUTE outnucastpkts NUMERIC
@ATTRIBUTE outdiscards NUMERIC
@ATTRIBUTE outerrors NUMERIC
@ATTRIBUTE outqlen NUMERIC
@DATA
84.83,837,2835,128,27,973,24,15,110,30,0,572,0,7,1.97
27.89,157,2930,47,9,295,21,3,28,12,0,146,0,6,0.2
131.38,455,3027,95,25,593,130,19,59,25,0,175,0,6,5.96
158.64,681,2974,172,39,819,68,20,131,43,0,265,0,7,0.79
436.72,876,3130,118,19,1013,51,11,61,25,0,188,0,5,3.75
79.73,281,2885,71,24,418,24,27,57,18,0,170,0,6,0.48
190.99,634,3028,150,35,772,65,23,93,42,0,223,0,6,1.51
```

Εικόνα 1.1

1.3 Διαχείριση δικτύου, συστήματα και απαιτήσεις

Τα τελευταία χρόνια τα δίκτυα υπολογιστών και τα συστήματα κατανεμημένης επεξεργασίας έχουν γνωρίσει μεγάλη ανάπτυξη. Η τάση στην ανάπτυξη των συστημάτων αυτών είναι προς την κατεύθυνση μεγαλύτερων και περισσότερο πολύπλοκων δικτύων τα οποία θα υποστηρίζουν περισσότερες εφαρμογές και περισσότερους χρήστες. Συνεπώς, έχει αυξηθεί σημαντικά η πιθανότητα να συμβεί κάποιο λάθος και έτσι ολόκληρο το δίκτυο ή ένα μέρος του να τεθεί εκτός λειτουργίας ή να μειωθεί η αξιοπιστία και η απόδοση του. Ειδικά σε ένα μεγάλο τοπικό δίκτυο (που μπορεί να έχει έκταση ενός μεγάλου κτιριακού συγκροτήματος ή ενός Πανεπιστημίου), η συντήρηση και ο έλεγχος του μπορεί να είναι μια διαδικασία ασύμφορη, επίπονη και χρονοβόρα, που απαιτεί να ασχοληθούν αρκετοί άνθρωποι.

Για τους παραπάνω λόγους, καθώς και η πολυπλοκότητα των δικτύων και η ύπαρξη συσκευών που ανήκουν σε διαφορετικούς κατασκευαστές, έχουν κάνει αναγκαία την ανάπτυξη εργαλείων που θα βοηθήσουν στην αυτόματη και αποτελεσματική διαχείριση των δικτύων. Έτσι έχουν αναπτυχθεί τα ανάλογα πρωτόκολλα και βάσεις διαχείρισης πληροφοριών καθώς και το αντίστοιχο λογισμικό το οποίο χρησιμοποιείται για να είναι εφικτή η διαχείριση του δικτύου. Με τον όρο <<Διαχείριση Δικτύου>> εννοούμε η διαδικασία του αυτόματου (ή όσο το δυνατόν αυτοματοποιημένου) ελέγχου ενός οποιουδήποτε δικτύου υπολογιστών ώστε το κόστος συντήρησης του να είναι κατά το δυνατόν μικρότερο και η απόδοση του η καλύτερη δυνατή. Γενικά, οι βασικοί σκοποί της διαχείρισης του δικτύου είναι οι εξής:

- Η διατήρηση της ικανοποιητικής και αξιόπιστης λειτουργίας ακόμη και κάτω από συνθήκες υπερφόρτωσης ή βλάβης, καθώς επίσης και κάτω από αλλαγές της διαμόρφωσης του δικτύου (εισαγωγή νέων συσκευών ή υπηρεσιών).
- Η βελτίωση της απόδοσης του δικτύου, η οποία σχετίζεται με την ποιότητα και την ποσότητα των υπηρεσιών που παρέχονται στους χρήστες.

Παρακάτω αναφέρονται τα επιμέρους στοιχεία από τα οποία αποτελείται ένα σύστημα διαχείρισης δικτύου.

- **Network Management Console:** Ο σταθμός εργασίας όπου παρακολουθεί ο διαχειριστής την κατάσταση του δικτύου.
- **Network Management Protocol:** Το πρωτόκολλο με το οποίο θα επικοινωνεί με τις δικτυακές συσκευές.

- **Network Management Agent:** Το software που εγκαθίσταται στην δικτυακή συσκευή για χρήση του πρωτοκόλλου διαχείρισης.

• **Δικτυακές συσκευές ή και που να τρέχουν agents.:** router, switches, servers, applications.

Ακολουθούν οι βασικές ενέργειες διαχείρισης δικτύου.

- Έλεγχος λειτουργίας δικτύου
- Διοίκηση δικτύου (σε τακτικό επίπεδο)
- Ανάλυση δικτύου και βελτιστοποίηση (τακτικό και στρατηγικό επίπεδο).
- Σχεδίαση δικτύου (στρατηγικό επίπεδο)
- Ο απομακρυσμένος έλεγχος και αναδιαμόρφωση δικτυακών συσκευών.
- Η παρακολούθηση του δικτύου μέσω των συσκευών που το συνθέτουν.
- Η σύνθεση βάσεων δεδομένων με το “ιστορικό” της δραστηριότητας του δικτύου.
- Η δυνατότητα τοποθέτησης παγίδων (traps) και συναγερμών (alarms) στις δικτυακές συσκευές.

Λαμβάνοντας υπόψη τις λειτουργίες που ένα διαχειριστικό σύστημα απαιτείται να υποστηρίζει, τα παρακάτω γενικά χαρακτηριστικά - περιορισμοί ενισχύουν την λειτουργικότητα του συστήματος:

Το σύστημα πρέπει να παρέχει ένα γραφικό σύστημα παρουσίασης της τοπολογίας του δικτύου. Είναι προτιμότερο η παρουσίαση να γίνεται με ιεραρχικό τρόπο και να υπάρχουν λογικές συνδέσεις μεταξύ των διαφορετικών επιπέδων της ιεραρχίας. Για παράδειγμα, σε ένα επίπεδο

παρουσιάζονται μόνο τα LANs και οι συνδέσεις μεταξύ τους, ενώ σε κατώτερο επίπεδο παρουσιάζονται τα τμήματα (**segments**) του κάθε LAN, στο επόμενο επίπεδο οι κόμβοι των segments κ.ο.κ. Πρέπει ακόμη το σύστημα να είναι σε θέση να αναγνωρίζει τις συνδέσεις μεταξύ των επιπέδων και το πως αυτές συσχετίζονται με την απόδοση και την λειτουργία ολόκληρου του δικτύου. Η ενοποιημένη εικόνα του διαχειριζόμενου δικτύου διατηρείται από το σύστημα, ενώ ο χρήστης μπορεί να επικεντρώνει την προσοχή του σε ορισμένα επίπεδα της ιεραρχίας.

Το σύστημα πρέπει να είναι ικανό να συλλέγει όλες τις πληροφορίες από τους διαχειριζόμενους κόμβους, με όσο είναι δυνατόν μεγαλύτερη διαφάνεια και ιδανικά μέσω ενός μόνο πρωτοκόλλου διαχείρισης. Βέβαια, σε ετερογενή περιβάλλοντα το σύστημα πρέπει να είναι σε θέση να χρησιμοποιεί διαφορετικά πρωτόκολλα διαχείρισης.

Η επεκτασιμότητα (**expandability**) και η δυνατότητα προσαρμογής σε διαφορετικές ανάγκες διαχείρισης (**customization**) είναι δύο ακόμη σημαντικά χαρακτηριστικά - απαιτήσεις. Δεν υπάρχει σύστημα που να καλύπτει τις ανάγκες διαχείρισης κάθε δυνατού δικτύου. Έτσι το σύστημα πρέπει να επιτρέπει την εύκολη προσθήκη νέων δυνατοτήτων και εργαλείων διαχείρισης ανάλογα με τις απαιτήσεις της κάθε εφαρμογής.

Μια ακόμη βασική λειτουργία ενός συστήματος διαχείρισης είναι η δυνατότητα ανίχνευσης και αναφοράς λαθών και προβλημάτων στο δίκτυο. Καθώς το διαχειριζόμενο δίκτυο επεκτείνεται, μια τέτοια υπηρεσία γίνεται όλο και περισσότερο πολύτιμη. Έστω και αν η διαχείριση λαθών δεν υποστηρίζεται, η ανίχνευση και η ειδοποίηση είναι απαραίτητα χαρακτηριστικά ενός διαχειριστικού συστήματος.

Μια άλλη απαίτηση είναι η βιωσιμότητα του διαχειριστικού συστήματος σε κρίσιμες καταστάσεις. Όταν το διαχειριζόμενο δίκτυο πέφτει και γενικά σε καταστάσεις σημαντικών προβλημάτων και λαθών, το σύστημα πρέπει να παραμείνει σε λειτουργία (σε όποιο βαθμό είναι αυτό δυνατό). Όσο περισσότερο ανεκτικό στα λάθη του διαχειριζόμενου δικτύου είναι το σύστημα, τόσο καλύτερα εκπληρώνει τον ρόλο του σε περιπτώσεις προβλημάτων.

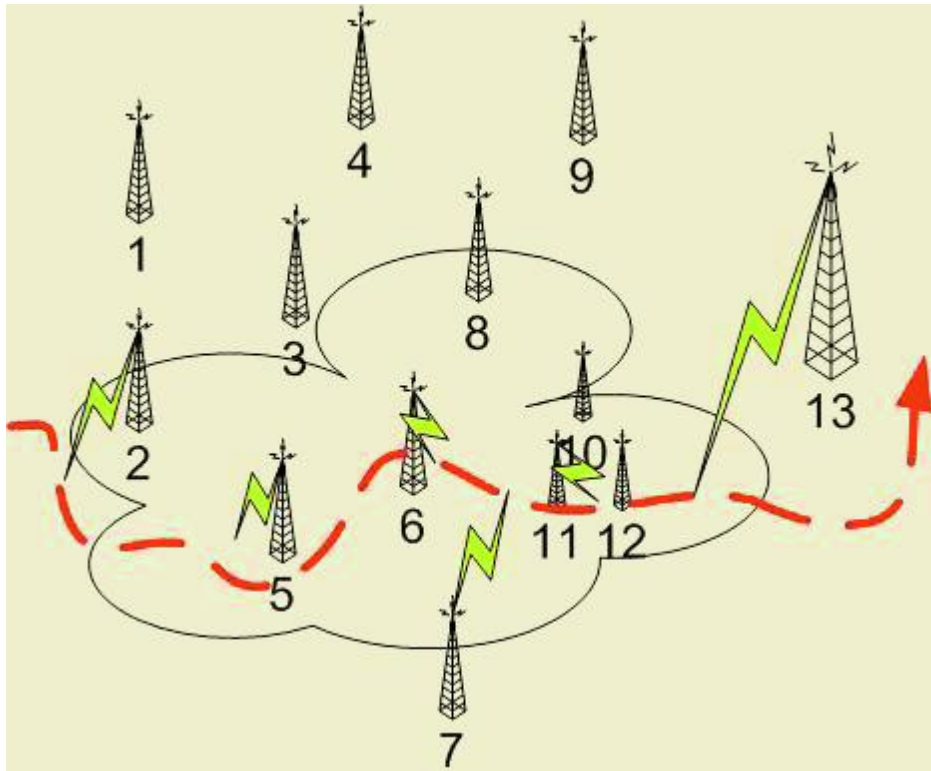
Η αρχιτεκτονική που προτείνεται και χρησιμοποιείται σήμερα για την διαχείριση τηλεπικοινωνιακών δικτύων και δικτύων υπολογιστών αποτελείται από το σύστημα διαχείρισης των δικτύων (**Network Management System, NMS**) ή το Σύστημα Λειτουργίας (**Operation**

Systems , OS) και τα στοιχεία εκείνα των δικτύων (**Network Elements** ,NE) τα οποία θέλουμε να διαχειριστούμε .Τέτοια NE σ ένα δίκτυο είναι κυρίως μηχανήματα αποθήκευσης ή επεξεργασίας πληροφοριών , όπως **hosts (terminal servers** κ.α.),καθώς και μηχανήματα διασύνδεσης δικτύων ,όπως **routers, repeaters** κ.α. στα οποία τρέχουν διαδικασίες διαχείρισης, που ονομάζονται (**agents**) και είναι υπεύθυνες για την εκτέλεση των συναρτήσεων που καλούν τα συστήματα διαχείρισης. Για την μεταφορά της πληροφορίας μεταξύ των διαχειριστικών συστημάτων και των διαχειριζόμενων στοιχείων χρησιμοποιούνται κατάλληλα πρωτόκολλα μεταφοράς της πληροφορίας που αφορά την διαχείριση. Τα πρωτόκολλα αυτά καθορίζουν με σαφήνεια τον τρόπο επικοινωνίας, τη μορφή και την σημασία των μηνυμάτων που θα ανταλλαχθούν, όπως επίσης και τον τρόπο ορισμού και περιγραφής των στοιχείων που θέλουμε να διαχειριστούμε . Ένα από τα γνωστότερα πρωτόκολλα αυτά είναι το SNMP (**Simple Network Management Protocol**).

1.4 Mobility prediction, ορισμός και χαρακτηριστικά

Αρχικά θα μπορούσαμε να πούμε ότι ο όρος <<**mobility prediction**>> έχει να κάνει με την πρόβλεψη της μελλοντικής θέσης ενός κινητού τερματικού(MT) μέσα σε ένα δίκτυο. Παρόλο που αυτή η εξήγηση είναι η γενική ιδέα του όρου, είναι απαραίτητο να τονιστούν κάποιες λεπτομέρειες που βοηθούν στην καλύτερη ερμηνεία του όρου.

Το πρώτο στοιχείο που χρειάζεται περισσότερη ανάλυση είναι η έννοια της τοποθεσίας(**location**). Η θέση του χρήστη που φέρει το MT, κατά κύριο λόγο θεωρείται, ως οι γεωγραφικές συντεταγμένες του. Σημειώνεται, ωστόσο, ότι υπάρχουν προβλήματα με την άμεση σύνδεση της τοποθεσία του MT με τη γεωγραφική θέση του χρήστη. Είναι καλύτερα να εξεταστεί η κίνηση του MT μέσω του δικτύου ως διαδοχική λίστα των συνδέσεων που πραγματοποιεί το MT. Αυτό ακριβώς σημαίνει ότι η θέση MT είναι πάντα ένα από ένα πεπερασμένο σύνολο από θέσεις που αντιπροσωπεύουν ένα από τα πιθανά σημεία πρόσβασης στο δίκτυο. Όλα αυτά φαίνονται καλύτερα στην εικόνα που ακολουθεί. Η διακεκομμένη γραμμή δείχνει την κίνηση που κάνει ο χρήστης μέσα στο δίκτυο καθώς συνδέεται σε διάφορους σταθμούς βάσης του δικτύου.



Εικόνα 1.2

Το δεύτερο στοιχείο που αξίζει να αναφέρουμε είναι η αίσθηση ότι η πρόβλεψη βασίζεται σε μια προηγούμενη κατάσταση-γνώση. Ο ακριβής προσδιορισμός της γνώσης που χρησιμοποιείται για την πρόβλεψη, είναι απαραίτητος για την επιλογή του καταλληλότερου μοντέλου πρόβλεψης, που ανταποκρίνεται στη λύση του εκάστοτε προβλήματος. Για παράδειγμα αν η πρόβλεψη βασίζεται σε ένα σύνολο δεδομένων που δεν είναι διαθέσιμα σε μια δοθείσα κατάσταση το μοντέλο που επιλέχθηκε είναι άχρηστο και άρα πρέπει να αντικατασταθεί. Αυτό γίνεται ευκολότερα κατανοητό με την έννοια της ασφάλειας. Αν το μοντέλο λάβει υπόψη του τις απαιτήσεις ασφάλειας των χρηστών, δηλαδή να μην αποκαλύπτει τα στοιχεία της προηγούμενης τους τοποθεσίας, δεν μπορεί να λειτουργήσει. Τα δεδομένα που χρειάζονται, μπορεί να υπάρχουν στο σύστημα αλλά δεν είναι προσβάσιμα και άρα το μοντέλο πρόβλεψης αχρηστεύεται.

Το τρίτο στοιχείο είναι η κατηγοριοποίηση της πρόβλεψης, δηλαδή η πρόβλεψη ότι θα συμβεί κάποιο γεγονός και είναι απαραίτητο να προβλέψουμε το χρόνο που θα συμβεί αυτό το γεγονός. Χαρακτηριστικό παράδειγμα είναι το φαινόμενο της διαπομπής, από ένα σταθμό βάσης σε έναν άλλο. Μια άλλη πρόβλεψη είναι να μπορέσουμε να προβέψουμε την κατάσταση ενός κινητού τερματικού σε ένα μελλοντικό χρόνο. Για παράδειγμα σε ποιο σταθμό βάσης του δικτύου θα συνδεθεί ένα MT σε μια συγκεκριμένη χρονική στιγμή t .

Το τέταρτο στοιχείο έχει να κάνει με το βαθμό ακρίβειας της πρόβλεψης. Αν το μοντέλο πρόβλεψης προβλέπει ένα γεγονός σε μια συγκεκριμένη χρονική στιγμή, κατά πόσο είναι ακριβής, σε λεπτά και σε δευτερόλεπτα. Αν συμβαίνει το αντίθετο, όπως είδαμε στην προηγούμενη περίπτωση τότε η τοποθεσία πρέπει να είναι ακριβής, μπορεί να προσδιοριστεί με πραγματικές γεωγραφικές συντεταγμένες.

Υπάρχουν πολλοί τρόποι για την επίλυση του προβλήματος του mobility prediction, άρα σημαίνει ότι έχουμε και πολλά διαφορετικά μοντέλα πρόβλεψης. Το κάθε ένα μοντέλο είναι διαφορετικό και χρησιμοποιείται για την επίλυση διαφορετικού προβλήματος χωρίς αυτό να σημαίνει ότι δεν μπορούν να συνδυαστούν διάφορα μοντέλα σε συνδυασμό προβλημάτων.

Η κινητικότητα μέσα σ ένα δίκτυο είναι πολύ σημαντική είτε πρόκειται για ένα δίκτυο που υποστηρίζεται από υποδομή, όπως ένα δίκτυο κυψελών είτε για δίκτυο που δεν υποστηρίζεται από υποδομή. Απλά στο δίκτυο με υποδομή η τοποθεσία ενός σημείου πρόσβασης στο δίκτυο είναι προκαθορισμένη σε αντίθεση με τη δεύτερη περίπτωση.

Όπως είδαμε με το mobility prediction μπορεί να αντιμετωπιστεί το πρόβλημα της διαπομπής γιατί ο χρήστης κινείται μέσα στο δίκτυο και είναι σημαντικό να γνωρίζουμε όχι μόνο το επόμενο σημείο που θα συνδεθεί αλλά και το γεγονός που θα συμβεί σ ένα MT, καθώς παρατηρείται πολύ συχνά το φαινόμενο της διαπομπής.

Ακόμη με το mobility prediction μπορούμε να προσδιορίσουμε το σύνολο από πιθανούς προορισμούς ενός χρήστη μέσα στο δίκτυο. Το λεγόμενο **path recognition**. Για παράδειγμα μπορούμε να προβλέψουμε την μελλοντική τοποθεσία ενός χρήστη που είναι συνδεδεμένος σ ένα σημείο πρόσβασης αν έχουμε στοιχεία για άλλους χρήστες που είναι συνδεδεμένοι στο ίδιο σημείο πρόσβασης.

Τέλος μια ακόμη διαφορά που έχει να κάνει με το μοντέλο πρόβλεψης είναι αν θα λάβουμε υπόψη μας μετρήσεις που αφορούν τους χρήστες ή τη συμπεριφορά των MT μέσα στο δίκτυο στο οποίο είναι συνδεδεμένα. Μια προσέγγιση βασισμένη σε μετρήσεις θα υπολογίσει την πιθανότητα να συμβεί κάποιο γεγονός με βάση ορισμένες παραμέτρους.

2. WEKA Explorer

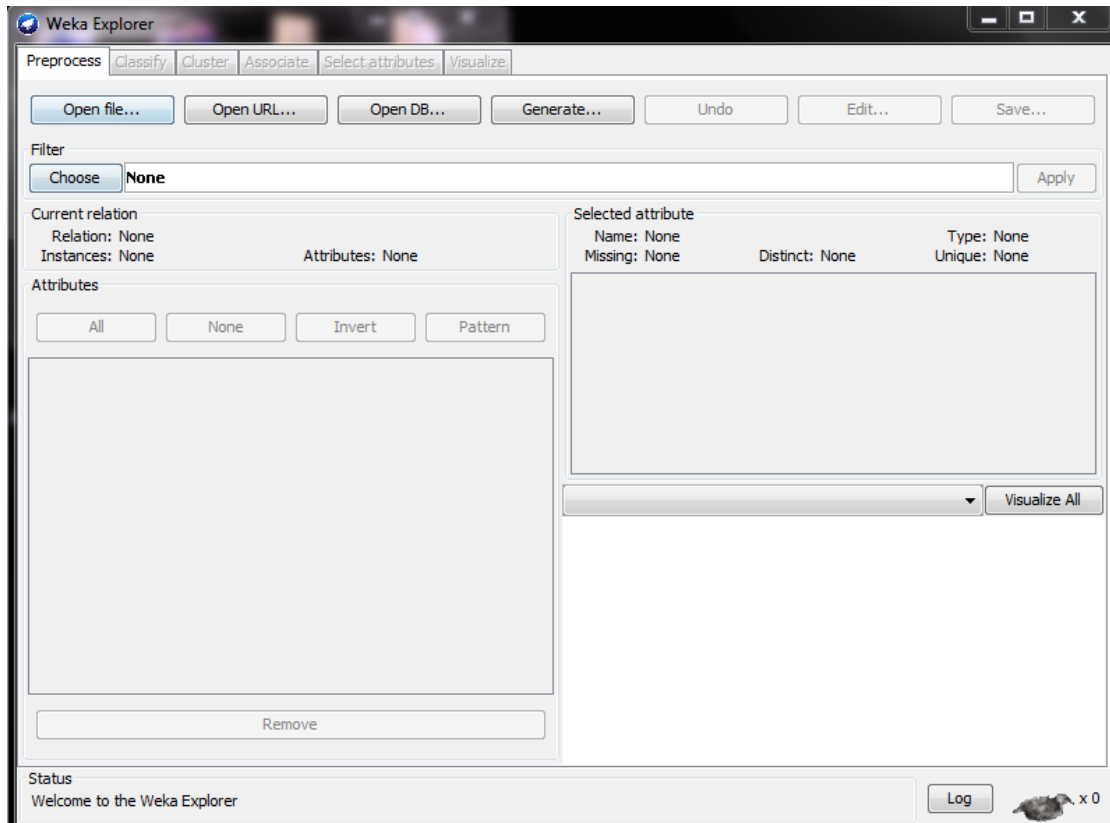
Το WEKA Explorer είναι το πρώτο και το κυριότερο περιβάλλον εργασίας του WEKA, που χρησιμοποιούν περισσότερο οι χρήστες. Παρέχει την ίδια λειτουργικότητα με το WEKA Experimenter και το WEKA Command Line. Στο κεφάλαιο αυτό θα δούμε αναλυτικότερα τις 6 βασικές λειτουργίες που προσφέρει στους χρήστες.

- **Preprocess:** Η λειτουργία αυτή δίνει τη δυνατότητα στους χρήστες να φορτώσουν τα δεδομένα τους στην πλατφόρμα του WEKA, είτε μέσω μιας βάσης δεδομένων, είτε με αρχεία τύπου csv και arff. Ακόμη στο σημείο αυτό, μπορεί να γίνει και τροποποίηση των δεδομένων, όπως αλλαγή τιμών και χαρακτηριστικών, ή ακόμη και διαγραφή και εισαγωγή νέων τιμών.
- **Classify:** Η λειτουργία αυτή δίνει τη δυνατότητα στους χρήστες να εφαρμόσουν τους αλγόριθμους μηχανικής μάθησης, οι οποίοι λέγονται classifiers στο WEKA.
- **Associate:** Η λειτουργία αυτή δίνει τη δυνατότητα στους χρήστες να καταλάβουν τη συσχέτιση που υπάρχει μεταξύ των δεδομένων τους.
- **Cluster:** Μέσω της λειτουργίας αυτής, δίνεται η δυνατότητα στους χρήστες να δημιουργήσουν ομάδες μεταξύ των όμοιων δεδομένων. Χαρακτηριστικό είναι το γεγονός ότι τα δεδομένα που ανήκουν σε ένα cluster είναι πιθανότερο να μοιάζουν μεταξύ τους, σε σχέση με άλλα που ανήκουν σε διαφορετικά clusters.
- **Select:** Μέσω τη λειτουργίας αυτής, δίνεται η δυνατότητα στους χρήστες να αναγνωρίζουν πιο εύκολα τα χαρακτηριστικά και τα δεδομένα που είναι ευκολότερο να προβλεφθούν κατά τη διαδικασία της μηχανικής μάθησης.
- **Visualize:** Μέσω της λειτουργίας αυτής, δίνεται η δυνατότητα στους χρήστες, μέσω γραφημάτων που αναπαριστούν τα δεδομένα με κουκίδες, να επιλέξουν τα δεδομένα εκείνα που χρειάζεται να

αναλύσουν περαιτέρω, μετά από την εφαρμογή κάποιου από τους classifiers του WEKA.

2.1 Preprocess

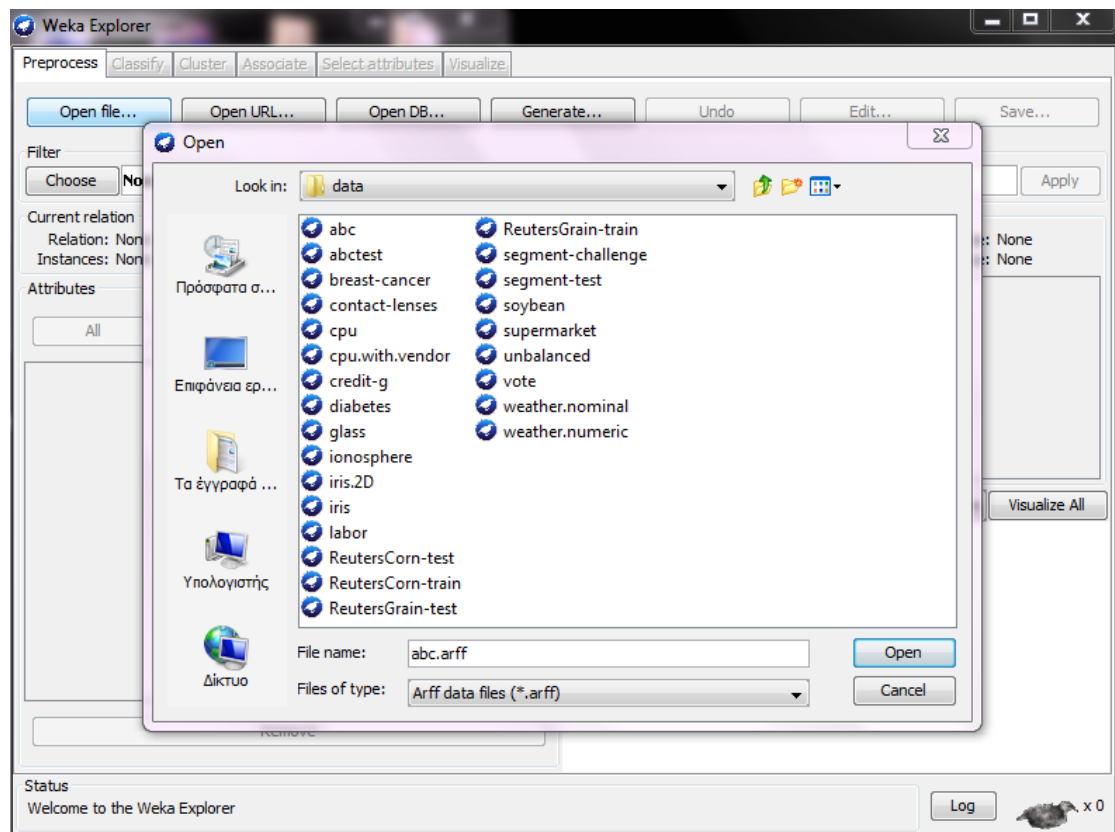
Η εικόνα που ακολουθεί είναι η πρώτη εικόνα που εμφανίζεται μόλις ανοίξουμε το WEKA Explorer. Μόνο η λειτουργία preprocess είναι διαθέσιμη, αφού δεν έχουμε φορτώσει ακόμη δεδομένα στο WEKA.



Εικόνα 2.1

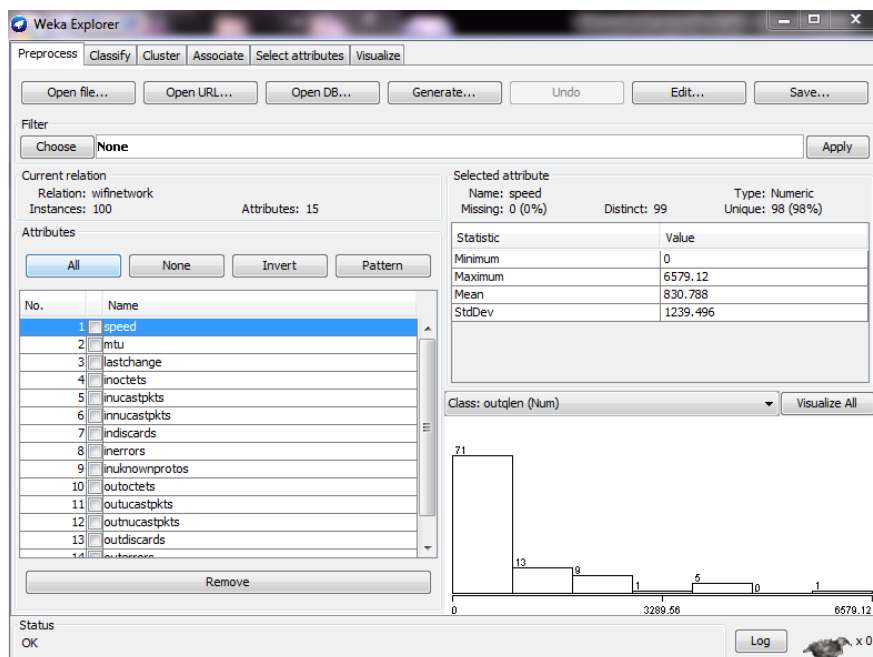
Όπως φαίνεται και στην εικόνα, μας δίνεται η δυνατότητα να φορτώσουμε τα αρχεία μας είτε από μια βάση δεδομένων, είτε από μια διεύθυνση url, είτε να δημιουργήσουμε τώρα τα δεδομένα μας με τη χρήση κάποιου φίλτρου.

Πάμε τώρα να φορτώσουμε τα δεδομένα μας στο WEKA όπως φαίνεται στη διαδικασία, που περιγράφεται από την εικόνα 2.2



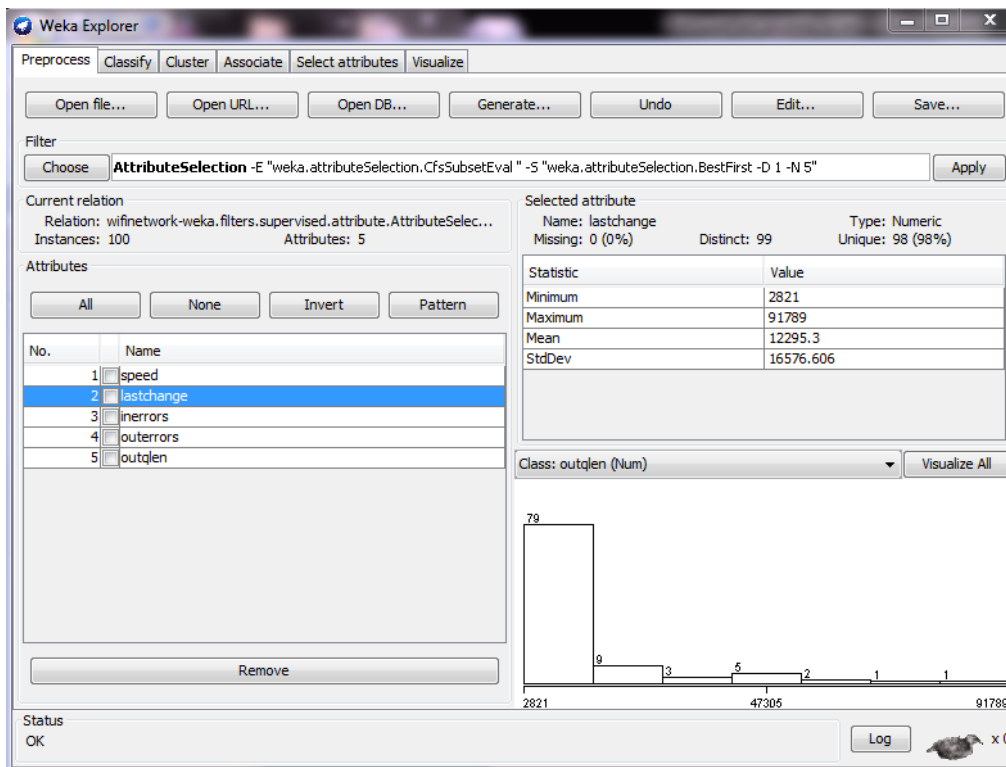
Εικόνα 2.2

Μόλις φορτώσουμε τα δεδομένα μας από το αρχείο data.arff που δημιουργήσαμε, μας εμφανίζεται η παρακάτω εικόνα. Σ αυτό το σημείο είμαστε έτοιμοι να ξεκινήσουμε να δουλεύουμε με τα δεδομένα μας, τα οποία αναλύθηκαν στην εισαγωγή.



Εικόνα 2.3

Έπειτα μπορούμε να εφαρμόσουμε κάποιο φίλτρο στα δεδομένα μας για πιο εύκολη εφαρμογή των αλγορίθμων μηχανικής μάθησης, όπως για παράδειγμα να επιλέξουμε ορισμένα μόνο απ τα χαρακτηριστικά-attributes των δεδομένων μας ή να εμφανίζεται το χαρακτηριστικό που έχει την μεγαλύτερη τιμή δεδομένων συγκριτικά με τα υπόλοιπα. Η διαδικασία φαίνεται στην παρακάτω εικόνα(εμφάνιση των 5 πρώτων χαρακτηριστικών με τις μεγαλύτερες τιμές δεδομένων).

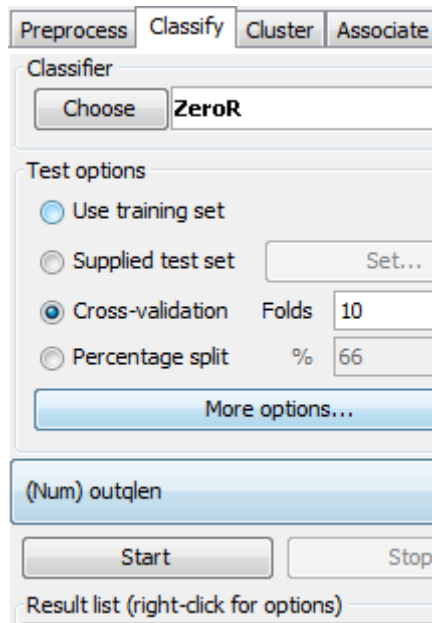


Εικόνα 2.4

Μπορούμε ακόμη να δούμε διαγράμματα που απεικονίζουν τις τιμές για κάθε χαρακτηριστικό, διακρίνοντας την υψηλότερη και μικρότερη τιμή ανάλογα με το χαρακτηριστικό που θα επιλέξουμε. Επιπλέον φαίνονται αυτές οι πληροφορίες και στο τμήμα της εικόνας πάνω από το σημείο της γραφικής αναπαράστασης.

Στη συνέχεια ακολουθούν παραδείγματα εφαρμογής αλγορίθμων μηχανικής μάθησης, οι λεγόμενοι classifiers, που για να τους δούμε όλους θα πρέπει να ανοίξουμε την καρτέλα **classify**.

2.2 Classify

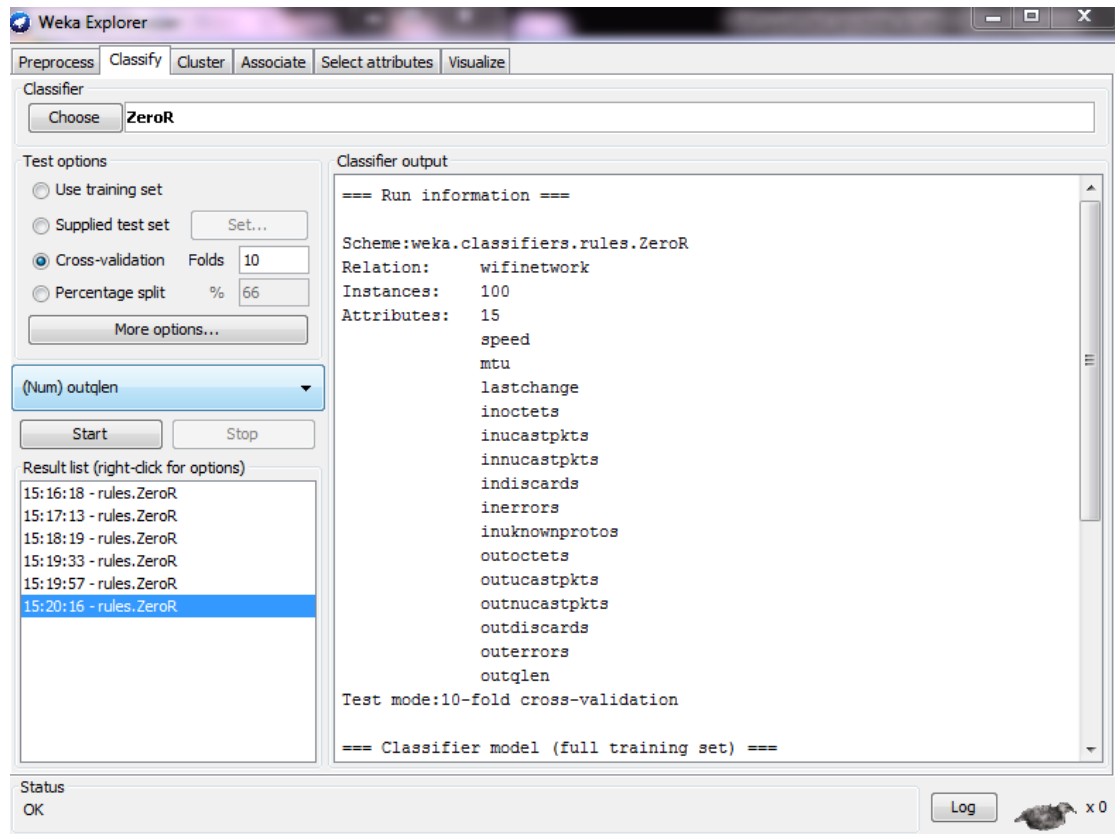


Εικόνα 2.5

Όπως φαίνεται και από την εικόνα 2.5 μπορούμε να ορίσουμε τις παραμέτρους για την εφαρμογή των classifiers μας. Στα παραδείγματά μας επιλέξαμε την τεχνική **cross validation**, που είναι μια τεχνική επικύρωσης στα στατιστικά μοντέλα, μέσω της οποίας δεδομένα μετατρέπονται σε ένα άλλο ανεξάρτητο σύνολο δεδομένων από το οποίο εξήχθησαν. Ακόμη υπάρχει η επιλογή **supplied test set** που μπορούμε να δημιουργήσουμε ένα υποσύνολο των δεδομένων προς εξέταση από το γενικό σύνολο των δεδομένων μας. Στην εργασία μας το σύνολο των δεδομένων προς εξέταση είναι το αρχείο **data.arff (training set)** και ένα παράδειγμα **test set** είναι το **datatest.arff**. Επιπλέον μπορούμε να ορίσουμε το ποσοστό % διάσπασης των δεδομένων μας με την επιλογή **percentage split**.

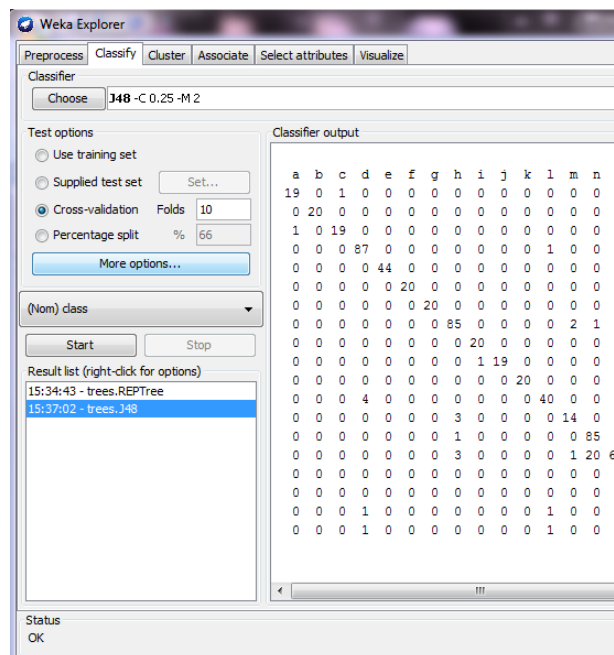
Αφού επιλέξουμε κάποιον πατάμε **start** και βλέπουμε τα αποτελέσματα.

Στην εικόνα 2.6 φαίνεται η εφαρμογή του classifier **ZeroR** (που προβλέπει την πλειοψηφία των τιμών μια κλάσης- attribute).



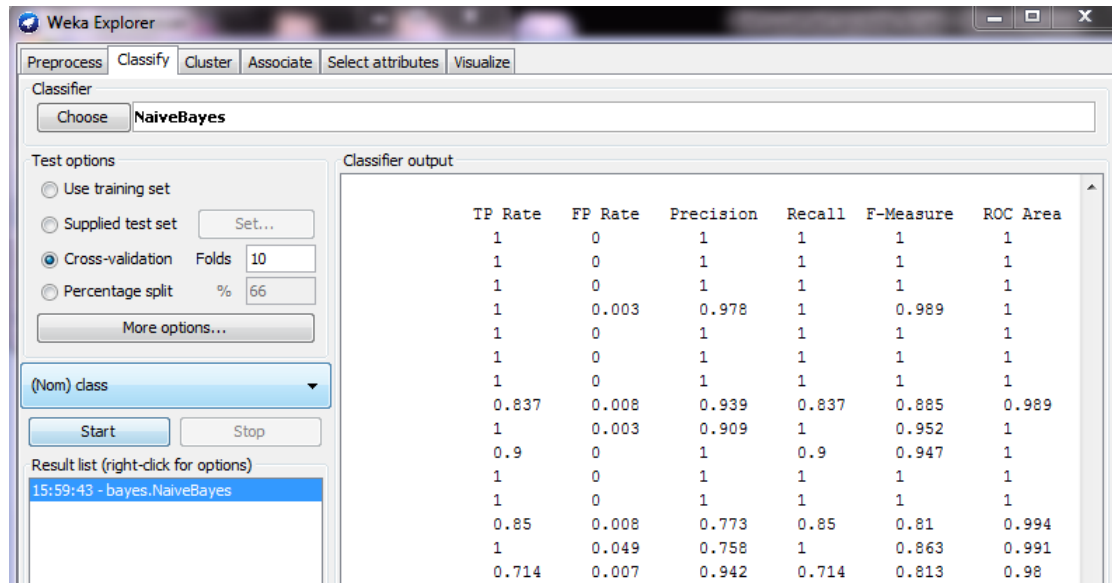
Εικόνα 2.6

Στην εικόνα 2.7 φαίνεται η εφαρμογή του classifier **J48** (στατιστικός ταξινομητής που χρησιμοποιείται για τη δημιουργία ενός δέντρου απόφασης.)



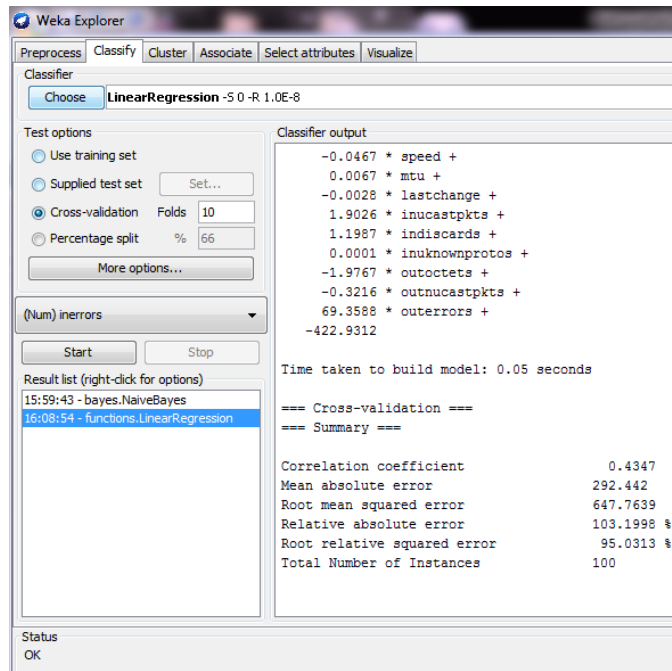
Εικόνα 2.7

Στην εικόνα 2.8 φαίνεται η εφαρμογή του classifier **NaiveBayes** (που είναι πιθανολογικός ταξινομητής, που βασίζεται στη θεωρία ότι τα attributes είναι ασοσχέτιστα μεταξύ τους).



Εικόνα 2.8

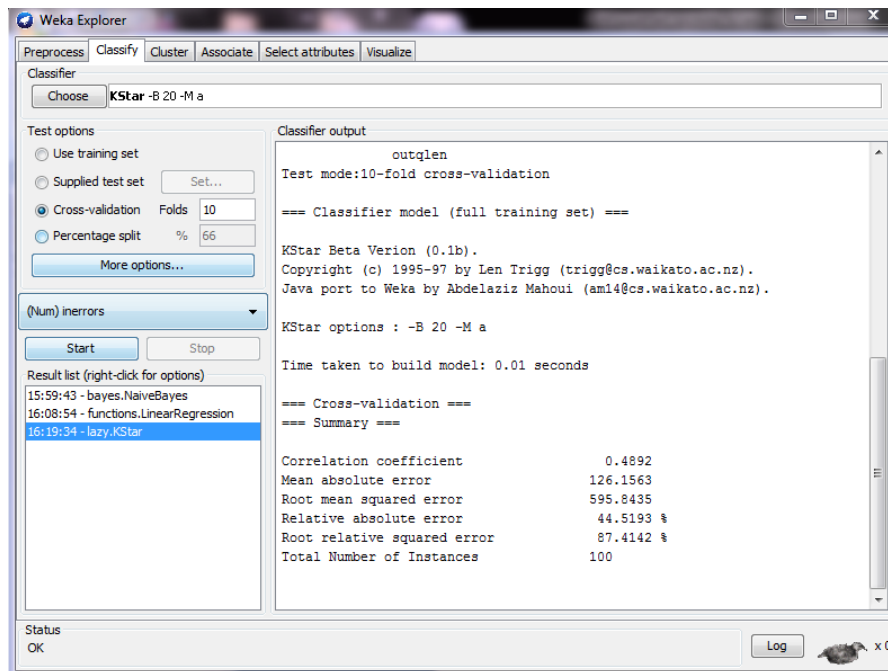
Στην εικόνα 2.9 φαίνεται η εφαρμογή του classifier **function logistics** (που είναι στατιστικές συναρτήσεις ταξινόμησης και παλινδρόμησης δεδομένων).



Εικόνα 2.9

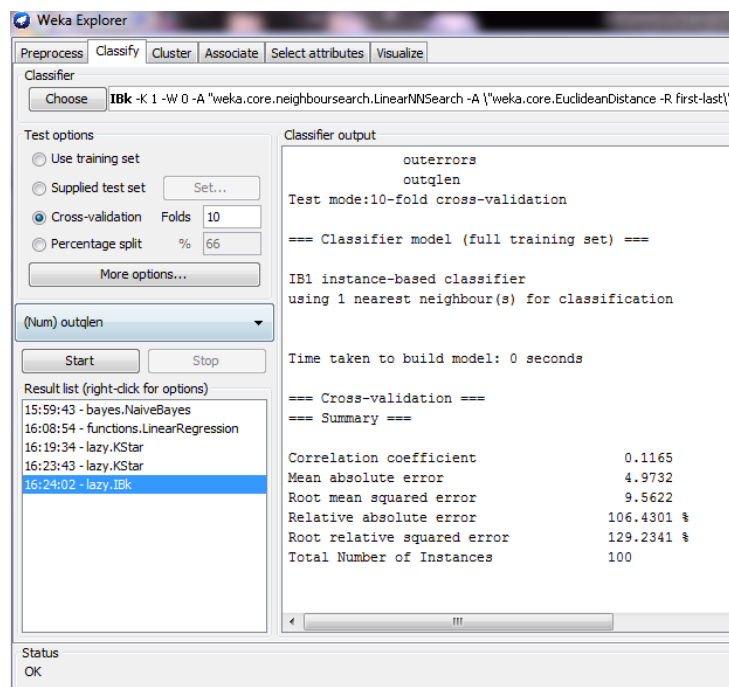
Βελτιστοποίηση διαχείρισης δικτύων με χρήση τεχνικών μηχανικής μάθησης

Στην εικόνα 2.10 φαίνεται η εφαρμογή του classifier **lazyKstar** (lazy machine learning που το σύστημα περιμένει να δημιουργήσει δεδομένα πριν τη χρήση queries).



Εικόνα 2.10

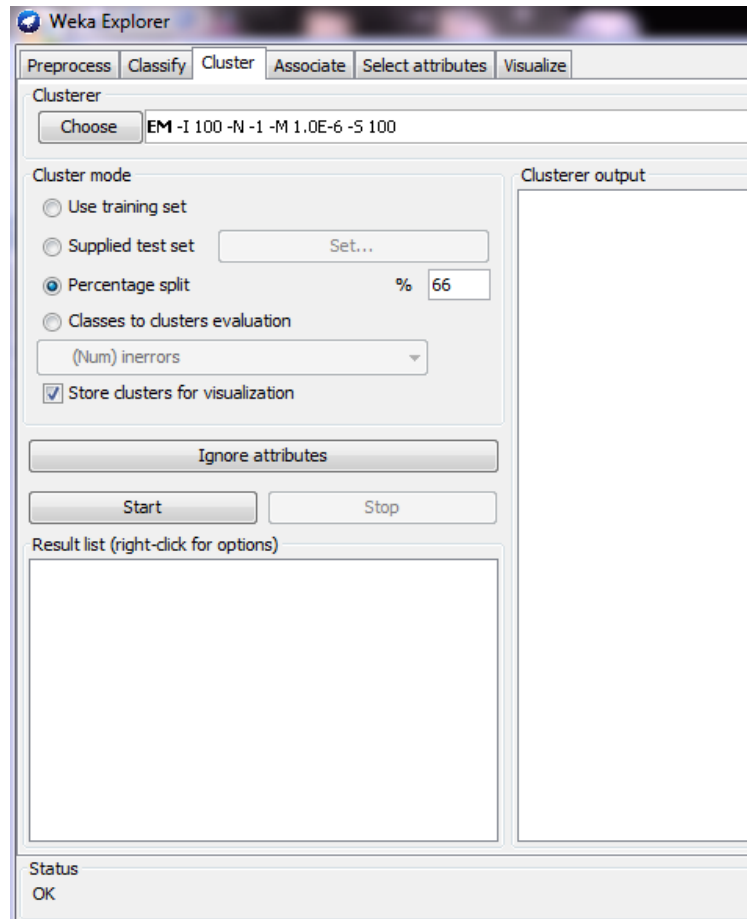
Στην εικόνα 2.11 φαίνεται η εφαρμογή του classifier **lazylbk**



Εικόνα 2.11

2.3 Cluster

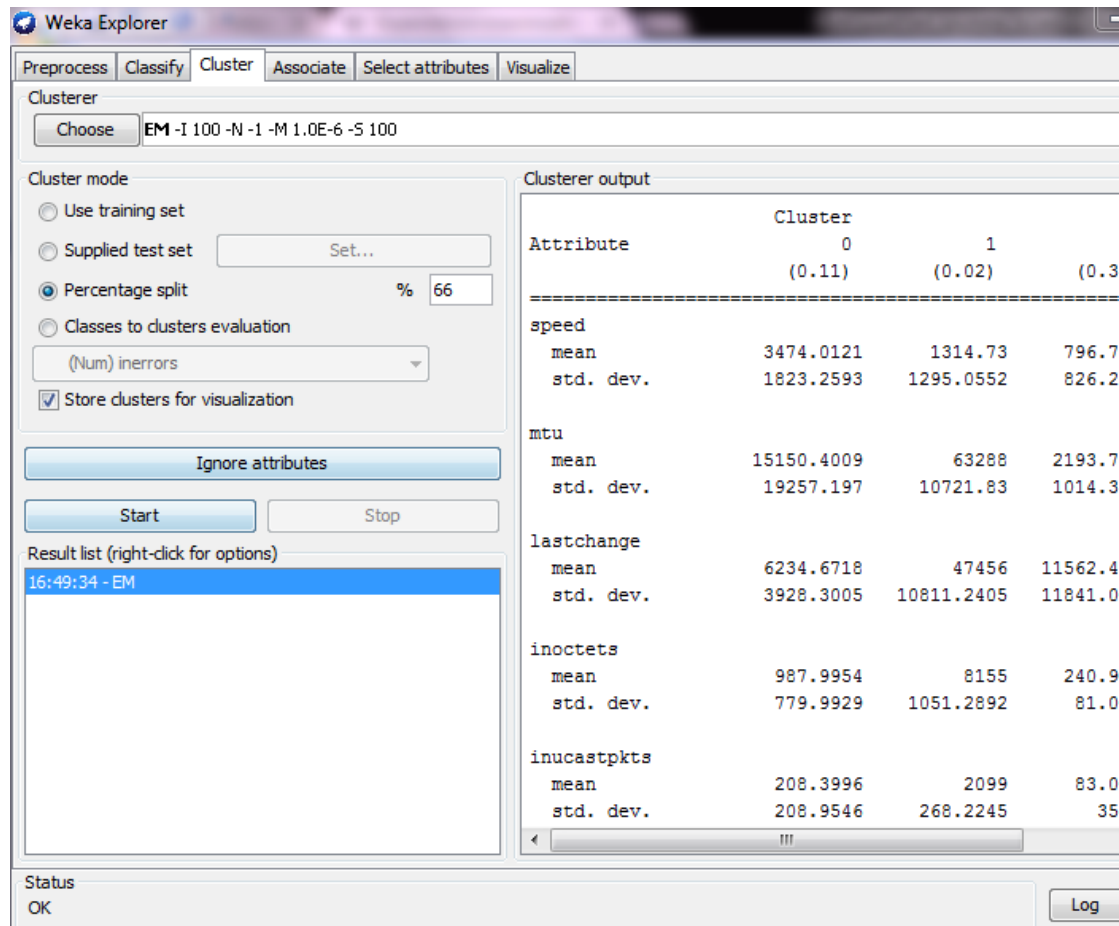
Στην προηγούμενη ενότητα είδαμε την λειτουργία Classify του WEKA Explorer. Πάμε τώρα να εξετάσουμε την λειτουργία **Cluster**. Προηγουμένως όμως πρέπει να επιλέξουμε την καρτέλα **Cluster** όπως φαίνεται στην παρακάτω εικόνα.



Εικόνα 2.12

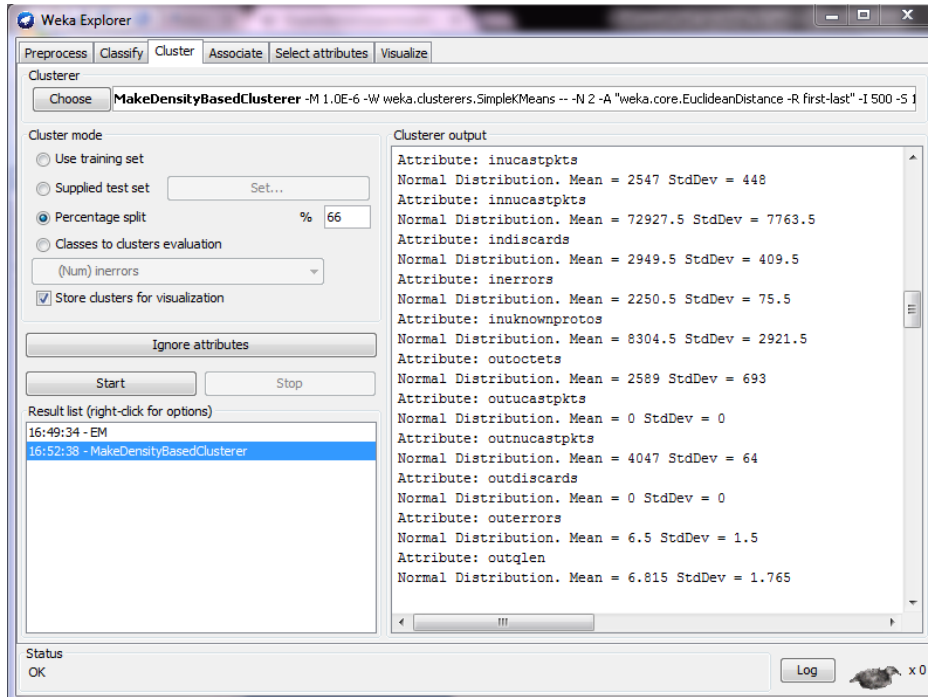
Κι εδώ υπάρχουν κάποιες παράμετροι που μπορούμε να ρυθμίσουμε τη διαδικασία του clustering. Η πιο σημαντική είναι να ορίσουμε το χαρακτηριστικό- attribute σύμφωνα με το οποίο να γίνει η ομαδοποίηση.

Ακολουθεί η εφαρμογή του **cluster**, που προτείνει το WEKA, του **EM(expectation-maximization)**. Είναι μια επαναληπτική μέθοδος που μεγιστοποιεί την πιθανότητα εύρεσης γειτονικής τιμής σ ένα σύνολο δεδομένων.



Εικόνα 2.13

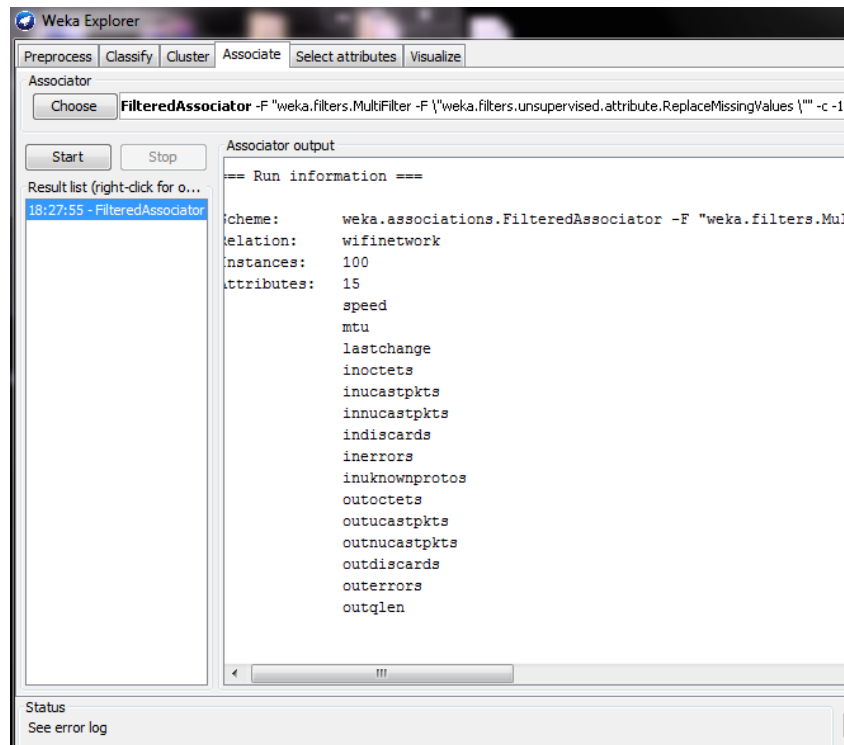
Στη συνέχεια ακολουθεί η εφαρμογή ενός άλλου **cluster** του **Makedensitybased** που έχει να κάνει με την πυκνότητα που εμφανίζουν οι τιμές ενός χαρακτηριστικού γύρω από μια ορισμένη τιμή.



Εικόνα 2.14

2.4 Associate

Στην ενότητα αυτή θα εξετάσουμε τη λειτουργία **associate** που μας βοηθά να καταλάβουμε τη συσχέτιση που υπάρχει μεταξύ των δεδομένων μας. Υπάρχει μια λίστα από associates διαθέσιμα, το WEKA όμως δεν άφησε να δουλέψουμε με κανένα γιατί έπειτα από αναζήτηση στο αρχείο, όλα τα δεδομένα ήταν τύπου **NUMERIC**. Οι associators του WEKA μπορούν να λειτουργήσουν μόνο με δεδομένα τύπου **nominal** Η μόνη εικόνα διαθέσιμη είναι η εξής:

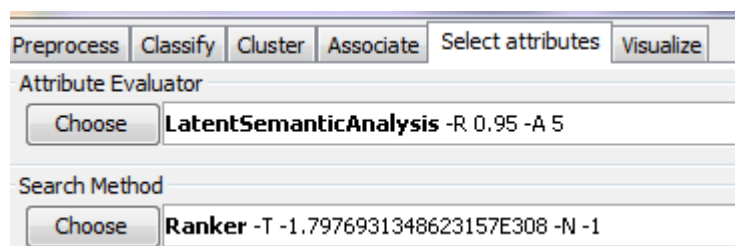


Εικόνα 2.15

2.5 Select

Στην ενότητα αυτή θα εξετάσουμε τη λειτουργία **select** που παρέχει αλγορίθμους για να καταλάβουμε ποια χαρακτηριστικά των δεδομένων μας είναι ευκολότερο να προβλεφθούν. Δίνονται κάποιοι διαθέσιμοι **selectors**, όμως εμείς επικεντρωθήκαμε στον **LaterSemantiAnalysis**.

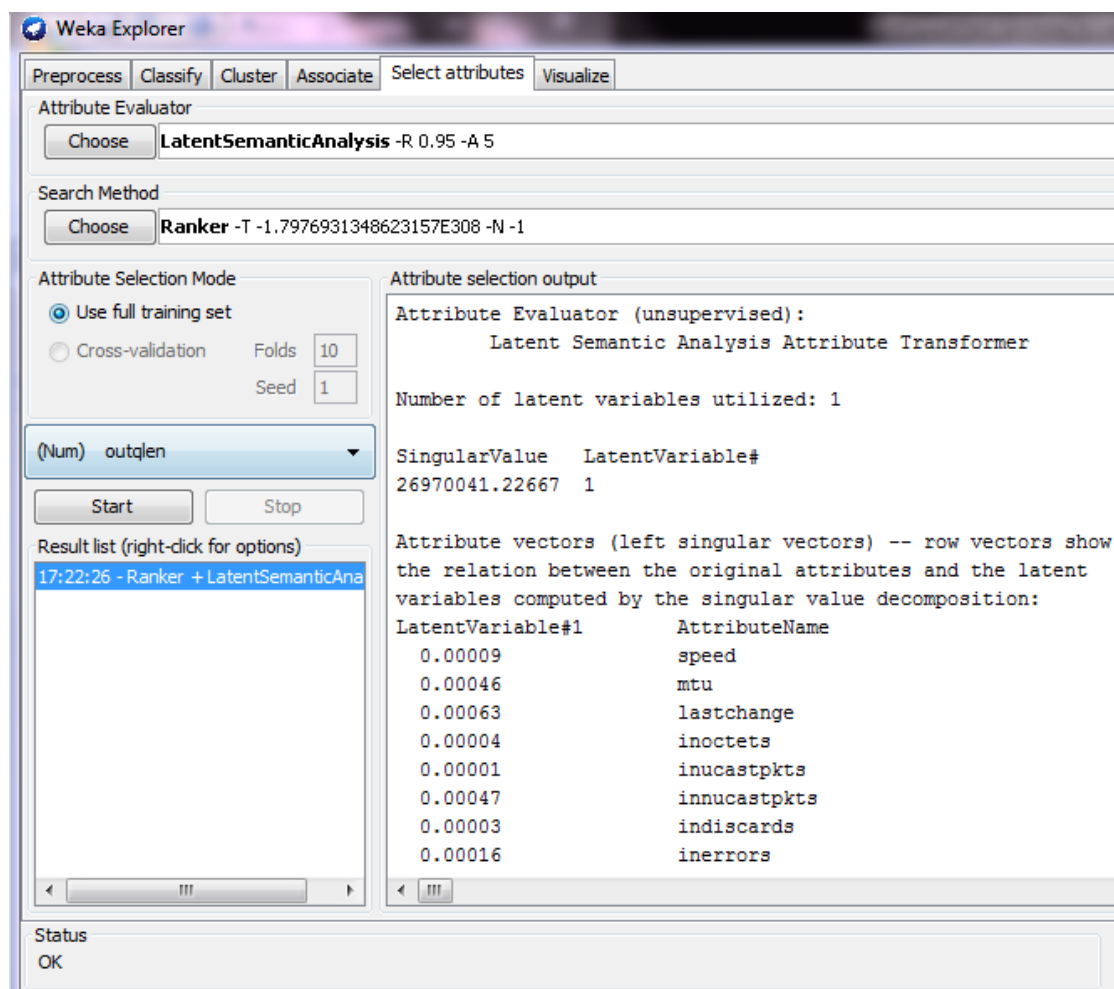
Οι παράμετροι για την εκτέλεσή του φαίνονται στην εικόνα 2.16



Εικόνα 2.16

Με την επιλογή ενός **attribute Evaluator** το WEKA συμπληρώνει αυτόματα το καταλληλότερο **search method**. Με την επιλογή **choose** μπορούμε να διαλέξουμε άλλο **selector** και το WEKA να συμπληρώσει αυτόματα το πεδίο που λείπει.

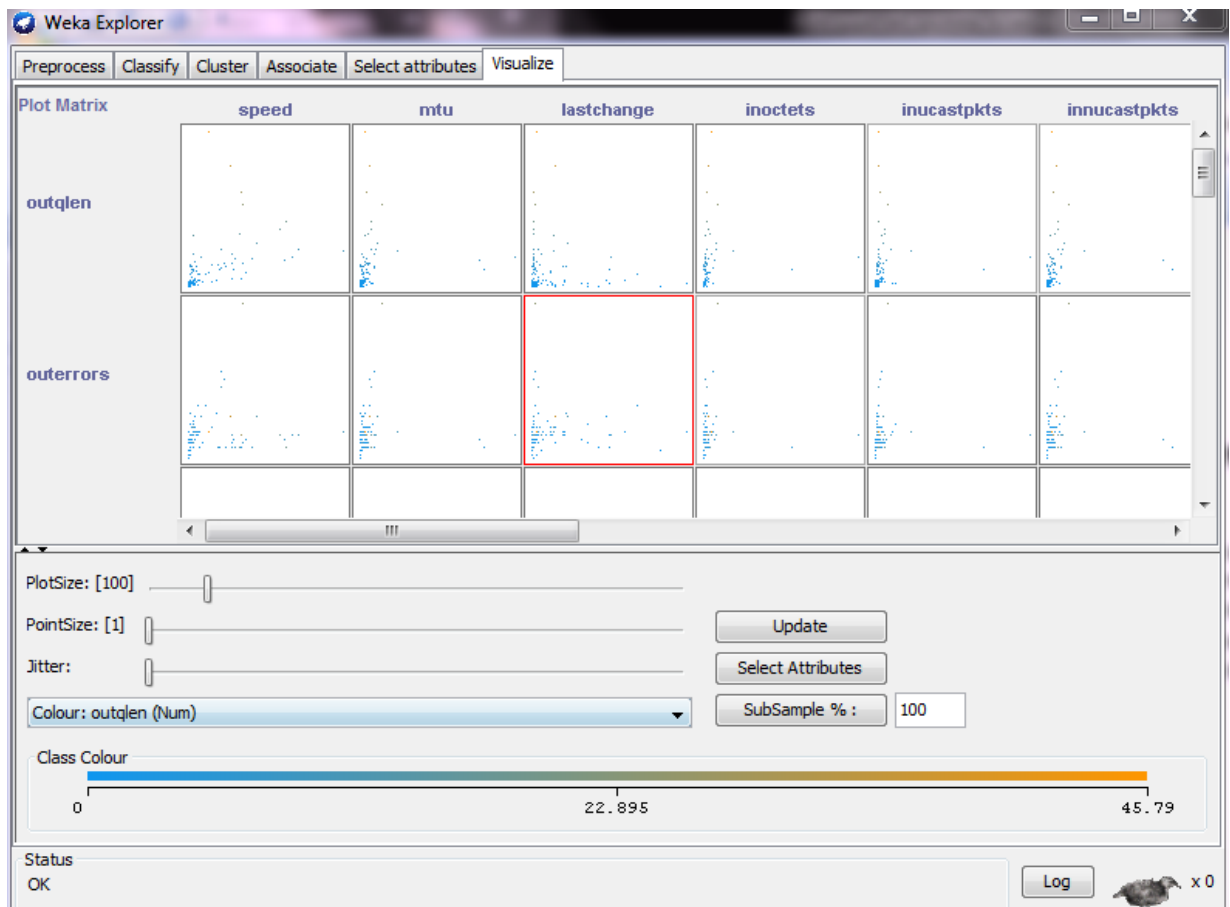
Στην εικόνα 2.17 ακολουθούν τα αποτελέσματα της εκτέλεσής του **LaterSemantiAnalysis**.



Εικόνα 2.17

2.6 Visualize

Στην ενότητα αυτή περιγράφεται η λειτουργία visualize που δεν είναι τίποτα άλλο από την γραφική αναπαράσταση με διαγράμματα των δεδομένων των τιμών για κάθε ένα από τα χαρακτηριστικά των δεδομένων που χρησιμοποιήθηκαν. Ενδεικτική είναι η εικόνα που ακολουθεί, όπου μετακινώντας την μπάρα βλέπουμε όλα τα διαγράμματα .



Εικόνα 2.18

3. WEKA Experimenter

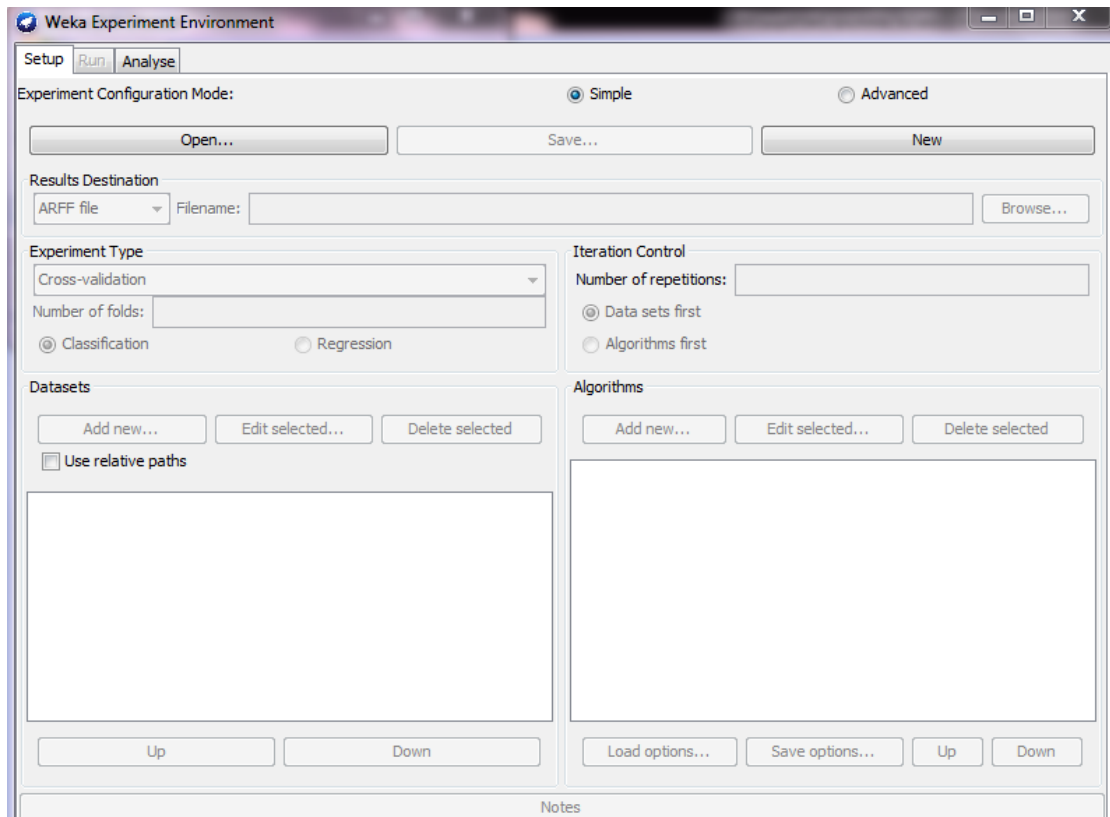
Το WEKA Experimenter αποτελεί το 2^ο περιβάλλον εργασίας της πλατφόρμας WEKA. Η βασική λειτουργικότητά του σε σχέση με το WEKA Explorer και το WEKA Command Line είναι ότι δίνει τη δυνατότητα στους χρήστες να δημιουργήσουν δικά τους πειράματα-test. Τα πειράματα αυτά είτε θα αποθηκεύονται στο σύστημα σαν arff αρχεία είτε σαν experiment configuration files. Επιπλέον οι χρήστες στο σημείο αυτό μπορούν να εφαρμόσουν ταυτόχρονα παραπάνω από ένα αλγόριθμο μηχανικής μάθησης, παράλληλα να προχωρήσουν σε τεχνικές clustering. Αυτό δεν γινόταν στον WEKA Explorer καθώς οι χρήστες πρέπει να μεταβαίνουν από καρτέλα σε καρτέλα για να εφαρμόσουν είτε κάποιο classifier, είτε να εφαρμόσουν μια τεχνική clustering στα δεδομένα τους.

Μια άλλη σημαντική δυνατότητα που μας δίνει το WEKA Experimenter είναι η δυνατότητα να επιλέγουμε εμείς αν θέλουμε να αποθηκεύονται τα αποτελέσματα της επεξεργασίας των δεδομένων καθώς και τον προορισμό αποθήκευσής τους. Δηλαδή μπορούμε να τα αποθηκεύσουμε σε αρχεία εύκολα προσβάσιμα, που θα επιτρέπουν καλύτερη στατιστική επεξεργασία στη συνέχεια για ασφαλέστερη εξαγωγή συμπερασμάτων. Γίνεται εύκολα αντιληπτό από τα παραπάνω ότι το WEKA Experimenter προσφέρει έναν εύκολο και βολικό τρόπο στους χρήστες του, για τη σύγκριση μεταξύ διαφορετικών αλγορίθμων μηχανικής μάθησης, δηλαδή των classifiers.

Τα τεστ στον WEKA Experimenter μπορούν να πάρουν μια από τις παρακάτω 2 μορφές: δηλαδή είτε θα είναι simple, είτε θα είναι advanced. Στην simple μορφή του ένα τεστ περιλαμβάνει μία ή πολλές τεχνικές μηχανικής μάθησης σε μια ομάδα δεδομένων, ενώ στην advanced μορφή του περιλαμβάνει και την εφαρμογή τεχνικών clustering στα δεδομένα .

Στη συνέχεια ακολουθούν διάφορες εικόνες που περιγράφουν καλύτερα τη λειτουργία του WEKA Experimenter.

3.1 Simple Mode



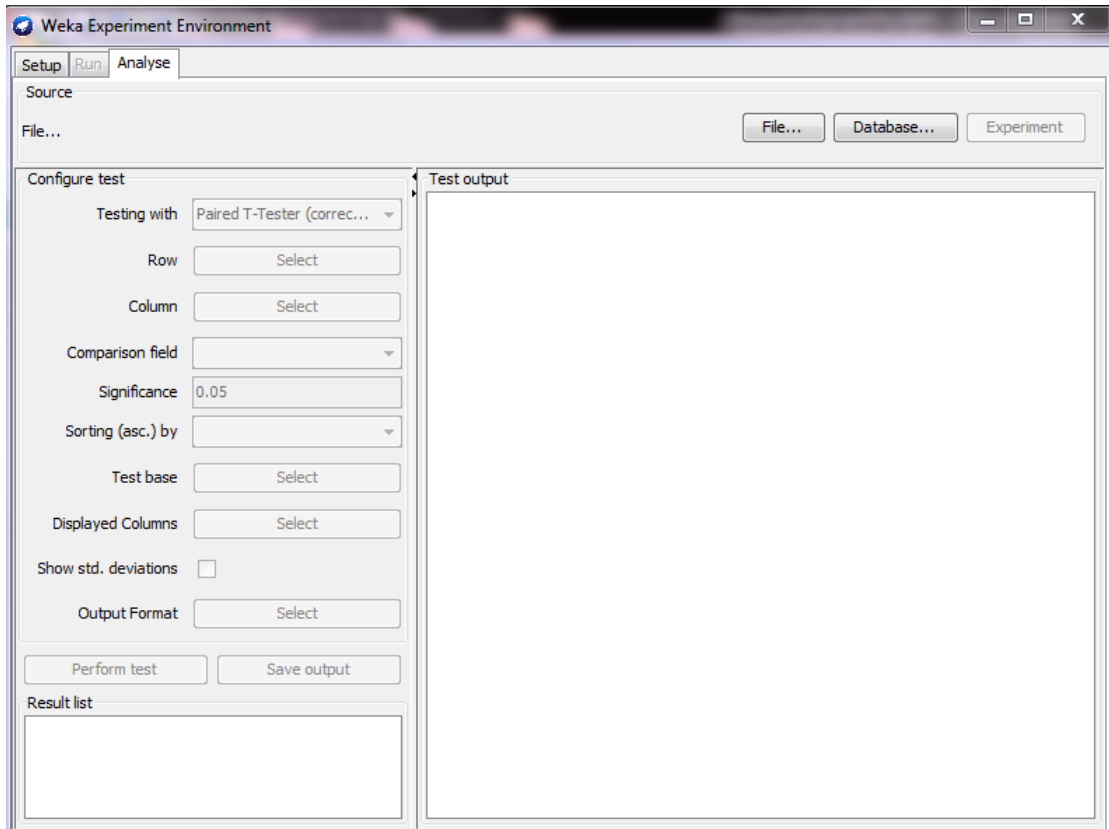
Εικόνα 3.1

Η εικόνα 3.1 είναι η πρώτη εικόνα που εμφανίζεται στους χρήστες μόλις επιλέξουν να ανοίξουν το περιβάλλον εργασίας του WEKA Experimenter. Χαρακτηριστικά βλέπουμε ότι υπάρχουν 3 βασικές καρτέλες στην κορυφή που περιγράφουν τις βασικές λειτουργίες του περιβάλλοντος, που επιλέχθηκε.

Αρχικά υπάρχει η **setup** λειτουργία που επιτρέπει στους χρήστες να φορτώσουν ένα πείραμα-experiment που έχουν αποθηκευμένο στο σύστημά τους ή να δημιουργήσουν ένα καινούριο και να το αποθηκεύσουν. Επιπλέον εύκολα μπορούμε να διακρίνουμε και τις καρτέλες επιλογής simple ή advanced ανάλογα με το είδος της εργασίας και επεξεργασίας που θέλουν να κάνουν στην ομάδα των δεδομένων τους.

Έπειτα υπάρχει η **run** λειτουργία που κατά την εκκίνηση του WEKA Experimenter είναι ανενεργή, γιατί οι χρήστες πρέπει να φορτώσουν δεδομένα στην πλατφόρμα τους και η **run** θα γίνει ενεργή μόλις ολοκληρωθεί αυτή η διαδικασία.

Τέλος υπάρχει και η λειτουργία **analyse**. Οι χρήστες στο σημείο αυτό μπορούν να κάνουν ανάλυση των αποτελεσμάτων, αφού έχουν φορτώσει τα δεδομένα και τρέξουν κάποιους αλγόριθμους. Ακολουθεί μια εικόνα που περιγράφει καλύτερα τη λειτουργία της **analyse**.

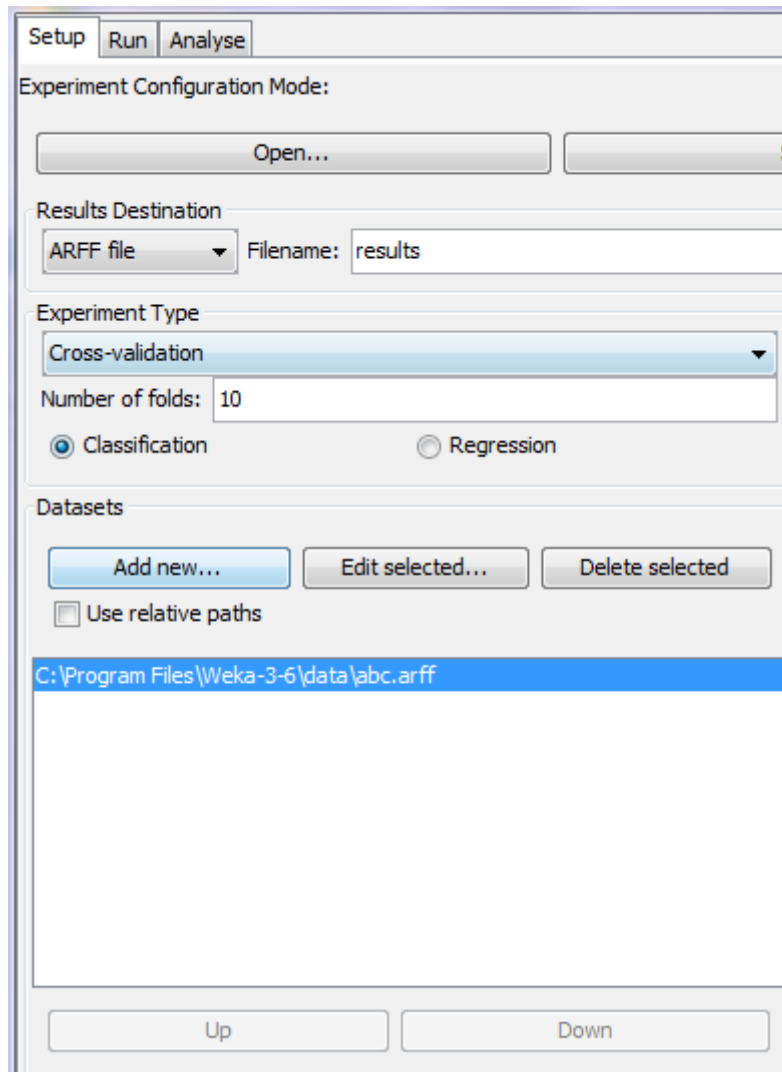


Εικόνα 3.2

Μέσω της εικόνας 3.2 βλέπουμε τι επιλογές παρέχει στους χρήστες η συγκεκριμένη λειτουργία. Πιο συγκεκριμένα βλέπουμε ότι οι χρήστες μπορούν να επιλέξουν συγκεκριμένη γραμμή και στήλη των δεδομένων τους, να συγκρίνουν διάφορα πεδία τιμών, να ορίσουν πόσο σημαντικό είναι ένα attribute κατά την εκτέλεση των αλγορίθμων. Επίσης μπορούν να επιλέξουν να εφαρμοστούν οι αλγόριθμοι με κάποια απόκλιση, καθώς και να καθορίσουν τη μορφή των εξαγόμενων αποτελεσμάτων.

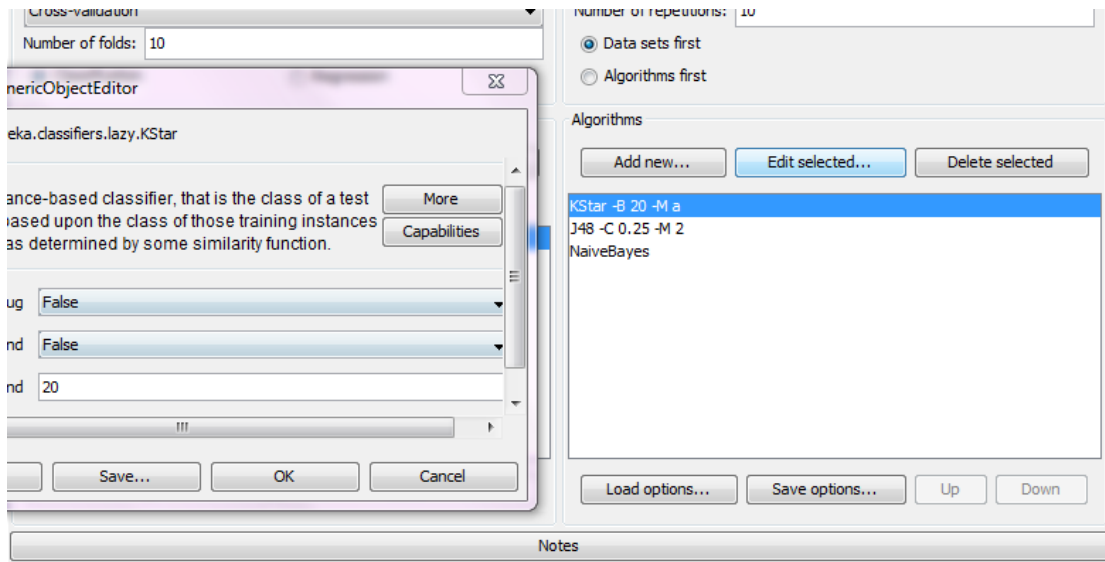
Επανερχόμαστε τώρα στη λειτουργία **setup** που είναι το πρώτο βήμα που πρέπει να κάνουν οι χρήστες για να μπορέσουν να δουλέψουν με το WEKA Experimenter. Πάμε να δούμε τι δυνατότητες παρέχει μέσα από ένα παράδειγμα εφαρμογής των αλγορίθμων μηχανικής μάθησης. Η εικόνα που ακολουθεί δείχνει το 1^ο μέρος του **setup** όπου έχουμε φορτώσει τα δεδομένα από το αρχείο μας **data.arff**, στη συνέχεια

επιλέγουμε την τεχνική του πειράματος (cross-validation) και τέλος επιλέγουμε τα αποτελέσματα του πειράματος να αποθηκευτούν σε ένα αρχείο με όνομα **results.arff**. Ακόμη βλέπουμε ότι επιλέξαμε το **Classification** και όχι το **Regression**, ανάλογα με τον τύπο των δεδομένων που έχουμε για επεξεργασία και το είδος των αλγορίθμων που θέλουμε να εφαρμόσουμε. Είναι προτιμότερο να γίνεται η αποθήκευση σε αρχεία τύπου arff γιατί δημιουργούνται εύκολα μόνο με τις κλάσεις που παρέχει το WEKA και αυτό μπορεί να αποδειχθεί χρήσιμο σε χρονοβόρα πειράματα.



Εικόνα 3.3

Στην εικόνα 3.4 που ακολουθεί φαίνεται το 2^ο μέρος της διαδικασίας δημιουργίας του παραδείγματός μας.



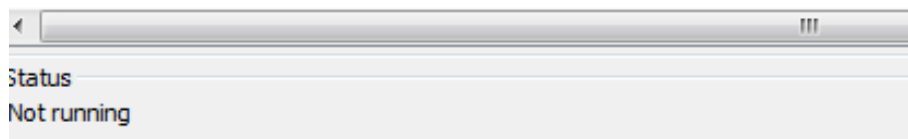
Εικόνα 3.4

Χαρακτηριστικά εφαρμόσαμε τον classifier lazyKstar, στη συνέχεια τον αλγόριθμο δομής δέντρου J-48 και τέλος τον αλγόριθμο NaiveBayes. Ακόμη βλέπουμε ότι για κάθε αλγόριθμο που εφαρμόζουμε, εμφανίζεται ένα αναδυόμενο παράθυρο με διάφορες επιπρόσθετες ρυθμίσεις, στο παράδειγμά μας αφήσαμε τις τιμές των παραμέτρων όπως τις είχε η πλατφόρμα. Επιπλέον μπορούμε με την επιλογή **Add new** να προσθέσουμε ένα νέο αλγόριθμο, με την επιλογή **Edit selected** να τον τροποποιήσουμε και με την επιλογή **delete selected** να τον διαγράψουμε.

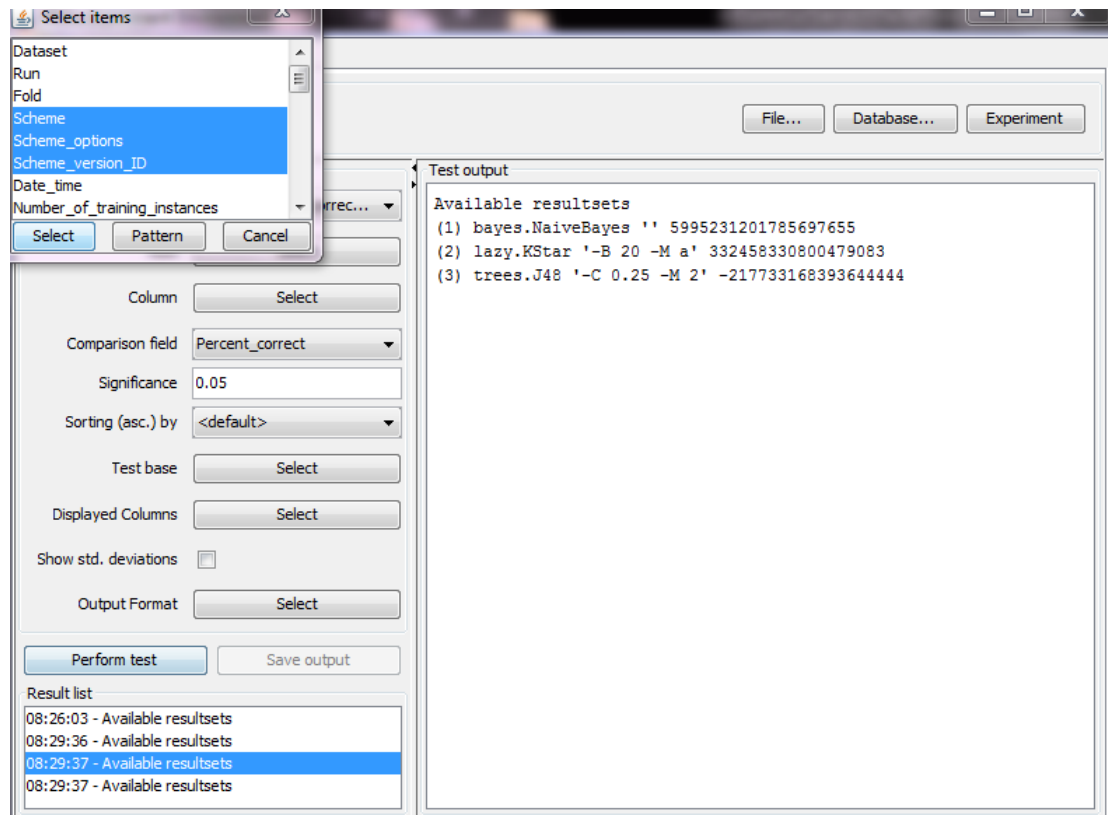
Η επόμενη εικόνα(3.5) που ακολουθεί είναι από τη λειτουργία **run**. Βεβαίως μόλις πατήσουμε **start** χρειάζεται να περιμένουμε λίγη ώρα μέχρι να τρέξουν όλοι οι αλγόριθμοι και να εμφανιστούν τα αποτελέσματα. Η εικόνα(3.6) μας δείχνει τα αποτελέσματα που εμφανίζονται στην καρτέλα της λειτουργίας **analyze**, καθώς και τις παραμέτρους που μπορούμε να εφαρμόσουμε στα αποτελέσματα.

Βελτιστοποίηση διαχείρισης δικτύων με χρήση τεχνικών μηχανικής μάθησης

```
08:24:02: Started
08:24:02: weka.classifiers.bayes.NaiveBayes: Cannot handle string attributes!
08:24:02: Interrupted
08:24:02: There was 1 error
08:24:33: Started
08:25:25: Finished
08:25:25: There were 0 errors
```



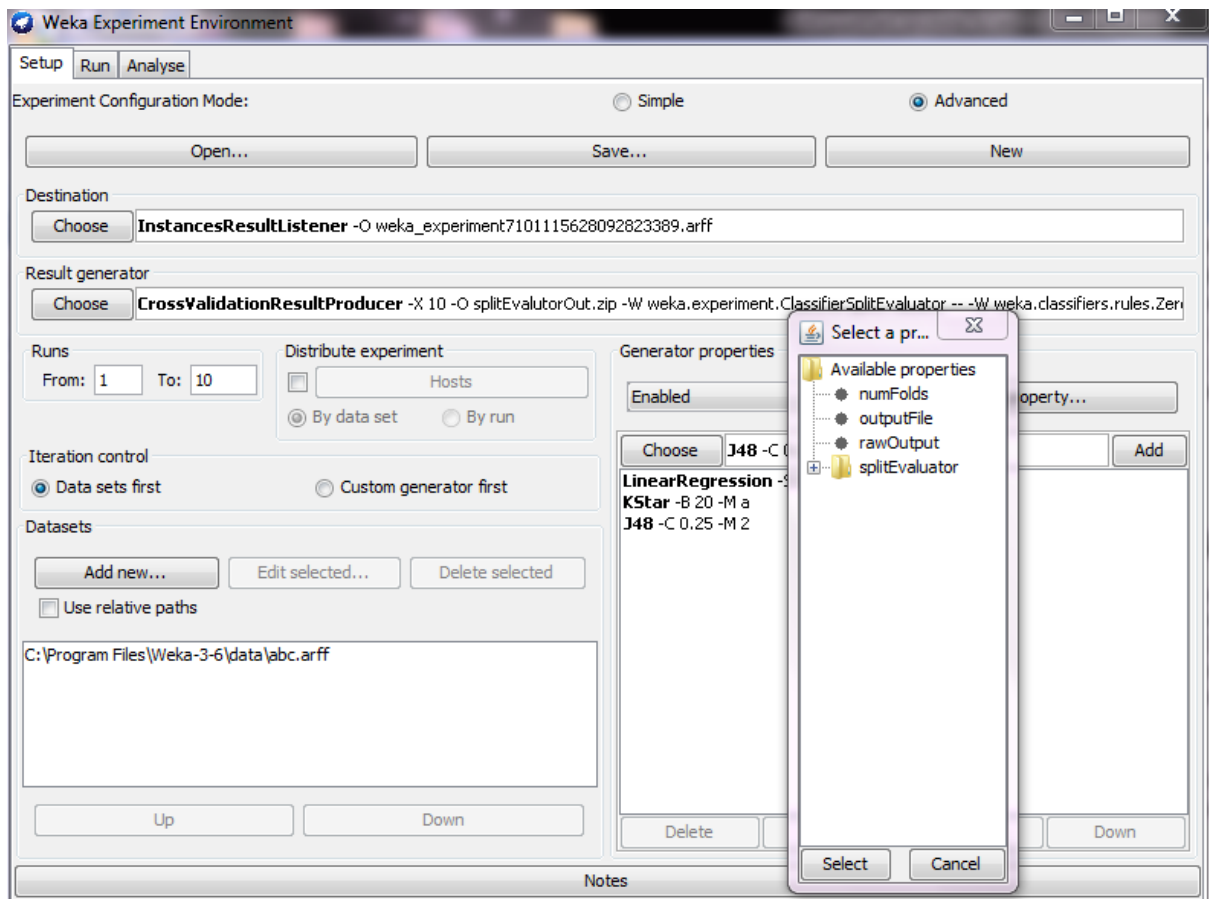
Εικόνα 3.5



Εικόνα 3.6

3.2 Advanced Mode

Στην εργασία μας ασχοληθήκαμε μόνο με πειράματα τύπου simple. Όπως είδαμε και προηγουμένως υπάρχουν και τα πειράματα τύπου advanced, που οι χρήστες μπορούν να εφαρμόσουν στα δεδομένα τους πέρα από τους classifiers και ορισμένες τεχνικές clustering. Η εικόνα (3.7) που ακολουθεί, περιγράφει με επαρκή τρόπο όλα τα απαραίτητα στοιχεία που πρέπει να γνωρίζουν οι χρήστες για να χρησιμοποιήσουν το WEKA Experimenter με πιο σύνθετο-advanced τρόπο.



Εικόνα 3.7

4. WEKA Knowledge Flow

Το WEKA Knowledge Flow είναι το 3^ο περιβάλλον εργασίας του WEKA. Η λειτουργικότητα που παρέχει στους χρήστες του, παρουσιάζει πολλά κοινά χαρακτηριστικά με το WEKA Explorer, όμως υπάρχουν κάποιες λειτουργίες που υπάρχουν αντίστοιχα στο ένα περιβάλλον εργασίας και δεν υπάρχουν στο άλλο.

Το WEKA Knowledge Flow παρουσιάζει στους χρήστες του ένα περιβάλλον εργασίας που βασίζεται στο data-flow(ροή δεδομένων). Συγκεκριμένα οι χρήστες του WEKA μπορούν να επιλέξουν τα κομμάτια του WEKA, που τους ενδιαφέρουν και εστιάζουν στο πρόβλημά τους, να τα τοποθετήσουν στο ειδικό layout που τους παρέχεται, με σκοπό να τα συνδέσουν μεταξύ τους και να δημιουργήσουν το λεγόμενο knowledge flow(ροή γνώσης). Σκοπός της ροής γνώσης είναι η επεξεργασία και η ανάλυση των δεδομένων.

Μια βασική διαφορά ανάμεσα στο WEKA Knowledge Flow και το WEKA Explorer, είναι ότι το πρώτο μπορεί να χειριστεί δεδομένα και σε ομάδες, π.χ ένα αρχείο arff, και αυξητικά, δηλαδή όταν αλλάζουν οι τιμές ενός χαρακτηριστικού από επεξεργασία κάποιου classifier. Αντίθετα το WEKA Explorer μπορεί να χειριστεί δεδομένα μόνο σε ομάδες. Προφανώς για να υπάρχει μάθηση με αυξητικό τρόπο θα πρέπει να υπάρχουν διαθέσιμοι classifiers που να είναι υπεύθυνοι γι αυτό.

4.1 Classifiers και Features

Ακολουθούν οι σημαντικότεροι classifiers που υπάρχουν στο συγκεκριμένο περιβάλλον εργασίας:

- **AODE**
- **IB1**

- **IBK**
- **NaiveBayesUpdateable**
- **Winnow**

Ακόμη υπάρχουν και 2 meta-classifiers:

- **RacedIncrementalLogitBoost**(για τη δημιουργία γνώσης από διακριτές τιμές δεδομένων).
- **LWL**(locally weighted learning).

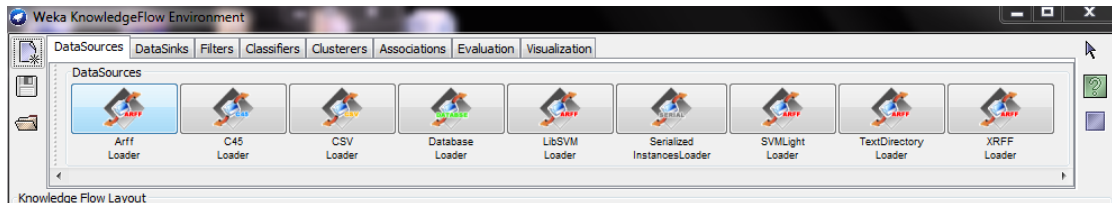
Ακολουθούν τα χαρακτηριστικά που παρέχει το WEKA Knowledge Flow στους χρήστες του.

- Επεξεργασία δεδομένων σε ομάδες ή αυξητικά.
- Παράλληλη επεξεργασία ομάδων ή ροών δεδομένων, η κάθε μια βέβαια στο δικό της κομμάτι.
- Συγκώνευση φίλτρων για την αποδοτικότερη εμφάνιση και επεξεργασία των δεδομένων.
- Προβολή των μοντέλων μάθησης που παράγονται από τους classifiers με τη χρήση της τεχνικής cross validation.
- Δυνατότητα προβολής της απόδοσης των classifiers κατά τη διάρκεια εφαρμογής τους στα δεδομένα.
- Ευδιάκριτο data-flow layout.

4.2 Knowledge Flow Components

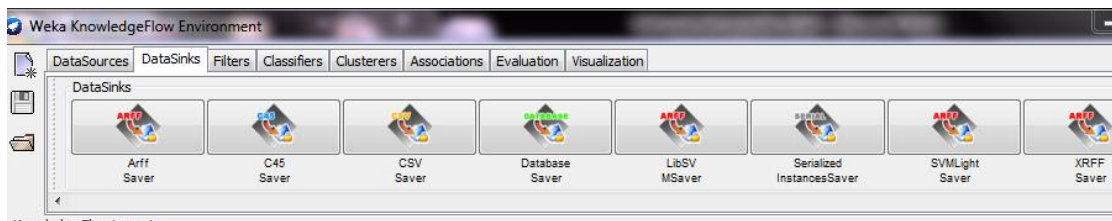
Στην ενότητα αυτή θα παρουσιάσουμε τα βασικά συστατικά του WEKA Knowledge Flow που χρησιμοποιούν οι χρήστες του.

- **Datasources**(όλες οι διαθέσιμες πηγές από όπου οι χρήστες μπορούν να φορτώσουν τα δεδομένα τους).



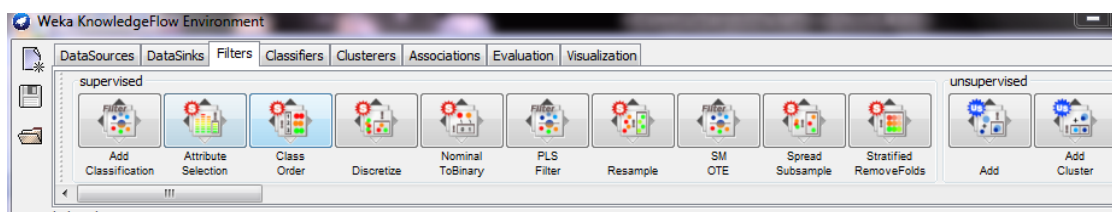
Εικόνα 4.1

- **Datasinks**(προορισμοί αποθήκευσης μοντέλων επεξεργασίας και δεδομένων)



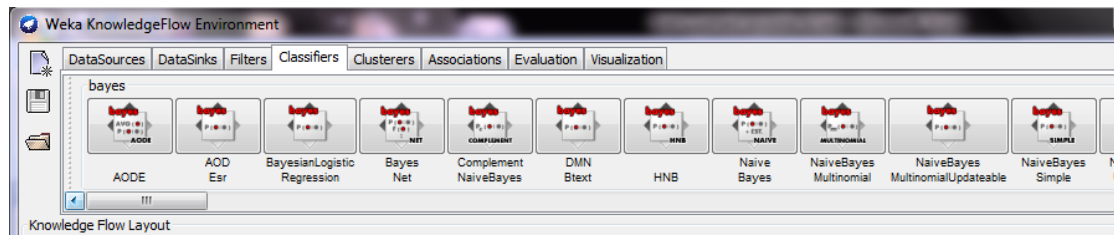
Εικόνα 4.2

- **Filters**



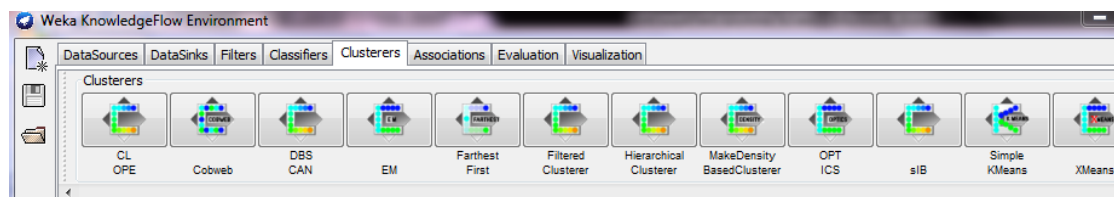
Εικόνα 4.3

- **Classifiers**



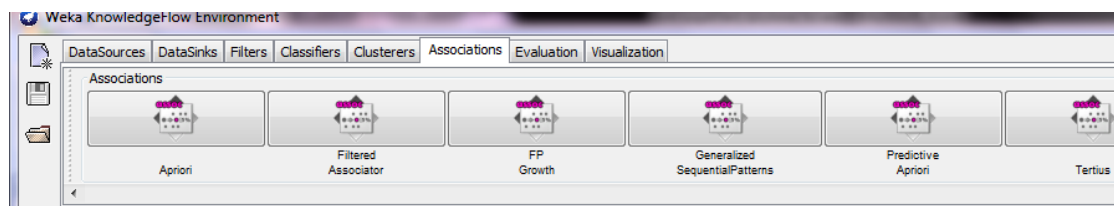
Εικόνα 4.4

- **Clusterers**



Εικόνα 4.5

- **Associations**

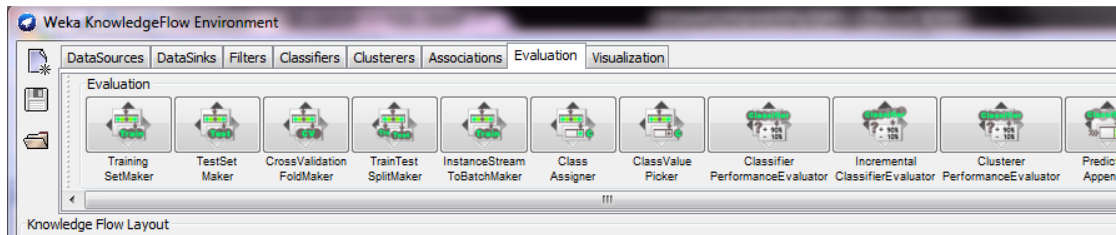


Εικόνα 4.6

- **Evaluation**

1. **TrainingSetMaker**(μετατροπή μιας ομάδας δεδομένων σε training)
2. **TestSetMaker**(μετατροπή μιας ομάδας δεδομένων σε test)
3. **CrossValidationFoldMaker**(διαχωρισμός ομάδων δεδομένων σε πεδία)
4. **TrainTestSplitMaker**(διαχωρισμός ομάδων δεδομένων σε training και test set αντίστοιχα)
5. **ClassAssigner**(ανάθεση ενός χαρακτηριστικού ως κύρια κλάση)
6. **ClassifierPerformanceEvaluator**(αξιολόγηση της απόδοσης ενός classifier)

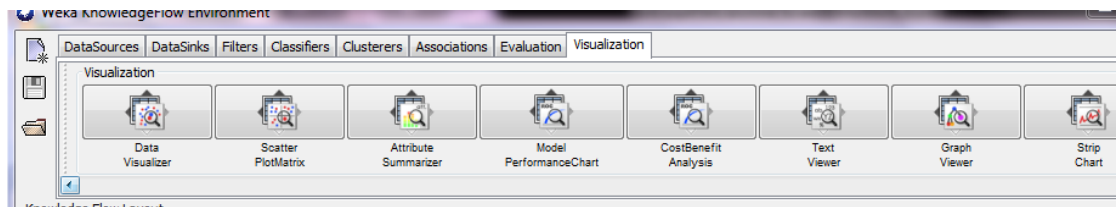
7. **IncrementalClassifierEvaluator**(αξιολόγηση απόδοσης classifier που λειτουργεί αυξητικά)
8. **ClustererPerformanceEvaluator**(αξιολόγηση απόδοσης ενός clusterer)
9. **PredictionAppender**(βοηθά τον classifier να κάνει πρόβλεψη σε ένα test set)



Εικόνα 4.7

- **Visualization**

1. **DataVisualizer**(δημιουργεί 2-D γράφημα)
2. **AttributeSummarizer**(δημιουργεί διάγραμμα που απεικονίζει τα attributes των δεδομένων)
3. **ModelPerformanceChart**(δημιουργεί διάγραμμα που απεικονίζει το threshold)
4. **TextViewer**(Στατιστική αναπαράσταση τιμών κειμένου, προβολή απόδοσης classifiers)
5. **GraphViewer**(Απεικόνιση μοντέλων που έχουν τη δομή δέντρου)

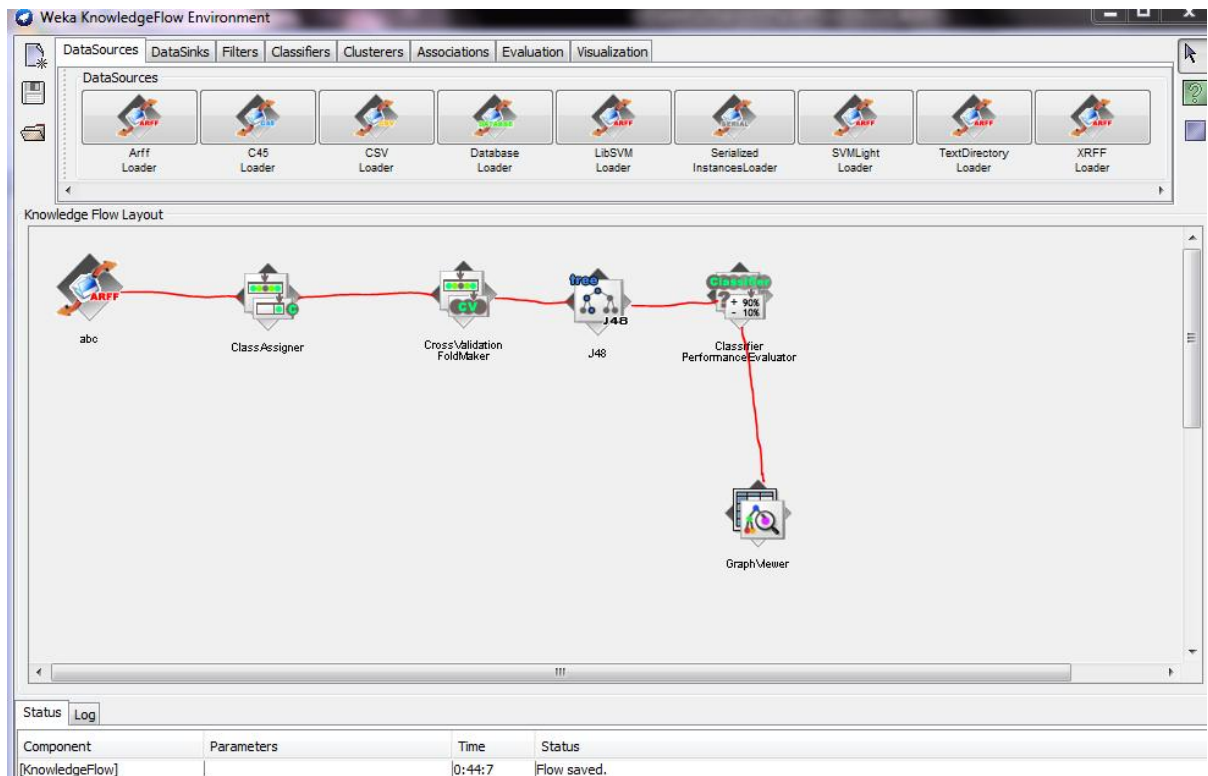


Εικόνα 4.8

4.3 Παράδειγμα χρήσης

Ακολουθεί ένα παράδειγμα χρήσης του WEKA Knowledge Flow, όπου εφαρμόσαμε τον classifier **J48**(που είναι μορφής δέντρου), στο αρχείο **data.arff**, με χρήση της τεχνικής **cross-validation**. Τα εργαλεία που χρησιμοποιήθηκαν είναι ο **ArffLoader** για τη φόρτωση του αρχείου, ο **ClassAssigner** για την ανάθεση ενός attribute ως κύρια κλάση, ο **J48 CrossValidationFoldMaker** για χρήση της τεχνικής **cross-validation**, ο **ClassifierPerformanceEvaluator** για την αξιολόγηση του μοντέλου και ο **GraphViewer** που είναι ο καταλληλότερος για την γραφική αναπαράσταση μοντέλων δέντρου.

Το παράδειγμά μας εξηγείται καλύτερα μέσω της εικόνας 4.9 που ακολουθεί.

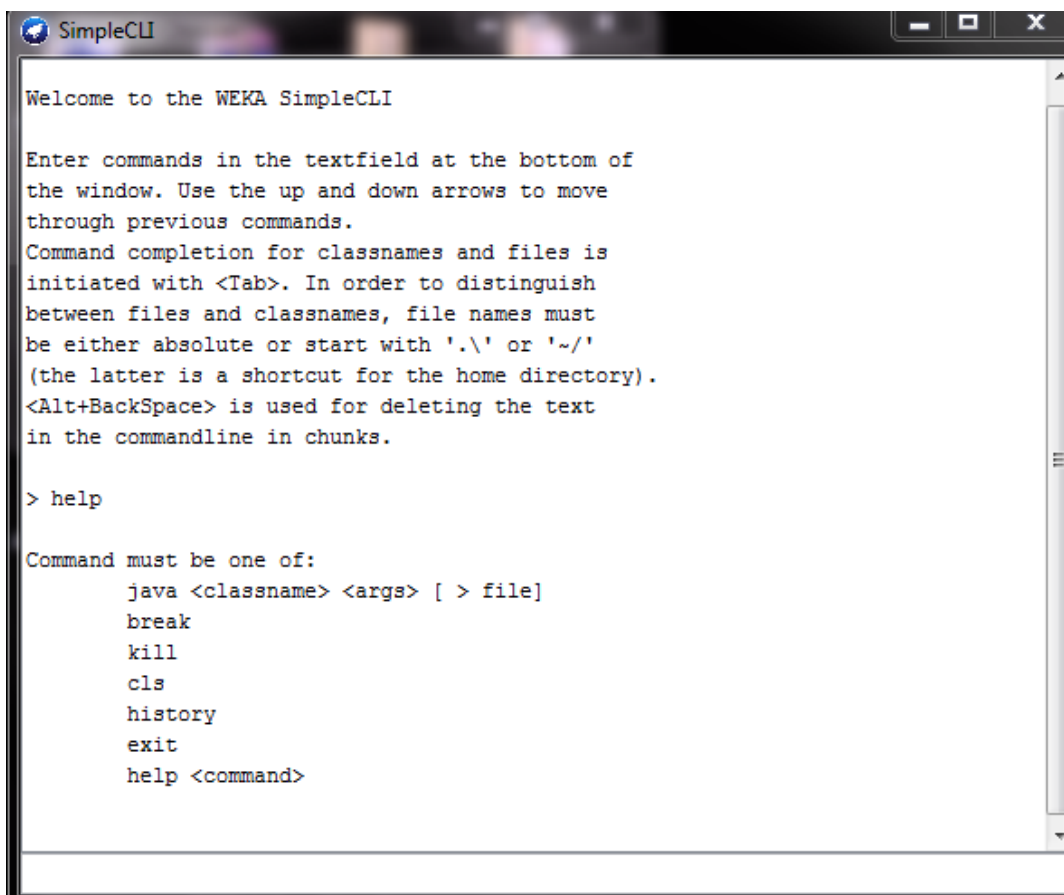


Εικόνα4.9

5. WEKA Command Line (CLI)

Το WEKA Command Line αποτελεί το 4^ο περιβάλλον εργασίας του WEKA. Στο σημείο αυτό θα πρέπει να τονίσουμε ότι η λειτουργικότητα που προσφέρει το WEKA Simple CLI είναι ίδια με το WEKA Explorer και το WEKA Experimenter. Το WEKA Simple CLI χρησιμοποιείται κυρίως από λειτουργικά συστήματα που δεν μπορούν να υποστηρίξουν το WEKA Graphical User Interface(GUI) λόγω των απαιτήσεων σε μνήμη.

Κατά την εκκίνηση του WEKA 3.6.9 μας εμφανίζεται η αρχική οθόνη που φαίνεται παρακάτω. Δεν είναι τίποτα άλλο απ το Simple Command Line του WEKA.



```
SimpleCLI
Welcome to the WEKA SimpleCLI

Enter commands in the textfield at the bottom of
the window. Use the up and down arrows to move
through previous commands.
Command completion for classnames and files is
initiated with <Tab>. In order to distinguish
between files and classnames, file names must
be either absolute or start with './' or '~/ '
(the latter is a shortcut for the home directory).
<Alt+BackSpace> is used for deleting the text
in the commandline in chunks.

> help

Command must be one of:
  java <classname> <args> [ > file]
  break
  kill
  cls
  history
  exit
  help <command>
```

Εικόνα 5.1

Οι ακόλουθες εντολές είναι διαθέσιμες στο Simple CLI

- **java <classname> [<args>]**

Καλεί μια κλάση της Java με τα δοθέντα χαρακτηριστικά (αν υπάρχουν βέβαια).

- **break**

Σταματάει την υπάρχουσα δέσμη ενεργειών. Για παράδειγμα όταν τρέχει ένας classifier(ταξινομητής δεδομένων).

- **kill**

Σταματάει την δέσμη ενεργειών με πιο σκληρό- απότομο τρόπο.

- **cls**

Καθαρισμός δεδομένων.

- **Exit**

Έξοδος από το Simple CLI.

- **Help [<command>]**

Επισκόπηση όλων των διαθέσιμων εντολών ή βοήθεια για την εντολή που είναι μέσα στα <>.

Επίσης μια πολύ σημαντική δυνατότητα που μας δίνει το WEKA φαίνεται μέσω του παρακάτω παραδείγματος:

```
Java weka.classifiers.trees.J48 -t data.arff > resj48.txt
```

Η παραπάνω εντολή αφορά την εκτέλεση του classifier J48 στο αρχείο με όνομα **data** που είναι arff και την εμφάνιση των αποτελεσμάτων σε ένα αρχείο με όνομα **resj48** που είναι αρχείο κειμένου και μας δίνει την δυνατότητα, για ευκολότερη επεξεργασία και μελέτη των αποτελεσμάτων.

Τώρα πρέπει να κάνουμε κάποιες παραδοχές που αφορούν τα δεδομένα που περιέχονται στο αρχείο arff.

1. Αλφαριθμητικά δεδομένα είναι τύπου string και πρέπει να περιέχονται ανάμεσα σε «».
2. Όλοι οι αριθμοί, που αναφέρονται σε δεδομένα είναι τύπου numeric.
3. Δεδομένα που προκύπτουν από μια λίστα τιμών με συγκεκριμένες τιμές πχ {yes,no,yes,no} και είναι nominal.
4. Ακόμη το WEKA προσφέρει τη δυνατότητα για τιμές που προκύπτουν από μετρήσεις, να είναι τύπου real.
5. Τέλος δεδομένα που αφορούν σε ημερομηνίες πραγματικού χρόνου είναι τύπου date.

Η μηχανική μάθηση στο WEKA, μέσω του Simple CLI , αφορά τη χρήση αλγορίθμων μηχανικής μάθησης , οι οποίοι αλγόριθμοι ονομάζονται classifiers σε όλα τα 4 περιβάλλοντα εργασίας, που

ορίστηκαν και αναλύθηκαν σ αυτή την εργασία . Η διαφορά έγκειται στο γεγονός ότι στο Simple CLI υπάρχει η δυνατότητα μεγαλύτερης ευελιξίας και η διαδικασία είναι <<φανερή>> ενώ στα προηγούμενα περιβάλλοντα, ο χρήστης απλά επιλέγει τον classifier που θέλει να τρέξει και βλέπει το αποτέλεσμα, ενώ η διαδικασία μένει κρυφή.

5.1 Classify με τη χρήση εντολών

Κάθε αλγόριθμος μάθησης στο WEKA προέρχεται από το `weka.classifiers.Classifier` . Ακολουθούν χαρακτηριστικά παραδείγματα classifiers και οι εντολές εκτέλεσης τους.

- **Trees.J48** (Απόφαση με τη χρήση δέντρου)
`Java.weka.classifiers.trees.J48 -t data.arff`
- **bayes.NaiveBayes**
`Java.weka.classifiers.bayes.NaiveBayes -t data.arff`
- **functions.Logistic** (Στατιστικό μοντέλο αναδρόμησης)
`Java.weka.classifiers.functions.Logistics -t data.arff`
- **lazy.Kstar** (Υπολογίζει την εντροπία των τιμών)
`Java.weka.classifiers.lazy.Kstar -t data.arff`
- **lazy.Ibk** (Υπολογισμός τιμής με βάση μια γειτονική)
`Java.weka.classifiers.lazy.Ibk -t data.arff`

Το κομμάτι των εντολών **-t** παίζει το μεγαλύτερο ρόλο κατά τη σύνταξη και την εκτέλεση των classifiers. Αυτό συμβαίνει γιατί το **-t** είναι παράμετρος. Στη συνέχεια ακολουθεί μια λίστα των σημαντικότερων παραμέτρων που υπάρχουν στους classifiers του WEKA και δίνονται σύντομες περιγραφές το πώς επηρεάζουν το αποτέλεσμα εκτέλεσης της εντολής που περιέχει την κάθε παράμετρο.

- **-t** επικεντρώνεται στο σύνολο των υπο-εξέταση δεδομένων(training file arff format).
- **-T** επικεντρώνεται σ ένα υποσύνολο των δεδομένων προς εξέταση (test file arff format). By default το υποσύνολο των δεδομένων αποτελείται από 10 τιμές δεδομένων.
- **-x** βοηθά το χρήστη να προσαρμόσει τον αριθμό των τιμών δεδομένων και να το αλλάξει από το 10 που είναι by default.

- **-c** Αυτή η παράμετρος καθορίζει την τάξη μεταβλητή (ένα attribute) και δείχνει σε μια τιμή (instance) του στοιχείου.
- **-d** Αυτή η παράμετρος αποθηκεύει το μοντέλο μάθησης που προκύπτει από την επεξεργασία ενός υποσυνόλου δεδομένων, μέσω κάποιου classifier . Αυτό συμβαίνει γιατί προκύπτουν διαφορετικά μοντέλα μάθησης από την εκτέλεση διαφορετικών classifiers.
- **-l** χρησιμοποιείται για τη <<φόρτωση>> ενός αποθηκευμένου μοντέλου.
- **-p** χρησιμοποιείται σε συνδυασμό με το -T και χρησιμοποιείται για την πρόβλεψη των τιμών των δεδομένων και σε ποιο attribute ταιριάζουν περισσότερο.
- **-I** μας δίνει περισσότερες λεπτομέρειες σε σχέση με την πρόβλεψη των επόμενων τιμών, π.χ ποσοστό επιτυχούς πρόβλεψης. Έχει να κάνει κυρίως με το αποτέλεσμα εκτέλεσης του κάθε classifier και όχι τόσο με τη σύνταξη της εντολής για την εκτέλεσή τους.

Όπως εύκολα γίνεται αντιληπτό από τα παραπάνω προκύπτουν διάφοροι συνδυασμοί εκτέλεσης των classifiers , ανάλογα με τις παραμέτρους που δίνει ο χρήστης σε κάθε εντολή εκτέλεσης. Αυτό που κάνει μοναδικά όμως το WEKA Simple CLI, σε σχέση με τα προηγούμενα περιβάλλοντα εργασίας (WEKA Explorer και WEKA Experimenter) είναι ότι μας δίνει σε βάθος την όλη διαδικασία που ακολουθείται. Για παράδειγμα εφαρμόζουμε στον WEKA Explorer τον J48 classifier, βλέπουμε το αποτέλεσμα αλλά αγνοούμε πλήρως την ύπαρξη της παραμέτρου - i που επηρεάζει την εμφάνιση του αποτελέσματος.

ΣΗΜΕΙΩΣΗ: Δεν έβαλα εικόνες στο κεφάλαιο αυτό της εργασίας γιατί είναι οι ίδιες με αυτές που δόθηκαν από την εκτέλεση των classifiers στα προηγούμενα περιβάλλοντα εργασίας του WEKA.

6. Markov based prediction model

Τα μοντέλα του Markov έχουν χρησιμοποιηθεί για την κατανόηση και μελέτη στοχαστικών διαδικασιών. Δείχνουν να ταιριάζουν τέλεια για τη μοντελοποίηση και πρόβλεψη της συμπεριφοράς της κίνησης ενός χρήστη σε κυψελωτά συστήματα. Σε γενικές γραμμές, τα δεδομένα εισόδου για αυτά τα προβλήματα, είναι η συχνότητα των επισκέψεων στα σημεία εισόδου στα οποία έχει πρόσβαση ο χρήστης. Ο σκοπός είναι να κατασκευάσουμε ένα μοντέλο Markov που να μπορεί να προβλέψει την επόμενη κίνηση του χρήστη, δηλαδή το επόμενο AP, το οποίο θα του παρέχει πρόσβαση στο δίκτυο.

6.1 Εισαγωγή στις αλυσίδες Markov

Η **αλυσίδα Μαρκόφ**, ή **Μαρκοβιανή αλυσίδα**, που πήρε το όνομα της από τον Αντρέι Μαρκόβ, είναι ένα μαθηματικό σύστημα που μεταβάλλεται από μια κατάσταση σε μια άλλη, ανάμεσα σε ένα πεπερασμένο αριθμό καταστάσεων. Είναι μια τυχαία διαδικασία που δε διατηρεί μνήμη για τις προηγούμενες μεταβολές: Η επόμενη κατάσταση εξαρτάται μόνο από την τωρινή κατάσταση και σε καμιά περίπτωση από αυτές που προηγήθηκαν. Αυτό το συγκεκριμένο είδος "αμνησίας" ονομάζεται μαρκοβιανή ιδιότητα. Οι Μαρκοβιανές Αλυσίδες έχουν πολλές εφαρμογές ως στατιστικά μοντέλα καθημερινών διαδικασιών.

Η **Μαρκοβιανή αλυσίδα** είναι μια στοχαστική διαδικασία με τη μαρκοβιανή ιδιότητα για ένα πεπερασμένο ή μετρήσιμο χώρο καταστάσεων. Ο όρος "**Μαρκοβιανή αλυσίδα**" αναφέρεται στην αλληλουχία (ή αλυσίδα) των καταστάσεων μέσω των οποίων κινείται μια τέτοια διαδικασία. Συνήθως μια **Μαρκοβιανή αλυσίδα** ορίζεται για μια διακριτή συλλογή χρόνων (**Μαρκοβιανή αλυσίδα** διακριτών χρόνων), παρόλο που μερικοί συγγραφείς χρησιμοποιούν την ίδια ορολογία για να αναφερθούν σε **Μαρκοβιανή αλυσίδα** συνεχής χρόνου. Η χρήση του όρου στη μεθοδολογία **Μαρκοβιανής αλυσίδας** Μόντε Κάρλο καλύπτει περιπτώσεις όπου η διαδικασία βρίσκεται σε διακριτό χρόνο (διακριτά αλγοριθμικά βήματα) με ένα συνεχές χώρο καταστάσεων. Οι πληροφορίες που ακολουθούν επικεντρώνονται στην περίπτωση διακριτού χρόνου και διακριτού χώρου καταστάσεων.

Μια τυχαία διαδικασία διακριτού χρόνου περιλαμβάνει ένα σύστημα που βρίσκεται σε μια συγκεκριμένη κατάσταση σε κάθε βήμα, με την

κατάσταση να μεταβάλλεται τυχαία μεταξύ των βημάτων. Τα βήματα συχνά θεωρούνται ως στιγμές στο χρόνο αλλά μπορούν εξίσου να αναφέρονται σε φυσική απόσταση ή οποιαδήποτε άλλη διακριτή μέτρηση: τυπικά τα βήματα είναι οι ακέραιοι ή φυσικοί αριθμοί και η τυχαία διαδικασία είναι η χαρτογράφηση τους σε καταστάσεις. Η Μαρκοβιανή ιδιότητα δηλώνει ότι η υποθετική πιθανότητα κατανομής του συστήματος στο επόμενο βήμα (και κατά βάση, σε όλα τα μελλοντικά βήματα) εξαρτάται μόνο από την παρούσα κατάσταση του συστήματος και όχι αθροιστικά από την κατάσταση του συστήματος σε προηγούμενα βήματα.

Καθώς το σύστημα μεταβάλλεται τυχαία, είναι γενικά αδύνατο να προβλεφθεί με βεβαιότητα η κατάσταση μιας Μαρκοβιανής αλυσίδας σε ένα δεδομένο μελλοντικό σημείο. Παρ' όλα αυτά, οι στατιστικές ιδιότητες του μέλλοντος του συστήματος μπορούν να προβλεφθούν. Σε πολλές εφαρμογές, αυτές οι στατιστικές ιδιότητες είναι σημαντικές. Οι αλλαγές κατάστασης του συστήματος ονομάζονται μεταβάσεις και οι πιθανότητες που σχετίζονται με τις διάφορες μεταβατικές καταστάσεις ονομάζονται πιθανότητες μετάβασης. Η διαδικασία χαρακτηρίζεται από ένα χώρο καταστάσεων, μια μήτρα μετάβασης που περιγράφει τις πιθανότητες μιας συγκεκριμένης μετάβασης και μια αρχική κατάσταση ή αρχική κατανομή στο χώρο καταστάσεων. Κατά συνθήκη, θεωρούμε ότι όλες οι δυνατές καταστάσεις και μεταβάσεις έχουν συμπεριληφθεί στον ορισμό των διαδικασιών, ώστε υπάρχει πάντα μια επόμενη κατάσταση και η διαδικασία συνεχίζεται για πάντα.

Στη συνέχεια ακολουθεί ο μαθηματικός ορισμός:

Μια Μαρκοβιανή Αλυσίδα είναι μια ακολουθία τυχαίων μεταβλητών X_1, X_2, X_3, \dots με τη Μαρκοβιανή ιδιότητα, δηλαδή με δεδομένη την παρούσα κατάσταση, οι παλαιότερες και οι μελλοντικές καταστάσεις είναι ανεξάρτητες. Ορίζουμε:

$\Pr(\mathbf{X}_{n+1}=\mathbf{x} \mid \mathbf{X}_1=\mathbf{x}_1, \mathbf{X}_2=\mathbf{x}_2, \dots, \mathbf{X}^n=\mathbf{x}_n) = \Pr(\mathbf{X}_{n+1}=\mathbf{x} \mid \mathbf{X}_n=\mathbf{x}_n)$ Οι πιθανές τιμές των X_i σχηματίζουν ένα αριθμησιμο σύνολο S που ονομάζουμε χώρο-καταστάσεων της αλυσίδας.

Οι Μαρκοβιανές Αλυσίδες συχνά περιγράφονται από ένα κατευθυνόμενο γράφημα που οι άκρες του επιγράφουν τις πιθανότητες μετάβασης από τη μια κατάσταση στις άλλες.

Μια **Μαρκοβιανή Αλυσίδα τάξης m** (ή Μαρκοβιανή Αλυσίδα με μήμη m), όπου το m είναι πεπερασμένο, είναι μια διαδικασία που ικανοποιεί:

$$\Pr(\mathbf{X}_n = \mathbf{x}_n \mid \mathbf{X}_{n-1} = \mathbf{x}_{n-1}, \mathbf{X}_{n-2} = \mathbf{x}_{n-2}, \dots, \mathbf{X}_1 = \mathbf{x}_1) =$$

$$\Pr(\mathbf{X}_n = \mathbf{x}_n \mid \mathbf{X}_{n-1} = \mathbf{x}_{n-1}, \mathbf{X}_{n-2} = \mathbf{x}_{n-2}, \dots, \mathbf{X}_{n-m} = \mathbf{x}_{n-m} \text{ for } n > m$$

Με άλλα λόγια η μελλοντική κατάσταση εξαρτάται από τις προηγούμενες m καταστάσεις. Είναι δυνατό να κατασκευάσουμε μια αλυσίδα (\mathbf{Y}_n) από (\mathbf{X}_n) που να έχει την κλασική Μαρκοβιανή ιδιότητα παίρνοντας ως χώρο καταστάσεων το σύνολο των m -πλειάδων του X , πχ. $\mathbf{Y}_n = (\mathbf{X}_n, \mathbf{X}_{n-1}, \dots, \mathbf{X}_{n-m+1})$.

Η πιθανότητα να πάει από την κατάσταση \mathbf{i} στην κατάσταση \mathbf{j} σε n χρονικά βήματα είναι:

$$\mathbf{P}_{ij}^{(n)} = \Pr(\mathbf{X}_n = \mathbf{j} \mid \mathbf{X}_0 = \mathbf{i})$$

Το απλό βήμα μετάδοσης

$$\mathbf{P}_{ij} = \Pr(\mathbf{X}_1 = \mathbf{j} \mid \mathbf{X}_0 = \mathbf{i})$$

Για μια ομογενή-χρονικά Μαρκοβιανή Αλυσίδα:

$$\mathbf{P}_{ij}^{(n)} = \Pr(\mathbf{X}_{k+n} = \mathbf{j} \mid \mathbf{X}_k = \mathbf{i})$$

$$\text{Και } \mathbf{P}_{ij} = \Pr(\mathbf{X}_{k+1} = \mathbf{j} \mid \mathbf{X}_k = \mathbf{i})$$

όπου S ο χώρος καταστάσεων της Μαρκοβιανής Αλυσίδας

Η οριακή κατανομή $\Pr(\mathbf{X}_n = \mathbf{x})$ είναι το μοίρασμα των καταστάσεων σε χρόνο n . Η αρχική κατανομή είναι $\Pr(\mathbf{X}_0 = \mathbf{x})$. Η εξέλιξη της διαδικασίας για ένα χρονικό βήμα περιγράφεται από τη σχέση:

$$\Pr(\mathbf{X}_n = \mathbf{j}) = \sum_{\mathbf{r} \in S} \mathbf{p}_{rj} \Pr(\mathbf{X}_{n-1} = \mathbf{r}) = \sum_{\mathbf{r} \in S} \mathbf{p}_{rj}^{(n)} \Pr(\mathbf{X}_0 = \mathbf{r})$$

6.2 Χρήση του WEKA για την κατασκευή του Bayesian Based prediction model

Στο σημείο αυτό πρέπει να τονιστεί το γεγονός ότι το περιβάλλον εργασίας του WEKA δεν υποστηρίζει το μοντέλο Markov, γι αυτό δίνεται θεωρητικά η αντιμετώπιση του προβλήματος με βάση το παραπάνω μοντέλο και η λύση του προβλήματος στο WEKA δίνεται με τη χρήση του μοντέλου NaiveBayes.

Στην μέθοδο **APC** αυτό που βοηθάει το μοντέλο στο να κάνει σωστή πρόβλεψη είναι το **Access Point(AP)**. Η συμμετοχή των ΜΗ είναι μικρή αφού το μόνο που κάνουν είναι να στέλνουν, σε κάθε διαπομπή το αναγνωριστικό του προηγούμενου AP. Αυτή η μέθοδος είναι πολύ χρήσιμη αν κύρια προτεραιότητά μας είναι η οικονομική διαχείριση των πόρων του δικτύου.

Η μέθοδος **MHC** από την άλλη έχει στην κορυφή της πυραμίδας της τα ΜΗ. Κάθε ένα από αυτά κατασκευάζει ένα μοντέλο σύμφωνα με τις πληροφορίες που παίρνει από τις ίδιες του τις κινήσεις. Είναι περισσότερο αξιόπιστη αυτή η μέθοδος γιατί είναι πιο συγκεκριμένη. Για παράδειγμα κινητά τερματικά που κινούνται στην ίδια περιοχή είναι πολύ πιθανό να εμφανίζουν ίδια συμπεριφορά. Το βασικότερο μειονέκτημα είναι ότι τα κινητά τερματικά έχουν λίγη μνήμη, πέφτει εύκολα η μπαταρία και δεν είναι προορισμένα για να τρέχουν έναν αλγόριθμο μηχανικής μάθησης.

Σύμφωνα με τις σχέσεις που δόθηκαν στην ενότητα 6.1 μπορούμε να βρούμε την πιθανότητα μετακίνησης ενός ΜΗ σ ένα AP γνωρίζοντας πράγματα για την προηγούμενη κατάστασή του. Συγκεκριμένα:

Έστω $L = \{L_1, L_2, L_3 \dots\}$ ένα σύνολο από τοποθεσίες και $L = L_1, L_2, L_3$ ένα ιστορικό τοποθεσιών. Η μαρκοβιανή υπόθεση δίνεται από την σχέση που ακολουθεί:

$$P(L_i = 1 | L_1, \dots, L_{i-1}) = P(L_i = 1 | L_{i-n}, \dots, L_{i-1}) \text{ για κάθε } 1 \in L, i > n$$

Η παραπάνω σχέση δηλώνει ότι η μεταβλητή που περιγράφει την πιθανότητα να πάει ένα ΜΗ στον επόμενο AP, είναι μια κατανομή που εξαρτάται μόνο απ το n.

Στη συνέχεια, αν δηλώσουμε L^m το ιστορικό τοποθεσιών ενός ΜΗ τότε ο κανόνας υπολογισμού του μοντέλου θα περιγράφεται από την σχέση που ακολουθεί.

$$P(L_i = 1 \mid L_{i-n}, \dots, L_{i-1}) = \frac{\sum_{m \in M} O(L_{i-n}^m, \dots, L_{i-1}^m, L^m)}{\sum_{m \in M} O(L_{i-n}^m, \dots, L_{i-1}^m, L^m)}$$

Όπου ο τελεστής O βρίσκει τον αριθμό των συμβάντων της πρώτης ακολουθίας co_k και το M αποτελεί τον αριθμό των ΜΗ που εμπλέκονται.

Όταν το μοντέλο είναι έτοιμο να κάνει μια πρόβλεψη ο πιο πιθανός επόμενος ΑΡ, σύμφωνα πάντα με την τελευταία τιμή του n , επιλέγεται.

Καμιά πρόβλεψη δεν μπορεί να πραγματοποιηθεί, δηλαδή το μοντέλο οδηγείται σε αστοχία, αν δεν έχει παρατηρηθεί προηγουμένως η τάξη μεγέθους n .

Αξίζει ακόμη να αναφέρουμε ότι, οι παράμετροι της διαδικασίας μάθησης του μαρκοβιανού μοντέλου και η διαδικασία της πρόβλεψης επικαλύπτονται. Αυτό εξηγείται απ εξής γεγονός: όταν ένα ΜΗ συνδέεται με ένα ΑΡ, το ΑΡ προβλέπει που θα πάει στη συνέχεια το ΜΗ. Μέχρι να γίνει γνωστό το επόμενο ΑΡ, η πληροφορία προστίθεται στο υπό μάθηση σύνολο των δεδομένων και αυτό επιτρέπει τη βελτίωση του μοντέλου της κινητικότητας. Τέλος ο βαθμός ακρίβειας της πρόβλεψης, είναι συνάρτηση του χρόνου που έχει στη διάθεσή του το σύστημα για να μάθει.

Αφού εξετάσαμε το μοντέλο Markov θα δώσουμε τώρα ορισμένα στοιχεία για το μοντέλο NaiveBayes.

Το μοντέλο NaiveBayes είναι μια βιώσιμη εναλλακτική μέθοδος κατηγοριοποίησης των δεδομένων ενός δικτύου, ερχόμενο σε αντιπαράθεση με τις προσεγγίσεις που στοχεύουν στην ελαχιστοποίηση του λάθους.

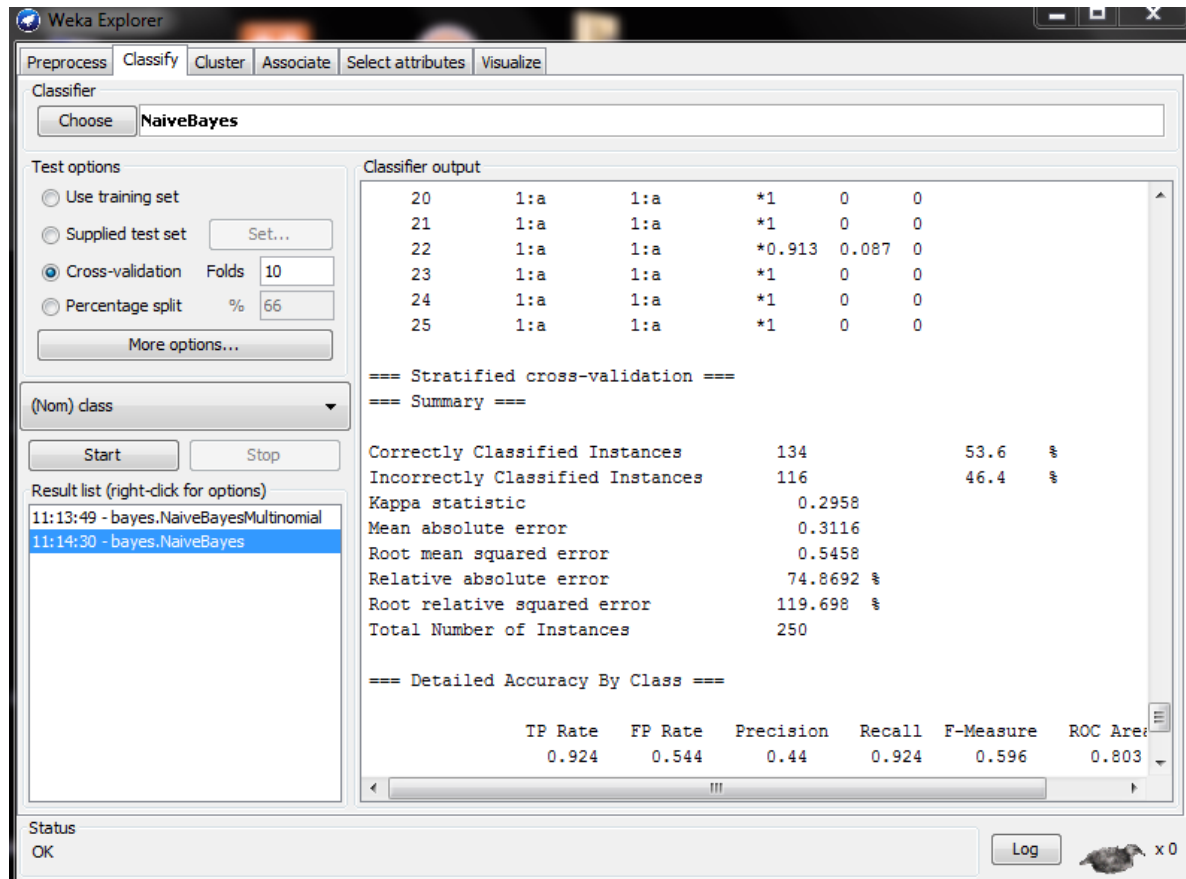
Οι Bayesian μέθοδοι χρησιμοποιούν πιθανότητα για την ποσοτικοποίηση της αβεβαιότητας σε συμπεράσματα και το αποτέλεσμα των Bayesian μεθόδων μάθησης, είναι μια κατανομή πιθανοτήτων, που εκφράζουν τις πεποιθήσεις μας σχετικά με το πόσο πιθανές είναι οι διαφορετικές προβλέψεις. Οι προβλέψεις, γίνονται με την ενσωμάτωση σε όλα τα μοντέλα πάνω από αυτό το οπίσθιο διανομής. Αυτή η μέθοδος εκμάθησης χρησιμοποιεί πιθανότητα να εκπροσωπεί την αβεβαιότητα σχετικά με τη σχέση που έμαθε. Πριν από την εμφάνιση των δεδομένων, προηγούμενες απόψεις, για το τι θα μπορούσε να είναι η πραγματική σχέση, μπορούν να εκφραστούν σε μια κατανομή πιθανότητας πάνω από τα βάρη του δικτύου που ορίζουν αυτή τη σχέση. Μετά παρουσιάζονται τα δεδομένα εκπαίδευσης και οι αναθεωρημένες απόψεις συλλαμβάνονται από μια οπίσθια διανομή πάνω από τα βάρη του δικτύου. Τα βάρη του δικτύου, που φάνηκαν πειστικά πριν, αλλά τα οποία δεν ταιριάζουν με τα δεδομένα πολύ καλά, θα πρέπει τώρα να θεωρηθούν ως πολύ λιγότερο πιθανά, ενώ η πιθανότητα για τις τιμές των βαρών που ταιριάζουν με τα δεδομένα να αυξηθεί.

Ο απλός ταξινομητής Bayes (**simple/naive Bayes classifier**) είναι μια πρακτική μέθοδος μάθησης που στηρίζεται σε στατιστικά στοιχεία (κατανομές πιθανότητας).

Η ποσότητα P που περιγράφει έναν απλό ταξινομητή Bayes, για ένα σύνολο παραδειγμάτων, εκφράζει την πιθανότητα να είναι c η τιμή της εξαρτημένης μεταβλητής C με βάση τις τιμές $\mathbf{x}=(x_1,x_2,\dots,x_n)$ και δίνεται από τη σχέση:

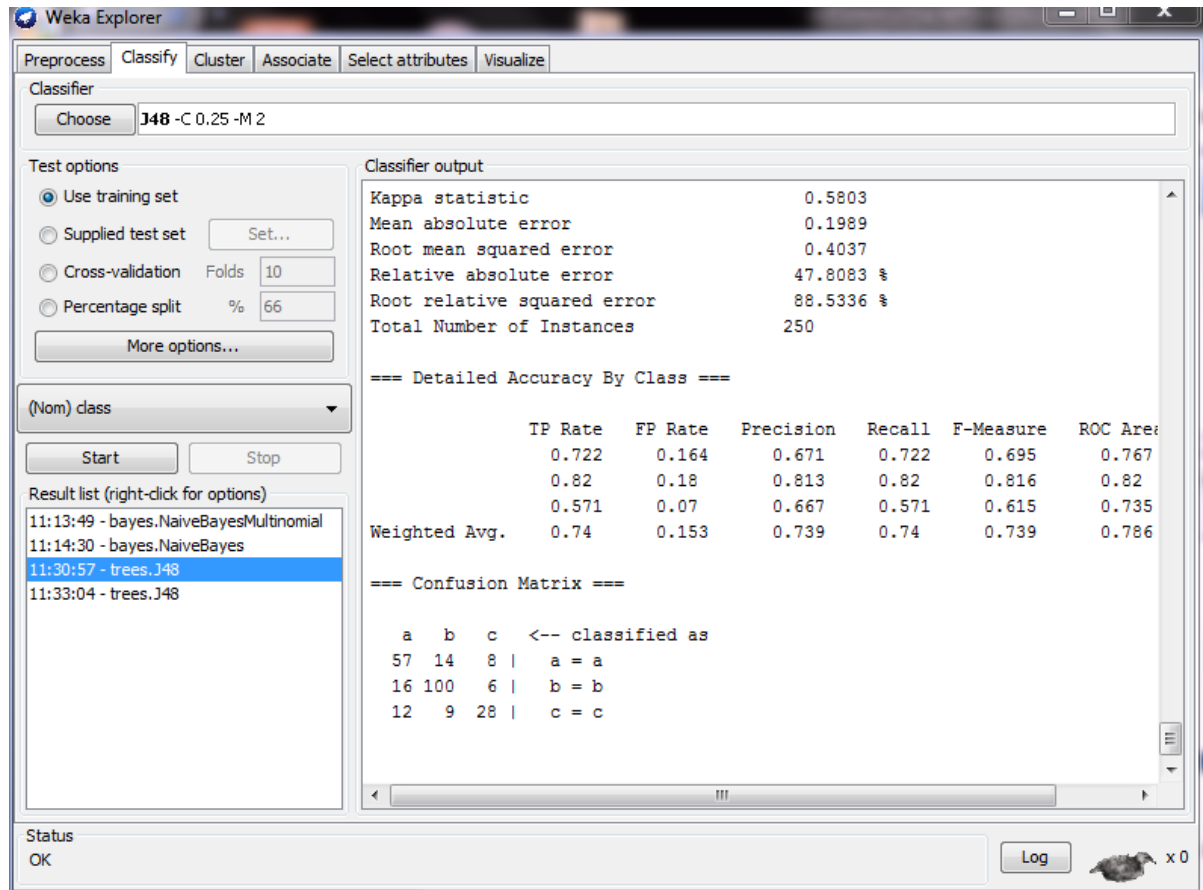
$$P(\mathbf{c} | \mathbf{x}) = P(\mathbf{c}) * \prod P(x_i | \mathbf{c})$$
 όπου τα χαρακτηριστικά X_i θεωρούνται ανεξάρτητα μεταξύ τους.

Στη συνέχεια ακολουθούν κάποιες εικόνες όπου φαίνεται καλύτερα η εφαρμογή των μεθόδων αυτών στο αρχείο **datafml.arff** απ το οποίο αντήσαμε τα δεδομένα του προβλήματός μας.



Εικόνα 6.1

Η εικόνα 6.1 περιγράφει την εφαρμογή του **classifier** NaiveBayes στα δεδομένα μας. Σε σύνολο 250 εγγραφών έχουν κατηγοριοποιηθεί σωστά οι 134. Έγινε σύγκριση και με τους άλλους classifiers της κατηγορίας NaiveBayes και προκύπτει ότι είναι ο καλύτερος για αυτό το είδος των δεδομένων. Επιπλέον και τα βάρη που προσθέτει ο classifier στα δεδομένα έχουν μεγαλύτερη βαρύτητα και μπορεί να οδηγήσουν σε μεγαλύτερη αξιοπιστία μελλοντικά.



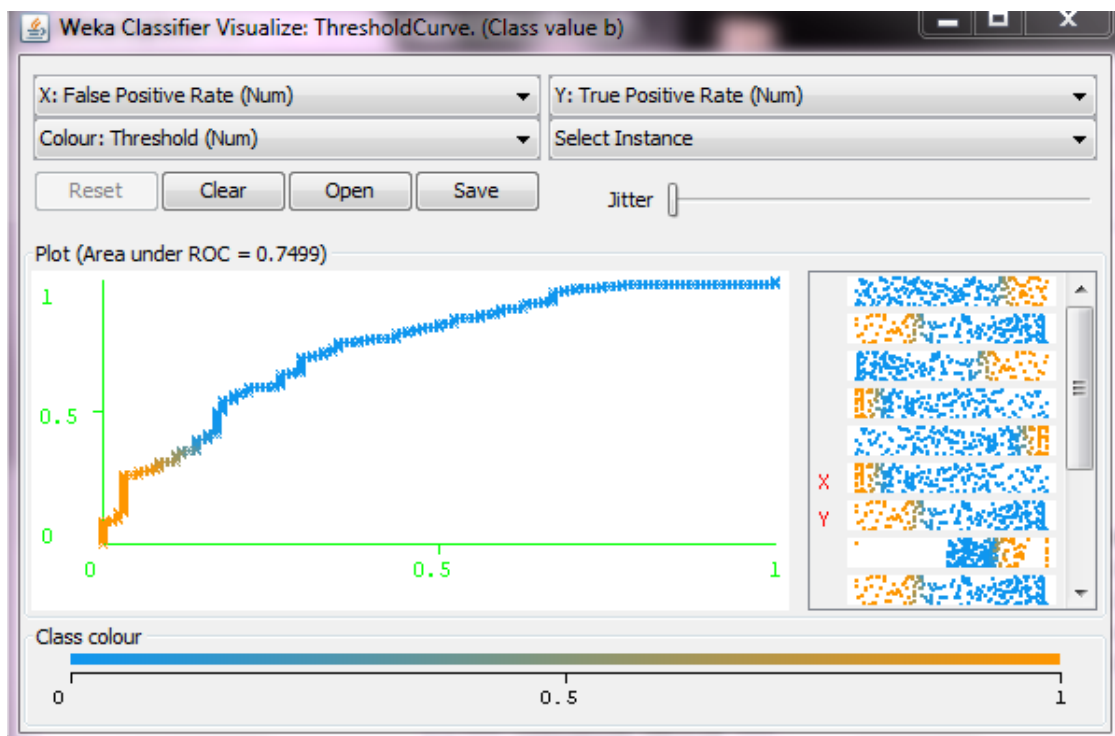
Εικόνα 6.2

Αυτό που αξίζει να τονίσουμε από την εικόνα 6.2 είναι οι επιλογές που μας δίνει η λειτουργία classify του WEKA.

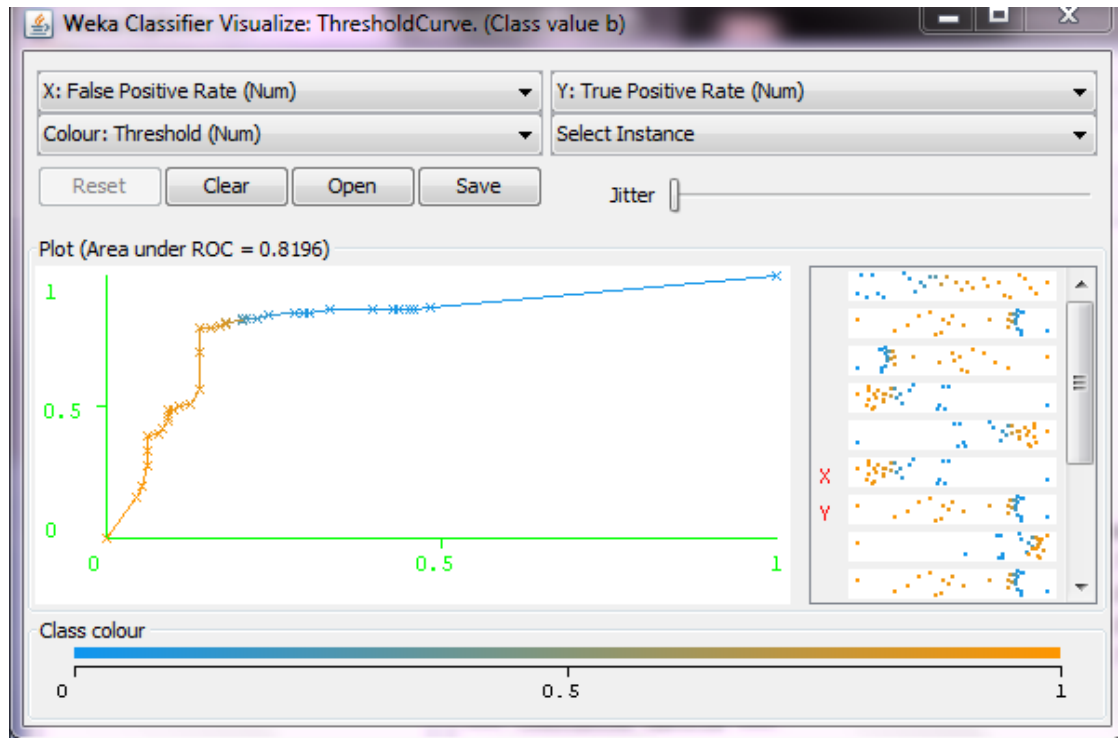
- **Use training set:** Ο classifier αποτιμάται στο πόσο καλά μπορεί να προβλέψει την κλάση των instances που εκπαιδεύτηκε.
- **Supplied test set:** αποτιμάται στο πόσο καλά προβλέπει την κλάση από το σετ των instances που φορτώθηκαν απ το αρχείο.
- **Cross validation:** αποτιμάται μέσα από τον υπολογισμό του μέσου όρου ακρίβειας για το σύνολο των επαναλήψεων που του δίνεται και εισάγεται στο κατάλληλο πεδίο
- **Percentage Split:** αποτιμάται στο πόσο καλά προβλέπει ένα certain percentage των δεδομένων που προσφέρονται για testing. Τα δεδομένα αυτά εξαρτώνται από την τιμή που εισάγεται στο πεδίο(ελέγχεται ένα μέρος των δεδομένων).

Βέβαια εδώ θα πρέπει να τονίσουμε ότι ο κάθε classifier έχει τις δικές του απαιτήσεις από τα δεδομένα, για να μπορεί να εφαρμοστεί όσο το δυνατόν με μεγαλύτερη αξιοπιστία και να δώσει ακριβή αποτελέσματα. Επομένως βλέπουμε ότι ο NaiveBayes αλγόριθμος ταξινόμησης απαιτεί και δεδομένα εισόδου τύπου binary. Δοκιμαστικά εφαρμόσαμε τον αλγόριθμο ταξινόμησης κατηγορίας δέντρου J-48 και είδαμε ότι πήραμε καλύτερα αποτελέσματα.

Η απόδοση κάθε ταξινομητή αναπαρίσταται ως ένα σημείο στην καμπύλη **ROC (Receiver Operating Characteristic Curve)**. Χαρακτηρίζει το trade-off μεταξύ positive hits(True Positive Rate) στον άξονα των ψ και false alarms (False Positive Rate) στον άξονα των χ. Η εικόνα 6.3 που ακολουθεί περιγράφει την απόδοση του ταξινομητή NaiveBayes και η εικόνα 6.4 την απόδοση του ταξινομητή της δομής δέντρου του J48.



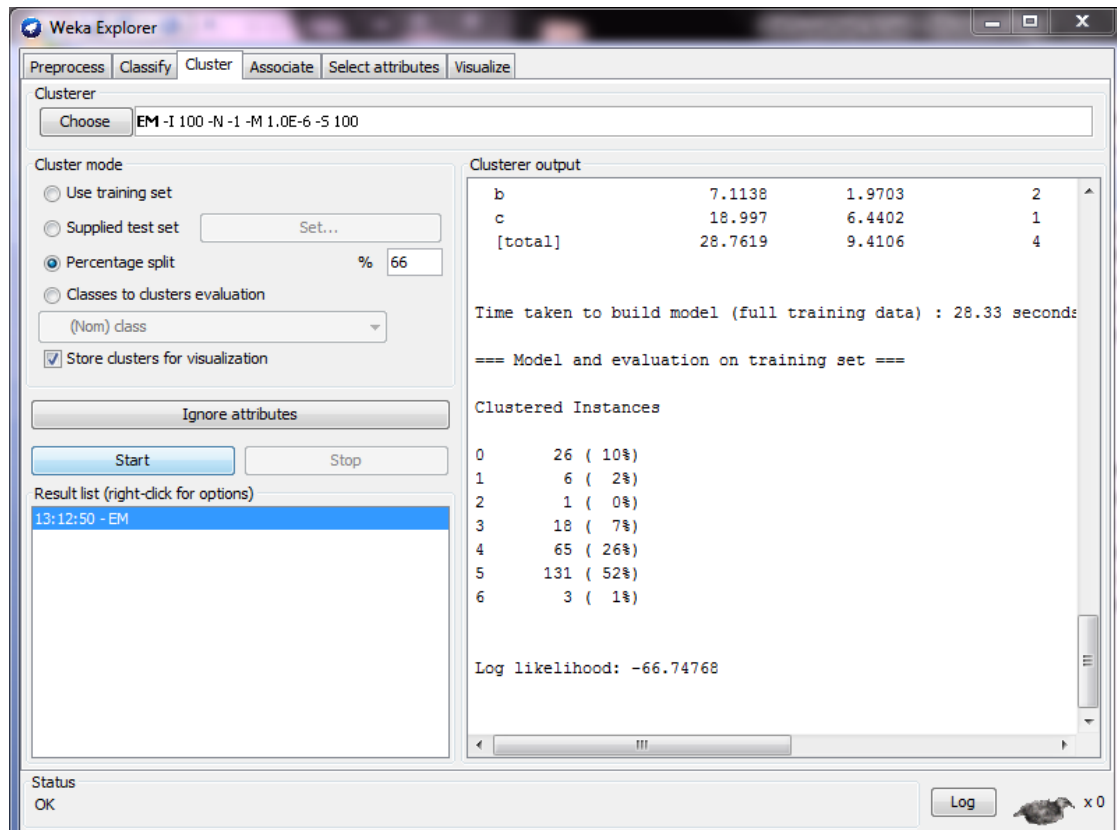
Εικόνα 6.3



Εικόνα 6.4

Συγκρίνοντας τα δεδομένα που προκύπτουν από τις εικόνες που περιγράφουν την απόδοση των δυο ταξινομητών προκύπτει ότι παίρνουμε καλύτερα αποτελέσματα από τον ταξινομητή J48. Επομένως τα δεδομένα που χρησιμοποιήθηκαν στην εργασία δεν είχαν το καλύτερο αποτέλεσμα, τουλάχιστον όχι στη μορφή που δόθηκαν.

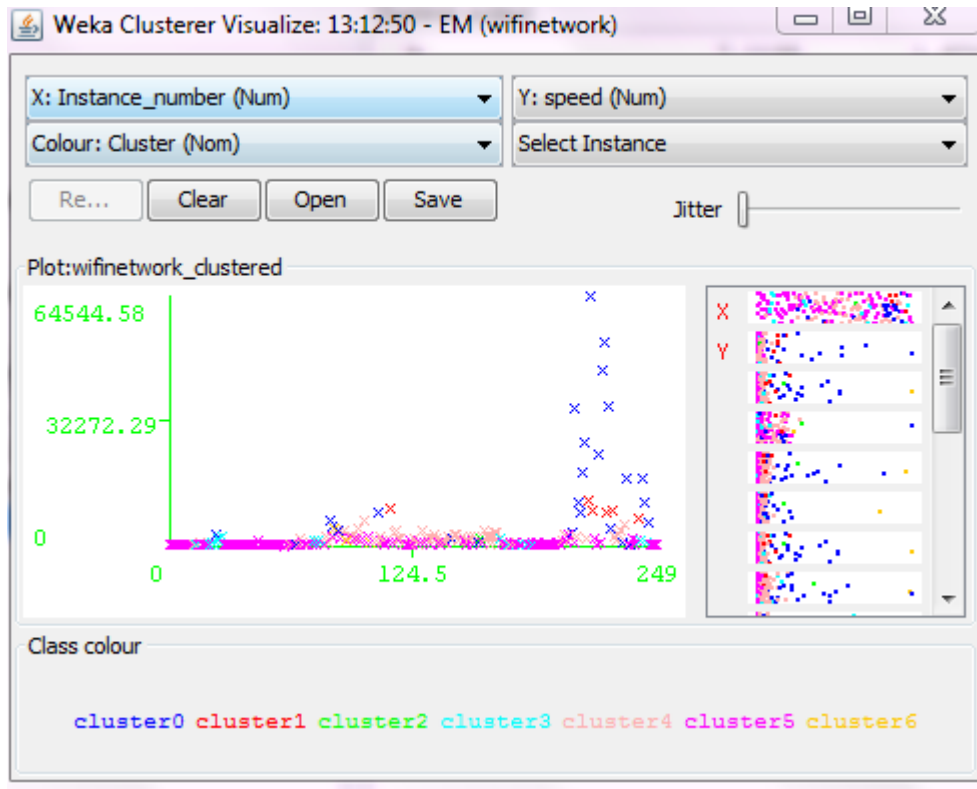
Ακολουθεί στην επόμενη εικόνα η εφαρμογή του clusterer, όπου φαίνονται διάφορα αποτελέσματα



Εικόνα 6.5

Από το διάγραμμα της παραπάνω εικόνας αντλούμε πολλές πληροφορίες για τη μέση τιμή κάθε χαρακτηριστικού των δεδομένων μας, την τυπική τους απόκλιση και διάφορα άλλα. Επίσης από τις πληροφορίες που παρέχει το WEKA διακρίνουμε ότι από τα δεδομένα μας δημιουργήθηκαν επτά clusters ή οι λεγόμενες ομάδες δεδομένων. Το **clustering** είναι μια διαδικασία στην οποία όμοια δεδομένα συγκεντρώνονται και ομαδοποιούνται. Ο σκοπός του είναι να αποκαλύψει δομές δεδομένων που σε άλλη περίπτωση κρύβονται στα δεδομένα. Είναι μια μη επιτηρούμενη διαδικασία μιας και οι αλγόριθμοι που εκτελούνται αναλαμβάνουν να συγκεντρώσουν τα δεδομένα, με τις προδιαγραφές που ορίζει βέβαια ο χρήστης. Το πιο σημαντικό βέβαια που παρέχει το clustering στο σύστημα είναι ότι, επιτρέπει την επαναχρησιμοποίηση των πόρων . Όλοι οι cluster μπορούν να τρέχουν όλοι μαζί, να μοιράζονται τους πόρους και να μην χρειάζεται για καθένα ξεχωριστά, να σπαταλά το σύστημα χωρητικότητα. Επομένως να ένας ακόμη πολύ ωραίος τρόπος, με τον οποίο βελτιώνεται η διαδικασία της βελτιστοποίησης της διαχείρισης δικτύων, με χρήση τεχνικών μηχανικής μάθησης.

Η εικόνα 6.6 που ακολουθεί περιγράφει την απόδοση του clusterer.



Εικόνα 6.6

Εύκολα διακρίνουμε εδώ το που κυμαίνονται οι εγγραφές-στιγμιότυπα του συνόλου των δεδομένων μας σε σχέση με την μέση τιμή και την τυπική απόκλιση. Μπορούμε ακόμη να καταλάβουμε και το βαθμό συσχέτισης που υπάρχει ανάμεσα στα δεδομένα μας ξεχωριστά, όσο και μεταξύ των ομάδων που δημιουργήθηκαν.

Όσον αφορά το πρόβλημα του mobility prediction στο αεροδρόμιο παρατηρήθηκε, τόσο απ την εφαρμογή του αλγορίθμου NaiveBayes, όσο και απ την εφαρμογή του J48, ότι πολλοί χρήστες κινούνται στην περιοχή B. Αυτή η περιοχή είναι η αίθουσα αναμονής και λογικό είναι οι χρήστες να αναπτύσσουν έντονη κινητικότητα στο δίκτυο, ενώ περιμένουν να επιβιβαστούν.

Είναι φανερό ότι το WEKA, όπως και άλλα περιβάλλοντα που υποστηρίζουν τεχνικές μηχανικής μάθησης διαθέτουν μια πληθώρα επιλογών, τόσο στους classifiers, όσο και στους clusterers. Είναι αναπόφευκτο, ότι όσο η τεχνολογία θ αναπτύσσεται, θ αυξάνονται και οι απαιτήσεις των χρηστών. Αυτό σημαίνει τόσο περισσότερα προβλήματα

μπορούν να παρουσιαστούν σ ένα δίκτυο. Για τη σωστή αντιμετώπιση όμως του προβλήματος, πρέπει να γίνει σωστά η συλλογή των δεδομένων που περιγράφουν το πρόβλημα. Μέσω της εργασίας αυτής είδαμε ότι, οι τεχνικές μηχανικής μάθησης συμβάλλουν με πολλούς τρόπους στην βελτίωση της διαχείρισης των δικτύων. Για παράδειγμα είδαμε ότι με τη χρήση αλγορίθμων και μεθόδων μηχανικής μάθησης, τα συστήματα από μόνα τους κάνουν εξοικονόμηση ενέργειας, μπορούν να προλάβουν αστοχίες και αυτό που δεν χωράει καμία αμφισβήτηση είναι ότι οι χρήστες απολαμβάνουν καλύτερης ποιότητας υπηρεσίες. Απλά αξίζει να σημειωθεί ότι κάθε τεχνική μηχανικής μάθησης εξαρτάται σε μεγάλο βαθμό και από τα δεδομένα του προβλήματος καθώς και τον τύπο του προβλήματος προς επίλυση. Πλέον με την βοήθεια κατάλληλων και εξειδικευμένων ανθρώπων το δίκτυο είναι σε θέση να γνωρίζει ανά πάσα στιγμή την κατάστασή του.

7. Συμπεράσματα

Στο κεφάλαιο αυτό της εργασίας θα αναφερθούμε σε κάποια συμπεράσματα που προέκυψαν, από τη χρήση των 4 περιβαλλόντων εργασίας, που παρέχει η πλατφόρμα WEKA στους χρήστες της. Ακόμη θα αναφερθούμε στα συνηθέστερα προβλήματα που έχουν να αντιμετωπίσουν όσοι ασχολούνται με το κομμάτι της διαχείρισης των δικτύων και την αντιμετώπισή τους.

1. Από τα πειράματα που έγιναν στο προηγούμενο κεφάλαιο, προκύπτει η διαπίστωση, ότι τα μοντέλα για το mobility - prediction χωρίζονται σε δυο μεγάλες κατηγορίες : **MH-Centric** και **AP Centric**. Στην πρώτη περίπτωση το μοντέλο πρόβλεψης βασίζεται στα MH του δικτύου και το μοντέλο χτίζεται με βάση τις συγκεκριμένες κινήσεις ενός MH. Αντίθετα στην δεύτερη περίπτωση το μοντέλο πρόβλεψης βασίζεται στα AP του δικτύου και ο κάθε AP καταγράφει τα MH του δικτύου που συνδέονται σε αυτόν. Όπως είδαμε και στην εισαγωγή ο τρόπος για να προσδιορίσουμε την κίνηση ενός MH σε ένα δίκτυο είναι δύσκολος γιατί η πληροφορία είναι διαφορετικής μορφής. Έτσι κάνουμε την υπόθεση ότι η πληροφορία που χρησιμοποιούμε για την κατασκευή του μοντέλου είναι, οι πληροφορίες που παρέχονται απ τον τελευταίο AP που συνδέθηκε το MH. Επιπλέον, για την κατασκευή του μοντέλου που περιέχει τις κινήσεις των MH του δικτύου είναι απαραίτητος ο **prediction agent ή next AP-predictor**.

2. Η διαφορά στην ακρίβεια μεταξύ των δυο κατηγοριών είναι απειροελάχιστη. Ακόμη η κατασκευή ενός μοντέλου που λαμβάνει σοβαρά υπόψη του το χρόνο, για παράδειγμα πόσοι MH συνδέονται σε ένα AP τις απογευματινές ώρες, δεν παρουσιάζει κάποια βελτίωση. Σε κάθε περίπτωση όμως το ποσοστό ακρίβειας των μοντέλων είναι χαμηλότερο από 60%, αλλά αυτό δικαιολογείται γιατί είναι ένα απ τα χαρακτηριστικά του δικτύου WiFi. Επίσης χρησιμοποιώντας μοντέλο κατηγορίας APC μπορούμε να επιτύχουμε καλύτερα αποτελέσματα για MH που συνδέονται για πρώτη φορά στο δίκτυο.

3. Οι περισσότερες μέθοδοι που χρησιμοποιήθηκαν στο προηγούμενο κεφάλαιο για **classify** και **clustering** επηρεάζονται άμεσα από μια

αλλαγή στο μονοπάτι κίνησης ενός ΜΗ μέσα στο δίκτυο. Γι αυτό το λόγο, η ακρίβεια της πρόβλεψης του μοντέλου, μειώνεται σε περίπτωση που δεν έχει γίνει σωστή συλλογή δεδομένων και ο θόρυβος είναι υψηλός. Αυτές οι μέθοδοι δεν είναι σε θέση να καταλάβουν την τυχαία κίνηση και τη διαφορά της από την κανονική κίνηση ενός χρήστη στο μονοπάτι που θα προβλέψει το μοντέλο. Σε γενικές γραμμές όμως οι χρήστες εμφανίζουν συνήθως κάποια σταθερή ροή κίνησης σ ένα δίκτυο και η κινητικότητα που παρουσιάζουν, λόγω ενός απρόοπτου συμβάντος, δεν είναι συχνό φαινόμενο.

4. Το περιβάλλον εργασίας WEKA Explorer είναι το κύριο περιβάλλον εργασίας του WEKA. Παρέχει στους χρήστες του όλη τη βασική λειτουργικότητα που απαιτείται για να κατασκευάσουν ένα μοντέλο μηχανικής μάθησης. Οι χρήστες μπορούν να επεξεργαστούν τα δεδομένα τους με τρόπο απόλυτα διαφανή, γνωρίζοντας κάθε φορά τη διαδικασία που ακολουθούν και χωρίς να μπλέκεται η μια λειτουργία με την άλλη. Για παράδειγμα, για το ίδιο πρόβλημα εφαρμόζουν τη λειτουργία **classify** και **clustering** και βλέπουν τα αποτελέσματά τους. Το WEKA Explorer είναι τόσο αποδεκτό μεταξύ των χρηστών, γιατί δεν απαιτεί εξειδικευμένες γνώσεις προγραμματισμού. Μέχρι και αρχάριοι χρήστες του WEKA μπορούν να το χρησιμοποιήσουν, για την κατασκευή μοντέλων μάθησης για τα δεδομένα τους.

5. Το περιβάλλον εργασίας WEKA Experimenter είναι χρήσιμο για χρήστες που έχουν κάποια εμπειρία από κατασκευή μοντέλων μηχανικής μάθησης. Χρησιμοποιείται σε πολύ μεγάλο βαθμό για την εφαρμογή διάφορων **classifiers** και **clusterers**, γιατί δίνει τη δυνατότητα στους χρήστες να συγκρίνουν τόσο τα αποτελέσματα, όσο και τις μεθόδους **clustering** και **classifying** που εφαρμόζουν. Μπορούν να τις συγκρίνουν για παράδειγμα, ως προς το ποσοστό επιτυχούς πρόβλεψης, την ταχύτητα εκτέλεσης, την μορφή εμφάνισης των αποτελεσμάτων και άλλα διάφορα χαρακτηριστικά που προσφέρει το WEKA Experimenter. Επιπλέον μπορούν να επιλέξουν να χρησιμοποιήσουν ήδη υπάρχοντα κατασκευαστικά μοντέλα ή να δημιουργήσουν από την αρχή δικά τους, απλά ή σύνθετα, ανάλογα με τις ανάγκες τους, επιλέγοντας **simple** ή **advanced**. Τέλος μπορούμε να πούμε ότι το WEKA Experimenter είναι το καταλληλότερο περιβάλλον εργασίας για χρονοβόρα πειράματα που εκτελούνται σε δεδομένα.

6. Το περιβάλλον εργασίας WEKA Knowledge Flow χρησιμοποιείται από τους χρήστες, για την σχηματική αναπαράσταση της κατασκευής της ροής της γνώσης. Βοηθά τους χρήστες να καταλάβουν τη διαδικασία που ακολουθείται, από την ανάγνωση ενός προβλήματος σε ένα δίκτυο, τα δεδομένα-τιμές που έχουν για αυτό, την εφαρμογή διάφορων μεθόδων μηχανικής μάθησης και τέλος την προβολή και αξιολόγηση των αποτελεσμάτων. Με λίγα λόγια, από την είσοδο απλών δεδομένων στο σύστημα, στην δημιουργία λύσεων που μπορούν να εφαρμοστούν για να βελτιώσουν την ομαλή λειτουργία του δικτύου.

7. Το περιβάλλον εργασίας WEKA Command Line χρησιμοποιείται από χρήστες, που έχουν πολύ καλές γνώσεις προγραμματισμού σχετικά με τη χρήση των εντολών σε διάφορα λειτουργικά συστήματα. Τα πλεονεκτήματα της χρήσης του είναι ότι οι απαιτήσεις του σε μνήμη είναι ελάχιστες σε σχέση με τα άλλα 3 περιβάλλοντα εργασίας του WEKA. Επίσης δίνει στους χρήστες την αίσθηση ότι ελέγχουν πλήρως την διαδικασία κατασκευής μοντέλων μηχανικής μάθησης. Όμως παρουσιάζει σημαντικό πρόβλημα στον τομέα της διαχείρισης προβλημάτων, που απαιτούν τη χρήση πολλών δεδομένων και την εφαρμογή πολλών και διαφορετικών μεθόδων **classifying** και **clustering**. Αυτό συμβαίνει γιατί, όταν οι χρήστες έχουν να διαχειριστούν τόσες πολλές γραμμές κώδικα εκτέλεσης εντολών, η πιθανότητα σφάλματος μεγαλώνει. Ακόμη δεν είναι εύκολη και η σύγκριση των αποτελεσμάτων από την εφαρμογή διαφορετικών μεθόδων στα δεδομένα.

8. Η τεχνική που εφαρμόστηκε περισσότερο από τις άλλες στην εργασία μας, κατά την εκτέλεση διάφορων μεθόδων **classifying** είναι το **cross-validation**. Αυτό συμβαίνει, γιατί χρησιμοποιείται ευρέως σε λειτουργίες, που ο στόχος τους είναι η πρόβλεψη και η εκτίμηση του ποσοστού ακρίβειας επιτυχούς πρόβλεψης από την εφαρμογή του σε ένα σύνολο δεδομένων. Ακόμη ο μηχανισμός αυτός χωρίζει τα δεδομένα σε ένα κύριο μέρος(**training set**) και εφαρμόζεται σε δεδομένα που το σύστημα τα θεωρεί καινούρια, προβλέποντας τη συμπεριφορά τους. Τέλος με τη χρήση του μηχανισμού αυτού, πετυχαίνουμε το συνδυασμό μετρήσεων του ποσοστού πρόβλεψης λάθους και οδηγούμαστε στην εξαγωγή μοντέλων με δεδομένα, που έχουν μεγαλύτερη ακρίβεια.

9. Ένα από τα συνηθέστερα προβλήματα που αντιμετωπίζουν τόσο οι χρήστες ενός δικτύου, όσο και αυτοί που ασχολούνται με την επίλυσή του, είναι το λεγόμενο **access-points overloaded**. Αυτό πρακτικά σημαίνει ότι κάποιοι χρήστες δεν μπορούν να συνδεθούν καθόλου σ ένα δίκτυο. Επειδή οι απαιτήσεις των χρηστών αυξάνονται διαρκώς , οι υπεύθυνοι για τη συντήρηση των δικτύων πρέπει να προχωρήσουν σε μετρήσεις που θα φαίνονται ξεκάθαρα οι ώρες αιχμής σ ένα δίκτυο. Ώρες αιχμής θα μπορούσαμε να χαρακτηρίσουμε την ύπαρξη πολλών χρηστών και την κίνηση μεγάλων ποσοτήτων δεδομένων. Επομένως θα μπορούσε να κατασκευαστεί ένα μοντέλο που θα ενημερώνει το δίκτυο για την πιθανότητα να δημιουργηθεί το συγκεκριμένο πρόβλημα. Άρα μεγαλύτερο ρόλο παίζει η πρόβλεψη και στη συνέχεια η αντιμετώπιση του προβλήματος από τους ειδικούς.

10. Άλλο σημαντικό πρόβλημα που μπορεί να προκύψει σ ένα δίκτυο είναι το φαινόμενο του **handover**. Αυτό συμβαίνει συχνότερα σε χρήστες που κινούνται μέσα σε ένα δίκτυο. Χαρακτηριστικά παραδείγματα είναι ένα πολυώροφο κτίριο, όπως μια υπηρεσία ή ένα πανεπιστήμιο. Επίσης σ ένα αεροδρόμιο μπορεί να παρουσιαστεί το συγκεκριμένο πρόβλημα γιατί οι χρήστες κινούνται διαρκώς στους χώρους του αεροδρομίου. Κι εδώ πρέπει να γίνουν μετρήσεις των δεδομένων που στέλνουν οι χρήστες, ώστε οι υπεύθυνοι να μελετήσουν τα δεδομένα και να γνωρίζουν τις θέσεις των χρηστών που συνήθως εμφανίζεται το πρόβλημα. Έτσι θα μπορούν οι χρήστες να εξυπηρετηθούν από ένα κοντινότερο σε αυτούς σταθμό βάσης.

11. Είναι ξεκάθαρο ότι η τεχνολογία προχωράει διαρκώς και οι απαιτήσεις των χρηστών αυξάνονται με γοργούς ρυθμούς. Λογικό είναι στον τομέα της διαχείρισης των δικτύων να εμφανίζονται διάφορα προβλήματα που κάνουν τη ζωή δύσκολη στους χρήστες. Γι αυτό είναι πολύ σημαντική η μελέτη των δεδομένων που προσφέρει το δίκτυο στους διαχειριστές του. Η μηχανική μάθηση παίζει πολύ μεγάλο ρόλο στη βελτιστοποίηση της διαχείρισης των δικτύων. Μέσω των μετρήσεων μπορούν να κατασκευαστούν μοντέλα μηχανικής μάθησης που θα βοηθούν εγκαίρως το δίκτυο να διαγνώσει το πρόβλημα. Γι αυτό λοιπόν, πριν την επίλυση ενός προβλήματος, είναι αναγκαίο να προβλεφθεί αυτό το πρόβλημα. Αυτό έχει στόχο να κάνει η μηχανική μάθηση . Δηλαδή με την είσοδο κάποιων δεδομένων σ ένα μοντέλο να μας βγάζει την πιθανότητα να οδηγηθούμε σε αστοχία, βασιζόμενο σε προηγούμενα δεδομένα χρήσης και ομαλής λειτουργίας του δικτύου.

8.Βιβλιογραφία-πηγές

1. A. Aljadhari, T. F. Znati, "Predictive Mobility Support for QoS Provisioning in MobileWireless Environments"
2. A. Jayasuriya, J. Asenstorfer, "Mobility Prediction for Cellular Networks Based on the Observed Traffic Patterns"
3. W.-S. Soh, H.S. Kim, "QoS provisioning in cellular networks based on mobility prediction techniques",
4. Kari Laasonen. Clustering and prediction of mobile user routes from cellular data.
5. B. H. Juang and L. H. Rabiner. The segmental k-means algorithm for estimating the parameters of hidden markov models. IEEE Transactions on Acoustics, Speech and Signal Processing,
6. David Lindsay and Sian Cox. Effective probability forecasting for time series data using standard machine learning techniques.
7. <http://natureofcode.com/book/chapter-10-neural-networks/>
8. <http://machinelearningmastery.com/>
9. <http://ireasoning.com/browser/help.shtml>
10. <http://ireasoning.com/snmpapi.shtml>
11. <http://www.ibm.com/developerworks/library/os-weka1/>

12. [https://en.wikipedia.org/wiki/Weka_\(machine_learning\)](https://en.wikipedia.org/wiki/Weka_(machine_learning))