Σχολή Θετικών Επιστημών και Τεχνολογίας

Μεταπτυχιακή Εξειδίκευση στα Πληροφοριακά Συστήματα

Διπλωματική Εργασία

Τεχνικές εξόρυξης δεδομένων και η εφαρμογή τους στα επιστημονικά ερευνητικά άρθρα

Γεωργία Κούτσου

Επιβλέπων καθηγητής: Γεώργιος Μαυρομμάτης

Αθήνα, Ιούλιος 2021

Τεχνικές εξόρυξης δεδομένων και η εφαρμογή τους στα επιστημονικά ερευνητικά άρθρα

Γεωργία Κούτσου

Επιτροπή Επίβλεψης Διπλωματικής Εργασίας

Επιβλέπων Καθηγητής:	Συνεπιβλέπων Καθηγητής:
Γεώργιος Μαυρομμάτης	Μιχαήλ Βασιλακόπουλος
Σ.Ε.Π. ΣΘΕΤ ΕΑΠ	Αν. Καθηγητής Πανεπιστημίου Θεσσαλίας

Αθήνα, Ιούλιος 2021

Ευχαριστίες

Αρχικά θα ήθελα ευχαριστήσω θερμά τον επιβλέποντα καθηγητή μου κ. Γ. Μαυρομμάτη για τη συνεχή του υποστήριξη, τις χρήσιμες συμβουλές του και γενικά για την πολύτιμη βοήθειά του όλο το διάστημα εκπόνησης της παρούσας εργασίας. Ακόμα, θα ήθελα να ευχαριστήσω ξεχωριστά το συνεπιβλέποντα κ. Μ. Βασιλακόπουλο για τις χρήσιμες παρατηρήσεις του οι οποίες συνέβαλαν στην επιτυχή ολοκλήρωση της εργασίας μου.

Ιδιαίτερα οφείλω να ευχαριστήσω τους συναδέλφους - συμφοιτητές μου της ομάδας «Μεταξύ μας» που συμμερίζονταν τις αγωνίες μου και συνέβαλαν με το δικό τους ξεχωριστό τρόπο στο να ολοκληρωθεί με επιτυχία αυτό το όμορφο ταξίδι, της φοίτησής μου στο ΕΑΠ.

Αφιερώνω την παρούσα εργασία στην οικογένειά μου, στο γιό μου Πάντιο, στην κόρη μου Ζωή και στο σύζυγό μου Κυριάκο τους οποίους οφείλω να ευχαριστήσω θερμά για την υπομονή τους, για την υποστήριξή τους και για την κατανόηση τους για τον προσωπικό χρόνο που τους στέρησα τα τελευταία χρόνια.

Τέλος οφείλω να ευχαριστήσω θερμά τη μητέρα μου Ζωή για την αδιάκοπη συμπαράστασή που μου πρόσφερε όλα αυτά τα χρόνια, χωρίς τη συμβολή της οποίας η επιτυχία μου θα ήταν ανέφικτη.

Περίληψη

Η μελέτη και η ανασκόπηση της επιστημονικής βιβλιογραφίας αποτελεί έναν από τους σημαντικότερους πυλώνες της επιστημονικής έρευνας. Η ανάλυση του σώματος της επιστημονικής βιβλιογραφίας συνεισφέρει, μεταξύ άλλων, στην κατανόηση των τάσεων της επιστήμης και στην εύρεση σχετικών με κάποιο τομέα άρθρων (Gulo, Rubio, Tabassum & Prado, 2015a).

Στόχος της παρούσας εργασίας ήταν να εφαρμοστούν οι κατάλληλες τεχνικές ανάλυσης κειμένου στην επιστημονική βιβλιογραφία και ειδικότερα σε ένα συγκεκριμένο επιστημονικό πεδίο, *το τεχνικό χρέος*, προκειμένου

- να ανακαλυφθεί νέα γνώση αναφορικά με αυτό το πεδίο,
- να εξαχθούν χρήσιμες πληροφορίες μέσα από μια διαδικασία αναγνώρισης και εξερεύνησης σημαντικών προτύπων,
- να ανακαλυφθούν οι τάσεις της έρευνας γύρω από αυτό το αντικείμενο,
- να διερευνηθούν τυχόν άλλοι τομείς ή κλάδοι στους οποίους επεκτείνεται,
- να δημιουργηθούν ένα ή περισσότερα μοντέλα ταξινόμησης για την κατηγοριοποίηση της έρευνας στο εν λόγω επιστημονικό πεδίο.

Στο πλαίσιο αυτό αρχικά κάναμε μια βιβλιογραφική έρευνα σχετικά με τις τεχνικές που εφαρμόζονται στα επιστημονικά ερευνητικά άρθρα. Στη συνέχεια εφαρμόσαμε αυτές τις τεχνικές στο σύνολο δεδομένων το οποίο αφορούσε το τεχνικό χρέος, έναν νέο τομέα της τεχνολογίας λογισμικού. Πιο συγκεκριμένα, εφαρμόσαμε από τους αλγορίθμους μη επιβλεπόμενης μηχανικής μάθησης το Topic Modeling, το K-means Clustering και το Hierarchical Clustering και από τους αλγόριθμους επιβλεπόμενης μηχανικής μάθησης εφαρμόσαμε το Decision Tree, το Support Vector Machine Linear, το K-Nearest Neighbor και το Naïve Bayes. Αντικείμενο εφαρμογής των αλγόριθμων ήταν δύο σύνολα δεδομένων, ένα που περιέχει τον όρο αναζήτησης “technical debt” OR “TD” (1ο dataset) και ένα που περιέχει μόνο τα σχετικά με το τεχνικό χρέος άρθρα (2ο dataset). Μια σημαντική παρατήρηση που προκύπτει, μεταξύ άλλων είναι, ότι το τεχνικό χρέος σχετίζεται με την ποιότητα του πηγαίου κώδικα και τις μετρικές ποιότητας. Επίσης η συσσώρευση τεχνικού χρέους και οι επιπτώσεις του είναι ένα ζήτημα για το οποίο γίνεται εκτενής αναφορά σε ένα μεγάλο αριθμό άρθρων. Επιπρόσθετα ένα άλλο ζήτημα που εντοπίζεται ότι απασχολεί τους ερευνητές είναι η διαχείρισή του. Αυτά προκύπτουν, όπως θα δούμε παρακάτω, από την k-

means συσταδοποίηση που εφαρμόζουμε στο 2ο dataset καθώς και από τα δέντρα απόφασης που προκύπτουν στο 2ο dataset του Σεναρίου 1.

Αρχικά ξεκινήσαμε με τη δημιουργία ενός νέφους λέξεων (word cloud) στο 1ο dataset προκειμένου να διαπιστώσουμε με οπτικό τρόπο τους όρους με τους οποίους συσχετίζεται το τεχνικό χρέος. Εν πρώτοις, διαπιστώσαμε ότι αυτό το dataset είχε μη σχετικά με το τεχνικό χρέος άρθρα. Με την τεχνική του topic modeling καταφέραμε να χωρίσουμε τα δεδομένα μας και να πετύχουμε έναν διαχωρισμό σε σχετικά και μη σχετικά με το τεχνικό χρέος άρθρα. Έτσι προέκυψε το 2ο dataset. Στη συνέχεια με τη k-means συσταδοποίηση στο 2ο dataset ομαδοποιήσαμε τα δεδομένα μας, τους προσδώσαμε μια ετικέτα κατηγορίας και με τη δημιουργία μοντέλων ταξινόμησης καταφέραμε να τα κατηγοριοποιήσουμε με ικανοποιητική ακρίβεια, 97,75%. Επιπρόσθετα με τη βοήθεια της ιεραρχικής συσταδοποίησης στο 2ο dataset καταφέραμε να εντοπίσουμε, μεταξύ άλλων, εκείνα που αφορούν τη συντήρηση λογισμικού.

Τέλος με την δική μας εμπειρική ομαδοποίηση στο 2ο dataset δημιουργήσαμε ένα μοντέλο ταξινόμησης το οποίο εντοπίζει άρθρα σχετικά με τη συντήρηση λογισμικού και των λοιπών κατηγοριών, που έχουμε εισάγει στο σύνολο δεδομένων μας, με αρκετά καλή ακρίβεια, 88,14%. Ωστόσο το μοντέλο αυτό γενικά είχε μικρότερη ακρίβεια σε σχέση με το μοντέλο στα οποίο η ετικέτα κατηγορίας προέκυψε από το k-means clustering στο 2ο dataset.

Λέξεις – Κλειδιά: Εξόρυξη Δεδομένων, Ανάλυση Κειμένου, Αλγόριθμοι Μηχανικής Μάθησης, Τεχνικό Χρέος

Abstract

The study and review of the scientific literature is one of the most important pillars of scientific research. The analysis of the corpus of the scientific literature contributes, among others, to the comprehension of the trends of science and to the identification of articles related to a specific field (Gulo, Rubio, Tabassum & Prado, 2015a).

The aim of this thesis was to apply the appropriate techniques of text analytics in the scientific literature and particularly in a specific scientific field, *technical debt*, in order to

- discover new knowledge concerning the specific field,
- extract useful information through a process of identifying and exploring important patterns,
- discover the research trends concerning this subject,
- investigate any other sectors or branches to which it extends,
- create one or more classification models to categorize research in that scientific field.

In this context we initially did a bibliographic research on the techniques applied in the scientific research articles. Then we applied these techniques to the technical debt dataset, a new area of software technology. More specifically, we applied Topic Modeling, K-means Clustering and Hierarchical Clustering from the unsupervised machine learning algorithms and the Decision Tree, Support Vector Machine Linear, K-Nearest Neighbor and Naïve Bayes from the supervised machine learning algorithms. The algorithms were applied to two sets of data, one containing the search term "technical debt" OR "TD" (1st dataset) and one containing only the articles related to technical debt (2nd dataset). An important observation that arises is that technical debt is related to source code quality and quality metrics. Also, the accumulation of technical debt and its effects is an issue that is extensively reported in a large number of articles. In addition, another issue that is noticed to concern researchers is its management. These result, as we will see, from the k-means clustering that we apply to the 2nd dataset as well as from the decision trees that result from the 2nd dataset of Scenario 1.

We first started by creating a word cloud in the 1st dataset in order to visually determine the terms to which the technical debt is related. First, we noticed that the dataset had non-

technical debt related articles. Using topic modeling technique we managed to separate our data and achieve a separation into articles related to and not related to technical debt. This is how the 2nd dataset created. Then applying k-means clustering technique in the 2nd dataset we grouped our data, we attached to them a category label and by creating classification models we managed to classify them with satisfactory accuracy, 97.75%. Also applying hierarchical clustering technique we managed to identify, among others, those related to software maintenance.

Finally, with our own empirical grouping, we created a classification model that identifies articles on software maintenance and other categories that we have inserted into our dataset, with good enough accuracy, 88.14%. However, these models were generally less accurate than the models in which the category label was derived from k-means clustering technique in the 2nd dataset.

Keywords: Data Mining, Text Analytics, Machine Learning Algorithms, Technical Debt

Πίνακας Περιεχομένων

ΕΥΡΕΤΗΡΙΟ ΕΙΚΟΝΩΝ

ΕΥΡΕΤΗΡΙΟ ΓΡΑΦΗΜΑΤΩΝ

ΕΥΡΕΤΗΡΙΟ ΠΙΝΑΚΩΝ

ΚΕΦΑΛΑΙΟ 1

1.1 Στόχος της εργασίας

Στόχος της εργασίας είναι μέσα από την εφαρμογή τεχνικών ανάλυσης κειμένου στην επιστημονική βιβλιογραφία και ειδικότερα σε ένα συγκεκριμένο επιστημονικό πεδίο, *το τεχνικό χρέος*, προκειμένου

- να ανακαλυφθεί νέα γνώση αναφορικά με αυτό το πεδίο,
- να εξαχθούν χρήσιμες πληροφορίες μέσα από μια διαδικασία αναγνώρισης και εξερεύνησης σημαντικών προτύπων,
- να ανακαλυφθούν οι τάσεις της έρευνας γύρω από αυτό το αντικείμενο,
- να διερευνηθούν τυχόν άλλοι τομείς ή κλάδοι στους οποίους επεκτείνεται,
- να δημιουργηθούν ένα ή περισσότερα μοντέλα ταξινόμησης για την κατηγοριοποίηση της έρευνας στο εν λόγω επιστημονικό πεδίο.

Εμείς αυτό που θέλουμε να διερευνήσουμε είναι η ανακάλυψη νέας γνώσης που θα προκύψει μέσα από την εφαρμογή τεχνικών εξόρυξης σε μια συλλογή κειμένων αξιοποιώντας την πληροφορία που προέρχεται από την περίληψη των επιστημονικών άρθρων. Η βασική μας επιδίωξη είναι η εύρεση ενός μοντέλου κατηγοριοποίησης (ή και περισσοτέρων) το οποίο θα αποτελέσει εργαλείο με το οποίο θα μπορέσει ένας ερευνητής να κατηγοριοποιήσει ένα σύνολο δεδομένων ώστε να εντοπίσει, να ανατρέξει και να μελετήσει το υποσύνολο που τον ενδιαφέρει.

Στη παρούσα διπλωματική θα εφαρμοστούν τεχνικές εξόρυξης γνώσης πάνω σε κείμενα επιστημονικών δημοσιεύσεων αυτού του πεδίου με σκοπό να εντοπιστούν τυχόν ομοιότητες, τάσεις, ομαδοποιήσεις, υποκατηγορίες με σκοπό την εύρεση νέας γνώσης. Θα εφαρμοστούν οι αλγόριθμοι μη επιβλεπόμενης μηχανικής μάθησης Topic Modeling, K-means Clustering και Hierarchical Clustering και οι αλγόριθμοι επιβλεπόμενης μάθησης Decision Tree, Support Vector Machine (linear), K-Nearest Neighbors και Naïve Bayes.

Πιο συγκεκριμένα θα εφαρμόσουμε τις τεχνικές ανάλυσης κειμένου και θα συγκρίνουμε τα ευρήματα κάθε τεχνικής τόσο μεταξύ τους όσο και με τα εμπειρικά δεδομένα στα οποία καταλήξαμε μετά από σχετική επεξεργασία του υπό ανάλυση συνόλου δεδομένων, προκειμένου να αξιολογήσουμε την αποτελεσματικότητά τους και να οδηγηθούμε στα

τελικά συμπεράσματα. Στη συνέχεια θα προτείνουμε ένα μοντέλο ταξινόμησης το οποίο θα μπορεί να αξιοποιήσει ένας ερευνητής ως εργαλείο για την περαιτέρω διερεύνηση του συγκεκριμένου επιστημονικού πεδίου.

1.2 Δομή της εργασίας

Το 1ο κεφάλαιο είναι εισαγωγικό. Αναλύεται ο στόχος της παρούσας εργασίας και παρουσιάζεται το βασικό αντικείμενό της.

Στο 2ο κεφάλαιο παρουσιάζεται όλο το θεωρητικό πλαίσιο της εργασίας. Παρουσιάζονται οι μέθοδοι ανάλυσης κειμένου, οι τεχνικές και τα μοντέλα ομαδοποίησης και κατηγοριοποίησης που εφαρμόζονται καθώς και οι μετρικές αξιολόγησής τους. Το κεφάλαιο ολοκληρώνεται με μια βιβλιογραφική ανασκόπηση στην εξόρυξη κειμένων στα ερευνητικά άρθρα.

Στο 3ο κεφάλαιο γίνεται μια σύντομη αναφορά στο θεωρητικό υπόβαθρο του τεχνικού χρέους που αποτελεί το βασικό αντικείμενο εφαρμογής αλγορίθμων εξόρυξης και παράλληλα τεκμηριώνεται η απόφαση επιλογής του συγκεκριμένου επιστημονικού πεδίου.

Στο κεφάλαιο 4 παρουσιάζονται τα εργαλεία που χρησιμοποιήθηκαν για την αντιμετώπιση του θέματος, η γλώσσα R και τα εργαλεία R-studio και JabRef.

Στο κεφάλαιο 5 παρουσιάζονται αναλυτικά και διεξοδικά η επεξεργασία του υλικού που επιλέχθηκε, οι τεχνικές και τα μοντέλα μηχανικής μάθησης που εφαρμόστηκαν. Οι βασικές τεχνικές που εφαρμόσαμε ήταν από την μη επιβλεπόμενη μάθηση τα Topic Modeling, K-means Clustering και Hierarchical Clustering και από την επιβλεπόμενη μάθηση οι Decision Tree, SVM (Linear), K – Nearest Neighbors και Naïve Bayes.

Στο κεφάλαιο 6 αναλύονται τα συμπεράσματα που προέκυψαν, συγκρίνονται τα αποτελέσματα των μοντέλων που εφαρμόστηκαν και προτείνονται προτάσεις βελτίωσης.

1.3 Συνοπτική περιγραφή των τεχνικών ανάλυσης κειμένου και των αλγορίθμων μηχανικής μάθησης που εφαρμόστηκαν

Στη παρούσα διπλωματική εφαρμόσαμε τεχνικές εξόρυξης γνώσης πάνω σε κείμενα επιστημονικών δημοσιεύσεων. Το πεδίο που επιλέχθηκε, όπως αναφέραμε ανωτέρω, είναι το αντικείμενο του τεχνικού χρέους όρος ο οποίος συνδέεται με τη συντήρηση λογισμικού στα έργα ανάπτυξης λογισμικού. Πιο συγκεκριμένα χρησιμοποιήσαμε από τους αλγόριθμους μη επιβλεπόμενης μηχανικής μάθησης τους εξής:

- Topic Modeling,
- K-means Clustering
- Hierarchical Clustering

και από τους αλγόριθμους επιβλεπόμενης μάθησης χρησιμοποιήσαμε τους παρακάτω:

- Decision Tree,
- Support Vector Machine (linear),
- K-Nearest Neighbors
- και Naïve Bayes.

Στο κεφάλαιο 5 περιγράφονται αναλυτικά οι αλγόριθμοι που εκτελέστηκαν και τα σχετικά αποτελέσματα που προέκυψαν ως χρήσιμη γνώση για το αντικείμενο αυτό. Ωστόσο θεωρούμε σημαντικό να παρουσιάσουμε συνοπτικά τα πειράματα που εκτελέσαμε.

Αρχικά, όπως παρουσιάζουμε στην ενότητα 5.3, κατεβάσαμε από το Scopus το σύνολο δεδομένων μας με 623 περιλήψεις το οποίο επεξεργαστήκαμε περαιτέρω αφαιρώντας διπλότυπα, ξενόγλωσσα και άρθρα χωρίς author. Το σύνολο που προέκυψε, το οποίο ονομάζουμε 1ο dataset, το επεξεργαστήκαμε στο εργαλείο JabRef και κάναμε μια δική μας εμπειρική ομαδοποίηση προκειμένου να την αξιοποιήσουμε στα μοντέλα ταξινόμησης που παρουσιάζουμε στο Σενάριο 2 της ενότητας 5.11. Επίσης με την ομαδοποίηση αυτή επιβεβαιώνουμε και τα αποτελέσματα του topic modeling της ενότητας 5.5.

Στην ενότητα 5.4 δημιουργούμε μέσω κώδικα ένα νέφος με τις λέξεις από τους τίτλους των άρθρων και διαπιστώνουμε ότι υπάρχουν μη σχετικά με το τεχνικό χρέος άρθρα.

Στην ενότητα 5.5 αξιοποιούμε την τεχνική του Topic Modeling και ομαδοποιούμε τα άρθρα με βάση το κυρίαρχο topic σε 6 ομάδες εκ των οποίων οι 3 είναι μόνο σχετικές με το τεχνικό χρέος τις οποίες κρατάμε και έτσι προκύπτει το 2ο dataset. Δηλαδή επιτυγχάνουμε να διαχωρίσουμε τα σχετικά από τα μη σχετικά τα οποία αφαιρούμε από το 1ο dataset. Το αποτέλεσμα του διαχωρισμού των άρθρων, σε σχετικά και μη, ταυτίζεται και με την εμπειρική ομαδοποίηση που κάναμε στην ενότητα 5.3 δηλαδή τα 3 σχετικά topics που βρήκαμε με το μοντέλο αυτό ταυτίζονται με τα άρθρα που εντοπίζουμε και εμείς στην ενότητα 5.3.

Στην ενότητα 5.6 επιχειρούμε να ομαδοποιήσουμε το 1ο dataset με την τεχνική του k-means clustering ώστε να διαχωρίσουμε τα σχετικά από τα μη σχετικά άρθρα. Εκτελέσαμε διαφορετικές δοκιμές με 4, 5, 6 και 7 συστάδες και με διαφορετικό weighting στους όρους του πίνακα Document Term Matrix. Ο αλγόριθμος δεν ήταν επιτυχής ώστε να τα ομαδοποιήσει όπως το topic modeling. Η δοκιμή έγινε με σκοπό να συγκρίνουμε τα αποτελέσματα αυτά με αυτά της ενότητας 5.5.

Από την ενότητα 5.7 μέχρι και την 5.11 παρουσιάζουμε τις δοκιμές που κάναμε στο 2ο dataset το οποίο ουσιαστικά είναι και αυτό που μας ενδιαφέρει. Στην ενότητα 5.7 εφαρμόζουμε k-means clustering και την ομαδοποίηση αυτή την κρατάμε ως ετικέτα κατηγορίας με σκοπό να ταξινομήσουμε το 2ο dataset (όπως παρουσιάζεται στο Σενάριο 1 της ενότητας 5.10).

Στην ενότητα 5.8 εφαρμόζουμε τεχνικές ιεραρχικής συσταδοποίησης στο 2ο dataset και επιτυγχάνουμε να χωρίσουμε τα άρθρα σε 6 συστάδες. Με τον αλγόριθμο αυτό σχηματίζονται, μεταξύ άλλων, κάποιες ενδιαφέρουσες συστάδες όπως πχ. μια που αφορά τη συντήρηση λογισμικού, μια σχετική με το αρχιτεκτονικό τεχνικό χρέος και μια που αναφέρεται στη ποιότητα του κώδικα.

Στην ενότητα 5.9 επιχειρήσαμε με την τεχνική του Topic Modeling να χωρίσουμε σε νέα topics το 2ο dataset με σκοπό να χρησιμοποιήσουμε αυτά τα labels ως ετικέτες κατηγορίας και στη συνέχεια να εκτελέσουμε τα 4 ανωτέρω μοντέλα ταξινόμησης. Τα μοντέλα αυτά είχαν μικρή ακρίβεια, περίπου 45 – 50%.

Στην ενότητα 5.10 εκτελούμε τα 4 μοντέλα ταξινόμησης στο 2ο dataset με ετικέτα κατηγορίας αυτή που προέκυψε από το k-means clustering της ενότητας 5.7. Την καλύτερη απόδοση είχε ο SVM linear με ακρίβεια 97,75 %.

Στην ενότητα 5.11 ολοκληρώνουμε τις δοκιμές εκτελώντας τα ίδια μοντέλα ταξινόμησης στο 2ο dataset αλλά με ετικέτα αυτή που προσδώσαμε εμείς εμπειρικά στο σύνολο δεδομένων μετά από την επεξεργασία του υλικού που παρουσιάζουμε στην ενότητα 5.3. Η μεγαλύτερη ακρίβεια που πετύχαμε ήταν 88,14%. Τα μοντέλα αυτά γενικά είχαν μικρότερη ακρίβεια σε σχέση με τα μοντέλα στα οποία η ετικέτα κατηγορίας προέκυψε από το k-means clustering.

ΚΕΦΑΛΑΙΟ 2

2.1 Μηχανική Μάθηση ή Εξόρυξη Δεδομένων

Αυτό που κάνει τη σημερινή εποχή μοναδική είναι ότι έχουμε εύκολη πρόσβαση σε μεγάλο όγκο δεδομένων. Μεγαλύτερα και περισσότερα σύνολα δεδομένων διαρκώς παράγονται από διάφορες πηγές και συσσωρεύονται ενώ παράλληλα είναι όλο και πιο εύκολα προσβάσιμα, για παράδειγμα μέσω αναζήτησης στον παγκόσμιο ιστό. Ο τομέας της επιστήμης που ασχολείται με την ανάπτυξη αλγορίθμων για τη μετατροπή δεδομένων σε σημαντική γνώση είναι γνωστός ως Μηχανική Μάθηση (Lantz, 2013).

Με άλλα λόγια, ο σκοπός της μηχανικής μάθησης είναι να χρησιμοποιεί αλγορίθμους για να ανακαλύψουμε γνώση, μέσα από σύνολα δεδομένων, την οποία μετέπειτα εφαρμόζουμε προκείμενου να λάβουμε τεκμηριωμένες αποφάσεις για το μέλλον (Nwanganga & Chapple, 2020). Εφαρμόζοντας δηλαδή διάφορους αλγορίθμους πάνω σε μεγάλα σύνολα δεδομένων επιτρέπουμε στις μηχανές να καταλαβαίνουν διάφορες καταστάσεις και βασισμένοι σε αυτές να λαμβάνουμε τις κατάλληλες αποφάσεις.

Η μηχανική μάθηση είναι χρήσιμη σε πολλούς και ετερογενείς τομείς της ανθρώπινης δραστηριότητας, ενδεικτικά αναφέρουμε (Nwanganga & Chapple, 2020):

- Τμηματοποίηση πελατών και προσδιορισμός μηνυμάτων μάρκετινγκ που θα προσελκύσουν διαφορετικές ομάδες πελατών
- Ανακάλυψη ανωμαλιών σε αρχεία καταγραφής συστημάτων και εφαρμογών, ενδεικτικά ενός ενδεχόμενου περιστατικού ασφάλειας στον κυβερνοχώρο
- Πρόβλεψη πωλήσεων προϊόντων με βάση τις συνθήκες της αγοράς και του περιβάλλοντος
- Καθορισμός τιμών συγκεκριμένων προϊόντων και αγαθών εκ των προτέρων με βάση την προβλεπόμενη ζήτηση
- Πρόβλεψη της επόμενης ταινίας που μπορεί να θέλει να παρακολουθήσει ένας πελάτης βάσει της προηγούμενης δραστηριότητάς του και των προτιμήσεων παρόμοιων πελατών.

Φυσικά, αυτά είναι μερικά μόνο παραδείγματα. Η μηχανική μάθηση μπορεί να φέρει αξία σχεδόν σε κάθε πεδίο όπου η ανακάλυψη προηγούμενων άγνωστων γνώσεων είναι χρήσιμη – άλλωστε δεν θα μπορούσαμε να σκεφτούμε ένα πεδίο όπου η γνώση δεν προσφέρει πλεονέκτημα.

Η μηχανική μάθηση είναι ένας ταχέως αναπτυσσόμενος κλάδος και μερικές κλασικές εφαρμογές μηχανικής μάθησης που σχετίζονται με την εξόρυξη δεδομένων είναι οι ακόλουθες (Han, Kamber & Pei, 2011):

- **Επιβλεπόμενη Μάθηση**: είναι σχετική με την κατηγοριοποίηση και η επίβλεψη προέρχεται από τα κατηγοριοποιημένα με ετικέτα παραδείγματα που βρίσκονται μέσα στο σύνολο εκπαίδευσης τα οποία επιβλέπουν τη μάθηση στο μοντέλο κατηγοριοποίησης
- **Μη επιβλεπόμενη Μάθηση**: είναι σχετική με τη συσταδοποίηση και η διαδικασία μάθησης είναι μη επιβλεπόμενη από τη στιγμή που τα παραδείγματα εισόδου δεν έχουν ετικέτα. Στην περίπτωση αυτή, όπου τα δεδομένα του συνόλου εκπαίδευσης δεν έχουν ετικέτα κατηγορίας, το μοντέλο αυτό δεν μπορεί να μας πει ποιο είναι το σημασιολογικό νόημα των συστάδων που δημιουργούνται.
- **Ημιεπιβλεπόμενη Μάθηση**: χρησιμοποιεί τεχνικές από τις δύο παραπάνω περιπτώσεις. Τα παραδείγματα με ετικέτα χρησιμοποιούνται για την εκμάθηση του μοντέλου ενώ τα παραδείγματα χωρίς ετικέτα κατηγορίας χρησιμεύουν για τη βελτίωση του μοντέλου.
- **Ενεργή Μάθηση**: είναι μια διαδικασία στην οποία οι χρήστες παίζουν ενεργό ρόλο στη διαδικασία μάθησης. Στη περίπτωση αυτή ζητείται από τον χρήστη να προσδώσει στο παράδειγμα ετικέτα κατηγορίας. Ο σκοπός είναι η βελτίωση του μοντέλου μέσα από τη γνώση που αποκτάται από τους χρήστες.

Ενώ σύμφωνα με τους Award & Khanna (2015) οι αλγόριθμοι μηχανικής μάθησης χωρίζονται σε 6 μεγάλες κατηγορίες:

- **η επιβλεπόμενη μάθηση** (Supervised Learning),
- **η μη επιβλεπόμενη μάθηση** (Unsupervised Learning)
- **ημι-επιβλεπόμενη μάθηση** (Semi-supervised Learning

- **η ενισχυτική μάθηση** (Reinforcement Learning)
- **η μεταγωγική μάθηση** (Transductive Learning)
- **και το επαγωγικό συμπέρασμα** (Inductive inference)

Η επιβλεπόμενη μάθηση έχει σαν κύριες μεθόδους την κατηγοριοποίηση (classification) και την παλινδρόμηση (regression), ενώ η μη επιβλεπόμενη μάθηση έχει το μετασχηματισμό και τη συσταδοποίηση (clustering). Και στις δύο περιπτώσεις, τα δεδομένα εισόδου πρέπει να έχουν σωστή αναπαράσταση για να μπορεί να τα καταλάβει ένας υπολογιστής. Η ενισχυτική μάθηση ασχολείται κυρίως με διάφορες οντότητες που ονομάζονται πράκτορες, οι οποίοι παίρνουν τις αποφάσεις τους από το περιβάλλον, με σκοπό να εκτελέσουν κάποια ενέργεια.

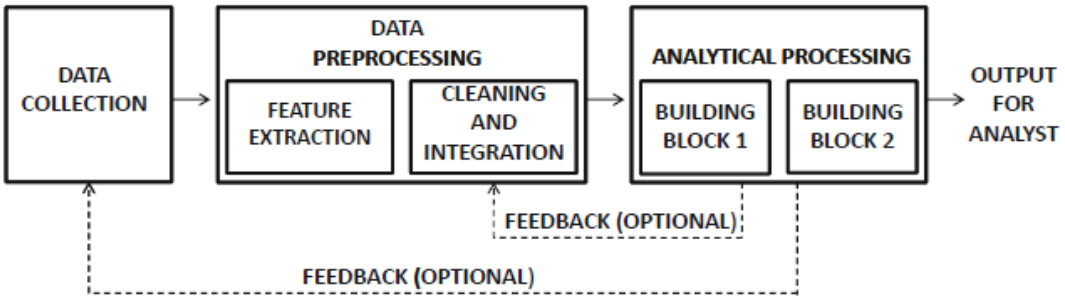
Από την άλλη πλευρά σχεδόν όλα τα αυτοματοποιημένα συστήματα δημιουργούν κάποια μορφή δεδομένων είτε για διαγνωστικούς σκοπούς είτε για σκοπούς ανάλυσης. Μερικά παραδείγματα διαφορετικών πηγών δεδομένων είναι, μεταξύ άλλων, το διαδίκτυο και οι οικονομικές συναλλαγές. Η Εξόρυξη Δεδομένων (ΕΔ) είναι η μελέτη συλλογής, καθαρισμού, επεξεργασίας, ανάλυσης και απόκτησης χρήσιμων πληροφοριών από τα δεδομένα (Aggarwal, 2015). Ο βασικός στόχος δηλαδή της Εξόρυξης Δεδομένων είναι η εξαγωγή μη τετριμμένης, προηγούμενα άγνωστης και πιθανά χρήσιμης πληροφορίας ή προτύπων από το σύνολο των δεδομένων (Βερύκιος, Καγκλής & Σταυρόπουλος, 2015).

Τα ανεπεξέργαστα δεδομένα μπορεί να είναι αυθαίρετα, μη δομημένα ή ακόμη και σε μορφή που δεν είναι αμέσως κατάλληλη για αυτοματοποιημένη επεξεργασία. Για παράδειγμα, τα δεδομένα που συλλέγονται με μη αυτόματο τρόπο μπορεί να προέρχονται από ετερογενείς πηγές σε διαφορετικές μορφές και ωστόσο πρέπει κατά κάποιο τρόπο να υποβληθούν σε επεξεργασία από ένα αυτοματοποιημένο πρόγραμμα υπολογιστή για να μας παρέχουν τις πληροφορίες που αναζητούμε. Για την αντιμετώπιση αυτού του ζητήματος, οι αναλυτές εξόρυξης δεδομένων χρησιμοποιούν μια σειρά βημάτων επεξεργασίας, όπου τα ακατέργαστα δεδομένα συλλέγονται, καθαρίζονται και μετατρέπονται σε τυποποιημένη μορφή (Aggarwal, 2015).

Η εξόρυξη δεδομένων γενικά περιλαμβάνει τις ακόλουθες φάσεις:

- Συλλογή των δεδομένων

- Εξαγωγή χαρακτηριστικών και προ-επεξεργασία των δεδομένων
- Αναλυτική επεξεργασία δεδομένων και εφαρμογή αλγορίθμων



Εικόνα 1. Ροή επεξεργασίας δεδομένων (Aggarwal, 2015)

Στην εξόρυξη δεδομένων χρησιμοποιούνται διάφορες τεχνικές και αλγόριθμοι, ενδεικτικά αναφέρουμε τους παρακάτω:

- Κατηγοριοποίηση (Classification)
- Συσταδοποίηση (Clustering)
- Παλινδρόμηση (Regression)
- Κανόνες συσχέτισης (Association Rules)
- Δέντρα αποφάσεων (Decision Trees)
- Μέθοδος πλησιέστερου γείτονα (Nearest Neighbor method) κα.
- Τεχνητή Νοημοσύνη (Artificial Intelligence)
- Νευρωνικά Δίκτυα (Neural Networks)
- Γενετικοί αλγόριθμοι (Genetic Algorithms)

2.1.1 Σύγκριση Μηχανικής Μάθησης με Εξόρυξη Δεδομένων

Η μηχανική μάθηση επικαλύπτεται με την εξόρυξη δεδομένων, ένας τομέας που επικεντρώνεται στην ανακάλυψη και την εξεύρεση γενικών προτύπων μέσα σε σύνολα δεδομένων. Δημοφιλείς αλγόριθμοι, όπως ομαδοποίηση k-means, ανάλυση συσχέτισης, και ανάλυση παλινδρόμησης, εφαρμόζονται τόσο στην εξόρυξη δεδομένων όσο και στη μηχανική μάθηση για την ανάλυση δεδομένων. Αλλά ενώ η μηχανική μάθηση επικεντρώνεται στην κρίσιμη διαδικασία της αυτο-μάθησης και μοντελοποίησης δεδομένων για τη διαμόρφωση προβλέψεων σχετικά με το μέλλον, η εξόρυξη δεδομένων

περιορίζεται στον καθαρισμό μεγάλων συνόλων δεδομένων για να συγκεντρώσει πολύτιμες πληροφορίες από το παρελθόν (Theobald, 2017).

Διαπιστώνουμε δηλαδή ότι πράγματι η μηχανική μάθηση και η εξόρυξη δεδομένων έχουν ομοιότητες σχετικά με τις τεχνικές που χρησιμοποιούν ωστόσο υπάρχουν και διαφορές μεταξύ τους. Στη μηχανική μάθηση η κατηγοριοποίηση και η συσταδοποίηση εστιάζουν στην ακρίβεια του μοντέλου ενώ στην εξόρυξη δεδομένων οι τεχνικές αυτές εστιάζουν στην αποδοτικότητα και επεκτασιμότητα των μεθόδων ιδίως σε μεγάλα σύνολα δεδομένων καθώς και στην εύρεση τρόπων χειρισμού πολύπλοκων τύπων δεδομένων αλλά και στην εύρεση νέων μεθόδων εξόρυξης (Han, et. al. 2011).

2.2 Εξόρυξη Κειμένου

Η εξόρυξη κειμένου μπορεί να οριστεί ευρέως σαν μια διαδικασία στην οποία ένας χρήστης αλληλοεπιδρά με μια συλλογή εγγράφων χρησιμοποιώντας μια σειρά εργαλείων ανάλυσης. Με τρόπο ανάλογο με την εξόρυξη δεδομένων, η εξόρυξη γνώσης κειμένων επιδιώκει να εξαγάγει χρήσιμες πληροφορίες από πηγές δεδομένων μέσα από μια διαδικασία αναγνώρισης και εξερεύνησης σημαντικών προτύπων. Στην περίπτωση της εξόρυξης κειμένου βέβαια, οι πηγές δεδομένων είναι συλλογές εγγράφων και τα ενδιαφέροντα μοτίβα δεν βρίσκονται ανάμεσα σε τυποποιημένες εγγραφές βάσεων δεδομένων αλλά μέσα σε μη δομημένα δεδομένα κειμένου μέσα στα έγγραφα αυτών των συλλογών (Feldman & Sanger, 2007).

Με πιο απλά λόγια η εξόρυξη γνώσης από κείμενο ή αλλιώς text analytics είναι η διαδικασία εξαγωγής χρήσιμης πληροφορίας από κείμενο (Anandarajan, Hill & Nolan, 2019).

Λόγω του σημαντικού ρόλου που κατέχει η διαχείριση κειμένου φυσικής γλώσσας, είναι αξιοσημείωτο να αναφέρουμε ότι η εξόρυξη κειμένων εξελίχθηκε και σε άλλους τομείς της επιστήμης των υπολογιστών που ασχολούνται με τον χειρισμό της φυσικής γλώσσας όπως η Υπολογιστική Γλωσσολογία (Feldman & Sanger, 2007).

Η ανάλυση κειμένου έχει επηρεαστεί από πολλά πεδία αλλά έχει συμβάλει και σημαντικά σε πολλούς κλάδους. Οι σύγχρονες εφαρμογές της καλύπτουν πολλούς κλάδους και διασταυρώνει πολλούς ερευνητικούς τομείς, όπως (Anandarajan, 2019):

- Βιβλιοθηκονομία και επιστήμη της πληροφορίας
- Κοινωνικές επιστήμες
- Επιστήμη των υπολογιστών
- Βάσεις δεδομένων
- Εξόρυξη δεδομένων
- Στατιστική
- Τεχνητή νοημοσύνη
- και Υπολογιστική γλωσσολογία όπως ήδη προαναφέραμε.

Επίσης εκμεταλλεύεται τεχνικές και μεθοδολογίες από τους τομείς της ανάκτησης πληροφοριών, της εξαγωγής γνώσης και της υπολογιστικής γλωσσολογίας (Feldman & Sanger, 2007).

2.2.1 Σύγκριση Εξόρυξης Κειμένου με Εξόρυξη Δεδομένων

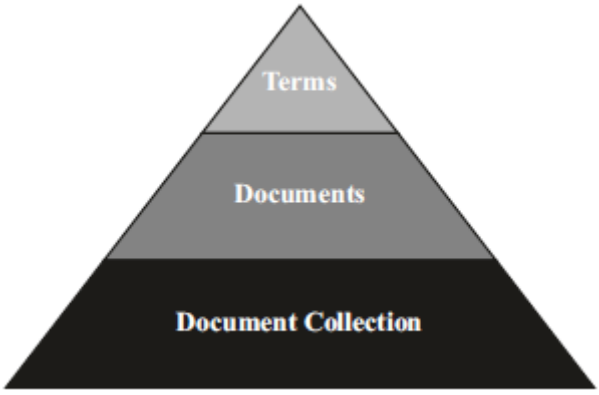
Η εξόρυξη κειμένου αντλεί μεγάλο μέρος της έμπνευσης και της κατεύθυνσής της από την εξόρυξη δεδομένων. Επομένως, δεν αποτελεί έκπληξη το γεγονός ότι τα συστήματα εξόρυξης κειμένων και εξόρυξης δεδομένων παρουσιάζουν πολλές αρχιτεκτονικές ομοιότητες υψηλού επιπέδου. Για παράδειγμα, και οι δύο βασίζονται σε τεχνικές προ-επεξεργασίας, αλγόριθμους ανακάλυψης προτύπων και σε εργαλεία οπτικοποίησης για την παρουσίαση των αποτελεσμάτων. Επιπλέον, η εξόρυξη κειμένου υιοθετεί πολλούς από τους συγκεκριμένους τύπους προτύπων στις βασικές λειτουργίες ανακάλυψης γνώσης οι οποίοι εφαρμόστηκαν αρχικά και εξετάστηκαν στην έρευνα εξόρυξης δεδομένων (Feldman & Sanger, 2007).

Εν τούτοις, υπάρχουν και σημαντικές διαφορές μεταξύ τους. Η βασική διαφορά είναι ο τύπος των δεδομένων που επεξεργάζονται προς ανάλυση. Η εξόρυξη δεδομένων χρησιμοποιεί δομημένα δεδομένα που βρίσκονται σε βάσεις δεδομένων ενώ η εξόρυξη κειμένου χρησιμοποιεί αδόμητα ή ημιδομημένα δεδομένα από μια ποικιλία πηγών όπως τα μέσα κοινωνικής δικτύωσης, τα μέσα μαζικής ενημέρωσης, ο παγκόσμιος ιστός και άλλες πηγές ηλεκτρονικών δεδομένων (Anandarajan, et. al. 2019).

Ειδικότερα, επειδή η εξόρυξη δεδομένων προϋποθέτει ότι τα δεδομένα τα προερχόμενα από μια βάση δεδομένων έχουν ήδη αποθηκευτεί σε δομημένη μορφή, μεγάλο μέρος της προ-επεξεργασίας επικεντρώνεται σε δύο κρίσιμες εργασίες: Απόξεση και ομαλοποίηση. Αντιθέτως, στην εξόρυξη κειμένων, οι λειτουργίες προ-επεξεργασίας επικεντρώνονται στον εντοπισμό και την εξαγωγή αντιπροσωπευτικών χαρακτηριστικών για έγγραφα φυσικής γλώσσας. Αυτές οι διαδικασίες προ-επεξεργασίας είναι υπεύθυνες για τη μετατροπή μη δομημένων δεδομένων που είναι αποθηκευμένα σε συλλογές εγγράφων σε μια σαφώς πιο δομημένη ενδιάμεση μορφή, οι οποίες σε καμία περίπτωση δεν σχετίζονται με τα περισσότερα συστήματα εξόρυξης δεδομένων (Feldman & Sanger, 2007).

2.3 Προ-επεξεργασία Κειμένου

Η διαδικασία της προ-επεξεργασίας κειμένων αποτελεί την πιο σημαντική διεργασία για ένα σύστημα εξόρυξης γνώσης-πληροφορίας. Πριν από την εφαρμογή τεχνικών ανάλυσης κειμένου είναι απαραίτητη η προετοιμασία και ο καθαρισμός των κειμένων της συλλογής. Σκοπός της διαδικασίας αυτής είναι η βελτίωση της αποτελεσματικότητας των τεχνικών που θα εφαρμοστούν στη συνέχεια και εν τέλει της αποδοτικότητας της ανάκτησης πληροφορίας. Το αποτέλεσμα της προ-επεξεργασίας κειμένων αφορά την εξαγωγή των χαρακτηριστικών όρων κάθε κειμένου οι οποίοι είναι κατάλληλοι για την αναπαράσταση του περιεχομένου κάθε κειμένου.



Εικόνα 2. Hierarchy of terms and documents (Anandarajan et. al., 2019)

Η προ-επεξεργασία κειμένων αποτελείται από τα παρακάτω στάδια:

1. Tokenize

2. Standardize & Cleanse
3. Stop Word Removal
4. Stem or Lemmatize

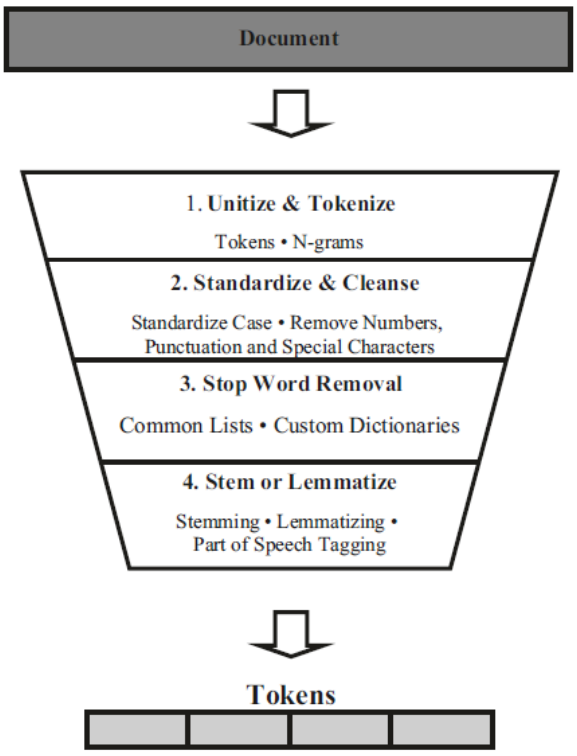
Tokenize : Σε αυτό το στάδιο το κείμενο διαχωρίζεται σε λήμματα (tokens). Τα λήμματα μπορεί να είναι λέξεις, αριθμοί, σύμβολα ή σημεία στίξης. Επειδή οι αριθμοί, τα σύμβολα και τα σημεία στίξης δεν προσφέρουν καμία πληροφορία και δεν έχουν καμία σχέση με το εννοιολογικό περιεχόμενο του κειμένου θα πρέπει να αναγνωριστούν και να αφαιρεθούν σε επόμενο στάδιο. Με αυτό τον τρόπο παραμένουν μόνο οι λέξεις του κειμένου. Τα tokens μπορεί να είναι είτε λέξεις ή φράσεις στις οποίες καθορίζουμε εμείς το μήκος των λέξεων γνωστά ως n-grams tokens μήκους 2 ή 3 λέξεων τα οποία θέλουμε να εντοπίσουμε μέσα στη συλλογή.

Standardize & Cleanse: Σε αυτό το στάδιο αναγνωρίζονται τα λήμματα που αφορούν σύμβολα, χαρακτήρες, σημεία στίξης, αριθμούς και αφαιρούνται από τη συλλογή μας. Επίσης σε αυτό το στάδιο μετατρέπονται τα κεφαλαία γράμματα σε μικρά. Έτσι μετασχηματίζουμε τα κείμενα σε ένα σύνολο λέξεων όρων, terms.

Stop Word Removal: Stop word θεωρείται ένας όρος ο οποίος έχει μεγάλη συχνότητα εμφάνισης μέσα στο κείμενο αλλά δεν σχετίζεται με το περιεχόμενο του. Για παράδειγμα ένας τέτοιος όρος μπορεί να είναι μία πρόθεση, ένα άρθρο ή ακόμη και ένας σύνδεσμος δύο προτάσεων. Αν οι όροι αυτοί δεν αφαιρεθούν λειτουργούν συνήθως ως θόρυβος με κίνδυνο τη μείωση της απόδοσης του αλγορίθμου που θα εφαρμοστεί στο επόμενο στάδιο. Με την αφαίρεση των stop words μειώνεται το μέγεθος του κειμένου μέσα τη συλλογή και παραμένουν οι πιο χρήσιμες από σημασιολογική άποψη λέξεις.

Stem or Lemmatize: Στόχος της διαδικασίας της αποκατάληξης είναι να αναγνωρισθούν οι ρίζες των λέξεων, ανεξάρτητα από τη πτώση ή το χρόνο στον οποίο βρίσκονται. Με αυτό τον τρόπο μειώνεται ακόμη περισσότερο το μέγεθος των όρων που τελικά θα χρησιμοποιηθούν για την αναπαράσταση των κειμένων. Για παράδειγμα οι λέξεις "argue", "argued", "argues", "arguing" μπορούν να αναχθούν στην κοινή ρίζα "argu". Αυτό διευκολύνει επιπλέον την απόδοση των αλγόριθμων εξόρυξης γνώσης. Ωστόσο αυτό ενδέχεται να μην έχει πάντα καλά αποτελέσματα πχ για τη λέξη train οι λέξεις trains, trained, training, trainer επιστρέφουν με το stemming την ίδια λέξη train.

Κατά τη διαδικασία της λεξικογραφικής ανάλυσης (part of speech tagging) αναγνωρίζεται τι μέρους του λόγου είναι η κάθε λέξη του κειμένου, δηλαδή ουσιαστικό, ρήμα, επίθετο κτλ. Κατά την υλοποίηση της διαδικασίας αυτής γίνεται η επιλογή των όρων. Επισημαίνεται ότι επιλέγονται κυρίως τα ουσιαστικά διότι φέρουν τη πιο σημαντική πληροφορία των κειμένων.



Εικόνα 3. Pre - processing process (Anandarajan et. al., 2019)

2.4 Αναπαράσταση κειμένου

Για να εφαρμοσθούν οι τεχνικές εξόρυξης κειμένων θα πρέπει τα κείμενα να αναπαρασταθούν σε μία μορφή που να είναι επεξεργάσιμη. Η πιο γνωστή μέθοδος αναπαράστασης κειμένων είναι με τη μορφή ενός πίνακα. Ο πίνακας αυτός αποτελείται από τόσες στήλες όσοι είναι και οι μοναδικοί όροι των κειμένων και τόσες γραμμές όσα είναι τα κείμενα της συλλογής και ονομάζεται document term matrix ή DTM. Αυτό προκύπτει από την ιδέα ότι το νόημα κάθε κειμένου μπορεί να εξαχθεί από τους όρους εκείνους που αντικατοπτρίζουν το σημασιολογικό του περιεχόμενο. Έτσι στον πίνακα DTM κάθε

κείμενο αναπαρίσταται από ένα σύνολο όρων. Για το λόγο αυτό, προκειμένου να εντοπισθούν οι μοναδικοί εκείνοι όροι που αποτυπώνουν το νόημα του κειμένου, οι οποίοι στη συνέχεια θα αποτελέσουν τις στήλες του πίνακα DTM, η προ-επεξεργασία κειμένων προηγείται της αναπαράστασης. Πολλές φορές για την αναπαράσταση κειμένων χρησιμοποιείται και ο term document matrix ή TDM ο οποίος είναι ουσιαστικά ο πίνακας DTM με εναλλαγή των γραμμών σε στήλες και των στηλών σε γραμμές .

DTM	**Term 1**	**Term 2**	**Term 3**
Document 1			
Document 2			
Document 3			

TDM	**Document 1**	**Document 2**	**Document 3**
Term 1			
Term 2			
Term 3			

Πίνακας 1. DTM και TDM Matrix

Υπάρχουν δύο βασικοί τρόποι που χρησιμοποιούνται για την αναπαράσταση κειμένων μέσα στον πίνακα.

Boolean Model: Σε ένα Boolean μοντέλο ο κάθε όρος του πίνακα μπορεί να πάρει τη τιμή 1 ή τη τιμή 0. Επομένως, η τιμή 1 σημαίνει ότι ο όρος εκείνος εμφανίζεται στο κείμενο, ενώ η τιμή 0 σημαίνει ότι ο όρος αυτός δεν υπάρχει στο κείμενο.

Term – Weight Model : Σε ένα Term – Weight μοντέλο σημαντικό ρόλο παίζει η συχνότητα εμφάνισης των όρων στα κείμενα. Αυτό σημαίνει ότι σε κάθε όρο του κειμένου (στήλη πίνακα DTM) αντιστοιχεί μία τιμή, η οποία υποδηλώνει τη συχνότητα εμφάνισης του εκάστοτε όρου στο κείμενο.

Μία μέθοδος αυτού του τύπου είναι το Term Frequency – Inverse Document Frequency Weighting (TF – IDF) το οποίο θεωρείται αρκετά αποτελεσματικό και θα το αναλύσουμε περαιτέρω.

2.4.1 Συχνότητα Όρου–Αντίστροφη Συχνότητα Κειμένου (TF – IDF)

Όπως αναφέραμε και στην προηγούμενη ενότητα, κάθε κείμενο αναπαρίσταται υπό τη μορφή πίνακα, όπου κάθε στήλη στον πίνακα DTM αποτελεί ένα μοναδικό όρο της συλλογής κειμένων. Επίσης, σε κάθε στήλη αντιστοιχεί ένας πραγματικός αριθμός ο οποίος εξαρτάται από τη συχνότητα εμφάνισης του εκάστοτε όρου στο κείμενο.

Η μέθοδος TF- IDF στοχεύει στο να σταθμίσει όλους τους όρους μιας συλλογής κειμένων. Με λίγα λόγια δηλαδή, στόχος της είναι να αποδώσει το αντίστοιχο βάρος σε κάθε όρο και κατά επέκταση στο στοιχείο του πίνακα. Αυτό συμβαίνει γιατί η απλή αρίθμηση ενός όρου σε ένα κείμενο δεν αρκεί για να μας πληροφορήσει για τη σημαντικότητα του όρου αυτού και τη βαρύτητα της πληροφορίας που περιέχει.

Η μέθοδος αυτή αποτελείται από τις ποσότητες TF και IDF. Η ποσότητα TF (συχνότητα όρου) υποδηλώνει το πόσες φορές εμφανίζεται ένας όρος σε ένα κείμενο. Από την άλλη η ποσότητα IDF υποδηλώνει το πόσο ένας όρος είναι διαδεδομένος σε ένα κείμενο αλλά και σε ολόκληρη τη συλλογή κειμένων. Επίσης, η ποσότητα IDF υπολογίζεται από το λογάριθμο του πηλίκου όλων των κειμένων προς τα κείμενα που περιέχουν τον όρο.

$$idf\ (t, D) = \log \frac{|D|}{|\{d \in D : t \in d\}|}$$

Όπου D όλα τα κείμενα και {d E D: t E D} εκείνα στα οποία εμφανίζεται ο όρος t.

Τελικά, το βάρος ενός όρου προκύπτει από τον πολλαπλασιασμό των ποσοτήτων TF και IDF όπως ακριβώς αποτυπώνεται από τον παρακάτω τύπο.

$$tf * idf\ (t, d, D) = tf\ (t, d) * idf\ (t, D)$$

Στόχος της μεθόδου αυτής μέσω του βάρους TF – IDF είναι η επιλογή εκείνων των όρων που αποτυπώνουν καλύτερα το περιεχόμενο ενός κειμένου. Για τον προσδιορισμό του βάρους ενός όρου είναι εξίσου σημαντικές και οι δύο ποσότητες TF και IDF όπως

προκύπτει από την παραπάνω εξίσωση. Αυτό επισημαίνεται διότι αν χρησιμοποιούσαμε μόνο τη συχνότητα εμφάνισης ενός όρου (TF) ως βάρος, αυτό θα είχε ως συνέπεια οι συχνότερα εμφανιζόμενοι όροι να θεωρούνται ως οι πιο σημαντικοί. Αυτή η υπόθεση θα μπορούσε να μας οδηγήσει σε λανθασμένη επιλογή όρων οι οποίοι εμφανίζονται σε πολλά κείμενα και δεν προσφέρουν κάποια ιδιαίτερη πληροφορία σε ένα κείμενο. Για παράδειγμα, η λέξη «χρέος» σε μία συλλογή κειμένων με θέμα «Διαχείριση του Τεχνικού Χρέους» θα εμφανίζεται με μεγάλη συχνότητα σε όλα τα κείμενα της συλλογής. Επομένως, ένας τέτοιος όρος παρότι θα μπορούσε από τη μία να εμφανίζεται αρκετές φορές σε ένα κείμενο, από την άλλη δεν θα μπορούσε να θεωρηθεί σημαντικός όρος γιατί δεν προσφέρει ένα ιδιαίτερο χαρακτηριστικό στο κείμενο σε σχέση με τα υπόλοιπα κείμενα της συλλογής. Εδώ λοιπόν, καταλαβαίνουμε τη σημαντικότητα της ποσότητας IDF στον υπολογισμό του βάρους ενός όρου. Σύμφωνα με τον παραπάνω τύπο υπολογισμού της ποσότητας IDF όταν ένας όρος εμφανίζεται σε πολλά κείμενα της συλλογής η τιμή της ποσότητας IDF είναι μικρή, ενώ όταν ένας όρος εμφανίζεται σε λίγα κείμενα της συλλογής η τιμή της ποσότητας IDF είναι μεγάλη. Επομένως, μεγάλο βάρος (TF*IDF) για έναν όρο προκύπτει όταν ο όρος αυτός εμφανίζεται πολλές φορές σε ένα κείμενο και λιγότερες φορές στο σύνολο των κειμένων.

2.5 Τεχνικές Ανάλυσης Κειμένου

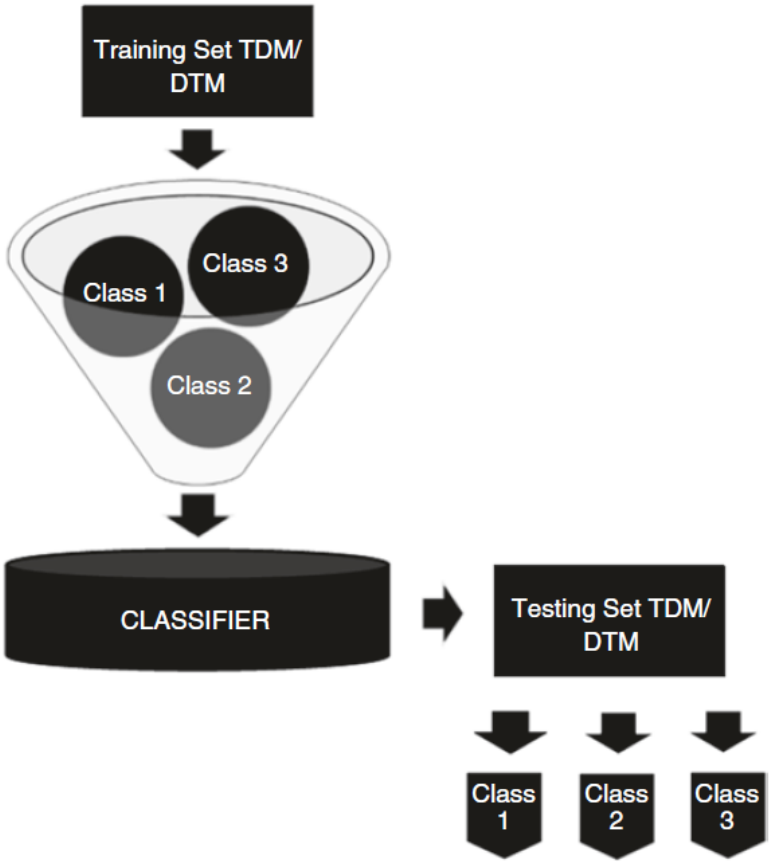
Οι τεχνικές ανάλυσης κειμένου όπως και στην εξόρυξη δεδομένων χωρίζονται σε δύο μεγάλες κατηγορίες (Anandarajan et. al., 2019):

- **Μέθοδοι Επιβλεπόμενης Ανάλυσης**: μέθοδοι οι οποίες είτε προσπαθούν να παράγουν ένα μοντέλο πρόβλεψης είτε να κατηγοριοποιήσουν τα δεδομένα πχ. Classification, Sentiment Analysis κα.
- **Μέθοδοι Μη Επιβλεπόμενης Ανάλυσης**: σε αυτές τις τεχνικές δεν έχουμε προηγούμενη γνώση των υπό ανάλυση κειμένων πχ. Topic Modelling, Latent Semantic Analysis, Clustering κα.

Στις επόμενες υπο-ενότητες που ακολουθούν θα αναφέρουμε τις πιο σημαντικές μεθόδους επιβλεπόμενης και μη επιβλεπόμενης μάθησης που χρησιμοποιούνται στην εξόρυξη κειμένου.

2.5.1 Μοντέλα Κατηγοριοποίησης – Classification Models

Τα μοντέλα κατηγοριοποίησης ανήκουν στη κατηγορία των μεθόδων επιβλεπόμενης ανάλυσης και τα χρησιμοποιούμε προκειμένου να ταξινομήσουμε τα documents. Στα μοντέλα ταξινόμησης χωρίζουμε τα δεδομένα μας σε δύο set, το training set (σύνολο εκπαίδευσης) και το testing set. Κάποιες φορές χρησιμοποιείται και το validation set. Χωρίζουμε λοιπόν τα δεδομένα μας, τα documents στη περίπτωση του text analytics, σε δύο σύνολα το training set και το testing set. Στο training set πρέπει να συμπεριλάβουμε ένα επαρκή αριθμό documents ώστε ο αλγόριθμος να είναι αποτελεσματικός και να μπορεί να προσδώσει τη σωστή ετικέτα κατηγορίας στα documents. Ένας καλός διαχωρισμός είναι 70% για το training set και 30% για το testing set (Anandarajan et. al., 2019). Μόλις χωρίσουμε τα documents τότε εφαρμόζουμε το μοντέλο πρόβλεψης, που έχουμε επιλέξει, στο document term matrix ή στο term document matrix του training set. Με αυτόν τον τρόπο εκπαιδεύουμε το μοντέλο μας ώστε να μπορεί να προβλέψει την κατηγορία του κάθε document στο testing set δηλαδή να μπορεί να προσδώσει τη σωστή ετικέτα κατηγορίας σε κάθε document του testing set. Η εικόνα 5 δείχνει τη διαδικασία του μοντέλου πρόβλεψης.



Εικόνα 4. Classification Analysis Process (Anandarajan et. al., 2019)

Μετά την εκτέλεση, για να ελέγξουμε την ποιότητα του μοντέλου χρησιμοποιούμε τη μήτρα σύγχυσης η οποία είναι ο πίνακας που περιέχει τις πραγματικές κατηγορίες και αυτές που προέβλεψε το μοντέλο. Αν σε ένα πρόβλημα ταξινόμησης οι κατηγορίες ή αλλιώς κλάσεις είναι Yes και No τότε ο πίνακας θα έχει την παρακάτω μορφή :

		Actual		
		Yes	*No*	**Total Predicted**
Predicted	*Yes*	3	6	9
	No	7	4	11
	Total Actual	10	10	20

Πίνακας 2. Μήτρα Σύγχυσης

Η μήτρα σύγχυσης χρησιμοποιείται για να αξιολογήσουμε την αποτελεσματικότητα του μοντέλου που εφαρμόζουμε κάθε φορά. Οι μετρικές που υπολογίζουμε είναι οι ακόλουθες: Accuracy, Error Rate και F-measure. Το Accuracy είναι το πηλίκο των σωστών προβλέψεων προς το σύνολο των προβλέψεων. Το Error Rate είναι το πηλίκο των λανθασμένων προβλέψεων προς το σύνολο των προβλέψεων και υπολογίζονται όπως παρακάτω:

$$\text{Accuracy} = \frac{\#\,\text{of correct predictions}}{\#\,\text{of total predictions}} * 100$$

$$\text{Error Rate} = \frac{\#\,\text{of incorrect predictions}}{\#\,\text{of total predictions}} * 100 = (100 - \text{Accuracy})$$

Η μετρική F-measure ανήκει στις μετρικές Class – Specific και μας βοηθάει να αξιολογήσουμε το μοντέλο πρόβλεψης για κάθε κλάση ξεχωριστά. Αυτό βοηθάει τον αναλυτή στην κατάλληλη επιλογή του μοντέλου με βάση το κριτήριο εντοπισμού μιας συγκεκριμένης κλάσης. Οι μετρικές Class – Specific είναι οι ακόλουθες : Precision, Recall και F-measure. Precision είναι το Accuracy της εξεταζόμενης κλάσης i δηλαδή το πηλίκο των σωστών προβλέψεων για την συγκεκριμένη κλάση i προς το σύνολο των προβλέψεων στην κλάση αυτή.

$$\text{Precision}_i = \text{Accuracy}_i = \frac{\#\,\text{of correct predicted}_i}{\text{total}\,\#\,\text{predicted}_i}$$

Το Recall μετράει πόσες προβλέψεις της εξεταζόμενης κλάσης i είναι σωστές προς το συνολικό αριθμό των documents που ανήκουν πραγματικά σε αυτή την κλάση.

$$\text{Recall}_i = \frac{\#\,\text{of correct predicted}_i}{\text{total}\,\#\,\text{of actual}_i}$$

Αφού υπολογίσουμε τις παραπάνω μετρικές για την κλάση i τότε η μετρική F για την ίδια κλάση i δίνεται από τον ακόλουθο τύπο:

$$F_i = 2 * \frac{\text{precision} * \text{recall}}{\text{precision} + \text{recall}}$$

Αν για παράδειγμα σε ένα πρόβλημα ταξινόμησης έχουμε δυο κλάσεις Yes και No τότε υπολογίζουμε την F(Yes) και την F(No) για να αξιολογήσουμε την αποτελεσματικότητα του μοντέλου να προβλέψει την κάθε κλάση ξεχωριστά. Όσο μεγαλύτερη είναι η τιμή αυτή τόσο πιο αποτελεσματικό είναι το μοντέλο και η μέγιστη δυνατή τιμή που μπορεί να έχει είναι το 1.

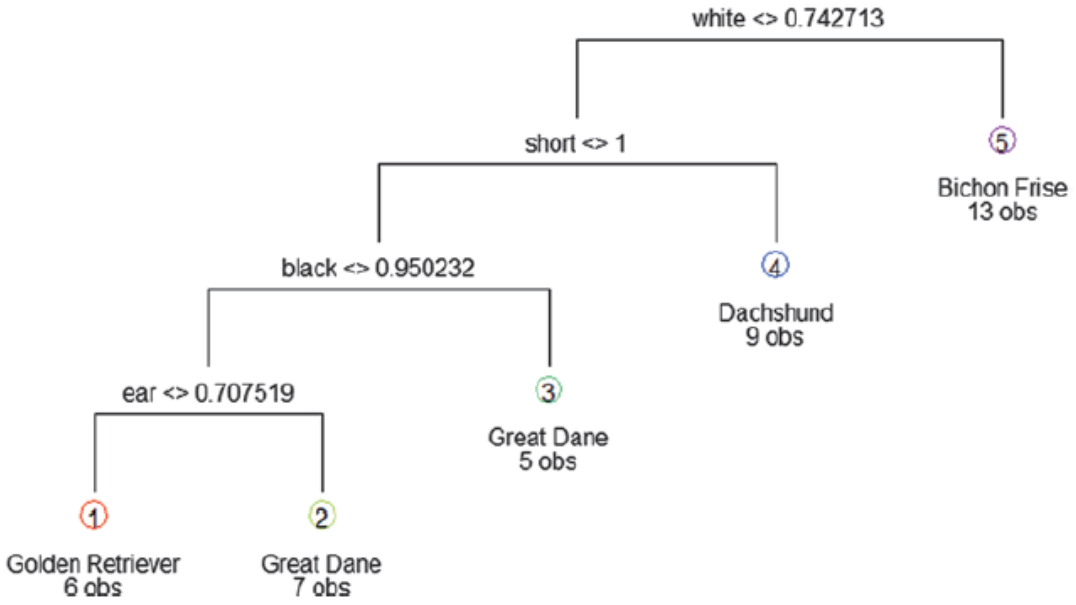
Για να ταξινομήσουμε τα documents σε ένα dataset υπάρχουν αρκετά μοντέλα πρόβλεψης. Αυτά είναι τα ακόλουθα: Decision Trees, Naïve Bayes, K – Nearest Neighbor, Support Vector Machines, Random Forest και Neural Networks. Παρακάτω αναλύουμε κάποια από αυτά.

2.5.1.1 Decision Tree

Το δέντρο απόφασης έχει τη μορφή ενός δέντρου σε ανάποδη φορά και αποτελεί την ιεραρχική δομή μιας σειράς ερωτήσεων με τις πιο πιθανές απαντήσεις. Αποτελείται από κόμβους τριών τύπων.

- Κόμβος ρίζα η οποία δεν έχει εισερχόμενες ακμές
- Εσωτερικός κόμβος ο οποίος έχει ακριβώς μια εισερχόμενη ακμή και δύο ή περισσότερες εξερχόμενες ακμές
- Φύλλο ή τερματικός κόμβος ο οποίος έχει ακριβώς μία εισερχόμενη και μια εξερχόμενη ακμή

Κάθε φύλλο σχετίζεται με μια ετικέτα κατηγορίας. Οι μη τερματικοί κόμβοι, δηλαδή η ρίζα και οι εσωτερικοί κόμβοι, περιέχουν τις συνθήκες ελέγχου των χαρακτηριστικών οι οποίες ορίζονται χρησιμοποιώντας ένα χαρακτηριστικό και κάθε πιθανό αποτέλεσμα της συνθήκης ελέγχου χαρακτηριστικού σχετίζεται με ένα παιδί ακριβώς αυτού του κόμβου.



Εικόνα 5. Δέντρο Απόφασης

Στην εικόνα 5 βλέπουμε ένα δέντρο απόφασης το οποίο κατηγοριοποιεί 4 διαφορετικά είδη σκύλων με βάση τα χαρακτηριστικά τους. Στο συγκεκριμένο παράδειγμα το training set αποτελεί το 70% του dataset. Μια σημαντική επισήμανση στο συγκεκριμένο παράδειγμα είναι ότι στις 40 εγγραφές του training set υπάρχουν 10 εγγραφές για κάθε είδος σκύλου και στο testing set 4 εγγραφές για κάθε είδος σκύλου.

Τα δέντρα απόφασης είναι απλά και εύκολα στη κατανόηση μέσω της οπτικοποίησης των αποτελεσμάτων. Επίσης δεν είναι ευαίσθητα στους outliers/noise data. Ένα μειονέκτημα είναι ότι είναι επιρρεπή στην υπερπροσαρμογή δηλαδή μπορεί να προσεγγίζουν τέλεια τα δεδομένα αλλά δεν μπορούν να γενικεύσουν. Αυτός είναι ο λόγος για τον οποίο είναι ευαίσθητα έστω και σε μικρές αλλαγές του training set (Anandarajan et. al., 2019).

2.5.1.2 Naïve Bayes

Ο κατηγοριοποιητής αυτός βασίζεται στο θεώρημα του Bayes το οποίο περιγράφεται με τον ακόλουθο τύπο:

$$P(X \mid Y) = \frac{P(Y \mid X)\, P(X)}{P(Y)}$$

όπου το P(X|Y) εκφράζει την πιθανότητα να συμβεί το X δεδομένου του συμβάντος Y,

P(X) είναι η πιθανότητα του συμβάντος X και P(Y) είναι η πιθανότητα του Y

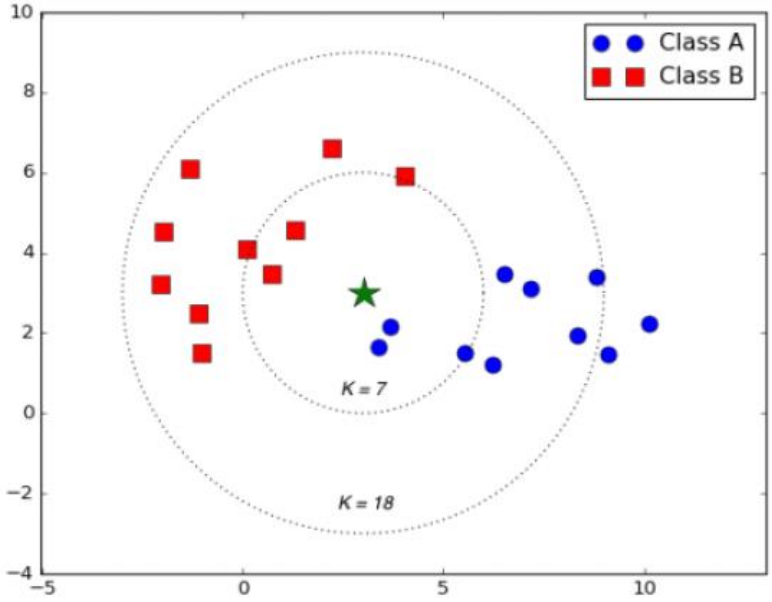
Ο Naive Bayes θεωρεί ότι οι όροι (terms) που μας δίνουν την κλάση είναι μεταξύ τους ανεξάρτητοι. Στη περίπτωση του text mining το θεώρημα του NB μας λέει ότι η πιο πιθανή ταξινόμηση ενός document D προκύπτει όπως παρακάτω:

$$\hat{C} = \arg\max P(D \mid C) * P(C)$$

Όπου P(D|C) είναι η εξαρτώμενη πιθανότητα του document D δεδομένης της κλάσης C και P(C) είναι η πιθανότητα της κλάσης. Όπως γνωρίζουμε τα documents αποτελούνται από terms (όρους), μπορούμε λοιπόν να προσδιορίσουμε την πιθανότητα ενός document δεδομένης της κλάσης πολλαπλασιάζοντας τις πιθανότητες κάθε όρου (term) δεδομένου ότι ανήκει σε αυτή τη κλάση.

2.5.1.3 K - Nearest Neighbors

Στη κατηγοριοποίηση πλησιέστερων γειτόνων χρησιμοποιείται ως κριτήριο ταξινόμησης η ευκλείδεια απόσταση. Βασίζεται στη παραδοχή ότι τα documents σε ένα group των k πλησιέστερων γειτόνων ανήκουν στην ίδια κλάση. Έτσι η μέθοδος υπολογίζει για κάθε document του test set την ευκλείδεια απόσταση του από τους κ πλησιέστερους γείτονες και το εντάσσει σε αυτήν στην οποία είναι πιο κοντά.

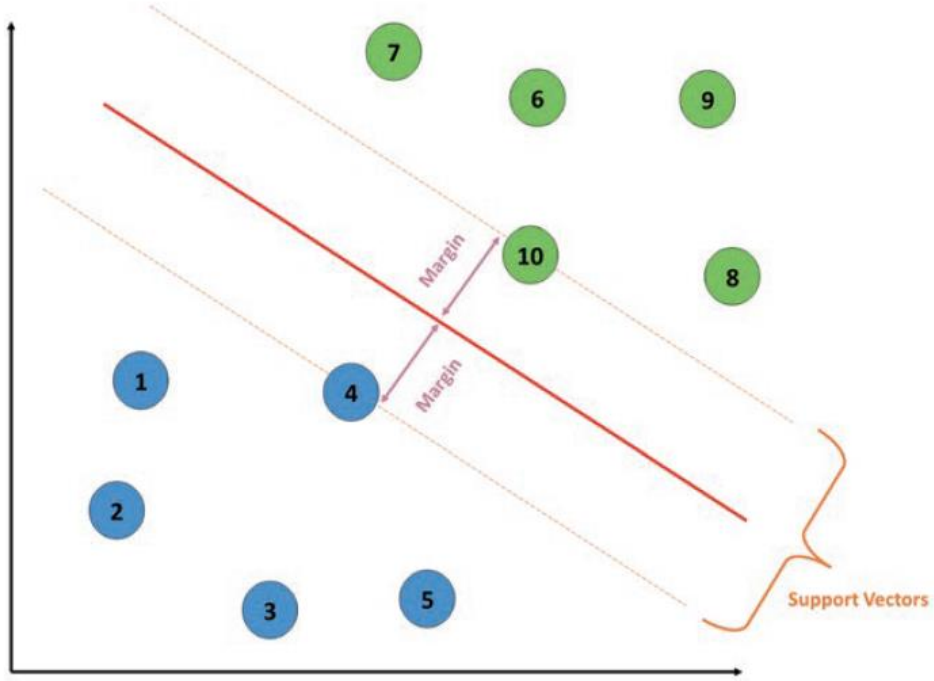


Εικόνα 6. K-Nearest Neighbors

Για παράδειγμα το πράσινο σημείο προκειμένου να ταξινομηθεί σε μία κλάση θα πρέπει να υπολογιστεί η απόσταση του από τις υπάρχουσες κλάσεις μπλε και κόκκινη. Για κ=7 το πράσινο σημείο θα ενταχθεί στην μπλε κλάση διότι βρίσκεται πιο κοντά σε αυτήν. Η ιδιαιτερότητα της μεθόδου είναι ότι πρέπει να ορίσουμε την τιμή του K και το αποτέλεσμα εξαρτάται από την επιλογή της τιμής αυτής. Η πιο δόκιμη τιμή για να ξεκινήσουμε τον αλγόριθμο, αφού αφορά επιβλεπόμενη μάθηση, είναι το πλήθος των κλάσεων που υπάρχουν στο training set. Εν τούτοις, η μέθοδος δεν θεωρείται αποτελεσματική για μεγάλα σύνολα δεδομένων (Anandarajan et. al., 2019).

2.5.1.4 Support Vector Machine

Ο ταξινομητής SVM αναζητά το ιδανικό υπερ-επίπεδο το οποίο χωρίζει τις κλάσεις μεταξύ τους στο μέγιστο δυνατό περιθώριο (margin). Αν υποθέσουμε ότι θέλουμε να ταξινομήσουμε 10 έγγραφα όπως φαίνεται στην επόμενη εικόνα: τα documents 1-5 ανήκουν στη μπλε κλάση και τα 6-10 στην πράσινη κλάση. Το ιδανικό υπερ-επίπεδο είναι η κόκκινη γραμμή.



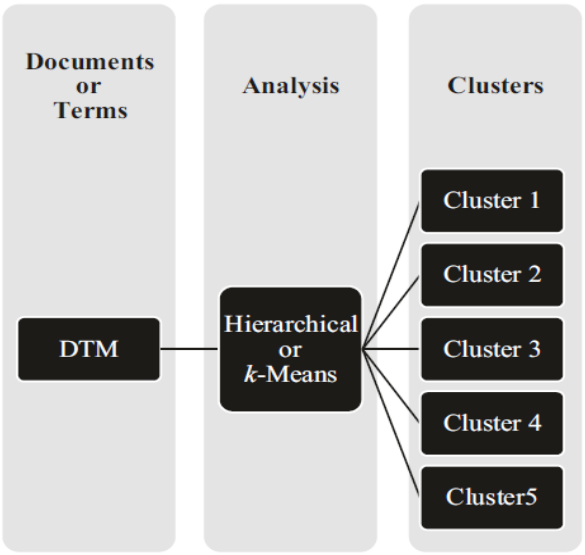
Εικόνα 7. Support Vector Machine

Ο SVM θεωρείται αρκετά αποτελεσματικός για την ταξινόμηση κειμένων διότι ανταποκρίνεται ικανοποιητικά στους πίνακες document term matrix (DTM) μεγάλων διαστάσεων και μεγάλης σποραδικότητας (sparsity). Επίσης υπερτερεί άλλων μεθόδων όπως του Naïve Bayes ή του Decision Tree αλλά μπορεί ενώ έχει ικανοποιητικό precision να έχει χαμηλό recall (Anandarajan et. al., 2019).

2.6 Συσταδοποίηση – Clustering

Στο πρόβλημα της συσταδοποίησης μας δίνεται ένα σύνολο δεδομένων, χωρίς τις αντίστοιχες κλάσεις ή ετικέτες και χρειαζόμαστε κάποιον αλγόριθμο, ο οποίος θα τα ομαδοποιήσει σε συστάδες. Οι συστάδες που δημιουργούνται θέλουμε να διαχωρίζουν ορθά τα δεδομένα. Αυτό πρακτικά σημαίνει ότι μια συστάδα θέλουμε να αποτελείται από δεδομένα π.χ. έγγραφα όπου κάθε έγγραφο είναι πιο κοντά σε κάθε άλλο έγγραφο της ίδιας συστάδας απ’ ότι σε κάποιο άλλο διαφορετικής συστάδας. Η συσταδοποίηση λοιπόν είναι μια μέθοδος μη επιβλεπόμενης μάθησης, που σημαίνει ότι π.χ. στη περίπτωση μιας συλλογής εγγράφων, δεν γνωρίζουμε την κατηγορία του εγγράφου μέσα στη συλλογή. Οι μέθοδοι συσταδοποίησης εφαρμόζουν αλγόριθμους που βασίζονται στο μέτρο απόστασης

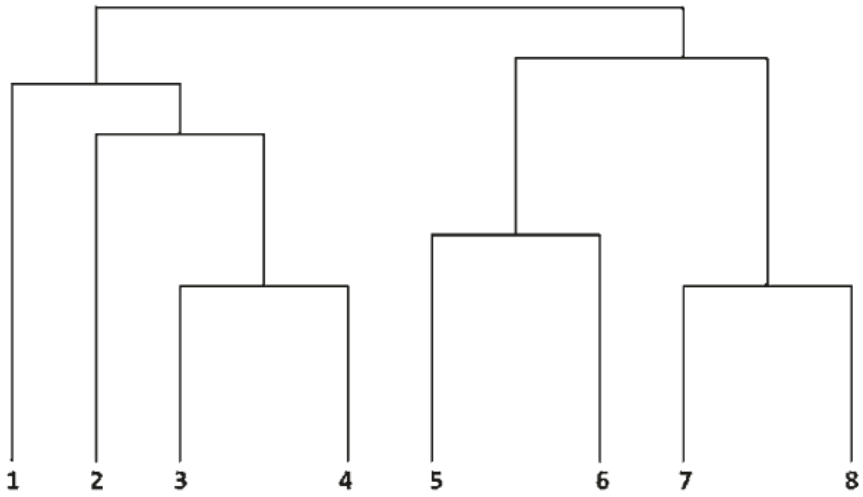
ή ομοιότητας προκειμένου να ομαδοποιηθούν τα δεδομένα ή τα έγγραφα μέσα στις συστάδες. Για την εφαρμογή του μοντέλου χρησιμοποιείται είτε ο term – document matrix (TDM) ή ο document – term matrix (DTM). Στην ανάλυση κειμένου η συσταδοποίηση μπορεί να χρησιμοποιηθεί για να βρούμε ομοιότητες είτε στις σχέσεις document-document ή term-term στον πίνακα term – document matrix ή στον πίνακα document – term matrix. Υπάρχουν δύο μέθοδοι συσταδοποίησης η ιεραρχική και η k-means.



Εικόνα 8. Cluster Analysis (Anandarajan et. al., 2019)

2.6.1 Ιεραρχική Συσταδοποίηση – Hierarchical Clustering

Η ιεραρχική συσταδοποίηση ανήκει στην κατηγορία της συσσωρετικής (agglomerative) συσταδοποίησης. Το αποτέλεσμα της μεθόδου καταλήγει σε ένα δενδρόγραμμα το οποίο μοιάζει με ένα δέντρο σε ανάποδη φορά. Αν κόψουμε το δενδρόγραμμα σε διαφορετικά επίπεδα ή ύψη τότε λαμβάνουμε και διαφορετικό αποτέλεσμα συσταδοποίησης. Στην παρακάτω εικόνα φαίνεται ένα τέτοιο δενδρόγραμμα. Οι οριζόντιες γραμμές δείχνουν πως χωρίζονται μεταξύ τους οι συστάδες και οι κάθετες γραμμές αναπαριστούν τα terms ή τα documents που βρίσκονται στην ίδια συστάδα.



Εικόνα 9. Ιεραρχική Συσταδοποίηση - Δενδρόγραμμα

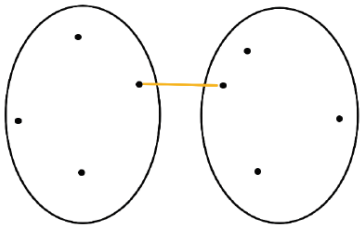
Ο αλγόριθμος ξεκινάει από το κάτω τμήμα του δενδρογράμματος. Έστω ότι στο ανωτέρω δενδρόγραμμα τα 1 – 8 είναι έγγραφα. Αρχικά κάθε document είναι και μια συστάδα και όσο ανεβαίνουμε προς τα πάνω τα έγγραφα ομαδοποιούνται σε συστάδες ώσπου να φτάσουμε στην κορυφή όπου και τα 8 documents ανήκουν σε μια μεγάλη συστάδα.

Για τη δημιουργία των συστάδων υπολογίζεται το μέτρο της απόστασης μεταξύ τους ώστε τα δεδομένα να ομαδοποιηθούν και να σχηματίσουν τις συστάδες. Το κριτήριο για το μέτρο της απόστασης μπορεί να είναι :

- Ελάχιστης απόστασης ή απλού συνδέσμου (single linkage).
- Μέγιστης απόστασης ή πλήρους συνδέσμου (complete linkage).
- Μέσου όρου της συστάδας (group average).
- Απόσταση κεντρικών σημείων (centroid distance)
- Μέθοδος του Ward.

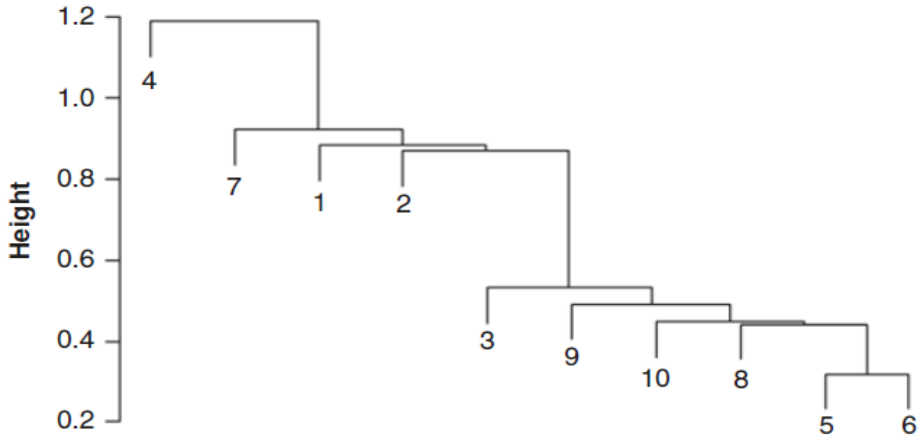
Απλού Συνδέσμου : Με βάση το κριτήριο του απλού συνδέσμου, η ομοιότητα μεταξύ δύο συστάδων βασίζεται στα δύο πιο όμοια (πιο γειτονικά) σημεία στις διαφορετικές συστάδες, (Εικόνα 10) δηλαδή στα σημεία με την ελάχιστη απόσταση μεταξύ τους. Είναι γνωστή και ως μέθοδος συσταδοποίησης πλησιέστερου γείτονα. Τα προτερήματα αυτής της μεθόδου είναι ότι δημιουργούνται συνεχόμενες συστάδες, ενώ μπορεί να χειριστεί μη ελλειπτικά

σχήματα. Το βασικό μειονέκτημα είναι η ευαισθησία στον θόρυβο και στις ακραίες τιμές (outliers) (Βερύκιος κα., 2015).



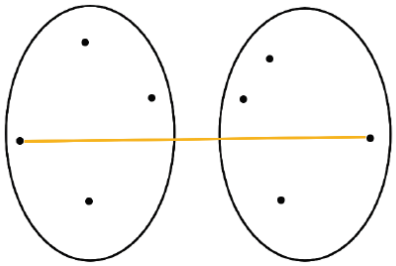
Εικόνα 10. Ομοιότητα συστάδων Απλού Συνδέσμου (Single Linkage)

Η πρώτη συστάδα που σχηματίζεται είναι η {5,6} διότι αυτά τα documents έχουν την κοντινότερη απόσταση στο document term matrix. Στη συνέχεια στην υπάρχουσα συστάδα μπαίνει και το document 8 και έτσι σχηματίζονται και οι επόμενες συστάδες με βάση το κριτήριο της μικρότερης απόστασης.



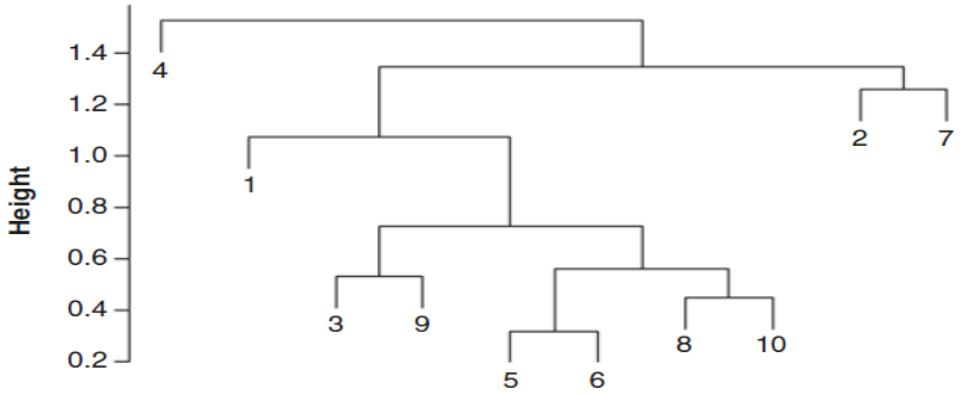
Εικόνα 11. Δενδρόγραμμα Απλού Συνδέσμου

<u>Πλήρους Συνδέσμου</u> : Με βάση το κριτήριο πλήρους συνδέσμου, η ομοιότητα μεταξύ δύο συστάδων βασίζεται στα δύο πιο ανόμοια (πιο μακρινά μεταξύ τους) σημεία στις διαφορετικές συστάδες (Εικόνα 12), δηλαδή στα σημεία με τη μέγιστη απόσταση μεταξύ τους. Το βασικό πλεονέκτημα αυτού του τρόπου σύνδεσης είναι η μικρή ευαισθησία στον θόρυβο και στις ακραίες τιμές (outliers). Τα μειονεκτήματα που έχει είναι ότι τείνει να διασπά μεγάλες συστάδες. Γενικά όμως δημιουργεί πιο συμπαγείς συστάδες σε σχέση με τις συστάδες του απλού συνδέσμου (Βερύκιος κα., 2015).



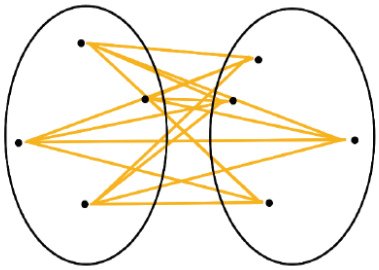
Εικόνα 12. Ομοιότητα βάσει Πλήρους Συνδέσμου (Complete Linkage)

Σε αυτή την περίπτωση η πρώτη συστάδα που σχηματίζεται είναι η {5,6} και η επόμενη είναι η {8,10} κοκ.



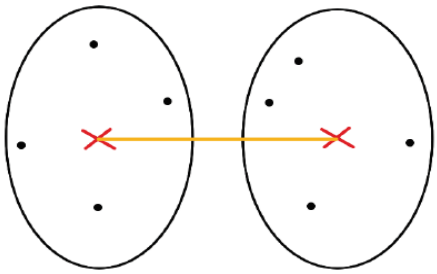
Εικόνα 13. Δενδρόγραμμα Πλήρους Συνδέσμου

<u>Μέσου όρου της συστάδας</u> : Ο μέσος όρος συστάδων είναι ουσιαστικά η μέση τιμή των αποστάσεων μεταξύ κάθε πιθανού ζεύγους μεταξύ των σημείων των δύο συστάδων (Εικόνα 14). Βρίσκεται κάπου ανάμεσα στην ελάχιστη και τη μέγιστη απόσταση. Έχει μικρότερη ευαισθησία σε θόρυβο και σε ακραίες τιμές (outliers), αλλά ευνοεί τις συστάδες με κυκλικό σχήμα (Βερύκιος κα., 2015)

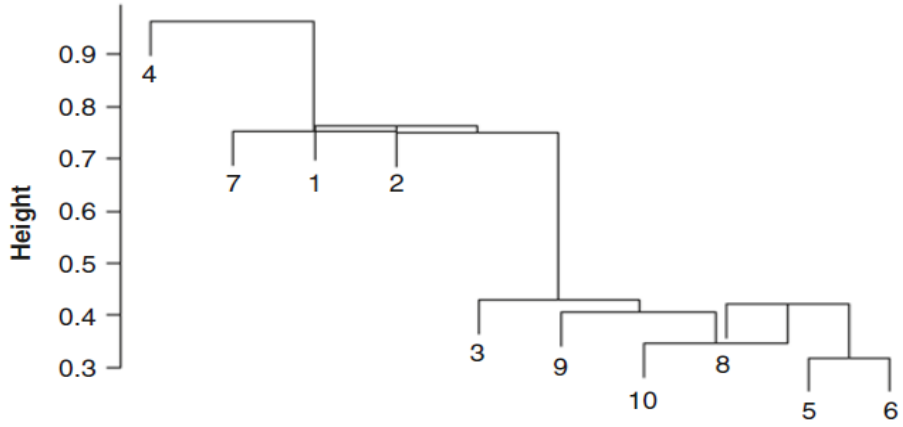


Εικόνα 14. Ομοιότητα συστάδων βάσει μέσου όρου (group average)

Απόστασης Κεντρικών Σημείων : Η απόσταση κεντρικών σημείων είναι η απόσταση μεταξύ των κέντρων των συστάδων. Το πρόβλημα με αυτή την απόσταση είναι ότι δεν έχει μονότονη αύξηση. Έτσι, δύο συστάδες που συγχωνεύονται μπορεί να έχουν μικρότερη απόσταση από συστάδες, οι οποίες έχουν συγχωνευτεί σε προηγούμενα βήματα (Βερύκιος κα., 2015)



Εικόνα 15. Ομοιότητα συστάδας βάσει Απόστασης Κεντρικών Σημείων

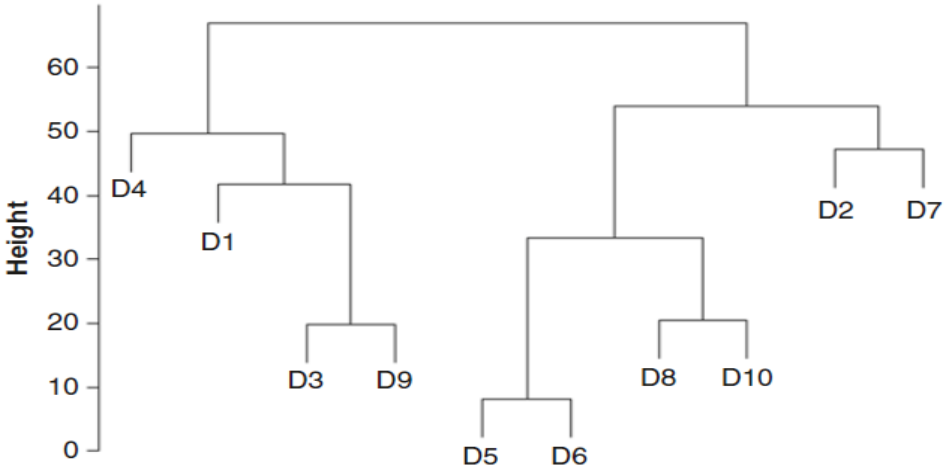


Εικόνα 16. Δενδρόγραμμα Κεντρικών Σημείων

Μέθοδος του Ward : Η μέθοδος αυτή υπολογίζει το Squared Error (SSE) και επιλέγει τη συγχώνευση των συστάδων με κριτήριο τη μικρότερη τιμή του SSE. Αν έχουμε n τελικές συστάδες και m documents τότε το Sum Squared Error (SSE) της συστάδας i υπολογίζεται όπως παρακάτω:

$$SSE_i = \sum_{k=1}^{n} \sum_{i=1}^{m} \left[x_k - \mu_i \right]^2 .$$

Τα μέσα των συστάδων προκύπτουν βάσει του μέσου όρου της συστάδας.

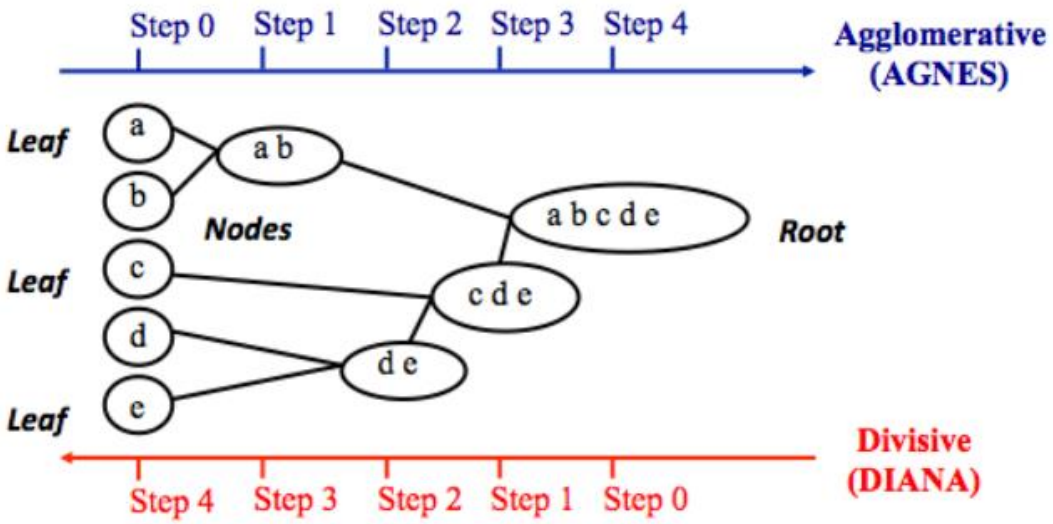


Εικόνα 17. Δενδρόγραμμα Ward

Το ισχυρό πλεονέκτημα της ιεραρχικής συσταδοποίησης είναι η οπτική αναπαράσταση του αποτελέσματος με τη μορφή δενδρογράμματος το οποίο εξυπηρετεί στην παρουσίαση και ερμηνεία των αποτελεσμάτων της ανάλυσης. Βέβαια όταν η συλλογή είναι πολύ μεγάλη τότε αυτό μετατρέπεται σε μειονέκτημα καθώς είναι δύσκολη η ανάγνωση του δενδρογράμματος. Ένα ακόμα μειονέκτημα της ιεραρχικής συσταδοποίησης είναι η ευαισθησία στις ακραίες τιμές καθώς η παρουσία τους τους επηρεάζει το τελικό αποτέλεσμα.

2.6.2 Διαιρετική Συσταδοποίηση – Divisive Clustering

Οι διαιρετικοί αλγόριθμοι ξεκινάνε με όλα τα δεδομένα να ανήκουν σε μια ενιαία συστάδα. Σε κάθε βήμα, μια ομάδα διασπάται σε δύο. Αυτό γίνεται επαναληπτικά, μέχρι να καταλήξουμε σε n ομάδες. Η πολυπλοκότητα των διαιρετικών αλγορίθμων είναι μεγαλύτερη από αυτή των συσσωρευτικών διότι η διάσπαση μιας ομάδας σε δύο μπορεί να γίνει κατά $2^{n-1}-1$ τρόπους. Η επιλογή της βέλτιστης διάσπασης πρακτικά είναι αδύνατη ακόμα και για μικρό n. Στην πράξη η διάσπαση γίνεται, αλλά όχι κατά τον βέλτιστο τρόπο. Η όλη διαδικασία του αλγορίθμου μπορεί να αναπαρασταθεί, όπως και στους συσσωρευτικούς, με δενδρόγραμμα (Βερύκιος κα., 2015).



Εικόνα 18. AGNES - DIANA

Στην εικόνα 18 φαίνεται η διαφορά των δύο κατηγοριών συσταδοποίησης. Σε αυτό το σημείο να προσθέσουμε ότι οι συσσωρευτικοί αλγόριθμοι θεωρούνται καταλληλότεροι για τη δημιουργία μικρών συστάδων ενώ οι διαιρετικοί είναι καλύτεροι για τη δημιουργία μεγάλων συστάδων.

2.6.3 K-means Clustering

Ο Αλγόριθμος k–means αποτελεί έναν από αυτής βασικότερους αλγόριθμους συσταδοποίησης, ο οποίος ανήκει παράλληλα στην κατηγορία αυτής διαχωριστικής συσταδοποίησης. Βασικό σημείο διάκρισης των διαφόρων αλγορίθμων της κατηγορίας αυτής είναι ο τρόπος αναπαράστασης της κάθε συστάδας. Κάθε ομάδα αναπαρίσταται σαν ένα επίπεδο του m – διάστατου χώρου, όπου στην πιο απλή περίπτωση, το επίπεδο αυτό αντιστοιχεί σε ένα και μόνο σημείο του χώρου που αντιπροσωπεύει το κέντρο της συστάδας. Οι αλγόριθμοι της κατηγορίας αυτής στηρίζονται σε ένα επαναληπτικό σχήμα, το οποίο ξεκινά από έναν αρχικό διαμερισμό του χώρου. Συνήθως ο διαχωρισμός αυτός γίνεται με τυχαίο τρόπο. Μετέπειτα, σε κάθε βήμα αρχικά, κάθε σημείο των δεδομένων εισόδου τοποθετείται σε μία συστάδα και στη συνέχεια ανανεώνεται το επίπεδο της κάθε συστάδας με βάση τα στοιχεία που έχουν τοποθετηθεί σε αυτή. Θα πρέπει όμως να επισημανθεί, ότι βασικό μειονέκτημα των αλγορίθμων της κατηγορίας αυτής είναι ότι τα

αποτελέσματά τους εξαρτώνται σε μεγάλο βαθμό από τον τρόπο που αρχικοποιούνται οι συστάδες.

Ο αλγόριθμος k-means ξεκινά διαχωρίζοντας το σύνολο των δεδομένων σε k συστάδες (clusters), όπου το k καθορίζεται από το χρήστη. Ο αλγόριθμος λοιπόν ξεκινάει με k τυχαία σημεία, τα οποία ονομάζονται κεντροειδή της συστάδας και δηλώνουν το κέντρο βάρους της συστάδας. Το k υποδηλώνει πόσες συστάδες θέλουμε ο αλγόριθμος να δημιουργήσει. Ο αλγόριθμος εκτελεί επαναληπτικά δύο βήματα. Το πρώτο βήμα αφορά την ανάθεση σε κάποια συστάδα, ενώ το δεύτερο βήμα αφορά τον επαναπροσδιορισμό και τη μετατόπιση του κεντροειδούς κάθε συστάδας.

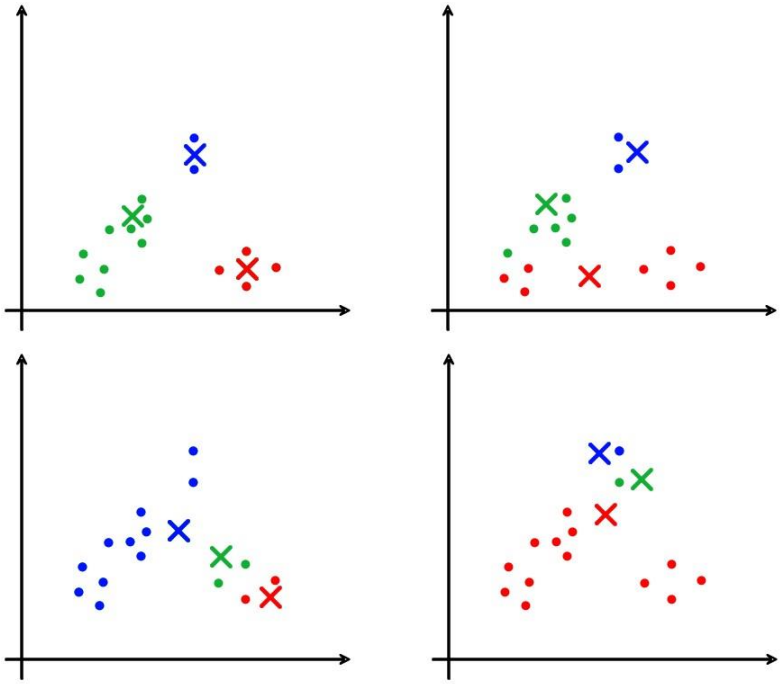
Πιο αναλυτικά, όσον αφορά στο πρώτο βήμα, δηλαδή την ανάθεση σε κάποια συστάδα, ο αλγόριθμος εξετάζει κάθε δείγμα σε σχέση με τα κεντροειδή των συστάδων. Με χρήση κάποιου μέτρου απόστασης, αναθέτει το εξεταζόμενο δείγμα στη συστάδα, της οποίας το κεντροειδές είναι το πλησιέστερο ως προς το συγκεκριμένο δείγμα. Στο δεύτερο βήμα, παίρνοντας τον μέσο όρο των δειγμάτων κάθε συστάδας, επανυπολογίζονται τα κεντροειδή της κάθε συστάδας, ώστε το κεντροειδές να είναι πιο αντιπροσωπευτικό στην πρόσφατα διαμορφωμένη συστάδα.

Εν συνεχεία, μέσω διαδοχικών επαναλήψεων κατατάσσει τα δεδομένα σε κάποια συστάδα με βάση την απόσταση τους από το κεντροειδές αυτής της συστάδας. Η απόσταση του δείγματος από το κεντροειδές της συστάδας υπολογίζεται με την συνάρτηση Sum Squared Error:

$$E = \sum_{i=1}^{k} \sum_{\mathbf{x} \in S_i} \|\mathbf{x} - \boldsymbol{\mu}_i\|^2$$

Όπου μi είναι το κέντρο βάρους της ομάδας Si και $\|x - \mu i\|^2$ είναι το τετράγωνο της Ευκλείδειας απόστασης μεταξύ του δεδομένου x και του κέντρου βάρους μi της εκάστοτε συστάδας.

Τα παραπάνω βήματα επαναλαμβάνονται μέχρι να διαπιστωθεί ότι η σύσταση των συστάδων δεν έχει αλλάξει σημαντικά από την προηγούμενη επανάληψη.



Εικόνα 19. Τυχαία αρχικοποίηση κεντροειδών του k-means (Βερύκιος κα., 2015)

Στην εικόνα 19 βλέπουμε τα διαφορετικά αποτελέσματα του αλγορίθμου σε σχέση με την αρχικοποίηση των κεντροειδών. Η πάνω αριστερά συσταδοποίηση είναι η πιο επιτυχής, η πάνω δεξιά λιγότερο επιτυχής ενώ οι δύο κάτω κρίνονται ανεπιτυχείς καθώς μια συστάδα εκ των τριών συγκεντρώνει σχεδόν όλα τα δείγματα..

Χαρακτηριστικά της k – means συσταδοποίησης:

- Εμφανίζει ικανοποιητική απόδοση όταν εφαρμόζεται σε μεγάλα σύνολα δεδομένων. Συνήθως, η πολυπλοκότητα που παρουσιάζει εξαρτάται από τον αριθμό των δεδομένων που θέλουμε να ομαδοποιήσουμε και από τον αριθμό των επαναλήψεων που θα πραγματοποιηθούν.
- Βασικό χαρακτηριστικό του αλγορίθμου είναι ο προσδιορισμός των αρχικών κέντρων των συστάδων. Τα αρχικά κέντρα των συστάδων διαδραματίζουν σημαντικό ρόλο καθώς επηρεάζουν τη σύγκλιση του αλγορίθμου σε τοπικό ή σε ολικό ελάχιστο.
- Υπάρχει περίπτωση ο αλγόριθμος να εντοπίσει ένα τοπικό ελάχιστο που να αντιστοιχεί σε ένα υποσύνολο δεδομένων και να σταματήσει τις επαναλήψεις. Αυτό συμβαίνει διότι δεν είναι σε θέση να ξεχωρίζει την εύρεση του ολικού ή του τοπικού

ελαχίστου. Έτσι υπάρχει η πιθανότητα ο αλγόριθμος να σταματήσει χωρίς να έχει καταλήξει στη βέλτιστη ελαχιστοποίηση της συνάρτησης.

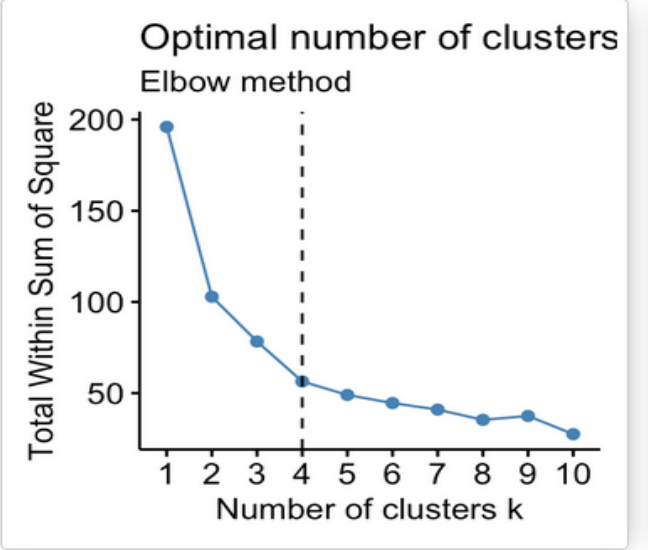
- Αξιοσημείωτο είναι επίσης το γεγονός ότι μία από τις κύριες παραμέτρους επιτυχίας του αλγορίθμου k–means είναι ο προσδιορισμός του αριθμού των συστάδων. Συχνά, για το καθορισμό του βέλτιστου αριθμού των συστάδων κρίνεται απαραίτητη η εκτέλεση του αλγορίθμου για διαφορετικό κάθε φορά αριθμό συστάδων. Μετά από κάθε επανάληψη αναλύονται τα αποτελέσματα της διαδικασίας προκειμένου να εντοπισθεί ο καλύτερος διαχωρισμός. Εκτός από αυτόν τον εμπειρικό τρόπο υπάρχουν τρεις γραφικές μέθοδοι για την επιλογή του αριθμού των συστάδων που αναλύονται παρακάτω.

- Ένας δείκτης της ποιότητας του διαχωρισμού, που εκτελεί ο αλγόριθμος της k–means συσταδοποίησης, είναι το πηλίκο του Between Clusters Sum of Squares / Total Within Cluster Sum of Squares το οποίο όσο υψηλότερο είναι τόσο καλύτερη θεωρείται η συσταδοποίηση.

2.6.4 Επιλογή αριθμού συστάδων

Για να επιλέξουμε τον αριθμό συστάδων υπάρχουν τρεις γραφικές μέθοδοι: Elbow, Silhouette και Gap Statistic.

Elbow Method : Με βάση την πρώτη μέθοδο, η οποία βασίζεται στον υπολογισμό του Sum of Square Error ανάμεσα στους clusters (within clusters) WSS, δημιουργείται το γράφημα 1 και η επιλογή του αριθμού των συστάδων προκύπτει με βάση το γράφημα. Στο παράδειγμά μας επιλέγουμε 4 συστάδες. Ωστόσο σε αρκετές περιπτώσεις η καμπύλη είναι πιο ομαλή, δηλαδή δεν έχει το σχήμα του «αγκώνα», οπότε η επιλογή μας δεν είναι ξεκάθαρη.



Γράφημα 1. Elbow method

Silhoutte Method: Στη μέθοδο αυτή υπολογίζεται ο συντελεστής Si με βάση την παρακάτω σχέση

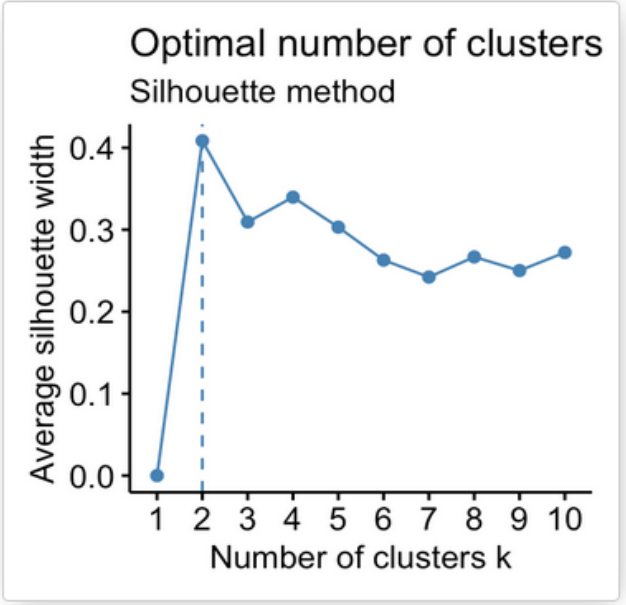
$$s_i = \frac{b_i - a_i}{\max(a_i, b_i)}$$

όπου

ai : είναι η μέση απόσταση του i data point από τα υπόλοιπα data points μέσα στο ίδιο cluster

bi : υπολογίζει την μέση απόσταση του i data point από τα υπόλοιπα data points των άλλων cluster και λαμβάνει την ελάχιστη τιμή από τις παραπάνω τιμές, δηλαδή τη μέση τιμή από το κοντινότερο cluster

Στη συνέχεια αφού υπολογιστεί ο συντελεστής Si για κάθε συστάδα υπολογίζεται η μέση τιμή του για διαφορετικά πλήθη συστάδων και έτσι προκύπτει η τιμή του Average Silhouette για διαφορετικό αριθμό συστάδων. Επιλέγουμε τις συστάδες εκεί που μεγιστοποιείται ο συντελεστής. Στο γράφημα 2 επιλέγουμε 2 cluster.



Γράφημα 2. Average Silhouete Method

Gap Statistics : Η μέθοδος αυτή χρησιμοποιεί την uniform κατανομή της στατιστικής και υπολογίζεται με βάση τον παρακάτω τύπο :

$$Gap_n(k) = E_n^*\{\log(W_k')\} - \log(W_k)$$

E^*_n : είναι η μέση τιμή των n σημείων δεδομένων (data points) της uniform κατανομής

Xij : ένα σύνολο των data points

X'ij : ένα σύνολο data points που ακολουθούν την uniform κατανομή στο διάστημα [minXij, maxXij]

K είναι το πλήθος των cluster

nr είναι το πλήθος των datapoints που ανήκουν στη συστάδα r

Ορίζεται επίσης ως d'ii η ευκλείδια απόσταση μεταξύ των data points

$$d_{ii'} = \sum_j (x_{ij} - x'_{ij})^2$$

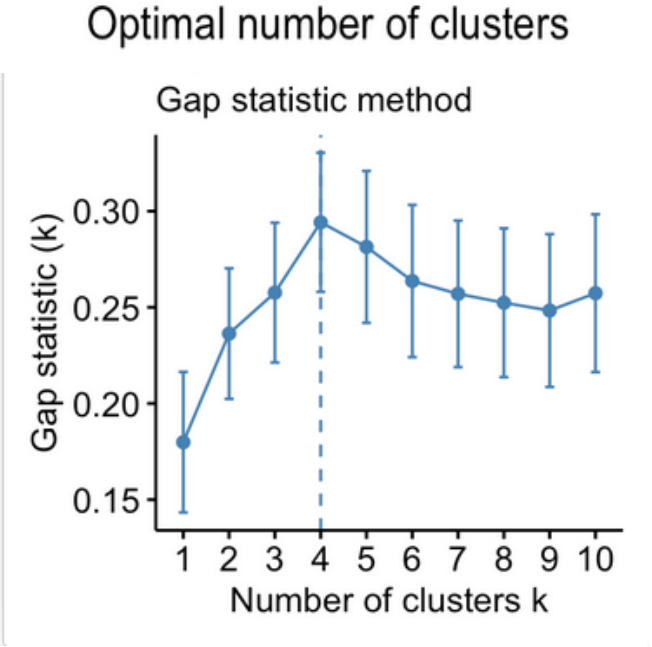
Ορίζεται ως Dr το άθροισμα των αποστάσεων εντός της συστάδας r (intra distance)

$$D_r = \sum_{i,i' \in C_r} d_{ii'}$$

και Wk είναι το άθροισμα των intra distance για όλες τις συστάδες και υπολογίζεται όπως παρακάτω

$$W_k = \sum_{r=1}^{k} \frac{1}{2n_r} D_r$$

Με βάση αυτόν τον αλγόριθμο υπολογίζεται το gap statistic και επιλέγουμε το K εκεί που μεγιστοποιείται το Gap.



Γράφημα 3. Gap Statistic Method

2.6.5 Αξιολόγηση μοντέλου συσταδοποίησης

Για να αξιολογήσουμε την αποτελεσματικότητα ενός μοντέλου που εφαρμόσαμε υπάρχουν τρεις τύποι εγκυρότητας: εσωτερική, εξωτερική και σχετική. Η εσωτερική βασίζεται στις μετρήσεις που γίνονται στα δεδομένα και στο μοντέλο που εφαρμόστηκε. Η εξωτερική προέρχεται από εξωτερική πληροφορία όπως η γνώμη κάποιου ειδικού ή οι ετικέτες κατηγορίας. Η σχετική συγκρίνει διαφορετικές λύσεις αλλάζοντας τις τιμές κάποιων

παραμέτρων. Για παράδειγμα ο συντελεστής Shilouette είναι ένα μέτρο σχετικής εγκυρότητας διότι αλλάζοντας τον αριθμό των συστάδων επιλέγουμε την παράμετρο.

Ένα μέτρο εσωτερικής εγκυρότητας που χρησιμοποιούμε προκειμένου να αξιολογήσουμε την αποτελεσματικότητα του clustering είναι ο δείκτης dunn ο οποίος υπολογίζεται με βάση τον ακόλουθο τύπο:

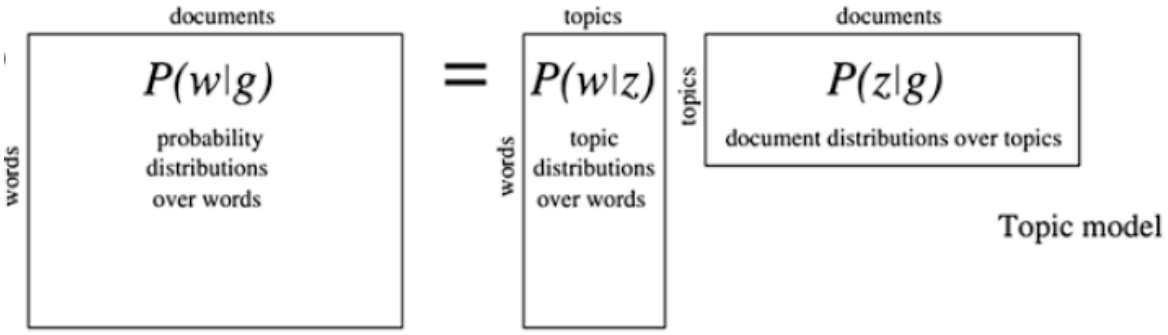
$$\text{Dunn Index} = \frac{\min\{d_{\text{inter}}\}}{\max\{d_{\text{intra}}\}}$$

όπου d_{inter} είναι η ελάχιστη απόσταση ανάμεσα σε δύο σημεία διαφορετικών συστάδων

και d_{intra} είναι η μέγιστη απόσταση ανάμεσα σε δύο σημεία της ίδιας συστάδας. Όσο υψηλότερος ο δείκτης τόσο καλύτερη είναι και η συσταδοποίηση (Anandarajan et. al., 2019).

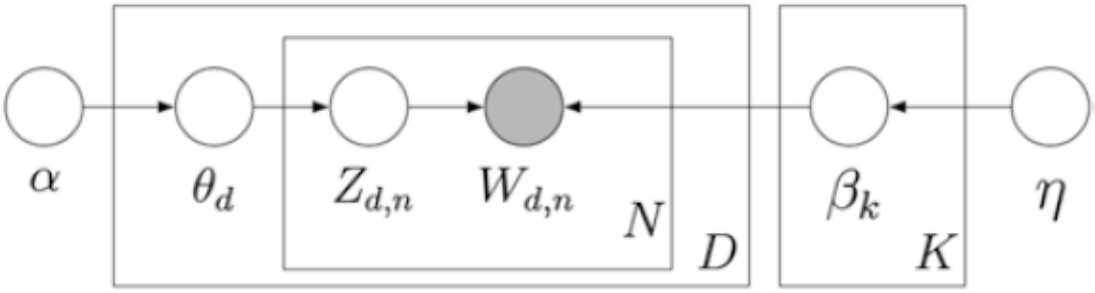
2.7 Topic Modeling

Το topic modeling είναι ένα πιθανοτικό μοντέλο που ανήκει στις μεθόδους της μη επιβλεπόμενης ανάλυσης και μας παρέχει θεματική πληροφορία μέσα σε ένα κείμενο. Τα topics αναπαριστώνται ως κατανομή πιθανότητας πάνω στους όρους του κειμένου. Ανάλογα με το μοντέλο ένα κείμενο μπορεί ανήκει σε ένα ή σε περισσότερα topics. Εμείς στη παρούσα εργασία θα ασχοληθούμε με το μοντέλο της δεύτερης περίπτωσης. Σε αυτή τη περίπτωση ο αριθμός των topics είναι προκαθορισμένος πριν την εφαρμογή του μοντέλου. Για την ανάλυση χρησιμοποιείται είτε ο πίνακας Term Document Matrix ή ο Document Term Matrix που προκύπτει από την προεπεξεργασία των κειμένων.



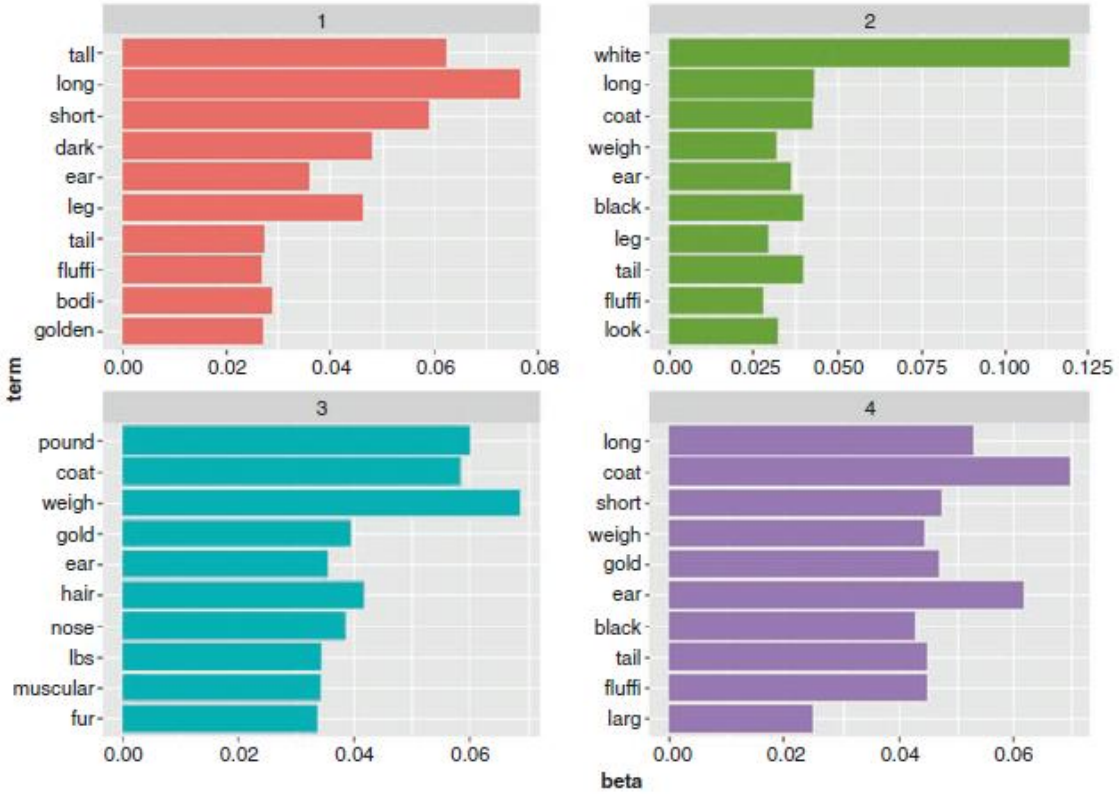
Εικόνα 20. Topic Modeling (Πηγή: Anandarajan et. al., 2019)

Ένα από τα μοντέλα που εφαρμόζει η μέθοδος του Topic Modeling είναι η κατανομή Latent Dirichlet Allocation ή LDA. Αν K ο αριθμός των topics, D ο αριθμός των documents, N είναι ο συνολικός αριθμός των λέξεων σε ένα document, Zd,n η κατανομή των όρων (terms), $W_{d,n}$ η εξεταζόμενη λέξη, α είναι η παράμετρος Dirichlet, η είναι η παράμετρος topic τότε κάθε document είναι μια μίξη θεμάτων με ποσοστά θ_d και κάθε term του documnet έχει β_κ αναθέσεις στα κ topics με βάση την εικόνα 21 η οποία αναπαριστά το μοντέλο. Οι Sun & Nie (2017) αναφέρουν την LDA ως ένα ιεραρχικό Bayesian μοντέλο.



Εικόνα 21. Latent Dirichlet Allocation (Anandarajan et. al., 2019)

Ας αναφέρουμε ένα παράδειγμα. Έστω ότι έχουμε μια συλλογή εγγράφων η οποία αναφέρεται σε διαφορετικές ράτσες σκύλων και για την οποία μας ενδιαφέρει να βρούμε τα χαρακτηριστικά κάθε ράτσας. Αν εφαρμόσουμε τον αλγόριθμο επιλέγοντας 4 topics τότε έχουμε το παρακάτω γράφημα που μας δείχνει τους δέκα πρώτους όρους ανά topic. Στο μοντέλο αυτό οι συνώνυμες λέξεις συνήθως βρίσκονται στο ίδιο topic πχ. στο topic 3 οι λέξεις pound weigh είναι συνώνυμες και για αυτό είναι πιο πιθανό να βρίσκονται μαζί στο ίδιο topic κάτι το οποίο δεν ισχύει σε άλλα μοντέλα (Anandarajan et. al., 2019).



Γράφημα 4. 4-Topic Model

Το μοντέλο αυτό κάνει μια ομαδοποίηση όπως η ασαφής συσταδοποίηση (soft clustering) και ουσιαστικά παράγει τα πιο πιθανά topics για να τα αναθέσει στα documents και επίσης παράγει τα πιο πιθανά terms για καθένα από τα topics.

2.7.1 Μετρικές του topic modeling

Για να εφαρμόσει ένας αναλυτής LDA Topic Modeling πρέπει πρώτα να επιλέξει τον ιδανικό αριθμό των topics. Για να το επιτύχει αυτό θα χρειαστεί να υπολογίσει 4 βασικές μετρικές οι οποίες υπάρχουν στον επίσημο ιστότοπο της Cran R στον σύνδεσμο https://cran.r-project.org/web/packages/ldatuning/vignettes/topics.html.

Αυτές είναι οι παρακάτω :

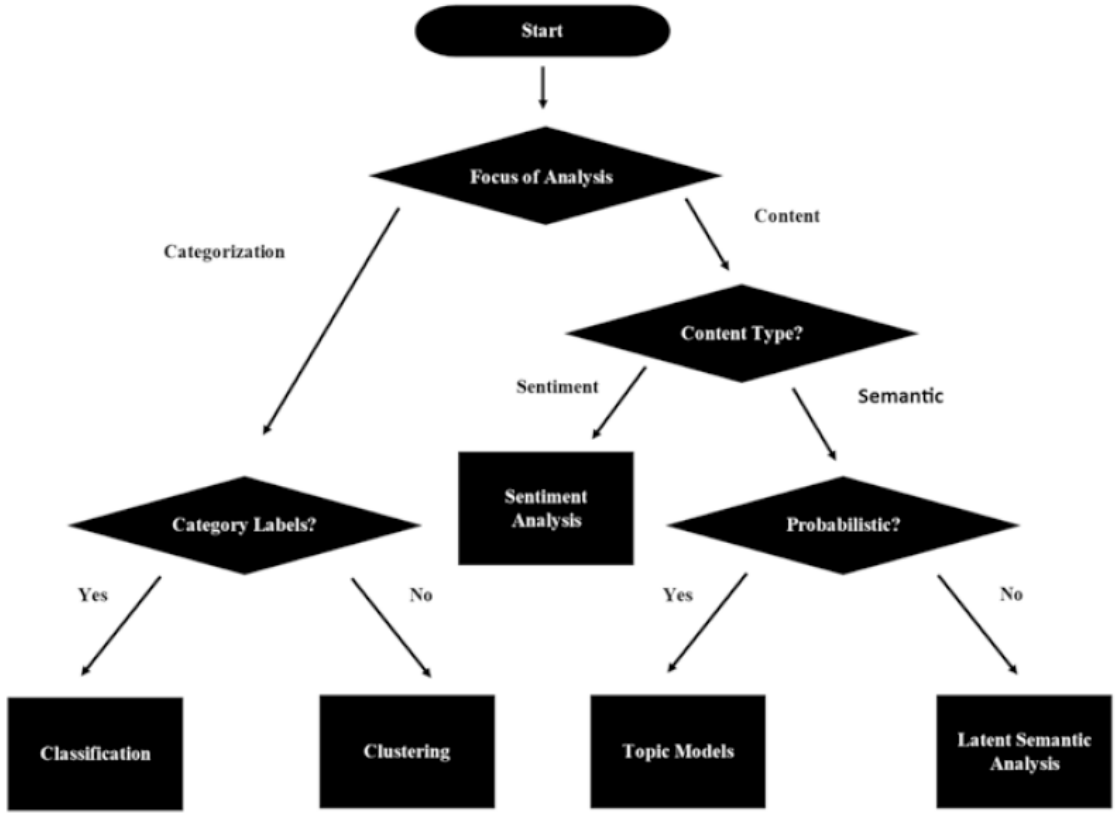
- CaoJuan2009 η οποία υπολογίζει την πυκνότητα του topic Z σε ακτίνα r, Density (Z, r). Αν Z είναι το topic και r μία ακτίνα, υπολογίζοντας τη μέση απόσταση (συνιμητόνου) ανάμεσα στο Z και τα άλλα topics, η πυκνότητα του topic Z σε ακτίνα

r, είναι ο αριθμός των θεμάτων που υπάρχουν μέσα στην ακτίνα r. Επιλέγουμε το πλήθος των topics με βάση την ελάχιστη τιμή της πυκνότητας ανάμεσα στα topics.

- Arun2010 υπολογίζει τις αποστάσεις των λέξεων για διαφορετικές τιμές των topics και επιλέγουμε τον αριθμό των topics με βάση την ελάχιστη δυνατή τιμή της.
- Griffiths2004 υπολογίζει τη λογαριθμική πιθανότητα εμφάνισης λέξεων (ήτοι δεδομένων) για διαφορετικές τιμές του αριθμού των topics. Επιλέγουμε εκείνο τον αριθμό topics όταν λαμβάνει τη μέγιστη δυνατή τιμή της ή στη περιοχή που μεγιστοποιείται η τιμή της.
- Deveaud2014 υπολογίζει τις αποστάσεις των λέξεων μεταξύ των topics για διαφορετικές τιμές των topics και επιλέγουμε τη μέγιστη δυνατή τιμή της.

2.8 Επιλογή Μοντέλου Εξόρυξης

Η επιλογή της μεθόδου που θα εφαρμόσουμε κρίνεται από το σκοπό της ανάλυσης, αν δηλαδή θέλουμε να ομαδοποιήσουμε τα έγγραφά μας ή αν μας ενδιαφέρει το περιεχόμενο των κειμένων π.χ. που αφορά την ανάλυση συναισθήματος που προκύπτει ή αν αυτό που αναζητούμε είναι σημασιολογική πληροφορία. Αν τα δεδομένα μας έχουν ετικέτα κατηγορίας συχνά επιλέγουμε την ανάλυση κατηγοριοποίησης η οποία μας δίνει και τη δυνατότητα πρόβλεψης μέσω ενός μοντέλου. Αν πάλι δεν έχουμε τα δεδομένα μας κατηγοριοποιημένα τότε συνήθως επιλέγουμε την συσταδοποίηση. Τέλος, αν μας ενδιαφέρει το σημασιολογικό περιεχόμενο μιας συλλογής εγγράφων τότε επιλέγουμε είτε το Topic Modeling είτε το Latent Semantic Analysis. Στην παρακάτω εικόνα απεικονίζεται η καταλληλότερη μέθοδος ανά περίπτωση (Anandarajan et. al., 2019).



Εικόνα 22. Επιλογή μεθόδου (Anandarajan et. al., 2019)

2.9 Εξόρυξη κειμένων στην επιστημονική βιβλιογραφία

Η μελέτη και η ανασκόπηση της επιστημονικής βιβλιογραφίας αποτελεί έναν από τους σημαντικότερους πυλώνες της επιστημονικής έρευνας. Η ανάλυση του σώματος της επιστημονικής βιβλιογραφίας βρίσκει εφαρμογή σε περιπτώσεις όπως η καλύτερη σύσταση σχετικών με κάποιο τομέα άρθρων (Gulo, et. al., 2015a), ο εντοπισμός περιπτώσεων απάτης, η κατανόηση των τάσεων της επιστήμης, η συσχέτιση ανάμεσα στην επιστημονική έρευνα και την τεχνολογική εξέλιξη (Gopal, 2018), ο εντοπισμός των ανερχόμενων θεμάτων στις επιστήμες (Small, et al., 2014) κ.ά.

Επίσης η βιβλιογραφική ανασκόπηση επιστημονικών άρθρων αποτελεί την πιο σημαντική φάση της έρευνας που έρχεται να διαπιστώσει την τελευταία λέξη της τεχνολογίας σε ένα συγκεκριμένο ερευνητικό πεδίο και επιπρόσθετα να αναγνωρίσει το χάσμα και τις τυχόν προκλήσεις στην εκάστοτε ερευνητική περιοχή (Gulo et. al. 2015a).

Από την άλλη πλευρά οι τεχνικές εξόρυξης κειμένου σε συνδυασμό με την βιβλιομετρική ανάλυση άρθρων συμβάλουν στην ανακάλυψη άγνωστων προτύπων στο πεδίο της έρευνας, στον εντοπισμό νέων μεγάλων ακαδημαϊκών κλάδων και εν γένει στην ανίχνευση των τάσεων της έρευνας (Nie & Sun, 2017).

Στη βιβλιογραφική ανασκόπηση ο ερευνητής μπορεί, με τα εργαλεία της μηχανικής μάθησης, να εντοπίσει με ευστοχία τα επικρατέστερα θέματα γύρω από το αντικείμενο της ερευνάς του, να ανακαλύψει τις πιο συναφείς/σχετικές θεωρίες σχετικά με αυτό αλλά και να συλλέξει τις επικρατέστερες μεθοδολογίες που εφαρμόζονται. Τέτοιες τεχνικές, για την ανίχνευση υποκατηγοριών, που περιέχει το αντικείμενο της έρευνας, είναι η ιεραρχική συσταδοποίηση και η ανάλυση δικτύου αναφορών. Οι Valerde-Berrosco, Garrido-Arroyo, Burgos-Vileda & Morales-Cevallos (2020) σε μια βιβλιογραφική ανασκόπηση σχετικά με το e-learning εφαρμόζουν, μεταξύ άλλων, ιεραρχική συσταδοποίηση στις λέξεις-κλειδιά των άρθρων και έτσι εντοπίζουν μέσω της ομαδοποίησης τους (στα φύλλα του δενδρογράμματος) πιθανές κατηγορίες και υποκατηγορίες σχετικά με το e-learning.

Μια άλλη ενδιαφέρουσα προσέγγιση σχετικά με την εξόρυξη κειμένων στα επιστημονικά άρθρα (Gulo et. al., 2015a) προτείνει, για την ταξινόμηση των κειμένων, τη μέθοδο Naïve Bayes. Στη συγκεκριμένη εργασία το σύνολο εκπαίδευσης προέκυψε μετά από την ομαδοποίηση που έκαναν ειδικοί σε μια συλλογή κειμένων (περιλήψεων) του ερευνητικού τους πεδίου. Στη συνέχεια χρησιμοποιήθηκε ο classifier Naïve Bayes ο οποίος ταξινόμησε το σετ δοκιμών με Accuracy 98,22%.

Άλλες τεχνικές, μεταξύ άλλων, είναι το topic modeling με τον αλγόριθμο Latent Dirichlet Allocation (που έχουμε αναφέρει στην ενότητα 2.7), ενώ ακόμα μια ευρέως χρησιμοποιούμενη μέθοδος είναι η τεχνική της συσταδοποίησης k-means. Οι Nie & Sun (2017) σε μια μελέτη περίπτωσης σχετικά με το Design Research επιτυγχάνουν, με τις παραπάνω τεχνικές (topic modeling και k-means clustering), να εντοπίσουν μεγάλους ακαδημαϊκούς κλάδους στον τομέα αυτό. Με το topic modeling βρίσκουν τα topics που αφορούν το κάθε άρθρο και στη συνέχεια εφαρμόζουν k-means clustering. Με βάση τα τρία πρώτα topics που εμφανίζονται σε κάθε cluster και τις δέκα πιο συχνές λέξεις που υπάρχουν σε κάθε topic καταφέρνουν να προσδιορίσουν τους κλάδους του Design. Τέλος εφαρμόζοντας βιβλιομετρική ανάλυση και ανάλυση δικτύου αναφορών εντοπίζουν τις

τάσεις των βασικών θεμάτων του κάθε κλάδου π.χ. στο κλάδο του Interaction design υπερτερεί η φράση “social media”, στο Ergonomics design το “biomechanics” κλπ.

Από την άλλη πλευρά οι Gulo και Rubio (2015b) προτείνουν μια μεθοδολογία για τον άμεσο εντοπισμό θεμάτων μέσα σε μια μεγάλη συλλογή άρθρων χρησιμοποιώντας τη γλώσσα R. Στην ανάλυση αυτή έχουν ένα dataset 506 papers στο οποίο εκτελούν προ-επεξεργασία κειμένου στις περιλήψεις και με τη δημιουργία νέφους λέξεων εντοπίζουν οπτικά τις επικρατέστερες λέξεις. Στη συνέχεια εφαρμόζουν topic modeling και δημιουργούν topic networks αποτυπώνοντας με οπτικό τρόπο τις συνδέσεις των λέξεων.

Είναι επίσης σημαντικό να αναφέρουμε ότι οι πιο συχνά χρησιμοποιούμενες τεχνικές εξόρυξης στην επιστημονική βιβλιογραφία είναι κυρίως η συσταδοποίηση και η παλινδρόμηση. Σε μια βιβλιογραφική ανασκόπηση σχετικά με τις τεχνικές του data mining που εφαρμόζονται στις ακαδημαϊκές βιβλιοθήκες οι Siguenza-Guzman, L., Saquicela, V., Avila-Ordóñez, E., Vandewalle, J., & Cattrysse, D. (2015) καταλήγουν στο συμπέρασμα ότι οι πιο συχνές τεχνικές είναι από την συσταδοποίηση η k-means, η association rules από το association, οι linear και logistic regression από την παλινδρόμηση και τέλος τα decision trees από τους αλγορίθμους classification.

Εν κατακλείδι, η τυπική αναζήτηση ερευνητικών άρθρων από βιβλιογραφικές βάσεις δεδομένων δεν οδηγεί πάντα στην ανάκτηση των πιο σχετικών κειμένων με την προς αναζήτηση ερευνητική περιοχή αν και οι μηχανές αναζήτησης εντοπίζουν άρθρα στα οποία εμφανίζονται οι λέξεις αναζήτησης ή οι λέξεις κλειδιά (Gulo et. al., 2015a).

Έτσι λοιπόν αυτό το πρόβλημα έρχεται να επιλύσει με τις μεθόδους της η εξόρυξη γνώσης από κείμενο ή αλλιώς ανάλυση κειμένου και να συμβάλει στην ανακάλυψη γνώσης. Με την εφαρμογή των κατάλληλων μοντέλων, ανάλογα με το ζητούμενο της έρευνας, ο ερευνητής είναι σε θέση να κατηγοριοποιήσει ένα σύνολο μιας συλλογής εγγράφων, να εντοπίσει το σημασιολογικό τους περιεχόμενο καθώς και να ανακαλύψει νέες υποκατηγορίες/τομείς. Σε κάθε περίπτωση η επιλογή των κριτηρίων ανάλυσης εξαρτάται από τις προσδοκίες του ερευνητή, τον αντικειμενικό σκοπό της έρευνας και το ζητούμενο της ανάλυσης που επιδιώκει να εντοπίσει, ήτοι η τελική επιλογή των τεχνικών εξόρυξης στην επιστημονική βιβλιογραφία εξαρτάται και από τη φύση και το σκοπό της έρευνας (Siguenza-Guzman et. al., 2015).

ΚΕΦΑΛΑΙΟ 3

Πριν αναλύσουμε το θεωρητικό πλαίσιο αναφορικά με το αντικείμενο του τεχνικού χρέους, κρίνουμε σκόπιμο να τεκμηριώσουμε την επιλογή μας αυτή, δηλαδή του αντικειμένου αυτού, για την εφαρμογή των μεθόδων και αλγορίθμων μηχανικής μάθησης.

Η βιωσιμότητα ενός έργου πληροφορικής εξαρτάται από την δυνατότητα συντήρησης του, προκειμένου να ικανοποιεί τις λειτουργικές ανάγκες των χρηστών του οργανισμού ή της επιχείρησης. Κάποια έργα εγκαταλείπονται διότι η συντήρησή τους είτε είναι αναποτελεσματική ή κρίνεται ασύμφορη για τον οργανισμό.

Από τι εξαρτάται η συντήρηση λογισμικού; Πως εκτιμάται το κόστος συντήρησης; Υπάρχει μεθοδολογία ή κάποιες τεχνικές για τη μέτρησή του; Αυτά είναι κάποια σημαντικά ερωτήματα που απασχολούν τόσο τους μηχανικούς, σχεδιαστές, αναλυτές, developers αλλά κυρίως τους project managers (Ampatzoglou, Ampatzoglou, Avgeriou & Chatzigeorgiou, 2015a). Θεωρήσαμε λοιπόν ιδιαίτερα χρήσιμο να αναζητήσουμε κάθε περαιτέρω γνώση για το αντικείμενο αυτό μέσω της εξόρυξης δεδομένων και μηχανικής μάθησης.

3.1 Τεχνικό χρέος - Ορισμός

Το τεχνικό χρέος είναι μία μεταφορά που χρησιμοποιήθηκε αρχικά για να περιγράψει τον κώδικα που δεν είναι εντελώς σωστός, σε προγράμματα λογισμικού, αλλά επεκτάθηκε για να αναφερθεί σε θέματα, όπως είναι ο ανώριμος σχεδιασμός, η ελλιπής τεκμηρίωση και οι ημιτελείς δοκιμές (Cunningham, 1992).

Ο όρος αυτός χρησιμοποιείται όλο και περισσότερο για να περιγράψει την επίδραση της καθυστέρησης ορισμένων εργασιών συντήρησης λογισμικού, σε προγράμματα λογισμικού και είναι γεγονός ότι οι επαγγελματίες κατανοούν διαισθητικά ότι το τεχνικό χρέος μπορεί να μετατραπεί σε σοβαρό πρόβλημα εάν παραμείνει χωρίς επίβλεψη. Ωστόσο, άγνωστο παραμένει το πόσο σοβαρό θα είναι το πρόβλημα και κατά πόσο είναι χρήσιμη η ρητή μέτρηση και η διαχείριση του τεχνικού χρέους (Guo et al., 2011).

Ο Cunningham (1992) χρησιμοποίησε πρώτος τον όρο τεχνικό χρέος ως μεταφορά για την παράδοση του κώδικα στον πελάτη δηλαδή όταν για λόγους εξοικονόμησης χρόνου παραδίδεται κώδικας ο οποίος είναι μεν λειτουργικός αλλά δεν έχει την απαιτούμενη

ποιότητα. Η παραδοχή αυτή ταυτίζεται με τον όρο του τεχνικού χρέους. Εάν αυτό το χρέος είναι μικρό και εξοφληθεί άμεσα τότε η ζημιά είναι μικρή ως αμελητέα αφού το κέρδος που προκύπτει από την έγκαιρη παράδοση του έργου είναι σαφώς μεγαλύτερο. Ωστόσο, μετά την παράδοση θα πρέπει να αφιερωθεί χρόνος ώστε να ξαναγραφτούν κάποια κομμάτια του κώδικα με τον σωστό πλέον τρόπο. Εάν όμως αυτό δεν συμβεί τότε το έργο κινδυνεύει καθώς γίνεται συνεχώς όλο και πιο δύσκολη η συντήρηση και η επέκτασή του.

Το κόστος που προκύπτει από το τεχνικό χρέος και πρέπει να αποπληρωθεί αποτελείται από δύο μέρη. Το πρώτο αφορά την αποπληρωμή του ίδιου του χρέους. Ο χρόνος δηλαδή που απαιτείται για να διορθωθεί ο προβληματικός κώδικας αυτός καθ' αυτός και αυτό αναφέρεται ως κεφάλαιο του χρέους (principal). Το δεύτερο αφορά την αποπληρωμή των τόκων. Εάν ο αρχικός προβληματικός κώδικας αποτελεί βάση πάνω στην οποία αναπτύχθηκαν και άλλα κομμάτια του έργου, τότε πρέπει να αφιερωθεί χρόνος ώστε να διορθωθούν και αυτά (Mayer et. al., 2014). Ο τόκος με την σειρά του αποτελείται και αυτός από δύο μέρη, το ποσό που εκφράζει την επιπλέον προσπάθεια που θα χρειαστεί (interest amount) και την πιθανότητα αυτό να προκαλέσει αλυσιδωτά προβλήματα και στον υπόλοιπο κώδικα (interest probability) (Seaman et. al., 2011).

Μια άλλη προσέγγιση δίνεται από τους Ampatzoglou, Ampatzoglou, Avgeriou, & Chatzigeorgiou (2015b) σύμφωνα με την οποία το κεφάλαιο εκφράζει την προσπάθεια που απαιτείται για να διαχειριστούμε τη διαφορά ανάμεσα στο τρέχον και στο ιδανικό επίπεδο ποιότητας σε ένα τεχνούργημα λογισμικού ή σε ένα ολόκληρο σύστημα λογισμικού ενώ ο τόκος εμπλέκει την εκτίμηση των μελλοντικών δραστηριοτήτων συντήρησης.

Πέρα από το γεγονός ότι το τεχνικό χρέος αποτελεί ένα πρόβλημα στην ανάπτυξη λογισμικού, το ενδιαφέρον της έρευνας οφείλεται και στη διττή φύση του καθώς εμπλέκει δύο επιστημονικούς τομείς αυτόν του λογισμικού και αυτού των οικονομικών (Ampatzoglou et. al., 2015b). Ο παραλληλισμός αυτός ήταν εύστοχος καθώς αποτελεί την γέφυρα μεταξύ των μηχανικών λογισμικού και των διευθυντικών στελεχών στους οποίους οι οικονομικοί όροι είναι πιο οικείοι και έτσι μπορούν να αντιληφθούν την σημασία του. Άλλωστε το αποτέλεσμα του τεχνικού χρέους σε ένα έργο δεν είναι εμφανές με την πρώτη ματιά καθώς είναι ένα ποιοτικό μέγεθος το οποίο μπορεί να προκαλέσει προβλήματα μακροπρόθεσμα ακόμα και αν την δεδομένη στιγμή φαίνεται πως όλα λειτουργούν όπως πρέπει.

Είναι λοιπόν σημαντικό να γνωρίζουμε το μέγεθος του τεχνικού χρέους το οποίο είναι αποδεκτό να υπάρχει ως ένα βαθμό αλλά σε κάθε περίπτωση πρέπει να είμαστε σε θέση να αξιολογήσουμε ποιο ποσοστό του πρέπει να αποπληρωθεί και σε ποιο θα δώσουμε προτεραιότητα. Αυτό εξαρτάται από την κρισιμότητα της μονάδας λογισμικού σε σχέση με το σύνολο του προγράμματος αλλά και άλλους παράγοντες όπως οι απαιτήσεις του έργου η τρέχουσα κατάστασή του καθώς και η προγραμματισμένη επαναχρησιμοποίηση συγκεκριμένου μέρους του κώδικα (Eisenberg, 2012).

Ως εκ τούτου, η διαχείριση του τεχνικού χρέους είναι απαραίτητη και αυτό που απαιτείται είναι η ποσοτικοποίηση τόσο του κεφαλαίου που απαιτείται για την ανάπτυξη λογισμικού όσο και του τόκου που αφορά δηλαδή τη συντήρησή του (Ampatzoglou, Michailidis, Sarikyriakidis, Ampatzoglou, Chatzigeorgiou, & Avgeriou, 2018).

Εκτιμώντας λοιπόν την αξία που θα είχε, για τη βιωσιμότητα του λογισμικού, μία αξιόπιστη πρόβλεψη της μελλοντικής συντήρησης ενός συστήματος λογισμικού, και επειδή πρόκειται για έναν νέο τομέα θεωρήσαμε χρήσιμη την εφαρμογή τεχνικών εξόρυξης στο αντικείμενο αυτό προκειμένου να ανακαλύψουμε τυχόν άλλους τομείς/υποκατηγορίες που επεκτείνεται καθώς και κάθε ευρύτερη γνώση που ενδεχομένως υπάρχει σχετικά με το ζήτημα του τεχνικού χρέους.

3.2 Κατηγοριοποίηση του τεχνικού χρέους

Υπάρχουν διαφορετικοί τρόποι κατηγοριοποίησης του τεχνικού χρέους οι οποίοι εξαρτώνται από τη σκοπιά με την οποία το εξετάζει κάποιος. Η κατηγοριοποίηση αυτή μπορεί να αφορά το τι επηρεάζει το χρέος, από ποιον προήλθε ή την αιτία από την οποία δημιουργήθηκε.

Μια πρώτη βασική κατηγοριοποίηση βασίζεται στη πρόθεση του προγραμματιστή αν δηλαδή το τεχνικό χρέος προκύπτει ηθελημένα ή όχι από τον ίδιο. Χωρίζεται λοιπόν σε ακούσιο και εκούσιο. Υπάρχουν περιπτώσεις όπου οι ίδιοι οι προγραμματιστές δεν έχουν την απαραίτητη γνώση και οι τεχνικές που χρησιμοποιούν δεν είναι οι ενδεδειγμένες. Αυτό είναι το ακούσιο χρέος. Υπάρχουν όμως και περιπτώσεις όπου το χρέος δημιουργείται εκούσια, εν γνώσει δηλαδή των προγραμματιστών και είναι μια απόφαση που λαμβάνεται

λόγω των χρονικών περιορισμών παράδοσης του προϊόντος στον πελάτη (Mayr, Plosch, & C. Korner, 2014).

Μία άλλη ταξινόμηση δίνεται από τους Seaman & Guo (2011) με βάση την οποία έχουμε τις εξής βασικές κατηγορίες:

- **Χρέος ελέγχου (testing debt):** Όταν ένα κομμάτι κώδικα βγαίνει στην παραγωγή χωρίς να έχει γίνει επιθεώρησή του και χωρίς να υπάρχουν δομές όπως τα unit tests που να εγγυώνται την σωστή λειτουργία του, τότε δημιουργείται χρέος.
- **Χρέος ελαττωμάτων (defect debt):** Υπάρχουν αρκετές περιπτώσεις στις οποίες εντοπίζεται ένα ελάττωμα στην λειτουργία του προγράμματος ωστόσο λαμβάνεται η απόφαση να μην διορθωθεί την δεδομένη χρονική στιγμή.
- **Χρέος τεκμηρίωσης (documentation debt):** Κάθε κομμάτι κώδικα πρέπει να συνοδεύεται από επαρκή τεκμηρίωση ώστε να μπορεί να το επεξεργαστεί και να το τροποποιήσει κατάλληλα οποιοσδήποτε προγραμματιστής και όχι μόνο ο αρχικός δημιουργός. Αυτό το κομμάτι τεχνικού χρέους είναι αρκετά σύνηθες ιδιαίτερα στα μεγάλα έργα λογισμικού. Η τεκμηρίωση συνήθως περιορίζεται μόνο στα ιδιαίτερα πολύπλοκα μέρη του έργου ενώ πολλές φορές ακόμα και αυτή είναι ελλιπής.
- **Σχεδιαστικό χρέος (design debt):** Αυτό το κομμάτι του χρέους αφορά παραβιάσεις που γίνονται στην χρήση των ενδεδειγμένων προγραμματιστικών τεχνικών αλλά και του αρχιτεκτονικού μοντέλου του έργου.

Οι Alves, Mendes, de Mendonça, Spínola, Shull & Seaman (2016) προχωρούν σε μία πιο αναλυτική περιγραφή κατηγοριών. Αρχικά αναγνωρίζουν πως υπάρχει το χρέος ελέγχου, το χρέος τεκμηρίωσης καθώς και το χρέος ελαττωμάτων αλλά η κατηγορία του σχεδιαστικού χρέους αναλύεται περισσότερο ενώ τέλος εισάγονται και νέες κατηγορίες.

Σύμφωνα με τους Mendes et. al. (2016) το σχεδιαστικό χρέος αναλύεται στις εξής τρεις υποκατηγορίες:

- **Σχεδιαστικό χρέος (design debt)**: Προβλήματα που αφορούν τις αποφάσεις κατά την σχεδίαση του λογισμικού.

- **Χρέος κώδικα (code debt)**: Προβλήματα που αφορούν αμιγώς το κομμάτι του κώδικα όπως για παράδειγμα η μεγάλη πολυπλοκότητα και δυσχεραίνουν την συντήρησή του.
- **Αρχιτεκτονικό χρέος (architectural debt)**: Προβλήματα που αφορούν την δομή του έργου όπως οι παραβιάσεις αρθρωτότητας (modularity violations).

Επιπρόσθετα όμως πέρα από την αναγνώριση του χρέους ελέγχου (test debt) εδώ εντοπίζεται και ένα δεύτερο σχετικό μέγεθος:

- **Χρέος αυτοματοποιημένου ελέγχου (automated test debt):** Περιλαμβάνει τον χρόνο που απαιτείται προκειμένου να υλοποιηθούν οι αυτόματοι έλεγχοι που θα διασφαλίζουν την σωστή λειτουργία του λογισμικού.

Επίσης οι νέες κατηγορίες χρέους που εισάγονται από τους ανωτέρω συγγραφείς αναφέρονται παρακάτω:

- **Χρέος προδιαγραφών (requirements debt)**: Υπάρχουν περιπτώσεις όπου παρόλο που η σωστή υλοποίηση είναι γνωστή και εφικτή αναγκαστικά παραβιάζονται κάποιες αρχές προκειμένου το τελικό προϊόν να ανταποκρίνεται στις ανάγκες του πελάτη με συνεπακόλουθη επίπτωση στις προδιαγραφές ασφάλειας.
- **Χρέος υποδομών (infrastructure debt)**: Όταν οι διαθέσιμες υποδομές και το υλικό δεν ικανοποιούν τις απαιτήσεις του έργου τότε η διαδικασία ανάπτυξης μπορεί να επιβραδυνθεί σημαντικά και να καθυστερεί μια αναβάθμιση ή μια επιδιόρθωση.
- **Χρέος ανθρώπων (person debt)**: Υπάρχουν φορές όπου το ανθρώπινο δυναμικό δεν επαρκεί για να υλοποιήσει όλο το έργο στον απαιτούμενο χρόνο. ενώ η πρόσληψη νέου προσωπικού δεν σημαίνει ότι θα επιταχύνει την διαδικασία υλοποίησης. Αντιθέτως βραχυπρόθεσμα θα την επιβραδύνει καθώς αυτά τα νέα στελέχη πρέπει αρχικά να εκπαιδευτούν από τους πιο έμπειρους συναδέλφους προκειμένου να συνεισφέρουν στο συγκεκριμένο έργο.

- **Χρέος διαδικασιών (process debt)**: Κάθε εταιρεία ακολουθεί συγκεκριμένες διαδικασίες και υπάρχουν περιπτώσεις όπου οι διαδικασίες αυτές μπορεί να μην είναι το ίδιο αποτελεσματικές για όλα τα έργα ενός οργανισμού.
- **Χρέος οικοδόμησης έργου (build debt)**: Αφορά την διαδικασία που ακολουθείται για να "χτιστεί" το έργο η οποία μπορεί να είναι σημαντικά αργή ειδικά όταν ο κώδικας δεν είναι σωστά δομημένος και όταν υπάρχουν μέσα σε αυτόν χαρακτηριστικά που δεν επιθυμεί ο πελάτης.
- **Χρέος υπηρεσιών (service debt)**: Αυτό το είδος χρέους εμφανίζεται όταν γίνεται λανθασμένη επιλογή των web services κατά την υλοποίηση με αποτέλεσμα να υπάρχουν ασυμφωνίες μεταξύ της πληροφορίας που παρέχεται από αυτές και αυτής που χρειάζεται τελικά το σύστημα για να λειτουργήσει.
- **Χρέος χρηστικότητας (usability debt)**: Αποφάσεις χρηστικότητας οι οποίες δεν είναι οι κατάλληλες και τελικά θα χρειαστεί να αναθεωρηθούν και να προσαρμοστούν.
- **Χρέος εκδόσεων (versioning debt)**: Προβλήματα μεταξύ των διαφορετικών εκδόσεων του κώδικα.

ΚΕΦΑΛΑΙΟ 4

Εργαλεία – Λογισμικά

Στο παρόν κεφάλαιο θα περιγράψουμε τα εργαλεία που χρησιμοποιήσαμε τόσο για την εξόρυξη των δεδομένων όσο και για την επεξεργασία του dataset που επιλέξαμε.

4.1 Η γλώσσα προγραμματισμού R

Η R είναι μια γλώσσα που χρησιμοποιείται κυρίως για στατιστικούς υπολογισμούς, για την παραγωγή γραφικών απεικονίσεων και για την επεξεργασία και ανάλυση των δεδομένων κατά την Εξόρυξη Δεδομένων.

Η υλοποίηση της R βασίστηκε στη γλώσσα προγραμματισμού S, την οποία δημιούργησε ο John Chambers, όσο βρισκόταν στα Bell Labs. Η R δημιουργήθηκε από τους Ross Ihaka και Robert Gentleman, στο πανεπιστήμιο Auckland στη Νέα Ζηλανδία. Τα τελευταία χρόνια έχει γίνει πολύ δημοφιλής και πλέον αναπτύσσεται από μια ομάδα ανθρώπων, γνωστή ως R Development Core Team.

Οι βασικότεροι λόγοι για τους οποίους έγινε τόσο δημοφιλής η R είναι η ευκολία στην εκμάθησή της, η συμβατότητά της με τα πιο διαδεδομένα λειτουργικά συστήματα δηλαδή Linux, Mac OS και Windows, και το ότι διαθέτει έναν μεγάλο αριθμό έτοιμων πακέτων με καλογραμμένα εγχειρίδια χρήσης, και τέλος το γεγονός ότι είναι δωρεάν διαθέσιμη (Βερύκιος κ.α., 2015).

Παρέχει όλες τις κοινές αλλά και λιγότερο κοινές τεχνικές όπως και τις πιο πρόσφατες προσεγγίσεις της εξόρυξης δεδομένων. Ο βασικός τρόπος λειτουργίας της, είναι η δημιουργία scripts. Εκτός από την δημιουργία απλών εντολών, ο χρήστης μόλις εξοικειωθεί μπορεί να δημιουργήσει πολύ πιο σύνθετα σενάρια και να εκτελέσει πολύπλοκες εργασίες σε ότι αφορά την εξόρυξη δεδομένων. Είναι πολύ σημαντικό ότι αυτά τα σενάρια με τις εντολές μπορούν να σωθούν σε αρχείο με .R επέκταση. Μπορούν στη συνέχεια να επαναλαμβάνονται και να διορθώνονται σύμφωνα με τις απαιτήσεις του χρήστη και να παράγουν χρήσιμες πληροφορίες.

Μερικά από τα πλεονεκτήματα της χρήσης της R ως εργαλείο εξόρυξης δεδομένων είναι, μεταξύ άλλων, τα παρακάτω:

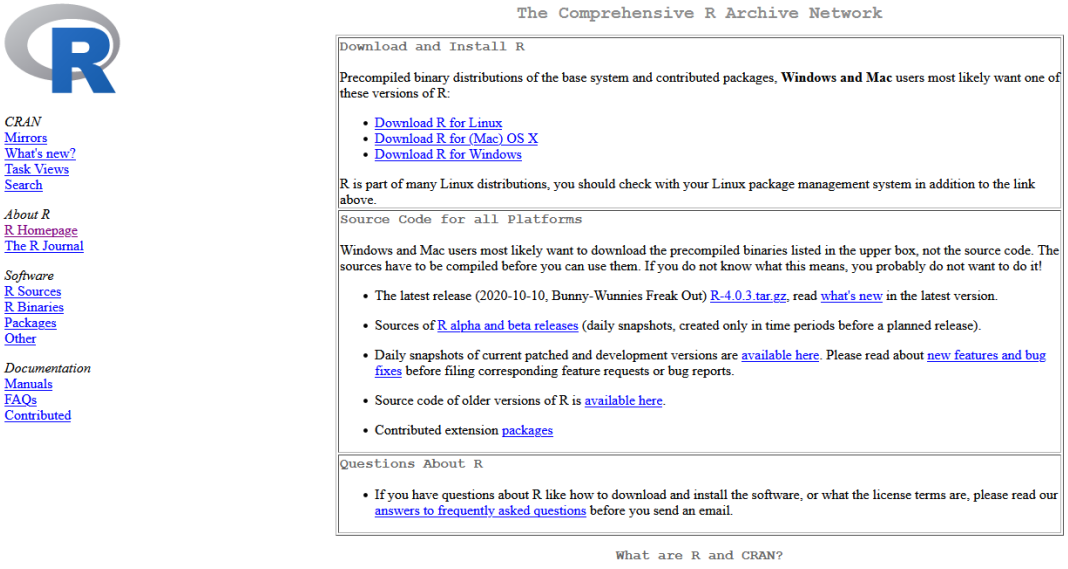
- Είναι το πιο ολοκληρωμένο διαθέσιμο πακέτο στατιστικής ανάλυσης. Ενσωματώνει όλες τις βασικές στατιστικές δοκιμές, μοντέλα και αναλύσεις καθώς και μία ολοκληρωμένη γλώσσα διαχείρισης και επεξεργασίας δεδομένων.
- Συντηρείται από μία βασική ομάδα 19 προγραμματιστών, μεταξύ αυτών και σημαντικοί στατιστικολόγοι.
- Οι γραφικές δυνατότητες της είναι εξαιρετικές, προσφέροντας μία πλήρως προγραμματιζόμενη γλώσσα γραφικών η οποία υπερτερεί των περισσότερων στατιστικών πακέτων και πακέτων γραφικών.
- Είναι ελεύθερο λογισμικό ανοικτού κώδικα, πράγμα που σημαίνει ότι οποιονδήποτε μπορεί να την χρησιμοποιήσει και το σημαντικότερο να την τροποποιήσει/προσαρμόσει ανάλογα με τις απαιτήσεις του.
- Δεν έχει περιορισμούς στην άδεια χρήσης, μπορεί να λειτουργήσει σε οποιοδήποτε λειτουργικό σύστημα και οποιοσδήποτε μπορεί να παρέχει διορθώσεις σφαλμάτων, τροποποιήσεις, βελτιώσεις κώδικα ακόμα και νέα πακέτα.
- Μπορούμε να επεκτείνουμε τη λειτουργικότητά της με βιβλιοθήκες που ονομάζονται πακέτα ενώ ο πιο δημοφιλής χώρος διαμοιρασμού πακέτων είναι το Comprehensive R Archive Network (CRAN) το οποίο περιέχει πάνω από 10.000 πακέτα. Πακέτα παρέχει και σε πολλά άλλα αποθετήρια που ειδικεύονται σε θέματα όπως η οικονομετρία, εξόρυξη δεδομένων, χωρική ανάλυση, βιοπληροφορική κ.α.
- Συνεργάζεται καλά με άλλα εργαλεία, όπως π.χ. για την εισαγωγή δεδομένων μπορεί να φορτώσει αρχεία από CSV, SAS και SPSS, ακόμη και κατευθείαν από Excel, SQL κτλ. Επίσης οι εξαγωγές γραφικών μπορούν να παραχθούν σε πολλές μορφές όπως PDF, JPG, PNG και SVG.
- Υπάρχουν πολλά φόρουμ χρηστών στα οποία μπορούν να τεθούν ερωτήσεις και να απαντηθούν εύστοχα από ικανούς προγραμματιστές ενίοτε και από τους ίδιους τους δημιουργούς.

Κάποια μειονεκτήματα της είναι :

- Κάνει κακή διαχείριση της μνήμης και γενικά καταναλώνει πολλή μνήμη από το σύστημα στο οποίο εκτελείται
- Θεωρείται αργή αλλά με τη χρήση πακέτων έχει αναβαθμίσει τις επιδόσεις της
- Δεν μπορεί να χρησιμοποιηθεί σαν back-end server για υπολογισμούς διότι υστερεί σε θέματα ασφάλειας.

4.2 Εγκατάσταση της R και του R studio

Από την επίσημη ιστοσελίδα της R https://cran.r-project.org/bin/windows/base/ κατεβάζουμε την R στον υπολογιστή μας όπως φαίνεται παρακάτω:

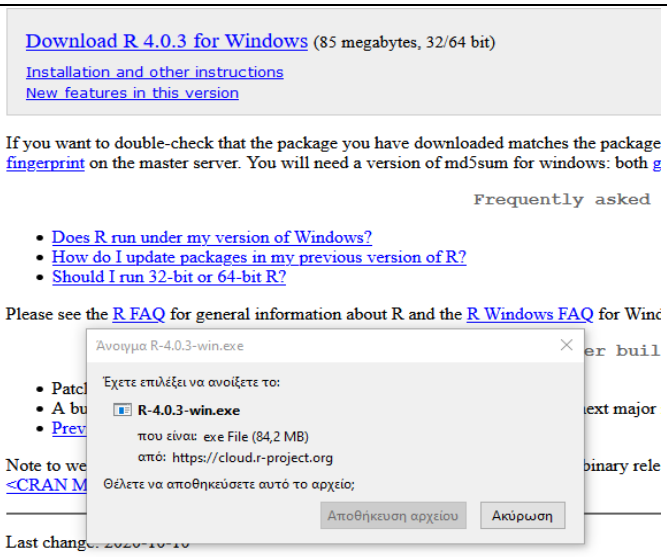


Εικόνα 23. Εγκατάσταση της R

Ο υπολογιστής που θα εργαστούμε έχει τα παρακάτω χαρακτηριστικά:

Windows 10 64 bit επεξεργαστή Intel Core i7 στα 3.4 GHZ και RAM 12GB.

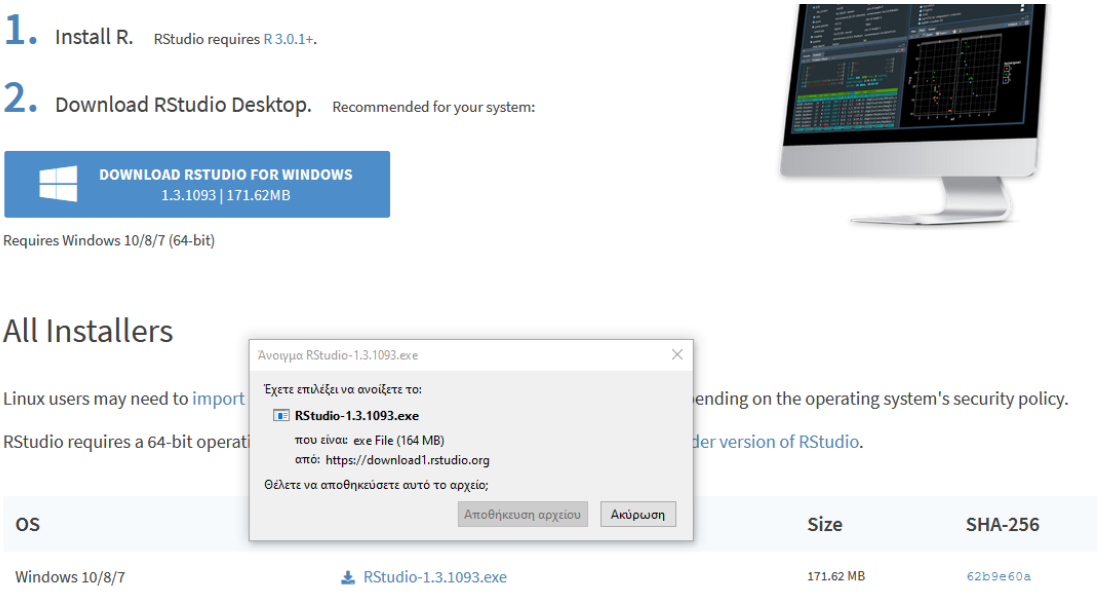
Επιλέγουμε την τρέχουσα έκδοση που υπάρχει για λειτουργικό των Windows και εκτελούμε το σχετικό αρχείο . exe.



Εικόνα 24. Εγκατάσταση της R

Ακολουθούμε τις σχετικές οδηγίες και ολοκληρώνουμε την εγκατάσταση.

Στη συνέχεια πρέπει να κατεβάσουμε και το R studio με το οποίο θα εργαστούμε. Από την επίσημη ιστοσελίδα https://www.rstudio.com/products/rstudio/download/ επιλέγουμε την Open Source άδεια Desktop όπως φαίνεται παρακάτω και εκτελούμε ομοίως το αντίστοιχο αρχείο exe στον υπολογιστή μας.



Εικόνα 25. Εγκατάσταση του R Studio

Για την εξόρυξη των δεδομένων μας χρησιμοποιήσαμε μια σειρά πακέτων και βιβλιοθηκών που διαθέτει η R για αυτό το σκοπό. Αναφέρουμε τα πιο σημαντικά :

- tm, stringi, NLP για την εξόρυξη κειμένου
- tidyverse το οποίο περιέχει μεταξύ άλλων τα πακέτα dplyr, tidyr ggplot2 για την διαχείριση των δεδομένων
- cluster για την εκτέλεση των αλγορίθμων συσταδοποίησης
- quanteda, topic models, ldaunting , devtools για την εκτέλεση του topic modeling
- caret, naivebayes, e1071 για την εφαρμογή του Naïve Bayes
- party, zoo, rpart για την εφαρμογή και αναπαράσταση του Decision Tree
- mlbench, class για την εκτέλεση του KNN πλησιέστερων γειτόνων
- gmodels για την αναπαράσταση της μήτρας σύγχυσης
- factoextra για την αναπαράσταση γραφημάτων
- RColorBrower, wordcloud για τη δημιουργία αναπαράστασης λέξεων ή φράσεων

4.3 Εργαλείο JabRef

Ένα άλλο εργαλείο το οποίο θα χρειαστούμε για την επεξεργασία του dataset είναι το λογισμικό JabRef. Το JabRef είναι ένα λογισμικό ανοιχτού κώδικα για τη διαχείριση και επεξεργασία αναφορών και παραπομπών, αρχείων τύπου bibtex, μέσω κατάλληλης διεπαφής που διαθέτει για την εισαγωγή, επεξεργασία και αναζήτηση δεδομένων που ανακτώνται από online διαδικτυακές βάσεις δεδομένων. Η εφαρμογή είναι υλοποιημένη σε Java και είναι συμβατή με Windows, Linux και Mac. Από την έκδοση 3.6 και έπειτα αδειοδοτείται από το MIT.

Υποστηρίζει πολλούς online επιστημονικούς καταλόγους όπως: ACM Portal, CiteSeer, CrossRef, DBLP, DOAJ, GVK, Google Scholar, IEEEXplore, INSPIRE-HEP, Medline, MathSciNet, SAO/NASA Astrophysics Data System, Springer, arXiv and zbMATH.

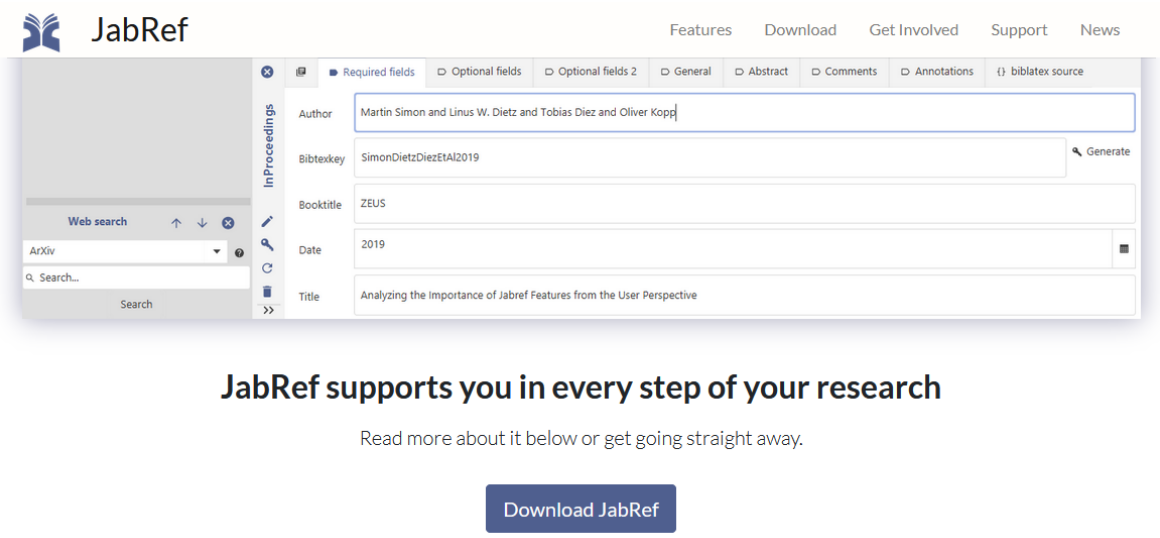
Έχει πολλές δυνατότητες για τη διαχείριση των αρχείων όπως:

- Διαθέτει λειτουργικότητες για την αναζήτηση, το φιλτράρισμα και την ανίχνευση διπλότυπων άρθρων.
- Υποστηρίζει δυνατότητες ιεραρχικής ομαδοποίησης βασισμένη σε λέξεις-κλειδιά σε όρους αναζήτησης κλπ.

Διαθέτει μεταξύ άλλων δυνατότητες παραμετροποίησης όπως στα αρχεία μεταδεδομένων, στα κλειδιά των παραπομπών στη μετονομασία των αρχείων κ.α.

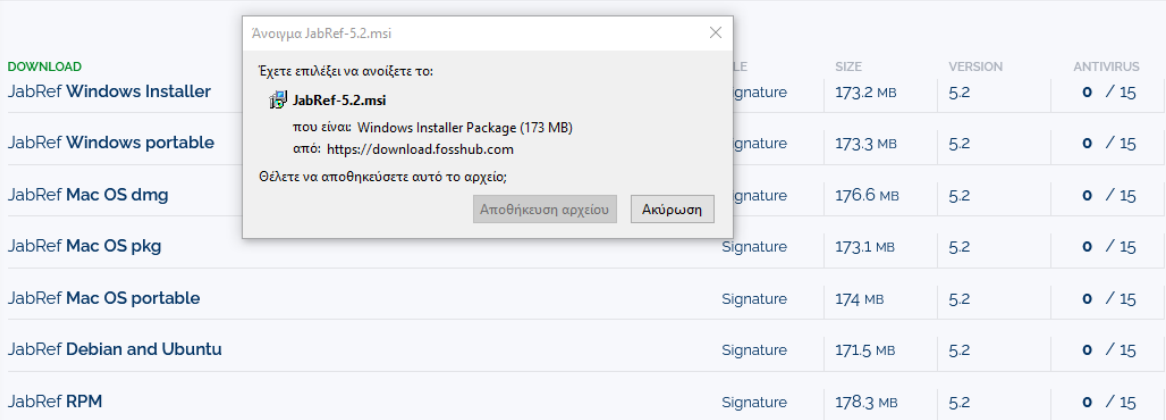
4.4 Εγκατάσταση JabRef

Από την επίσημη ιστοσελίδα του Jabref https://www.jabref.org/ κατεβάσαμε το λογισμικό όπως φαίνεται παρακάτω:



Εικόνα 26. Κατέβασμα του JabRef

Και στη συνέχεια εκτελούμε το αντίστοιχο αρχείο στον υπολογιστή μας.



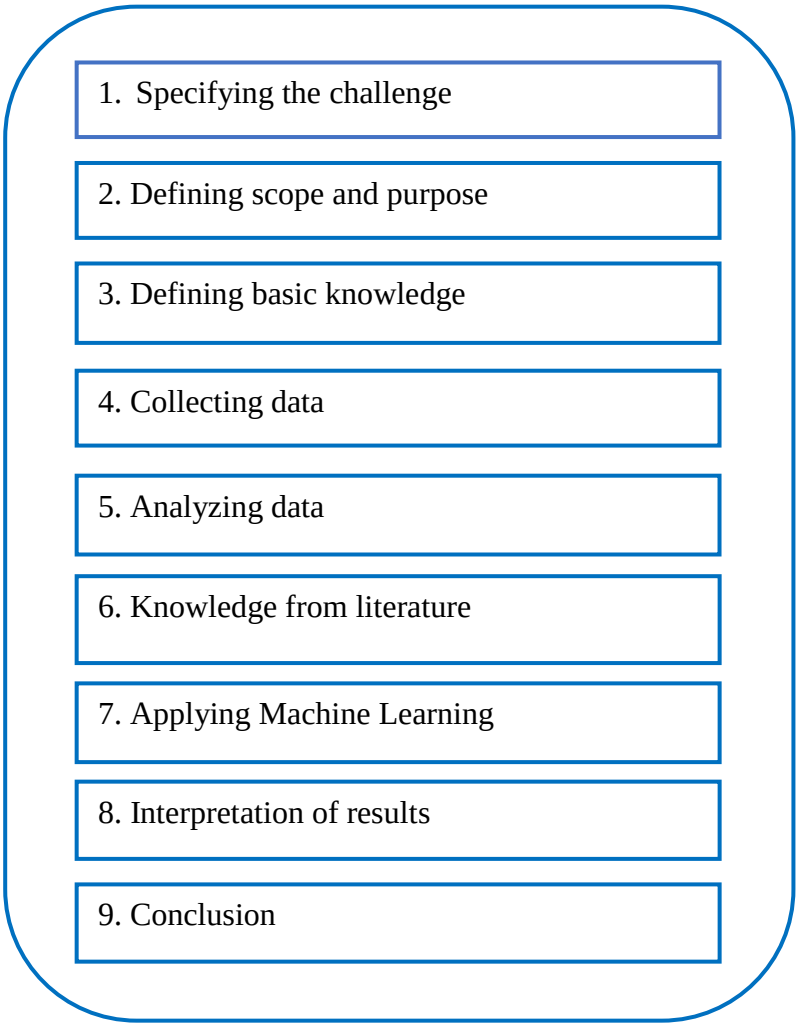
Εικόνα 27. Εγκατάσταση του JabRef

Με το εργαλείο αυτό θα επεξεργαστούμε το dataset πριν εφαρμόσουμε τις τεχνικές εξόρυξης δεδομένων.

ΚΕΦΑΛΑΙΟ 5

5.1 Μεθοδολογία Ερευνητικής Προσέγγισης

Πριν εισέλθουμε στην περιγραφή του προβλήματος και στα εργαλεία με τα οποία θα επεξεργαστούμε τα δεδομένα μας και θα εφαρμόσουμε τις κατάλληλες τεχνικές για την εξόρυξη γνώσης από την επιστημονική βιβλιογραφία, σχετικά με το αντικείμενο του τεχνικού χρέους, κρίνουμε σκόπιμο να κάνουμε μια εισαγωγή για τη μεθοδολογία που θα ακολουθήσουμε η οποία περιγράφει και τον τρόπο με τον οποίο εργαστήκαμε.



Εικόνα 28. Φάσεις της μελέτης περίπτωσης (Μπρασινίκας, 2020)

Η 1η φάση αφορά τον προσδιορισμό του προβλήματος. Στη φάση αυτή αναλύουμε το ερευνητικό πεδίο που θα μελετήσουμε με τις τεχνικές μηχανικής μάθησης, και περιγράφουμε το αντικείμενο της ανάλυσης – το τεχνικό χρέος στο τομέα της τεχνολογίας λογισμικού. Επίσης τεκμηριώνουμε την αξία σε γνώση που προσφέρει η μηχανική μάθηση στη μελέτη του τεχνικού χρέους, έναν νέο τομέα στην τεχνολογία λογισμικού που απασχολεί τόσο τους επαγγελματίες στο χώρο της ανάπτυξης έργων λογισμικού όσο και τους ακαδημαϊκούς όπως αποτυπώνεται στην επιστημονική βιβλιογραφία (Ampatzoglou, et. al., 2015a)

Η 2η φάση έχει να κάνει με το εύρος και την επιλογή του συνόλου δεδομένων που θα χρησιμοποιήσουμε καθώς και με τον προσδιορισμό του στόχου της ανάλυσης κειμένων, ο οποίος είναι η ανακάλυψη νέας γνώσης σχετικά με το τεχνικό χρέος καθώς και δημιουργία ενός μοντέλου ταξινόμησης για την κατηγοριοποίηση του συνόλου δεδομένων.

Η 3η φάση έχει να κάνει με τη βασική γνώση που απαιτήθηκε για την αντιμετώπιση του προβλήματος. Αφορά την απαιτούμενη τεχνική γνώση σχετικά με την εξόρυξη δεδομένων, καθώς και την εκμάθηση της γλώσσας R και των εργαλείων R studio και JabRef. Επίσης περιλαμβάνει τη τεχνική γνώση γύρω από τη μηχανική μάθηση, τις τεχνικές, μεθοδολογίες και τους αλγόριθμους. Εκτός αυτών, περιλαμβάνει και την βιβλιογραφική ανασκόπηση σχετικά με το τεχνικό χρέος, προς απόκτηση της απαραίτητης γνώσης που είναι προ απαιτούμενη για την κατανόηση, ερμηνεία και επεξήγηση των αποτελεσμάτων που θα προκύψουν από τους αλγορίθμους μηχανικής μάθησης.

Η 4η φάση αφορά τη συλλογή των δεδομένων από μια μεγάλη βιβλιογραφική βάση δεδομένων, τη Scopus στην οποία έχουμε πρόσβαση μέσω του ιδρύματος του ΕΑΠ.

Η 5η φάση περιλαμβάνει την απαιτούμενη επεξεργασία και προετοιμασία των δεδομένων με τη βοήθεια του εργαλείου JabRef καθώς και την περιγραφική και διερευνητική ανάλυση των δεδομένων. Στη φάση αυτή μελετάται το περιεχόμενο των περιλήψεων και στη συνέχεια, για ένα υποσύνολο δεδομένων, τα πλήρη κείμενα. Η φάση αυτή θεωρείται καθοριστική για την διεξαγωγή των επόμενων φάσεων και ειδικότερα για την κατανόηση και την ερμηνεία των αποτελεσμάτων.

Η 6η φάση έχει να κάνει με τη βιβλιογραφική ανασκόπηση σχετικά με την εξόρυξη κειμένων στην επιστημονική βιβλιογραφία. Μελετήθηκαν δηλαδή άρθρα που αφορούν την ανάλυση

κειμένων επιστημονικών ερευνητικών άρθρων και αποκτήθηκε πολύτιμη γνώση σχετικά με την επιλογή των μεθόδων μηχανικής μάθησης. Η φάση αυτή βοήθησε στην τελική επιλογή των αλγορίθμων που εφαρμόστηκαν.

Στη φάση 7 εφαρμόζονται αλγόριθμοι επιβλεπόμενης και μη επιβλεπόμενης μηχανικής μάθησης στο εργαλείο R-studio. Εκτελούνται οι απαραίτητες δοκιμές, βελτιώσεις και στη συνέχεια συγκρίνονται και αξιολογούνται τα αποτελέσματα των μοντέλων ταξινόμησης που εφαρμόστηκαν.

Στη φάση 8 συγκρίνουμε τα αποτελέσματα της μηχανικής μάθησης με τα εμπειρικά μας αποτελέσματα και στη συνέχεια ερμηνεύουμε συνολικά τα αποτελέσματα που προέκυψαν τα οποία θα μας βοηθήσουν στην εξαγωγή χρήσιμων συμπερασμάτων που αποτελούν τη φάση 9 στην οποία, μεταξύ άλλων, παρουσιάζουμε και τις μελλοντικές μας προτάσεις σχετικά με την ανάλυση κειμένων στο αντικείμενο του τεχνικού χρέους.

5.2 Περιγραφή Προβλήματος

Στη παρούσα εργασία επιλέχθηκε να διερευνηθεί το αντικείμενο του τεχνικού χρέους, όρος ο οποίος συνδέεται με τη συντήρηση λογισμικού στα έργα ανάπτυξης λογισμικού. Όπως αναφέρθηκε αναλυτικά στο κεφάλαιο 3 το τεχνικό χρέος είναι ένα πεδίο το οποίο απασχολεί όλο και περισσότερο τα τελευταία χρόνια τους αναλυτές, σχεδιαστές, προγραμματιστές και κυρίως τους project managers στα έργα ανάπτυξης λογισμικού. Αυτό αποτυπώνεται και στην ερευνητική βιβλιογραφία με τη δημοσίευση όλο και περισσότερων σχετικών με το αντικείμενο άρθρων (Ampatzoglou et. al., 2015a). Πρόκειται λοιπόν για έναν νέο τομέα για τον οποίο θεωρούμε ότι η εφαρμογή τεχνικών εξόρυξης θα συμβάλλει σημαντικά στην ανακάλυψη τυχόν άλλων τομέων/υποκατηγοριών που ενδεχομένως επηρεάζει ή/και επεκτείνεται και εν γένει σε κάθε ευρύτερη γνώση που ενδεχομένως συσχετίζεται.

Για την εφαρμογή μεθόδων εξόρυξης κειμένου στην ερευνητικά άρθρα συνήθως ως σύνολο δεδομένων χρησιμοποιείται ο τίτλος και η περίληψη των κειμένων (Gulo et. al., 2015b). Οι αναλυτές συνήθως κατεβάζουν από μια ή περισσότερες βιβλιογραφικές βάσεις δεδομένων ένα σύνολο δεδομένων σε μορφή csv με βασικά πεδία τον τίτλο, τον συγγραφέα, το έτος και την περίληψη. Ανάλογα με το αντικείμενο της έρευνας χρησιμοποιούν τις κατάλληλες

λέξεις αναζήτησης, και ανάλογα με τον σκοπό της έρευνας εφαρμόζουν τις κατάλληλες τεχνικές εξόρυξης προκειμένου να καταλήξουν στην αναζητούμενη γνώση.

Εμείς αυτό που θέλουμε να διερευνήσουμε είναι η ανακάλυψη νέας γνώσης που θα προκύψει μέσα από την εφαρμογή τεχνικών εξόρυξης σε μια συλλογή κειμένων αξιοποιώντας την πληροφορία που προέρχεται από την περίληψη των επιστημονικών άρθρων. Η βασική μας επιδίωξη είναι η εύρεση ενός μοντέλου κατηγοριοποίησης (ή και περισσοτέρων) το οποίο θα αποτελέσει εργαλείο με το οποίο θα μπορέσει ένας ερευνητής να κατηγοριοποιήσει ένα σύνολο δεδομένων ώστε να εντοπίσει, να ανατρέξει και να μελετήσει το υποσύνολο που τον ενδιαφέρει.

5.3 Επεξεργασία Βιβλιογραφικού Υλικού

Επιλέξαμε τη βιβλιογραφική βάση δεδομένων του Scopus διότι πρόκειται για μια μεγάλη βάση δεδομένων και επίσης έχουμε δυνατότητα πρόσβασης μέσω του ιδρύματος του ΕΑΠ. Το Scopus περιλαμβάνει περιλήψεις και αναφορές για ακαδημαϊκά άρθρα περιοδικών και εισαγάγαμε τον όρο αναζήτησης "*technical debt"* OR "*TD*" στον τίτλο του άρθρου, προκειμένου να εντοπίσουμε επιστημονικά άρθρα σε αυτό το αντικείμενο. Επιλέξαμε το πλήρες όνομα αλλά και το ακρωνύμιο TD προκειμένου να συμπεριλάβουμε όλα τα άρθρα με τη συγκεκριμένη ορολογία και να μην αποκλείσουμε κανένα άρθρο που πιθανόν να είχε μόνο το ακρωνύμιο στον τίτλο.

Βάση Δεδομένων	Query: Search term	Πλήθος αποτελεσμάτων
Scopus	"Technical debt" OR "TD"	623

Πίνακας 3. Search term

Από την αναζήτηση προέκυψε μια συλλογή 623 επιστημονικών άρθρων σε μορφή bibtex αρχείου το οποίο επεξεργαστήκαμε με το λογισμικό JabRef.

Η κατανομή των άρθρων ανά έτος φαίνεται στον ακόλουθο πίνακα. Η άντληση των δεδομένων έγινε πριν το τέλος του έτους 2020 κατά συνέπεια το dataset το οποίο θα επεξεργαστούμε περιλαμβάνει ένα μέρος από τα 84 άρθρα του έτους 2020.

Πλήθος Άρθρων	Έτος
84	2020
89	2019
66	2018
69	2017
69	2016
50	2015
55	2014
43	2013
48	2012
36	2011
28	2010
17	2009
6	2008
10	2007
2	2006
4	2005
2	2004
16	ως το 2003

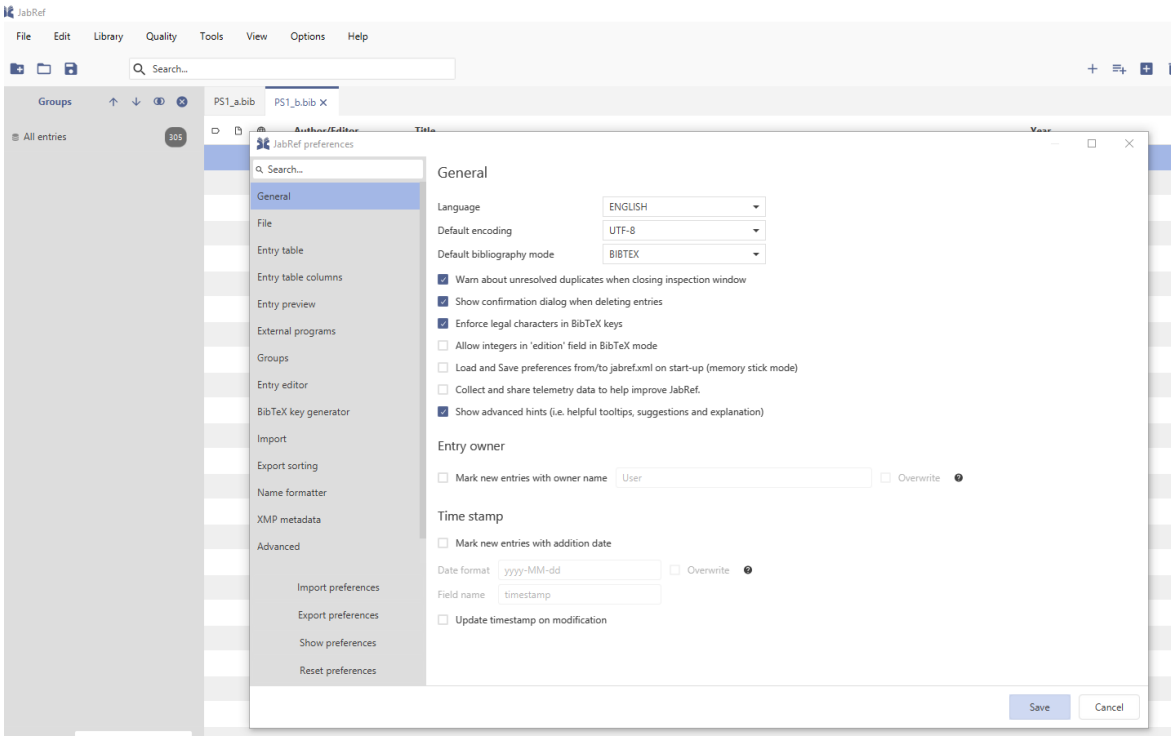
Πίνακας 4. Άρθρα ανά έτος

Γράφημα 5. Κατανομή Δημοσιεύσεων ανά Έτος

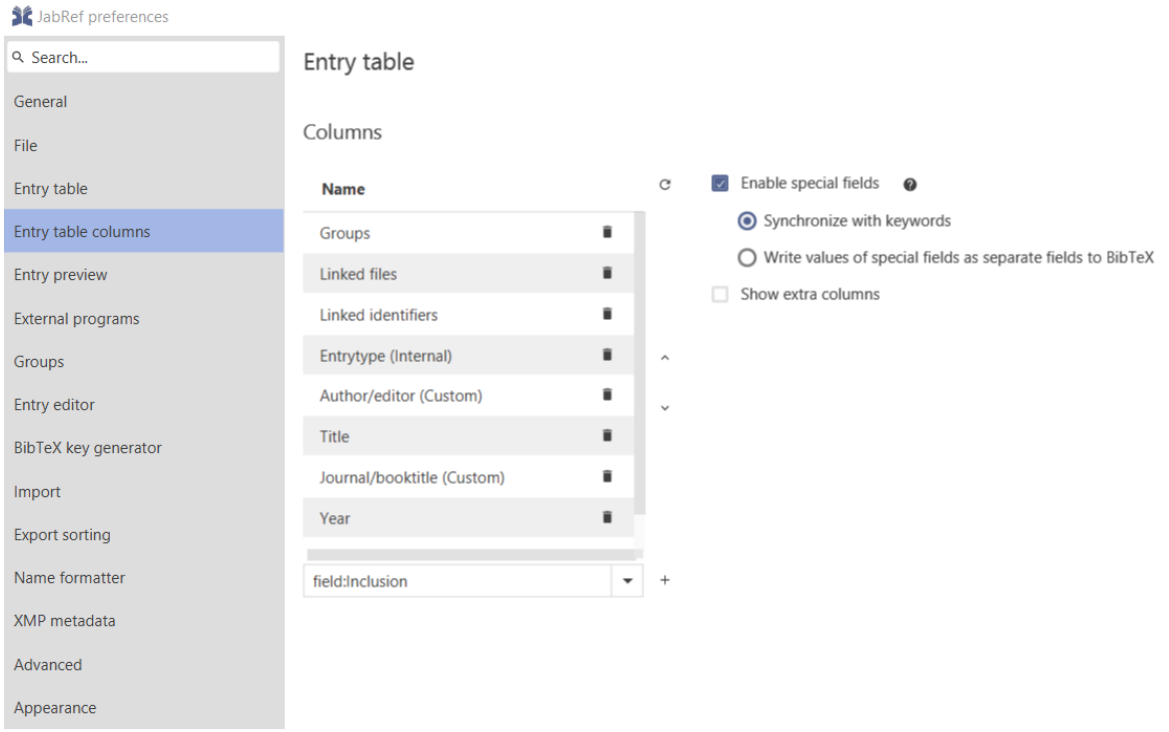
Στη συνέχεια κάναμε import into new library το αρχείο PS1.bib, το οποίο κατεβάσαμε από τη βιβλιογραφική βάση δεδομένων του Scopus, στο εργαλείο JabRef για περαιτέρω επεξεργασία.

Το JabRef διαθέτει τα απαιτούμενα πεδία για την επεξεργασία των επιστημονικών άρθρων και επίσης παρέχει τη δυνατότητα στον χρήστη να τα προσαρμόσει σε σχέση με τις ανάγκες επεξεργασίας του υλικού. Πριν προβούμε σε ανάλυση των περιλήψεων του αρχείου αυτού κρίνουμε σκόπιμο να το προ επεξεργαστούμε προκειμένου να εντοπίσουμε και εμπειρικά τομείς ή κατηγορίες που υπάρχουν γύρω από το θέμα του τεχνικού χρέους. Με αυτό τον τρόπο θα μπορέσουμε να συγκρίνουμε τα αποτελέσματα που θα προκύψουν από την εξόρυξη κειμένου με αυτά.

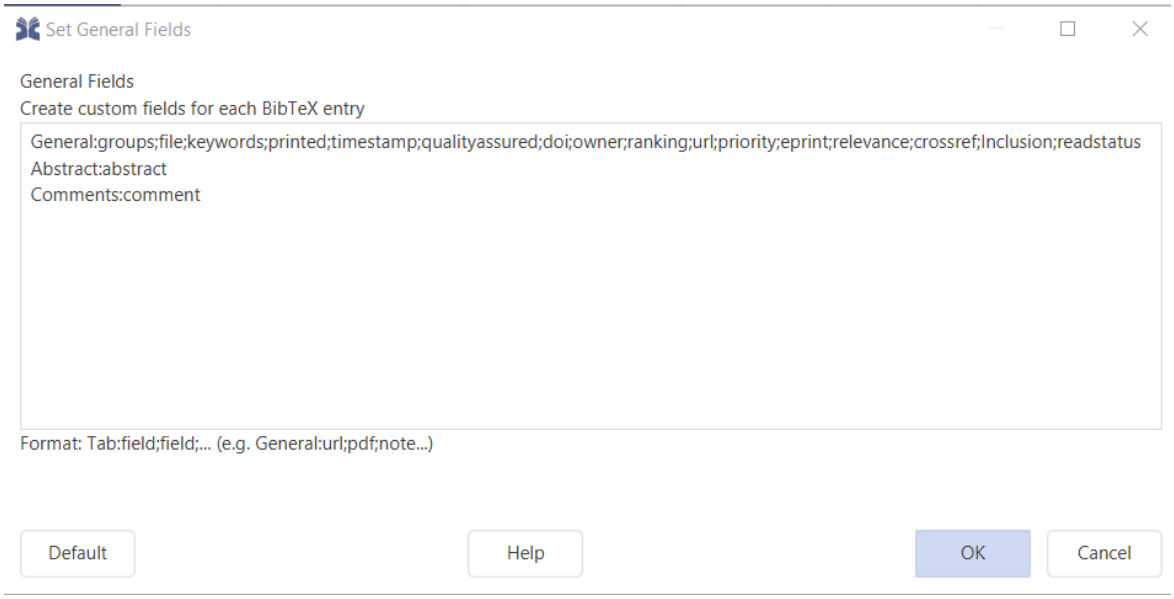
Στην εξεταζόμενη περίπτωση προκειμένου να κατηγοριοποιήσουμε το υλικό προσθέσαμε ένα επιπλέον πεδίο στο οποίο αποτυπώνουμε την κατηγορία στην οποία ανήκει το άρθρο μας.



Εικόνα 29. Δημιουργία νέου πεδίου επεξεργασίας

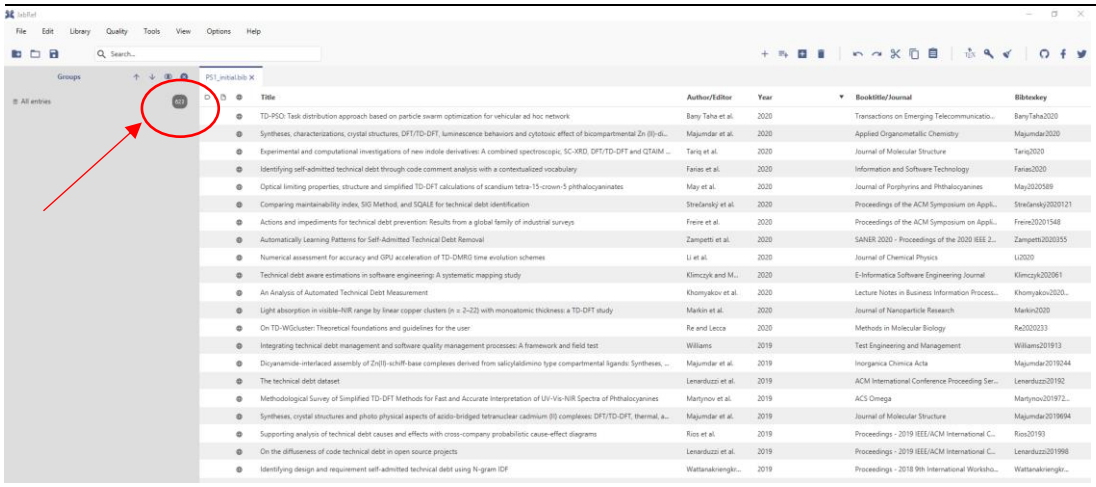


Εικόνα 30. Ορισμός ονόματος νέου πεδίου



Εικόνα 31. Ρυθμίσεις εμφάνισης του πεδίου *Inclusion*

Με αυτό τον τρόπο μας δίνεται η δυνατότητα να εντοπίζουμε πολύ εύκολα μέσω της αναζήτησης τα κείμενα που αφορούν την επιλεγμένη κατηγορία.



Εικόνα 32. Επεξεργασία υλικού

Στη συνέχεια προχωρήσαμε σε καθαρισμό των δεδομένων:

- Αφαίρεση διπλότυπων άρθρων
- Αφαίρεση άρθρων σε άλλη γλώσσα (πλην της αγγλικής)
- Αφαίρεση άρθρων με κενό το πεδίο Author

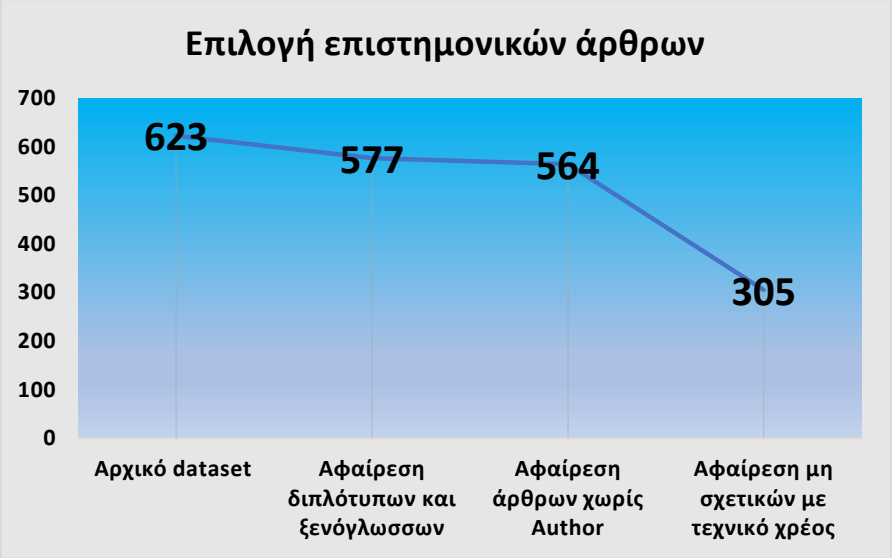
και έτσι το dataset τελικά έχει 564 άρθρα.

Dataset	Πλήθος
Αρχικό dataset	623
Αφαίρεση διπλότυπων και ξενόγλωσσων	577
Αφαίρεση άρθρων χωρίς Author	564

Πίνακας 5. Dataset

Στην επόμενη φάση αρχίσαμε να εξετάζουμε τα άρθρα με βάση τον τίτλο και την περίληψη προκειμένου να τα κατηγοριοποιήσουμε. Εδώ να αναφέρουμε ότι αρχικά δεν εφαρμόσαμε κάποιο/α κριτήρια για να κατηγοριοποιήσουμε τα δεδομένα μας, απλώς διαβάσαμε τους τίτλους και τις περιλήψεις τους προκειμένου να σχηματίσουμε μια πρώτη εικόνα για τα θέματα στα οποία επεκτείνεται. Στη φάση αυτή κατά την ανάγνωση τίτλων και περιλήψεων διαπιστώσαμε ότι το dataset περιλάμβανε και μη σχετικά με το τεχνικό χρέος άρθρα. Αυτό το πρόβλημα θα το αντιμετωπίσουμε με τις τεχνικές εξόρυξης κειμένου, δηλαδή θα εντοπίσουμε αυτές τις επιπλέον μη σχετικές κατηγορίες και θα τις εξαιρέσουμε από την περαιτέρω ανάλυση. Αυτά τα άρθρα αρχικά τα ομαδοποιήσαμε στην κατηγορία 0.

Στο ακόλουθο γράφημα φαίνεται η τελική επιλογή των άρθρων. Εδώ αναφέρουμε ότι τα σχετικά με το τεχνικό χρέος άρθρα ήταν συνολικά 305 όπως φαίνεται και στο γράφημα.

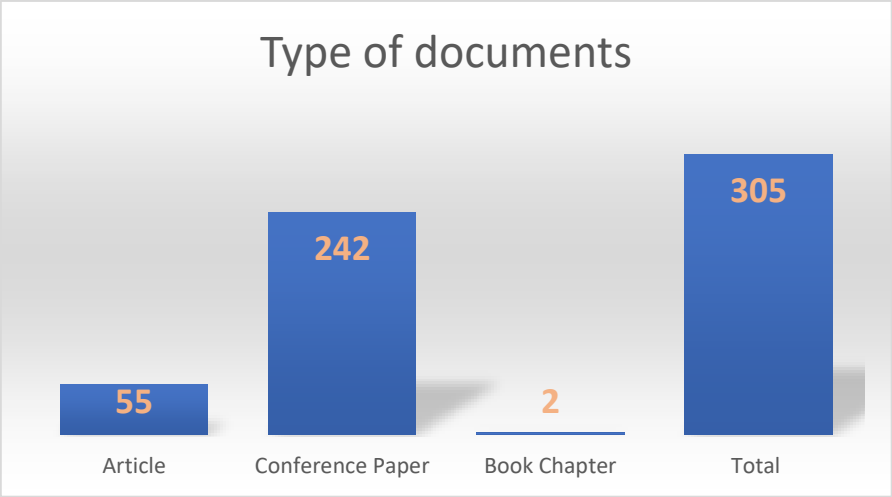


Γράφημα 6. Βήματα επιλογής άρθρων

Στη συνέχεια κατεβάσαμε αυτά τα 305 άρθρα – τα πλήρη κείμενα τα οποία και διαβάσαμε. Οι τύποι των κειμένων αυτών ήταν κυρίως Conference Paper και Article και φαίνονται στον ακόλουθο πίνακα και στο σχετικό γράφημα.

Document Type	Number of documents
Article	55
Conference Paper	242
Book Chapter	2
Total	305

Πίνακας 6. Τύποι των documents



Γράφημα 7. Type of Documents

Αφού λοιπόν μελετήσαμε το περιεχόμενο των 305 κειμένων, συλλέξαμε χρήσιμη γνώση σχετικά με το αντικείμενο αυτό και καταλήξαμε στις εξής 4 βασικές κατηγορίες :

1. Γενικά θέματα και θέματα διαχείρισης του τεχνικού χρέους,
2. Αιτίες και επιπτώσεις συσσώρευσης τεχνικού χρέους,
3. Τεχνικό χρέος και συντήρηση λογισμικού,
4. Βιβλιογραφική ανασκόπηση στο τεχνικό χρέος

Η πρώτη κατηγορία, η οποία ήταν και η πιο μεγάλη, αφορούσε γενικά θέματα (ορολογία, ορισμοί κλπ.) και θέματα διαχείρισης, μοντέλα και μεθοδολογίες υπολογισμού. Η δεύτερη κατηγορία αφορούσε κυρίως τους παράγοντες και τις επιπτώσεις συσσώρευσης τεχνικού χρέους ενώ η τρίτη κατηγορία ήταν σχετικά με το πως το τεχνικό χρέος συνδέεται και επηρεάζει τις εργασίες συντήρησης του λογισμικού. Μια ακόμα κατηγορία άρθρων που εντοπίσαμε, η τέταρτη κατηγορία ήταν η βιβλιογραφική ανασκόπηση γύρω από το συγκεκριμένο θέμα. Σίγουρα μέσα σε αυτές υπάρχουν και άλλες υποκατηγορίες για πχ τα διαφορετικά είδη του τεχνικού χρέους, για τα open source συστήματα, για συγκεκριμένες τεχνολογίες ανάπτυξης λογισμικού κλπ. Μια άλλη μεγάλη ομάδα/κατηγορία σχετίζεται με το θεωρητικό μέρος του θέματος, δηλαδή την ορολογία που υπάρχει στον τομέα αυτό, τις διαφορετικές προσεγγίσεις γύρω από τον ορισμό του και το πώς το αντιλαμβάνονται οι αναλυτές κλπ. Σε κάθε περίπτωση η κατηγοριοποίηση των δεδομένων καθορίζεται και

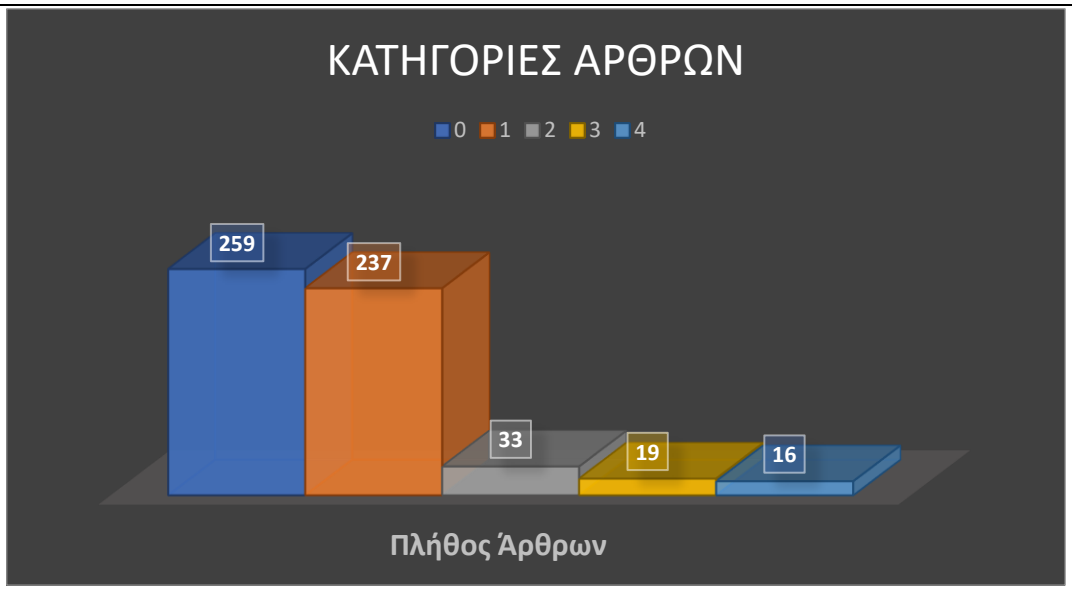
εξαρτάται από το ζητούμενο της έρευνας το οποίο θα καθορίσει και τα κριτήρια της κατηγοριοποίησης.

Τελικά από τη μελέτη των κειμένων προέκυψε ότι τα 33 από αυτά ανήκουν στη κατηγορία 2 ενώ τα υπόλοιπα 19 ανήκουν στην κατηγορία 3. Τα 16 αποτελούν SLR άρθρα (Systematic Literature Review) και τα υπόλοιπα 237 αναφέρονται γενικά στο τεχνικό χρέος και στη διαχείρισή του. Εδώ να κάνουμε μια αξιοσημείωτη παρατήρηση, η κατάταξη των άρθρων στη μια ή στην άλλη κατηγορία εξαρτάται και από τον σκοπό της έρευνας δηλαδή αν μας ενδιαφέρει π.χ. να εντοπίσουμε μόνο τα άρθρα που υπολογίζουν τη συντήρηση λογισμικού με οικονομικούς όρους και με μια συγκεκριμένη φόρμουλα τότε ενδεχομένως τα παρακάτω αποτελέσματα να είναι διαφορετικά και αυτό γιατί αφενός μεν δεν υπάρχει κοινή ορολογία στο συγκεκριμένο αντικείμενο αφετέρου κάποια μοντέλα προκύπτουν εμπειρικά. Διαβάζοντας τα πλήρη κείμενα διαπιστώσαμε ότι η επικρατέστερη ορολογία σχετικά με τη συντήρηση λογισμικού η οποία σχετίζεται με το τεχνικό χρέος περιέχει φράσεις όπως “interest amount”, “interest probability”, “maintenance effort” και “technical debt interest”. Ωστόσο υπάρχουν και άλλες λιγότερο συχνές φράσεις όπως “rework cost”, “wasted time” και “TD liability” και αυτές τις εντοπίσαμε σε μόλις 3 άρθρα. Αν συμπεριλάβουμε όλα τα άρθρα με αυτές τις διαφορετικές ορολογίες τότε καταλήγουμε σε μια κατάταξη όπως φαίνεται παρακάτω.

Κατηγορία	**Περιγραφή**	**Πλήθος άρθρων**
0	Μη σχετικά με το αντικείμενο έρευνας	259
1	Τεχνικό χρέος και θέματα διαχείρισης	237
2	Αιτίες και επιπτώσεις συσσώρευσης χρέους	33
3	Τεχνικό χρέος και συντήρηση λογισμικού	19
4	Βιβλιογραφική ανασκόπηση στο τεχνικό χρέος	16
Σύνολο		564

Πίνακας 7. Κατηγορίες Άρθρων

Στο ακόλουθο γράφημα έχουμε τη κατηγοριοποίηση του πίνακα 7.



Γράφημα 8. Κατηγοριοποίηση Άρθρων

Ανάλογα με τα ερευνητικά ενδιαφέροντα του εκάστοτε ερευνητή σχετικά με το αντικείμενο του τεχνικού χρέους θα μπορούσαμε να πούμε ότι τα άρθρα κατηγορίας 2 αφορούν, μεταξύ άλλων, και τη συντήρηση λογισμικού και παρέχουν χρήσιμη πληροφορία σε ό,τι αφορά τους παράγοντες συσσώρευσης τεχνικού χρέους ενώ κάποια από αυτά προτείνουν λύσεις για την μείωση του κόστους συντήρησης και αποτελούν σημαντικό κομμάτι της έρευνας για κάποιον που ενδιαφέρεται να γνωρίζει τις επικρατέστερες πρακτικές για την αντιμετώπιση του τεχνικού χρέους.

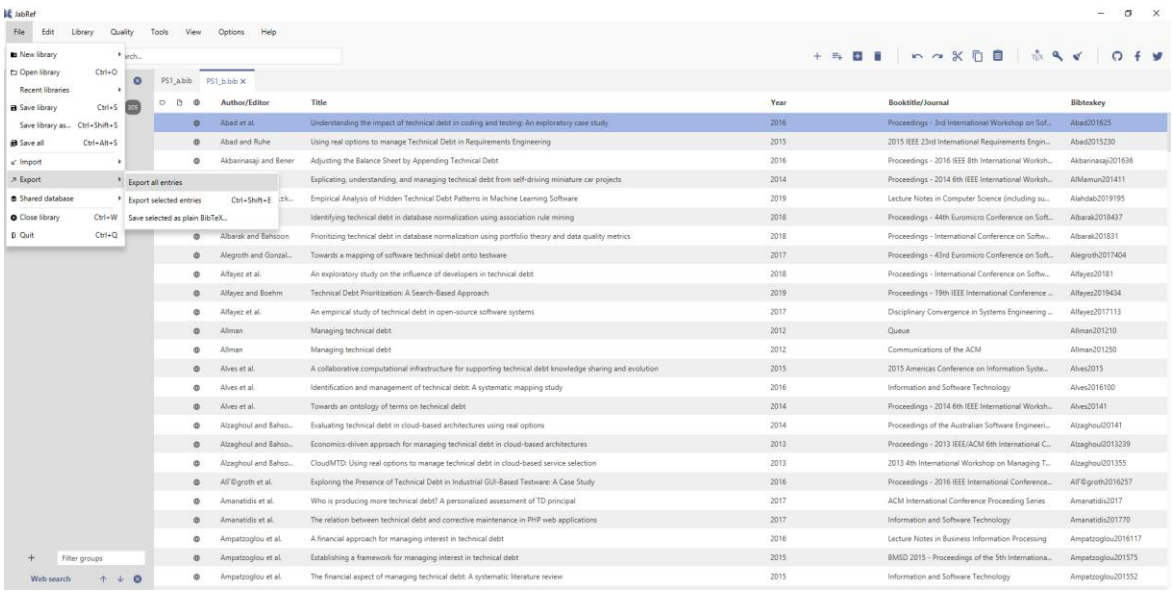
Τα άρθρα κατηγορίας 3 προτείνουν κάποιο μοντέλο για τον υπολογισμό της συντήρησης λογισμικού μέσα από τον υπολογισμό του τεχνικού χρέους. Συνεπώς συμβάλλουν στην ευρύτερη γνώση περί του συγκεκριμένου αντικειμένου καθόσον η αποτελεσματική διαχείριση του τεχνικού χρέους συμβάλει κατ' επέκταση και στην αντιμετώπιση της συντήρησης λογισμικού κατά τη διάρκεια του κύκλου ζωής ενός έργου λογισμικού.

Η προ επεξεργασία αυτή και ο εντοπισμός των κατηγοριών των άρθρων που μας ενδιαφέρουν γίνεται με σκοπό να συγκρίνουμε τα αποτελέσματα που θα προκύψουν από την εφαρμογή των τεχνικών εξόρυξης με αυτά στα οποία καταλήξαμε εμπειρικά και στη συνέχεια να επιβεβαιώσουμε ή/όχι την αποτελεσματικότητα των μεθόδων που θα εκτελέσουμε προκειμένου να εντοπίσουμε το σύνολο άρθρων που μας ενδιαφέρει. Άλλωστε όπως έχει προαναφερθεί στο κεφάλαιο 1 μια κατηγορία μάθησης είναι και η ενεργή μάθηση

στην οποία οι χρήστες ζητούνται να προσδώσουν ετικέτα κατηγορίας με σκοπό τη βελτίωση του μοντέλου (Han, Kamber & Pei, 2011).

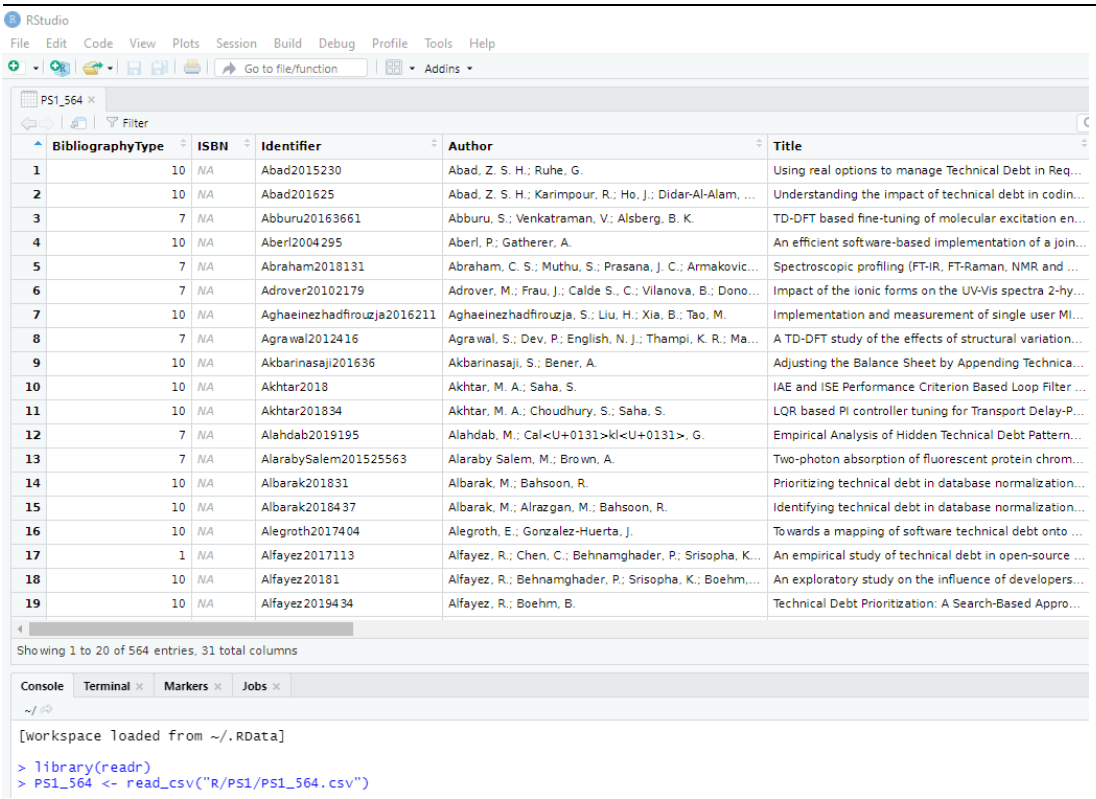
Βέβαια, ο ευρύτερος σκοπός της παρούσας εργασίας δεν περιορίζεται μόνο στη συντήρηση λογισμικού αλλά στοχεύει μεταξύ άλλων και στον εντοπισμό τομέων που επεκτείνεται το τεχνικό χρέος ή/και των θεμάτων που περιλαμβάνονται σε αυτό όπως π.χ. παράγοντες συσσώρευσης τεχνικού χρέους, άλλες υποκατηγορίες του, μεθοδολογίες και μοντέλα που χρησιμοποιούνται για τη διαχείριση του, τυχόν εργαλεία μέτρησής του, μεθοδολογίες και τεχνικές για την εκτίμηση των εργασιών συντήρησης λογισμικού και γενικά ότι αφορά το συγκεκριμένο θέμα.

Στη συνέχεια για την εφαρμογή των τεχνικών ανάλυσης κειμένου εξάγουμε μέσα από το εργαλείο JabRef το επιλεγμένο αρχείο PS1 σε μορφή csv.



Εικόνα 33. Εξαγωγή αρχείου

Το αρχείο αυτό θα το επεξεργαστούμε στο R studio οπότε το εισάγουμε με το όνομα PS1_564. Αυτό το αρχείο έχει ήδη σε ξεχωριστή στήλη την κατηγοριοποίηση που έχουμε ολοκληρώσει με εμπειρικό τρόπο διαβάζοντας περιλήψεις και πλήρη κείμενα όπως αναφέρθηκε παραπάνω. Οι τεχνικές που ακολουθούν στις επόμενες ενότητες (πλην της τελευταίας) δεν βασίζονται σε αυτή την κατηγοριοποίηση καθόσον ο αρχικός μας στόχος είναι η διερεύνηση των τομέων που εντοπίζουν οι τεχνικές αυτές προς αναζήτηση νέας γνώσης.



Εικόνα 34. Αρχείο προς επεξεργασία

Στη συνέχεια στις επόμενες ενότητες που ακολουθούν παρουσιάζουμε τις τεχνικές εξόρυξης και μηχανικής μάθησης που εφαρμόσαμε στα ακόλουθα σύνολα δεδομένων.

Dataset	Πλήθος άρθρων	Επεξεργασία υλικού
1ο	**564**	Προέκυψε μετά την αφαίρεση διπλότυπων, ξενόγλωσσων κλπ. (επεξεργασία στο JabRef)
2ο	**305**	Μετά την ανάγνωση τίτλων και περιλήψεων Μετά την εφαρμογή του topic modeling στο 1^{ο} dataset

Πίνακας 8. Dataset προς ανάλυση

5.4 Εξερεύνηση των Τίτλων

Πριν αρχίσουμε την εφαρμογή αλγορίθμων θα επιχειρήσουμε να εξερευνήσουμε το περιεχόμενο των τίτλων μέσω οπτικοποίησης των λέξεων. Αυτό θα μας βοηθήσει να

εντοπίσουμε πολύ εύκολα τα επικρατέστερα θέματα με τα οποία ασχολείται το τεχνικό χρέος. Πέρα από αυτό όμως θα μας βοηθήσει να εντοπίσουμε τις επικρατέστερες λέξεις ή/ και τα ακρωνύμια που εμφανίζονται στους τίτλους (Gulo et. al., 2015b).

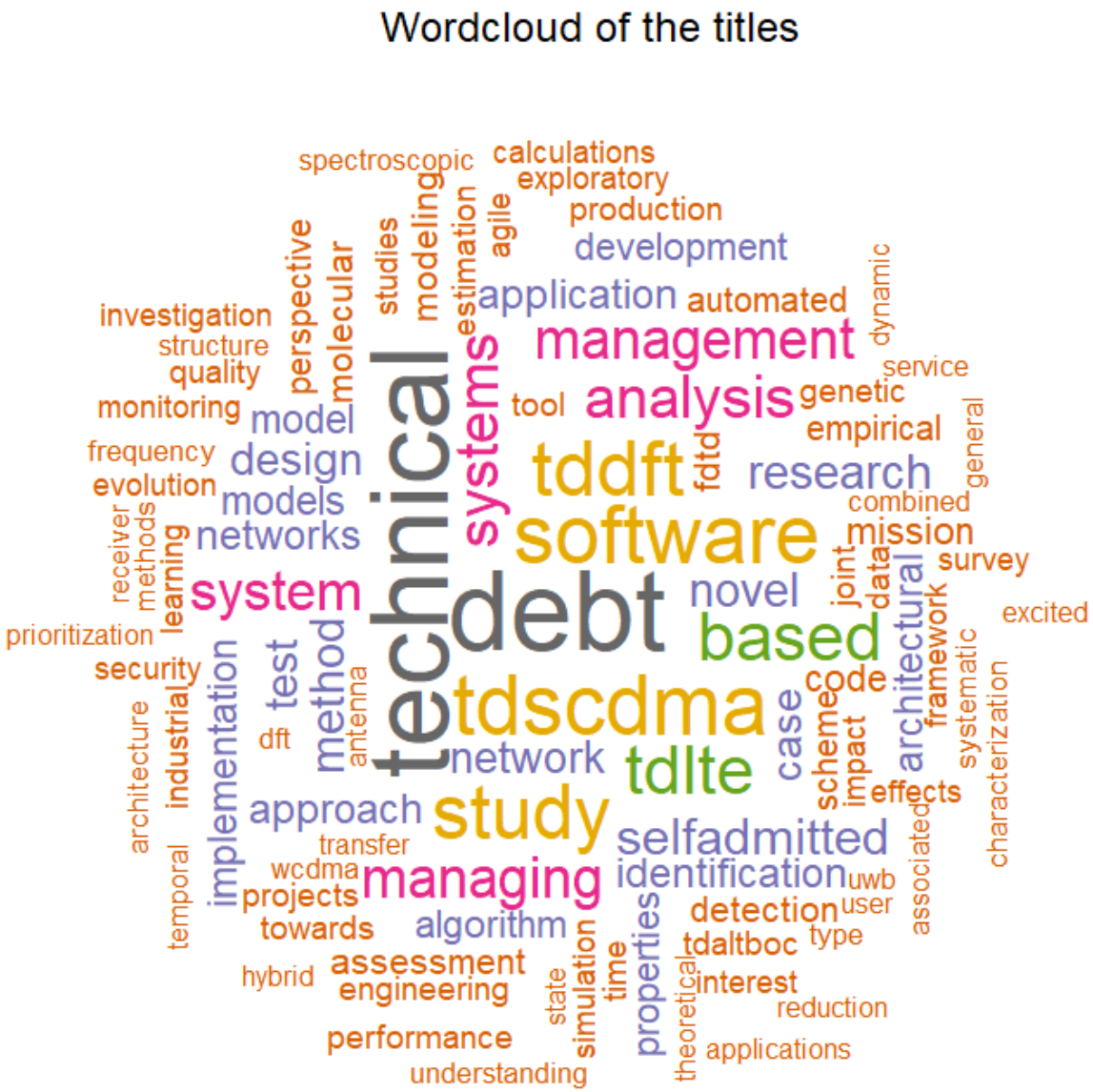
Με τη βοήθεια του παρακάτω κώδικα εισάγουμε το αρχείο μας στο R-studio και στη συνέχεια χρησιμοποιούμε τους τίτλους με τους οποίους φτιάχνουμε έναν πίνακα document term matrix. Αρχικά διαβάζουμε το αρχείο μας (γραμμή κώδικα 7) και δημιουργούμε ένα πλαίσιο δεδομένων – data frame (γραμμή 8). Αποθηκεύουμε στο αντικείμενο paper.title τους τίτλους (γραμμή 10). Έπειτα φροντίζουμε να καθαρίσουμε τους τίτλους από αριθμούς, σύμβολα, σημεία στίξης, χαρακτήρες, κενά κλπ. (γραμμές κώδικα 12-24). Αφαιρούμε κάποιες λέξεις όπως ρήματα, προθέσεις, συνδέσμους κ.λπ. οι οποίοι δεν προσφέρουν κάποια πληροφορία σχετικά με το σημασιολογικό περιεχόμενο του τίτλου (γραμμή 21) και εφαρμόζουμε τη μέθοδο tfidf (γραμμή 25) για τη δημιουργία του document term matrix.

```
1  # WORD CLOUD OF THE TITLES
2  library(tm)
3  library(NLP)
4  library(stringi)
5  setwd("~/R/PS1")
6  #read data
7  r<-read.csv("/Users/User/Documents/R/PS1/PS1_564.csv")
8  rdt<-as.data.frame(r)
9  rdt2<-subset(rdt, select = - c(BibliographyType,ISBN,Author, Journal,Volume, Number, Month, Pages,Note, URL,Address, E
10 paper.title <- VCorpus(VectorSource(rdt2$Title))
11 inspect(paper.title)
12 toSpace <- content_transformer(function(x,pattern) gsub(pattern, " ", x))
13 tcorpu <- tm_map(paper.title, toSpace, "/|@|\\|")
14 tcorpu<-tm_map(tcorpu,content_transformer(tolower))
15 removeNumPunct <- function(x) gsub("[^[:alpha:][:space:]]*", "", x) #remove anything other than English letters or spa
16 tcorpu<-tm_map(tcorpu, content_transformer(removeNumPunct))
17 tcorpu<-tm_map(tcorpu,removeWords,stopwords("english"))
18 removeUnicode <- function(x) stri_replace_all_regex(x,"[^\x20-\x7E]","")
19 tcorpu <- tm_map(tcorpu, content_transformer(removeUnicode))
20 #remove extra words
21 tcorpu<-tm_map(tcorpu,removeWords, c("use","can","get","could","have", "will","using", "would" ,"also","say","one","wa
22 tcorpu<-tm_map(tcorpu,removePunctuation, UCP=TRUE)
23 tcorpu<-tm_map(tcorpu,removeNumbers)
24 tcorpu<-tm_map(tcorpu,stripWhitespace) #remove extra whitespace
25 tdtm<-DocumentTermMatrix(tcorpu, control = list(weighting=weightTfIdf,minWordLength=4, bounds = list(global = c(3, Inf
26 inspect(tdtm)
```

Στη συνέχεια δημιουργούμε το νέφος των λέξεων με τη μεγαλύτερη συχνότητα εμφάνισης (γραμμές 28-35).

```
28 #create word cloud
29 library(wordcloud)
30 library(RColorBrewer)
31 m<-as.matrix(tdtm)
32 v<-sort(colSums(m), decreasing=TRUE)
33 d<-data.frame(word=names(v), freq=v)
34 wordcloud(d$word, d$freq,random.order=FALSE, rot.per=0.3,scale=c(4,.5),max.words=101,colors=brewer.pal(8,"Dark2"))
35 title(main="Wordcloud of the titles", font.main=1, cex.main=1.5)
```

Παρακάτω φαίνεται το νέφος των λέξεων των τίτλων που προκύπτει με τον ανωτέρω κώδικα.



Εικόνα 35. Word Cloud Τίτλων

Ο πιο μεγάλες στο νέφος λέξεις είναι και οι πιο συχνά εμφανιζόμενες στους τίτλους. Στην εικόνα 35 παρατηρούμε ότι οι πιο μεγάλες λέξεις είναι οι technical, debt, αναμενόμενο αποτέλεσμα αφού αυτές ανήκουν στο query που χρησιμοποιήσαμε.

Οι πιο σημαντικές λέξεις σχετικά με το τεχνικό χρέος είναι οι:

- management, managing: αναφέρονται βασικά στο τρόπο διαχείρισης του τεχνικού χρέους
- software, development: αναφέρονται στο λογισμικό και την ανάπτυξη λογισμικού

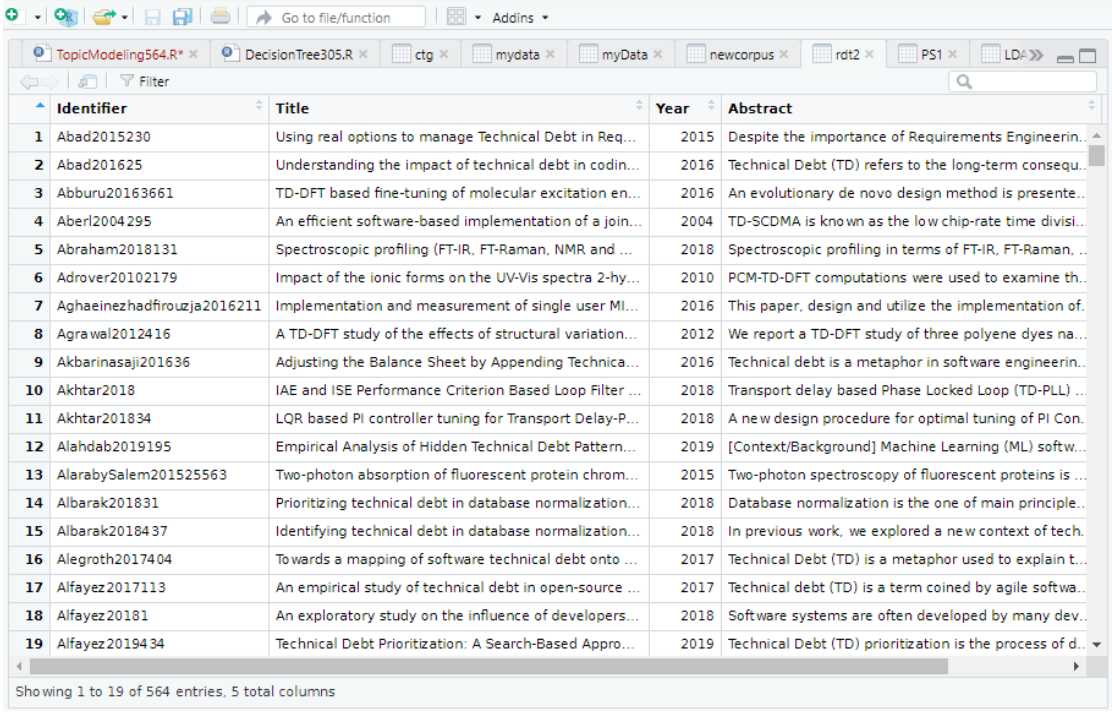
- selfadmitted: αναφέρονται στο εκούσιο τεχνικό χρέος
- design: αφορούν ενδεχομένως το design debt
- interest: αναφέρεται στη συντήρηση λογισμικού (technical debt interest)
- architectural, architecture: αφορά το αρχιτεκτονικό χρέος.

Επίσης παρατηρούμε ότι υπάρχουν λέξεις ή ακρωνύμια μη σχετικά με το τεχνικό χρέος όπως tdlte, tdscdma, tdaltboc, spectroscopic κα. Συμπεραίνουμε λοιπόν ότι το dataset, που έχουμε, περιέχει και μη σχετικά άρθρα κατά συνέπεια θα πρέπει με την εφαρμογή του κατάλληλου αλγορίθμου εξόρυξης να αφαιρέσουμε αυτά τα μη σχετικά με το τεχνικό χρέος άρθρα πριν εφαρμόσουμε περαιτέρω τεχνικές εξόρυξης.

5.5 Topic Modeling στο 1ο dataset

Το σενάριο αυτό θα εφαρμοστεί στο dataset που έχουμε εισάγει στο R - studio.

Εισάγουμε στο R studio το csv αρχείο το οποίο προέκυψε μετά από αναζήτηση με τον όρο «technical debt OR TD» στο Scopus. Αποτελείται από 564 άρθρα. Εδώ να πούμε ότι με το εργαλείο Jabref έχουμε κάνει μια προ επεξεργασία των δεδομένων για να επιβεβαιώσουμε τα αποτελέσματα στα οποία καταλήγουμε και παρουσιάζονται παρακάτω. Με τον όρο προ επεξεργασία αναφερόμαστε σε όλη την προεργασία που κάναμε δηλαδή το να διαβάσουμε τίτλους, περιλήψεις και πλήρη κείμενα, όπου χρειάστηκε, και να προσδώσουμε σε κάθε ένα άρθρο μια ετικέτα κατηγορίας (εμπειρικά) με βάση το περιεχόμενό τους.



	Identifier	Title	Year	Abstract
1	Abad2015230	Using real options to manage Technical Debt in Req...	2015	Despite the importance of Requirements Engineerin..
2	Abad201625	Understanding the impact of technical debt in codin...	2016	Technical Debt (TD) refers to the long-term consequ..
3	Abburu20163661	TD-DFT based fine-tuning of molecular excitation en...	2016	An evolutionary de novo design method is presente..
4	Aberl2004295	An efficient software-based implementation of a join...	2004	TD-SCDMA is known as the low chip-rate time divisi..
5	Abraham2018131	Spectroscopic profiling (FT-IR, FT-Raman, NMR and ...	2018	Spectroscopic profiling in terms of FT-IR, FT-Raman, ..
6	Adrover20102179	Impact of the ionic forms on the UV-Vis spectra 2-hy...	2010	PCM-TD-DFT computations were used to examine th..
7	Aghaeinezhadfirouzja2016211	Implementation and measurement of single user MI...	2016	This paper, design and utilize the implementation of.
8	Agrawal2012416	A TD-DFT study of the effects of structural variation...	2012	We report a TD-DFT study of three polyene dyes na...
9	Akbarinasaji201636	Adjusting the Balance Sheet by Appending Technica...	2016	Technical debt is a metaphor in software engineerin..
10	Akhtar2018	IAE and ISE Performance Criterion Based Loop Filter ...	2018	Transport delay based Phase Locked Loop (TD-PLL) ...
11	Akhtar201834	LQR based PI controller tuning for Transport Delay-P...	2018	A new design procedure for optimal tuning of PI Con.
12	Alahdab2019195	Empirical Analysis of Hidden Technical Debt Pattern...	2019	[Context/Background] Machine Learning (ML) softw...
13	AlarabySalem201525563	Two-photon absorption of fluorescent protein chrom...	2015	Two-photon spectroscopy of fluorescent proteins is ...
14	Albarak201831	Prioritizing technical debt in database normalization...	2018	Database normalization is the one of main principle..
15	Albarak2018437	Identifying technical debt in database normalization...	2018	In previous work, we explored a new context of tech.
16	Alegroth2017404	Towards a mapping of software technical debt onto ...	2017	Technical Debt (TD) is a metaphor used to explain t...
17	Alfayez2017113	An empirical study of technical debt in open-source ...	2017	Technical debt (TD) is a term coined by agile softwa..
18	Alfayez20181	An exploratory study on the influence of developers...	2018	Software systems are often developed by many dev..
19	Alfayez2019434	Technical Debt Prioritization: A Search-Based Appro...	2019	Technical Debt (TD) prioritization is the process of d..

Showing 1 to 19 of 564 entries, 5 total columns

Εικόνα 36. Αρχείο 1ο dataset

Το βασικό πρόβλημα που έχουμε να αντιμετωπίσουμε είναι η εφαρμογή της κατάλληλης τεχνικής εξόρυξης η οποία θα καταφέρει να εντοπίσει με ακρίβεια το υποσύνολο δεδομένων που μας ενδιαφέρει για περαιτέρω ανάλυση. Δηλαδή όπως έχουμε αναφέρει στην υποενότητα 5.3 το dataset που κατεβάσαμε από τη βάση του Scopus είχε και πολλά μη σχετικά με το τεχνικό χρέος άρθρα τα οποία δεν μας χρειάζονται.

Σύμφωνα με τους Anandarajan, Hill and Nolan (2019) όταν θέλουμε να βρούμε το περιεχόμενο ενός συνόλου δεδομένων τότε εφαρμόζουμε τη μέθοδο topic modeling. Εμείς αυτό που προσπαθούμε να πετύχουμε είναι να χωρίσουμε το σύνολο σε ξεχωριστά topic τα οποία θα περιέχουν μόνο τα σχετικά με το τεχνικό χρέος άρθρα ή τα μη σχετικά, δηλαδή θέλουμε να πετύχουμε έναν καθαρό διαχωρισμό των κειμένων μας. Για να το πετύχουμε κάναμε πολλές δοκιμές και ξεκινήσαμε με sparsity 0,99 προκειμένου να μην αφαιρέσουμε πολλές λέξεις από τον αρχικό πίνακα Document Term Matrix. Το αποτέλεσμα μας έφερε σε 3 topics 313 άρθρα εκ των οποίων μόνο 8 ήταν μη σχετικά. Στη συνέχεια μειώσαμε το sparsity σε 0,95 μειώνοντας έτσι τις στήλες του πίνακα document term matrix και το αποτέλεσμα μας έφερε 3 topics με 307 άρθρα εκ των οποίων μόνο 2 ήταν μη σχετικά, αυτά ήταν τα άρθρα με αύξοντα αριθμό 162 και 163. Στη συνέχεια ελέγξαμε τις stopwords και χρειάστηκε να προσθέσουμε μερικές ακόμα (περίπου 12 λέξεις) στον κώδικα – γραμμή 38

οι οποίες μας είχαν αρχικά διαφύγει (πχ. addition, although, several κλπ). Μετά την επανεκτέλεση του κώδικα σχηματίστηκαν 6 topics στα 3 από τα οποία είχαμε ακριβώς τα 305 άρθρα τα οποία έπρεπε να βρούμε.

Παρακάτω παρουσιάζουμε αναλυτικά τον κώδικα και τα βήματα που ακολουθήσαμε :

Αρχικά εκτελούμε τον ακόλουθο κώδικα προκειμένου να παραχθεί ο πίνακας document term matrix τον οποίο θα χρειαστούμε για να εφαρμόσουμε τα μοντέλα για την κατηγοριοποίηση των δεδομένων μας. Διαβάζουμε το csv αρχείο (γραμμή 21), δημιουργούμε πλαίσιο δεδομένων (γραμμή 22), αφαιρούμε κάποιες στήλες (γραμμή 23) και τέλος κρατάμε στο αντικείμενο paper.abstract (γραμμή 26) τις περιλήψεις των κειμένων. Στη συνέχεια αφαιρούμε από αυτές αριθμούς, σύμβολα, σημεία στίξης, χαρακτήρες, κενά, αγγλικές stopwords και επιπλέον stopwords που έχουμε ορίσει εμείς στον κώδικα (γραμμές 29-43). Δημιουργούμε τον πίνακα document term matrix (γραμμή 44).

```
1  # 1.TOPIC MODELING 564 CSV IN ABSTRACTS
2  # 2. CLUSTERING
3  # 3. DECISION TREE
4
5  library(tm) #required for text mining
6  library(pdftools)
7  library(NLP)
8  library(topicmodels)
9  library(RColorBrewer)
10 library(lda) # lattent dirichlet allocation
11 library(ldatuning)#to find number of topics
12 library(wordcloud)# to make a wordcloud
13 library(SnowballC) #for stemming
14 library(quanteda) #required for lattent dirichlet allocation function
15 library(ggplot2)
16 library(stringi)
17
18 setwd("~/R/564")
19
20 #read data
21 r<-read.csv("/Users/User/Documents/R/564/PS1_564.csv")
22 rdt<-as.data.frame(r)
23 rdt2<-subset(rdt, select = - c(BibliographyType,ISBN,Author, Journal,Volume, Number, Month, Pages,Note, URL,Addres
24 names(rdt2)[4]="Abstract"
25 View(rdt2)
26 paper.abstract <- VCorpus(VectorSource(rdt2$Abstract))
27 inspect(paper.abstract)
28
29 toSpace <- content_transformer(function(x,pattern) gsub(pattern, " ", x))
30 corpu <- tm_map(paper.abstract, toSpace, "/|@|\\|")
31 corpu<-tm_map(corpu,content_transformer(tolower))
32 removeNumPunct <- function(x) gsub("[^[:alpha:][:space:]]*", "", x) #remove anything other than English letters or
33 corpu<-tm_map(corpu, content_transformer(removeNumPunct))
34 corpu<-tm_map(corpu,removeWords,stopwords("english"))
35 removeUnicode <- function(x) stri_replace_all_regex(x,"[^\x20-\x7E]","")
36 corpu <- tm_map(corpu, content_transformer(removeUnicode))
37 #remove extra words
38 mystopwords<-c("show","finally","although","addition","four","several","better","therefore","significant","springe
39 mystopwords<-sort(mystopwords)
40 corpu<-tm_map(corpu,removeWords, mystopwords)
41 corpu<-tm_map(corpu,removePunctuation, UCP=TRUE)
42 corpu<-tm_map(corpu,removeNumbers)
43 corpu<-tm_map(corpu,stripWhitespace) #remove extra whitespace
44 dtm<-DocumentTermMatrix(corpu, control = list(weighting=weightTf,stopwords=T,minWordLength=c(4,15), bounds = list(
45
```

Εικόνα 37. Κώδικας R - Topic Modeling 1ο dataset

Ο πίνακας φαίνεται ως ακολούθως:

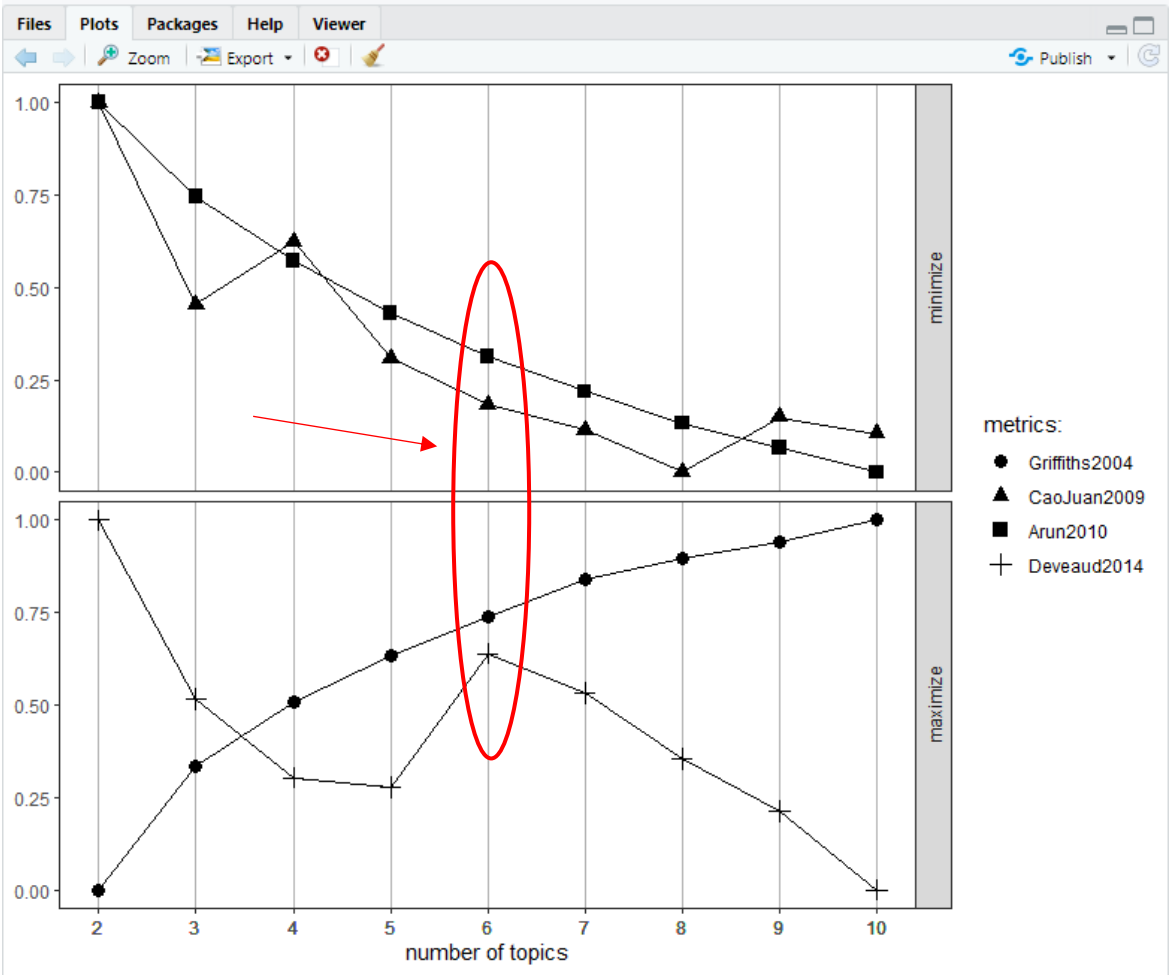
```
<<DocumentTermMatrix (documents: 564, terms: 184)>>
Non-/sparse entries: 8763/95013
Sparsity           : 92%
Maximal term length: 15
Weighting          : term frequency (tf)
Sample             :
     Terms
Docs  analysis code data development management method model quality results time
  145        4    5    0           0          0      1     0       0       3    0
  179        5   10    0           1          0      0     1       5       0    0
  274        2    6    0           0          0      0     0       0       2    6
  31         0    0    1           0          1      1     0       0       2    0
  338        0    0    0           3          0      1     0       0       2    1
  350        3    2    1           3          0      0     0       1       2    1
  365        0    2    0           4          6      2     0       2       0    1
  436        0    0    1           1          0      0     0       0       0    2
  527        0    6    0           1          0      0     5       1       1    0
  549        3    7    0           1          0      0     0       0       1    1
> |
```

Πίνακας 9. Document Term Matrix – 1ο dataset

Στη συνέχεια θα εφαρμόσουμε τον αλγόριθμο του topic modeling. Αρχικά θα επιλέξουμε τον ιδανικό αριθμό των topics με βάση τον παρακάτω κώδικα. Στον κώδικα ορίζουμε 2-10 topics και εισάγουμε τις μετρικές μας.

```
#find optimum number of topics

#Arun2020 maximize, CaoJuan minimize, Griffiths minimize
optimal.topics <- FindTopicsNumber(
  dtm ,
  topics = c(2:10),
  metrics = c("Griffiths2004", "CaoJuan2009", "Arun2010","Deveaud2014"),
  method = "Gibbs",
  control = list(seed = 12345),
  mc.cores = 4L,
  verbose = TRUE
)
FindTopicsNumber_plot(optimal.topics)
```

Εικόνα 38. Μετρικές του Topic Modeling - 1ο dataset



Γράφημα 9. Μετρικές του Topic Modeling – 1ο dataset

Με βάση τις μετρικές CaoJuan2009, Arun2010, Griffiths2004 και Deveaud2014 στο ανωτέρω γράφημα επιλέγουμε 6 topics.

Στη συνέχεια εκτελούμε τον ακόλουθο κώδικα

```
set.seed(222)
m=LDA(dtm, method="Gibbs", k=6, control=list(alpha=0.1))
#for a specific topic we can find topwords
topic = 6
words = posterior(m)$terms[topic, ]
topwords = head(sort(words, decreasing = T), n=50)
head(topwords)
```

Εικόνα 39. Δημιουργία των topics - 1ο dataset

Με τη συνάρτηση LDA του topic modeling ορίζουμε 6 topics και χρησιμοποιούμε τη στατιστική συνάρτηση α η οποία καθορίζει το πλήθος των topics που ανατίθενται σε κάθε

κείμενο, ήτοι μικρή τιμή του α σημαίνει ότι σε κάθε κείμενο ανατίθενται λιγότερα topics. Με τη συνάρτηση posterior αναθέτουμε στα κείμενα topics. Στη συνέχεια με την topwords εμφανίζουμε τις πιο δημοφιλείς λέξεις από όλα τα topics.

Αυτά είναι τα αποτελέσματα του κώδικα από την εκτέλεση του αλγορίθμου:

Βλέπουμε τις επικρατέστερες λέξεις από όλα τα topics.

```
> FindTopicsNumber_plot(optimal.topics)
> set.seed(222)
> m=LDA(dtm, method="Gibbs", k=6, control=list(alpha=0.1))
> #for a specific topic we can find topwords
> topic = 6
> words = posterior(m)$terms[topic, ]
> topwords = head(sort(words, decreasing = T), n=50)
> head(topwords)
   results      tddft     theory    density functional   analysis
```

Παρακάτω έχουμε τις 15 πρώτες λέξεις ανά topic

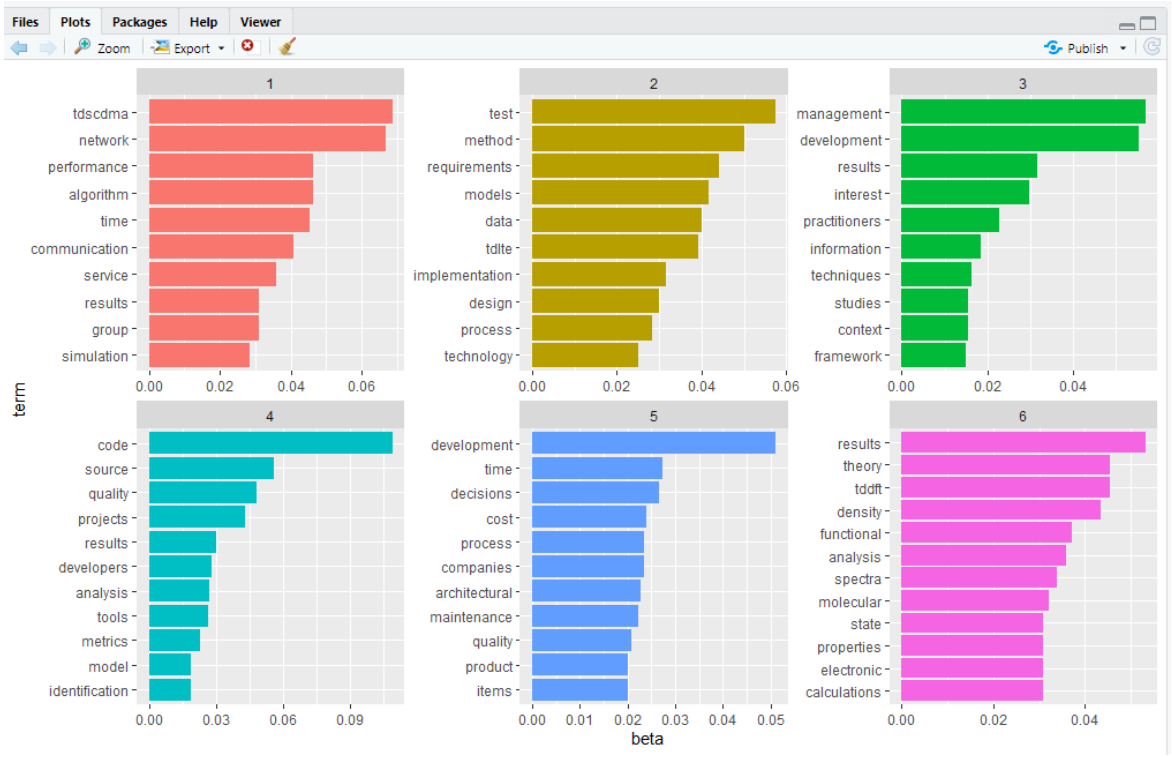
```
> terms(m,15)
      Topic 1          Topic 2          Topic 3          Topic 4          Topic 5          Topic 6
 [1,] "tdscdma"        "test"           "management"     "code"           "development"    "results"
 [2,] "network"        "method"         "development"    "source"         "time"           "tddft"
 [3,] "algorithm"      "requirements"   "results"        "quality"        "decisions"      "theory"
 [4,] "performance"    "models"         "interest"       "projects"       "cost"           "density"
 [5,] "time"           "data"           "practitioners"  "results"        "companies"      "functional"
 [6,] "communication"  "tdlte"          "information"    "developers"     "process"        "analysis"
 [7,] "service"        "implementation" "techniques"     "analysis"       "architectural"  "spectra"
 [8,] "group"          "design"         "context"        "tools"          "maintenance"    "molecular"
 [9,] "results"        "process"        "studies"        "metrics"        "quality"        "calculations"
[10,] "simulation"     "technology"     "framework"      "identification" "items"          "electronic"
[11,] "frequency"      "development"    "method"         "model"          "product"        "properties"
[12,] "solution"       "order"          "tools"          "items"          "results"        "state"
[13,] "presented"      "control"        "practices"      "open"           "impact"         "method"
[14,] "users"          "important"      "projects"       "time"           "model"          "experimental"
[15,] "compared"       "time"           "maintenance"    "evolution"      "costs"          "complex"
```

Είναι εμφανές ότι τα 2 πρώτα topics καθώς και το τελευταίο, δηλαδή το 6ο δεν αφορούν το τεχνικό χρέος.

Ο κατωτέρω κώδικας μας δίνει τα γραφήματα των topics:

```
library(tidytext)

ap_topics <- tidy(m, matrix = "beta")
ap_topics

library(ggplot2)
library(dplyr)
#plots
ap_top_terms <- ap_topics %>%
  group_by(topic) %>%
  top_n(10, beta) %>%
  ungroup() %>%
  arrange(topic, -beta)

ap_top_terms %>%
  mutate(term = reorder_within(term, beta, topic)) %>%
  ggplot(aes(beta, term, fill = factor(topic))) +
  geom_col(show.legend = FALSE) +
  facet_wrap(~ topic, scales = "free") +
  scale_y_reordered()
```

Το παρακάτω γράφημα μας δείχνει τις πιο συχνές λέξεις ανά topic και ουσιαστικά και μας δίνει χρήσιμη πληροφορία για το περιεχόμενο του κάθε topic .



Γράφημα 10. 6 - Topic Model - 1ο dataset

Από το ανωτέρω γράφημα διαπιστώνουμε ότι τα topic 1,2,6 αφορούν άλλο επιστημονικό πεδίο και όχι το τεχνικό χρέος της ανάπτυξης λογισμικού. Αυτό συνέβη διότι ο όρος

αναζήτησης TD στο "technical debt OR TD" έφερε και άρθρα που είχαν στον τίτλο το ακρωνύμιο TD όπως είναι τα ακρωνύμια tddft, tdscdma τα οποία περιέχουν τον όρο TD και εμφανίζονται στα topic 1, 2, 6. Αυτό το έχουμε ήδη επιβεβαιώσει τόσο με το word cloud όσο με το Jabref διαβάζοντας τίτλους και περιλήψεις.

Από τις εμφανιζόμενες λέξεις μπορούμε να περιγράψουμε το περιεχόμενο του κάθε topic το οποίο παρουσιάζεται στον παρακάτω πίνακα.

Topic	Περιγραφή
1	Αφορά μη σχετικό με το τεχνικό χρέος τομέα (αφορά τις τηλεπικοινωνίες)
2	Αφορά μη σχετικό με το τεχνικό χρέος τομέα (αφορά τις τηλεπικοινωνίες)
3	Θέματα διαχείρισης του τεχνικού χρέους και της συντήρησης λογισμικού
4	Θέματα ποιότητας και μετρικών του πηγαίου κώδικα που συσχετίζονται με το τεχνικό χρέος
5	Θέματα κόστους και χρόνου που αφορούν την ανάπτυξη λογισμικού και το αρχιτεκτονικό χρέος
6	Αφορά μη σχετικό με το τεχνικό χρέος τομέα (αφορά κλάδο της χημείας)

Πίνακας 10. Περιγραφή των 6 Topics 1ου dataset

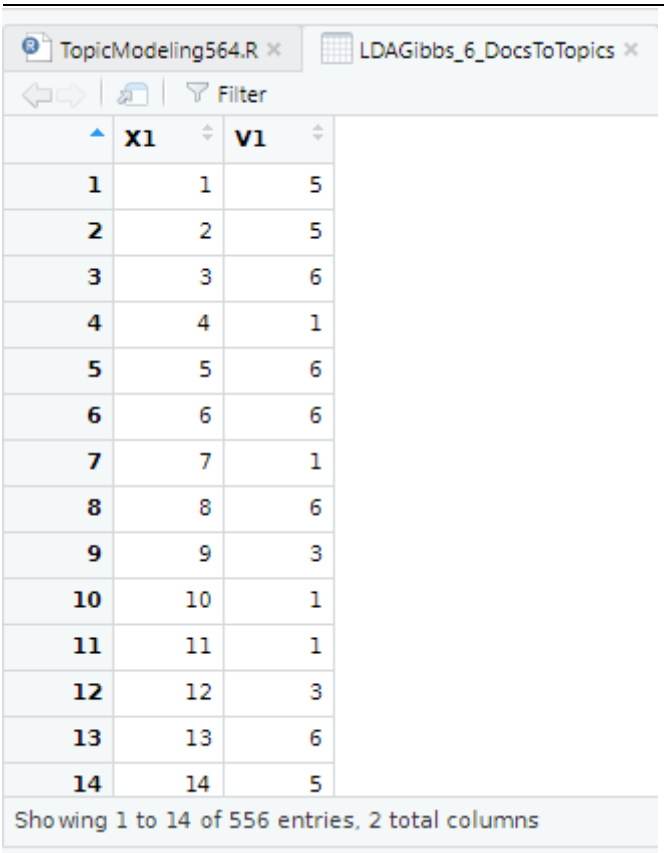
Σε αυτό το σημείο μπορούμε να συγκρίνουμε τα δικά μας αποτελέσματα με αυτά της παραπάνω μεθόδου. Ο αλγόριθμος εντοπίζει θεματικά τα κείμενα και τα χωρίζει σε κατηγορίες με βάση την εμφάνιση των λέξεων σε αυτά. Εμείς στη δική μας εμπειρική κατηγοριοποίηση θεωρήσαμε τα μη σχετικά άρθρα ως μια κατηγορία όλα μαζί, καθόσον δεν αφορούν το αντικείμενό μας, ενώ ο αλγόριθμος, όπως ήταν αναμενόμενο, τα εντάσσει σε διαφορετικές κατηγορίες ανάλογα με το περιεχόμενό τους. Τα σχετικά με το τεχνικό χρέος κείμενα τα χωρίζει σε 3 κατηγορίες, ενώ εμείς σε 4 διότι έχουμε θεωρήσει ως ξεχωριστή κλάση τα άρθρα της βιβλιογραφικής ανασκόπησης. Από την άλλη, το topic modeling δεν φαίνεται να μπορεί σε καμία περίπτωση να κάνει έναν τέτοιο διαχωρισμό. Για τις λοιπές κατηγορίες παρατηρούμε ότι ο αλγόριθμος εντοπίζει τα άρθρα τα σχετικά με τη διαχείριση του τεχνικού χρέους όπως και εμείς με τη διαφορά ότι περιέχει και άρθρα με τη συντήρηση λογισμικού. Το topic 4 που αφορά τα θέματα ποιότητας είναι μια ομάδα κειμένων εν μέρει σχετική η οποία συνδέεται και με τα αίτια συσσώρευσης τεχνικού χρέους.

Το topic 5 περιέχει άρθρα σχετικά με τον υπολογισμό (σε κόστος ή χρόνο) του κεφαλαίου και του τόκου του τεχνικού χρέους καθώς και ότι αφορά το αρχιτεκτονικό χρέος. Η ομαδοποίηση αυτή χωρίζει τα άρθρα σε 3 topics το 3, το 4 και το 5 με 101, 97, 107 άρθρα αντίστοιχα, ως εκ τούτου, απέχει από τη δική μας ομαδοποίηση. Εν τούτοις, καταφέρνει να διαχωρίσει τα σχετικά από τα μη σχετικά με απόλυτη επιτυχία και αυτός ήταν ο στόχος σε αυτή τη φάση.

Στη συνέχεια κρατάμε τα αποτελέσματα του κώδικα και τα αποθηκεύουμε σε ένα αρχείο τα άρθρα με τα topics, στα οποία ανήκουν, και σε ένα δεύτερο αρχείο αποθηκεύουμε τις πιθανότητες των κειμένων ανά topic με τη βοήθεια του παρακάτω κώδικα.

```
#topic propabilities
topicProbabilities <- as.data.frame(m@gamma)
#probabilities for the articles to the topics
write.csv(topicProbabilities,file=paste("LDAGibbs", 6,"TopicProbabilities.csv"))
#the top 6 terms for every topic
ldaOut.terms <- as.matrix(terms(m,10))

#write out results
#docs to topics
ldaOut.topics <- as.matrix(topics(m))
write.csv(ldaOut.topics,file=paste("LDAGibbs",6,"DocsToTopics.csv"))
#----------------------------------------------------------------------------#
```

Τα αρχεία φαίνονται παρακάτω. Στον ακόλουθο πίνακα βλέπουμε σε ποιο topic ανήκει το κάθε άρθρο.



TopicModeling564.R | LDAGibbs_6_DocsToTopics

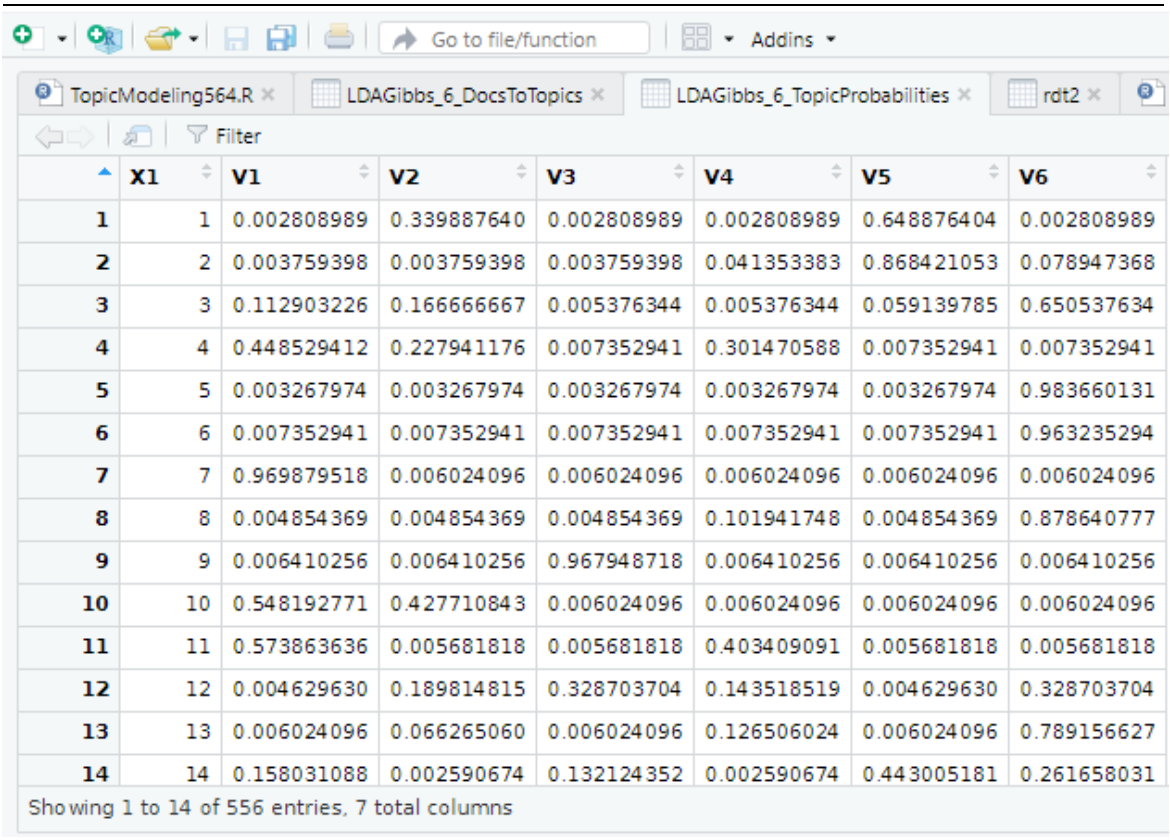
Filter

	X1	V1
1	1	5
2	2	5
3	3	6
4	4	1
5	5	6
6	6	6
7	7	1
8	8	6
9	9	3
10	10	1
11	11	1
12	12	3
13	13	6
14	14	5

Showing 1 to 14 of 556 entries, 2 total columns

Πίνακας 11. Documents and Topics – 1ο dataset

Στον επόμενο πίνακα βλέπουμε την πιθανότητα που έχει το κάθε άρθρο σε καθένα topic. Τελικά το κάθε άρθρο εντάσσεται σε εκείνο το topic με τη μεγαλύτερη πιθανότητα πχ. το άρθρο 1 ανήκει στο topic 5 κοκ.



Go to file/function | Addins

TopicModeling564.R | LDAGibbs_6_DocsToTopics | LDAGibbs_6_TopicProbabilities | rdt2

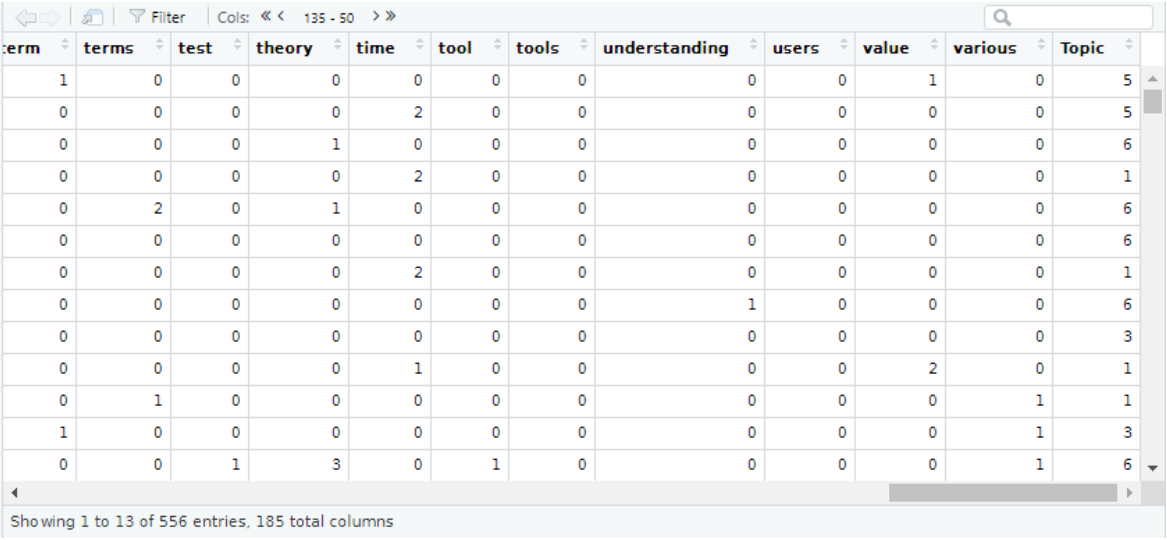
Filter

	X1	V1	V2	V3	V4	V5	V6
1	1	0.002808989	0.339887640	0.002808989	0.002808989	0.648876404	0.002808989
2	2	0.003759398	0.003759398	0.003759398	0.041353383	0.868421053	0.078947368
3	3	0.112903226	0.166666667	0.005376344	0.005376344	0.059139785	0.650537634
4	4	0.448529412	0.227941176	0.007352941	0.301470588	0.007352941	0.007352941
5	5	0.003267974	0.003267974	0.003267974	0.003267974	0.003267974	0.983660131
6	6	0.007352941	0.007352941	0.007352941	0.007352941	0.007352941	0.963235294
7	7	0.969879518	0.006024096	0.006024096	0.006024096	0.006024096	0.006024096
8	8	0.004854369	0.004854369	0.004854369	0.101941748	0.004854369	0.878640777
9	9	0.006410256	0.006410256	0.967948718	0.006410256	0.006410256	0.006410256
10	10	0.548192771	0.427710843	0.006024096	0.006024096	0.006024096	0.006024096
11	11	0.573863636	0.005681818	0.005681818	0.403409091	0.005681818	0.005681818
12	12	0.004629630	0.189814815	0.328703704	0.143518519	0.004629630	0.328703704
13	13	0.006024096	0.066265060	0.006024096	0.126506024	0.006024096	0.789156627
14	14	0.158031088	0.002590674	0.132124352	0.002590674	0.443005181	0.261658031

Showing 1 to 14 of 556 entries, 7 total columns

Πίνακας 12. Documents and Topic Probabilities – 1ο dataset

Στη συνέχεια από το αρχείο που έχουμε αποθηκεύσει (με τον αύξοντα αριθμό των άρθρων και το topic στο οποίο ανήκει) εισάγουμε στον πίνακα document term matrix ως τελευταία στήλη το topic όπως φαίνεται στην παρακάτω οθόνη



Filter | Cols: « < 135 - 50 > »

erm	terms	test	theory	time	tool	tools	understanding	users	value	various	Topic
1	0	0	0	0	0	0	0	0	1	0	5
0	0	0	0	2	0	0	0	0	0	0	5
0	0	0	1	0	0	0	0	0	0	0	6
0	0	0	0	2	0	0	0	0	0	0	1
0	2	0	1	0	0	0	0	0	0	0	6
0	0	0	0	0	0	0	0	0	0	0	6
0	0	0	0	2	0	0	0	0	0	0	1
0	0	0	0	0	0	0	1	0	0	0	6
0	0	0	0	0	0	0	0	0	0	0	3
0	0	0	0	1	0	0	0	0	2	0	1
0	1	0	0	0	0	0	0	0	0	1	1
1	0	0	0	0	0	0	0	0	0	1	3
0	0	1	3	0	1	0	0	0	0	1	6

Showing 1 to 13 of 556 entries, 185 total columns

Πίνακας 13. Document Term Matrix με τα Topics – 1ο dataset

Για την προσθήκη της ανωτέρω στήλης δημιουργούμε ένα data frame από το document term matrix, διαβάζουμε το αρχείο που περιέχει τα topics ανά άρθρο, μετονομάζουμε τη στήλη σε Category και τέλος δημιουργούμε ένα νέο data frame με το όνομα newcorpus το οποίο έχει ως τελευταία στήλη το topic. Τέλος μετονομάζουμε την στήλη αυτή σε Topic για να είναι πιο κατατοπιστική για εμάς. Ο σχετικός κώδικας φαίνεται ακολούθως:

```
#keep only topics 3,4,5 : 305 abstracts and then apply a classification model
matrixdtm<-as.matrix(dtm)
datadtm<-as.data.frame(matrixdtm)
library(readr)
LDAGibbs_6_DocsToTopics <- read_csv("LDAGibbs 6 DocsToTopics.csv")
#rename columns
names(LDAGibbs_6_DocsToTopics)[2]="Category"
#create corpus
newcorpus<-data.frame(datadtm,LDAGibbs_6_DocsToTopics$Category)
View(newcorpus)
#rename last column
names(newcorpus)[185]="Topic"
```

Στη συνέχεια αφαιρούμε, με τη βοήθεια του παρακάτω κώδικα, τα topics 1, 2, 6 τα οποία αφορούν άλλο επιστημονικό πεδίο, προκειμένου να επεξεργαστούμε περαιτέρω αυτά που αφορούν αμιγώς το τεχνικό χρέος. Επίσης αφαιρούμε τη στήλη Topic η οποία δε μας χρειάζεται πλέον.

```
#remove rows in topics 1,2,6
corpus<-newcorpus[ !(newcorpus$Topic %in% c(1,2,6)), ]
View(corpus)
Corpus<-corpus[,-185]  #remove column named Topic
View(Corpus) # data set 305 entries
```

Το αποτέλεσμα του κώδικα μας δίνει τον ακόλουθο πίνακα.

Filter | Cols: « < 0 - 50 > »

	accumulation	activities	algorithm	amount	analysis	analyze	analyzed	application	applications
1	0	0	0	0	0	0	0	1	
2	0	0	0	0	0	0	0	0	
9	0	2	0	0	0	0	0	0	
12	0	0	0	1	0	0	0	0	
14	1	0	0	0	0	0	0	0	
15	0	0	0	0	0	0	0	0	
16	0	0	0	0	3	0	0	0	
17	0	0	0	0	1	0	0	0	
18	0	0	0	0	2	0	0	0	
19	0	0	1	0	0	0	0	0	
20	0	0	0	0	0	0	0	0	
21	0	0	1	0	0	0	0	0	
22	1	0	0	0	0	0	0	0	

Showing 1 to 13 of 305 entries, 185 total columns

Πίνακας 14. Total Document Term Matrix – 2ο dataset

Μετά την εφαρμογή του topic modeling απομένουν 305 άρθρα τα οποία αφορούν αποκλειστικά το αντικείμενό μας και είναι αυτά που χρειαζόμαστε (2ο dataset). Το αποτέλεσμα αυτό το επιβεβαιώσαμε εξετάζοντας τον πίνακα και τους αύξοντες αριθμούς των άρθρων, που απομένουν, και διαπιστώνουμε ότι είναι τα ίδια ακριβώς άρθρα με αυτά στα οποία καταλήξαμε διαβάζοντας τίτλους και περιλήψεις στο JabRef.

5.6 K - means Clustering στο 1ο Dataset

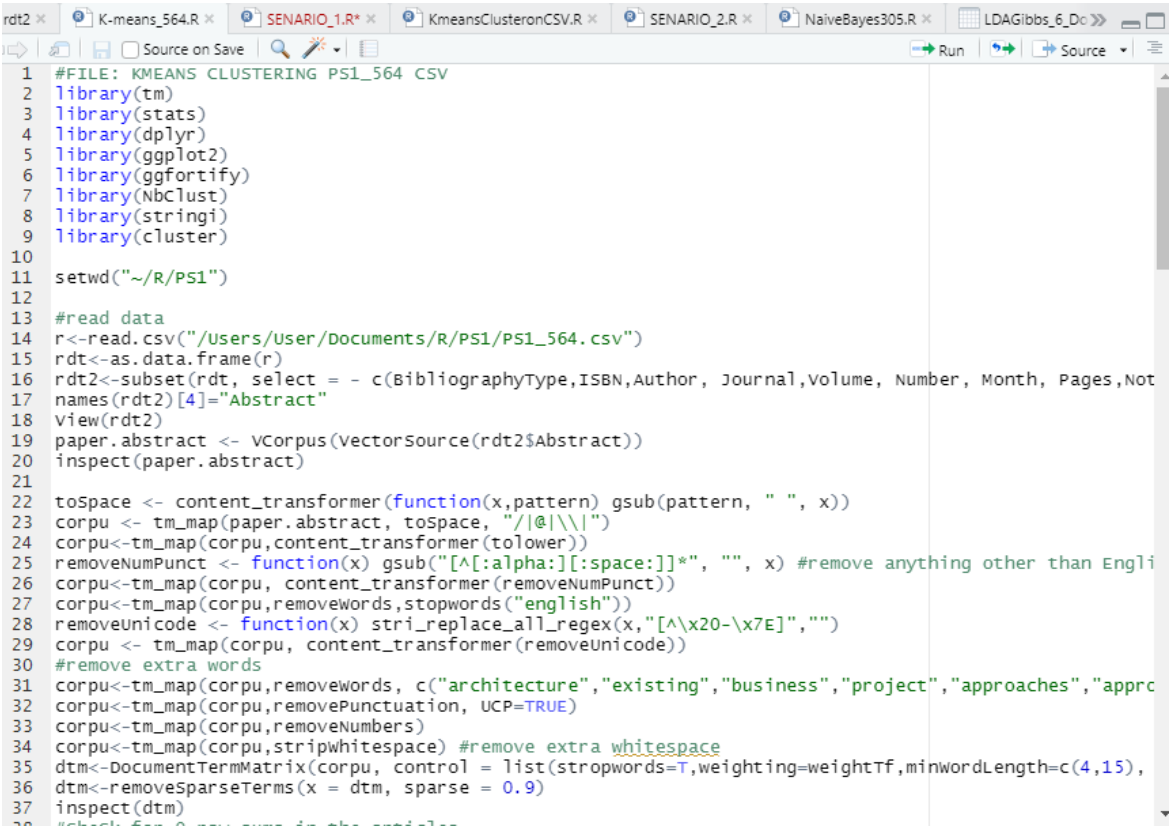
Αφού ολοκληρώσαμε με επιτυχία την ομαδοποίηση του αρχικού dataset με το topic modeling, στη συνέχεια θα επιχειρήσουμε εκ νέου μια συσταδοποίηση των δεδομένων μας με τη βοήθεια της μεθόδου k-means clustering αυτή τη φορά, προκειμένου να διαπιστώσουμε αν μπορούμε να τα ομαδοποιήσουμε επιτυχώς ώστε να κρατήσουμε μόνο τις ομάδες που αφορούν τεχνικό χρέος. Οι δοκιμές μας θα έχουν τις παρακάτω παραμέτρους και επεξηγούνται παρακάτω αναλυτικά.

Test	Weighting of Terms	Number of clusters
1	Term frequency - tf	4
2	Term frequency - tf	5
3	Term frequency – Inverse Document Frequency	6

Test	Weighting of Terms	Number of clusters
4	Term frequency – Inverse Document Frequency	7

Πίνακας 15. Παράμετροι Δοκιμών k - means clustering

Test 1**:** Εκτελούμε τον ακόλουθο κώδικα τον οποίο επεξηγούμε.



Εικόνα 40. Κώδικας R - 1ο dataset

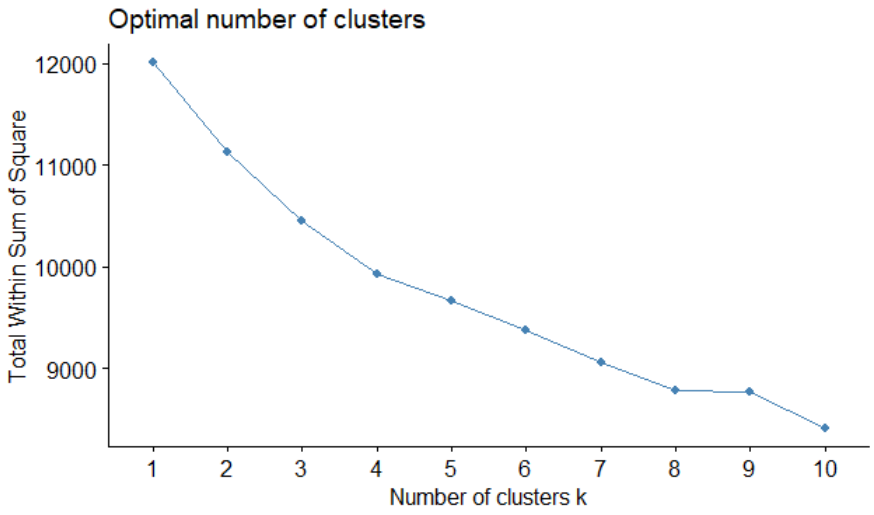
Διαβάζουμε το σχετικό αρχείο (γραμμή 14), μειώνουμε τον αριθμό των στηλών διότι δεν μας χρειάζονται όλες (γραμμή 16) και κρατάμε στο αντικείμενο paper.abstract τις περιλήψεις (γραμμή 19). Στη συνέχεια στις γραμμές 22-34 αφαιρούμε χαρακτήρες, σύμβολα, κενά, σημεία στίξης, αγγλικές stopwords και επιπλέον λέξεις stopwords που έχουμε δηλώσει εμείς (γραμμή 31). Θέτουμε το sparse του πίνακα document term matrix ίσο με 0,9 (γραμμή 36). Έπειτα παράγουμε τον πίνακα document term matrix ο οποίος έχει συνολικά 45 στήλες και φαίνεται ακολούθως:

```
> inspect(dtm)
<<DocumentTermMatrix (documents: 564, terms: 45)>>
Non-/sparse entries: 3991/21389
Sparsity           : 84%
Maximal term length: 13
Weighting          : term frequency (tf)
Sample             :
     Terms
Docs  analysis code development management method proposed quality results system time
  132        0    8           2          0      0        0       2       2      1    0
  133        1    0           1          0      1        1       0       2      0    1
  179        5   10           1          0      0        0       5       0      1    0
  25         0    0           0          4      1        3       0       2      0    0
  274        2    6           0          0      0        0       0       2      0    6
  365        0    2           4          6      2        0       2       0      0    1
  370        0    0           5          3      0        4       1       0      0    2
  391        2    0           0          0      0        0       0       2      5    0
  527        0    6           1          0      0        2       1       1      0    0
  549        3    7           1          0      0        1       0       1      1    1
```

Πίνακας 16. Test 1 - Document Term Matrix 1ου dataset

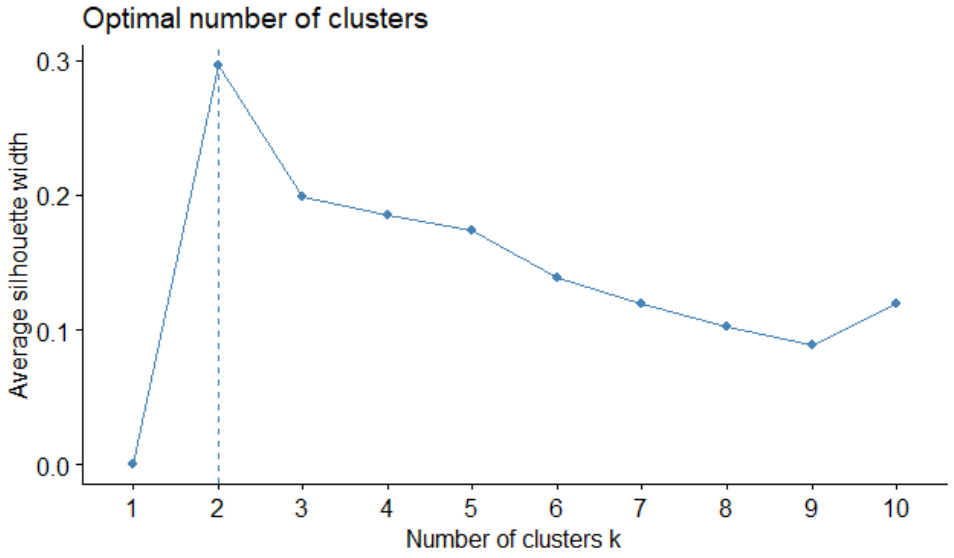
Στον πίνακα βλέπουμε στη 1η στήλη τον αύξοντα αριθμό των άρθρων και οι επόμενες στήλες είναι οι λέξεις με τη μεγαλύτερη συχνότητα εμφάνισης. Το επόμενο ζήτημα είναι να επιλέξουμε τον ιδανικό αριθμό συστάδων. Αυτή η επιλογή θα προκύψει μέσω γραφημάτων με τη βοήθεια του παρακάτω κώδικα.

```
library(ggpubr)
library(factoextra)

fviz_nbclust(myData, kmeans, method = "wss")
fviz_nbclust(myData, kmeans, method = "silhouette")
fviz_nbclust(myData, kmeans, method = "gap_stat")
```



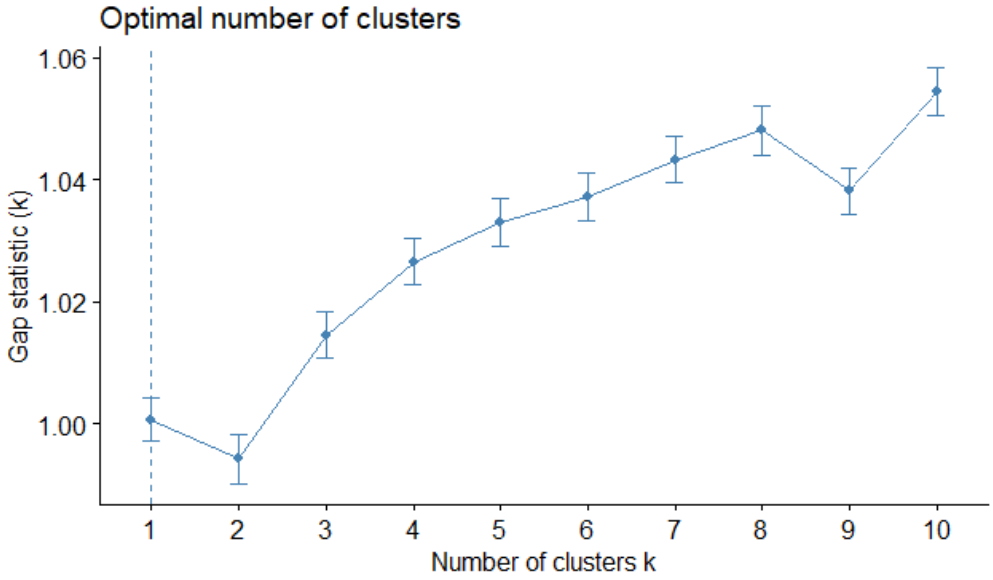
Γράφημα 11. Test1 - Elbow Method

Η TWSS μας προτείνει 4 ή περισσότερες συστάδες.



Γράφημα 12. Test 1- Silhouete Method

Η Average Silhouette προτείνει 2 και κάθε άλλη τιμή από τις μεγαλύτερες είναι υποψήφια προς δοκιμή.



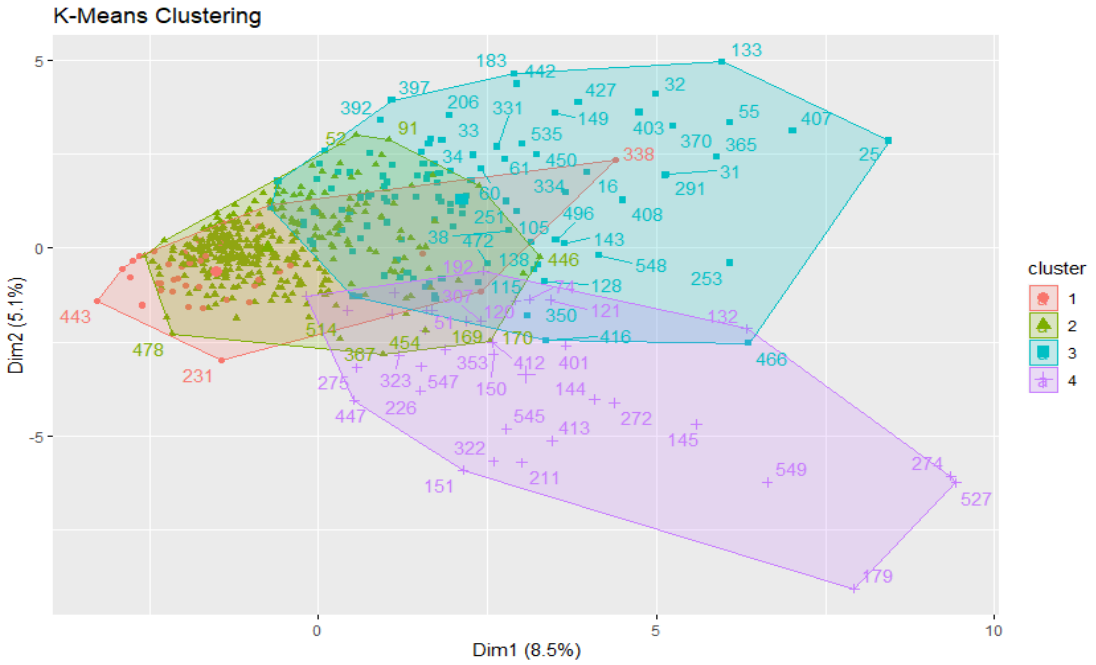
Γράφημα 13. Test 1 - Gap Statistic

Από τη Gap Statistic επιλέγουμε κάποια τιμή από τις μεγαλύτερες.

Επιλέγουμε αρχικά 4 συστάδες και έχουμε τα ακόλουθο γράφημα συσταδοποίησης.

```
# k-means clustering
set.seed(1234)
KM=kmeans(myData, 4, nstart=30)
```

```
# plot clusters
fviz_cluster(KM, data = myData, repel = TRUE,  main="K-Means Clustering")
```



Γράφημα 14. Test 1 - 4 clusters στο 1ο dataset

Παρατηρούμε ότι υπάρχει επικάλυψη των σημείων – data points το οποίο συμβαίνει στην αναπαράσταση ενός 45-διάστατου χώρου στο 2-διάστατο χώρο του επιπέδου.

Στη συνέχεια βλέπουμε τα μεγέθη των συστάδων, τα κέντρα τους και τη συχνότητα εμφάνισης των όρων (terms) ανά συστάδα (cluster).

Console Terminal Markers Jobs

~/R/PS1/

```
> KM$size
[1]  40 378 111  35
> KM$centers
   analysis       case       code   compared    context       cost       data     design developers development engineering
1 0.2000000 0.05000000 0.0500000 0.22500000 0.05000000 0.17500000 0.5000000 0.5500000 0.05000000   0.1500000   0.1250000
2 0.3201058 0.13756614 0.2142857 0.13227513 0.08201058 0.11640212 0.2566138 0.2089947 0.08994709   0.1931217   0.1349206
3 0.3873874 0.36036036 0.2342342 0.03603604 0.27027027 0.31531532 0.3063063 0.1891892 0.42342342   1.9279279   0.3423423
4 0.9142857 0.05714286 4.5428571 0.17142857 0.31428571 0.08571429 0.2000000 0.5428571 0.85714286   0.5428571   0.1142857
  identified  identify     impact  important  interest     level   longterm maintenance management   metaphor     method
1 0.02500000 0.0750000 0.15000000 0.07500000 0.0250000 0.1000000 0.07500000  0.42500000 0.05000000 0.00000000 0.5250000
2 0.08201058 0.1005291 0.08994709 0.10846561 0.0978836 0.1216931 0.06349206  0.07142857 0.11640212 0.08994709 0.2883598
3 0.25225225 0.3603604 0.27927928 0.23423423 0.6036036 0.2432432 0.26126126  0.45045045 1.28828829 0.35135135 0.4324324
4 0.40000000 0.8571429 0.22857143 0.05714286 0.1714286 0.5428571 0.22857143  0.40000000 0.08571429 0.34285714 0.3428571
    methods      model      order performance  performed practitioners   present   process  projects  proposed    provide
1 0.0750000 0.2500000 0.2750000  0.95000000 0.05000000    0.00000000 0.1750000 0.0250000 0.0000000 0.3500000 0.27500000
2 0.1428571 0.2671958 0.1455026  0.17460317 0.09259259    0.07936508 0.1322751 0.1693122 0.1005291 0.2539683 0.08730159
3 0.2432432 0.2972973 0.1621622  0.07207207 0.23423423    0.51351351 0.1261261 0.4504505 0.6036036 0.3963964 0.23423423
4 0.2857143 0.6571429 0.2571429  0.17142857 0.28571429    0.02857143 0.1428571 0.2285714 1.2571429 0.5142857 0.08571429
    quality  research   results    several     source   studies    system   systems   tdscdma      time      tools      work
1 0.0250000 0.1750000 0.7500000 0.0500000 0.0000000 0.0250000 3.7000000 0.4250000 1.4750000 0.6250000 0.10000000 0.1500000
2 0.2169312 0.1746032 0.5026455 0.1031746 0.1216931 0.1005291 0.2804233 0.3280423 0.2089947 0.3730159 0.05555556 0.1481481
3 0.8828829 0.7297297 0.8378378 0.2882883 0.1531532 0.4504505 0.1621622 0.1711712 0.0000000 0.5495495 0.51351351 0.4054054
4 0.6285714 0.1714286 1.0285714 0.1142857 2.0571429 0.4857143 0.5428571 0.2571429 0.0000000 0.5142857 0.51428571 0.5428571
```

Console Terminal Markers Jobs

~/R/PS1/

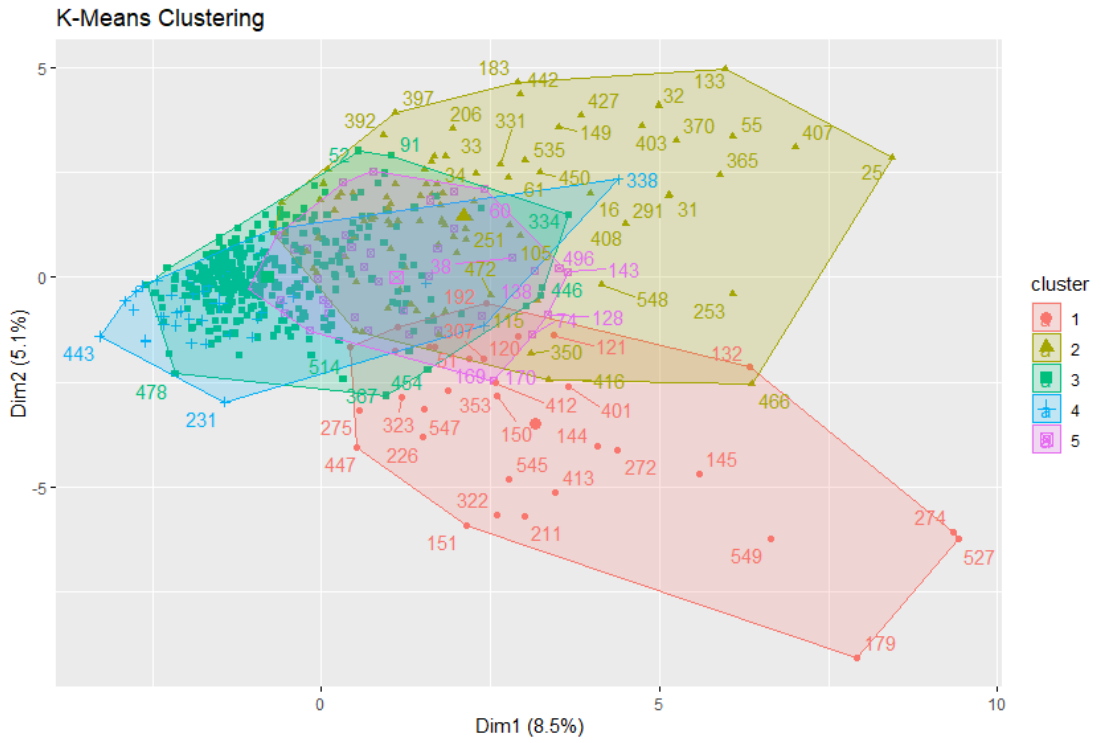
```
> allsums<-data.frame(c1,c2,c3,c4)
> allsums
               c1  c2  c3  c4
analysis        8 121  43  32
case            2  52  40   2
code            2  81  26 159
compared        9  50   4   6
context         2  31  30  11
cost            7  44  35   3
data           20  97  34   7
design         22  79  21  19
developers      2  34  47  30
development     6  73 214  19
engineering     5  51  38   4
identified      1  31  28  14
identify        3  38  40  30
impact          6  34  31   8
important       3  41  26   2
interest        1  37  67   6
level           4  46  27  19
longterm        3  24  29   8
maintenance    17  27  50  14
management      2  44 143   3
metaphor        0  34  39  12
method         21 109  48  12
methods         3  54  27  10
model          10 101  33  23
order          11  55  18   9
performance    38  66   8   6
performed       2  35  26  10
practitioners   0  30  57   1
present         7  50  14   5
process         1  64  50   8
projects        0  38  67  44
proposed       14  96  44  18
provide        11  33  26   3
quality         1  82  98  22
research        7  66  81   6
results        30 190  93  36
several         2  39  32   4
source          0  46  17  72
studies         1  38  50  17
system        148 106  18  19
systems        17 124  19   9
tdscdma        59  79   0   0
time           25 141  61  18
tools           4  21  57  18
work            6  56  45  19
```

Παρατηρούμε ότι στην 1η και 2η συστάδα υπάρχει το ακρωνύμιο tdscdma το οποίο γνωρίζουμε ότι δεν συσχετίζεται με το τεχνικό χρέος. Κατά συνέπεια οι δύο πρώτες συστάδες έχουν, μεταξύ άλλων, και μη σχετικά άρθρα. Συνολικά αυτές οι δύο έχουν 40+378=418>305, ενώ οι άλλες δύο έχουν συνολικά 146<305 άρθρα. Συμπεραίνουμε λοιπόν ότι ο διαχωρισμός αυτός δεν είναι αποτελεσματικός. Παρακάτω έχουμε τα αποτελέσματα των δεικτών ποιότητας διαχωρισμού των συστάδων οι οποίοι έχουν μικρές τιμές και μας δείχνουν ότι η συσταδοποίηση δεν είναι πετυχημένη.

```
> #quality of partinioning
> bss<-KM$betweenss
> tss<-KM$totss
> qual<-(bss/tss)*100
> qual
[1] 17.32385
```

```
> # Statistics for k-means clustering
> km_stats <- cluster.stats(dist(myData),  KM$cluster)
> # Dun index
> km_stats$dunn
[1] 0.07537784
```

Test 2 : Εφόσον η δοκιμή δεν ήταν επιτυχής θα συνεχίσουμε και θα χωρίσουμε το dataset σε 5 συστάδες αυτή τη φορά προκειμένου να διαπιστώσουμε αν επιτυγχάνουμε καλύτερη ομαδοποίηση. Με επανάληψη του ίδιου κώδικα, αλλάζουμε μόνο το πλήθος των συστάδων και προκύπτει το ακόλουθο γράφημα.



Γράφημα 15. Test 2 - 5 clusters στο 1ο dataset

Ομοίως παρουσιάζουμε το πλήθος των άρθρων ανά συστάδα, τα κέντρα τους και τη συχνότητα εμφάνισης λέξεων.

```
> KM$size
[1]  33  93 358  40  40
> KM$centers
   analysis       case      code   compared    context       cost      data    design developers development engineering
1 0.9090909 0.06060606 4.6363636 0.18181818 0.33333333 0.09090909 0.2121212 0.5757576  0.9090909   0.5151515  0.09090909
2 0.3978495 0.33333333 0.2258065 0.04301075 0.30107527 0.30107527 0.2258065 0.1935484  0.4086022   2.1827957  0.36559140
3 0.3240223 0.13966480 0.1927374 0.13128492 0.07821229 0.11731844 0.2513966 0.2011173  0.0698324   0.1871508  0.11731844
4 0.2000000 0.05000000 0.0500000 0.22500000 0.05000000 0.17500000 0.5000000 0.5500000  0.0500000   0.1500000  0.12500000
5 0.3250000 0.27500000 0.5750000 0.07500000 0.12500000 0.22500000 0.5000000 0.2500000  0.4500000   0.4750000  0.35000000
  identified  identify     impact  important  interest     level   longterm maintenance management  metaphor     method
1 0.42424242 0.8484848 0.24242424 0.06060606 0.1818182 0.5454545 0.21212121  0.39393939 0.09090909 0.3333333 0.3636364
2 0.25806452 0.3763441 0.26881720 0.13978495 0.5053763 0.1505376 0.30107527  0.46236559 1.46236559 0.3118280 0.3763441
3 0.08100559 0.1033520 0.07541899 0.11452514 0.1145251 0.1368715 0.06424581  0.06703911 0.12011173 0.0698324 0.3184358
4 0.02500000 0.0750000 0.15000000 0.07500000 0.0250000 0.1000000 0.07500000  0.42500000 0.05000000 0.0000000 0.5250000
5 0.15000000 0.2000000 0.32500000 0.32500000 0.4000000 0.2750000 0.07500000  0.27500000 0.20000000 0.5000000 0.2000000
    methods     model     order performance  performed practitioners     present    process    projects  proposed     provide
1 0.2727273 0.6969697 0.2727273  0.18181818 0.30303030    0.03030303 0.15151515 0.2424242 1.33333333 0.5454545 0.09090909
2 0.1935484 0.2365591 0.1397849  0.02150538 0.23655914    0.45161290 0.09677419 0.4408602 0.60215054 0.3655914 0.23655914
3 0.1480447 0.2541899 0.1508380  0.18435754 0.08938547    0.10055866 0.13407821 0.1424581 0.09776536 0.2737430 0.08379888
4 0.0750000 0.2500000 0.2750000  0.95000000 0.05000000    0.00000000 0.17500000 0.0250000 0.00000000 0.3500000 0.27500000
5 0.2750000 0.5250000 0.1500000  0.15000000 0.17500000    0.22500000 0.17500000 0.5500000 0.35000000 0.2000000 0.17500000
     quality  research   results    several     source   studies    system   systems    tdscdma      time      tools      work
1 0.51515152 0.1818182 1.0606061 0.1212121 2.1515152 0.5151515 0.4848485 0.1818182 0.0000000 0.5454545 0.45454545 0.5757576
2 0.34408602 0.8602151 0.8064516 0.2580645 0.1397849 0.5161290 0.1505376 0.1612903 0.0000000 0.6451613 0.47311828 0.4086022
3 0.09217877 0.1703911 0.5111732 0.1117318 0.1061453 0.1089385 0.2681564 0.3072626 0.2206704 0.3826816 0.05865922 0.1201117
4 0.02500000 0.1750000 0.7500000 0.0500000 0.0000000 0.0250000 3.7000000 0.4250000 1.4750000 0.6250000 0.10000000 0.1500000
5 3.00000000 0.1500000 0.6500000 0.1750000 0.3250000 0.0250000 0.4250000 0.5250000 0.0000000 0.1250000 0.40000000 0.5000000
> |
```

Από τα παρακάτω αποτελέσματα διαπιστώνουμε ότι η 3η και 4η συστάδα έχουν και μη σχετικά (με το τεχνικό χρέος) άρθρα και περιέχουν συνολικά 358+40=398>305 ενώ οι υπόλοιπες έχουν 33+93+40=166<305. Συμπεραίνουμε λοιπόν ότι και σε αυτή τη περίπτωση ο διαχωρισμός δεν είναι επιτυχής.

Console Terminal Markers Jobs

~/R/PS1/

```
> allsums<-data.frame(c1,c2,c3,c4,c5)
> allsums
                c1  c2  c3  c4  c5
analysis        30  37 116   8  13
case             2  31  50   2  11
code           153  21  69   2  23
compared         6   4  47   9   3
context         11  28  28   2   5
cost             3  28  42   7   9
data             7  21  90  20  20
design          19  18  72  22  10
developers      30  38  25   2  18
development     17 203  67   6  19
engineering      3  34  42   5  14
identified      14  24  29   1   6
identify        28  35  37   3   8
impact           8  25  27   6  13
important        2  13  41   3  13
interest         6  47  41   1  16
level           18  14  49   4  11
longterm         7  28  23   3   3
maintenance     13  43  24  17  11
management       3 136  43   2   8
metaphor        11  29  25   0  20
method          12  35 114  21   8
methods          9  18  53   3  11
model           23  22  91  10  21
order            9  13  54  11   6
performance      6   2  66  38   6
performed       10  22  32   2   7
practitioners    1  42  36   0   9
present          5   9  48   7   7
process          8  41  51   1  22
projects        44  56  35   0  14
proposed        18  34  98  14   8
provide          3  22  30  11   7
quality         17  32  33   1 120
research         6  80  61   7   6
results         35  75 183  30  26
several          4  24  40   2   7
source          71  13  38   0  13
studies         17  48  39   1   1
system          16  14  96 148  17
systems          6  15 110  17  21
tdscdma          0   0  79  59   0
time            18  60 137  25   5
tools           15  44  21   4  16
work            19  38  43   6  20
```

Οι τιμές των δεικτών διαχωρισμού φαίνονται παρακάτω και διαπιστώνουμε ότι δεν είναι ικανοποιητικές.

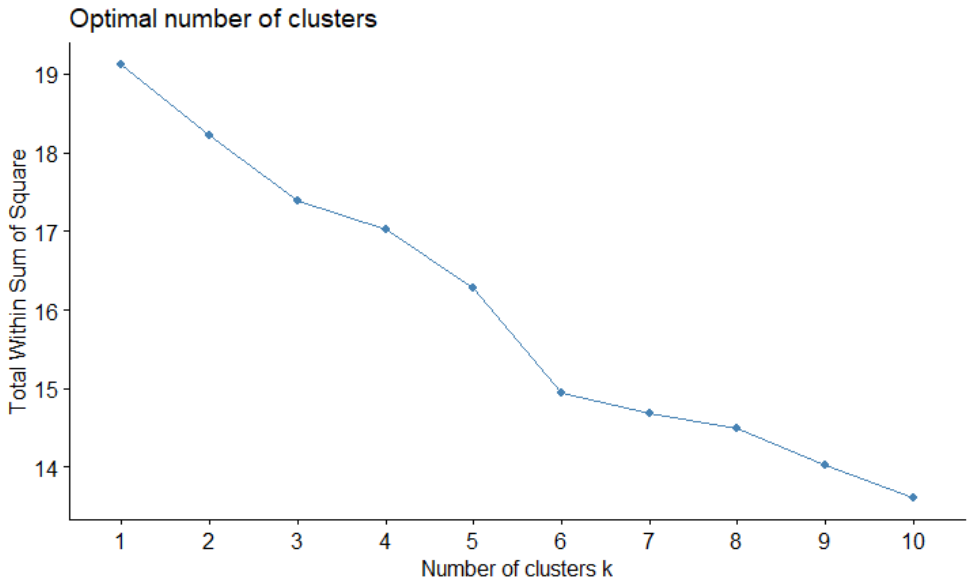
```
> #quality of partinioning
> bss<-KM$betweenss
> tss<-KM$totss
> qual<-(bss/tss)*100
> qual
[1] 20.15415
```

```
> #Dunn Index
> library(fpc)
> # Statistics for k-means clustering
> km_stats <- cluster.stats(dist(myData),  KM$cluster)
> # Dun index
> km_stats$dunn
[1] 0.07537784
```

Test 3: Στη συνέχεια θα εκτελέσουμε τον κώδικα με weighting το tf-idf το οποίο εντοπίζει με ευστοχία εκείνες τις λέξεις που επικρατούν στο τρέχον κείμενο και λιγότερο στο σύνολο των υπόλοιπων κειμένων. Παραθέτουμε τον πίνακα document term matrix.

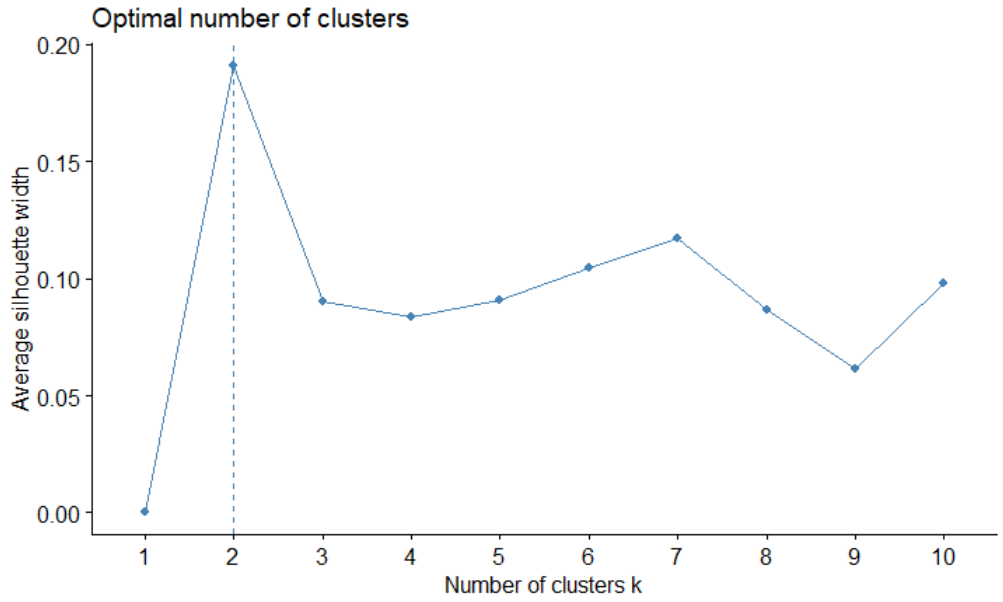
```
> inspect(dtm)
<<DocumentTermMatrix (documents: 564, terms: 45)>>
Non-/sparse entries: 3991/21389
Sparsity           : 84%
Maximal term length: 13
Weighting          : term frequency - inverse document frequency (normalized) (tf-idf)
Sample             :
     Terms
Docs    analysis       code development management      model    quality     system    systems   tdscdma
  147 0.05130221 0.05611834  0.04262308 0.06312104 0.00000000 0.00000000 0.00000000 0.00000000 0.0000000
  166 0.00000000 0.20406669  0.00000000 0.00000000 0.00000000 0.21319414 0.00000000 0.00000000 0.0000000
  192 0.00000000 0.19953187  0.03788718 0.00000000 0.00000000 0.00000000 0.04203609 0.00000000 0.0000000
  196 0.13680590 0.00000000  0.00000000 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000 0.0000000
  236 0.00000000 0.00000000  0.00000000 0.00000000 0.00000000 0.00000000 0.11464387 0.00000000 0.2983246
  276 0.08208354 0.08978934  0.00000000 0.00000000 0.20081558 0.28141626 0.00000000 0.00000000 0.0000000
  466 0.00000000 0.02522173  0.03831288 0.02836901 0.00000000 0.00000000 0.00000000 0.00000000 0.0000000
  473 0.10260443 0.00000000  0.08524616 0.00000000 0.00000000 0.11725677 0.00000000 0.25396385 0.0000000
  539 0.00000000 0.15480921  0.00000000 0.00000000 0.08655844 0.00000000 0.00000000 0.00000000 0.0000000
  70  0.00000000 0.00000000  0.00000000 0.00000000 0.00000000 0.04342843 0.00000000 0.09406069 0.0000000
```

Ακολουθούν τα γραφήματα με τα οποία θα επιλέξουμε το πλήθος των συστάδων.



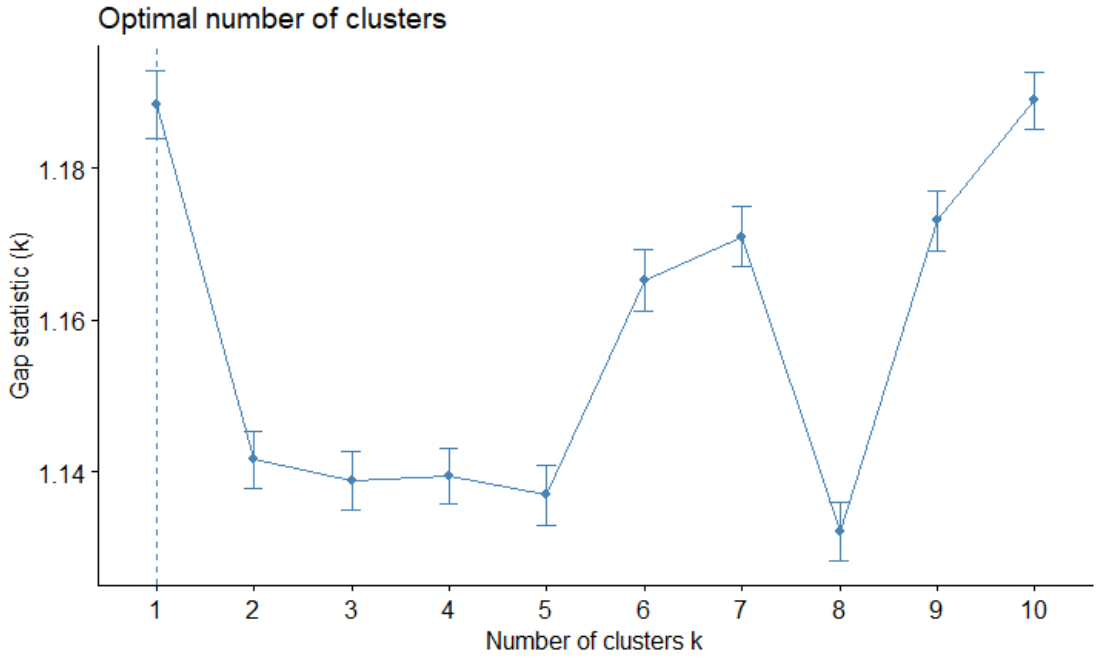
Γράφημα 16. Test 3 - Elbow Method

Από το γράφημα επιλέγουμε 6 συστάδες διότι εκεί σχηματίζεται «αγκώνας».



Γράφημα 17. Test 3 - Silhouete Method

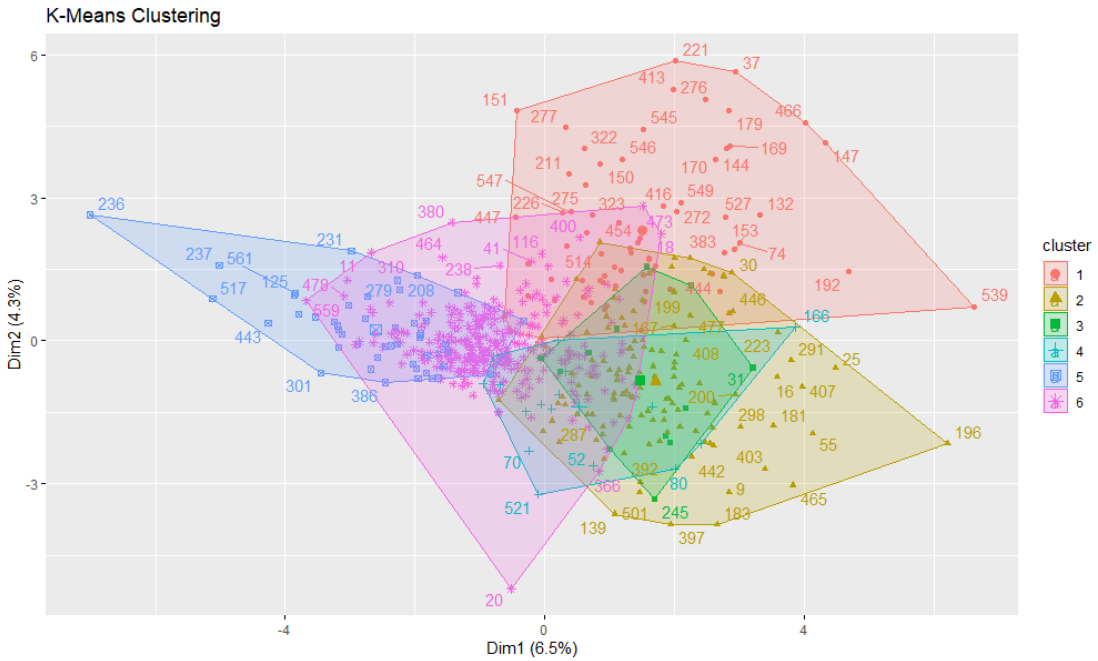
Εδώ από το γράφημα συμπεραίνουμε ότι η καλύτερη επιλογή μετά τις 2 συστάδες είναι οι 7.



Γράφημα 18. Test 3 - Gap Statistic Method

Εδώ προκύπτει ότι μια καλή επιλογή είναι οι 6 συστάδες και μετά οι 7.

Συνδυάζοντας τα ανωτέρω αποτελέσματα επιλέγουμε 6 συστάδες. Με επανεκτέλεση του κώδικα με αριθμό συστάδων 6 λαμβάνουμε το ακόλουθο γράφημα.



Γράφημα 19. Test 3 - 6 clusters στο 1ο dataset

Στη συνέχεια παρουσιάζουμε τα άρθρα ανά συστάδα, τα κέντρα τους και τη συχνότητα εμφάνισης λέξεων (tf-idf) σε αυτές.

```
> KM$size
[1]  70 124  12  16  48 294
> KM$centers
     analysis        case        code    compared     context        cost        data      design
1 0.019530846 0.006374428 0.090670733 0.004943686 0.010656167 0.004288602 0.007026518 0.015914880
2 0.008064570 0.012823091 0.005142839 0.001240094 0.009382984 0.013049862 0.005528463 0.005215221
3 0.000000000 0.011280454 0.024715270 0.004610365 0.018784521 0.013043959 0.008013799 0.004435050
4 0.015081026 0.007305201 0.012754168 0.000000000 0.007603826 0.015946238 0.023101953 0.012239524
5 0.006397181 0.001838006 0.002753670 0.012536526 0.000000000 0.002122231 0.011118869 0.025686628
6 0.011336225 0.004854415 0.003614220 0.006719859 0.002874767 0.007461413 0.011093060 0.008079850
   developers development engineering  identified    identify      impact   important    interest
1 0.021879928 0.011765415 0.003226663 0.008946356 0.019523014 0.012242278 0.003248128 0.002867056
2 0.012578396 0.037676329 0.009544532 0.010828852 0.013127270 0.012838033 0.008728102 0.008590559
3 0.004952658 0.014028600 0.000000000 0.006888878 0.004603960 0.000000000 0.006595192 0.227118564
4 0.007037988 0.008773774 0.213067806 0.003827154 0.003683168 0.000000000 0.002095628 0.009024182
5 0.000000000 0.004145387 0.000000000 0.000000000 0.000000000 0.002062233 0.005080074 0.000000000
6 0.004022479 0.006639484 0.002991788 0.003065887 0.002498244 0.003943538 0.006535364 0.003143114
       level    longterm  maintenance  management    metaphor      method     methods       model
1 0.009054816 0.005357263 0.0144982469 0.003184673 0.007917982 0.008139849 0.008353032 0.019061249
2 0.006993759 0.011677542 0.0160812070 0.048068009 0.020439085 0.009198743 0.008139781 0.006272368
3 0.044672546 0.003444439 0.0328708480 0.026818941 0.003635590 0.001644302 0.007030755 0.005229572
4 0.000000000 0.007251450 0.0000000000 0.007720482 0.048343438 0.009786473 0.004605590 0.001651444
5 0.001176440 0.000000000 0.0009020913 0.000000000 0.000000000 0.012090038 0.009682490 0.012800384
6 0.005995156 0.003787299 0.0032424390 0.001678906 0.001423190 0.011225324 0.005892619 0.011840127
       order performance   performed practitioners     present     process    projects    proposed
1 0.009138838 0.004302911 0.009489643   0.004707663 0.004513742 0.003223979 0.041955987 0.007851344
2 0.007835424 0.001018040 0.006158653   0.026965638 0.003118358 0.017042994 0.015428385 0.008532591
3 0.003447040 0.000000000 0.009407253   0.000000000 0.007321321 0.004406222 0.006072508 0.010572047
4 0.001474251 0.000000000 0.000000000   0.001695026 0.002397964 0.016119481 0.000000000 0.009926227
5 0.005251454 0.033308168 0.000000000   0.000000000 0.004705715 0.003853571 0.001739286 0.025571971
6 0.006852039 0.010616531 0.005325787   0.001414754 0.008195902 0.006916447 0.001656224 0.009777271
      provide     quality    research     results     several      source     studies      system
1 0.004613968 0.023273072 0.006103432 0.011181095 0.008356885 0.048283464 0.012118457 0.009112063
2 0.008332056 0.025190974 0.028533367 0.009980663 0.011497541 0.003652005 0.011106739 0.004935493
3 0.006740049 0.013752338 0.023399310 0.013986121 0.007321321 0.007409518 0.010759516 0.000000000
4 0.001587532 0.020561585 0.032344706 0.007341937 0.000000000 0.000000000 0.006951082 0.009028648
5 0.006721238 0.001814920 0.004244331 0.013468026 0.001002324 0.004348371 0.001767635 0.050136677
6 0.004965986 0.006918923 0.002776124 0.009838657 0.004403726 0.002271332 0.005200604 0.016940502
      systems     tdscdma        time       tools        work
1 0.007779779 0.000000000 0.010341112 0.026632163 0.017382052
2 0.004360000 0.000000000 0.005415036 0.013445795 0.012322456
3 0.000000000 0.000000000 0.009138279 0.007911614 0.004825757
4 0.025631950 0.000000000 0.007605059 0.006119884 0.006518785
5 0.004859002 0.167347955 0.010181362 0.000000000 0.002979050
6 0.017267835 0.002190281 0.015802020 0.001583505 0.004502252
```

Console Terminal Markers Jobs

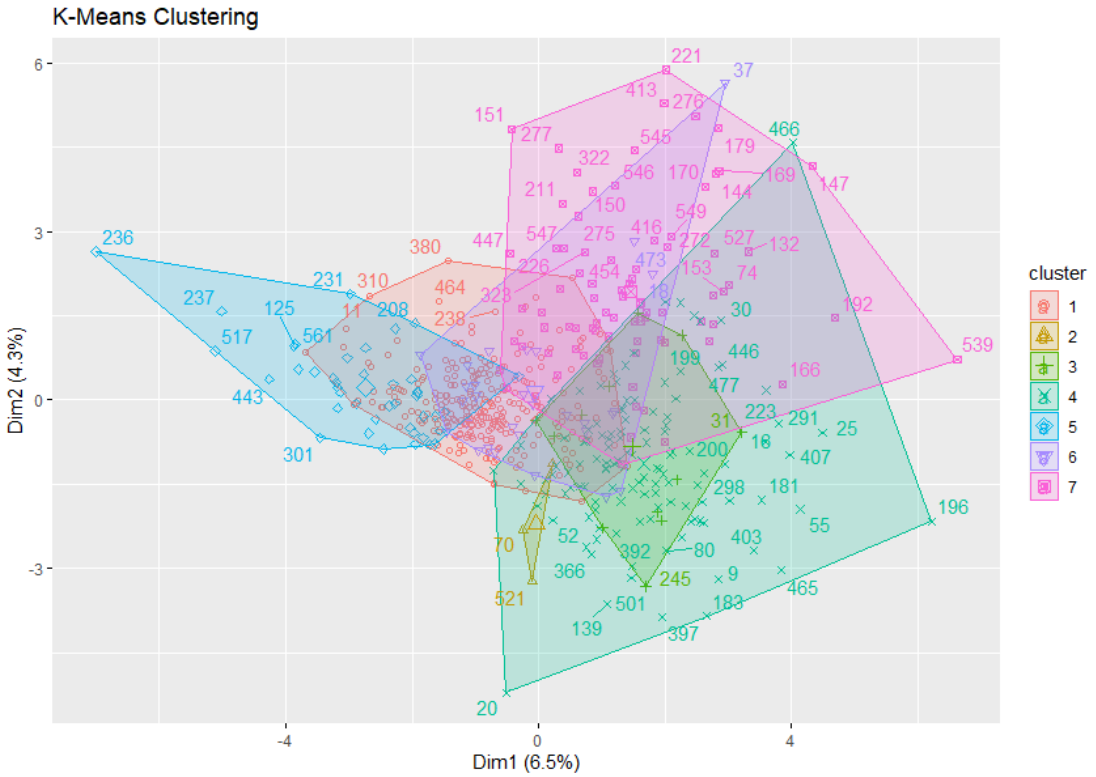
~/R/PS1/

```
> allsums<-data.frame(c1,c2,c3,c4,c5,c6)
> allsums
                    c1        c2         c3         c4         c5        c6
analysis      1.3671592 1.0000067 0.00000000 0.24129642 0.30706468 3.3328502
case          0.4462100 1.5900633 0.13536545 0.11688322 0.08822427 1.4271981
code          6.3469513 0.6377120 0.29658324 0.20406669 0.13217616 1.0625805
compared      0.3460580 0.1537716 0.05532438 0.00000000 0.60175324 1.9756386
context       0.7459317 1.1634900 0.22541425 0.12166122 0.00000000 0.8451814
cost          0.3002022 1.6181828 0.15652751 0.25513980 0.10186711 2.1936554
data          0.4918562 0.6855294 0.09616558 0.36963125 0.53370572 3.2613597
design        1.1140416 0.6466874 0.05322060 0.19583239 1.23295814 2.3754758
developers    1.5315950 1.5597211 0.05943190 0.11260781 0.00000000 1.1826089
development   0.8235790 4.6718648 0.16834320 0.14038038 0.19897857 1.9520083
engineering   0.2258664 1.1835219 0.00000000 3.40908489 0.00000000 0.8795858
identified    0.6262450 1.3427776 0.08266653 0.06123447 0.00000000 0.9013708
identify      1.3666109 1.6277815 0.05524751 0.05893068 0.00000000 0.7344838
impact        0.8569595 1.5919161 0.00000000 0.00000000 0.09898719 1.1594002
important     0.2273690 1.0822846 0.07914231 0.03353005 0.24384356 1.9213971
interest      0.2006939 1.0652293 2.72542277 0.14438691 0.00000000 0.9240754
level         0.6338371 0.8672262 0.53607055 0.00000000 0.05646913 1.7625759
longterm      0.3750084 1.4480152 0.04133327 0.11602320 0.00000000 1.1134659
maintenance   1.0148773 1.9940697 0.39445018 0.00000000 0.04330038 0.9532771
management    0.2229271 5.9604331 0.32182729 0.12352771 0.00000000 0.4935983
metaphor      0.5542587 2.5344465 0.04362708 0.77349501 0.00000000 0.4184180
method        0.5697894 1.1406441 0.01973162 0.15658357 0.58032183 3.3002454
methods       0.5847122 1.0093329 0.08436906 0.07368943 0.46475952 1.7324300
model         1.3342874 0.7777736 0.06275487 0.02642310 0.61441845 3.4809974
order         0.6397187 0.9715926 0.04136448 0.02358801 0.25206980 2.0144993
performance   0.3012038 0.1262369 0.00000000 0.00000000 1.59879204 3.1212602
performed     0.6642750 0.7636730 0.11288703 0.00000000 0.00000000 1.5657813
practitioners 0.3295364 3.3437391 0.00000000 0.02712042 0.00000000 0.4159377
present       0.3159619 0.3866763 0.08785585 0.03836743 0.22587430 2.4095953
process       0.2256785 2.1133313 0.05287466 0.25791170 0.18497140 2.0334356
projects      2.9369191 1.9131198 0.07287010 0.00000000 0.08348574 0.4869298
proposed      0.5495941 1.0580413 0.12686456 0.15881963 1.22745460 2.8745177
provide       0.3229778 1.0331749 0.08088058 0.02540051 0.32261945 1.4600000
quality       1.6291151 3.1236808 0.16502805 0.32898537 0.08711614 2.0341633
research      0.4272402 3.5381375 0.28079172 0.51751530 0.20372791 0.8161805
results       0.7826766 1.2376022 0.16783346 0.11747099 0.64646524 2.8925650
several       0.5849819 1.4256951 0.08785585 0.00000000 0.04811154 1.2946953
source        3.3798424 0.4528486 0.08891421 0.00000000 0.20872182 0.6677715
studies       0.8482920 1.3772356 0.12911419 0.11121731 0.08484647 1.5289776
system        0.6378444 0.6120011 0.00000000 0.14445837 2.40656051 4.9805075
systems       0.5445846 0.5406400 0.00000000 0.41011120 0.23323211 5.0767436
tdscdma       0.0000000 0.0000000 0.00000000 0.00000000 8.03270185 0.6439426
time          0.7238778 0.6714644 0.10965935 0.12168094 0.48870539 4.6457939
tools         1.8642514 1.6672785 0.09493937 0.09791815 0.00000000 0.4655503
work          1.2167437 1.5279845 0.05790909 0.10430056 0.14299439 1.3236620
```

Παρατηρούμε ότι μόνο οι δύο τελευταίες συστάδες έχουν μεταξύ άλλων και τον όρο tdscdma που σημαίνει ότι περιέχουν και μη σχετικά άρθρα. Συνολικά και οι δύο μαζί έχουν 48+294 = 342 >259 άρθρα. Οι υπόλοιπες συστάδες έχουν συνολικά 70+124+12+16 = 222<305. Συνεπώς καταλαβαίνουμε ότι ο αλγόριθμος δεν διαχωρίζει επιτυχώς τα άρθρα που έχουμε. Ωστόσο ο δείκτης bss/tss είναι μεγαλύτερος ενώ ο dunn είναι μικρότερος σε σχέση με την προηγούμενη δοκιμή.

```
> #quality of partinioning
> bss<-KM$betweenss
> tss<-KM$totss
> qual<-(bss/tss)*100
> qual
[1] 21.78934
>
> #Dunn Index
> library(fpc)
> # Statistics for k-means clustering
> km_stats <- cluster.stats(dist(myData),  KM$cluster)
> # Dun index
> km_stats$dunn
[1] 0.06170569
```

Test 4: Στη συνέχεια θα επιχειρήσουμε ακόμα μία δοκιμή με 7 συστάδες. Η μέθοδος μας δίνει το ακόλουθο γράφημα.



Γράφημα 20. Test 4 - 7 clusters στο 1ο dataset

Ακολουθούν τα άρθρα ανά συστάδα, τα κέντρα τους και η συχνότητα εμφάνισης των λέξεων (tf-idf) ανά συστάδα.

```
> KM$size
[1] 272   3  12 122  42  29  84
> KM$centers
     analysis        case         code     compared      context         cost         data       design  developers development
1 0.011212848 0.005287557 0.0033504275 0.006984191 0.001915513 0.003947591 0.010619584 0.006373114 0.002338604 0.005127048
2 0.000000000 0.017967486 0.0000000000 0.000000000 0.000000000 0.019807494 0.105996023 0.030974953 0.000000000 0.000000000
3 0.000000000 0.011280454 0.0247152698 0.004610365 0.018784521 0.013043959 0.008013799 0.004435050 0.004952658 0.014028600
4 0.008513459 0.010508620 0.0041891615 0.001223015 0.010442724 0.021615198 0.004594762 0.005495819 0.013485860 0.040240186
5 0.006559383 0.001034376 0.0009543935 0.013152068 0.000000000 0.002425407 0.012707279 0.029356146 0.000000000 0.004113071
6 0.017930505 0.009059328 0.0043510567 0.004592408 0.010475087 0.005354293 0.014637106 0.019564287 0.020124648 0.010823438
7 0.016242404 0.007002916 0.0808908042 0.004080321 0.009255414 0.005258146 0.007343150 0.015092512 0.018113605 0.011860984
  engineering  identified    identify      impact   important    interest       level    longterm maintenance  management
1 0.002848472 0.002868504 0.002541129 0.003010234 0.006571735 0.002991208 0.005731848 0.002747740 0.001756360 0.001355973
2 0.476025100 0.020411490 0.000000000 0.000000000 0.000000000 0.019682508 0.000000000 0.000000000 0.000000000 0.000000000
3 0.000000000 0.006888878 0.004603960 0.000000000 0.006595192 0.227118564 0.044672546 0.003444439 0.032870848 0.026818941
4 0.017496917 0.010262515 0.013149855 0.014336689 0.006766637 0.008430948 0.005718846 0.014650956 0.015491149 0.047702413
5 0.000000000 0.000000000 0.000000000 0.002356838 0.004639002 0.000000000 0.001344503 0.000000000 0.001030962 0.000000000
6 0.030854004 0.005306103 0.003497864 0.003867659 0.009247787 0.001839077 0.005830831 0.004920775 0.013906467 0.005619581
7 0.005545350 0.008145911 0.016558326 0.011050642 0.005147129 0.004521672 0.009973611 0.004464386 0.014181979 0.005345231
     metaphor      method     methods       model       order  performance   performed practitioners     present
1 0.001525786 0.012707057 0.006929926 0.011395168 0.007545480 0.0115259207 0.005257038   0.003250877 0.007318658
2 0.017918264 0.000000000 0.000000000 0.000000000 0.000000000 0.0000000000 0.000000000   0.000000000 0.000000000
3 0.003635590 0.001644302 0.007030755 0.005229572 0.003447040 0.0000000000 0.009407253   0.000000000 0.007321321
4 0.017550347 0.007770268 0.008095091 0.004240864 0.006185967 0.0007879722 0.004966277   0.022096224 0.003032787
5 0.000000000 0.010058782 0.006056773 0.011564060 0.006001662 0.0363122784 0.000000000   0.000000000 0.005377959
6 0.000000000 0.009446727 0.002888497 0.014191288 0.004600239 0.0031031173 0.008149534   0.002864749 0.015053246
7 0.019889373 0.007700810 0.007788480 0.020473702 0.008439679 0.0035857596 0.008590390   0.005396165 0.004206929
      process     projects     proposed      provide      quality     research      results      several       source      studies
1 0.006933107 0.002200865 0.011430001 0.005130093 0.005912811 0.001907644 0.010096065 0.004128995 0.001904952 0.004537166
2 0.033291455 0.000000000 0.000000000 0.000000000 0.014476145 0.000000000 0.007425391 0.000000000 0.000000000 0.000000000
3 0.004406222 0.006072508 0.010572047 0.006740049 0.013752338 0.023399310 0.013986121 0.007321321 0.007409518 0.010759516
4 0.016603234 0.012353550 0.006977867 0.007946103 0.010068522 0.033076333 0.009174791 0.010222452 0.002835388 0.012438904
5 0.004404081 0.000000000 0.023586150 0.007681415 0.002074194 0.004850664 0.014197697 0.001145513 0.004969567 0.002020154
6 0.004303863 0.002693675 0.006823442 0.001276885 0.013608660 0.007070745 0.009351272 0.008057874 0.002954437 0.007680590
7 0.005884179 0.038530588 0.008567488 0.005234686 0.045728818 0.006426549 0.010970948 0.008350814 0.042270472 0.010611072
       system      systems      tdscdma         time        tools         work
1 0.015838182 0.005522864 0.004423078 0.014813575 0.001591071 0.005174461
2 0.000000000 0.031353562 0.000000000 0.000000000 0.019100083 0.000000000
3 0.000000000 0.000000000 0.000000000 0.009138279 0.007911614 0.004825757
4 0.005289362 0.004477579 0.000000000 0.010883949 0.013672536 0.011687245
5 0.054989385 0.005553145 0.177942076 0.007909972 0.000000000 0.001778366
6 0.025586038 0.132794393 0.000000000 0.006063678 0.014938199 0.000000000
7 0.009244463 0.006886876 0.000000000 0.009360985 0.017900830 0.016758290
```

```
> allsums<-data.frame(c1,c2,c3,c4,c5,c6,c7)
> allsums
                    c1         c2         c3         c4         c5         c6         c7
analysis      3.0498945 0.00000000 0.00000000 1.03864203 0.27549408 0.51998466 1.3643619
case          1.4382155 0.05390246 0.13536545 1.28205159 0.04344377 0.26272052 0.5882450
code          0.9113163 0.00000000 0.29658324 0.51107770 0.04008453 0.12618064 6.7948275
compared      1.8997000 0.00000000 0.05532438 0.14920783 0.55238687 0.13317984 0.3427469
context       0.5210196 0.00000000 0.22541425 1.27401236 0.00000000 0.30377753 0.7774548
cost          1.0737448 0.05942248 0.15652751 2.63705416 0.10186711 0.15527450 0.4416843
data          2.8885268 0.31798807 0.09616558 0.56056097 0.53370572 0.42447609 0.6168246
design        1.7334871 0.09292486 0.05322060 0.67048996 1.23295814 0.56736431 1.2677710
developers    0.6361003 0.00000000 0.05943190 1.64527487 0.00000000 0.58361481 1.5215428
development   1.3945570 0.00000000 0.16834320 4.90930275 0.17274898 0.31387969 0.9963226
engineering   0.7747843 1.42807530 0.00000000 2.13462393 0.00000000 0.89476612 0.4658094
identified    0.7802330 0.06123447 0.08266653 1.25202681 0.00000000 0.15387699 0.6842566
identify      0.6911872 0.00000000 0.05524751 1.60428232 0.00000000 0.10143806 1.3908994
impact        0.8187837 0.00000000 0.00000000 1.74907606 0.09898719 0.11216211 0.9282539
important     1.7875118 0.00000000 0.07914231 0.82552973 0.19483810 0.26818581 0.4323588
interest      0.8136085 0.05904752 2.72542277 1.02857569 0.00000000 0.05333325 0.3798205
level         1.5590625 0.00000000 0.53607055 0.69769926 0.05646913 0.16909411 0.8377833
longterm      0.7473852 0.00000000 0.04133327 1.78741661 0.00000000 0.14270247 0.3750084
maintenance   0.4777300 0.00000000 0.39445018 1.88992017 0.04330038 0.40328754 1.1912863
management    0.3688246 0.00000000 0.32182729 5.81969433 0.00000000 0.16296785 0.4489994
metaphor      0.4150138 0.05375479 0.04362708 2.14114228 0.00000000 0.00000000 1.6707073
method        3.4563196 0.00000000 0.01973162 0.94797275 0.42246886 0.27395508 0.6468680
methods       1.8849398 0.00000000 0.08436906 0.98760109 0.25438446 0.08376643 0.6542323
model         3.0994858 0.00000000 0.06275487 0.51738544 0.48569051 0.41154735 1.7197910
order         2.0523706 0.00000000 0.04136448 0.75468800 0.25206980 0.13340692 0.7089330
performance   3.1350504 0.00000000 0.00000000 0.09613261 1.52511569 0.08999040 0.3012038
performed     1.4299143 0.00000000 0.11288703 0.60588585 0.00000000 0.23633648 0.7215927
practitioners 0.8842386 0.00000000 0.00000000 2.69573937 0.00000000 0.08307773 0.4532778
present       1.9906749 0.00000000 0.08785585 0.37000003 0.22587430 0.43654412 0.3533820
process       1.8858051 0.09987437 0.05287466 2.02559450 0.18497140 0.12481204 0.4942711
projects      0.5986353 0.00000000 0.07287010 1.50713311 0.00000000 0.07811658 3.2365694
proposed      3.1089604 0.00000000 0.12686456 0.85129973 0.99061832 0.19787981 0.7196690
provide       1.3953854 0.00000000 0.08088058 0.96942455 0.32261945 0.03702966 0.4397136
quality       1.6082846 0.04342843 0.16502805 1.22835967 0.08711614 0.39465115 3.8412207
research      0.5188792 0.00000000 0.28079172 4.03531261 0.20372791 0.20505162 0.5398301
results       2.7461297 0.02227617 0.16783346 1.11932450 0.59630326 0.27118688 0.9215596
several       1.1230865 0.00000000 0.08785585 1.24713914 0.04811154 0.23367834 0.7014684
source        0.5181468 0.00000000 0.08891421 0.34591737 0.20872182 0.08567867 3.5507196
studies       1.2341090 0.00000000 0.12911419 1.51754629 0.08484647 0.22273710 0.8913301
system        4.3079856 0.00000000 0.00000000 0.64530214 2.30955416 0.74199511 0.7765349
systems       1.5022191 0.09406069 0.00000000 0.54626462 0.23323211 3.85103740 0.5784976
tdscdma       1.2030773 0.00000000 0.00000000 0.00000000 7.47356721 0.00000000 0.0000000
time          4.0292925 0.00000000 0.10965935 1.32784179 0.33221883 0.17584666 0.7863228
tools         0.4327713 0.05730025 0.09493937 1.66804939 0.00000000 0.43320778 1.5036697
work          1.4074535 0.00000000 0.05790909 1.42584384 0.07469137 0.00000000 1.4076963
> |
```

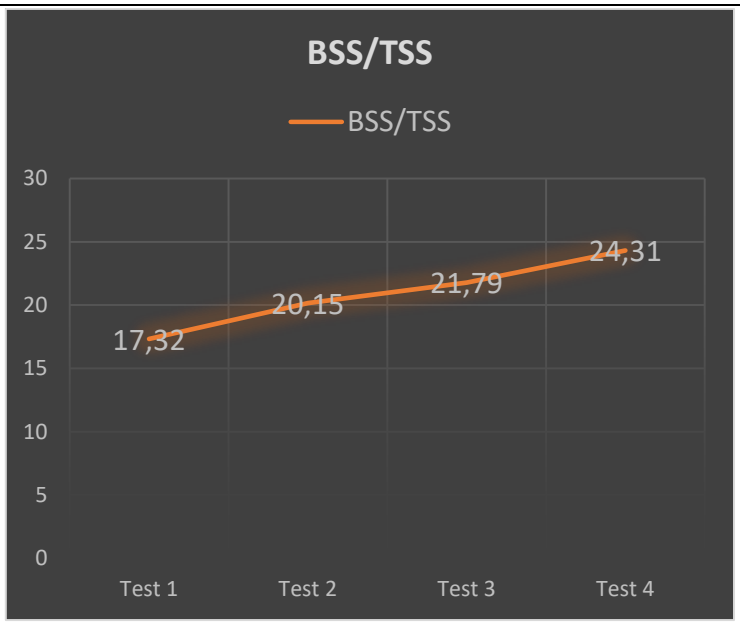
Με βάση τα ανωτέρω αποτελέσματα παρατηρούμε ότι οι συστάδες 1 και 5 έχουν σίγουρα μη σχετικά άρθρα και περιέχουν συνολικά και οι δύο μαζί 272+42 = 314 κείμενα περισσότερα από τα 259 που γνωρίζουμε ότι πρέπει να εντοπιστούν. Συνεπώς η ομαδοποίηση δεν είναι αποτελεσματική. Ακολουθούν οι δείκτες αξιολόγησης.

```
> #quality of partinioning
> bss<-KM$betweenss
> tss<-KM$totss
> qual<-(bss/tss)*100
> qual
[1] 24.31384
> #Dunn Index
> library(fpc)
> # Statistics for k-means clustering
> km_stats <- cluster.stats(dist(myData),  KM$cluster)
> # Dun index
> km_stats$dunn
[1] 0.0860899
```

Παρακάτω παρουσιάζουμε συνοπτικά τα αποτελέσματα των 4 δοκιμών μας σε πίνακα και με τα σχετικά γραφήματα.

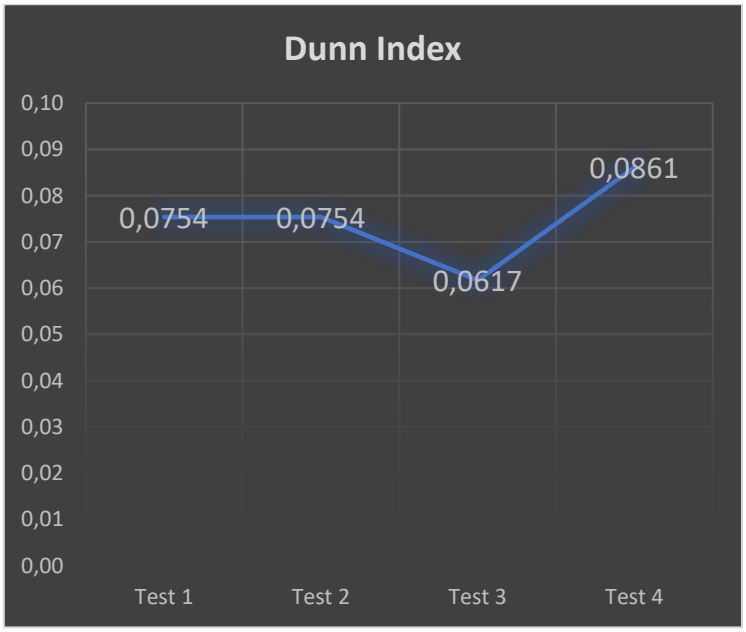
Αποτελέσματα	Clusters	BSS/TSS	Dunn Index
Test 1	4	17.32	0.0754
Test 2	5	20.15	0.0754
Test 3	6	21.79	0.0617
Test 4	7	24.31	0.0861

Πίνακας 17. Αποτελέσματα K-Means Clustering



Γράφημα 21. Δείκτης BSS/TSS

Από το γράφημα αλλά και από τον πίνακα συμπεραίνουμε ότι η μέθοδος tf-idf επιτυγχάνει καλύτερο διαχωρισμό των συστάδων και μάλιστα ο δείκτης βελτιώνεται με την αύξηση των συστάδων.



Γράφημα 22. Δείκτης Dunn

Ο δείκτης dunn για την $1^{η}$ και $2^{η}$ δοκιμή παραμένει σταθερός παρά την αύξηση των συστάδων γεγονός που σημαίνει ότι η ελάχιστη απόσταση μεταξύ διαφορετικών συστάδων

δεν αυξήθηκε ούτε η μέγιστη απόσταση σημείων της ίδιας συστάδας ελαττώθηκε. Στη 3η δοκιμή ο δείκτης μειώνεται ακόμα περισσότερο αλλά στη 4η δοκιμή παρουσιάζει τη μέγιστη τιμή του.

Με βάση τα αποτελέσματα των ανωτέρω δοκιμών συμπεραίνουμε ότι η μέθοδος k–means clustering δεν ήταν αποτελεσματική για τον διαχωρισμό των συστάδων μας σε αντίθεση με τη μέθοδο του topic modeling η οποία μας διαχώρισε τα άρθρα στις σωστές ομάδες.

5.7 K - means Clustering στο 2ο Dataset

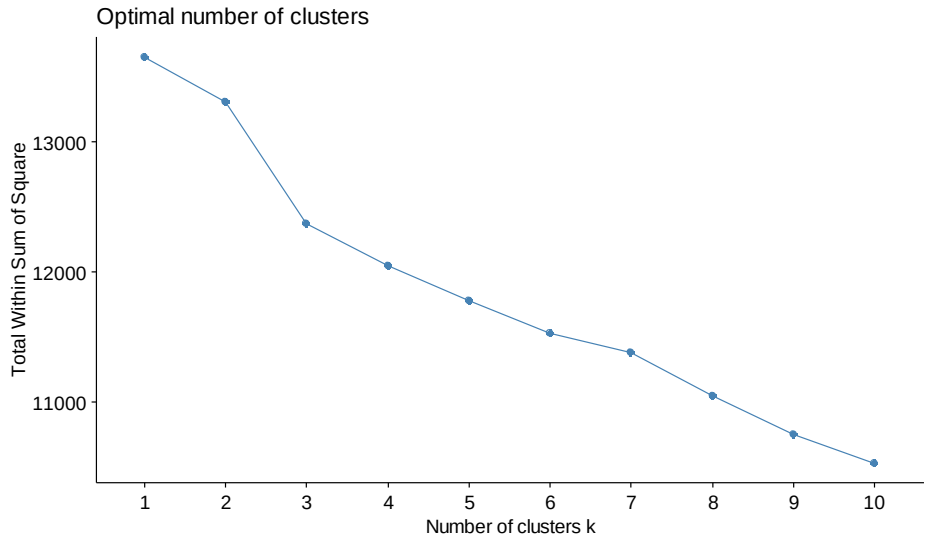
Προκειμένου να δημιουργήσουμε ένα μοντέλο κατηγοριοποίησης/ταξινόμησης, το οποίο θα μπορεί να χρησιμοποιήσει ένας ερευνητής ως εργαλείο για να διερευνήσει το συγκεκριμένο πεδίο, θα πρέπει προηγουμένως να έχουμε ομαδοποιήσει τα άρθρα μας τα εντοπισμένα με το τεχνικό χρέος (2ο dataset). Μια τέτοια ομαδοποίηση μπορεί να προκύψει με την τεχνική του k means clustering. Εφαρμόζουμε λοιπόν k means clustering για τα 305 άρθρα, που αφορούν το τεχνικό χρέος, και προέκυψαν από την τεχνική του topic modeling που ολοκληρώσαμε στην ενότητα 5.5.

Αρχικά πριν την εκτέλεση του κώδικα αφαιρούμε τις στήλες με μηδενικές τιμές που είχαν απομείνει στον πίνακα μας με το όνομα Corpus. Όταν αφαιρέσαμε τα άρθρα των topics 1, 2, 6 στον πίνακα document term matrix οι λέξεις (στήλες) που ανήκαν σε αυτά τα topics παρέμειναν στον πίνακα αλλά με μηδενικές τιμές διότι τα άρθρα που κρατήσαμε ανήκουν στα topic 3,4,5 και δεν περιέχουν αυτές τις λέξεις. Έτσι διαγράψαμε αυτές τις περιττές στήλες. Στη συνέχεια θα πρέπει να εντοπίσουμε τον κατάλληλο αριθμό των συστάδων για να ομαδοποιήσουμε τα δεδομένα μας.

Με τον ακόλουθο κώδικα δημιουργούμε τα γραφήματα που ακολουθούν και δημιουργούμε και τις συστάδες.

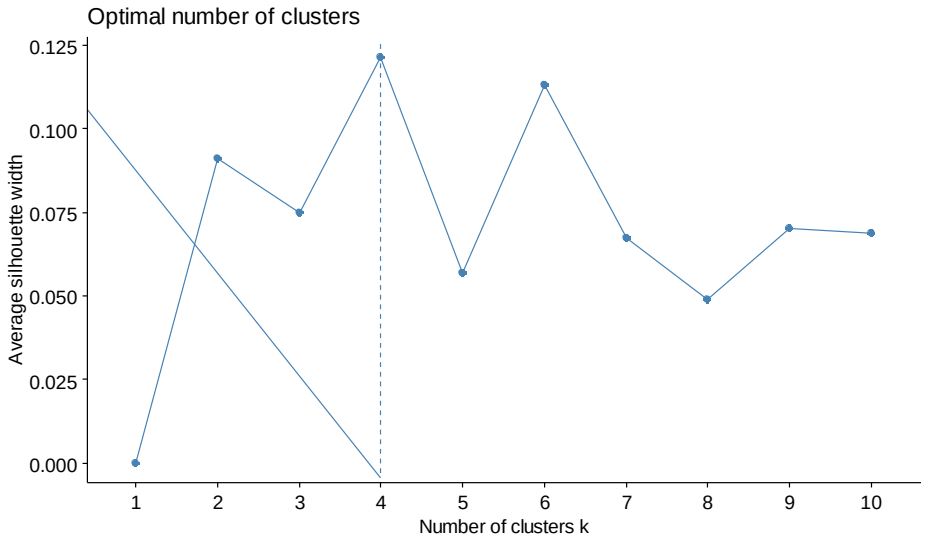
```
#-----------------K MEANS CLUSTERING--------------#
#remove comlumns with zero values
myData<-Corpus[,colSums(Corpus[])>0]

View(myData)
library(ggpubr)
library(factoextra)

fviz_nbclust(myData, kmeans, method = "wss")
fviz_nbclust(myData, kmeans, method = "silhouette")
fviz_nbclust(myData, kmeans, method = "gap_stat")
library(cluster)
set.seed(222)
KM=kmeans(myData,4, nstart=25) #nstart to change initial centers
#evaluating cluster analysis
autoplot(KM, data=myData, frame=TRUE)
```

Θα επιλέξουμε τον ιδανικό αριθμό συστάδων με βάση τις παρακάτω μεθόδους : Elbow, Silhouette και Gap Statistic.



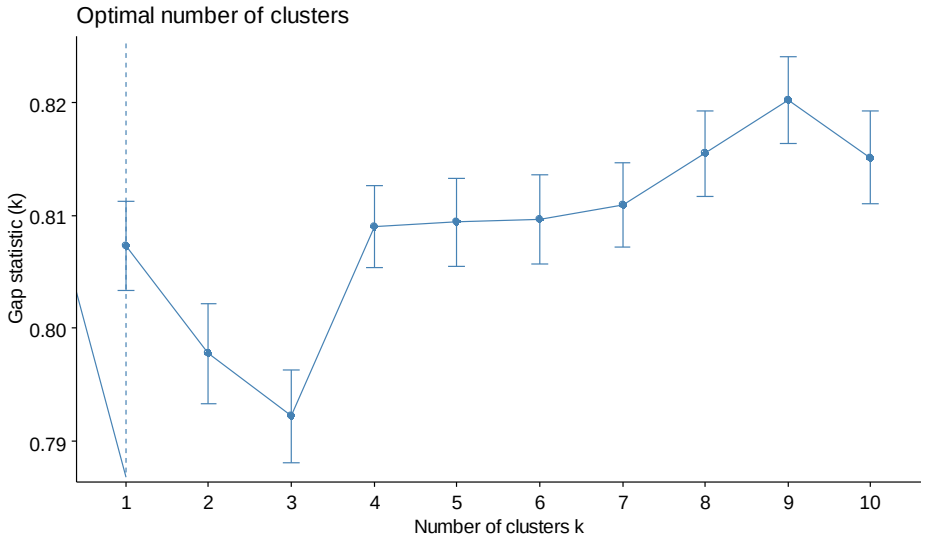
Γράφημα 23. Elbow Method για το 2ο Dataset

Ιδανικός αριθμός συστάδων με βάση τη μέθοδο αυτή είναι 3 ή μεγαλύτερο του 3.



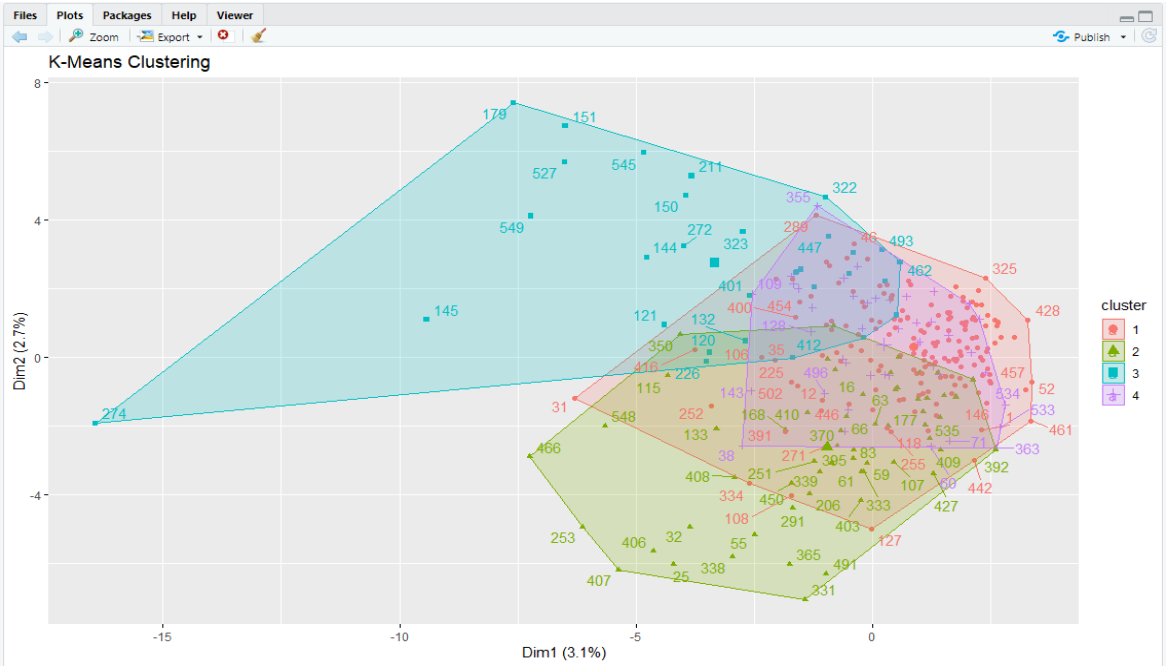
Γράφημα 24. Silhouete Method για το 2ο Dataset

Εδώ το γράφημα μας προτείνει 4 .



Γράφημα 25. Gap Statistic για το 2ο dataset

Με βάση το γράφημα μπορούμε να επιλέξουμε 4 ή περισσότερες συστάδες. Με βάση και τις τρεις αυτές μεθόδους επιλέγουμε 4 συστάδες. Μετά την εκτέλεση του αλγορίθμου συσταδοποίησης προκύπτει το παρακάτω γράφημα.



Γράφημα 26. K – Means Clustering στο 2ο Dataset

Το πλήθος των άρθρων ανά συστάδα φαίνεται παρακάτω:

```
> KM$size
[1] 173  58  31  43
>
```

Στη συνέχεια θα βρούμε το πλήθος των όρων ανά συστάδα με τον παρακάτω κώδικα

```
[1] 173  58  31  43
> cluster1<-myData[KM$cluster==1,]
> cluster2<-myData[KM$cluster==2,]
> cluster3<-myData[KM$cluster==3,]
> cluster4<-myData[KM$cluster==4,]
> c1<-colSums(cluster1)
> c2<-colSums(cluster2)
> c3<-colSums(cluster3)
> c4<-colSums(cluster4)
> allsums<-data.frame(c1,c2,c3,c4)
> allsums
                c1  c2  c3  c4
accumulation    20   8   0   7
activities      27  18   1   3
algorithm        5   0   1   0
amount          31   8   1   2
analysis        55  26  30  13
analyze         14   6   6   2
analyzed         9  10   3   4
application      6   5   1   7
applications    13   2   1   5
applied          6   7   3   4
architectural   40  28   6   2
aspects         21   5   2   6
assess          14   6   0   7
associated      16   4   5   2
automated       11  12  10   6
available       16   4   7   1
basis            3   3   1   3
become          11   3   1   4
benefits        24  17   4   6
calculations     1   0   0   0
characteristics 11   3   4   5
code            62  12 149  25
```

Μετά από επεξεργασία των ανωτέρω αποτελεσμάτων παρουσιάζουμε τις πιο συχνές λέξεις ανά συστάδα οι οποίες μας παρέχουν χρήσιμη πληροφορία για το περιεχόμενο της κάθε μιας. Ειδικότερα στον πίνακα 18 βλέπουμε τη συχνότητα εμφάνισης της κάθε λέξης μέσα στη συστάδα ενώ στον πίνακα 19 βλέπουμε το ποσοστό εμφάνισης της κάθε λέξης ανά άρθρο μέσα στη συστάδα. Δηλαδή έχουμε διαιρέσει τη συχνότητα εμφάνισης του όρου με το πλήθος των άρθρων της εκάστοτε συστάδας. Οι πιο συχνές λέξεις φαίνονται σκιασμένες στους παρακάτω πίνακες και αυτές μας δίνουν πληροφορία για τις συστάδες.

	Terms										
Cluster	**code**	**decisions**	**development**	**interest**	**items**	**management**	**quality**	**projects**	**results**	**time**	**Articles**
1	**62**	**66**	**79**	**66**	**27**	**75**	**30**	**67**	**87**	**55**	**173**
2	**12**	**3**	**156**	**21**	**40**	**98**	**20**	**26**	**56**	**43**	**58**
3	**149**	**1**	**15**	**6**	**22**	**2**	**16**	**39**	**35**	**17**	**31**
4	**25**	**4**	**27**	**16**	**2**	**12**	**124**	**15**	**27**	**10**	**43**

Πίνακας 18. Συχνότητα εμφάνισης λέξεων

	Terms										
Cluster	**code**	**decisions**	**development**	**interest**	**items**	**management**	**quality**	**projects**	**results**	**time**	**Articles**
1	**0,358**	**0,382**	**0,457**	**0,382**	**0,156**	**0,434**	**0,173**	**0,387**	**0,503**	**0,318**	**173**
2	**0,207**	**0,052**	**2,690**	**0,362**	**0,690**	**1,690**	**0,345**	**0,448**	**0,966**	**0,741**	**58**
3	**4,806**	**0,032**	**0,484**	**0,194**	**0,710**	**0,065**	**0,516**	**1,258**	**1,129**	**0,548**	**31**
4	**0,581**	**0,093**	**0,628**	**0,372**	**0,047**	**0,279**	**2,884**	**0,349**	**0,628**	**0,233**	**43**

Πίνακας 19. Ποσοστό εμφάνισης λέξεων

Με βάση τα αποτελέσματα αυτά συμπεραίνουμε ότι η 1η συστάδα, η οποία είναι αρκετά μεγαλύτερη συγκρινόμενη με τις υπόλοιπες, επικεντρώνεται στο κομμάτι της διαχείρισης του τεχνικού χρέους και σε κάποια κείμενα παρουσιάζονται αποτελέσματα από διάφορα projects. Η εμφάνιση της λέξης decision αφορά κυρίως τα projects στα οποία λαμβάνονται αποφάσεις οι οποίες οδηγούν, μεταξύ άλλων, στη συσσώρευση τεχνικού χρέους. Επίσης σε κάποια data points της συστάδας γίνεται αναφορά στη συντήρηση λογισμικού καθόσον υπάρχει ο όρος interest. Τέλος στη συστάδα αυτή παρατηρούμε ότι οι πιο πολλές λέξεις εκτός από τις items, quality και time έχουν μεγάλη συχνότητα εμφάνισης κάτι το οποίο θεωρείται αναμενόμενο διότι αυτή είναι κατά πολύ μεγαλύτερη από τις άλλες. Από αυτό καταλαβαίνουμε επίσης ότι η ομάδα αυτή ενδεχομένως συγκεντρώνει και κείμενα κάπως πιο γενικά στο ζήτημα του τεχνικού χρέους.

Η 2η συστάδα επίσης αφορά το κομμάτι της διαχείρισης, στη φάση της ανάπτυξης λογισμικού και περιλαμβάνει, μεταξύ άλλων, και τυχόν αποτελέσματα ενδεχομένως από projects ή από μελέτες ή τεχνικές που εφαρμόστηκαν. Τέλος γίνεται αναφορά και στο ζήτημα του *χρόνου* (time) μια λέξη που συνδέεται με το τεχνικό χρέος και συνήθως μεταφράζεται σε ανθρωπο-προσπάθεια.

Η 3η συστάδα ασχολείται περισσότερο με θέματα του κώδικα και πιθανόν αναφέρεται στο code debt των projects. Σε αυτό το συμπέρασμα μας οδηγεί και η λέξη items η οποία αναφέρεται στα TD items δηλαδή στα τμήματα του κώδικα στα οποία εμφανίζεται τεχνικό χρέος.

Τέλος η 4η συστάδα αναφέρεται κυρίως σε θέματα ποιότητας του κώδικα. Δηλαδή ασχολείται ίσως εκτενέστερα με τις μετρικές ποιότητας κώδικα.

Σε σχέση με τη δική μας εμπειρική ομαδοποίηση μπορούμε να πούμε ότι και εμείς έχουμε μια μεγάλη κατηγορία που περιέχει άρθρα που αναφέρονται σε γενικά θέματα του τεχνικού χρέους όπως έχουμε και εδώ. Στις υπόλοιπες δικές μας ομάδες έχουμε διαχωρίσει τα άρθρα που ασχολούνται με το ζήτημα της συντήρησης κάτι που δεν ισχύει στη τρέχουσα συσταδοποίηση. Αυτό το διαπιστώνουμε από τις τιμές του όρου interest και στις 4 κλάσεις. Δηλαδή παρόλο που στην 1η κλάση παρατηρείται η μεγαλύτερη συχνότητα εμφάνισης του όρου, από τον πίνακα 19 διαπιστώνουμε ότι στις κλάσεις 1, 2 και 4 ο όρος εμφανίζεται

σχεδόν με το ίδιο ποσοστό, που σημαίνει ότι τα άρθρα μοιράζονται και στις 3 αυτές κλάσεις. Επίσης, όπως έχουμε προαναφέρει, εμείς έχουμε μια ξεχωριστή ομάδα για τα κείμενα που αφορούν τη βιβλιογραφική ανασκόπηση κάτι που δεν συμβαίνει με το clustering. Στον ακόλουθο πίνακα συνοψίζουμε τα αποτελέσματα της συταδοποίησης.

Συστάδα	Πλήθος άρθρων	Επεξήγηση
1	**178**	**Είναι η μεγαλύτερη συστάδα και περιλαμβάνει γενικά θέματα, θέματα διαχείρισης και συντήρησης λογισμικού**
2	**58**	**Επικεντρώνεται σε θέματα διαχείρισης, συντήρησης λογισμικού και σε σχέση με τα αποτελέσματα από διάφορα projects ανάπτυξης λογισμικού**
3	**31**	**Επικεντρώνεται στο χρέος κώδικα - code debt από διάφορα projects**
4	**43**	**Αναφέρεται στις μετρικές ποιότητας κώδικα και στη συντήρηση λογισμικού**

Πίνακας 20. K-Means Clustering 2ο dataset

Στη συνέχεια υπολογίζουμε τις μετρικές της συσταδοποίησης.

Μετρικές της συσταδοποίησης

Υπολογίζουμε το πηλίκο bss/tss παρακάτω και παρατηρούμε ότι η τιμή του είναι μικρή.

```
> #quality o partinioning
> bss<-KM$betweenss
> tss<-KM$totss
> qual<-(bss/tss)*100
> qual #the higher the percentage the better the score
[1] 11.73899
```

Παρακάτω υπολογίζουμε τον δείκτη dunn ο οποίος είναι χαμηλός αλλά εμφανώς υψηλότερος από αυτόν της ενότητας 5.6

```
> #Dunn Index as max as possible
> library(fpc)
> # Statistics for k-means clustering
> km_stats <- cluster.stats(dist(myData),  KM$cluster)
> # Dun index
> km_stats$dunn
[1] 0.1803828
```

Στη συνέχεια προσθέτουμε στα δεδομένα μας σε μια επιπλέον στήλη την συστάδα στην οποία ανήκει κάθε άρθρο όπως φαίνεται παρακάτω.

```
#-------------------------------------------------------------------------------------------#
#------Create new data frame with category to myData adding a column with cluster category----#
mydata<-data.frame(myData, KM$cluster)
View(mydata)
#rename column KM.cluster
names(mydata)[179]="cg"
view(mydata)
```

Έτσι έχουμε προετοιμάσει το σύνολο δεδομένων με ετικέτες κατηγορίας και το χωρίζουμε σε σύνολο εκπαίδευσης και σύνολο δοκιμών προκειμένου να εφαρμόσουμε τα μοντέλα ταξινόμησης που έχουμε επιλέξει και παρουσιάζουμε παρακάτω στην ενότητα 5.10.

5.8 Hierarchical Clustering στο 2ο Dataset

Μια άλλη δοκιμή που θα κάνουμε είναι να εφαρμόσουμε την ιεραρχική συσταδοποίηση για να δούμε τις ομάδες που δημιουργούνται με αυτή τη μέθοδο και να διερευνήσουμε το περιεχόμενό τους δηλαδή να κατανοήσουμε πως διαχωρίζονται μεταξύ τους και ποιες κατηγορίες εντοπίζει ο αλγόριθμος. Πριν παρουσιάσουμε τα αποτελέσματά μας, είναι σημαντικό να αναφέρουμε ότι κάναμε δοκιμές με τις εξής μεθόδους ιεραρχικής συσταδοποίησης single, complete, average, ward.D και ward.D2. Επίσης εφαρμόσαμε και διαιρετική συσταδοποίηση. Μεταξύ όλων αυτών, η μέθοδος με τα καλύτερα αποτελέσματα είναι η ward.D2 την οποία παρουσιάζουμε παρακάτω:

Δημιουργούμε με τη βοήθεια αλγορίθμου δύο πίνακες και δύο ειδών δενδρογράμματα με απολήξεις:

- τα άρθρα ώστε να εντοπίσουμε οπτικά ποια ομαδοποιούνται μεταξύ τους
- τους όρους (terms) του πίνακα term document matrix ώστε να εντοπίσουμε το σύνολο των λέξεων ανά συστάδα

Αρχικά δημιουργούμε τον πίνακα document term matrix με τον ακόλουθο κώδικα.

```
1  # HIERARCHICAL CLUSTERING ON CSV
2  library(tm)
3  library(stats)
4  library(dplyr)
5  library(ggplot2)
6  library(Snowballc)
7  library(NbClust)
8  library(stringi)
9  library(tidyverse)  # data manipulation
10 library(cluster)    # clustering algorithms
11 library(factoextra) # clustering visualization
12 library(dendextend) # for comparing two dendrograms
13
14 setwd("~/R/PS1")
15 #read data
16 r<-read.csv("/Users/User/Documents/R/PS1/PS1.csv")
17 rdt<-as.data.frame(r)
18 rdt2<-subset(rdt, select = - c(BibliographyType,ISBN,Author, Journal,Volume, Number, Month, Pages,Note, URL,Address, B
19 names(rdt2)[4]="Abstract"
20 View(rdt2)
21 paper.abstract <- VCorpus(VectorSource(rdt2$Abstract))
22 inspect(paper.abstract)
23 toSpace <- content_transformer(function(x,pattern) gsub(pattern, " ", x))
24 corpu <- tm_map(paper.abstract, toSpace, "/|@|\\|")
25 corpu<-tm_map(corpu,content_transformer(tolower))
26 removeNumPunct <- function(x) gsub("[^[:alpha:][:space:]]*", "", x) #remove anything other than English letters or spa
27 corpu<-tm_map(corpu, content_transformer(removeNumPunct))
28 corpu<-tm_map(corpu,removeWords,stopwords("english"))
29 removeUnicode <- function(x) stri_replace_all_regex(x,"[^\x20-\x7E]","")
30 corpu <- tm_map(corpu, content_transformer(removeUnicode))
31 #remove extra words
32 corpu<-tm_map(corpu,removeWords, c("engineering","systems","architecture","existing","business","project","approaches"
33 corpu<-tm_map(corpu,removePunctuation, UCP=TRUE)
34 corpu<-tm_map(corpu,removeNumbers)
35 corpu<-tm_map(corpu,stripWhitespace) #remove extra whitespace
36 dtm<-DocumentTermMatrix(corpu, control = list(stropwords=T,weighting=weightTfIdf,minWordLength=c(4,15), bounds = list(
37 dtm<-removeSparseTerms(x = dtm, sparse = 0.9)
38 inspect(dtm)
```

Διαβάζουμε το σχετικό αρχείο (γραμμή 16), μειώνουμε τον αριθμό των στηλών διότι δεν μας χρειάζονται όλες (γραμμή 18) και κρατάμε στο αντικείμενο paper.abstract τις περιλήψεις (γραμμή 21). Στη συνέχεια στις γραμμές 23-35 αφαιρούμε χαρακτήρες, σύμβολα, κενά, σημεία στίξης, αγγλικές stopwords και επιπλέον λέξεις stopwords που έχουμε δηλώσει εμείς (γραμμή 32). Δημιουργούμε τον πίνακα document term matrix (γραμμή 36) και θέτουμε το sparse του πίνακα ίσο με 0,9 (γραμμή 37).

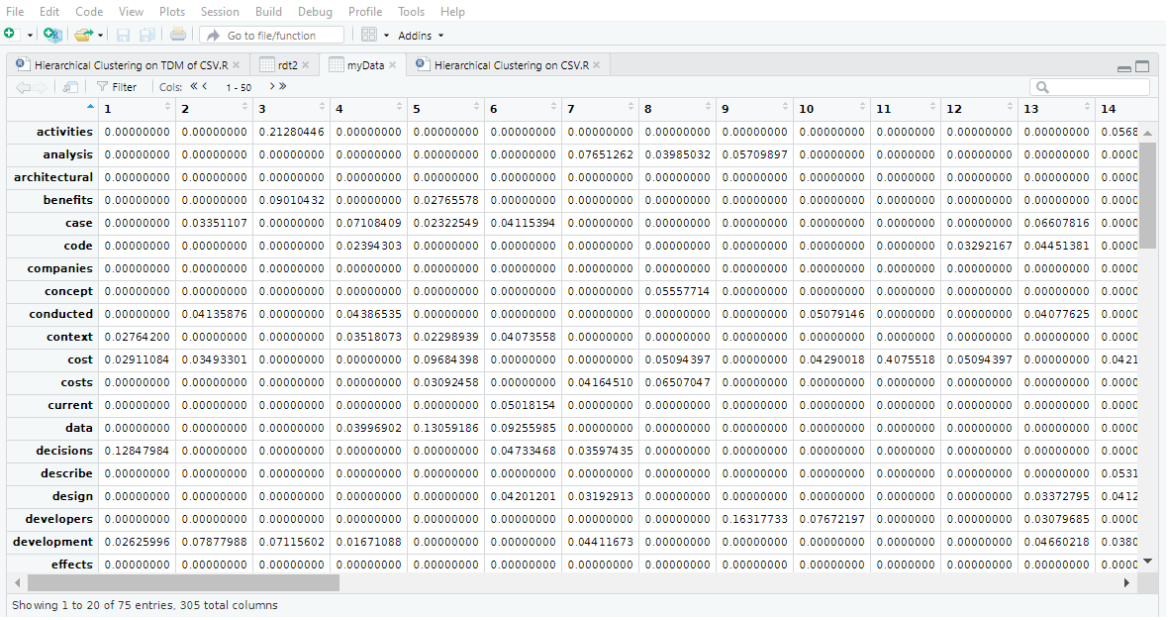
information	interest	investigate	issues	items	knowledge	level	longterm	maintenance	manage	management
0.00000000	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000	0.03248933	0.03005649	0.00000000	0.03819371	0.00000000
0.00000000	0.00000000	0.00000000	0.03811004	0.00000000	0.00000000	0.00000000	0.03606779	0.00000000	0.00000000	0.00000000
0.00000000	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000	0.08144339	0.00000000	0.00000000	0.00000000
0.00000000	0.00000000	0.00000000	0.08083948	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000
0.00000000	0.04999495	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000	0.02499748	0.00000000	0.00000000	0.00000000
0.00000000	0.00000000	0.00000000	0.00000000	0.05205725	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000
0.00000000	0.00000000	0.00000000	0.00000000	0.23738106	0.04397959	0.00000000	0.03366327	0.00000000	0.00000000	0.00000000
0.00000000	0.05259885	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000
0.00000000	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000
0.00000000	0.00000000	0.00000000	0.00000000	0.15617175	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000
0.00000000	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000
0.00000000	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000	0.05259885	0.00000000	0.00000000	0.00000000
0.00000000	0.00000000	0.00000000	0.03757328	0.00000000	0.04645731	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000
0.05608044	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000
0.19133326	0.00000000	0.00000000	0.00000000	0.00000000	0.19402760	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000
0.03127563	0.02427639	0.02818017	0.00000000	0.00000000	0.00000000	0.02624138	0.00000000	0.02144517	0.00000000	0.06531064
0.00000000	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000	0.09166489	0.00000000
0.00000000	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000

Showing 1 to 18 of 305 entries, 75 total columns

Πίνακας 21. Document Term Matrix - 2ο Dataset

Στη συνέχεια δημιουργούμε και τον πίνακα term document matrix με τον εξής κώδικα

```
tdm<-TermDocumentMatrix(corpu, control = list(stropwords=T,weighting=weightTfIdf,minWordLength=c(4,15)
tdm<-removeSparseTerms(x = tdm, sparse = 0.9)
```



	1	2	3	4	5	6	7	8	9	10	11	12	13	14
activities	0.00000000	0.00000000	0.21280446	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000	0.0000000	0.00000000	0.00000000	0.0568
analysis	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000	0.07651262	0.03985032	0.05709897	0.00000000	0.0000000	0.00000000	0.00000000	0.0000
architectural	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000	0.0000000	0.00000000	0.00000000	0.0000
benefits	0.00000000	0.00000000	0.09010432	0.00000000	0.02765578	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000	0.0000000	0.00000000	0.00000000	0.0000
case	0.00000000	0.03351107	0.00000000	0.07108409	0.02322549	0.04115394	0.00000000	0.00000000	0.00000000	0.00000000	0.0000000	0.00000000	0.06607816	0.0000
code	0.00000000	0.00000000	0.00000000	0.02394303	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000	0.0000000	0.03292167	0.04451381	0.0000
companies	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000	0.0000000	0.00000000	0.00000000	0.0000
concept	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000	0.05557714	0.00000000	0.00000000	0.0000000	0.00000000	0.00000000	0.0000
conducted	0.00000000	0.04135876	0.00000000	0.04386535	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000	0.05079146	0.0000000	0.00000000	0.04077625	0.0000
context	0.02764200	0.00000000	0.00000000	0.03518073	0.02298939	0.04073558	0.00000000	0.00000000	0.00000000	0.00000000	0.0000000	0.00000000	0.00000000	0.0000
cost	0.02911084	0.03493301	0.00000000	0.00000000	0.09684398	0.00000000	0.00000000	0.05094397	0.00000000	0.04290018	0.4075518	0.05094397	0.00000000	0.0421
costs	0.00000000	0.00000000	0.00000000	0.00000000	0.03092458	0.00000000	0.04164510	0.06507047	0.00000000	0.00000000	0.0000000	0.00000000	0.00000000	0.0000
current	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000	0.05018154	0.00000000	0.00000000	0.00000000	0.00000000	0.0000000	0.00000000	0.00000000	0.0000
data	0.00000000	0.00000000	0.00000000	0.03996902	0.13059186	0.09255985	0.00000000	0.00000000	0.00000000	0.00000000	0.0000000	0.00000000	0.00000000	0.0000
decisions	0.12847984	0.00000000	0.00000000	0.00000000	0.00000000	0.04733468	0.03597435	0.00000000	0.00000000	0.00000000	0.0000000	0.00000000	0.00000000	0.0000
describe	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000	0.0000000	0.00000000	0.00000000	0.0531
design	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000	0.04201201	0.03192913	0.00000000	0.00000000	0.00000000	0.0000000	0.00000000	0.03372795	0.0412
developers	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000	0.16317733	0.07672197	0.0000000	0.00000000	0.03079685	0.0000
development	0.02625996	0.07877988	0.07115602	0.01671088	0.00000000	0.00000000	0.04411673	0.00000000	0.00000000	0.00000000	0.0000000	0.00000000	0.04660218	0.0380
effects	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000	0.0000000	0.00000000	0.00000000	0.0000

Showing 1 to 20 of 75 entries, 305 total columns

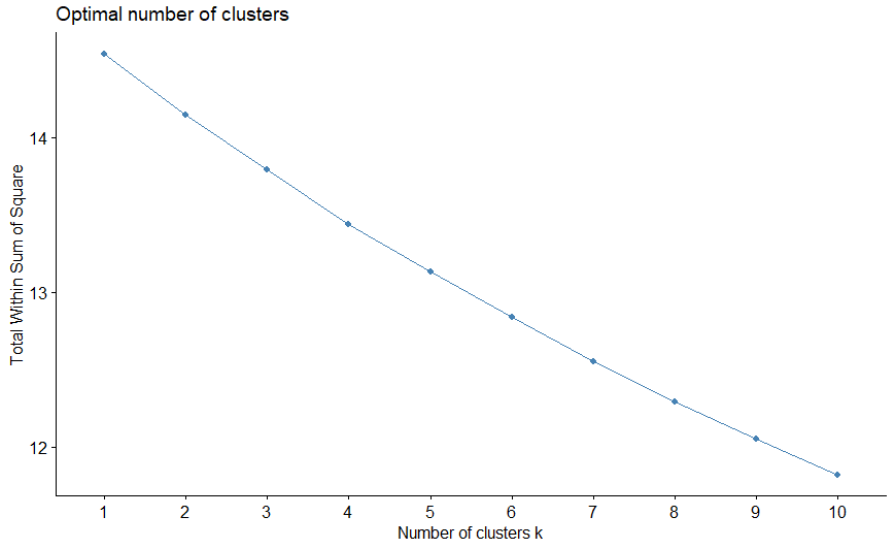
Πίνακας 22. Term Document Matrix - 2ο Dataset

και χρησιμοποιούμε ως κριτήριο ομοιότητας την ευκλείδεια απόσταση.

```
# Compute distance matrix
m<-as.matrix(myData)
distMatrix <- dist(m, method="euclidean")
```

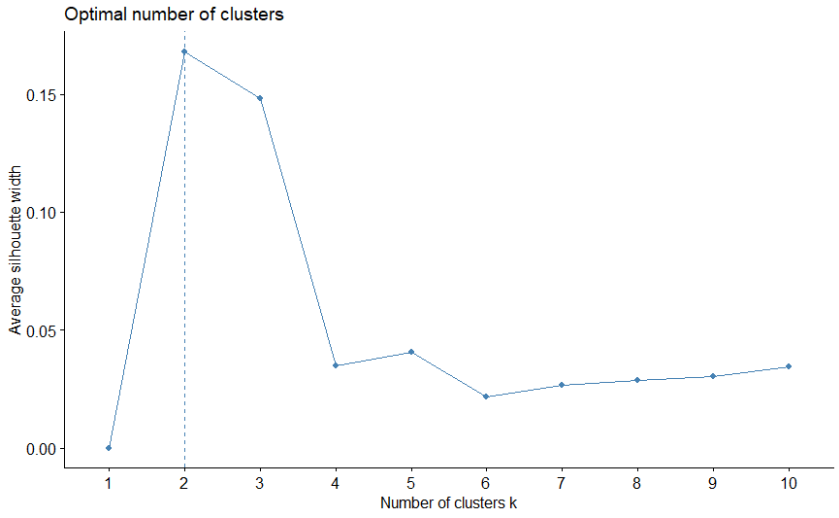
Πριν ξεκινήσουμε τη συσταδοποίηση θα προσδιορίσουμε τον αριθμό των συστάδων με τις μεθόδους των 3 γραφημάτων.

```
#-------------------------------------------------#
#Optimal number of clusters
fviz_nbclust(myData, FUN = hcut, method = "wss")
fviz_nbclust(myData, FUN = hcut, method = "silhouette")
gap_stat <- clusGap(myData, FUN = hcut, nstart = 25, K.max = 10, B = 50)
fviz_gap_stat(gap_stat)
#-------------------------------------------------#
```



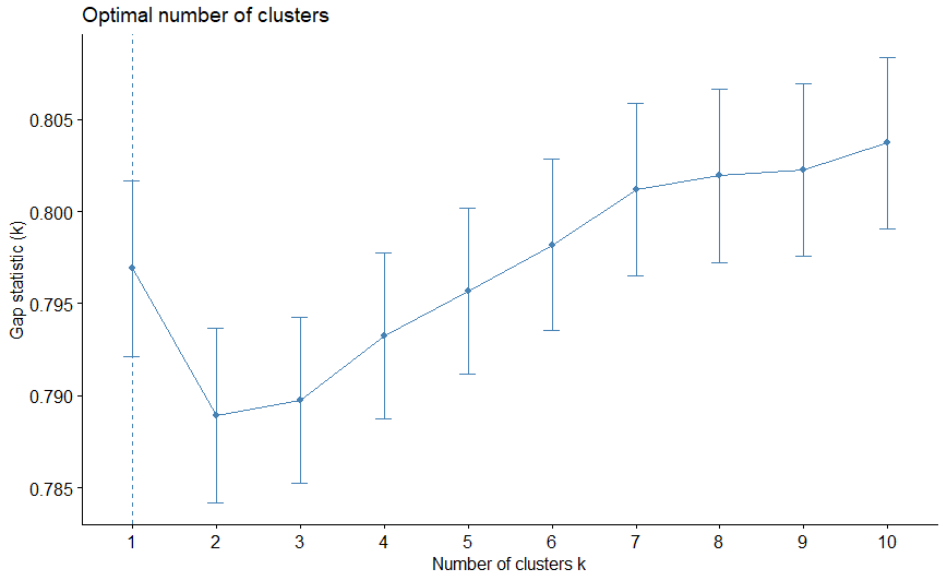
Γράφημα 27. Elbow Method στο 2ο dataset (για hierarchical clustering)

Δυστυχώς το γράφημα δεν μας βοηθάει στην επιλογή μας καθόσον δεν υπάρχει η εμφάνιση «αγκώνα».



Γράφημα 28. Silhouete Method στο 2ο dataset (για hierarchical clustering)

Εδώ η μέθοδος μας προτείνει 2 συστάδες. Επόμενη καλύτερη τιμή θεωρείται κάποια από αμέσως επόμενες μεγαλύτερες.



Γράφημα 29. Gap statistic Method στο 2ο dataset (για hierarchical clustering)

Η μέθοδος αυτή δείχνει ότι όσο περισσότερες είναι οι συστάδες τόσο αυξάνεται το gap statistic. Για να επιλέξουμε θα πρέπει να συνδυάσουμε τις παραπάνω επιλογές. Στη πράξη κάναμε δοκιμές με 4, 5 και 6 συστάδες διότι τα γραφήματα στην εξεταζόμενη περίπτωση δεν ήταν ιδιαίτερα κατατοπιστικά και τελικά καταλήξαμε στις 6 συστάδες.

Στη συνέχεια εφαρμόζουμε συσσωρευτική συσταδοποίηση και συγκεκριμένα τη μέθοδο ward.D2 η οποία μας έδωσε καλύτερα αποτελέσματα από τις άλλες μεθόδους.

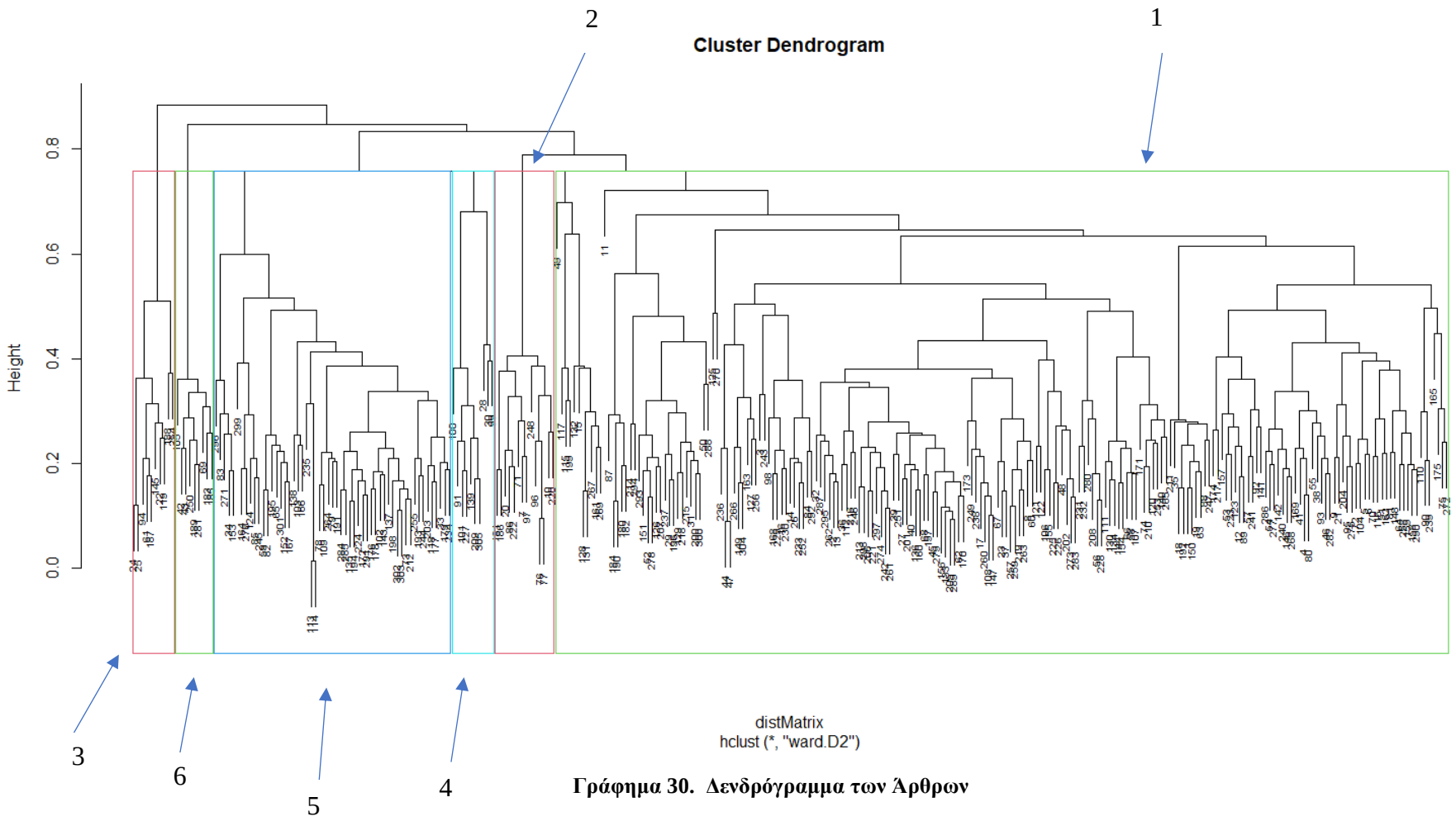
Παρουσιάζουμε τον κώδικα δημιουργίας των 6 συστάδων της μεθόδου wardD2 και με κριτήριο την ευκλείδια απόσταση.

```
#-----------------------------------------------#
# AGGLOMERATIVE CLUSTERING
# METHOD WARD D2
hc_wardD2 <- hclust(distMatrix, method = "ward.D2" )
plot(hc_wardD2, cex = 0.6)
rect.hclust(hc_wardD2, k = 6, border = 2:5)
```

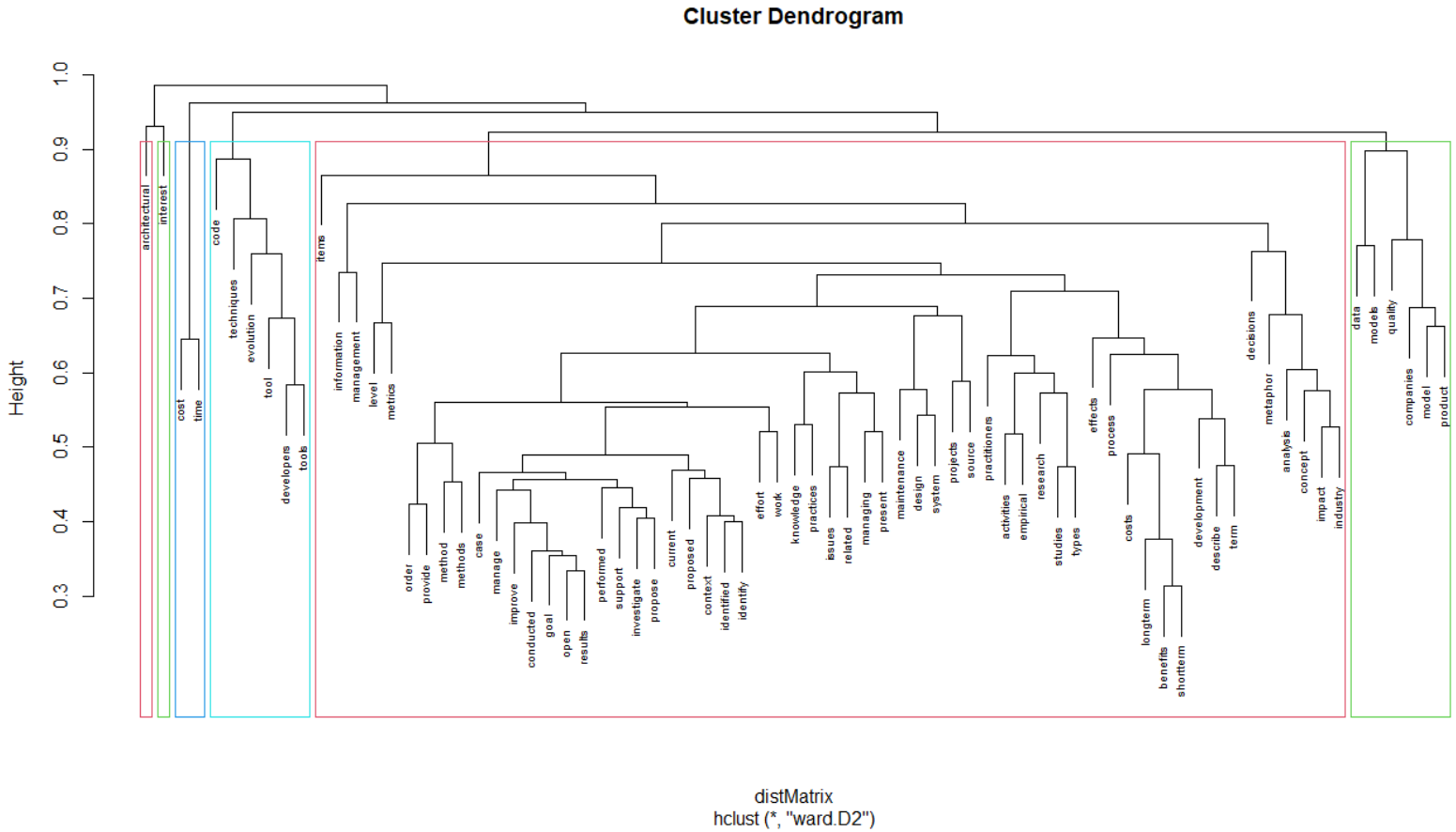
Όταν ο πίνακας του distMatrix έχει όρισμα, στον παρακάτω κώδικα, τον document term matrix τότε το δενδρόγραμμα θα έχει φύλλα τα άρθρα (αύξοντα αριθμό) ενώ όταν έχει όρισμα τον term document matrix τότε το δενδρόγραμμα έχει ως φύλλα τους όρους/λέξεις.

```
# Compute distance matrix
m<-as.matrix(myData)
distMatrix <- dist(m, method="euclidean")
```

Στη συνέχεια ακολουθούν τα αντίστοιχα δενδρογράμματα, δηλαδή ακολουθεί το δενδρόγραμμα με φύλλα τον αύξοντα αριθμό των άρθρων και το δενδρόγραμμα με τους όρους του term document matrix. Στο δενδρόγραμμα με φύλλα τους αριθμούς των άρθρων φαίνεται και ο αύξοντας αριθμός της κάθε συστάδας.



Γράφημα 30. Δενδρόγραμμα των Άρθρων



Γράφημα 31. Δενδρόγραμμα των terms ανά cluster

Από το 2ο δενδρόγραμμα μπορούμε να δούμε τους όρους ανά συστάδα και να κατανοήσουμε την ομαδοποίηση που εκτελεί ο αλγόριθμος. Στον πίνακα που ακολουθεί παρουσιάζουμε τα αποτελέσματα μας τα οποία επιβεβαιώθηκαν μετά από έλεγχο του πίνακα myData.

Συστάδα	Πλήθος άρθρων	Επεξήγηση
1	**207**	**Είναι η μεγαλύτερη συστάδα και περιλαμβάνει θέματα διαχείρισης χρέους, αιτιών, επιπτώσεων, ορολογία, ορισμοί κλπ.**
2	**14**	**Επικεντρώνεται σε θέματα τεχνικού χρέους και ποιότητας του κώδικα**
3	**10**	**Επικεντρώνεται στη συντήρηση λογισμικού**
4	**10**	**Αναφέρεται στη σχέση μεταξύ τεχνικού χρέους και τεχνικών/ εργαλείων ανάπτυξης κώδικα**
5	**55**	**Είναι η επόμενη μεγαλύτερη συστάδα και ασχολείται με το κόστος του τεχνικού χρέους και το χρόνο (πχ ανθρωπο-προσπάθεια)**
6	**9**	**Αναφέρεται στο αρχιτεκτονικό χρέος**

Πίνακας 23. Αποτελέσματα Ιεραρχικής Συσταδοποίησης (ward.D2)

Η κατανομή των άρθρων ανά συστάδα φαίνεται αμέσως παρακάτω:

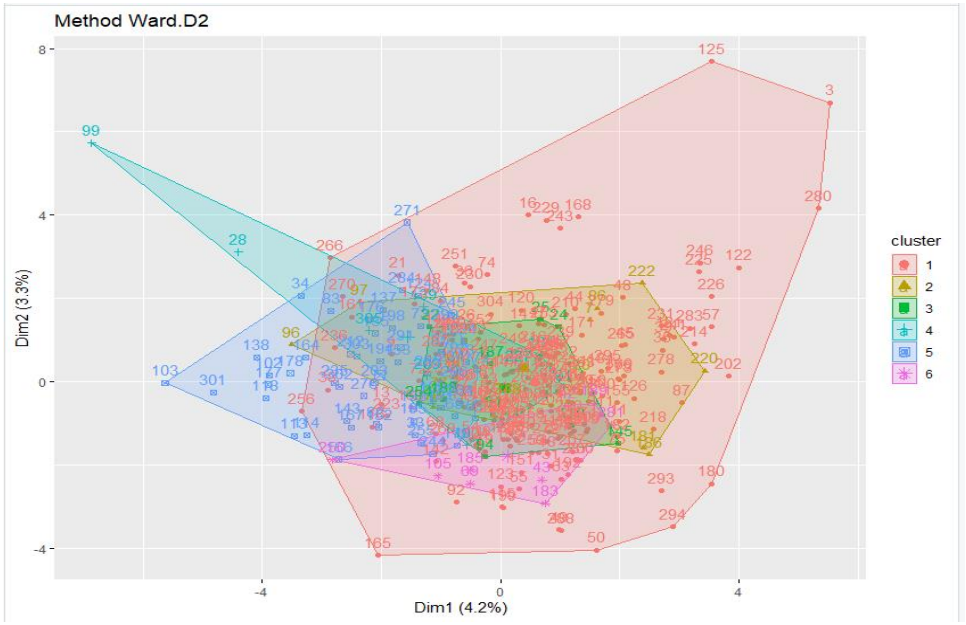
```
sub_wardD2
  1   2   3   4   5   6
207  14  10  10  55   9
>
```

Το πλήθος των όρων ανά συστάδα είναι ως ακολούθως:

```
sub_wardD2
 1  2  3  4  5  6
59  1  6  6  2  1
>
```

Με βάση τον πίνακα μπορούμε να ανατρέξουμε στη συστάδα που μας ενδιαφέρει και διαβάζοντας το δενδρόγραμμα των άρθρων να εντοπίσουμε ποια είναι. Για παράδειγμα

παρατηρούμε ότι με βάση τα αποτελέσματα του ανωτέρω πίνακα εντοπίζονται 10 άρθρα που αφορούν το θέμα της συντήρησης λογισμικού. Από το δενδρόγραμμα, στο γράφημα 30, αυτά είναι τα ακόλουθα : 22, 24, 25, 61, 94, 119, 145, 187, 188, 254. Συνεπώς με τη μέθοδο αυτή μπορεί κάποιος ερευνητής να εντοπίσει και να κατεβάσει τα προαναφερόμενα άρθρα που αφορούν τη συντήρηση λογισμικού. Με ανάλογο τρόπο μπορεί να ανατρέξει αρχικά στο δενδρόγραμμα των όρων και να βρει την ομάδα που τον ενδιαφέρει. Στη συνέχεια θα πρέπει να διαβάσει το δενδρόγραμμα των άρθρων και να εντοπίσει τα άρθρα που ανήκουν στην ομάδα που τον ενδιαφέρει.



Γράφημα 32. Clusters of ward.D2 Method

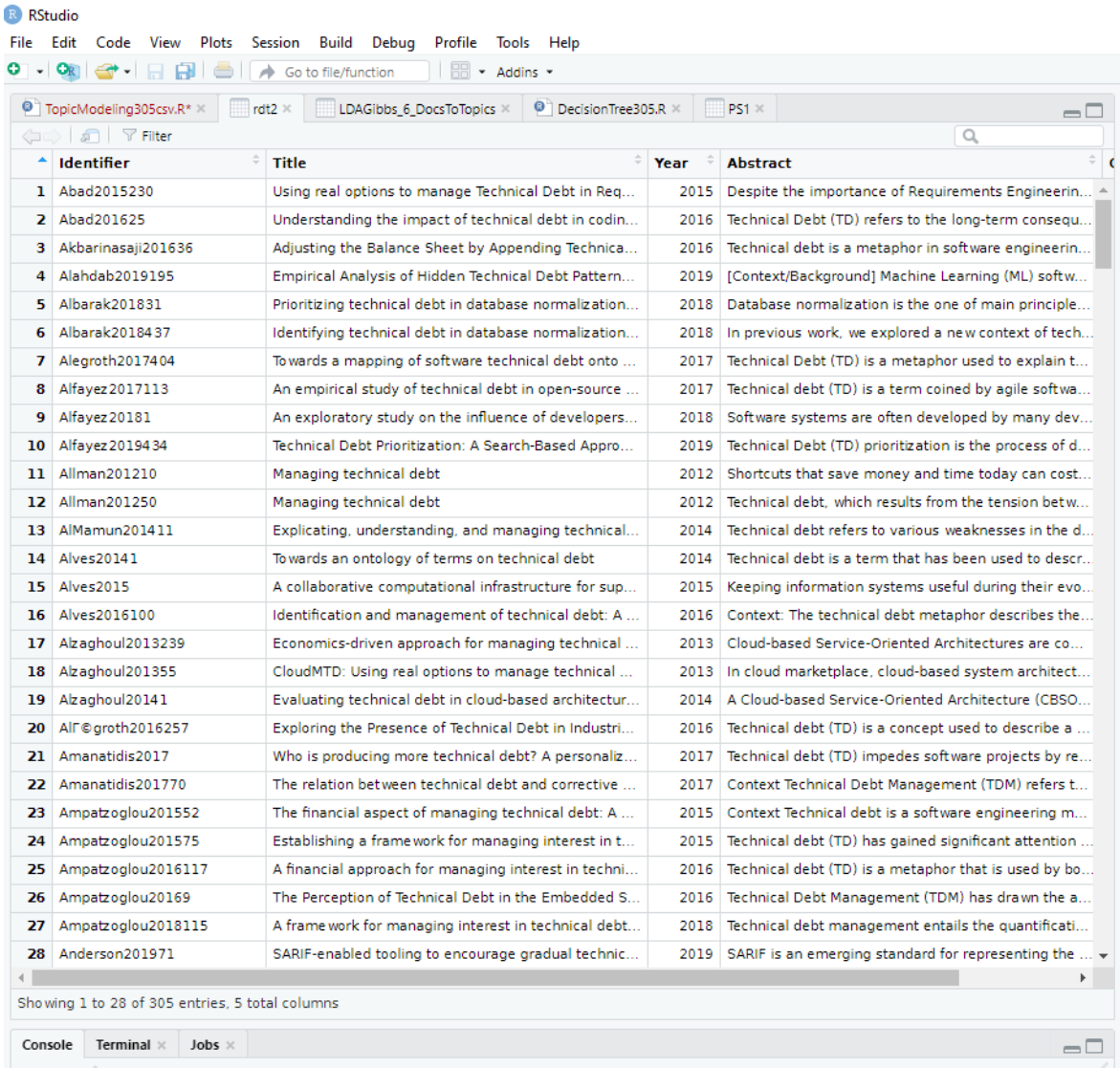
Στο παραπάνω γράφημα βλέπουμε τις 6 συστάδες που δημιούργησε η μέθοδος ward.D2. Η συστάδα που αναφέραμε για τη συντήρηση λογισμικού είναι αυτή με τον αριθμό 3 (έντονο πράσινο χρώμα).

5.9 Topic Modeling στο 2ο dataset και αποτελέσματα των μοντέλων ταξινόμησης με ετικέτες κατηγορίας τα topics

Στη δοκιμή αυτή θα επιχειρήσουμε να ομαδοποιήσουμε τα δεδομένα μας με τη μέθοδο του Topic Modeling, να δούμε πως ομαδοποιεί ο συγκεκριμένος αλγόριθμος και στη συνέχεια

να συγκρίνουμε τα αποτελέσματα με τα δικά μας εμπειρικά αποτελέσματα. Επίσης θα αξιοποιήσουμε τα topics ως ετικέτες κατηγορίας και θα εκτελέσουμε 4 αλγορίθμους ταξινόμησης ώστε να ταξινομήσουμε το 2ο dataset με βάση αυτές τις κατηγορίες. Κατά βάση ο σκοπός μας είναι να δημιουργήσουμε ένα ή/και περισσότερα μοντέλα ταξινόμησης ικανοποιητικής ακρίβειας αξιοποιώντας τις ετικέτες κατηγορίας που θα μας προσδώσει η τεχνική του topic modeling.

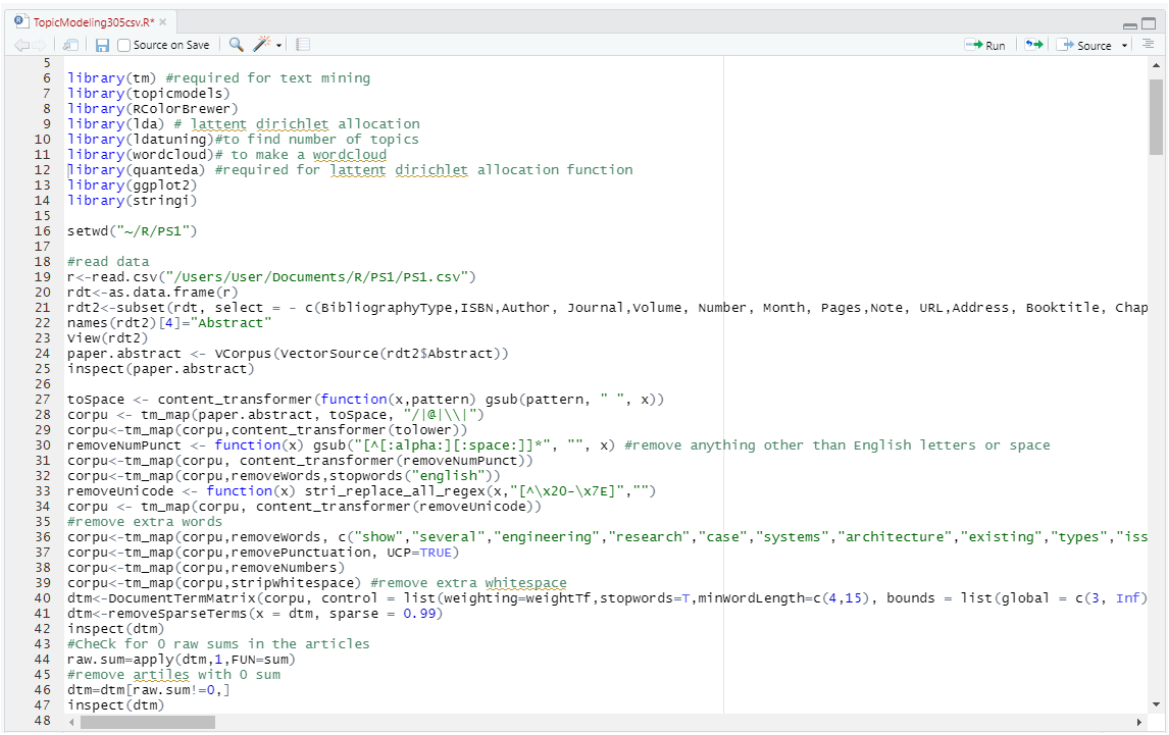
Αρχικά εισάγουμε το 2ο dataset στο R-studio και η στήλη που μας ενδιαφέρει είναι η στήλη Abstract όπως φαίνεται στην παρακάτω οθόνη του R-studio.



	Identifier	Title	Year	Abstract
1	Abad2015230	Using real options to manage Technical Debt in Req...	2015	Despite the importance of Requirements Engineerin...
2	Abad201625	Understanding the impact of technical debt in codin...	2016	Technical Debt (TD) refers to the long-term consequ...
3	Akbarinasaji201636	Adjusting the Balance Sheet by Appending Technica...	2016	Technical debt is a metaphor in software engineerin...
4	Alahdab2019195	Empirical Analysis of Hidden Technical Debt Pattern...	2019	[Context/Background] Machine Learning (ML) softw...
5	Albarak201831	Prioritizing technical debt in database normalization...	2018	Database normalization is the one of main principle...
6	Albarak2018437	Identifying technical debt in database normalization...	2018	In previous work, we explored a new context of tech...
7	Alegroth2017404	Towards a mapping of software technical debt onto ...	2017	Technical Debt (TD) is a metaphor used to explain t...
8	Alfayez2017113	An empirical study of technical debt in open-source ...	2017	Technical debt (TD) is a term coined by agile softwa...
9	Alfayez20181	An exploratory study on the influence of developers...	2018	Software systems are often developed by many dev...
10	Alfayez2019434	Technical Debt Prioritization: A Search-Based Appro...	2019	Technical Debt (TD) prioritization is the process of d...
11	Allman201210	Managing technical debt	2012	Shortcuts that save money and time today can cost...
12	Allman201250	Managing technical debt	2012	Technical debt, which results from the tension betw...
13	AlMamun201411	Explicating, understanding, and managing technical...	2014	Technical debt refers to various weaknesses in the d...
14	Alves20141	Towards an ontology of terms on technical debt	2014	Technical debt is a term that has been used to descr...
15	Alves2015	A collaborative computational infrastructure for sup...	2015	Keeping information systems useful during their evo...
16	Alves2016100	Identification and management of technical debt: A ...	2016	Context: The technical debt metaphor describes the...
17	Alzaghoul2013239	Economics-driven approach for managing technical ...	2013	Cloud-based Service-Oriented Architectures are co...
18	Alzaghoul201355	CloudMTD: Using real options to manage technical ...	2013	In cloud marketplace, cloud-based system architect...
19	Alzaghoul20141	Evaluating technical debt in cloud-based architectur...	2014	A Cloud-based Service-Oriented Architecture (CBSO...
20	AlΓ©groth2016257	Exploring the Presence of Technical Debt in Industri...	2016	Technical debt (TD) is a concept used to describe a ...
21	Amanatidis2017	Who is producing more technical debt? A personaliz...	2017	Technical debt (TD) impedes software projects by re...
22	Amanatidis201770	The relation between technical debt and corrective ...	2017	Context Technical Debt Management (TDM) refers t...
23	Ampatzoglou201552	The financial aspect of managing technical debt: A ...	2015	Context Technical debt is a software engineering m...
24	Ampatzoglou201575	Establishing a framework for managing interest in t...	2015	Technical debt (TD) has gained significant attention ...
25	Ampatzoglou2016117	A financial approach for managing interest in techni...	2016	Technical debt (TD) is a metaphor that is used by bo...
26	Ampatzoglou20169	The Perception of Technical Debt in the Embedded S...	2016	Technical Debt Management (TDM) has drawn the a...
27	Ampatzoglou2018115	A framework for managing interest in technical debt...	2018	Technical debt management entails the quantificati...
28	Anderson201971	SARIF-enabled tooling to encourage gradual technic...	2019	SARIF is an emerging standard for representing the ...

Εικόνα 41. Αρχείο 2ο dataset

Στην επόμενη εικόνα παρουσιάζουμε τον σχετικό κώδικα τον οποίο επεξηγούμε. Εισάγουμε τις απαραίτητες βιβλιοθήκες (γραμμές 6-14) προκειμένου να επεξεργαστούμε τα δεδομένα μας. Διαβάζουμε το αρχείο μας (γραμμή 19), κρατάμε τις στήλες που θέλουμε (γραμμή 21) και στη συνέχεια κρατάμε στο αντικείμενο paper.abstract τις περιλήψεις του dataset (γραμμή 24). Κατόπιν αφαιρούμε χαρακτήρες, σύμβολα, κενά, σημεία στίξης, αγγλικές stopwords και επιπλέον λέξεις stopwords που έχουμε δηλώσει εμείς (γραμμές 27-39).



Εικόνα 42. Κώδικας R – Topic Modeling 2ο dataset

Δημιουργούμε τον πίνακα document term matrix (γραμμή 40) και έχουμε το ακόλουθο αποτέλεσμα. Εδώ να αναφέρουμε ότι επιλέξαμε sparse 0.99 (γραμμή 41) και ο πίνακας έχει 1198 στήλες (με λέξεις – terms).

```
> inspect(dtm)
<<DocumentTermMatrix (documents: 305, terms: 1198)>>
Non-/sparse entries: 14246/351144
Sparsity           : 96%
Maximal term length: 17
Weighting          : term frequency (tf)
Sample             :
     Terms
Docs  analysis code developers development management projects quality results source
  118        5   10          0           1          0        0       5       0      5
  134        0    0          0           3          4        0       0       1      0
  149        1    2          0           1          1        1       0       2      1
  163        2    6          7           0          0        1       0       2      0
  189        0    0          0           3          0        0       0       2      0
  206        0    0          0           5          3        4       1       0      0
  228        0    0          0           1          2        1       1       4      0
  291        0    6          2           1          0        2       1       1      4
  305        3    7          0           1          0        0       0       1      7
  97         4    5          0           0          0        1       0       3      1
```

Πίνακας 24. Document Term Matrix - 2ο dataset

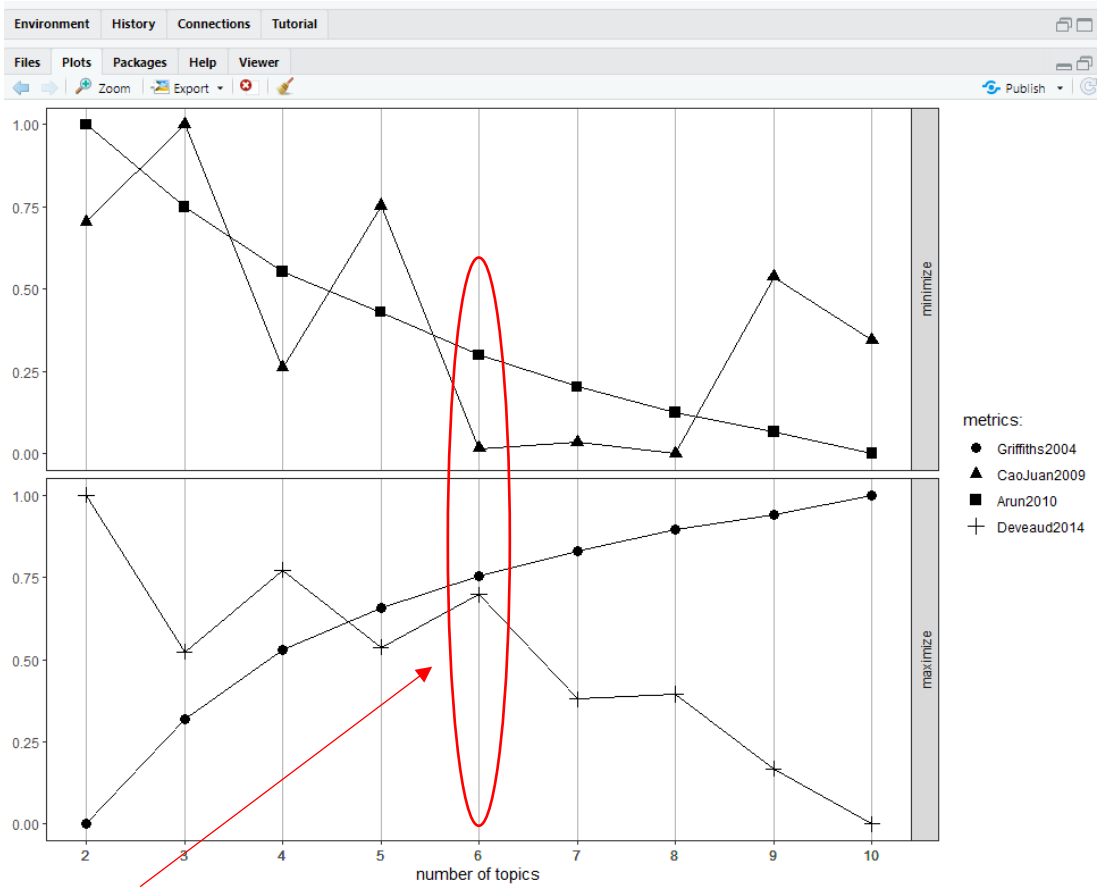
Στη συνέχεια θα εφαρμόσουμε τη μέθοδο του topic modeling για να χωρίσουμε το σύνολο δεδομένων σε ομάδες με βάση το περιεχόμενό τους.

Προκειμένου να βρούμε το θέμα κάθε άρθρου πρέπει να επιλέξουμε τον ενδεδειγμένο αριθμό των topics που πρέπει να χωρίσουμε τα δεδομένα μας . Αυτό θα προκύψει με την μέτρηση 4 μετρικών των Arun2010, CaoJuan2009, Griffiths2004, Deveaud2014 . Εκτελούμε το κατωτέρω τμήμα κώδικα. Προεπιλέγουμε τον αριθμό των topics 2-10,

```
#Arun2020 maximize, CaoJuan minimize, Griffiths minimize
optimal.topics <- FindTopicsNumber(
  dtm ,
  topics = c(2:10),
  metrics = c("Griffiths2004", "CaoJuan2009", "Arun2010","Deveaud2014"),
  method = "Gibbs",
  control = list(seed = 12345),
  mc.cores = 4L,
  verbose = TRUE
)
FindTopicsNumber_plot(optimal.topics)
```

Εικόνα 43. Μετρικές του Topic Modeling – 2ο dataset

και λαμβάνουμε το παρακάτω γράφημα



Εικόνα 44. Γράφημα Μετρικών του Topic Modeling – 2ο dataset

Με βάση το ανωτέρω γράφημα ο ενδεδειγμένος αριθμός των topics που προκύπτει είναι 6 δεδομένου ότι οι δύο μετρικές (Arun2010, CaoJuan2009) θέλουμε να είναι όσο γίνεται μικρότερες και οι άλλες δύο (Griffiths2004, Deveaud2014) όσο γίνεται μεγαλύτερες. Εδώ να προσθέσουμε ότι η επόμενη καλύτερη επιλογή μας είναι τα 4 topics.

Στη συνέχεια εκτελούμε τον ακόλουθο κώδικα επιλέγοντας τελικά 6 topics.

```
set.seed(1)
m=LDA(dtm, method="Gibbs", k=6, control=list(alpha=0.1))
#for a specific topic we can find topwords
topic = 6
words = posterior(m)$terms[topic, ]
topwords = head(sort(words, decreasing = T), n=50)
head(topwords)
```

Εικόνα 45. Δημιουργία των 6 Topics - 2ο dataset

Και έχουμε τα παρακάτω topics με τις 15 πρώτες λέξεις για το καθένα.

```
> terms(m,15)
      Topic 1          Topic 2          Topic 3      Topic 4          Topic 5          Topic 6
 [1,] "development"    "management"     "code"       "code"           "development"    "interest"
 [2,] "management"     "decisions"      "quality"    "comments"       "time"           "quality"
 [3,] "results"        "service"        "projects"   "source"         "architectural"  "framework"
 [4,] "practitioners"  "cloud"          "source"     "projects"       "companies"      "data"
 [5,] "analysis"       "requirements"   "tools"      "selfadmitted"   "teams"          "cost"
 [6,] "studies"        "decision"       "level"      "items"          "process"        "metaphor"
 [7,] "tools"          "information"    "metrics"    "developers"     "maintenance"    "development"
 [8,] "process"        "cost"           "analysis"   "classification" "product"        "maintenance"
 [9,] "support"        "problem"        "results"    "effort"         "refactoring"    "managing"
[10,] "method"         "services"       "developers" "introduced"     "developers"     "amount"
[11,] "effects"        "estimation"     "time"       "studies"        "prioritization" "design"
[12,] "metaphor"       "implementation" "evolution"  "bugs"           "interviews"     "production"
[13,] "important"      "various"        "techniques" "changes"        "test"           "domain"
[14,] "concept"        "making"         "static"     "detect"         "costs"          "present"
[15,] "identification" "provide"        "model"      "identification" "practices"      "product"
```

Οι πιο σημαντικές λέξεις ανάμεσα σε όλα τα topics φαίνονται ακολούθως :

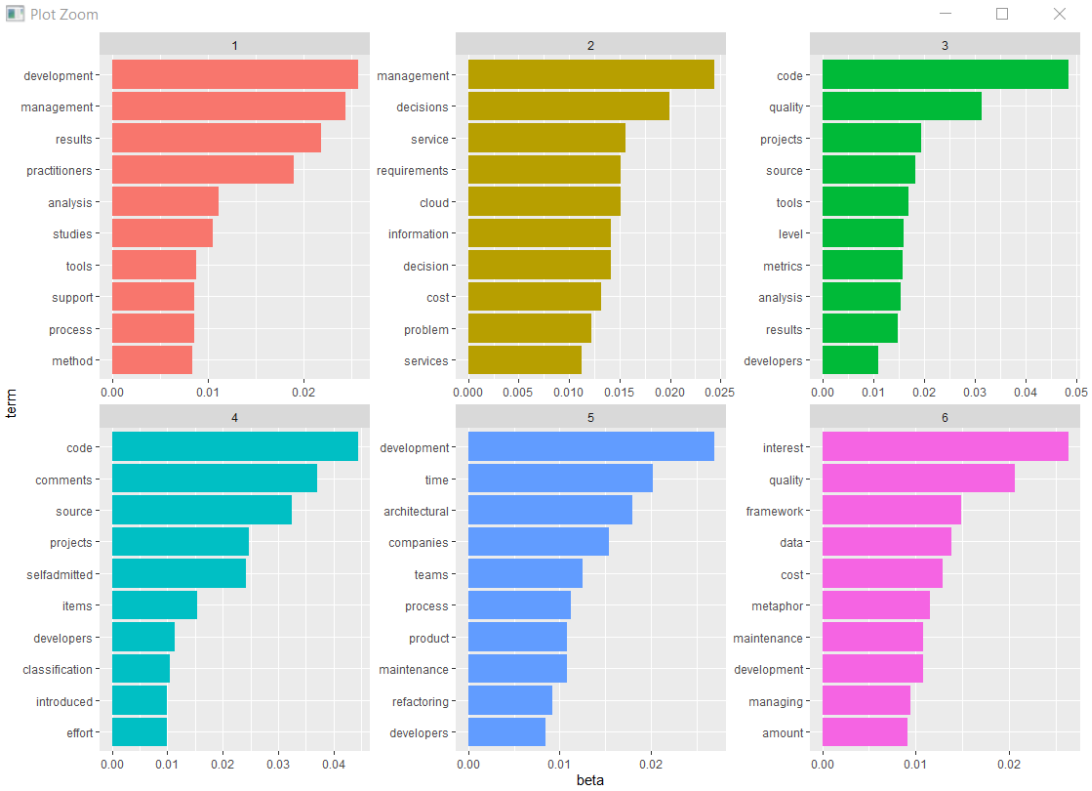
```
> head(topwords)
  interest    quality  framework       data       cost   metaphor
```

Στη συνέχεια εκτελούμε τον παρακάτω κώδικα:

```
library(tidytext)

ap_topics <- tidy(m, matrix = "beta")
ap_topics

library(ggplot2)
library(dplyr)
#plots
ap_top_terms <- ap_topics %>%
  group_by(topic) %>%
  top_n(10, beta) %>%
  ungroup() %>%
  arrange(topic, -beta)

ap_top_terms %>%
  mutate(term = reorder_within(term, beta, topic)) %>%
  ggplot(aes(beta, term, fill = factor(topic))) +
  geom_col(show.legend = FALSE) +
  facet_wrap(~ topic, scales = "free") +
  scale_y_reordered()
```

Εικόνα 46. Κώδικας δημιουργίας γραφήματος -Topic Modeling 2ο dataset

και έχουμε το ακόλουθο γράφημα



Γράφημα 33. Top 10 Terms per topic – 2ο dataset

Σε αυτό το σημείο αξίζει να κάνουμε μερικά σχόλια. Για ευνόητους λόγους αφαιρέσαμε τις λέξεις *technical debt* διότι προφανώς εμφανίζονται σε όλα τα topics και αυτό που μας ενδιαφέρει είναι το θέμα του κάθε άρθρου μέσα σε αυτό το αντικείμενο.

Οι πιο συχνές λέξεις που εμφανίζονται σε κάθε ομάδα υποδηλώνουν το θέμα των άρθρων που ανήκουν στην ομάδα αυτή. Έτσι τα topic 1 και 2 αναφέρονται στη διαχείριση του τεχνικού χρέους, το topic 3 σε θέματα ποιότητας και μετρικών του κώδικα, το topic 4 αναφέρεται στο εκούσιο τεχνικό χρέος, το topic 5 στο αρχιτεκτονικό χρέος και τέλος το topic 6 αναφέρεται στα θέματα συντήρησης του λογισμικού και στο κόστος αυτού. Στον παρακάτω πίνακα φαίνεται το περιεχόμενο του κάθε topic.

Topic	Περιγραφή	Πλήθος
1	Διαχείριση του τεχνικού χρέους και εργαλεία/μέθοδοι	80
2	Διαχείριση του τεχνικού χρέους σε ότι αφορά υπηρεσίες μέσω cloud	28
3	Μετρικές καθώς και εργαλεία μέτρησης ποιότητας του κώδικα	53

Topic	Περιγραφή	Πλήθος
4	Το εκούσιο τεχνικό χρέος	30
5	Το αρχιτεκτονικό χρέος και θέματα ανακατασκευής του κώδικα	67
6	Θέματα σχετικά με τη συντήρηση λογισμικού και το κόστος	47

Πίνακας 25. Περιεχόμενο των 6 Topics – 2ο dataset

Η ομαδοποίηση αυτή στην οποία καταλήγει ο αλγόριθμος παρατηρούμε ότι απέχει από την δική μας διότι π.χ. εντάσσει σε ξεχωριστή ομάδα τα κείμενα τα σχετικά με το αρχιτεκτονικό χρέος και τα σχετικά με το εκούσιο χρέος. Εν τούτοις, καταχωρίζει τα σχετικά με τη συντήρηση λογισμικού άρθρα σε μια ομάδα ξεχωριστά, όπως και εμείς.

Στη συνέχεια θα αποθηκεύσουμε σε μορφή csv αρχείου τα αποτελέσματα με τη βοήθεια του παρακάτω κώδικα:

το ένα αρχείο περιέχει την πιθανότητα στο κάθε topic και το άλλο το topic στο οποίο ανήκει το κάθε document.

```
#topic propabilities
topicProbabilities <- as.data.frame(m@gamma)
#probabilities for the articles to the topics
write.csv(topicProbabilities,file=paste("LDAGibbs", 6,"TopicProbabilities.csv"))
#the top 6 terms for every topic
ldaOut.terms <- as.matrix(terms(m,10))

#write out results
#docs to topics
ldaOut.topics <- as.matrix(topics(m))
write.csv(ldaOut.topics,file=paste("LDAGibbs",6,"DocsToTopics.csv"))
```

Ακολούθως φαίνονται τα σχετικά αρχεία. Στον επόμενο πίνακα βλέπουμε την πιθανότητα που έχει το κάθε άρθρο σε καθένα topic. Τελικά το κάθε άρθρο εντάσσεται σε εκείνο το topic με τη μεγαλύτερη πιθανότητα πχ. το άρθρο 1 ανήκει στο topic 2 κοκ.

LDAGibbs_6_DocsToTopics | TopicModeling305csv.R | PS1 | LDAGibbs_6_TopicProbabilities

Filter

	X1	V1	V2	V3	V4	V5	V6
1	1	0.001305483	0.6801566580	0.0013054830	0.0013054830	0.2101827676	0.1057441253
2	2	0.286786787	0.0015015015	0.0465465465	0.0015015015	0.6621621622	0.0015015015
3	3	0.692028986	0.0036231884	0.0036231884	0.0036231884	0.2934782609	0.0036231884
4	4	0.282178218	0.0016501650	0.1171617162	0.3151815182	0.0181518152	0.2656765677
5	5	0.013480392	0.0012254902	0.0379901961	0.0012254902	0.1360294118	0.8100490196
6	6	0.002403846	0.0264423077	0.0024038462	0.0024038462	0.0024038462	0.9639423077
7	7	0.403735632	0.0014367816	0.0301724138	0.0158045977	0.3031609195	0.2456896552
8	8	0.002293578	0.0022935780	0.2316513761	0.0022935780	0.2316513761	0.5298165138
9	9	0.064465409	0.0172955975	0.8663522013	0.0015723270	0.0015723270	0.0487421384
10	10	0.199604743	0.0217391304	0.0810276680	0.0019762846	0.5750988142	0.1205533597
11	11	0.015151515	0.0151515152	0.1666666667	0.0151515152	0.0151515152	0.7727272727
12	12	0.071100917	0.3233944954	0.0022935780	0.0022935780	0.3922018349	0.2087155963
13	13	0.001677852	0.0016778523	0.3875838926	0.1023489933	0.5050335570	0.0016778523
14	14	0.692164179	0.0764925373	0.0018656716	0.0018656716	0.0018656716	0.2257462687

Showing 1 to 14 of 305 entries, 7 total columns

Πίνακας 26. Documents and Topics – 2ο dataset

Στον ακόλουθο πίνακα βλέπουμε σε ποιο topic ανήκει το κάθε άρθρο.

LDAGibbs_6_DocsToTopics | TopicModeling305csv.R

Filter

	X1	Topic
1	1	2
2	2	5
3	3	1
4	4	4
5	5	6
6	6	6
7	7	1
8	8	6
9	9	3
10	10	5
11	11	6
12	12	5
13	13	5
14	14	1

Showing 1 to 14 of 305 entries, 2 total columns

Πίνακας 27. Documents and Topic Probabilities - 2ο dataset

Στη συνέχεια με ετικέτα κατηγορίας στα δεδομένα μας, αυτή που προέκυψε από το topic modeling, εκτελέσαμε 4 αλγορίθμους ταξινόμησης (Decision Tree, SVM Linear, KNN και Naïve Bayes). Δυστυχώς η απόδοση τους δεν ήταν ικανοποιητική (45 - 50 %) και αυτό μας οδηγεί στο συμπέρασμα ότι η συγκεκριμένη ομαδοποίηση (με την επιλογή των 6 topics) δεν απέδωσε στα παραπάνω μοντέλα ταξινόμησης. Την καλύτερη απόδοση είχε ο αλγόριθμος SVM Linear με ακρίβεια μόλις 51,43 %.

Στη συνέχεια θα επιχειρήσουμε να χωρίσουμε το σύνολο δεδομένων σε 4 topics και να δούμε την ομαδοποίηση που δημιουργεί. Εκτελούμε τον ίδιο κώδικα απλώς επιλέγουμε 4 topics όπως φαίνεται παρακάτω:

```
49 set.seed(1)
50 m=LDA(dtm, method="Gibbs", k=4, control=list(alpha=0.1))
51 #for a specific topic we can find topwords
52 topic = 4
53 words = posterior(m)$terms[topic, ]
54 topwords = head(sort(words, decreasing = T), n=50)
55 head(topwords)
```

Εικόνα 47. Κώδικας για 4 -Topics του 2ου dataset

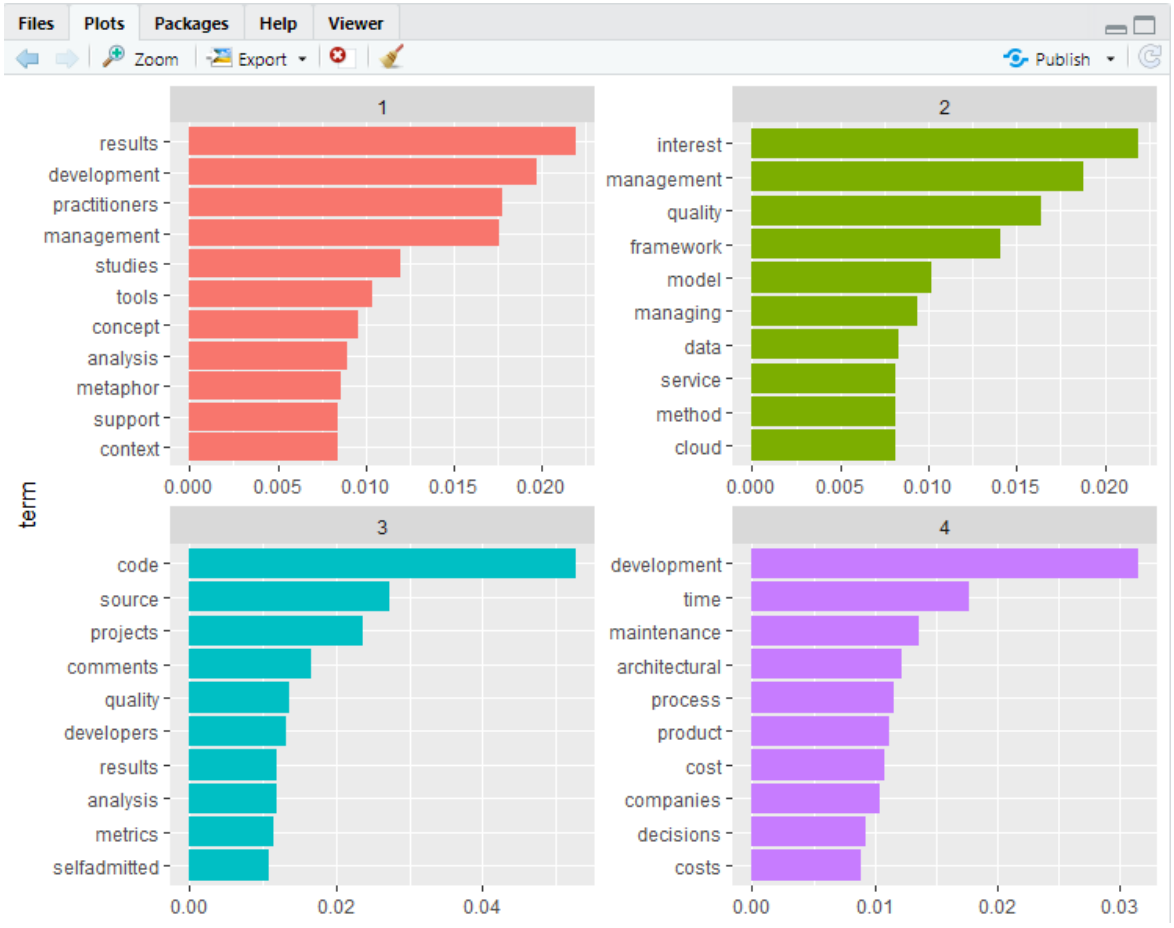
Και λαμβάνουμε τις παρακάτω ως επικρατέστερες λέξεις. Παρατηρούμε ότι λαμβάνουμε διαφορετικές λέξεις σε σχέση με την προηγούμενη ομαδοποίηση (interest, quality, framework, data, cost, metaphor)

```
> head(topwords)
  development          time   maintenance architectural       process       product
```

Οι 15 πρώτες λέξεις ανά topic φαίνονται ακολούθως

```
> #find 15 terms of every topic
> terms(m,15)
      Topic 1         Topic 2        Topic 3          Topic 4
 [1,] "results"       "interest"     "code"           "development"
 [2,] "development"   "management"   "source"         "time"
 [3,] "practitioners" "quality"      "projects"       "maintenance"
 [4,] "management"    "framework"    "comments"       "architectural"
 [5,] "studies"       "model"        "quality"        "process"
 [6,] "tools"         "managing"     "developers"     "product"
 [7,] "concept"       "data"         "analysis"       "cost"
 [8,] "analysis"      "cloud"        "results"        "companies"
 [9,] "metaphor"      "method"       "metrics"        "decisions"
[10,] "context"       "service"      "selfadmitted"   "costs"
[11,] "support"       "models"       "tools"          "benefits"
[12,] "important"     "decisions"    "level"          "teams"
[13,] "method"        "requirements" "identification" "quality"
[14,] "industry"      "context"      "model"          "results"
[15,] "information"   "tool"         "open"           "shortterm"
```

Στη συνέχεια με τον ίδιο κώδικα δημιουργίας γραφήματος (εικόνα 47) προκύπτει το ακόλουθο γράφημα:



Γράφημα 34. 4 - Topic Model 2ο dataset

Το μοντέλο χωρίζει το σύνολο δεδομένων σε 4 topics όπως παρακάτω:

Topic	Περιγραφή	Πλήθος
1	Διαχείριση του τεχνικού χρέους και εργαλεία/μέθοδοι	76
2	Διαχείριση τεχνικού χρέους, συντήρηση λογισμικού, ποιότητα κώδικα και υπηρεσίες μέσω cloud	66
3	Θέματα ποιότητας, μετρικών του πηγαίου κώδικα και εκούσιο τεχνικό χρέος	77
4	Θέματα συντήρησης, χρόνου, κόστους και αρχιτεκτονικό χρέος	86

Πίνακας 28. Περιγραφή 4 Topics - 2ο dataset

Εδώ μπορούμε να κάνουμε κάποια σχόλια. Είναι αναμενόμενο ότι ο αλγόριθμος αφού χωρίζει το σύνολο δεδομένων σε λιγότερα topics ουσιαστικά εμπλουτίζει τις ομάδες σε περιεχόμενο. Έτσι αν συγκρίνουμε τις ομάδες των 6 topics με εκείνες των 4 topics παρατηρούμε ότι το topic 4 (των 6 topics) που αφορούσε το εκούσιο χρέος συμπτύσσεται με το topic 3 (των 6 topics) το οποίο περιέχει θέματα μετρικών και ποιότητας κώδικα και σχηματίζουν το topic 3 (των 4 topics). Επίσης το topic 5 (των 6 topics) που αφορούσε το αρχιτεκτονικό χρέος συμπτύσσεται με το topic 6 (των 6 topics) το οποίο περιέχει θέματα συντήρησης λογισμικού και σχηματίζουν το 4 topic (των 4 topics). Το topic 2 (των 6 topics) που αφορούσε τη διαχείριση χρέους και υπηρεσίες μέσω cloud διαμοιράζεται με τα topic 3 και 6 (των 6 topics) τα οποία περιέχουν θέματα ποιότητας/μετρικών κώδικα και θέματα συντήρησης λογισμικού και σχηματίζεται το topic 2 (των 4 topics). Τέλος το topic 1 και στις δυο περιπτώσεις έχει το ίδιο περιεχόμενο δηλαδή αναφέρεται στη διαχείριση χρέους και στα εργαλεία/μεθόδους και μετά από έλεγχο των δύο αρχείων με τα topics διαπιστώνουμε ότι έχουν τα ίδια άρθρα και συγκεκριμένα το topic 1 (των 6 topics) έχει 4 άρθρα επιπλέον.

Στον ακόλουθο πίνακα βλέπουμε το περιεχόμενο των 6 και των 4 topics. Για λόγους καλύτερης επεξήγησης ονομάσαμε τα 4 topics σε Α, Β, Γ και Δ. Στη τελευταία στήλη του πίνακα βλέπουμε ποια topics συνενώνονται για να σχηματίσουν τα Α, Β, Γ και Δ.

Topic Modeling - 6 Topics			**Topic Modeling - 4 Topics**			
Topic	**Περιγραφή**	**Πλήθος**	**Topic**	**Περιγραφή**	**Πλήθος**	**Επεξήγηση**
1	Διαχείριση του τεχνικού χρέους και εργαλεία/μέθοδοι	80	Α	Διαχείριση του τεχνικού χρέους και εργαλεία/μέθοδοι	76	1
2	Διαχείριση του τεχνικού χρέους σε ότι αφορά υπηρεσίες μέσω cloud	28	Β	Διαχείριση τεχνικού χρέους, συντήρηση λογισμικού, ποιότητα κώδικα και υπηρεσίες μέσω cloud	66	1,2,3,6
3	Μετρικές καθώς και εργαλεία μέτρησης ποιότητας του κώδικα	53	Γ	Θέματα ποιότητας, μετρικών του πηγαίου κώδικα και εκούσιο τεχνικό χρέος	77	3,4
4	Το εκούσιο τεχνικό χρέος	30	Δ	Θέματα συντήρησης, χρόνου, κόστους και αρχιτεκτονικό χρέος	86	5,6
5	Το αρχιτεκτονικό χρέος και θέματα ανακατασκευής του κώδικα	67				
6	Θέματα σχετικά με τη συντήρηση λογισμικού και το κόστος	47				

Πίνακας 29. Σύγκριση περιεχομένου των topics

Σε σχέση με τη δική μας ομαδοποίηση πρέπει να πούμε ότι εδώ αυτό το αποτέλεσμα απέχει από το δικό μας διότι, πρώτα από όλα, τα άρθρα σχετικά με τη συντήρηση λογισμικού δεν ανήκουν σε ένα topic αλλά σε περισσότερα. Επίσης και τα σχετικά με τη διαχείριση τεχνικού χρέους ανήκουν σε 2 topics.

Στη συνέχεια με ετικέτα κατηγορίας, στα δεδομένα μας, αυτή που προέκυψε από το topic modeling, εκτελέσαμε και πάλι τους 4 αλγορίθμους ταξινόμησης (Decision Tree, SVM Linear, KNN και Naïve Bayes). Οι αλγόριθμοι έδωσαν μικρό Accuracy (κάτω του 50 %) με εξαίρεση τον Naïve Bayes με 55,56 % και τον SVM Linear με 65,71 %. Συνεπώς δεν παρουσιάζουμε περαιτέρω το σενάριο αυτό, απλώς καταγράψαμε τα αποτελέσματα και παρατηρούμε ότι τα συγκεκριμένα μοντέλα δεν απέδωσαν, με τις συγκεκριμένες ετικέτες κατηγορίας που προσδώσαμε στο 2ο dataset από την τεχνική topic modeling.

5.10 Σενάριο 1: Μοντέλα ταξινόμησης - Classification Models στο 2ο Dataset με ετικέτες κατηγορίας τις συστάδες του k-means clustering

Με τη μέθοδο του topic modeling στο 1ο dataset καταφέραμε να εντοπίσουμε τα σχετικά με τεχνικό χρέος άρθρα αλλά και τα μη σχετικά και μάλιστα με απόλυτη ευστοχία. Εφαρμόσαμε k-means clustering στο 2ο dataset και με βάση την ομαδοποίηση αυτή θα ταξινομήσουμε τα δεδομένα μας εφαρμόζοντας τους ακόλουθους αλγόριθμους ταξινόμησης:

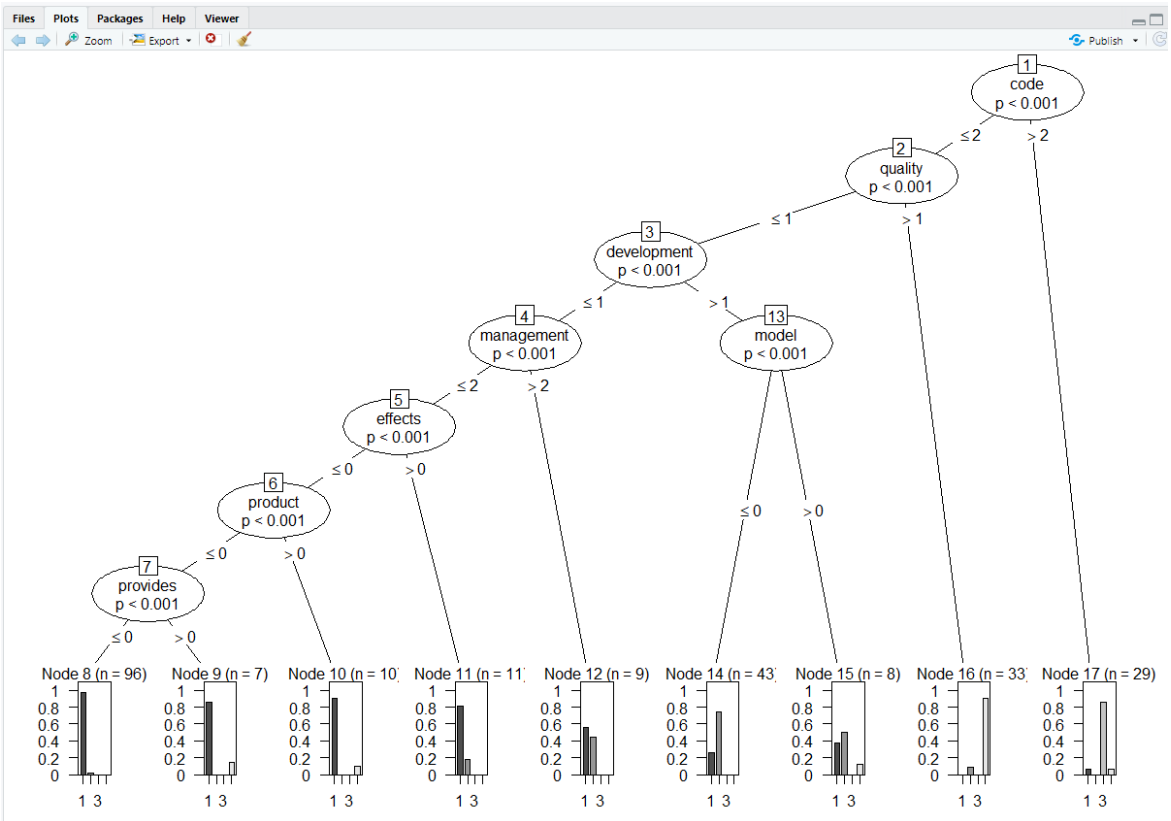
- Decision Tree
- Support Vector Machine (Linear)
- K – Nearest Neighbors
- Naïve Bayes

Κατά την εκτέλεση των μοντέλων έγιναν δοκιμές πριν την τελική καταγραφή και συλλογή των αποτελεσμάτων που παρουσιάζουμε παρακάτω. Για το διαχωρισμό των δεδομένων επιλέχθηκε, κατά περίπτωση, είτε το 80% ως σύνολο εκπαίδευσης και το υπόλοιπο 20% ως σύνολο δοκιμής, είτε το 70 %, 30 % αντίστοιχα, ανάλογα με την απόδοση του κάθε αλγορίθμου.

1) Εφαρμόζουμε το μοντέλο του Decision Tree με τον ακόλουθο κώδικα: Εδώ χωρίσαμε το σύνολο δεδομένων σε 80% και 20 %.

```
#----------------------1. DECISION TREE----------------------#
library(party)
library(zoo)
library(caret)

mydata$cg<-as.factor(mydata$cg)
set.seed(1234)
intrain<-createDataPartition(mydata$cg,p=0.8, list=F)
training<-mydata[intrain,]
testing<-mydata[-intrain,]
tree<-ctree(cg~.,data=training, controls=ctree_control(mincriterion = 0.8, minsplit=40))
plot(tree)
```

Και παράγουμε το ακόλουθο δενδρόγραμμα:

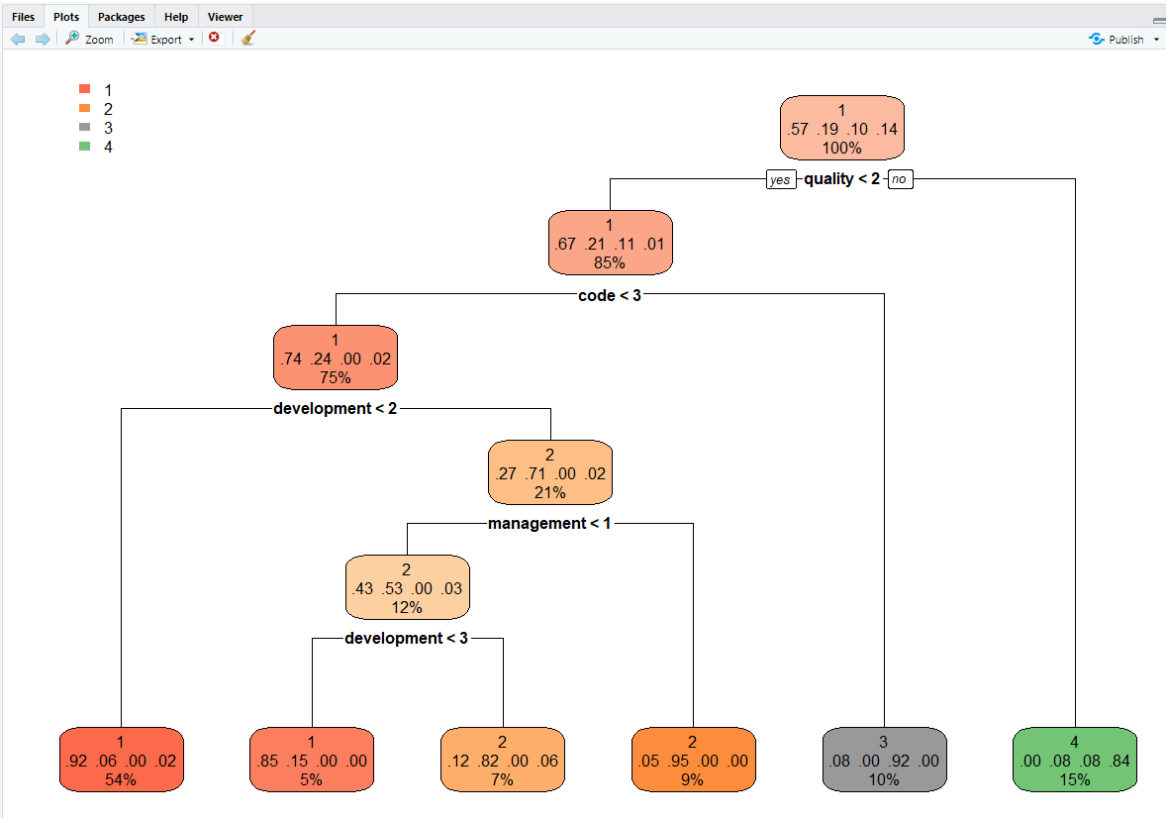


Εικόνα 48. Δενδρόγραμμα (α) - Σενάριο 1

Στο δενδρόγραμμα βλέπουμε τις πιθανότητες κατάταξης ενός άρθρου σε μια από τις 4 κατηγορίες - κλάσεις ανάλογα με την συχνότητα εμφάνισης των όρων, που φαίνονται στους κόμβους του δένδρου. Τα κείμενα με εμφάνιση του όρου code εντάσσονται στην 3η κλάση.

Αν αναφέρονται στο quality ανήκουν στην 4η κλάση. Όσα εστιάζουν στο development αφορούν την 2η κλάση και όσα σχολιάζουν τη διαχείριση τεχνικού χρέους και τις επιπτώσεις του είναι της κλάσης 1.

Στη συνέχεια αναπτύσσουμε δενδρόγραμμα για το σύνολο εκπαίδευσης στο οποίο φαίνεται η πιθανότητα για κάθε κλάση. Στα φύλλα του δένδρου βλέπουμε τα τελικά ποσοστά σε ποια κλάση ανήκουν. Ως εκ τούτου το 59 % ανήκει στην κλάση 1 που είναι και η μεγαλύτερη, το 9% στη 2η κλάση, το 10% στην 3η και το 15% στη 4η κλάση.



Εικόνα 49. Δενδρόγραμμα (β) - Σενάριο 1

Στη συνέχεια αν θέλουμε να δούμε για το σύνολο test σε ποια κλάση εντάσσεται το κάθε κείμενο τότε εκτελούμε τον παρακάτω κώδικα

```
#Prediction
predict(tree1,testing)#prediction for each article
```

Τα αποτελέσματα απόδοσης του αλγόριθμου φαίνονται ακολούθως μετά την εκτέλεση του κατωτέρω κώδικα:

Δημιουργούμε το μοντέλο όπως παρακάτω και σχηματίζουμε τη μήτρα σύγχυσης για το σετ δοκιμών

```
testPred<-predict(tree,newdata=testing)
tab<-table(testPred,testing$cg )
print(tab)
```

```
library(gmodels)
CrossTable(testPred,testing$cg,prop.chisq=F, prop.t=F,dnn=c("Predicted " , "Actual"))
```

Console Terminal Jobs

~/R/564/

```
               Accuracy : 0.9322
                 95% CI : (0.8354, 0.9812)
    No Information Rate : 0.5763
    P-Value [Acc > NIR] : 1.107e-09

                  Kappa : 0.8878

 Mcnemar's Test P-Value : NA

Statistics by Class:

                     Class: 1 Class: 2 Class: 3 Class: 4
Sensitivity            0.9412   0.8182   1.0000   1.0000
Specificity            0.9200   0.9583   1.0000   1.0000
Pos Pred Value         0.9412   0.8182   1.0000   1.0000
Neg Pred Value         0.9200   0.9583   1.0000   1.0000
Prevalence             0.5763   0.1864   0.1017   0.1356
Detection Rate         0.5424   0.1525   0.1017   0.1356
Detection Prevalence   0.5763   0.1864   0.1017   0.1356
Balanced Accuracy      0.9306   0.8883   1.0000   1.0000
> |
```

Εικόνα 50. Αποτελέσματα Decision Tree - Σενάριο 1

Το μοντέλο έχει ικανοποιητικό recall και για τις 4 κλάσεις, όπως φαίνεται στο sensitivity της παραπάνω εικόνας.

Total Observations in Table: 59

Predicted	Actual 1	2	3	4	Row Total
1	32	2	0	0	34
	0.941	0.059	0.000	0.000	0.576
	0.941	0.182	0.000	0.000	
2	2	9	0	0	11
	0.182	0.818	0.000	0.000	0.186
	0.059	0.818	0.000	0.000	
3	0	0	6	0	6
	0.000	0.000	1.000	0.000	0.102
	0.000	0.000	1.000	0.000	
4	0	0	0	8	8
	0.000	0.000	0.000	1.000	0.136
	0.000	0.000	0.000	1.000	
Column Total	34	11	6	8	59
	0.576	0.186	0.102	0.136	

Πίνακας 30. Confusion Matrix of Decision Tree - Σενάριο 1

Στον ανωτέρω πίνακα παρουσιάζουμε τη μήτρα σύγχυσης για το σετ δοκιμών, τα 59 άρθρα.

2) Support Vector Machine Linear

Στη συνέχεια εφαρμόζουμε τον Support Vector Machine με διαφορετικές παραμέτρους στο μοντέλο κατηγοριοποίησης. Αυτός που είχε τη μεγαλύτερη ακρίβεια ήταν ο Linear Support Vector Machine και είναι ο παρακάτω:

```
#------------2. SVM----------------#
library(e1071) #for SVM
set.seed(12345)
intrain<-createDataPartition(mydata$cg,p=0.7, list=F)
training<-mydata[intrain,]
testing<-mydata[-intrain,]
```

Χωρίζουμε το σύνολο δεδομένων σε 70% - 30% και στη συνέχεια δημιουργούμε το μοντέλο SVM Linear.

```
#------------ SVM LINEAR -----------------#
mymodel_linear<-svm(cg~., data=training, kernel="linear")
summary(mymodel_linear)
pred_linear<-predict(mymodel_linear,testing)
tab<-table(Predicted=pred_linear, Actual=testing$cg)
tab
confusionMatrix(table(pred_linear,testing$cg))
CrossTable(pred_linear,testing$cg,prop.chisq=F,prop.t=F,dnn=c("Predicted","Actual"))
```

Για να δούμε τα αποτελέσματα του μοντέλου εκτελούμε την pred_linear και βλέπουμε την ταξινόμηση των άρθρων που εκτελεί ο αλγόριθμος για το σύνολο δοκιμών

```
> pred_linear
 14  16  22  25  32  35  37  39  50  54  55  67  71  72  80  91 104 107 108 116 118 128 130 131 132 133
  4   2   1   2   2   1   1   1   1   1   2   1   4   4   1   1   1   2   1   1   1   4   1   1   3   2
134 143 144 155 156 168 170 179 180 183 197 199 220 222 224 226 244 260 276 291 307 311 316 321 323 325
  1   4   3   1   1   1   4   3   1   2   1   1   1   1   1   3   1   1   4   2   4   1   1   1   3   1
335 336 339 353 355 357 372 379 391 395 396 400 405 406 408 423 445 446 449 453 456 460 473 490 492 493
  2   1   2   3   4   1   1   1   1   2   1   1   2   2   2   4   1   1   1   1   1   2   1   1   4   3
502 504 506 513 518 530 531 534 538 545 553
  1   1   1   1   2   1   1   4   1   3   1
Levels: 1 2 3 4
```

Ακολούθως έχουμε την αποδοτικότητα του μοντέλου:

```
Console   Terminal ×   Jobs ×
~/R/564/

               Accuracy : 0.9775
                 95% CI : (0.9212, 0.9973)
    No Information Rate : 0.573
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.9624

 Mcnemar's Test P-Value : NA

Statistics by Class:

                     Class: 1 Class: 2 Class: 3 Class: 4
Sensitivity            1.0000   0.9412  0.88889   1.0000
Specificity            0.9474   1.0000  1.00000   1.0000
Pos Pred Value         0.9623   1.0000  1.00000   1.0000
Neg Pred Value         1.0000   0.9863  0.98765   1.0000
Prevalence             0.5730   0.1910  0.10112   0.1348
Detection Rate         0.5730   0.1798  0.08989   0.1348
Detection Prevalence   0.5955   0.1798  0.08989   0.1348
Balanced Accuracy      0.9737   0.9706  0.94444   1.0000
>
```

Εδώ παρατηρούμε ότι το μοντέλο έχει αρκετά μεγάλη ακρίβεια 97,75% με αρκετά υψηλό recall (sensitivity) και στις 4 κλάσεις.

Total Observations in Table: 89

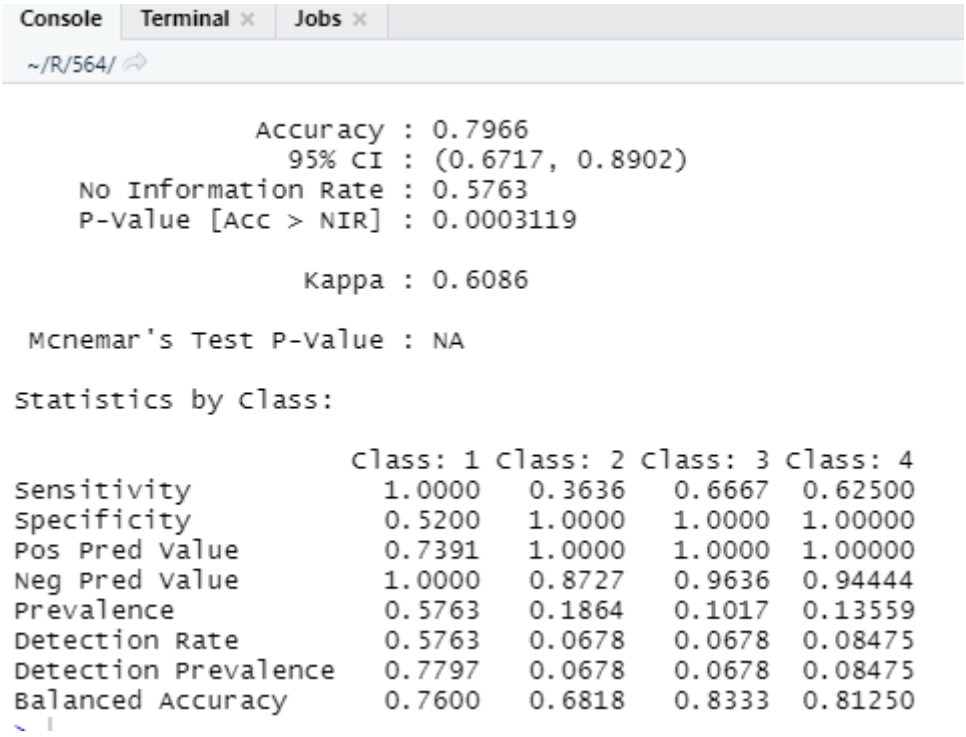
	Actual				
Predicted	1	2	3	4	Row Total
1	51	1	1	0	53
	0.962	0.019	0.019	0.000	0.596
	1.000	0.059	0.111	0.000	
2	0	16	0	0	16
	0.000	1.000	0.000	0.000	0.180
	0.000	0.941	0.000	0.000	
3	0	0	8	0	8
	0.000	0.000	1.000	0.000	0.090
	0.000	0.000	0.889	0.000	
4	0	0	0	12	12
	0.000	0.000	0.000	1.000	0.135
	0.000	0.000	0.000	1.000	
Column Total	51	17	9	12	89
	0.573	0.191	0.101	0.135	

Πίνακας 31. Confusion Matrix of SVM Linear – Σενάριο 1

3) Συνεχίζουμε με το μοντέλο KNN πλησιέστερων γειτόνων με βάση τον παρακάτω κώδικα: χωρίζουμε στο σύνολο δεδομένων σε 80 % - 20 %

```
#------------3. KNN Model-------------------#
library(caret)
library(pROC)
library(mlbench)
library(class)
library(tidyverse)

set.seed(1)
intrain<-createDataPartition(mydata$cg,p=0.8, list=F)
training<-mydata[intrain,]
testing<-mydata[-intrain,]

train_labels<-as.factor(pull(training, cg))
test_labels<-as.factor(pull(testing, cg))
training<-data.frame(select(training, - cg))
testing<-data.frame(select(testing, -cg))
knn_pred<-knn(train=training, test=testing, cl=train_labels,k=4,prob=T)
knn_pred
tb<-table(test_labels, knn_pred)
tb
sum(diag(tb))/nrow(testing)
confusionMatrix(table(Predicted=knn_pred, Actual=test_labels ))
```

Εδώ να αναφέρουμε ότι έγιναν δοκιμές με την παράμετρο κ, το πλήθος των πλησιέστερων γειτόνων με τους οποίους θα υπολογιστεί η ευκλείδεια απόσταση του εξεταζόμενου σημείου- άρθρου του συνόλου εκπαίδευσης. Ο αλγόριθμος είχε το καλύτερο accuracy για k=4 όπως φαίνεται παρακάτω:



Ο αλγόριθμος έχει χαμηλό recall (sensitivity) στη 2η κλάση και υψηλό στη 1η κλάση.

Total Observations in Table: 59

	Actual				
Predicted	1	2	3	4	Row Total
1	34	7	2	3	46
	0.739	0.152	0.043	0.065	0.780
	1.000	0.636	0.333	0.375	
2	0	4	0	0	4
	0.000	1.000	0.000	0.000	0.068
	0.000	0.364	0.000	0.000	
3	0	0	4	0	4
	0.000	0.000	1.000	0.000	0.068
	0.000	0.000	0.667	0.000	
4	0	0	0	5	5
	0.000	0.000	0.000	1.000	0.085
	0.000	0.000	0.000	0.625	
Column Total	34	11	6	8	59
	0.576	0.186	0.102	0.136	

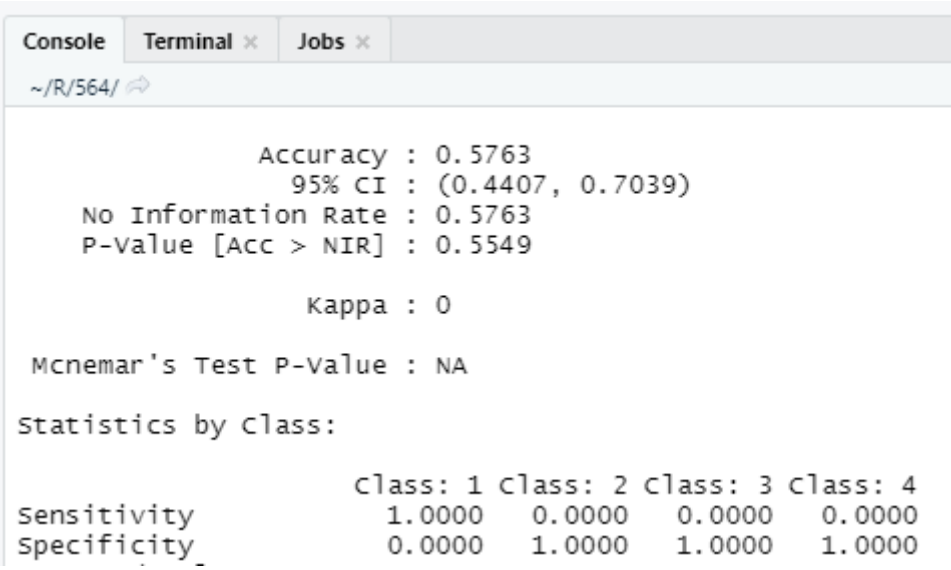
Πίνακας 32. Confusion Matrix of KNN – Σενάριο 1

Στον πίνακα φαίνεται η μήτρα σύγχυσης για το σετ δοκιμών.

4) Ο αλγόριθμος με την πιο περιορισμένη απόδοση από όλους όσους χρησιμοποιήσαμε ήταν ο Naïve Bayes με τις παρακάτω παραμέτρους όπως φαίνονται στον ακόλουθο κώδικα:

```
#--------------4 NAIVE BAYES --------------#

myCntrl<-trainControl(method="repeatedcv" , number=10, repeats=10 )
set.seed(1234)
tc<-train(cg ~., data=training, method="nb", trControl=myCntrl)
tc
#prediction
pre_nb<-predict(tc, newdata=testing)
table(pre_nb,testing$cg)
#confusion matrix
confusionMatrix(table(pre_nb,testing$cg) )
```

Η μέθοδος που χρησιμοποιήσαμε ήταν η repeated cross validation με 10 folds και 10 επαναλήψεις. Ανάμεσα στις δοκιμές που εκτελέσαμε αυτή είχε την υψηλότερη απόδοση. Τα αποτελέσματα φαίνονται ακολούθως :



Console | Terminal | Jobs

~/R/564/

```
               Accuracy : 0.5763
                 95% CI : (0.4407, 0.7039)
    No Information Rate : 0.5763
    P-Value [Acc > NIR] : 0.5549

                  Kappa : 0

 Mcnemar's Test P-Value : NA

Statistics by Class:

                     Class: 1 Class: 2 Class: 3 Class: 4
Sensitivity            1.0000   0.0000   0.0000   0.0000
Specificity            0.0000   1.0000   1.0000   1.0000
```

pre	testing$cg 1	2	3	4	Row Total
1	34	11	6	8	59
	0.576	0.186	0.102	0.136	
Column Total	34	11	6	8	59

Πίνακας 33. Confusion Matrix of Naive Bayes – Σενάριο 1

Από τα ανωτέρω αποτελέσματα διαπιστώνουμε ότι ο αλγόριθμος έχει recall μόνο για την 1η κλάση , στις άλλες είναι 0 συνεπώς δεν μπορεί να ταξινομήσει το σετ δεδομένων.

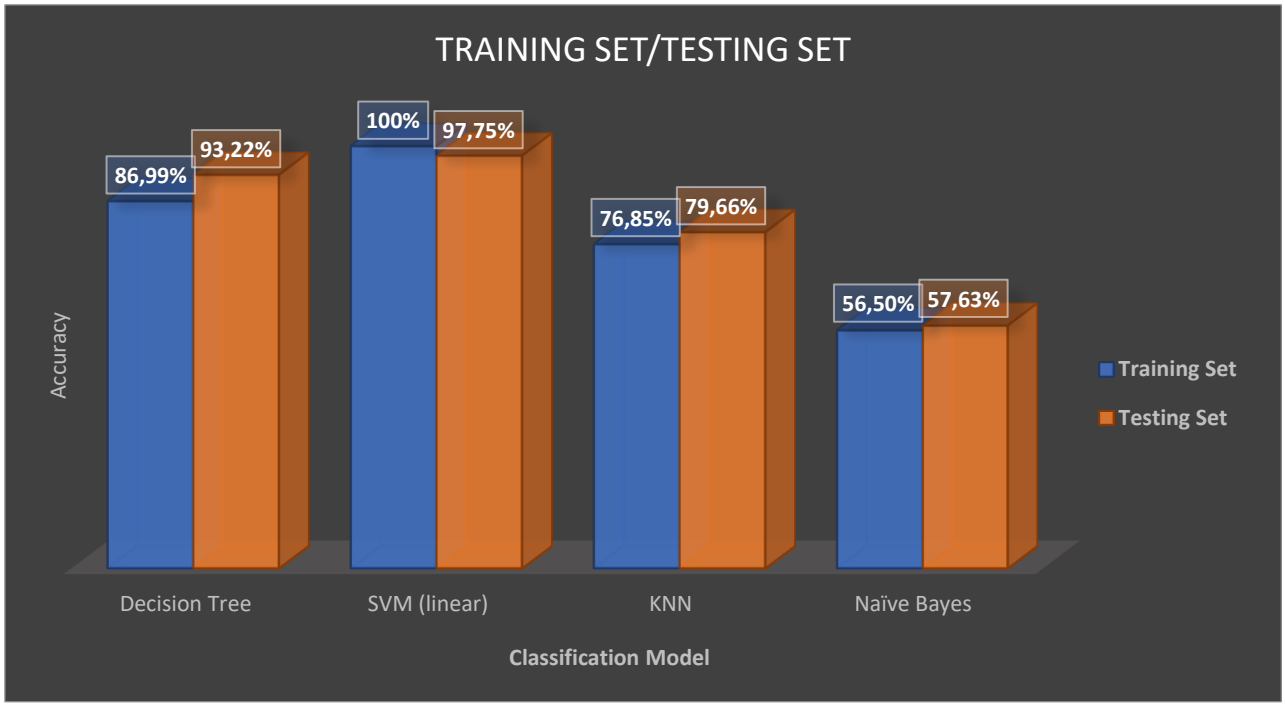
Στη συνέχεια εκτελέσαμε όλους τους παραπάνω αλγορίθμους και υπολογίσαμε το Accuracy τόσο για το σύνολο εκπαίδευσης όσο και για το σύνολο δοκιμών.

Παρουσιάζουμε τα αποτελέσματά μας συγκεντρωτικά:

	Accuracy		
Classification Model	**Training Set**	**Testing Set**	**Διαφορά**
Decision Tree	**86,99 %**	**93,22%**	**6,23%**
SVM (Linear)	**100 %**	**97,75%**	**2,25%**
K – Nearest Neighbors	**76,85 %**	**79,66%**	**2,81%**
Naïve Bayes	**56,5 %**	**57,63%**	**1,13%**

Πίνακας 34. Accuracy of the Models (training set & testing set) – Σενάριο 1

Στον ανωτέρω πίνακα διαπιστώνουμε ότι τη μεγαλύτερη διαφορά στο accuracy μεταξύ των δύο συνόλων training set και testing set παρουσιάζει ο Decision Tree ενώ τη μικρότερη ο Naïve Bayes που σημαίνει ότι έχει τη μικρότερη τάση υπερπροσαρμογής και ακολουθούν ο SVM linear και ο KNN. Στο ακόλουθο γράφημα φαίνονται τα αποτελέσματα του Accuracy για τα 4 μοντέλα και στα δύο σύνολα εκπαίδευσης και δοκιμής.



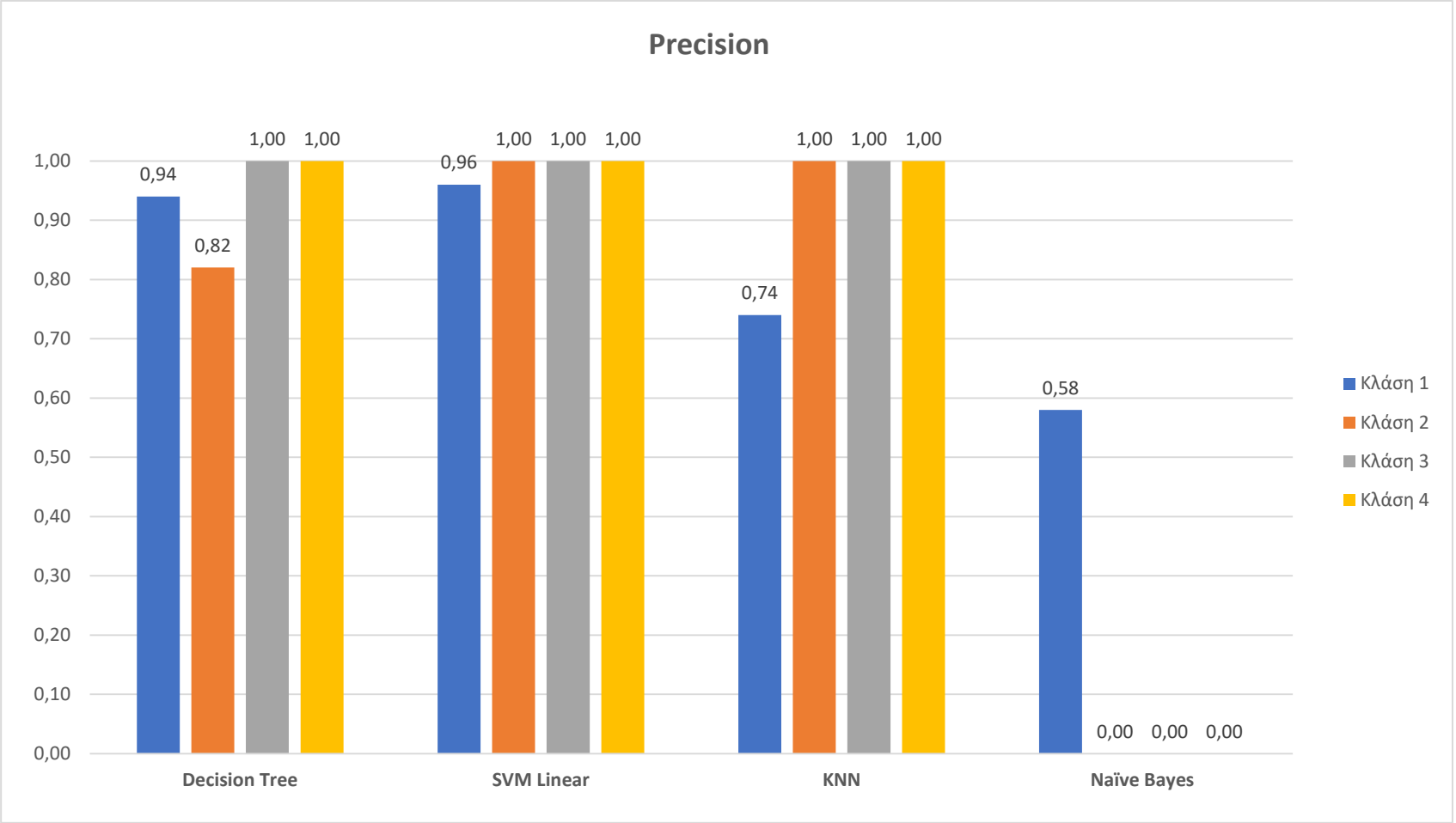
Γράφημα 35. Accuracy (training & testing set) – Σενάριο 1

Στη συνέχεια παρουσιάζουμε όλες τις μετρικές απόδοσης των αλγορίθμων για κάθε κλάση ξεχωριστά.

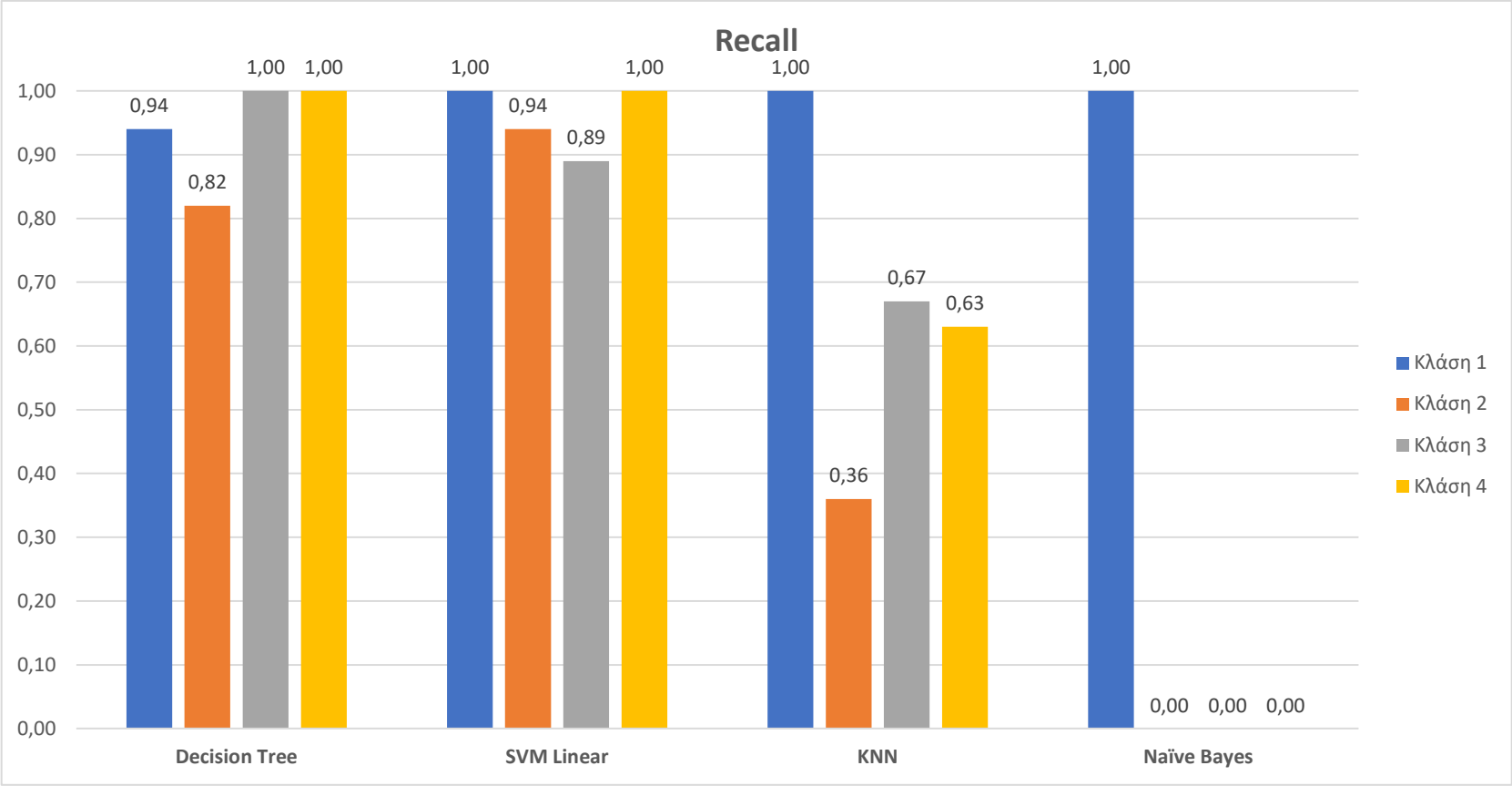
		Precision				Recall				F1			
		Class Labels				Class Labels				Class Labels			
Classification Model	**Accuracy**	**1**	**2**	**3**	**4**	**1**	**2**	**3**	**4**	**1**	**2**	**3**	**4**
Decision Tree	**93,22%**	**0,94**	**0,82**	**1,00**	**1,00**	**0,94**	**0,82**	**1,00**	**1,00**	**0,94**	**0,82**	**1,00**	**1,00**
SVM (Linear)	**97,75%**	**0,96**	**1,00**	**1,00**	**1,00**	**1,00**	**0,94**	**0,89**	**1,00**	**0,98**	**0,97**	**0,94**	**1,00**
K – Nearest Neighbors	**79,66%**	**0,74**	**1,00**	**1,00**	**1,00**	**1,00**	**0,36**	**0,67**	**0,63**	**0,85**	**0,53**	**0,50**	**0,77**
Naïve Bayes	**57,63%**	**0,58**	**0**	**0**	**0**	**1,00**	**0**	**0**	**0**	**0,73**	**0**	**0**	**0**

Πίνακας 35. Μετρικές Απόδοσης Αλγορίθμων – Σενάριο 1

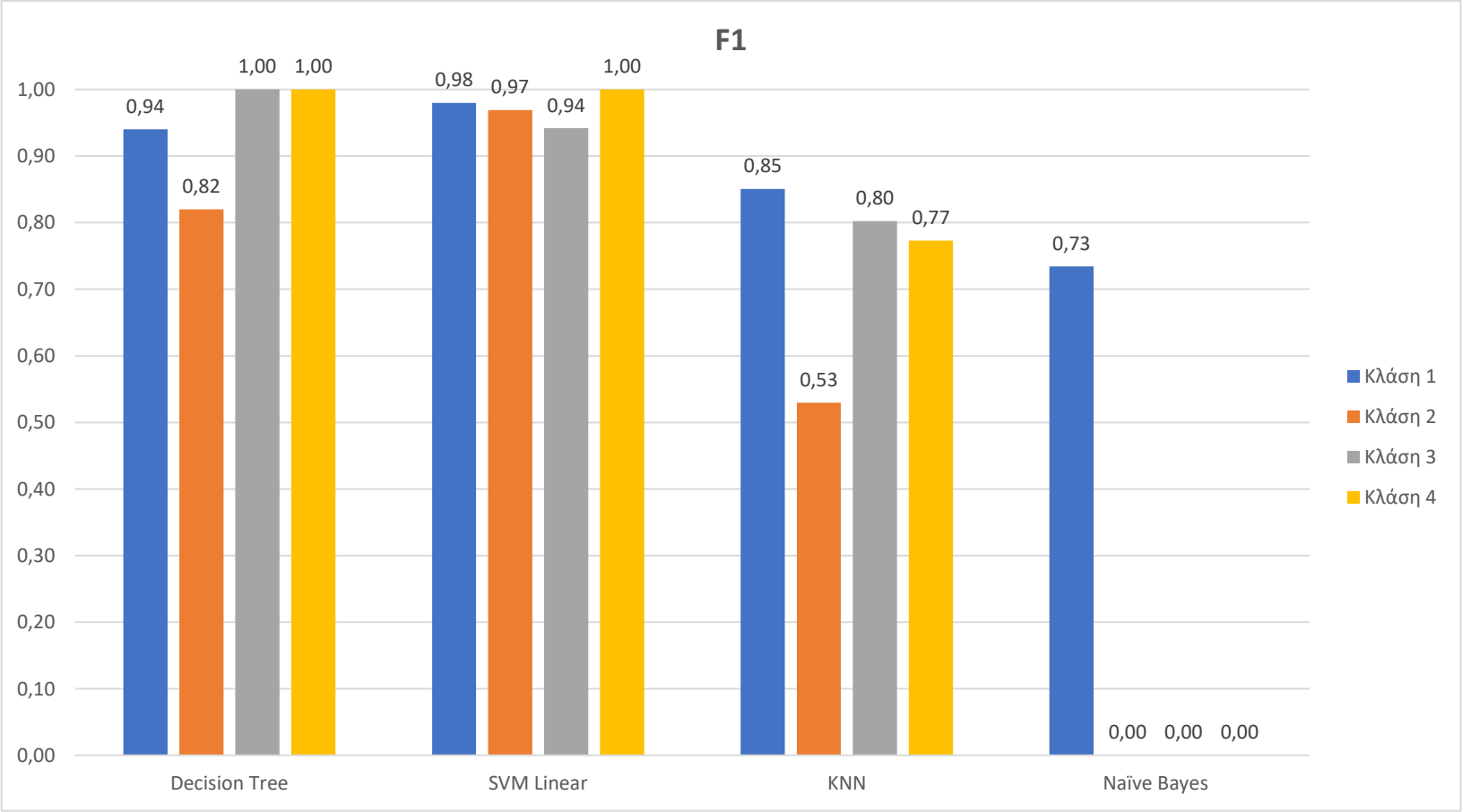
Ακολουθούν τα γραφήματα ανά μετρική για τα 4 μοντέλα .



Γράφημα 36. Precision of the models - Σενάριο 1

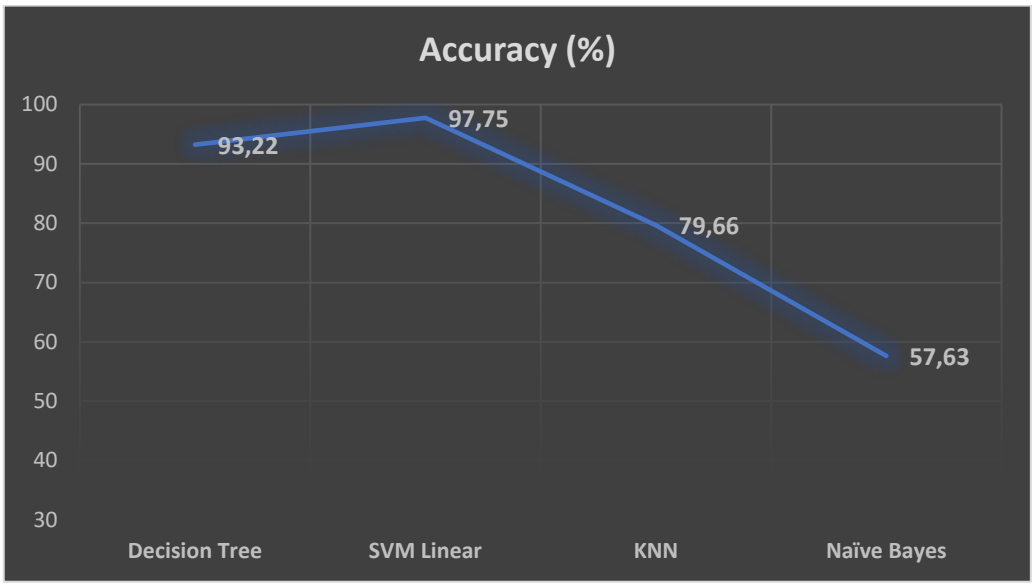


Γράφημα 37. Recall of the models - Σενάριο 1



Γράφημα 38. F1 of the models - Σενάριο 1

Στο ακόλουθο γράφημα βλέπουμε την ακρίβεια των 4 αλγορίθμων πρόβλεψης που εφαρμόσαμε.



Γράφημα 39. Accuracy - Σενάριο 1

Ο πιο αποδοτικός αλγόριθμος για το σύνολο δεδομένων που εξετάζουμε είναι ο SVM Linear και ακολουθούν ο Decision Tree, ο KNN και με μεγάλη διαφορά έπεται ο Naïve Bayes.

5.11 Σενάριο 2: Μοντέλα ταξινόμησης - Classification Models μετά από εμπειρική ομαδοποίηση στο 2ο dataset

Στο σενάριο αυτό θα χρησιμοποιήσουμε τις ετικέτες κατηγορίας που έχουμε εισάγει εμείς μετά από την μελέτη των περιλήψεων και ολόκληρων των κειμένων του 2ου dataset. Κατεβάσαμε τα άρθρα (πλήρη κείμενα) και τα διαβάσαμε ώστε να είμαστε σε θέση να προσδώσουμε σε αυτά τη σωστή ετικέτα κατηγορίας. Στη συνέχεια εκτελούμε εκ νέου τα μοντέλα ταξινόμησης, δηλαδή με ανάλογο τρόπο όπως στο προηγούμενο σενάριο και εφαρμόζουμε :

- Decision Tree
- Support Vector Machine (Linear)
- K – Nearest Neighbors
- Naïve Bayes

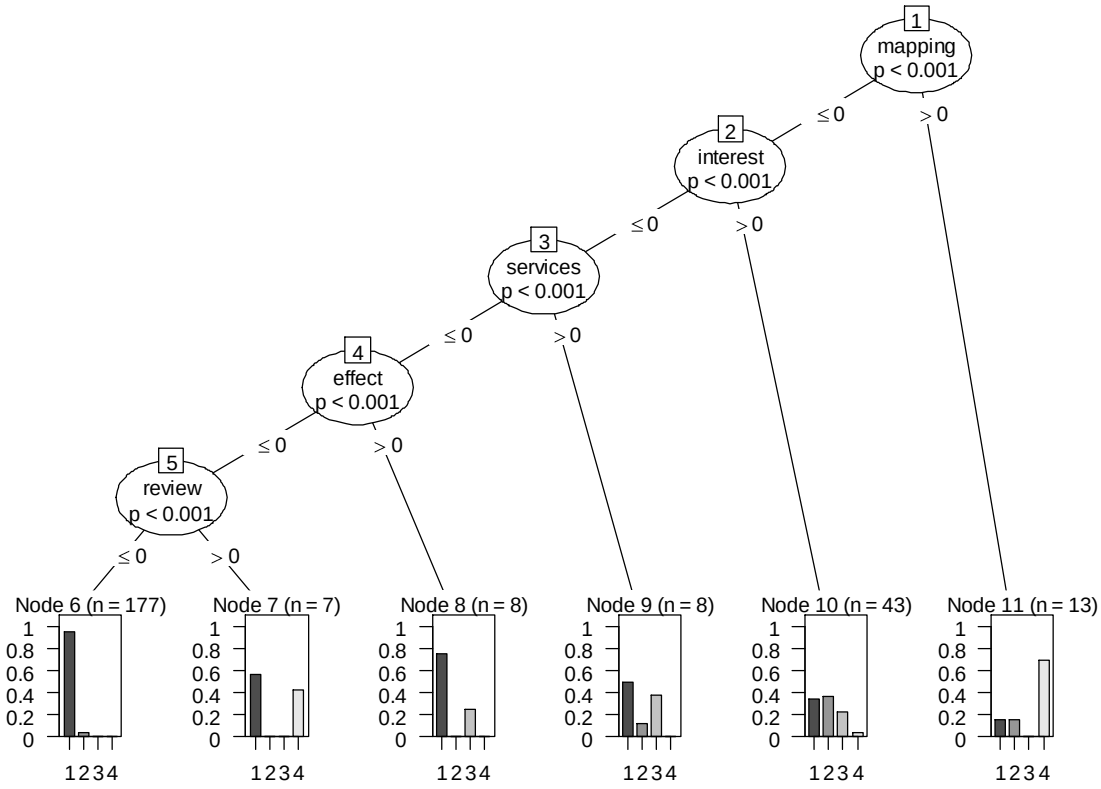
Κατόπιν θα τα αξιολογήσουμε τόσο μεταξύ τους όσο και με τα αποτελέσματα του Σεναρίου 1.

1) Decision Tree

Ξεκινάμε με το Decision Tree και εκτελούμε τον ακόλουθο κώδικα. Εδώ να σημειώσουμε ότι χωρίσαμε το dataset σε 80% και 20% και αυτό διότι ο αλγόριθμος μας έδωσε καλύτερη ακρίβεια .

```
#-----------DECISION TREE-----------------#
library(party)
library(zoo)
tree<-ctree(articles.Label~.,data=training, controls=ctree_control(mincriterion = 0.8, minsplit=40))

plot(tree)
```

Παράγεται το ακόλουθο δενδρόγραμμα



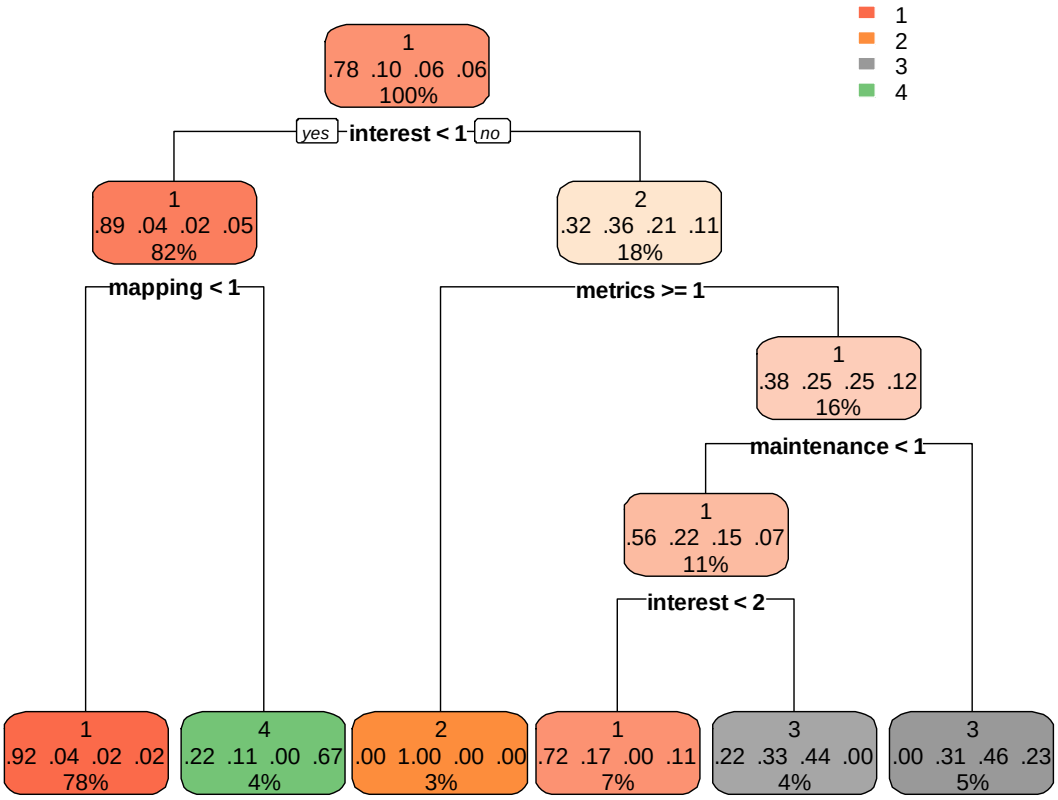
Εικόνα 51. Δενδρόγραμμα (α) - Σενάριο 2

Στη κατηγορία – κλάση 4 ανήκουν τα άρθρα της βιβλιογραφικής ανασκόπησης (node 11), τα κείμενα που αναφέρονται στη συντήρηση λογισμικού αφορούν την 2η κλάση αλλά και την 1η και 3η (node 10). Τα άρθρα της κλάσης 1 αφορούν κυρίως τις επιπτώσεις του τεχνικού χρέους στα έργα ανάπτυξης λογισμικού (node 8). Επίσης οι κλάσεις 1 και 3 αφορούν τον τομέα των υπηρεσιών (node 9) του λογισμικού (Software As a Service).

Στη συνέχεια αναπτύσσουμε δενδρόγραμμα για το σύνολο εκπαίδευσης στο οποίο φαίνεται η πιθανότητα για κάθε κλάση. Στα φύλλα του δένδρου βλέπουμε τα τελικά ποσοστά σε ποια κλάση ανήκουν. Ως εκ τούτου το 85 % ανήκει στην κλάση 1 που είναι και η μεγαλύτερη, το 3% στη 2η κλάση, το 9% στην 3η και το 4% στη 4η κλάση.

Πρώτα εισάγουμε τις απαραίτητες βιβλιοθήκες για το δέντρο απόφασης τις rpart και rpart.plot όπως βλέπουμε παρακάτω και δημιουργούμε το μοντέλο ταξινόμησης:

```
88  #Decision tree with rpart
89  library(rpart)
90  library(rpart.plot)
91
92  tree1<-rpart(articles.Label~., training)
93  rpart.plot(tree1)
```



Εικόνα 52. Δενδρόγραμμα (β) – Σενάριο 2

Σύμφωνα με το παραπάνω δέντρο να πούμε ότι η 4η κλάση είναι τα κείμενα της βιβλιογραφικής ανασκόπησης τα οποία εντοπίζονται από τη λέξη mapping εκ του mapping study που επικρατεί ως φράση στις ανασκοπήσεις. Η 3η κλάση αφορά τη συντήρηση λογισμικού εκ των όρων maintenance και interest. Η 2η κλάση αναφέρεται στις μετρικές ποιότητας κώδικα.

Τα αποτελέσματα του αλγορίθμου φαίνονται ακολούθως

```
Overall Statistics

               Accuracy : 0.7959
                 95% CI : (0.6566, 0.8976)
    No Information Rate : 0.7551
    P-Value [Acc > NIR] : 0.3171

                  Kappa : 0.362

 Mcnemar's Test P-Value : NA

Statistics by Class:

                     Class: 1 Class: 2 Class: 3 Class: 4
Sensitivity            0.9459  0.57143  0.00000  0.00000
Specificity            0.3333  0.95238  1.00000  1.00000
```

Εικόνα 53. Αποτελέσματα Decision Tree - Σενάριο 2

Παρατηρούμε ότι δυστυχώς το μοντέλο έχει μηδενικό recall για τις κλάσεις 3 και 4 κάτι που διαπιστώνουμε και στη μήτρα σύγχυσης που ακολουθεί στην οποία ο πίνακας έχει 2 γραμμές αντί για 4 που είναι κλάσεις.

Η μήτρα σύγχυσης μας δίνει την ταξινόμηση του σετ δοκιμών, δηλαδή για τα 49 άρθρα.

Total Observations in Table: 49

	Actual				
Predicted	1	2	3	4	Row Total
1	35	3	4	1	43
	0.814	0.070	0.093	0.023	0.878
	0.946	0.429	1.000	1.000	
2	2	4	0	0	6
	0.333	0.667	0.000	0.000	0.122
	0.054	0.571	0.000	0.000	
Column Total	37	7	4	1	49
	0.755	0.143	0.082	0.020	

Πίνακας 36. Confusion Matrix of Decision Tree - Σενάριο 2

2) SVM Linear

Και σε αυτή τη περίπτωση χωρίσαμε σε 80% και 20% το σύνολο δεδομένων και εφαρμόσαμε τον SVM Linear που είχε την καλύτερη ακρίβεια.

```
#------------------SVM---------------------#
library(e1071) #for SVM
set.seed(222)
intrain<-createDataPartition(newcorpus$articles.Label,p=0.8, list=F)
training<-newcorpus[intrain,]
testing<-newcorpus[-intrain,]
```

```
#------------------- SVM LINEAR -------------------#
mymodel_linear<-svm(articles.Label~., data=training, kernel="linear")
summary(mymodel_linear)
pred_linear<-predict(mymodel_linear,testing)
tab<-table(Predicted=pred_linear, Actual=testing$articles.Label)
tab
confusionMatrix(table(pred_linear,testing$articles.Label))
CrossTable(pred_linear,testing$articles.Label,prop.chisq=F, prop.t=F, dnn=c("Predicted " , "Actual"))
```

```
               Accuracy : 0.8814
                 95% CI : (0.7707, 0.9509)
    No Information Rate : 0.7966
    P-Value [Acc > NIR] : 0.06657

                  Kappa : 0.5911

 Mcnemar's Test P-Value : NA

Statistics by Class:

                     Class: 1 Class: 2 Class: 3 Class: 4
Sensitivity            0.9787   0.3333  0.66667  0.66667
Specificity            0.5000   1.0000  1.00000  0.98214
Pos Pred Value         0.8846   1.0000  1.00000  0.66667
Neg Pred Value         0.8571   0.9298  0.98246  0.98214
Prevalence             0.7966   0.1017  0.05085  0.05085
Detection Rate         0.7797   0.0339  0.03390  0.03390
Detection Prevalence   0.8814   0.0339  0.03390  0.05085
Balanced Accuracy      0.7394   0.6667  0.83333  0.82440
```

Εικόνα 54. Αποτελέσματα SVM Linear - Σενάριο 2

Παρατηρούμε ότι το μοντέλο έχει καλό recall για τη 1η κλάση και χαμηλότερο για την 3η , 4η και αρκετά μικρότερο για την 2η κλάση. Ακολουθεί η μήτρα σύγχυσης για το σετ δοκιμών.

Total Observations in Table: 59

Predicted	Actual 1	2	3	4	Row Total
1	46	4	1	1	52
	0.885	0.077	0.019	0.019	0.881
	0.979	0.667	0.333	0.333	
2	0	2	0	0	2
	0.000	1.000	0.000	0.000	0.034
	0.000	0.333	0.000	0.000	
3	0	0	2	0	2
	0.000	0.000	1.000	0.000	0.034
	0.000	0.000	0.667	0.000	
4	1	0	0	2	3
	0.333	0.000	0.000	0.667	0.051
	0.021	0.000	0.000	0.667	
Column Total	47	6	3	3	59
	0.797	0.102	0.051	0.051	

Πίνακας 37. Confusion Matrix of SVM Linear - Σενάριο 2

3) KNN πλησιέστερων γειτόνων

Στη συνέχεια προχωράμε με το μοντέλο KNN πλησιέστερων γειτόνων με βάση τον παρακάτω κώδικα. Έχουμε κάνει τις απαραίτητες δοκιμές και έχουμε χωρίσει το σύνολο δεδομένων σε 80% - 20%. Η μεγαλύτερη ακρίβεια επετεύχθη με κ=3.

```
#--------------3 KNN MODEL -------------------------#
library(caret)
library(pROC)
library(mlbench)
library(class)
library(tidyverse)
set.seed(222)
intrain<-createDataPartition(newcorpus$articles.Label,p=0.8, list=F)
training<-newcorpus[intrain,]
testing<-newcorpus[-intrain,]
train_labels<-as.factor(pull(training, articles.Label))
test_labels<-as.factor(pull(testing, articles.Label))
training<-data.frame(select(training, - articles.Label))
testing<-data.frame(select(testing, -articles.Label))
knn_pred<-knn(train=training, test=testing, cl=train_labels,k=3,prob=T)
knn_pred
tb<-table(test_labels, knn_pred)
tb
sum(diag(tb))/nrow(testing)
confusionMatrix(table(Predicted=knn_pred, Actual=test_labels))
CrossTable(knn_pred,test_labels,prop.chisq=F, prop.t=F,dnn=c("Predicted " , "Actual"))
```

Τα αποτελέσματα μας δείχνουν ακρίβεια 83,05% αν και διαπιστώνουμε ότι ικανοποιητικό recall υπάρχει μόνο για την 1η κλάση και μέτριο για την 3η κλάση.

```
                Accuracy : 0.8305
                  95% CI : (0.7103, 0.9156)
    No Information Rate : 0.7966
    P-Value [Acc > NIR] : 0.3229

                   Kappa : 0.2588

 Mcnemar's Test P-Value : NA

Statistics by Class:

                     Class: 1 Class: 2 Class: 3 Class: 4
Sensitivity            1.0000   0.0000  0.66667  0.00000
Specificity            0.1667   1.0000  1.00000  1.00000
Pos Pred Value         0.8246      NaN  1.00000      NaN
Neg Pred Value         1.0000   0.8983  0.98246  0.94915
Prevalence             0.7966   0.1017  0.05085  0.05085
Detection Rate         0.7966   0.0000  0.03390  0.00000
Detection Prevalence   0.9661   0.0000  0.03390  0.00000
Balanced Accuracy      0.5833   0.5000  0.83333  0.50000
```

Εικόνα 55. Αποτελέσματα KNN - Σενάριο 2

Ακολουθεί η μήτρα σύγχυσης και εδώ διαπιστώνουμε από τις γραμμές του πίνακα ότι το μοντέλο δεν ταξινομεί άρθρα στην 2η και 4η κλάση. Ωστόσο για την 3η κλάση έχει 66,67%.

Total Observations in Table: 59

	Actual				
Predicted	1	2	3	4	Row Total
1	47	6	1	3	57
	0.825	0.105	0.018	0.053	0.966
	1.000	1.000	0.333	1.000	
3	0	0	2	0	2
	0.000	0.000	1.000	0.000	0.034
	0.000	0.000	0.667	0.000	
Column Total	47	6	3	3	59
	0.797	0.102	0.051	0.051	

Πίνακας 38. Confusion Matrix KNN - Σενάριο 2

4) Naïve Bayes

Ολοκληρώνουμε τις δοκιμές με το μοντέλο Naïve Bayes, με το οποίο παρά τις δοκιμές που κάναμε αλλάζοντας τις παραμέτρους όπως ο διαχωρισμός των δεδομένων σε 70 - 30 ή 80 - 20 καθώς και το repeated cross validation, δεν πετύχαμε ικανοποιητική ακρίβεια. Επίσης ο αλγόριθμος αυτός ήταν και ο πιο χρονοβόρος λόγω του cross validation. Τελικά επιλέγουμε , όπως φαίνεται και παρακάτω, διαχωρισμό 80 % και 20 % στα σετ εκπαίδευσης και δοκιμών αντίστοιχα και cross validation με fold 10 και 10 επαναλήψεις.

```
#-------------------4 NAIVE BAYES ------------------------#
library(e1071)
library(caret)
library(lattice)
set.seed(222)
intrain<-createDataPartition(newcorpus$articles.Label,p=0.8, list=F)
training<-newcorpus[intrain,]
testing<-newcorpus[-intrain,]
myCntrl<-trainControl(method="repeatedcv" ,number=10, repeats=10)
tc<-train(articles.Label ~., data=training, method="nb", trControl=myCntrl)
tc
#prediction
prenb<-predict(tc, newdata=testing)
confusionMatrix(table(Predicted=prenb,Actual=testing$articles.Label))
```

```
               Accuracy : 0.7966
                 95% CI : (0.6717, 0.8902)
    No Information Rate : 0.7966
    P-Value [Acc > NIR] : 0.5765

                  Kappa : 0

 Mcnemar's Test P-Value : NA

Statistics by Class:

                     Class: 1 Class: 2 Class: 3 Class: 4
Sensitivity            1.0000   0.0000  0.00000  0.00000
Specificity            0.0000   1.0000  1.00000  1.00000
```

Εικόνα 56. Αποτελέσματα Naive Bayes - Σενάριο 2

Το μοντέλο, όπως φαίνεται, έχει αποδοτικό recall μόνο για την 1η κλάση και δεν ενδείκνυται για τις υπόλοιπες κλάσεις. Αυτό φαίνεται και από την μήτρα σύγχυσης. Στην επόμενη εικόνα έχουμε τη μήτρα σύγχυσης.

prenb	1	2	3	4
1	47	6	3	3
2	0	0	0	0
3	0	0	0	0
4	0	0	0	0

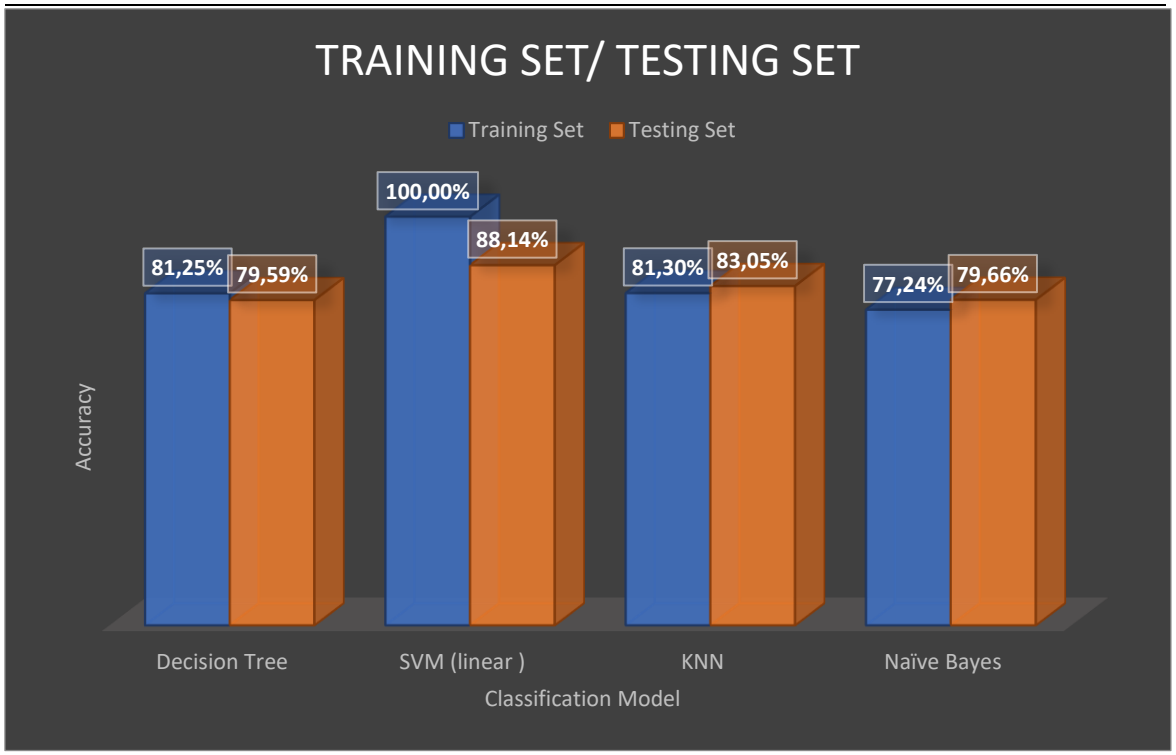
Πίνακας 39. Confusion Matrix Naive Bayes - Σενάριο 2

Στη συνέχεια εκτελέσαμε όλους τους παραπάνω αλγορίθμους και υπολογίσαμε το Accuracy τόσο για το σύνολο εκπαίδευσης όσο και για το σύνολο δοκιμών. Παρουσιάζουμε τα αποτελέσματά μας συγκεντρωτικά:

	Accuracy		
Classification Model	**Training Set**	**Testing Set**	**Διαφορά**
Decision Tree	**81,25 %**	**79,59 %**	**1,66 %**
SVM (linear)	**100 %**	**88,14%**	**11,86 %**
K – Nearest Neighbors	**81,3 %**	**83,05 %**	**1,75 %**
Naïve Bayes	**77,24 %**	**79,66 %**	**2,42 %**

Πίνακας 40. Accuracy of the Models (training set & testing set) - Σενάριο 2

Στο ακόλουθο γράφημα φαίνονται τα αποτελέσματα του Accuracy για τα 4 μοντέλα και στα δύο σύνολα εκπαίδευσης και δοκιμής.



Γράφημα 40. Accuracy (training & testing set) – Σενάριο 2

Παρατηρούμε ότι τη μικρότερη τάση υπερπροσαρμογής έχουν οι Decision Tree και KNN ενώ στο Σενάριο 1 τη μικρότερη τάσης υπερπροσαρμογής παρουσίαζε ο Naïve Bayes. Επίσης παρατηρούμε ότι εδώ ο Naïve Bayes εμφανίζει μεγαλύτερο Accuracy. Επιπρόσθετα να αναφέρουμε ότι και στα δύο σενάρια ο SVM linear έχει 100% Accuracy στο σύνολο εκπαίδευσης. Ωστόσο το Σενάριο 1 έχει καλύτερες αποδόσεις ταξινόμησης στους Decision Tree και SVM linear ενώ οι KNN και Naïve Bayes έχουν μεγαλύτερη ακρίβεια στο Σενάριο 2. Παρακάτω έχουμε τον πίνακα σύγκρισης των δύο Σεναρίων.

	Σενάριο 1		**Σενάριο 2**	
Classification Model	**Training Set**	**Testing Set**	**Training Set**	**Testing Set**
Decision Tree	**86,99 %**	**93,22 %**	**81,25 %**	**79,59 %**
SVM (Linear)	**100 %**	**97,75 %**	**100 %**	**88,14%**
K - Nearest Neighbors	**76,85 %**	**79,66 %**	**81,3 %**	**83,05 %**
Naïve Bayes	**56,5 %**	**57,63 %**	**77,24 %**	**79,66 %**

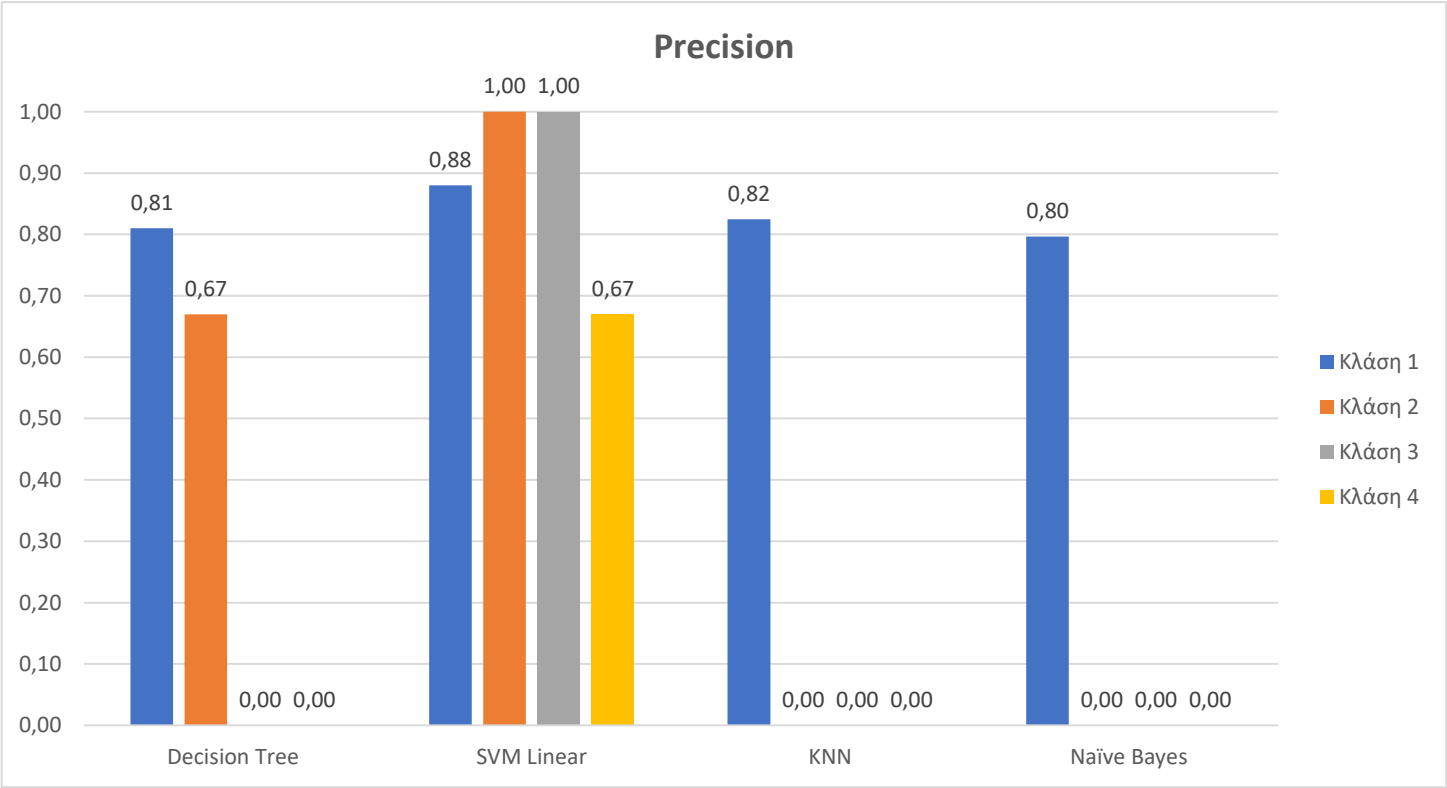
Πίνακας 41. Σύγκριση Αποτελεσμάτων του Accuracy

Στη συνέχεια παρουσιάζουμε όλες τις μετρικές απόδοσης των αλγορίθμων του Σεναρίου 2 για κάθε κλάση ξεχωριστά.

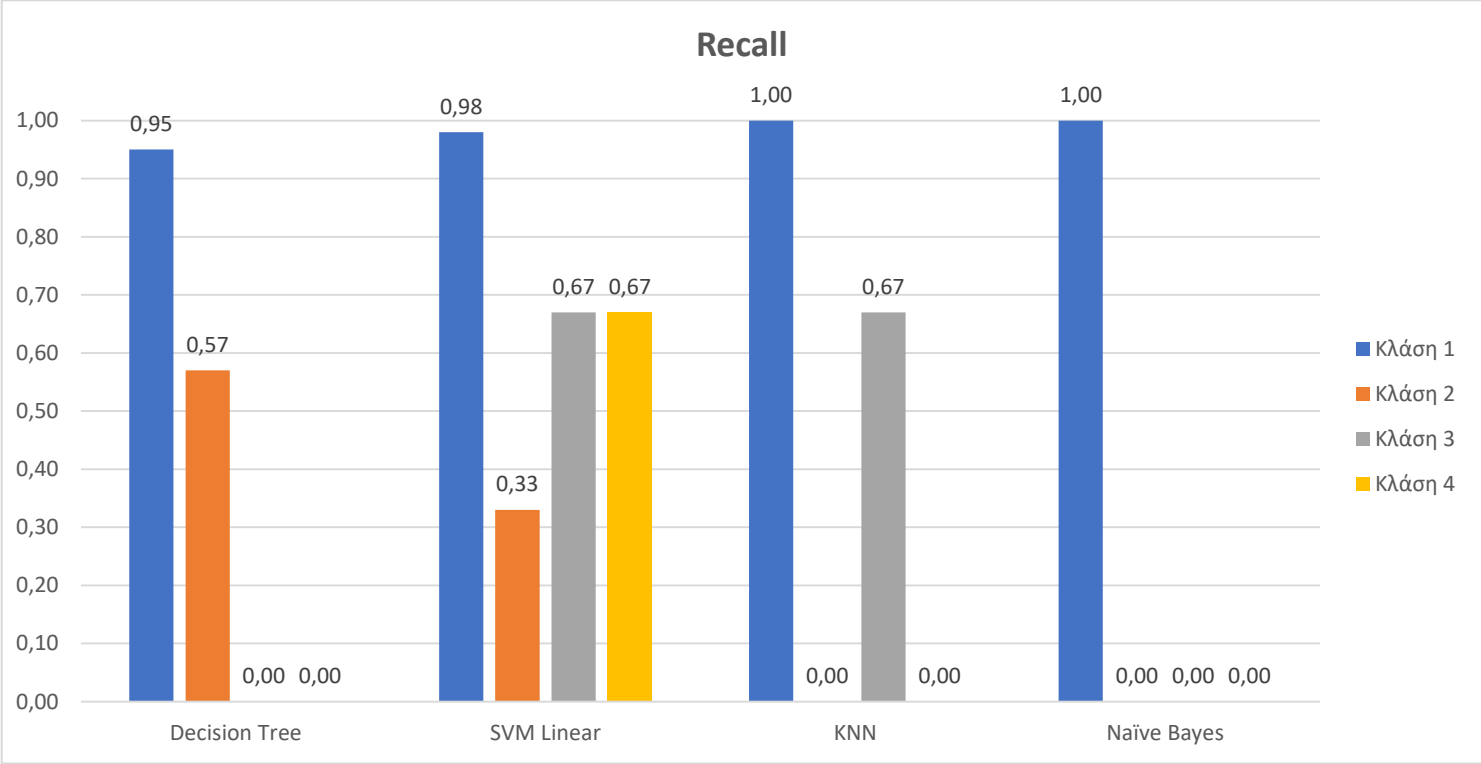
		Precision				Recall				F1			
		Class Labels				Class Labels				Class Labels			
Classification Model	**Accuracy**	**1**	**2**	**3**	**4**	**1**	**2**	**3**	**4**	**1**	**2**	**3**	**4**
Decision Tree	**79,59 %**	**0,81**	**0,67**	**0,00**	**0,00**	**0,95**	**0,57**	**0,00**	**0,00**	**0,88**	**0,62**	**0,00**	**0,00**
SVM (Linear)	**88,14 %**	**0,88**	**1,00**	**1,00**	**0,67**	**0,98**	**0,33**	**0,67**	**0,67**	**0,93**	**0,50**	**0,80**	**0,67**
K – Nearest Neighbors	**83,05 %**	**0,82**	**0,00**	**0,00**	**0,00**	**1,00**	**0,00**	**0,67**	**0,00**	**0,90**	**0,00**	**0,00**	**0,00**
Naïve Bayes	**79,66 %**	**0,80**	**0,00**	**0,00**	**0,00**	**1,00**	**0,00**	**0,00**	**0,00**	**0,90**	**0,00**	**0,00**	**0,00**

Πίνακας 42. Μετρικές Απόδοσης Αλγορίθμων - Σενάριο 2

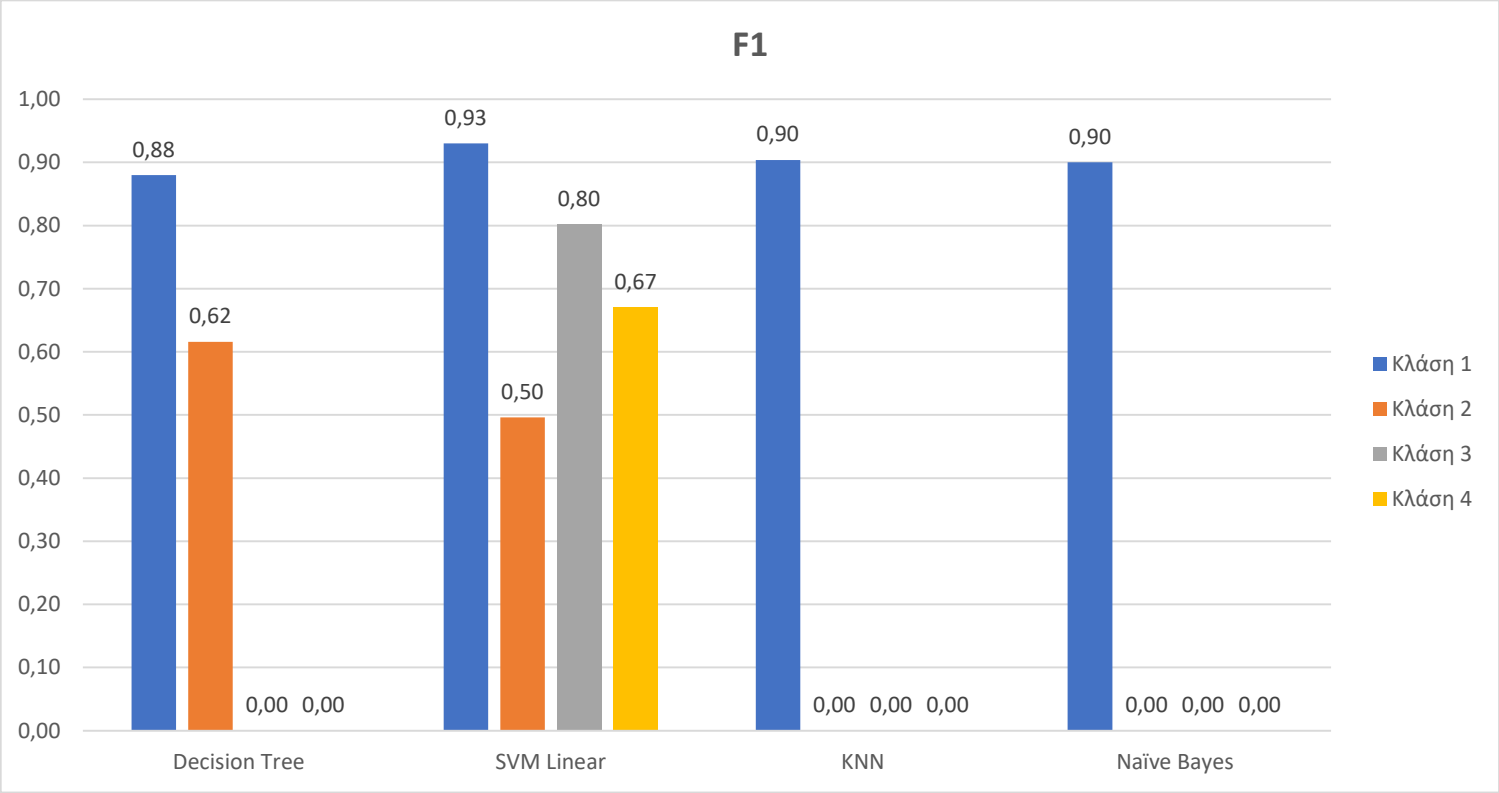
Ακολουθούν τα γραφήματα των μετρικών απόδοσης των αλγορίθμων.



Γράφημα 41. Precision of the models - Σενάριο 2

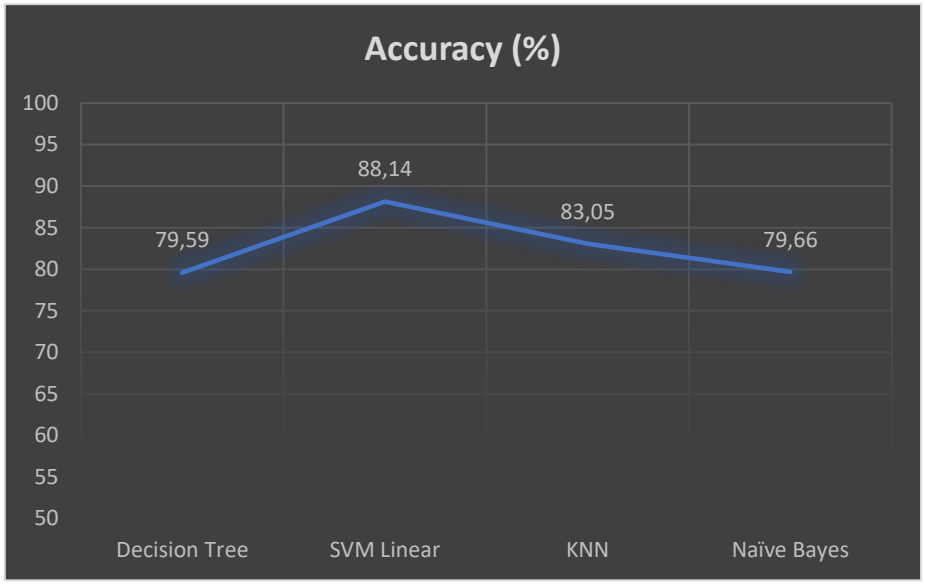


Γράφημα 42. - Recall of the models - Σενάριο 2



Γράφημα 43. F1 of the models - Σενάριο 2

Ακολουθεί το γράφημα του Accuracy των 4 μοντέλων ταξινόμησης που εφαρμόσαμε .



Γράφημα 44. Accuracy - Σενάριο 2

Φαίνεται ότι για τη δική μας ομαδοποίηση το μοντέλο που έχει καλύτερη απόδοση είναι ο SVM (Linear). Αυτό φαίνεται και από τις μετρικές του μοντέλου οι οποίες έχουν καλύτερες τιμές και στις 4 κλάσεις, ενώ τα υπόλοιπα μοντέλα δεν έχουν καλή απόδοση σε καμία κλάση εκτός από τη 1η.

Συνολικά και για τα δύο σενάρια που εκτελέσαμε διαπιστώνουμε ότι, για το σύνολο δεδομένων που μελετάμε, ο SVM Linear είχε καλύτερη απόδοση από τα υπόλοιπα μοντέλα.

ΚΕΦΑΛΑΙΟ 6

6.1 Συμπεράσματα

Στο κεφάλαιο 5 εφαρμόσαμε κάποια βασικά μοντέλα επιβλεπόμενης και μη επιβλεπόμενης μηχανικής μάθησης που χρησιμοποιούνται ευρέως στην ανάλυση κειμένου προκειμένου να εντοπίσουμε τα θέματα με τα οποία συσχετίζεται το τεχνικό χρέος, όρος ο οποίος συνδέεται με τη συντήρηση λογισμικού. Αυτό που προέκυψε από την k-means συσταδοποίηση και τα δέντρα απόφασης είναι ότι το τεχνικό χρέος σχετίζεται με την ποιότητα του πηγαίου κώδικα και τις μετρικές ποιότητας. Επίσης η συσσώρευση τεχνικού χρέους και οι επιπτώσεις του είναι ένα ζήτημα για το οποίο γίνεται εκτενής αναφορά σε ένα μεγάλο αριθμό άρθρων. Επιπρόσθετα ένα άλλο ζήτημα που εντοπίζεται ότι απασχολεί τους ερευνητές είναι η διαχείρισή του.

Αρχικά ξεκινήσαμε με τη δημιουργία ενός νέφους λέξεων (word cloud) προκειμένου να διαπιστώσουμε με οπτικό τρόπο τους όρους με τους οποίους συσχετίζεται το τεχνικό χρέος. Εν πρώτοις, διαπιστώσαμε ότι το dataset είχε μη σχετικά με το τεχνικό χρέος άρθρα. Στη συνέχεια με τη τεχνική του topic modeling καταφέραμε να χωρίσουμε τα δεδομένα σε κατηγορίες με βάση το topic, στο οποίο τα κατέταξε το μοντέλο, και από αυτά επιλέξαμε το υποσύνολο δεδομένων το οποίο αφορούσε το θέμα μας – το τεχνικό χρέος. Δηλαδή με το μοντέλο αυτό ξεχωρίσαμε από το σύνολο δεδομένων μας τα σχετικά από τα μη σχετικά άρθρα. Αυτό το επιβεβαιώνουμε και μετά από τη σχετική ανάγνωση των τίτλων και περιλήψεων του αρχικού dataset. Αυτό το επιχειρήσαμε και με τη βοήθεια του k-means clustering αλλά χωρίς επιτυχία.

Στη συνέχεια εφαρμόσαμε το topic modeling για να ομαδοποιήσουμε τα δεδομένα μας, τους προσδώσαμε μια ετικέτα κατηγορίας με βάση το topic και εκτελέσαμε 4 μοντέλα ταξινόμησης αλλά χωρίς ικανοποιητική ακρίβεια. Κατόπιν με την k-means συσταδοποίηση ομαδοποιήσαμε εκ νέου τα δεδομένα μας, τους προσδώσαμε μια νέα ετικέτα κατηγορίας και με τη δημιουργία μοντέλων ταξινόμησης καταφέραμε να τα κατηγοριοποιήσουμε με ικανοποιητική ακρίβεια, 97,75%. Επιπρόσθετα, με τη βοήθεια της ιεραρχικής συσταδοποίησης καταφέραμε να εντοπίσουμε, μεταξύ άλλων, εκείνα που αφορούν τη συντήρηση λογισμικού.

Τέλος με την δική μας εμπειρική ομαδοποίηση δημιουργήσαμε μοντέλα ταξινόμησης τα οποία εντοπίζουν άρθρα σχετικά με τη συντήρηση λογισμικού και των λοιπών κατηγοριών, που έχουμε εισάγει στο σύνολο δεδομένων μας, με αρκετά καλή ακρίβεια, 88,14%. Ωστόσο τα μοντέλα αυτά γενικά είχαν μικρότερη ακρίβεια σε σχέση με τα μοντέλα στα οποία η ετικέτα κατηγορίας προέκυψε από το k-means clustering.

6.2 Προτάσεις

Εμείς για την αντιμετώπιση του προβλήματος επιλέξαμε να επεξεργαστούμε τους τίτλους των άρθρων και κυρίως τις περιλήψεις τους. Στην επιλογή μας αυτή συντέλεσε καθοριστικά το γεγονός ότι στη βιβλιογραφική έρευνα που κάναμε οι περισσότερες μελέτες είχαν ως αντικείμενο εξόρυξης κειμένου τις περιλήψεις (Gulo et. al., 2015b). Εν τούτοις, θεωρούμε ότι η εφαρμογή των τεχνικών εξόρυξης και μηχανικής μάθησης σε ολόκληρο το σώμα των άρθρων θα μπορούσε να μας δώσει περισσότερη εις βάθος πληροφορία σχετικά με τις τάσεις της επιστήμης αναφορικά με το τεχνικό χρέος ή/και τις υποκατηγορίες/τομείς στους οποίους επεκτείνεται. Η δημιουργία ενός μοντέλου ταξινόμησης που να κατηγοριοποιεί τα πλήρη κείμενα των άρθρων θα μπορούσε να είναι ένα χρήσιμο εργαλείο σε κάποιον ερευνητή. Για το σκοπό αυτό αφιερώσαμε πολύ χρόνο για να μελετήσουμε τα πλήρη κείμενα, εργασία η οποία μας βοήθησε σημαντικά στο να κατανοήσουμε και να ερμηνεύσουμε τα αποτελέσματά μας αλλά κυρίως θα μας βοηθούσε να αξιολογήσουμε τα αποτελέσματα μιας τέτοιας δοκιμής. Ωστόσο για το πρόβλημα αυτό απαιτούνταν υψηλοί υπολογιστικοί πόροι τους οποίους δεν διαθέταμε για να το διερευνήσουμε περαιτέρω.

Τέλος, μια άλλη πρόταση έχει να κάνει με το σύνολο δεδομένων προς επεξεργασία, δηλαδή αν είχαμε τη δυνατότητα ελεύθερης πρόσβασης σε περισσότερες βιβλιογραφικές βάσεις δεδομένων θα μπορούσαμε να έχουμε ένα μεγαλύτερο σύνολο δεδομένων από διαφορετικές βιβλιοθήκες. Με τη χρήση των ανωτέρω αλγορίθμων επιβλεπόμενης και μη επιβλεπόμενης μάθησης σε ένα μεγαλύτερο σύνολο δεδομένων θα μπορούσαμε ίσως να εξάγουμε επιπλέον χρήσιμες πληροφορίες και νέα γνώση σχετικά με το αντικείμενο αυτό, καθώς και να ανακαλύψουμε περισσότερες υποκατηγορίες ή/και να εντοπίσουμε και άλλους τομείς κάτι που θα ήταν χρήσιμο σε κάποιον που αναζητά σχετική βιβλιογραφία πάνω στο αντικείμενο αυτό.

Βιβλιογραφικές Αναφορές

Aggarwal, C. (2015). *Data Mining:The textbook* . New York. Spinger

Alves, N. S. R., Mendes, T. S., de Mendonça, M. G., Spínola, R. O., Shull, F. & Seaman, C. (2016) Identification and management of technical debt: A systematic mapping study. Information Software Technology, Vol. 70, pp. 100–121. doi: 10.1016/j.infsof.2015.10.008

Anandarajan, M., Hill, C., Nolan, T. (2019). *Practical Text Analytics: Maximizing the value of text data.* Switzerland: Springer

Ampatzoglou, A., Ampatzoglou, A.; Avgeriou, P. and Chatzigeorgiou, A. (2015a). Establishing a Framework for Managing Interest in Technical Debt. *Proceedings of the Fifth International Symposium on Business Modeling and Software Design - BMSD,* 6-8 Ιουλίου 2015 (pp. 75-85), Μιλάνο: Boris Shishkov. Ανακτήθηκε στις 2/2/2021 από https://www.scitepress.org/Papers/2015/58857/58857.pdf

Ampatzoglou A., Ampatzoglou A., Chatzigeorgiou A., Avgeriou P. (2015b). The Financial Aspect of Managing Technical Debt: A Systematic Literature Review. In Information and Software Technology, 64, pp. 52-73, Elsevier.

Ampatzoglou, A., Michailidis, A., Sarikyriakidis, C., Ampatzoglou, A., Chatzigeorgiou, A. & Avgeriou, P. **(**2018**)**. A Framework for Managing Interest in Technical Debt: An Industrial Validation. Proceedings of the International Conference on Software Engineering, 2018, pp**.**115-124

Arun, R., Suresh, V., Madhavan, V. & Narasimha M. (2010). On finding the natural number of topics with latent dirichlet allocation: Some observations. In *Advances in knowledge discovery and data mining*, Mohammed J. Zaki, Jeffrey Xu Yu, Balaraman Ravindran and Vikram Pudi (eds.). Springer Berlin Heidelberg, pp. 391–402. Ανακτήθηκε στις 2/4/2021 από http://doi.org/10.1007/978-3-642-13657-3_43

Award, M. & Khanna, R. (2015). *Learning Machines. Theories, Concepts and Applications for Engineers and System Designers*. Apress

Cunningham, W. (1992). The WyCash Portfolio Management System. Addendum to the Proceedings on Object-oriented Programming Systems, Languages, and Applications (Addendum) pp. 29–30. New York, NY, SA: ACM. https://doi.org/10.1145/157709.157715

Deveaud, R., SanJuan, E. & Bellot. P. (2014). Accurate and effective latent concept modeling for ad hoc information retrieval. Vol 17 (No 1), pp: 61–84. Ανακτήθηκε στις 31/3/2021 από http://doi.org/10.3166/dn.17.1.61-84

Eisenberg, R. J (2012). A threshold based approach to technical debt. *ACM SIGSOFT Software Engineering Notes*, Vol 37 (No 2), p.1. Ανακτήθηκε στις 1/2/2021, από https://doi.org/10.1145/2108144.2108151

Feldman, R. and Sanger, J. (2007). *The text mining handbook: Advanced Approaches in Analyzing Unstructured Data*, New York: Cambridge University Press

Gopal S. P. (2018). Scientific Literature Text Mining and the Case for Open Access. Ανακτήθηκε στις 10/11/20 από https://peerj.com/preprints/2566v2/

Griffiths, T., and Steyvers. M. (2004). Finding scientific topics. *Proceedings of the National Academy of Sciences* Vol 101 (supply 1), pp. 5228–5235. Ανακτήθηκε στις 2/4/2021 από https://www.pnas.org/content/pnas/101/suppl_1/5228.full.pdf

Gulo, C. A. S. J., Rúbio T. R. P. M., Tabassum, S. and Prado S. G. D. (2015a). Mining Scientific Articles Powered by Machine Learning Techniques, ICCSW 2015.

Gulo, C. A. S. J., Rúbio T. R. P. M. (2015b). Text Mining Articles using R Language. Proceedings of the 10th Doctoral Symposium in Informatics Engineering ,2015. 1st Edition, pp. 60-68. Ανακτήθηκε στις 10/2/2021 από https://paginas.fe.up.pt/~prodei/dsie15/web/papers/dsie15_submission_10.pdf

Guo, Y., Seaman, C., Gomes, R., Cavalcanti, A., Tonin, G., Silva, F. Q. B. D & Siebra, C. (2011). Tracking technical debt. An exploratory case study. *27th IEEE International Conference on Software Maintenance ICSM*, pp.28–531.doi:10.1109/ICSM.2011.6080824

Gupta, M.K., Chandra, P. A. (2020). A Comprehensive survey of data mining. International Journal of Information Technology, 12, 1243–1257 (2020).

Han, J., Kamber, M. & Pei, J. (2011). *Data Mining. Concepts and techniques*. 3rd Ed. USA: Kaufman Publishers

Juan, C., Tian, X., Jintao, L., Yongdong, Z. & Tang Sheng, T. (2009). A density-based method for adaptive lda model selection. *Neurocomputing — 16th European Symposium on*

Artificial Neural Networks Vol. 72, pp.1775–1781. Ανακτήθηκε στις 2/4/2021 από http://doi.org/10.1016/j.neucom.2008.06.011

Lantz, B. (2013). *Machine Learning with R.* Birmingham: Packt Publishin Ltd.

Nie, B. & Sun, S. (2017).Using Text Mining to Identify Research Trends: A Case Study of Design Research. China: MPDI. Ανακτήθηκε στις 15/2/2021 από . https://www.mdpi.com/2076-3417/7/4/401

Nwanganga, F. & Chapple, M. (2020). *Practical Machine Learning in R*. Indianapolis USA: Wiley

Mayr, A. Plosch, R. and Korner, C. (2014). A Benchmarking-Based Model for Technical Debt Calculation. *14th International Conference on Quality Software*, 2014, (pp. 305–314). Allen, TX, USA: IEEE

Seaman, C. and Guo, Y. (2011). Measuring and Monitoring Technical Debt, *Advances in Computers*, Vol. 82, Elsevier, pp. 25–46. doi: 10.1016/B978-0-12-385512-1.00002-5

Small, H., Boyack, K. & Klavans, R. (2014). *Identifying emerging topics in science and technology*. Research Policy, Vol 43 (No 8), 1450–1467.

Siguenza-Guzman, L., Saquicela, V., Avila-Ordóñez, E., Vandewalle, J., & Cattrysse, D. (2015). Literature Review of Data Mining Applications in Academic Libraries. The Journal of Academic Librarianship, Vol 41 (No 4), 499–510. doi:10.1016/j.acalib.2015.06.007

Theobald, O. (2017). *Machine Learning For Absolute Beginners.*

Valerde-Berrocoso, J., Garrido-Arroyo, M., Burgos-Voleda, C. and Morales-Cevallos, M. (2020). Trends in Educational Research about e-Learning: A Systematic Literature Review (2009/2018). Vol. 12, Sustainability. Ανακτήθηκε στις 25/5/2021 από https://www.mdpi.com/2071-1050/12/12/5153

Ελληνόγλωσση

Βερύκιος, Β., Καγκλής. Β., Σταυρόπουλος. Η., (2015). *Η επιστήμη των δεδομένων μέσα από τη γλώσσα R*. Αθήνα. ΣΕΑΒ.

Μπρασινίκας Ι. (2020). Τεχνικές Εξόρυξης Δεδομένων με σκοπό την ταξινόμηση Καρδιαγγειακών Νόσων. Διπλωματική Εργασία, Ελληνικό Ανοικτό Πανεπιστήμιο.

Διαδικτυακοί τόποι

https://towardsdatascience.com/the-complete-guide-to-clustering-analysis-10fe13712787

https://cran.r-project.org/web/packages/ldatuning/vignettes/topics.html.

Παράρτημα

Παρατίθεται ο κώδικας όλων των δοκιμών και σεναρίων του κεφαλαίου 5

Κώδικας ενότητας 5.4

```
# WORD CLOUD OF THE TITLES

library(tm)

library(NLP)

library(stringi)

setwd("~/R/PS1")

#read data

r<-read.csv("/Users/User/Documents/R/PS1/PS1_564.csv")

rdt<-as.data.frame(r)

rdt2<-subset(rdt, select = - c(BibliographyType,ISBN,Author,
Journal,Volume, Number, Month, Pages,Note, URL,Address, Booktitle,
Chapter, Edition,Editor, Series,Publisher,
ReportType,Howpublished,Institution, Organizations,School, Annote,
Custom3,Custom4,Custom5 ))

paper.title <- VCorpus(VectorSource(rdt2$Title))

inspect(paper.title)

toSpace <- content_transformer(function(x,pattern) gsub(pattern, " ",
x))

tcorpu <- tm_map(paper.title, toSpace, "/|@|\\|")

tcorpu<-tm_map(tcorpu,content_transformer(tolower))

removeNumPunct <- function(x) gsub("[^[:alpha:][:space:]]*", "", x)
#remove anything other than English letters or space

tcorpu<-tm_map(tcorpu, content_transformer(removeNumPunct))

tcorpu<-tm_map(tcorpu,removeWords,stopwords("english"))

removeUnicode <- function(x) stri_replace_all_regex(x,"[^\x20-\x7E]","")

tcorpu <- tm_map(tcorpu, content_transformer(removeUnicode))

#remove extra words

tcorpu<-tm_map(tcorpu,removeWords, c("use","can","get","could","have",
"will","using",
"would","also","say","one","way","however","tell","much","need","take","
tend","even","like","particular","rather","different","said","well","mak
e","ask","come","end","first","two","help","often","may","might","see","
something","thing","point","post","look","right","now","think","another"
,"put","set","new","good","want","sure","kind","large","yes","day","etc"
,"queit","sinc","attempt","lack","seen","aware","little","ever","moreove
r","though","found","able","enough","far","early","earlier","away","achi
eve","draw","last","never","brief","bit","entire","brief","great","lot",
"doi","ieee","the","paper"))

tcorpu<-tm_map(tcorpu,removePunctuation, UCP=TRUE)
```

```
tcorpu<-tm_map(tcorpu,removeNumbers)

tcorpu<-tm_map(tcorpu,stripWhitespace) #remove extra whitespace

tdtm<-DocumentTermMatrix(tcorpu, control =
list(weighting=weightTfIdf,minWordLength=4, bounds = list(global = c(3,
Inf))))

inspect(tdtm)

#create word cloud

library(wordcloud)

library(RColorBrewer)

m<-as.matrix(tdtm)

v<-sort(colSums(m), decreasing=TRUE)

d<-data.frame(word=names(v), freq=v)

wordcloud(d$word, d$freq,random.order=FALSE,
rot.per=0.3,scale=c(4,.5),max.words=101,colors=brewer.pal(8,"Dark2"))

title(main="Wordcloud of the titles", font.main=1, cex.main=1.5)
```

Κώδικας ενότητας 5.5

```
# 1.TOPIC MODELING 564 CSV IN ABSTRACTS

library(tm) #required for text mining

library(topicmodels)

library(RColorBrewer)

library(NLP)

library(lda) # lattent dirichlet allocation

library(ldatuning)#to find number of topics

library(wordcloud)# to make a wordcloud

library(quanteda) #required for lattent dirichlet allocation function

library(ggplot2)

library(stringi)

setwd("~/R/564")

#read data

r<-read.csv("/Users/User/Documents/R/564/PS1_564.csv")

rdt<-as.data.frame(r)

rdt2<-subset(rdt, select = - c(BibliographyType,ISBN,Author,
Journal,Volume, Number, Month, Pages,Note, URL,Address, Booktitle,
Chapter, Edition,Editor, Series,Publisher,
ReportType,Howpublished,Institution, Organizations,School, Annote,
Custom3,Custom4,Custom5 ))

names(rdt2)[4]="Abstract"

View(rdt2)
```

```
paper.abstract <- VCorpus(VectorSource(rdt2$Abstract))

inspect(paper.abstract)

toSpace <- content_transformer(function(x,pattern) gsub(pattern, " ",
x))

corpu <- tm_map(paper.abstract, toSpace, "/|@|\\|")

corpu<-tm_map(corpu,content_transformer(tolower))

removeNumPunct <- function(x) gsub("[^[:alpha:][:space:]]*", "", x)
#remove anything other than English letters or space

corpu<-tm_map(corpu, content_transformer(removeNumPunct))

corpu<-tm_map(corpu,removeWords,stopwords("english"))

removeUnicode <- function(x) stri_replace_all_regex(x,"[^\x20-\x7E]","")

corpu <- tm_map(corpu, content_transformer(removeUnicode))

#remove extra words

mystopwords<-
c("show","finally","although","addition","four","several","better","ther
efore","significant","springer","engineering","research","case","systems
","architecture","elsevier","existing","types","issues","identify","iden
tified","work","within","propose","proposed","business","system","projec
t","approaches","approach","debt","related","atd","technical","useful","
years","understand","study","satd","software","use","used","using","can"
,"get","could","have","due","will","would","also","say","one","way","amo
ng","thus",
"however","future","based","tell","much","many","need","take","tend","ev
en","like","particular","rather","different","said","well","make","ask",
"come","end","first","two","help","often","may","might","see","something
","thing","point","post","look","right","now","think","another","put","s
et","new","good","want","sure","kind","large","yes","day","etc","quit","
since","attempt","lack","seen","awar","little","ever","moreover","though
","found","able","enough","far","early","away","achieve","draw","last","
never","brief","bit","entire","brief","great","lot","doi","the","paper",
"acm","ieee","three")

mystopwords<-sort(mystopwords)

corpu<-tm_map(corpu,removeWords, mystopwords)

corpu<-tm_map(corpu,removePunctuation, UCP=TRUE)

corpu<-tm_map(corpu,removeNumbers)

corpu<-tm_map(corpu,stripWhitespace) #remove extra whitespace

dtm<-DocumentTermMatrix(corpu, control =
list(weighting=weightTf,stopwords=T,minWordLength=c(4,15), bounds =
list(global = c(3, Inf))))

dtm<-removeSparseTerms(x = dtm, sparse = 0.95)

inspect(dtm)

#CheCk for 0 raw sums in the articles

raw.sum=apply(dtm,1,FUN=sum)

#remove artiles with 0 sum

dtm=dtm[raw.sum!=0,]
```

```
inspect(dtm)
#find optimum number of topics
#Arun2020 maximize, CaoJuan minimize, Griffiths minimize
optimal.topics <- FindTopicsNumber(
  dtm ,
  topics = c(2:10),
  metrics = c("Griffiths2004", "CaoJuan2009", "Arun2010","Deveaud2014"),
  method = "Gibbs",
  control = list(seed = 12345),
  mc.cores = 4L,
  verbose = TRUE)
FindTopicsNumber_plot(optimal.topics)
set.seed(222)
m=LDA(dtm, method="Gibbs", k=6, control=list(alpha=0.1))
#for a specific topic we can find topwords
topic = 6
words = posterior(m)$terms[topic, ]
topwords = head(sort(words, decreasing = T), n=50)
head(topwords)
#find 15 terms of every topic
terms(m,15)
library(tidytext)
ap_topics <- tidy(m, matrix = "beta")
ap_topics
library(ggplot2)
library(dplyr)
#plots
ap_top_terms <- ap_topics %>%
  group_by(topic) %>%
  top_n(10, beta) %>%
  ungroup() %>%
  arrange(topic, -beta)
ap_top_terms %>%
  mutate(term = reorder_within(term, beta, topic)) %>%
  ggplot(aes(beta, term, fill = factor(topic))) +
  geom_col(show.legend = FALSE) +
  facet_wrap(~ topic, scales = "free") +
  scale_y_reordered()
```

```
#end of graphs

#topic propabilities

topicProbabilities <- as.data.frame(m@gamma)

#probabilities for the articles to the topics

write.csv(topicProbabilities,file=paste("LDAGibbs",
6,"TopicProbabilities.csv"))

#the top 6 terms for every topic

ldaOut.terms <- as.matrix(terms(m,10))

#write out results

#docs to topics

ldaOut.topics <- as.matrix(topics(m))

write.csv(ldaOut.topics, file=paste("LDAGibbs",6,"DocsToTopics.csv"))

#keep only topics 3,4,5 : 305 abstracts and then apply a classification
model

matrixdtm<-as.matrix(dtm)

datadtm<-as.data.frame(matrixdtm)

library(readr)

LDAGibbs_6_DocsToTopics <- read_csv("LDAGibbs 6 DocsToTopics.csv")

#rename columns

names(LDAGibbs_6_DocsToTopics)[2]="Category"

#create corpus

newcorpus<-data.frame(datadtm,LDAGibbs_6_DocsToTopics$Category)

View(newcorpus)

#rename last column

names(newcorpus)[185]="Topic"

View(newcorpus)

#remove rows in topics 1,2,6

corpus<-newcorpus[ !(newcorpus$Topic %in% c(1,2,6)), ]

View(corpus)

Corpus<-corpus[,-185]  #remove column named Topic

View(Corpus) # data set 305 entries

#-------------------------------------------------------------------------#
```

Κώδικας ενότητας 5.6

```
#FILE: KMEANS CLUSTERING PS1_564 CSV

library(tm)

library(stats)

library(dplyr)
```

```
library(ggplot2)

library(ggfortify)

library(NbClust)

library(stringi)

library(cluster)

setwd("~/R/PS1")

#read data

r<-read.csv("/Users/User/Documents/R/PS1/PS1_564.csv")

rdt<-as.data.frame(r)

rdt2<-subset(rdt, select = - c(BibliographyType,ISBN,Author,
Journal,Volume, Number, Month, Pages,Note, URL,Address, Booktitle,
Chapter, Edition,Editor, Series,Publisher,
ReportType,Howpublished,Institution, Organizations,School, Annote,
Custom3,Custom4,Custom5 ))

names(rdt2)[4]="Abstract"

View(rdt2)

paper.abstract <- VCorpus(VectorSource(rdt2$Abstract))

inspect(paper.abstract)


toSpace <- content_transformer(function(x,pattern) gsub(pattern, " ",
x))

corpu <- tm_map(paper.abstract, toSpace, "/|@|\\|")

corpu<-tm_map(corpu,content_transformer(tolower))

removeNumPunct <- function(x) gsub("[^[:alpha:][:space:]]*", "", x)
#remove anything other than English letters or space

corpu<-tm_map(corpu, content_transformer(removeNumPunct))

corpu<-tm_map(corpu,removeWords,stopwords("english"))

removeUnicode <- function(x) stri_replace_all_regex(x,"[^\x20-\x7E]","")

corpu <- tm_map(corpu, content_transformer(removeUnicode))

#remove extra words

corpu<-tm_map(corpu,removeWords,
c("architecture","existing","business","project","approaches","approach"
,"significant","three","debt","technical","useful","years","understand",
"study","software","use","used","using","can","get","could","have","due"
, "will", "would" ,"also","say","one","way",
"however","future","based","tell","much","need","take","tend","even","li
ke","particular","rather","different",
"said","well","make","ask","come","end","first","two","help","often","ma
y","might","see","something","thing","point","post","look","right","now"
,"think","another","put","set","new","good","want","sure","kind","large"
,"yes","day","etc","quit","sinc","attempt","lack","seen","aware","little
","show","therefore","ever","moreover","though","found","able",
"enough","far","earli","away","achieve","draw",
"last","never","brief","bit","entire","brief","great","lot","doi","ieee"
,"the","paper","acm","elsevier"))
```

```
corpu<-tm_map(corpu,removePunctuation, UCP=TRUE)

corpu<-tm_map(corpu,removeNumbers)

corpu<-tm_map(corpu,stripWhitespace) #remove extra whitespace

dtm<-DocumentTermMatrix(corpu, control =
list(stropwords=T,weighting=weightTf,minWordLength=c(4,15), bounds =
list(global = c(3, Inf))))

dtm<-removeSparseTerms(x = dtm, sparse = 0.9)

inspect(dtm)

datam<-as.matrix(dtm)

myData<-as.data.frame(datam)

View(myData)

library(ggpubr)

library(factoextra)

fviz_nbclust(myData, kmeans, method = "wss")

fviz_nbclust(myData, kmeans, method = "silhouette")

fviz_nbclust(myData, kmeans, method = "gap_stat")

# k-means clustering

set.seed(1234)

KM=kmeans(myData, 4, nstart=30) #nstart to change initial centers

# plot clusters

fviz_cluster(KM, data = myData, repel = TRUE,  main="K-Means
Clustering") #ellipse.type="norm" to print better shape

#results

KM$size

KM$centers

KM$cluster

#find the articles in the clusters

myData[KM$cluster==1,]

myData[KM$cluster==2,]

myData[KM$cluster==3,]

myData[KM$cluster==4,]

myData[KM$cluster==5,]

myData[KM$cluster==6,]

myData[KM$cluster==7,]

#names of raws of cluster 1

row.names(myData[KM$cluster==1,])

row.names(myData[KM$cluster==2,])

row.names(myData[KM$cluster==3,])

row.names(myData[KM$cluster==4,])
```

```
row.names(myData[KM$cluster==5,])
row.names(myData[KM$cluster==6,])
row.names(myData[KM$cluster==7,])
cluster1<-myData[KM$cluster==1,]
cluster2<-myData[KM$cluster==2,]
cluster3<-myData[KM$cluster==3,]
cluster4<-myData[KM$cluster==4,]
cluster5<-myData[KM$cluster==5,]
cluster6<-myData[KM$cluster==6,]
cluster7<-myData[KM$cluster==7,]
c1<-colSums(cluster1)
c2<-colSums(cluster2)
c3<-colSums(cluster3)
c4<-colSums(cluster4)
c5<-colSums(cluster5)
c6<-colSums(cluster6)
c7<-colSums(cluster7)
allsums<-data.frame(c1,c2,c3,c4,C5,C6,C7)
allsums
#quality of partinioning
bss<-KM$betweenss
tss<-KM$totss
qual<-(bss/tss)*100
qual
#Dunn Index
library(fpc)
# Statistics for k-means clustering
km_stats <- cluster.stats(dist(myData),  KM$cluster)
# Dun index
km_stats$dunn
```

Κώδικας ενότητας 5.7

```
#-----------------K MEANS CLUSTERING-------------#
#remove comlumns with zero values
myData<-Corpus[,colSums(Corpus[])>0]
View(myData)
View(myData)
```

```
library(ggpubr)
library(factoextra)
fviz_nbclust(myData, kmeans, method = "wss")
fviz_nbclust(myData, kmeans, method = "silhouette")
fviz_nbclust(myData, kmeans, method = "gap_stat")
library(cluster)
set.seed(222)
KM=kmeans(myData,4, nstart=25) #nstart to change initial centers
fviz_cluster(KM, data = myData, repel = TRUE,  main="K-Means Clustering")
#ellipse.type="norm" to print better shape
KM$size
cluster1<-myData[KM$cluster==1,]
cluster2<-myData[KM$cluster==2,]
cluster3<-myData[KM$cluster==3,]
cluster4<-myData[KM$cluster==4,]
c1<-colSums(cluster1)
c2<-colSums(cluster2)
c3<-colSums(cluster3)
c4<-colSums(cluster4)
allsums<-data.frame(c1,c2,c3,c4)
allsums
#names of raws of cluster 1
row.names(myData[KM$cluster==1,])
row.names(myData[KM$cluster==2,])
row.names(myData[KM$cluster==3,])
row.names(myData[KM$cluster==4,])
#quality of partinioning
bss<-KM$betweenss
tss<-KM$totss
qual<-(bss/tss)*100
qual #the higher the percentage the better the score
#Dunn Index as max as possible
library(fpc)
# Statistics for k-means clustering
km_stats <- cluster.stats(dist(myData),  KM$cluster)
# Dun index
km_stats$dunn
#------Create new data frame with category to myData adding a column with
cluster category----#
```

```
mydata<-data.frame(myData, KM$cluster)
View(mydata)
#rename column KM.cluster
names(mydata)[179]="cg"
View(mydata)
write.csv(mydata, "/Users/User/Documents/R/564/myData.csv")
View(myData)
```

Κώδικας ενότητας 5.8

```
Ο dtm του κώδικα της ενότητας 5.6
datam<-as.matrix(dtm)
myData<-as.data.frame(datam)
View(myData)

# Compute distance matrix
m<-as.matrix(myData)
distMatrix <- dist(m, method="euclidean")
#-------------------------------------------------#
#Optimal number of clusters
fviz_nbclust(myData, FUN = hcut, method = "wss")
fviz_nbclust(myData, FUN = hcut, method = "silhouette")
gap_stat <- clusGap(myData, FUN = hcut, nstart = 25, K.max = 10, B = 50)
fviz_gap_stat(gap_stat)
#-------------------------------------------------#
# AGGLOMERATIVE CLUSTERING
# METHOD WARD D2
hc_wardD2 <- hclust(distMatrix, method = "ward.D2" )
plot(hc_wardD2, cex = 0.9)
rect.hclust(hc_wardD2, k = 6, border = 2:5)
# Cut tree into 6 groups
sub_wardD2 <- cutree(hc_wardD2, k = 6)
# Number of members in each cluster
table(sub_wardD2)
fviz_cluster(list(data = myData, cluster = sub_wardD2), main ="Cluster
Plot: Method Ward D2")
```

Κώδικας ενότητας 5.9

```
library(tm) #required for text mining
library(topicmodels)
library(RColorBrewer)
library(lda) # lattent dirichlet allocation
library(ldatuning)#to find number of topics
library(wordcloud)# to make a wordcloud
library(quanteda) #required for lattent dirichlet allocation function
library(ggplot2)
library(stringi)
setwd("~/R/PS1")
#read data
r<-read.csv("/Users/User/Documents/R/PS1/PS1.csv")
rdt<-as.data.frame(r)
rdt2<-subset(rdt, select = - c(BibliographyType,ISBN,Author, Journal,Volume, Number, Month, Pages,Note, URL,Address, Booktitle, Chapter, Edition,Editor, Series,Publisher, ReportType,Howpublished,Institution,Organizations,School,Annote,Custom3, Custom4,Custom5 ))
names(rdt2)[4]="Abstract"
View(rdt2)
paper.abstract <- VCorpus(VectorSource(rdt2$Abstract))
inspect(paper.abstract)
toSpace <- content_transformer(function(x,pattern) gsub(pattern, " ", x))
corpu <- tm_map(paper.abstract, toSpace, "/|@|\\|")
corpu<-tm_map(corpu,content_transformer(tolower))
removeNumPunct <- function(x) gsub("[^[:alpha:][:space:]]*", "", x) #remove anything other than English letters or space
corpu<-tm_map(corpu, content_transformer(removeNumPunct))
corpu<-tm_map(corpu,removeWords,stopwords("english"))
removeUnicode <- function(x) stri_replace_all_regex(x,"[^\x20-\x7E]","")
corpu <- tm_map(corpu, content_transformer(removeUnicode))
#remove extra words
corpu<-tm_map(corpu,removeWords,
c("show","several","engineering","research","case","systems","architectu
re","existing","types","issues","identify","identified","work","longterm
","propose","proposed","business","system","project","approaches","appro
ach","debt","related","atd","technical","useful","years","understand","s
tudy","satd","software","use","used","using","can","get","could","have",
"due","will","would","also","say","one","way","however","future","based"
,"tell","much","need","take","tend","even","like","particular","rather",
"different","said","well","make","ask","come","end","first","two","help"
,"often","may","might","see","something","thing","point","post","look","
right","now","think","another","put","set","new","good","want","sure","k
```

```
ind","large","yes","day","etc","quit","sinc","attempt","lack","seen","aw
ar","littl","ever","moreover","though","found","able","enough","far","ea
rli","away","achieve","draw",
"last","never","brief","bit","entire","brief","great","lot","doi","the",
"paper","acm","ieee"))

corpu<-tm_map(corpu,removePunctuation, UCP=TRUE)

corpu<-tm_map(corpu,removeNumbers)

corpu<-tm_map(corpu,stripWhitespace) #remove extra whitespace

dtm<-
DocumentTermMatrix(corpu,control=list(weighting=weightTf,stopwords=T,min
WordLength=c(4,15), bounds = list(global = c(3, Inf))))

dtm<-removeSparseTerms(x = dtm, sparse = 0.99)

inspect(dtm)

#CheCk for 0 raw sums in the articles

raw.sum=apply(dtm,1,FUN=sum)

#remove artiles with 0 sum

dtm=dtm[raw.sum!=0,]

inspect(dtm)

set.seed(1)

m=LDA(dtm, method="Gibbs", k=4, control=list(alpha=0.1))

#for a specific topic we can find topwords

topic = 4

words = posterior(m)$terms[topic, ]

topwords = head(sort(words, decreasing = T), n=50)

head(topwords)

#find 15 terms of every topic

terms(m,15)

library(tidytext)

ap_topics <- tidy(m, matrix = "beta")

ap_topics

library(ggplot2)

library(dplyr)

#plots

ap_top_terms <- ap_topics %>%

  group_by(topic) %>%

  top_n(10, beta) %>%

  ungroup() %>%

  arrange(topic, -beta)

ap_top_terms %>%

  mutate(term = reorder_within(term, beta, topic)) %>%
```

```
  ggplot(aes(beta, term, fill = factor(topic))) +
  geom_col(show.legend = FALSE) +  facet_wrap(~ topic, scales = "free") +
scale_y_reordered()
#end of graphs
#topic propabilities
topicProbabilities <- as.data.frame(m@gamma)
#probabilities for the articles to the topics
write.csv(topicProbabilities,file=paste("LDAGibbs",
4,"TopicProbabilities.csv"))
#the top 10 terms for every topic
ldaOut.terms <- as.matrix(terms(m,10))
#write out results
#docs to topics
ldaOut.topics <- as.matrix(topics(m))
write.csv(ldaOut.topics,file=paste("LDAGibbs",4,"DocsToTopics.csv"))
```

Σενάριο 1

```
#----------------------APPLY CLASSIFICATION MODELS -------------------#
#--------------------------- 1. DECISION TREE   ---------------------#
#--------------------------- 2. SVM  --------------------------------#
#--------------------------- 4. NAIVE BAYES   -----------------------#
#--------------------------- 3. KNN     -----------------------------#
#---------------------1. DECISION TREE----------------------#
library(party)
library(zoo)
library(caret)

mydata$cg<-as.factor(mydata$cg)
set.seed(1234)
intrain<-createDataPartition(mydata$cg,p=0.8, list=F)
training<-mydata[intrain,]
testing<-mydata[-intrain,]
tree<-ctree(cg~.,data=training,   controls=ctree_control(mincriterion   =
0.8, minsplit=40))
plot(tree)
tree
#Decision tree with rpart and training data
library(rpart)
```

```
library(rpart.plot)
tree1<-rpart(cg~., training)
rpart.plot(tree1)
rpart.plot(tree1, extra=1)#
rpart.plot(tree1, extra=2)
rpart.plot(tree1, extra=4)# show probabilities
#Prediction
predict(tree1,testing)#gives prediction for each article
#misclassificatin error for training data
tab<-table(predict(tree), training$cg)
print(tab)
1-sum(diag(tab))/sum(tab)
sum(diag(tab))/sum(tab)
#misclassification error for testing data
testPred<-predict(tree,newdata=testing)
tab<-table(testPred,testing$cg )
print(tab)
library(gmodels)
CrossTable(testPred,testing$cg,prop.chisq=F,  prop.t=F,dnn=c("Predicted  "
, "Actual"))
confusionMatrix(table(Predicted=testPred, Actual=testing$cg) )
#------------2. SVM----------------#
library(e1071) #for SVM
set.seed(12345)
intrain<-createDataPartition(mydata$cg,p=0.7, list=F)
training<-mydata[intrain,]
testing<-mydata[-intrain,]
#------------ SVM LINEAR -----------------#
mymodel_linear<-svm(cg~., data=training, kernel="linear")
summary(mymodel_linear)
pred_linear<-predict(mymodel_linear,testing)
tab<-table(Predicted=pred_linear, Actual=testing$cg)
tab
confusionMatrix(table(pred_linear,testing$cg))
CrossTable(pred_linear,testing$cg,prop.chisq=F,prop.t=F,dnn=c("Predicted
","Actual"))
#accuracy of training data
pred_train<-predict(mymodel_linear, training)
confusionMatrix(table(pred_train,training$cg))
```

```
#-----------------------------------------#
#------------3. KNN Model------------------#
library(caret)
library(pROC)
library(mlbench)
library(class)
library(tidyverse)
set.seed(1)
intrain<-createDataPartition(mydata$cg,p=0.8, list=F)
training<-mydata[intrain,]
testing<-mydata[-intrain,]
train_labels<-as.factor(pull(training, cg))
test_labels<-as.factor(pull(testing, cg))
training<-data.frame(select(training, - cg))
testing<-data.frame(select(testing, -cg))
knn_pred<-knn(train=training, test=testing, cl=train_labels,k=4,prob=T)
knn_pred
tb<-table(test_labels, knn_pred)
tb
sum(diag(tb))/nrow(testing)
confusionMatrix(table(Predicted=knn_pred, Actual=test_labels ))
CrossTable(knn_pred,test_labels,prop.chisq=F, prop.t=F, dnn=c("Predicted
" , "Actual"))
#accuracy of training data
knn_pred_train<-knn(train=training,
test=training,cl=train_labels,k=4,prob=T)
confusionMatrix(table(Predicted=knn_pred_train, Actual=train_labels ))
#--------------4 NAIVE BAYES --------------#
myCntrl<-trainControl(method="repeatedcv" , number=10, repeats=10 )
set.seed(1234)
tc<-train(cg ~., data=training, method="nb", trControl=myCntrl)
tc
#prediction
pre_nb<-predict(tc, newdata=testing)
table(pre_nb,testing$cg)
#confusion matrix
confusionMatrix(table(pre_nb,testing$cg) )
#accuracy of training data
prenb_train<-predict(tc,newdata=training)
```

```
confusionMatrix(table(prenb_train,training$cg))
```

Σενάριο 2

```
#-----------1  DECISION TREE  ------------------#
library(party)
library(zoo)
tree<-ctree(articles.Label~.,data=training,
controls=ctree_control(mincriterion = 0.8, minsplit=40))
plot(tree)
tree
#probabilities for the category
predict(tree, testing,type="prob")
predict(tree, testing)
#Decision tree with rpart
library(rpart)
library(rpart.plot)

tree1<-rpart(articles.Label~., training)
rpart.plot(tree1)
rpart.plot(tree1, extra=1)
rpart.plot(tree1, extra=2)
rpart.plot(tree1, extra=4)
#Prediction
predict(tree1,testing)
#misclassificatin error for training data
tab<-table(predict(tree), training$articles.Label)
print(tab)
1-sum(diag(tab))/sum(tab)
sum(diag(tab))/sum(tab)
#alternative way is to calculate confusion matrix
confusionMatrix(table(predict(tree),training$articles.Label))
library(gmodels)
CrossTable(predict(tree),training$articles.Label,prop.chisq=F,  prop.t=F,
dnn=c("Predicted " , "Actual"))
#misclassification error for testing data
testPred<-predict(tree,newdata=testing)
tab<-table(testPred,testing$articles.Label )
print(tab)
```

```
1-sum(diag(tab))/sum(tab)
sum(diag(tab))/sum(tab)
#alternative way is to calculate confusion matrix
confusionMatrix(table(testPred,testing$articles.Label))
library(gmodels)
CrossTable(testPred,testing$articles.Label,prop.chisq=F,        prop.t=F,
dnn=c("Predicted " , "Actual"))
#------------------2  SVM  ---------------------#
library(e1071) #for SVM
set.seed(222)
intrain<-createDataPartition(newcorpus$articles.Label,p=0.8, list=F)
training<-newcorpus[intrain,]
testing<-newcorpus[-intrain,]
#------------------- SVM LINEAR -------------------#
mymodel_linear<-svm(articles.Label~., data=training, kernel="linear")
summary(mymodel_linear)
pred_linear<-predict(mymodel_linear,testing)
tab<-table(Predicted=pred_linear, Actual=testing$articles.Label)
tab
confusionMatrix(table(pred_linear,testing$articles.Label))
CrossTable(pred_linear,testing$articles.Label,prop.chisq=F,     prop.t=F,
dnn=c("Predicted " , "Actual"))
#accuracy of training data
pred_train<-predict(mymodel_linear, training)
confusionMatrix(table(pred_train,training$articles.Label))
#--------------3 KNN MODEL -------------------------#
library(caret)
library(pROC)
library(mlbench)
library(class)
library(tidyverse)
set.seed(222)
intrain<-createDataPartition(newcorpus$articles.Label,p=0.8, list=F)
training<-newcorpus[intrain,]
testing<-newcorpus[-intrain,]
train_labels<-as.factor(pull(training, articles.Label))
test_labels<-as.factor(pull(testing, articles.Label))
training<-data.frame(select(training, - articles.Label))
testing<-data.frame(select(testing, -articles.Label))
```

```
knn_pred<-knn(train=training, test=testing, cl=train_labels,k=3,prob=T)

knn_pred

tb<-table(test_labels, knn_pred)

tb

sum(diag(tb))/nrow(testing)

confusionMatrix(table(Predicted=knn_pred, Actual=test_labels))

CrossTable(knn_pred,test_labels,prop.chisq=F, prop.t=F,dnn=c("Predicted "
, "Actual"))

# accuracy of training data

knn_pred_train<-knn(train=training,
test=training,cl=train_labels,k=3,prob=T)

confusionMatrix(table(knn_pred_train, train_labels))

#-------------------------------------------------#

#------------------4 NAIVE BAYES ----------------------#

library(e1071)

library(caret)

library(lattice)

set.seed(222)

intrain<-createDataPartition(newcorpus$articles.Label,p=0.8, list=F)

training<-newcorpus[intrain,]

testing<-newcorpus[-intrain,]

myCntrl<-trainControl(method="repeatedcv" ,number=10, repeats=10)

tc<-train(articles.Label      ~.,      data=training,      method="nb",
trControl=myCntrl)

tc

#prediction

prenb<-predict(tc, newdata=testing)

confusionMatrix(table(Predicted=prenb,Actual=testing$articles.Label))

CrossTable(prenb,testing$articles.Label,prop.chisq=F,
prop.t=F,dnn=c("Predicted " , "Actual"))

# accuracy of training data

prenb_train<-predict(tc,newdata=training)

confusionMatrix(table(prenb_train,training$articles.Label))
```