ΤΕΧΝΟΛΟΓΙΚΟ ΕΚΠΑΙΔΕΥΤΙΚΟ ΙΔΡΥΜΑ ΣΕΡΡΩΝ
ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΚΩΝ ΕΦΑΡΜΟΓΩΝ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ & ΕΠΙΚΟΙΝΩΝΙΩΝ

ΘΕΜΑ

«Επιλογή χαρακτηριστικών (feature selection) από βάσεις δεδομένων με φίλτρο αμοιβαίας πληροφορίας (mutual information filter)»

Σπουδαστής

Kakoyan Manvel

Επιβλέπων

Δρ. Τσιμπίρης Αλκιβιάδης

ΣΕΡΡΕΣ 2010

Περιεχόμενα

Εισαγωγή **7**

1. Επιλογή χαρακτηριστικών στο Data Mining **11**

1.1. Σκοπός και η λειτουργία της εργασίας 11

1.2. Μέθοδοι επιλογής χαρακτηριστικών 12

1.2.1. Μέθοδος φίλτρου (Filter) 12

1.2.1.1. Ανάλυση χαρακτηριστικών με επόπτη 13

1.2.1.1.1. Έλεγχος της ευστάθειας (Stability) 13

1.2.1.1.2. Έλεγχος της ικανότητας διαχωρισμού (Seperability) 13

1.2.1.1.3. Ανάλυση της ομοιότητας (Correlation) 14

1.2.1.2. Αμοιβαία πληροφορία (Mutual information) 14

1.2.2. Μέθοδος περιτυλίγματος (Wrapper) 15

2. Μαθηματικό Μέρος **17**

2.1. Θεωρία πιθανοτήτων 17

2.2. Θεωρία πληροφορίας 18

2.2.1. Πληροφορία 18

2.2.2. Μετρό πληροφορίας του Shannon 19

2.2.3. Εντροπία - Μέση πληροφορία. (Entropy) 19

2.2.5. Από Κοινού Εντροπία (Joint Entropy) 20

2.2.4. Υπό Συνθήκη Εντροπία (Conditional entropy) 20

2.2.5. Αμοιβαία Πληροφορία (Mutual Information) 21

2.3. Υπό Συνθήκη Αμοιβαία Πληροφορία (Conditional Mutual Information-CMI) 22

2.4. Επιλογή χαρακτηριστικών βασισμένη στην Αμοιβαία πληροφορία Μέγιστης Σχετικότητας και Ελάχιστου Πλεονασμού (Feature Selection Based on Mutual Information: Criteria of Max-Dependency, Max-Relevance,and Min-Redundancy) 24

2.4.1. Η Σχέση Ανάμεσα Στην Μέγιστη Εξάρτιση , Μέγιστη Σχετικότητα, και Ελάχιστος Πλεονασμός 26

2.4.1.1. Μέγιστη Εξάρτιση (Max Dependency) 26

2.4.1.2 Μέγιστη Σχετικότητα Ελάχιστος Πλεονασμός (Max-Relevance and Min-Redundancy) 26

3. Εφαρμογή **31**

3.1. Σχεδιάγραμμα 33

3.2. Παράδειγμα 34

3.2.1. Πρόβλημα 34
3.2.2. Δεδομένα 34
3.2.3. Λύση 35

3.3. PHP 38

3.4. Μεταφορά αρχείου από τον χρήστη στον Server 38

3.5. Εκτέλεση αρχείων Matlab μέσω της exec 39

3.6. Δημιουργία μοναδικού πίνακα στην MySql και προβολή του 40

3.7. Εμφάνιση παλιότερων αποτελεσμάτων 42

3.8. Υλοποίηση στο Matlab του αλγορίθμου επιλογής χαρακτηριστικών με το φίλτρο της αμοιβαίας πληροφορίας 43

3.9. Υλοποίηση της υπό συνθήκης αμοιβαίας πληροφορίας στο Matlab (Conditional Mutual Information-CMI) 45

3.10. Αλγόριθμος υπό συνθήκη αμοιβαία πληροφορία (Conditional Mutual Information) στο Matlab 48

3.10.1. Εύρεση πρώτου Βέλτιστου χαρακτηριστικού I(Fi,C) 48
3.10.2. Εύρεση των χαρακτηριστικών με τα μεγαλύτερα σκορ 50
3.10.3. Εύρεση επόμενων βέλτιστων χαρακτηριστικών 51
3.10.4. Αποθήκευση των αποτελεσμάτων στην βάση μας 52

3.11. Αλγόριθμος Μέγιστη Σχετικότητα Ελάχιστος Πλεονασμός (Max Relevance Min Redundancy) 54

3.11.1. Εύρεση επόμενων βέλτιστων χαρακτηριστικών 54

3.12. Αλγόριθμος κανονικοποίησης των τιμών σε έναν πίνακα 57

3.13. Αλγόριθμος τοποθέτησης των κανονικοποιημένων τιμών σε χωρία (bins) 59

3.14. Αποτελέσματα 60

3.14.1. Σύγκριση αποτελεσμάτων με την ιστοσελίδα του Hanchuan Peng 61
3.14.2. Σύγκριση αποτελεσμάτων με το Spider 64

3.15. Συμπεράσματα 67

4. Πίνακες 68

4.1. Πίνακας εξισώσεων 68

4.2. Πίνακας εικόνων 69

4.3. Πίνακας πινάκων 69

4.4. Πίνακας γραφημάτων 70

5. Παράρτημα Α 71

5.1. Matlab 71

5.2. Βάσεις Δεδομένων 71

5.3. Ο οδηγός ODBC 72

5.4. Δημιουργία Και Ρύθμιση ODBC driver 73

5.5.Flash 74

5.6. Αρχεία CSS 75

Βιβλιογραφία 76

Κώδικας 76

Εισαγωγή

Ένα πρόβλημα που εμφανίζεται σε αρκετές εφαρμογές μηχανικής μάθησης είναι ο μεγάλος αριθμός χαρακτηριστικών (features) και η μεγάλη διαστασιμότητα (dimensionality) αποτελεί πρόβλημα τόσο σημαντικό που πολλές φορές αναφέρεται ως κατάρα της διαστασιμότητας (curse of dimensionality).

Οι περισσότεροι αλγόριθμοι εκμάθησης έχουν σχεδιαστεί ώστε να επιλέγουν τα πλέον κατάλληλα χαρακτηριστικά για τη διαμόρφωση των αποφάσεών τους. Η ύπαρξη περισσότερων χαρακτηριστικών επομένως οδηγεί θεωρητικά σε περισσότερο αποδοτική εκπαίδευση. Ωστόσο : "Ποια η διαφορά μεταξύ θεωρίας και πράξης; Δεν υπάρχει διαφορά-στη θεωρία. Αλλά στην πράξη υπάρχει ". Στην πράξη, η προσθήκη μη σχετικών χαρακτηριστικών συχνά 'συγχύζει' τα συστήματα μηχανικής μάθησης.

Σκοπός της πτυχιακής εργασίας είναι η μελέτη αλγορίθμων επιλογής χαρακτηριστικών (feature selection) από βάσεις δεδομένων με φίλτρο αμοιβαίας πληροφορίας (mutual information filter). Στη συνέχεια η ανάπτυξη μιας εφαρμογής σε Matlab που θα επικοινωνεί με βάσεις δεδομένων μέσω ODBC για λήψη και αποθήκευση των δεδομένων. Στην εφαρμογή αυτή θα δοκιμαστεί ο αλγόριθμος επιλογής χαρακτηριστικών (feature selection) με φίλτρο αμοιβαίας πληροφορίας.

Θα αναπτυχθεί επίσης ένα διαδραστικό περιβάλλον για τους χρήστες, σε διαδικτυακή γλώσσα (PHP, HTML και Java), ώστε να μπορούν να εκτελούν την εφαρμογή που αναπτύχθηκε σε Matlab, μέσω του διαδικτύου χωρίς να έχουν απαραίτητα εγκατεστημένο το Matlab στον Η/Υ τους.

Η δομή της πτυχιακής αυτής είναι ως εξής: Στο κεφάλαιο 1. Επιλογή χαρακτηριστικών στο Data Mining) περιγράφουμε τις πιο γνωστές μεθόδους επιλογής χαρακτηριστικών που υπάρχουν. Στο κεφαλαίο

2. Μαθηματικό Μέρος) επικεντρώνουμε την προσοχή μας στην αμοιβαία πληροφορία και τον μαθηματικό τρόπο υπολογισμού με διάφορες παραλλαγές. Στο κεφαλαίο

3. Εφαρμογή) εξηγούμε τον τρόπο που αναπτύξαμε την εφαρμογή μας, παρουσιάζουμε κάποια αποτελέσματα και συγκρίσεις μεταξύ των διαφορετικών αλγορίθμων και στο τέλος παραθέτουμε κάποια συμπεράσματα .

1. Επιλογή χαρακτηριστικών στο Data Mining

Η επιλογή χαρακτηριστικών χρησιμοποιείτε στους τομείς αναγνώριση προτύπων, στατιστική, και κυρίως στην εξόρυξη δεδομένων. Η κύρια ιδέα της επιλογής χαρακτηριστικών είναι να επιλέγει ένα υποσύνολο μεταβλητών εξαλείφοντας στοιχεία με μικρή ή περιττή πληροφορία. Η επιλογή χαρακτηριστικών μπορεί να βελτιώσει σημαντικά την απόδοση των ταξινομητών (classifiers) και να βοηθήσει στην δημιουργία ενός μοντέλου που επεξεργάζεται καλύτερα περίπλοκα δεδομένα.

Συνήθως η εύρεση του καλύτερου υποσύνολου των βέλτιστων χαρακτηριστικών είναι ένα σημαντικό πρόβλημα. Η επιλογή χαρακτηριστικών στην επιβλεπόμενη μάθηση (supervised learning) έχει μελετηθεί επαρκώς. Κύριος στόχος είναι να βρεθεί ένα υποσύνολο χαρακτηριστικών που προσφέρει μεγαλύτερη ακρίβεια ταξινόμησης.

Ωστόσο, οι παραδοσιακές προσεγγίσεις της επιλογής χαρακτηριστικών με κριτήριο την μαζική αξιολόγηση έχουν δείξει περιορισμένη ικανότητα όσων αφορά στην ανακάλυψης της γνώσης και στην επιλογή της σωστής απόφασης. Γενικά κανένα κριτήριο για την επιλογή χαρακτηριστικών γνωρισμάτων χωρίς επόπτη (unsupervised learning) δεν θεωρείτε καλύτερο για κάποια εφαρμογή και μόνο η απόφαση του κατασκευαστή μπορεί να καθορίσει τη σχετική βαρύτητα των κριτηρίων για την εφαρμογή της [1].

1.1. Σκοπός και η λειτουργία της εργασίας

Ο απώτερος σκοπός και στόχος μας είναι να δημιουργήσουμε ένα σύνολο προγραμμάτων γραμμένο σε διάφορες γλώσσες όπως html, php, Matlab ώστε με τον κατάλληλο συνδυασμό τους να μπορεί ένας απλός χριστής από το σπίτι του (χωρίς να έχει εγκατεστημένο Matlab) να στείλει έναν μεγάλο πίνακα δεδομένων στον server και αυτός αφού τρέξει τους αλγόριθμους επιλογής χαρακτηριστικών με διάφορες παραλλαγές του φίλτρου της αμοιβαίας πληροφορίας να επιστρέφει στον χρήστη τα καλύτερα N

χαρακτηριστικά γνωρίσματα από αυτόν τον πινάκα που ο προηγούμενος έχει ζητήσει. Παρακάτω θα εξηγήσουμε αναλυτικά και βήμα προς βήμα την λειτουργία της εργασίας αυτής.

1.2. Μέθοδοι επιλογής χαρακτηριστικών

Κατά την προσέγγιση επιλογής χαρακτηριστικών, βασικός σκοπός είναι να εντοπιστούν και απορριφθούν εκείνα τα χαρακτηριστικά τα οποία παρέχουν περιττή πληροφορία. Με τον τρόπο αυτό μειώνεται η διάσταση των δεδομένων αφού τα περιττά διανύσματα δεν συμμετέχουν στις περαιτέρω διαδικασίες. Η επιλογή χαρακτηριστικών περιλαμβάνει μεθόδους οι οποίες αναφέρονται είτε στο σύνολο των δεδομένων του κάθε διανύσματος (μη επιβλεπόμενες μέθοδοι) είτε στη γραφική και στατιστική ανάλυση προκειμένου να καθοριστεί ο βαθμός της διαχωριστικότητας μεταξύ των κλάσεων (επιβλεπόμενες μέθοδοι) [2].

Οι πιο γνωστοί μέθοδοι επιλογής υποσυνόλου χαρακτηριστικών είναι οι παρακάτω δυο:

- Μέθοδος φίλτρου (Filter): Η ανεξάρτητη επιλογή, που είναι βασισμένη στα γενικά χαρακτηριστικά των δεδομένων .
- Μέθοδος περιτυλίγματος (Wrapper): Για να βρεθεί το διάστημα όλων των υποσυνόλων χαρακτηριστικών γνωρισμάτων, ένας αλγόριθμος αναζήτησης είναι συνέχεια τυλιγμένος (wrapped) γύρω από το πρότυπο ταξινόμησης

1.2.1. Μέθοδος φίλτρου (Filter)

Οι μέθοδοι φίλτρων αξιολογούν τη σχετικότητα των χαρακτηριστικών ερευνώντας μόνο τις ιδιότητες των στοιχείων. Η πιο κοινή διαδικασία είναι βασισμένη στον υπολογισμό της σχετικότητας χαρακτηριστικών γνωρισμάτων. Τα χαρακτηριστικά γνωρίσματα με τη χαμηλή σχετικότητα αφαιρούνται. Τα υπόλοιπα χαρακτηριστικά γνωρίσματα χρησιμοποιούνται ως εισαγωγή στον αλγόριθμο ταξινόμησης .

Οι τεχνικές φίλτρων έχουν το πλεονέκτημα να είναι απλές και γρήγορες, να είναι εφικτές από άποψη υπολογιστικής πολυπλοκότητας που εφαρμόζεται στα δεδομένα υψηλής διάστασης (όπως δεδομένα DNA, κείμενο κτλ) και να είναι ανεξάρτητες από τον αλγόριθμο ταξινόμησης. Το τελευταίο πλεονέκτημα είναι πολύ σημαντικό δεδομένου ότι κάποιος μπορεί να εφαρμόσει μια μέθοδο φίλτρων παράγοντας το βέλτιστο υποσύνολο

χαρακτηριστικών γνωρισμάτων που μπορεί να αξιολογηθεί χρησιμοποιώντας τους διαφορετικούς ταξινομητές (classifiers) όπως κάνει το spider [3].

Μειονέκτημα των μεθόδων φίλτρων είναι ότι δεν υπάρχει οποιαδήποτε αλληλεπίδραση με τον ταξινομητή. Αυτό σημαίνει ότι κάθε χαρακτηριστικό γνώρισμα ελέγχεται χωριστά, κατά συνέπεια οι εξαρτήσεις χαρακτηριστικών γνωρισμάτων να αγνοούνται. Αυτό το γεγονός μπορεί να οδηγήσει στη χαμηλότερη απόδοση ταξινόμησης σε σύγκριση με άλλες τεχνικές επιλογής χαρακτηριστικών γνωρισμάτων. Σε μια προσπάθεια να ελεγχθούν οι εξαρτήσεις χαρακτηριστικών γνωρισμάτων πολλών μεταβλητών οι τεχνικές φίλτρων έχουν βελτιωθεί στοχεύοντας στην ενσωμάτωση των εξαρτήσεων χαρακτηριστικών γνωρισμάτων μέχρι ενός ορισμένου βαθμού [4].

1.2.1.1. Ανάλυση χαρακτηριστικών με επόπτη

Μια μέθοδος επιλογής χαρακτηριστικών φίλτρου βασίζεται στον παράγοντα διαχωριστικότητας. Η επιλογή των ισχυρών χαρακτηριστικών είναι κρίσιμη για την αποτελεσματικότητα της ταξινόμησης. Για να ελαττωθεί το πλήθος των χαρακτηριστικών με την επιλογή ενός συνόλου βέλτιστων, που οδηγούν στην ορθή εκπαίδευση, εκτελείται μια διαδικασία αξιολόγησης της ευστάθειας (stability), της διαχωριστικότητας (separability) και της ομοιότητας (similarity) των χαρακτηριστικών [5].

1.2.1.1.1. Έλεγχος της ευστάθειας (Stability)

Με τον έλεγχο της ευστάθειας προσδιορίζονται τα χαρακτηριστικά που παρουσιάζουν σημαντικά σταθερή συμπεριφορά. Υπολογίζονται οι μέσες τιμές και οι τυπικές αποκλίσεις των χαρακτηριστικών μέσα σε κάθε κλάση. Οι τιμές των τυπικών αποκλίσεων κανονικοποιούνται κατόπιν στο ίδιο διάστημα, διαιρούμενες με τις μέσες τιμές τους. Τα χαρακτηριστικά διαιρούνται σε τρεις ομάδες ανάλογα με την κανονικοποιημένη τυπική απόκλισή τους σ^*. Η πρώτη ομάδα περιλαμβάνει τα χαρακτηριστικά υψηλής ευστάθειας με $\sigma^*<0.1$. Η δεύτερη ομάδα περιλαμβάνει τα ασταθή χαρακτηριστικά με $0.1<\sigma^*<0.9$ και η τρίτη τα πολύ ασταθή με $\sigma^*>0.9$.

1.2.1.1.2. Έλεγχος της ικανότητας διαχωρισμού (Seperability)

Για τον έλεγχο της ικανότητας διαχωρισμού καθορίζεται ο παράγοντας διαχωριστικότητας (seperability factor) S_v μεταξύ δύο κλάσεων ω_i και ω_j για κάθε χαρακτηριστικό v σύμφωνα με την εξίσωση:

$$S_v = \frac{\left|\mu_v^i - \mu_v^j\right|}{\sqrt{(\sigma_v^i)^2 + (\sigma_v^j)^2}}$$

εξ. 1

Όπου μ^j_v , μ^i_v είναι οι μέσες τιμές και σ^j_v, σ^i_v οι τυπικές αποκλίσεις σε κάθε κλάση. Μεγάλη διαχωριστικότητα σημαίνει ότι το χαρακτηριστικό έχει μεγάλη ικανότητα να διαχωρίζει τις δύο κλάσεις μεταξύ τους.

1.2.1.1.3. Ανάλυση της ομοιότητας (Correlation)

Για την ανάλυση της ομοιότητας των χαρακτηριστικών εκτιμάται ο *συντελεστής συσχέτισης (correlation factor)* για κάθε δύο χαρακτηριστικά *ν* και *λ* που ανήκουν στην ίδια τάξη *p*, σύμφωνα με την εξίσωση:

$$C_{v\lambda}^p = \frac{\frac{1}{K_p}\sum_{\kappa=1}^{K_p}(x_{v\kappa} - \mu_v)(x_{\lambda\kappa} - \mu_\lambda)}{\sigma_v \sigma_\lambda}$$

εξ. 2

K_p είναι το πλήθος των στοιχείων της κλάσης p, $X_{v\kappa}$ και $X_{\lambda\kappa}$ οι τιμές των χαρακτηριστικών, μ_v, μ_λ, σ_v, σ_λ οι μέσες τιμές και οι τυπικές αποκλίσεις των χαρακτηριστικών στην τάξη p.

Ο παράγοντας συσχέτισης μετρά την ομοιότητα μεταξύ των δύο χαρακτηριστικών και παίρνει τιμές μεταξύ –1 και +1. Μια τιμή κοντά στο +1 ή στο -1 σημαίνει ότι τα δύο χαρακτηριστικά συσχετίζονται πολύ ή συσχετίζονται αντίστροφα. Μια τιμή κοντά στο μηδέν δείχνει ότι τα χαρακτηριστικά είναι κατά πολύ ασυσχέτιστα. Έτσι χρησιμοποιώντας τις τρεις παραπάνω απλές ιδιότητες και τύπους μπορούμε να επιλέξουμε τα καλύτερα χαρακτηριστικά ως προς τις δοθείσες κατηγορίες χωρίς να κάνουμε πολλούς υπολογισμούς και χωρίς να δουλέψουμε με πιθανότητες.

1.2.1.2. Αμοιβαία πληροφορία (Mutual information)

Στην εργασία μας έχουμε επιλέξει να χρησιμοποιήσουμε επιλογή χαρακτηριστικών με φίλτρο αμοιβαίας πληροφορίας, θα εξηγήσουμε τον ορισμό της και πως υλοποιούμε τον τύπο της μαθηματικά στο Matlab (κεφάλαιο 2). Είναι παρόμοια με τον συντελεστή συσχέτισης με την διάφορα ότι λαμβάνει υπόψη και τις μη γραμμικές εξαρτήσεις μεταξύ των γνωρισμάτων.

1.2.2. Μέθοδος περιτυλίγματος (Wrapper)

Εκτιμώντας ότι οι μέθοδοι φίλτρων επιλέγουν χαρακτηριστικά γνωρίσματα βασισμένα στα κριτήρια επιλογής χωρίς χρησιμοποίηση ταξινομητή, οι μέθοδοι περιτυλίγματος ενσωματώνουν τους ταξινομητές (classifiers) μέσα στην αναζήτηση υποσυνόλων χαρακτηριστικών γνωρισμάτων.

Επιπλέον, στις μεθόδους Wrapper τα χαρακτηριστικά γνωρίσματα συνήθως αξιολογούνται σε ομάδες και όχι χωριστά. Όλα τα ήδη διαθέσιμα χαρακτηριστικά γνωρίσματα χρησιμοποιούνται για να παραγάγουν τα υποσύνολα χαρακτηριστικών γνωρισμάτων που αξιολογούνται. Η αξιολόγηση ενός συγκεκριμένου υποσυνόλου χαρακτηριστικών γνωρισμάτων γίνεται με την κατάρτιση και τη δοκιμή ενός συγκεκριμένου προτύπου ταξινόμησης. Για να βρεθεί το διάστημα όλων των υποσυνόλων χαρακτηριστικών γνωρισμάτων, ένας αλγόριθμος αναζήτησης είναι συνέχεια τυλιγμένος (wrapped) γύρω από το πρότυπο ταξινόμησης .

Εντούτοις, καθώς το διάστημα του υποσυνόλου χαρακτηριστικών γνωρισμάτων αυξάνεται εκθετικά με τον αριθμό χαρακτηριστικών γνωρισμάτων, οι ευρετικές (heuristic) μέθοδοι αναζήτησης χρησιμοποιούνται για να καθοδηγήσουν την αναζήτηση ενός βέλτιστου υποσυνόλου. Αυτές οι μέθοδοι αναζήτησης μπορούν να ταξινομηθούν στους αιτιοκρατικούς (deterministic) και τυχαίους (random) αλγορίθμους αναζήτησης.

Στα πλεονεκτήματα των προσεγγίσεων του Wrapper περιλαμβάνεται η αλληλεπίδραση μεταξύ της αναζήτησης υποσυνόλων χαρακτηριστικών γνωρισμάτων, καθώς και η δυνατότητα να ληφθούν υπόψη οι εξαρτήσεις μεταξύ των χαρακτηριστικών γνωρισμάτων. Μειονέκτημα είναι ότι υπάρχει μεγαλύτερο ρίσκο για υπερφόρτωση (overfiting) από τις μεθόδους Wrapper σε αντίθεση με τις τεχνικές φίλτρων, πολύ συχνό φαινόμενο κατά την εκπαίδευση (training) του ταξινομητή .

2. Μαθηματικό Μέρος

Επειδή η εργασία μας αφορά την επιλογή χαρακτηριστικών με το φίλτρο της αμοιβαίας πληροφορίας, θα δούμε στο κεφαλαίο αυτό μια εισαγωγή στην θεωρία πιθανοτήτων εφόσον ο τύπος της αμοιβαίας πληροφορίας βασίζεται κυρίως σε πιθανότητες. Επίσης θα αναφερθούμε στην θεωρία πληροφορίας αφού έχουμε να κάνουμε με πίνακες που περιέχουν τιμές και ουσιαστικά πληροφορίες. Στην συνέχεια μιλάμε για τον Shannon, πατέρας της θεωρίας πληροφορίας, εξηγούμε και ορίζουμε τους μαθηματικούς τύπους: εντροπία, αμοιβαία πληροφορία, υπό συνθήκη αμοιβαία πληροφορία, και αμοιβαία πληροφορία mRmR.

2.1. Θεωρία πιθανοτήτων

Η θεωρία πιθανοτήτων είναι ο κλάδος των μαθηματικών ο οποίος ασχολείται με την ανάλυση τυχαίων φαινομένων. Κεντρικό ρόλο στη θεωρία πιθανοτήτων παίζει η έννοια της πιθανότητας, ενώ σημαντικές είναι οι τυχαίες μεταβλητές. Αν και τα γεγονότα που μελετώνται από τη θεωρία πιθανοτήτων, όπως π.χ. η ρίψη ενός ζαριού ή το στρίψιμο ενός κέρματος είναι τυχαία, όταν επαναλαμβάνονται πολλές φορές η αλληλουχία των τυχαίων γεγονότων παρουσιάζει ορισμένα στατιστικά μοτίβα τα οποία μπορούν να μελετηθούν και να προβλεφθούν .

Ως μαθηματικό θεμέλιο της στατιστικής, η θεωρία πιθανοτήτων είναι απαραίτητη σε πολλές δραστηριότητες που περιλαμβάνουν ανάλυση μεγάλων συνόλων δεδομένων. Μέθοδοι της θεωρίας πιθανοτήτων εφαρμόζονται και στην περιγραφή πολύπλοκων συστημάτων, όπως στη στατιστική μηχανική [6].

Στην εργασία αυτή θα παρουσιάσουμε τους συμβολισμούς που χρησιμοποιούμε για τις πιθανότητες και τις έννοιες που εκφράζουν αυτές. Για παράδειγμα όταν σε έναν τύπο

έχουμε την πιθανότητα P(x) εννοούμε την συχνότητα εμφάνισης του χ σε ένα διάσημα τιμών. Υπολογίζεται μετρώντας το πλήθος εμφάνισης του χ δια το σύνολο των στοιχειών.

2.2. Θεωρία πληροφορίας

Θεωρία Πληροφορίας είναι το πεδίο εκείνο που ασχολείται με την έννοια της «πληροφορίας», τα μέτρα και τις εφαρμογές της. Πιο συγκεκριμένα, στα θέματα που απασχολούν τη Θεωρία Πληροφορίας συγκαταλέγονται η εντροπία και οι μονάδες μέτρησης αυτής, η ροή πληροφορίας σε κανάλια και τα θεμελιώδη όρια της ποσότητας πληροφορίας που μπορούν να μεταδοθούν. Ακόμα, θέματα που την απασχολούν είναι η κατασκευή συστημάτων επεξεργασίας και επικοινωνίας πληροφορίας που μπορούν να προσεγγίσουν αυτά τα ανωτέρω όρια κ.ά. .

Η λέξη πληροφορία έχει πολλές έννοιες, οπότε πρέπει να γίνει διάκριση μεταξύ της πληροφορίας με την έννοια ποσότητα δεδομένων και της πληροφορίας με την έννοια σημαντικότητας δεδομένων .

Η Θεωρία Πληροφορίας απαντά, κατά βάση, σε δύο θεμελιώδεις ερωτήσεις: Ποια είναι η μεγαλύτερη δυνατή συμπίεση δεδομένων και ποιος ο μέγιστος δυνατός ρυθμός μετάδοσης σε ένα επικοινωνιακό κανάλι. Όριο της συμπίεσης δεδομένων αποτελεί η μέση ποσότητα πληροφορίας (εντροπία), ενώ όριο του ρυθμού μετάδοσης σε ένα κανάλι αποτελεί η χωρητικότητά του [7].

Όλοι οι λογάριθμοι που χρησιμοποιούνται στη εργασία αυτή θα είναι με βάση το δύο.

2.2.1. Πληροφορία

Ο κύριος στόχος της θεωρίας πληροφοριών είναι η ποσοτικοποίηση της εννοίας της πληροφορίας. Αν καταφέρουμε να κάνουμε κάτι τέτοιο θα μπορέσουμε να αναπτύξουμε μεθόδους ανάλυσης και σχεδιασμού συστημάτων πληροφορίας με τη χρήση της αριθμητικής και του λογισμού .

Κατά την κοινή αντίληψη πληροφορία είναι η νέα γνώση που αποκτάται μέσω ακοής, όρασης, κλπ.. Πληροφορία είναι μια συλλογή δεδομένων. Είναι το κοινό νόημα που αποδίδεται σε ένα απλό ή σύνθετο σύμβολο από δύο ή περισσότερα υποκείμενα. Ως έννοια είναι σύνθετη και αποτελείται από τις λέξεις "φέρω" και "πλήρης". Κατ' επέκταση η έννοια "πληροφορία" σηματοδοτεί εκείνο το νοηματικό περιεχόμενό που είναι ολοκληρωμένο και σαφές. Στην Πληροφορική η Πληροφορία σηματοδοτείται από την ποιοτική αξία του bit (0 ή

1). Άλλωστε η πραγματική "νοηματική" αξία για έναν Η/Υ είναι μια σωρεία από 0 και 1. Ο υπολογιστής επεξεργάζεται δεδομένα και παράγει επεξεργασμένα δεδομένα. Ο άνθρωπος προσδίδει νόημα στα επεξεργασμένα δεδομένα μετατρέποντας τα σε πληροφορία. Πηγή πληροφορίας είναι κάθε σύστημα που παράγει στην έξοδο του πληροφορία .

2.2.2. Μετρό πληροφορίας του Shannon

Ο Κλοντ Σάνον (Claude Elwood Shannon, 30 Απριλίου 1916 – 24 Φεβρουαρίου 2001) ήταν Αμερικανός μαθηματικός και ηλεκτρολόγος μηχανικός γνωστός ως "ο πατέρας της θεωρίας της πληροφορίας" και της κρυπτογραφίας .

Ο Shannon θεμελίωσε στη δεκαετία του '40 τη θεωρία της πληροφορίας και ανέδειξε την πληροφορία σε μετρήσιμο μέγεθος. Έθεσε έτσι τα θεμέλια για τα τηλεπικοινωνιακά δίκτυα και βοήθησε να αναπτυχθεί η σημερινή Κοινωνία της Πληροφορίας. Είχε επίσης εξαιρετική συνεισφορά στην ανάπτυξη των ηλεκτρονικών υπολογιστών, αφού πρώτος έδειξε ότι με βάση την άλγεβρα Boole είναι δυνατή η δημιουργία ψηφιακών ηλεκτρικών κυκλωμάτων για την επίλυση μαθηματικών εξισώσεων. Επίσης, θεωρείται από τους πατέρες της τεχνητής νοημοσύνης, ενώ η εργασία του στη θεωρία της πληροφορίας έχει βρει εφαρμογή και σε άλλους τομείς όπως η γλωσσολογία, η φωνητική, η θεωρία του Χάους και η κρυπτογραφία [8].

2.2.3. Εντροπία - Μέση πληροφορία. (Entropy)

Η εντροπία είναι ένα μέτρο της αβεβαιότητας των διαφόρων πιθανοτήτων. Όσο μεγαλύτερη είναι η εντροπία, τόσο μεγαλύτερη είναι η αβεβαιότητα, επομένως τόσο μεγαλύτερη είναι η ποσότητα της μεταφερόμενης πληροφορίας .

Επιπλέον, όταν υπάρχουν Μ μεταβλητές, μπορεί να αποδειχτεί ότι η μέση πληροφορία ανά μεταβλητή γίνεται μέγιστη όταν όλες οι μεταβλητές είναι ισοπίθανες και η τιμή της εντροπίας δίνεται από την εξίσωση :

$$H_{max} = \sum_{i=1}^{M} \frac{1}{M} \log(M) = \log(M) \qquad \text{εξ. 3}$$

Αν Χ είναι μια διακριτή τυχαία μεταβλητή με δειγματοχώρο Χ = $\{x_1, x_2, \ldots, x_n\}$ και συνάρτηση πιθανότητας πυκνότητας p(xi), τότε η μέση ποσότητα πληροφορίας (ή μέση πληροφορία ή μέσο πληροφορικό περιεχόμενο ή εντροπία) της Χ ,Η(Χ) , δίνεται από τη εξίσωση :

$$H(X) = -\sum_{i=1}^{n} p(Xi) \log(Xi) \qquad \text{εξ. 4}$$

2.2.5. Από Κοινού Εντροπία (Joint Entropy)

Πολλές φορές μάς ενδιαφέρει να εξετάσουμε την ποσότητα πληροφορίας ενός συνδυασμού δύο τυχαίων μεταβλητών, δηλαδή ενός πειράματος που αποτελείται από δύο υποπειράματα. Ένα τυχαίο πείραμα (X,Y) (εξίσωση 5) έχει ως δυνατά αποτελέσματα όλους τους συνδυασμούς των αποτελεσμάτων μεταξύ των $X=\{x_1, x_2, x_3,\ldots x_n\}$ και $Y=\{y_1, y_2, y_3,\ldots y_n\}$ επομένως το δειγματοχωρο :

$(X,Y)=\{(x_1,y_1),(x_1,y_2)\ldots.(x_1,y_n),(x_2,y_1),(x_2,y_2)\ldots\ldots(x_2,y_n)\ldots\ldots.(x_n,y_n)\}$ **εξ. 5**

Η κατανομή πιθανοτήτων δίνετε από την εξίσωση :

$P=\{p(x_1,y_1),p(x_1,y_2)\ldots.p(x_1,y_n),p(x_2,y_1),p(x_2,y_2)\ldots\ldots p(x_2,y_n)\ldots\ldots.p(x_n,y_n)\}$ **εξ. 6**

Αν (X,Y) είναι ένα τυχαίο πείραμα με δισδιάστατο δειγματοχώρο και η κατανομή πιθανοτήτων είναι όπως στην εξίσωση 6, τότε η από κοινού εντροπία (joint entropy) *H(X,Y)* ορίζεται όπως στην εξίσωση :

$$H(X,Y) = -\sum_{x\in\mathscr{X}}\sum_{y\in\mathscr{Y}} p(x,y)\log p(x,y).$$

εξ. 7

Όπου p(x,y) είναι η πιθανότητα εμφάνισης του x στο διάνυσμα του X και ταυτόχρονα y στο διάνυσμα του Y.

2.2.4. Υπό Συνθήκη Εντροπία (Conditional entropy)

Πολλές φορές μας ενδιαφέρει να υπολογίσουμε την ποσότητα πληροφορίας μιας τυχαίας μεταβλητής X, όταν δίνεται το αποτέλεσμα μιας άλλης τυχαίας μεταβλητής Y. Αυτή καλείται υπό συνθήκη (Conditional) ποσότητα πληροφορίας της X ως προς την Y. Η υπό συνθήκη ποσότητα πληροφορίας του αποτελέσματος x_i αν είναι γνωστό ότι έχει λάβει χώρα το αποτέλεσμα y_j δίνεται από την εξίσωση :

$$H(X\,|\,Y) = -\sum_{x}\sum_{y} p(x,y)\log p(x\,|\,y)$$

εξ. 8

Όπου p(x,y) είναι η πιθανότητα εμφάνισης του χ στο διάνυσμα του X και ταυτόχρονα y στο διάνυσμα του Y. P(x|y) είναι η πιθανότητα εμφάνισης του χ γνωρίζοντας πως έχει εμφανιστεί ήδη το y.

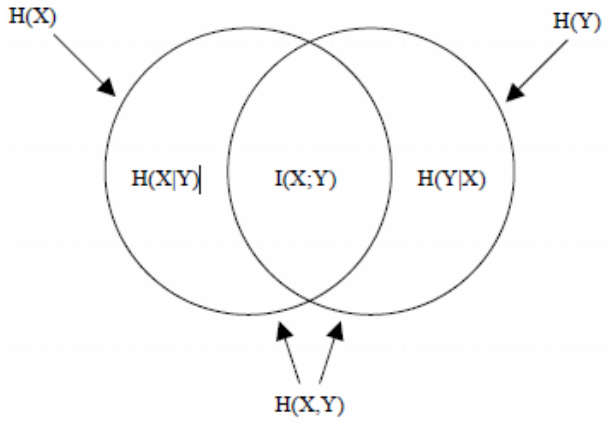
2.2.5. Αμοιβαία Πληροφορία (Mutual Information)

Η αμοιβαία πληροφορία είναι ένα μέτρο της ποσότητας της πληροφορίας που περιέχει μια τυχαία μεταβλητή για μια άλλη τυχαία μεταβλητή ή αλλιώς ένα μέτρο της εξάρτησης μεταξύ δύο τυχαίων μεταβλητών [9]. Η αμοιβαία πληροφορία δύο τυχαίων μεταβλητών *X* και *Y* ορίζεται από την εξίσωση :

$$\begin{aligned} I(X;Y) &= H(X) - H(X|Y) \\ &= H(Y) - H(Y|X) \\ &= H(X) + H(Y) - H(X,Y) \\ &= H(X,Y) - H(X|Y) - H(Y|X) \end{aligned}$$

εξ. 9

Εικόνα 1: εικονική αναπαράσταση της αμοιβαίας πληροφορίας όπου βλέπουμε πως η αμοιβαία πληροφορία είναι η κοινή πληροφορία ανάμεσα στην εντροπία H(X) και H(Y).



Ύστερα από πράξεις στην εξίσωση 9 καταλήγουμε στην εξίσωση :

$$I(X;Y) = \sum_{y \in Y} \sum_{x \in X} p(x,y) \log \left(\frac{p(x,y)}{p_1(x)\, p_2(y)} \right),$$

εξ. 10

Όπου η p(x) είναι η πιθανότητα να έχουμε την τιμή x στο διάνυσμα του X και αντίστοιχα p(y) η πιθανότητα να έχουμε την τιμή y στο διάνυσμα του Y. Η p(x,y) είναι η δεσμευμένη πιθανότητα (joint probability) να έχουμε x στο διάνυσμα του X και ταυτόχρονα y στο διάνυσμα του Y. Στο τύπο αυτό έχουμε άθροισμα μέσα σε άθροισμα και πάντα το τελικό αποτέλεσμα διαιρείται με το συνολικό αριθμό τον αθροισμάτων για να βγάλουμε έναν

μέσο όρο. Το σκορ αυτό μας δείχνει κατά πόσο τα δυο διανύσματα μοιράζονται κοινή πληροφορία μεταξύ τους.

Η αμοιβαία πληροφορία I(X;Y) μετρά πόσο (κατά μέσον όρο) η πραγματοποίηση της τυχαίας μεταβλητής Y μας λέει για την πραγματοποίηση της X, δηλαδή πώς η εντροπία της X μειώνεται εάν ξέρουμε την πραγματοποίηση της Y .

Για παράδειγμα, η αμοιβαία πληροφορία μεταξύ ενός θορύβου περιβάλλοντος και του ήχου περιβάλλοντος μας δείχνουν πόσο κατά μέσον όρο ο θόρυβος δίνει πληροφορίες για το περιβάλλον. Εάν ο θόρυβος είναι τέλειος πληροφοριακά (ίδιος με τον ήχο), και αναφέρεται αποκλειστικά στο περιβάλλον ,τίποτα επιπλέον, τότε η αμοιβαία πληροφορία μεταξύ του θορύβου και ήχου περιβάλλοντος είναι απλά η εντροπία του ήχου όπως βλέπουμε στην εξίσωση :

$$I(X; Y) = H(X) - H(X|Y) = H(X) - H(X|X) = H(X) \quad \textbf{εξ. 11}$$

Με άλλα λόγια, η αμοιβαία πληροφορία μεταξύ μίας τυχαίας μεταβλητής και της ίδιου είναι η εντροπία της όπως βλέπουμε στην εξίσωση :

$$I(X;X) = H(X) \quad \textbf{εξ. 12}$$

Η αμοιβαία πληροφορία είναι συμμετρική. Η μεταβλητή X μας λέει ακριβώς τόσα για την Y όσα η Y για την X ,όπως φαίνεται στην εξίσωση :

$$I(X; Y) = H(X) - H(X|Y) = H(Y) - H(Y|X) = I(Y,X). \quad \textbf{εξ. 13}$$

2.3. Υπό Συνθήκη Αμοιβαία Πληροφορία (Conditional Mutual Information-CMI)

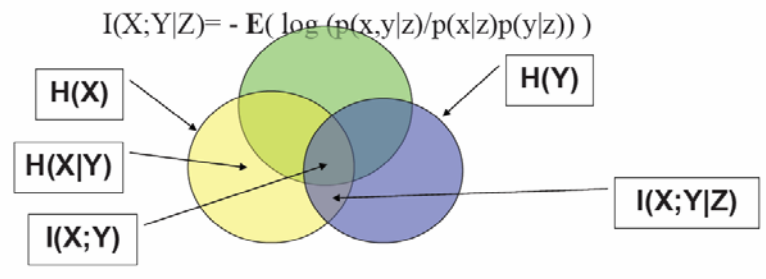
Στην περίπτωση της επιλογής χαρακτηριστικών πολύ συχνά χρησιμοποιείται η CMI μεταξύ χαρακτηριστικών F και των κλάσεων C στις οποίες χωρίζονται τα δεδομένα.

Αν ο αριθμός των χαρακτηριστικών είναι μεγάλος και θέλουμε να επιλέξουμε μοναδικά χαρακτηριστικά χρησιμοποιώντας την αμοιβαία πληροφορία $I(F_1,\ldots,F_k; C)$, είναι αδύνατον να το επιτύχουμε με τον απλό τύπο. Ο χρόνος εκτέλεσης του αλγορίθμου θα ήταν τεράστιος και θα ήταν ανούσιο να κάνουμε κάτι τέτοιο [10], γιατί θα πρέπει να υπολογίζουμε όλες τους συνδυασμούς μεταξύ όλων των πιθανοτήτων, μεταξύ όλων των διαστάσεων. Υπάρχουν δυο τρόποι τοποθέτησης τιμών σε χωρία (bins), Equidistance και Equiprobable.

Η Equidistance χωρίζει όλο το εύρος των τιμών σε όσα τμήματα θέλουμε εμείς και βάζει τις τιμές που βρίσκονται ανάμεσα σε αυτές στα αντίστοιχα χωρία (bins). Η Equiprobable αφού πάρει την μεγαλύτερη και την μικρότερη τιμή, διαιρεί το εύρος αυτό με τον

επιθυμητό αριθμό χωρίων δημιουργώντας τα αντίστοιχα τμήματα. Σαρώνοντας όλες τις τιμές ελέγχει σε πιο τμήμα ανήκει η κάθε μια και την τοποθετεί στο αντίστοιχο χωρίο. Στην εργασία μας έχουμε επιλέξει την Equiprobable και πριν από την χρήση της κάνουμε κανονικοποιήση των τιμών, έτσι ώστε να έχουμε ελάχιστο 0 και μέγιστο 1.

Εικόνα 2: Εικονική αναπαράσταση της υπό συνθήκη αμοιβαία πληροφορίας.



Για να αποφύγουμε τον απλό τύπο μπορούμε να χρησιμοποιήσουμε την **Conditional Mutual Information** [11] όπως την βλέπουμε στην απλούστερη μορφή της στην εξίσωση **:**

$$I(X;Y \mid Z) = H(X,Z) + H(Y,Z) - H(X,Y,Z) - H(Z) \qquad \textbf{εξ. 14}$$

Παρακάτω βλέπουμε τον ψευδοκώδικα από τον αλγόριθμο επιλογής χαρακτηριστικών με υπό συνθήκη αμοιβαία πληροφορία (Conditional Mutual Information)

Παράμετροι Εισόδου:

n – Ο αριθμός των χαρακτηριστικών που θέλουμε να επιλέξουμε.

v – Ο αριθμός των χαρακτηριστικών συνολικά.

Παράμετροι Εξόδου:

F – Το σετ των επιλεγμένων χαρακτηριστικών.

1. Ορίζουμε το F ίσο με κενό.
2. θέτουμε m=1
3. Προσθήκη F_i Μέσα στο F, όπου $F_i = \text{argmax}_{i=1..v}\, I(F_i;C)$
4. Επανάληψη
5. m++ (Αύξηση του m κατά ένα)
6. Προσθήκη F_i Μέσα στο F, όπου $F_i = \text{argmax}_{i=1..v}\, \{\min_{F\text{j ανήκει } F} I(F_i;C|F_j)\}$
7. Μεχρις ότου m=n

Με λίγα λόγια στην αρχή του αλγόριθμου χρησιμοποιούμε τον απλό τύπο της αμοιβαίας πληροφορίας για να βρούμε το χαρακτηριστικό με το μεγαλύτερο σκορ σχετικότητας ως προς την κλάση C.

Αργότερα για να βρούμε τα υπόλοιπα χαρακτηριστικά χρησιμοποιούμε την υπό συνθήκη αμοιβαία πληροφορία (Conditional Mutual Information). Σε κάθε επανάληψη επιλεγούμε το μεγαλύτερο σκορ σχετικότητας από το σύνολο των σκορ που επιλεχτήκαν ανάμεσα στα μικρότερα της υπό συνθήκη αμοιβαίας πληροφορίας και στο χαρακτηριστικό Fi, που ελέγχεται από το σύνολο των χαρακτηριστικών και τα ήδη επιλεγμένα F. Δηλαδή αν στον τύπο του CMI βάλουμε στο F_i , F_j το χαρακτηριστικό που έχουμε επιλέξει ήδη, τότε το σκορ που θα μας δώσει είναι το 0. Όταν εμείς θα επιλέξουμε από το σύνολο των σκορ το μεγαλύτερο τότε αποκλείουμε την περίπτωση να επιλέξουμε ξανά το ίδιο χαρακτηριστικό η παρόμοιο.

Έτσι επιλεγούμε σε κάθε επανάληψη μονό καλά χαρακτηριστικά προς την κλάση μας και όχι όμοια η ίδια μεταξύ τους.

$$
\begin{aligned}
I(F^*; C \mid F_1,\ldots,F_k) &\approx \min I(F^*; C \mid \underbrace{F_i,\ldots,F_j}_{k-1}) \\
I(F^*; C \mid F_1,\ldots,F_k) &\approx \min I(F^*; C \mid \underbrace{F_i,\ldots,F_j}_{k-2}) \\
&\vdots \\
I(F^*; C \mid F_1,\ldots,F_k) &\approx \min I(F^*; C \mid F_i, F_j) \\
I(F^*; C \mid F_1,\ldots,F_k) &\approx \min I(F^*; C \mid F_i)
\end{aligned}
\qquad \text{εξ. 15}
$$

Ως πηγή του παραπάνω αλγορίθμου σύμφωνα με την εξ.15 και εικόνα 2.α πήραμε στοιχεία από την αναφορά [12] και .

2.4. Επιλογή χαρακτηριστικών βασισμένη στην Αμοιβαία πληροφορία Μέγιστης Σχετικότητας και Ελάχιστου Πλεονασμού (Feature Selection Based on Mutual Information: Criteria of Max-Dependency, Max-Relevance,and Min-Redundancy)

Η επιλογή χαρακτηριστικών γνωρισμάτων είναι ένα σημαντικό πρόβλημα για τα συστήματα ταξινόμησης. Μελετάμε πώς να επιλέξουμε τα καλά χαρακτηριστικά γνωρίσματα σύμφωνα με το κριτήριο μέγιστης εξάρτησης βασισμένο στην αμοιβαία πληροφορία. Η επιλογή χαρακτηριστικών γνωρισμάτων με μέγιστη εξάρτηση μπορεί να

επιλέξει έναν πολύ μικρό αριθμό χαρακτηριστικών γνωρισμάτων. Όμως όταν ο αριθμός των δειγμάτων είναι μεγάλος, δεν είναι θετικό για τις εφαρμογές που έχουν στόχο την υψηλή ακρίβεια και την ταχύτατη επιλογή χαρακτηριστικών.

Επειδή είναι μια αρκετά αργή μέθοδος, όταν ο αριθμός κοινών δειγμάτων αυξάνεται πολύ γρήγορα και πλησιάζει με τον αριθμό δειγμάτων, η κοινή πιθανότητα αυτών των χαρακτηριστικών γνωρισμάτων, καθώς επίσης και οι αμοιβαίες πληροφορίες, δεν μπορούν να υπολογιστούν σωστά . Λόγω της δυσκολίας να εφαρμόσουμε άμεσα το όρο μέγιστης εξάρτησης, παράγουμε αρχικά μια ισοδύναμη μορφή, αποκαλούμενη κριτήριο ελάχιστου πλεονασμού και μέγιστης σχετικότητας (mRmR). Αυτό επιτρέπει σε μας να επιλέξουμε ένα σύνολο βέλτιστων χαρακτηριστικών γνωρισμάτων με πολύ χαμηλότερο υπολογιστικό κόστος.

Μια από τις δημοφιλέστερες προσεγγίσεις για να πραγματοποιήσουμε επιλογή των χαρακτηριστικών γνωρισμάτων με την υψηλότερη σχετικότητα ως προς τη κλάση C. είναι η μέγιστη σχετικότητα (max relevance). Η σχετικότητα χαρακτηρίζεται συνήθως από το συσχετισμό ή την αμοιβαία πληροφορία μεταξύ των χαρακτηριστικών. Η αμοιβαία πληροφορία είναι ένα από τα ευρέως χρησιμοποιημένα μέτρα για να καθοριστεί η εξάρτηση μεταβλητών. Στην εργασία μας εστιάζουμε απόλυτα στη επιλογή χαρακτηριστικών γνωρισμάτων βασισμένη στη αμοιβαία πληροφορία.

Στην μέγιστη σχετικότητα (Max-Relevance), τα επιλεγμένα χαρακτηριστικά γνωρίσματα x_i κρατούνται χωριστά για να έχουν τις μεγαλύτερες αμοιβαίες πληροφορίες $I(X_i;c)$ ως προς την κλάση C. Μετά από διαδοχικές αναζητήσεις, το m θα περιέχει τα καλύτερα μεμονωμένα χαρακτηριστικά γνωρίσματα [13].

Στην επιλογή χαρακτηριστικών γνωρισμάτων, έχει αναγνωριστεί ότι οι συνδυασμοί καλών χαρακτηριστικών γνωρισμάτων δεν είναι πάντα ορθά στην σωστή εκτέλεση ταξινόμησης. Με άλλα λόγια, τα καλύτερα χαρακτηριστικά γνωρίσματα m δεν είναι πάντα τα καλύτερα». Μερικοί ερευνητές έχουν μελετήσει τρόπους για να μειώσουν τον πλεονασμό μεταξύ χαρακτηριστικών γνωρισμάτων και να επιλέξουν τα χαρακτηριστικά γνωρίσματα με τον ελάχιστο πλεονασμό (minimal redundancy). Σύμφωνα με τους Hanchuan Peng, Fuhui Long, και Chris Ding προτείνεται ένα συνδυαστικό πλαίσιο ελάχιστου πλεονασμού μέγιστης σχετικότητας (minimal redundancy maximal relevance, mRMR) για να ελαχιστοποιηθεί ο πλεονασμός και να χρησιμοποιηθούν μια σειρά από τεχνικές που προσπαθούν να

προβλέψουν ποια είναι τα βέλτιστα χαρακτηριστικά που θα παρουσιάσουν τον ελάχιστο πλεονασμό και τη μεγίστη σχετικότητα [14].

2.4.1. Η Σχέση Ανάμεσα Στην Μέγιστη Εξάρτιση , Μέγιστη Σχετικότητα, και Ελάχιστος Πλεονασμός

2.4.1.1. Μέγιστη Εξάρτιση (Max Dependency)

Εφαρμόζοντας την αμοιβαία πληροφορία, για την επιλογή χαρακτηριστικών ο σκοπός είναι να βρεθεί το σετ χαρακτηριστικών S με m χαρακτηριστικά γνωρίσματα $\{x_{i=1\ldots m}\}$, τα οποία έχουν από κοινού τη μεγαλύτερη εξάρτηση στην κατηγορία C. Αυτό το σχέδιο, αποκαλούμενο μέγιστη εξάρτηση(Max-Dependency) έχει την ακόλουθη μορφή:

$$\max D(S,c), \quad D = I(\{x_i, i = 1, \ldots, m\}; c)$$

εξ. 16

Όπου D είναι το σετ με όλα τα σκορ εξάρτισης ανάμεσα στα χαρακτηριστικά x_i, c και max D είναι το μέγιστο σκορ από αυτά.

Προφανώς, όταν m είναι ίσο με το 1, η λύση είναι το χαρακτηριστικό γνώρισμα που μεγιστοποιεί την $I(X_j;c)$ με (1 =< j =< M).

Παρά τη θεωρητική αξία της μέγιστης εξάρτησης, είναι συχνά δύσκολα να πάρει μια ακριβή εκτίμηση για πολλές πυκνότητες πιθανότητας $p(x_1 \ldots\ldots x_m)$ και $p(x_1 \ldots\ldots x_m, c)$, λόγω δύο δυσκολιών που παρουσιάζονται : α) το μεγάλο διάστημα δεδομένων β) η απότομη αύξηση κοινής περιοχής. Παραδείγματος χάριν, ας υποθέσουμε ότι κάθε χαρακτηριστικό γνώρισμα έχει N δείγματα που ανήκουν σε τρείς κατηγορίες. Τα χαρακτηριστικά γνωρίσματα K θα μπορούσαν να έχουν μια μέγιστη 3^K έως ελάχιστη N κοινή περιοχή. Όταν ο αριθμός από τη κοινή περιοχή αυξάνεται πολύ γρήγορα και πλησιάζουν στον αριθμό των δειγμάτων N η κοινή πιθανότητα αυτών των χαρακτηριστικών γνωρισμάτων, καθώς επίσης και οι αμοιβαίες πληροφορίες, δεν μπορούν να υπολογιστούν σωστά [15].

2.4.1.2 Μέγιστη Σχετικότητα Ελάχιστος Πλεονασμός (Max-Relevance and Min-Redundancy)

Δεδομένου ότι το κριτήριο μέγιστης εξάρτησης είναι δύσκολο να εφαρμοστεί, η εναλλακτική λύση είναι να επιλεχτούν τα χαρακτηριστικά γνωρίσματα βασισμένα στο κριτήριο μέγιστης σχετικότητας. Η μέγιστη σχετικότητα πρόκειται να ψάξει τα χαρακτηριστικά γνωρίσματα που ικανοποιούν τον παρακάτω τύπο, ο οποίος προσεγγίζει το

σετ D(S,c) με τη μέση αξία όλων των αμοιβαίων τιμών πληροφοριών μεταξύ του μεμονωμένου χαρακτηριστικού γνωρίσματος xi και της κατηγορίας c στην παρακάτω εξίσωση όπου το max D(S,c) είναι το μέγιστο σκορ σχετικότητας.

$$\max D(S,c), \quad D = \frac{1}{|S|}\sum_{x_i \in S} I(x_i;c).$$

εξ. 17

Είναι πιθανό τα χαρακτηριστικά γνωρίσματα που θα επιλέξουμε σύμφωνα με την εξίσωση 17 να έχουν πλούσιο πλεονασμό, δηλαδή η εξάρτηση μεταξύ αυτών των χαρακτηριστικών γνωρισμάτων θα μπορούσε να είναι μεγάλη. Όταν δύο ιδιαίτερα χαρακτηριστικά γνωρίσματα εξαρτούνται το ένα από το άλλο στην αντίστοιχη κατηγορία (κλάση C) τότε η ισχύς τους δεν θα άλλαζε πολύ εάν αφαιρούσαμε ένα από τα δυο . Επομένως, ο ελάχιστος πλεονασμός (Min Redundancy) μπορεί να προστεθεί για να επιλέξει τα αποκλειστικά χαρακτηριστικά γνωρίσματα στην εξίσωση 18 όπου το R(S) περιέχει το σκορ πλεονασμού ανάμεσα στα ήδη επιλεγμένα χαρακτηριστικά S και x_i. Έτσι αν τα $x_i \in F$ και $x_j \in S$ μας δώσουν μικρό σκορ σημαίνει πως έχουμε ελάχιστο πλεονασμό δηλαδή δεν έχουμε κοινά ή όμοια χαρακτηριστικά. Έτσι για να επιλέξουμε μοναδικά χαρακτηριστικά, τα ελέγχουμε όλα με τα ήδη επιλεγμένα και αν είναι διαφορετικά μεταξύ τους θα μας δώσουν μικρό σκορ.

$$\min R(S), \quad R = \frac{1}{|S|^2}\sum_{x_i, x_j \in S} I(x_i, x_j).$$

εξ. 18

Το κριτήριο που συνδυάζει τους ανωτέρω δύο περιορισμούς καλείται ελάχιστος πλεονασμός και μέγιστη σχετικότητα (mRMR). Καθορίζουμε δυο εξισώσεις που συνδυάζουν το D και το R. Στην πρώτη (εξίσωση 19) γίνεται αφαίρεση ανάμεσα στην μέγιστη σχετικότητα D και του ελάχιστου πλεονασμού R. Στην δεύτερη η οποία είναι ίδια με την εξίσωση 19 αντί για αφαίρεση έχουμε διαίρεση.

Στην πράξη, οι παραπάνω μέθοδοι αναζήτησης μπορούν να χρησιμοποιηθούν για να βρουν τα βέλτιστα χαρακτηριστικά γνωρίσματα. Υποθέστε ότι έχουμε ήδη σετ χαρακτηριστικών S(m-1) και χαρακτηριστικά γνωρίσματα m-1. Στόχος είναι να επιλέξουμε το $m^{-\text{οστό}}$ χαρακτηριστικό από το σετ { X – S(m-1)}, αυτό πετυχαίνεται επιλέγοντας το χαρακτηριστικό που δίνει το μέγιστο στην εξίσωση 20 ή 21. Με λίγα λόγια επιλέγουμε

χαρακτηριστικά με μέγιστη σχετικότητα αλλά φροντίζουμε να μην είναι όμοια η ίδια με τα ήδη επιλεγμένα.

$$\max_{x_j \in X - S_{m-1}} \left[I(x_j; c) - \frac{1}{m-1} \sum_{x_i \in S_{m-1}} I(x_j; x_i) \right].$$

εξ. 19

Προσπαθούμε εδώ να δούμε εάν η προσέγγιση mRMR είναι αποτελεσματική με απλούστερες μορφές του. Υλοποιούμε την εξίσωση 20 Αμοιβαία πληροφορία με διαφορά (Mutual information difference) και εξίσωση 21 Αμοιβαία πληροφορία με πηλίκο (Mutual information quotient).

$$\max_{i \in \Omega_S} [I(i,h) - \frac{1}{|S|} \sum_{j \in S} I(i,j)]$$

εξ. 20

$$\max_{i \in \Omega_S} \{I(i,h) / [\frac{1}{|S|} \sum_{j \in S} I(i,j)]\}$$

εξ. 21

Ουσιαστικά στους δυο παραπάνω τύπους για να βρούμε το βέλτιστο χαρακτηριστικό κάθε φορά επιλέγουμε το μέγιστο από το σετ των διαφορών η πηλίκων που έχουμε στο πρώτο μέρος την αμοιβαία πληροφορία ανάμεσα στο διάνυσμα x_i και το διάνυσμα κατηγοριών h . Στο δεύτερο μέρος,τον μέσο όρο των αμοιβαίων πληροφοριών ανάμεσα στην x_i και τα ήδη επιλεγμένα χαρακτηριστικά $x_{j=1...m-1}$. Ο σκοπός μας είναι κάθε φορά, υπολογίζοντας το πρώτο μέρος που η μέγιστη Σχετικότητα είναι ανάμεσα στο διάνυσμα x_i και την κλάση h, να αφαιρούμε ή να διαιρούμε με το δεύτερο μέρος όπου ο ελάχιστος πλεονασμός είναι ανάμεσα στο διάνυσμα x_i και τα είδη επιλεγμένα διανύσματα x_j . Με την μεθοδολογία αυτή προσπαθούμε να επιλέγουμε χαρακτηριστικά όσο το δυνατόν βέλτιστα αλλά όχι ιδία μεταξύ τους . Ο ψευδοκώδικας του αλγορίθμου μέγιστης σχετικότητας και ελάχιστου πλεονασμού δίνεται παρακάτω.

Παράμετροι Εισόδου:

n – Ο αριθμός των χαρακτηριστικών που θέλουμε να επιλέξουμε.

X – Ο αριθμός των χαρακτηριστικών συνολικά.

m – Ο αριθμός των επιλεγμένων χαρακτηριστικών.

Παράμετροι Εξόδου:

S – Το σετ των επιλεγμένων χαρακτηριστικών.

1. Ορίζουμε το S ίσο με καινό.

2. θέτουμε m=1

3. Προσθήκη x_i στο S, όπου xi= $\text{argmax}_{i=1..v}$ $I(x_i;h)$

4. Αφαίρεση βέλτιστου διανύσματος από το σύνολο των διανυσμάτων

5. Επανάληψη

6. m++ (Αύξηση του m κατά ένα)

7. Προσθήκη xi στο S,όπου xi=$\text{argmax}_{i=1..v}$ $\{I(xi;h)- (\Sigma I(x_i; x_{j=1..m-1}))/ m-1\}$

8. Αφαίρεση του νέου βέλτιστου από το σύνολο των διανυσμάτων

9.Μεχρις ότου m=n

3. Εφαρμογή

Στο κεφάλαιο αυτό θα παρουσιάσουμε την εφαρμογή μας η οποία περιλαμβάνει μια αρχική σελίδα (εικόνα 3). Είναι μια flash σελίδα με την οποία ο χρήστης θα έρχεται σε επαφή για να επιλέξει είτε έναν από τους τρεις αλγόριθμους είτε να ξαναδεί παλιότερα αποτελέσματα. Εάν ο χρήστης επιλέξει έναν από τους τρεις αλγορίθμους θα εμφανιστεί στην οθόνη του χρηστή η σελίδα εισαγωγής δεδομένων και παραμέτρων (εικόνα 4) όπου ο χρήστης θα ορίσει τον αριθμό των χαρακτηριστικών που θέλει, θα δώσει το αρχείο που περιέχει τον πίνακα με τα χαρακτηριστικά, θα ορίσει αν ο πίνακας είναι χωρισμένος σε χωρία η όχι και ανάλογα αν το διάνυσμα με τις κατηγορίες είναι μαζί με τον πίνακα χαρακτηριστικών θα επιλέξει την ανάλογη φόρμα. Επιλέγοντας done, οι ρυθμίσεις και τα αρχεία θα αποσταλούν στον server όπως εξηγούμε στην παράγραφο *3.4. Μεταφορά αρχείου από τον χρήστη στον Server*). Στην συνέχεια γίνεται η εκτέλεση του αλγορίθμου επιλογής χαρακτηριστικών στο Matlab όπως εξηγούμε στην παράγραφο *3.5. Εκτέλεση αρχείων Matlab μέσω της exec*). Δημιουργείται μοναδικός πίνακας και αφού περιμένει η php την ολοκλήρωση της εκτέλεσης του αλγορίθμου επιλογής χαρακτηριστικών από το Matlab γίνεται προβολή του όπως βλέπουμε στην παράγραφο *3.6. Δημιουργία μοναδικού πίνακα στην MySql και προβολή του*)

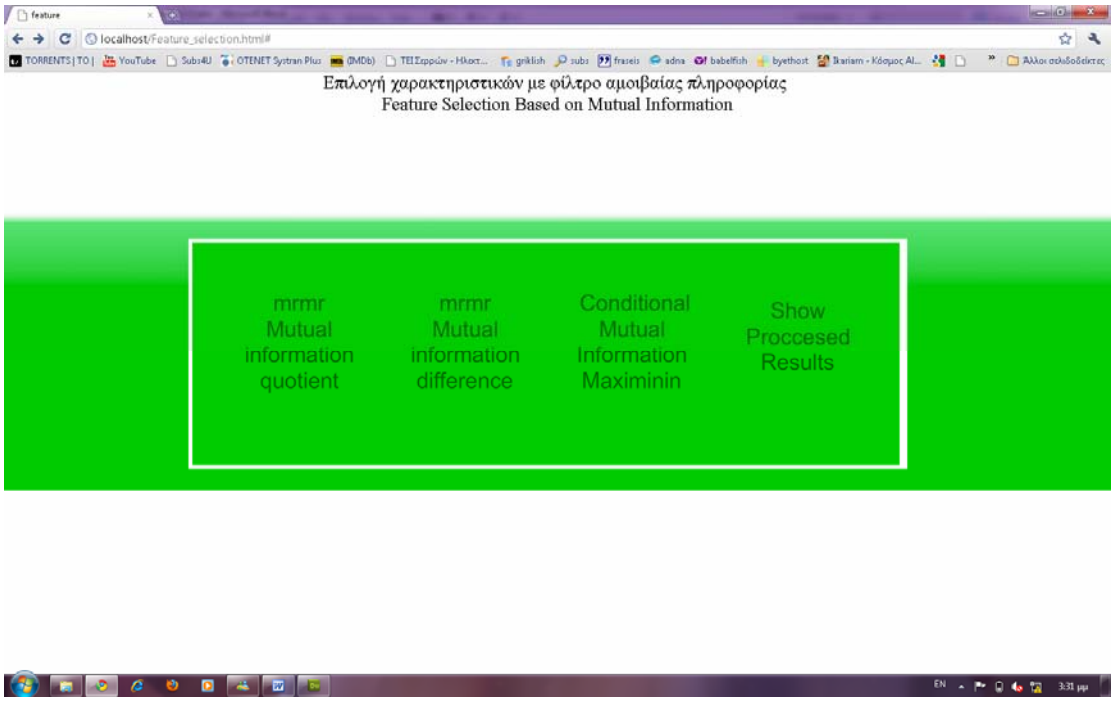
Αν ο χρήστης επιλέξει τον αλγόριθμο επιλογής χαρακτηριστικών με υπό συνθήκη αμοιβαίας πληροφορίας θα γίνει η αναζήτηση του πρώτου βέλτιστου χαρακτηριστικού με τον απλό τύπο της αμοιβαία πληροφορίας τον οποίον τον αναλύουμε στην παράγραφο *3.8. Υλοποίηση στο Matlab του αλγορίθμου επιλογής χαρακτηριστικών με το φίλτρο της αμοιβαίας πληροφορίας*). Αργότερα θα γίνει χρήση του αλγορίθμου υπό συνθήκη αμοιβαίας πληροφορίας σε κάθε επανάληψη όπως μπορούμε να δούμε στην παράγραφο *3.10. Αλγόριθμος υπό συνθήκη αμοιβαία πληροφορία (Conditional Mutual Information) στο Matlab*

Αν ο χρήστης επιλέξει τον αλγόριθμο επιλογής χαρακτηριστικών με αμοιβαία πληροφορία μέγιστης σχετικότητας ελάχιστου Πλεονασμού θα γίνει πάλι η αναζήτηση του πρώτου βέλτιστου χαρακτηριστικού με τον απλό τύπο της αμοιβαία πληροφορίας τον οποίον τον αναλύουμε στην παράγραφο *3.8. Υλοποίηση στο Matlab του αλγορίθμου επιλογής χαρακτηριστικών με το φίλτρο της αμοιβαίας πληροφορίας*) στη συνέχεια θα γίνει χρήση του αλγορίθμου μέγιστης σχετικότητας ελάχιστου πλεονασμόυ σε κάθε επανάληψη όπως μπορούμε να δούμε στην παράγραφο *3.11. Αλγόριθμος Μέγιστη Σχετικότητα Ελάχιστος Πλεονασμός (Max Relevance Min Redundancy)*).

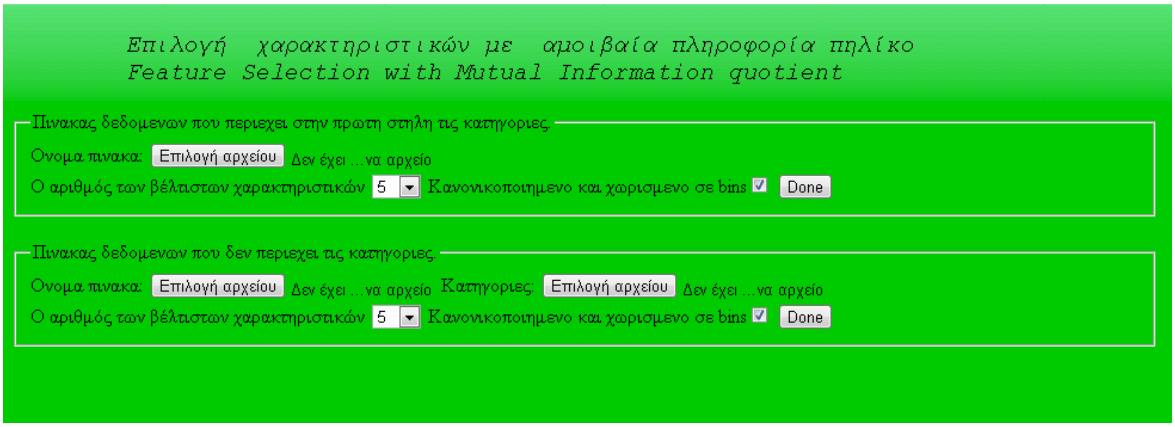
Όταν ο χρηστής επιλέξει να δει παλιότερα αποτελέσματα πληκτρολογεί τον κωδικό του πίνακα (αν ο κωδικός είναι σωστός και ο πίνακας υπάρχει) τότε γίνεται εμφάνιση του παλαιού πίνακα με τα επιλεγμένα χαρακτηριστικά όπως βλέπουμε στην παράγραφο *3.7. Εμφάνιση παλιότερων αποτελεσμάτων*).

Σε περίπτωση που τα δεδομένα του χρήστη δεν είναι κανονικοποιημένα και χωρισμένα σε χωρία (bins) ο χρήστης μπορεί να επιλέξει να κάνει κανονικοποίηση, όπως βλέπουμε στην παράγραφο *3.12. Αλγόριθμος κανονικοποίησης των τιμών σε έναν πίνακα*) και αργότερα να βάλει τις τιμές σε χωρία (bins) παράγραφος *3.13. Αλγόριθμος τοποθέτησης των κανονικοποιημένων τιμών σε χωρία (bins)*)

Εικόνα 3: Η αρχική σελίδα της εφαρμογής μας.



Εικόνα 4: Σελίδα εισαγωγής παραμέτρων και δεδομένων. Είναι παρόμοια και στους τρεις αλγορίθμους.



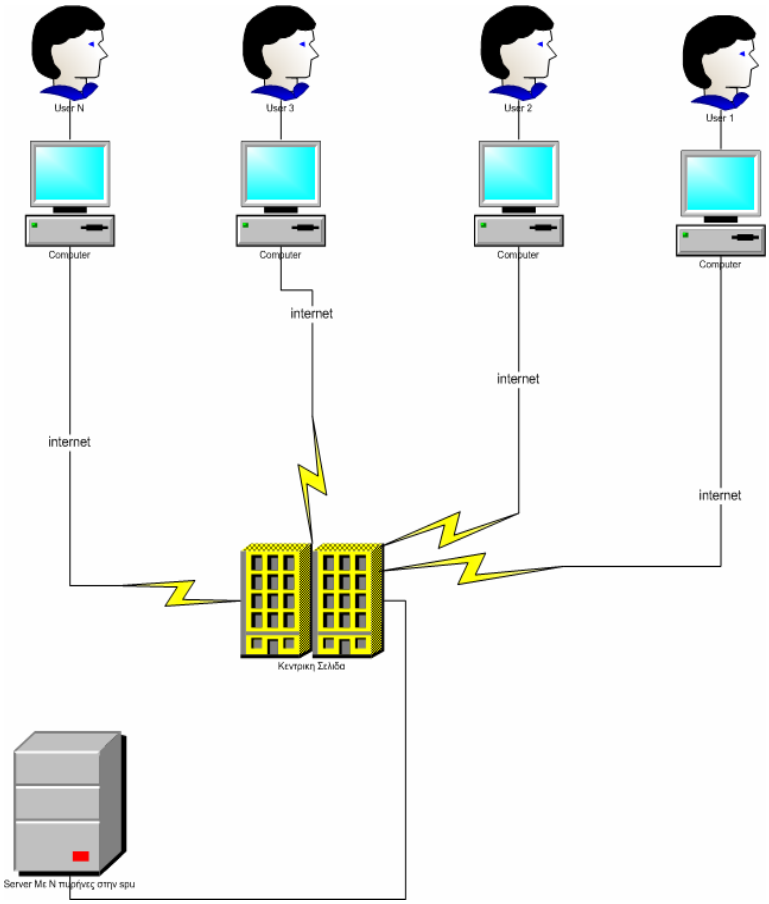
3.1. Σχεδιάγραμμα

Στο σχεδιάγραμμα της εικόνας 5 βλέπουμε πως η εφαρμογή μας θα δίνει την δυνατότητα να επισκεφτούν την σελίδα μας και να εκτελέσουν πολλοί χρήστες ταυτόχρονα Matlab. Οι χρήστες είναι τόσοι όσοι είναι και οι πυρήνες της CPU του server. Ο server αφιερώνει κάθε φορά έναν πυρήνα σε κάθε εκτέλεση Matlab. Όταν πλέον τελειώσει την εκτέλεση του m file τερματίζει το Matlab έτσι ώστε να μπορούν και άλλοι χρήστες να χρησιμοποιήσουν τον server.

Αν σε περίπτωση που ένας χρήστης επιθυμεί να εκτελέσει κάποιον αλγόριθμο στον server και αυτός έχει ελεύθερο πυρήνα για να το κάνει, τότε όλα θα κυλίσουν ομαλά. Αλλιώς ο χρηστής θα πρέπει να αποπειραθεί κάποια άλλη χρονική στιγμή. Η αίτηση του θα μπει στην ουρά αναμονής από τον επεξεργαστή και όταν βρεθεί ελεύθερος πυρήνας θα γίνει εκτέλεση του αλγορίθμου. Τα αποτελέσματα δεν θα είναι επιθυμητά σε περίπτωση μεγάλης αναμονής γιατί η php περιμένει συγκεκριμένα δευτερόλεπτα την Matlab. Αν αυτή αργήσει , η php θα συμπεραίνει πως υπάρχει κάποιο πρόβλημα και θα εμφανίσει αντί των αποτελεσμάτων ένα μήνυμα αποτυχίας .

Η περίπτωση της αποτυχίας (αν τα ζητούμενα και τα δοθέντα δεν είναι λάθος) θα είναι σπάνια γιατί η συνολική εκτέλεση του προγράμματος θα διαρκεί μερικά δευτερόλεπτα. Έτσι πχ αν ένας server έχει 8 πυρήνες θα είναι αρκετά σπάνιο φαινόμενο η αποτυχία της επιλογής βέλτιστων χαρακτηριστικών εφόσον μπορούν να εξυπηρετηθούν 8 χρήστες ταυτόχρονα.

Εικόνα 5: Σχεδιάγραμμα Της εφαρμογής. Αναπαράσταση επικοινωνίας χρήστη με server.



3.2. Παράδειγμα

3.2.1. Πρόβλημα

Στο κεφάλαιο αυτό θα παρουσιάσουμε ένα μικρό παράδειγμα επιλογής χαρακτηριστικών και για να είναι ορατά τα αποτελέσματα χρησιμοποιούμε έναν μικρό πίνακα δεδομένων 10 x 11 στην αρχή χωρισμένο σε χωρία (bins) πίνακας 1.α και στην συνεχεία τον πίνακα 3.α ο οποίος δεν είναι κανονικοποιημένος και δεν είναι χωρισμένος σε χωρία (bins) .

Σκοπός του παραδείγματος είναι να εκτελέσουμε τους αλγορίθμους επιλογής χαρακτηριστικών και να εξετάσουμε τα αποτελέσματα κρίνοντας αν είναι αποτελεσματικοί οι αλγόριθμοι η όχι.

3.2.2. Δεδομένα

Τα δεδομένα μας είναι σε έναν μικρό δοκιμαστικό πίνακα δεδομένων 10 x 11 που περιέχει στην πρώτη του στήλη τις κατηγορίες (οι πρώτες πέντε είναι κατηγορία 1 και οι υπόλοιπες είναι κατηγορία 2) και στις υπόλοιπες 10 στήλες περιέχει διανύσματα με τιμές από τις

οποίες οι 1,2,5,6,8,10 έχουν χρήσιμες πληροφορίες όπως άλλωστε φαίνεται στον πινάκα 1.α .

3.2.3. Λύση

Εκτελούμε τους αλγορίθμους με τον πίνακα 1.α Όπως ήταν αναμενόμενο ο αλγόριθμος επέλεξε τα διανύσματα F2,F8,F6,F5,F1 σαν βέλτιστα και δοκιμάζοντας και τους υπόλοιπους δυο αλγόριθμους θα παρατηρήσουμε πως έχουμε πάλι τα ίδια 5 καλύτερα διανύσματα όπως φαίνεται στους παρακάτω πίνακες .

Πίνακας 1: α)Μικρός δοκιμαστικός Πινάκας δεδομένων χωρισμένος σε χωριά (bins). β)Τα αποτελέσματα με την επιλογή χαρακτηριστικών με τον αλγόριθμο Αμοιβαία πληροφορία πηλίκο (MIQ).

α)

Κατηγορίες	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10
1	1	1	0	-1	5	6	-2	1	-3	1
1	1	2	0	-1	1	6	-2	1	-3	10
1	1	2	0	-1	1	6	-2	8	-3	10
1	1	2	0	-1	5	6	-2	8	-3	10
1	1	1	0	-1	5	6	-2	8	-3	1
2	1	5	0	-1	5	5	-2	8	-3	10
2	5	5	0	-1	5	6	-2	8	-3	5
2	1	2	0	-1	5	6	-2	8	-3	10
2	1	2	0	-1	5	6	-2	8	-3	5
2	5	2	0	-1	5	5	-2	8	-3	10

β)

Minimum Redundancy – Maximum Relevance for Categorical (Discrete) Variables With ***Mutual information quotient***

Results

Order	Selected Features	Score
1	2	0.4
2	8	1.8184
3	6	1.79247
4	5	1.88774
5	1	2.04346

Results code :1284576828

Πίνακας 2: α) Αμοιβαία πληροφορία διαφορά (MID). β) υπό συνθήκη Αμοιβαία πληροφορία (CMI).

α)

Minimum Redundancy - Maximum Relevance for Categorical (Discrete) Variables With ***Mutual information difference***

Results

Order	Selected Features	Score
1	5	0.0831286
2	1	0.0908938
3	2	0.0577219
4	6	0.0542453
5	8	0.0509494

Results code :1284653351

β)

Feature Selection with ***Conditional Mutual Information***

Results

Order	Selected Features	Score
1	2	0.4
2	5	0.275489
3	8	0.2
4	1	0.114525
5	6	0.114525

Results code :1284654395

Στο παραπάνω παράδειγμα βέβαια είχαμε πίνακα με διακριτές τιμές και χωρισμένο σε μικρό αριθμό χωρία (bins). Υπάρχει περίπτωση ο πίνακας μας να μην είναι χωρισμένος σε χωρία (bins) όπως βλέπουμε τον παρακάτω πίνακα που είναι παρόμοιος με τον προηγούμενο απλά αυτός έχει μεγάλες και μικρές τιμές.

Πίνακας 3: α) Μικρός δοκιμαστικός Πινάκας δεδομένων μη κανονικοποιημένος και δεν είναι χωρισμένος σε χωριά. β)Κανονικοποιηση του πίνακα 3.α και τοποθέτηση των τιμών σε 10 χωρία (bins). Δηλαδή από τιμές 1 μέχρι 100 τοποθέτιουνται στο χωρίο 1, από 101 μέχρι 200 τοποθέτιουνται στο χωρίο 2, κτλ.

α)

Κατηγορίες	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10
1	100	100	0	-1	500	600	-2	100	-3	100
1	100	200	0	-1	100	600	-2	100	-3	1000
1	100	200	0	-1	100	600	-2	800	-3	1000
1	100	200	0	-1	500	600	-2	800	-3	1000
1	100	100	0	-1	500	600	-2	800	-3	100
2	100	500	0	-1	500	500	-2	800	-3	1000
2	500	500	0	-1	500	600	-2	800	-3	500
2	100	200	0	-1	500	600	-2	800	-3	1000
2	100	200	0	-1	500	600	-2	800	-3	500
2	500	200	0	-1	500	500	-2	800	-3	10

β)

Κατηγορίες	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10
1	1	1	0	-1	11	11	-2	1	-3	1
1	1	3	0	-1	1	11	-2	1	-3	11
1	1	3	0	-1	1	11	-2	11	-3	11
1	1	3	0	-1	11	11	-2	11	-3	11
1	1	1	0	-1	11	11	-2	11	-3	1
2	1	11	0	-1	11	1	-2	11	-3	11
2	11	11	0	-1	11	11	-2	11	-3	5
2	1	3	0	-1	11	11	-2	11	-3	11
2	1	3	0	-1	11	11	-2	11	-3	5
2	11	3	0	-1	11	1	-2	11	-3	1

Απλά αυτή την φορά κανονικοποιηούμε πρώτα τον πίνακα 3.α και μετά τον χωρίζουμε σε χωρία (bins) στον πίνακα 3.β και τελικά τρέχουμε τον αλγόριθμο.

Πίνακας 4: α)Αποτελέσματα από την εκτέλεση του πρώτου και δευτέρου αλγορίθμου με δεδομένα από τον πίνακα 3.β β) Αποτελέσματα από την εκτέλεση του τρίτου αλγορίθμου με δεδομένα από τον πίνακα 3.β .

α)

Minimum Redundancy - Maximum Relevance for Categorical (Discrete) Variables With ***Mutual information difference***

Results

Order	Selected Features	Score
1	5	0.0831286
2	1	0.0908938
3	2	0.0577219
4	6	0.0542453
5	8	0.0509494

Results code :1284659813

β)

Feature Selection with ***Conditional Mutual Information***

Results

Order	Selected Features	Score
1	2	0.4
2	5	0.275489
3	10	0.21257
4	8	0.2
5	1	0.114525

Results code :1284661424

Όπως βλέπαμε μετά την εκτέλεση του πρώτου και δευτέρου αλγορίθμου mRMR στον πίνακα 4.α και τρίτου αλγόριθμου CMI στον πίνακα 4.β έχουμε πάλι τα ίδια επιθυμητά αποτελέσματα με πριν.

3.3. PHP

Στην εργασία μας θα χρησιμοποιήσουμε την php για να στείλουμε ένα αρχείο στον server , ώστε να εκτελέσουμε τους αλγόριθμους επιλογής χαρακτηριστικών που είναι γραμμένοι σε γλώσσα Matlab και επιλέγουν χαρακτηριστικά με την αμοιβαία πληροφορία. δημιουργούμε κάθε φορά έναν μοναδικό πίνακα στην βάση μας όπου θα αποθηκευτούν τα αποτελέσματα μετά την εκτέλεση των αλγόριθμων. Το μοναδικό όνομα του πινάκα που δημιουργούμε κάθε φορά είναι ουσιαστικά η ώρα του συστήματος εκείνη την στιγμή και χρησιμοποιούμε την συνάρτηση Time(). Τέλος προβάλουμε τα αποτελέσματα μας μέσω μιας ιστοσελίδας.

Η PHP είναι μια γλώσσα προγραμματισμού για τη δημιουργία σελίδων web με δυναμικό περιεχόμενο. Μια σελίδα PHP περνά από επεξεργασία από ένα συμβατό διακομιστή του Παγκόσμιου Ιστού (π.χ. Apache), ώστε να παραχθεί σε πραγματικό χρόνο το τελικό περιεχόμενο, που θα σταλεί στο πρόγραμμα περιήγησης των επισκεπτών σε μορφή κώδικα HTML.

3.4. Μεταφορά αρχείου από τον χρήστη στον Server

Για να ανεβάσουμε ένα αρχείο στον server βάζουμε κάποια συγκεκριμένα components στην html form της αρχικής σελίδας τα οποία είναι:

```
<form action="upload_filecon.php" method="post" enctype="multipart/form-data">
<label for="file">TableFilename:</label>
<input type="file" name="file" id="file" />
<input type="submit" name="submit1" value="Done"/>
</form>
```

Το <label for=”file”> είναι ένα απλό label που δείχνει το όνομα και κατάληξη του αρχείου όταν το φορτώσουμε με το <input type=”file”> και όταν φορτώσουμε αρχείο στο <input type=”file”> το οποίο θα πρέπει να είναι χωρισμένο σε στήλες και γραμμές, όπου οι γραμμές είναι το σύνολο των δειγμάτων και οι στήλες είναι τα χαρακτηριστικά. Πατώντας το button submit αποστέλλετε το αρχείο στον server. Αντίστοιχα στην upload_filecon.php θα έχουμε τον εξής κώδικα:

```
<?php
if ($_FILES["file"]["size"] <1000000)// Το μέγεθος το ορίζουμε εμείς
        {
```

```
        if ($_FILES["file"]["error"] > 0)
                {
                echo "Return Code: " . $_FILES["file"]["error"] . "<br />";
                }
        else
                {
                $_FILES["file"]["name"]=$fsTime.".dat";
                move_uploaded_file($_FILES["file"]["tmp_name"],"".$_FILES["file"]
                ["name"]);
                }
        }
else
        {
        echo "Invalid file";
        }?>
```

Η *$_FILES["file"]["size"]* περιέχει το μέγεθος του αρχείου που έχουμε ανεβάσει. Το *$_FILES["file"]["error"]* περιέχει τον κωδικό σφάλματος αν υπάρχει, αν δεν υπάρχει είναι 0. Στο *$_FILES["file"]["name"]* βάζουμε το τελικό όνομα του αρχείου που ανεβάσαμε στον server. Αν δεν θέλουμε να αποθηκευτεί το αρχείο στον ίδιο φάκελο στον server μαζί με το όνομα δίνουμε και το path. Τέλος με την εντολή *move_uploaded_file* μεταφέρουμε το αρχείο από το προσωρινό φάκελο στο path που του δώσαμε, αν δώσαμε, αλλιώς παίρνει απλά το όνομα που δώσαμε και αποθηκεύετε στον ίδιο φάκελο με το php αρχείο. Στην εργασία μας το όνομα που δίνουμε στο αρχείο είναι το μοναδικό όνομα που δίνουμε και στον πινάκα στην βάση μας το οποίο είναι η ώρα του συστήματος.

3.5. Εκτέλεση αρχείων Matlab μέσω της exec

Στην php η exec μας δίνει την δυνατότητα να εκτελούμε αρχεία Matlab. Στην εργασία μας καλούμε το αλγόριθμο που χρειάζεται κάθε φορά με την εξής σύνταξη.

```
echo exec(' matlab -r mrmr_miq('.$fsTime.','.$_POST["fetsel"].')');
```

Ουσιαστικά τρέχουμε την exec και μέσα δίνουμε παραμέτρους Matlab, σημαίνει πως το αρχείο που θα τρέξουμε είναι αρχείο Matlab και mrmr_miq που είναι το όνομα του αρχείου Matlab που καλούμε και μέσα σε αυτήν περνάμε άλλες δυο παραμέτρους το

πρώτο είναι το όνομα του μοναδικού πίνακα που θα δημιουργήσουμε στην βάση μας και το δεύτερο είναι ο αριθμός των βέλτιστων χαρακτηριστικών που θέλουμε να μας επιλέξει ο αλγόριθμος.

3.6. Δημιουργία μοναδικού πίνακα στην MySql και προβολή του

Για να δοθεί η δυνατότητα να εκτελέσουν τον ίδιο αλγόριθμο δυο η και παραπάνω χρήστες ταυτόχρονα δημιουργούμε για τον κάθε χρήστη και για κάθε εκτέλεση του προγράμματος μοναδικό πίνακα παίρνοντας από το ρόλοι του συστήματος την ώρα σε μορφή milliseconds και θέτοντας τον σαν όνομα για να εξασφαλίσουμε αποκλειστικότητα κάθε φορά, για να αποφύγουμε την πιθανή σύγχυση των πινάκων και λανθασμένη επιλογή χαρακτηριστικών σε περίπτωση χρήσης της εφαρμογής από πολλούς χρήστες. Αυτό το πετυχαίνουμε με τον παρακάτω κώδικα :

```
$fsTime=time();
$con=mysql_connect("localhost","root","");
mysql_select_db("feature_selection",$con);
$sql = "CREATE TABLE `feature_selection`.`".$fsTime."` (`selFeatures` VARCHAR( 4 ) NOT NULL,`Scor` float NOT NULL) ENGINE = MYISAM" ;
mysql_query($sql,$con);
```

Αφού δημιουργήσουμε μια σύνδεση με τον MySql συντάσσουμε και το ανάλογο query με το όνομα της βάσης και το όνομα του πίνακα εκτελούμε το query. Μετά την εκτέλεση του αλγορίθμου η Matlab αποθηκεύει τα αποτελέσματα στον πίνακα που έχουμε δημιουργήσει είδη για να τα προβάλει στον αντίστοιχο χρήστη που περιμένει να τα δει. Μετά την δημιουργία του πίνακα δίνουμε εντολή στην php να περιμένει το Matlab να εκτελέσει και να αποθηκεύσει τα αποτελέσματα και να τα προβάλουμε στην συνέχεια. Αυτό πετυχαίνεται με την εντολή sleep() και παίρνει σαν παράμετρο τα δευτερόλεπτα που πρέπει να περιμένει η php για να συνεχίσει την εκτέλεση των παρακάτω εντολών.

```
if ($_POST["fetsel"]==5)
                {
                        sleep(25);
                }
 else if($_POST["fetsel"]==10)
                {
```

```
                    sleep(59);
            }
```

Επίσης γίνεται έλεγχος πόσα βέλτιστα χαρακτηριστικά πρέπει να επιλέξει ο αλγόριθμος και όσο πιο πολλά βέλτιστα τόσο περισσότερο πρέπει να περιμένει η php επειδή ο αλγόριθμος αργεί περισσότερο όταν μεγαλώνει ο αριθμός των βέλτιστων γιατί έχει να εκτελέσει περισσότερες πράξεις. Και τέλος γίνεται η προβολή των αποτελεσμάτων όπως θα δούμε παρακάτω.

```
<?php
 $con=mysql_connect("localhost","root","");
 mysql_select_db("feature_selection",$con);
$result=mysql_query("SELECT * FROM `feature_selection`.`".$fsTime."`");
 $row=mysql_fetch_array($result);
 if($row)
 {
        $order=1;
        ?>
       <table  width="100%" border bgcolor="#00CC00">
          <tr>
            <th>Order
            <th>Selected Features
            <th>Score
          </tr>
    <?php
        while($row)
                {
        echo "<tr><td>".$order."</td>";
                echo "<td>".$row['selFeatures']."</td>";
                echo "<td>".$row['Scor']."</td>";
                echo "</tr>";
        $row = mysql_fetch_array($result);
                $order=$order+1;                }
        ?>
```

```
        </table>
 <?php
 echo "Results code :".$fsTime; }
else
        echo "Erorr No Selected features"; ?>
```

Παραπάνω αφού δημιουργήσουμε μια νέα σύνδεση επιλεγούμε με ποια ΒΔ θα συνδεθούμε και με το query παίρνουμε τα βέλτιστα χαρακτηριστικά και το σκορ της σχετικότητας από τον μοναδικό πίνακα μας. Παρακάτω εκτελούμε την εντολή $row=mysql_fetch_array($result); που ουσιαστικά παίρνει κάθε φορά που το εκτελούμε την πρώτη γραμμή από τον πίνακα μας και αν υπάρχει πρώτη γραμμή δηλαδή αν δεν είναι άδειος ο πίνακας μας μες στο loop while($row) και μέσω html κώδικα δημιουργούμε τον πίνακα που θα δει ο χρήστης (εικόνα 5). Το loop σταματά όταν κάνουμε mysql_fetch_array και μας επιστρέψει μηδενική τιμή δηλαδή όταν φτάσει στο τέλος ο πίνακας. Δείχνουμε επίσης και το όνομα του πίνακα μας δίπλα από την φράση *Results code* για να μπορέσει ο χρήστης να δει τα αποτελέσματα όσες φορές θέλει και όποτε θέλει δίνοντας απλά το όνομα του πίνακα που περιέχει επιλεγμένα χαρακτηριστικά από παλιότερους υπολογισμούς.

Πίνακας 5: Παράδειγμα Πίνακα με αποτελέσματα ύστερα από εκτέλεση κάποιου αλγορίθμου.

Minimum Redundancy - Maximum Relevance for Categorical (Discrete) Variables With ***Mutual information difference***

Results

Order	Selected Features	Score
1	23	0.773383
2	126	0.555003
3	244	0.566919
4	133	0.533325
5	243	0.53842

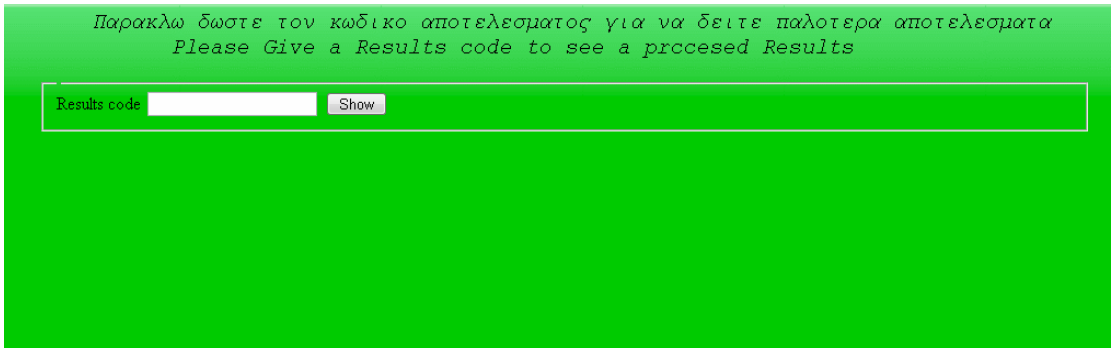
Results code :1284760422

3.7. Εμφάνιση παλιότερων αποτελεσμάτων

Στην αρχική σελίδα όταν επιλέξουμε εμφάνιση παλιότερων αποτελεσμάτων (πίνακας 6.α) και δώσουμε το όνομα του πινάκα που θέλουμε να δούμε γίνεται η προβολή των βέλτιστων χαρακτηριστικών ακριβώς με τον ίδιο τρόπο που περιγράψαμε στην προηγούμενη παράγραφο (πίνακας 6.β).

Πίνακας 6: α) Φόρμα όπου δίνουμε τον κωδικό παλιότερων αποτελεσμάτων. β)Εμφάνιση παλιότερων αποτελεσμάτων.

α)



β)

Order	Selected Features	Score
1	23	0.773383
2	164	0.691109
3	244	0.682766
4	19	0.619789
5	126	0.605096

3.8. Υλοποίηση στο Matlab του αλγορίθμου επιλογής χαρακτηριστικών με το φίλτρο της αμοιβαίας πληροφορίας

Παρακάτω καλούμε μέσα σε ένα loop την συνάρτηση mutualinfo που υπολογίζει την αμοιβαία πληροφορία και εκτελείται τόσες φορές όσα είναι τα χαρακτηριστικά γνωρίσματα στο d. Τα αποτελέσματα αποθηκεύονται στον πίνακα argg.

```
function [argg]=mutualin_arg(d,c)
{
 for i=1:size(d,2),
   argg(i)=mutualinfo(d(:,i),c); %#ok<AGROW>
end;
 return;
}
```

Παρακάτω στην συνάρτηση mutualinfo γίνεται η εφαρμογή του τύπου της αμοιβαίας πληροφορίας όπως τον περιγράψαμε στην παράγραφο

2.2.5. Αμοιβαία Πληροφορία (Mutual Information) και είναι η εξίσωση 10.

```
function mi=mutualinfo(fi,c)
{
atrisma=0;
for j=1:size(c,1),
    p1=pithand(j,fi,c);     // p(x,y)
    p2=pithan(j,fi);        //p(x)
    p3=pithan(j,c);          //p(y)
     if p1~=0
       atrisma=( (log(p1/(p2*p3) ))/log(2) )+atrisma;
   end;
   end;
mi=atrisma/(size(fi,1));    // μέσος ορος αμοιβαίων πληροφοριών.
  return;
}
```

$$I(X;Y) = \sum_{y \in Y} \sum_{x \in X} p(x,y) \log \left(\frac{p(x,y)}{p_1(x)\, p_2(y)} \right),$$ **εξ.10**

Με λίγα λόγια μηδενίζουμε την μεταβλητή atrisma η οποία περιέχει το άθροισμα των αμοιβαίων πληροφοριών όλων τον στοιχείων των δυο διανυσμάτων. Υπολογίζουμε τις πιθανότητες των ζευγών fi(j) και c(j) για όλα τα στοιχειά των δυο διανυσμάτων κατά ζεύγη και αφού υλοποιήσουμε τον τύπο αθροίζουμε το αποτέλεσμα με τα υπόλοιπα αθροίσματα για να πάρουμε τον μέσο όρο των αθροισμάτων. Η συνάρτηση pithand(j,fi,c) υπολογίζει την πιθανότητα p(x,y) η συνάρτηση pithan(j,fi), pithan(j,c) υπολογίζουν την πιθανότητα p(x),p(y) αντίστοιχα.

```
function pit=pithand(i,x,y)
{
pit(1)=0;
 for t=1:size(x,1),
   if (x(t)==x(i)&&y(t)==y(i)),
      pit(1)=pit(1)+1;
   end;
 end;
```

```
pit(1)=pit(1)/size(x,1);
return;
 }
```

Η παραπάνω συνάρτηση *pithand* υπολογίζει την p(x,y). Ουσιαστικά μετρά την συχνότητα εμφάνισης των ζευγών x και y , δηλαδή αφού μετρήσουμε πόσες φορές εμφανίστηκε το ζεύγος διαιρούμε με τον αριθμό των δειγμάτων .

```
function pit=pithan(i,x)
{
pit(1)=0;
 for t=1:size(x,1),
   if (x(t)==x(i)),
      pit(1)=pit(1)+1;
   end;
end;
pit(1)=pit(1)/size(x,1);
return;
}
```

Ομοίως με την *pithand* και η *pithan* υπολογίζει την συχνότητα εμφάνισης του x, δηλαδή αφού μετρήσουμε πόσες φορές εμφανίστηκε το x διαιρούμε με τον αριθμό των δειγμάτων .

3.9. Υλοποίηση της υπό συνθήκης αμοιβαίας πληροφορίας στο Matlab (Conditional Mutual Information-CMI)

Παρακάτω το loop1 θα προσπελαστεί όσες φορές είναι ο αριθμός των χαρακτηριστικών συνολικά για να υπολογιστεί το σκορ σχετικότητας για όλα τα χαρακτηριστικά γνωρίσματα με το διάνυσμα κατηγοριών. Το loop2 θα προσπελαστεί όσες φόρες είναι ο αριθμός τον βέλτιστων χαρακτηριστικών που έχουν επιλεγεί είδη έτσι ώστε παρακάτω στο τέλος του loop2 να επιλεγεί το μικρότερο για να επιτύχουμε αργότερα στην επιλογή του μέγιστου σκορ να μην έχουμε ίδια χαρακτηριστικά γνωρίσματα ή όμοια.

```
function [argu]=minARG(d,c,Fid)
{
   for i=1:size(d,2),   // loop1
      for j=1:size(Fid,2),  //loop2
```

```
        r=d(:,Fid(j));
        mic(j)=mutualinfoc(c,d(:,i),r);
      end;
      argu(i)=min(mic);//επιλογή πάντα του μικρότερου.
  end;
 return;
}
```

Όπως έχουμε δει ήδη στην παράγραφο 2.3. Υπό Συνθήκη Αμοιβαία Πληροφορία (Conditional Mutual Information-CMI) Η υλοποίηση του υπό συνθήκη αμοιβαία πληροφορία εφαρμόζουμε την εξίσωση 14.

```
function  mic=mutualinfoc(c,di,r)
{
a1=jentropy(c,r);
a2=jentropy(di,r);
a3=trjentropy(c,di,r);
a4=entropy(r);
mic=a1+a2-a3-a4;
 return;
}
```

$$I(X;Y \mid Z) = H(X,Z) + H(Y,Z) - H(X,Y,Z) - H(Z) \qquad \text{εξ. 14}$$

Ο τύπος υπολογίζεται με το άθροισμα δυο ορών οι οποίοι είναι από κοινού εντροπίες (joint entropy) των ζευγών των μεταβλητών X και Z στον πρώτο όρο, Y και Z αντίστοιχα στον δεύτερο όρο. Και με την διαφορά δυο ορών από τους οποίους ο πρώτος είναι μια από κοινού εντροπία (joint entropy) με τρεις μεταβλητές X,Y και Z, και ο τελευταίος ορος είναι η εντροπία (entropy) της μεταβλητής Z.

```
function a=jentropy(x,y)
{
atrisma=0;
for i=1:size(x,1),
```

```
    p=pithand(i,x,y);
    if p~=0,
        atrisma=( ( ( log(p) )/log(2)) )+atrisma;
    end;
   end;
a=-atrisma/(size(y,1));
return;
}
 function a=trjentropy(x,y,z)
{
atrisma=0;
 for i=1:size(x,1),
   p=pithant(i,x,y,z);
   if p~=0,
        atrisma=( (( log(p) )/log(2)) )+atrisma;
    end;
end;
a=-atrisma/(size(y,1));
return;
}
 function a=entropy(x)
{
atrisma=0;
for i=1:size(x,1),
    p=pithan(i,x);
    if p~=0,
        atrisma=( (log(p)/log(2)) )+atrisma;
    end;
 end;
a=-atrisma/size(x,1);
return;
}
```

Οι παραπάνω εντροπίες υπολογίζονται με τους τύπους που προαναφέραμε στις παραγράφους 5.2 και 5.3 οι οποίοι περιέχουν πιθανότητες που είδαμε πιο πάνω πως υπολογίζονται.

3.10. Αλγόριθμος υπό συνθήκη αμοιβαία πληροφορία (Conditional Mutual Information) στο Matlab

Ο αλγόριθμος αυτός υλοποιείτε σε τρία στάδια .

1. Εύρεση πρώτου Βέλτιστου χαρακτηριστικού I(Fi,C).
2. Εύρεση επόμενων βέλτιστων χαρακτηριστικών.
3. Αποθήκευση των αποτελεσμάτων στην βάση μας στον αντίστοιχο μοναδικό πίνακα που δημιουργήθηκε στην έναρξη .

3.10.1. Εύρεση πρώτου Βέλτιστου χαρακτηριστικού I(Fi,C)

Παρακάτω στην αρχή καλούμε απλά την συνάρτηση mutualin_arg(d,c) η οποία παίρνει σαν παράμετρο α) τον πίνακα d που περιέχει όλα τα διανύσματα και β) το διάνυσμα κατηγοριών c και μας επιστρέφει έναν πίνακα με σκορ σχετικότητας όλων των διανυσμάτων. Στην συνέχεια αφού επιλεγεί το μέγιστο σκορ με την εντολή max(arg) και αποθηκευτεί στην μεταβλητή Fi γίνεται αναζήτηση της θέσης του πρώτου βέλτιστου χαρακτηριστικού που έχει επιλεγεί. Η αναζήτηση του γίνεται μέσα στο loop for και στον έλεγχο if βλέπουμε αν το τρέχον arg(i) είναι το μέγιστο και αν το done είναι false. Η μεταβλητή done υπάρχει απλώς και μόνο για να επιλεγεί το πρώτο max που θα βρεθεί. Τα αλλά αν υπάρχουν θα απορριφθούν αυτόματα όταν θα επιλεγούμε μόνο τα max από τα min.

```
head=1:size(d,2);
arg=mutualin_arg(d,c); %arg=I(Fi;c),i=1..u
 Fi=max(arg);
done=false;
for i=1:size(arg,2),
        if arg(i)==max(arg)&&done~=true,
                best_Feature=head(i); // η θέση του μέγιστου
                Fid=[i];  // η Fid θα περιέχει τα βέλτιστα επιλεγμένα χαρακτηριστικά.
                done=true;
```

```
        end;
end;
F=Fi;
```

Παρακάτω καλούμε την συνάρτηση mutualinfo() για κάθε διάνυσμα ξεχωριστά και επιστρέφει έναν πίνακα όλα τα σκορ σχετικότητας. Παρακάτω βλέπουμε την υλοποίηση στον τύπο αμοιβαίας πληροφορίας όπως προαναφέρει είδη στο ανάλογο κεφαλαίο.

```
function [argg]=mutualin_arg(d,c)
{
argg=0;
for i=1:size(d,2),
        argg(i)=mutualinfo(d(:,i),c);
end;
return;
}
 function mi=mutualinfo(fi,c)
{
atrisma=0;
for j=1:size(c,1),
        p1=pithand(j,fi,c);
        p2=pithan(j,fi);
        p3=pithan(j,c);
        if p1~=0
                atrisma=( (log(p1/(p2*p3) ))/log(2) )+atrisma;
        end;
end;
mi=atrisma/(size(fi,1));
return;
}
```

3.10.2. Εύρεση των χαρακτηριστικών με τα μεγαλύτερα σκορ

Παραπάνω απλά εντοπίζουμε την θέση του μέγιστου σκορ αποθηκεύουμε την τιμή του, την θέση του και βάζουμε στην θέση του 0. Επαναλαμβάνουμε τα παραπάνω βήματα μέχρι να βρούμε τον επιθυμητό αριθμό των μέγιστων σκορ.

```
parg=arg
done=false;
find=1;
i=1;
finish=false;
% παρακάτω απλώς μέσω του loop αποθηκεύουμε τα 5 μεγαλύτερα σκορ και την
% θέση τους  σε περίπτωση που θελήσουμε να δείξουμε ποιά είναι.
while finish~=true,
   if i==size(parg,2),
      finish=true;
   else
      i=i+1;
   end;
   if parg(i)==max(parg)&&done~=true,
      if find==5,
         done=true;
      end;
     bestf_skor(find)=max(parg);
     bestf(find)=i;
     parg(i)=0;
     find=find+1;
     i=0;
   end;
end;
bestf
bestf_skor
```

3.10.3. Εύρεση επόμενων βέλτιστων χαρακτηριστικών

Παρακάτω n είναι ο επιθυμητός αριθμός των βέλτιστων επιλεγμένων χαρακτηριστικών και ο m είναι ο αριθμός των ήδη επιλεγμένων χαρακτηριστικών. Έτσι η while θα εκτελέσει τις εσωτερικές εντολές μέχρι να επιλεγούν τα επιθυμητά χαρακτηριστικά. Η minARG παίρνει σαν παράμετρο τον πίνακα των χαρακτηριστικών d, η c είναι το διάνυσμα κατηγοριών και το Fid πίνακας που περιέχει τις θέσεις των επιλεγμένων χαρακτηριστικών και επιστρέφει έναν πίνακα με τα μικρότερα σκορ που επιλέχτηκαν από τις υπό συνθήκη αμοιβαίες πληροφορίες όλων των διανυσμάτων, του διανύσματος κατηγοριών και των ειδή επιλεγμένα χαρακτηριστικών. Στην while γίνετε η εύρεση της θέσης του πρώτου μεγίστου όπως έχουμε εξηγήσει είδη στο παραπάνω κεφάλαιο.

```
find=false;
m=1;
while(m<n)
  [argu]=minARG(d,c,Fid); // υπολογισμός των μικρότερων σκορ.
  l=1;
  while l<=size(argu,2)&&find==false,
          if argu(l)==max(argu),
          best_Featurec=head(l);
          best_Feature=[best_Feature,best_Featurec];
          Fid=[Fid,l];
          Fi=argu(l);
          F=[F,Fi];
          find=true;
         else
          l=l+1;
          end;
   end;
   m=m+1;
   find=false;
end;
```

Παρακάτω καλείται η συνάρτηση mutualinfoc() για κάθε συνδυασμό διανύσματος κατηγοριών, όλων των διανυσμάτων και των ήδη επιλεγμένων χαρακτηριστικών. Μέσα στο

loop2 καλείται η συνάρτηση mutualinfoc() και παίρνει παράμετρο ένα διάνυσμα από το d, το διάνυσμα κατηγοριών c και ένα διάνυσμα από τα επιλεγμένα χαρακτηριστικά. Μετά την εκτέλεση των πρώτων δυο με όλα τα επιλεγμένα χαρακτηριστικά, κρατάμε το μικρότερο σκορ για το διάνυσμα d(:,i). Αυτό γιατί όταν διαλέξουμε το μεγαλύτερο σκορ το διάνυσμα που θα επιλεγεί να μην είναι ίδιο η όμοιο με τα επιλεγμένα.

```
function [argu]=minARG(d,c,Fid)
{
 for i=1:size(d,2), // loop1 για την προσπέλαση όλων των χαρακτηριστικών.
        for j=1:size(Fid,2),  //loop2
                r=d(:,Fid(j));
                mic(j)=mutualinfoc(c,d(:,i),r);
        end;
        argu(i)=min(mic);// επιλογή του μικρότερου
  end;
return;
}
-----------------------------------------------------------------------
function  mic=mutualinfoc(c,di,r)
{
a1=jentropy(c,r);
a2=jentropy(di,r);
a3=trjentropy(c,di,r);
a4=entropy(r);
 mic=a1+a2-a3-a4;
 return;}
```

3.10.4. Αποθήκευση των αποτελεσμάτων στην βάση μας

Στην αρχή αφού πάρουμε το όνομα του μοναδικού πίνακα και το μετατρέψουμε από αριθμό σε κείμενο το ενώνουμε με την κατάληξη .dat για να εισάγουμε τον πίνακα με τα χαρακτηριστικά από το αρχείο και διαγράφουμε το αντίστοιχο αρχείο. Η παράμετρος n είναι ο αριθμός των επιθυμητών χαρακτηριστικών. Το dbname περιέχει το όνομα της βάσης μας και ενώνετε με το όνομα του πίνακα για την χρήση του στην παρακάτω συνάρτηση.

```
function condmutualinfo(fstime,n)
{
fstime=num2str(fstime);
filename=strcat(fstime,'.dat');
t=importdata(filename);
delete(filename);
d=t(:,2:size(t,2));
c=t(:,1);
 dbname='feature_selection.';
dbtable_name=strcat(dbname,fstime);
fet(d,c,n,dbtable_name);
return;
}
```

Στην κυρίως συνάρτηση fet αφού κάνουμε σύνδεση με την βάση μας μέσα στο loop for στο colnames βάζουμε τα ονόματα των στηλών και στο exdata βάζουμε το σκορ και την θέση των επιλεγμένων χαρακτηριστικών έτσι εισάγουμε έναν κάθε φορά με την insert δίνοντας παράμετρο την σύνδεση το όνομα της βάσης ενωμένο με το όνομα του πίνακα τα ονόματα των στύλων και τα δεδομένα μας. Μετά την for κλείνουμε την σύνδεση με την βάση μας.

```
function fet(d, c, n, dbtable_name)
{
….
conn = database('McManus','root','');
for i=1:size(F,2),
colnames = {'selFeatures','Scor'};
exdata = {Fid(i),F(i)};
insert(conn,dbtable_name, colnames, exdata);
end;
close(conn);
}
```

3.11. Αλγόριθμος Μέγιστη Σχετικότητα Ελάχιστος Πλεονασμός (Max Relevance Min Redundancy)

Ο αλγόριθμος αυτός υλοποιείτε σε τρία στάδια .

1. Εύρεση πρώτου Βέλτιστου χαρακτηριστικού I(Fi,C).
2. Εύρεση επόμενων βέλτιστων χαρακτηριστικών.
3. Αποθήκευση των αποτελεσμάτων στην βάση μας στον αντίστοιχο μοναδικό πίνακα που δημιουργήθηκε στην έναρξη.

Τα στάδια 1 , 3 υλοποιείται ομοίως με το προηγούμενο κεφάλαιο. Με την διαφορά ότι στο στάδιο 1 μετά την εύρεση του βέλτιστου χαρακτηριστικού με το διάνυσμα κατηγοριών το βέλτιστο διάνυσμα αφαιρείται από το σύνολο των διανυσμάτων.

3.11.1. Εύρεση επόμενων βέλτιστων χαρακτηριστικών

Παρακάτω μέσα στην loop1 η οποία τερματίζει όταν βρεθούν τα επιθυμητά βέλτιστα χαρακτηριστικά αφού καλέσουμε την miRG μέσα στην loop2 γίνετε η εύρεση του επόμενου βέλτιστου χαρακτηριστικού που έχει το μέγιστο σκορ των γινομένων ή διαφορών και πάντα μέσα στην loop2 επιλέγουμε το πρώτο μέγιστο σκορ που θα συναντήσουμε και αν υπάρχουν και άλλα δεν τα χριζόμαστε γιατί θα έχει τις ίδιες πληροφορίες με αυτό που έχουμε επιλέξει είδη έτσι με το ελάχιστο πλεονασμό το όμοιο θα προκαλέσει μικρό σκορ με αποτέλεσμα να μην επιλεγεί. Δηλαδή η διαφορά ή το πηλίκο της μέγιστης σχετικότητας και του ελάχιστου πλεονασμού θα έχει μικρό σκορ πάντα για χαρακτηριστικά που είναι ίδια η όμοια μετάξι του εξεταζομένου χαρακτηριστικού και των είδη επιλεγμένων. Η miRG παίρνει σαν παράμετρο όλα τα μη επιλεγμένα χαρακτηριστικά , διάνυσμα κατηγοριών και όλα τα είδη επιλεγμένα.

```
finde=false;
while(m<n) // loop1
        [argu]=miRG(d,c,Fidt);
        l=1;
        while l<size(argu,2)&&finde==false, // loop2
                if argu(l)==max(argu),
                best_Featurec=head(l);
                best_Feature=[best_Feature,best_Featurec];
```

```
                Fid=[Fid,head(l)]; // αποθήκευση του νέου βέλτιστου.
                Fi=argu(l);  // η τιμή του σκορ του νέου βέλτιστου.
                F=[F,Fi];
                finde=true;
                Fidt=[Fidt,d(:,l)]; //αποθήκευση του διασίματος του βέλτιστου.
                d(:,l)=[]; // αφαίρεση του νέου βέλτιστου από τα συνολικά .
                head(l)=[]; // αφαιρείτε και η θέση του.
                else
                l=l+1;
                end;
        end;
m=m+1;
finde=false;
end;
```

Παρακάτω μέσα στην loop1 η οποία τερματίζει όταν βρεθούν τα επιθυμητά βέλτιστα χαρακτηριστικά αφού καλέσουμε την miRG μέσα στην loop2 γίνετε η εύρεση του επόμενου βέλτιστου χαρακτηριστικού που έχει το μέγιστο σκορ των γινομένων ή διαφορών και πάντα μέσα στην loop2 επιλέγουμε το πρώτο μέγιστο σκορ που θα συναντήσουμε και αν υπάρχουν και άλλα δεν τα χρειαζόμαστε γιατί θα έχει τις ίδιες πληροφορίες με αυτό που έχουμε επιλέξει είδη έτσι με το ελάχιστο πλεονασμό το όμοιο θα προκαλέσει μικρό σκορ με αποτέλεσμα να μην επιλεγεί. Δηλαδή η διαφορά ή το πηλίκο της μέγιστης σχετικότητας και του ελάχιστου πλεονασμού θα έχει μικρό σκορ πάντα για χαρακτηριστικά που είναι ίδια η όμοια μετάξι του εξεταζομένου χαρακτηριστικού και των είδη επιλεγμένων. Η miRG παίρνει σαν παράμετρο όλα τα μη επιλεγμένα χαρακτηριστικά, διάνυσμα κατηγοριών και όλα τα είδη επιλεγμένα.

```
 function [argu]=miARG(d,c,Fid)
{
mia=0;
for i=1:size(d,2), //loop1
        mi=mutualinfo2(d(:,i),c);
        for j=1:size(Fid,2),//loop2
                temp=mutualinfo2(d(:,i),Fid(:,j));
```

```
            mia=temp+mia;
        end;
        argu(i)=( mi/( mia /size(Fid,2))  );  %#ok<AGROW>
        mia=0;
end;
return;
}
```

Στην loop1 γίνεται η σάρωση του συνόλου όλων των μη επιλεγμένων χαρακτηριστικών. Και μέσα σε αυτήν καλείται η mutualinfo2 που παίρνει παράμετρο το τρέχον χαρακτηριστικό και το διάνυσμα κατηγοριών και επιστρέφει το σκορ για την αμοιβαία πληροφορίας τους. Μέσα στο loop2 αθροίζονται οι αμοιβαίες πληροφορίες των ήδη επιλεγμένων χαρακτηριστικών με το τρέχον χαρακτηριστικό. Μετά το loop2 γίνεται είτε αφαίρεση είτε διαίρεση ανάμεσα στο σκορ μέγιστης σχετικότητας και ελάχιστου πλεονασμού. Έτσι τα μη επιλεγμένα δυνατά χαρακτηριστικά θα έχουν μεγάλα σκορ και αν δεν είναι όμοια η ιδία θα έχουν μικρό σκορ πλεονασμού και η αφαίρεση ή διαίρεση τους θα επιφέρει μεγάλο σκορ γιατί θα έχει μικρό έως μηδαμινό σκορ πλεονασμού και αυτά που είναι ίδια η όμοια επειδή θα έχουν μεγάλο σκορ πλεονασμού αντίστοιχα θα έχουν μικρό σκορ στην αφαίρεση ή διαίρεση με αποτέλεσμα να μην επιλέγουν αργότερα. Τέλος παρακάτω στην mutualinfo2 γίνεται η εφαρμογή του κλασικού τύπου όπως έχουμε προαναφέρει κεφάλαιο 2.2.5.

Παρακάτω γίνεται η εφαρμογή του τύπου αμοιβαίας πληροφορίας.

```
function mi=mutualinfo2(x,y)
{
atrisma=0;
for i=1:size(x,1),
        p2=pithan(i,x);
        p1=pithand(i,x,y);
        p3=pithan(i,y);
        If p1~=0
                atrisma=( (log(p1/(p2*p3)) )/log(2) )+atrisma;
        end;
end;
mi=atrisma/(size(x,1));    return;}
```

3.12. Αλγόριθμος κανονικοποίησης των τιμών σε έναν πίνακα

Στην normalazationq αφού πάρουμε σαν παράμετρο το μοναδικό όνομα του αρχείου που περιέχει τον πίνακα με τα χαρακτηριστικά ενώνουμε στο όνομα την κατάληξη ".dat" και εισάγουμε τον πίνακα στην συνάρτηση μας. Παράδειγμα ο πίνακας 6.α ο οποίος περιέχει δεδομένα από προσομοιώσεις που έγιναν με το δυναμικό σύστημα Mackey_Glass για τιμές παραμέτρους D=100,200,300 πάνω στις οποίες υπολογιστήκαν 351 χαρακτηριστικά και 150 δείγματα το καθένα. Τα χαρακτηριστικά αυτά είναι γραμμικά, μη γραμμικά, ταλάντωσης και διάφορα άλλα στατιστικά μέτρα. Χωρίζονται σε τρεις κατηγορίες για κάθε Δ. Τα πρώτα 50 δείγματα ανήκουν σε χρόνο με Δ=100, τα άλλα 50 για Δ=200 και τα υπόλοιπα 50 για Δ=300.

```
function normalazationq(fstime,n)
{
fstime=num2str(fstime);
filename=strcat(fstime,'.dat');
b=importdata(filename);
delete(filename);
norm=b(:,1); // αποθηκεύουμε διάνυσμα κατηγοριών.
for i=2:size(b,2),
    for j=1:size(b,1) ,
            norm(j,i)=(b(j,i)-min(b(:,i)))/(max(b(:,i))-min(b(:,i)));
    end;
end;
save(filename, 'norm');
metatrop_normalzedq(fstime,n);
return;
}
```

Σαρώνοντας όλες τις στήλες εφαρμόζουμε τον τύπο της εξίσωσης 22, όπου στον αριθμητή από το στοιχείο που βρίσκετε στην θέση *b(i,j)* αφαιρείται το μικρότερο στοιχείο από την αντίστοιχη στήλη και στον παρανομαστή έχουμε την αφαίρεση του μικρότερου από το μεγαλύτερο στοιχείο στην αντίστοιχη στήλη. Με αποτέλεσμα όλα τα νούμερα μας στον νέο πίνακα *norm* (πίνακας 6.β) να έχουν εύρος τιμών από το 0 μέχρι το 1 όπως βλέπουμε στους πίνακες παρακάτω .

$$f'(j,i) = \frac{\mathrm{b(j,i)} - \min(\mathrm{b(:,i)})}{\max(\mathrm{b(:,i)}) - \min(\mathrm{b(:,i)})} \qquad \textbf{εξ. 22}$$

Πίνακας 7: α) Πίνακας δεδομένων με 150 γραμμές και 351 στήλες (ορατές στην εικόνα 32 γραμμές και 18 στήλες) όπου οι τιμές δεν είναι κανονικοποιημένες. β)Ο πίνακας 6.α κανονικοποιημένος .

α)

MGD2 <150x351 double>

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
1	0.0572	0.5900	0.2360	0.1024	15.7969	0.7206	16	0.2027	0.1212	0.0453	0.0204	4.4029	3.1962	11	-0.3009	-0.1000	0.0167	0.0115	-4
2	0.9015	0.7140	0.4292	0.2240	18.7476	10.0435	23	0.3419	0.1730	0.0672	0.0466	5.5302	3.9022	12	-0.4955	-0.1835	-0.0346	0.0208	-6
3	0.9063	0.7194	0.4073	0.2220	18.7167	10.0924	22	0.3477	0.1701	0.0592	0.0408	5.4552	3.9631	12	-0.5135	-0.1862	-0.0315	0.0156	-6
4	0.8747	0.6315	0.2844	0.1031	14.7998	9.4878	17	0.3140	0.1471	0.0460	0.0324	4.8513	3.6968	12	-0.4102	-0.1073	0.0038	0.0271	-4
5	0.9223	0.7687	0.4967	0.3210	21.5235	11.0999	27	0.3660	0.1981	0.0817	0.0494	6.1212	4.3492	13	-0.5846	-0.2488	-0.0599	-0.0050	-8
6	0.8894	0.6769	0.3712	0.2127	17.6147	9.7879	21	0.3289	0.1622	0.0531	0.0324	5.0885	3.7715	12	-0.4537	-0.1429	-0.0210	0.0093	-5
7	0.8867	0.6675	0.3005	0.0412	14.9111	9.7303	18	0.3101	0.1458	0.0414	0.0325	4.7115	3.5725	12	-0.4613	-0.1491	-0.0059	0.0317	-5
8	0.9194	0.7954	0.4703	0.2632	20.1745	10.3421	25	0.3604	0.1878	0.0024	0.0426	5.9260	4.1027	12	-0.5729	-0.2350	-0.0426	0.0067	-7
9	0.9200	0.7602	0.4812	0.2543	20.1461	11.0363	25	0.3678	0.1963	0.0721	0.0385	5.8414	4.3128	13	-0.5887	-0.2360	-0.0595	0.0051	-7
10	0.9173	0.7539	0.4722	0.2406	19.0997	10.9813	25	0.3660	0.1875	0.0811	0.0470	5.9574	4.2091	13	-0.5630	-0.2325	-0.0450	0.0151	-7
11	0.8425	0.5266	0.0398	-0.1169	12.0514	8.3841	13	0.2881	0.1122	0.0363	0.0271	4.0480	3.0436	11	-0.3505	-0.0502	0.0355	0.0202	-3
12	0.8932	0.6815	0.3212	0.0491	15.4864	9.8339	19	0.3164	0.1547	0.0491	0.0312	4.8849	3.6485	12	-0.4827	-0.1576	-0.0054	0.0300	-5
13	0.8688	0.6126	0.2389	0.0622	13.7656	9.3601	16	0.2889	0.1312	0.0424	0.0235	4.4337	3.3592	12	-0.4139	-0.1040	0.0130	0.0215	-4
14	0.8974	0.6923	0.3315	0.1382	16.2577	9.9066	19	0.3308	0.1624	0.0558	0.0386	5.2113	3.7746	12	-0.4873	-0.1638	-0.0024	0.0289	-5
15	0.9209	0.7849	0.5387	0.3415	22.1900	11.2410	29	0.3707	0.2006	0.0841	0.0553	6.3157	4.3875	13	-0.6220	-0.2701	-0.0073	-0.0067	-8
16	0.8547	0.5765	0.1705	0.0259	12.3140	8.6357	15	0.2805	0.1195	0.0360	0.0233	4.1428	3.1697	11	-0.3752	-0.0824	0.0212	0.0230	-4
17	0.9126	0.7304	0.4053	0.3629	21.6179	10.2231	30	0.3537	0.1956	0.0840	0.0607	6.0056	4.0369	12	-0.5406	-0.2084	-0.0494	-0.0099	-7
18	0.8870	0.6627	0.3108	0.1600	16.2806	10.2393	18	0.3163	0.1492	0.0493	0.0262	4.7763	3.7261	13	-0.4563	-0.1400	-0.0015	0.0133	-5
19	0.8972	0.6874	0.3584	0.2031	17.8271	9.8832	20	0.3214	0.1524	0.0608	0.0354	5.0230	3.6846	12	-0.4960	-0.1674	-0.0180	0.0144	-6
20	0.8670	0.6085	0.2316	0.0014	13.6260	8.8216	16	0.2889	0.1306	0.0397	0.0314	4.4401	3.2650	11	-0.4078	-0.1006	0.0121	0.0314	-4
21	0.9183	0.7524	0.4586	0.2969	20.7798	10.3199	25	0.3605	0.1915	0.0757	0.0524	6.0050	4.1436	12	-0.5664	-0.2260	-0.0423	0.0062	-7
22	0.9332	0.8004	0.5603	0.3917	23.5092	10.6397	32	0.3947	0.2294	0.1108	0.0770	7.1929	4.5503	12	-0.6209	-0.2823	-0.0777	-0.0063	-8
23	0.8695	0.6325	0.3358	0.2042	16.6051	9.4708	19	0.2944	0.1372	0.0441	0.0346	4.5916	3.4280	12	-0.4111	-0.1183	-0.0158	0.0133	-5
24	0.9006	0.7025	0.3773	0.1375	16.7007	10.5665	21	0.3317	0.1664	0.0583	0.0293	5.1551	3.9273	13	-0.5016	-0.1737	-0.0105	0.0190	-6
25	0.9222	0.7668	0.4733	0.2423	19.7841	10.4088	26	0.3627	0.1919	0.0737	0.0497	6.0285	4.1468	12	-0.5874	-0.2497	-0.0531	0.0194	-7
26	0.9115	0.7295	0.3912	0.1492	17.4257	10.7060	21	0.3479	0.1795	0.0609	0.0349	5.4059	4.1140	13	-0.5405	-0.2003	-0.0222	0.0236	-6
27	0.9103	0.7338	0.4141	0.1661	17.8154	10.1869	22	0.3442	0.1746	0.0650	0.0415	5.5605	3.9332	12	-0.5376	-0.2121	-0.0291	0.0276	-6
28	0.8905	0.6719	0.3039	0.0908	15.1697	9.7895	18	0.3157	0.1436	0.0479	0.0311	4.7977	3.6084	12	-0.4719	-0.1567	-6.0174e-04	0.0270	-5
29	0.9305	0.7901	0.5178	0.2964	21.3834	11.2764	27	0.3853	0.2198	0.0940	0.0533	6.6115	4.6010	13	-0.6190	-0.2697	-0.0621	0.0074	-8
30	0.9048	0.7184	0.4021	0.1868	17.8844	10.0801	22	0.3390	0.1772	0.0678	0.0416	5.5261	3.9233	12	-0.5146	-0.1858	-0.0204	0.0239	-6
31	0.9006	0.6992	0.3523	0.1208	16.4962	9.9595	20	0.3341	0.1697	0.0590	0.0360	5.3326	3.8496	12	-0.4993	-0.1659	-0.0065	0.0276	-6
32	0.9195	0.7693	0.4651	0.2175	19.2585	10.3623	24	0.3557	0.1879	0.0677	0.0387	5.6624	4.0688	12	-0.5781	-0.2417	-0.0542	0.0144	-7

β)

x1287692217 <150x351 double>

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
1	0.1045	0.1690	0.2364	0.3279	0.1655	0.0069	0.0345	0.0450	0.0252	0.0389	0.0203	0.0309	0.0077	0	0.9594	0.9233	0.9553	0.8985	0
2	0.4189	0.4437	0.4691	0.3735	0.2958	0.0334	0.1149	0.2192	0.1701	0.1103	0.0934	0.1292	0.0431	0.0145	0.8067	0.7949	0.8338	0.9324	0
3	0.4532	0.4564	0.4428	0.3713	0.2945	0.0343	0.1034	0.2364	0.1845	0.0842	0.0704	0.1227	0.0462	0.0145	0.7827	0.7907	0.8410	0.9134	0
4	0.2285	0.2484	0.2947	0.2410	0.1214	0.0222	0.0460	0.1371	0.0977	0.0412	0.0366	0.0700	0.0278	0.0145	0.9204	0.9121	0.9249	0.9550	0
5	0.5668	0.5731	0.5504	0.4799	0.4185	0.0546	0.1609	0.2903	0.2403	0.1578	0.1049	0.1808	0.0656	0.0290	0.6879	0.6945	0.7738	0.8390	0
6	0.3329	0.3534	0.3992	0.3611	0.2458	0.0282	0.0920	0.1811	0.1399	0.0642	0.0367	0.0907	0.0365	0.0145	0.8624	0.8573	0.8659	0.8908	0
7	0.3139	0.3337	0.3141	0.1732	0.1263	0.0271	0.0575	0.1256	0.0939	0.0261	0.0370	0.0579	0.0266	0.0145	0.8522	0.8478	0.9017	0.9717	0
8	0.5463	0.5418	0.5186	0.4164	0.3589	0.0394	0.1379	0.2736	0.2116	0.1600	0.0775	0.1637	0.0532	0.0145	0.7036	0.7157	0.8149	0.8814	0
9	0.5502	0.5532	0.5317	0.4067	0.3576	0.0533	0.1379	0.2956	0.2354	0.1264	0.0610	0.1564	0.0637	0.0290	0.7092	0.7156	0.7746	0.8754	0
10	0.5311	0.5381	0.5210	0.4005	0.3467	0.0522	0.1379	0.2632	0.2107	0.1559	0.0953	0.1665	0.0585	0.0290	0.7157	0.7195	0.8091	0.9118	0
11	0	0	0	0	0	0	0	0.0020	0	0.0096	0.0152	0	0	0	1	1	1	0.9302	
12	0.3598	0.3668	0.3391	0.1818	0.1518	0.0291	0.0690	0.1442	0.1190	0.0512	0.0318	0.0730	0.0304	0.0145	0.8238	0.8348	0.9030	0.9657	0
13	0.1868	0.2035	0.2399	0.1963	0.0757	0.0196	0.0345	0.0633	0.0532	0.0294	0.0305	0.0336	0.0158	0.0145	0.9165	0.9172	0.9467	0.9350	0
14	0.3899	0.3923	0.3515	0.2796	0.1858	0.0306	0.0690	0.1867	0.1405	0.0732	0.0614	0.1014	0.0367	0.0145	0.8176	0.8252	0.9101	0.9618	0
15	0.6138	0.6117	0.6010	0.5023	0.4480	0.0574	0.1839	0.3040	0.2474	0.1654	0.1285	0.1977	0.0675	0.0290	0.6370	0.6493	0.7089	0.8327	0
16	0.0871	0.1181	0.1575	0.1565	0.0116	0.0051	0.0230	0.0387	0.0205	0.0085	0	0.0083	0.0063	0	0.9670	0.9504	0.9681	0.9402	0
17	0.4979	0.5015	0.5367	0.5257	0.4227	0.0370	0.1954	0.2540	0.2054	0.1679	0.1503	0.1777	0.0499	0.0145	0.7465	0.7567	0.7986	0.8212	0
18	0.3160	0.3222	0.3265	0.3034	0.1869	0.0373	0.0575	0.1440	0.1034	0.0520	0.0117	0.0635	0.0343	0.0290	0.8590	0.8618	0.9123	0.9051	0
19	0.3887	0.3808	0.3839	0.3506	0.2552	0.0301	0.0805	0.1591	0.1126	0.0568	0.0487	0.0860	0.0322	0.0145	0.8061	0.8196	0.8732	0.9091	0
20	0.1745	0.1939	0.2311	0.1296	0.0696	0.0088	0.0345	0.0633	0.0516	0.0206	0.0325	0.0342	0.0111	0	0.9236	0.9225	0.9445	0.9707	0
21	0.5387	0.5347	0.5045	0.4534	0.3856	0.0389	0.1379	0.2739	0.2219	0.1382	0.1168	0.1706	0.0552	0.0145	0.7122	0.7295	0.8155	0.8796	0
22	0.6444	0.6482	0.6271	0.5573	0.5062	0.0453	0.2184	0.3746	0.3279	0.2525	0.2157	0.2742	0.0756	0.0145	0.6289	0.6429	0.7317	0.8342	0
23	0.1921	0.2509	0.3666	0.3519	0.2012	0.0218	0.0690	0.0795	0.0699	0.0348	0.0454	0.0474	0.0193	0.0145	0.9192	0.8952	0.8784	0.9052	0
24	0.4128	0.4165	0.4066	0.2787	0.2054	0.0439	0.0920	0.1892	0.1517	0.0813	0.0241	0.0965	0.0444	0.0290	0.7986	0.8099	0.8720	0.9286	0
25	0.5661	0.5664	0.5222	0.3936	0.3416	0.0407	0.1379	0.2804	0.2229	0.1316	0.1059	0.1727	0.0554	0.0145	0.6842	0.6931	0.7899	0.9273	0
26	0.4902	0.4803	0.4233	0.2916	0.2374	0.0483	0.0920	0.2370	0.1882	0.0896	0.0465	0.1184	0.0538	0.0290	0.7467	0.7691	0.8632	0.9425	0
27	0.4813	0.4907	0.4510	0.3101	0.2547	0.0362	0.1034	0.2260	0.1746	0.1030	0.0733	0.1319	0.0447	0.0145	0.7506	0.7509	0.8467	0.9568	0
28	0.3412	0.3440	0.3182	0.2275	0.1378	0.0278	0.0575	0.1423	0.0877	0.0472	0.0313	0.0654	0.0284	0.0145	0.8382	0.8361	0.9143	0.9546	0
29	0.6250	0.6240	0.5759	0.4529	0.4123	0.0581	0.1609	0.3470	0.3012	0.1979	0.1207	0.2235	0.0782	0.0290	0.6421	0.6624	0.7687	0.8837	0
30	0.4424	0.4541	0.4365	0.3327	0.2577	0.0341	0.1034	0.2107	0.1818	0.1124	0.0736	0.1289	0.0442	0.0145	0.7812	0.7914	0.8675	0.9434	0
31	0.4130	0.4086	0.3765	0.2692	0.1965	0.0317	0.0805	0.1963	0.1608	0.0861	0.0511	0.1120	0.0405	0.0145	0.8017	0.8220	0.9004	0.9572	0
32	0.5470	0.5509	0.5123	0.3664	0.3184	0.0398	0.1264	0.2599	0.2119	0.1119	0.0617	0.1408	0.0515	0.0145	0.6966	0.7054	0.7874	0.9093	0

3.13. Αλγόριθμος τοποθέτησης των κανονικοποιημένων τιμών σε χωρία (bins)

Παρακάτω αφού εισάγουμε τον πίνακα μας από το αρχείο όπως έχουμε εξηγήσει ήδη ορίζουμε μια μεταβλητή με το όνομα ran που θα περιέχει τον αριθμό των χορίων και μια μεταβλητή er που θα περιέχει το εύρος του κάθε χωρίου που στην περίπτωση μας είναι 1/ran=0,1 επειδή ο πίνακας μας έχει εύρος τιμών 0-1 γι' αυτό και διαιρούμε 1/ran. Μέσα στην while γίνεται έλεγχος αν το ανάλογο κελί δεν περιέχει αριθμό και έχει NaN η null τοποθετείται στο πρώτο χωρίο 0. Μετά στην elseif γίνετε ο έλεγχος του κελιού αν ανήκει στο τρέχων χωρίο και αν ναι τοποθετείται εκεί και τερματίζει η while επειδή δίνουμε τιμή true στην done, που σημαίνει πως έχει βρεθεί σε ποιό χωρίο ανήκει το κελί αλλιώς στην else αυξάνουμε το τρέχον χωρίο κατά μια τιμή er2=er2+er και εκτελούμε έτσι τις εντολές της while μέχρι να βρεθεί σε ποιο χωρίο ανήκει το κελί. Παρακάτω βλέπουμε τον πίνακα που παρουσιάσαμε στο προηγούμενο κεφάλαιο μετά από τοποθέτηση των τιμών σε χωρία.

```
function metatrop_normalzedq(fstime,n)
{
fstime=num2str(fstime);
filename=strcat(fstime,'.dat');
b=importdata(filename);
delete(filename);
 ran=10; // σε ποσά χωρία θέλουμε να κατατάξουμε τα δεδομένα.
er=1/ran;
er2=er; // τι εύρος θα έχει κάθε χωρίο. 0.1 στην περίπτωση αυτή.
bin=b;
done=false;
binfinal=b(:,1); // αποθηκεύουμε διάνυσμα κατηγοριών.
for i=2:size(b,2),
        for j=1:size(b,1), // Σάρωση όλων τον δεδομένων μέσο διπλής  loop .
                while done==false,
                 if isnan(bin(j,i)), // αν δεν είναι αριθμός.
                        binfinal(j,i)=0;
                        done=true;
```

```
                elseif bin(j,i)<=er2,
                        binfinal(j,i)=int8(er2*10); //τοποθέτηση στο ανάλογο χωρίο.
                        bin(j,i)=nan; // σβήσιμο των παλιών δεδομένων.
                        done=true;
                        er2=er; // Επαναφορά του er2 στην πρώτη τιμή 0,1 .
                    else
                        er2=er2+er;
                    end;
                end;
                done=false;
            end;
end;
 bin=double(binfinal);  // μετατροπή πίνακα σε double.
 save(filename, 'bin')
mrmr_miq(fstime,n);
return;
}
```

Πίνακας 8: Ο πίνακας 6.β με τοποθετημένες τις τιμές σε χωρία.

x1287692928 <150x351 double>

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
1	2	2	3	4	2	1	1	1	1	1	1	1	1	1	10	10	10	9	
2	5	5	5	4	3	1	2	3	2	2	1	2	1	1	9	8	9	10	
3	5	5	5	4	3	1	2	3	2	1	1	2	1	1	8	8	9	10	
4	3	3	3	3	2	1	1	2	1	1	1	1	1	1	10	10	10	10	
5	6	6	6	5	5	1	2	3	3	2	2	2	1	1	7	7	8	9	
6	4	4	4	4	3	1	1	2	2	1	1	1	1	1	9	9	9	9	
7	4	4	4	2	2	1	1	2	1	1	1	1	1	1	9	9	10	10	
8	6	6	6	5	4	1	2	3	3	2	1	2	1	1	8	8	9	9	
9	6	6	6	5	4	1	2	3	3	2	1	2	1	1	8	8	9	9	
10	6	6	6	5	4	1	2	3	3	2	1	2	1	1	8	8	9	10	
11	1	1	1	1	1	1	1	1	1	1	1	1	1	1	11	11	11	10	
12	4	4	4	2	2	1	1	2	2	1	1	1	1	1	9	9	10	10	
13	2	3	3	2	1	1	1	1	1	1	1	1	1	1	10	10	10	10	
14	4	4	4	3	2	1	1	2	2	1	1	2	1	1	9	9	10	10	
15	7	7	7	6	6	1	2	4	3	2	2	2	1	1	7	7	8	9	
16	1	2	2	2	1	1	1	1	1	1	1	1	1	1	10	10	10	10	
17	5	6	6	6	5	1	2	3	3	2	2	2	1	1	8	8	8	9	
18	4	4	4	4	2	1	1	2	2	1	1	1	1	1	9	9	10	10	
19	4	4	4	4	3	1	1	2	2	1	1	1	1	1	9	9	9	10	
20	2	2	3	2	1	1	1	1	1	1	1	1	1	1	10	10	10	10	
21	6	6	6	5	4	1	2	3	3	2	2	2	1	1	8	8	9	9	
22	7	7	7	6	6	1	3	4	4	3	3	3	1	1	7	7	8	9	
23	2	3	4	4	3	1	1	1	1	1	1	1	1	1	10	9	9	10	
24	5	5	5	3	3	1	1	2	2	1	1	1	1	1	8	9	9	10	
25	6	6	6	4	4	1	2	3	3	2	2	2	1	1	7	7	8	10	
26	5	5	5	3	3	1	1	3	2	1	1	2	1	1	8	8	9	10	
27	5	5	5	4	3	1	2	3	2	2	1	2	1	1	8	8	9	10	
28	4	4	4	3	2	1	1	2	1	1	1	1	1	1	9	9	10	10	
29	7	7	6	5	5	1	2	4	4	2	2	3	1	1	7	7	8	9	
30	5	5	5	4	3	1	2	3	2	2	1	2	1	1	8	8	9	10	
31	5	5	4	3	2	1	1	2	2	1	1	2	1	1	9	9	10	10	
32	6	6	6	4	4	1	2	3	3	2	1	2	1	1	7	8	8	10	

3.14. Αποτελέσματα

Ύστερα από πολλές εκτελέσεις των αλγορίθμων μας συγκρίνουμε τα αποτελέσματα με τα αποτελέσματα από την ιστοσελίδα του Hanchuan Peng στην παράγραφο 3.14.1. , με τα

αποτελέσματα από την εκτέλεση του spider στην παράγραφο 3.14.2. , και παρατηρούμε πως έχουμε τα επιθυμητά αποτελέσματα .

Στην διαδικασία σύγκρισης με την σελίδα του Hanchuan Peng και με το spider παρατηρήσαμε πως για να έχουμε αποτελέσματα ίδια και με τους δυο όταν χρησιμοποιούμε τον τύπο αμοιβαίας πληροφορίας p(x,y)log(p(x,y)/p(x)p(y)) χωρίς την κοινή πιθανότητα μπροστά δηλαδή μόνο log(p(x,y)/p(x)p(y)). Αυτό συμβαίνει γιατί ο τύπος είναι γενικός και στην περίπτωση μας έχουμε δισδιάστατο πίνακα με διανύσματα και τα χαρακτηριστικά ελέγχονται διανυσματικά κατά ζεύγη. Αν τα δεδομένα μας αφορούσαν αρχείο κειμένου θα είχαμε κάνει χρήση του γενικού τύπου. Παρ' όλα αυτά μπούμε να βάλουμε απλά την κοινή πιθανότητα μπροστά και μας δίνει τα ίδια αποτελέσματα τις περισσότερες φόρες με μικρότερο σκορ σχετικότητας.

3.14.1. Σύγκριση αποτελεσμάτων με την ιστοσελίδα του Hanchuan Peng

Η δοκιμή των αλγορίθμων έχει γίνει όπως είδαμε ήδη στο κεφάλαιο εφαρμογή με δεδομένα δικά μας αλλά επίσης έχει γίνει σύγκριση των αποτελεσμάτων και με την ηλεκτρονική σελίδα του Hanchuan Peng στο http://penglab.janelia.org/proj /mRMR/index.htm ο οποίος κάνει εφαρμογή τους mRmR διαίρεση και πηλίκο λαμβάνονται το σετ δεδομένων από την σελίδα του και εκτελώντας τους δικούς μας και τους αντίστοιχα δικούς του αλγορίθμους παρατηρήσαμε πως τα αποτελέσματα είναι τα ιδία.

Στις παρακάτω εικόνες βλέπουμε τα αποτελέσματα από την εκτέλεση των δικών μας αλγορίθμων και τα αποτελέσματα από την ιστοσελίδα του Hanchuan Peng (Εικόνα 6.α) χρησιμοποιώντας το σετ δεδομένων το όποιο είναι microrray στοιχεία του καρκίνου του πνεύμονα με 7 κατηγορίες και το οποίο το προμηθευτήκαμε από την ιστοσελίδα του Hanchuan Peng.

Εικόνα 6: Αρχική σελίδα του Hanchuan Peng.

mRMR (minimum Redundancy Maximum Relevance Feature Selection)

[Frequently Asked Questions | Online Version | C/C++ Version with Source Codes | Matlab Version | Sample I
Publications | BIBM'07 Tutorial Slides]

- **Online Version**: You can run the program using your own data through the following form, -- you can also download the program and run on your own ma

  * Data file (Standard CSV file format, where each row is a sample and each column is a variable/attribute/feature. MAKE SURE YOUR DATA IS SEP BUT NOT BLANK SPACE OR OTHER CHARACTERS!! The first row must be the feature names, and the first column must be the classes for sam testing example data set here, which is microrray data of lung cancer (7 classes). The data has been discretized as 3-states. Note that the web-based pr data file with the maximum size 2M bytes, and maximum number of variables = 10000 -- if you have a larger data set, you should download the prograr machine (see download links below).

  Επιλογή αρχείου Δεν έχει ... νο αρχείο

  * What is the feature selection scheme you want to use:

  MID (Mutual Information Difference)

  * How many features you want to select:

  50

  * What is the type of your data? "Categorical" means each attribute/variable/feature in your data is discretized as a few categorical states. "Continuous" numerical values. If you choose "Categorical" then the last option below will have no effect. For mutual information based feature selection methods lik mRMR, you might want to discretize your own data first as a few categorical states, -- empirically this leads to better results than continuous-value mut computation. You can also use the option below to discretize your data using two thresholds mean+/-k*std.

  Categorical
  Continuous

  * In case you have continuous variables in your data file, what is the threshold you want to discretize your data, i.e. mean +/- k*std:

  k=1

  Submit Job | Reset Input

Εικόνα 7: Αποτελέσματα Επιλογής 5 χαρακτηριστικών γνωρισμάτων από τον αλγόριθμο mRmR διαίρεση από την σελίδα του Hanchuan Peng.

* You have uploaded a file with the size 56730 bytes. Now it is being parsed and computed ..... Results are

```
You have specified parameters: no discretization, #fea=5 selection method=MIQ #max

Target classification variable (#1 column in the input data) has name=class     er

*** MaxRel features ***
Order     Fea      Name     Score
1         23       v23      0.773
2         11       v11      0.766
3         20       v20      0.756
4         30       v30      0.748
5         151      v151     0.736

*** mRMR features ***
Order     Fea      Name     Score
1         23       v23      0.773
2         140      v140     24.104
3         275      v275     13.216
4         105      v105     10.895
5         235      v235     12.982


 *** This program and the respective minimum Redundancy Maximum Relevance (mRMR)
     algorithm were developed by Hanchuan Peng for
     the paper
     "Feature selection based on mutual information: criteria of
      max-dependency, max-relevance, and min-redundancy,"
      Hanchuan Peng, Fuhui Long, and Chris Ding,
      IEEE Transactions on Pattern Analysis and Machine Intelligence,
      Vol. 27, No. 8, pp.1226-1238, 2005.
```

Πίνακας 9: Αποτελέσματα Επιλογής 5 χαρακτηριστικών γνωρισμάτων από τον αλγόριθμο mRmR διαίρεση από την δική μας εφαρμογή με το ίδιο σετ δεδομένων με την εικόνα 4.α .

Minimum Redundancy - Maximum Relevance for Categorical (Discrete) Variables With ***Mutual information quotient***

Results

Order	Selected Features	Score
1	23	0.773383
2	140	24.3563
3	275	13.2737
4	105	10.9178
5	235	13.0185

Results code :1285708181

Εικόνα 8: Αποτελέσματα Επιλογής 5 χαρακτηριστικών γνωρισμάτων από τον αλγόριθμο mRmR διαφορά από την σελίδα του Hanchuan Peng.

* You have uploaded a file with the size 56730 bytes. Now it is being parsed and computed ..... Results ar

```
You have specified parameters: no discretization, #fea=5 selection method=MID #m

Target classification variable (#1 column in the input data) has name=class

*** MaxRel features ***
Order     Fea      Name     Score
1         23       v23      0.773
2         11       v11      0.766
3         20       v20      0.756
4         30       v30      0.748
5         151      v151     0.736

*** mRMR features ***
Order     Fea      Name     Score
1         23       v23      0.773
2         126      v126     0.555
3         244      v244     0.567
4         133      v133     0.533
5         243      v243     0.538

 *** This program and the respective minimum Redundancy Maximum Relevance (mRMR)
     algorithm were developed by Hanchuan Peng for
     the paper
     "Feature selection based on mutual information: criteria of
      max-dependency, max-relevance, and min-redundancy,"
      Hanchuan Peng, Fuhui Long, and Chris Ding,
      IEEE Transactions on Pattern Analysis and Machine Intelligence,
      Vol. 27, No. 8, pp.1226-1238, 2005.
```

Πίνακας 10: Αποτελέσματα Επιλογής 5 χαρακτηριστικών γνωρισμάτων από τον αλγόριθμο mRmR διαφορά από την δική μας εφαρμογή με το ίδιο σετ δεδομένων με την εικόνα 5.α.

Minimum Redundancy - Maximum Relevance for Categorical (Discrete) Variables With ***Mutual information difference***

Results

Order	Selected Features	Score
1	23	0.773383
2	126	0.555003
3	244	0.566919
4	133	0.533325
5	243	0.53842

Results code :1285708686

3.14.2. Σύγκριση αποτελεσμάτων με το Spider

Εκτός από την σελίδα του Hanchuan Peng έχουμε χρησιμοποιήσει και τα έτοιμα εργαλεία που μας παρέχει το spider http://www.kyb.mpg.de/bs/people/spider/ κατεβάζοντας τα Core files και από τα extras το εργαλείο mutinf Feature selection by mutual information τα αποσυμπιέζουμε και βάζουμε τον φάκελο @mutinf του mutinf μέσα στο φάκελο C:\spider\feat_sel και αφού δώσουμε και τις διαδρομές στο Matlab για να μπορεί να τα δει είμαστε έτοιμοι.

Χρησιμοποιώντας τα εργαλείο του spider και εκτελώντας τις παρακάτω εντολές παρατηρούμε πως τα 10 επιλεγμένα χαρακτηριστικά που είναι στην εικόνα 9.α το spider ναι μεν κάνει επιλογή χαρακτηριστικών επιλέγει όμως χαρακτηριστικά που είναι όμοια έως και ιδία μεταξύ τους έτσι αν του ζητήσουμε να μας βρει λίγα χαρακτηριστικά σε ένα πίνακα δεδομένων που έχει πολλά κοινά δυνατά χαρακτηριστικά αυτό ουσιαστικά θα μας επιλέξει ένα σετ με ιδία χαρακτηριστικά. Εκτελώντας τον δικό μας αλγόριθμο επιλογή χαρακτηριστικών με αμοιβαία πληροφορία διαίρεση (mRmRq) με τα ίδια δεδομένα όπως παρατηρούμε στην εικόνα 6.β βλέπουμε πως τα 5 χαρακτηριστικά (bestf) με το μεγαλύτερο σκορ σχετικότητας είναι ιδία με τα επιλεγμένα του spider και πως τα σκορ σχετικότητας τους είναι ίδιο και στα 5 (bestf_skor). Στην εικόνα 6.β στα best_Feature βλέπουμε πως ο δικός μας αλγόριθμος απορρίπτει τα χαρακτηριστικά 6 και 26 γιατί είναι όμοια η ίδια με τα υπόλοιπα επιλεγμένα και επιλεγεί 58, 75 που έχουν μικρότερο σκορ άλλα είναι λιγότερα όμοια με τα υπόλοιπα .

Κατά την διαδικασία σύγκρισης έχουμε χρησιμοποιήσει δεδομένα από προσομοιώσεις που έγιναν με το δυναμικό σύστημα Mackey_Glass για τιμές παραμέτρους Δ=100,200,300 πάνω στις οποίες υπολογιστήκαν 351 χαρακτηριστικά και 150 δείγματα το καθένα. Τα χαρακτηριστικά αυτά είναι γραμμικά, μη γραμμικά, ταλάντωσης και διάφορα άλλα στατιστικά μέτρα.

```
fno=10;
d=data(x,y);
a=mutinf(svm);
a.method='classification';
a.feat=fno;
[r,a]=train(a,d);
er=loss(test(a,d));
f=a.rank(1:fno)
plot(x(:,f(1:fno)))
```

Εικόνα 9: α) Εκτέλεση του Spider και επιλογή των 10 βέλτιστων χαρακτηριστικών από αυτόν(10 γιατί παρακάτω θα επιλέξουμε κάποια χαρακτηριστικά που είναι μέσα στην δεκάδα). β) Η επιλογή των 5 καλύτερων χαρακτηριστικών από την δική μας εφαρμογή με το ίδιο σετ δεδομένων με την εικόνα 6.α

α)

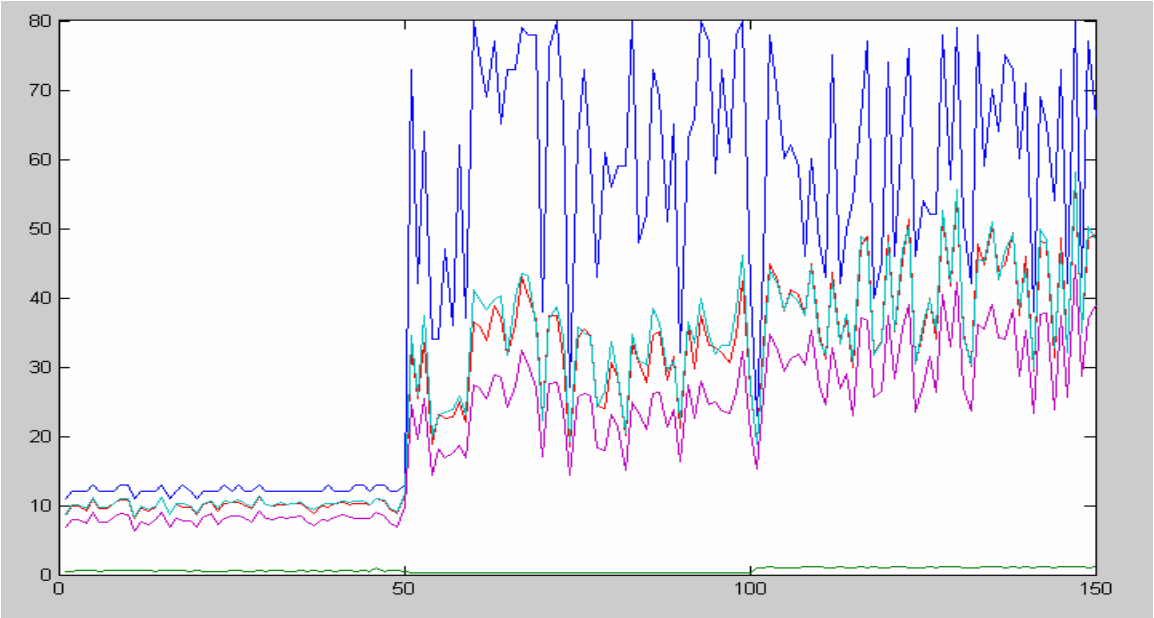
```
>> mi_spider
training mutinf feat=10....
training svm kernel linear....

f =

    14
    54
    26
     6
    32
    13
   227
    71
    73
    74

>>
```

β)

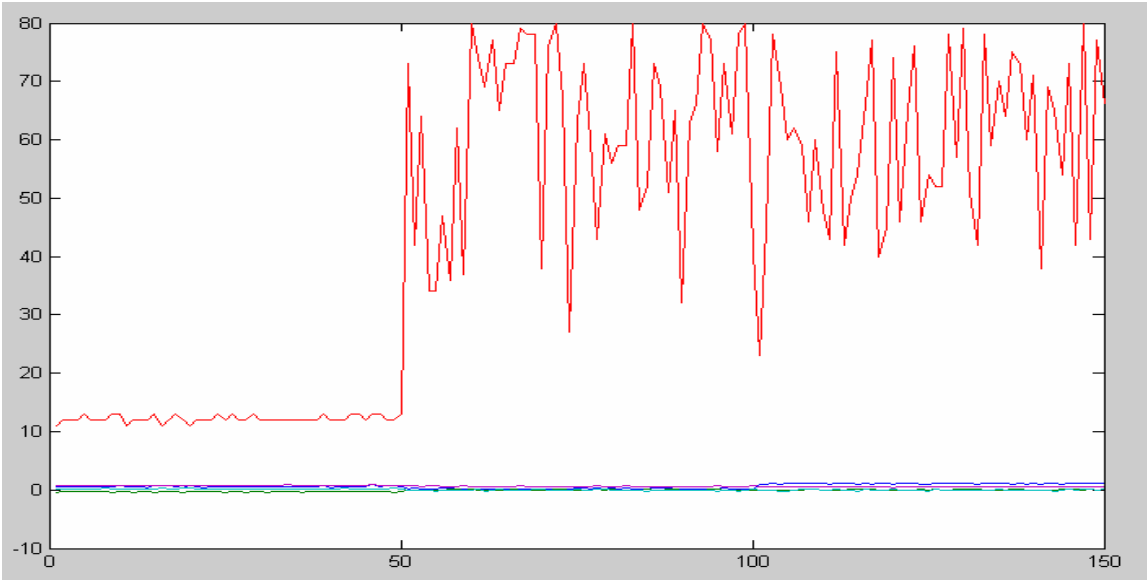
```
bestf =

    54   227     6    14    26


bestf_skor =

    0.9183    0.9183    0.9183    0.9183    0.9183


best_Feature =

    54   227    14    58    75


F =

    0.9183    0.8636    0.8030    0.7780    0.8125

>>
```

Βλέπουμε στο γράφημα 1 τα επιλεγμένα χαρακτηριστικά από το spider βλέπουμε πως όντως τα χαρακτηριστικά είναι όμοια μεταξύ τους και πως γίνεται ο χωρισμός σε δυο κατηγορίες (1 μέχρι 50 και 51 μέχρι 150). Και στο γράφημα 2 βλέπουμε τα επιλεγμένα χαρακτηριστικά από τον δικό μας αλγόριθμο. Αν βγάλουμε από το γράφημα 2 το χαρακτηριστικό 14 επειδή έχει μεγάλες τιμές θα έχουμε μια καλύτερη απεικόνιση στο γράφημα 3 και θα δούμε πως τα χαρακτηριστικά δεν έχουν μεγάλη ομοιότητα και είναι χωρισμένα καλά σε κατηγορίες.

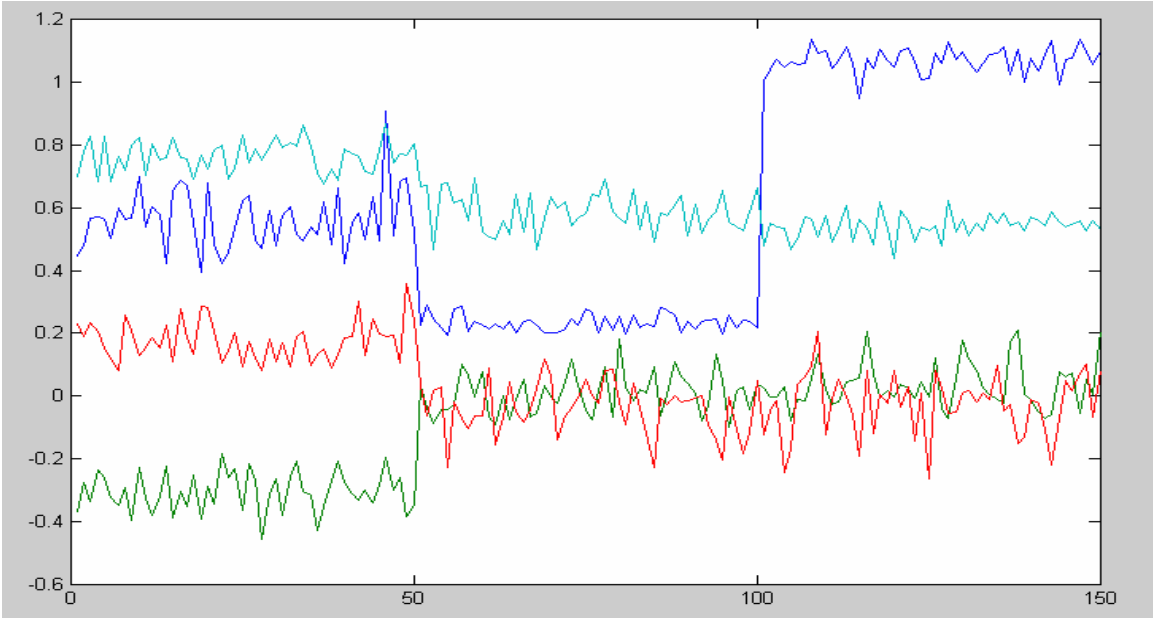
Γράφημα 1: Τιμές των επιλεγμένων χαρακτηριστικών φ14(μπλε),φ54(πράσινο), φ26(κόκκινο), φ6(κυανό),φ32(μοβ) από το Spider, σε γράφημα.



Γράφημα 2: Τιμές των επιλεγμένων χαρακτηριστικών από την δική μας εφαρμογή φ54(μπλε),φ227(πράσινο), φ14(κόκκινο), φ58(κυανό),φ75(μοβ), σε γράφημα.



Γράφημα 3: Οι τιμές από το γράφημα της εικόνας 11.ι χωρίς το χαρακτηριστικό 14, φ54(μπλε),φ227(πράσινο), φ58(κόκκινο),φ75(κυανό).



3.15. Συμπεράσματα

Με την εφαρμογή που υλοποιήθηκε στα πλαίσια αυτής της πτυχιακής εργασίας, δίνεται η δυνατότητα στο χρήστη να κατανοήσει πως λειτουργεί η επιλογή χαρακτηριστικών με φίλτρο αμοιβαίας πληροφορίας. Και μέσω της εφαρμογής αυτής δίνουμε την δυνατότητα σε οποιονδήποτε χρήστη να εκτελέσει τους αλγορίθμους στέλνοντας τα δεδομένα στον

server και λαμβάνοντος τα αποτελέσματα από αυτόν χωρίς ο ίδιος να έχει Matlab εγκατεστημένο στον υπολογιστή του και χωρίς να έχει τις απαραίτητες μαθηματικές γνώσεις. Την εφαρμογή μας μπορούν να την χρησιμοποιήσουν τόσοι χρήστες ταυτόχρονα (online) όσοι είναι οι πύρινες του επεξεργαστή του server επειδή ο server αφιερώνει έναν πύρινα σε κάθε νέα εκτέλεση του Matlab και στο τέλος τερματίζει το ανοιχτό παράθυρο Matlab.

Έτσι ένας οποιοσδήποτε χρήστης που έχει πρόσβαση στο διαδίκτυο θα μπορεί να επιλέξει κάποιον μικρό αριθμό βέλτιστων χαρακτηριστικών από έναν πίνακα δεδομένων που μπορεί να έχει εκατοντάδες χαρακτηριστικά γνωρίσματα.

Η εφαρμογή μας είναι αποτελεσματική καθώς μας παρέχει τα ίδια έως και καλύτερα αποτελέσματα από το spider και ιδία με την σελίδα του Hanchuan Peng. Οι τρεις αλγόριθμοι που παρουσιάσαμε στα πλαίσια αυτής της πτυχιακής εργασίας θα μπορούσαν να χρησιμοποιηθούν για την εκπαίδευση νευρωνικών δικτύων, στην ψηφιακή επεξεργασία εικόνας, σε αρχεία κειμένων(text document), στην ιατρική όπως για την γνωμάτευση και λήψη απόφασης ενός ιατρού στο αν πρέπει η όχι να γίνει μια κρίσιμη επέμβαση σε έναν ασθενή και σε άλλους τομείς που έχουμε να κάνουμε με μεγάλη διάσταση δεδομένων και υπάρχει η ανάγκη να τα ελαττώσουμε σημαντικά. Η εφαρμογή μας μπορεί να επεκταθεί αργότερα εισάγοντας και άλλους μεθόδους.

4. Πίνακες

4.1. Πίνακας εξισώσεων

Παράγοντας διαχωριστικότητας	*εξ. 1*	*14*
Παράγοντας συσχέτισης	*εξ. 2*	*14*
Η μέγιστη μέση πληροφορία ανά μεταβλητή	*εξ. 3*	*197*
Μέση ποσότητα πληροφορίας.	*εξ. 4*	*19*
Ποσότητα πληροφορίας ενός συνδυασμού δύο τυχαίων μεταβλητών	*εξ. 5*	*208*
Κατανομή πιθανοτήτων ενός συνδυασμού δύο τυχαίων μεταβλητών	*εξ. 6*	*208*
Από κοινού εντροπία (joint entropy) H(X,Y)	*εξ. 7*	*208*
Υπό Συνθήκη Εντροπία (Conditional entropy)	*εξ. 8*	*208*
Αμοιβαία Πληροφορία τύπος με εντροπίες	*εξ. 9*	*219*
Αμοιβαία Πληροφορία μετά από πράξεις	*εξ. 10*	*219*
Αμοιβαία πληροφορία μεταξύ ενός θορύβου και του περιβάλλοντος	*εξ. 11*	*20*
Αμοιβαία πληροφορία μεταξύ ενός τυχαίου μεταβλητού και του ίδιου	*εξ. 12*	*22*
Αμοιβαία πληροφορία είναι συμμετρική	*εξ. 13*	*22*

Υπό Συνθήκη Αμοιβαία Πληροφορία	*εξ. 14 ..... 231*
Αλγόριθμος Υπό Συνθήκης Αμοιβαίας Πληροφορίας	*εξ. 15 ..... 242*
Ανώτατη εξάρτηση(Max-Dependency)	*εξ. 16 ..... 264*
Μέγιστη Σχετικότητα (Max-Relevance)	*εξ. 17..... 275*
Ελάχιστος Πλεονασμός (Min-Redundancy)	*εξ. 18..... 275*
Αναλυτική μορφή ελάχιστος πλεονασμός μέγιστη σχετικότητα	*εξ. 19..... 286*
MID(Αμοιβαία πληροφορία διαφορά)	*εξ. 20 ..... 286*
MIQ(Αμοιβαία πληροφορία πηλίκο)	*εξ. 21..... 286*
Τυπος κανονικοποίησης	*εξ. 22 ..... 586*

4.2. Πίνακας εικόνων

Εικόνα 1: εικονική αναπαράσταση της αμοιβαίας πληροφορίας όπου βλέπουμε πως η αμοιβαία πληροφορία είναι η κοινή πληροφορία ανάμεσα στην εντροπία H(X) και H(Y). 21
Εικόνα 2: Εικονική αναπαράσταση της υπό συνθήκη αμοιβαία πληροφορίας. 23
Εικόνα 3: Η αρχική σελίδα της εφαρμογής μας. 32
Εικόνα 4: Σελίδα εισαγωγής παραμέτρων και δεδομένων. Είναι παρόμοια και στους τρεις αλγορίθμους..... 33
Εικόνα 5: Σχεδιάγραμμα Της εφαρμογής. Αναπαράσταση επικοινωνίας χρήστη με server.. 33
Εικόνα 6: Αρχική σελίδα του Hanchuan Peng. 62
Εικόνα 7: Αποτελέσματα Επιλογής 5 χαρακτηριστικών γνωρισμάτων από τον αλγόριθμο mRmR διαίρεση από την σελίδα του Hanchuan Peng. 62
Εικόνα 8: Αποτελέσματα Επιλογής 5 χαρακτηριστικών γνωρισμάτων από τον αλγόριθμο mRmR διαφορά από την σελίδα του Hanchuan Peng. 63
Εικόνα 9: α) Εκτέλεση του Spider και επιλογή των 10 βέλτιστων χαρακτηριστικών από αυτόν(10 γιατί παρακάτω θα επιλέξουμε κάποια χαρακτηριστικά που είναι μέσα στην δεκάδα). β) Η επιλογή των 5 καλύτερων χαρακτηριστικών από την δική μας εφαρμογή με το ίδιο σετ δεδομένων με την εικόνα 6.α..... 65
Εικόνα 10: α)Λίστα με τα διαθέσιμους οδηγούς και η επιλογή του κατάλληλου. β)Τα πεδία με τις απαραίτητες ρυθμίσεις για την δημιουργία του οδηγού ODBC..... 73

4.3. Πίνακας πινάκων

Πίνακας 1: α)Μικρός δοκιμαστικός Πινάκας δεδομένων χωρισμένος σε χωριά (bins). β)Τα αποτελέσματα με την επιλογή χαρακτηριστικών με τον αλγόριθμο Αμοιβαία πληροφορία πηλίκο (MIQ). 35
Πίνακας 2: α) Αμοιβαία πληροφορία διαφορά (MID). β) υπό συνθήκη Αμοιβαία πληροφορία (CMI)..... 35
Πίνακας 3: α) Μικρός δοκιμαστικός Πινάκας δεδομένων μη κανονικοποιημένος και δεν είναι χωρισμένος σε χωριά. β)Κανονικοποιηση του πίνακα 3.α και τοποθέτηση των τιμών σε 10 χωρία (bins). Δηλαδή από τιμές 1 μέχρι 100 τοποθέτιουνται στο χωρίο 1, από 101 μέχρι 200 τοποθέτιουνται στο χωρίο 2, κτλ. 36

Πίνακας 4: α)Αποτελέσματα από την εκτέλεση του πρώτου και δευτέρου αλγορίθμου με δεδομένα από τον πίνακα 3.β β) Αποτελέσματα από την εκτέλεση του τρίτου αλγορίθμου με δεδομένα από τον πίνακα 3.β 37

Πίνακας 5: Παράδειγμα Πίνακα με αποτελέσματα ύστερα από εκτέλεση κάποιου αλγορίθμου. 42

Πίνακας 6: α) Φόρμα όπου δίνουμε τον κωδικό παλιότερων αποτελεσμάτων. β)Εμφάνιση παλιότερων αποτελεσμάτων. 43

Πίνακας 7: α) Πίνακας δεδομένων με 150 γραμμές και 351 στήλες (ορατές στην εικόνα 32 γραμμές και 18 στήλες) όπου οι τιμές δεν είναι κανονικοποιημένες. β)Ο πίνακας 6.α κανονικοποιημένος 58

Πίνακας 8: Ο πίνακας 6.β με τοποθετημένες τις τιμές σε χωρία..... 60

Πίνακας 9: Αποτελέσματα Επιλογής 5 χαρακτηριστικών γνωρισμάτων από τον αλγόριθμο mRmR διαίρεση από την δική μας εφαρμογή με το ίδιο σετ δεδομένων με την εικόνα 4.α . 63

Πίνακας 10: Αποτελέσματα Επιλογής 5 χαρακτηριστικών γνωρισμάτων από τον αλγόριθμο mRmR διαφορά από την δική μας εφαρμογή με το ίδιο σετ δεδομένων με την εικόνα 5.α.. 64

4.4. Πίνακας γραφημάτων

Γράφημα 1: α)Τιμές των επιλεγμένων χαρακτηριστικών φ14(μπλε),φ54(πράσινο), φ26(κόκκινο), φ6(κυανό),φ32(μοβ) από το Spider, σε γράφημα. 664

Γράφημα 2: α)Τιμές των επιλεγμένων χαρακτηριστικών από την δική μας εφαρμογή φ54(μπλε),φ227(πράσινο), φ14(κόκκινο), φ58(κυανό),φ75(μοβ), σε γράφημα. 675

Γράφημα 3: α)Οι τιμές από το γράφημα της εικόνας 11.ι χωρίς το χαρακτηριστικό 14, φ54(μπλε),φ227(πράσινο), φ58(κόκκινο),φ75(κυανό). 675

5. Παράρτημα Α

5.1. Matlab

Το όνομα Matlab προέρχεται από τις λέξεις «Matrix laboratory» . Επινοήθηκε στα τέλη του 1970 από τον Cleve Moler ο οποίος ήθελε να δώσει στους μαθητές του πρόσβαση στα πακέτα LINPACK και EISPACK χωρίς να χρειάζεται η εκμάθηση της Fortran. Το Matlab κέρδισε δημοτικότητα κυρίως από στόμα σε στόμα επειδή δεν διατίθενται επισήμως. Στην δεκαετία του 1980, το Matlab γράφτηκε σε γλώσσα C με περισσότερη λειτουργικότητα όπως ρουτίνες για παραγωγή γραφικών παραστάσεων. Η εταιρεία Mathworks, Inc. που ιδρύθηκε το 1984 είναι στο παρόν υπεύθυνη για την ανάπτυξη και υποστήριξη του Matlab.

Τα πλεονεκτήματα του Matlab: Είναι σχετικά εύκολο στην εκμάθηση. Ο κώδικας είναι βελτιστοποιημένος ώστε να είναι σχετικά γρήγορος στον υπολογισμό πινάκων. Μπορεί να συμπεριφερθεί ως γλώσσα προγραμματισμού ή ως μία αριθμομηχανή. Είναι μία interpreted γλώσσα προγραμματισμού, δηλαδή κάθε εντολή μεταφράζετε την προκειμένη στιγμή σε γλώσσα μηχανής και έτσι τα λάθη είναι ευκολότερο να διορθωθούν.

Μειονεκτήματα του Matlab: Δεν είναι γλώσσα προγραμματισμού γενικής χρήσης. Είναι μία interpreted γλώσσα προγραμματισμού (καθιστώντας το για παράδειγμα πιο αργό από μία γλώσσα προγραμματισμού όπως η C++ που είναι compiled). Είναι σχεδιασμένο για επιστημονικούς υπολογισμούς και δεν είναι κατάλληλο για άλλες χρήσεις.

5.2. Βάσεις Δεδομένων

Ένας από τους όρους που οι περισσότεροι άνθρωποι έχουν συνηθίσει να ακούνε είτε στο χώρο εργασίας τους ή κατά την περιήγηση στο διαδίκτυο είναι αυτός της βάσης δεδομένων. Βάση δεδομένων είναι μία δομημένη συλλογή εγγραφών που είναι αποθηκευμένη σε ένα υπολογιστή. Οι βάσεις δεδομένων όμως δεν υφίστανται απόλυτα μέσα στα όρια του υπολογιστή. Παραδείγματα τέτοιων βάσεων δεδομένων αποτελούν ένας τηλεφωνικός κατάλογος ή ένα λεξικό.

Οι βάσεις δεδομένων σε υπολογιστές είναι συνήθως οργανωμένες σε έναν ή περισσότερους πίνακες. Ένας πίνακας αποτελείται από ένα αριθμό στηλών και γραμμών. Οι βάσεις που μπορούν να αποθηκευτούν σε ένα υπολογιστή, μπορούν και να διαχειριστούν με κάποιο πρόγραμμα. Αυτά τα προγράμματα καλούνται σύστημα διαχείρισης βάσης δεδομένων (DBMS - database management system). Ως το εργαλείο που χρησιμοποιείται κυρίως για τη διαχείριση των βάσεων δεδομένων, υπάρχουν πολλά συστήματα διαχείρισης βάσης δεδομένων διαθέσιμα στην αγορά. Μερικά δημοφιλή παραδείγματα αποτελούν τα MySql, Microsoft Access, FileMaker και Oracle. Όλα αυτά τα προϊόντα παρέχουν τη δυνατότητα δημιουργίας μίας σειράς δικαιωμάτων που μπορούν να αποδοθούν σε ένα συγκεκριμένο χρήστη. Είναι δυνατό να οριστεί ένας ή περισσότεροι διαχειριστές που μπορούν να παρέχουν στους άλλους χρήστες διάφορα δικαιώματα διαχείρισης. Αυτή η ευελιξία καθιστά το έργο της επίβλεψης ενός συστήματος με χρήση ενός συστήματος διαχείρισης βάσης δεδομένων κάτι που μπορεί να ελέγχεται κεντρικά ή να είναι διαμοιρασμένο σε διάφορους χρήστες. Τα δεδομένα που είναι καταχωρημένα στη βάση μπορούμε να τα επεξεργαστούμε με μία γλώσσα που ονομάζεται SQL.

Τα αρχικά SQL σημαίνουν Structured Query Language και τα οποία στην ελληνική ορολογία μεταφράζονται ως δομημένη γλώσσα Ερωτημάτων. Συνοπτικά με την SQL μπορούμε να έχουμε πρόσβαση σε μια βάση δεδομένων, να εκτελούμε αναζητήσεις με κριτήρια, να εισάγουμε νέα δεδομένα, να διαγράψουμε υπάρχοντα δεδομένα και να κάνουμε αλλαγές σε υπάρχοντα δεδομένα. Η SQL άρχισε ως δημιουργία της IBM από τους Andrew Richardson, Donald C. Messerly και Raymond F. Boyce στις αρχές του 1970. Αρχικά ήταν γνωστή ως SEQUEL και αργότερα τυποποιήθηκε από το αμερικανικό εθνικό ίδρυμα προτύπων (ANSI) και το διεθνή οργανισμό για την τυποποίηση (ISO). Υπάρχει μεγάλος αριθμός διαφορετικών εκδόσεων της γλώσσας SQL αλλά κάποιες βασικές εντολές συναντώνται σε όλες τις βάσεις δεδομένων.

Στα πλαίσια της συγκεκριμένης πτυχιακής εργασίας έγινε χρήση του συστήματος διαχείρισης βάσης δεδομένων MySql Μέσω του πακέτου XAMPP (Basis Package) version 1.7.3 .

5.3. Ο οδηγός ODBC

Τα αρχικά προέρχονται από τις λέξεις Open DataBase Connectivity. Είναι ένα πρότυπο πρόσβασης βάσης δεδομένων που αναπτύχθηκε από την ομάδα SQL Access το 1992. Η

ομάδα SQL Access ήταν μία ομάδα εταιρειών λογισμικού που ιδρύθηκε το 1989 για τον καθορισμό και την προώθηση προτύπων βάσης δεδομένων για τη φορητότητα και τη βέλτιστη λειτουργικότητα του. Ο στόχος του προτύπου ODBC είναι να καθιστά δυνατό να μπορούν να προσπελαστούν οποιαδήποτε δεδομένα από οποιαδήποτε εφαρμογή ανεξάρτητα με τη σύστημα διαχείρισης βάσης δεδομένων διαχειρίζεται τα δεδομένα (DBMS) .

Το ODBC καταφέρνει κάτι τέτοιο εισάγοντας ένα ενδιάμεσο επίπεδο που καλείται database driver, μεταξύ της εφαρμογής και του DBMS. Ο σκοπός αυτό του επιπέδου είναι να μεταφράζει τα δεδομένα σε εντολές που ο DBMS να καταλαβαίνει. Για να λειτουργήσει κάτι τέτοιο πρέπει και η εφαρμογή και ο DBMS να είναι "συμβατά" με το ODBC, που σημαίνει ότι η εφαρμογή πρέπει να είναι ικανή να δίνει εντολές ODBC και ο DBMS να μπορεί να ανταποκριθεί σε αυτές. Με λίγα λόγια είναι η γέφυρα επικοινωνίας ανάμεσα στον DBMS και εφαρμογής που δουλεύει με πίνακες από βάση δεδομένων.

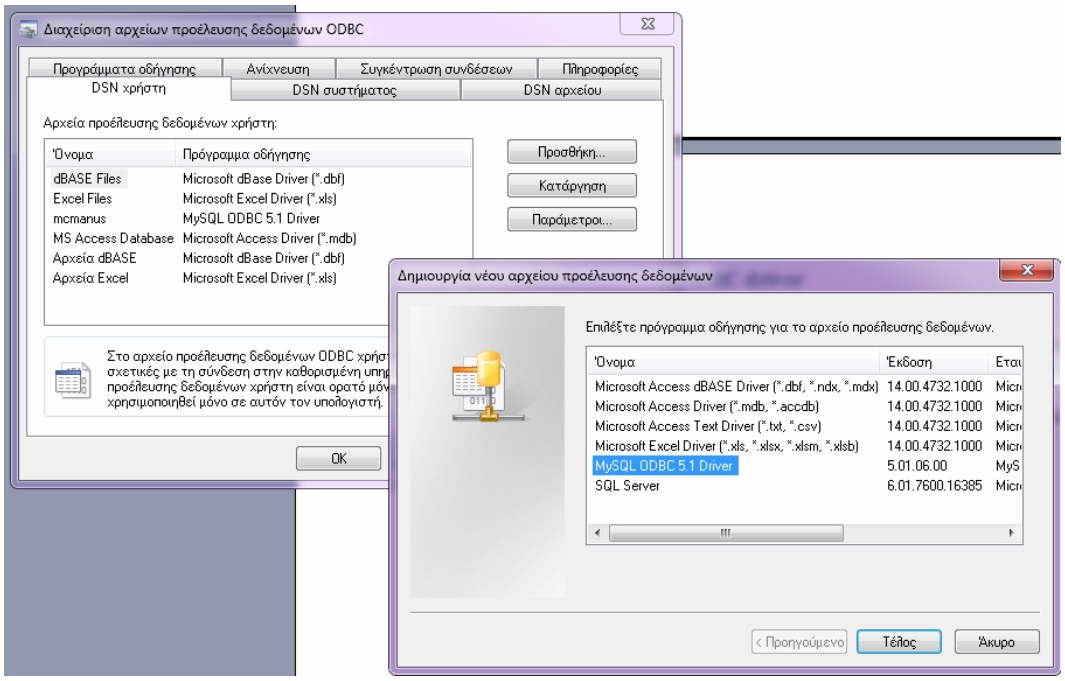
5.4. Δημιουργία Και Ρύθμιση ODBC driver

Εφόσον προμηθευτούμε από το διαδίκτυο ή τον κατασκευαστή DBMS το αρχείο εγκατάστασης το οποίο διατίθεται δωρεάν και το κάνουμε εγκατάσταση πηγαίνουμε στον πίνακα έλεγχου, Εργαλεία διαχείρισης→Αρχεία προέλευσης δεδομένων (ODBC) → στην καρτέλα DSN χρήστη και επιλέγουμε προσθήκη από την λίστα με τους διαθέσιμους drivers, επιλέγουμε MySql ODBC 5.1 driver όπως φαίνεται στην εικόνα 10.α και δίνουμε όνομα driver, περιγραφή, τον server που θα γίνεται η σύνδεση, όνομα χρηστή, κωδικός πρόσβασης αν έχει ο server και τέλος αν δώσαμε σωστά στοιχεία θα μας δίνετε η δυνατότητα να επιλέγουμε και την βάση που θα γίνεται σύνδεση από μια λίστα διαθέσιμων βάσεων όπως φαίνεται στην εικόνα 10.β και επιλέγουμε τέλος.

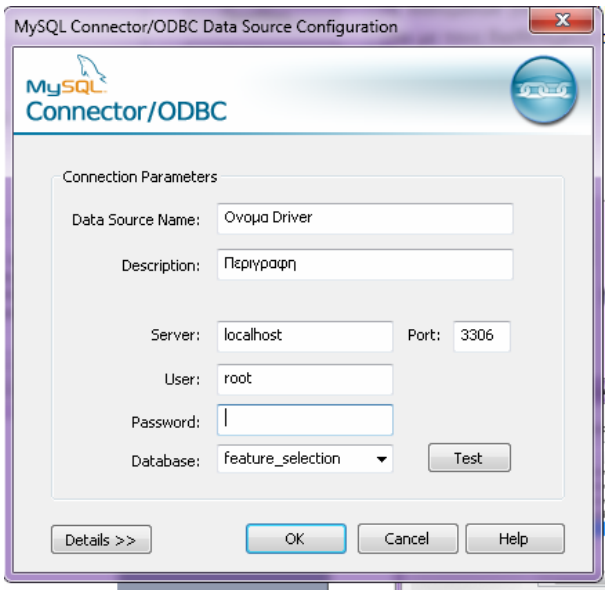
Στην εργασία μας έχουμε βάλει όνομα οδηγού mcmanus , server localhost, χρήστης root χωρίς κωδικό πρόσβασης και το έχουμε σύνδεση με την βάση feature_selection.

Εικόνα 10: α)Λίστα με τα διαθέσιμους οδηγούς και η επιλογή του κατάλληλου. β)Τα πεδία με τις απαραίτητες ρυθμίσεις για την δημιουργία του οδηγού ODBC.

α)



β)



5.5.Flash

Το Flash αρχείο της ιστοσελίδας έχει δημιουργηθεί στον adobe flash professional με τα εξής βήματα. Επιλέγουμε File -> new -> actionscript3. Αλλάζουμε τις διαστάσεις, χρώμα background και δίνουμε ότι θέλουμε και χριζόμαστε εμείς από το Properties -> size ->edit. Στην περίπτωση μας έχουμε δώσει 960 x 290 pxl και χρώμα πράσινο επειδή και η περιοχή όπου τοποθετούμε το flash έχει αυτές τις διαστάσεις και χρώμα. Τα τέσσερα button είναι ουσιαστικά τέσσερα text τα οποία με f8 τα κάνουμε convert to symbol και επιλέγουμε το

button και δίνουμε ένα όνομα για να ξέρουμε πως θα αναφερθούμε σε αυτό. Τέλος για να μεταφερόμαστε σε άλλη σελίδα με το πάτημα ενός button πατάμε f9 και μας ανοίγει γραμμή εντολών και εκεί γράφουμε τον εξής κώδικα που σημαίνει πως μετά το πάτημα ενός αντίστοιχου button θα μεταφερθούμε στην σελίδα που είναι μέσα σαν παράμετρος στην getURL.

```
on (release)
        {
        getURL("fs_mrmr_miq.html");  }
```

Τέλος πάμε file -> publish settings επιλέγουμε σε τι μορφή θέλουμε αυτό που δημιουργήσαμε , δίνουμε το path και πατάμε publish.

5.6. Αρχεία CSS

Στην σελίδα αυτή κάνουμε εισαγωγή ένα css αρχείο το οποίο περιέχει τις βασικές ρυθμίσεις της σελίδας μας και μας δίνεται η δυνατότητα να εισάγουμε το css αρχείο σε όσες σελίδες θέλουμε ώστε να αλλάξουμε την μορφή και το χρώμα της ιστοσελίδας. Έτσι σε κάθε νέα σελίδα δεν χρειάζεται να γράφουμε κάθε φορά τις απαραίτητες ρυθμίσεις για να πάρει την μορφή που επιθυμούμε αρκεί να κάνουμε εισαγωγή του .CSS στο <head> την εξής γραμμή .

```
<link href="layout.css" rel="stylesheet" type="text/css" />
```

Το CSS αρχείο μας περιέχει τέσσερις ομάδες εντολών. Τα τρία είναι div δηλαδή αφόρα ρυθμίσεις για μια συγκεκριμένη περιοχή που θα καταλαμβάνει και το τέταρτο είναι η body δηλαδή γενικές ρυθμίσεις για όλη την html σελίδα.

```
Body {
  padding:0;
  margin:0;
  color:#fff;
  background-image:url(images/main_background.gif);
  background-repeat:repeat-x;
  background-color:#FFF;
}
```

Στην body δίνουμε περιθώρια 0,χρωμα γραφής άσπρο, background μια εικόνα η οποία επαναλαμβάνετε στον άξονα χ με την εντολή background-repeat:repeat-x;

Και background-color:#FFF; δίνουμε χρώμα background άσπρο.

```
#container {
  width:960px;
  margin:0 auto;
  padding-top:145px;
}
```

```
#middle {
  width:960px;
  height:440px;
  float:left;
}
```

```
#middle_inside {
  width:930px;
  height:290px;
  float:left;
  background-color:#fff;
  margin:15px 0 0 0;
  padding:5px 0 5px 5px;}
```

Οι ιδιότητες των παραπάνω ρυθμίσεων είναι απλά οι διαστάσεις τους , float:left; Είναι η τοποθέτηση στα αριστερά, padding:5px 0 5px 5px; Είναι τα γύρο περιθώρια.

Βιβλιογραφία

[1] "Feature Selection in Data Mining", Yong Seog Kim, W. Nick Street, and Filippo Menczer, University of Iowa USA 2000, (http://dollar.biz.uiowa.edu/~street/research /dmoc.pdf).

[2] "Ανίχνευση και Εξαγωγή του Οδικού δικτύου με Αντικειμενοστραφή Ανάλυση Υπερφασματικών Εικόνων Hyperion και Πολυφασματικών Εικόνων Landsat TM και σύγκριση των αποτελεσμάτων", Πέντζιος Βασίλειος, Διπλωματική εργασία από το Εθνικό Μετσόβιο Πολυτεχνείο Σχολή Αγρονόμων & Τοπογράφων Μηχανικών Εργαστήριο Τηλεπισκόπησης, Αθήνα 2009, (http://dspace.lib.ntua.gr/bitstream/ 123456789/3165/3/rentziosv_ecognition.pdf)

[3]"Feature selection by mutual information ", Spider Extra Algorithms, (http://www.kyb.mpg.de/bs/people/spider/)

[4] "Λειτουργική ταξινόμηση πρωτεϊνών με δεδομένα φασματογραφίας μάζας και διερεύνηση της συχνότητας ταυτοποίησής τους σε πρωτεϊνική ανάλυση", Πουγιούκος Παναγιώτης, B.Sc Τεχνολογία Ιατρικών Οργάνων M.Sc. Ιατρική Φυσική , Πάτρα Μάιος 2009 (http://nemertes.lis.upatras.gr/dspace/bitstream/123456789/2486/1/Bougioukos%20Panos%20Phd_revised_1.pdf)

[5] "Αναγνώριση Προτύπων - Νευρωνικά Δίκτυα", Δρ. Χαράλαμπος Π. Στρουθόπουλος, ΤΕΙ Σερρών Τομέας Υπολογιστικών Τεχνικών & Συστημάτων Σέρρες 2003, (http://teiserron.gr/index.php?action=dlattach;topic=558.0;attach=780)

[6] "Θεωρία πιθανοτήτων Από τη Βικιπαίδεια, την ελεύθερη εγκυκλοπαίδεια", (http://el.wikipedia.org/wiki/%CE%98%CE%B5%CF%89%CF%81%CE%AF%CE%B1_%CF%80%CE%B9%CE%B8%CE%B1%CE%BD%CE%BF%CF%84%CE%AE%CF%84%CF%89%CE%BD)

[7] "Θεωρία Πληροφορίας και Κωδικοποίηση", Βασίλειος Ζορκάδης, ΕΛΛΗΝΙΚΟ ΑΝΟΙΚΤΟ ΠΑΝΕΠΙΣΤΗΜΙΟ Σχολή Θετικών Επιστημών και Τεχνολογίας, Πάτρα 2002, (http://www.scribd.com/doc/900534/-)

[8] " Κλοντ Σάνον Από τη Βικιπαίδεια, την ελεύθερη εγκυκλοπαίδεια" (http://el.wikipedia.org/wiki/%CE%9A%CE%BB%CE%BF%CE%BD%CF%84_%CE%A3%CE%AC%CE%BD%CE%BF%CE%BD)

[9] "Mutual information – Wikipedia", (http://en.wikipedia.org/wiki/Mutual_information)

[10] "Feature selection with conditional mutual", G. Wang and F. H. Lochovsky. Qiang Yang, CIKM'04 Washington, DC USA., November 8–13, 2004, (http://www.cs.ust.hk/~wanggang/research/cikm04_CMIM.pdf)

[11] "Conditional Mutual information – Wikipedia", (http://en.wikipedia.org/wiki/ Conditional_mutual_information)

[12] "Feature Selection with Conditional Mutual Information Maximinin Text Categorization", Department of Computer Science, Hong Kong University of Science and Technology, Slides (http://www.cs.ust.hk/~wanggang/research/cikm04_slides.pdf)

[13] "Feature selection based on mutual information: criteria of max-dependency, max-relevance, and min-redundancy,", Hanchuan Peng, Fuhui Long, and Chris Ding, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 27, No. 8, pp.1226-1238, 2005. (http://penglab.janelia.org/papersall/docpdf/2005_TPAMI_FeaSel.pdf)

[14] "A Minimum Redundancy Feature Selection from Microarray Gene Expression Data" Hanchuan Peng, Chris Ding, University of California, Berkeley, CA, 94720, USA, Proceedings of the Computational Systems Bioinformatics (CSB'03) (http://penglab.janelia.org/papersall/docpdf/2003_CSB_feasel.pdf)

[15] "Minimum redundancy feature selection from microarray gene expression data," Hanchuan Peng, Chris Ding, Journal of Bioinformatics and Computational Biology, Vol. 3, No. 2, pp.185-205, 2005. (http://penglab.janelia.org/papersall/docpdf/2004_JBCB_feasel-04-06-15.pdf)