



**ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ**  
**ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ**  
**ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΜΕ ΕΦΑΡΜΟΓΕΣ**  
**ΣΤΗ ΒΙΟΪΑΤΡΙΚΗ**

**Εφαρμογή Διαχείρισης Πληροφορίας σε Πηγές Νομικών Κειμένων**

**Σπυρίδων Κάτσαμπος**

**ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ**

**Υπεύθυνος**

**Ιωάννης Αναγνωστόπουλος**

**Επίκουρος Καθηγητής**

**Λαμία, 9/9 έτος 2015**





**ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ**  
**ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ**  
**ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΜΕ ΕΦΑΡΜΟΓΕΣ ΣΤΗ**  
**ΒΙΟΪΑΤΡΙΚΗ**

**Εφαρμογή Διαχείρισης Πληροφορίας σε Πηγές Νομικών Κειμένων**

**Σπυρίδων Κάτσαμπος**

**ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ**

**Επιβλέπων**

**Ιωάννης Αναγνωστόπουλος**

**Επίκουρος Καθηγητής**

**Λαμία, 9/9 έτος 2015**

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

1. Δεν παραθέτω κομμάτια βιβλίων ή άρθρων ή εργασιών άλλων αυτολεξεί **χωρίς να τα περικλείω σε εισαγωγικά** και χωρίς να αναφέρω το συγγραφέα, τη χρονολογία, τη σελίδα. Η αυτολεξεί παράθεση χωρίς εισαγωγικά χωρίς αναφορά στην πηγή, είναι λογοκλοπή. Πέραν της αυτολεξεί παράθεσης, λογοκλοπή θεωρείται και η παράφραση εδαφίων από έργα άλλων, συμπεριλαμβανομένων και έργων συμφοιτητών μου, καθώς και η παράθεση στοιχείων που άλλοι συνέλεξαν ή επεξεργάστηκαν, χωρίς αναφορά στην πηγή. Αναφέρω πάντοτε με πληρότητα την πηγή κάτω από τον πίνακα ή σχέδιο, όπως στα παραθέματα.
2. Δέχομαι ότι η αυτολεξεί **παράθεση χωρίς εισαγωγικά**, ακόμα κι αν συνοδεύεται από αναφορά στην πηγή σε κάποιο άλλο σημείο του κειμένου ή στο τέλος του, είναι αντιγραφή. Η αναφορά στην πηγή στο τέλος π.χ. μιας παραγράφου ή μιας σελίδας, δεν δικαιολογεί συρραφή εδαφίων έργου άλλου συγγραφέα, έστω και παραφρασμένων, και παρουσίασή τους ως δική μου εργασία.
3. Δέχομαι ότι υπάρχει επίσης περιορισμός στο μέγεθος και στη συχνότητα των παραθεμάτων που μπορώ να εντάξω στην εργασία μου εντός εισαγωγικών. Κάθε μεγάλο παράθεμα (π.χ. σε πίνακα ή πλαίσιο, κλπ), προϋποθέτει ειδικές ρυθμίσεις, και όταν δημοσιεύεται προϋποθέτει την άδεια του συγγραφέα ή του εκδότη. Το ίδιο και οι πίνακες και τα σχέδια.
4. Δέχομαι όλες τις συνέπειες σε περίπτωση λογοκλοπής ή αντιγραφής.

Ημερομηνία: ...../...../20.....

Ο – Η Δηλ.

(Υπογραφή)

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

**Εφαρμογή Διαχείρισης Πληροφορίας σε Πηγές Νομικών Κειμένων  
Σπυρίδων Κάτσαμπος**

**Τριμελής Επιτροπή:**

- Ιωάννης Αναγνωστόπουλος, Επιβλέπων, Επίκουρος Καθηγητής, Τμήμα Πληροφορικής με εφαρμογές στη Βιοϊατρική, Πανεπιστήμιο Θεσσαλίας
- Βασίλης Πλαγιανάκος, Αναπληρωτής Καθηγητής, Τμήμα Πληροφορικής με εφαρμογές στη Βιοϊατρική, Πανεπιστήμιο Θεσσαλίας
- Αθανάσιος Λουκόπουλος, Λέκτορας, Τμήμα Πληροφορικής με εφαρμογές στη Βιοϊατρική, Πανεπιστήμιο Θεσσαλίας

## ΠΕΡΙΛΗΨΗ

---

Αντικείμενο της πτυχιακής εργασίας είναι η διαχείριση της πληροφορίας σε πηγές νομικών κειμένων. Ειδικότερα, η διαδικασία της διαχείρισης στην παρούσα εργασία περιλαμβάνει την εξαγωγή κεφαλίδας/περίληψης, την επισήμανση των νομικών αναφορών και την κατηγοριοποίηση των νομικών αποφάσεων που εξήχθησαν από τον Άρειο Πάγο. Με αυτό τον τρόπο, δημιουργήθηκε μια εφαρμογή που να περιλαμβάνει λειτουργίες προς διευκόλυνση των χρηστών ως προς την αναζήτηση και εύρεση συγκεκριμένων χωρίων.

Αρχικά, χρησιμοποιήθηκαν νομικά κείμενα του ποινικού, πολιτικού και του τμήματος νόμου 3068/2002 για την εξαγωγή περιλήψεων και εντοπίστηκαν κάποια βασικά σημεία μέσα στα ίδια τα κείμενα τα οποία έχουν αυστηρά καθορισμένη δομή και ακολουθούν συγκεκριμένες και επαναλαμβανόμενες νόρμες, γεγονός που βοήθησε στην γρήγορη και αυτοματοποιημένη εξαγωγή των ζητούμενων χαρακτηριστικών. Για την επισήμανση αναφορών ομαδοποιήθηκαν σε κατηγορίες τα στοιχεία σύμφωνα με τη θεματολογία που αναφέρονται. Έτσι δημιουργήθηκε μία αυτοματοποιημένη διαδικασία που πραγματοποιούσε τον εντοπισμό των αναφορών κάθε κατηγορίας και τις επισήμανε με διαφορετικό χρώμα. Ακόμα, παράγαγε ένα διάγραμμα του οποίου κάθε στοιχείο, δήλωνε την συχνότητα εμφάνισης κάθε κατηγορίας. Έπειτα τα διαγράμματα χρησιμοποιήθηκαν ως είσοδοι για διάφορα μοντέλα νευρωνικών δικτύων. Πιο αναλυτικά τα δεδομένα χρησιμοποιήθηκαν για την εκπαίδευση και την δοκιμή των μοντέλων που χρησιμοποιήθηκαν. Τα μοντέλα που δοκιμάστηκαν είναι τα εξής: MultiLayerPerceptron, RBFNetwork, SMO και Simple Logistic.

Τέλος, έγινε εκτίμηση της αποτελεσματικότητας του συστήματος αναλύοντας τις μετρικές, «ανάκληση» και «ακρίβεια» για κάθε μία από τις τρεις λειτουργίες της εφαρμογής ξεχωριστά. Μ' αυτόν τον καταλήξαμε σε διάφορα συμπεράσματα για την απόδοση του συστήματος κατά την εξαγωγή περίληψης, τον εντοπισμό των αναφορών και την κατηγοριοποίηση κειμένων.

### Λέξεις Κλειδιά

Νομικά Κείμενα, Εξόρυξη Κειμένου, Νομικές Αναφορές, Εξαγωγή Περίληψης, Επισήμανση Αναφορών, Ταξινόμηση Κειμένων, Μηχανική Μάθηση, MultiLayerPerceptron, RBFNetwork, SMO, Simple Logistic, Μετρικές Εκτίμησης Απόδοσης, Ανάκληση, Ακρίβεια.

# ABSTRACT

---

The subject of this thesis is the management of information in sources of legal documents. In particular, the process of information management in this body of work involves extraction header summarisation, labeling of legal references and categorization of legal decisions drawn from the Greek Supreme Court. To assist with this process, an application was created which includes features that facilitate users' searching activities in order to find specific passages.

Primarily, the texts used for the summarization were from criminal and civil law and a section from Law 3068/2002. Within those texts, some key points which have a strictly defined structure and follow specific and repetitive norms were identified. Those key points help in the fast and automated export of the requested features. In order to highlight the references they were grouped into categories of data, according to their identifying themes. This created an automated process that detected reports from each category and highlighted them with different colours. Furthermore, it produced a vector, its value indicating the frequency of each category. The vectors were then used as inputs for various neural network models. More detailed data was used for training and testing these models. The models tested include MultiLayerPerceptron, RBFNetwork, SMO and Simple Logistic.

Finally, there was an assessment of the effectiveness of the system by individually analyzing the metrics, "recall" and "precision" of each of the three functions of the application. As such we have reached several conclusions about the performance of the export summary system, the identification of references and text categorization.

## Key Words

Legal Texts, Text Mining, Legal Reports, Header Summarisation, Petitions Labelling, Texts Classification, Machine Learning, MultiLayerPerceptron, RBFNetwork, SMO, Simple Logistic, Measuring Performance Assessment, Recall, Precision.

# ΠΕΡΙΕΧΟΜΕΝΑ

---

<b>ΠΕΡΙΛΗΨΗ.....</b>	<b>6</b>
<i>Λέξεις Κλειδιά</i> .....	6
<b>ABSTRACT.....</b>	<b>7</b>
<i>Key Words</i> .....	7
<b>ΠΕΡΙΕΧΟΜΕΝΑ.....</b>	<b>8</b>
<b>ΚΑΤΑΛΟΓΟΣ ΕΙΚΟΝΩΝ .....</b>	<b>10</b>
<b>ΚΑΤΑΛΟΓΟΣ ΣΧΗΜΑΤΩΝ.....</b>	<b>11</b>
<b>ΚΑΤΑΛΟΓΟΣ ΠΙΝΑΚΩΝ .....</b>	<b>11</b>
<b>ΠΡΟΛΟΓΟΣ.....</b>	<b>12</b>
<b>ΚΕΦΑΛΑΙΟ 1 ΕΙΣΑΓΩΓΗ .....</b>	<b>13</b>
1.1 <i>ΑΝΤΙΚΕΙΜΕΝΟ – ΣΚΟΠΟΣ</i> .....	13
1.2 <i>ΓΕΝΙΚΗ ΕΠΙΣΚΟΠΗΣΗ</i> .....	13
<b>ΚΕΦΑΛΑΙΟ 2 ΔΙΑΧΕΙΡΙΣΗ ΠΛΗΡΟΦΟΡΙΑΣ ΣΕ ΝΟΜΙΚΕΣ ΠΗΓΕΣ .....</b>	<b>15</b>
2.1 <i>ΜΕΘΟΔΟΙ ΕΠΕΞΕΡΓΑΣΙΑΣ ΚΕΙΜΕΝΩΝ</i> .....	15
2.1.1 Εξαγωγή Χαρακτηριστικών (Feature Extraction).....	15
2.1.2 Πλοήγηση με βάση το κείμενο (Text Based Navigation) .....	16
2.1.3 Αναζήτηση και Ανάκτηση (Search and Retrieval).....	16
2.1.4 Κατηγοριοποίηση (Classification) .....	16
2.1.5 Ομαδοποίηση (Clustering).....	17
2.1.6 Περιληπτική παρουσίαση της πληροφορίας (Summarization) .....	18
2.1.7 Γλωσσικός προσδιορισμός (Language Identification) και απόδοση κειμένου στο συγγραφέα.....	18
2.1.8 Συσχετίσεις (Associations).....	18
2.1.9 Απεικόνιση – Οπτικοποίηση (Visualization).....	18
2.2 <i>ΠΕΡΙΓΡΑΦΗ ΤΩΝ ΝΟΜΙΚΩΝ ΠΗΓΩΝ</i> .....	19
2.2.1 Η Δομή των Νομικών Κειμένων.....	19
2.2.2 Εξαγωγή Επικεφαλίδας σε Νομικά Κείμενα.....	21
2.2.3 Επισήμανση Αναφορών σε Νομικά Κείμενα.....	24
2.3 <i>ΜΕΘΟΔΟΙ ΜΗΧΑΝΙΚΗΣ ΜΑΘΗΣΗΣ ΚΑΙ ΤΑΞΙΝΟΜΙΣΗΣ ΚΕΙΜΕΝΩΝ</i> .....	26
2.3.1 Βασικές Έννοιες της Μηχανικής Μάθησης.....	26
2.3.2 Μοντέλο Perceptron Πολλών Στρωμάτων (MultiLayer Perceptron).....	27
2.3.3. Μοντέλο Δικτύου Ακτινικών Συναρτήσεων Βάσης (RBF Network) .....	29
2.3.3 Η μέθοδος Cross-Validation .....	31
<b>ΚΕΦΑΛΑΙΟ 3 ΑΠΟΤΕΛΕΣΜΑΤΑ – ΑΞΙΟΛΟΓΙΣΗ ΑΠΟΔΟΣΗΣ.....</b>	<b>33</b>
3.1 <i>ΜΕΤΡΙΚΕΣ ΕΚΤΙΜΗΣΗΣ ΑΠΟΔΟΣΗΣ</i> .....	33
3.1.1 Βασικές Έννοιες Απόδοσης.....	33
3.1.2 Ανάκληση και Ακρίβεια (Recall and Precision) .....	34
3.1.3 Συνολική Απόδοση και σύγκριση συστημάτων ανάκτησης .....	38



3.2 ΑΠΟΤΕΛΕΣΜΑΤΑ ΑΠΟΔΟΣΗΣ ΣΥΣΤΗΜΑΤΟΣ.....	39
3.2.1 Ανάκληση και Ακρίβεια Επικεφαλίδας.....	39
3.2.2 Εντοπισμός Αναφορών .....	40
3.2.3 Ταξινόμηση Κειμένων .....	43
<b>ΚΕΦΑΛΑΙΟ 4 ΠΡΟΤΕΙΝΟΜΕΝΟ ΣΥΣΤΗΜΑ .....</b>	<b>47</b>
4.1 ΣΚΟΠΟΣ – ΠΕΡΙΓΡΑΦΗ .....	47
4.2 ΤΕΧΝΟΛΟΓΙΕΣ .....	47
4.2.1 Πρόσθετα πακέτα/βιβλιοθήκες .....	47
4.2.2 Δομή και περιγραφή του κώδικα εφαρμογής.....	48
4.3 ΠΕΡΙΓΡΑΦΗ ΣΥΣΤΗΜΑΤΟΣ ΔΙΑΧΕΙΡΗΣΗΣ ΝΟΜΙΚΩΝ ΚΕΙΜΕΝΩΝ .....	55
4.3.1 Εξαγωγή περίληψης .....	56
4.3.2 Επισήμανση αναφορών .....	57
4.4 ΠΕΡΙΓΡΑΦΗ ΣΥΣΤΗΜΑΤΟΣ ΚΑΤΗΓΟΡΙΟΠΟΙΗΣΗΣ ΚΕΙΜΕΝΩΝ .....	61
4.4.1 Κατηγοριοποίηση με το μοντέλο Multilayer Perceptron .....	61
4.4.2 Κατηγοριοποίηση με το μοντέλο RBFNetwork.....	64
4.4.3 Κατηγοριοποίηση με το μοντέλο SMO.....	65
4.4.4 Κατηγοριοποίηση με το μοντέλο Simple Logistic .....	66
4.5 ΤΟ ΕΡΓΑΛΕΙΟ WEKA .....	67
<b>ΚΕΦΑΛΑΙΟ 5 ΣΥΜΠΕΡΑΣΜΑΤΑ – ΜΕΛΛΟΝΤΙΚΗ ΕΡΕΥΝΑ .....</b>	<b>72</b>
5.1 ΣΥΜΠΕΡΑΣΜΑΤΑ.....	72
5.2 ΜΕΛΛΟΝΤΙΚΗ ΕΡΕΥΝΑ.....	76
<b>ΒΙΒΛΙΟΓΡΑΦΙΑ.....</b>	<b>77</b>
<b>ΠΑΡΑΡΤΗΜΑ Ι : ΚΩΔΙΚΑΣ.....</b>	<b>79</b>
Κλάση MainFrameForm .....	79
Κλάση HeaderSummarisationFrame.....	85
Κλάση HeaderSummarisationResults.....	90
Κλάση ExportRefMenu .....	96
Κλάση ExportReferences .....	108
Κλάση ExportRefResults.....	119
Κλάση ReadAndWriteDocx .....	124
Κλάση CreateVectorExcel.....	125
Κλάση AllFinalsVariables .....	127
Κλάση PerceptronFrame.....	128
Κλάση RBFNetworkFrame.....	139
Κλάση SMOFrame .....	149
Κλάση SimpleLogisticFrame .....	159
<b>ΠΑΡΑΡΤΗΜΑ ΙΙ : ΑΠΟΤΕΛΕΣΜΑΤΑ ΚΑΤΗΓΟΡΙΟΠΟΙΗΣΗΣ ΝΟΜΙΚΩΝ</b>	
<b>ΚΕΙΜΕΝΩΝ .....</b>	<b>168</b>
Αποτελέσματα 10-fold CrossValidation στο WEKA .....	168
Αποτελέσματα Εκπαίδευσης των μοντέλων .....	171
Αποτελέσματα Ελέγχου/Δοκιμής των μοντέλων.....	174

---

## ΚΑΤΑΛΟΓΟΣ ΕΙΚΟΝΩΝ

---

Εικόνα 1 Αρχιτεκτονική Δικτύου MultiLayer Perceptron .....	27
Εικόνα 2 Τελικό στάδιο εφαρμογής αλγορίθμου ανάστροφης διάδοσης σε MLP με δύο κρυφά επίπεδα .....	28
Εικόνα 3 Αρχιτεκτονική Δικτύου RBFNetwork .....	29
Εικόνα 4 10 folds Cross-Validation .....	32
Εικόνα 5 Αρχικό παράθυρο εφαρμογής διαχείρισης νομικών κειμένων .....	55
Εικόνα 6 Header Summarisation Frame .....	56
Εικόνα 7 Παράθυρο επιλογής αρχείου.....	56
Εικόνα 8 Απόσπασμα αποτελεσμάτων Εξαγωγής Περίληψης .....	57
Εικόνα 9 Μενού επιλογών για επισήμανση αναφορών.....	58
Εικόνα 10 Παράθυρο αποτελεσμάτων επισήμανσης αναφορών .....	59
Εικόνα 11 Ποσοστό(%) εμφάνισης αναφορών ανά κατηγορία .....	59
Εικόνα 12 Επισήμανση αναφορών .....	60
Εικόνα 13 Δημιουργία Διανυσμάτων.....	60
Εικόνα 14 Κατηγοριοποίηση με το μοντέλο MultiLayer Perceptron.....	62
Εικόνα 15 Αποτελέσματα εκπαίδευσης του MultiLayer Percetron .....	63
Εικόνα 16 Αποτελέσματα σε ποινικές αποφάσεις με το MultiLayer Perceptron.....	63
Εικόνα 17 Κατηγοριοποίηση με το RBFNetwork .....	64
Εικόνα 18 Κατηγοριοποίηση με το μοντέλο SMO .....	65
Εικόνα 19 Κατηγοριοποίηση με το μοντέλο Simple Logistic.....	66
Εικόνα 20 Γραφικό περιβάλλον του Weka 3.6 .....	67
Εικόνα 21 Το παράθυρο Explorer του WEKA .....	69
Εικόνα 22 Το παράθυρο του Classify στο WEKA .....	71
Εικόνα 23 Οπτικοποίηση του αποτελέσματος της κατηγοριοποίησης .....	71

## ΚΑΤΑΛΟΓΟΣ ΣΧΗΜΑΤΩΝ

---

Figure 1 Αναπαράσταση συνόλου εγγράφων .....	33
Figure 2 Συνδυασμοί ανάκλησης – ακρίβεια .....	35
Figure 3 Αρχική καμπύλη ανάκλησης και ακρίβειας.....	36
Figure 4 Καμπύλη ανάκλησης - ακρίβειας με την τεχνική της παρεμβολής .....	37
Figure 5 Σύγκριση συστημάτων.....	38
Figure 6 Ανάκληση και ακρίβεια για Header Summarization για τα κείμενα που χρησιμοποιήθηκαν για την υλοποίηση του συστήματος .....	39
Figure 7 Ανάκληση και ακρίβεια για το Header Summarization.....	40
Figure 8 Ανάκληση και ακρίβεια στον εντοπισμό των αναφορών για τα κείμενα που χρησιμοποιήθηκαν για την υλοποίηση του συστήματος.....	41
Figure 9 Ανάκληση και ακρίβεια στον εντοπισμό των αναφορών .....	43
Figure 10 Ανάκληση και ακρίβεια με το μοντέλο Multilayer Perceptron .....	44
Figure 11 Ανάκληση και ακρίβεια με το μοντέλο RBF Network .....	44
Figure 12 Ανάκληση και ακρίβεια με το μοντέλο SMO.....	45
Figure 13 Ανάκληση και ακρίβεια με το μοντέλο Simple Logistic .....	45

## ΚΑΤΑΛΟΓΟΣ ΠΙΝΑΚΩΝ

---

Πίνακας 1 Αποτελέσματα της λειτουργίας HeaderSummarisation (παράδειγμα 1).....	22
Πίνακας 2 Αποτελέσματα της λειτουργίας HeaderSummarisation (παράδειγμα 2).....	23
Πίνακας 3 Κατηγοριοποίηση εγγράφων .....	34
Πίνακας 4 Τιμές ανάκλησης και ακρίβειας.....	36
Πίνακας 5 Τιμές ακρίβειας με παρεμβολή για τα 11 επίπεδα ανάκλησης .....	37
Πίνακας 6 Τιμές ανάκλησης και ακρίβειας για όλες τις κατηγορίες αναφορών από τα κείμενα που χρησιμοποιήθηκαν για την υλοποίηση του συστήματος.....	41
Πίνακας 7 Τιμές ανάκλησης και ακρίβειας για τον εντοπισμό των αναφορών .....	42
Πίνακας 8 Τιμές ανάκλησης και ακρίβειας για όλες τις κατηγορίες του HeaderSummarisation ...	72
Πίνακας 9 Τιμές ανάκλησης και ακρίβειας για όλες τις κατηγορίες αναφορών κατά τον εντοπισμό τους .....	73
Πίνακας 10 Τιμές ανάκλησης, ακρίβειας και σωστά ταξινομημένων διανυσμάτων για τα μοντέλα που χρησιμοποιήθηκαν στο 10-fold cross validation.....	74
Πίνακας 11 Τιμές ανάκλησης, ακρίβειας και σωστά ταξινομημένων διανυσμάτων για τα μοντέλα που χρησιμοποιήθηκαν στο training set.....	74
Πίνακας 12 Τιμές ανάκλησης, ακρίβειας και σωστά ταξινομημένων διανυσμάτων για τα μοντέλα που χρησιμοποιήθηκαν στο test set για κείμενα από το πολιτικό τμήμα .....	75
Πίνακας 13 Τιμές ανάκλησης, ακρίβειας και σωστά ταξινομημένων διανυσμάτων για τα μοντέλα που χρησιμοποιήθηκαν στο test set για κείμενα από το ποινικό τμήμα.....	75
Πίνακας 14 Τιμές ανάκλησης, ακρίβειας και σωστά ταξινομημένων διανυσμάτων για τα μοντέλα που χρησιμοποιήθηκαν στο test set για κείμενα από το τμήμα Νόμου 3068/2002.....	75

## ΠΡΟΛΟΓΟΣ

---

Η παρούσα πτυχιακή εργασία με θέμα «Εφαρμογή Διαχείρισης Πληροφορίας σε Πηγές Νομικών Κειμένων» πραγματοποιήθηκε εξ ολοκλήρου στο Τμήμα Πληροφορικής με Εφαρμογές στην Βιοϊατρική στη Σχολή Θετικών Επιστημών του Πανεπιστημίου Θεσσαλίας, κατά την διάρκεια του ακαδημαϊκού έτους 2014-2015 υπό την επίβλεψη του Επίκουρου Καθηγητή Ιωάννη Αναγνωστόπουλου.

Στο σημείο αυτό θα ήθελα να ευχαριστήσω ιδιαίτερα τον επιβλέποντα καθηγητή μου για την ευκαιρία που μου έδωσε να ασχοληθώ με ένα τόσο ενδιαφέρον θέμα καθώς και για την πολύτιμη καθοδήγησή και τις συμβουλές του κατά την διάρκεια εκπόνησης της εργασίας. Ακόμα θέλω να ευχαριστήσω τον κ. Βασίλη Πλαγιανάκο, Αναπληρωτή Καθηγητή του τμήματός μου, για τις παρατηρήσεις του με σκοπό την βελτίωση της εργασίας. Επιπλέον ευχαριστώ τον συμφοιτητή μου Παναγιώτη Παπαδόπουλο για τις υποδείξεις του κατά την διάρκεια της σχεδίασης του γραφικού περιβάλλοντος της εφαρμογής, και τον συμφοιτητή μου Κωνσταντίνο Γούτσια για τις επισημάνσεις του κατά την διάρκεια της συγγραφής του κειμένου.

Τέλος, ένα μεγάλο ευχαριστώ στην οικογένεια μου και στους φίλους μου για την υποστήριξη και την αγάπη τους.

# ΚΕΦΑΛΑΙΟ 1

## ΕΙΣΑΓΩΓΗ

---

### 1.1 ΑΝΤΙΚΕΙΜΕΝΟ – ΣΚΟΠΟΣ

Την σημερινή εποχή η τεχνολογία αναπτύσσεται με γοργούς ρυθμούς με αποτέλεσμα να δημιουργείται η ανάγκη για την υλοποίηση νέων ιδεών και πρωτότυπων εφαρμογών. Σε αυτή την κατεύθυνση συμβάλει και η ραγδαία ανάπτυξη και εξέλιξη του Παγκόσμιου Ιστού τα τελευταία χρόνια.

Πριν από πολλά χρόνια ήταν δύσκολο να φανταστεί κανείς πως θα μπορούσε μέσα σε λίγα δευτερόλεπτα να εξερευνήσει και να διαχειριστεί πηγές πληροφορίας μεγάλου όγκου. Πιο συγκεκριμένα προσεγγίζοντας το αντικείμενο της πτυχιακής εργασίας, έρχονται στο προσκήνιο οι πηγές νομικών κειμένων, που περιλαμβάνουν χρήσιμη πληροφορία μέσα σε μεγάλο όγκο δεδομένων από κείμενα. Έτσι καθίσταται σημαντική η ύπαρξη μιας εφαρμογής που θα μειώσει δραματικά τον χρόνο εξερεύνησης και εξαγωγής χρήσιμης πληροφορίας από νομικά κείμενα. Η πρόκληση και ο προβληματισμός πηγάζει από το μέγεθος της διαθέσιμης αυτής πληροφορίας που αναπόφευκτα οδηγεί στην ανάγκη για δημιουργία σύγχρονων λογισμικών που θα έχουν την δυνατότητα να εντοπίζουν, να εξαγουν και να ταξινομούν το εν λόγω υλικό αυτόματα καθώς ο χρόνος επεξεργασίας κρίνεται πολύτιμος για να σπαταληθεί σε μια εποχή που χαρακτηρίζεται από ανάγκες που αλλάζουν καθημερινά και ανεξέλεγκτα.

Μέσα σε αυτή την φιλοσοφία βρίσκει πρόσφορο έδαφος ο σκοπός της παρούσας εργασίας καθώς συγκεντρώθηκαν πηγές νομικών κειμένων για την δημιουργία μιας εφαρμογής που θα είναι σε θέση να εντοπίζει σημαντικά σημεία σε ένα νομικό κείμενο, καθώς και την γενικότερη κατηγορία στην οποία αυτό ανήκει.

### 1.2 ΓΕΝΙΚΗ ΕΠΙΣΚΟΠΗΣΗ

Η πτυχιακή εργασία που υλοποιήθηκε με τίτλο «Εφαρμογή Διαχείρισης Πληροφορίας σε Πηγές Νομικών Κειμένων» περιλαμβάνει τα παρακάτω κεφάλαια:

- Εισαγωγή
- Διαχείριση Πληροφορίας σε Νομικές Πηγές
- Αποτελέσματα – Αξιολόγηση Απόδοσης
- Προτεινόμενο Σύστημα
- Συμπεράσματα – Μελλοντική Έρευνα

Ποιο αναλυτικά η διάρθρωση της πτυχιακής εργασίας παρουσιάζεται παρακάτω:

Στο πρώτο κεφάλαιο περιλαμβάνονται εισαγωγικές έννοιες για την εργασία.

Στο δεύτερο κεφάλαιο αναφέρθηκαν και αναλύθηκαν οι βασικές μέθοδοι κατά την εξόρυξη κειμένου και έγινε μια περιγραφή των νομικών κειμένων καθώς και του τρόπου με τον οποίο έγινε η εξαγωγή περίληψης και η επισήμανση των νομικών αναφορών. Επίσης εξετάστηκαν βασικές έννοιες μηχανικής μάθησης και αναλύθηκαν κάποιοι βασικοί αλγόριθμοι.

Στο τρίτο κεφάλαιο αναπτύχθηκαν οι βασικές μετρικές αποτίμησης αποτελεσματικότητας, ανάκληση και ακρίβεια. Ακόμα παρουσιάστηκε πως εντοπίζεται η συνολική απόδοση ενός συστήματος ανάκτησης πληροφορίας ώστε να μπορεί να γίνει σύγκριση μεταξύ συστημάτων ανάκτησης και αναφέρθηκαν τα αποτελέσματα ανάκτησης της εφαρμογής όσον αφορά την εξαγωγή περίληψης, την επισήμανση των αναφορών και την κατηγοριοποίηση κειμένων.

Στο τέταρτο κεφάλαιο αναφέρονται οι τεχνολογίες που χρησιμοποιήθηκαν κατά την υλοποίηση της εφαρμογής, κάποιες βασικές κλάσεις και μέθοδοι. Επιπλέον γίνεται μια παρουσίαση της εφαρμογής που να προσομοιώνει όσο περισσότερο γίνεται την εκτέλεση της εφαρμογής σε πραγματικό χρόνο και τέλος παρουσιάζεται το εργαλείο WEKA που αποδείχθηκε χρήσιμο κατά την υλοποίηση της παρούσας εφαρμογής.

Στο πέμπτο κεφάλαιο παρουσιάζονται τα συμπεράσματα που εξήχθησαν από την εκπόνηση της εργασίας και αναφέρονται μελλοντικές επεκτάσεις της και προοπτικές.

# ΚΕΦΑΛΑΙΟ 2

## ΔΙΑΧΕΙΡΙΣΗ ΠΛΗΡΟΦΟΡΙΑΣ ΣΕ ΝΟΜΙΚΕΣ ΠΗΓΕΣ

---

### 2.1 ΜΕΘΟΔΟΙ ΕΠΕΞΕΡΓΑΣΙΑΣ ΚΕΙΜΕΝΩΝ

Οι σημαντικότεροι τομείς για την επεξεργασία κειμένων είναι η ανακάλυψη γνώσεις σε κείμενο (Knowledge Discovery in Text) καθώς και ο τομέας της εξόρυξης κειμένου (Text Mining).

Είναι τομείς οι οποίοι αναπτύχθηκαν λόγω της μεγάλης αύξησης των δεδομένων από κείμενα, γεγονός το οποίο καθιστά αναγκαία την ύπαρξη μεθόδων για την ανάλυση τους. Ο κύριος παράγοντας της αύξησης αυτής οφείλεται στην αλματώδη ανάπτυξη του Διαδικτύου.

Χρησιμοποιούμε συνήθως τον όρο ανακάλυψη γνώσεις σε κείμενο για να δηλώσουμε την διαδικασία της μετατροπής των μη δομημένων δεδομένων από κείμενα σε πληροφορίες και γνώσεις, ενώ ο όρος εξόρυξη κειμένου αναφέρεται για ένα από τα βήματα της ανακάλυψης γνώσεις σε κείμενο το οποίο έχει να κάνει με την εξαγωγή προτύπων από τα δεδομένα των κειμένων[12].

Οι Κυριότερες κατηγορίες των μεθόδων που χειρίζεται η εξόρυξη κειμένου είναι:

- Εξαγωγή χαρακτηριστικών (Feature Extraction)
- Πλοήγηση με βάση το κείμενο (Text Based Navigation)
- Αναζήτηση και Ανάκτηση (Search and Retrieval)
- Κατηγοριοποίηση (Classification)
- Ομαδοποίηση (Clustering)
- Περιληπτική παρουσίαση της πληροφορίας (Summarization)
- Γλωσσικός προσδιορισμός (Language Identification) και απόδοση κειμένου στο συγγραφέα
- Συσχετίσεις (Associations)
- Απεικόνιση – Οπτικοποίηση (Visualization)

#### 2.1.1 Εξαγωγή Χαρακτηριστικών (Feature Extraction)

Έχει ως στόχο τον προσδιορισμό γεγονότων και σχέσεων στο κείμενο, διακρίνοντας συχνά εάν κάποια ονομαστική φράση είναι πρόσωπο, θέση, οργανισμός ή άλλο διακριτό αντικείμενο.

Οι αλγόριθμοι εξαγωγής χαρακτηριστικών χρησιμοποιούν λεξικά ή λεκτικά πρότυπα για να αναγνωρίσουν κάποιους όρους. Για παράδειγμα το όνομα ενός

οργανισμού όπως «ΣΔΟΕ» μπορεί να μην υπάρχει σε ένα λεξικό αλλά ένας αλγόριθμος εξαγωγής χαρακτηριστικών θα μπορούσε να το αναγνωρίσει ως έναν σημαντικό όρο. Επίσης ανιχνεύουν ως ίδιες λέξεις διαφορετικές λέξεις που ουσιαστικά προέρχονται από την ίδια κατηγορία, για παράδειγμα οι λέξεις «ερευνώ» και «ερευνητής» θα εντοπισθούν και θα αναγνωρισθούν από το σύστημα ως ίδια λέξη.

Τέλος η διεργασία της εξαγωγής χαρακτηριστικών θα πρέπει να μας παρουσιάζει τον αριθμό των εμφανίσεων κάθε όρου (word frequency).

### 2.1.2 Πλοήγηση με βάση το κείμενο (Text Based Navigation)

Η πλοήγηση με βάση το κείμενο επιτρέπει την πλοήγηση των χρηστών σε μια συλλογή εγγράφων με βάση σημαντικά θέματα ή σημαντικούς όρους. Αυτό βοηθάει στο να αναγνωριστούν έννοιες και συσχετίσεις μεταξύ σημαντικών όρων. Για παράδειγμα όταν αναζητούμε έγγραφα με τον όρο «ΣΔΟΕ» θα πρέπει να έχουμε γρήγορη πρόσβαση και να μετακινούμαστε σε έννοιες όπως «διευθυντής του ΣΔΟΕ» και «υπάλληλος του ΣΔΟΕ» και γενικά σε φράσεις περιέχουν τον όρο «ΣΔΟΕ». Η διεργασία αυτή έχει ως αποτέλεσμα την ικανότητα να μπορούμε να βλέπουμε ότι δύο όροι εμφανίζονται συχνά μαζί και έτσι να υποθέτουμε ότι πιθανόν έχουν κάποια σχέση μεταξύ τους.

### 2.1.3 Αναζήτηση και Ανάκτηση (Search and Retrieval)

Περιλαμβάνει την αναζήτηση σε εσωτερικές συλλογές εγγράφων ή σε συλλογές που βρίσκονται στον Παγκόσμιο Ιστό. Το βασικό του χαρακτηριστικό είναι οι διάφορες επιλογές αναζήτησης κειμένου. Μετά τη δημιουργία ευρετηρίου που είναι το πρώτο βήμα, μπορούν να προσφερθούν πολλές επιλογές αναζήτησης. Αυτές μπορούν να περιλαμβάνουν από απλές επιλογές αναζήτησης όπως επιλογές Boole (and/or/not) ως και πιο σύνθετες επιλογές αναζήτησης όπως (relevancy – ranked natural language searching) ή ασαφής αναζήτηση (fuzzy searching).

### 2.1.4 Κατηγοριοποίηση (Classification)

Η κατηγοριοποίηση είναι μία τεχνική κατά την οποία ένα στοιχείο ανατίθεται σε ένα προκαθορισμένο σύνολο κατηγοριών.

Οι κατηγορίες μπορούν να είναι προκαθορισμένες ή μπορεί να αφήνεται να προσδιοριστούν από τον χρήστη. Υπάρχουν δυο τρόποι για να δημιουργήσουμε κατηγορίες. Στην πρώτη περίπτωση δημιουργείται ένας θησαυρός οποίος να ορίζει ένα σύνολο από όρους σχετικά με το θέμα καθώς και σχέσεις μεταξύ τους (οι πιο κοινές σχέσεις είναι οι διευρυμένος όρος, κοντινότερος όρος, συνώνυμο και



σχετικός όρος). Ο κατηγοριοποιητής μπορεί τότε να ορίσει το αντικείμενο του κειμένου σύμφωνα με την συχνότητα των όρων σχετικά με το πεδίο που υπάρχουν στο έγγραφο. Στην δεύτερη περίπτωση ο κατηγοριοποιητής εκπαιδεύεται με πρότυπα έγγραφα. Αναλύονται λεκτικά πρότυπα (linguistic patterns) όπως λεκτικές συγγένειες (lexical affinities), συχνότητες λέξεων από πρότυπα κείμενα τα οποία είναι προ-κατηγοριοποιημένα και τα οποία ανήκουν σε κάθε ξεχωριστή κατηγορία με σκοπό να δημιουργηθεί μία στατιστική υπογραφή για κάθε κατηγορία. Μετά χρησιμοποιείται αυτή η στατιστική υπογραφή για να καταταχθούν τα καινούργια έγγραφα στις κατηγορίες. Το πλεονέκτημα της δεύτερης προσέγγισης είναι ότι δεν χρειάζεται να δημιουργηθεί κάποιος θησαυρός εννοιών, πράγμα εξαιρετικά δύσκολο για μεγάλους τομείς.

Για να αποφύγουμε την λάθος κατάταξη ενός εγγράφου είναι μερικές φορές απαραίτητο να εφοδιάζουμε με διάφορες κατηγορίες κάθε έγγραφο.

### 2.1.5 Ομαδοποίηση (Clustering)

Μια ομάδα είναι μια συλλογή από κείμενα που έχουν κάποια σχέση μεταξύ τους. Ομαδοποίηση είναι η διαδικασία της δημιουργίας ομάδων εγγράφων βάση κάποιας ομοιότητας, αυτόματα χωρίς να έχουμε από πριν προσδιορίσει τις κατηγορίες. Οι πιο γνωστοί αλγόριθμοι κατάταξης που υπάρχουν είναι ιεραρχικοί (hierarchical), δυαδικοί σχεσιακοί (binary relational) και ασαφής (fuzzy).

Οι ιεραρχικοί αλγόριθμοι ομαδοποίησης κατασκευάζουν ένα δέντρο με όλα τα έγγραφα στη ρίζα του και ένα έγγραφο σε κάθε φύλλο του. Οι ενδιάμεσοι κόμβοι περιέχουν κάποιον αριθμό εγγράφων και γίνονται όλο και πιο ειδικοί όσο πλησιάζουν προς τα φύλλα. Αυτό είναι εξαιρετικά χρήσιμοι όταν εξερευνούμε μια καινούργια συλλογή κειμένων και θέλουμε να έχουμε μια γενική επισκόπηση της συλλογής.

Η δυαδική σχεσιακή ομαδοποίηση χωρίζει τα έγγραφα σε μια οριζόντια δομή όπου κάθε ομάδα τοποθετείται μόνο σε ένα σύνολο.

Στην ασαφή κατηγοριοποίηση όλα τα έγγραφα περιέχονται σε όλες τις ομάδες αλλά με διαφορετικούς βαθμούς κυριότητας.

Ο πιο σημαντικός παράγοντας στην ομαδοποίηση είναι το μέτρο ομοιότητας. Όλοι οι αλγόριθμοι ομαδοποίησης βασίζονται σε μέτρα ομοιότητας και υπάρχουν διάφορα είδη μέτρων ομοιότητας. Ένας τύπος χρησιμοποιεί λέξεις οι οποίες εμφανίζονται συχνά μαζί (λεκτικές συγγένειες π.χ. Ιατρική Πληροφορική) σαν κοινά χαρακτηριστικά για να κατατάξει τα έγγραφα στις ομάδες. Ένας άλλος τύπος χρησιμοποιεί χαρακτηριστικά που έχουν εξαχθεί όπως για παράδειγμα «άρ. 23 παρ. 2 του ΚΠΔ».

### 2.1.6 Περιληπτική παρουσίαση της πληροφορίας (Summarization)

Η εξαγωγή περίληψης είναι η διαδικασία κατά την οποία μειώνεται η ποσότητα του κειμένου διατηρώντας όμως πάντα το βασικό του νόημα. Σε αυτού του είδους τις διεργασίες ο χρήστης καλείται τις πιο πολλές φορές να καθορίσει ένα πλήθος παραμέτρων όπως παραδείγματος χάριν το πλήθος των λέξεων που θα εξαχθούν ή το ποσοστό επί του συνολικού κειμένου το οποίο θα αποτελεί την περίληψη. Το αποτέλεσμα περιλαμβάνει συνήθως τις πιο σημαντικές προτάσεις του κειμένου.

### 2.1.7 Γλωσσικός προσδιορισμός (Language Identification) και απόδοση κειμένου στο συγγραφέα

Ένα εργαλείο language identification μπορεί να προσδιορίσει σε ποια γλώσσα είναι γραμμένο ένα κείμενο, ή και τι ποσοστό του κειμένου είναι γραμμένο σε κάθε γλώσσα, εάν αυτό είναι γραμμένο σε περισσότερες. Επιπλέον, υπάρχει η δυνατότητα προσδιορισμού του συγγραφέα στον οποίο ανήκει το κείμενο, χρησιμοποιώντας τεχνικές data mining.

### 2.1.8 Συσχετίσεις (Associations)

Στις συσχετίσεις αναγνωρίζονται σχέσεις μεταξύ χαρακτηριστικών γνωρισμάτων που έχουν εξαχθεί από τη συλλογή εγγράφων, και ορίζεται ένα πρότυπο με τη χρήση μιας αντικειμενικής συσχέτισης. Το πρότυπο αυτό, εκφράζει έναν κανόνα που αναφέρει ότι αν βρεθεί η υπολέξη που περιέχεται στο πρότυπο, ακολουθούμενη από μία άλλη δεδομένη υπολέξη, σε συγκεκριμένη απόσταση μεταξύ τους, τότε η αντικειμενική συνθήκη θα διατηρηθεί με μεγάλη πιθανότητα. Οι κανόνες αυτοί είναι πολύ ευέλικτοι για την περιγραφή των τοπικών ομοιοτήτων που περιέχονται στα δεδομένα του κειμένου.

### 2.1.9 Απεικόνιση – Οπτικοποίηση (Visualization)

Το visualization χρησιμοποιεί την εξαγωγή χαρακτηριστικών γνωρισμάτων και το ευρετήριο βασικών όρων για να κατασκευάσει μια γραφική αναπαράσταση μιας συλλογής εγγράφων. Η προσέγγιση αυτή βοηθάει το χρήστη να αναγνωρίζει πολύ γρήγορα τα κύρια θέματα και τις βασικές έννοιες των κειμένων, με βάση τη σπουδαιότητα (πχ μέγεθος) αυτών στην αναπαράσταση.

## 2.2 ΠΕΡΙΓΡΑΦΗ ΤΩΝ ΝΟΜΙΚΩΝ ΠΗΓΩΝ

Με τον όρο νομικές πηγές αναφερόμαστε σε κείμενα νομικού περιεχομένου. Στην κατηγορία των νομικών κειμένων υπάγονται κείμενα αποφάσεων του Αρείου Πάγου. Ο Άρειος Πάγος είναι το ανώτατο ακυρωτικό δικαστήριο της πολιτικής και ποινικής δικαιοσύνης της Ελλάδος. Οι αποφάσεις που παίρνει χωρίζονται σε αποφάσεις Πολιτικές, Ποινικές και σε αποφάσεις του Νόμου 3068/2002 [15]. Κάθε μία από τις κατηγορίες αυτές χωρίζονται σε αποφάσεις που λαμβάνονται από τα εξής τμήματα:

- Α
- Α1
- Α2
- Β
- Β1
- Β2
- Δ
- Ε
- ΣΤ
- Ζ
- Ολομέλεια

### 2.2.1 Η Δομή των Νομικών Κειμένων

Τα νομικά κείμενα έχουν ένα συγκεκριμένο πρότυπο δομής που ακολουθεί ένα σύνολο κανόνων. Μερικοί από τους κανόνες περιγράφονται συνοπτικά παρακάτω:

- Όλα τα κείμενα έχουν έναν κωδικό αριθμό ο οποίος είναι μοναδικός για κάθε δικαστική απόφαση και βρίσκεται πάντα στην αρχή (π.χ. Αριθμός 1017/2011) .
- Στην δεύτερη γραμμή δηλώνεται πάντα σε ποια κατηγορία και σε ποιο τμήμα ανήκει η απόφαση (π.χ. Ε΄ ΠΟΙΝΙΚΟ ΤΜΗΜΑ).
- Στην πρώτη παράγραφο του κειμένου εντοπίζονται όλοι όσοι συμμετείχαν στην νομική διαδικασία για την λήψη της απόφασης (Δικαστές, Εισηγητές, Εισαγγελείς, Δικηγόροι).
- Μετά την πρώτη παράγραφο ακολουθεί η φράση «ΣΕΦΘΗΚΕ ΣΥΜΦΩΝΑ ΜΕ ΤΟ ΝΟΜΟ» και εν συνεχεία γίνεται η ανάλυση της νομικής απόφασης η οποία περιλαμβάνει ένα σύνολο αναφορών.

Παρακάτω παρουσιάζεται απόσπασμα ενός τέτοιου κειμένου:

**Αριθμός 1104/2011**

**ΣΤ' ΠΟΙΝΙΚΟ ΤΜΗΜΑ**

*Συγκροτήθηκε από τους Δικαστές: Αιμιλία Λίτινα, Προεδρεύουσα Αρεοπαγίτη, ως αρχαιότερο μέλος στη σύνθεση και σύμφωνα με την 101/21-7-2010 Πράξη του Προέδρου του Αρείου Πάγου, Ανδρέα Τσόλια-Εισηγητή, Παναγιώτη Ρουμπή, Βασίλειο Φράγγο, σύμφωνα με την 197/28-12-2010 Πράξη του Προέδρου του Αρείου Πάγου, για συμπλήρωση της σύνθεσης, και Αθανάσιο Γεωργόπουλο, Αρεοπαγίτες.*

*Συνήλθε σε δημόσια συνεδρίαση στο Κατάστημά του στις 8 Μαρτίου 2011, με την παρουσία του Αντεισαγγελέα του Αρείου Πάγου Μιλτιάδη Ανδρειωτέλλη (γιατί κωλύεται ο Εισαγγελέας) και της Γραμματέως Πελαγίας Λόζιου, για να δικάσει την αίτηση του αναιρεσιόντος-κατηγορουμένου Χ. Α. του Γ., κατοίκου ..., που εκπροσωπήθηκε από τον πληρεξούσιο δικηγόρο του Αλέξανδρο Δημάκη, για αναίρεση της υπ'αριθ.4260/2010 αποφάσεως του Τριμελούς Εφετείου Αθηνών. Με πολιτικώς ενάγουσα την Β. Κ., κάτοικο ..., που δεν παρέστη.*

**ΣΚΕΦΘΗΚΕ ΣΥΜΦΩΝΑ ΜΕ ΤΟ ΝΟΜΟ**

*Επειδή, η καταδικαστική απόφαση έχει την απαιτούμενη από τις διατάξεις των άρθρων 93 παρ. 3 του Συντάγματος και 139 του Κώδικα Ποινικής Δικονομίας ειδική και εμπειριστατωμένη αιτιολογία, η έλλειψη της οποίας ιδρύει τον προβλεπόμενο από το άρθρο 510 παρ. 1 στοιχ. Δ' του ιδίου Κώδικα λόγο αναίρεσεως, όταν εκτίθενται σε αυτή με σαφήνεια, πληρότητα και χωρίς αντιφάσεις ή λογικά κενά, τα πραγματικά περιστατικά που προέκυψαν και συγκροτούν την αντικειμενική και υποκειμενική υπόσταση του εγκλήματος για το οποίο καταδικάστηκε ο κατηγορούμενος, και αναφέρονται οι αποδείξεις που τα θεμελιώνουν και οι νομικοί συλλογισμοί με βάση τους οποίους υπήχθησαν τα περιστατικά που αποδείχθηκαν στην ουσιαστική ποινική διάταξη που εφαρμόστηκε.[...] Επομένως, ο σχετικός και μοναδικός από το άρθρο 510 παρ. 1 στοιχ. Δ του Κώδικα Ποινικής Δικονομίας λόγος της κρινόμενης αιτήσεως αναίρεσεως, για έλλειψη ειδικής και εμπειριστατωμένης αιτιολογίας είναι βάσιμος και πρέπει να γίνει δεκτή η κρινόμενη αίτηση αναίρεσεως, να ανααιρεθεί η προσβαλλόμενη και να παραπεμφθεί η υπόθεση για νέα συζήτηση στο ίδιο δικαστήριο, αφού είναι δυνατή η συγκρότηση του από άλλους δικαστές, εκτός εκείνων που δίκασαν προηγουμένως (519 ΚΠΔ).*

**ΓΙΑ ΤΟΥΣ ΛΟΓΟΥΣ ΑΥΤΟΥΣ**

*Αναιρεί Παραπέμπει*

Στην παρούσα εργασία χρησιμοποιήθηκαν τέτοιου είδους κείμενα με απώτερο σκοπό την εξαγωγή επικεφαλίδας, μέσω της περιληπτικής παρουσίασης κάποιων βασικών στοιχείων που εμφανίζονται στο κείμενο καθώς και την επισήμανση των νομικών αναφορών που περιέχουν τα κείμενα αυτά.

## 2.2.2 Εξαγωγή Επικεφαλίδας σε Νομικά Κείμενα

Μελετώντας τη δομή των κειμένων παρατηρήθηκε πως εξάγοντας κάποια βασική πληροφορία από την πρώτη παράγραφο των πηγών που μπορεί να χρησιμοποιηθεί ως επικεφαλίδα του αντίστοιχου κειμένου ακολουθώντας μία καθορισμένη δομή και συγκεκριμένους κανόνες. Οι κανόνες αυτοί είναι οι ακόλουθοι:

- Όπου εντοπίζουμε τη λέξη ΑΡΙΘΜΟΣ πρέπει να την εξάγουμε ως Αρίθμ. και να ακολουθήσει ο αριθμός της αποφάσεως.
- Ο πρώτος Δικαστής ορίζεται ως Προεδρεύων ο Αντιπρόεδρος / Προεδρεύουσα η Αντιπρόεδρος
- Ο Πρόεδρος του Αρείου Πάγου που είναι Εισηγητής ορίζεται ως Εισηγητής ο Αεροπαγίτης / Εισηγήτρια η Αεροπαγίτης.
- Ο Αντεισαγγελέας του Αρείου Πάγου ορίζεται ως Εισαγγελεύς.
- Ο Δικηγόρος παραμένει ως έχει.
- Αν πρόκειται για ανδρικό όνομα εξάγεται μόνο το αρχικό γράμμα του ονόματος και το επώνυμο κανονικά, για παράδειγμα αν έχουμε το όνομα «Γεώργιο Χρυσικό» γίνεται «Γ. Χρυσικός»
- Τα γυναικεία ονόματα παραμένουν ως έχουν.

Παρακάτω ακολουθούν 2 χαρακτηριστικά παραδείγματα για τον τρόπο με τον οποίο παράγεται η επικεφαλίδα/περίληψη.

Παράδειγμα 1 (έχουν επισημανθεί τα σημεία που εντοπίζει το σύστημα).

### **ΑΡΙΘΜΟΣ 1018/2011**

#### **ΣΤ' ΠΟΙΝΙΚΟ ΤΜΗΜΑ**

*Συγκροτήθηκε από τους Δικαστές: Χαράλαμπο Δημάδη, Προεδρεύοντα Αεροπαγίτη, ως αρχαιότερο μέλος στη σύνθεση και σύμφωνα με την 21/27.1.2011 Πράξη του Προέδρου του Αρείου Πάγου, Ανδρέα Τσόλια, Παναγιώτη Ρουμπή-Εισηγητή, Ιωάννη Γιαννακόπουλο, σύμφωνα με την 17/24.1.2011 Πράξη του Προέδρου του Αρείου Πάγου για συμπλήρωση της σύνθεσης και Αθανάσιο Γεωργόπουλο, Αεροπαγίτες*

*Συνήλθε σε δημόσια συνεδρίαση στο Κατάστημά του στις 8 Φεβρουαρίου 2011, με την παρουσία του Αντεισαγγελέα του Αρείου Πάγου Αθανάσιου Κονταξή (γιατί κωλύεται ο Εισαγγελέας του Αρείου Πάγου) και της Γραμματέως Πελαγίας Λόζιου, για να δικάσει τις αίτηση των αναιρεσιόντων-κατηγορουμένων: 1.Η. Α. του Δ., κατοίκου ... και 2.Φ. Μ. του Ν., κατοίκου ..., που εκπροσωπήθηκαν από τον πληρεξούσιο δικηγόρο τους Εμμανουήλ Παπαδάκη, περί αναιρέσεως της υπ' αριθμ. 3081/2008, 3159/2008 και 399/2009 αποφάσεως του Πενταμελούς Εφετείου Αθηνών. Με συγκατηγορούμενο τον Α. Κ. του Γ..*

<b>Αυτό που εντόπισε το σύστημα</b>	<b>Αυτό που εμφάνισε</b>
<i>ΑΡΙΘΜΟΣ 1018/2011</i>	Αριθμ. 1018/2011
Χαράλαμπο Δημάδη,	Προεδρεύων ο Αντιπρόεδρος <b>Χ. Δημάδης</b>
Παναγιώτη Ρουμπή-Εισηγητή,	Εισηγητής ο Αρεοπαγίτης <b>Π. Ρουμπής</b>
Αθανάσιου Κονταζή	Εισαγγελεύς <b>Α. Κονταζής</b>
Εμμανουήλ Παπαδάκη,	Δικηγόρος <b>Ε. Παπαδάκης</b>

Πίνακας 1 Αποτελέσματα της λειτουργίας HeaderSummarisation (παράδειγμα 1)

## Παράδειγμα 2

### **ΑΡΙΘΜΟΣ 1725/2011**

#### **ΣΤ' ΠΟΙΝΙΚΟ ΤΜΗΜΑ**

*Συγκροτήθηκε από τους Δικαστές: Γεώργιο Χρυσικό, Αντιπρόεδρο του Αρείου Πάγου, Αιμιλία Λίτινα, Ανδρέα Τσόλια, Κωνσταντίνο Φράγκο, Ειρήνη Κιουρκτσόγλου-Πετρουλάκη- Εισηγήτρια, Αρεοπαγίτες.*

*Συνήλθε σε δημόσια συνεδρίαση στο Κατάστημά του στις 15 Νοεμβρίου 2011, με την παρουσία του Αντεισαγγελέως του Αρείου Πάγου Γεωργίου Κολιοκώστα (γιατί κωλύεται ο Εισαγγελέας) και της Γραμματέως Πελαγίας Λόζιου, για να δικάσει την αίτηση της αναιρεσείουσας-κατηγορουμένης, Χ. Π. του Η., κατοίκου ..., που εκπροσωπήθηκε από την πληρεξούσια δικηγόρο της Ελένη Γκόνου περί αναιρέσεως της 1970/2010 αποφάσεως Τριμελούς Πλημμελειοδικείου Μυτιλήνης. Με πολιτικώς ενάγουσας τη Ρ. Κ. του Γ., κατοίκου ..., που δεν παρέστη.*

<b>Αυτό που εντόπισε το σύστημα</b>	<b>Αυτό που εμφάνισε</b>
ΑΡΙΘΜΟΣ 1725/2011	Αριθμ. 1725/2011
Γεώργιο Χρυσικό	Προεδρεύων ο Αντιπρόεδρος <b>Γ. Χρυσικός</b>
Ειρήνη Κιουρκτσόγλου-Πετρουλάκη-Εισηγήτρια,	Εισηγήτρια η Αρεοπαγίτης <b>Ειρήνη Κιουρκτσόγλου-Πετρουλάκη</b>
Γεωργίου Κολιοκώστα	Εισαγγελεύς <b>Γ. Κολιοκώστας</b>
Ελένη Γκόνου	Δικηγόρος <b>Ελένη Γκόνου</b>

Πίνακας 2 Αποτελέσματα της λειτουργίας HeaderSummarisation (παράδειγμα 2)

Τελικά αποτελέσματα των παραδειγμάτων 1 και 2 αντίστοιχα είναι:

**Αριθμ. 1018/2011**

Προεδρεύων ο Αντιπρόεδρος **Χ. Δημάδης**

Εισηγητής ο Αρεοπαγίτης **Π. Ρουμπής**

Εισαγγελεύς **Α. Κονταξής**

Δικηγόροι **Ε. Παπαδάκης**

**Αριθμ. 1725/2011**

Προεδρεύων ο Αντιπρόεδρος **Γ. Χρυσικός**

Εισηγήτρια η Αρεοπαγίτης **Ειρήνη Κιουρκτσόγλου-Πετρουλάκη**

Εισαγγελεύς **Γ. Κολιοκώστας**

Δικηγόρος **Ελένη Γκόνου**

### 2.2.3 Επισήμανση Αναφορών σε Νομικά Κείμενα

Νομική αναφορά: Είναι μια διευκρίνιση που επιτρέπει τον εντοπισμό ενός παραθέματος, τη μνεία ενός έργου ή μέρους έργου που έχει σχέση με το αντικείμενο του κειμένου ή την παραπομπή, μέσα στο ίδιο κείμενο ή σε υποσημειώσεις. (π.χ. άρθρο 510 παρ. 1 στοιχ. Δ του Κώδικα Ποινικής Δικονομίας).

Μελετώντας ένα ευρύ φάσμα τέτοιων νομικών αποφάσεων παρατηρήθηκαν συντομογραφίες κατά την διατύπωση των αναφορών. Κυριότερες εκ των οποίων είναι οι ακόλουθες [14]:

- Το άρθρο μνημονεύεται ως «άρ.» ή «άρθρο»
- Οι παράγραφοι μνημονεύονται ως «παρ.»
- Η περίπτωση μνημονεύεται ως «περ.»
- Η υποπερίπτωση ως «υποπερ.»
- Το στοιχείο αναφέρεται ως «στοιχ.»
- Το εδάφιο αναφέρεται ως «εδάφιο» ή «εδ.» και αριθμείται ολογράφως. Όταν συναντάται ο όρος «περίοδος» σε κωδικοποιούμενο κείμενο, στο κωδικοποιητικό κείμενο γίνεται χρήση του πλέον κατάλληλου από τους όρους που αναφέρονται στο παρόν εγχειρίδιο.
- Τα προεδρικά διατάγματα μνημονεύονται ως «π.δ.». Στη γενική πτώση, η συντομογραφία είναι «π.δ/τος». Συντομογραφία χρησιμοποιείται μόνο όταν μνημονεύεται συγκεκριμένο προεδρικό διάταγμα με τα στοιχεία του (π.δ. 18/1989). Αντιθέτως, η συντομογραφία δεν χρησιμοποιείται σε άλλες περιπτώσεις, όπως στις εξουσιοδοτικές διατάξεις (π.χ. «με προεδρικό διάταγμα, που εκδίδεται ...» και όχι «με π.δ. που εκδίδεται...»).
- Με Συντ. εμφανίζεται η αναφορά του Συντάγματος.
- Το Σώμα Δίωξης Οικονομικού Εγκλήματος αναφέρεται ως ΣΔΟΕ.
- Ο Κώδικας Πολιτικής Δικονομίας αναφέρεται ως ΚΠολΔ.
- Το Διεθνές Σύμφωνο για τα Ατομικά και Πολιτικά Δικαιώματα αναφέρεται ως ΔΣΑΠΔ
- Η Ευρωπαϊκή Σύμβαση για τα Δικαιώματα του Ανθρώπου αναφέρεται ως ΕΣΔΑ.
- Ο Ποινικός Κώδικας αναφέρεται ως ΠΚ.
- Ο Κώδικας Ποινικής Δικονομίας αναφέρεται ως ΚΠΔ
- Οι νόμοι μνημονεύονται ως «ν.» ή «Ν.» (π.χ. ν. 1558/1985)
- Τα Νομοθετικά Διατάγματα μνημονεύονται ως «ν.δ» ή «Ν.Δ»
- Ο Αστικός Κώδικας μνημονεύεται ως ΑΚ ή Α.Κ.
- Υπάρχουν αναφορές που φέρουν τους όρους Δικαστηρίου, Πλημμελειοδικείου, Εφετείου, Πρωτοδικείου και Ευρωπαϊκής Συμβάσεως.
- Αναφορές που καταλήγουν στις φράσεις «του ιδίου κώδικα», «του ιδίου νόμου» ανήκουν στην αμέσως προηγούμενη κατηγορία που υπάρχει στο κείμενο.



Προς διευκόλυνση του εντοπισμού και της επισήμανσης των αναφορών έγινε αναγκαία η ομαδοποίησή τους με σκοπό την εξαγωγή κανόνων για κάθε κατηγορία.

Αρχικά παρατηρήθηκε ότι όλες οι αναφορές αρχίζουν με τους όρους «άρθρο» ή «άρ.» ή «παρ.» ή με μία λέξη που περιέχει «/», π.χ. 1024/71. Μετά εντοπίστηκε πώς όλες οι αναφορές τελειώνουν με έναν όρο ανάλογα με την κατηγορία την οποία ανήκουν. Έτσι ομαδοποιήθηκαν στις ακόλουθες κατηγορίες:

- Συντ.
- ΣΔΟΕ
- ΚΠΟΛΔ
- ΔΣΑΠΔ
- ΕΣΔΑ
- ΠΚ
- ΚΠΔ
- Ν.
- Ν.Δ.
- Δικαστήρια
- Πλημμελειοδικεία
- Εφετεία
- Πρωτοδικεία
- Ευρωπαϊκής Συμβάσεως
- Αστικού Κώδικα

Όπως είδαμε στην παράγραφο 2.2.2 κάθε απόφαση νομικού κειμένου αρχίζει με την λέξη ΑΡΙΘΜΟΣ. Έτσι μπορούμε σε ένα νομικό κείμενο που περιλαμβάνει πολλές αποφάσεις να ξεχωρίσουμε πότε αρχίζει και τελειώνει κάθε απόφαση. Ακόμα αφού έχουμε χωρίσει σε κατηγορίες τις αναφορές που συναντά κανείς στα κείμενα αυτά, μετρώντας τις συναντήσεις με κάθε μια από αυτές τις αναφορές μπορεί να παραχθεί ένα διάγραμμα μεγέθους  $n$  ( $n$  ο αριθμός των διαφορετικών κατηγοριών των αναφορών) που σε κάθε θέση του να περιλαμβάνει την συχνότητα εμφάνισης της αντίστοιχης αναφοράς. Οπότε μπορούμε για κάθε απόφαση να εξάγουμε ένα διάγραμμα μεγέθους  $n$ . Τέλος έχοντας συλλέξει ένα σύνολο διανυσμάτων ίσο σε αριθμό με τις αποφάσεις του κειμένου οδηγούμαστε σε μεθόδους μηχανικής μάθησης κάνοντας μια επιβλεπόμενη ταξινόμηση στα κείμενα ανάλογα με την κατηγορία που ανήκουν (Πολιτικές, Ποινικές, Νόμου 3068/2002).

## 2.3 ΜΕΘΟΔΟΙ ΜΗΧΑΝΙΚΗΣ ΜΑΘΗΣΗΣ ΚΑΙ ΤΑΞΙΝΟΜΙΣΗΣ ΚΕΙΜΕΝΩΝ

### 2.3.1 Βασικές Έννοιες της Μηχανικής Μάθησης

Η Μηχανική Μάθηση είναι ένας τομέας της Επιστήμης Υπολογιστών που ασχολείται με την ευφυή συμπεριφορά υπολογιστικών μηχανών.

Ο τομέας αυτός έχει μεγάλο εύρος εφαρμογής, για παράδειγμα μπορεί να χρησιμοποιηθεί σε μηχανές αναζήτησης, για αναγνώριση προτύπων, επεξεργασία φυσικής γλώσσας, βιοπληροφορική, ιατρική διάγνωση. Τα συστήματα μηχανικής μάθησης έχουν έναν τρόπο να μαθαίνουν και να βελτιώνονται. Η τóπος με τον οποίο «μαθαίνουν» μπορεί να χωριστεί σε τρεις κατηγορίες:

- Επιβλεπόμενη μάθηση (Supervised learning):

Στην επιβλεπόμενη μάθηση δημιουργείται μία συνάρτηση που αντιστοιχεί δεδομένα εισόδου σε επιθυμητές εξόδους. Η κατηγοριοποίηση αποτελεί κλασική μορφή επιβλεπόμενης μάθησης. Σε αυτή δίνονται στο σύστημα αντικείμενα που ανήκουν σε μία κατηγορία που είναι γνωστή και το σύστημα πρέπει να εντοπίσει τα κοινά χαρακτηριστικά των αντικειμένων αυτών.

- Μη επιβλεπόμενη μάθηση (Unsupervised learning):

Στη μη επιβλεπόμενη μάθηση δεν δίνονται στο σύστημα δεδομένα εκπαίδευσης. Το σύστημα προσπαθεί να εντοπίσει μόνο του κοινά χαρακτηριστικά σε ένα σύνολο αντικειμένων και να τα τοποθετήσει σε κατηγορίες που έχει δημιουργήσει μόνο του. Η ομαδοποίηση (clustering) και οι κανόνες συσχέτισης (associations rules) είναι δύο κατηγορίες που χρησιμοποιεί η μη επιβλεπόμενη μάθηση.

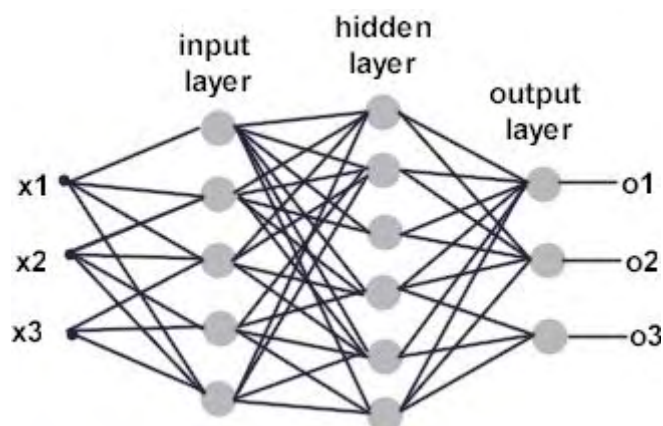
- Ενισχυτική μάθηση (Reinforcement learning):

Η ενισχυτική μάθηση χρησιμοποιείται κυρίως σε ευφυείς πράκτορες οι οποίοι κάνουν συλλογισμούς για το περιβάλλον στο οποίο βρίσκονται και δρουν πάνω σε αυτό για την επίτευξη αποτελεσμάτων. Η μάθηση του συστήματος έγκειται στο ότι υπάρχει επιβράβευση του πράκτορα ανάλογα με το πόσο σωστός είναι ο τρόπος με τον οποίο επιλέγει να μεταβεί από μία κατάσταση σε μία άλλη. Η διαφορά της από την επιβλεπόμενη μάθηση είναι ότι δεν χρησιμοποιεί γνώση που να είναι γνωστή από πριν.

### 2.3.2 Μοντέλο Perceptron Πολλών Στρωμάτων (MultiLayer Perceptron)

Η δομή ενός δικτύου πολλών στρωμάτων (MLP – multilayer perceptron) έχει ως εξής: Το δίκτυο αποτελείται από το επίπεδο εισόδου όπου και εφαρμόζονται οι είσοδοι στο σύστημα. Στη συνέχεια ακολουθούν τα ενδιάμεσα κρυμμένα επίπεδα. Και τέλος έχουμε το επίπεδο εξόδου.

Οι νευρώνες κάθε επιπέδου συνδέονται με τους νευρώνες των επόμενων επιπέδων με τα βάρη  $w_{ij}^m$ , όπου  $w_{ij}^m$  είναι το βάρος που συνδέει το  $i$  νευρώνα του  $m$  επιπέδου με το  $j$  νευρώνα του  $m+1$  επιπέδου. Στα δίκτυα αυτά κάθε νευρώνας συνδέεται μόνο με νευρώνες επόμενων επιπέδων γι' αυτό ανήκουν στην κατηγορία των δικτύων προσοτροφοδότησης (feedforward). Γενικά συνηθίζεται κάθε νευρώνας να συνδέεται με όλους (ή κάποιους από) τους νευρώνες του αμέσως επόμενου επιπέδου, οπότε και το νευρωνικό δίκτυο ονομάζεται δομημένο και έχει τη μορφή του παρακάτω σχήματος.



Εικόνα 1 Αρχιτεκτονική Δικτύου MultiLayer Perceptron

Η έξοδος κάθε νευρώνα είναι ίση με το άθροισμα του γινομένου των εισόδων του επί τα αντίστοιχα βάρη τους αυξημένο κατά μια σταθερά, το κατώφλι και περασμένο μέσα από μια συνάρτηση ενεργοποίησης  $f$ . Η έξοδος ενός νευρώνα εφαρμόζεται ως είσοδος στους νευρώνες των επόμενων επιπέδων με τους οποίους συνδέεται μέχρι το επίπεδο εξόδου. Πιο αναλυτικά αν συμβολίσουμε με  $w_{ij}^m$  το βάρος που συνδέει το  $i$  νευρώνα του  $m$  επιπέδου με το  $j$  νευρώνα του  $m+1$ , με  $b_i^m$  το κατώφλι του και με  $y_i^m$  την έξοδό του ( $i$  νευρώνα στο  $m$  επίπεδο), τότε η κάθε έξοδος είναι ίση με:

$$y_i^m = \begin{cases} f(\sum_j w_{ij}^{m-1} * y_j^{m-1}) + b_i^m & m \geq 2 \\ x_i & m = 1 \end{cases}$$

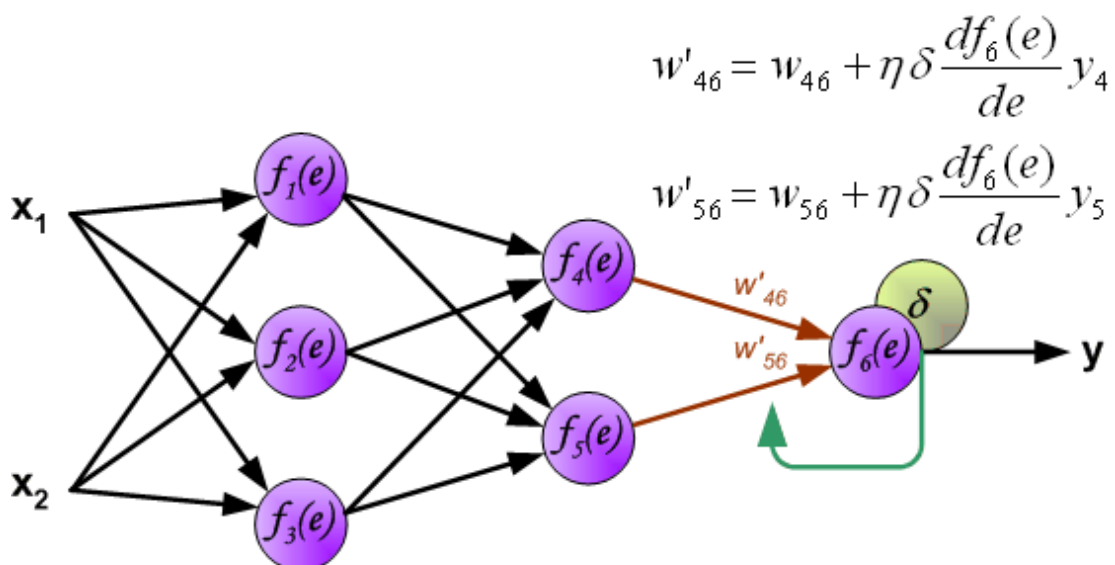
,όπου  $f$  η συνάρτηση ενεργοποίησης και  $x_i$  οι είσοδοι. Παρατηρούμε ότι για το πρώτο επίπεδο, δηλαδή για  $m = 1$  η έξοδος του κάθε νευρώνα είναι ίδια με την είσοδο.

Η εκπαίδευση του δικτύου πολλών στρωμάτων είναι η διαδικασία ρύθμισης των βαρών του έτσι ώστε να ικανοποιείται κάποιο κριτήριο καταλληλότητας. Σε ένα δίκτυο MLP δίνεται εκτός από την είσοδο και η επιθυμητή έξοδος οπότε έχουμε εκπαίδευση με επίβλεψη (supervised).

Για τα πολυστρωματικά νευρωνικά δίκτυα ο πιο διαδεδομένος αλγόριθμος εκπαίδευσης με επίβλεψη είναι ο αλγόριθμος της ανάστροφης διάδοσης (Back Propagation).

Ο αλγόριθμος ανάστροφης διάδοσης λειτουργεί ως εξής:

Αρχικά παρουσιάζουμε κάποιο παράδειγμα στην είσοδο του δικτύου και από την έξοδο του υπολογίζουμε την τιμή της συνάρτησης σφάλματος. Στη συνέχεια υπολογίζουμε με φορά από την έξοδο προς την είσοδο (ανάστροφη) τα σήματα σφάλματος  $\delta$ . Τέλος με βάση τα σήματα αυτά καθώς και τις εισόδους των νευρώνων (Hebbian Rule) βελτιώνουμε τα βάρη. Κάθε σύνολο δεδομένων εκπαίδευσης χρειάζεται να περάσει πολλές φορές από ένα δίκτυο ώστε να συγκλίνουν τα βάρη σε τέτοιες τιμές που να εξασφαλίζουν ότι το νευρωνικό δίκτυο «έμαθε» (μέχρι κάποιο βαθμό πάντα) τα παραδείγματα. Η εκπαίδευση ενός πολυστρωματικού νευρωνικού δικτύου τύπου MLP είναι μια επαναληπτική διαδικασία κατά την οποία παρουσιάζονται στο δίκτυο ένα σύνολο από δεδομένα, τα δεδομένα εκπαίδευσης, και επιβάλλονται στο δίκτυο οι επιθυμητές έξοδοι. Κάθε παρουσίαση των δεδομένων εκπαίδευσης στο δίκτυο ονομάζεται εποχή (epoch). Πολλοί αλγόριθμοι εκπαίδευσης αντί να αλλάζουν τα βάρη με κάθε παράδειγμα που τους παρουσιάζεται κάνουν συνολική αλλαγή των βαρών μετά το πέρας όλου του συνόλου εκπαίδευσης, δηλαδή στο τέλος κάθε εποχής [17] (Κεφάλαιο 3 του [3]).



Εικόνα 2 Τελικό στάδιο εφαρμογής αλγορίθμου ανάστροφης διάδοσης σε MLP με δύο κρυφά επίπεδα

### 2.3.3. Μοντέλο Δικτύου Ακτινικών Συναρτήσεων Βάσης (RBF Network)

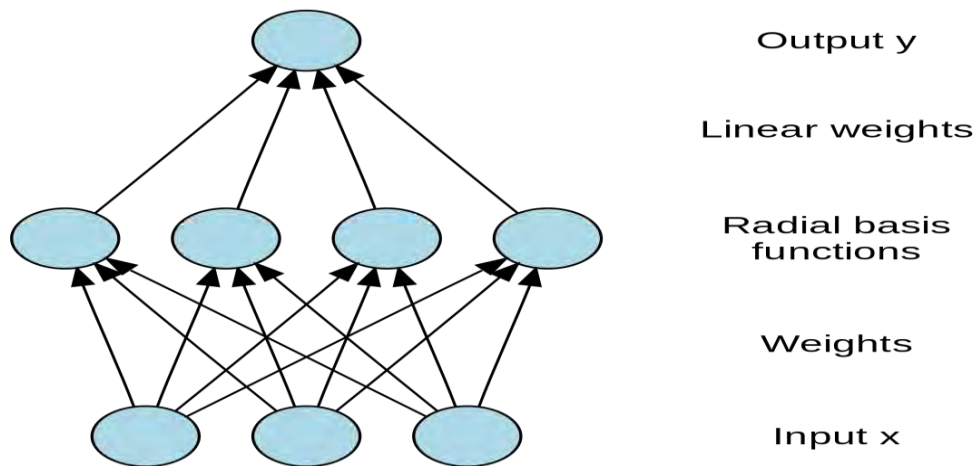
Μία συνάρτηση  $f(x)$  λέγεται ακτινικού τύπου (radial function) αν υπάρχει κάποιο διάνυσμα  $c$  το οποίο καλούμε κέντρο (center ή centroid) και η τιμή της συνάρτησης εξαρτάται μόνο από την απόσταση του  $x$  από το κέντρο.

$$f(x) = f(\|x - c\|)$$

Για έναν κύκλο με κέντρο το  $c$  και ακτίνα  $r = \|x - c\|$  η τιμή της  $f$  είναι σταθερή.

Τα δίκτυα ακτινικών συναρτήσεων βάσης (Radial Basis Function Networks - RBF) αποτελούνται αυστηρά από τρία στρώματα (δύο αν δεν υπολογίσουμε το στρώμα εισόδου). Ο λόγος που δεν χρησιμοποιούνται περισσότερα στρώματα είναι ότι δεν υπάρχει ικανοποιητικός αλγόριθμος εκπαίδευσης για περισσότερα από 3 στρώματα.

Το RBF διαφοροποιείται από το MLP στο γεγονός ότι στο κρυμμένο επίπεδο χρησιμοποιεί ακτινικές συναρτήσεις ενεργοποίησης.



Εικόνα 3 Αρχιτεκτονική Δικτύου RBFNetwork

#### Αλγόριθμος ανάκλησης δικτύου RBF (Κεφάλαιο 5 του [3])

Είσοδοι:

Ένα διάνυσμα εισόδου  $x = [x_1, x_2, \dots, x_n]^T$

Πλήθος νευρώνων: στρώμα εισόδου =  $n$ , κρυφό στρώμα =  $K$ , στρώμα εξόδου =  $m$

Κάθε κρυφός νευρώνας  $i$  έχει δικό του κέντρο  $c_i$  και εύρος συνάρτησης  $\sigma_i$

Εξοδοι:

$y_1, y_2, \dots, y_m$

Μέθοδος:

- Για κάθε κρυφό νευρώνα  $i = 1, \dots, K$  {
  - $a_i = f(\|x - c_i\|, \sigma_i)$}
- Για κάθε νευρώνα εξόδου  $i = 1, \dots, m$  {
  - $y_i = \sum_{j=1}^K w_{ij} a_j + w_{i0}$}

Μία πολύ σημαντική ικανότητα των δικτύων RBF, όπως και στην περίπτωση των δικτύων MLP είναι η δυνατότητα που έχουν να «μαθαίνουν», δηλαδή να προσαρμόζουν τις διάφορες παραμέτρους τους, ώστε να παράγουν εξόδους οι οποίες να βρίσκονται πιο κοντά στις επιθυμητές. Στην περίπτωση των δικτύων RBF, η εκπαίδευσή γίνεται με χρήση τεχνικής επιβλεπόμενης μάθησης (supervised learning). Σκοπός της διαδικασίας μάθησης είναι και εδώ η όσο το δυνατόν καλύτερη προσέγγιση των εξόδων του δικτύου στις επιθυμητές εξόδους των προτύπων, που παρουσιάζονται στο δίκτυο κατά τη φάση της εκπαίδευσης[3].

Η ιδιομορφία που παρουσιάζουν τα δίκτυα ακτινικών συναρτήσεων βάσης σε σχέση με τα δίκτυα πολλών στρωμάτων είναι ότι αποτελούνται από διαφορετικά είδη νευρώνων. Εξαιτίας της ιδιομορφίας τους αυτής, αν και τα δίκτυα αυτού του τύπου είναι δυνατόν να εκπαιδευτούν με αλγόριθμο ανάστροφης διάδοσης ωστόσο κάτι τέτοιο δεν είναι αποτελεσματικό. Αυτό οφείλεται στο γεγονός ότι οι παράμετροι είναι ανομοιογενείς και επηρεάζουν με διαφορετικό τρόπο η καθεμιά τη λειτουργία του δικτύου. Πιο συγκεκριμένα τα δίκτυα ακτινικών συναρτήσεων βάσης έχουν τις εξής παραμέτρους, που πρέπει να υπολογιστούν:

- Τα κέντρα των ακτινικών συναρτήσεων βάσης, τα οποία είναι διανύσματα με διάσταση ίση με τον αριθμό εισόδων του δικτύου.
- Οι ακτίνες των ακτινικών συναρτήσεων βάσης, που είναι καθαροί αριθμοί
- Τα βάρη του γραμμικού στρώματος εξόδου

Έτσι για ένα δίκτυο RBF που έχει  $X$  εισόδους,  $N$  νευρώνες στο κρυμμένο επίπεδο και  $Y$  εξόδους πρέπει να υπολογιστούν  $X \cdot N$  παράμετροι για τα κέντρα,  $N$  για τις ακτίνες και  $(N + 1) \cdot Y$  βάρη και κατώφλια.

### 2.3.3 Η μέθοδος Cross-Validation

Η μέθοδος αυτή είναι ένας τρόπος για να βελτιώσουμε την ικανότητα γενίκευσης μιας μηχανής μάθησης. Η βελτίωση αυτή μπορεί να επιτευχθεί με το να εκπαιδεύσουμε πολλά μοντέλα πάνω σε διαφορετικές οικογένειες συναρτήσεων, να εκτιμήσουμε την ικανότητα γενίκευσης της τελικής συνάρτησης που έμαθε κάθε μοντέλο, και τέλος να επιλέξουμε την συνάρτηση που επιτυγχάνει την καλύτερη γενίκευση.

Για παράδειγμα, έστω ότι έχουμε να εκπαιδεύσουμε το μοντέλο MLP δύο στρωμάτων και διαθέτουμε και ένα σύνολο προτύπων  $P$ . Για να εκτιμήσουμε την ικανότητα γενίκευσης του μοντέλου το αφήνουμε να εκπαιδευτεί με ένα σύνολο προτύπων  $P-S$ . Οπότε, αν το σύνολο των προτύπων είναι 1000 ( $P = 1000$ ) και το σύνολο των προτύπων που αφαιρούνται είναι  $0.1*P$  ( $S = 0.1*P$ ) θα εκπαιδεύσουμε το μοντέλο με 900 από αυτά και με τα υπόλοιπα 100 θα ελέγξουμε το μοντέλο. Άρα μπορούμε να ορίσουμε  $I_{train} = \{1,2,\dots,900\}$  και  $I_{test} = \{901, 902,\dots,1000\}$ . Γενικά θα εκπαιδεύσουμε το δίκτυο χρησιμοποιώντας τα ζεύγη  $(x^{(1)}, d^{(1)}), \dots, (x^{(900)}, d^{(900)})$  και για τον έλεγχο τα ζεύγη  $(x^{(901)}, d^{(901)}), \dots, (x^{(1000)}, d^{(1000)})$ .

Όταν ολοκληρωθεί η εκπαίδευση το μοντέλο μαθαίνει μια συνάρτηση  $f(x, \theta)$ , όπου  $\theta$  είναι το διάνυσμα των παραμέτρων του μοντέλου (Κεφάλαιο 10 του [3]).

Στην μέθοδο Cross-Validation υπάρχουν δυο είδη σφαλμάτων.

- Το σφάλμα εκπαίδευσης: Δείχνει το πόσο καλά προσεγγίζει τα πρότυπα το μοντέλο που εκπαιδεύτηκε.

$$J_{train} = \frac{1}{|I_{train}|} \sum_{i \in I_{train}} \|d^{(i)} - f(x^{(i)}, \theta)\|^2$$

- Το σφάλμα ελέγχου: Δείχνει το πόσο καλά γενικεύει το μοντέλο στα πρότυπα που αφαιρέθηκαν κατά την εκπαίδευση.

$$J_{test} = \frac{1}{|I_{test}|} \sum_{i \in I_{test}} \|d^{(i)} - f(x^{(i)}, \theta)\|^2$$

Σε περίπτωση που επιλέγαμε άλλο σύνολο προτύπων για εκπαίδευση και έλεγχο του μοντέλου είναι πολύ πιθανόν να παίρναμε διαφορετικά σφάλματα  $J_{train}$  και  $J_{test}$ . Για να υπάρξει λύση σε αυτό χωρίζουμε το σύνολο των προτύπων σε  $k$  ομάδες με  $k = 1/S$  και με  $S*P$  πρότυπα η κάθε μία και εκπαιδεύουμε το μοντέλο  $k$  φορές. Την  $i$ -οστή φορά που εκπαιδεύουμε το μοντέλο θέτουμε την  $i$ -οστή ομάδα ως σύνολο ελέγχου (Κεφάλαιο 10 του [3]). Έτσι η ικανότητα γενίκευσης του μοντέλου είναι:

$$\overline{J_{test}} = \frac{1}{N} \sum_{i=1}^N J_{test}(i)$$

### Αλγόριθμος Cross-Validation [3]:

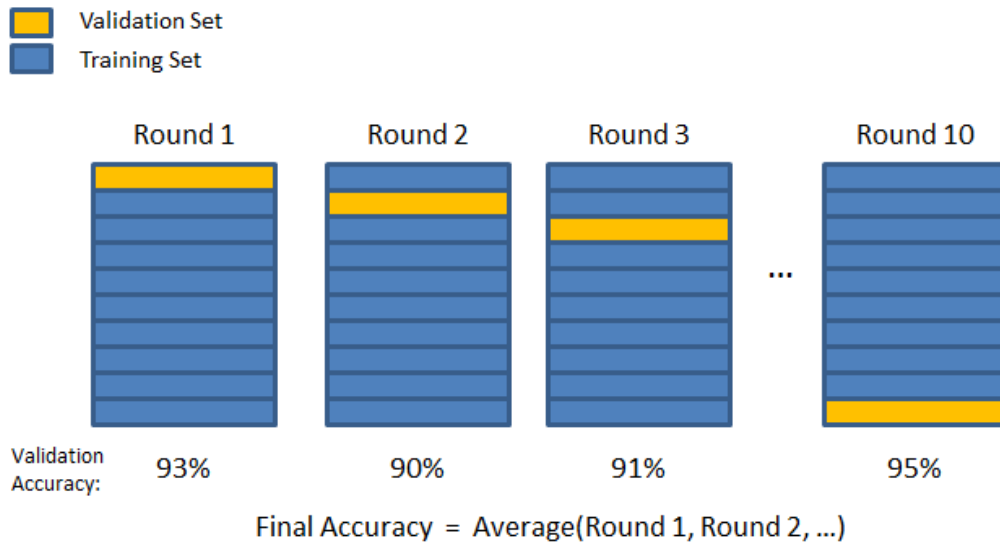
Είσοδος:  $P$  πρότυπα  $\chi^{(1)}, \dots, \chi^{(P)}$ , με στόχους  $d^{(1)}, \dots, d^{(P)}$

Ένα συγκεκριμένο νευρωνικό μοντέλο

Εξοδος: Εκτίμηση  $\bar{J}_{test}$  της ικανότητας γενίκευσης του μοντέλου

Μέθοδος:

- Χώρισε τα πρότυπα σε  $N$  ίσες ομάδες,  $G_1, G_2, \dots, G_N$
- Για κάθε επανάληψη  $i = 1, \dots, N$  {
  - Εκπαίδευσε το μοντέλο χρησιμοποιώντας τα πρότυπα από όλες τις ομάδες εκτός από την  $G_i$
  - Υπολόγισε το σφάλμα ελέγχου  $\bar{J}_{test}$  για τα πρότυπα της ομάδας  $G_i$
- Υπολόγισε το  $\bar{J}_{test}$  από το μέσο όρο  $\bar{J}_{test}(i)$



Εικόνα 4 10 folds Cross-Validation



## ΚΕΦΑΛΑΙΟ 3

### ΑΠΟΤΕΛΕΣΜΑΤΑ – ΑΞΙΟΛΟΓΙΣΗ ΑΠΟΔΟΣΗΣ

#### 3.1 ΜΕΤΡΙΚΕΣ ΕΚΤΙΜΗΣΗΣ ΑΠΟΔΟΣΗΣ

Σε ένα σύστημα ανάκτησης της πληροφορίας τίθεται πάντα το ζήτημα της επάρκειας των αποτελεσμάτων στα ερωτήματα των χρηστών. Η επάρκεια των αποτελεσμάτων αποκαλείται *αποτελεσματικότητα* (effectiveness) του συστήματος. Στο Κεφάλαιο αυτό παρουσιάζονται οι βασικές μετρικές εκτίμησης απόδοσης (αποτελεσματικότητας) ενός συστήματος διαχείρισης και ανάκτησης πληροφορίας. Οι μετρικές αυτές είναι η *ανάκληση* και η *ακρίβεια* και μας κατατοπίζουν για το πότε ένα σύστημα είναι καλύτερο από ένα άλλο, συνεπώς μας δείχνουν ποιο σύστημα έχει καλύτερη απόδοση [5]. Η αποτελεσματικότητα ενός συστήματος εξαρτάται από πολλούς παράγοντες όπως :

- Τη συλλογή εγγράφων που διαχειρίζεται το σύστημα
- Τα ερωτήματα των χρηστών
- Το μοντέλο ανάκτησης που χρησιμοποιείται

##### 3.1.1 Βασικές Έννοιες Απόδοσης

Έχοντας την δυνατότητα να διαπιστώσουμε αν ένα έγγραφο  $d$  είναι σχετικό ή μη σχετικό με μία συλλογή  $C$  για ένα ερώτημα  $q$ , μετά την επεξεργασία του ερωτήματος το σύστημα επέστρεψε δεδομένα που αποτελούν το σύνολο της απάντησης  $A$ . Έχοντας επίσης ένα σύνολο  $R$  που περιέχει όλα τα έγγραφα της συλλογής  $C$  που είναι σχετικά με το ερώτημα με  $A_r$  συμβολίζουμε το υποσύνολο του  $A$  που περιέχει τα σχετικά ως προς το  $q$  έγγραφα.

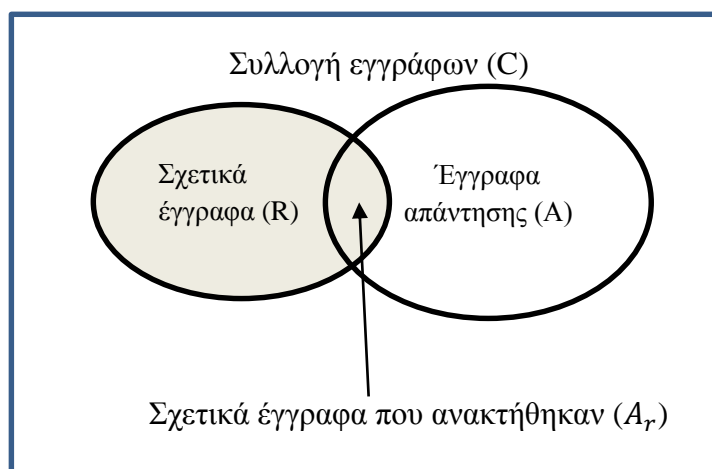


Figure 1 Αναπαράσταση συνόλου εγγράφων

Στο παρακάτω πίνακάκι (Πίνακας 3.1) δίνονται οι χαρακτηρισμοί των εγγράφων για ένα ερώτημα. Η ιδανική περίπτωση φαίνεται να είναι τα έγγραφα να ανήκουν στο κάτω αριστερό τεταρτημόριο και στο πάνω δεξιό τεταρτημόριο.

	Ανακτηθέν	Μη ανακτηθέν
Μη σχετικό	Μη σχετικά έγγραφα που έχουν ανακτηθεί ( $A - A_r$ )	Μη σχετικά έγγραφα που δεν έχουν ανακτηθεί ( $C - (R \cup A)$ )
Σχετικό	Σχετικά έγγραφα που έχουν ανακτηθεί ( $A_r$ )	Σχετικά έγγραφα που δεν έχουν ανακτηθεί ( $R - A$ )

Πίνακας 3 Κατηγοριοποίηση εγγράφων

### 3.1.2 Ανάκληση και Ακρίβεια (Recall and Precision)

Ανάκληση (Recall) είναι ο λόγος του αριθμού των σχετικών εγγράφων που ανακτήθηκαν προς το συνολικό αριθμό σχετικών εγγράφων της συλλογής :

$$recall(q) = \frac{\text{αριθμός σχετικών εγγράφων που ανακτήθηκαν}}{\text{αριθμός σχετικών εγγράφων στη συλλογή}} = \frac{|A_r|}{|R|}$$

Ουσιαστικά η ανάκληση μετράει το ποσοστό των σχετικών εγγράφων που μπόρεσε το σύστημα να ανακτήσει σε σχέση με όλα τα σχετικά έγγραφα που υπάρχουν στη συλλογή.

Ακρίβεια (Precision) είναι ο λόγος του αριθμού των σχετικών εγγράφων που ανακτήθηκαν προς τον συνολικό αριθμό των εγγράφων που ανακτήθηκαν :

$$precision(q) = \frac{\text{αριθμός σχετικών εγγράφων που ανακτήθηκαν}}{\text{αριθμός εγγράφων που ανακτήθηκαν}} = \frac{|A_r|}{|A|}$$

Η καλύτερη δυνατή ανάκληση και ακρίβεια που μπορεί να επιτευχθεί σε ένα σύστημα ανάκτησης είναι 100%.

Παράδειγμα : Ένα σύστημα για ένα ερώτημα πρέπει να επιστρέψει 100 σχετικά έγγραφα. Το σύστημα τελικά επιστρέφει 125 έγγραφα που από αυτά σχετικά είναι τα 80. Άρα  $recall = \frac{80}{100} = 0.80 = 80\%$  και  $precision = \frac{80}{125} = 0.64 = 64\%$

Γενικά όσο ζητάμε από ένα σύστημα να μας επιστρέψει περισσότερα σχετικά έγγραφα , τόσο αυξάνουμε και τις πιθανότητες να μειωθεί η ακρίβεια κατά την ανάκτηση. Δηλαδή όταν το ένα μέγεθος αυξάνεται τότε το άλλο παρουσιάζει τάση μείωσης.

Στο Σχήμα 3.2 παρουσιάζονται οι περιπτώσεις ανάκλησης – ακρίβειας σε ένα σύστημα ανάκτησης. Πιο αναλυτικά η πάνω αριστερά περίπτωση βλέπουμε ότι έχει χαμηλή ανάκληση και χαμηλή ακρίβεια που σημαίνει ότι στην χειρότερη περίπτωση το σύστημα δεν θα εντοπίσει κανένα σχετικό έγγραφο. Άρα αυτή η περίπτωση θα πρέπει αποφεύγεται. Πάνω δεξιά βλέπουμε μια περίπτωση όπου η ακρίβεια είναι στο 100% και η ανάκληση είναι χαμηλή, που σημαίνει ότι το σύστημα κατάφερε να επιστρέψει καλής ποιότητας αποτελέσματα αλλά δεν εντόπισε πολλά σχετικά έγγραφα. Κάτω αριστερά στο σχήμα παρουσιάζεται η αντίθετη περίπτωση, δηλαδή η ανάκληση είναι στο 100% και η ακρίβεια είναι χαμηλή, πράγμα που σημαίνει πως το σύστημα εντόπισε όλα τα σχετικά έγγραφα αλλά και πολλά άλλα έγγραφα που δεν είναι σχετικά. Τέλος κάτω δεξιά φαίνεται η ιδανική περίπτωση όπου ανάκληση και ακρίβεια είναι σε πολύ υψηλά επίπεδα.

Συμπερασματικά καταλήγουμε πως γενικά ένα σύστημα ανάκλησης και ακρίβειας κυμαίνεται μεταξύ των περιπτώσεων πάνω δεξιά και κάτω αριστερά.

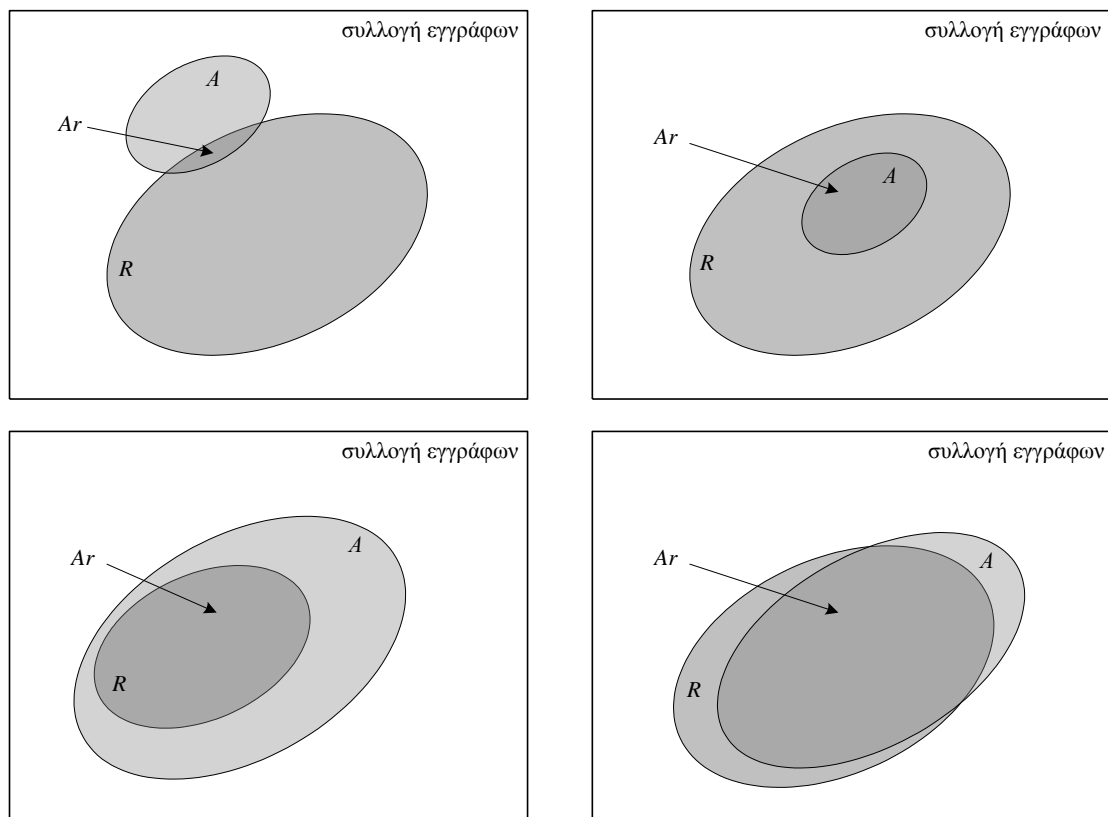


Figure 2 Συνδυασμοί ανάκλησης – ακρίβεια

Οι μηχανές αναζήτησης είναι ένα κλασικό παράδειγμα ενός συστήματος ανάκτησης εγγράφων. Αν εξετάσουμε τα αποτελέσματά τους στον παγκόσμιο ιστό βλέπουμε πως εμφανίζονται κατά ομάδες. Έτσι τα έγγραφα από ένα σημείο και μετά είναι σχετικά

αλλά έχουν χαμηλό βαθμό, άρα μικρή σχετικότητα ως προς το ερώτημα (Κεφάλαιο 2 του [5]).

Για παράδειγμα ας θεωρήσουμε ότι ως για ένα ερώτημα  $q$  τα σχετικά έγγραφα είναι τα  $d_1, d_4, d_5, d_8$  και  $d_9$ . Το σύστημα δίνει ως αποτέλεσμα τα  $d_4, d_5, d_2, d_3, d_7, d_9, d_8, d_6$  και  $d_1$ . Στη συνέχεια καταγράφουμε τις τιμές τις ανάκλησης και της ακρίβειας που παίρνουμε ως αποτέλεσμα και δίνονται στον παρακάτω πίνακα (Κεφάλαιο 2 του [5]).

$\alpha / \alpha$	έγγραφο	σχετικό	ανάκληση	ακρίβεια
1	$d_4$	<i>NAI</i>	20%	100%
2	$d_5$	<i>NAI</i>	40%	100%
3	$d_2$	<i>OXI</i>	40%	66%
4	$d_3$	<i>OXI</i>	40%	50%
5	$d_7$	<i>OXI</i>	40%	40%
6	$d_9$	<i>NAI</i>	60%	50%
7	$d_8$	<i>NAI</i>	80%	57%
8	$d_6$	<i>OXI</i>	80%	50%
9	$d_1$	<i>NAI</i>	100%	55%

Πίνακας 4 Τιμές ανάκλησης και ακρίβειας

Με βάση αυτόν τον πίνακα μπορούμε να δημιουργήσουμε την καμπύλη ανάκλησης-ακρίβειας η οποία δείχνει τον τόπο μεταβολής της μιας μετρικής σε σχέση με την άλλη.

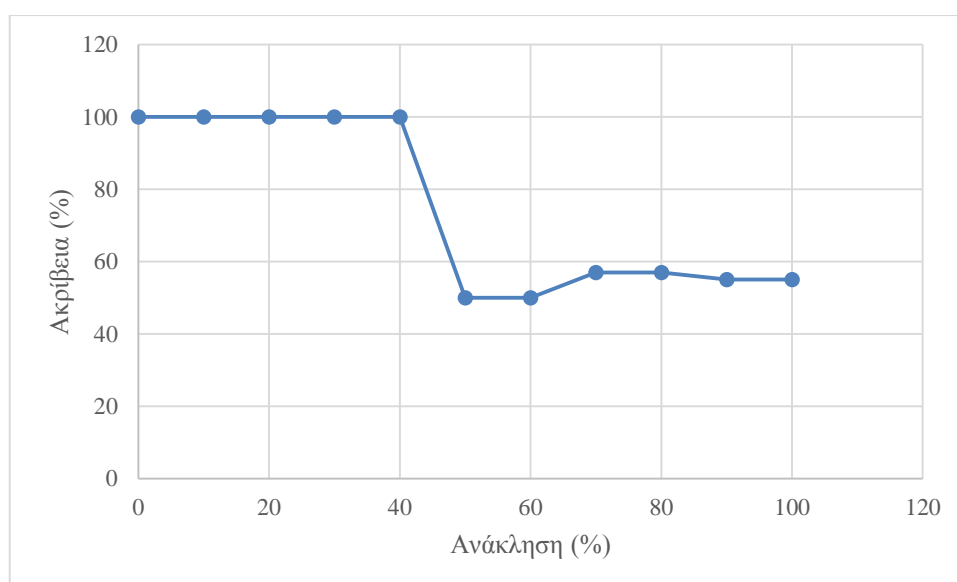


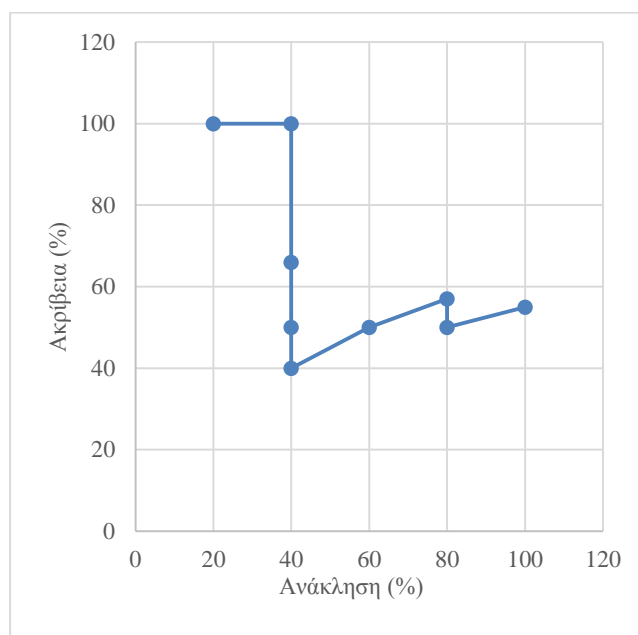
Figure 3 Αρχική καμπύλη ανάκλησης και ακρίβειας

Η καμπύλη του Σχήματος 3.3 δεν χρησιμοποιείται, αλλά συνήθως διαμορφώνεται κατάλληλα χρησιμοποιώντας πάντα 11 σημεία ανάκλισης (0%, 10%, ... , 100%) και σχηματίζεται αντιστοιχίζοντας τις τιμές της ακρίβειας με αυτές τις τιμές ανάκλισης. Όταν δεν υπάρχει αντίστοιχη τιμή ανάκλισης χρησιμοποιείται μία τεχνική που ονομάζεται *τεχνική της παρεμβολής* (interpolation). Με την τεχνική αυτή αντιστοιχίζουμε στην τιμή ανάκλισης που δεν υπάρχει την μέγιστη τιμή ακρίβειας που υπάρχει για το αμέσως επόμενο επίπεδο ανάκλισης (Κεφάλαιο 2 σελ. 29-30 του [5]). Δηλαδή αν  $r_i$  το  $i$ -οστό επίπεδο ανάκλισης, με  $p(r)$  την πραγματική τιμή της ακρίβειας για το επίπεδο της ανάκλισης  $r$  και με  $\pi(i)$  την παρεμβαλλόμενη τιμή της ακρίβειας για το επίπεδο  $r_i$  τότε η τιμή του  $\pi(r_i)$  θα είναι:

$$\pi(r_i) = \max_{r_i \leq r \leq r_{i+1}} \{p(r)\}$$

Οι τιμές της ακρίβειας για τα 11 επίπεδα της ανάκλισης παρουσιάζονται στον ακόλουθο πίνακα (Πίνακας 3.3). Από τον πίνακα αυτό συμπεραίνουμε πως η καμπύλη ανάκλισης – ακρίβειας θα έχει αλλάξει, κάτι που φαίνεται στο Σχήμα 3.4.

<i>α/α</i>	<i>ανάκλιση</i>	<i>ακρίβεια</i>
1	0%	100%
2	10%	100%
3	20%	100%
4	30%	100%
5	40%	100%
6	50%	50%
7	60%	50%
8	70%	57%
9	80%	57%
10	90%	55%
11	100%	55%



Πίνακας 5 Τιμές ακρίβειας με παρεμβολή για τα 11 επίπεδα ανάκλισης

Figure 4 Καμπύλη ανάκλισης - ακρίβειας με την τεχνική της παρεμβολής

### 3.1.3 Συνολική Απόδοση και σύγκριση συστημάτων ανάκτησης

Στο παράδειγμα της υπό-ενότητας 3.1.2 παρουσιάστηκε η ανάλυση της ανάκλησης – ακρίβειας για ένα μόνο ερώτημα σε ένα σύστημα ανάκτησης πληροφορίας. Για τον υπολογισμό της συνολικής απόδοσης ενός συστήματος απαιτείται μια διαφορετική διαδικασία.

Η τεχνική που χρησιμοποιείται σε αυτή την περίπτωση είναι η χρήση της μέσης τιμής της ακρίβειας για τα διαφορετικά επίπεδα ανάκλησης. Με τον τρόπο αυτό παράγεται μία μόνο καμπύλη που αντιπροσωπεύει όλα τα ερωτήματα του συστήματος, έτσι η μελέτη της συνολικής απόδοσης και η σύγκριση 2 συστημάτων ανάκτησης μπορεί να επιτευχθεί πιο εύκολα. Πιο αναλυτικά, έστω ότι έχουμε  $N_q$  διαφορετικά ερωτήματα για τα οποία γνωρίζουμε τις τιμές της ακρίβειας για το καθένα για τα 11 επίπεδα ανάκλησης που πήραμε εφαρμόζοντας την διαδικασία της παρεμβολής. Αν με  $\overline{P}(r)$  συμβολίσουμε την μέση τιμή για το επίπεδο ανάκλησης  $r$  και με  $P(r)$  την τιμή της ακρίβειας του  $j$ -οστού ερωτήματος για το επίπεδο ανάκλησης  $r$  τότε η μέση ακρίβεια είναι η εξής:

$$\overline{P}(r) = \sum_{j=1}^{N_q} \frac{P_j(r)}{N_q}$$

Οι καμπύλες που απεικονίζονται στο Σχήμα 3.5 δείχνουν την μορφή που έχει μια τυπική γραφική παράσταση της ακρίβειας σε σχέση με την ανάκληση για τα συστήματα A και B. Οι δυο καμπύλες διασταυρώνονται, έτσι δεν είναι προφανές πιο από τα δυο συστήματα έχει καλύτερη απόδοση. Για να λυθεί αυτό, πρέπει να θα πρέπει να ληφθούν υπόψη και άλλες παράμετροι όπως η *μονότιμη σύνοψη* (single valued summary), ο *αρμονικός μέσος* (harmonic mean) και η Ε-μετρική (Κεφ. 2 σελ 31 του [5]).

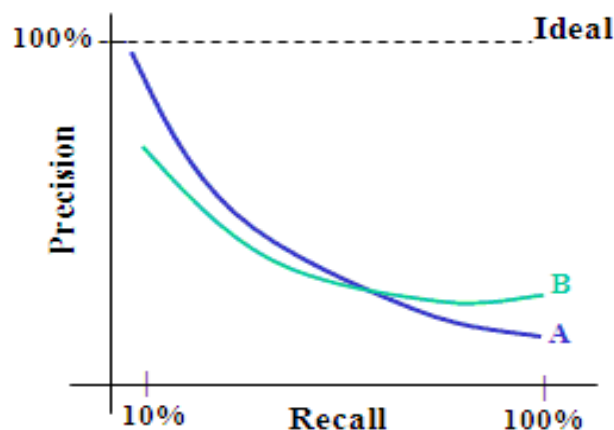


Figure 5 Σύγκριση συστημάτων

## 3.2 ΑΠΟΤΕΛΕΣΜΑΤΑ ΑΠΟΔΟΣΗΣ ΣΥΣΤΗΜΑΤΟΣ

Για την διαχείριση των νομικών κειμένων χρησιμοποιήθηκε ένα σύστημα ανάκτησης πληροφορίας και κατηγοριοποίησης των νομικών αποφάσεων. Στο κεφάλαιο αυτό αναλύονται τα αποτελέσματα για την αποτελεσματικότητα στην εξαγωγή επικεφαλίδας, στον εντοπισμό των αναφορών καθώς και στην κατηγοριοποίηση των νομικών αποφάσεων με τα διάφορα μοντέλα που χρησιμοποιήθηκαν.

### 3.2.1 Ανάκληση και Ακρίβεια Επικεφαλίδας

Κατά την εξαγωγή της επικεφαλίδας από τα κείμενα κρίθηκε απαραίτητη η μελέτη της ανάκλησης και ακρίβειας για την εκτίμηση της απόδοσης του συστήματος.

Πιο αναλυτικά, η μελέτη αυτή έγινε ξεχωριστά για κάθε μια από τις ακόλουθες κατηγορίες :

- Αριθμός
- Προεδρεύων ο αντιπρόεδρος
- Εισηγητής ο αεροπαγίτης
- Εισαγγελεύς
- Δικηγόρος

Για την υλοποίηση του συστήματος χρησιμοποιήθηκε ένα σύνολο νομικών αποφάσεων, εξετάστηκαν τα αποτελέσματά του (Figure 6) και για να έχουμε μια πιο εμπειρισταωμένη εκτίμηση χρησιμοποιήθηκαν επιπλέον (νέα) κείμενα για την εξέταση της ανάκτησης της επικεφαλίδας (Figure 7).

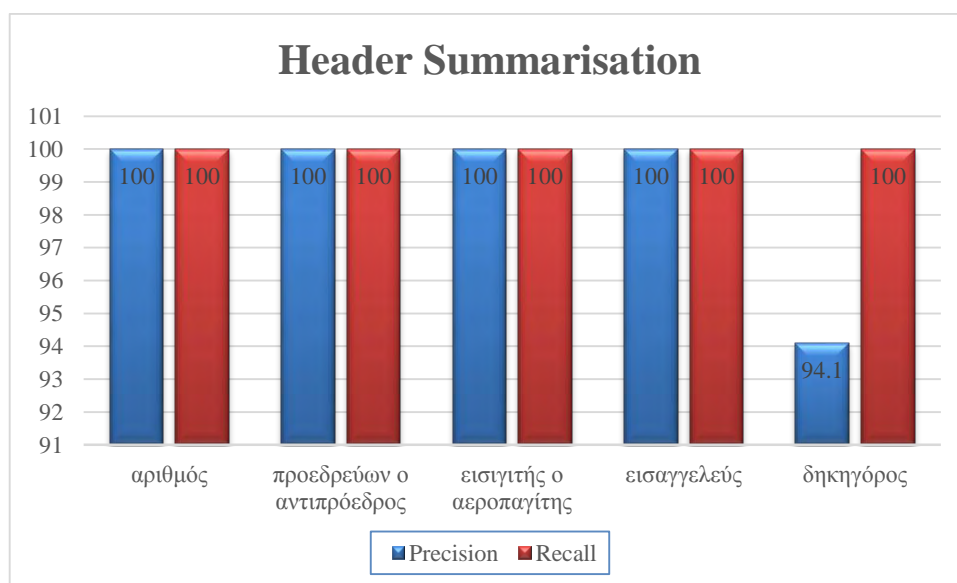


Figure 6 Ανάκληση και ακρίβεια για Header Summarization για τα κείμενα που χρησιμοποιήθηκαν για την υλοποίηση του συστήματος.

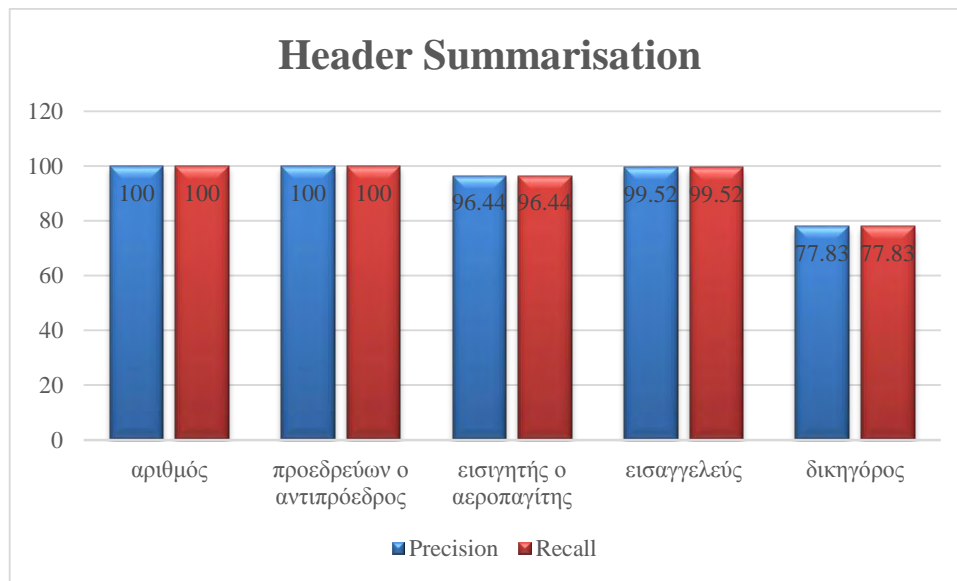


Figure 7 Ανάκληση και ακρίβεια για το Header Summarization

Παρατηρούμε πως στο Figure 6 η ανάκληση και η ακρίβεια είναι στο 100% για τις κατηγορίες αριθμός, προεδρεύων ο αντιπρόεδρος, εισηγητής ο αεροπαγίτης και εισαγγελεύς ενώ για την κατηγορία δικηγόρος η ακρίβεια είναι στο 94.1% και η ανάκληση στο 100%. Στο Figure 7 που είναι τα αποτελέσματα για τα νέα κείμενα παρατηρούμε πως διατηρούνται σε υψηλά επίπεδα η ανάκληση και η ακρίβεια όλων των κατηγοριών. Η κατηγορίες αριθμός και προεδρεύων ο αντιπρόεδρος παραμένουν στο 100%. Στην ανάκληση και η ακρίβεια των υπόλοιπων κατηγοριών παρατηρείτε πτώση. Η κατηγορία προεδρεύων ο αντιπρόεδρος έχει και για τις 2 μετρικές 96.4%, ο εισαγγελεύς 99.52 ενώ η κατηγορία δικηγόρος παρουσιάζει την μεγαλύτερη μείωση με τιμές 77.83%.

### 3.2.2 Εντοπισμός Αναφορών

Για την εκτίμηση της απόδοσης του συστήματος στον εντοπισμό των αναφορών υπολογίστηκαν οι μετρικές απόδοσης για κάθε κατηγορία αναφορών ξεχωριστά. Η εξαγωγή αυτών έγινε από ένα κείμενο είκοσι νομικών αποφάσεων που χρησιμοποιήθηκαν για την εξαγωγή κανόνων κατά την υλοποίηση του συστήματος ανάκτησης αναφορών.

Στο παρακάτω πίνακα (Πίνακας 6) και το Figure 8 φαίνονται τα αποτελέσματα της ανάκλησης και της ακρίβειας για κάθε κατηγορία.



<i>Κατηγορίες νομικών αναφορών</i>	<i>Ανάκληση</i>	<i>Ακρίβεια</i>
<i>Συντ</i>	<i>100%</i>	<i>100%</i>
<i>ΣΔΟΕ</i>	<i>100%</i>	<i>100%</i>
<i>ΚΠολΔ</i>	<i>100%</i>	<i>100%</i>
<i>ΔΣΑΠΔ</i>	<i>100%</i>	<i>100%</i>
<i>ΕΣΔΑ</i>	<i>81.81%</i>	<i>100%</i>
<i>ΠΚ</i>	<i>90.62%</i>	<i>100%</i>
<i>ΚΠΔ</i>	<i>89.81%</i>	<i>100%</i>
<i>Ν.</i>	<i>100%</i>	<i>90.66%</i>
<i>ν.δ</i>	<i>80%</i>	<i>100%</i>
<i>Δικαστήρ</i>	<i>100%</i>	<i>100%</i>
<i>Πλημ</i>	<i>83.33%</i>	<i>100%</i>
<i>Εφετ</i>	<i>88.23%</i>	<i>100%</i>
<i>Πρωτοδικ</i>	<i>100%</i>	<i>100%</i>
<i>ΕΣυμβ</i>	<i>100%</i>	<i>100%</i>
<i>ΑΚ</i>	<i>100%</i>	<i>100%</i>

Πίνακας 6 Τιμές ανάκλησης και ακρίβειας για όλες τις κατηγορίες αναφορών από τα κείμενα που χρησιμοποιήθηκαν για την υλοποίηση του συστήματος

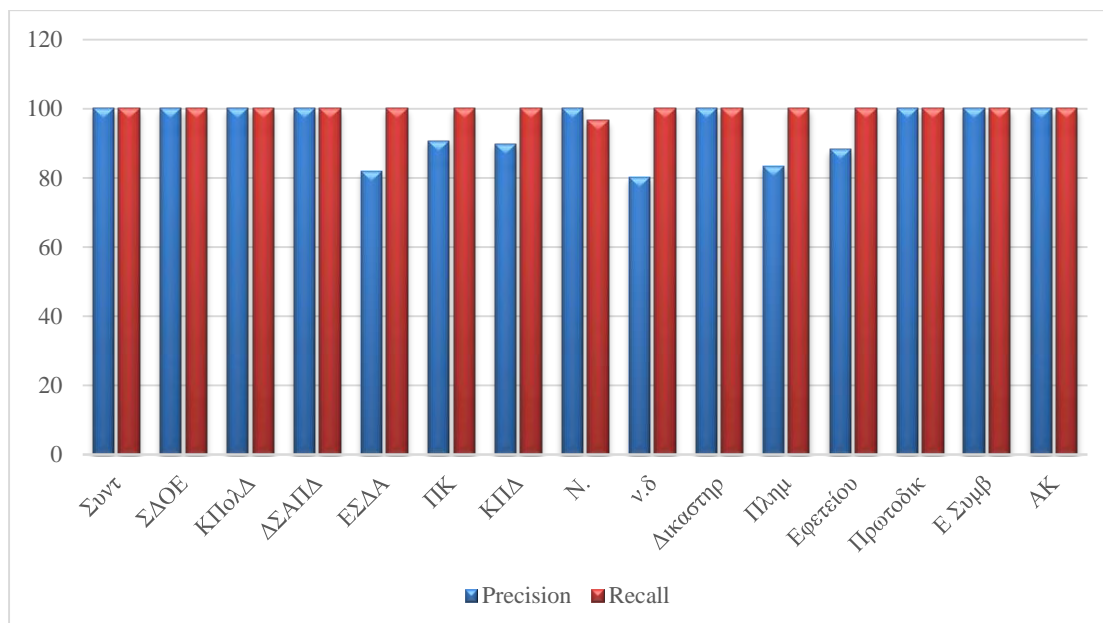


Figure 8 Ανάκληση και ακρίβεια στον εντοπισμό των αναφορών για τα κείμενα που χρησιμοποιήθηκαν για την υλοποίηση του συστήματος

Στο Figure 8 βλέπουμε τα ποσοστά των μετρικών να κυμαίνονται σε υψηλά επίπεδα με μόλις μια κατηγορία (N.) να παρουσιάζει ανάκληση (κάτω από 100%) στο 96.66% και τις κατηγορίες ΕΣΔΑ, ΠΚ ΚΠΔ,ν.δ, Πλημ, Εφετ να έχουν ακρίβεια κάτω από 100% με χαμηλότερη την κατηγορία ν.δ στο 80%.

Για πιο ακριβή αποτύπωση των μετρικών απόδοσης επιλέχθηκε ένα τυχαίο σύνολο νομικών αποφάσεων με τα αποτελέσματα του συστήματος να φαίνονται στον Πίνακα 7 και στο Figure 9.

<b>Κατηγορίες νομικών αναφορών</b>	<b>Ανάκληση</b>	<b>Ακρίβεια</b>
Συντ	100%	93.33%
ΣΔΟΕ	0%	0%
ΚΠολΔ	100%	68%
ΔΣΑΠΔ	0%	0%
ΕΣΔΑ	100%	100%
ΠΚ	100%	80%
ΚΠΔ	98.18%	98.18%
N.	96.77%	73.61%
ν.δ	100%	90.9%
Δικαστήρ	0%	0%
Πλημ	100%	78.57%
Εφετ	100%	73.68%
Πρωτοδικ	100%	96.77%
ΕΣυμβ	100%	100%
ΑΚ	95%	95%

**Πίνακας 7 Τιμές ανάκλησης και ακρίβειας για τον εντοπισμό των αναφορών**

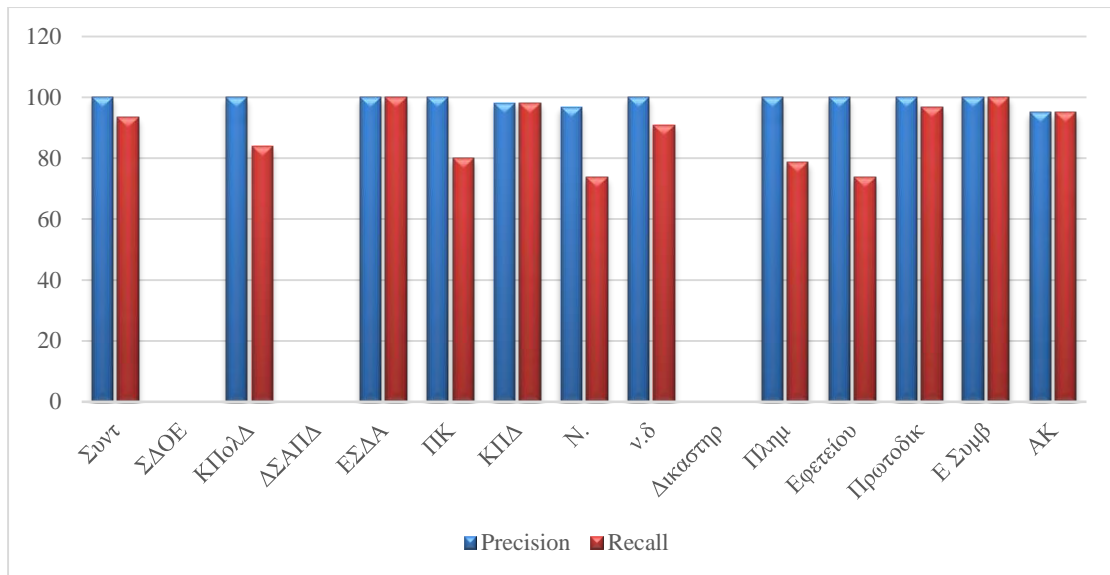


Figure 9 Ανάκληση και ακρίβεια στον εντοπισμό των αναφορών

Στο Figure 9 αρχικά παρατηρούμε πως οι κατηγορίες ΣΔΟΕ, ΔΣΑΠΔ και Δικαστηρ παρουσιάζουν μηδενική αποτελεσματικότητα, πράγμα που σημαίνει πως δεν υπήρχαν στο σύνολο αποφάσεων που εξετάστηκε και συνεπώς δεν ανακτήθηκαν από το σύστημα. Αυτό της κατατάσσει στις μη σημαντικές κατηγορίες για τον γενικό χαρακτηρισμό και την απόδοση του κειμένου σε μία κατηγορία αποφάσεων. Δηλαδή οι εν λόγω κατηγορίες θα μπορούσαν κάλλιστα να μην χρησιμοποιηθούν κατά την διαδικασία της κατηγοριοποίησης των νομικών αποφάσεων, χωρίς να επηρεάσουν το τελικό αποτέλεσμα.

Για της υπόλοιπες κατηγορίες βλέπουμε πως η χαμηλότερη ακρίβεια που σημειώνεται είναι στο 95% ενώ η χαμηλότερη ανάκληση στο 73.68%.

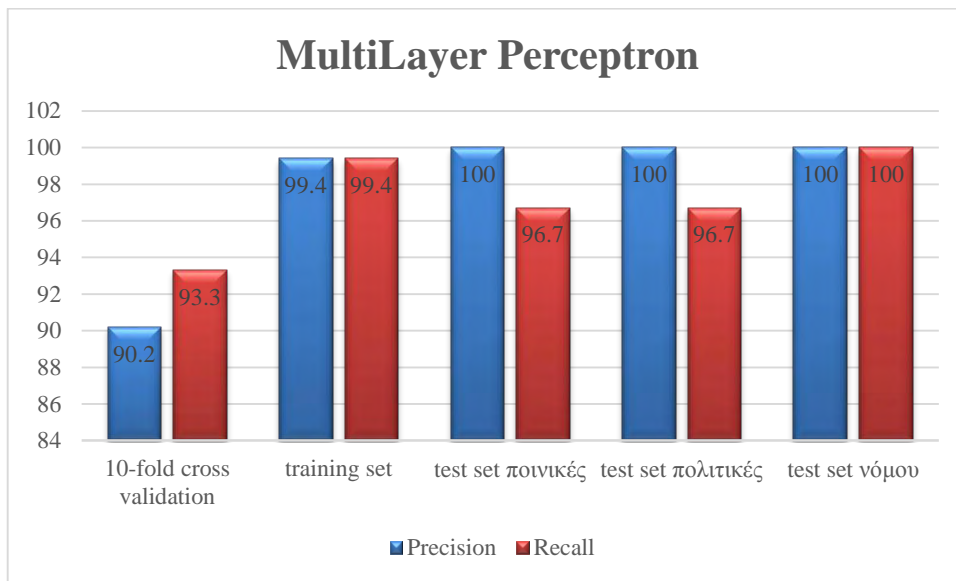
### 3.2.3 Ταξινόμηση Κειμένων

Η ταξινόμηση κειμένων, όπως έχει προαναφερθεί στην ενότητα 2.1.4 είναι μια διαδικασία που μας επιτρέπει την οργάνωση ενός σώματος κειμένων με βάση κάποιο κοινό χαρακτηριστικό, διευκολύνοντας την αναζήτηση στο σώμα αυτό.

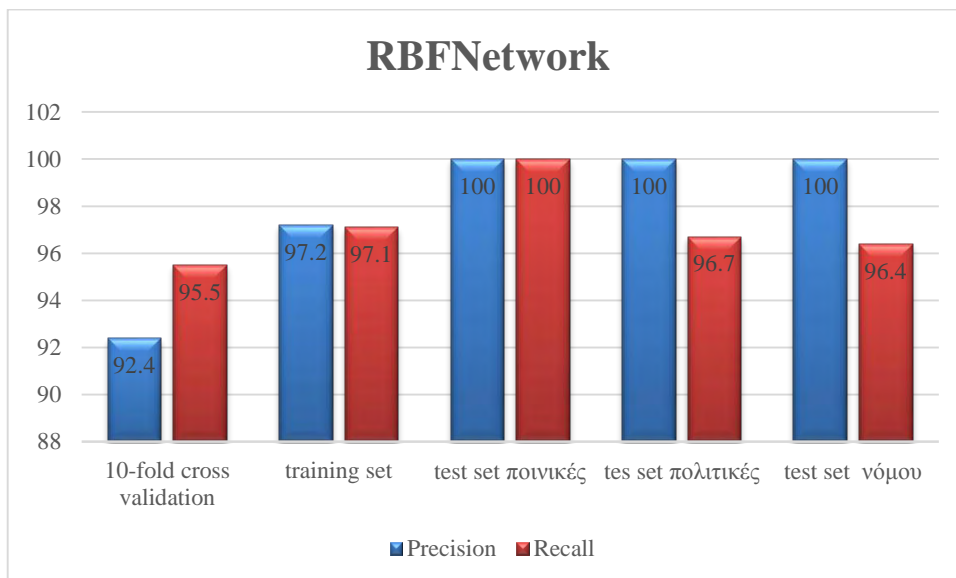
Για την ταξινόμηση χρησιμοποιήθηκαν τα παρακάτω μοντέλα :

- Multilayer Perceptron
- RBFNetwork
- SMO
- Simple Logistic

Στα Figure 10, 11, 12 και 13 φαίνονται οι τιμές ανάκλησης και ακρίβειας που έδωσαν τα μοντέλα κατά την εκπαίδευσή τους καθώς και κατά τον έλεγχο απόδοσής τους ανά κατηγορία νομικών κειμένων.



**Figure 10** Ανάκληση και ακρίβεια με το μοντέλο Multilayer Perceptron



**Figure 11** Ανάκληση και ακρίβεια με το μοντέλο RBF Network

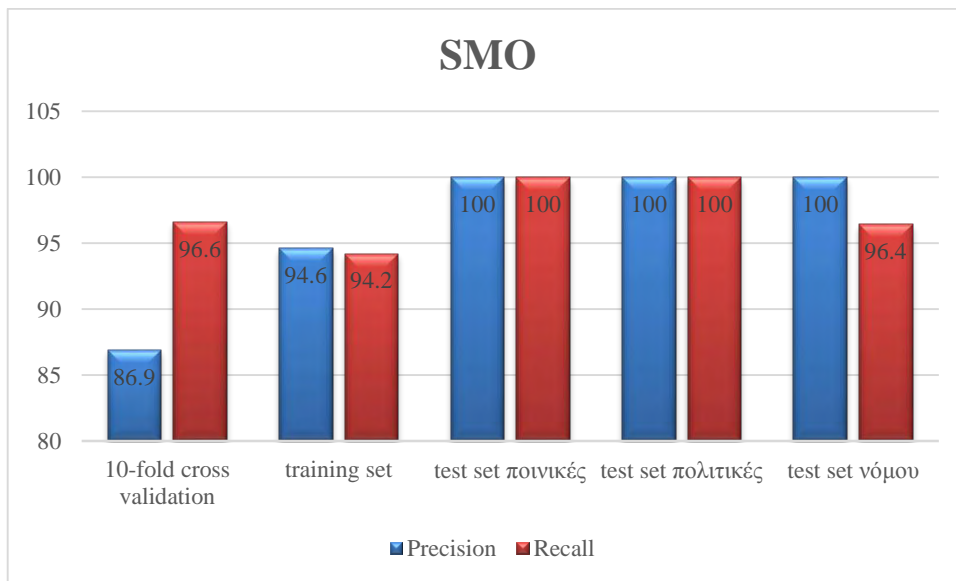


Figure 12 Ανάκληση και ακρίβεια με το μοντέλο SMO

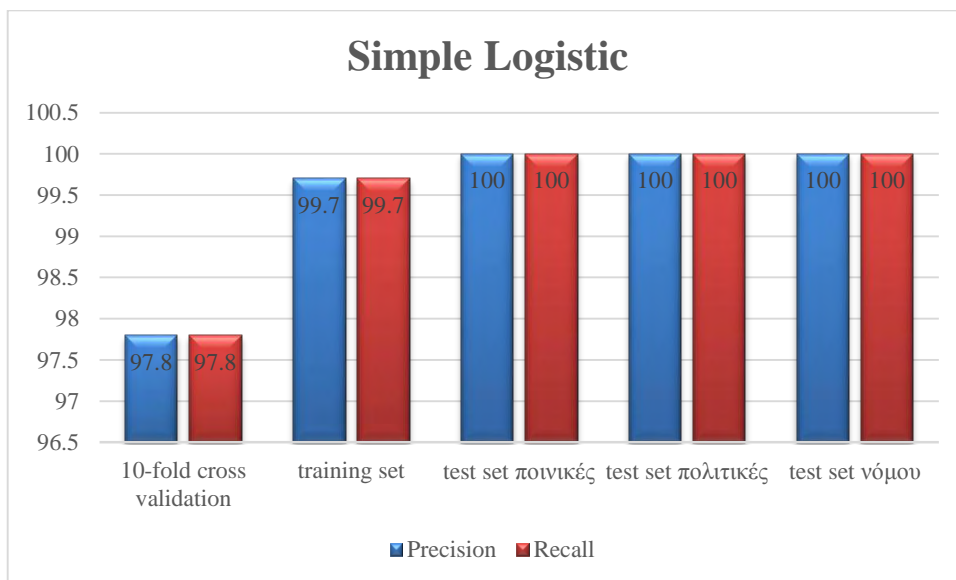


Figure 13 Ανάκληση και ακρίβεια με το μοντέλο Simple Logistic

Εφαρμόζοντας την τεχνική 10-fold cross validation παρατηρούμε πως το μοντέλο Simple Logistic παρουσιάζει την καλύτερη ικανότητα γενίκευσης.

Η εκπαίδευση με το μοντέλο Multilayer Perceptron παρουσιάζει ανάκληση και ακρίβεια στο 99.4%, με το RBFNetwork έχουμε ακρίβεια στο 97.2% και ανάκληση στο 97.1%, με το SMO παίρνουμε ακρίβεια 94.6% και ανάκληση 94.2% και με το μοντέλο Simple Logistic καταγράφηκαν τα μεγαλύτερα ποσοστά κατά την εκπαίδευση με τιμή 99.7 και για τις 2 μετρικές εκτίμησης αποτελεσματικότητας.

Για τον έλεγχο των μοντέλων χρησιμοποιήθηκε ένα σύνολο νομικών αποφάσεων για κάθε κατηγορία (ποινικές, πολιτικές, νόμου 3068) και τα αποτελέσματα ήταν τα εξής:

- Για την κατηγορία «ποινικές» όλα τα μοντέλα που χρησιμοποιήθηκαν έδωσαν ακρίβεια 100% και ανάκληση 96.7% το Multilayer Perceptron και όλα τα υπόλοιπα 100%.
- Για την κατηγορία «πολιτικές» τα μοντέλα Multilayer Perceptron και RBFNetwork παρουσιάζουν ίδιες τιμές με ακρίβεια στο 100% και ανάκληση στο 96.7%, ενώ τα SMO και Simple Logistic ανάκληση και ακρίβεια στο 100%.
- Για την κατηγορία «νόμου 3068» το Multilayer Perceptron έδωσε 100% και για τις 2 μετρικές, τα RBFNetwork και SMO έδωσαν 100% ακρίβεια και 96.4% ανάκληση και ο Simple Logistic 100% ακρίβεια και 100% ανάκληση.

Συμπερασματικά βλέπουμε πως μεγαλύτερη αποτελεσματικότητα κατά την κατηγοριοποίηση μας έδωσε το μοντέλο της λογιστικής παλινδρόμησης (Simple Logistic).

# ΚΕΦΑΛΑΙΟ 4

## ΠΡΟΤΕΙΝΟΜΕΝΟ ΣΥΣΤΗΜΑ

---

### 4.1 ΣΚΟΠΟΣ – ΠΕΡΙΓΡΑΦΗ

Με την πάροδο των χρόνων όλο και περισσότερο αυξάνεται ο όγκος της πληροφορίας και των δεδομένων. Έτσι είναι αναγκαία συστήματα τα οποία να μπορούν να διαχειρίζονται την πληροφορία αυτή με σχετικά γρήγορο τρόπο με αποτέλεσμα να μειώνεται σημαντικά ο χρόνος και ο όγκος της εργασίας.

Τα νομικά κείμενα είναι κείμενα τα οποία διαθέτουν χρήσιμη πληροφορία και έχουν μεγάλο όγκο πολλές φορές, άρα κρίνεται απαραίτητο να υπάρχει κάποιο σύστημα για την διαχείρισή τους. Το συγκεκριμένο σύστημα που θα παρουσιαστεί παρακάτω είναι ένα τέτοιο σύστημα το οποίο περιλαμβάνει κάποιες βασικές λειτουργίες και εργασίες που θα μπορούσαν να επιτευχθούν πάνω σε τέτοιου είδους κείμενα. Οι κύριες λειτουργίες του συστήματος είναι η εξαγωγή περίληψης από κείμενα νομικής φύσης, η εξαγωγή και επισήμανση χρήσιμης πληροφορίας και η κατηγοριοποίηση με επιβλεπόμενη μάθηση των κειμένων ανάλογα με την κατηγορία στην οποία ανήκουν.

### 4.2 ΤΕΧΝΟΛΟΓΙΕΣ

Η εφαρμογή για την διαχείριση των νομικών κειμένων υλοποιήθηκε σε Java χρησιμοποιώντας τον μεταγλωττιστή NetBeans IDE. Πριν γίνει η περιγραφή της υλοποίησης είναι σημαντικό να αναφερθεί πως για την ολοκλήρωσή της εγκαταστάθηκαν βιβλιοθήκες από το εργαλείο μηχανικής μάθησης WEKA οι οποίες χρησιμοποιήθηκαν στην κατηγοριοποίηση, βιβλιοθήκες JFreeChart για την αυτόματη εξαγωγή γραφικών παραστάσεων καθώς και βιβλιοθήκες από τον Apache poi για την ανάγνωση επεξεργασία και εγγραφή σε κείμενα από το word office (\*.docx) και αρχεία excel (\*.xls).

#### 4.2.1 Πρόσθετα πακέτα/βιβλιοθήκες

Apache poi: Οι βιβλιοθήκες που εγκαταστάθηκαν βρίσκονται στο πακέτο poi-3.11 και είναι οι παρακάτω [20]:

- poi-3.11-20141221.jar
- poi-examples-3.11-20141221.jar
- poi-excelant-3.11-20141221.jar
- poi-ooxml-3.11-20141221.jar
- poi-ooxml-schemas-3.11-20141221.jar

- poi-scratchpad-3.11-20141221.jar
- xmlbeans-2.6.0.jar

JFreeChart: Οι βιβλιοθήκες που εγκαταστάθηκαν βρίσκονται στο πακέτο jfreechart-1.0.19 και είναι οι παρακάτω [21]:

- hamcrest-core-1.3.jar
- jcommon-1.0.23.jar
- jfreechart-1.0.19.jar
- jfreechart-1.0.19-experimental.jar
- jfreechart-1.0.19-swt.jar
- jfreesvg-2.0.jar
- junit-4.11.jar
- orsoncharts-1.4-eval-nofx.jar
- orsonpdf-1.6-eval.jar

WEKA: Για την χρήση των κλάσεων του weka έγινε η χρήση της βιβλιοθήκης weka.jar .

#### 4.2.2 Δομή και περιγραφή του κώδικα εφαρμογής

##### Η κλάση MainFrameForm

Το σύστημα αποτελείται από την κύρια κλάση MainFrameForm και δημιουργεί το κύριο παράθυρο της εφαρμογής το οποίο δίνει την δυνατότητα επιλογής εκτέλεσης μιας από τις τρεις λειτουργίες που προσφέρονται. Ουσιαστικά το κύριο σύστημα αποτελείτε από 3 υποσυστήματα. Η κλάσει αυτή αποτελείτε από τις παρακάτω μεθόδους:

- Η μέθοδος HeaderSummarisationActionPerformed: Όταν καλείτε αυτή η μέθοδος δίνεται στην μεταβλητή choice η τιμή 1.
- Η μέθοδος InformationExtractionActionPerformed: Όταν καλείτε αυτή η μέθοδος δίνεται στην μεταβλητή choice η τιμή 2.
- Η μέθοδος ClassifyActionPerformed: Όταν καλείτε αυτή η μέθοδος δίνεται στην μεταβλητή choice η τιμή 3.
- Η μέθοδος ComboBoxActionPerformed: Δέχεται μία τιμή String που επιλέγεται από τον χρήστη και η οποία καθορίζει ποιο μοντέλο θα χρησιμοποιηθεί για την κατηγοριοποίηση.
- Η μέθοδος NextButtonActionPerformed: Η μέθοδος αυτή καθορίζει το υποσύστημα το οποίο θα εκτελεστεί ανάλογα με την επιλογή που έχει κάνει ο χρήστης.
- Η μέθοδος ExitButtonActionPerformed: Τερματίζει την εφαρμογή.



### **Η κλάση HeaderSummarisationFrame**

Στην κλάση αυτή υλοποιείτε το γραφικό περιβάλλον του υποσυστήματος για την εξαγωγή της περίληψης. Περιλαμβάνει τις παρακάτω μεθόδους:

- **OpenFileButtonActionPerformed:** Καταχωρεί στην μεταβλητή path το μονοπάτι του αρχείου που επιλέγει ο χρήστης.
- **FinalHSButtonActionPerformed:** Αρχικά ανοίγει το αρχείο που επιλέγει ο χρήστης διαβάζοντας το μονοπάτι που λαμβάνει το σύστημα από την συνάρτηση **OpenFileButtonActionPerformed**. Στη συνέχεια δημιουργεί ένα αντικείμενο της κλάσης **Summarisation** και καλεί την μέθοδο **findHeader** η οποία είναι υπεύθυνη για την αναζήτηση και την δημιουργία της κεφαλίδας/περίληψης.
- **CloseButtonActionPerformed:** Τερματίζει το παράθυρο του υποσυστήματος.

### **Η κλάση Summarisation**

Η κλάση αυτή περιλαμβάνει τη μέθοδο **findHeader** ως κύρια συνάρτηση που περιλαμβάνει τον αλγόριθμο για την εξαγωγή περίληψης και καλεί τις παρακάτω μεθόδους.

- **FindNumber:** Για τον εντοπισμό του αριθμού του νομικού κειμένου.
- **FindJudge:** Για τον εντοπισμό του Προεδρέυων Αντιπρόεδρου.
- **FindProponent:** Για τον εντοπισμό του εισηγητή.
- **FindAdvocate:** Για τον εντοπισμό του Εισαγγελεύς.
- **FindLawyers:** Εντοπίζει τους δικηγόρους σε περίπτωση που υπάρχουν δυο.
- **FindLawyer:** Για τον εντοπισμό του δικηγόρου.

### **Η κλάση SecondStepMenu**

Η κλάση αυτή υλοποιεί το γραφικό περιβάλλον του υποσυστήματος για την επισήμανση των αναφορών. Αποτελείτε από μια μέθοδο για κάθε κατηγορία αναφορών. Αν γίνει επιλογή μιας ή περισσότερων κατηγοριών καταχωρείτε σε μια μεταβλητή ένα σύμβολο το οποίο αντιστοιχεί σε μια συγκεκριμένη κατηγορία, έτσι το σύστημα γνωρίζει τι επέλεξε ο χρήστης για εντοπισμό. Επίσης περιλαμβάνει τις παρακάτω μεθόδους:

- **OpenFileButtonActionPerformed:** Καταχωρεί στην μεταβλητή path το μονοπάτι του αρχείου που επιλέγει ο χρήστης.
- **NextButtonActionPerformed:** Η μέθοδος αυτή αρχικά ανοίγει το αρχείο που επιλέγει ο χρήστης και δημιουργεί ένα String με τα περιεχόμενα του. Στη συνέχεια δημιουργεί ένα αντικείμενο της κλάσης **Classification** όπου γίνεται ο εντοπισμός των αναφορών. Αφού εντοπίσει τις αναφορές γράφει σε νέο αρχείο

το κείμενο με τις αναφορές που εντόπισε και επισήμανε και ανοίγει νέο παράθυρο για την προβολή των αποτελεσμάτων.

- CloseButtonActionPerformed: Τερματίζει το παράθυρο του υποσυστήματος.

### **Η κλάση Classification**

Στην κλάση αυτή γίνεται ο εντοπισμός, η επισήμανση των αναφορών και άλλες λειτουργίες που έχουν σχέση με την διαχείρισή τους.. Πιο αναλυτικά περιλαμβάνει:

- Την μέθοδο Allreferences: Η μέθοδος αυτή περιλαμβάνει τον αλγόριθμο εντοπισμού των αναφορών. Εντοπίζει τις αναφορές και καλεί μεθόδους για την μορφοποίηση που θα λάβουν στο νέο κείμενο που θα εγγραφούν. Επίσης μετράει την συχνότητα εμφάνισης της κάθε κατηγορίας και δημιουργεί ένα διάγραμμα για κάθε νομική απόφαση που υπάρχει μέσα στο κείμενο. Έτσι δημιουργεί και εμφανίζει μια γραφική παράσταση που περιγράφει το επί τις εκατό ποσοστό εμφάνισης τις κάθε κατηγορίας.
- Την μέθοδο CreateTextVector: Η μέθοδος αυτή δημιουργεί έναν δυναμικό πίνακα που κάθε στοιχείο του περιέχει το όνομα μιας κατηγορίας αναφορών.
- Την μέθοδο CreateRateList: Η μέθοδος αυτή δημιουργεί έναν δυναμικό πίνακα που κάθε στοιχείο του περιέχει τον αριθμό που εμφανίστηκε μια κατηγορία στο κείμενο.
- Την μέθοδο FormatingType1: Η μέθοδος αυτή δέχεται ως παραμέτρους ένα String που περιέχει την αναφορά που εντοπίστηκε και ένα String που περιέχει το έναν κωδικό που αντιστοιχεί στο χρώμα που θα πάρει η συγκεκριμένη αναφορά. Έτσι προσθέτει στο νέο κείμενο την αναφορά χρωματισμένη και υπογραμμισμένη.
- Την μέθοδο FormatingType2: Η μέθοδος αυτή κάνει την ίδια διαδικασία με την μέθοδο FormatingType1.
- Την μέθοδο Find\_IKOD\_Ref: Η μέθοδος αυτή καλείτε όταν εντοπίζεται στο κείμενο αναφορά τύπου «αρ.5 παρ 10 του Ιδίου Κώδικα». Αυτό σημαίνει πως η κατηγορία της αναφοράς αυτής είναι ίδια με την αμέσως προηγούμενη κατηγορία που υπάρχει στο κείμενο και είναι μια από τις ΚΠολΔ,ΑΚ,ΠΚ,ΚΠΔ. Η δουλειά της μεθόδου αυτής είναι να βρει σε ποια από τις κατηγορίες αυτές ανήκει.
- Την μέθοδο Find\_INOMOY\_Ref: Η μέθοδος αυτή καλείτε όταν εντοπίζεται στο κείμενο αναφορά τύπου «αρ.5 παρ 10 του Ιδιού Νόμου». Αυτό σημαίνει πως η κατηγορία της αναφοράς αυτής είναι ίδια με την αμέσως προηγούμενη κατηγορία που υπάρχει στο κείμενο και είναι μια από τις Ν. ή ν.δ. . Η δουλειά της μεθόδου αυτής είναι να βρει σε ποια από τις κατηγορίες αυτές ανήκει.

### **Η κλάση FinalFrame**

Στην κλάση αυτή υλοποιείτε το γραφικό παράθυρο που παρουσιάζει τα τελικά αποτελέσματα του υποσυστήματος επισημάνσης αναφορών. Πιο αναλυτικά περιλαμβάνεται:

- Η μέθοδος `NewTextButtonActionPerformed`: Η μέθοδος αυτή εμφανίζει το `.docx` αρχείο που έχει δημιουργηθεί, στο οποίο έχουν επισημανθεί οι αναφορές που ζήτησε ο χρήστης.
- Η μέθοδος `VectorsTableButtonActionPerformed`: Η μέθοδος αυτή εμφανίζει ένα `.xls` αρχείο το οποίο περιέχει διανύσματα, τόσα όσες και οι αποφάσεις που υπήρχαν μέσα στο κείμενο.
- Η μέθοδος `CloseButtonActionPerformed`: Τερματίζει το παράθυρο του υποσυστήματος.

### **Η κλάση CreateVectorExcel**

Στην κλάση αυτή δημιουργείτε ένα νέο αρχείο `.xls` και καταχωρούνται τα διανύσματα των αποφάσεων από τα νομικά κείμενα. Κάθε διάνυσμα αντιστοιχεί σε μια απόφαση και κάθε στοιχείο του διανύσματος σε μια κατηγορία αναφορών. Περιλαμβάνει τις παρακάτω μεθόδους:

- `Excel`: Στην μέθοδο αυτή γίνεται η εγγραφή της πληροφορίας στο αρχείο.
- `OpenExcel`: Η μέθοδος αυτή δημιουργεί ένα αρχείο `.xls` για έγγραφη.
- `CloseExcel`: Η μέθοδος αυτή κλείνει το αρχείο.
- `openFinalExcel`: Εκτελώντας την μέθοδο αυτή ανοίγει αυτόματα το αρχείο `excel` που είχε δημιουργηθεί.

### **Η κλάση JFreeChartBar**

Στην κλάση αυτή δημιουργείται η γραφική παράσταση που απεικονίζει το ποσοστό εμφάνισης των αναφορών που εντοπίστηκαν μέσα στο κείμενο. Περιλαμβάνει τις εξής μεθόδους:

- `createDataset`: Αυτή η μέθοδος καταχωρεί τα ποσοστά στην γραφική παράσταση στην οποία απεικονίζονται μόνο οι κατηγορίες που σημειώνουν ποσοστό μεγαλύτερο του μηδενός.
- `Rates`: Η μέθοδος αυτή λαμβάνει ως παραμέτρους δυο τιμές, κάνει τον υπολογισμό του ποσοστού και επιστρέφει την τιμή του.

### **Η κλάση AllFinalsVariables**

Η κλάση αυτή περιλαμβάνει όλες τις σταθερές που ήταν απαραίτητες για την δημιουργία του συστήματος.

### **Η κλάση ReadAndWriteDocx**

Η κλάση αυτή υλοποιεί τις βασικές λειτουργίες για ανάγνωση, εγγραφή και επεξεργασία σε ένα αρχείο .docx και περιλαμβάνει τις μεθόδους:

- ReadDocx: Η μέθοδος αυτή διαβάζει από ένα αρχείο .docx. Το αρχείο αυτό επιλέγεται από τον χρήστη.
- WriteDocx: Η μέθοδος αυτή δημιουργεί ένα αντικείμενο της κλάσης XWPFDocument και το επιστρέφει.
- OpenNewDocx: Η μέθοδος αυτή δημιουργεί ένα νέο αρχείο .docx
- openFinalNewDocx: Η μέθοδος αυτή εμφανίζει αυτόματα στην οθόνη το νέο αρχείο .docx που έχει δημιουργηθεί.

### **Η κλάση PerceptronFrame**

Η κλάση αυτή δημιουργεί το γραφικό περιβάλλον για την εφαρμογή του μοντέλου MultiLayer Perceptron στο υποσύστημα κατηγοριοποίησης κειμένων. Προσφέρει λειτουργίες για την δημιουργία και την τροποποίηση του μοντέλου, για την εκπαίδευση του και για την εκτέλεση της κατηγοριοποίησης. Ποιο αναλυτικά περιλαμβάνει τις παρακάτω μεθόδους:

- TrainingButtonActionPerformed: Στην μέθοδο αυτή γίνεται η δημιουργία και η εκπαίδευση του μοντέλου. Αρχικά δημιουργείτε μια συμβολοσειρά που καθορίζει την δημιουργία του και στη συνέχεια ανοίγει ένα αρχείο .arff για ανάγνωση το οποίο περιέχει πρότυπα από νομικά κείμενα όλων των κατηγοριών. Τέλος δημιουργεί ένα παράθυρο διαλόγου στο οποίο εμφανίζονται τα αποτελέσματα της εκπαίδευσης.
- ClassificationButtonActionPerformed: Η μέθοδος αυτή ανοίγει το αρχείο που επέλεξε ο χρήστης (path) για ανάγνωση και στη συνέχεια πραγματοποιεί ταξινόμηση των προτύπων με το μοντέλο MLP. Τέλος δημιουργεί ένα παράθυρο διαλόγου στο οποίο εμφανίζονται τα αποτελέσματα της ταξινόμησης.
- OpenARFFfileActionPerformed: Η μέθοδος αυτή καταχωρεί στην μεταβλητή path το μονοπάτι του .arff αρχείου που επιλέγει ο χρήστης.
- ClosePerceptronFrameActionPerformed: Η μέθοδος αυτή τερματίζει την λειτουργία του υποσυστήματος.
- CloseDialogButtonActionPerformed: Η μέθοδος αυτή κλείνει το παράθυρο διαλόγου.

### **Η κλάση RBFNetworkFrame**

Η κλάση αυτή δημιουργεί το γραφικό περιβάλλον για την εφαρμογή του μοντέλου RBFNetwork στο υποσύστημα κατηγοριοποίησης κειμένων. Προσφέρει λειτουργίες για την δημιουργία και την τροποποίηση του μοντέλου, για την εκπαίδευση του και για την εκτέλεση της κατηγοριοποίησης. Ποιο αναλυτικά περιλαμβάνει τις παρακάτω μεθόδους:

- **TrainingButtonActionPerformed:** Στην μέθοδο αυτή γίνεται η δημιουργία και η εκπαίδευση του μοντέλου. Αρχικά δημιουργείτε μια συμβολοσειρά που καθορίζει την δημιουργία του και στη συνέχεια ανοίγει ένα αρχείο .arff για ανάγνωση το οποίο περιέχει πρότυπα από νομικά κείμενα όλων των κατηγοριών. Τέλος δημιουργεί ένα παράθυρο διαλόγου στο οποίο εμφανίζονται τα αποτελέσματα της εκπαίδευσης.
- **ClassificationButtonActionPerformed:** Η μέθοδος αυτή ανοίγει το αρχείο που επέλεξε ο χρήστης (path) για ανάγνωση και στη συνέχεια πραγματοποιεί ταξινόμηση των προτύπων με το μοντέλο RBFNetwork. Τέλος δημιουργεί ένα παράθυρο διαλόγου στο οποίο εμφανίζονται τα αποτελέσματα της ταξινόμησης.
- **OpenARFFfileActionPerformed:** Η μέθοδος αυτή καταχωρεί στην μεταβλητή path το μονοπάτι του .arff αρχείου που επιλέγει ο χρήστης.
- **ClosePerceptronFrameActionPerformed:** Η μέθοδος αυτή τερματίζει την λειτουργία του υποσυστήματος.
- **CloseDialogButtonActionPerformed:** Η μέθοδος αυτή κλείνει το παράθυρο διαλόγου.

### **Η κλάση SMOFrame**

Η κλάση αυτή δημιουργεί το γραφικό περιβάλλον για την εφαρμογή του μοντέλου SMO στο υποσύστημα κατηγοριοποίησης κειμένων. Προσφέρει λειτουργίες για την δημιουργία και την τροποποίηση του μοντέλου, για την εκπαίδευση του και για την εκτέλεση της κατηγοριοποίησης. Ποιο αναλυτικά περιλαμβάνει τις παρακάτω μεθόδους:

- **TrainingButtonActionPerformed:** Στην μέθοδο αυτή γίνεται η δημιουργία και η εκπαίδευση του μοντέλου. Αρχικά δημιουργείτε μια συμβολοσειρά που καθορίζει την δημιουργία του και στη συνέχεια ανοίγει ένα αρχείο .arff για ανάγνωση το οποίο περιέχει πρότυπα από νομικά κείμενα όλων των κατηγοριών. Τέλος δημιουργεί ένα παράθυρο διαλόγου στο οποίο εμφανίζονται τα αποτελέσματα της εκπαίδευσης.
- **ClassificationButtonActionPerformed:** Η μέθοδος αυτή ανοίγει το αρχείο που επέλεξε ο χρήστης (path) για ανάγνωση και στη συνέχεια πραγματοποιεί ταξινόμηση των προτύπων με το μοντέλο SMO. Τέλος δημιουργεί ένα παράθυρο διαλόγου στο οποίο εμφανίζονται τα αποτελέσματα της ταξινόμησης.

- `OpenARFFfilrActionPerformed`: Η μέθοδος αυτή καταχωρεί στην μεταβλητή `path` το μονοπάτι του `.arff` αρχείου που επιλέγει ο χρήστης.
- `ClosePerceptronFrameActionPerformed`: Η μέθοδος αυτή τερματίζει την λειτουργία του υποσυστήματος.
- `CloseDialogButtonActionPerformed`: Η μέθοδος αυτή κλείνει το παράθυρο διαλόγου.

### **Η κλάση SimpleLogisticFrame**

Η κλάση αυτή δημιουργεί το γραφικό περιβάλλον για την εφαρμογή του μοντέλου Simple Logistic στο υποσύστημα κατηγοριοποίησης κειμένων. Προσφέρει λειτουργίες για την δημιουργία και την τροποποίηση του μοντέλου, για την εκπαίδευση του και για την εκτέλεση της κατηγοριοποίησης. Ποιο αναλυτικά περιλαμβάνει τις παρακάτω μεθόδους:

- `TrainingButtonActionPerformed`: Στην μέθοδο αυτή γίνεται η δημιουργία και η εκπαίδευση του μοντέλου. Αρχικά δημιουργείτε μια συμβολοσειρά που καθορίζει την δημιουργία του και στη συνέχεια ανοίγει ένα αρχείο `.arff` για ανάγνωση το οποίο περιέχει πρότυπα από νομικά κείμενα όλων των κατηγοριών. Τέλος δημιουργεί ένα παράθυρο διαλόγου στο οποίο εμφανίζονται τα αποτελέσματα της εκπαίδευσης.
- `ClassificationButtonActionPerformed`: Η μέθοδος αυτή ανοίγει το αρχείο που επέλεξε ο χρήστης (`path`) για ανάγνωση και στη συνέχεια πραγματοποιεί ταξινόμηση των προτύπων με το μοντέλο Simple Logistic. Τέλος δημιουργεί ένα παράθυρο διαλόγου στο οποίο εμφανίζονται τα αποτελέσματα της ταξινόμησης.
- `OpenARFFfilrActionPerformed`: Η μέθοδος αυτή καταχωρεί στην μεταβλητή `path` το μονοπάτι του `.arff` αρχείου που επιλέγει ο χρήστης.
- `ClosePerceptronFrameActionPerformed`: Η μέθοδος αυτή τερματίζει την λειτουργία του υποσυστήματος.
- `CloseDialogButtonActionPerformed`: Η μέθοδος αυτή κλείνει το παράθυρο διαλόγου.

## 4.3 ΠΕΡΙΓΡΑΦΗ ΣΥΣΤΗΜΑΤΟΣ ΔΙΑΧΕΙΡΗΣΗΣ ΝΟΜΙΚΩΝ ΚΕΙΜΕΝΩΝ

Σε αυτή την ενότητα θα παρουσιαστεί η εφαρμογή που δημιουργήθηκε στα πλαίσια της παρούσας εργασίας. Στην Εικόνα 5 απεικονίζεται το κύριο παράθυρο της εφαρμογής όπου δίνει την δυνατότητα στον χρήστη να επιλέξει την λειτουργία που επιθυμεί να εκτελέσει. Διαθέτει επιλογή για εξαγωγή κεφαλίδας/περίληψης (Header Summarisation), επιλογή για επισήμανση αναφορών σε νομικά κείμενα (Hightlight References) και επιλογή για κατηγοριοποίηση νομικών κειμένων (Classify) δίνοντας παράλληλα την δυνατότητα πλοκής αλγορίθμου. Οι αλγόριθμοι που θα κληθεί να επιλέξει ο χρήστης είναι οι εξής:

- MultiLayerPerceptron
- RBFNetwork
- SMO
- Simple Logistic

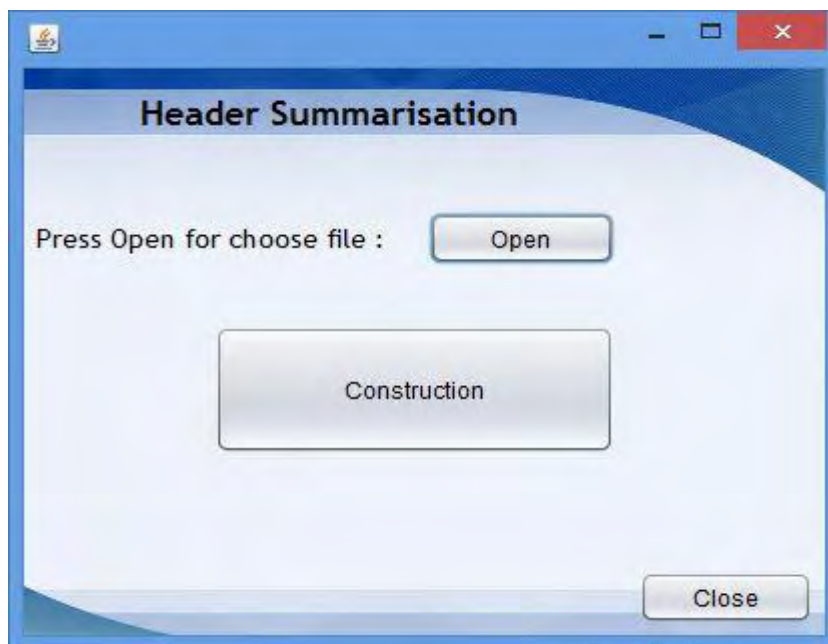
Τέλος επιλέγοντας το κουμπί Next το σύστημα μεταβαίνει στην επόμενη κατάσταση η οποία είναι ανάλογη με την επιλογή που έχει γίνει πιο πριν και επίσης διαθέτει κουμπί Exit για έξοδο και τερματισμό των διεργασιών της εφαρμογής.



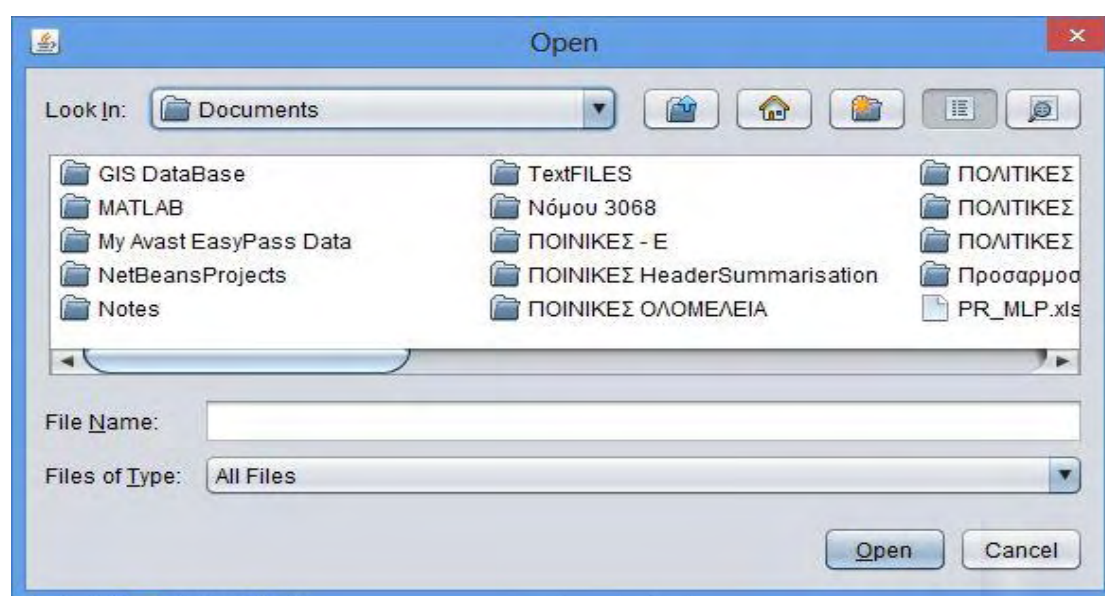
Εικόνα 5 Αρχικό παράθυρο εφαρμογής διαχείρισης νομικών κειμένων

### 4.3.1 Εξαγωγή περίληψης

Επιλέγοντας στο κύριο παράθυρο της εφαρμογής την επιλογή Header Summarisation το σύστημα μεταβαίνει σε ένα άλλο παράθυρο το οποίο φαίνεται στην Εικόνα 6. Αυτό διαθέτει ένα κουμπί Open το οποίο εμφανίζει ένα παράθυρο (Εικόνα 7) που προσφέρεται για την επιλογή ενός αρχείου .docx με νομικό περιεχόμενο. Εν συνεχεία πιέζοντας το κουμπί Construction ανοίγει αυτόματα ένα έγγραφο .docx (Εικόνα 8) το οποίο περιέχει την περίληψη/επικεφαλίδα από κάθε νομική απόφαση που περιείχε το αρχικό κείμενο.



Εικόνα 6 Header Summarisation Frame



Εικόνα 7 Παράθυρο επιλογής αρχείου



**Αριθμ. 1713/2011**

**Προεδρεύων ο Αντιπρόεδρος Γ. Χρυσικός  
Εισηγητής ο Αρεοπαγίτης Κ. Φράγκος  
Εισαγγελεύς Γ. Χατζίκος  
Δικηγόρος Ν. Μαυρομάτης**

**Αριθμ. 1723/2001**

**Προεδρεύων ο Αντιπρόεδρος Ν. Ζαΐρης  
Εισηγητής ο Αρεοπαγίτης Α. Ξένος  
Εισαγγελεύς Μ. Ανδρειωτέλλης**

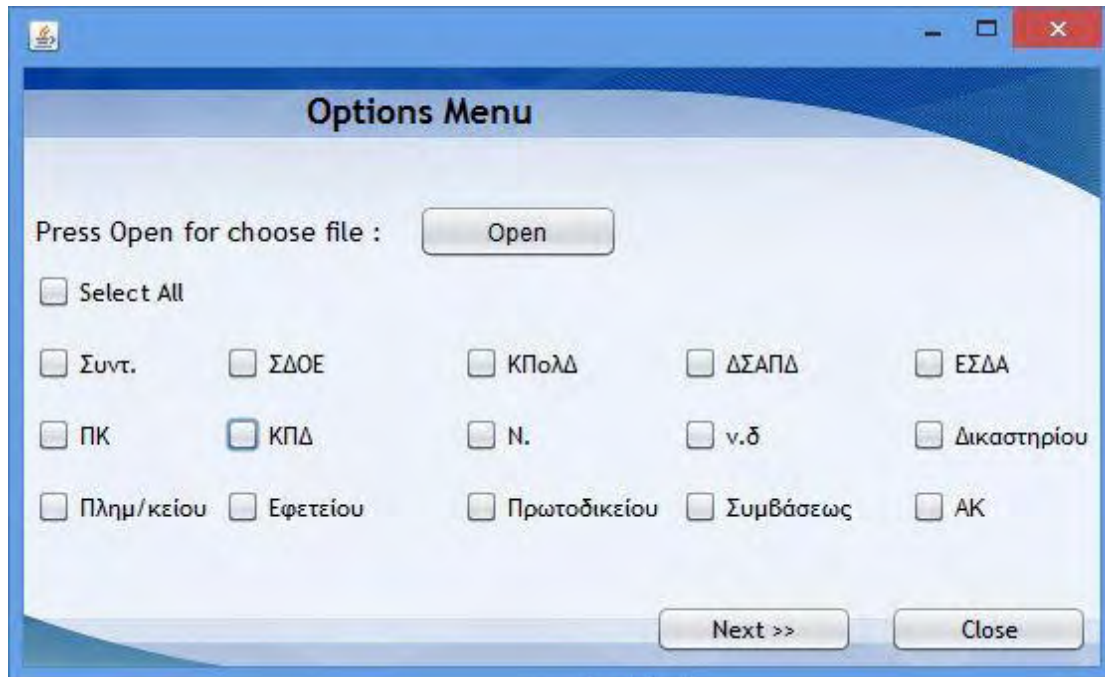
**Αριθμ. 1725/2011**

**Προεδρεύων ο Αντιπρόεδρος Γ. Χρυσικός  
Εισηγήτρια η Αρεοπαγίτης Ειρήνη Κιουρκτσόγλου-Πετρουλάκη  
Εισαγγελεύς Γ. Κολιοκώστας  
Δικηγόρος Ελένη Γκόνου**

Εικόνα 8 Απόσπασμα αποτελεσμάτων Εξαγωγής Περίληψης

#### 4.3.2 Επισήμανση αναφορών

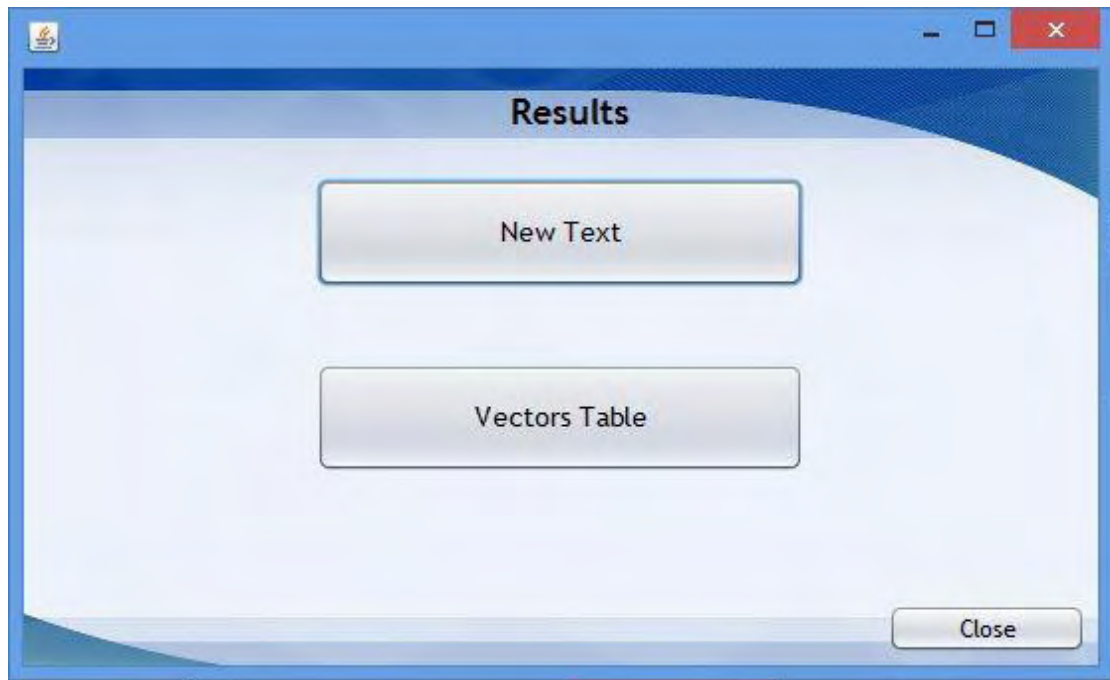
Για την επισήμανση των αναφορών προβάλλεται ένα παράθυρο που δίνει την δυνατότητα επιλογής του αρχείου που θέλει ο χρήστης να επεξεργαστεί πατώντας το κουμπί Open. Το παράθυρο που θα εμφανιστεί για την επιλογή φαίνεται στην Εικόνα 7. Ακόμα δίνεται ένα μενού για την επιλογή των κατηγοριών που επιθυμεί να επισημάνει ο χρήστης. Η επιλογή αυτή μπορεί να είναι μία κατηγορία ή περισσότερες. Τέλος προς διευκόλυνση υπάρχει η επιλογή Select All που επιλέγει αυτόματα όλες τις κατηγορίες.



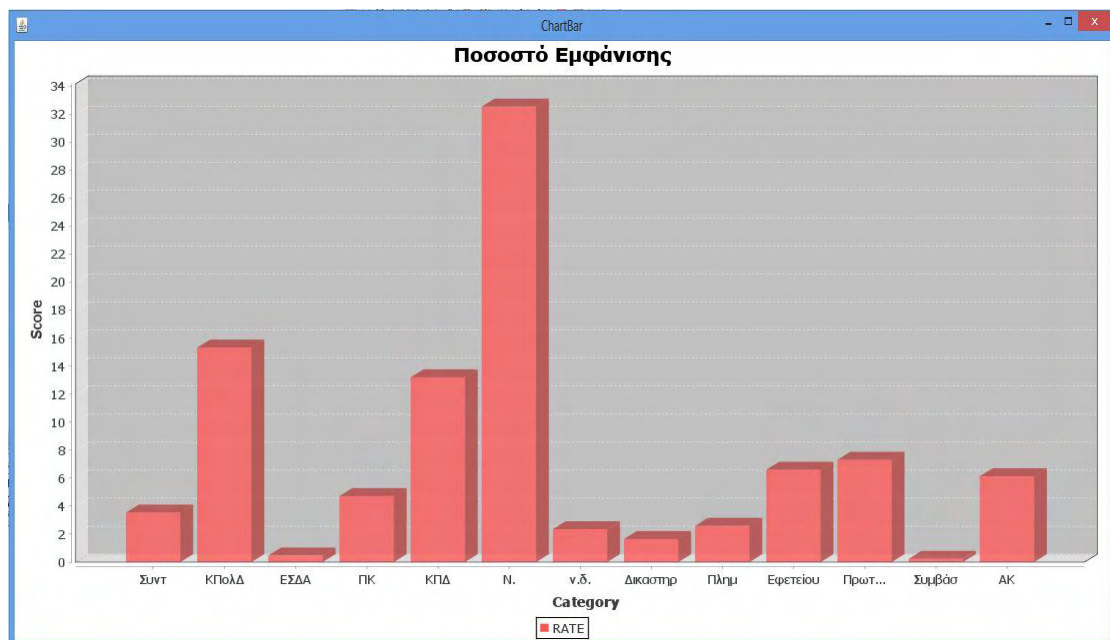
Εικόνα 9 Μενού επιλογών για επισήμανση αναφορών

Μετά την επιλογή του κατάλληλου αρχείου και την επιλογή των κατηγοριών προς επισήμανση, πατώντας το κουμπί Next οδηγούμαστε προς τα αποτελέσματα τα οποία εμφανίζονται σε δύο παράθυρα γραφικών (Εικόνα 10, Εικόνα 11).

Στην Εικόνα 11 παρουσιάζεται το επί της εκατό ποσοστό εμφάνισης της κάθε κατηγορίας στο αρχείο που επεξεργάστηκε δίνοντας έτσι σημαντικές πληροφορίες για το περιεχόμενο του κειμένου. Στο παράθυρο της Εικόνας 10 παρουσιάζονται αποτελέσματα τα οποία διαιρούνται σε 2 κατηγορίες. Αρχικά πιέζοντας το κουμπί New Text δημιουργείτε και ανοίγει αυτόματα ένα νέο .docx αρχείο το οποίο περιέχει το αρχικό νομικό κείμενο αλλά με επισημασμένες τις αναφορές που έχει εντοπίσει. Κάθε κατηγορία αναφορών υιοθετεί ένα συγκεκριμένο χρώμα έτσι ώστε να είναι ξεκάθαρος ο μεταξύ τους διαχωρισμός (Εικόνα 12). Πιέζοντας το δεύτερο κουμπί που δίνεται και ονομάζεται Vectors Table εμφανίζεται αυτόματα ένα excel αρχείο το οποίο περιέχει διανύσματα που χαρακτηρίζουν το συγκεκριμένο νομικό κείμενο. Ο αριθμός των διανυσμάτων που δημιουργούνται είναι ανάλογος με τον αριθμό των νομικών αποφάσεων που περιέχει το κείμενο. Κάθε διάνυσμα λοιπόν αντιστοιχεί σε μία νομική απόφαση (Εικόνα 13) και κάθε στοιχείο του διανύσματος αντιστοιχεί στην συχνότητα εμφάνισης μιας κατηγορίας .



Εικόνα 10 Παράθυρο αποτελεσμάτων επισήμανσης αναφορών



Εικόνα 11 Ποσοστό(%) εμφάνισης αναφορών ανά κατηγορία

προσβαλλόμενη 27858/2012 απόφαση του Τριμελούς Πλημ/κείου, που εκδόθηκε κατ' έφεση, ο αναιρεσείων καταδικάστηκε και πάλι για τις ίδιες ως άνω δύο μερικότερες πράξεις της μη απόδοσης των ίδιων ως άνω ποσών Φ.Π.Α. στο Δημόσιο κατά τις ίδιες χρονικές περιόδους, αλλά υπό τον νομικό χαρακτηρισμό της μη αποδόσεως Φ.Π.Α. προς το Δημόσιο, κατ' εξακολούθηση (άρ. 98 του ΠΚ και 18 του Ν. 2523/1997), σε φυλάκιση ενός (1) έτους, η οποία μετατράπηκε προς πέντε (5) ευρώ ημερησίως. Τοιουτοτρόπως, το Τριμελές Αθηνών, καταδικάζοντας τον αναιρεσείοντα με την προσβαλλόμενη 27858/2012 απόφασή του, για πράξη για την οποία προηγουμένως είχε καταδικαστεί αμετακλήτως με την 23722/2011 απόφαση, παραβίασε παραβίασε τις περί δεδικασμένου διατάξεις του άρ. 57 του ΚΠΑ και αναιρετέα κατέστησε την απόφασή του, κατά τον από την διάταξη του άρ. 510 παρ. 1 στοιχ. ΣΤ# του ΚΠΑ λόγο αναιρέσεως, ο οποίος ερευνάται και αυτεπαγγέλτως από το Δικαστήριο του Αρείου Πάγου, κατ' εφαρμογή του άρ. 511 του ΚΠΑ. Επομένως, πρέπει να αναιρεθεί η προσβαλλόμενη απόφαση και να κηρυχθεί απαράδεκτη η ποινική δίωξη κατά του αναιρεσείοντος για την πράξη για την οποία καταδικάστηκε (άρ. 57 του ΚΠΑ). ΔΕΔΙΚΑΣΜΕΝΟ/ΦΟΡΟΛΟΓΙΚΕΣ ΠΑΡΑΒΑΣΕΙΣ [+++ Προσοχή! Θωμά, δεν είμαι σίγουρη αν κατάλαβα αυτή την απόφαση...--] Αριθμ. 363/2013 Προεδρεύων ο Αντιπρόεδρος Ν. Εισηγητής ο Αρεοπαγίτης Κ. Φράγκος, Εισαγγελεύς Μ. Ανδρειωτέλλης, Δικηγόρος Θ. Μπαλίδης Δεδικασμένο επί φορολογικών παραβάσεων (άρ. 57, 510 παρ. 1 στοιχ. ΣΤ# ΚΠΑ· άρ. 25 του Ν. 1882/1990· άρ. 19 του Ν. 2523/1997): Απορρίπτεται ο περί παραβιάσεως του δεδικασμένου λόγος αναιρέσεως του κατηγορουμένου, διότι η αμετάκλητη αθώωσή του για αποδοχή εικονικών τιμολογίων δεν ασκεί έννομη επιρροή επί της καταδίκης για μη καταβολή χρεών προς το Δημόσιο, επειδή η τελευταία πράξη προέκυψε από βεβαιωθέν μεταγενέστερα (στην αρμόδια ΔΟΥ) πρόστιμο για την φορολογική παράβαση της αποδοχής εικονικών Στην προκειμένη περίπτωση, ο αναιρεσείων κατηγορούμενος με την κρινόμενη αίτηση αναιρέσεως, προσβάλλει την 15419/2011 απόφαση του Τριμελούς Πλημμελειοδικείου Θεσσαλονίκης, που τον καταδίκασε σε δεύτερο βαθμό, για την πράξη της μη καταβολή χρεών προς το Δημόσιο, σε φυλάκιση τριών ετών με τριετή αναστολή, για τους λόγους: [...] β) παραβίασης δεδικασμένου αποφάσεων διοικητικών δικαστηρίων επί του ίδιου βεβαιωθέντος χρέους του και της με αρ. 2465/2001 αποφάσεως του Τριμελούς Εφετείου Θεσσαλονίκης, με την οποία αθώωθηκε αμετάκλητα, για την πράξη της αποδοχής εικονικών τιμολογίων (άρ. 510 παρ. 1 στοιχ. ΣΤ του ΚΠΑ) [...] [...] Με τις παραπάνω παραδοχές, το δικάσαν σε δεύτερο βαθμό

Εικόνα 12 Επισημάνση αναφορών

0	0	15	0	0	0	0	20	2	1	0	1	0	0	7
0	0	2	0	0	0	0	0	0	1	0	1	1	0	0
0	0	15	0	0	0	0	1	0	1	0	4	0	0	0
0	0	8	0	0	0	0	0	0	1	0	1	0	0	0
0	0	7	0	0	0	0	2	0	1	0	2	3	0	6
0	0	12	0	0	0	0	12	0	1	0	2	0	0	11
3	0	6	0	0	0	0	9	0	1	0	0	3	0	1
0	0	5	0	0	0	0	1	0	1	0	2	0	0	20
2	0	7	0	0	1	0	6	3	0	0	0	3	0	4
0	0	3	0	0	0	0	1	0	1	0	2	0	0	14
9	0	2	0	3	1	0	28	0	1	0	1	0	0	7
0	0	2	0	0	0	0	0	0	1	0	1	1	0	5
0	0	5	0	0	0	0	7	0	1	0	1	3	0	1
6	0	9	0	0	0	0	37	0	2	0	0	1	0	0
0	0	3	0	0	0	0	0	0	0	0	1	0	0	0
0	0	5	0	0	0	0	0	0	1	0	4	3	0	0
2	0	21	0	0	0	0	2	0	0	0	6	2	0	13
0	0	19	0	0	0	0	2	0	1	0	3	2	0	0
0	0	10	0	0	0	0	0	0	1	0	1	2	0	4
0	0	11	0	0	0	0	0	0	1	0	3	1	0	9
1	0	12	0	0	0	0	5	0	1	0	3	3	2	0
0	0	15	0	0	0	0	2	0	5	0	4	19	0	4
0	0	17	0	0	0	0	2	0	2	0	3	2	0	6
0	0	12	0	0	0	0	0	0	1	0	6	1	0	1
2	0	9	0	0	0	0	4	0	0	0	2	0	0	6
5	0	6	0	1	0	0	6	2	5	0	3	1	0	1
4	0	5	0	1	0	0	1	1	5	0	3	2	0	1
0	0	18	0	0	0	0	1	0	2	0	10	10	0	0
0	0	4	0	0	0	0	0	0	2	0	4	1	0	1
6	0	12	0	0	0	0	13	0	1	0	0	3	0	3
0	0	9	0	0	0	0	10	0	1	0	1	2	0	1
5	0	2	0	0	0	0	8	0	1	0	1	0	0	2

Εικόνα 13 Δημιουργία Διανυσμάτων

## 4.4 ΠΕΡΙΓΡΑΦΗ ΣΥΣΤΗΜΑΤΟΣ ΚΑΤΗΓΟΡΙΟΠΟΙΗΣΗΣ ΚΕΙΜΕΝΩΝ

### 4.4.1 Κατηγοριοποίηση με το μοντέλο Multilayer Perceptron

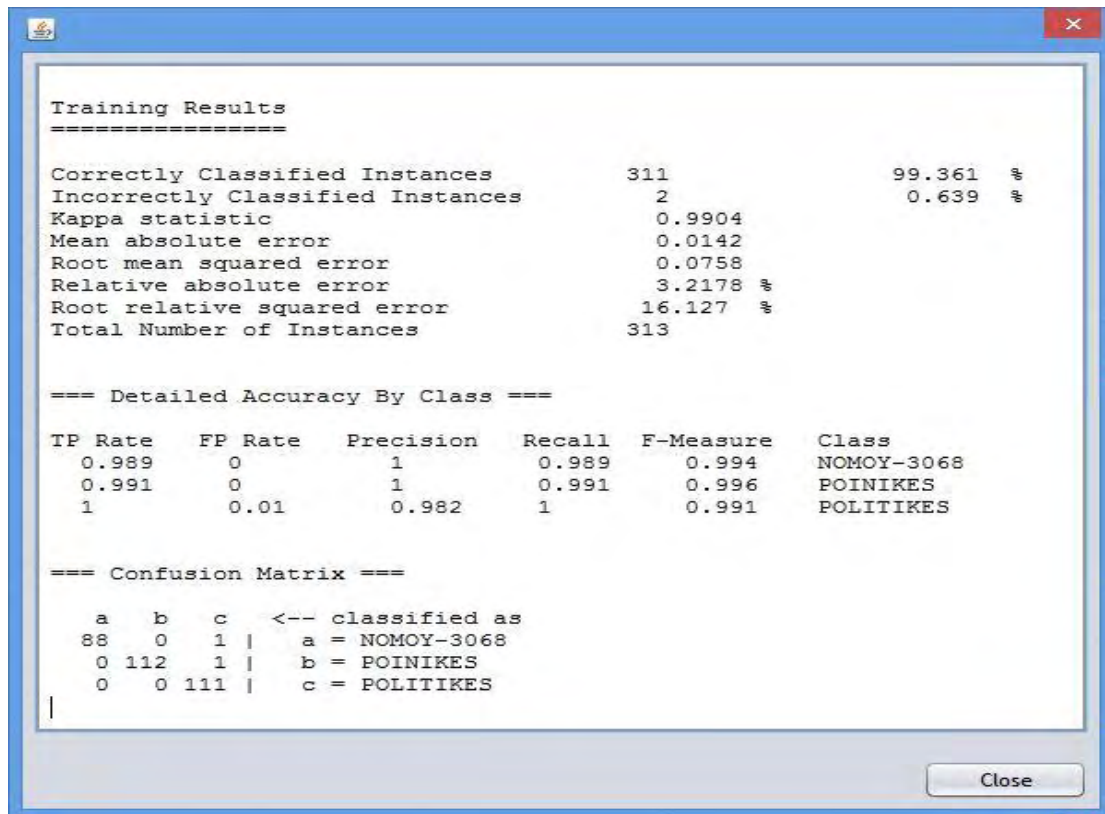
Στην Εικόνα 14 φαίνεται το γραφικό περιβάλλον που εμφανίζεται με την επιλογή του Classify και MLP στην αρχή της εφαρμογής. Σε αυτό το σημείο δίνεται αρχικά η δυνατότητα στο χρήστη να δημιουργήσει το μοντέλο με τον τρόπο που αυτός επιθυμεί ρυθμίζοντας μια σειρά παραμέτρων. Οι παράμετροι αυτοί είναι:

- Κρυφά επίπεδα (Hidden Layers) : Είναι ο αριθμός των κρυφών στρωμάτων του μοντέλου και έχει δοθεί ως τιμή προεπιλογής ο χαρακτήρας 'a' ο οποίος δηλώνει τον αυτόματο προσδιορισμό των κόμβων για κάθε κρυφό επίπεδο. Η τιμή πρέπει να είναι μια λίστα από αριθμούς διαχωρισμένους με κόμμα, με κάθε αριθμό να αντιστοιχεί στον αριθμό των κόμβων για κάθε επίπεδο.
- Ρυθμός μάθησης (Learning Rate) : Εκφράζει τον ρυθμό μάθησης για τον αλγόριθμο backpropagation και μπορεί να πάρει τιμές από 0 έως 1, με τιμή προεπιλογής το 0.3.
- Ορμή (Momentum) : Εκφράζει τον ρυθμό της ορμής για τον αλγόριθμο backpropagation και μπορεί να λάβει τιμές από 0 έως 1 με τιμή προεπιλογής το 0.2.
- Χρόνος εκπαίδευσης (Training Time) : Εκφράζει τον αριθμό των εποχών για την εκπαίδευση του δικτύου. Εποχή (epoch) ονομάζεται ένας κύκλος εκπαίδευσης. Η προεπιλεγμένη τιμή του είναι συνήθως το 500.
- Validation Set Size: Ορίζεται ως προεπιλογή 0 και εκφράζει το μέγεθος του συνόλου επικύρωσης.
- Validation Threshold: Εκφράζει τον αριθμό των διαδοχικών λαθών που επιτρέπονται για τις δοκιμές επικύρωσης πριν από τον τερματισμό του δικτύου. Η τιμή προεπιλογής του είναι το 20.
- Seed : Η τιμή του πρέπει να είναι πάντα μεγαλύτερη ή ίση του μηδενός. Έχει οριστεί ως προεπιλογή η τιμή 0.

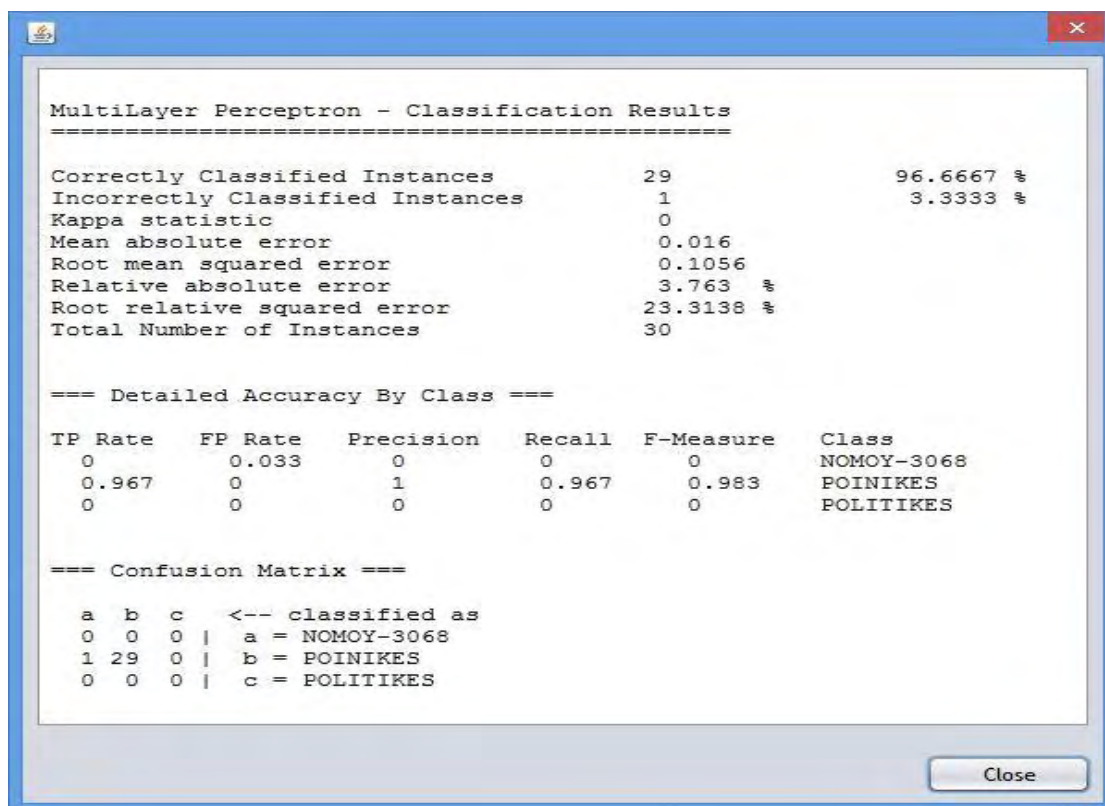
Για την εκπαίδευση του μοντέλου υπάρχει το κουμπί Training. Η εκπαίδευση του δικτύου γίνεται από ένα σύνολο διανυσμάτων που έχουν συλλεχθεί από κείμενα νομικών αποφάσεων και έχει δημιουργηθεί ένα αρχείο .arff και τελικά εμφανίζονται τα αποτελέσματα της εκπαίδευσης όπως φαίνονται στην Εικόνα 15. Ύστερα με το κουμπί Open ανοίγει το παράθυρο της εικόνας 7 για την επιλογή ενός .arff αρχείου ώστε να χρησιμοποιηθεί για τον έλεγχο του μοντέλου. Τέλος πατώντας Classification κάνει την κατηγοριοποίηση και εμφανίζει τα αποτελέσματα στη μορφή που παρουσιάζονται στην Εικόνα 16.



Εικόνα 14 Κατηγοριοποίηση με το μοντέλο MultiLayer Perceptron



Εικόνα 15 Αποτελέσματα εκπαίδευσης του MultiLayer Percetron



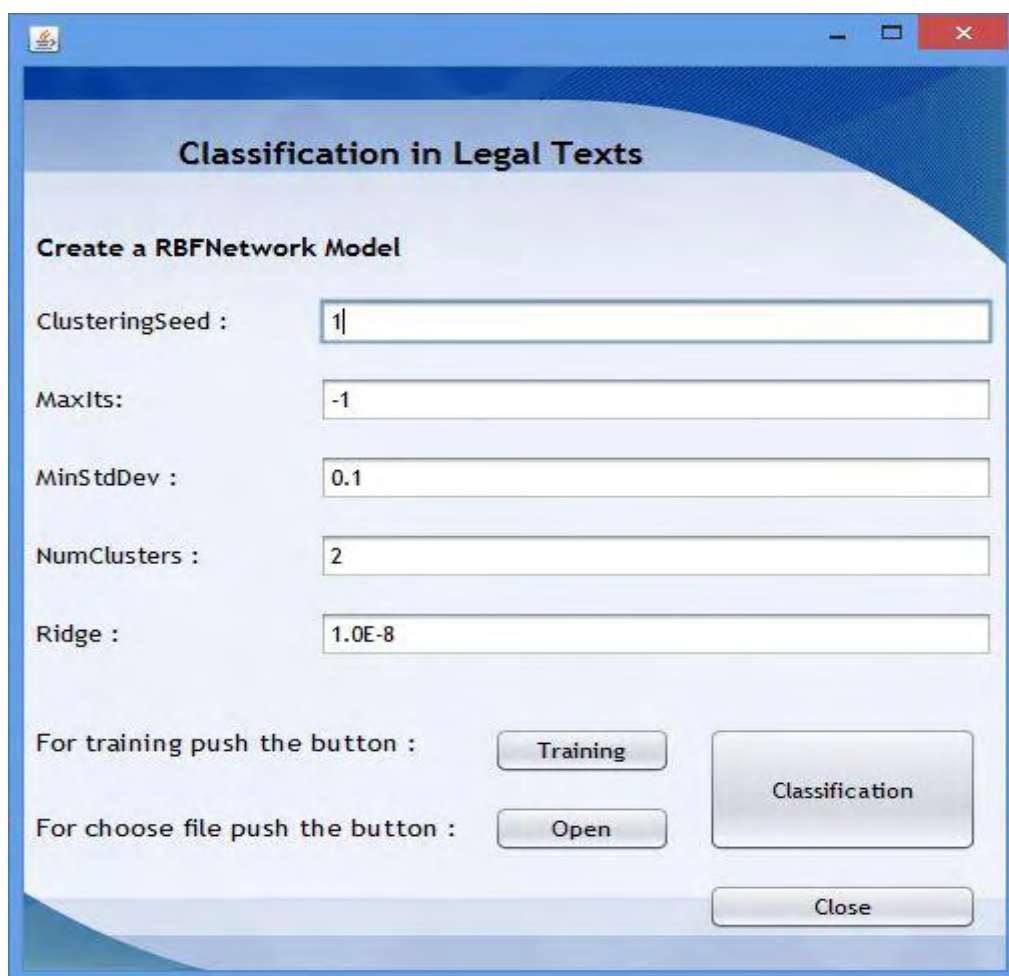
Εικόνα 16 Αποτελέσματα σε ποινικές αποφάσεις με το MultiLayer Perceptron

#### 4.4.2 Κατηγοριοποίηση με το μοντέλο RBFNetwork

Στην Εικόνα 17 φαίνεται το γραφικό περιβάλλον που εμφανίζεται με την επιλογή του Classify και RBFNetwork που δίνεται στην έναρξη της εφαρμογής. Σε αυτό το σημείο δίνεται αρχικά η δυνατότητα στο χρήστη να δημιουργήσει το μοντέλο με τον τρόπο που αυτός επιθυμεί ρυθμίζοντας μια σειρά παραμέτρων. Οι παράμετροι αυτοί είναι:

- **Clustering Seed:** Ρυθμίζει την παραγωγή τυχαίων αριθμών για την χρήση τους στον αλγόριθμο k-means. Η τιμή που δίνεται είναι το 1.
- **MaxIts:** Καθορίζει τον μέγιστο αριθμό των επαναλήψεων για την λογιστική παλινδρόμηση και έχει ως προεπιλογή τον αριθμό 1.
- **MinStdDev :** Ρυθμίζει την ελάχιστη τυπική απόκλιση για τις συστάδες. Η τιμή που δίνεται από το σύστημα είναι τι 0.1.
- **Num of Clusters:** Καθορίζει τον αριθμό των συναρτήσεων βάσης του μοντέλου ο οποίος έχει οριστεί ως προεπιλογή 2.
- **Ridge:** Ορίζει την ανώτατη τιμή τις λογιστικής παλινδρόμησης με τιμή προεπιλογής το 1.0E-8.

Η διαδικασία της εκπαίδευσης και του ελέγχου του μοντέλου είναι όμοια με αυτήν της υποενότητας 4.4.2.



Εικόνα 17 Κατηγοριοποίηση με το RBFNetwork

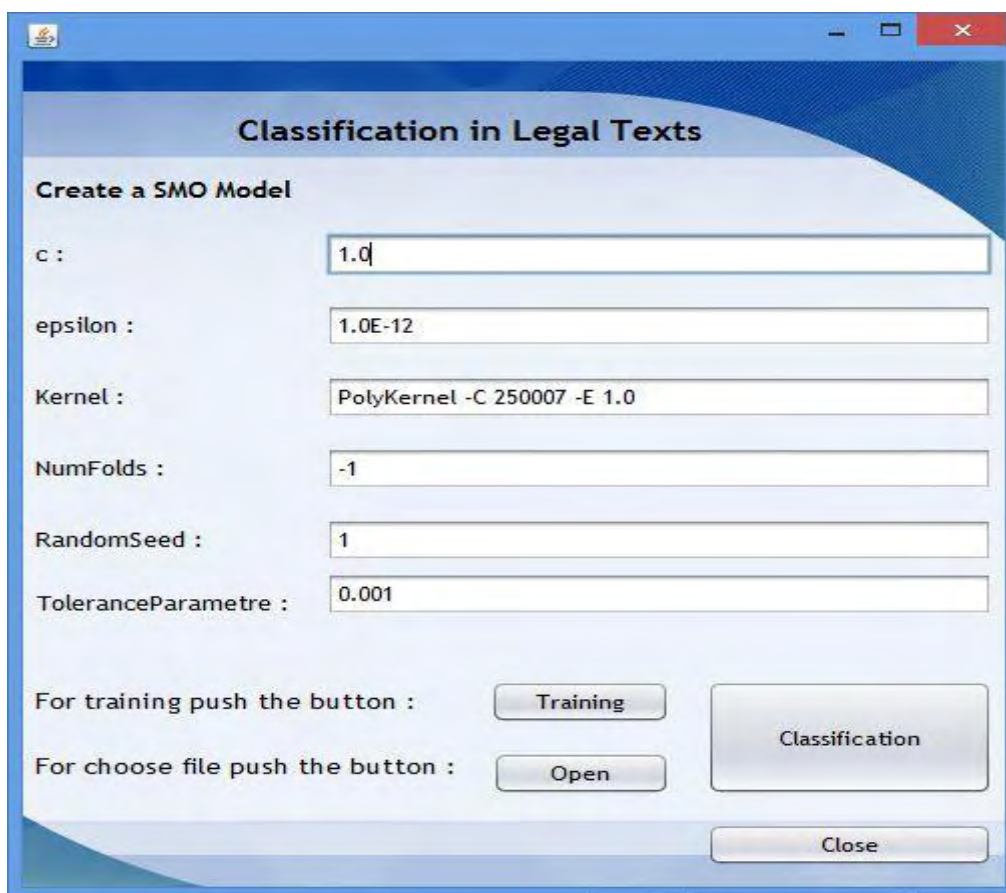


#### 4.4.3 Κατηγοριοποίηση με το μοντέλο SMO

Το γραφικό περιβάλλον για το μοντέλο SMO φαίνεται στην Εικόνα 18. Το μοντέλο έχει μια σειρά παραμέτρων για τον καθορισμό της δομής και της λειτουργίας του για τις οποίες το σύστημα δίνει τιμές προεπιλογής καθώς δίνει και την δυνατότητα στον χρήστη να τις τροποποιήσει. . Οι παράμετροι αυτοί είναι:

- Η σταθερά C (c) : Καθορίζει την πολυπλοκότητα και έχει τιμή προεπιλογής 1.
- Epsilon : Καθορίζει την στρογγυλοποίηση του σφάλματος και η τιμή που δίνεται από το σύστημα είναι το 1.0E-12.
- Kernel: Καθορίζει τον πυρήνα που χρησιμοποιείτε για την δημιουργία του μοντέλου. Ως προεπιλογή δίνεται ο PolyKernel με μέγεθος μνήμης 250007 και με τιμή εκθέτη 1.0.
- NumFolds: Καθορίζει τον αριθμό των πτυχώσεων και έχει τιμή προεπιλογής το -1.
- Random Seed: Είναι η γεννήτρια τυχαίων αριθμών και έχει τιμή προεπιλογής το 1.
- Tolerance Parametre : Η παράμετρος ανοχής έχει ως τιμή προεπιλογής την 0.001.

Η διαδικασία της εκπαίδευσης και του ελέγχου του μοντέλου είναι όμοια με αυτήν της υποενότητας 4.4.2.



Εικόνα 18 Κατηγοριοποίηση με το μοντέλο SMO

#### 4.4.4 Κατηγοριοποίηση με το μοντέλο Simple Logistic

Στην Εικόνα 19 φαίνεται το γραφικό περιβάλλον που εμφανίζεται με την επιλογή του Classify και RBFNetwork στην αρχή της εφαρμογής. Σε αυτό το σημείο δίνεται αρχικά η δυνατότητα στο χρήστη να δημιουργήσει το μοντέλο με τον τρόπο που αυτός επιθυμεί ρυθμίζοντας μια σειρά παραμέτρων. Οι παράμετροι αυτοί είναι:

- Heuristic Stop: Έχει από προεπιλογή την τιμή 50.
- Max Boosting Iterations: Ο μέγιστος αριθμός των επαναλήψεων έχει οριστεί με τιμή 500.
- Num Boosting Iterations: Ο σταθερός αριθμός των επαναλήψεων έχει οριστεί με μηδενική τιμή.
- Weight Trim Beta: Έχει οριστεί με τιμή 0 για να μην έχει κανένα βάρος.

Η διαδικασία της εκπαίδευσης και του ελέγχου του μοντέλου είναι όμοια με αυτήν της υποενότητας 4.4.2.



Εικόνα 19 Κατηγοριοποίηση με το μοντέλο Simple Logistic

## 4.5 ΤΟ ΕΡΓΑΛΕΙΟ WEKA

Το WEKA (Waikato Environment for Knowledge Analysis) είναι ένα πρόγραμμα ανάπτυξης εφαρμογών μηχανικής μάθησης και εξόρυξης γνώσης από δεδομένα (data mining), το οποίο αναπτύχθηκε στο τμήμα Επιστήμης Υπολογιστών του Waikato της Νέας Ζηλανδίας. Είναι ένα πακέτο λογισμικού ανοιχτού κώδικα υλοποιημένο σε Java, και χρησιμοποιείται ευρέως στον τομέα της εξόρυξης δεδομένων[2].

Παρέχει ένα σύνολο από υλοποιήσεις αλγορίθμων μηχανικής μάθησης και εξόρυξης δεδομένων, το οποίο περιλαμβάνει μεθόδους φιλτραρίσματος, επιλογής χαρακτηριστικών, ταξινόμησης, εύρεσης κανόνων συσχέτισης (association), κατηγοριοποίησης (categorization) και ομαδοποίησης (clustering), καθώς και μηχανισμούς για την προ-επεξεργασία δεδομένων (preprocessing) και μετέπειτα επεξεργασία αποτελεσμάτων (post-processing)[2].



Εικόνα 20 Γραφικό περιβάλλον του Weka 3.6

Οι ρουτίνες είναι υλοποιημένες σαν classes και ταξινομημένες σε packages, ενώ περιέχεται και ένα αναλυτικό γραφικό περιβάλλον (GUI Interface). Οι χρήστες έχουν τη δυνατότητα να χρησιμοποιούν τις υλοποιήσεις των αλγορίθμων τόσο από τη γραμμή εντολών όσο και από το γραφικό περιβάλλον του WEKA. Οι προγραμματιστές έχουν την δυνατότητα να χρησιμοποιούν τις υλοποιήσεις αυτές καλώντας τις αντίστοιχες κλάσεις του WEKA από τα δικά τους προγράμματα. Έτσι το WEKA μπορεί να αποτελέσει μια βιβλιοθήκη υλοποιήσεων αλγορίθμων εξόρυξης δεδομένων, οι κλάσεις της οποίας μπορούν να χρησιμοποιούνται για τη δημιουργία νέων προγραμμάτων[2].

Το WEKA χρησιμοποιεί αρχεία τα οποία περιέχουν εγγραφές, στα οποία κάθε εγγραφή περιγράφεται σε μια μόνο γραμμή, τα δε πεδία των εγγραφών έχουν σταθερό πλάτος και χωρίζονται με κενά για την περιγραφή των δεδομένων. Η είσοδος στο WEKA δίνεται ως σύνολο δεδομένων (data set), μέσω διαφόρων μορφών αρχείων όπως CSV

(comma separated values: \*.csv), binary serialized instances (\*.bsi) με πιο συχνή την χρήση της μορφής ARFF (\*.arff) η οποία παράγεται από το ίδιο το WEKA. Τα δεδομένα επίσης μπορούν να διαβαστούν και από μια ηλεκτρονική διεύθυνση ή από μια βάση δεδομένων (με JDBC)[2] .

Παραδείγματα αρχείων .arff παρουσιάζονται παρακάτω:

#### Παράδειγμα 1

```
@relation heart-disease-simplified
@attribute age numeric
@attribute sex {female, male}
@attribute chest_pain_type {typ_angina, asympt,
non_anginal, atyp_angina}
@attribute cholesterol numeric
@attribute exercise_induced_angina { no, yes}
@attribute class {present, not present}
@data
63,male,typ_angina,233,no,not_present
67,male,asympt,286,yes,present
67,male,asympt,229,yes,present
38,female,non_anginal,?,no,not_present
...
```

#### Παράδειγμα 2

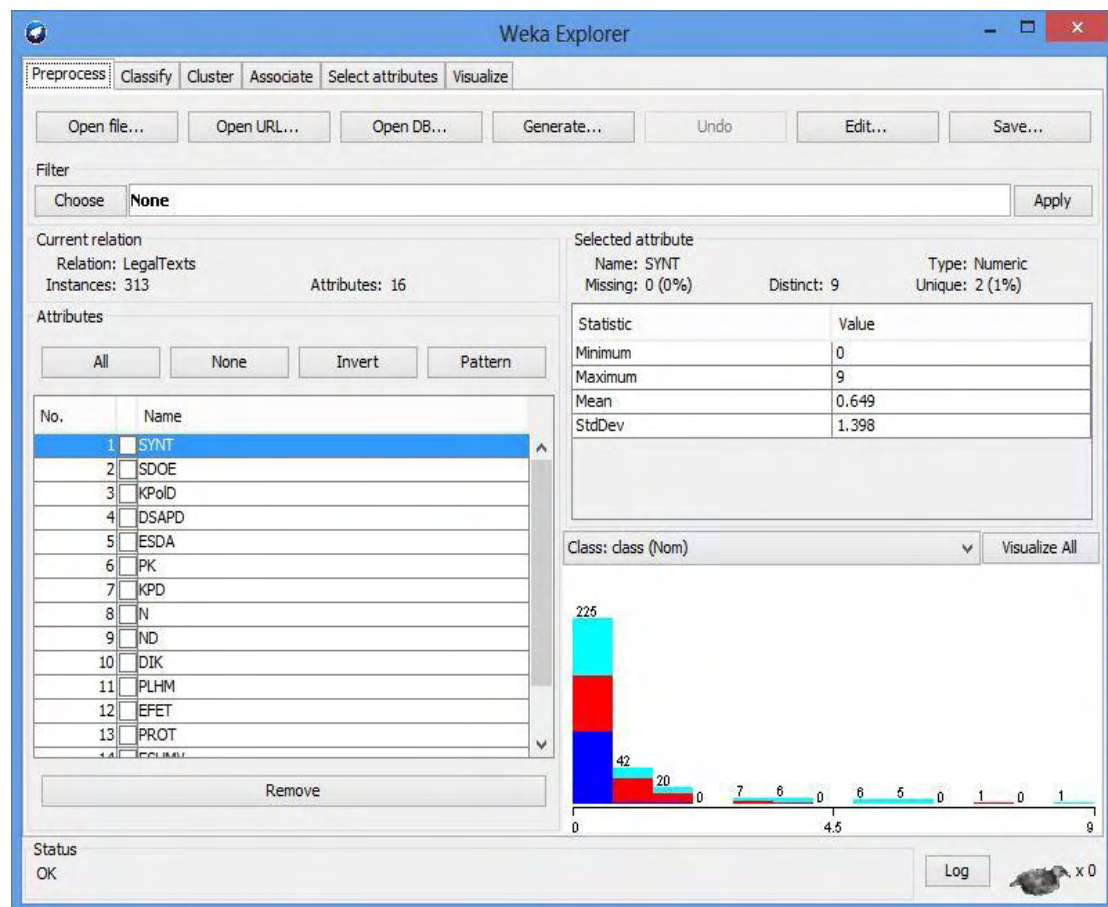
```
@RELATION LegalTexts

@ATTRIBUTE SYNT REAL
@ATTRIBUTE KPolD REAL
@ATTRIBUTE ESDA REAL
@ATTRIBUTE PK REAL
@ATTRIBUTE KPD REAL
@ATTRIBUTE N REAL
@ATTRIBUTE PLHM REAL
@ATTRIBUTE EFET REAL
@ATTRIBUTE AK REAL
@ATTRIBUTE class {NOMOY-3068, POINIKES, POLITIKES}

@DATA
0 0 0 0 0 0 0 5 0 NOMOY-3068
1 0 0 0 0 5 3 0 0 POINIKES
0 0 0 0 0 2 5 3 0 POINIKES
```

1	0	0	0	0	0	0	2	0	POINIKES
1	0	0	0	0	0	2	0	0	POINIKES
0	0	0	0	1	10	3	3	0	POINIKES
0	0	9	0	0	0	0	6	0	POLITIKES
7	0	7	0	0	0	0	31	0	POLITIKES
0	0	8	0	0	0	0	0	0	POLITIKES

Όπως αναφέρθηκε το WEKA μπορεί να χρησιμοποιηθεί για κατηγοριοποίηση δεδομένων. Ανοίγοντας το πρόγραμμα μέσω του μενού Explorer → Open file δίνεται η δυνατότητα να επιλεγεί ένα σύνολο δεδομένων (data set , αρχείο.arff) στο οποίο μπορούμε να εφαρμόσουμε τεχνικές που αφορούν την προ-επεξεργασία (preprocess), κατηγοριοποίηση (classify) , ομαδοποίηση (cluster), συσχέτιση (associate), επιλογή χαρακτηριστικών (select attributes) και οπτικοποίηση (visualize). Αφού επιλέξουμε ένα σύνολο δεδομένων, μπορούμε να δούμε τα δεδομένα να εμφανίζονται γραφικά για κάθε ένα από τα γνωρίσματα (attributes) ξεχωριστά καθώς και στατιστικές πληροφορίες για αυτά.



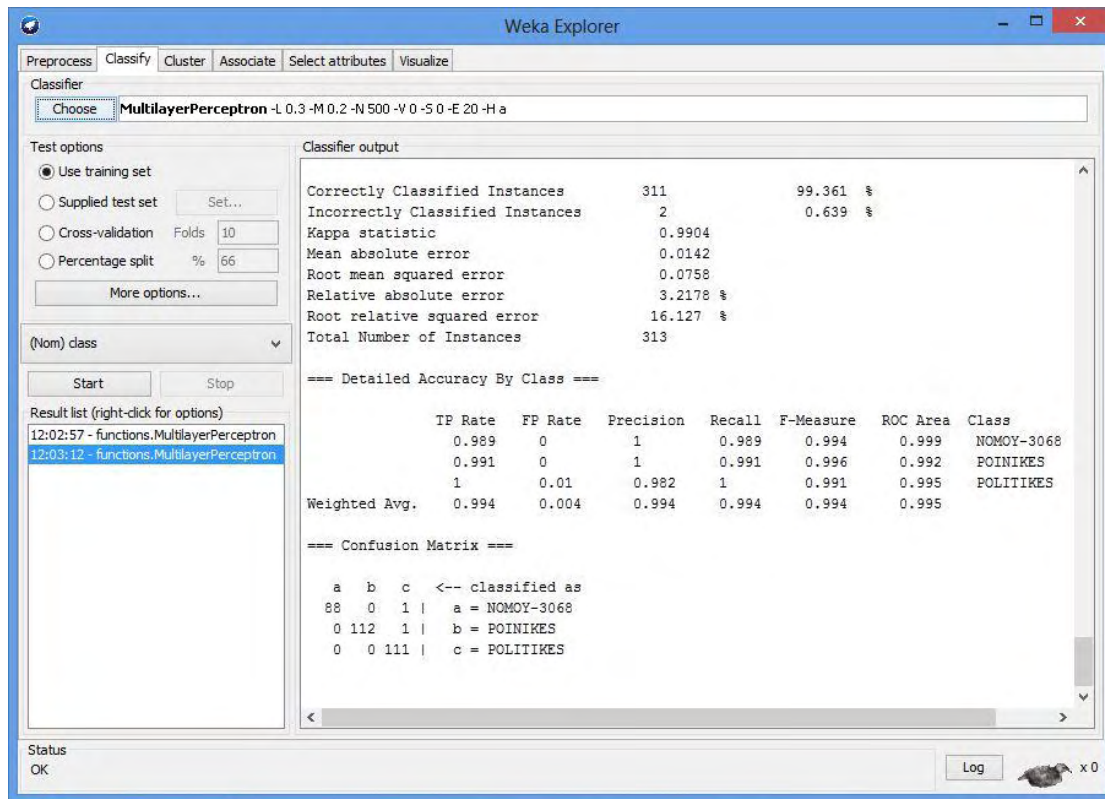
Εικόνα 21 Το παράθυρο Explorer του WEKA

Αφού έχει επιλεγεί ένα σύνολο δεδομένων είναι δυνατόν να γίνει κατηγοριοποίηση (classification), μεταβαίνοντας στην καρτέλα classify και επιλέγοντας έναν αλγόριθμο με βάση τον οποίο γίνεται η κατηγοριοποίηση. Με το κουμπί Choose δίνεται η δυνατότητα στο χρήστη να επιλέξει ποιόν αλγόριθμο θέλει. Το Weka προσφέρει πληθώρα αλγορίθμων για κατηγοριοποίηση, μερικοί από τους οποίους αναφέρονται παρακάτω:

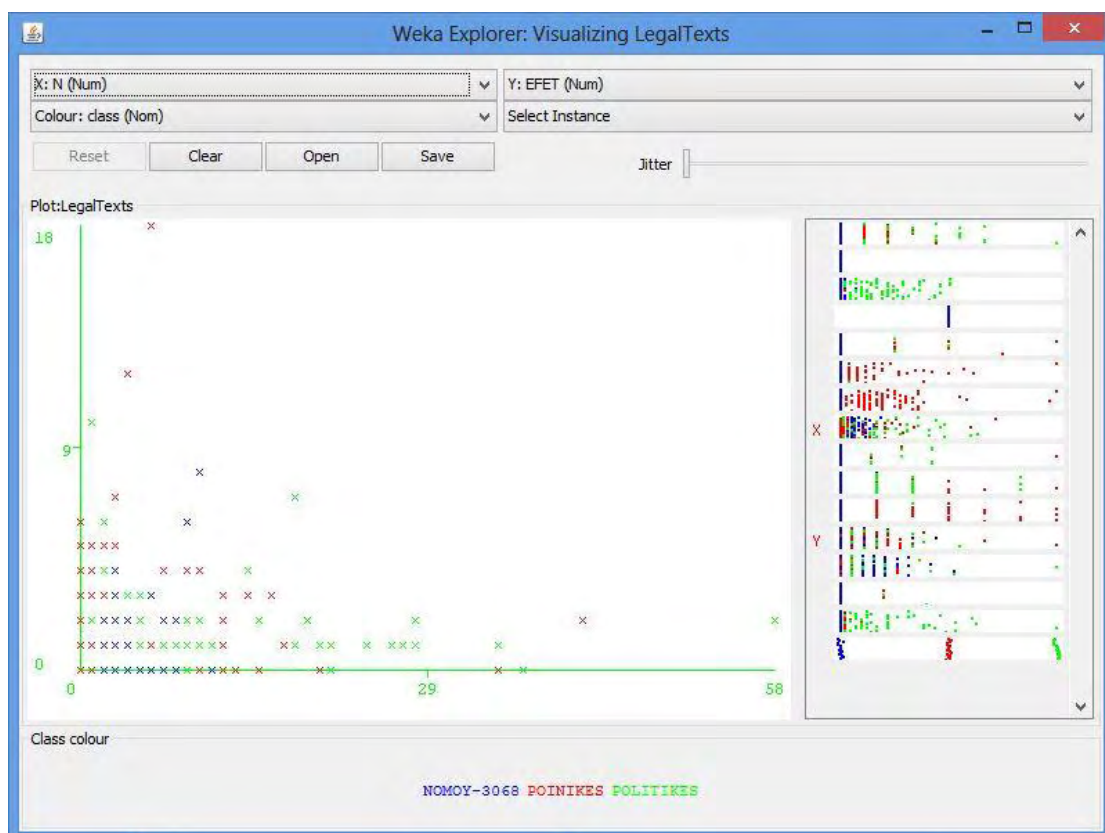
- NaiveBayes
- libSVM
- Logistic
- MultilayerPerceptron
- RBFNetwork
- SMO
- KStar
- ZeroR
- J48
- RandomForest
- RandomTree

Περιλαμβάνει επιλογή *training set* για την εκπαίδευση του μοντέλου που έχουμε αποφασίσει να χρησιμοποιήσουμε, επιλογή *cross validation* για την βελτίωση της ικανότητας γενίκευσης του μοντέλου και επιλογή *test set* για τον έλεγχο της ικανότητας γενίκευσης του μοντέλου.

Πατώντας το κουμπί Start μπορεί να ξεκινήσει η εκτέλεση του αλγορίθμου. Τα αποτελέσματα της κατηγοριοποίησης εμφανίζονται στο δεξί μέρος της καρτέλας, ενώ επίσης παρέχεται η δυνατότητα οπτικοποίησης (visualize) του αποτελέσματος με εμφάνιση γραφικής παράστασης των δεδομένων ανάλογα με τις ομάδες που είχαν καθοριστεί κατά την κατηγοριοποίηση.



Εικόνα 22 Το παράθυρο του Classify στο WEKA



Εικόνα 23 Οπτικοποίηση του αποτελέσματος της κατηγοριοποίησης

## ΚΕΦΑΛΑΙΟ 5

### ΣΥΜΠΕΡΑΣΜΑΤΑ – ΜΕΛΛΟΝΤΙΚΗ ΕΡΕΥΝΑ

#### 5.1 ΣΥΜΠΕΡΑΣΜΑΤΑ

Όπως προκύπτει τελικά από την παρούσα πτυχιακή εργασία, υπάρχουν αρκετοί τρόποι επεξεργασίας κειμένων με σκοπό την εξόρυξη γνώσης από αυτά. Η αυστηρά καθορισμένη δομή των νομικών κειμένων που χρησιμοποιήθηκαν αποτέλεσε σημαντικό επίτευγμα στην εξαγωγή αποτελεσμάτων κάνοντας πιο εύκολη την επεξεργασία τους σε σχέση με τα κείμενα φυσικής γλώσσας και ελεύθερου λόγου.

Τα τελικά αποτελέσματα που εξήχθησαν αφορούσαν την εξαγωγή περίληψης, την αναζήτηση και ανάκτηση και την κατηγοριοποίηση από τις μεθόδους επεξεργασίας κειμένων.

Για την εξαγωγή περιληπτικής παρουσίασης της πληροφορίας τα αποτελέσματα της απόδοσης του συστήματος κατά την διαδικασία αυτή, όπως είδαμε ήταν τα ακόλουθα (Πίνακας 8):

	αριθμός	Προεδρεύων ο αντιπρόεδρος	Εισηγητής ο αεροπαγίτης	Εισαγγελεύς	Δικηγόρος
Precision	100%	100%	96.44%	99.52%	77.83%
Recall	100%	100%	96.44%	99.52%	77.83%

Πίνακας 8 Τιμές ανάκλησης και ακρίβειας για όλες τις κατηγορίες του HeaderSummarisation

Η πρώτη παρατήρηση αρχικά είναι πως τα επίπεδα ανάκλησης και ακρίβεια κυμαίνονται σε υψηλά επίπεδα, εκτός από την κατηγορία «δικηγόρος» όπου είναι κάτω από 90% (77.83%) και για τις δυο μετρικές. Αυτό συμβαίνει γιατί η κατηγορία αυτή παρουσιάζεται πιο αυθαίρετα μέσα στα κείμενα σε σχέση με τις υπόλοιπες κατηγορίες.

Η δεύτερη παρατήρηση είναι πως σε όλες τις κατηγορίες η ανάκληση είναι ίση με την ακρίβεια γιατί πάντα το σύστημα εντοπίζει ότι υπάρχει σχετική πληροφορία, αλλά δεν ανακτά πάντα την σωστή πληροφορία.

Όπως είδαμε έγινε και εντοπισμός και επισήμανση των αναφορών που συναντώνται στα νομικά κείμενα. Οι αναφορές αυτές έχουν συγκεκριμένη δομή μέσα στα κείμενα πράγμα που επιτρέπει την παραγωγή κανόνων για τον αυτόματο εντοπισμό τους. Οι αναφορές αυτές ανάλογα με το είδος τους χωρίστηκαν σε κατηγορίες και για κάθε κατηγορία μελετήθηκε η αποτελεσματικότητα του συστήματος για την ανάκτησή τους.



Τα αποτελέσματα αποτυπώνονται στον Πίνακα 9 :

<i>Κατηγορίες νομικών αναφορών</i>	<i>Ανάκληση</i>	<i>Ακρίβεια</i>
<b>Συντ</b>	100%	93.33%
<b>ΣΔΟΕ</b>	0%	0%
<b>ΚΠολΔ</b>	100%	68%
<b>ΔΣΑΠΔ</b>	0%	0%
<b>ΕΣΔΑ</b>	100%	100%
<b>ΠΚ</b>	100%	80%
<b>ΚΠΔ</b>	98.18%	98.18%
<b>Ν.</b>	96.77%	73.61%
<b>ν.δ</b>	100%	90.9%
<b>Δικαστήρ</b>	0%	0%
<b>Πλημ</b>	100%	78.57%
<b>Εφετ</b>	100%	73.68%
<b>Πρωτοδικ</b>	100%	96.77%
<b>ΕΣυμβ</b>	100%	100%
<b>ΑΚ</b>	95%	95%

Πίνακας 9 Τιμές ανάκλησης και ακρίβειας για όλες τις κατηγορίες αναφορών κατά τον εντοπισμό τους

Αρχικά παρατηρούμε πως οι κατηγορίες ΣΔΟΕ, ΔΣΑΠΔ και Δικαστηρ παρουσιάζουν μηδενική αποτελεσματικότητα, πράγμα που σημαίνει πως δεν υπήρχαν στο σύνολο αποφάσεων που εξετάστηκε και συνεπώς δεν ανακτήθηκαν από το σύστημα. Αυτό τις κατατάσσει στις μη σημαντικές κατηγορίες για τον γενικό χαρακτηρισμό και την απόδοση του κειμένου σε μία κατηγορία αποφάσεων. Δηλαδή οι εν λόγω κατηγορίες θα μπορούσαν κάλλιστα να μην χρησιμοποιηθούν κατά την διαδικασία της κατηγοριοποίησης των νομικών αποφάσεων, χωρίς να επηρεάσουν το τελικό αποτέλεσμα.

Ύστερα για της υπόλοιπες κατηγορίες βλέπουμε πως η χαμηλότερη ακρίβεια που σημειώνεται είναι στο 95% ενώ η χαμηλότερη ανάκληση στο 73.68%. Συνεπώς μπορούμε να πούμε πως το σύστημα ανάκτησης παρουσιάζει καλή αποτελεσματικότητα.

Το σύστημα κατά τον εντοπισμό των αναφορών έκανε και μέτρηση αυτών, για κάθε κατηγορία ξεχωριστά, με στόχο την παραγωγή ιστογράμματος για κάθε κείμενο. Ουσιαστικά προχωρούσε στην δημιουργία ενός διανύσματος (προτύπου) με κάθε στοιχείο του να είναι ένας ακέραιος αριθμός που αντιστοιχεί σε μία κατηγορία αναφορών. Παρατηρώντας τα πρότυπα αυτά διαπιστώθηκε ότι οι κατηγορίες που παραπέμπουν άμεσα στο τμήμα από όπου προήλθε το κείμενο παρουσιάζουν υψηλές συχνότητες. Για παράδειγμα στα κείμενα που ανήκουν στην κατηγορία «Πολιτικά»

υπερισχύει η κατηγορία του Κώδικα Πολιτικής Δικονομίας (ΚΠολΔ), ενώ σε κείμενα «Ποινικά» υπερτερούν οι κατηγορίες του Ποινικού Κώδικα (ΠΚ) και του Κώδικα Ποινικής Δικονομίας (ΚΠΔ). Έτσι συμπερασματικά οδηγηθήκαμε σε μεθόδους Μηχανικής Μάθησης με αποτέλεσμα να είναι δυνατόν να επιτευχθεί η δημιουργία ενός συστήματος επιβλεπόμενης μάθησης που επιτυγχάνει ταξινόμηση των νομικών κειμένων με βάσει τα πρότυπα τους.

Για την ταξινόμηση των κειμένων μελετήθηκαν τα μοντέλα MultiLayer Perceptron, RBFNetwork, SMO και Simple Logistic. Για κάθε ένα από τα μοντέλα που αναφέρθηκαν εφαρμόστηκε η τεχνική 10-folds cross validation για την εκτίμηση της ικανότητας γενίκευσης του κάθε μοντέλου. Τα αποτελέσματα παρουσιάζονται στον Πίνακα 10 :

	MultiLayer Perceptron	RBFNetworks	SMO PolyKernel	Simple Logistic
Correctly Classified Instances	94.2492%	96.1661%	93.6102%	97.7636%
Precision	94.3%	96.2%	93.9%	97.8%
Recall	94.2%	96.2%	93.6%	97.8%

Πίνακας 10 Τιμές ανάκλησης, ακρίβειας και σωστά ταξινομημένων διανυσμάτων για τα μοντέλα που χρησιμοποιήθηκαν στο 10-fold cross validation

Όπως φαίνεται από το πινακάκι καλύτερα αποτελέσματα επιτυγχάνει το Simple Logistic έχοντας την καλύτερη ικανότητα γενίκευσης και τα υψηλότερα ποσοστά σε ανάκληση και ακρίβεια. Για να επιβεβαιώσουμε αυτό το συμπέρασμα ελέγχουμε τα αποτελέσματα των μοντέλων κατά την εκπαίδευση (training set) και κατά το σύνολο ελέγχου (test set). Τα αποτελέσματα κατά την εκπαίδευση φαίνονται παρακάτω (Πίνακας 11):

	MultiLayer Perceptron	RBFNetworks	SMO PolyKernel	Simple Logistic
Correctly Classified Instances	99.361%	97.1246%	94.2492%	99.6805%
Precision	99.4%	97.2%	94.6%	99.7%
Recall	99.4%	97.1%	94.2%	99.7%

Πίνακας 11 Τιμές ανάκλησης, ακρίβειας και σωστά ταξινομημένων διανυσμάτων για τα μοντέλα που χρησιμοποιήθηκαν στο training set

Βλέπουμε πως στην εκπαίδευση υπερσχύει το μοντέλο Simple Logistic έναντι των υπολοίπων.

Έπειτα δοκιμάστηκαν κείμενα πολιτικών αποφάσεων σε όλα τα μοντέλα που χρησιμοποιήθηκαν με τα αποτελέσματα να φαίνονται στον Πίνακα 12 :

	MultiLayer Perceptron	RBFNetworks	SMO PolyKernel	Simple Logistic
Correctly Classified Instances	96.667%	96.667%	100%	100%
Precision	100%	100%	100%	100%
Recall	96.7%	96.7%	100%	100%

Πίνακας 12 Τιμές ανάκλησης, ακρίβειας και σωστά ταξινομημένων διανυσμάτων για τα μοντέλα που χρησιμοποιήθηκαν στο test set για κείμενα από το πολιτικό τμήμα

Τα αποτελέσματα για κείμενα αποφάσεων από το ποινικό τμήμα είναι τα εξής (Πίνακας 13) :

	MultiLayer Perceptron	RBFNetworks	SMO PolyKernel	Simple Logistic
Correctly Classified Instances	96.667%	100%	100%	100%
Precision	100%	100%	100%	100%
Recall	96.7%	100%	100%	100%

Πίνακας 13 Τιμές ανάκλησης, ακρίβειας και σωστά ταξινομημένων διανυσμάτων για τα μοντέλα που χρησιμοποιήθηκαν στο test set για κείμενα από το ποινικό τμήμα

Τα αποτελέσματα για κείμενα αποφάσεων από το τμήμα Νόμου 3068/2002 είναι τα εξής (Πίνακας 14):

	MultiLayer Perceptron	RBFNetworks	SMO PolyKernel	Simple Logistic
Correctly Classified Instances	100%	96.4286%	96.4286%	100%
Precision	100%	100%	100%	100%
Recall	100%	96.4%	96.4%	100%

Πίνακας 14 Τιμές ανάκλησης, ακρίβειας και σωστά ταξινομημένων διανυσμάτων για τα μοντέλα που χρησιμοποιήθηκαν στο test set για κείμενα από το τμήμα Νόμου 3068/2002

Τελικά καταλήγουμε στο συμπέρασμα πως η καλύτερη μέθοδος ταξινόμησης από τις παραπάνω είναι ο αλγόριθμος Simple Logistic καθώς παρουσιάζει μεγαλύτερα ποσοστά στην τεχνική cross validation, στο σύνολο εκπαίδευσης (training set) και στο

σύνολο ελέγχου όπου έχει 100% επιτυχία και για τις τρεις κατηγορίες νομικών κειμένων (Ποινικές αποφάσεις, Πολιτικές αποφάσεις, αποφάσεις Νόμου 3068/2002).

## 5.2 ΜΕΛΛΟΝΤΙΚΗ ΕΡΕΥΝΑ

Συμπερασματικά, μπορούμε να δούμε πως η ανάπτυξη της εφαρμογής βοήθησε στην εξοικονόμηση χρόνου για την διαχείριση της πληροφορίας στα νομικά κείμενα. Βέβαια θα μπορούσε να λάβει κάποιες σημαντικές επεκτάσεις που θα βοηθούσαν ακόμα περισσότερο στην ανάλυση τέτοιου είδους κειμένων.

Αρχικά θα μπορούσαν να αναλυθούν και να χρησιμοποιηθούν μέθοδοι μη επιβλεπόμενης μάθησης όπως η ομαδοποίηση (clustering) με διάφορους αλγορίθμους που παρέχει το εργαλείο WEKA, όπως DBSCAN, EM, Simple kMeans, XMeans. Έτσι μπορούν να συγκριθούν τα αποτελέσματα της κατηγοριοποίησης και της ομαδοποίησης με αποτέλεσμα να εξαχθούν κάποια συμπεράσματα.

Επιπλέον, η οπτικοποίηση (visualization) των αποτελεσμάτων της κατηγοριοποίησης και της ομαδοποίησης θα βοηθούσε ακόμη περισσότερο το χρήστη να δει τις ομάδες και να μελετήσει τις ιδέες, καθώς και να δει ποιες ομάδες βρίσκονται μεταξύ τους πιο κοντά νοηματικά για να βγάλει επιπλέον συμπεράσματα.

Τέλος θα μπορούσε να υλοποιηθεί ένα σύστημα που να εξερευνά απευθείας από τον Παγκόσμιο Ιστό τέτοιου είδους κείμενα, έτσι θα μειωθεί ακόμα περισσότερο ο όγκος της δουλειάς του χρήστη.

## BIBΛΙΟΓΡΑΦΙΑ

---

1. Vishal Gupta and Gurpreet S. Lehal, “A Survey of Text Mining Techniques and Applications”, Journal of Emerging Technologies in Web Intelligence, VOL. 1, NO. 1, August 2009.
2. Remco R. Bouckaert, Eibe Frank, Mark Hall, Richard Kirkbly, Peter Reuremann, Alex Seewald and David Scuse, “WEKA Manual for Version 3-6-12”, University of Waikato, Hamilton, New Zealand, 2002-2004
3. Κωνσταντίνος Διαμαντάρας, «ΤΕΧΝΗΤΑ ΝΕΥΡΩΝΙΚΑ ΔΙΚΤΥΑ», Κλειδάριθμος, Αθήνα 2007.
4. Pang-Ning Tan, Michael Steinbach and Vipin Kumar, «ΕΙΣΑΓΩΓΗ ΣΤΗΝ ΕΞΟΡΥΞΗ ΔΕΔΟΜΕΝΩΝ», Τζιόλας, Αθήνα 2010.
5. Ιωάννης Μανωλόπουλος και Απόστολος Ν. Παπαδόπουλος, «Ανάκτηση Πληροφορίας», [http://delab.csd.auth.gr/courses/c\\_ir/irbook.pdf](http://delab.csd.auth.gr/courses/c_ir/irbook.pdf)
6. D. Michie, D.J. Spiegelhalter and C.C. Taylor, “Machine Learning Neural and Statistical Classification”, February 17, 1994
7. Ah-Hwee Tan, “Text Mining: The state of the art and the challenges”, Kent Ridge Digital Labs, 21 Heng Mui Keng Terrace, Singapore 11963
8. Ron Kohavi, “A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection”, International Joint Conference on Artificial Intelligence (IJCAI), Computer Science Department, Stanford University, Stanford 1995
9. Jerome Euzenat, “Semantic Precision and Recall for Ontology Alignment Evaluation”, IJCAI, INRIA Rhone-Alpes, Monbonnot, France
10. Qiuming Zhu, Yao Cai and Luzheng Liu, “A global learning algorithm for a RBF network”, Neural Networks 12, Digital Imaging and Computer Vision Laboratory, University of Nebraska at Omaha, Omaha, USA, 10 November 1998, pp.527-540
11. Dennis W. Ruck, Steven K. Rogers and Matthew Kabrisky, “Feature Selection Using a Multilayer Perceptron”, Journal of Neural Network Computing, Volume 2, Number 2, Ohio, 1990, pp. 40-48
12. Jeffrey L. Solka, “Text Date Mining: Theory and Methods”, Natural Surface Warfare Center Dahlgren Division Attention Jeff Solka, 18444 Frontage Rd, Suite 328, Dahlgren Virginia, 22448-5161
13. M. Martinez-Gonzalez, “Reference extraction and resolution for legal texts”, PReMI’05 Proc. First Int. Conf. Pattern Recognit. Mach. Intell., pp. 218–221, 2005. [http://link.springer.com/chapter/10.1007/11590316\\_29](http://link.springer.com/chapter/10.1007/11590316_29)
14. Γενική Γραμματεία της Κυβέρνησης, «Εγχειρίδιο οδηγιών για την κωδικοποίηση της νομοθεσίας», [http://www.ggk.gov.gr/wpcontent/uploads/2010/02/teliko\\_egxeiridio\\_odigion\\_gia\\_tin\\_kodikopoiisi\\_tis\\_nomothesias.pdf](http://www.ggk.gov.gr/wpcontent/uploads/2010/02/teliko_egxeiridio_odigion_gia_tin_kodikopoiisi_tis_nomothesias.pdf)
15. AREIOS PAGOS, Supreme Civil and Criminal Court of Greece, <http://www.areiospagos.gr/>

16. Mariusz Bernacki and Przemyslaw Wlodarczyk, “Principles of training multi-layer neural network using backpropagation”, 2005.  
[http://home.agh.edu.pl/~vlsi/AI/backp\\_t\\_en/backprop.html](http://home.agh.edu.pl/~vlsi/AI/backp_t_en/backprop.html)
17. SANN Overviews – Network Types, “The Multilayer Perceptron Neural Networks”,  
<http://documentation.statsoft.com/STATISTICAHelp.aspx?path=SANN/Overview/SANNOverviewsNetworkTypes>
18. Christos Stergiou and Dimitrios Siganos, “Neural Networks”,  
[http://www.doc.ic.ac.uk/~nd/surprise\\_96/journal/vol4/cs11/report.html](http://www.doc.ic.ac.uk/~nd/surprise_96/journal/vol4/cs11/report.html)
19. Apache POI – the Java API for Microsoft Documents  
<https://poi.apache.org/>
20. JFree Chart, sourceforge download,  
<http://sourceforge.net/projects/jfreechart/files/1.%20JFreeChart/>

# ΠΑΡΑΡΤΗΜΑ Ι : ΚΩΔΙΚΑΣ

---

Ο κώδικας της εφαρμογής που χρησιμοποιήθηκε για την υλοποίηση της εφαρμογής διαχείρισης νομικών κειμένων γράφτηκε και γλώσσα Java και απαρτίζεται από δεκατέσσερις κλάσεις.

1. MainFrameForm
2. HeaderSummarisationFrame
3. HeaderSummarisationResults
4. ExportRefMenu
5. ExportReferences
6. ExportRefResults
7. ReadAndWriteDocx
8. CreateVectorExcel
9. JFreeChartBar
10. AllFinalsVariables
11. PerceptronFrame
12. RBFNetworkFrame
13. SMOFrame
14. SimpleLogisticFrame

## Κλάση MainFrameForm

```
package textdatamining;

import java.awt.Graphics2D;
import java.awt.Image;
import java.awt.RenderingHints;
import java.awt.Toolkit;
import java.awt.image.BufferedImage;
import javax.swing.ImageIcon;
import javax.swing.JLabel;

/**
 *
 * @author spiros
 */
public class MainFrameForm extends javax.swing.JFrame {

    private String path;
    private int choice;
    String itemText = "MultiLayerPerceptron";//comboBox, αρχικοποιείτε με
    μια τιμή

    public MainFrameForm() { //constructor
        Image img = Toolkit.getDefaultToolkit().createImage("Blueprint
border.jpg");
        ImageIcon im = new ImageIcon();
    }
}
```

```

BufferedImage img1 = scaleImage(470, 300, "Blueprint border.jpg");
im.setImage(img1);
setContentPane(new JLabel(im));

initComponents();

choice = 0;
this.path = path;
}

/**
 * This method is called from within the constructor to initialize the
form.
 * WARNING: Do NOT modify this code. The content of this method is
always
 * regenerated by the Form Editor.
 */
@SuppressWarnings("unchecked")
// <editor-fold defaultstate="collapsed" desc="Generated Code">
private void initComponents() {

    JLabel1 = new javax.swing.JLabel();
    NextButton = new javax.swing.JButton();
    ExitButton = new javax.swing.JButton();
    HeaderSummarisation = new javax.swing.JRadioButton();
    InformationExtraction = new javax.swing.JRadioButton();
    Classify = new javax.swing.JRadioButton();
    ComboBox = new javax.swing.JComboBox();

    setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
    setCursor(new java.awt.Cursor(java.awt.Cursor.DEFAULT_CURSOR));

    JLabel1.setFont(new java.awt.Font("Trebuchet MS", 1, 18)); // NOI18N
    JLabel1.setText("Text Mining in Legal Texts");
    JLabel1.setVerticalTextPosition(javax.swing.SwingConstants.TOP);

    NextButton.setFont(new java.awt.Font("Trebuchet MS", 0, 12)); //
NOI18N
    NextButton.setText("Next >>");
    NextButton.setPreferredSize(new java.awt.Dimension(100, 25));
    NextButton.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            NextButtonActionPerformed(evt);
        }
    });

    ExitButton.setFont(new java.awt.Font("Trebuchet MS", 0, 12)); //
NOI18N
    ExitButton.setText("Exit");
    ExitButton.setPreferredSize(new java.awt.Dimension(100, 25));
    ExitButton.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            ExitButtonActionPerformed(evt);
        }
    });

    HeaderSummarisation.setFont(new java.awt.Font("Trebuchet MS", 1,
14)); // NOI18N

```



```

HeaderSummarisation.setText("Header Summarisation");
HeaderSummarisation.addActionListener(new
java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        HeaderSummarisationActionPerformed(evt);
    }
});

InformationExtraction.setFont(new java.awt.Font("Trebuchet MS", 1,
14)); // NOI18N
InformationExtraction.setText("Information Extraction");
InformationExtraction.addActionListener(new
java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        InformationExtractionActionPerformed(evt);
    }
});

Classify.setFont(new java.awt.Font("Trebuchet MS", 1, 14)); //
NOI18N
Classify.setText("Classify");
Classify.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        ClassifyActionPerformed(evt);
    }
});

ComboBox.setFont(new java.awt.Font("Trebuchet MS", 0, 12)); //
NOI18N
ComboBox.setModel(new javax.swing.DefaultComboBoxModel(new String[]
{ "MultiLayerPerceptron", "RBFNetwork", "SMO", "SimpleLogistic" }));
ComboBox.setToolTipText("");
ComboBox.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        ComboBoxActionPerformed(evt);
    }
});

javax.swing.GroupLayout layout = new
javax.swing.GroupLayout(getContentPane());
getContentPane().setLayout(layout);
layout.setHorizontalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
layout.createSequentialGroup()
        .addGap(18, 18, 18)
        .addComponent(NextButton,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addGap(18, 18, 18)
        .addComponent(ExitButton,
javax.swing.GroupLayout.PREFERRED_SIZE, 92,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addGap(53, 53, 53)

```

```

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
NG)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)
    .addGroup(layout.createSequentialGroup()
        .addComponent(HeaderSummarisation)
        .addGap(18, 18, 18)
        .addComponent(InformationExtraction,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE))
    .addGroup(layout.createSequentialGroup()
        .addComponent(Classify)
        .addGap(18, 18, 18)
        .addComponent(ComboBox,
javax.swing.GroupLayout.PREFERRED_SIZE, 267,
javax.swing.GroupLayout.PREFERRED_SIZE))
        .addComponent(jLabel1))
        .addContainerGap(54, Short.MAX_VALUE))
);
layout.setVerticalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(layout.createSequentialGroup()
        .addContainerGap()
        .addComponent(jLabel1,
javax.swing.GroupLayout.PREFERRED_SIZE, 40,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addGap(44, 44, 44)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
INE)
    .addComponent(HeaderSummarisation)
    .addComponent(InformationExtraction))
    .addGap(18, 18, 18)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
INE)
    .addComponent(Classify)
    .addComponent(ComboBox,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 63,
Short.MAX_VALUE)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)
ING)
    .addComponent(ExitButton,
javax.swing.GroupLayout.PREFERRED_SIZE, 29,
javax.swing.GroupLayout.PREFERRED_SIZE)
    .addComponent(NextButton,
javax.swing.GroupLayout.PREFERRED_SIZE, 28,
javax.swing.GroupLayout.PREFERRED_SIZE))
    .addContainerGap()
);

```

```

        pack();
    } // </editor-fold>

    private void NextButtonActionPerformed(java.awt.event.ActionEvent evt) {
        //Επιλογή λειτουργίας
        if (choice == 1) {
            new HeaderSummarisationFrame().setVisible(true);
        }
        if (choice == 2) {
            new SecondStepMenu().setVisible(true);
        }
        if (choice == 3) {
            if (itemText == "MultiLayerPerceptron") {
                new PerceptronFrame().setVisible(true);
            } else if (itemText == "RBFNetwork") {
                new RBFNetworkFrame().setVisible(true);
            } else if (itemText == "SMO") {
                new SMOFrame().setVisible(true);
            } else if (itemText == "SimpleLogistic"){
                new SimpleLogisticFrame().setVisible(true);
            }
        }
    }

}

private void ExitButtonActionPerformed(java.awt.event.ActionEvent evt) {
    System.exit(0); //Τερματίζει την εφαρμογή
}

private void
HeaderSummarisationActionPerformed(java.awt.event.ActionEvent evt) {
    choice = 1;
    InformationExtraction.setSelected(false);
    Classify.setSelected(false);
}

private void
InformationExtractionActionPerformed(java.awt.event.ActionEvent evt) {
    choice = 2;
    HeaderSummarisation.setSelected(false);
    Classify.setSelected(false);
}

private void ClassifyActionPerformed(java.awt.event.ActionEvent evt) {
    choice = 3;
    HeaderSummarisation.setSelected(false);
    InformationExtraction.setSelected(false);
}

private void ComboBoxActionPerformed(java.awt.event.ActionEvent evt) {
    ComboBox.getSelectedItem();
    itemText = (String) ComboBox.getSelectedItem();
}

/**
 * @param args the command line arguments
 */
public static void main(String args[]) {

```

```

        /* Set the Nimbus look and feel */
        //<editor-fold defaultstate="collapsed" desc=" Look and feel setting
code (optional) ">
        /* If Nimbus (introduced in Java SE 6) is not available, stay with
the default look and feel.
        * For details see
http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
        */
        try {
            for (javax.swing.UIManager.LookAndFeelInfo info :
javax.swing.UIManager.getInstalledLookAndFeels()) {
                if ("Nimbus".equals(info.getName())) {

javax.swing.UIManager.setLookAndFeel(info.getClassName());
                    break;
                }
            }
        } catch (ClassNotFoundException ex) {

java.util.logging.Logger.getLogger(MainFrameForm.class.getName()).log(java.u
til.logging.Level.SEVERE, null, ex);
        } catch (InstantiationException ex) {

java.util.logging.Logger.getLogger(MainFrameForm.class.getName()).log(java.u
til.logging.Level.SEVERE, null, ex);
        } catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(MainFrameForm.class.getName()).log(java.u
til.logging.Level.SEVERE, null, ex);
        } catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(MainFrameForm.class.getName()).log(java.u
til.logging.Level.SEVERE, null, ex);
        }
    }
    //</editor-fold>

    /* Create and display the form */
    java.awt.EventQueue.invokeLater(new Runnable() {
        public void run() {
            new MainFrameForm().setVisible(true);
        }
    });
}

// Variables declaration - do not modify
private javax.swing.JRadioButton Classify;
private javax.swing.JComboBox ComboBox;
private javax.swing.JButton ExitButton;
private javax.swing.JRadioButton HeaderSummarisation;
private javax.swing.JRadioButton InformationExtraction;
private javax.swing.JButton NextButton;
private javax.swing.JLabel jLabell;
// End of variables declaration

BufferedImage scaleImage(int WIDTH, int HEIGHT, String filename) {
//Φωρτώνει την εικόνα στο gui
    BufferedImage bi = null;

    try {

```

```

        ImageIcon ii = new ImageIcon(filename); // path to image
        bi = new BufferedImage(WIDTH, HEIGHT,
BufferedImage.TYPE_INT_RGB);
        Graphics2D g2d = (Graphics2D) bi.createGraphics();
        g2d.addRenderingHints(new
RenderingHints(RenderingHints.KEY_RENDERING,
RenderingHints.VALUE_RENDER_QUALITY));
        g2d.drawImage(ii.getImage(), 0, 0, WIDTH, HEIGHT, null);
    } catch (Exception e) {
        return null;
    }
    return bi;
}
}

```

## Κλάση HeaderSummarisationFrame

```

package textdatamining;

import java.awt.Graphics2D;
import java.awt.Image;
import java.awt.RenderingHints;
import java.awt.Toolkit;
import java.awt.image.BufferedImage;
import java.io.IOException;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.swing.ImageIcon;
import javax.swing.JLabel;
import org.apache.poi.xwpf.extractor.XWPFFWordExtractor;
import org.apache.poi.xwpf.usermodel.XWPFFDocument;
import org.apache.poi.xwpf.usermodel.XWPFFParagraph;
import org.apache.poi.xwpf.usermodel.XWPFFRun;

/**
 *
 * @author spiros
 */
public class HeaderSummarisationFrame extends javax.swing.JFrame {

    private String path = null;

    public HeaderSummarisationFrame() {
        Image img = Toolkit.getDefaultToolkit().createImage("Blueprint
border.jpg");
        ImageIcon im = new ImageIcon();

        BufferedImage img1 = scaleImage(400, 300, "Blueprint border.jpg");

        im.setImage(img1);
        setContentPane(new JLabel(im));
        initComponents();
    }

}

```



```

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(layout.createSequentialGroup()
        .addContainerGap()
        .addComponent(jLabel2)
        .addGap(18, 18, 18)
        .addComponent(OpenFileButton,
javax.swing.GroupLayout.PREFERRED_SIZE, 93,
javax.swing.GroupLayout.PREFERRED_SIZE))
        .addGroup(layout.createSequentialGroup()
            .addGap(58, 58, 58)
            .addComponent(jLabel4))
    .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
layout.createSequentialGroup()
        .addContainerGap()
        .addComponent(FinalHSButton,
javax.swing.GroupLayout.PREFERRED_SIZE, 200,
javax.swing.GroupLayout.PREFERRED_SIZE))
        .addGap(0, 91, Short.MAX_VALUE)
        .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
layout.createSequentialGroup()
            .addGap(0, 0, Short.MAX_VALUE)
            .addComponent(CloseButton,
javax.swing.GroupLayout.PREFERRED_SIZE, 87,
javax.swing.GroupLayout.PREFERRED_SIZE))
        .addContainerGap()
    );
layout.setVerticalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(layout.createSequentialGroup()
        .addContainerGap()
        .addComponent(jLabel4,
javax.swing.GroupLayout.PREFERRED_SIZE, 35,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addGap(35, 35, 35)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
    .addComponent(jLabel2)
    .addComponent(OpenFileButton))
    .addGap(32, 32, 32)
    .addComponent(FinalHSButton,
javax.swing.GroupLayout.DEFAULT_SIZE, 68, Short.MAX_VALUE)
    .addGap(62, 62, 62)
    .addComponent(CloseButton)
    .addContainerGap()
);

pack();
} // </editor-fold>

private void FinalHSButtonActionPerformed(java.awt.event.ActionEvent
evt) {
    if (path != null) {
        try {

```

```

        ReadAndWriteDocx open = new ReadAndWriteDocx(path);
//ανοίγει το αρχείο που δίνει ο χρήστης

        XWPFFWordExtractor extract = open.ReadDocx();
        String text = extract.getText();
        XWPFFDocument newDoc = open.WriteDocx();
        XWPFFParagraph para = newDoc.createParagraph(); //Paragraph
        XWPFFRun run = para.createRun();

        HeaderSummarisationResults file = new
HeaderSummarisationResults(text, para, run);
        file.findHeader();

        ReadAndWriteDocx open1 = new ReadAndWriteDocx(newDoc);
        open1.openNewDocx();
        open1.openFinalNewDocx();
    } catch (IOException ex) {

Logger.getLogger(MainFrameForm.class.getName()).log(Level.SEVERE, null, ex);
    }
}

private void OpenFileButtonActionPerformed(java.awt.event.ActionEvent
evt) {
    if (jFileChooser1.showOpenDialog(this) ==
jFileChooser1.APPROVE_OPTION) {
        path = jFileChooser1.getSelectedFile().getAbsolutePath();
        System.out.println(path);
    }
}

private void CloseButtonActionPerformed(java.awt.event.ActionEvent evt)
{
    this.dispose();
}

/**
 * @param args the command line arguments
 */
public void main() {
    /* Set the Nimbus look and feel */
    //<editor-fold defaultstate="collapsed" desc=" Look and feel setting
code (optional) ">
    /* If Nimbus (introduced in Java SE 6) is not available, stay with
the default look and feel.
    * For details see
http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
    */
    try {
        for (javax.swing.UIManager.LookAndFeelInfo info :
javax.swing.UIManager.getInstalledLookAndFeels()) {
            if ("Nimbus".equals(info.getName())) {

javax.swing.UIManager.setLookAndFeel(info.getClassName());
                break;
            }
        }
    } catch (ClassNotFoundException ex) {

```



```

java.util.logging.Logger.getLogger(HeaderSummarisationFrame.class.getName())
.log(java.util.logging.Level.SEVERE, null, ex);
    } catch (InstantiationException ex) {

java.util.logging.Logger.getLogger(HeaderSummarisationFrame.class.getName())
.log(java.util.logging.Level.SEVERE, null, ex);
    } catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(HeaderSummarisationFrame.class.getName())
.log(java.util.logging.Level.SEVERE, null, ex);
    } catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(HeaderSummarisationFrame.class.getName())
.log(java.util.logging.Level.SEVERE, null, ex);
    }
//</editor-fold>

/* Create and display the form */
java.awt.EventQueue.invokeLater(new Runnable() {
    public void run() {
        new HeaderSummarisationFrame().setVisible(true);
    }
});
}

// Variables declaration - do not modify
private javax.swing.JButton CloseButton;
private javax.swing.JButton FinalHSButton;
private javax.swing.JButton OpenFileButton;
private javax.swing.JFileChooser jFileChooser1;
private javax.swing.JLabel jLabel2;
private javax.swing.JLabel jLabel4;
// End of variables declaration

BufferedImage scaleImage(int WIDTH, int HEIGHT, String filename) {
    BufferedImage bi = null;

    try {
        ImageIcon ii = new ImageIcon(filename); // path to image
        bi = new BufferedImage(WIDTH, HEIGHT,
BufferedImage.TYPE_INT_RGB);
        Graphics2D g2d = (Graphics2D) bi.createGraphics();
        g2d.addRenderingHints(new
RenderingHints(RenderingHints.KEY_RENDERING,
RenderingHints.VALUE_RENDER_QUALITY));
        g2d.drawImage(ii.getImage(), 0, 0, WIDTH, HEIGHT, null);
    } catch (Exception e) {
        return null;
    }
    return bi;
}
}

```

## Κλάση HeaderSummarisationResults

```
package textdatamining;

import java.io.IOException;
import java.util.Scanner;
import org.apache.poi.xwpf.usermodel.ParagraphAlignment;
import org.apache.poi.xwpf.usermodel.UnderlinePatterns;
import org.apache.poi.xwpf.usermodel.XWPFParagraph;
import org.apache.poi.xwpf.usermodel.XWPFRun;

public class HeaderSummarisationResults {

    String number1 = "Αριθμός";
    String number2 = "ΑΡΙΘΜΟΣ";

    String dikastes = ":";

    String eishghths1 = "Εισηγητή";
    String eishghths3 = "Εισηγήτρια";

    String key = "του"; //άντρας
    String key1 = "της"; //γυναίκα
    String antieisag = "Αντεισαγγ"; /***

    String plhrej1 = "πληρεξούσιους"; //άντρας
    String plhrej2 = "πληρεξούσιο"; //άντρας
    String plhrej3 = "πληρεξούσια"; //γυναίκα

    private String text = null;
    private XWPFParagraph para = null;
    private XWPFRun run = null;

    public HeaderSummarisationResults(String text, XWPFParagraph para,
XWPFRun run) {
        this.text = text;
        this.para = para;
        this.run = run;
    }

    public void setPara(XWPFParagraph para) {
        this.para = para;
    }

    public void setRun(XWPFRun run) {
        this.run = run;
    }

    public void findHeader() throws IOException {
        String word = null;
        String word1 = null;
        String word2 = null;
        String word3 = null;

        int sumJudge = 0;
        int sumProponent = 0;
        int sumAdvocate = 0;
    }
}
```

```

int sumLawers = 0;
int sumLawer = 0;

String text1 = text.replace("-", " "); //για thn periptvsh tou
eisigith vgazw paules
Scanner scanner = new Scanner(text1);
int cnt = 0;
while (scanner.hasNext()) { //Διάβαζε λέξεις μέχρι το τέλος του
κειμένου
    para.setAlignment(ParagraphAlignment.LEFT);
    run = para.createRun();

    word3 = word2;
    word2 = word1;
    word1 = word;
    word = scanner.next();

    if (word.indexOf(number1) > -1 || word.indexOf(number2) > -1) {
        cnt++;
        if (cnt > 1) { //σε περίπτωση που βρεί και άλλα κείμενα
            sumJudge = 0;
            sumProponent = 0;
            sumAdvocate = 0;
            sumLawers = 0;
            sumLawer = 0;

            run.addBreak();
            cnt--;
        }
        FindNumber(scanner);
    }
    if (word.indexOf(dikastes) > -1 && sumJudge == 0) {
        sumJudge++;
        FindJudge(scanner);
    }
    if ((word.indexOf(eishghths1) > -1 || word.indexOf(eishghths3) >
-1) && sumProponent == 0) {
        sumJudge = 1; //an vre meta ton eisigith ton dikasth na mhn
mpei...tha einai lathos kok
        sumProponent++;
        FindProponent(word, word1, word2, word3);
    }
    if (word.indexOf(antieisag) > -1 && sumAdvocate == 0) {
        sumJudge = 1;
        sumProponent = 1;
        sumAdvocate++;
        FindAdvocate(scanner, word, word1);
    }
    if (word.indexOf(plhrej1) > -1 && sumLawers == 0 ) {
        sumJudge = 1;
        sumProponent = 1;
        sumAdvocate = 1;
        sumLawers++;
        FindLawers(scanner, word);
    }
    if ((word.indexOf(plhrej2) > -1 || word.indexOf(plhrej3) > -1)
&& (sumLawer <= 1) && (sumLawers == 0)) {
        sumJudge = 1;
        sumProponent = 1;
    }
}

```

```

        sumAdvocate = 1;
        sumLawer++;
        FindLawer(scanner, word, sumLawer);
    }
}
System.out.println();
}

private void FindNumber(Scanner scanner) {
    //εντοπίζει τον αριθμό του κειμένου
    String number = null;
    String word = scanner.next();
    if (word.indexOf("/") > -1){
        number = "Αριθμ. " + word;
    }
    else{ //se periptvsh pou o arithmos toy keimenoy den einai h
        epvmenh lejh
        word = scanner.next();
        number = "Αριθμ. " + word;
    }
    run.addBreak();
    run.setText(number);
    run.setColor("0000FF");
    run.setBold(true);
    run.setItalic(true);
    run.setUnderline(UnderlinePatterns.THICK);
    run.addBreak();
}

private void FindJudge(Scanner scanner) {
    //Προεδρεύων ο Αντιπρόεδρος
    String Judge = scanner.next();
    Judge = Judge + " " + scanner.next();
    String word = scanner.next();

    //Δήμητρα Παπαντωνοπούλου, Βιολέττα Κυτία , Θεοδώρα Γκοϊνη
    if (word.indexOf("Προεδρεύουσα") > -1 || Judge.indexOf("Δήμητρα
    Παπαντωνοπούλου") > -1
        || Judge.indexOf("Βιολέττα Κυτία") > -1 ||
    Judge.indexOf("Θεοδώρα Γκοϊνη") > -1) {
        Judge = Judge.replace(", ", " ");
        run.setText("Προεδρεύουσα η Αντιπρόεδρος " + Judge);
        run.setColor("0000FF");
        run.setBold(true);
        run.addBreak();
    } else {
        String[] result = Judge.split(" ");
        Judge = (String) result[0].subSequence(0, 1);
        Judge = Judge.concat(".");
        Judge = Judge.concat(" ");
        Judge = Judge.concat(result[1]);

        if (Judge.indexOf(',') > -1) {
            Judge = Judge.replace(", ", "ς");
        } else {
            Judge = Judge.concat("ς");
        }
        run.setText("Προεδρεύων ο Αντιπρόεδρος " + Judge);
        run.setColor("0000FF");
    }
}

```

```

        run.setBold(true);
        run.addBreak();
    }
}

private void FindProponent(String word, String word1, String word2,
String word3) {
    //Εισαγγητής/τρια
    String Proponent = null;

    if (word.indexOf(eishghths1) > -1) {

        Proponent = word2 + " " + word1;
        String[] result = Proponent.split(" ");
        Proponent = (String) result[0].subSequence(0, 1);
        Proponent = Proponent.concat(".");
        Proponent = Proponent.concat(" ");
        Proponent = Proponent.concat(result[1]);

        if (Proponent.indexOf(',') > -1) {
            Proponent = Proponent.replace(",", "ς");
        } else {
            Proponent = Proponent.concat("ς");
        }
        run.setText("Εισαγγητής ο Αρροπαγίτης " + Proponent);
        run.setColor("0000FF");
        run.setBold(true);
        //run.setText(", ");
        run.addBreak();

    } else {
        Proponent = word3 + " " + word2 + "-" + word1;
        Proponent = Proponent.replace(", ", " ");
        run.setText("Εισαγγήτρια η Αρροπαγίτης " + Proponent);
        run.setColor("0000FF");
        run.setBold(true);
        //run.setText(", ");
        run.addBreak();
    }
}

private void FindAdvocate(Scanner scanner, String word, String word1) {
    //Εισαγγελεύς
    String Advocate = null;

    if (word1.indexOf(key) > -1) {
        word = scanner.next();
        word = scanner.next();
        word = scanner.next();
        word = scanner.next();
        Advocate = word.concat(" " + scanner.next());

        String[] result = Advocate.split(" ");

        char[] Str2 = new char[2];
        result[1].getChars(result[1].length() - 2, result[1].length(),
Str2, 0);

        if (Str2[0] == 'ο' && Str2[1] == 'υ') {

```

```

        result[1] = result[1].replace("υ", "ς");
    } else {
        result[1] = result[1].concat("ς");
    }

    Advocate = (String) result[0].subSequence(0, 1);
    Advocate = Advocate.concat(".");
    Advocate = Advocate.concat(" ");
    Advocate = Advocate.concat(result[1]);
    run.setText("Εισαγγελεύς " + Advocate);
    run.setColor("0000FF");
    run.setBold(true);
    //run.setText(", ");
    run.addBreak();
}
if (word1.indexOf(key1) > -1) {
    word = scanner.next();
    word = scanner.next();
    word = scanner.next();
    word = scanner.next();
    Advocate = word.concat(scanner.next());
    Advocate = Advocate.replace("ς", " ");
    run.setText("Εισαγγελεύς " + Advocate);
    run.setColor("0000FF");
    run.setBold(true);
    //run.setText(", ");
    run.addBreak();
}
}

private void FindLawers(Scanner scanner, String word) {
    //Δικηγόροι
    String Lawers1 = null;
    String Lawers2 = null;

    word = scanner.next();
    word = scanner.next();
    word = scanner.next();
    Lawers1 = word.concat(" " + scanner.next());
    word = scanner.next();
    word = scanner.next();
    Lawers2 = word.concat(" " + scanner.next());

    String[] result = Lawers1.split(" ");
    Lawers1 = (String) result[0].subSequence(0, 1);
    Lawers1 = Lawers1.concat(".");
    Lawers1 = Lawers1.concat(" ");
    Lawers1 = Lawers1.concat(result[1]);

    if (Lawers1.indexOf(',') > -1) {
        Lawers1 = Lawers1.replace(", ", "ς");
    } else {
        Lawers1 = Lawers1.concat("ς");
    }

    result = Lawers2.split(" ");
    Lawers2 = (String) result[0].subSequence(0, 1);
    Lawers2 = Lawers2.concat(".");
}

```

```

Lawers2 = Lawers2.concat(" ");
Lawers2 = Lawers2.concat(result[1]);

if (Lawers2.indexOf(',') > -1) {
    Lawers2 = Lawers2.replace(",", "ς");
} else {
    Lawers2 = Lawers2.concat("ς");
}
run.setText("Δικηγόροι " + Lawers1 + " και " + Lawers2);
run.setColor("0000FF");
run.setBold(true);
//run.setText(", ");
run.addBreak();

}

private void FindLawer(Scanner scanner, String word, int cnt) {
    //Δικηγόρος
    String Lawer = null;

    String key3 = word;
    word = scanner.next();
    word = scanner.next();
    word = scanner.next();
    Lawer = word.concat(" " + scanner.next());
    if (key3.indexOf(plhrej2) > -1) {/**άντρας**
        String[] result = Lawer.split(" ");
        Lawer = (String) result[0].subSequence(0, 1);
        Lawer = Lawer.concat(".");
        Lawer = Lawer.concat(" ");
        Lawer = Lawer.concat(result[1]);

        if (Lawer.indexOf(',') > -1) {
            Lawer = Lawer.replace(",", "ς");
        } else {
            Lawer = Lawer.concat("ς");
        }
    } else {/**γυναίκα**
        Lawer = Lawer.replace(",", " ");
    }
    if (cnt == 1) {
        run.setText("Δικηγόρος " + Lawer);
        run.setColor("0000FF");
        run.setBold(true);
        run.addBreak();
        if (cnt == 2) {
            System.out.println(", " + Lawer);
            run.setText(", " + Lawer);
            run.setColor("0000FF");
            run.setBold(true);
            run.addBreak();
        }
    }
}
}
}

```

## Κλάση ExportRefMenu

```
package textdatamining;

import java.awt.Graphics2D;
import java.awt.Image;
import java.awt.RenderingHints;
import java.awt.Toolkit;
import java.awt.image.BufferedImage;
import java.io.IOException;
import javax.swing.ImageIcon;
import javax.swing.JFileChooser;
import javax.swing.JLabel;
import org.apache.poi.xwpf.extractor.XWPFWordExtractor;
import org.apache.poi.xwpf.usermodel.XWPFDocument;
import org.apache.poi.xwpf.usermodel.XWPFParagraph;
import org.apache.poi.xwpf.usermodel.XWPFRun;

public class ExportRefMenu extends javax.swing.JFrame {

    private String path;
    private String input;

    public ExportRefMenu() {
        Image img = Toolkit.getDefaultToolkit().createImage("Blueprint
border.jpg");
        ImageIcon im = new ImageIcon();

        BufferedImage img1 = scaleImage(541, 300, "Blueprint border.jpg");

        im.setImage(img1);
        setContentPane(new JLabel(im));

        initComponents();
        input = "";
    }

    /**
     * This method is called from within the constructor to initialize the
form.
     * WARNING: Do NOT modify this code. The content of this method is
always
     * regenerated by the Form Editor.
     */
    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated Code">
    private void initComponents() {

        jFileChooserSS = new javax.swing.JFileChooser();
        Sunt = new javax.swing.JCheckBox();
        Sdoe = new javax.swing.JCheckBox();
        KPolD = new javax.swing.JCheckBox();
        DSAPD = new javax.swing.JCheckBox();
        ESDA = new javax.swing.JCheckBox();
        PK = new javax.swing.JCheckBox();
        KPD = new javax.swing.JCheckBox();
        N = new javax.swing.JCheckBox();
    }
}
```



```

ND = new javax.swing.JCheckBox();
Dikastirioy = new javax.swing.JCheckBox();
Plhm = new javax.swing.JCheckBox();
Efet = new javax.swing.JCheckBox();
Prot = new javax.swing.JCheckBox();
ESumv = new javax.swing.JCheckBox();
SelectAll = new javax.swing.JCheckBox();
CloseButton = new javax.swing.JButton();
NextButton = new javax.swing.JButton();
AK = new javax.swing.JCheckBox();
jLabel2 = new javax.swing.JLabel();
OpenFileButton = new javax.swing.JButton();
jLabel3 = new javax.swing.JLabel();

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);

Sunt.setFont(new java.awt.Font("Trebuchet MS", 0, 12)); // NOI18N
Sunt.setText("ΣΥΝΤ.");
Sunt.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        SuntActionPerformed(evt);
    }
});

Sdoe.setFont(new java.awt.Font("Trebuchet MS", 0, 12)); // NOI18N
Sdoe.setText("ΣΔΟΕ");
Sdoe.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        SdoeActionPerformed(evt);
    }
});

KPolD.setFont(new java.awt.Font("Trebuchet MS", 0, 12)); // NOI18N
KPolD.setText("ΚΠΟΛΔ");
KPolD.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        KPolDActionPerformed(evt);
    }
});

DSAPD.setFont(new java.awt.Font("Trebuchet MS", 0, 12)); // NOI18N
DSAPD.setText("ΔΣΑΠΔ");
DSAPD.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        DSAPDActionPerformed(evt);
    }
});

ESDA.setFont(new java.awt.Font("Trebuchet MS", 0, 12)); // NOI18N
ESDA.setText("ΕΣΔΑ");
ESDA.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        ESDAActionPerformed(evt);
    }
});

PK.setFont(new java.awt.Font("Trebuchet MS", 0, 12)); // NOI18N
PK.setText("ΠΚ");
PK.addActionListener(new java.awt.event.ActionListener() {

```

```

        public void actionPerformed(java.awt.event.ActionEvent evt) {
            PKActionPerformed(evt);
        }
    });

    KPD.setFont(new java.awt.Font("Trebuchet MS", 0, 12)); // NOI18N
    KPD.setText("ΚΠΔ");
    KPD.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            KPDActionPerformed(evt);
        }
    });

    N.setFont(new java.awt.Font("Trebuchet MS", 0, 12)); // NOI18N
    N.setText("Ν.");
    N.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            NActionPerformed(evt);
        }
    });

    ND.setFont(new java.awt.Font("Trebuchet MS", 0, 12)); // NOI18N
    ND.setText("ν.δ");
    ND.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            NDActionPerformed(evt);
        }
    });

    Dikastirioy.setFont(new java.awt.Font("Trebuchet MS", 0, 12)); //
NOI18N
    Dikastirioy.setText("Δικαστήριο");
    Dikastirioy.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            DikastirioyActionPerformed(evt);
        }
    });

    Plhm.setFont(new java.awt.Font("Trebuchet MS", 0, 12)); // NOI18N
    Plhm.setText("Πλημ/κείου");
    Plhm.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            PlhmActionPerformed(evt);
        }
    });

    Efet.setFont(new java.awt.Font("Trebuchet MS", 0, 12)); // NOI18N
    Efet.setText("Εφετείου");
    Efet.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            EfetActionPerformed(evt);
        }
    });

    Prot.setFont(new java.awt.Font("Trebuchet MS", 0, 12)); // NOI18N
    Prot.setText("Πρωτοδικείου");
    Prot.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            ProtActionPerformed(evt);
        }
    });

```

```

    }
});

ESumv.setFont(new java.awt.Font("Trebuchet MS", 0, 12)); // NOI18N
ESumv.setText("Συμβόλεως");
ESumv.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        ESumvActionPerformed(evt);
    }
});

SelectAll.setFont(new java.awt.Font("Trebuchet MS", 0, 12)); //
NOI18N
SelectAll.setText("Select All");
SelectAll.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        SelectAllActionPerformed(evt);
    }
});

CloseButton.setFont(new java.awt.Font("Trebuchet MS", 0, 12)); //
NOI18N
CloseButton.setText("Close");
CloseButton.setMaximumSize(new java.awt.Dimension(95, 25));
CloseButton.setPreferredSize(new java.awt.Dimension(100, 25));
CloseButton.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        CloseButtonActionPerformed(evt);
    }
});

NextButton.setFont(new java.awt.Font("Trebuchet MS", 0, 12)); //
NOI18N
NextButton.setText("Next >>");
NextButton.setMaximumSize(new java.awt.Dimension(95, 25));
NextButton.setMinimumSize(new java.awt.Dimension(95, 25));
NextButton.setPreferredSize(new java.awt.Dimension(100, 25));
NextButton.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        NextButtonActionPerformed(evt);
    }
});

AK.setText("AK");
AK.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        AKActionPerformed(evt);
    }
});

jLabel2.setFont(new java.awt.Font("Trebuchet MS", 0, 14)); // NOI18N
jLabel2.setText("Press Open for choose file :");

OpenFileButton.setText("Open");
OpenFileButton.addActionListener(new java.awt.event.ActionListener()
{
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        OpenFileButtonActionPerformed(evt);
    }
}

```

```

    });

    jLabel13.setFont(new java.awt.Font("Trebuchet MS", 1, 18)); // NOI18N
    jLabel13.setText("Options Menu");

    javax.swing.GroupLayout layout = new
javax.swing.GroupLayout(getContentPane());
    getContentPane().setLayout(layout);
    layout.setHorizontalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup()
            .addGap(10, 10, 10)
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addGroup(layout.createSequentialGroup()
                    .addGroup(layout.createSequentialGroup()
                        .addGap(0, 0, Short.MAX_VALUE)
                        .addComponent(NextButton,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
                        .addGap(18, 18, 18)
                        .addComponent(CloseButton,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
                    .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
layout.createSequentialGroup()

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addGroup(layout.createSequentialGroup()
                    .addComponent(Sunt)
                    .addComponent(PK)
                    .addGap(44, 44, 44)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addGroup(layout.createSequentialGroup()
                    .addComponent(Sdoe)
                    .addComponent(KPD)
                    .addComponent(Efet))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)
                .addGroup(layout.createSequentialGroup()
                    .addComponent(N,
javax.swing.GroupLayout.Alignment.LEADING)
                    .addComponent(Prot,
javax.swing.GroupLayout.Alignment.LEADING)
                    .addComponent(KPold,
javax.swing.GroupLayout.Alignment.LEADING))
                .addGap(12, 12, 12)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addGroup(layout.createSequentialGroup()
                    .addComponent(DSAPD)
                    .addGroup(layout.createSequentialGroup()

```

```

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addComponent(ND)
    .addComponent(ESumv)
    .addGap(30, 30, 30)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addComponent(AK)
    .addComponent(ESDA)
    .addComponent(Dikastirioy))))
.addGroup(layout.createSequentialGroup())

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addComponent(Plhm)
    .addGroup(layout.createSequentialGroup()
        .addComponent(jLabel2)
        .addGap(18, 18, 18)
        .addComponent(OpenFileButton,
javax.swing.GroupLayout.PREFERRED_SIZE, 101,
javax.swing.GroupLayout.PREFERRED_SIZE))
        .addComponent(SelectAll)
        .addGroup(layout.createSequentialGroup()
            .addGap(136, 136, 136)
            .addComponent(jLabel3))
            .addGap(0, 0, Short.MAX_VALUE))
        .addContainerGap())
    );
layout.setVerticalGroup(

layout.createSequentialGroup()
    .addGroup(layout.createSequentialGroup()
        .addContainerGap()
        .addComponent(jLabel3,
javax.swing.GroupLayout.PREFERRED_SIZE, 31,
javax.swing.GroupLayout.PREFERRED_SIZE))
        .addGap(31, 31, 31)

.addGroup(layout.createParallelGroup(BASELINE)
    .addComponent(jLabel2)
    .addComponent(OpenFileButton))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 27,
Short.MAX_VALUE)
    .addComponent(SelectAll)
    .addGap(18, 18, 18)

.addGroup(layout.createParallelGroup(BASELINE)
    .addComponent(Sunt)
    .addComponent(Sdoe)
    .addComponent(DSAPD)
    .addComponent(ESDA)
    .addComponent(KPoLD))
    .addGap(18, 18, 18)

```

```

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
    .addComponent(PK)
    .addComponent(KPD)
    .addComponent(N)
    .addComponent(ND)
    .addComponent(Dikastirioy)
    .addGap(18, 18, 18)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
    .addComponent(Plhm)
    .addComponent(Efet)
    .addComponent(Prot)
    .addComponent(ESumv)
    .addComponent(AK)
    .addGap(40, 40, 40)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
    .addComponent(CloseButton,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
    .addComponent(NextButton,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
    .addContainerGap()
);

pack();
} // </editor-fold>

private void SuntuActionPerformed(java.awt.event.ActionEvent evt) {
    if (Suntu.isSelected()) {
        if (input.contains("a")) {
            input += "a";
        }
    } else {
        input = input.replace("a", "");
    }
}

private void SdoeActionPerformed(java.awt.event.ActionEvent evt) {
    if (Sdoe.isSelected()) {
        if (input.indexOf("b") == -1) {
            input += "b";
        }
    } else {
        input = input.replace("b", "");
    }
}

private void KPolDActionPerformed(java.awt.event.ActionEvent evt) {
    if (KPolD.isSelected()) {
        if (input.indexOf("c") == -1) {
            input += "c";
        }
    }
}

```

```

    }
} else {
    input = input.replace("c", "");
}
}

private void DSAPDActionPerformed(java.awt.event.ActionEvent evt) {
    if (DSAPD.isSelected()) {
        if (input.indexOf("e") == -1) {
            input += "e";
        }
    } else {
        input = input.replace("e", "");
    }
}

private void ESDAAActionPerformed(java.awt.event.ActionEvent evt) {
    if (ESDA.isSelected()) {
        if (input.indexOf("f") == -1) {
            input += "f";
        }
    } else {
        input = input.replace("f", "");
    }
}

private void PKActionPerformed(java.awt.event.ActionEvent evt) {
    if (PK.isSelected()) {
        if (input.indexOf("g") == -1) {
            input += "g";
        }
    } else {
        input = input.replace("g", "");
    }
}

private void KPDActionPerformed(java.awt.event.ActionEvent evt) {
    if (KPD.isSelected()) {
        if (input.indexOf("h") == -1) {
            input += "h";
        }
    } else {
        input = input.replace("h", "");
    }
}

private void NActionPerformed(java.awt.event.ActionEvent evt) {
    if (N.isSelected()) {
        if (input.indexOf("i") == -1) {
            input += "i";
        }
    } else {
        input = input.replace("i", "");
    }
}

private void NDAActionPerformed(java.awt.event.ActionEvent evt) {
    if (ND.isSelected()) {
        if (input.indexOf("j") == -1) {

```

```

        input += "j";
    }
} else {
    input = input.replace("j", "");
}
}

private void DikastirioyActionPerformed(java.awt.event.ActionEvent evt)
{
    if (Dikastirioy.isSelected()) {
        if (input.indexOf("k") == -1) {
            input += "k";
        }
    } else {
        input = input.replace("k", "");
    }
}

private void PlhmActionPerformed(java.awt.event.ActionEvent evt) {
    if (Plhm.isSelected()) {
        if (input.indexOf("l") == -1) {
            input += "l";
        }
    } else {
        input = input.replace("l", "");
    }
}

private void EfetActionPerformed(java.awt.event.ActionEvent evt) {
    if (Efet.isSelected()) {
        if (input.indexOf("m") == -1) {
            input += "m";
        }
    } else {
        input = input.replace("m", "");
    }
}

private void ProtActionPerformed(java.awt.event.ActionEvent evt) {
    if (Prot.isSelected()) {
        if (input.indexOf("n") == -1) {
            input += "n";
        }
    } else {
        input = input.replace("n", "");
    }
}

private void ESumvActionPerformed(java.awt.event.ActionEvent evt) {
    if (ESumv.isSelected()) {
        if (input.indexOf("o") == -1) {
            input += "o";
        }
    } else {
        input = input.replace("o", "");
    }
}

private void SelectAllActionPerformed(java.awt.event.ActionEvent evt) {

```



```

    if (SelectAll.isSelected()) {
        input = "p";
    } else {
        input = "";
    }

    Sunt.setSelected(SelectAll.isSelected());
    Sdoe.setSelected(SelectAll.isSelected());
    KPolD.setSelected(SelectAll.isSelected());
    DSAPD.setSelected(SelectAll.isSelected());
    ESDA.setSelected(SelectAll.isSelected());
    PK.setSelected(SelectAll.isSelected());
    KPD.setSelected(SelectAll.isSelected());
    N.setSelected(SelectAll.isSelected());
    ND.setSelected(SelectAll.isSelected());
    Dikastirioy.setSelected(SelectAll.isSelected());
    Plhm.setSelected(SelectAll.isSelected());
    Efet.setSelected(SelectAll.isSelected());
    Prot.setSelected(SelectAll.isSelected());
    ESumv.setSelected(SelectAll.isSelected());
    AK.setSelected(SelectAll.isSelected());
}

private void CloseButtonActionPerformed(java.awt.event.ActionEvent evt)
{
    this.dispose();
}

private void NextButtonActionPerformed(java.awt.event.ActionEvent evt) {
    if (path != null) {
        ReadAndWriteDocx open = new ReadAndWriteDocx(path);

        XWPFFWordExtractor extract = null;
        try {
            extract = open.ReadDocx();
        } catch (IOException ex) {
        }
        String text = extract.getText();
        XWPFFDocument newDoc = open.WriteDocx();
        XWPFFParagraph para = newDoc.createParagraph();
        XWPFFRun run = para.createRun();
        try {
            ExportReferences exportAll = new ExportReferences(text,
para, run, input);
            exportAll.Allreferences();
            ReadAndWriteDocx open1 = new ReadAndWriteDocx(newDoc);
            open1.openNewDocx();
            ExportRefResults frame = new ExportRefResults();
            frame.setDoc(newDoc);
            frame.setList(exportAll.getList());
            frame.setSumALL(exportAll.getSumALL());
            frame.setHeight(exportAll.getHeight());
            frame.main();
            this.dispose();//gia na kllhsei to frame
        } catch (Exception ex) {
            System.out.println("Exception Error");
        }
    }
}

```

```

}

private void AKActionPerformed(java.awt.event.ActionEvent evt) {
    if (AK.isSelected()) {
        if (input.indexOf("d") == -1) {
            input += "d";
        }
    } else {
        input = input.replace("d", "");
    }
}

private void OpenFileButtonActionPerformed(java.awt.event.ActionEvent
evt) {
    if (jFileChooserSS.showOpenDialog(this) ==
JFileChooser.APPROVE_OPTION) {
        path = jFileChooserSS.getSelectedFile().getAbsolutePath();
        System.out.println(path);
    }
}

public void Main() {
    /* Set the Nimbus look and feel */
    //<editor-fold defaultstate="collapsed" desc=" Look and feel setting
code (optional) ">
    /* If Nimbus (introduced in Java SE 6) is not available, stay with
the default look and feel.
    * For details see
http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
    */
    try {
        for (javax.swing.UIManager.LookAndFeelInfo info :
javax.swing.UIManager.getInstalledLookAndFeels()) {
            if ("Nimbus".equals(info.getName())) {
                javax.swing.UIManager.setLookAndFeel(info.getClassName());
                break;
            }
        }
    } catch (ClassNotFoundException ex) {
        java.util.logging.Logger.getLogger(ExportRefMenu.class.getName()).log(java.u
til.logging.Level.SEVERE, null, ex);
    } catch (InstantiationException ex) {
        java.util.logging.Logger.getLogger(ExportRefMenu.class.getName()).log(java.u
til.logging.Level.SEVERE, null, ex);
    } catch (IllegalAccessException ex) {
        java.util.logging.Logger.getLogger(ExportRefMenu.class.getName()).log(java.u
til.logging.Level.SEVERE, null, ex);
    } catch (javax.swing.UnsupportedLookAndFeelException ex) {
        java.util.logging.Logger.getLogger(ExportRefMenu.class.getName()).log(java.u
til.logging.Level.SEVERE, null, ex);
    }
}
//</editor-fold>
//</editor-fold>
//</editor-fold>

```

```

//</editor-fold>

/* Create and display the form */
java.awt.EventQueue.invokeLater(new Runnable() {
    public void run() {
        new ExportRefMenu().setVisible(true);
    }
});
}

// Variables declaration - do not modify
private javax.swing.JButton CloseButton;
private javax.swing.JCheckBox DSAPD;
private javax.swing.JCheckBox Dikastirioy;
private javax.swing.JCheckBox ESDA;
private javax.swing.JCheckBox ESumv;
private javax.swing.JCheckBox Efet;
private javax.swing.JCheckBox KPD;
private javax.swing.JCheckBox KPolD;
private javax.swing.JCheckBox N;
private javax.swing.JCheckBox ND;
private javax.swing.JButton NextButton;
private javax.swing.JButton OpenFileButton;
private javax.swing.JCheckBox PK;
private javax.swing.JCheckBox Plhm;
private javax.swing.JCheckBox Prot;
private javax.swing.JCheckBox Sdoe;
private javax.swing.JCheckBox SelectAll;
private javax.swing.JCheckBox Sunt;
private javax.swing.JFileChooser jFileChooserSS;
private javax.swing.JLabel jLabel2;
private javax.swing.JLabel jLabel3;
private javax.swing.JCheckBox AK;
// End of variables declaration

BufferedImage scaleImage(int WIDTH, int HEIGHT, String filename) {
    BufferedImage bi = null;

    try {
        ImageIcon ii = new ImageIcon(filename); // path to image
        bi = new BufferedImage(WIDTH, HEIGHT,
BufferedImage.TYPE_INT_RGB);
        Graphics2D g2d = (Graphics2D) bi.createGraphics();
        g2d.addRenderingHints(new
RenderingHints(RenderingHints.KEY_RENDERING,
RenderingHints.VALUE_RENDER_QUALITY));
        g2d.drawImage(ii.getImage(), 0, 0, WIDTH, HEIGHT, null);
    } catch (Exception e) {
        return null;
    }
    return bi;
}
}

```

## Κλάση ExportReferences

```
package textdatamining;

import java.io.IOException;
import java.util.ArrayList;
import java.util.Scanner;
import java.util.regex.Matcher;
import java.util.regex.Pattern;
import org.apache.poi.xwpf.usermodel.XWPFParagraph;
import org.apache.poi.xwpf.usermodel.XWPFRun;
import org.apache.poi.xwpf.usermodel.ParagraphAlignment;
import org.apache.poi.xwpf.usermodel.UnderlinePatterns;
import org.jfree.ui.RefineryUtilities;

public class ExportReferences {

    final static int GUARD = 15;
    private String text;
    private XWPFParagraph para;
    private XWPFRun run;
    private String In;

    public ExportReferences(String text, XWPFParagraph para, XWPFRun run,
String In) {
        this.text = text;
        this.para = para;
        this.run = run;
        this.In = In;
    }

    float sumSuntforChart = 0;
    float sumSdoeforChart = 0;
    float sumDSAPDforChart = 0;
    float sumDikforChart = 0;
    float sumEsdaforChart = 0;
    float sumEfetforChart = 0;
    float sumKpdforChart = 0;
    float sumKPolDforChart = 0;
    float sumNdotforChart = 0;
    float sumNdotDforChart = 0;
    float sumPkforChart = 0;
    float sumPlhmforChart = 0;
    float sumProtforChart = 0;
    float sumESumvforChart = 0;
    float sumAkforChart = 0;

    float sumALL = 0;
    float sumSunt = 0;
    float sumSdoe = 0;
    float sumDSAPD = 0;
    float sumDik = 0;
    float sumEsda = 0;
    float sumEfet = 0;
    float sumKpd = 0;
```

```

float sumKPolD = 0;
float sumNdot = 0;
float sumNdotD = 0;
float sumPk = 0;
float sumPlhm = 0;
float sumProt = 0;
float sumESumv = 0;
float sumAk = 0;

ArrayList<Float> freqList = new ArrayList<>();//h lista pou peri;exei
thn suxnothta emfanishs gia kathe apofash
ArrayList<Float> ratesList = new ArrayList<>();//h lista pou periexei to
sunoliko athroisma gia olla ta pososta
int height = 600;//gia grafikh

public void Allreferences() throws IOException, Exception {
    AllFinalsVariables finals = new AllFinalsVariables();

    String number1 = "Αριθμ";
    String number2 = "ΑΡΙΘΜ";
    String reference = null;
    String previousRef = null;
    String word1 = null;

    CreateVectorExcel excel = new CreateVectorExcel();
    excel.OpenExcel();

    Scanner scanner = new Scanner(text);
    para.setAlignment(ParagraphAlignment.BOTH);//gia morfopoihsh
    int textcnt = 0;
    while (scanner.hasNext()) { //Διάβαζε λέξεις μέχρι το τέλος του
κειμένου
        run = para.createRun();
        run.setText(" ");

        word1 = scanner.next();
        reference = word1;

        if (word1.indexOf(finals.ar1) > -1
            || word1.indexOf(finals.ar2) > -1
            || word1.indexOf(finals.parKey) > -1
            || word1.indexOf(finals.arthro) > -1
            || word1.indexOf(finals.bslashKey) > -1) {

            int cnt = 0;
            while (word1.indexOf(number1) <= -1 &&
word1.indexOf(number2) <= -1
                && word1.indexOf(finals.sunt) <= -1
                && word1.indexOf(finals.sdoe) <= -1
                && word1.indexOf(finals.idioukwдика1) <= -1
                && word1.indexOf(finals.idioukwδικα2) <= -1
                && word1.indexOf(finals.idiounomou1) <= -1
                && word1.indexOf(finals.idiounomou2) <= -1
                && word1.indexOf(finals.ankwδικα) <= -1
                && word1.indexOf(finals.daspd) <= -1
                && word1.indexOf(finals.kPolD) <= -1
                && word1.indexOf(finals.esda1) <= -1
                && word1.indexOf(finals.esda2) <= -1
                && word1.indexOf(finals.pk) <= -1

```

```

        && word1.indexOf(finals.pk1) <= -1
        && word1.indexOf(finals.pk2) <= -1
        && word1.indexOf(finals.kpd) <= -1
        && word1.indexOf(finals.kpd1) <= -1
        && word1.indexOf(finals.kpd2) <= -1
        && word1.indexOf(finals.kpd3) <= -1
        && word1.indexOf(finals.kpd4) <= -1
        && word1.indexOf(finals.n) <= -1
        && word1.indexOf(finals.n1) <= -1
        && word1.indexOf(finals.ndotd) <= -1
        && word1.indexOf(finals.ndotd1) <= -1
        && word1.indexOf(finals.dikasthriou) <= -1
        && word1.indexOf(finals.plhmmleiodikeiou) <= -1
        && word1.indexOf(finals.efeteiou) <= -1
        && word1.indexOf(finals.protodikeiou) <= -1
        && word1.indexOf(finals.eurvpaikhsSumvasews) <= -1
        && word1.indexOf(finals.Ak1) <= -1
        && word1.indexOf(finals.Ak2) <= -1) {

word1 = scanner.next();
reference = reference + " " + word1;
cnt++;

    if (cnt > GUARD) {
        run.setText(reference);
        run.setText(" ");
        break;
    }
    if (reference.indexOf(number1) > -1 ||
reference.indexOf(number2) > -1) {
        break;
    }

        if ((reference.indexOf(finals.idiokwdika1) > -1 ||
reference.indexOf(finals.idiokwdika2) > -1)
            || (reference.indexOf(finals.anwkwidika) > -1)
|| (reference.indexOf(finals.Ak1) > -1)
            || (reference.indexOf(finals.Ak2) > -1))
            && (In.indexOf("p") > -1)) {

            cnt = 50;
            sumALL++;
            Find_IKOD_Ref(previousRef, reference);
        }
        if ((reference.indexOf(finals.idiounomou1) > -1 ||
reference.indexOf(finals.idiounomou2) > -1)
            && (In.indexOf("p") > -1)) {
            cnt = 50;
            sumALL++;
            Find_INOMOY_Ref(scanner, previousRef, word1,
reference);
        }

//*****
        if ((reference.indexOf(finals.esda1) > -1 ||
reference.indexOf(finals.esda2) > -1)
            && (In.indexOf("f") > -1 || (In.indexOf("p") >
-1))) {

```

```

        cnt = 50;
        sumALL++;
        sumEsda++;
        sumEsdaforChart++;
        FormattingType1(reference, finals.DarkCyan);
        previousRef = reference;
    }

//*****
    if ((reference.indexOf(finals.pk) > -1 ||
reference.indexOf(finals.pk1) > -1 || reference.indexOf(finals.pk2) > -1)
        && (In.indexOf("g") > -1 || In.indexOf("p") >
-1))) {

        cnt = 50;
        sumALL++;
        sumPk++;
        sumPkforChart++;
        FormattingType1(reference, finals.BlueViolet);
        previousRef = reference;
    }

//*****
    if ((reference.indexOf(finals.Ak1) > -1 ||
reference.indexOf(finals.Ak2) > -1)
        && (In.indexOf("d") > -1 || In.indexOf("p") >
-1)) {

        cnt = 50;
        sumALL++;
        sumAk++;
        sumAkforChart++;
        FormattingType1(reference, finals.SaddleBrown);
        previousRef = reference;
    }

//*****
    if ((reference.indexOf(finals.dikasthriou) > -1)
        && (In.indexOf("k") > -1 || In.indexOf("p") >
-1)) {

        cnt = 50;
        sumALL++;
        sumDik++;
        sumDikforChart++;
        word1 = scanner.next();
        reference = reference + " " + word1;
        FormattingType2(reference, finals.DarkOlive);
        previousRef = reference;
    }

//*****
    if ((reference.indexOf(finals.plhmmleiodikeiou) > -1)
        && (In.indexOf("l") > -1 || In.indexOf("p") >
-1)) {

        cnt = 50;

```

```

        sumALL++;
        sumPlhm++;
        sumPlhmforChart++;
        word1 = scanner.next();
        reference = reference + " " + word1;
        FormattingType2(reference, finals.DarkOrange);
        previousRef = reference;
    }

//*****
    if ((reference.indexOf(finals.efeteiou) > -1)
        && (In.indexOf("m") > -1 || In.indexOf("p") >
-1)) {

        cnt = 50;
        sumALL++;
        sumEfet++;
        sumEfetforChart++;
        word1 = scanner.next();
        reference = reference + " " + word1;
        FormattingType2(reference, finals.Fuchia);
        previousRef = reference;
    }

//*****
    if ((reference.indexOf(finals.protodikeiou) > -1)
        && (In.indexOf("n") > -1 || In.indexOf("p") >
-1)) {

        cnt = 50;
        sumALL++;
        sumProt++;
        sumProtforChart++;
        word1 = scanner.next();
        reference = reference + " " + word1;
        FormattingType2(reference, finals.GoldenRod);
        previousRef = reference;
    }

//*****
    if ((reference.indexOf(finals.kpd) > -1 ||
reference.indexOf(finals.kpd1) > -1
        || reference.indexOf(finals.kpd2) > -1 ||
reference.indexOf(finals.kpd3) > -1 || reference.indexOf(finals.kpd4) > -1)
        && (In.indexOf("h") > -1 || In.indexOf("p") >
-1)) {

        cnt = 50;
        sumALL++;
        sumKpd++;
        sumKpdforChart++;
        FormattingType1(reference, finals.Red);
        previousRef = reference;
    }
}

```



```

//*****
    if ((reference.indexOf(finals.sunt) > -1)
        && (In.contains("a") || In.indexOf("p") > -1))
{
    cnt = 50;
    sumALL++;
    sumSunt++;
    sumSuntforChart++;
    FormattingType1(reference, finals.IndianRed);
    previousRef = reference;
}

//*****
    if ((reference.indexOf(finals.kPold) > -1)
        && (In.indexOf("c") > -1 || In.indexOf("p") >
-1)) {
    cnt = 50;
    sumALL++;
    sumKPold++;
    sumKPoldforChart++;
    FormattingType1(reference, finals.LimeGreen);
    previousRef = reference;
}

//*****
    if ((reference.indexOf(finals.daspd) > -1)
        && (In.indexOf("e") > -1 || In.indexOf("p") >
-1)) {
    cnt = 50;
    sumALL++;
    sumDSAPD++;
    sumDSAPDforChart++;
    FormattingType1(reference, finals.Maroon);
    previousRef = reference;
}

//*****
    if ((reference.indexOf(finals.n) > -1 ||
reference.indexOf(finals.n1) > -1)
        && (In.indexOf("i") > -1 || In.indexOf("p") >
-1)) {

        if (reference.indexOf(finals.ndotd) == -1 &&
reference.indexOf(finals.ndotd1) == -1) {
            cnt = 50;
            word1 = scanner.next();
            Pattern p =
Pattern.compile("(1|2|3|4|5|6|7|8|9|0)"); //an meta to N. arithmos ok aliws
oxi

            Matcher m = p.matcher(word1);
            if (m.find()) {
                sumNdot++;
                sumALL++;
                sumNdotforChart++;
                reference = reference + " " + word1;

```

```

        FormattingType2(reference,
finals.SteelBlue);
        previousRef = reference;
    }
}

//*****
    if ((reference.indexOf(finals.ndotd) > -1 ||
reference.indexOf(finals.ndotd1) > -1)
        && (In.indexOf("j") > -1 || In.indexOf("p") >
-1)) {

        cnt = 50;
        sumALL++;
        sumNdotD++;
        sumNdotDforChart++;
        word1 = scanner.next();
        reference = reference + " " + word1;
        FormattingType2(reference, finals.Olive);
        previousRef = reference;

    }

//*****
    if ((reference.indexOf(finals.sdoe) > -1)
        && (In.indexOf("b") > -1 || In.indexOf("p") >
-1)) {

        cnt = 50;
        sumALL++;
        sumSdoe++;
        sumSdoeforChart++;
        FormattingType1(reference, finals.Tomato);
        previousRef = reference;

    }

//*****
    if ((reference.indexOf(finals.eurvpaikhSumvasews) > -1)
        && (In.indexOf("o") > -1 || In.indexOf("p") >
-1)) {

        cnt = 50;
        sumALL++;
        sumESumv++;
        sumESumvforChart++;
        FormattingType1(reference, finals.SaddleBrown);
        previousRef = reference;

    }
}
} else if ((word1.indexOf(finals.n) > -1 ||
word1.indexOf(finals.n1) > -1)
        && (In.indexOf("i") > -1 || In.indexOf("p") > -1)) {

    if (word1.indexOf(finals.ndotd) == -1 &&
word1.indexOf(finals.ndotd1) == -1) {
        word1 = scanner.next();
        Pattern p = Pattern.compile("(1|2|3|4|5|6|7|8|9|10)");
        Matcher m = p.matcher(word1);

```

```

        if (m.find()) {
            sumALL++;
            sumNdot++;
            sumNdotforChart++;
            reference = reference + " " + word1;
            FormattingType1(reference, finals.SteelBlue);
            previousRef = reference;
        }
    }
} else if ((word1.indexOf(finals.ndotd) > -1 ||
word1.indexOf(finals.ndotd1) > -1)
    && (In.indexOf("j") > -1 || In.indexOf("p") > -1)) {
    sumALL++;
    sumNdotD++;
    sumNdotDforChart++;
    word1 = scanner.next();
    FormattingType1(reference, finals.Olive);
    previousRef = reference;
} else {
}
if (word1.indexOf(number1) > -1 || word1.indexOf(number2) > -1)
{
    //System.out.println("\n" + textcnt);
    textcnt++;
    if (textcnt > 1) {
        freqList = CreateTextVector();
        excel.setList(freqList);
        excel.setLinePos(textcnt);
        excel.Excel();
    }
    sumSunt = 0;
    sumSdoe = 0;
    sumDSAPD = 0;
    sumDik = 0;
    sumEsda = 0;
    sumEfet = 0;
    sumKpd = 0;
    sumKPolD = 0;
    sumNdot = 0;
    sumNdotD = 0;
    sumPk = 0;
    sumPlhm = 0;
    sumProt = 0;
    sumESumv = 0;
    sumAk = 0;
}

if (reference.indexOf(finals.sunt) > -1
    || reference.indexOf(finals.sdoe) > -1
    || reference.indexOf(finals.kPolD) > -1
    || reference.indexOf(finals.daspd) > -1
    || reference.indexOf(finals.idioukwдика1) > -1
    || reference.indexOf(finals.idioukwдика2) > -1
    || reference.indexOf(finals.idiounomou1) > -1
    || reference.indexOf(finals.idiounomou2) > -1
    || reference.indexOf(finals.ankwдика) > -1
    || reference.indexOf(finals.esda1) > -1
    || reference.indexOf(finals.esda2) > -1
    || reference.indexOf(finals.pk) > -1

```

```

|| reference.indexOf(finals.pk1) > -1
|| reference.indexOf(finals.pk2) > -1
|| reference.indexOf(finals.kpd) > -1
|| reference.indexOf(finals.kpd1) > -1
|| reference.indexOf(finals.kpd2) > -1
|| reference.indexOf(finals.kpd3) > -1
|| reference.indexOf(finals.kpd4) > -1
|| reference.indexOf(finals.dikasthriou) > -1
|| reference.indexOf(finals.plhmmleiodikeiou) > -1
|| reference.indexOf(finals.efeteiou) > -1
|| reference.indexOf(finals.n) > -1
|| reference.indexOf(finals.n1) > -1
|| reference.indexOf(finals.ndotd) > -1
|| reference.indexOf(finals.ndotd1) > -1
|| reference.indexOf(finals.protodikeiou) > -1
|| reference.indexOf(finals.eurvpaikhsSumvasews) > -1
|| reference.indexOf(finals.Ak1) > -1
|| reference.indexOf(finals.Ak2) > -1) {

    if (scanner.hasNext() == false) { //An h teleutea leksh eina
anafora stamata
        break;
    }

    word1 = scanner.next();
}
run.setText(word1);
run.setText(" ");
}

textcnt++;
freqList = CreateTextVector();
excel.setList(freqList);
excel.setLinePos(textcnt);
excel.Excel();
excel.CloseExcel();
//excel.openFinalExcel();//Anoigei automata to excel

//*****
//Gia grafikh AWT Based Application
ratesList = CreateRateList();

if (sumALL != 0) { //ama den exei kamia anafora mhn ftiakseis
grafikh
    JFreeChartBar chart = new JFreeChartBar("ChartBar", "Ποσοστό
Εμφάνιση", ratesList, sumALL, height);
    chart.pack();
    RefineryUtilities.centerFrameOnScreen(chart);
    chart.setVisible(true);
}

//*****
}

public ArrayList<Float> getList() {
    return freqList;
}
}

```

```

public float getSumALL() {
    return sumALL;
}

public int getHeight() {
    return height;
}

private ArrayList<Float> CreateTextVector() {
    ArrayList<Float> freqList = new ArrayList<>();
    System.out.println("\n" + sumSunt + "," + sumSdoe + "," + sumKPolD +
        "," + sumDSAPD + "," + sumEsda
            + "," + sumPk + "," + sumKpd + "," + sumNdot + "," +
sumNdotD + "," + sumDik + ","
            + sumPlhm + "," + sumEfet + "," + sumProt + "," + sumESumv
+ "," + sumAk + "\n");
    freqList.add(sumSunt);
    freqList.add(sumSdoe);
    freqList.add(sumKPolD);
    freqList.add(sumDSAPD);
    freqList.add(sumEsda);
    freqList.add(sumPk);
    freqList.add(sumKpd);
    freqList.add(sumNdot);
    freqList.add(sumNdotD);
    freqList.add(sumDik);
    freqList.add(sumPlhm);
    freqList.add(sumEfet);
    freqList.add(sumProt);
    freqList.add(sumESumv);
    freqList.add(sumAk);

    return freqList;
}

private ArrayList<Float> CreateRateList() {
    ArrayList<Float> ratesList = new ArrayList<>();
    ratesList.add(sumSuntforChart);
    ratesList.add(sumSdoeforChart);
    ratesList.add(sumKPolDforChart);
    ratesList.add(sumDSAPDforChart);
    ratesList.add(sumEsdaforChart);
    ratesList.add(sumPkforChart);
    ratesList.add(sumKpdforChart);
    ratesList.add(sumNdotforChart);
    ratesList.add(sumNdotDforChart);
    ratesList.add(sumDikforChart);
    ratesList.add(sumPlhmforChart);
    ratesList.add(sumEfetforChart);
    ratesList.add(sumProtforChart);
    ratesList.add(sumESumvforChart);
    ratesList.add(sumAkforChart);

    return ratesList;
}

private void FormattingType1(String reference, String colour) {
    run.setText(reference + " ");
    run.setColor(colour);
}

```

```

        run.setBold(true);
        run.setItalic(true);
        run.setUnderline(UnderlinePatterns.THICK);
        System.out.println(reference);
    }

    private void FormattingType2(String reference, String colour) {
        run.setColor(colour);
        run.setBold(true);
        run.setItalic(true);
        run.setUnderline(UnderlinePatterns.THICK);
        System.out.println(reference);
    }

    private void Find_IKOD_Ref(String previousRef, String reference) {
        AllFinalsVariables finals = new AllFinalsVariables();

        if (previousRef.indexOf(finals.kPold) > -1) {
            sumKPolD++;
            sumKPolDforChart++;
            FormattingType2(reference, finals.LimeGreen);
        } else if ((previousRef.indexOf(finals.Ak1) > -1 ||
previousRef.indexOf(finals.Ak2) > -1)) {
            sumAk++;
            sumAkforChart++;
            FormattingType2(reference, finals.SaddleBrown);
        } else if ((previousRef.indexOf(finals.pk) > -1 ||
previousRef.indexOf(finals.pk1) > -1 || previousRef.indexOf(finals.pk2) > -
1)) {
            sumPk++;
            sumPkforChart++;
            FormattingType2(reference, finals.BlueViolet);
        } else if ((previousRef.indexOf(finals.kpd) > -1 ||
previousRef.indexOf(finals.kpd1) > -1
            || previousRef.indexOf(finals.kpd2) > -1 ||
previousRef.indexOf(finals.kpd3) > -1 || previousRef.indexOf(finals.kpd4) >
-1)) {
            sumKpd++;
            sumKpdforChart++;
            FormattingType2(reference, finals.Red);
        } else {
            run.setText(reference);
        }
    }

    private void Find_INOMOY_Ref(Scanner scanner, String previousRef, String
word1, String reference) {
        AllFinalsVariables finals = new AllFinalsVariables();

        if (previousRef.indexOf(finals.ndotd) > -1 ||
previousRef.indexOf(finals.ndotd1) > -1) {
            sumNdotD++;
            sumNdotDforChart++;
            word1 = scanner.next();
            FormattingType1(reference, finals.Olive);
        } else if ((word1.indexOf(finals.n) > -1 || word1.indexOf(finals.n1)
> -1)) {

```

```

        if (word1.indexOf(finals.ndotd) == -1 &&
word1.indexOf(finals.ndotd1) == -1) {
            word1 = scanner.next();
            Pattern p = Pattern.compile("(1|2|3|4|5|6|7|8|9|10)");
            Matcher m = p.matcher(word1);
            if (m.find()) {
                sumNdot++;
                sumNdotforChart++;
                reference = reference + " " + word1;
                FormattingType1(reference, finals.SteelBlue);
            }
        }
    } else {
        run.setText(reference);
    }
}

//*****
}

```

## Κλάση ExportRefResults

```

package textdatamining;

import java.awt.Graphics2D;
import java.awt.Image;
import java.awt.RenderingHints;
import java.awt.Toolkit;
import java.awt.image.BufferedImage;
import java.util.ArrayList;
import javax.swing.ImageIcon;
import javax.swing.JLabel;
import org.apache.poi.xwpf.usermodel.XWPFDocument;

/**
 *
 * @author spiros
 */
public class ExportRefResults extends javax.swing.JFrame {

    private ArrayList<Float> freqList;
    private float sumALL;
    private int height;
    private XWPFDocument newDoc;

    public ExportRefResults() {
        Image img = Toolkit.getDefaultToolkit().createImage("Blueprint
border.jpg");
        ImageIcon im = new ImageIcon();

        BufferedImage img1 = scaleImage(541, 300, "Blueprint border.jpg");

        im.setImage(img1);
        setContentPane(new JLabel(im));
    }
}

```

```

        initComponents();
        this.newDoc = newDoc;
    }
    public void setDoc(XWPFDocument newDoc){
        this.newDoc = newDoc;
    }

    public void setList(ArrayList<Float> freqList) {
        this.freqList = freqList;
    }

    public void setSumALL(float sumALL) {
        this.sumALL = sumALL;
    }

    public void setHeight(int height){
        this.height = height;
    }

    /**
     * This method is called from within the constructor to initialize the
    form.
     * WARNING: Do NOT modify this code. The content of this method is
    always
     * regenerated by the Form Editor.
     */
    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated Code">
    private void initComponents() {

        NewTextButton = new javax.swing.JButton();
        VectorsTableButton = new javax.swing.JButton();
        CloseButton = new javax.swing.JButton();
        jLabel1 = new javax.swing.JLabel();

        setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);

        NewTextButton.setFont(new java.awt.Font("Trebuchet MS", 0, 14)); //
    NOI18N
        NewTextButton.setText("New Text");
        NewTextButton.setPreferredSize(new java.awt.Dimension(100, 25));
        NewTextButton.addActionListener(new java.awt.event.ActionListener()
    {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            NewTextButtonActionPerformed(evt);
        }
    });

        VectorsTableButton.setFont(new java.awt.Font("Trebuchet MS", 0,
    14)); // NOI18N
        VectorsTableButton.setText("Vectors Table");
        VectorsTableButton.setPreferredSize(new java.awt.Dimension(100,
    25));
        VectorsTableButton.addActionListener(new
    java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            VectorsTableButtonActionPerformed(evt);
        }
    });
    }

```



```

CloseButton.setFont(new java.awt.Font("Trebuchet MS", 0, 12)); //
NOI18N
CloseButton.setText("Close");
CloseButton.setPreferredSize(new java.awt.Dimension(100, 25));
CloseButton.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        CloseButtonActionPerformed(evt);
    }
});

jLabel1.setFont(new java.awt.Font("Trebuchet MS", 1, 18)); // NOI18N
jLabel1.setText("Results");

javax.swing.GroupLayout layout = new
javax.swing.GroupLayout(getContentPane());
getContentPane().setLayout(layout);
layout.setHorizontalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
layout.createSequentialGroup()
    .addGap(0, 147, Short.MAX_VALUE)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING
ING)
    .addComponent(VectorsTableButton,
javax.swing.GroupLayout.PREFERRED_SIZE, 246,
javax.swing.GroupLayout.PREFERRED_SIZE)
    .addComponent(NewTextButton,
javax.swing.GroupLayout.PREFERRED_SIZE, 246,
javax.swing.GroupLayout.PREFERRED_SIZE))
    .addGap(148, 148, 148))
    .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
layout.createSequentialGroup()
    .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING
NG)
    .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
layout.createSequentialGroup()
    .addComponent(CloseButton,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
    .addContainerGap())
    .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
layout.createSequentialGroup()
    .addComponent(jLabel1)
    .addGap(236, 236, 236)))
);
layout.setVerticalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(layout.createSequentialGroup()
    .addContainerGap()

```

```

        .addComponent(jLabel1,
javax.swing.GroupLayout.PREFERRED_SIZE, 31,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addGap(18, 18, 18)
        .addComponent(NewTextButton,
javax.swing.GroupLayout.PREFERRED_SIZE, 55,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addGap(38, 38, 38)
        .addComponent(VectorsTableButton,
javax.swing.GroupLayout.PREFERRED_SIZE, 55,
javax.swing.GroupLayout.PREFERRED_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 56,
Short.MAX_VALUE)
        .addComponent(CloseButton,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addContainerGap()
    );

    pack();
} // </editor-fold>

private void NewTextButtonActionPerformed(java.awt.event.ActionEvent
evt) {
    ReadAndWriteDocx open = new ReadAndWriteDocx(newDoc);
    open.openFinalNewDocx();
}

private void
VectorsTableButtonActionPerformed(java.awt.event.ActionEvent evt) {
    CreateVectorExcel excel = new CreateVectorExcel();
    excel.openFinalExcel();
}

private void CloseButtonActionPerformed(java.awt.event.ActionEvent evt)
{
    this.dispose();
}

/**
 * @param args the command line arguments
 */
public void main() {
    /* Set the Nimbus look and feel */
    //<editor-fold defaultstate="collapsed" desc=" Look and feel setting
code (optional) ">
    /* If Nimbus (introduced in Java SE 6) is not available, stay with
the default look and feel.
    * For details see
http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
    */
    try {
        for (javax.swing.UIManager.LookAndFeelInfo info :
javax.swing.UIManager.getInstalledLookAndFeels()) {
            if ("Nimbus".equals(info.getName())) {
                javax.swing.UIManager.setLookAndFeel(info.getClassName());
            }
        }
    } catch (ClassNotFoundException ex) {
        java.util.logging.Logger.getLogger(Main.class).log(Level.SEVERE,
null, ex);
    } catch (InstantiationException ex) {
        java.util.logging.Logger.getLogger(Main.class).log(Level.SEVERE,
null, ex);
    } catch (IllegalAccessException ex) {
        java.util.logging.Logger.getLogger(Main.class).log(Level.SEVERE,
null, ex);
    }
}

```

```

        break;
    }
}
} catch (ClassNotFoundException ex) {

java.util.logging.Logger.getLogger(ExportRefResults.class.getName()).log(java.
a.util.logging.Level.SEVERE, null, ex);
} catch (InstantiationException ex) {

java.util.logging.Logger.getLogger(ExportRefResults.class.getName()).log(java.
a.util.logging.Level.SEVERE, null, ex);
} catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(ExportRefResults.class.getName()).log(java.
a.util.logging.Level.SEVERE, null, ex);
} catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(ExportRefResults.class.getName()).log(java.
a.util.logging.Level.SEVERE, null, ex);
}
//</editor-fold>
//</editor-fold>

/* Create and display the form */
java.awt.EventQueue.invokeLater(new Runnable() {
    public void run() {
        new ExportRefResults().setVisible(true);
    }
});
}

// Variables declaration - do not modify
private javax.swing.JButton CloseButton;
private javax.swing.JButton NewTextButton;
private javax.swing.JButton VectorsTableButton;
private javax.swing.JLabel jLabell;
// End of variables declaration

BufferedImage scaleImage(int WIDTH, int HEIGHT, String filename) {
    BufferedImage bi = null;

    try {
        ImageIcon ii = new ImageIcon(filename); // path to image
        bi = new BufferedImage(WIDTH, HEIGHT,
BufferedImage.TYPE_INT_RGB);
        Graphics2D g2d = (Graphics2D) bi.createGraphics();
        g2d.addRenderingHints(new
RenderingHints(RenderingHints.KEY_RENDERING,
RenderingHints.VALUE_RENDER_QUALITY));
        g2d.drawImage(ii.getImage(), 0, 0, WIDTH, HEIGHT, null);
    } catch (Exception e) {
        return null;
    }
    return bi;
}
}

```

## Κλάση ReadAndWriteDocx

```
package textdatamining;

import java.awt.Desktop;
import java.io.File;
import java.io.FileInputStream;
import java.io.FileNotFoundException;
import java.io.FileOutputStream;
import java.io.IOException;
import org.apache.poi.xwpf.extractor.XWPFFWordExtractor;
import org.apache.poi.xwpf.usermodel.XWPFDocument;

public class ReadAndWriteDocx {
    private String In = null;
    private XWPFDocument newDoc;

    public ReadAndWriteDocx(XWPFDocument newDoc) {
        this.newDoc = newDoc;
    }

    public ReadAndWriteDocx(String In) {
        this.In = In;
    }

    public XWPFFWordExtractor ReadDocx() throws FileNotFoundException,
    IOException {
        File DocxFile = new File(In);
        FileInputStream Docxfis = new
    FileInputStream(DocxFile.getAbsolutePath());
        XWPFDocument document = new XWPFDocument(Docxfis);
        XWPFFWordExtractor extract = new XWPFFWordExtractor(document);
        return extract;
    }

    public XWPFDocument WriteDocx() {
        XWPFDocument newDoc = new XWPFDocument(); //Doc to write new doc to
        return newDoc;
    }

    //H openNewDocx δημιουργεί ένα αρχείο .docx
    public void openNewDocx() {

        try {
            newDoc.write(new
    FileOutputStream("C:\\Users\\spiros\\Documents\\NetBeansProjects\\TextDataMi
    ning\\FinalText.docx"));
        } catch (IOException e) {
            System.out.println("IOException Error!");
            e.printStackTrace();
        }
    }

    //Εμφανίζει αυτόματα το .docx που δημιουργήθηκε
    public void openFinalNewDocx() {

        try {
            if (Desktop.isDesktopSupported()) {
```



```

        rowhead.createCell(10).setCellValue("Πλημ/κείου");
        rowhead.createCell(11).setCellValue("Εφετείου");
        rowhead.createCell(12).setCellValue("Πρωτοδικ");
        rowhead.createCell(13).setCellValue("Συμβάσεως");
        rowhead.createCell(14).setCellValue("ΑΚ");

        HSSFRow row = sheet.createRow((short) this.linepos);
        for (int i = 0; i < this.freqList.size(); i++) {
            row.createCell(i).setCellValue(freqList.get(i));
        }

    } catch (Exception ex) {
        System.out.println(ex);
    }
}

public void OpenExcel() {
    filename =
"C:\\Users\\spiros\\Documents\\NetBeansProjects\\TextDataMining\\ExcelVector
sFile.xls";
    workbook = new HSSFWorkbook();
    sheet = workbook.createSheet("FirstSheet");
}

public void CloseExcel() throws FileNotFoundException, IOException {
    FileOutputStream fileOut = new FileOutputStream(filename);
    workbook.write(fileOut);
    fileOut.close();
    System.out.println("Your excel file has been generated!");
}

public void openFinalExcel() {
    try {
        if (Desktop.isDesktopSupported()) {
            Desktop.getDesktop().open(new
File("C:\\Users\\spiros\\Documents\\NetBeansProjects\\TextDataMining\\ExcelV
ectorsFile.xls"));
        }
    } catch (IOException e) {
        e.printStackTrace();
    }
}
}
}

```

## Κλάση AllFinalsVariables

```
package textdatamining;

public class AllFinalsVariables {

    final String ar1 = "άρ.";
    final String ar2 = "αρ.";
    //final String numKey = "98";
    final String parKey = "παρ.";
    final String arthro = "άρθρο";
    final String bslashKey = "/";

    final String sunt = "Συντ";
    final String sdoe = "ΣΔΟΕ";
    final String kPold = "ΚΠολΔ";
    final String idioukwдика1 = "ίδιου Κώδ";
    final String idioukwδικα2 = "ίδιου Κώδ";
    final String ankwδικα = "άνω Κώδ";
    final String idiounomoul = "ίδιου Νόμου";
    final String idiounomou2 = "ίδιου νόμου";
    //final String idiouarthrou = "ίδιου άρθρου";
    final String daspd = "ΔΣΑΠΔ";
    final String esda1 = "ΕΣΔΑ";
    final String esda2 = "Ε.Σ.Δ.Α.";
    final String pk = "ΠΚ";
    final String pk1 = "Π.Κ.";
    final String pk2 = "Ποινικού Κώδικα";
    final String kpd = "ΚΠΔ";
    final String kpd1 = "Κ.Π.Δ.";//!!!!!!!!!!1 και χωρίς τελεί
    final String kpd2 = "ΚΠολνΔ";
    final String kpd3 = "Κ.Ποιν.Δ.";
    final String kpd4 = "Κώδικα Ποινικής Δικονομίας";
    final String n = "Ν.";
    final String n1 = "ν.";
    final String ndotd = "ν.δ.";
    final String ndotd1 = "Ν.Δ.";
    final String dikasthriou = "Δικαστηρ";
    final String plhmmleiodikeiou = "Πλημ"; // "Πλημ/κείου" "Πλημμ."
    "Πλημμελειοδικείου"
    final String efeteiou1 = "Εφετείου";
    final String protodikeiou = "Πρωτοδικ";
    final String eurvpaikhsSumvasews = "Συμβάσ";
    final String Ak1 = "ΑΚ";
    final String Ak2 = "Α.Κ.";
    //*****

    //COLORS
    final String Red = "FF0000";
    final String Tomato = "FF6347";
    final String IndianRed = "CD5C5C";
    final String GoldenRod = "B8860B";
    final String DarkOrange = "FF8C00";
    final String Olive = "808000";
    final String DarkOlive = "808000";
    final String Maroon = "800000";
    final String Lime = "00FF00";
    final String LimeGreen = "32CD32";
```

```

    final String DarkCyan = "008B8B";
    final String SteelBlue = "4682B4";
    final String Blue = "0000FF";
    final String BlueViolet = "8A2BE2";
    final String Fuchsia = "FF00FF";
    final String SaddleBrown = "8B4513";
}

```

## Κλάση PerceptronFrame

```

package textdatamining;

import java.io.*;
import weka.core.*;
import weka.core.Instances;
import weka.classifiers.Evaluation;
import weka.classifiers.functions.MultilayerPerceptron;
import java.awt.Graphics2D;
import java.awt.Image;
import java.awt.RenderingHints;
import java.awt.Toolkit;
import java.awt.image.BufferedImage;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.swing.ImageIcon;
import javax.swing.JFileChooser;
import javax.swing.JLabel;

/**
 *
 * @author spiros
 */

public class PerceptronFrame extends javax.swing.JFrame {

    String createModel = null;
    String path = null;
    Instances train;
    MultilayerPerceptron mlp = new MultilayerPerceptron();

    /*KMeans.buildClusterer(train);
    ClusterEvaluation evalKM = new ClusterEvaluation();
    evalKM.evaluateClusterer(train);
    System.out.printf("CLUSTERER \n" + evalKM.getClassesToClusters());
    */

    public PerceptronFrame() {
        Image img = Toolkit.getDefaultToolkit().createImage("Blueprint
border.jpg");
        ImageIcon im = new ImageIcon();

        BufferedImage img1 = scaleImage(490, 600, "Blueprint border.jpg");

```



```

        im.setImage(img1);
        setContentPane(new JLabel(im));

        initComponents();
    }

    /**
     * This method is called from within the constructor to initialize the
    form.
     * WARNING: Do NOT modify this code. The content of this method is
    always
     * regenerated by the Form Editor.
     */
    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated Code">
    private void initComponents() {

        ChooserFileARFF = new javax.swing.JFileChooser();
        ResultsDialog = new javax.swing.JDialog();
        jScrollPane1 = new javax.swing.JScrollPane();
        jTextArea1 = new javax.swing.JTextArea();
        CloseDialogButton = new javax.swing.JButton();
        jLabel11 = new javax.swing.JLabel();
        jLabel12 = new javax.swing.JLabel();
        jLabel13 = new javax.swing.JLabel();
        jLabel14 = new javax.swing.JLabel();
        jLabel15 = new javax.swing.JLabel();
        jLabel16 = new javax.swing.JLabel();
        jLabel17 = new javax.swing.JLabel();
        jLabel18 = new javax.swing.JLabel();
        jLabel19 = new javax.swing.JLabel();
        jLabel110 = new javax.swing.JLabel();
        jLabel111 = new javax.swing.JLabel();
        TrainingButton = new javax.swing.JButton();
        OpenARFFfile = new javax.swing.JButton();
        ClassificationButton = new javax.swing.JButton();
        ClosePerceptronFrame = new javax.swing.JButton();
        HiddenLayersField = new javax.swing.JTextField();
        LearningRateField = new javax.swing.JTextField();
        MomentumField = new javax.swing.JTextField();
        TrainingTimeField = new javax.swing.JTextField();
        ValidationSetSizeField1 = new javax.swing.JTextField();
        ValidationThresholdField = new javax.swing.JTextField();
        SeedField = new javax.swing.JTextField();

        ResultsDialog.setMinimumSize(new java.awt.Dimension(600, 550));

        jTextArea1.setColumns(20);
        jTextArea1.setFont(new java.awt.Font("Courier New", 0, 13)); //
NOI18N
        jTextArea1.setRows(5);
        jScrollPane1.setViewportView(jTextArea1);

        CloseDialogButton.setFont(new java.awt.Font("Trebuchet MS", 0, 12));
// NOI18N
        CloseDialogButton.setText("Close");
        CloseDialogButton.addActionListener(new
java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {

```

```

        CloseDialogButtonActionPerformed(evt);
    }
});

javax.swing.GroupLayout ResultsDialogLayout = new
javax.swing.GroupLayout(ResultsDialog.getContentPane());
ResultsDialog.getContentPane().setLayout(ResultsDialogLayout);
ResultsDialogLayout.setHorizontalGroup(

ResultsDialogLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(ResultsDialogLayout.createSequentialGroup()
        .addContainerGap()
        .addGroup(ResultsDialogLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addComponent(jScrollPane1,
                javax.swing.GroupLayout.DEFAULT_SIZE, 530, Short.MAX_VALUE)
            .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
                ResultsDialogLayout.createSequentialGroup()
                    .addGap(0, 0, Short.MAX_VALUE)
                    .addComponent(CloseDialogButton,
                        javax.swing.GroupLayout.PREFERRED_SIZE, 90,
                        javax.swing.GroupLayout.PREFERRED_SIZE)))
            .addContainerGap())
    .addGroup(ResultsDialogLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(ResultsDialogLayout.createSequentialGroup()
            .addContainerGap()
            .addGroup(ResultsDialogLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)
                .addComponent(jScrollPane1,
                    javax.swing.GroupLayout.DEFAULT_SIZE, 367, Short.MAX_VALUE)
                .addGap(18, 18, 18)
                .addComponent(CloseDialogButton)
                .addContainerGap())
            .addContainerGap())
        .addGroup(ResultsDialogLayout.createSequentialGroup()
            .addContainerGap()
            .addComponent(jLabel1,
                javax.swing.GroupLayout.PREFERRED_SIZE, 18, Short.MAX_VALUE)
            .addContainerGap())
        .addGroup(ResultsDialogLayout.createSequentialGroup()
            .addContainerGap()
            .addComponent(jLabel2,
                javax.swing.GroupLayout.PREFERRED_SIZE, 14, Short.MAX_VALUE)
            .addContainerGap())
        .addGroup(ResultsDialogLayout.createSequentialGroup()
            .addContainerGap()
            .addComponent(jLabel3,
                javax.swing.GroupLayout.PREFERRED_SIZE, 13, Short.MAX_VALUE)
            .addContainerGap())
        .addGroup(ResultsDialogLayout.createSequentialGroup()
            .addContainerGap()
            .addComponent(jLabel4,
                javax.swing.GroupLayout.PREFERRED_SIZE, 13, Short.MAX_VALUE)
            .addContainerGap())
        .addGroup(ResultsDialogLayout.createSequentialGroup()
            .addContainerGap()
            .addComponent(jLabel5,
                javax.swing.GroupLayout.PREFERRED_SIZE, 13, Short.MAX_VALUE)
            .addContainerGap())
        .addGroup(ResultsDialogLayout.createSequentialGroup()
            .addContainerGap()
            .addComponent(jLabel6,
                javax.swing.GroupLayout.PREFERRED_SIZE, 13, Short.MAX_VALUE)
            .addContainerGap())
    );

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);

jLabel1.setFont(new java.awt.Font("Trebuchet MS", 1, 18)); // NOI18N
jLabel1.setText("Classification in Legal Texts");

jLabel2.setFont(new java.awt.Font("Trebuchet MS", 1, 14)); // NOI18N
jLabel2.setText("Create a MultiLayerPerceptron Model");

jLabel3.setFont(new java.awt.Font("Trebuchet MS", 0, 13)); // NOI18N
jLabel3.setText("Hidden Layers :");

jLabel4.setFont(new java.awt.Font("Tahoma", 0, 13)); // NOI18N
jLabel4.setText("Learning Rate :");

jLabel5.setFont(new java.awt.Font("Tahoma", 0, 13)); // NOI18N
jLabel5.setText("Momentum :");

jLabel6.setFont(new java.awt.Font("Tahoma", 0, 13)); // NOI18N
jLabel6.setText("Training Time :");

```

```

jLabel17.setFont(new java.awt.Font("Tahoma", 0, 13)); // NOI18N
jLabel17.setText("Validation Set Size :");

jLabel18.setFont(new java.awt.Font("Tahoma", 0, 13)); // NOI18N
jLabel18.setText("Validation Threshold :");

jLabel19.setFont(new java.awt.Font("Tahoma", 0, 13)); // NOI18N
jLabel19.setText("Seed :");

jLabel10.setFont(new java.awt.Font("Trebuchet MS", 0, 14)); //
NOI18N
jLabel10.setText("For training push the button :");

jLabel11.setFont(new java.awt.Font("Trebuchet MS", 0, 14)); //
NOI18N
jLabel11.setText("For choose file push the button :");

TrainingButton.setFont(new java.awt.Font("Trebuchet MS", 0, 12)); //
NOI18N
TrainingButton.setText("Training");
TrainingButton.addActionListener(new java.awt.event.ActionListener()
{
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        TrainingButtonActionPerformed(evt);
    }
});

OpenARFFfile.setFont(new java.awt.Font("Trebuchet MS", 0, 12)); //
NOI18N
OpenARFFfile.setText("Open");
OpenARFFfile.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        OpenARFFfileActionPerformed(evt);
    }
});

ClassificationButton.setFont(new java.awt.Font("Trebuchet MS", 0,
12)); // NOI18N
ClassificationButton.setText("Classification");
ClassificationButton.addActionListener(new
java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        ClassificationButtonActionPerformed(evt);
    }
});

ClosePerceptronFrame.setFont(new java.awt.Font("Trebuchet MS", 0,
12)); // NOI18N
ClosePerceptronFrame.setText("Close");
ClosePerceptronFrame.addActionListener(new
java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        ClosePerceptronFrameActionPerformed(evt);
    }
});

HiddenLayersField.setFont(new java.awt.Font("Trebuchet MS", 0, 12));
// NOI18N

```

```

HiddenLayersField.setText("a");

LearningRateField.setFont(new java.awt.Font("Trebuchet MS", 0, 12));
// NOI18N
LearningRateField.setText("0.3");

MomentumField.setFont(new java.awt.Font("Trebuchet MS", 0, 12)); //
NOI18N
MomentumField.setText("0.2");

TrainingTimeField.setFont(new java.awt.Font("Trebuchet MS", 0, 12));
// NOI18N
TrainingTimeField.setText("500");

ValidationSetSizeField1.setFont(new java.awt.Font("Trebuchet MS", 0,
12)); // NOI18N
ValidationSetSizeField1.setText("0");

ValidationThresholdField.setFont(new java.awt.Font("Trebuchet MS",
0, 12)); // NOI18N
ValidationThresholdField.setText("20");

SeedField.setFont(new java.awt.Font("Trebuchet MS", 0, 12)); //
NOI18N
SeedField.setText("0");

javax.swing.GroupLayout layout = new
javax.swing.GroupLayout(getContentPane());
getContentPane().setLayout(layout);
layout.setHorizontalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(layout.createSequentialGroup()
        .addComponent(jLabel10)
        .addGap(38, 38, 38)
        .addComponent(TrainingButton,
javax.swing.GroupLayout.PREFERRED_SIZE, 90,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addGap(18, 18, 18)

    .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addComponent(ClosePerceptronFrame,
javax.swing.GroupLayout.Alignment.TRAILING,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE)
        .addComponent(ClassificationButton,
javax.swing.GroupLayout.DEFAULT_SIZE, 134, Short.MAX_VALUE)))
    .addGroup(layout.createSequentialGroup()

    .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup()
            .addComponent(jLabel10)
            .addGap(38, 38, 38)
            .addComponent(TrainingButton,
javax.swing.GroupLayout.PREFERRED_SIZE, 90,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addGap(18, 18, 18)

```

```

        .addComponent(jLabel11))
        .addComponent(jLabel12)
        .addGroup(layout.createSequentialGroup())
        .addComponent(jLabel11)
        .addGap(18, 18, 18)
        .addComponent(OpenARFFfile,
javax.swing.GroupLayout.PREFERRED_SIZE, 90,
javax.swing.GroupLayout.PREFERRED_SIZE))
        .addGap(0, 0, Short.MAX_VALUE)
        .addGroup(layout.createSequentialGroup()

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
NG)

        .addComponent(jLabel13)
        .addComponent(jLabel14)
        .addComponent(jLabel15)
        .addComponent(jLabel16)
        .addComponent(jLabel17)
        .addComponent(jLabel18)
        .addComponent(jLabel19))
        .addGap(18, 18, 18)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
NG)

        .addComponent(SeedField)
        .addComponent(ValidationThresholdField)
        .addComponent(ValidationSetSizeField1)
        .addComponent(TrainingTimeField)
        .addComponent(MomentumField)
        .addComponent(LearningRateField)
        .addComponent(HiddenLayersField)))
        .addContainerGap()
    );
    layout.setVerticalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup()
            .addGap(31, 31, 31)
            .addComponent(jLabel11,
javax.swing.GroupLayout.PREFERRED_SIZE, 32,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addGap(34, 34, 34)
            .addComponent(jLabel12,
javax.swing.GroupLayout.PREFERRED_SIZE, 23,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addGap(18, 18, 18)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
INE)

        .addComponent(jLabel13)
        .addComponent(HiddenLayersField,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
        .addGap(18, 18, 18)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
INE)

        .addComponent(jLabel14)

```

```

        .addComponent( LearningRateField,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
        .addGap(18, 18, 18)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
.addComponent(jLabel15)
.addComponent(MomentumField,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
        .addGap(18, 18, 18)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
.addComponent(jLabel16)
.addComponent(TrainingTimeField,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
        .addGap(18, 18, 18)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
.addComponent(jLabel17)
.addComponent(ValidationSetSizeField1,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
        .addGap(18, 18, 18)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
.addComponent(jLabel18)
.addComponent(ValidationThresholdField,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
        .addGap(18, 18, 18)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
.addComponent(jLabel19)
.addComponent(SeedField,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 31,
Short.MAX_VALUE)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
.addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
layout.createSequentialGroup())

```

```

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
    .addComponent(jLabel10)
    .addComponent(TrainingButton))
.addGap(18, 18, 18)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
    .addComponent(jLabel11)
    .addComponent(OpenARFFfile))
.addComponent(ClassificationButton,
javax.swing.GroupLayout.Alignment.TRAILING,
javax.swing.GroupLayout.PREFERRED_SIZE, 72,
javax.swing.GroupLayout.PREFERRED_SIZE)
.addGap(18, 18, 18)
.addComponent(ClosePerceptronFrame)
.addContainerGap()
);

pack();
} // </editor-fold>

private void TrainingButtonActionPerformed(java.awt.event.ActionEvent
evt) {
    createModel = "-L " + LearningRateField.getText() + " -M " +
MomentumField.getText() + " -N " + TrainingTimeField.getText()
        + " -V " + ValidationSetSizeField1.getText() + " -S " +
SeedField.getText() + " -E " + ValidationThresholdField.getText()
        + " -H " + HiddenLayersField.getText();
    System.out.println(createModel);
    try {
        FileReader trainreader = new
FileReader("C:\\Users\\spiros\\Documents\\TrainingSet.arff");
        train = new Instances(trainreader);
        train.setClassIndex(train.numAttributes() - 1);

        mlp.setOptions(Utils.splitOptions(createModel));
        mlp.buildClassifier(train);

        Evaluation eval = new Evaluation(train);
        eval.evaluateModel(mlp, train);

        JTextArea1.setText(eval.toSummaryString("\nTraining Results\n"
            + "=====\n",
false) + "\n\n"
            + eval.toClassDetailsString() + "\n\n" +
eval.toMatrixString());

        ResultsDialog.setVisible(true);

        System.out.println(eval.toSummaryString("\nTRAIN
Results\n=====\n", false));
        System.out.println(eval.toClassDetailsString());
        System.out.println(eval.confusionMatrix());
        System.out.println(eval.toMatrixString());

        trainreader.close();
    } catch (FileNotFoundException ex) {

```

```

Logger.getLogger(PerceptronFrame.class.getName()).log(Level.SEVERE, null,
ex);
    } catch (IOException ex) {

Logger.getLogger(PerceptronFrame.class.getName()).log(Level.SEVERE, null,
ex);
    } catch (Exception ex) {

Logger.getLogger(PerceptronFrame.class.getName()).log(Level.SEVERE, null,
ex);
    }
}

private void OpenARFFfileActionPerformed(java.awt.event.ActionEvent evt)
{
    if (ChooserFileARFF.showOpenDialog(this) ==
JFileChooser.APPROVE_OPTION) {
        path = ChooserFileARFF.getSelectedFile().getAbsolutePath();
        System.out.println(path);
    }
}

private void
ClassificationButtonActionPerformed(java.awt.event.ActionEvent evt) {
    try {
        FileReader testreader = new FileReader(path);
        Instances test = new Instances(testreader);
        test.setClassIndex(test.numAttributes() - 1);

        Evaluation eval = new Evaluation(train);
        eval.evaluateModel(mlp, test);
        JTextArea1.setText(eval.toSummaryString("\nMultiLayer Perceptron
- Classification Results\n"
+
"===== \n", false) + "\n\n"
+ eval.toClassDetailsString() + "\n\n" +
eval.toMatrixString());

        ResultsDialog.setVisible(true);

        System.out.println(eval.toSummaryString("\nResults\n===== \n",
false));
        System.out.println(eval.toClassDetailsString());
        System.out.println(eval.confusionMatrix());
        System.out.println(eval.toMatrixString());
        testreader.close();
    } catch (FileNotFoundException ex) {
    } catch (IOException ex) {
    } catch (Exception ex) {

Logger.getLogger(PerceptronFrame.class.getName()).log(Level.SEVERE, null,
ex);
    }
}

private void
ClosePerceptronFrameActionPerformed(java.awt.event.ActionEvent evt) {

```



```

        this.dispose();
    }

    private void CloseDialogButtonActionPerformed(java.awt.event.ActionEvent
    evt) {
        ResultsDialog.setVisible(false);
    }

    public void Main() {
        /* Set the Nimbus look and feel */
        //<editor-fold defaultstate="collapsed" desc=" Look and feel setting
code (optional) ">
        /* If Nimbus (introduced in Java SE 6) is not available, stay with
the default look and feel.
         * For details see
http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
         */
        try {
            for (javax.swing.UIManager.LookAndFeelInfo info :
                javax.swing.UIManager.getInstalledLookAndFeels()) {
                if ("Nimbus".equals(info.getName())) {
                    javax.swing.UIManager.setLookAndFeel(info.getClassName());
                    break;
                }
            }
        } catch (ClassNotFoundException ex) {
            java.util.logging.Logger.getLogger(PerceptronFrame.class.getName()).log(java
            .util.logging.Level.SEVERE, null, ex);
        } catch (InstantiationException ex) {
            java.util.logging.Logger.getLogger(PerceptronFrame.class.getName()).log(java
            .util.logging.Level.SEVERE, null, ex);
        } catch (IllegalAccessException ex) {
            java.util.logging.Logger.getLogger(PerceptronFrame.class.getName()).log(java
            .util.logging.Level.SEVERE, null, ex);
        } catch (javax.swing.UnsupportedLookAndFeelException ex) {
            java.util.logging.Logger.getLogger(PerceptronFrame.class.getName()).log(java
            .util.logging.Level.SEVERE, null, ex);
        }
        //</editor-fold>
        //</editor-fold>

        /* Create and display the form */
        java.awt.EventQueue.invokeLater(new Runnable() {
            public void run() {
                new PerceptronFrame().setVisible(true);
            }
        });
    }

    // Variables declaration - do not modify
    private javax.swing.JFileChooser ChooserFileARFF;
    private javax.swing.JButton ClassificationButton;
    private javax.swing.JButton CloseDialogButton;
    private javax.swing.JButton ClosePerceptronFrame;

```

```

private javax.swing.JTextField HiddenLayersField;
private javax.swing.JTextField LearningRateField;
private javax.swing.JTextField MomentumField;
private javax.swing.JButton OpenARFFfile;
private javax.swing.JDialog ResultsDialog;
private javax.swing.JTextField SeedField;
private javax.swing.JButton TrainingButton;
private javax.swing.JTextField TrainingTimeField;
private javax.swing.JTextField ValidationSetSizeField1;
private javax.swing.JTextField ValidationThresholdField;
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel10;
private javax.swing.JLabel jLabel11;
private javax.swing.JLabel jLabel12;
private javax.swing.JLabel jLabel13;
private javax.swing.JLabel jLabel14;
private javax.swing.JLabel jLabel15;
private javax.swing.JLabel jLabel16;
private javax.swing.JLabel jLabel17;
private javax.swing.JLabel jLabel18;
private javax.swing.JLabel jLabel19;
private javax.swing.JScrollPane jScrollPane1;
private javax.swing.JTextArea jTextArea1;
// End of variables declaration

BufferedImage scaleImage(int WIDTH, int HEIGHT, String filename) {
    BufferedImage bi = null;

    try {
        ImageIcon ii = new ImageIcon(filename); // path to image
        bi = new BufferedImage(WIDTH, HEIGHT,
BufferedImage.TYPE_INT_RGB);
        Graphics2D g2d = (Graphics2D) bi.createGraphics();
        g2d.addRenderingHints(new
RenderingHints(RenderingHints.KEY_RENDERING,
RenderingHints.VALUE_RENDER_QUALITY));
        g2d.drawImage(ii.getImage(), 0, 0, WIDTH, HEIGHT, null);
    } catch (Exception e) {
        return null;
    }
    return bi;
}
}

```

## Κλάση RBFNetworkFrame

```
package textdatamining;

import java.awt.Graphics2D;
import java.awt.Image;
import java.awt.RenderingHints;
import java.awt.Toolkit;
import java.awt.image.BufferedImage;
import java.io.FileNotFoundException;
import java.io.FileReader;
import java.io.IOException;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.swing.ImageIcon;
import javax.swing.JFileChooser;
import javax.swing.JLabel;
import weka.classifiers.Evaluation;
import weka.classifiers.functions.RBFNetwork;
import weka.core.Instances;
import weka.core.Utils;

/**
 *
 * @author spiros
 */
public class RBFNetworkFrame extends javax.swing.JFrame {

    String createModel = null;
    String path = null;
    Instances train;

    RBFNetwork rbf = new RBFNetwork();

    public RBFNetworkFrame() {
        Image img = Toolkit.getDefaultToolkit().createImage("Blueprint
border.jpg");
        ImageIcon im = new ImageIcon();

        BufferedImage img1 = scaleImage(490, 520, "Blueprint border.jpg");

        im.setImage(img1);
        setContentPane(new JLabel(im));

        initComponents();
    }

    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated Code">
    private void initComponents() {

        ResultsDialog = new javax.swing.JDialog();
        CloseDialogButton = new javax.swing.JButton();
        jScrollPane1 = new javax.swing.JScrollPane();
        jTextArea1 = new javax.swing.JTextArea();
        ChooserFileARFF = new javax.swing.JFileChooser();
        jLabel1 = new javax.swing.JLabel();
        jLabel2 = new javax.swing.JLabel();
    }
}

```

```

jLabel13 = new javax.swing.JLabel();
jLabel14 = new javax.swing.JLabel();
jLabel15 = new javax.swing.JLabel();
jLabel16 = new javax.swing.JLabel();
jLabel17 = new javax.swing.JLabel();
ClusteringSeedField = new javax.swing.JTextField();
MaxItsField = new javax.swing.JTextField();
MinStdDevField = new javax.swing.JTextField();
NumClustersField = new javax.swing.JTextField();
RidgeField = new javax.swing.JTextField();
jLabel18 = new javax.swing.JLabel();
jLabel19 = new javax.swing.JLabel();
TrainingButton = new javax.swing.JButton();
OpenARFFfile = new javax.swing.JButton();
ClussificationButton = new javax.swing.JButton();
CloseRBFNetworkFrame = new javax.swing.JButton();

ResultsDialog.setMinimumSize(new java.awt.Dimension(600, 550));

CloseDialogButton.setFont(new java.awt.Font("Trebuchet MS", 0, 12));
// NOI18N
CloseDialogButton.setText("Close");
CloseDialogButton.addActionListener(new
java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        CloseDialogButtonActionPerformed(evt);
    }
});

jTextArea1.setColumns(20);
jTextArea1.setFont(new java.awt.Font("Courier New", 0, 13)); //
NOI18N
jTextArea1.setRows(5);
jScrollPane1.setViewportView(jTextArea1);

javax.swing.GroupLayout ResultsDialogLayout = new
javax.swing.GroupLayout(ResultsDialog.getContentPane());
ResultsDialog.getContentPane().setLayout(ResultsDialogLayout);
ResultsDialogLayout.setHorizontalGroup(

ResultsDialogLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(ResultsDialogLayout.createSequentialGroup()
        .addContainerGap(450, Short.MAX_VALUE)
        .addComponent(CloseDialogButton,
javax.swing.GroupLayout.PREFERRED_SIZE, 90,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addContainerGap()
        .addComponent(jScrollPane1,
javax.swing.GroupLayout.Alignment.TRAILING,
javax.swing.GroupLayout.DEFAULT_SIZE, 550, Short.MAX_VALUE)
    );
ResultsDialogLayout.setVerticalGroup(

ResultsDialogLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
ResultsDialogLayout.createSequentialGroup()
        .addContainerGap()

```

```

        .addComponent(jScrollPane1,
javax.swing.GroupLayout.DEFAULT_SIZE, 367, Short.MAX_VALUE)
        .addGap(18, 18, 18)
        .addComponent(CloseDialogButton)
        .addContainerGap()
    );

    setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);

    jLabel1.setFont(new java.awt.Font("Trebuchet MS", 1, 18)); // NOI18N
    jLabel1.setText("Classification in Legal Texts");

    jLabel2.setFont(new java.awt.Font("Trebuchet MS", 1, 14)); // NOI18N
    jLabel2.setText("Create a RBFNetwork Model");

    jLabel3.setFont(new java.awt.Font("Trebuchet MS", 0, 13)); // NOI18N
    jLabel3.setText("ClusteringSeed :");

    jLabel4.setFont(new java.awt.Font("Trebuchet MS", 0, 13)); // NOI18N
    jLabel4.setText("MaxIts:");

    jLabel5.setFont(new java.awt.Font("Trebuchet MS", 0, 13)); // NOI18N
    jLabel5.setText("MinStdDev :");

    jLabel6.setFont(new java.awt.Font("Trebuchet MS", 0, 13)); // NOI18N
    jLabel6.setText("NumClusters :");

    jLabel7.setFont(new java.awt.Font("Trebuchet MS", 0, 13)); // NOI18N
    jLabel7.setText("Ridge :");

    ClusteringSeedField.setFont(new java.awt.Font("Trebuchet MS", 0,
12)); // NOI18N
    ClusteringSeedField.setText("1");

    MaxItsField.setFont(new java.awt.Font("Trebuchet MS", 0, 12)); //
NOI18N
    MaxItsField.setText("-1");

    MinStdDevField.setFont(new java.awt.Font("Trebuchet MS", 0, 12)); //
NOI18N
    MinStdDevField.setText("0.1");

    NumClustersField.setFont(new java.awt.Font("Trebuchet MS", 0, 12));
// NOI18N
    NumClustersField.setText("2");

    RidgeField.setFont(new java.awt.Font("Trebuchet MS", 0, 12)); //
NOI18N
    RidgeField.setText("1.0E-8");

    jLabel8.setFont(new java.awt.Font("Trebuchet MS", 0, 14)); // NOI18N
    jLabel8.setText("For training push the button :");

    jLabel9.setFont(new java.awt.Font("Trebuchet MS", 0, 14)); // NOI18N
    jLabel9.setText("For choose file push the button :");

    TrainingButton.setFont(new java.awt.Font("Trebuchet MS", 0, 12)); //
NOI18N
    TrainingButton.setText("Training");

```



```

        .addComponent (MinStdDevField)
        .addComponent (MaxItsField)
        .addComponent (RidgeField)))
    .addGroup (layout.createSequentialGroup ())
        .addComponent (jLabel6)
        .addGap (59, 59, 59)
        .addComponent (NumClustersField)
    .addGroup (layout.createSequentialGroup ())

    .addGroup (layout.createParallelGroup (javax.swing.GroupLayout.Alignment.LEADING)
    NG)
        .addComponent (jLabel2,
javax.swing.GroupLayout.PREFERRED_SIZE, 232,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent (jLabel5)
        .addComponent (jLabel7)
        .addGroup (layout.createSequentialGroup ())

    .addGroup (layout.createParallelGroup (javax.swing.GroupLayout.Alignment.LEADING)
    NG)
        .addComponent (jLabel9)
        .addComponent (jLabel8)
        .addGap (18, 18, 18)

    .addGroup (layout.createParallelGroup (javax.swing.GroupLayout.Alignment.LEADING)
    NG, false)
        .addComponent (OpenARFFfile,
javax.swing.GroupLayout.DEFAULT_SIZE, 89, Short.MAX_VALUE)
        .addComponent (TrainingButton,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE))
        .addGap (18, 18, 18)

    .addGroup (layout.createParallelGroup (javax.swing.GroupLayout.Alignment.LEADING)
    NG, false)
        .addComponent (CloseRBFNetworkFrame,
javax.swing.GroupLayout.Alignment.TRAILING,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE)
        .addComponent (ClussificationButton,
javax.swing.GroupLayout.Alignment.TRAILING,
javax.swing.GroupLayout.DEFAULT_SIZE, 135, Short.MAX_VALUE)))
        .addGap (0, 0, Short.MAX_VALUE))
        .addContainerGap ()
        .addGroup (layout.createSequentialGroup ())
            .addGap (77, 77, 77)
            .addComponent (jLabel11)
            .addContainerGap (javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE))
        );
    layout.setVerticalGroup (

    layout.createParallelGroup (javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup (layout.createSequentialGroup ())
            .addGap (33, 33, 33)
            .addComponent (jLabel11,
javax.swing.GroupLayout.PREFERRED_SIZE, 33,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addGap (27, 27, 27)

```

```

        .addComponent(jLabel2,
javax.swing.GroupLayout.PREFERRED_SIZE, 22,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addGap(18, 18, 18)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
        .addComponent(jLabel3)
        .addComponent(ClusteringSeedField,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
        .addGap(18, 18, 18)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
        .addComponent(jLabel4)
        .addComponent(MaxItsField,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
        .addGap(18, 18, 18)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
        .addComponent(jLabel5)
        .addComponent(MinStdDevField,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
        .addGap(18, 18, 18)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
        .addComponent(jLabel6)
        .addComponent(NumClustersField,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
        .addGap(18, 18, 18)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
        .addComponent(jLabel7)
        .addComponent(RidgeField,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup()
            .addGap(40, 40, 40)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup()
            .addComponent(jLabel18)
            .addGap(23, 23, 23))

```



```

.addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
layout.createSequentialGroup()
    .addComponent(TrainingButton)
    .addGap(18, 18, 18))

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
    .addComponent(jLabel9)
    .addComponent(OpenARFFfile))
layout.createSequentialGroup()

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
    .addComponent(ClusificationButton,
javax.swing.GroupLayout.PREFERRED_SIZE, 72,
javax.swing.GroupLayout.PREFERRED_SIZE))
    .addGap(18, 18, 18)
    .addComponent(CloseRBFNetworkFrame)
    .addContainerGap(65, Short.MAX_VALUE)
);

pack();
} // </editor-fold>

private void TrainingButtonActionPerformed(java.awt.event.ActionEvent
evt) {
    createModel = "-B " + NumClustersField.getText() + " -S " +
ClusteringSeedField.getText() + " -R " + RidgeField.getText()
        + " -M " + MaxItsField.getText() + " -W " +
MinStdDevField.getText();
    System.out.println(createModel);
    try {
        FileReader trainreader = new
FileReader("C:\\Users\\spiros\\Documents\\TrainingSet.arff");
        train = new Instances(trainreader);
        train.setClassIndex(train.numAttributes() - 1);

        rbf.setOptions(Utils.splitOptions(createModel));
        rbf.buildClassifier(train);

        Evaluation eval = new Evaluation(train);
        eval.evaluateModel(rbf, train);

        JTextArea1.setText(eval.toSummaryString("\nTraining Results\n"
            + "=====\n",
false) + "\n\n"
            + eval.toClassDetailsString() + "\n\n" +
eval.toMatrixString());

        ResultsDialog.setVisible(true);

        System.out.println(eval.toSummaryString("\nTRAIN
Results\n=====\n", false));
        System.out.println(eval.toClassDetailsString());
        System.out.println(eval.confusionMatrix());
        System.out.println(eval.toMatrixString());

        trainreader.close();

```

```

        } catch (FileNotFoundException ex) {

Logger.getLogger(PerceptronFrame.class.getName()).log(Level.SEVERE, null,
ex);
        } catch (IOException ex) {

Logger.getLogger(PerceptronFrame.class.getName()).log(Level.SEVERE, null,
ex);
        } catch (Exception ex) {

Logger.getLogger(PerceptronFrame.class.getName()).log(Level.SEVERE, null,
ex);
    }
}

private void OpenARFFfileActionPerformed(java.awt.event.ActionEvent evt)
{
    if (ChooserFileARFF.showOpenDialog(this) ==
ChooserFileARFF.APPROVE_OPTION) {
        path = ChooserFileARFF.getSelectedFile().getAbsolutePath();
        System.out.println(path);
    }
}

private void
ClassificationButtonActionPerformed(java.awt.event.ActionEvent evt) {
    try {
        FileReader testreader = new FileReader(path);
        Instances test = new Instances(testreader);
        test.setClassIndex(test.numAttributes() - 1);

        Evaluation eval = new Evaluation(train);
        eval.evaluateModel(rbf, test);
        JTextArea1.setText(eval.toSummaryString("\nRBFNetwork -
Classification Results\n"
+
"=====\n", false) + "\n\n"
+ eval.toClassDetailsString() + "\n\n" +
eval.toMatrixString());

        ResultsDialog.setVisible(true);

        System.out.println(eval.toSummaryString("\nResults\n=====\n",
false));
        System.out.println(eval.toClassDetailsString());
        System.out.println(eval.confusionMatrix());
        System.out.println(eval.toMatrixString());
        testreader.close();
    } catch (FileNotFoundException ex) {
    } catch (IOException ex) {
    } catch (Exception ex) {

Logger.getLogger(PerceptronFrame.class.getName()).log(Level.SEVERE, null,
ex);
    }
}
}

```

```

private void
CloseRBFNetworkFrameActionPerformed(java.awt.event.ActionEvent evt) {
    this.dispose();
}

private void CloseDialogButtonActionPerformed(java.awt.event.ActionEvent
evt) {
    ResultsDialog.setVisible(false);
}

/**
 * @param args the command line arguments
 */
public void main() {
    /* Set the Nimbus look and feel */
    //<editor-fold defaultstate="collapsed" desc=" Look and feel setting
code (optional) ">
    /* If Nimbus (introduced in Java SE 6) is not available, stay with
the default look and feel.
    * For details see
http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
    */
    try {
        for (javax.swing.UIManager.LookAndFeelInfo info :
javax.swing.UIManager.getInstalledLookAndFeels()) {
            if ("Nimbus".equals(info.getName())) {
                javax.swing.UIManager.setLookAndFeel(info.getClassName());
                break;
            }
        }
    } catch (ClassNotFoundException ex) {
        java.util.logging.Logger.getLogger(RBFNetworkFrame.class.getName()).log(java
.util.logging.Level.SEVERE, null, ex);
    } catch (InstantiationException ex) {
        java.util.logging.Logger.getLogger(RBFNetworkFrame.class.getName()).log(java
.util.logging.Level.SEVERE, null, ex);
    } catch (IllegalAccessException ex) {
        java.util.logging.Logger.getLogger(RBFNetworkFrame.class.getName()).log(java
.util.logging.Level.SEVERE, null, ex);
    } catch (javax.swing.UnsupportedLookAndFeelException ex) {
        java.util.logging.Logger.getLogger(RBFNetworkFrame.class.getName()).log(java
.util.logging.Level.SEVERE, null, ex);
    }
}
//</editor-fold>

/* Create and display the form */
java.awt.EventQueue.invokeLater(new Runnable() {
    public void run() {
        new RBFNetworkFrame().setVisible(true);
    }
});
}

// Variables declaration - do not modify

```

```

private javax.swing.JFileChooser ChooserFileARFF;
private javax.swing.JButton CloseDialogButton;
private javax.swing.JButton CloseRBFNetworkFrame;
private javax.swing.JButton ClussificationButton;
private javax.swing.JTextField ClusteringSeedField;
private javax.swing.JTextField MaxItsField;
private javax.swing.JTextField MinStdDevField;
private javax.swing.JTextField NumClustersField;
private javax.swing.JButton OpenARFFfile;
private javax.swing.JDialog ResultsDialog;
private javax.swing.JTextField RidgeField;
private javax.swing.JButton TrainingButton;
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel2;
private javax.swing.JLabel jLabel3;
private javax.swing.JLabel jLabel4;
private javax.swing.JLabel jLabel5;
private javax.swing.JLabel jLabel6;
private javax.swing.JLabel jLabel7;
private javax.swing.JLabel jLabel8;
private javax.swing.JLabel jLabel9;
private javax.swing.JScrollPane jScrollPane1;
private javax.swing.JTextArea jTextArea1;
// End of variables declaration

BufferedImage scaleImage(int WIDTH, int HEIGHT, String filename) {
    BufferedImage bi = null;

    try {
        ImageIcon ii = new ImageIcon(filename); // path to image
        bi = new BufferedImage(WIDTH, HEIGHT,
BufferedImage.TYPE_INT_RGB);
        Graphics2D g2d = (Graphics2D) bi.createGraphics();
        g2d.addRenderingHints(new
RenderingHints(RenderingHints.KEY_RENDERING,
RenderingHints.VALUE_RENDER_QUALITY));
        g2d.drawImage(ii.getImage(), 0, 0, WIDTH, HEIGHT, null);
    } catch (Exception e) {
        return null;
    }
    return bi;
}
}

```

## Κλάση SMOFrame

```
package textdatamining;

import java.awt.Graphics2D;
import java.awt.Image;
import java.awt.RenderingHints;
import java.awt.Toolkit;
import java.awt.image.BufferedImage;
import java.io.FileNotFoundException;
import java.io.FileReader;
import java.io.IOException;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.swing.ImageIcon;
import javax.swing.JLabel;
import weka.classifiers.Evaluation;
import weka.core.Instances;
import weka.classifiers.functions.SMO;
import weka.core.Utils;

/**
 *
 * @author spiros
 */
public class SMOFrame extends javax.swing.JFrame {

    String createModel = null;
    String path = null;
    Instances train;

    SMO smo = new SMO();

    public SMOFrame() {
        Image img = Toolkit.getDefaultToolkit().createImage("Blueprint
border.jpg");
        ImageIcon im = new ImageIcon();

        BufferedImage img1 = scaleImage(490, 520, "Blueprint border.jpg");

        im.setImage(img1);
        setContentPane(new JLabel(im));

        initComponents();
    }

    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated Code">
    private void initComponents() {

        ChooserFileARFF = new javax.swing.JFileChooser();
        ResultsDialog = new javax.swing.JDialog();
        CloseDialogButton = new javax.swing.JButton();
        jScrollPane1 = new javax.swing.JScrollPane();
        JTextArea1 = new javax.swing.JTextArea();
        jLabel1 = new javax.swing.JLabel();
        jLabel2 = new javax.swing.JLabel();
        jLabel3 = new javax.swing.JLabel();
    }
}
```

```

jLabel14 = new javax.swing.JLabel();
jLabel15 = new javax.swing.JLabel();
jLabel16 = new javax.swing.JLabel();
jLabel17 = new javax.swing.JLabel();
jLabel18 = new javax.swing.JLabel();
cField = new javax.swing.JTextField();
epsilonField = new javax.swing.JTextField();
KernelField = new javax.swing.JTextField();
NumFoldsField = new javax.swing.JTextField();
RandomSeedField = new javax.swing.JTextField();
ToleranceParametreField = new javax.swing.JTextField();
jLabel19 = new javax.swing.JLabel();
jLabel10 = new javax.swing.JLabel();
TrainingButton = new javax.swing.JButton();
OpenARFFFFile = new javax.swing.JButton();
ClassificationButton = new javax.swing.JButton();
CloseSMOFrame = new javax.swing.JButton();

ResultsDialog.setMinimumSize(new java.awt.Dimension(550, 430));

CloseDialogButton.setFont(new java.awt.Font("Trebuchet MS", 0, 12));
// NOI18N
CloseDialogButton.setText("Close");
CloseDialogButton.addActionListener(new
java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        CloseDialogButtonActionPerformed(evt);
    }
});

jTextArea1.setColumns(20);
jTextArea1.setFont(new java.awt.Font("Courier New", 0, 13)); //
NOI18N
jTextArea1.setRows(5);
jScrollPane1.setViewportView(jTextArea1);

javax.swing.GroupLayout ResultsDialogLayout = new
javax.swing.GroupLayout(ResultsDialog.getContentPane());
ResultsDialog.getContentPane().setLayout(ResultsDialogLayout);
ResultsDialogLayout.setHorizontalGroup(

ResultsDialogLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(ResultsDialogLayout.createSequentialGroup()
        .addContainerGap()
        .addGroup(ResultsDialogLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addComponent(CloseDialogButton,
                javax.swing.GroupLayout.PREFERRED_SIZE, 90,
                javax.swing.GroupLayout.PREFERRED_SIZE)
            .addComponent(jScrollPane1,
                javax.swing.GroupLayout.DEFAULT_SIZE, 530, Short.MAX_VALUE)
            .addGap(10, 10, 10))
        .addContainerGap(10, true));
ResultsDialogLayout.setVerticalGroup(

```

```

ResultsDialogLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
ResultsDialogLayout.createSequentialGroup()
    .addContainerGap()
    .addComponent(jScrollPane1,
javax.swing.GroupLayout.DEFAULT_SIZE, 367, Short.MAX_VALUE)
    .addGap(18, 18, 18)
    .addComponent(closeDialogButton)
    .addContainerGap()
);

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);

jLabel11.setFont(new java.awt.Font("Trebuchet MS", 1, 18)); // NOI18N
jLabel11.setText("Classification in Legal Texts");

jLabel12.setFont(new java.awt.Font("Trebuchet MS", 1, 14)); // NOI18N
jLabel12.setText("Create a SMO Model");

jLabel13.setFont(new java.awt.Font("Trebuchet MS", 0, 13)); // NOI18N
jLabel13.setText("c :");

jLabel14.setFont(new java.awt.Font("Trebuchet MS", 0, 13)); // NOI18N
jLabel14.setText("epsilon :");

jLabel15.setFont(new java.awt.Font("Trebuchet MS", 0, 13)); // NOI18N
jLabel15.setText("Kernel :");

jLabel16.setFont(new java.awt.Font("Trebuchet MS", 0, 13)); // NOI18N
jLabel16.setText("NumFolds :");

jLabel17.setFont(new java.awt.Font("Trebuchet MS", 0, 13)); // NOI18N
jLabel17.setText("RandomSeed :");

jLabel18.setFont(new java.awt.Font("Trebuchet MS", 0, 13)); // NOI18N
jLabel18.setText("ToleranceParametre :");

cField.setFont(new java.awt.Font("Trebuchet MS", 0, 12)); // NOI18N
cField.setText("1.0");

epsilonField.setFont(new java.awt.Font("Trebuchet MS", 0, 12)); //
NOI18N
epsilonField.setText("1.0E-12");

KernelField.setFont(new java.awt.Font("Trebuchet MS", 0, 12)); //
NOI18N
KernelField.setText("PolyKernel -C 250007 -E 1.0");

NumFoldsField.setFont(new java.awt.Font("Trebuchet MS", 0, 12)); //
NOI18N
NumFoldsField.setText("-1");

RandomSeedField.setFont(new java.awt.Font("Trebuchet MS", 0, 12));
// NOI18N
RandomSeedField.setText("1");

```

```

ToleranceParametreField.setFont(new java.awt.Font("Trebuchet MS", 0,
12)); // NOI18N
ToleranceParametreField.setText("0.001");

jLabel9.setFont(new java.awt.Font("Trebuchet MS", 0, 14)); // NOI18N
jLabel9.setText("For training push the button :");

jLabel10.setFont(new java.awt.Font("Trebuchet MS", 0, 14)); //
NOI18N
jLabel10.setText("For choose file push the button :");

TrainingButton.setFont(new java.awt.Font("Trebuchet MS", 0, 12)); //
NOI18N
TrainingButton.setText("Training");
TrainingButton.addActionListener(new java.awt.event.ActionListener()
{
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        TrainingButtonActionPerformed(evt);
    }
});

OpenARFFFFile.setFont(new java.awt.Font("Trebuchet MS", 0, 12)); //
NOI18N
OpenARFFFFile.setText("Open");
OpenARFFFFile.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        OpenARFFFFileActionPerformed(evt);
    }
});

ClassificationButton.setFont(new java.awt.Font("Trebuchet MS", 0,
12)); // NOI18N
ClassificationButton.setText("Classification");
ClassificationButton.addActionListener(new
java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        ClassificationButtonActionPerformed(evt);
    }
});

CloseSMOFrame.setFont(new java.awt.Font("Trebuchet MS", 0, 12)); //
NOI18N
CloseSMOFrame.setText("Close");
CloseSMOFrame.addActionListener(new java.awt.event.ActionListener()
{
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        CloseSMOFrameActionPerformed(evt);
    }
});

javax.swing.GroupLayout layout = new
javax.swing.GroupLayout(getContentPane());
getContentPane().setLayout(layout);
layout.setHorizontalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(layout.createSequentialGroup()
        .addContainerGap()

```



```
.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
```

```
    .addGroup(layout.createSequentialGroup()  
        .addComponent(jLabel12)  
        .addGap(0, 0, Short.MAX_VALUE))  
    .addGroup(layout.createSequentialGroup()  
        .addComponent(jLabel13)  
        .addGap(130, 130, 130)  
        .addComponent(cField))  
    .addGroup(layout.createSequentialGroup()  
        .addComponent(jLabel14)  
        .addGap(95, 95, 95)  
        .addComponent(epsilonField))  
    .addGroup(layout.createSequentialGroup()  
        .addComponent(jLabel15)  
        .addGap(100, 100, 100)  
        .addComponent(kernelField))  
    .addGroup(layout.createSequentialGroup()  
        .addComponent(jLabel16)  
        .addGap(80, 80, 80)  
        .addComponent(numFoldsField))  
    .addGroup(layout.createSequentialGroup()  
        .addComponent(jLabel17)  
        .addGap(61, 61, 61)  
        .addComponent(randomSeedField))  
    .addGroup(layout.createSequentialGroup()  
        .addComponent(jLabel18)  
        .addGap(18, 18, 18)  
        .addComponent(toleranceParametreField))  
    .addGroup(layout.createSequentialGroup()
```

```
.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)
```

```
    .addGroup(layout.createSequentialGroup()  
        .addComponent(jLabel19)  
        .addGap(37, 37, 37)  
        .addComponent(trainingButton,
```

```
javax.swing.GroupLayout.PREFERRED_SIZE, 90,
```

```
javax.swing.GroupLayout.PREFERRED_SIZE))
```

```
    .addGroup(layout.createSequentialGroup()  
        .addComponent(jLabel110)  
        .addGap(18, 18, 18)  
        .addComponent(openARFFFile,
```

```
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,  
Short.MAX_VALUE)))
```

```
    .addGap(18, 18, 18)
```

```
.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
```

```
    .addComponent(closeSMOFrame,
```

```
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,  
Short.MAX_VALUE)
```

```
    .addComponent(classificationButton,
```

```
javax.swing.GroupLayout.DEFAULT_SIZE, 135, Short.MAX_VALUE)))
```

```
    .addContainerGap()
```

```
    .addGroup(layout.createSequentialGroup()
```

```
        .addGap(107, 107, 107)
```

```

        .addComponent (jLabel1,
javax.swing.GroupLayout.PREFERRED_SIZE, 248,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addContainerGap (javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE)
    );
    layout.setVerticalGroup(

layout.createParallelGroup (javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup (layout.createSequentialGroup ()
        .addGap (33, 33, 33)
        .addComponent (jLabel1)
        .addGap (18, 18, 18)

.addGroup (layout.createParallelGroup (javax.swing.GroupLayout.Alignment.TRAILING)
    .addGroup (layout.createSequentialGroup ()
        .addComponent (jLabel2)
        .addGap (18, 18, 18)

.addGroup (layout.createParallelGroup (javax.swing.GroupLayout.Alignment.BASELINE)
    .addComponent (jLabel3)
    .addComponent (cField,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
        .addGap (18, 18, 18)

.addGroup (layout.createParallelGroup (javax.swing.GroupLayout.Alignment.BASELINE)
    .addComponent (jLabel4)
    .addComponent (epsilonField,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
        .addGap (18, 18, 18)

.addGroup (layout.createParallelGroup (javax.swing.GroupLayout.Alignment.BASELINE)
    .addComponent (jLabel5)
    .addComponent (KernelField,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
        .addGap (18, 18, 18)

.addGroup (layout.createParallelGroup (javax.swing.GroupLayout.Alignment.BASELINE)
    .addComponent (jLabel6)
    .addComponent (NumFoldsField,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
        .addGap (18, 18, 18)

.addGroup (layout.createParallelGroup (javax.swing.GroupLayout.Alignment.BASELINE)
    .addComponent (jLabel7)

```

```

        .addComponent(RandomSeedField,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
        .addGap(18, 18, 18)
        .addComponent(jLabel8)
        .addComponent(ToleranceParametreField,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

        .addGroup(layout.createSequentialGroup()
            .addGap(41, 41, 41)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

            .addComponent(jLabel9)
            .addComponent(TrainingButton))
            .addGap(18, 18, 18)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addComponent(jLabel10)
            .addComponent(OpenARFFFile))
        .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
layout.createSequentialGroup()

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
            .addComponent(ClassificationButton,
javax.swing.GroupLayout.PREFERRED_SIZE, 72,
javax.swing.GroupLayout.PREFERRED_SIZE))
            .addGap(18, 18, 18)
            .addComponent(ClosesMOFrame)
            .addContainerGap(45, Short.MAX_VALUE)
        );

pack();
} // </editor-fold>

private void TrainingButtonActionPerformed(java.awt.event.ActionEvent
evt) {
    createModel = "-C " + cField.getText() + " -L " +
ToleranceParametreField.getText() + " -P " + epsilonField.getText()
        + " -N 0" + " -V " + NumFoldsField.getText() + " -W" +
RandomSeedField.getText() + " -K weka.classifiers.function.supportVector." +
KernelField.getText();
    System.out.println(createModel);
    try {
        FileReader trainreader = new
FileReader("C:\\Users\\spiros\\Documents\\TrainingSet.arff");
        train = new Instances(trainreader);
        train.setClassIndex(train.numAttributes() - 1);

        smo.setOptions(Utils.splitOptions(createModel));
        smo.buildClassifier(train);

        Evaluation eval = new Evaluation(train);

```

```

        eval.evaluateModel(smo, train);

        JTextArea1.setText(eval.toSummaryString("\nTraining Results\n"
                                                + "=====\n",
false) + "\n\n"
                                                + eval.toClassDetailsString() + "\n\n" +
eval.toMatrixString());

        ResultsDialog.setVisible(true);

        System.out.println(eval.toSummaryString("\nTRAIN
Results\n=====\n", false));
        System.out.println(eval.toClassDetailsString());
        System.out.println(eval.confusionMatrix());
        System.out.println(eval.toMatrixString());

        trainreader.close();
    } catch (FileNotFoundException ex) {

Logger.getLogger(PerceptronFrame.class.getName()).log(Level.SEVERE, null,
ex);
    } catch (IOException ex) {

Logger.getLogger(PerceptronFrame.class.getName()).log(Level.SEVERE, null,
ex);
    } catch (Exception ex) {

Logger.getLogger(PerceptronFrame.class.getName()).log(Level.SEVERE, null,
ex);
    }
}

private void OpenARFFFileActionPerformed(java.awt.event.ActionEvent evt)
{
    if (ChooserFileARFF.showOpenDialog(this) ==
ChooserFileARFF.APPROVE_OPTION) {
        path = ChooserFileARFF.getSelectedFile().getAbsolutePath();
        System.out.println(path);
    }
}

private void
ClassificationButtonActionPerformed(java.awt.event.ActionEvent evt) {
    try {
        FileReader testreader = new FileReader(path);
        Instances test = new Instances(testreader);
        test.setClassIndex(test.numAttributes() - 1);

        Evaluation eval = new Evaluation(train);
        eval.evaluateModel(smo, test);
        JTextArea1.setText(eval.toSummaryString("\nSMO - Classification
Results\n"
                                                +
"=====\n", false) + "\n\n"
                                                + eval.toClassDetailsString() + "\n\n" +
eval.toMatrixString());

        ResultsDialog.setVisible(true);

```

```

        System.out.println(eval.toSummaryString("\nResults\n=====\n",
false));
        System.out.println(eval.toClassDetailsString());
        System.out.println(eval.confusionMatrix());
        System.out.println(eval.toMatrixString());
        testreader.close();
    } catch (FileNotFoundException ex) {
    } catch (IOException ex) {
    } catch (Exception ex) {

Logger.getLogger(PerceptronFrame.class.getName()).log(Level.SEVERE, null,
ex);
    }
}

private void CloseSMOFrameActionPerformed(java.awt.event.ActionEvent
evt) {
    this.dispose();
}

private void CloseDialogButtonActionPerformed(java.awt.event.ActionEvent
evt) {
    ResultsDialog.setVisible(false);
}

/**
 * @param args the command line arguments
 */
public void main() {
    /* Set the Nimbus look and feel */
    //<editor-fold defaultstate="collapsed" desc=" Look and feel setting
code (optional) ">
    /* If Nimbus (introduced in Java SE 6) is not available, stay with
the default look and feel.
    * For details see
http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
    */
    try {
        for (javax.swing.UIManager.LookAndFeelInfo info :
javax.swing.UIManager.getInstalledLookAndFeels()) {
            if ("Nimbus".equals(info.getName())) {
                javax.swing.UIManager.setLookAndFeel(info.getClassName());
                break;
            }
        }
    } catch (ClassNotFoundException ex) {

java.util.logging.Logger.getLogger(SMOFrame.class.getName()).log(java.util.l
ogging.Level.SEVERE, null, ex);
    } catch (InstantiationException ex) {

java.util.logging.Logger.getLogger(SMOFrame.class.getName()).log(java.util.l
ogging.Level.SEVERE, null, ex);
    } catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(SMOFrame.class.getName()).log(java.util.l
ogging.Level.SEVERE, null, ex);

```

```

    } catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(SMOFrame.class.getName()).log(java.util.l
ogging.Level.SEVERE, null, ex);
    }
    //</editor-fold>

    /* Create and display the form */
    java.awt.EventQueue.invokeLater(new Runnable() {
        public void run() {
            new SMOFrame().setVisible(true);
        }
    });
}

// Variables declaration - do not modify
private javax.swing.JFileChooser ChooserFileARFF;
private javax.swing.JButton ClassificationButton;
private javax.swing.JButton CloseDialogButton;
private javax.swing.JButton CloseSMOFrame;
private javax.swing.JTextField KernelField;
private javax.swing.JTextField NumFoldsField;
private javax.swing.JButton OpenARFFFile;
private javax.swing.JTextField RandomSeedField;
private javax.swing.JDialog ResultsDialog;
private javax.swing.JTextField ToleranceParametreField;
private javax.swing.JButton TrainingButton;
private javax.swing.JTextField cField;
private javax.swing.JTextField epsilonField;
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel10;
private javax.swing.JLabel jLabel2;
private javax.swing.JLabel jLabel3;
private javax.swing.JLabel jLabel4;
private javax.swing.JLabel jLabel5;
private javax.swing.JLabel jLabel6;
private javax.swing.JLabel jLabel7;
private javax.swing.JLabel jLabel8;
private javax.swing.JLabel jLabel9;
private javax.swing.JScrollPane jScrollPane1;
private javax.swing.JTextArea jTextArea1;
// End of variables declaration

BufferedImage scaleImage(int WIDTH, int HEIGHT, String filename) {
    BufferedImage bi = null;

    try {
        ImageIcon ii = new ImageIcon(filename); // path to image
        bi = new BufferedImage(WIDTH, HEIGHT,
BufferedImage.TYPE_INT_RGB);
        Graphics2D g2d = (Graphics2D) bi.createGraphics();
        g2d.addRenderingHints(new
RenderingHints(RenderingHints.KEY_RENDERING,
RenderingHints.VALUE_RENDER_QUALITY));
        g2d.drawImage(ii.getImage(), 0, 0, WIDTH, HEIGHT, null);
    } catch (Exception e) {
        return null;
    }
    return bi;
}

```

```
}  
}
```

## Κλάση SimpleLogisticFrame

```
package textdatamining;  
  
import java.awt.Graphics2D;  
import java.awt.Image;  
import java.awt.RenderingHints;  
import java.awt.Toolkit;  
import java.awt.image.BufferedImage;  
import java.io.FileNotFoundException;  
import java.io.FileReader;  
import java.io.IOException;  
import java.util.logging.Level;  
import java.util.logging.Logger;  
import javax.swing.ImageIcon;  
import javax.swing.JLabel;  
import weka.classifiers.Evaluation;  
import weka.classifiers.functions.SimpleLogistic;  
import weka.core.Instances;  
import weka.core.Utils;  
  
/**  
 *  
 * @author spiros  
 */  
public class SimpleLogisticFrame extends javax.swing.JFrame {  
  
    String createModel = null;  
    String path = null;  
    Instances train;  
  
    SimpleLogistic slog = new SimpleLogistic();  
  
    public SimpleLogisticFrame() {  
        Image img = Toolkit.getDefaultToolkit().createImage("Blueprint  
border.jpg");  
        ImageIcon im = new ImageIcon();  
  
        BufferedImage img1 = scaleImage(490, 465, "Blueprint border.jpg");  
  
        im.setImage(img1);  
        setContentPane(new JLabel(im));  
  
        initComponents();  
    }  
  
    @SuppressWarnings("unchecked")  
    // <editor-fold defaultstate="collapsed" desc="Generated Code">  
    private void initComponents() {  
  
        ChooserFileARFF = new javax.swing.JFileChooser();
```

```

ResultsDialog = new javax.swing.JDialog();
jButton1 = new javax.swing.JButton();
jScrollPane1 = new javax.swing.JScrollPane();
jTextAreal = new javax.swing.JTextArea();
jLabel11 = new javax.swing.JLabel();
jLabel12 = new javax.swing.JLabel();
jLabel13 = new javax.swing.JLabel();
jLabel14 = new javax.swing.JLabel();
jLabel15 = new javax.swing.JLabel();
jLabel16 = new javax.swing.JLabel();
jLabel17 = new javax.swing.JLabel();
jLabel18 = new javax.swing.JLabel();
TrainingButton = new javax.swing.JButton();
OpenARFFFFile = new javax.swing.JButton();
ClassificationButton = new javax.swing.JButton();
CloseSimpleLogisticButton = new javax.swing.JButton();
HeuristicStopField = new javax.swing.JTextField();
MaxBoostingIterationsField = new javax.swing.JTextField();
NumBoostingIterationsField = new javax.swing.JTextField();
WeightTrimBetaField = new javax.swing.JTextField();

ResultsDialog.setMinimumSize(new java.awt.Dimension(550, 430));

NOI18N
jButton1.setFont(new java.awt.Font("Trebuchet MS", 0, 12)); //
jButton1.setText("Close");
jButton1.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButton1ActionPerformed(evt);
    }
});

jTextAreal.setColumns(20);
jTextAreal.setFont(new java.awt.Font("Courier New", 0, 13)); //
NOI18N
jTextAreal.setRows(5);
jTextAreal.setMinimumSize(new java.awt.Dimension(550, 430));
jScrollPane1.setViewportView(jTextAreal);

javax.swing.GroupLayout ResultsDialogLayout = new
javax.swing.GroupLayout(ResultsDialog.getContentPane());
ResultsDialog.getContentPane().setLayout(ResultsDialogLayout);
ResultsDialogLayout.setHorizontalGroup(

ResultsDialogLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(ResultsDialogLayout.createSequentialGroup()
        .addContainerGap()

    .addGroup(ResultsDialogLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
ResultsDialogLayout.createSequentialGroup()
        .createSequentialGroup()
            .addGap(0, 0, Short.MAX_VALUE)
            .addComponent(jButton1,
javax.swing.GroupLayout.PREFERRED_SIZE, 90,
javax.swing.GroupLayout.PREFERRED_SIZE))
        .addComponent(jScrollPane1,
javax.swing.GroupLayout.DEFAULT_SIZE, 530, Short.MAX_VALUE))

```



```

        .addContainerGap()
    );
    ResultsDialogLayout.setVerticalGroup(

ResultsDialogLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LE
ADING)
        .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
ResultsDialogLayout.createSequentialGroup()
        .addContainerGap()
        .addComponent(jScrollPane1,
javax.swing.GroupLayout.DEFAULT_SIZE, 367, Short.MAX_VALUE)
        .addGap(18, 18, 18)
        .addComponent(jButton1)
        .addContainerGap()
    );

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);

jLabel1.setFont(new java.awt.Font("Trebuchet MS", 1, 18)); // NOI18N
jLabel1.setText("Classification in Legal Texts");

jLabel2.setFont(new java.awt.Font("Trebuchet MS", 1, 14)); // NOI18N
jLabel2.setText("Create o SimpleLogistic Model");

jLabel3.setFont(new java.awt.Font("Trebuchet MS", 0, 13)); // NOI18N
jLabel3.setText("HeuristicStop :");

jLabel4.setFont(new java.awt.Font("Trebuchet MS", 0, 13)); // NOI18N
jLabel4.setText("MaxBoostingIterations :");

jLabel5.setFont(new java.awt.Font("Trebuchet MS", 0, 13)); // NOI18N
jLabel5.setText("NumBoostingIterations :");

jLabel6.setFont(new java.awt.Font("Trebuchet MS", 0, 13)); // NOI18N
jLabel6.setText("WeightTrimBeta :");

jLabel7.setFont(new java.awt.Font("Trebuchet MS", 0, 14)); // NOI18N
jLabel7.setText("For training push the butoon :");

jLabel8.setFont(new java.awt.Font("Trebuchet MS", 0, 14)); // NOI18N
jLabel8.setText("For choose file push the button :");

TrainingButton.setFont(new java.awt.Font("Trebuchet MS", 0, 12)); //
NOI18N
TrainingButton.setText("Training");
TrainingButton.addActionListener(new java.awt.event.ActionListener()
{
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        TrainingButtonActionPerformed(evt);
    }
});

OpenARFFFFile.setFont(new java.awt.Font("Trebuchet MS", 0, 12)); //
NOI18N
OpenARFFFFile.setText("Open");
OpenARFFFFile.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        OpenARFFFFileActionPerformed(evt);
    }
}

```



```

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)
    .addComponent(jLabel12)
    .addGroup(layout.createSequentialGroup()

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, NG)
    .addComponent(jLabel4)
    .addComponent(jLabel6)
    .addComponent(jLabel3))
    .addGap(18, 18, 18)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, NG, false)
    .addComponent(WeightTrimBetaField,
javax.swing.GroupLayout.DEFAULT_SIZE, 312, Short.MAX_VALUE)

.addComponent(MaxBoostingIterationsField)
    .addComponent(HeuristicStopField))
    .addGroup(layout.createSequentialGroup()

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, NG)
    .addGroup(layout.createSequentialGroup()
        .addComponent(jLabel18)
        .addGap(18, 18, 18)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, NG, false)
    .addComponent(TrainingButton,
javax.swing.GroupLayout.DEFAULT_SIZE, 90, Short.MAX_VALUE)
    .addComponent(OpenARFFFFile,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
    .addComponent(jLabel17))
    .addGap(18, 18, 18)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, NG)
    .addComponent(CloseSimpleLogisticButton,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
    .addComponent(ClassificationButton,
javax.swing.GroupLayout.DEFAULT_SIZE, 134, Short.MAX_VALUE)))
    .addGap(0, 0, Short.MAX_VALUE))
    .addContainerGap()
);
layout.setVerticalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(layout.createSequentialGroup()
        .addGap(26, 26, 26)
        .addComponent(jLabel11,
javax.swing.GroupLayout.PREFERRED_SIZE, 22,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addGap(30, 30, 30)
        .addComponent(jLabel12)
        .addGap(36, 36, 36)

```

```

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
    .addComponent(jLabel3)
    .addComponent(HeuristicStopField,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
    .addGap(18, 18, 18)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
    .addComponent(jLabel4)
    .addComponent(MaxBoostingIterationsField,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
    .addGap(18, 18, 18)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
    .addComponent(jLabel5)
    .addComponent(NumBoostingIterationsField,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
    .addGap(18, 18, 18)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
    .addComponent(jLabel6)
    .addComponent(WeightTrimBetaField,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(layout.createSequentialGroup()
        .addGap(36, 36, 36)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
    .addComponent(jLabel7)
    .addComponent(TrainingButton))
    .addGap(18, 18, 18)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
    .addComponent(jLabel8)
    .addComponent(OpenARFFFile)))
    .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
layout.createSequentialGroup()

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
    .addComponent(ClassificationButton,
javax.swing.GroupLayout.PREFERRED_SIZE, 72,
javax.swing.GroupLayout.PREFERRED_SIZE))
    .addGap(18, 18, 18)

```

```

        .addComponent(CloseSimpleLogisticButton)
        .addContainerGap(54, Short.MAX_VALUE)
    );

    pack();
} // </editor-fold>

private void TrainingButtonActionPerformed(java.awt.event.ActionEvent
evt) {
    createModel = "-I " + NumBoostingIterationsField.getText() + " -M "
+ MaxBoostingIterationsField.getText() + " -H "
        + HeuristicStopField.getText();
    System.out.println(createModel);
    try {
        FileReader trainreader = new
FileReader("C:\\Users\\spiros\\Documents\\TrainingSet.arff");
        train = new Instances(trainreader);
        train.setClassIndex(train.numAttributes() - 1);

        slog.setOptions(Utils.splitOptions(createModel));
        slog.buildClassifier(train);

        Evaluation eval = new Evaluation(train);
        eval.evaluateModel(slog, train);

        JTextArea1.setText(eval.toSummaryString("\nTraining Results\n"
                                                + "=====\n",
false) + "\n\n"
                                                + eval.toClassDetailsString() + "\n\n" +
eval.toMatrixString());

        ResultsDialog.setVisible(true);

        System.out.println(eval.toSummaryString("\nTRAIN
Results\n=====\n", false));
        System.out.println(eval.toClassDetailsString());
        System.out.println(eval.confusionMatrix());
        System.out.println(eval.toMatrixString());

        trainreader.close();
    } catch (FileNotFoundException ex) {

Logger.getLogger(PerceptronFrame.class.getName()).log(Level.SEVERE, null,
ex);
        } catch (IOException ex) {

Logger.getLogger(PerceptronFrame.class.getName()).log(Level.SEVERE, null,
ex);
        } catch (Exception ex) {

Logger.getLogger(PerceptronFrame.class.getName()).log(Level.SEVERE, null,
ex);
        }
    }

private void OpenARFFFileActionPerformed(java.awt.event.ActionEvent evt)
{
    if (ChooserFileARFF.showOpenDialog(this) ==
ChooserFileARFF.APPROVE_OPTION) {

```

```

        path = ChooserFileARFF.getSelectedFile().getAbsolutePath();
        System.out.println(path);
    }
}

private void
ClassificationButtonActionPerformed(java.awt.event.ActionEvent evt) {
    try {
        FileReader testreader = new FileReader(path);
        Instances test = new Instances(testreader);
        test.setClassIndex(test.numAttributes() - 1);

        Evaluation eval = new Evaluation(train);
        eval.evaluateModel(slog, test);
        JTextArea1.setText(eval.toSummaryString("\nSimple Logistic -
Classification Results\n"
+
"=====\n", false) + "\n\n"
+ eval.toClassDetailsString() + "\n\n" +
eval.toMatrixString());

        ResultsDialog.setVisible(true);

        System.out.println(eval.toSummaryString("\nResults\n=====\n",
false));
        System.out.println(eval.toClassDetailsString());
        System.out.println(eval.confusionMatrix());
        System.out.println(eval.toMatrixString());
        testreader.close();
    } catch (FileNotFoundException ex) {
    } catch (IOException ex) {
    } catch (Exception ex) {

Logger.getLogger(PerceptronFrame.class.getName()).log(Level.SEVERE, null,
ex);
    }
}

private void
CloseSimpleLogisticButtonActionPerformed(java.awt.event.ActionEvent evt) {
    this.dispose();
}

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
    ResultsDialog.setVisible(false);
}

/**
 * @param args the command line arguments
 */
public void main() {
    /* Set the Nimbus look and feel */
    //<editor-fold defaultstate="collapsed" desc=" Look and feel setting
code (optional) ">
    /* If Nimbus (introduced in Java SE 6) is not available, stay with
the default look and feel.
    * For details see
http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
    */
}

```

```

        try {
            for (javax.swing.UIManager.LookAndFeelInfo info :
                javax.swing.UIManager.getInstalledLookAndFeels()) {
                if ("Nimbus".equals(info.getName())) {
                    javax.swing.UIManager.setLookAndFeel(info.getClassName());
                    break;
                }
            }
        } catch (ClassNotFoundException ex) {
            java.util.logging.Logger.getLogger(SimpleLogisticFrame.class.getName()).log(
                java.util.logging.Level.SEVERE, null, ex);
        } catch (InstantiationException ex) {
            java.util.logging.Logger.getLogger(SimpleLogisticFrame.class.getName()).log(
                java.util.logging.Level.SEVERE, null, ex);
        } catch (IllegalAccessException ex) {
            java.util.logging.Logger.getLogger(SimpleLogisticFrame.class.getName()).log(
                java.util.logging.Level.SEVERE, null, ex);
        } catch (javax.swing.UnsupportedLookAndFeelException ex) {
            java.util.logging.Logger.getLogger(SimpleLogisticFrame.class.getName()).log(
                java.util.logging.Level.SEVERE, null, ex);
        }
    }
}

//</editor-fold>

/* Create and display the form */
java.awt.EventQueue.invokeLater(new Runnable() {
    public void run() {
        new SimpleLogisticFrame().setVisible(true);
    }
});
}

// Variables declaration - do not modify
private javax.swing.JFileChooser ChooserFileARFF;
private javax.swing.JButton ClassificationButton;
private javax.swing.JButton CloseSimpleLogisticButton;
private javax.swing.JTextField HeuristicStopField;
private javax.swing.JTextField MaxBoostingIterationsField;
private javax.swing.JTextField NumBoostingIterationsField;
private javax.swing.JButton OpenARFFFile;
private javax.swing.JDialog ResultsDialog;
private javax.swing.JButton TrainingButton;
private javax.swing.JTextField WeightTrimBetaField;
private javax.swing.JButton jButton1;
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel2;
private javax.swing.JLabel jLabel3;
private javax.swing.JLabel jLabel4;
private javax.swing.JLabel jLabel5;
private javax.swing.JLabel jLabel6;
private javax.swing.JLabel jLabel7;
private javax.swing.JLabel jLabel8;
private javax.swing.JScrollPane jScrollPane1;
private javax.swing.JTextArea jTextArea1;
// End of variables declaration

```

```

BufferedImage scaleImage(int WIDTH, int HEIGHT, String filename) {
    BufferedImage bi = null;

    try {
        ImageIcon ii = new ImageIcon(filename); // path to image
        bi = new BufferedImage(WIDTH, HEIGHT,
BufferedImage.TYPE_INT_RGB);
        Graphics2D g2d = (Graphics2D) bi.createGraphics();
        g2d.addRenderingHints(new
RenderingHints(RenderingHints.KEY_RENDERING,
RenderingHints.VALUE_RENDER_QUALITY));
        g2d.drawImage(ii.getImage(), 0, 0, WIDTH, HEIGHT, null);
    } catch (Exception e) {
        return null;
    }
    return bi;
}
}

```

## ΠΑΡΑΡΤΗΜΑ II : ΑΠΟΤΕΛΕΣΜΑΤΑ ΚΑΤΗΓΟΡΙΟΠΟΙΗΣΗΣ ΝΟΜΙΚΩΝ ΚΕΙΜΕΝΩΝ

---

Παρακάτω παρουσιάζονται τα αποτελέσματα του συστήματος κατά την κατηγοριοποίηση των νομικών αποφάσεων με τα μοντέλα που χρησιμοποιήθηκαν.

### Αποτελέσματα 10-fold CrossValidation στο WEKA

#### Μοντέλο MytliLayerPerceptron

Correctly Classified Instances	295	94.2492 %
Incorrectly Classified Instances	18	5.7508 %
Kappa statistic	0.9134	
Mean absolute error	0.0407	
Root mean squared error	0.1529	
Relative absolute error	9.2174 %	
Root relative squared error	32.5232 %	
Total Number of Instances	313	



=== Detailed Accuracy By Class ===

Class	TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area
NOMOY-3068	0.933	0.04	0.902	0.933	0.917	0.987
POINIKES	0.947	0.015	0.973	0.947	0.96	0.987
POLITIKES	0.946	0.03	0.946	0.946	0.946	0.991
Weighted Avg.	0.942	0.027	0.943	0.942	0.943	0.988

=== Confusion Matrix ===

```
a  b  c  <-- classified as
83  1  5 |  a = NOMOY-3068
 5 107  1 |  b = POINIKES
 4   2 105 |  c = POLITIKES
```

## **Μοντέλο RBFNetwork**

Correctly Classified Instances	301	96.1661 %
Incorrectly Classified Instances	12	3.8339 %
Kappa statistic	0.9422	
Mean absolute error	0.039	
Root mean squared error	0.1505	
Relative absolute error	8.8221 %	
Root relative squared error	32.0089 %	
Total Number of Instances	313	

=== Detailed Accuracy By Class ===

Class	TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area
NOMOY-3068	0.955	0.031	0.924	0.955	0.939	0.963
POINIKES	0.956	0.01	0.982	0.956	0.969	0.988
POLITIKES	0.973	0.015	0.973	0.973	0.973	0.982
Weighted Avg.	0.962	0.018	0.962	0.962	0.962	0.979

=== Confusion Matrix ===

```
a  b  c  <-- classified as
85  1  3 |  a = NOMOY-3068
```

```

5 108 0 | b = POINIKES
2 1 108 | c = POLITIKES

```

### Μοντέλο SMO

```

Correctly Classified Instances      293      93.6102 %
Incorrectly Classified Instances    20        6.3898 %
Kappa statistic                    0.904
Mean absolute error                 0.2386
Root mean squared error            0.3003
Relative absolute error            53.9612 %
Root relative squared error        63.8686 %
Total Number of Instances          313

```

=== Detailed Accuracy By Class ===

Class	TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area
NOMOY-3068	0.966	0.058	0.869	0.966	0.915	0.959
POINIKES	0.885	0.015	0.971	0.885	0.926	0.967
POLITIKES	0.964	0.02	0.964	0.964	0.964	0.979
Weighted Avg.	0.936	0.029	0.939	0.936	0.936	0.969

=== Confusion Matrix ===

```

a  b  c  <-- classified as
86  2  1 | a = NOMOY-3068
10 100 3 | b = POINIKES
3  1 107 | c = POLITIKES

```

### Μοντέλο SimpleLogistic

```

Correctly Classified Instances      306      97.7636 %
Incorrectly Classified Instances    7        2.2364 %
Kappa statistic                    0.9663
Mean absolute error                 0.0196

```

```

Root mean squared error          0.1018
Relative absolute error          4.4414 %
Root relative squared error      21.6521 %
Total Number of Instances       313

```

=== Detailed Accuracy By Class ===

Class	TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area
NOMOY-3068	0.966	0.013	0.966	0.966	0.966	0.998
POINIKES	0.982	0	1	0.982	0.991	0.995
POLITIKES	0.982	0.02	0.965	0.982	0.973	0.997
Weighted Avg.	0.978	0.011	0.978	0.978	0.978	0.997

=== Confusion Matrix ===

```

  a   b   c   <-- classified as
86   0   3 |   a = NOMOY-3068
 1 111   1 |   b = POINIKES
 2   0 109 |   c = POLITIKES

```

## Αποτελέσματα Εκπαίδευσης των μοντέλων

### Multilayer Perceptron

Training Results

=====

```

Correctly Classified Instances      311          99.361 %
Incorrectly Classified Instances    2            0.639 %
Kappa statistic                    0.9904
Mean absolute error                 0.0142
Root mean squared error             0.0758
Relative absolute error             3.2178 %
Root relative squared error         16.127 %
Total Number of Instances          313

```

=== Detailed Accuracy By Class ===

TP Rate	FP Rate	Precision	Recall	F-Measure	Class
---------	---------	-----------	--------	-----------	-------

0.989	0	1	0.989	0.994	NOMOY-3068
0.991	0	1	0.991	0.996	POINIKES
1	0.01	0.982	1	0.991	POLITIKES

=== Confusion Matrix ===

a	b	c	<-- classified as
88	0	1	a = NOMOY-3068
0	112	1	b = POINIKES
0	0	111	c = POLITIKES

## RBFNetwork

Training Results

=====

Correctly Classified Instances	304	97.1246 %
Incorrectly Classified Instances	9	2.8754 %
Kappa statistic	0.9567	
Mean absolute error	0.0296	
Root mean squared error	0.1232	
Relative absolute error	6.7037 %	
Root relative squared error	26.1974 %	
Total Number of Instances	313	

=== Detailed Accuracy By Class ===

TP Rate	FP Rate	Precision	Recall	F-Measure	Class
0.966	0.027	0.935	0.966	0.95	NOMOY-3068
0.965	0.005	0.991	0.965	0.978	POINIKES
0.982	0.01	0.982	0.982	0.982	POLITIKES

=== Confusion Matrix ===

a	b	c	<-- classified as
86	1	2	a = NOMOY-3068
4	109	0	b = POINIKES

**SMO**

Training Results

=====

Correctly Classified Instances	295	94.2492 %
Incorrectly Classified Instances	18	5.7508 %
Kappa statistic	0.9136	
Mean absolute error	0.2364	
Root mean squared error	0.2971	
Relative absolute error	53.4826 %	
Root relative squared error	63.1981 %	
Total Number of Instances	313	

=== Detailed Accuracy By Class ===

TP Rate	FP Rate	Precision	Recall	F-Measure	Class
0.966	0.058	0.869	0.966	0.915	NOMOY-3068
0.894	0.01	0.981	0.894	0.935	POINIKES
0.973	0.015	0.973	0.973	0.973	POLITIKES

=== Confusion Matrix ===

a	b	c	<-- classified as
86	2	1	a = NOMOY-3068
10	101	2	b = POINIKES
3	0	108	c = POLITIKES

## Simple Logistic

Training Results

=====

Correctly Classified Instances	312	99.6805 %
Incorrectly Classified Instances	1	0.3195 %
Kappa statistic	0.9952	
Mean absolute error	0.0044	
Root mean squared error	0.0396	
Relative absolute error	0.9932 %	
Root relative squared error	8.414 %	
Total Number of Instances	313	

=== Detailed Accuracy By Class ===

TP Rate	FP Rate	Precision	Recall	F-Measure	Class
0.989	0	1	0.989	0.994	NOMOY-3068
1	0	1	1	1	POINIKES
1	0.005	0.991	1	0.996	POLITIKES

=== Confusion Matrix ===

a	b	c	<-- classified as
88	0	1	a = NOMOY-3068
0	113	0	b = POINIKES
0	0	111	c = POLITIKES

## Αποτελέσματα Ελέγχου/Δοκιμής των μοντέλων

### MultiLayerPerceptron

#### 1. Ποινικές Αποφάσεις

MultiLayer Perceptron - Classification Results

=====

Correctly Classified Instances	29	96.6667 %
Incorrectly Classified Instances	1	3.3333 %
Kappa statistic	0	
Mean absolute error	0.016	
Root mean squared error	0.1056	
Relative absolute error	3.763 %	
Root relative squared error	23.3138 %	
Total Number of Instances	30	

=== Detailed Accuracy By Class ===

TP Rate	FP Rate	Precision	Recall	F-Measure	Class
0	0.033	0	0	0	NOMOY-3068
0.967	0	1	0.967	0.983	POINIKES
0	0	0	0	0	POLITIKES

=== Confusion Matrix ===

a	b	c	<-- classified as
0	0	0	a = NOMOY-3068
1	29	0	b = POINIKES
0	0	0	c = POLITIKES

## 2. Πολιτικές Αποφάσεις

MultiLayer Perceptron - Classification Results

=====

Correctly Classified Instances	29	96.6667 %
Incorrectly Classified Instances	1	3.3333 %
Kappa statistic	0	
Mean absolute error	0.0154	
Root mean squared error	0.0913	
Relative absolute error	3.5811 %	
Root relative squared error	19.9482 %	
Total Number of Instances	30	

=== Detailed Accuracy By Class ===

TP Rate	FP Rate	Precision	Recall	F-Measure	Class
0	0.033	0	0	0	NOMOY-3068
0	0	0	0	0	POINIKES
0.967	0	1	0.967	0.983	POLITIKES

=== Confusion Matrix ===

```

a b c <-- classified as
0 0 0 | a = NOMOY-3068
0 0 0 | b = POINIKES
1 0 29 | c = POLITIKES

```

### 3. Αποφάσεις Νόμου 3068/2002

MultiLayer Perceptron - Classification Results

=====

Correctly Classified Instances	28	100	%
Incorrectly Classified Instances	0	0	%
Kappa statistic	1		
Mean absolute error	0.0124		
Root mean squared error	0.0186		
Relative absolute error	2.6069	%	
Root relative squared error	3.679	%	
Total Number of Instances	28		

=== Detailed Accuracy By Class ===

TP Rate	FP Rate	Precision	Recall	F-Measure	Class
1	0	1	1	1	NOMOY-3068
0	0	0	0	0	POINIKES
0	0	0	0	0	POLITIKES

=== Confusion Matrix ===

```

a b c <-- classified as
28 0 0 | a = NOMOY-3068
0 0 0 | b = POINIKES
0 0 0 | c = POLITIKES

```



## RBFNetwork

### 1. Ποινικές Αποφάσεις

RBFNetwork - Classification Results

=====

Correctly Classified Instances	30	100
%		
Incorrectly Classified Instances	0	0
%		
Kappa statistic	1	
Mean absolute error	0.0016	
Root mean squared error	0.0042	
Relative absolute error	0.387	%
Root relative squared error	0.9232	%
Total Number of Instances	30	

=== Detailed Accuracy By Class ===

TP Rate	FP Rate	Precision	Recall	F-Measure	Class
0	0	0	0	0	NOMOY-3068
1	0	1	1	1	POINIKES
0	0	0	0	0	POLITIKES

=== Confusion Matrix ===

a	b	c	<-- classified as
0	0	0	a = NOMOY-3068
0	30	0	b = POINIKES
0	0	0	c = POLITIKES

### 2. Πολιτικές Αποφάσεις

RBFNetwork - Classification Results

=====

Correctly Classified Instances	29	96.6667
%		
Incorrectly Classified Instances	1	3.3333
%		
Kappa statistic	0	
Mean absolute error	0.037	
Root mean squared error	0.1503	
Relative absolute error	8.5953	%
Root relative squared error	32.8465	%
Total Number of Instances	30	

=== Detailed Accuracy By Class ===

TP Rate	FP Rate	Precision	Recall	F-Measure	Class
---------	---------	-----------	--------	-----------	-------

0	0	0	0	0	NOMOY-3068
0	0.033	0	0	0	POINIKES
0.967	0	1	0.967	0.983	POLITIKES

```

=== Confusion Matrix ===
  a  b  c  <-- classified as
  0  0  0 | a = NOMOY-3068
  0  0  0 | b = POINIKES
  0  1 29 | c = POLITIKES

```

### 3. Αποφάσεις Νόμου 3068/2002

RBFNetwork - Classification Results

=====

Correctly Classified Instances	27	96.4286
%		
Incorrectly Classified Instances	1	3.5714
%		
Kappa statistic	0	
Mean absolute error	0.0326	
Root mean squared error	0.1245	
Relative absolute error	6.8346	%
Root relative squared error	24.6143	%
Total Number of Instances	28	

=== Detailed Accuracy By Class ===

TP Rate	FP Rate	Precision	Recall	F-Measure	Class
0.964	0	1	0.964	0.982	NOMOY-3068
0	0.036	0	0	0	POINIKES
0	0	0	0	0	POLITIKES

```

=== Confusion Matrix===
  a  b  c  <-- classified as
 27  1  0 | a = NOMOY-3068
  0  0  0 | b = POINIKES
  0  0  0 | c = POLITIKES

```

## SMO

### 1. Ποινικές Αποφάσεις

SMO - Classification Results

=====

Correctly Classified Instances	30	100	%
Incorrectly Classified Instances	0	0	%
Kappa statistic	1		
Mean absolute error	0.2222		
Root mean squared error	0.2722		
Relative absolute error	52.1452	%	
Root relative squared error	60.0934	%	
Total Number of Instances	30		

=== Detailed Accuracy By Class ===

TP Rate	FP Rate	Precision	Recall	F-Measure	Class
0	0	0	0	0	NOMOY-3068
1	0	1	1	1	POINIKES
0	0	0	0	0	POLITIKES

=== Confusion Matrix ===

```
a b c <-- classified as
0 0 0 | a = NOMOY-3068
0 30 0 | b = POINIKES
0 0 0 | c = POLITIKES
```

## 2. Πολιτικές Αποφάσεις

SMO - Classification Results

=====

Correctly Classified Instances	30	100	%
Incorrectly Classified Instances	0	0	%
Kappa statistic	1		
Mean absolute error	0.2222		
Root mean squared error	0.2722		
Relative absolute error	51.634	%	
Root relative squared error	59.4847	%	
Total Number of Instances	30		

=== Detailed Accuracy By Class ===

TP Rate	FP Rate	Precision	Recall	F-Measure	Class
0	0	0	0	0	NOMOY-3068
0	0	0	0	0	POINIKES
1	0	1	1	1	POLITIKES

```

=== Confusion Matrix ===
  a  b  c  <-- classified as
  0  0  0 | a = NOMOY-3068
  0  0  0 | b = POINIKES
  0  0 30 | c = POLITIKES

```

### 3. Αποφάσεις Νόμου 30-68/2002

SMO - Classification Results

=====

Correctly Classified Instances	27	96.4286 %
Incorrectly Classified Instances	1	3.5714 %
Kappa statistic	0	
Mean absolute error	0.2302	
Root mean squared error	0.2864	
Relative absolute error	48.2722 %	
Root relative squared error	56.6269 %	
Total Number of Instances	28	

=== Detailed Accuracy By Class ===

TP Rate	FP Rate	Precision	Recall	F-Measure	Class
0.964	0	1	0.964	0.982	NOMOY-3068
0	0.036	0	0	0	POINIKES
0	0	0	0	0	POLITIKES

```

=== Confusion Matrix ===
  a  b  c  <-- classified as
 27  1  0 | a = NOMOY-3068
  0  0  0 | b = POINIKES
  0  0  0 | c = POLITIKES

```

## Simple Logistic

### 1. Ποινικές Αποφάσεις

SimpleLogistic - Classification Results

```

=====
Correctly Classified Instances      30          100    %
Incorrectly Classified Instances    0           0    %
Kappa statistic                    1
Mean absolute error                0
Root mean squared error            0
Relative absolute error             0    %
Root relative squared error         0    %
Total Number of Instances          30

```

=== Detailed Accuracy By Class ===

TP Rate	FP Rate	Precision	Recall	F-Measure	Class
0	0	0	0	0	NOMOY-3068
1	0	1	1	1	POINIKES
0	0	0	0	0	POLITIKES

=== Confusion Matrix ===

```

a  b  c  <-- classified as
0  0  0  |  a = NOMOY-3068
0  30 0  |  b = POINIKES
0  0  0  |  c = POLITIKES

```

## 2. Πολιτικές Αποφύσεις

SimpleLogistic - Classification Results

```

=====
Correctly Classified Instances      30          100    %
Incorrectly Classified Instances    0           0    %
Kappa statistic                    1
Mean absolute error                0.0018
Root mean squared error            0.0088
Relative absolute error             0.4149    %
Root relative squared error         1.9278    %
Total Number of Instances          30

```

=== Detailed Accuracy By Class ===

TP Rate	FP Rate	Precision	Recall	F-Measure	Class
0	0	0	0	0	NOMOY-3068
0	0	0	0	0	POINIKES
1	0	1	1	1	POLITIKES

=== Confusion Matrix ===

```
a b c <-- classified as
0 0 0 | a = NOMOY-3068
0 0 0 | b = POINIKES
0 0 30 | c = POLITIKES
```

### **3. Αποφάσεις Νόμου 3068/2002**

SimpleLogistic - Classification Results

=====

Correctly Classified Instances	28	100	%
Incorrectly Classified Instances	0	0	%
Kappa statistic	1		
Mean absolute error	0.0008		
Root mean squared error	0.0027		
Relative absolute error	0.1617	%	
Root relative squared error	0.5369	%	
Total Number of Instances	28		

=== Detailed Accuracy By Class ===

TP Rate	FP Rate	Precision	Recall	F-Measure	Class
1	0	1	1	1	NOMOY-3068
0	0	0	0	0	POINIKES
0	0	0	0	0	POLITIKES

=== Confusion Matrix ===

```
a b c <-- classified as
28 0 0 | a = NOMOY-3068
0 0 0 | b = POINIKES
0 0 0 | c = POLITIKES
```



