ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΙΓΑΙΟΥ

ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΑΚΩΝ ΚΑΙ ΕΠΙΚΟΙΝΩΝΙΑΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

ΠΡΟΓΡΑΜΜΑ ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ

ΠΛΗΡΟΦΟΡΙΑΚΑ ΚΑΙ ΕΠΙΚΟΙΝΩΝΙΑΚΑ ΣΥΣΤΗΜΑΤΑ

ΜΕΤΑΠΤΥΧΙΑΚΗ ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Δια – τομεακή Αναγνώριση Γένους Συγγραφέα

Cross – domain Author Gender Prediction

του

Παναγούλια Κωνσταντίνου

Α.Μ.: 3252020035

Επιβλέπων:

Σταματάτος Ευστάθιος

Καθηγητής Πανεπιστημίου Αιγαίου

Σάμος, Ιούνιος 2022

Η σελίδα αυτή είναι σκόπιμα κενή.

Πρόλογος

Η συγκεκριμένη διπλωματική εργασία πραγματοποιήθηκε εντός των πλαισίων του μεταπτυχιακού προγράμματος σπουδών «Πληροφοριακά και Επικοινωνιακά Συστήματα» του Τμήματος Μηχανικών Πληροφοριακών και Επικοινωνιακών Συστημάτων του Πανεπιστημίου Αιγαίου, κατά το χειμερινό και εαρινό εξάμηνο του Ακαδημαϊκού Έτους 2021 – 2022. Επιβλέπων καθηγητής και καθοδηγητής της διπλωματικής εργασίας ήταν ο καθηγητής κ. Ευστάθιος Σταματάτος. Το θέμα της εργασίας είναι: «Cross – domain Author Gender Prediction».

Ευχαριστίες

Με το πέρας της διπλωματικής μου εργασίας, θα ήθελα να ευχαριστήσω θερμά όλους τους ανθρώπους που συνείσφεραν με το δικό τους τρόπο καθ' όλο το διάστημα της πορείας μου.

Αρχικά, θα ήθελα να ευχαριστήσω τον επιβλέποντα καθηγητή μου κ. Σταματάτο για την άψογη συνεργασία και την πολύτιμη βοήθεια και καθοδήγηση που μου προσέφερε κατά την διάρκεια της συγκεκριμένης εργασίας.

Επιπλέον, οφείλω να ευχαριστήσω και την οικογένειά μου για την πολύτιμη βοήθεια και συμπαράσταση που μου δίνουν, όσον αφορά τις σπουδές μου, αλλά και γενικώς.

Τέλος, θα ήθελα να ευχαριστήσω τους φίλους μου, οι οποίοι μου έδιναν δύναμη σε όλη την πορεία της διπλωματικής μου εργασίας στηρίζοντάς με και πιστεύοντας σε μένα.

Η σελίδα αυτή είναι σκόπιμα κενή.

Πίνακας περιεχομένων

Κατάλογος Εικόνων

Κατάλογος Πινάκων

Η σελίδα αυτή είναι σκόπιμα κενή.

Στους γονείς μου, Γιώργο & Αγγελική

Περίληψη

Η συγκεκριμένη εργασία πραγματεύεται την αναγνώριση του προφίλ συγγραφέα (Author Profiling) μέσα από μια διαδικασία αναπαράστασης εγγράφου (Document Representation) και χρήσης αλγορίθμων Μηχανικής Μάθησης (Machine Learning). Στόχος της είναι η ταξινόμηση των συγγραφέων ως προς το φύλο τους, εξετάζοντας παραλλαγές στη διαδικασία της αναπαράστασης του κειμένου.

Στο θεωρητικό μέρος της εργασίας, μελετώνται αρχικά κάποιες τεχνικές εξόρυξης γνώσης από έγγραφα (Text Mining), αναπαράστασης εγγράφου, καθώς και μετρικές αξιολόγησής τους. Παρατίθενται, επίσης, πληροφορίες για την φάση της προεπεξεργασίας των δεδομένων που εξάγονται από τη διαδικασία της αναπαράστασης, ώστε να μετατραπούν στην κατάλληλη μορφή για την ταξινόμησή τους από κάποιον αλγόριθμο μηχανικής μάθησης. Ειδική αναφορά γίνεται στον όρο της μηχανικής μάθησης, στα είδη αυτής, όπως και σε ορισμένους σημαντικούς αλγορίθμους που κατατάσσονται σε αυτό το πεδίο. Στη συνέχεια, περιγράφεται ο όρος της ταξινόμησης εγγράφου (Text Classification), οι διαφορετικοί αλγόριθμοι υλοποίησης και οι μετρικές αξιολόγησης. Τέλος, γίνεται λόγος για το θέμα της αναγνώρισης προφίλ συγγραφέα, για τις διάφορες εφαρμογές του, καθώς και για τη μελέτη του ζητήματος στο πλαίσιο του διαγωνισμού PAN.

Στο πειραματικό μέρος, εφαρμόζονται κάποιες από τις τεχνικές αναπαράστασης και ταξινόμησης κειμένου που αναφέρθηκαν για υλοποίηση συστήματος αναγνώρισης προφίλ συγγραφέα στη συλλογή δεδομένων CMCC Corpus. Η αναπαράσταση γίνεται σύμφωνα με τη θεωρία N – Gram, σε επίπεδο χαρακτήρων, λέξεων και συντακτικών όρων, για ένα εύρος αριθμών N. Η αναγνώριση του συγγραφικού προφίλ των συγγραφέων με βάση το φύλο πραγματοποιείται με αλγόριθμο μηχανικής μάθησης και αποτελεί πρόβλημα ταξινόμησης σε 2 κλάσεις (Male – Αρσενικό, Female – Θηλυκό).

Λέξεις-κλειδιά: *Document Representation, Text Mining, Text Classification, Machine Learning, N – Gram theory, Author Profiling*

Παναγούλιας Κωνσταντίνος

Τμήμα Μηχανικών Πληροφοριακών και Επικοινωνιακών Συστημάτων

Πανεπιστήμιο Αιγαίου

Abstract

This thesis discusses the development of an Author Profiling model, utilising document representation and Machine Learning algorithms. The aim is to classify authors by their gender, while examining multiple variations of document representation.

In the theoretical part of the thesis, text mining techniques are studied, along with document representation approaches and evaluation metrics. The preprocessing phase is also analysed, for the data extracted by the representation to be ultimately inserted into a Machine Learning classification algorithm. Then, the concept of Machine Learning, with its various subfields and some important algorithms, are thoroughly analysed. More focus is put specifically on text classification implementations of Machine Learning. Finally, the author profiling problem is presented, with its numerous applications and in the context of the PAN competition.

In the experimental part, some of the aforementioned document representation and text classification techniques are utilised to perform author profiling on the CMCC Corpus dataset. For document representation, the N – gram theory is implemented, on word, character, and syntactic terms, for a range of N. Author profiling with regard to the author's gender is performed by a Machine Learning algorithm, as a binary classification problem, the two classes being: M – Male, F – Female.

Keywords: *Document Representation, Text Mining, Text Classification, Machine Learning, N – Gram theory, Author Profiling*

Panagoulias Konstantinos

Department of Information and Communication Systems Engineering

University of the Aegean

Εισαγωγή

Αναγνώριση προφίλ συγγραφέα (Author Profiling)

Η αναγνώριση προφίλ συγγραφέα (Author Profiling) περιγράφει τη διαδικασία ανάλυσης και εξόρυξης δεδομένων από κείμενα, με σκοπό τον προσδιορισμό κάποιων προσδιοριστικών παραμέτρων για τον συγγραφέα – δημιουργό του εκάστοτε κειμένου. Οι παράμετροι μπορεί να αφορούν το φύλο του συγγραφέα, την ηλικία του, το μορφωτικό του επίπεδο, τον τύπο προσωπικότητάς του, αλλά και αναγνώριση της γλώσσας του κειμένου, διαλέκτων ή αν η γλώσσα είναι μητρική ή ξένη για τον δημιουργό [1], [2]. Ακόμα, η αναγνώριση μπορεί να αφορά και προσδιορισμό της ίδιας της ταυτότητας του συγγραφέα ενός κειμένου, έχοντας ως δεδομένα άλλα κείμενά του [3].

Η εξαγωγή χαρακτηριστικών του συγγραφέα, καθώς και η δυνατότητα προβλέψεων με βάση αυτά, βρίσκει ποικίλες εφαρμογές, από χρήση για κατάλληλη στόχευση διαφήμισης, μέχρι αναγνώριση ενόχων στην εγκληματολογία [1]. Λόγω, λοιπόν, του έντονου ενδιαφέροντος, αλλά και της πληθώρας δεδομένων κειμένων, κυρίως από τα μέσα κοινωνικής δικτύωσης, η έρευνα στον συγκεκριμένο τομέα έχει σημειώσει σημαντική πρόοδο. Οι προσεγγίσεις μπορεί να διαφέρουν αρκετά, ανάλογα με το είδος του κειμένου, π.χ. αν τα κείμενα προέρχονται από προφορικό λόγο, αν περιέχουν ειδικούς χαρακτήρες, όπως Emoticons [4]. Επειδή το ύφος του συγγραφέα επηρεάζεται σημαντικά από το θέμα συζήτησης, αλλά και από το είδος του κειμένου που παράγει (Chat, Email, Tweet), η μελέτη των μεμονωμένων ειδών/θεμάτων μπορεί να παρέχει συμπεράσματα από διαφορετικές οπτικές [3]. Έτσι, θα παρουσίαζε ενδιαφέρον η μελέτη δειγμάτων διαφορετικών ειδών κειμένου και θεμάτων περιεχομένου για τον ίδιο συγγραφέα, ώστε να παρατηρηθεί αν υπάρχουν μοτίβα που να ενισχύουν τη δημιουργία ενός προφίλ.

Αντικείμενο Διπλωματικής

Στην παρούσα διπλωματική εργασία, η αναγνώριση του προφίλ του συγγραφέα πραγματοποιείται με βάση το φύλο του. Τα δεδομένα, το CMCC Corpus, προέρχονται από 6 διαφορετικά είδη κειμένων (Blog, Chat, Discussion, Email, Essay, Phone Interview), συνδυάζοντας είδη προφορικού και γραπτού λόγου, και το περιεχόμενο αφορά 6 διαφορετικά θέματα.

Για την αναπαράσταση των κειμένων, χρησιμοποιούνται 3 διαφορετικά μοντέλα της θεωρίας N – Gram, που θα αναφερθεί αναλυτικά στη συνέχεια, με μεταβλητές τιμές N, ώστε να ταιριάζουν στις ιδιαιτερότητες κάθε είδους κειμένου (π.χ. αραιά/πυκνά). Μετά την αναπαράσταση, εξάγεται συγκεκριμένο λεξιλόγιο για κάθε είδος κειμένου, το οποίο διαμορφώνεται βάσει μετρικών επιλογής χαρακτηριστικών κειμένου. Η ύπαρξη λεξιλογίου συμβάλει στο να φιλτραριστεί η χρήσιμη πληροφορία και επομένως να γίνει πιο αποτελεσματική η διαδικασία αναγνώρισης.

Η αναγνώριση του φύλου αντιμετωπίζεται ως πρόβλημα ταξινόμησης και επιλύεται με τεχνικές Μηχανικής Μάθησης. Ορισμένα μοντέλα ταξινομητών εκπαιδεύονται σε ένα μέρος των δεδομένων με επιβλεπόμενη μάθηση. Στη συνέχεια, δοκιμάζονται στο υπόλοιπο σύνολο δεδομένων, προβλέποντας το φύλο των συγγραφέων. Έτσι, εφόσον οι πραγματικές τιμές των φύλων είναι γνωστές, υπολογίζεται η ακρίβεια των αλγορίθμων στην πρόβλεψη.

Διάρθρωση της Διπλωματικής

Στο Κεφάλαιο 1, γίνεται αναφορά στην έννοια της εξόρυξης εγγράφων (Text Mining). Στο Κεφάλαιο 2, περιγράφεται πιο αναλυτικά η διαδικασία ταξινόμησης κειμένου (Text Classification) και οι αλγόριθμοι Μηχανικής Μάθησης που χρησιμοποιούνται σε αυτή. Λόγος για την εφαρμογή της ταξινόμησης κειμένου στην αναγνώριση προφίλ συγγραφέα (Author Profiling) γίνεται στο Κεφάλαιο 3. Στο Κεφάλαιο 4, αναφέρονται τα διάφορα εργαλεία και βιβλιοθήκες που αξιοποιήθηκαν για την υλοποίηση, ενώ στο Κεφάλαιο 5 γίνεται μία περιγραφή του συνόλου δεδομένων που χρησιμοποιήθηκε. Τα πειράματα και οι δοκιμές που πραγματοποιήθηκαν πάνω στον κώδικα αναλύονται στα

Κεφάλαια 6 και 7, ενώ στο Κεφάλαιο 8 γίνεται η παρουσίαση των συμπερασμάτων που εξήχθησαν από αυτές, καθώς και μελλοντικών επεκτάσεων. Στο Παράρτημα Ι παρατίθεται ο κώδικας που υλοποιήθηκε, καθώς και πιο αναλυτικά αποτελέσματα από τα πειράματα στο Παράρτημα ΙΙ.

1 Εξόρυξη Γνώσης από Έγγραφα (Text Mining)

1.1 Εισαγωγή

Το πρόβλημα της εξόρυξης εγγράφων έχει αποκτήσει ολοένα και μεγαλύτερη προσοχή τα τελευταία χρόνια, λόγω μιας ποικιλίας εφαρμογών κοινωνικού δικτύου, ιστού και άλλων εφαρμογών που δημιουργούν συνεχώς τεράστιες ποσότητες δεδομένων εγγράφου. Τα μη δομημένα δεδομένα είναι η συνηθέστερη και κυρίαρχη μορφή δεδομένων που μπορούν να δημιουργηθούν σε οποιαδήποτε εφαρμογή. Ως αποτέλεσμα, δημιουργήθηκε μια τεράστια ανάγκη να σχεδιαστούν μέθοδοι και αλγόριθμοι, οι οποίοι να μπορούν να επεξεργάζονται αποτελεσματικά μια μεγάλη ποικιλία εφαρμογών εγγράφου. Σε αυτό το κεφάλαιο, θα πραγματοποιήσουμε μια επισκόπηση των διαφορετικών μεθόδων και αλγορίθμων που είναι οι πιο διαδεδομένες στον τομέα του εγγράφου, με ιδιαίτερη έμφαση στις μεθόδους εξόρυξης.

1.2 Ορισμός

Η εξόρυξη εγγράφων αφορά την διαδικασία εξαγωγής των σημαντικών πληροφοριών από δεδομένα εγγράφων. Προσπαθεί να ανακαλύψει τη γνώση από αδόμητα έγγραφα. Είναι επίσης γνωστή ως Εξόρυξη Δεδομένων Εγγράφων (Text Data Mining). Η διαδικασία εξόρυξης εγγράφων συμπίπτει με την εξόρυξη δεδομένων, εκτός από τα εργαλεία εξόρυξης δεδομένων που έχουν σχεδιαστεί για να χειρίζονται τα δομημένα δεδομένα, ενώ η εξόρυξη εγγράφων μπορεί να χειριστεί μη δομημένα ή ημιδομημένα σύνολα δεδομένων, όπως HTML αρχεία, Emails, έγγραφα πλήρους κειμένου, κλπ. [5].

Η εξόρυξη εγγράφων χρησιμοποιείται για την εύρεση των νέων, παλιότερα άγνωστων πληροφοριών από διαφορετικές πηγές εγγράφων [6].

1.3 Τεχνικές του Text Mining

1.3.1 Ανάκτηση Πληροφορίας (Information Retrieval – IR)

Η ανάκτηση πληροφορίας σχετίζεται εδώ και πολλά χρόνια με τα συστήματα βάσεων δεδομένων. Ανάκτηση πληροφορίας σημαίνει ανάκτηση και συσχέτιση σημαντικών πληροφοριών από έναν αριθμό εγγράφων [7]. Μερικά προβλήματα ανάκτησης πληροφορίας, όπως μη δομημένα έγγραφα, εκτιμώμενη αναζήτηση με βάση λέξεις – κλειδιά και η συνάφεια, δεν ανήκουν στα συστήματα βάσεων δεδομένων. Επίσης, και μερικά προβλήματα στα συστήματα αυτά δεν ανήκουν στην ανάκτηση πληροφορίας, όπως η επαναφορά, η διαχείριση αλλαγών, η ενημέρωση και ο έλεγχος ταυτόχρονης λειτουργίας. Η ανάκτηση πληροφορίας έχει πολλές εφαρμογές σε πολλούς τομείς εξαιτίας του τεράστιου όγκου πληροφορίας εγγράφου, όπως τα συστήματα διαχείρισης εγγράφων και οι εξελιγμένες μηχανές αναζήτησης στο διαδίκτυο.

1.3.2 Εξαγωγή Πληροφορίας (Information Extraction – IE)

Η εξαγωγή πληροφορίας στοχεύει στην εξαγωγή χρήσιμων πληροφοριών, καθώς προσδιορίζει τις οντότητες και τις σχέσεις σε ημιδομημένα και αδόμητα έγγραφα. Οι κυριότερες πληροφορίες σε ένα έγγραφο είναι το όνομα ενός ατόμου, η τοποθεσία και ο οργανισμός [8].

1.3.3 Ταξινόμηση (Classification)

Η ταξινόμηση εγγράφων αποτελεί ένα είδος επιβλεπόμενης μάθησης, στην οποία σε όλη την διάρκεια της εκπαίδευσης εγγράφων οι κλάσεις είναι ήδη γνωστές και

σταθερές. Η ταξινόμηση αποτελεί την ανάθεση εγγράφων σε ένα προκαθορισμένο σύνολο θεμάτων ανάλογα με το περιεχόμενό τους. Πρόκειται για ένα σώμα εγγράφων, όπου σε κάθε έγγραφο εξελίσσεται η διαδικασία εύρεσης ακριβών θεμάτων. Την εμφάνισή της έχει κάνει και η αυτόματη ταξινόμηση εγγράφων πάνω σε εξατομικευμένες διαφημίσεις, φιλτράρισμα ανεπιθύμητων μηνυμάτων, δημιουργία μεταδεδομένων και ανίχνευση είδους εγγράφου, κ.ά. [9]. Πρόκειται για έναν σημαντικό ερευνητικό τομέα της μηχανικής μάθησης σήμερα.

1.3.4 Ομαδοποίηση (Clustering)

Η ομαδοποίηση είναι ένα είδος μη επιβλεπόμενης μάθησης, της οποίας στόχος αποτελεί ο εντοπισμός εγγενών δομών πληροφόρησης και η οργάνωσή τους σε υποομάδες για περαιτέρω μελέτη και ανάλυση. Το κύριο πρόβλημα είναι η ομαδοποίηση μιας μη ταξινομημένης συλλογής σε συστάδες, χωρίς προηγούμενη πληροφόρηση. Κάθε συστάδα λαμβάνει μια ετικέτα, η οποία σχετίζεται με ένα αντικείμενο που ανήκει αποκλειστικά στα δεδομένα. Για παράδειγμα, η ομαδοποίηση εγγράφων συμβάλλει στην ανάκτηση, δημιουργώντας συνδέσμους μεταξύ σχετικών εγγράφων, και στη συνέχεια αυτά επιτρέπουν την ανάκτησή τους, μόλις κάποιο θεωρηθεί σχετικό με ένα συγκεκριμένο ερώτημα [8]. Η ομαδοποίηση έχει εφαρμογή σε διάφορα πεδία, όπως η βιολογία, η εξόρυξη δεδομένων, η αναγνώριση προτύπων, η ανάκτηση και ταξινόμηση εγγράφων, η ασφάλεια, η επιχειρηματική ευφυΐα και η αναζήτηση στο διαδίκτυο. Μπορεί να χρησιμοποιηθεί ως αυτόνομο εργαλείο εξόρυξης εγγράφου ή ως ένα βήμα προεπεξεργασίας για άλλους αλγόριθμους εξόρυξης εγγράφου που λειτουργούν στις ανιχνευόμενες συστάδες.

1.3.5 Περιληπτική Παρουσίαση Πληροφορίας (Summarization)

Η σύνοψη είναι μια διαδικασία αυτόματης δημιουργίας συμπιεσμένης έκδοσης ενός συγκεκριμένου εγγράφου, το οποίο παρέχει σημαντικές πληροφορίες για έναν χρήστη. Παρόλο που διαθέτει περιεχόμενο μειωμένου μήκους, προσπαθεί να διατηρήσει και τη γενική ουσία ενός ή περισσότερων πρωτότυπων εγγράφων, επισημαίνοντας τις

σημαντικότερες πληροφορίες. Η σύνοψη κειμένου περιλαμβάνει διάφορες μεθόδους που αναπτύσσονται στην κατηγοριοποίηση κειμένου, όπως νευρωνικά δίκτυα, δέντρα απόφασης, μοντέλα παλινδρόμησης, δέντρα απόφασης, κ.ά. Ωστόσο, το κοινό πρόβλημα όλων αυτών των μεθόδων είναι η ποιότητα ανάπτυξης των ταξινομητών, η οποία εξαρτάται σημαντικά από το είδος του εγγράφου προς σύνοψη.

1.3.6 Αναγνώριση Γλώσσας (Language Identification)

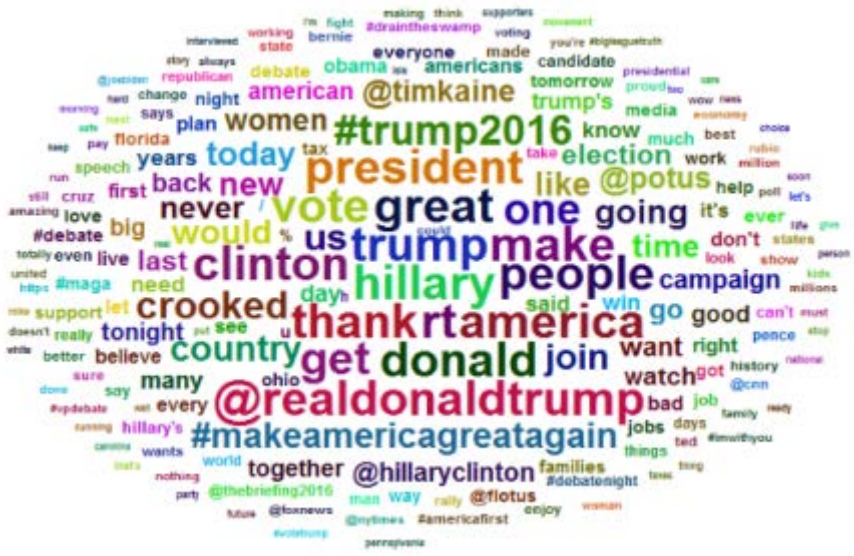
Αφορά τον προσδιορισμό μιας γλώσσας στην οποία είναι γραμμένη ένα έγγραφο ή, αν το έγγραφο περιέχει τμήματα σε διάφορες γλώσσες, τον προσδιορισμό αυτών. Έρευνες έχουν πραγματοποιηθεί σε γλωσσολογικά και στατιστικά μοντέλα για αυτή την διαδικασία. Υπάρχουν και μέθοδοι αναγνώρισης γλώσσας, όπως αυτή με βάση την κατανομή των γραμμάτων στο έγγραφο και η N – Gram προσέγγιση που θα αναφερθεί στη συνέχεια.

1.3.7 Κανόνες Συσχέτισης (Association Rules)

Στην ανάλυση των κανόνων συσχετίσεων ουσιαστικά μιλάμε για τη σχέση μεταξύ των χαρακτηριστικών που εξάγονται από ένα έγγραφο [10]. Υπάρχει, λοιπόν, μια συνθήκη με βάση ένα πρότυπο εγγράφου. Δηλαδή, αν ένας όρος μέσα σε ένα έγγραφο περιέχεται μέσα στο πρότυπο και βρίσκεται σε μια ορισμένη απόσταση από έναν άλλο δεδομένο όρο, τότε η συνθήκη αυτή διατηρείται με μεγάλη πιθανότητα.

1.3.8 Οπτικοποίηση (Visualization)

Το εργαλείο αυτό με τη χρήση της εξαγωγής χαρακτηριστικών (Feature Extraction) μπορεί να αναπαραστήσει γραφικά μια συλλογή εγγράφων. Η προσέγγιση αυτή συντελεί στον άμεσο εντοπισμό κάποιων σημαντικών στοιχείων που μας ενδιαφέρουν. Για παράδειγμα, αν ένας όρος αποτυπώνεται με μεγάλη συμβολοσειρά, θεωρείται σημαντικός (Word Cloud).



Εικόνα 1: Παράδειγμα Word Cloud[1]

1.4 Αναπαράσταση Εγγράφου (Document Representation)

Τα έγγραφα δεν έχουν μια προκαθορισμένη μορφή, αποτελούν, δηλαδή, μια αδόμητη μορφή πληροφορίας. Αυτό σημαίνει ότι προσπαθούμε να επεξεργαστούμε ένα σύνολο από λέξεις, γνωστό ως «Σακούλα των λέξεων» (Bag of Words – BOW), όπου βρίσκονται όλες οι λέξεις του εγγράφου. Ο δημοφιλέστερος τρόπος αναπαράστασης εγγράφου είναι η διανυσματική αναπαράσταση (Vector Representation). Κατά τη διανυσματική αναπαράσταση, κάθε έγγραφο απεικονίζεται ως ένα διάνυσμα όρων (Term Vector), όπου κάθε όρος αποτελεί ένα μοναδικό και ανεξάρτητο χαρακτηριστικό (Feature). Σε κάθε όρο – διάσταση του διανύσματος αποδίδεται μια τιμή, η οποία εκφράζει τον αριθμό εμφάνισης του συγκεκριμένου όρου μέσα στο έγγραφο.

Όπως δηλώνει και το όνομα «Bag of Words», το κομμάτι εγγράφου αντιμετωπίζεται ως σάκος λέξεων. Μειονέκτημα της τεχνικής αυτής αποτελεί το ότι δεν μας ενδιαφέρει η σειρά των λέξεων, αλλά μόνο η παρουσία ή/και η συχνότητα με την οποία εμφανίζονται οι λέξεις στο έγγραφο. Επίσης, επειδή αγνοούμε τη σημασία και τη σύνταξη των λέξεων, προτάσεις με τις ίδιες λέξεις θα αναπαρασταθούν με τον ίδιο τρόπο.

[1] (https://www.displayr.com/sentiment-analysis-example)

Το πλεονέκτημα της τεχνικής αυτής είναι ότι, λόγω της παραγωγής αραιών διανυσμάτων και μεγάλων διαστάσεων στις περισσότερες περιπτώσεις επιλογής χαρακτηριστικών, είναι και γραμμικώς διαχωρίσιμα.

1.4.1 Μοντέλο Διανυσματικού Χώρου (Vector Space Model – VSM)

Το Μοντέλο Διανυσματικού Χώρου αναπαριστά τα έγγραφα σε μορφή διανυσμάτων. Η διαδικασία του μοντέλου διανυσματικού χώρου μπορεί να διαιρεθεί σε τρία στάδια. Σε αρχικό στάδιο γίνεται η δεικτοδότηση εγγράφων (Document Indexing), όπου οι όροι που φέρουν περιεχόμενο, ανακτώνται από το κείμενο του εγγράφου. Το δεύτερο στάδιο είναι η στάθμιση των όρων που έχουν βρεθεί, για τη βελτίωση της ανάκτησης του εγγράφου που αφορά τον χρήστη. Στο τελευταίο στάδιο, το έγγραφο ταξινομείται σε σχέση με ένα ερώτημα σύμφωνα με ένα μέτρο ομοιότητας [11].

1.4.1.1 Δεικτοδότηση Εγγράφων (Document Indexing)

Υπάρχουν πολλές από τις λέξεις σε ένα έγγραφο που δεν φέρουν περιεχόμενο, όπως “be”. Κάνοντας χρήση αυτόματης δεικτοδότησης εγγράφων, αυτές οι ασήμαντες λειτουργικές λέξεις αφαιρούνται από το διάνυσμα εγγράφων, οπότε το έγγραφο θα εκπροσωπείται μόνο από λέξεις που έχουν «αξία» [12]. Αυτή η δεικτοδότηση μπορεί να βασιστεί στη συχνότητα των όρων. Όροι που έχουν υψηλή και χαμηλή συχνότητα μέσα σε ένα έγγραφο θεωρούνται λειτουργικές λέξεις [13], [14]. Αποτελείται από ένα κατάλληλο σύνολο λέξεων – κλειδιών που βασίζεται σε όλο το σώμα του εγγράφου και αντιστοιχεί βάρη σε εκείνες τις λέξεις – κλειδιά για κάθε συγκεκριμένο έγγραφο, μετατρέποντας έτσι κάθε έγγραφο σε ένα διάνυσμα από βάρη λέξεων – κλειδιών. Το βάρος σχετίζεται με τη συχνότητα εμφάνισης του συγκεκριμένου όρου στο έγγραφο και με το πλήθος των εγγράφων που χρησιμοποιούν αυτόν τον όρο.

1.4.1.2 Στάθμιση Όρων

Η στάθμιση όρων καθορίζει την επιτυχία ή την αποτυχία του συστήματος ταξινόμησης. Σε ένα έγγραφο, διαφορετικοί όροι έχουν διαφορετικό επίπεδο σημαντικότητας, οπότε το βάρος ενός όρου σχετίζεται με κάθε όρο ως ένας σημαντικός δείκτης [15]. Τα τρία βασικά συστατικά που επηρεάζουν τη σημαντικότητα ενός όρου σε ένα έγγραφο είναι:

Συχνότητα του Όρου (Term Frequency – TF)

Συχνότητα όρου, $tf(t,d)$, είναι η συχνότητα του όρου t σε ένα έγγραφο d και δίνεται από την εξής σχέση:

$$tf(t,d) = \frac{f_{t,d}}{\sum_{t' \in d} f_{t',d}} \quad (1)$$

όπου $f_{t,d}$ είναι η ακατέργαστη καταμέτρηση ενός όρου σε ένα έγγραφο, δηλαδή πόσες φορές συναντάται ο όρος t στο έγγραφο d. Υπάρχουν επίσης παραλλαγές συχνότητας όρου που αναφέρονται στην παρακάτω Εικόνα [16].

weighting scheme	tf weight
binary	$0, 1$
raw count	$f_{t,d}$
term frequency	$f_{t,d} \Big/ \sum_{t' \in d} f_{t',d}$
log normalization	$\log(1 + f_{t,d})$
double normalization 0.5	$0.5 + 0.5 \cdot \frac{f_{t,d}}{\max_{\{t' \in d\}} f_{t',d}}$
double normalization K	$K + (1 - K)\frac{f_{t,d}}{\max_{\{t' \in d\}} f_{t',d}}$

Εικόνα 2: Παραλλαγές Συχνότητας Όρου[2]

[2] https://en.wikipedia.org/wiki/Tf-idf

Αντίστροφη Συχνότητα του Εγγράφου (Inverse Document Frequency – IDF)

Η αντίστροφη συχνότητα εγγράφου είναι μία μετρική που εκφράζει το πόσες πληροφορίες παρέχει ένας όρος, δηλαδή, αν είναι κοινός ή σπάνιος στο συγκεκριμένο σώμα εγγράφων. Υπολογίζεται ως το λογαριθμικά κλιμακωτό αντίστροφο κλάσμα από όλα τα έγγραφα που περιέχουν τον όρο και δίνεται από την εξής σχέση:

$$idf(t, D) = log \frac{N}{|\{d \in D: t \in d\}|} \quad (2)$$

όπου N= $|D|$ είναι το πλήθος των εγγράφων σε ένα σώμα και $|\{d \in D: t \in d\}|$ είναι το πλήθος των εγγράφων, όπου ο όρος t εμφανίζεται (δηλ. $tf(t, d) \neq 0$). Αν ο όρος δεν ανήκει στο σώμα, ο παρονομαστής θα γίνει 0. Γι' αυτό τον γράφουμε ως: $1 + |\{d \in D: t \in d\}|$.

Παραλλαγές της αντίστροφης συχνότητας εγγράφου αναφέρονται στην παρακάτω Εικόνα 3.

weighting scheme	idf weight ($n_t = \|\{d \in D : t \in d\}\|$)
unary	1
inverse document frequency	$\log \frac{N}{n_t} = -\log \frac{n_t}{N}$
inverse document frequency smooth	$\log\left(\frac{N}{1+n_t}\right) + 1$
inverse document frequency max	$\log\left(\frac{\max_{\{t' \in d\}} n_{t'}}{1+n_t}\right)$
probabilistic inverse document frequency	$\log \frac{N-n_t}{n_t}$

Εικόνα 3: Παραλλαγές Αντίστροφης Συχνότητας Όρου[3]

[3] https://en.wikipedia.org/wiki/Tf-idf

Η TF – IDF [15], [17] είναι μια τεχνική, η οποία χρησιμοποιεί τόσο την TF όσο και την IDF για να προσδιοριστεί το βάρος ενός όρου ενός εγγράφου. Το TF – IDF σχήμα είναι πολύ δημοφιλές στο πεδίο της ταξινόμησης εγγράφων και σχεδόν όλα τα άλλα σχήματα είναι παραλλαγές αυτού. Δοθέντος ενός σώματος D, ενός όρου t και ενός ανεξάρτητου εγγράφου d, το TF – IDF περιγράφεται από την παρακάτω σχέση:

$$tf - idf(t, d, D) = tf(t, d) \times idf(t, D) \tag{3}$$

Παραλλαγές της τεχνικής αναφέρονται στην παρακάτω Εικόνα.

weighting scheme	document term weight	query term weight
1	$f_{t,d} \cdot \log \frac{N}{n_t}$	$\left(0.5 + 0.5 \frac{f_{t,q}}{\max_t f_{t,q}}\right) \cdot \log \frac{N}{n_t}$
2	$\log(1 + f_{t,d})$	$\log\left(1 + \frac{N}{n_t}\right)$
3	$(1 + \log f_{t,d}) \cdot \log \frac{N}{n_t}$	$(1 + \log f_{t,q}) \cdot \log \frac{N}{n_t}$

Εικόνα 4: Παραλλαγές TF – IDF[4]

Η τεχνική TF – IDF βασίζεται στη λογική ότι όσο συχνότερα συναντάμε έναν όρο σε ένα έγγραφο, τόσο πιο αντιπροσωπευτική πληροφορία μπορεί να μας παρέχει για τη σημασία του. Ακόμα, αξιοποιεί την πρόταση ότι όσο αυξάνεται ο αριθμός εγγράφων στα οποία συναντάται ένας όρος, τόσο μειώνεται η αξία του σχετικά με την πληροφορία που προσφέρει.

Ουσιαστικά, η TF – IDF μετρά την αξία ενός όρου σε σχέση με ένα έγγραφο μόνο ως συνάρτηση της συχνότητας εμφάνισής του σε αυτό, χωρίς να λαμβάνεται υπόψη η σειρά με την οποία εμφανίζονται οι όροι σε αυτό. Τα έγγραφα πρέπει να αναπαρασταθούν με διανύσματα ίσου μήκους, στα οποία εφαρμόζονται στα βάρη w_k, τα οποία κανονικοποιούνται στο διάστημα [0,1] σύμφωνα με την παρακάτω εξίσωση:

[4] https://en.wikipedia.org/wiki/Tf-idf

$$w_k = \frac{tf - idf(t_k, d, D)}{\sqrt{\sum_{i=1}^{|T|}(tf - idf(t_i, d, D))^2}}$$

(4)

όπου $|T|$ το πλήθος των όρων των εγγράφων (Sahoo, 2018). Εκτός από τη δεικτοδότηση μέσω TF – IDF, έχουν χρησιμοποιηθεί και άλλες μέθοδοι, όπως μέθοδοι βασισμένες στις πιθανότητες εμφάνισης (Gövert et al., 1999) ή μέθοδοι που εφαρμόζουν δεικτοδότηση σε δομημένα έγγραφα (Larkey and Croft, 1996).

Κανονικοποίηση του Μήκους του Εγγράφου (Document Length Normalization)

Σε ένα μεγάλο σώμα μπορεί να περιέχονται έγγραφα με μεγάλες διαφορές στα μεγέθη μεταξύ τους. Έτσι, μία μέθοδος δεικτοδότησης θα είναι πολωμένη ως προς τα έγγραφα με μεγαλύτερο μήκος, καθώς οι περισσότεροι όροι τους έχουν και μεγαλύτερη πιθανότητα εμφάνισης. Για να αντιμετωπιστεί αυτό, εφαρμόζουμε κανονικοποίηση, διαιρώντας με το μήκος εγγράφου. Το μήκος μπορεί να αναπαρασταθεί με διαφορετικού τρόπους, ανάλογα με την αναπαράσταση εγγράφου που επιλέγεται:

- **Αναπαράσταση ως ακολουθία bytes:** Το έγγραφο αναπαρίσταται ως ακολουθία bytes, οπότε ως μέγεθος του εγγράφου υπολογίζεται ως ο συνολικός αριθμός bytes που αποτελούν τους όρους. Πλεονέκτημα αυτής της μεθόδου αποτελεί το γεγονός ότι χαρακτηρίζεται από απλότητα.
- **Αναπαράσταση ως ακολουθία λέξεων:** Το έγγραφο αναπαρίσταται ως σύνολο των λέξεων που το αποτελούν, οπότε το μέγεθος του εγγράφου υπολογίζεται ως ο συνολικός αριθμός των λέξεων του. Στα θετικά αυτής της μεθόδου περιλαμβάνεται η απλότητά της, αλλά υπεισέρχεται μία πολυπλοκότητα λόγω της ανάγκης χωρισμού σε λέξεις.
- **Αναπαράσταση ως νόρμες διανυσμάτων:** Το έγγραφο d αναπαρίσταται ως διανύσματα των όρων που το αποτελούν, αφού εφαρμοστεί μία από τις μεθόδους στάθμισης όρων που αναφέρθηκαν. Σε κάθε όρο αντιστοιχίζεται ένα ορθοκανονικό μοναδιαίο διάνυσμα σε ένα χώρο t διαστάσεων (όπου t ο συνολικός αριθμός των όρων). Ο κάθε όρος k_i του εγγράφου d αντιστοιχίζεται

με ένα διάνυσμα k_i, που υπολογίζεται ως $tf - idf_{t,d} \times k_i$. Έτσι, το συνολικό έγγραφο αναπαρίσταται ως η συνισταμένη των επιμέρους διανυσμάτων όλων των όρων.

1.4.1.3 Μετρικές Ομοιότητας

Η ομοιότητα στα μοντέλα διανυσματικού χώρου προσδιορίζεται χρησιμοποιώντας σχετικούς συντελεστές, που βασίζονται στο εσωτερικό γινόμενο του διανύσματος εγγράφων και του διανύσματος ερωτήματος, όπου η αλληλοεπικάλυψη λέξεων εκφράζει ομοιότητα. Το πιο δημοφιλές μέτρο ομοιότητας είναι το διάνυσμα συνημίτονου. Δεδομένου ενός εγγράφου d και ενός ερωτήματος q, έχουμε την παρακάτω σχέση:

$$S_c(d,q) = cos(d,q) = \frac{d \cdot q}{\|d\|\|q\|} \quad (5)$$

1.5 Προεπεξεργασία Δεδομένων

Η φάση της προεπεξεργασίας μετατρέπει τα αρχικά δεδομένα σε μια δομή έτοιμη για εξόρυξη δεδομένων, όπου εντοπίζονται σημαντικά χαρακτηριστικά κειμένου, που χρησιμεύουν για τη διαφοροποίηση μεταξύ κειμένων – κατηγοριών. Πρόκειται για τη διαδικασία ενσωμάτωσης ενός νέου εγγράφου σε ένα σύστημα ανάκτησης πληροφορίας [18]. Η απόκτηση σημαντικών χαρακτηριστικών ή όρων – κλειδιών από έγγραφα, η ενίσχυση της συνάφειας λέξης – εγγράφου και η συνάφεια όρου – κλάσης αποτελούν τους βασικούς στόχους της προεπεξεργασίας. Ο στόχος πίσω από αυτήν είναι ότι κάθε έγγραφο αναπαρίσταται ως ένα διάνυσμα, το οποίο χωρίζει το έγγραφο σε μεμονωμένους όρους και η επιλογή όρου – κλειδιού είναι απαραίτητη για την ευρετηρίαση εγγράφων. Αυτό οδηγεί στο στάδιο της ταξινόμησης, οπότε είναι σημαντικό να βρούμε όρους που φέρουν σημασία και να απορρίψουμε όρους που δε συμβάλλουν στη διάκριση μεταξύ των εγγράφων.

1.5.1 Λεκτική Ανάλυση (Lexical Analysis)

Λεκτική ανάλυση είναι η διαδικασία κατά την οποία μια ακολουθία χαρακτήρων μετατρέπεται σε ακολουθία λεκτικών μονάδων (Tokens). Για να επιτευχθεί αυτή η μετατροπή, θα πρέπει να διαχωρίσουμε τις μονάδες αυτές με βάση τους κενούς χαρακτήρες, μία αρκετά απλή διαδικασία. Ωστόσο, προστίθεται μία πολυπλοκότητα στη διαδικασία, καθώς οι όροι που χρησιμοποιούμε ως λεκτικές μονάδες κάθε φορά μπορεί να ανήκουν σε διαφορετικές κατηγορίες και να απαιτούν σημασιολογική ανάλυση. Μερικές κατηγορίες λεκτικών μονάδων είναι οι εξής: Δεσμευμένοι Όροι (Reserved Words), Σταθερές (Constants), Τελεστές (Operators), Διακριτικά (Identifiers), Διαχωριστές (Delimiters).

Η διαδικασία μετατροπής ενός κειμένου σε ακολουθία λεκτικών μονάδων (Tokenization) περιλαμβάνει την αναγνώριση των λεκτικών μονάδων και την κατηγοριοποίησή τους με βάση τις κατηγορίες που αναφέρθηκαν. Σε ορισμένες περιπτώσεις μπορεί να απαιτείται επιπλέον ανάλυση για τον σωστό διαχωρισμό, όπως σε περίπτωση ύπαρξης Emoticons, ειδικών χαρακτήρων με συγκεκριμένη σημασία ανάλογα με το πλαίσιο χρήσης (π.χ. “#”, “@” στα μέσα κοινωνικής δικτύωσης), ηλεκτρονικών συνδέσμων, μίξης πεζών – κεφαλαίων χαρακτήρων, κλπ.

1.5.2 Αφαίρεση των Stop Words

«Stop Words» ονομάζονται οι όροι, οι οποίοι χρησιμοποιούνται συχνά σε κάθε γλώσσα και είναι απαραίτητοι για τη σύνταξή της. Αυτοί οι όροι είναι άχρηστοι στην ανάκτηση πληροφορίας και την εξόρυξη κειμένου και δεν περιέχουν πληροφορίες, όπως αντωνυμίες, προθέσεις, συζεύξεις κ.α. Συγκεκριμένα, η αγγλική γλώσσα περιλαμβάνει περίπου 400 – 500 Stop Words, όπως “the”, “of”, “and”, “to”. Αυτό είναι το πρώτο σημαντικό βήμα κατά την προεπεξεργασία [9].

1.5.3 Λημματοποίηση (Stemming)

Η λημματοποίηση χρησιμοποιείται για την ανακάλυψη της ρίζας ενός όρου. Μετατρέπει τους όρους στις ρίζες τους, η οποία ενσωματώνει πολλές εκδοχές του όρου αυτού. Όροι με την ίδια ρίζα, που περιγράφουν ίδιες ή σχετικά στενές έννοιες σε ένα έγγραφο, μπορούν να συγχωνευθούν με αυτή τη διαδικασία. Για παράδειγμα, οι όροι “user”, “users”, “used”, “using” μπορούν να λημματοποιηθούν στον όρο “use”.

1.5.4 POS Tagging

Στην επεξεργασία φυσικής γλώσσας, η επισήμανση μερών του λόγου (POS Tagging ή POST), η οποία ονομάζεται επίσης γραμματική σήμανση ή αποσαφήνιση κατηγοριών όρων, είναι η διαδικασία σήμανσης ενός όρου σε ένα έγγραφο, το οποίο αντιστοιχεί σε ένα συγκεκριμένο μέρος της ομιλίας, με βάση τόσο τον ορισμό του όσο και το πλαίσιο του – δηλαδή τη σχέση του με τους γειτονικούς και συναφείς όρους σε μια φράση, πρόταση ή παράγραφο. Μια απλοποιημένη μορφή αυτού διδάσκεται συνήθως στα παιδιά της σχολικής ηλικίας, στην ταύτιση των όρων ως ουσιαστικά, ρήματα, επίθετα, επιρρήματα (π.χ. ο όρος “συντομεύσεις”, ανάλογα με τα συμφραζόμενα, θα μπορούσε να λειτουργεί ως ρήμα ή ως ουσιαστικό) [19].

Η διαδικασία της επισήμανσης των μερών του λόγου μιας πρότασης πραγματοποιεί την αντιστοίχιση του κάθε όρου που έχει σχηματιστεί από την λεξικολογικής ανάλυσης, σε ένα μέρος του λόγου (Bird Steven et al., 2009).

1.5.5 Μείωση Διαστασιμότητας (Dimensionality Reduction)

Το πλήθος των εγγράφων όπου εμφανίζεται ένας όρος, ονομάζεται συχνότητα εγγράφων (Document Frequency – DF). Η μείωση λεξιλογίου γίνεται με μια απλή τεχνική, το DF κατώφλι. Όροι υψηλής συχνότητας, που δεν σχετίζονται με την διαδικασία ταξινόμησης και σπάνιοι όροι, αφαιρούνται με την εξάλειψη των Stop Words. Δηλαδή, όλοι οι όροι που εμφανίζονται σε έγγραφα μικρότερα από ένα

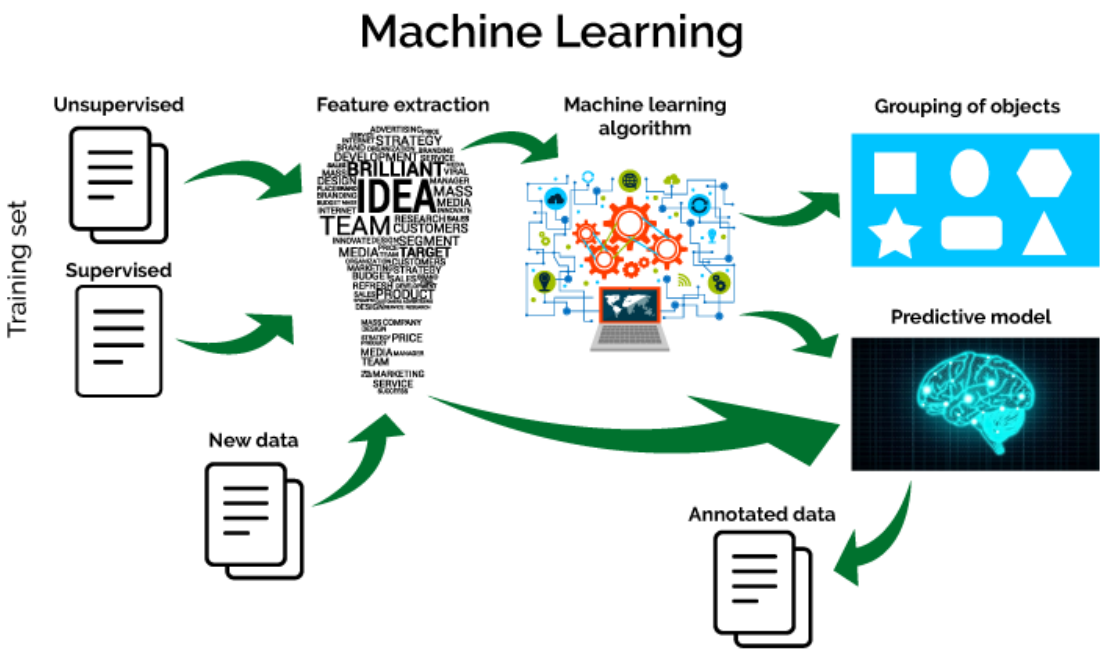
συγκεκριμένο ως προκαθορισμένο όριο ενός σώματος εγγράφων, δεν θεωρούνται χαρακτηριστικά. Το DF κατώφλι βασίζεται στην υπόθεση ότι οι σπάνιοι όροι δεν έχουν «αξία» για την πρόβλεψη κλάσεων.

1.6 Μηχανική Μάθηση (Machine Learning)

Η Μηχανική Μάθηση αποτελεί ένα υποπεδίο της Τεχνητής Νοημοσύνης που αφορά την ανάπτυξη μεθόδων που έχουν την ικανότητα να «μαθαίνουν». Πιο συγκεκριμένα, τα μοντέλα που αναπτύσσονται έχουν τη δυνατότητα να εκπαιδεύονται πάνω σε σύνολα δεδομένων, ώστε να είναι σε θέση να προβλέψουν ή να πάρουν αποφάσεις σχετικά με αυτά, χωρίς να προγραμματιστούν ρητά για το πώς να το επιτύχουν [20], [22].

Ο όρος εισήχθη το 1959 από τον Arthur Samuel, ενώ την ίδια περίοδο χρησιμοποιούταν ο όρος «αυτοδίδακτοι υπολογιστές» [21]. Στις αρχές της δεκαετίας του 1960, μία πειραματική «μηχανή μάθησης» με το όνομα “Cybertron” είχε αναπτυχθεί από την εταιρεία Raytheon, ώστε να αναλύει μοτίβα ομιλίας και ήχου, χρησιμοποιώντας ενισχυμένη μάθηση. Το ενδιαφέρον για τον τομέα, κυρίως για την αναγνώριση προτύπων, αυξήθηκε κατά τη δεκαετία του 1970, ενώ το 1981 αναφέρεται η ανάπτυξη ενός νευρωνικού δικτύου που αναγνωρίζει 40 χαρακτήρες [23], [24].

Η μηχανική μάθηση αξιοποιεί σε πολλές υποκατηγορίες προβλημάτων εργαλεία υπολογιστικής στατιστικής και μαθηματικής βελτιστοποίησης, ωστόσο στις περισσότερες περιπτώσεις οι προσεγγίσεις φτάνουν αρκετά πιο μακριά και διαφοροποιούνται από τη μαθηματική ανάλυση. Οι εφαρμογές της μηχανικής μάθησης εμφανίζονται σε πολλούς και διαφορετικούς τομείς, όπως η υπολογιστική όραση, τα οικονομικά και χρηματιστηριακά, καθώς και εφαρμογές με βιολογικό και ιατρικό ενδιαφέρον [25], [26].



Εικόνα 5: Machine Learning Process[5]

1.6.1 Υπολογιστική Νοημοσύνη και Μηχανική Μάθηση

Ως επιστημονική προσπάθεια, η μηχανική μάθηση εξελίχθηκε μέσα από την αναζήτηση για υπολογιστική νοημοσύνη. Στα πρώτα χρόνια της υπολογιστικής νοημοσύνης ως ακαδημαϊκού τομέα, οι ερευνητές προσανατολίζονταν προς το να εκπαιδεύουν τις μηχανές πάνω σε δεδομένα. Προσέγγιζαν το πρόβλημα με «νευρωνικά δίκτυα» της εποχής (κυρίως μοντέλα perceptron και άλλα μοντέλα π.χ. ADALINE), που τελικά αποτελούσαν κυρίως παραλλαγές των γραμμικών μοντέλων της στατιστικής.

Η αυξανόμενη έμφαση στη «συμβολική» τεχνητή νοημοσύνη (προσέγγιση των προβλημάτων με λογική και συμβολισμούς σε υψηλό επίπεδο, εύκολα κατανοητούς από την ανθρώπινη λογική) προκάλεσε μία αλλαγή κατεύθυνσης μεταξύ μηχανικής μάθησης και τεχνητής νοημοσύνης. Τη δεκαετία του 1980, τα συστήματα «ειδικών» (expert systems), υπολογιστικά συστήματα που προσομοίωναν την ικανότητα λήψης αποφάσεων από ανθρώπους μέσα από κανόνες “if…else”, κυριάρχησαν στην υπολογιστική νοημοσύνη, ενώ οι μέθοδοι στατιστικής παρουσίαζαν πλέον ερευνητικό

[5] https://towardsdatascience.com/

ενδιαφέρον κυρίως για τα πεδία της αναγνώρισης προτύπων και ανάκτησης πληροφορίας.

Η μηχανική μάθηση, αναγνωρισμένη ως ξεχωριστός κλάδος, άνθισε κυρίως από τη δεκαετία του 1990 και άλλαξε το στόχο της από την επίτευξη τεχνητής νοημοσύνης στην επίλυση πιο πρακτικών προβλημάτων. Η βασική διαφορά μεταξύ μηχανικής μάθησης και υπολογιστικής νοημοσύνης μπορούμε να πούμε ότι είναι ότι η πρώτη περιλαμβάνει εκπαίδευση και πρόβλεψη βασισμένη στην παρατήρηση δεδομένων, ενώ η τεχνητή νοημοσύνη υπονοεί την ύπαρξη ενός πράκτορα που αλληλεπιδρά με το περιβάλλον, ώστε να μάθει και να δράσει με σκοπό να επιτύχει τους στόχους του. Μέχρι σήμερα, πολλές πηγές συνεχίζουν να χαρακτηρίζουν τη μηχανική μάθηση ως υποκατηγορία της υπολογιστικής νοημοσύνης.

1.6.2 Τύποι Προβλημάτων και Εργασιών

Η μηχανική μάθηση μπορεί να χωριστεί σε κάποιες μεγάλες κατηγορίες, ανάλογα με το είδος προβλημάτων προς επίλυση και τον τρόπο εκπαίδευσης που ακολουθείται. Οι βασικές κατηγορίες που προκύπτουν με βάση τον τρόπο εκπαίδευσης είναι οι εξής [27]:

- **Επιβλεπόμενη Μάθηση (Supervised Learning):** Το μοντέλο λαμβάνει κάποια είσοδο δεδομένων, καθώς και κάποιες ετικέτες σχετικές με τα δεδομένα. Σκοπός είναι το μοντέλο να δημιουργήσει έναν κανόνα με βάση αυτή την αντιστοίχιση, ώστε να μπορεί να παράγει αποτελέσματα εξόδου και για νέα δεδομένα εισόδου.
- **Μη Επιβλεπόμενη Μάθηση (Unsupervised Learning):** Το μοντέλο δε λαμβάνει κάποια αποτελέσματα σχετικά με τα δεδομένα εισόδου, αλλά θα πρέπει να ανακαλύψει κάποιον κανόνα σχετικά με αυτά. Βασίζεται κυρίως σε εύρεση ομοιοτήτων στα δεδομένα και μπορεί να κάνει προβλέψεις πάνω σε νέα δεδομένα με βάση την παρουσία ή την έλλειψη αυτών των ομοιοτήτων.
- **Ημι – επιβλεπόμενη Μάθηση (Semi – supervised Learning):** Βρίσκεται μεταξύ της επιβλεπόμενης και μη επιβλεπόμενης μάθησης, καθώς στον αλγόριθμο παρέχονται τόσο δεδομένα εισόδου αντιστοιχισμένα με ετικέτες αποτελεσμάτων, όσο και δεδομένα χωρίς ετικέτες.

- **Ενισχυτική Μάθηση (Reinforcement Learning):** Ο αλγόριθμος περιλαμβάνει έναν πράκτορα, ο οποίος παίρνει αποφάσεις και δρα μέσα σε ένα περιβάλλον, ενώ συνεχώς λαμβάνει ανατροφοδότηση για την τωρινή κατάσταση και για την «ανταμοιβή» (Reward) που λαμβάνει για τις αποφάσεις του.
- **Διαδικασία Εκμάθησης (Meta Learning):** Βασίζεται στην αξιοποίηση υπάρχουσας εκπαίδευσης και εμπειρίας, με σκοπό το μοντέλο μηχανικής μάθησης να βελτιώσει τις ικανότητές του. Ως δεδομένα εισόδου λαμβάνει τα αποτελέσματα άλλου ήδη εκπαιδευμένου μοντέλου και δεδομένα σχετικά με την εκπαίδευση (Metadata).
- **Μάθηση με Κανόνες Συσχέτισης (Association Rule Learning):** Αποτελεί μέθοδο που στηρίζεται στην εύρεση κανόνων για συσχέτιση μεταβλητών, συνήθως σε μεγάλες βάσεις δεδομένων. Οι κανόνες συσχέτισης συνήθως προκύπτουν αναζητώντας στα δεδομένα συχνά μοτίβα “if…then” και έπειτα αποφασίζοντας ποια είναι τα πιο σημαντικά με βάση κάποιο κριτήριο.

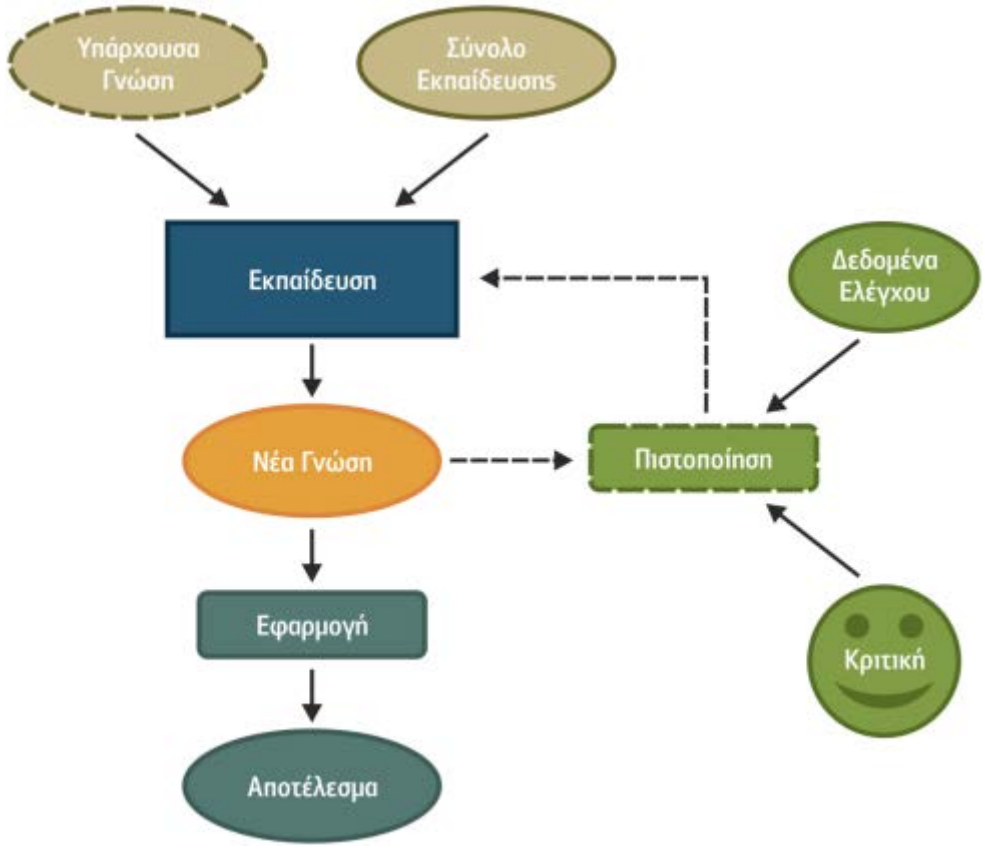
Οι κατηγορίες που προκύπτουν με βάση το είδος προβλήματος και στόχου της επίλυσης είναι οι εξής [18]:

- **Ταξινόμηση (Classification):** Το μοντέλο θα πρέπει να αποδώσει στα δεδομένα εισόδου μία ετικέτα (Label), ανάλογα με την κλάση που πιστεύει ότι κατατάσσονται. Το μοντέλο μπορεί να χωρίζει τα δεδομένα σε δύο ή περισσότερες κλάσεις (Multi – Label Classification). Παράδειγμα προβλήματος ταξινόμησης σε δύο κλάσεις αποτελεί η κατάταξη των μηνυμάτων ηλεκτρονικού ταχυδρομείου (Emails) σε “Spam” ή όχι [28]. Συνήθως πραγματοποιείται με επιβλεπόμενη μάθηση.
- **Ομαδοποίηση (Clustering):** Το μοντέλο πρέπει να ομαδοποιήσει τα δεδομένα ανάλογα με χαρακτηριστικά που παρουσιάζουν ομοιότητες μεταξύ τους. Συνήθως οι ομάδες που θα προκύψουν δεν είναι γνωστές πριν τη διαδικασία εκπαίδευσης, επομένων δεν υπάρχουν ετικέτες για τα δεδομένα και η εκπαίδευση συνήθως είναι μη επιβλεπόμενη.
- **Παλινδρόμηση (Regression):** Εφαρμόζεται σε συνεχή δεδομένα, με σκοπό την εύρεση μίας συσχέτισης μίας εξαρτημένης μεταβλητής και μίας ή πολλών ανεξάρτητων μεταβλητών.

- **Εκτίμηση Πυκνότητας (Density Estimation):** Αποτελεί μέθοδο στατιστικής και πιθανοτήτων. Το μοντέλο κατασκευάζει μία προσέγγιση, με βάση τα παρατηρούμενα δεδομένα, για μία μη παρατηρήσιμη συνάρτηση πυκνότητας πιθανότητας.
- **Μείωση Διαστασιμότητας (Dimensionality Reduction):** Περιλαμβάνει μία διαδικασία μείωσης των διαστάσεων των χαρακτηριστικών εισόδου, διατηρώντας αυτά τα οποία παρέχουν την απαραίτητη πληροφορία. Μία δημοφιλής μέθοδος μείωσης διαστάσεων είναι η Ανάλυση Κύριων Συνιστωσών (Principal Component Analysis).

Στην Εικόνα 6, απεικονίζεται η αρχή λειτουργίας των περισσότερων μοντέλων μηχανικής μάθησης. Τα δεδομένα, μετά από την προεπεξεργασία που υφίστανται, εισέρχονται στη φάση της εκπαίδευσης, όπου το σύνολο εκπαίδευσης (Training Set), μαζί με την υπάρχουσα γνώση και εμπειρία, εισάγονται στο μοντέλο, με σκοπό την εκπαίδευσή του, την απόκτηση, δηλαδή, νέας γνώσης σχετικά με τα δεδομένα.

Αφού το μοντέλο εκπαιδευτεί, η γνώση θα πρέπει να αξιολογηθεί στη φάση της πιστοποίησης (Validation). Με εφαρμογή της γνώσης σε δεδομένα ελέγχου (Test Set) για εξαγωγή προβλέψεων, μπορεί να αξιολογηθεί η απόδοση του μοντέλου με χρήση διάφορων μετρικών που θα αναφερθούν και στη συνέχεια. Αφού πιστοποιηθεί η αξιοπιστία του μοντέλου, αυτό μπορεί να αξιοποιηθεί σε εφαρμογές για επίλυση πρακτικών προβλημάτων.



Εικόνα 6: Γενικός τρόπος λειτουργίας Μηχανικής Μάθησης [29]

1.6.3 Αλγόριθμοι Μηχανικής Μάθησης

Γνωστοί αλγόριθμοι επιβλεπόμενης μάθησης είναι ενδεικτικά οι εξής [30]:

- Naïve Bayes
- Μέθοδος k – Πλησιέστερων Γειτόνων (k – NN)
- Μηχανές Διανυσμάτων Υποστήριξης (Support Vector Machines – SVM)
- Κανόνες Ταξινόμησης (Classification Rules)
- Δένδρα απόφασης (Decision Trees)
- K – Means
- Τυχαία Δάση (Random Forests)

Αργότερα, θα αναλύσουμε μερικούς από αυτούς, τους οποίους χρησιμοποιούμε και στην υλοποίησή μας.

2 Ταξινόμηση Κειμένου (Text Classification)

2.1 Εισαγωγή

Η διαδικασία της ταξινόμησης των εγγράφων πραγματοποιείται με την απόδοση ενός εγγράφου σε μια ή περισσότερες κλάσεις, με βάση τα χαρακτηριστικά του εγγράφου [31], [17]. Συνήθως αντιμετωπίζεται ως πρόβλημα επιβλεπόμενης μάθησης και οι κλάσεις, όπου θέλουμε να ταξινομήσουμε τα έγγραφα, είναι προκαθορισμένες [32], [33]. Ανάλογα με την περίπτωση, μπορεί να αντιμετωπιστεί είτε ως πρόβλημα δυαδικής ταξινόμησης, πολλαπλής ταξινόμησης ή ταξινόμησης πολλαπλών ετικετών.

Πιο συγκεκριμένα, έστω $D = \{d_1, \dots, d_n\}$ ένα σώμα εγγράφων και $C = \{c_1, \dots, c_m\}$ το σύνολο των προκαθορισμένων κλάσεων. Εμείς αναζητούμε μια συνάρτηση $f: D \to C$, τέτοια ώστε κάθε έγγραφο d_i να αντιστοιχεί σε μία ή περισσότερες κλάσεις c_i. Δηλαδή, δημιουργούμε έναν ταξινομητή (Classifier), ο οποίος εκχωρεί ετικέτες στα έγγραφα.

Σε πρώτο στάδιο, στα δεδομένα εκπαίδευσης και δοκιμής, κάθε έγγραφο αναπαρίσταται ως ένα διάνυσμα όρων – χαρακτηριστικών [17]. Στη συνέχεια, το μοντέλο μάθησης εκπαιδεύεται μέσω κάποιου αλγορίθμου ταξινόμησης, ο οποίος μαθαίνει τις παραμέτρους του μοντέλου από τα δεδομένα εκπαίδευσης. Έτσι, γίνεται χρήση του μοντέλου για την εκχώρηση ετικετών σε καινούργια έγγραφα στα δεδομένα δοκιμής.

Ο συνδυασμός διανυσμάτων, που προέρχονται από έγγραφα μαζί με τις ετικέτες τους, αποτελούν την είσοδο για τον αλγόριθμο μηχανικής μάθησης. Οι διανυσματικές αυτές αναπαραστάσεις των εγγράφων σχετίζονται με την εξαγωγή χαρακτηριστικών, τα οποία βοηθούν στην διαφοροποίηση μεταξύ των κλάσεων. Όπως θα δούμε και παρακάτω, τα διανύσματα μπορεί να εκφράζουν περιεχόμενο που αντλείται από διάφορες κατηγορίες χαρακτηριστικών όπως συντακτικά, σημασιολογικά, κτλ. Η σωστή επιλογή χαρακτηριστικών αποτελεί σημαντικό βήμα για την δημιουργία ενός επιτυχημένου μοντέλου.

Η απόδοση των αλγορίθμων, που θα χρησιμοποιηθούν για την επιλογή του ταξινομητή, διαφέρει ανάλογα με τα ιδιαίτερα χαρακτηριστικά του κάθε προβλήματος [34]. Για

παράδειγμα, η μέθοδος της Λογιστικής Παλινδρόμησης (Logistic Regression – LR) αποδίδει καλύτερα από τη μέθοδο Naive Bayes, όταν υπάρχουν συσχετιζόμενα χαρακτηριστικά. Η Naive Bayes αποδίδει πολύ καλά σε μικρά σύνολα δεδομένων [35], ενώ η μέθοδος SVM δίνει καλύτερα αποτελέσματα, όταν εφαρμόζεται σε μεγάλα σύνολα δεδομένων ή μεγάλα έγγραφα [36]. Επίσης, το πλήθος των χαρακτηριστικών που εξάγονται και η πολυπλοκότητα που το χαρακτηρίζει, παίζουν σημαντικό ρόλο. Για παράδειγμα, ένα μοντέλο όπως το SVM είναι επιρρεπές σε υψηλό σφάλμα διακύμανσης και υπάρχει ο κίνδυνος να μη γενικεύει επαρκώς σε νέα δεδομένα. Γι' αυτό σε αρκετές έρευνες, όταν υπάρχουν πολλά χαρακτηριστικά, ακολουθείται η διαδικασία της επιλογής χαρακτηριστικών (Feature Selection) [32], ώστε να απομακρυνθούν χαρακτηριστικά που μπορεί να προκαλούν υπερφόρτωση (Overfitting).

2.2 Τύποι Ταξινόμησης Κειμένων

Ένα είδος ταξινόμησης αποτελεί αυτό της μονής ετικέτας (Single – Label Classification), όπου ισχύει ο περιορισμός ότι κάθε έγγραφο πρέπει να αντιστοιχίζεται σε μία και μοναδική κλάση. Πλέον, πολλά έγγραφα περιέχουν περισσότερα από ένα χαρακτηριστικά και δεν υπάρχει κανένας περιορισμός για το πλήθος των κλάσεων στις οποίες πρέπει να αντιστοιχίζεται ένα έγγραφο. Έτσι, με την πάροδο του χρόνου δημιουργήθηκε ένα νέο είδος ταξινόμησης (Multi – Label Classification), όπου ένα έγγραφο διαθέτει πολλές ετικέτες διαφορετικών κλάσεων [37].

Ένα ακόμη είδος ταξινόμησης αποτελεί η δυαδική ταξινόμηση εγγράφου, μία υποκλάση της περίπτωσης της αντιστοίχισης σε μία ετικέτα, όπου ένα έγγραφο αντιστοιχίζεται σε μία κλάση ή στο συμπλήρωμα αυτής. Ένα μοντέλο για ταξινόμηση μονής ή διπλής ετικέτας μπορεί να χρησιμοποιηθεί για ταξινόμηση σε πολλαπλές κλάσεις, όπου για την ταξινόμηση χρησιμοποιούνται ανεξάρτητα τμήματα του μοντέλου, με το καθένα να αποφασίζει αν η είσοδος ανήκει στην συγκεκριμένη κλάση ή όχι. Ωστόσο, ένα μοντέλο που πραγματοποιεί ταξινόμηση σε πολλαπλές ετικέτες δεν μπορεί να χρησιμοποιηθεί για αντιστοίχιση σε μονή ή διπλή ετικέτα. Σχετικά με τις εφαρμογές σε ταξινόμηση κειμένου, πιο συχνά χρησιμοποιείται η ταξινόμηση διπλής

ετικέτας, λόγω της απλότητας και αποτελεσματικότητάς του, ωστόσο κάθε εφαρμογή είναι διαφορετική και μπορεί να απαιτείται διαφορετική προσέγγιση.

Μία εφαρμογή της μηχανικής μάθησης που σχετίζεται με την ταξινόμηση κειμένου είναι η επεξεργασία φυσικής γλώσσας (Natural Language Processing). Ωστόσο, η σημασιολογία και η συντακτική υπόσταση της φυσικής γλώσσας προκαλεί αμφισημίες και την καθιστά ιδιαίτερα δύσκολη στην επεξεργασία και την κατανόησή της, με αποτέλεσμα η επικοινωνία με τον υπολογιστή και ο έλεγχός του να αποτελεί εξίσου πρόβλημα.

Επιπλέον, υπάρχουν και άλλα είδη, όπως η ταξινόμηση βασισμένη σε έγγραφα ή σε κλάσεις. Στην πρώτη περίπτωση, στόχος είναι να αναγνωρίσουμε όλες τις κλάσεις από τις οποίες χαρακτηρίζεται ένα έγγραφο, ενώ στη δεύτερη περίπτωση, ψάχνουμε όλα τα έγγραφα που κατηγοριοποιούνται σε αυτή. Επίσης, μπορεί να εφαρμοστεί ιεραρχική ταξινόμηση των εγγράφων. Στο πεδίο ταξινόμησης κειμένων εφαρμόζεται κυρίως η ταξινόμηση σε κλάσεις, όπως χρησιμοποιείται και στη συγκεκριμένη εργασία.

2.3 Επεξεργασία Φυσικής Γλώσσας (NLP)

Η Επεξεργασία Φυσικής Γλώσσας (Natural Language Processing – NLP) αποτελεί ένα διεπιστημονικό πεδίο, με στοιχεία από κλάδους όπως η Γλωσσολογία, η Φωνολογία, αλλά και η Πληροφορική, που ασχολείται με την ανάπτυξη υπολογιστικών μοντέλων και εργαλείων, με σκοπό την κατανόηση και δυνατότητα επεξεργασίας δεδομένων φυσικής γλώσσας [38]. Η διαδικασία αυτή περιλαμβάνει και τη δυνατότητα κατανόησης εννοιών που αντιλαμβάνονται οι άνθρωποι και έχουν να κάνουν με το πλαίσιο του λόγου, ώστε να συνάγονται τα σωστά συμπεράσματα.

Ο τομέας του Natural Language Processing περιλαμβάνει υποκατηγορίες όπως Αναγνώριση Ομιλίας (Speech Recognition), Παραγωγή Φυσικής Γλώσσας (Natural Language Generation) και Ανάκτηση Πληροφορίας (Information Retrieval). Μία πρόκληση στη διαδικασία της επεξεργασίας φυσικής γλώσσας είναι η αποτύπωση του νοήματος που μπορεί να αποκτά η γλώσσα, ανάλογα με τη χρήση της από τους ανθρώπους (όπως το ύφος και οι εκφράσεις στον προφορικό λόγο).

2.4 Προεπεξεργασία Δεδομένων

2.4.1 Θεωρία N – Gram

Στην προηγούμενη ενότητα αναφερθήκαμε στο μειονέκτημα του μοντέλου BOW, το οποίο αγνοεί τη σειρά εμφάνισης των όρων στις προτάσεις ενός εγγράφου. Το πρόβλημα αυτό αντιμετωπίζεται με την τεχνική του N – Gram μοντέλου. Το N – Gram αποτελεί γενίκευση της μεθόδου BOW και είναι μια συνεχόμενη ακολουθία N βασικών μονάδων (Tokens) που αποτελούν ένα έγγραφο. Τα Tokens μπορεί να είναι όροι (Word N – Gram), συλλαβές ή χαρακτήρες όρων (Character N – Gram) με N να είναι το μήκος του παραθύρου. Στην περίπτωση του μοντέλου N – Gram χαρακτήρων αντιμετωπίζονται προβλήματα λάθους συγγραφής των όρων και ορθογραφικά λάθη. Με την χρήση της τεχνικής αυτής, αναγνωρίζουμε εκφράσεις αποτελούμενες από πολλούς όρους. Όσο αυξάνεται το μέγεθος των N – Gram, τόσο αυξάνεται και το μέγεθος του λεξιλογίου και των χαρακτηριστικών διανυσμάτων. Επακόλουθο αυτής της αύξησης είναι η εισαγωγή θορύβου στο μοντέλο, διότι πολλοί από τους επιπλέον όρους δεν περιέχουν σημαντική πληροφορία εγγράφου. Επίσης, καθώς εξετάζεται μια N – Gram φράση και η σημασία της, πρέπει να προσέξουμε και τα σημεία στίξης, τα οποία παίζουν σημαντικό ρόλο. Τέλος, σχετίζεται με την ακολουθία Markov, στην οποία κάθε κατάσταση εξαρτάται μόνο από την προηγούμενή της. Παρακάτω παρουσιάζεται το N – Gram για την πρόταση:

“the dog saw the cat”

Unigrams (N = 1): {“the”, “dog”, “saw”, “the”, “cat”}

Bigrams (N = 2): {“the dog”, “dog saw”, “saw the”, “the cat”}

Trigrams (N = 3): {“the dog saw”, “dog saw the”, “saw the cat”}

2.4.1.1 Skip – Gram

Το μοντέλο αυτό αποτελεί γενίκευση της BOW αναπαράστασης των διανυσμάτων της ακολουθίας των όρων παραλείποντας όρους της. Στην παραπάνω πρόταση με 1 – Skip – Gram – 2 παράγονται τα εξής σύνολα δύο όρων με την παράλειψη ενός όρου:

{“the saw”, “dog the”, “saw cat”}

2.5 Επιλογή Χαρακτηριστικών

2.5.1 Τεχνική Επιλογής Χαρακτηριστικών

Στην ενότητα αυτή θα παρουσιάσουμε μερικές από τις πιο δημοφιλείς τεχνικές επιλογής χαρακτηριστικών για την επεξεργασία φυσικής γλώσσας (Forman George, 2003), (J. Novovicova and A. Malik, 2005).

Η σωστή επιλογή των χαρακτηριστικών στη φάση της επεξεργασίας οδηγεί στη μείωση των διαστάσεων των δεδομένων εισόδου, διατηρώντας αρκετά χαρακτηριστικά για να επιτευχθεί. Ωστόσο, η επιλογή πολύ λίγων ή των μη πραγματικά σημαντικότερων χαρακτηριστικών μπορεί να οδηγήσει σε μειωμένη ακρίβεια του αλγορίθμου.

Οι τεχνικές αυτές εφαρμόζονται στα έγγραφα και τις τοπικές αναπαραστάσεις αυτών. Για την ανάλυση των τεχνικών αυτών θεωρούμε τις εξής παραδοχές:

- Κάθε έγγραφο αποτελείται από όρους, τα λεγόμενα χαρακτηριστικά, από τα οποία ορισμένα μόνο χρησιμοποιούνται. Οι όροι προκύπτουν ανάλογα με την αναπαράσταση, οπότε μπορεί να είναι π.χ. λέξεις ή N – Gram.
- Ένα κείμενο μπορεί να αντιστοιχίζεται σε παραπάνω από μία κλάσεις.

Ο αλγόριθμος επιλογής χαρακτηριστικών αξιολογεί κάθε χαρακτηριστικό με βάση τη χρησιμότητά του και μπορούν να γίνουν οι παρακάτω επιλογές:

- Τα K καλύτερα χαρακτηριστικά με βάση τη χρησιμότητά τους.
- Το ποσοστό των K καλύτερων χαρακτηριστικών με βάση τη χρησιμότητά τους.

- Επιλογή χαρακτηριστικών με βάση κάποιο άνω ή κάτω όριο (Threshold) π.χ. 1000 λέξεις.

Στις τοπικές αναπαραστάσεις, οι τεχνικές επιλογής χαρακτηριστικών είναι:

1. **Συχνότητα Εμφάνισης** που αντιπροσωπεύει το ποσοστό των εγγράφων που εμφανίζεται ένας όρος.
2. **Πλήθος Εμφάνισης** που αντιπροσωπεύει τον αριθμό των εγγράφων, στα οποία εμφανίζεται ένας όρος.
3. **TF – IDF**

2.5.1.1 Μετρικές Αξιολόγησης Χαρακτηριστικών

Η αξιολόγηση χαρακτηριστικών γίνεται με κάποιες μετρικές της θεωρίας της πληροφορίας, όπως η Αμοιβαία Πληροφορία (Mutual Information), το Κέρδος Πληροφορίας (Information Gain) και η μετρική X^2 (Chi – Squared) (Luigi Galavotti et al, 2000).

2.5.1.1.1 Αμοιβαία Πληροφορία (Mutual Information)

Η μετρική αυτή υπολογίζει το μέγεθος της πληροφορίας που περιέχει ένας όρος συγκρινόμενη με την πληροφορία κάθε κλάσης, ώστε ο όρος να ταξινομηθεί σωστά στην συγκεκριμένη κλάση (D. Manning, P. Raghavan, H. Schütze, 2008). Η αμοιβαία πληροφορία δίνεται από την σχέση:

$$MI(t_k, c_j) = log \frac{t_k \wedge c_j}{p(t_k) \times p(c_j)} = log \frac{p(t_k, c_j)}{p(t_k) \times p(c_j)} \quad (6)$$

όπου συγκρίνεται η πιθανότητα εμφάνισης για τα t_k και c_j, καθώς και οι αντίστοιχες πιθανότητες για τα t_k και c_j. Εάν η από κοινού πιθανότητα εμφάνισης $p(t_k, c_j)$ είναι μεγαλύτερη από την $p(t_k) \times p(c_j)$, τότε η κλάση c_j συσχετίζεται με τον όρο t_k.

2.5.1.1.2 Σημειακή Αμοιβαία Πληροφορία (Pointwise Mutual Information)

Βάσει αυτής της μετρικής, υπολογίζεται η αμοιβαία πληροφορία μίας κλάσης και ενός όρου. Η σημειακή αμοιβαία πληροφορία τους δίνεται από τη σχέση:

$$PMI(t_k, c_j) = log \frac{t_k \wedge c_j}{p(t_k) \times p(c_j)} \simeq log \frac{B \times M}{(B + C) \times (B + T)} \quad (7)$$

όπου:

- Β είναι η συχνότητα εμφάνισης του όρου t_k και της κλάσης c_j μαζί στο έγγραφο.
- Μ είναι ο αριθμός εγγράφων που αντιστοιχίζονται στην κλάση c_j.
- C είναι η συχνότητα εμφάνισης της κλάσης c_j χωρίς τον όρο t_k.
- Τ είναι η συχνότητα εμφάνισης του όρου t_k χωρίς την κλάση c_j.

Για να υπολογιστεί η καθολική χρησιμότητα κάθε όρου t_k υπολογίζεται ο μέσος όρος σημειακής αμοιβαίας πληροφορίας για κάθε κατηγορία.

$$PMI_{avg}(t_k) = \sum_{j=1}^{|C|} p(c_j) PMI(t_k, c_j) \quad (8)$$

2.5.1.1.3 Κέρδος Πληροφορίας (Information Gain)

Η μετρική αυτή υπολογίζει την πληροφορία που λαμβάνεται, αν γνωρίζουμε ότι ένας όρος εμφανίζεται ή απουσιάζει από ένα κείμενο.

$$IG(t_k, c_j) = \sum_{c \in \{c_k, \overline{c_j}\}} \sum_{t \in \{t_k, \overline{t_k}\}} p(t, c) log \frac{p(t, c)}{p(t) p(c)} \quad (9)$$

2.5.1.1.4 X^2 (Chi-Squared)

Η x^2 για έναν ποιοτικό όρο, ποσοτικοποιεί το πόσο διαφέρει η συχνότητα με την οποία εμφανίζεται από τη συχνότητα με την οποία αναμένεται να εμφανιστεί. Έτσι, υπολογίζεται πόσο στενά σχετίζεται κάθε όρος με μία κλάση c_j:

$$x^2(t_k, c_j) = \frac{|T|\{p(t_k, c_j)p(\overline{t_k}, \overline{c_j}) - p(t_k, \overline{c_j})p(\overline{t_k}, c_j)\}^2}{p(t_k)p(\overline{t_k})p(c_j)p(\overline{c_j})}$$

(10)

2.5.2 Εξαγωγή Χαρακτηριστικών

Στην εξαγωγή χαρακτηριστικών (Feature Extraction), αναπαριστούμε τα αντικείμενα προς ταξινόμηση σε κλάσεις, σε διανύσματα χαρακτηριστικών. Η διαδικασία αυτή μπορεί να είναι πολύ απλή, αλλά και εξίσου σύνθετη. Προσπαθούμε μέσω ενός αρχικού συνόλου να δημιουργήσουμε ένα νέο σύνολο σύνθετων χαρακτηριστικών, με σκοπό να βελτιωθεί η απόδοση του αλγορίθμου ταξινόμησης. Σκοπός είναι να ανακαλυφθούν συσχετίσεις μεταξύ των χαρακτηριστικών, οι οποίες μπορούν να εκφραστούν τελικά μέσα από νέα χαρακτηριστικά που θα εξαχθούν. Τέτοιες σχέσεις που μπορεί να παρατηρηθούν στην επεξεργασία φυσικής γλώσσας είναι η συνωνυμία ή η ύπαρξη όρων που μπορεί να εμφανίζονται μαζί ως φράσεις.

2.5.2.1 Μείωση Διαστάσεων

Δύο κύριες τεχνικές για την μείωση διαστάσεων αποτελούν η Ανάλυση Κύριων Συνιστωσών (PCA) και η Ανάλυση σε Ιδιάζουσες Τιμές (SVD). Όσον αφορά την SVD (Scott Deerwester et al, 1990), υπολογίζεται ένα σύνολο R διανυσμάτων (όρων) ως ο γραμμικός συνδυασμός Μ διανυσμάτων. Αντιθέτως, η PCA βρίσκει τις δύο κύριες συνιστώσες στις οποίες παρατηρείται η μέγιστη διασπορά των υπόλοιπων

χαρακτηριστικών, ελαχιστοποιώντας τα αθροίσματα των τετραγώνων τους. Οι τεχνικές αυτές γενικά, συμπιέζουν τις διαστάσεις με μικρή διασπορά και διατηρούν αυτές με μεγάλη διασπορά. Κατά την διαδικασία μείωσης διαστάσεων, ένα μέρος της πληροφορίας χάνεται, επομένως δεν είναι εφικτό να ανακατασκευαστούν οι αρχικές διαστάσεις. Στη νέα απεικόνιση, στο νέο χώρο διαστάσεων, τα νέα χαρακτηριστικά εκφράζουν τις σχέσεις ανάμεσα στα αρχικά χαρακτηριστικά.

2.6 Κατηγορίες Αλγορίθμων Μηχανικής Μάθησης στην Ταξινόμηση Κειμένου

Υπάρχει μια μεγάλη ποικιλία αλγορίθμων ταξινόμησης κειμένου που συναντάμε στην βιβλιογραφία, οι οποίοι μπορούν να χωριστούν σε κατηγορίες, ανάλογα με το πρόβλημα που αντιμετωπίζουν. Η συγκεκριμένη εργασία αξιοποιεί μερικές κατηγορίες.

2.6.1 Naïve Bayes

Οι ταξινομητές Naïve Bayes αποτελούν ένα σύνολο απλών στατιστικών ταξινομητών που χρησιμοποιούν μια απλή εφαρμογή του θεωρήματος Bayes, με ορισμένες παραδοχές για ανεξαρτησία μεταξύ των χαρακτηριστικών εισόδου. Ο όρος “Naïve” αναφέρεται ακριβώς σ’ αυτή την παραδοχή, δηλαδή όλοι οι συγκεκριμένοι ταξινομητές κάνουν την παραδοχή ότι η τιμή κάθε χαρακτηριστικού δεν εξαρτάται από τα υπόλοιπα χαρακτηριστικά που κατηγοριοποιούνται στην ίδια κλάση. Το θεώρημα πιθανότητας του Bayes δηλώνει την ακόλουθη σχέση μεταξύ μιας δοθείσας ανεξάρτητης μεταβλητής (κλάσης, y) και του ανεξάρτητου διανύσματος χαρακτηριστικών $\boldsymbol{d} = (d_1, \dots, d_n)$:

$$P(y|\boldsymbol{d}) = \frac{P(y)P(\boldsymbol{d}|y)}{P(\boldsymbol{d})}$$

(11)

όπου:

- $P(y)$, η εκ των προτέρων πιθανότητα (Prior Probability) της κάθε κλάσης που προσδιορίζει την πιθανότητα εμφάνισης της y και υπολογίζεται ως εξής:

$$P(y_i) = \frac{αριθμός\ παρατηρήσεων\ κλάσης\ y_i}{αριθμός\ κλάσεων}$$

(1)

- $P(\boldsymbol{d})$, είναι η εκ των προτέρων πιθανότητα (Prior Probability) του κάθε εγγράφου **d** και προσδιορίζει την πιθανότητα εμφάνισης του **d**.
- $P(y|\boldsymbol{d})$, η εκ των υστέρων πιθανότητα (Posterior Probability) και προσδιορίζει την πιθανότητα κάποιο από τα έγγραφα του διανύσματος **d** να ανήκει στην κλάση y.
- $P(\boldsymbol{d}|y)$ προσδιορίζει την πιθανοφάνεια (Likelihood) και είναι η πιθανότητα παρατήρησης κάποιου εγγράφου του διανύσματος **d** δεδομένου ότι έχει ταξινομηθεί στην κλάση y. Διαισθητικά η πιθανοφάνεια απαντά στο ερώτημα «ποια η πιθανότητα παρατήρησης ενός εγγράφου με τα χαρακτηριστικά του εγγράφου **d** στην κλάση y».

Όσον αφορά τις εκ των προτέρων πιθανότητες $P(y_i)$, οι κλάσεις θεωρούμε ότι εμφανίζονται με ίση πιθανότητα:

$$P(y_i) = \frac{1}{αριθμός\ κλάσεων} = \frac{1}{k}, \forall i = 1, \dots, k$$

(2)

Επίσης, ένα έγγραφο αποτελείται από πλήθος χαρακτηριστικών – όρων, $\boldsymbol{d} = \{x_1, \dots, x_n\}$. Χρησιμοποιώντας την «αφελή» παραδοχή της ανεξαρτησίας των ενδεχομένων και τη σειρά των όρων (BOW) που δεν έχει σημασία, η παράσταση υπολογίζεται ως το γινόμενο από τις ανεξάρτητες μεταξύ τους δεσμευμένες πιθανότητες:

$$P(\boldsymbol{d}|y) = P(x_1|y) \dots P(x_n|y)$$

(3)

Για κάθε i, η σχέση (11) απλοποιείται ως εξής:

$$P(y|\boldsymbol{d}) = \frac{P(y)\prod_{i=1}^{n} P(x_i|y)}{P(\boldsymbol{d})}$$

(4)

Επειδή $P(\boldsymbol{d})$ είναι σταθερή δεδομένης της εισόδου και προσδιορίζει την εκ των προτέρων πιθανότητα (Prior Probability) για τα χαρακτηριστικά – όρους του κάθε εγγράφου, υπολογίζεται τελικά η κλάση $\hat{y}_{map}$ με την μέγιστη εκ των υστέρων πιθανότητα (Maximum a Posteriori).

$$P(y|\boldsymbol{d}) \propto P(y)\prod\nolimits_{i=1}^{n} P(x_i|y)$$

(5)

$$\Downarrow$$

$$\hat{y}_{map} = \underset{y}{argmax}\, P(y)\prod\nolimits_{i=1}^{n} P(x_i|y)$$

(17)

Συνεπώς, στην ταξινόμηση ενός εγγράφου υπολογίζουμε την $\hat{y}_{map}$ και αυτό μας δείχνει διαισθητικά την εκτίμηση της $P(y)$ και $P(x_i|y)$, με την πρώτη να αποτελεί την συχνότητα εμφάνισης της κλάσης y στο σύνολο εκπαίδευσης. Ο Naïve Bayes, παρά τις παραδοχές που πρέπει να γίνουν για ανεξαρτησία των χαρακτηριστικών, χρησιμοποιείται εκτενώς λόγω της καλής απόδοσής του και της ακρίβειάς του. Η φάση της εκπαίδευσής του ουσιαστικά στοχεύει στον υπολογισμό μίας εκτίμησης για τις συναρτήσεις πυκνότητας πιθανότητας $P(\boldsymbol{d}|y_i)$, δηλαδή των $P(x_j|y_i)$ για j=1,2,…,n και i=1,2,…,k αφού κάνουμε την παραδοχή για ανεξαρτησία μεταξύ των χαρακτηριστικών. Η μέθοδος αυτή προκύπτει αν θεωρήσουμε μία τυπική κατανομή για τα δεδομένα και προσεγγίσουμε τον υπολογισμό των παραμέτρων αυτής της κατανομής με την προσέγγιση μέγιστης πιθανοφάνειας (Maximum Likelihood) (Zhang, 2004). Υπάρχουν αρκετές παραλλαγές του ταξινομητή Naïve Bayes (Multinomial, Complement, Gaussian, Bernoulli), τις οποίες θα αναλύσουμε παρακάτω.

2.6.1.1 Gaussian Naïve Bayes

Θεωρούμε ότι για κάθε χαρακτηριστικό εισόδου x_j , j=1,2,…,n και κάθε κλάση y_i, i=1,2,…,k, η συνάρτηση πυκνότητας πιθανότητας είναι η Gaussian (κανονική κατανομή), δηλαδή:

$$P(x_j|y_i) = \frac{1}{\sqrt{2\pi{\sigma_{ji}}^2}} exp\left(-\frac{x-\mu_{ji}}{2{\sigma_{ji}}^2}\right)^2$$

(18)

2.6.1.2 Multinomial Naïve Bayes

Οι συγκεκριμένοι ταξινομητές χρησιμοποιούνται εκτενώς σε εφαρμογές ταξινόμησης κειμένου και συγκεκριμένα σε αναπαραστάσεις BOW. Η κατανομή του συνόλου των παρατηρήσεων παραμετροποιείται από τα διανύσματα $\boldsymbol{\theta_y} = (\theta_{y1}, \dots, \theta_{yn})$ για κάθε κλάση y, όπου n είναι το πλήθος των χαρακτηριστικών (στην ταξινόμηση εγγράφων, είναι το μέγεθος του λεξικού). Το θ_{yi} είναι η δεσμευμένη πιθανότητα $P(x_i|y_i)$ του χαρακτηριστικού i που εμφανίζεται στο δείγμα της κλάσης y_i. Οι παράμετροι διανυσμάτων $\boldsymbol{\theta_y}$ πραγματοποιούν εκτιμήσεις με μια παραλλαγή της μέγιστης πιθανοφάνειας (Maximum Likelihood – M):

$$\hat{\theta}_{yi} = \frac{N_{yi} + a}{N_y + an}$$

(19)

όπου $N_{yi} = \sum_{x \in T} x_i$ το πλήθος εμφάνισης του χαρακτηριστικού i σε δείγμα που ανήκει στην κλάση y στο σύνολο εκπαίδευσης T, και $N_y = \sum_{i=1}^{n} N_{yi}$ το συνολικό πλήθος όλων των χαρακτηριστικών – όρων που ανήκουν στην κλάση y. Είναι συχνά επιθυμητό να ενσωματώνεται μια διόρθωση μικρού δείγματος, ένας συντελεστής εξομάλυνσης $a \geq 0$, ο οποίος καταγράφει χαρακτηριστικά που δεν υπάρχουν στα δείγματα του συνόλου εκπαίδευσης και αποτρέπει τις μηδενικές πιθανότητες. Αυτή η ενσωμάτωση

ονομάζεται ψευδομέτρηση. Θέτοντας $a = 1$, έχουμε εξομάλυνση Laplace και $a < 1$ εξομάλυνση Lidstone (Manning, C.D. et al, 2008).

2.6.1.3 Complement Naïve Bayes

Μια πολύ διαδεδομένη παραλλαγή του Multinomial NB, ο Complement NB χρησιμοποιείται σε περιπτώσεις που υπάρχει ανισορροπία μεταξύ των κλάσεων. Ειδικότερα, ο Complement NB χρησιμοποιεί το χαρακτηριστικό του συμπληρώματος κάθε κλάσης για τον υπολογισμό των βαρών του μοντέλου. Οι εκτιμήσεις των παραμέτρων του Complement NB παρουσιάζουν μεγαλύτερη αξιοπιστία από αυτές του MNB. Η διαδικασία υπολογισμού των βαρών δίνεται από τις παρακάτω σχέσεις:

$$\hat{\theta}_{ci} = \frac{a_i + \sum_{j:y_j \neq c} d_{ij}}{a + \sum_{j:y_j \neq c} \sum_k d_{kj}}$$

(20)

$$w_{ci} = log\hat{\theta}_{ci}$$

(21)

$$w_{ci} = \frac{w_{ci}}{\sum_j |w_{ci}|}$$

(22)

Στην πρώτη σχέση αθροίζονται τα χαρακτηριστικά των j εγγράφων που δεν ανήκουν στην κλάση c, οπότε τα d_{ij} προσδιορίζουν το πλήθος εμφάνισης ή τις TF – IDF τιμές του i – οστού όρου στο έγγραφο j, το α είναι παράμετρος εξομάλυνσης, όπως στον MNB και είναι:

$$a = \sum_i a_i$$

(23)

Η δεύτερη σχέση ομαλοποίησης αντιμετωπίζει την τάση κυριαρχίας μεγαλύτερων σωμάτων εγγράφων, να κυριαρχούν στις εκτιμήσεις παραμέτρων του MNB. Ο κανόνας ταξινόμησης δίνεται από τη σχέση:

$$\hat{c} = \underset{c}{argmin} \sum_{i} t_i w_{ci}$$

(24)

Ο παραπάνω κανόνας μας δίνει την κλάση εκχώρησης κειμένων με την ελάχιστη αντιστοίχισή συμπληρώματος (Rennie, J.et al, 2003).

2.6.1.4 Bernoulli Naïve Bayes

Οι ταξινομητές Bernoulli Naïve Bayes είναι επίσης διαδεδομένοι σε εφαρμογές ταξινόμησης εγγράφων. Τα δεδομένα επεξεργάζονται δυαδικά χαρακτηριστικά, επειδή ακολουθούν κατανομή Bernoulli. Τα δείγματα αναπαρίστανται ως διανύσματα δυαδικών τιμών (0 ή 1). Όπως συμβαίνει στις μεταβλητές που ακολουθούν κατανομή Bernoulli, όταν συναντάται το 1 στο διάνυσμα κάποιου δείγματος, σημαίνει ότι το ενδεχόμενο που αναπαριστά η συγκεκριμένη θέση αληθεύει για το δείγμα, ενώ το 0 ότι δε συμβαίνει. Στην περίπτωση εφαρμογής BOW, τα ενδεχόμενα εκφράζουν παρουσία ή μη των όρων στο εκάστοτε έγγραφο (Term Occurrence), οπότε σε αυτό το μοντέλο NB δε μας απασχολεί η συχνότητα εμφάνισης του όρου. Αν τα χαρακτηριστικά των εγγράφων αναπαρίστανται με δυαδικές κλάσεις, μπορεί να χρησιμοποιηθεί ο NB για εκτίμηση της συνάρτησης πιθανότητας με Bernoulli:

$$P(x|y_i) = \prod_{j=1}^{n} {p_{ij}}^{x_j} \left(1 - p_{ij}\right)^{1-x_j}$$

(25)

όπου p_{ij} η πιθανότητα η κλάση y_i να παράγει το ενδεχόμενο i. Ο απλοϊκός ταξινομητής Bernoulli Naïve Bayes στην ταξινόμηση κειμένων λαμβάνει υπόψη τον αριθμό εμφάνισης ενός χαρακτηριστικού – όρου σε ένα διάνυσμα και όχι το πλήθος των όρων από ένα διάνυσμα. Ο ταξινομητής αυτός αποδίδει καλύτερες προβλέψεις ετικετών σε σύνολα δεδομένων με μικρά σώματα εγγράφων (V. Metsis, I. Androutsopoulos, G. Paliouras, 2006), (Manning, C.D. et al, 2008).

2.6.2 Δένδρα Απόφασης (Decision Trees)

Η χρήση της ταξινόμησης με δένδρα απόφασης είναι συχνή, επειδή το μοντέλο εκπαίδευσης αναπαρίσταται ως δενδρική δομή, ώστε να είναι κατανοητή η διαδικασία ταξινόμησης. Σε κάθε δένδρο διακρίνονται δύο είδη κόμβων. Οι εσωτερικοί κόμβοι περιέχουν γνωρίσματα δεδομένων, των οποίων το καθήκον είναι ο έλεγχος. Απ' την άλλη, οι τερματικοί κόμβοι αναθέτουν στα στιγμιότυπα που καταλήγουν σ' αυτούς μια ετικέτα κλάσης που πιθανώς ανήκουν. Σε κάθε ακμή που φεύγει από κάποιο κόμβο αναγράφονται οι κλάσεις των χαρακτηριστικών του κόμβου, εκτός και αν η ακμή καταλήγει στην τελική κλάση του στιγμιότυπου.

Δεδομένου ενός συνόλου δεδομένων εκπαίδευσης, χρησιμοποιείται μια συγκεκριμένη διαδικασία για την δημιουργία ενός δένδρου απόφασης. Σε πρώτο στάδιο, επιλέγεται ένα χαρακτηριστικό ως ρίζα του δένδρου, το οποίο διαχωρίζει καλύτερα τα στιγμιότυπα ως προς την συγκεκριμένη κλάση. Στη συνέχεια, τα δεδομένα χωρίζονται ως προς τις διαθέσιμες κλάσεις του συγκεκριμένου χαρακτηριστικού. Για κάθε υποσύνολο που περιέχει στιγμιότυπα διαφορετικών κλάσεων, επαναλαμβάνεται η διαδικασία ως ότου εξαντληθούν όλα τα χαρακτηριστικά του συνόλου εκπαίδευσης και κάθε υποσύνολο περιέχει μόνο μία κλάση.

Έστω ότι έχουμε ένα στιγμιότυπο και αναζητούμε την κλάση σε ένα δένδρο απόφασης με βάση το σύνολο εκπαίδευσης. Τότε η διαδικασία για την ταξινόμηση είναι αρκετά απλή. Αρχίζουμε από τη ρίζα του δένδρου και προχωράμε στην ακμή που μας οδηγεί στην σωστή ετικέτα (κλάση) για το χαρακτηριστικό που εξετάζουμε. Έπειτα, πηγαίνουμε στον επόμενο κόμβο και επαναλαμβάνουμε την ίδια διαδικασία μέχρι να φτάσουμε στον τελικό κόμβο που οδηγεί και στην κλάση του στιγμιότυπου.

Δεδομένου ενός συνόλου απόφασης, μπορούμε να δημιουργήσουμε διαφορετικά δένδρα. Το κριτήριο που τα διαφοροποιεί είναι η διαφορετική θέση των χαρακτηριστικών στους κόμβους. Η επιλογή του κατάλληλου χαρακτηριστικού γίνεται ανάλογα με το πόσο καλά διαχωρίζονται τα δείγματα με βάση τις κλάσεις. Έτσι, υπάρχουν αρκετοί αλγόριθμοι για την κατασκευή ενός δέντρου απόφασης, όπως ο ID3 και ο C4.5 που θα δούμε παρακάτω. Οι αλγόριθμοι αυτοί ακολουθούν την ίδια διαδικασία για τη δημιουργία δένδρου και διαφοροποιούνται στο κριτήριο με το οποίο

επιλέγουν τα χαρακτηριστικά σε κάθε κόμβο για να επιτύχουν καλύτερο διαχωρισμό [39], [40].

2.6.2.1 Αλγόριθμος ID3

Ο αλγόριθμος αυτός κάνει χρήση δύο μετρικών, του κέρδους πληροφορίας (Information Gain) και της εντροπίας (Entropy). Η πρώτη μετρική χρησιμοποιείται για τον καλύτερο διαχωρισμό των δεδομένων. Απ' την άλλη, η δεύτερη μετράει την ποσότητα του χάους σε κάθε υποσύνολο. Όσο πιο μεγάλη είναι η εντροπία, τόσο πιο χαοτικό είναι το υποσύνολο προς εξέταση. Μπορούμε να λάβουμε την εντροπία ως μέτρο «καθαρότητας», διότι υποσύνολα με χαμηλή εντροπία περιέχουν δεδομένα που τα περισσότερα ανήκουν στην ίδια κλάση. Δεδομένου ενός συνόλου παρατηρήσεων D, η εντροπία υπολογίζεται ως εξής:

$$E(D) = -\sum_{i=1}^{k} P(c_i|D) log_2 P(c_i|D)$$

(26)

όπου $P(c_i|D)$ είναι η πιθανότητα εμφάνισης της κλάσης c_i στο σύνολο D και k το πλήθος των κλάσεων της κλάσης. Για να επιβεβαιώσουμε ότι η επιλογή του χαρακτηριστικού ελαττώνει την συνολική εντροπία, ορίζουμε το κέρδος πληροφορίας για ένα συγκεκριμένο χαρακτηριστικό ως εξής:

$$InformationGain(X_i, D) = E(D) - \sum_{i,j} P(X_{ij}) E(X_i)$$

(27)

όπου X_i το κάθε χαρακτηριστικό του συνόλου D και X_{ij} κάθε κλάση του χαρακτηριστικού. Συνεπώς, κάθε φορά επιλέγουμε εκείνο το χαρακτηριστικό που διαθέτει το μεγαλύτερο κέρδος πληροφορίας και την μικρότερη εντροπία με αποτέλεσμα την αύξηση της «καθαρότητας» [39], [40], [41].

2.6.2.2 Αλγόριθμος J48 ή C4.5

Ο αλγόριθμος J48 είναι μια βελτιωμένη έκδοση του προηγούμενου ID3, αφού παρουσιάζει καλύτερη απόδοση στη διαδικασία «κλαδέματος» δέντρων (Pruning) και τελικά στην εξαγωγή κανόνων. Επίσης, παρουσιάζει βελτιωμένα αποτελέσματα ειδικά σε κείμενα με αριθμητικά δεδομένα και κενά πεδία. Κατά την κατασκευή του δένδρου απόφασης, ο αλγόριθμος εστιάζει στα δεδομένα που έχουν τιμή, οπότε δεν επηρεάζεται από την έλλειψη τιμής στα υπόλοιπα. Επίσης, κατά την ταξινόμηση ενός ελλιπούς δεδομένου ως προς ένα χαρακτηριστικό, προβλέπεται η τιμή του βάσει των υπόλοιπων δεδομένων ως προς το συγκεκριμένο χαρακτηριστικό. Όσον αφορά τα συνεχή δεδομένα, τα χαρακτηριστικά που αντιπροσωπεύουν συνεχή μεγέθη είναι εύκολο να διαχωριστούν στα αντίστοιχα συνεχή διαστήματα. Στο κλάδεμα, ο αλγόριθμος χρησιμοποιεί δύο σημαντικές τεχνικές. Η μία αφορά την αντικατάσταση υποδένδρου (Subtree Replacement) από φύλλο, το οποίο επιλέγεται αν το σφάλμα του νέου υποδένδρου είναι κοντά στο σφάλμα του αρχικού. Η άλλη αφορά την ανύψωση υποδένδρου (Subtree Raising), όπου ένα υποδένδρο που βρίσκεται σε υψηλότερο επίπεδο, αντικαθίσταται με το περισσότερο χρησιμοποιούμενο υποδένδρο του, λαμβάνοντας υπόψη την αύξηση στη συχνότητα λαθών. Στην περίπτωση της διάσπασης, ο αλγόριθμος κρατάει τα χαρακτηριστικά που οδηγούν σε πολλές διαιρέσεις. Το πρόβλημα σε αυτή την περίπτωση είναι ότι το μοντέλο μπορεί να παρουσιάσει υπερπροσαρμογή στο σύνολο εκπαίδευσης, γεγονός που μπορεί να αντιμετωπιστεί λαμβάνοντας υπόψη την παράμετρο της πληθικότητας. Στη διάσπαση ως μετρική αξιολόγησης χρησιμοποιείται ο λόγος κέρδους (Gain Ratio). Ο λόγος κέρδους υπολογίζεται ως το πηλίκο του κέρδους πληροφορίας ως προς την πληροφορία διαχωρισμού (Split Information) και συγκεκριμένα:

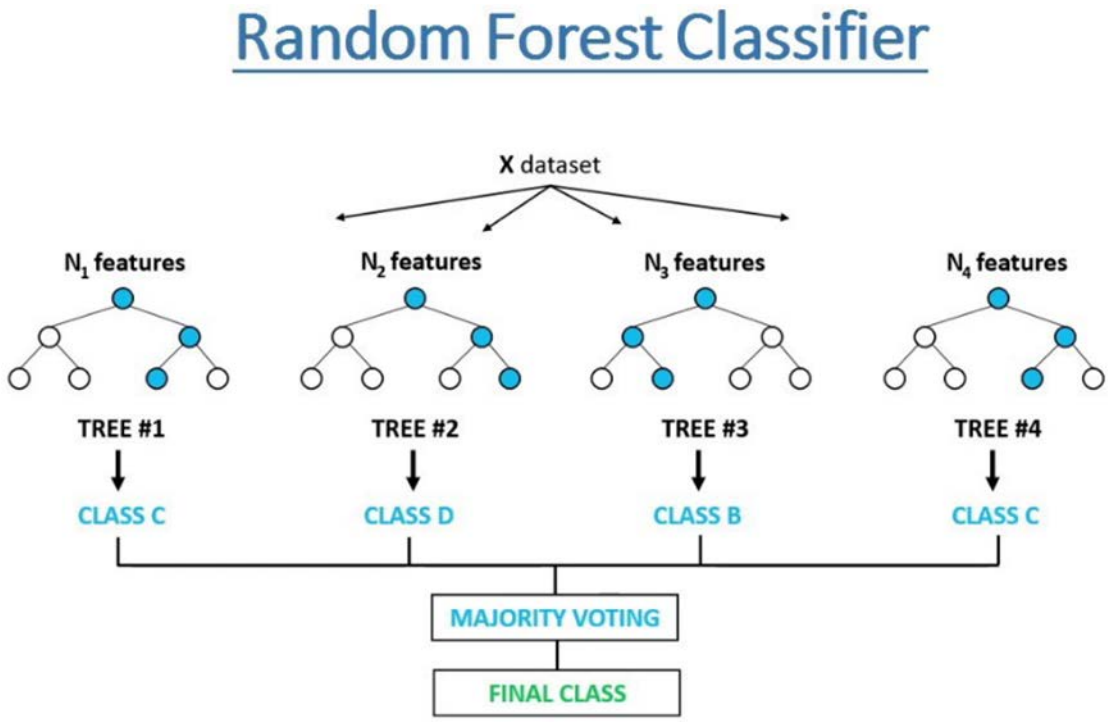
$$GainRatio(X_i, D) = \frac{InformationGain(X_i, D)}{SplitInformation(X_i, D)} \quad (28)$$

όπου $SplitInformation(X_i, D) = -\sum_{i,j} \frac{|X_{ij}|}{|X_i|} log_2 \frac{|X_{ij}|}{|X_i|}$. Ο διαχωρισμός πληροφορίας εκφράζει το βαθμό κατά τον οποίο ένας όρος οδηγεί το σύνολο σε διαχωρισμό. Ο λόγος κέρδους καταφέρνει να επιλέξει το χαρακτηριστικό με τη μεγαλύτερη ικανότητα

διαχωρισμού των παρατηρήσεων, αποφεύγοντας χαρακτηριστικά που τείνουν να δημιουργήσουν δένδρα με πολλές διακλαδώσεις [39], [40], [41].

2.6.2.3 Τυχαία Δάση (Random Forests)

Μια ειδική κατηγορία των συνδυαστικών μεθόδων ταξινόμησης αποτελούν τα τυχαία δάση (Random Forests), τα οποία χρησιμοποιούν για ταξινομητές δένδρα απόφασης (Ho, 1995), (L. Breiman, 2001). Ο Random Forest αποτελεί έναν αλγόριθμο που χρησιμοποιείται κυρίως σε προβλήματα ταξινόμησης. Τα δείγματα επιλέγονται από ένα σύνολο δεδομένων με τυχαίο τρόπο. Έπειτα, δημιουργείται ένα δένδρο απόφασης για κάθε δείγμα και υπολογίζεται η πρόβλεψη κλάσης για κάθε δένδρο. Τέλος, μέσω ψηφοφορίας επιλέγει το πιο υψηλό αποτέλεσμα πρόβλεψης. Το πλεονέκτημα έναντι του δέντρου απόφασης είναι ότι συνδυάζει τα αποτελέσματα από πολλά δένδρα απόφασης και αποφεύγει την υπερπροσαρμογή. Επίσης, τα τυχαία δάση έχουν μικρότερη διακύμανση, είναι ευέλικτα και πετυχαίνουν πολύ υψηλή ακρίβεια ακόμα και με έλλειψη ενός μεγάλου μέρους δεδομένων. Το κύριο μειονέκτημα αυτού του αλγόριθμου είναι η πολυπλοκότητα, καθώς για την εφαρμογή του απαιτούνται πολλοί υπολογιστικοί πόροι. Επιπλέον, η διαδικασία πρόβλεψης και κατασκευής τέτοιων τυχαίων δασών είναι δύσκολη και χρονοβόρα.



Εικόνα 7: Ταξινομητής Random Forest[6]

2.6.3 Μηχανές Διανυσμάτων Υποστήριξης (SVM)

Οι Μηχανές Διανυσμάτων Υποστήριξης (Support Vector Machines – SVM) αποτελούν μοντέλα που λειτουργούν με στόχο την εύρεση ενός υπερεπιπέδου σε έναν N – διάστατο χώρο (όπου N ο αριθμός χαρακτηριστικών), ώστε να τα διαχωρίσει αποτελεσματικά σε κλάσεις, βρίσκοντας το βέλτιστο υπερεπίπεδο μεταξύ όλων των πιθανών, δηλαδή αυτό που μεγιστοποιεί το περιθώριο για διαχωρισμό των κλάσεων [42]. Η SVM μπορεί να διαχωριστεί ανάλογα με το είδος του πυρήνα:

- **Γραμμική SVM:** Επιλύει γραμμικά διαχωρίσιμα προβλήματα, με διαχωρισμό δύο κλάσεων γραμμικά με ένα υπερεπίπεδο.
- **Μη γραμμική SVM:** Για μη γραμμικά διαχωρίσιμα προβλήματα, χρησιμοποιούνται συναρτήσεις πυρήνα (Kernel Functions) για απεικόνιση των χαρακτηριστικών εισόδου σε χώρους μεγαλύτερων διαστάσεων.

[6] https://medium.com/analytics-vidhya/random-forest-classifier-and-its-hyperparameters-8467bec755f6

2.6.3.1 Γραμμική SVM

Όπως αναφέρθηκε, διαχωρίζει τα δεδομένα γραμμικά με υπερεπίπεδο, ωστόσο μπορεί να χρησιμοποιηθεί και σε μη διαχωρίσιμες κλάσεις. Εν γένει προορίζεται για προβλήματα δυαδικής ταξινόμησης (2 – Class Classification), ωστόσο μπορεί να χρησιμοποιηθεί και για ταξινόμηση πολλών κλάσεων (Multi – Class Classification), αν θεωρήσουμε ως ανεξάρτητο πρόβλημα τη δυαδική ταξινόμηση αναγνώρισης ή μη της καθεμίας από τις κλάσεις.

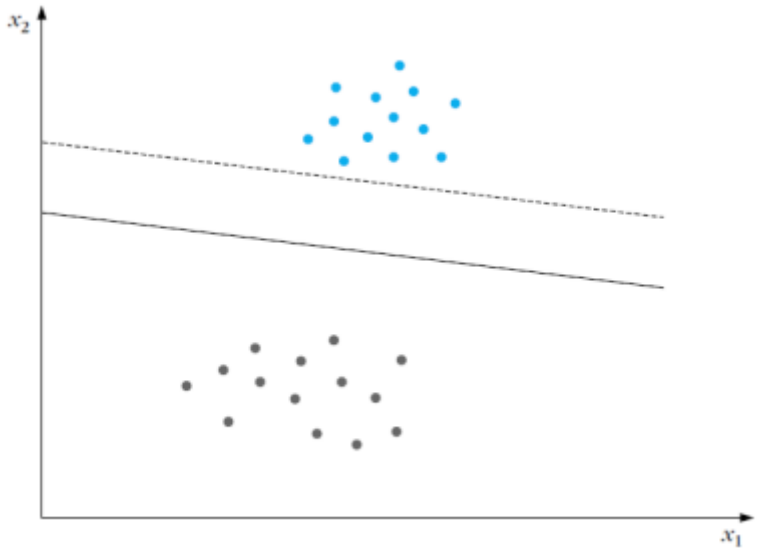
2.6.3.1.1 Διαχωρίσιμες Κλάσεις

Διαχωρίσιμες κλάσεις θεωρούνται αυτές, οι οποίες διαχωρίζονται πλήρως από ένα υπερεπίπεδο. Έστω $x_i,\ i = 1, \dots, N$ τα διανύσματα που αναπαριστούν τα δεδομένα εκπαίδευσης, X. Αυτά τα διανύσματα κατηγοριοποιούνται σε μία από τις δύο κλάσεις ω_1, ω_2 και θεωρείται ότι είναι γραμμικά διαχωρίσιμα. Στόχος είναι ο σχεδιασμός ενός υπερεπιπέδου:

$$g(\boldsymbol{x}) = \boldsymbol{w}^T \boldsymbol{x} + w_0 = 0$$

(29)

Το οποίο διαχωρίζει σωστά τα διανύσματα εκπαίδευσης. Όπως φαίνεται στην Εικόνα 8, το υπερεπίπεδο δεν είναι μοναδικό.

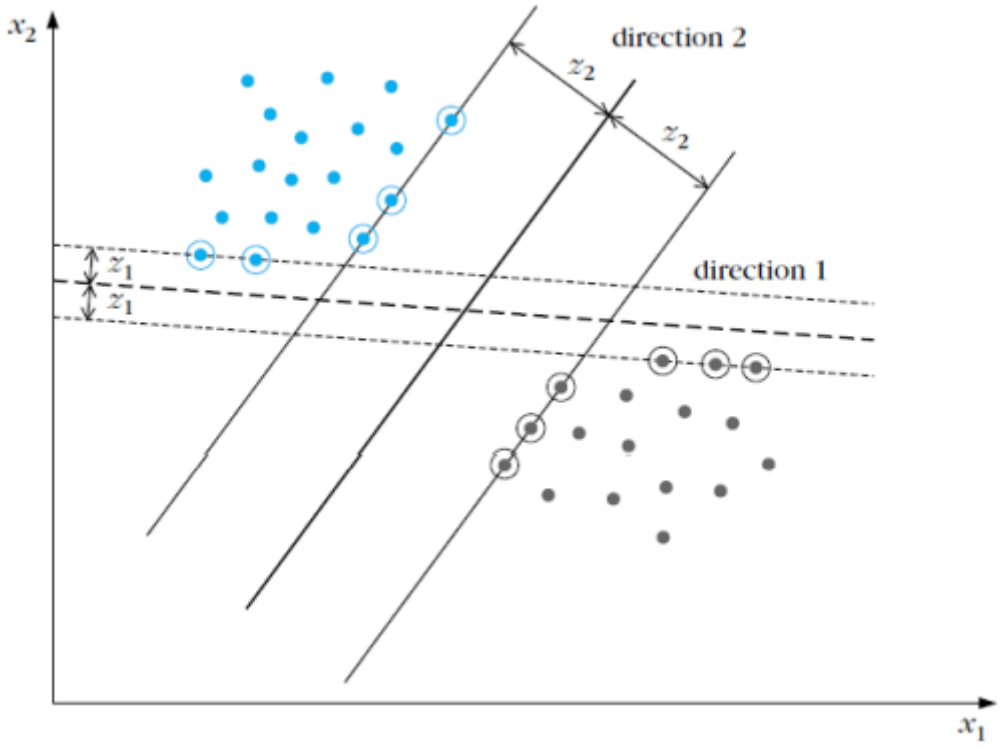


Εικόνα 8: Παραδείγματα Υπερεπιπέδων [43]

Σ' αυτήν την περίπτωση, το ερώτημα που τίθεται είναι ποιο υπερεπίπεδο είναι το κατάλληλο ως λύση του προβλήματος. Διαισθητικά, το κατάλληλο υπερεπίπεδο είναι αυτό που δημιουργεί ίσο περιθώριο μεταξύ των δύο κλάσεων ω_1, ω_2 (δηλαδή μεγιστοποιεί την απόσταση του περιθωρίου από τα κοντινότερα σε αυτό σημεία των δύο κλάσεων), καθώς όπως φαίνεται θα είναι λιγότερο ευαίσθητη σε νέα δεδομένα.

Από την σχέση (29), συμπεραίνεται ότι κάθε υπερεπίπεδο προσδιορίζεται από τον προσανατολισμό του από την παράμετρο **w** και από την μετατόπισή του που προκύπτει από το w_0.

Στην Εικόνα 9, η απόσταση του υπερεπιπέδου 1 είναι $2z_1$ και του υπερεπιπέδου 2 είναι $2z_2$. Στόχος αποτελεί η εύρεση ενός υπερεπιπέδου (μέσω των παραμέτρων που το προσδιορίζουν), το οποίο μεγιστοποιεί το περιθώριο. Η απόσταση κάθε σημείου από το υπερεπίπεδο διαχωρισμού ορίζεται ως: $z = \frac{g(x)}{\|w\|}$. Σε αυτό το σημείο θα ήταν εύχρηστο να εφαρμοστεί κανονικοποίηση στις παραμέτρους που χαρακτηρίζουν το υπερεπίπεδο, ώστε η απόσταση από το υπερεπίπεδο για το κοντινότερο σε αυτό σημείο κάθε κλάσης να είναι -1 και 1 αντίστοιχα.



Εικόνα 9: Παράδειγμα Διαχωρίσιμων Κλάσεων [43]

Διανύσματα υποστήριξης (Support Vectors) ονομάζονται τα κοντινότερα σημεία των κλάσεων που καθορίζουν τη θέση του υπερεπιπέδου, ενώ όλος ο ταξινομητής βέλτιστου υπερεπιπέδου είναι η Μηχανή Διανυσμάτων Υποστήριξης (Support Vector Machine).

Με βάση τα παραπάνω, προκύπτει ότι αν για το περιθώριο ισχύει:

$$\frac{1}{\|\boldsymbol{w}\|}+\frac{1}{\|\boldsymbol{w}\|}=\frac{2}{\|\boldsymbol{w}\|} \tag{30}$$

Αναγκαστικά πληρούνται οι εξής περιορισμοί:

$$\boldsymbol{w}^T\boldsymbol{x}+w_0 \geq 1, \forall \boldsymbol{x} \in \omega_1 \tag{31}$$

$$\boldsymbol{w}^T\boldsymbol{x}+w_0 \leq -1, \forall \boldsymbol{x} \in \omega_2 \tag{32}$$

Για κάθε $\boldsymbol{x_i}$ διάνυσμα, η κλάση στην οποία κατατάσσεται, ορίζεται ως y_i (1 για την ω_1 και -1 για την ω_2). Οπότε προσπαθούμε να υπολογίσουμε τις παραμέτρους $\boldsymbol{w}, w_0$ του υπερεπιπέδου ώστε:

- Να ελαχιστοποιείται το:

$$J(\boldsymbol{w}, w_0) = \frac{1}{2}\|\boldsymbol{w}\|^2$$

(33)

- δοθέντος του:

$$y_i(\boldsymbol{w}^T\boldsymbol{x_i} + w_0) \geq 1, i = 1, \dots, N$$

(34)

Για να μεγιστοποιηθεί το περιθώριο, θα πρέπει η σχέση (33) να ελαχιστοποιηθεί. Για την επίλυση αυτής της εξίσωσης, θα χρησιμοποιήσουμε εργαλεία βελτιστοποίησης για δευτεροβάθμια συνάρτηση με γραμμικούς περιορισμούς. Εφόσον η συνάρτηση κόστους είναι κυρτή, γνωρίζουμε ότι αν βρούμε κάποιο τοπικό (και μοναδικό) ελάχιστο, αυτό θα είναι και ολικό ελάχιστο. Άρα, το βέλτιστο υπερεπίπεδο που θα προκύψει βάσει αυτών των συνθηκών θα είναι μοναδικό. Για να λύσουμε την σχέση (33) με χρήση και της σχέσης (34) αναζητούμε το:

$$\max_{\lambda}\left(\sum_{i=1}^{N}\lambda_i - \frac{1}{2}\sum_{i,j}\lambda_i\,\lambda_j y_i y_j x_i^T x_j\right)$$

(35)

$$\text{Δοθέντος των } \sum_{i=1}^{N}\lambda_i y_i = 0$$

(36)

$$\text{και } \boldsymbol{\lambda} \geq 0$$

(37)

όπου $\boldsymbol{\lambda}$ είναι το διάνυσμα των πολλαπλασιαστών Lagrange. Αφού έχουμε τους πολλαπλασιαστές Lagrange, μπορούν να λυθούν οι σχέσεις (38), (39) αντίστοιχα, οπότε προκύπτουν και τα $\mathbf{w}$, w_0. Οι δύο αυτές σχέσεις είναι γνωστές ως συνθήκες Karush – Kuhn – Tucker (KKT).

$$\boldsymbol{w} = \sum_{i=1}^{N} \lambda_i y_i \boldsymbol{x_i}$$

(38)

$$\lambda_i [y_i (\boldsymbol{w}^T \boldsymbol{x_i} + w_0) - 1] = 0, i = 1, \dots, N$$

(39)

Οι πολλαπλασιαστές Lagrange παίρνουν μη αρνητικές τιμές. Για τις θετικές τιμές, το $\boldsymbol{w}$ προκύπτει ως γραμμικός συνδυασμός $N_S \leq N$ διανυσμάτων. Τα συγκεκριμένα διανύσματα είναι τα Διανύσματα Υποστήριξης που αναφέρθηκαν νωρίτερα και ικανοποιούν την σχέση:

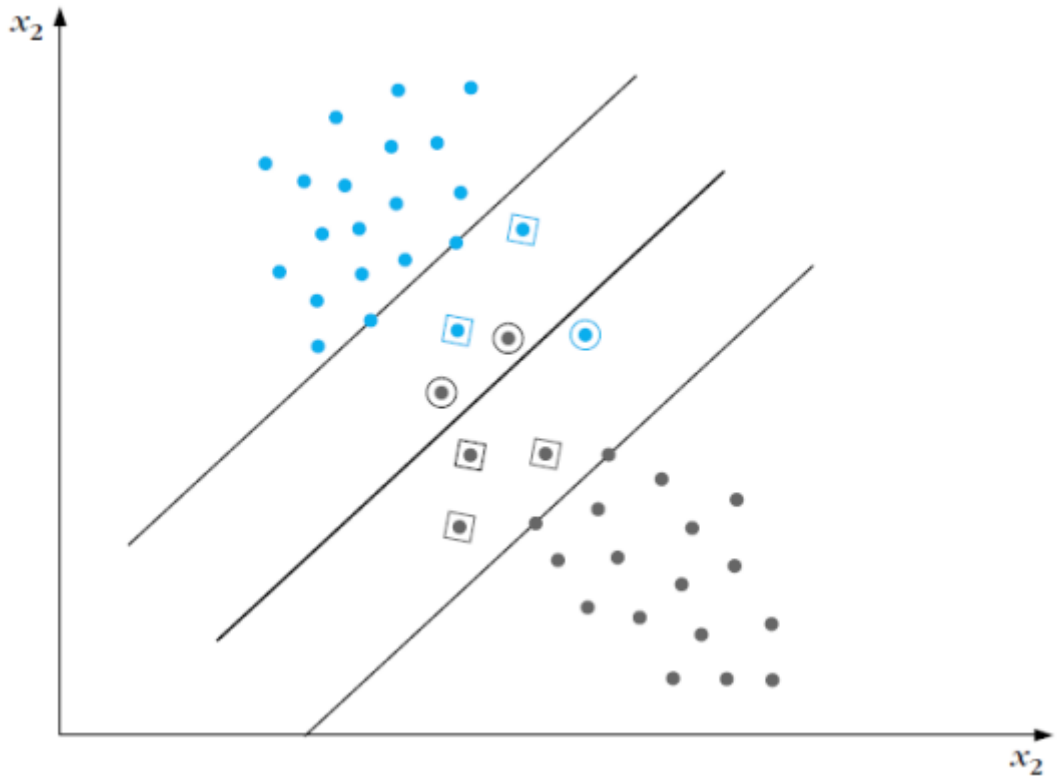
$$\boldsymbol{w}^T \boldsymbol{x} + w_0 = \pm 1$$

(40)

Τα επίπεδα που προκύπτουν από τη σχέση (40) οριοθετούν μία περιοχή, στην οποία δε συναντώνται σημεία από καμία από τις κλάσεις και ονομάζεται Περιοχή Διαχωρισμού Κλάσεων (Class Separation Band – CBS).

2.6.3.1.2 Μη Διαχωρίσιμες Κλάσεις

Ως μη διαχωρίσιμες κλάσεις χαρακτηρίζονται αυτές για τις οποίες δεν μπορεί να βρεθεί υπερεπίπεδο που να τις διαχωρίζει πλήρως, όπως στο παράδειγμα της Εικόνας 10.



Εικόνα 10: Παράδειγμα Μη Διαχωρίσιμων Κλάσεων [43]

Σε αυτή την περίπτωση, διανύσματα των δύο κλάσεων θα βρίσκονται και εντός της Περιοχής Διαχωρισμού Κλάσεων. Προκύπτουν οι εξής υποπεριπτώσεις για τα διανύσματα:

1. Εκτός CSB, σωστή ταξινόμηση: Είναι τα διανύσματα που επαληθεύουν την σχέση (34).
2. Εντός CSB, σωστή ταξινόμηση (τετράγωνα στην Εικόνα 10): Επαληθεύουν την σχέση:

$$0 \leq y_i(\boldsymbol{w}^T\boldsymbol{x} + w_0) < 1 \tag{41}$$

3. Εντός CSB, λανθασμένη ταξινόμηση (κύκλοι στην Εικ.10): Ικανοποιούν την σχέση:

$$y_i(\boldsymbol{w}^T\boldsymbol{x} + w_0) < 0 \tag{42}$$

Ενοποιημένη έκφραση για αυτές τις περιπτώσεις μπορεί να προκύψει, αν εισάγουμε παράμετρο ξ_i ώστε:

$$y_i(\boldsymbol{w}^T\boldsymbol{x} + w_0) \geq 1 - \xi_i$$

(43)

Για $\xi_i = 0$, ισχύει η πρώτη κατηγορία των διανυσμάτων που προαναφέρθηκε. Για $0 < \xi_i \leq 1$, ισχύει η δεύτερη κατηγορία και για $\xi_i > 1$, η τρίτη. Οι παράμετροι ξ_i ονομάζονται «χαλαρές μεταβλητές» (Slack Variables). Τώρα εκτός από την μεγιστοποίηση του περιθωρίου, πρέπει να ελαχιστοποιηθεί και ο αριθμός των σημείων με $\xi > 0$. Η συνάρτηση κόστους ορίζεται ως:

$$J(\boldsymbol{w}, w_0, \boldsymbol{\xi}) = \frac{1}{2}\|\boldsymbol{w}\|^2 + C\sum_{i=1}^{N}\xi_i$$

(44)

όπου $\boldsymbol{\xi}$ το διάνυσμα των παραμέτρων ξ_i $και$ C μια μη αρνητική παράμετρος. Οι σχέσεις (33), (34) ισοδύναμα εκφράζονται ως:

Ελαχιστοποίηση του $J(\boldsymbol{w}, w_0, \boldsymbol{\xi})$

(45)

Δοθέντος των $y_i(\boldsymbol{w}^T\boldsymbol{x_i} + w_0) \geq 1 - \xi_i$

(46)

και $\xi_i \geq 0$

(47)

Η w_0 υπολογίζεται επιλύοντας μία εκ των εξισώσεων:

$$\lambda_i[y_i(\boldsymbol{w}^T\boldsymbol{x_i} + w_0) - 1 + \xi_i] = 0, i = 1, \dots, N$$

(48)

Το τελικό πρόβλημα προς επίλυση είναι:

$$\max_{\lambda} \left(\sum_{i=1}^{N} \lambda_i - \frac{1}{2} \sum_{i,j} \lambda_i \, \lambda_j y_i y_j x_i^T x_j \right)$$

(49)

$$\text{Δοθέντος των } \sum_{i=1}^{N} \lambda_i y_i = 0$$

(50)

$$\text{και } 0 \leq \lambda_i \leq C, i = 1, \dots, N$$

(51)

Εδώ οι πολλαπλασιαστές Lagrange είναι άνω φραγμένοι της μεταβλητής C, ενώ στα γραμμικά διαχωρίσιμα μοντέλα θεωρούμε ότι $C \to \infty$.

2.6.3.2 Μη Γραμμική SVM

Αν τα δεδομένα εισόδου σε ένα πρόβλημα δεν μπορούν να μοντελοποιηθούν και να διαχωριστούν γραμμικά, δε μπορούν να χρησιμοποιηθούν γραμμικές SVM. Έτσι, εμφανίζονται οι μη γραμμικοί SVM ταξινομητές. Επομένως, το πρόβλημα ανάγεται σε έναν πολυδιάστατο χώρο, με διάσταση τον αριθμό χαρακτηριστικών, όπου τα διανύσματα εισόδου γίνονται γραμμικά διαχωρίσιμα.

Στη γραμμική SVM που εξετάσαμε μέχρι τώρα, η κατάταξη των σημείων σε κλάσεις γίνεται με τοποθέτησή τους γύρω από το υπερεπίπεδο σύμφωνα με το πρόσημο της διαχωριστικής συνάρτησης $g(\boldsymbol{x})$:

$$g(\boldsymbol{x}) = \boldsymbol{w}^T \boldsymbol{x} + w_0 = \sum_{i=1}^{N_S} \lambda_i y_i \boldsymbol{x_i}^T \boldsymbol{x} + w_0$$

(52)

Φαίνεται και από την σχέση (52) ότι η συνάρτηση διαχωρισμού εξαρτάται από το εσωτερικό γινόμενο των διανυσμάτων εισόδου που χρησιμοποιείται για την

ταξινόμηση. Αν απεικονίσουμε τα διανύσματα εισόδου $\boldsymbol{x}$ διάστασης l σε διανύσματα $\boldsymbol{y}$ σε ένα χώρο υψηλότερης διάστασης k:

$$\boldsymbol{x} \in \mathbb{R}^l \rightarrow \boldsymbol{y} \in \mathbb{R}^k$$

(53)

μπορούμε να βρούμε τρόπο ώστε τα διανύσματα να μπορούν να διαχωριστούν γραμμικά στον νέο χώρο. Για να γίνει η αντιστοίχιση, χρησιμοποιείται μία συνάρτηση $\boldsymbol{\varphi}$: $\boldsymbol{y} = \boldsymbol{\varphi}(\boldsymbol{x})$ (Θεώρημα Mercer), τέτοια ώστε και για το εσωτερικό γινόμενο των διανυσμάτων στον νέο χώρο να ισχύει:

$$\langle \boldsymbol{\varphi}(\boldsymbol{x}), \boldsymbol{\varphi}(\boldsymbol{z}) \rangle = K(x, z)$$

(54)

όπου $K(x, z)$ είναι μια θετικά ορισμένη συνεχής συνάρτηση – πυρήνας (Kernel). Προκύπτει ότι το εσωτερικό γινόμενο των διανυσμάτων, από το οποίο εξαρτάται η ταξινόμησή τους στον αρχικό χώρο, απεικονίζεται στο εσωτερικό γινόμενο των διανυσμάτων που παράχθηκαν στον νέο χώρο, και επομένως έχουμε πρακτικά μεταφέρει τη συνθήκη για την ταξινόμηση στον πολυδιάστατο χώρο.

Έτσι, έχουμε πλέον να επιλύσουμε ένα πρόβλημα μεγιστοποίησης, αντίστοιχο με αυτό της σχέσης (35):

$$\max_{\lambda} \left(\sum_{i=1}^{N} \lambda_i - \frac{1}{2} \sum_{i,j} \lambda_i \, \lambda_j y_i y_j K(x_i, x_j) \right)$$

(55)

$$\text{Δοθέντων των } \sum_{i=1}^{N} \lambda_i y_i = 0$$

(56)

$$\text{και } 0 \leq \lambda_i \leq C, i = 1, \dots, N$$

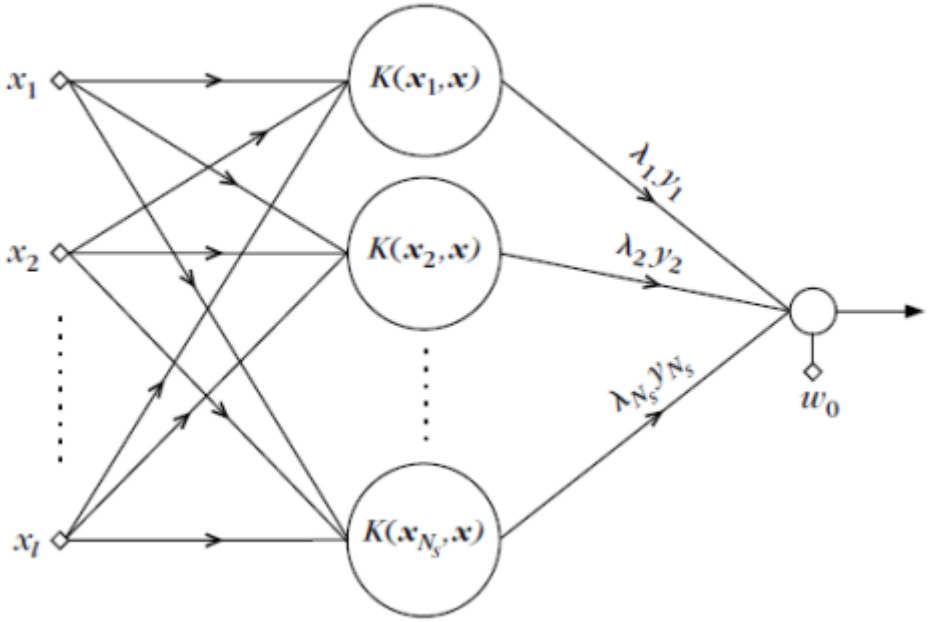
(57)

Η νέα συνάρτηση διαχωρισμού προκύπτει από την σχέση (58). Για $g(\boldsymbol{x}) > 0$, το αρχικό διάνυσμα εισόδου κατατάσσεται στην περιοχή της κλάσης ω_1, αλλιώς στην κλάση ω_2:

$$g(\boldsymbol{x}) = \sum_{i=1}^{N_S} \lambda_i y_i \, K(x_i, \boldsymbol{x}) + w_0$$

(58)

Στην Εικόνα 11, αποτυπώνεται η αρχιτεκτονική ενός μη γραμμικού ταξινομητή. Τα διανύσματα υποστήριξης καθορίζουν τον αριθμό κόμβων K, οι οποίοι με χρήση της συνάρτησης πυρήνα θα παράγουν τα νέα εσωτερικά γινόμενα.



Εικόνα 11: Αρχιτεκτονική Μη Γραμμικού Ταξινομητή [43]

Οι πιο δημοφιλείς συναρτήσεις πυρήνα είναι οι εξής:

- **Πολυωνυμικές (Polynomials)**

$$K(x, z) = (x^T z + 1)^q, q > 0$$

(59)

- **Gaussian Ακτινικές Συναρτήσεις Βάσης (RBF – Radial Basis Functions)**

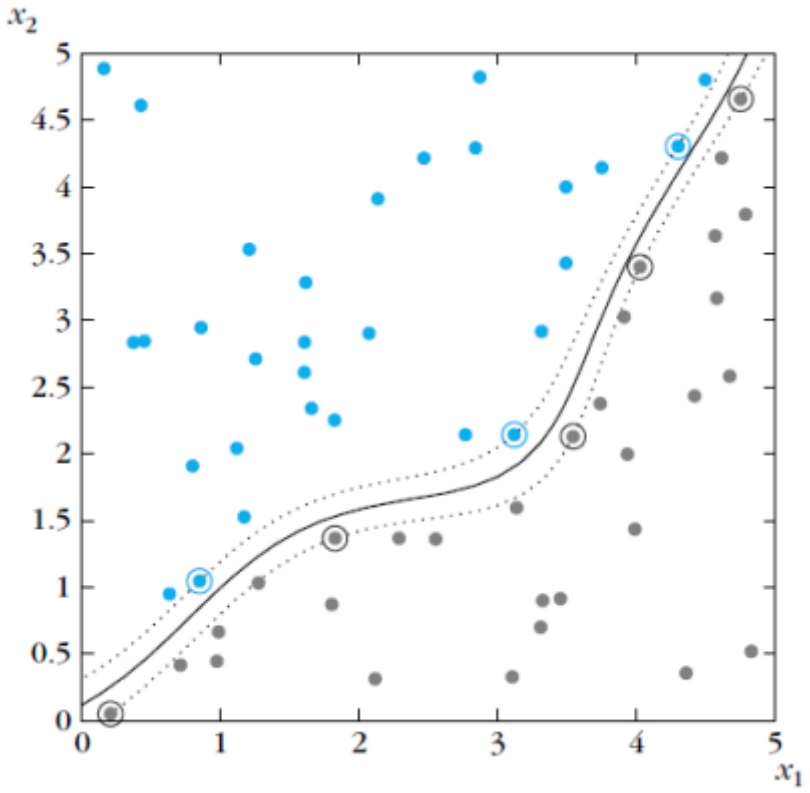
$$K(x, z) = exp\left(-\frac{\|x - z\|^2}{\sigma^2}\right)$$

(60)

- **Υπερβολική Εφαπτομένη (Hyperbolic Tangent)**

$$K(x, z) = tanh(\beta x^T z + \gamma) \quad (61)$$

Στην Εικόνα 12, απεικονίζεται μια μη γραμμική SVM για δύο μη γραμμικά διαχωρίσιμες κλάσεις, με συνάρτηση πυρήνα την Gaussian RBF με σ = 1.75.



Εικόνα 12: Μη Γραμμικός SVM Ταξινομητής [43]

2.6.4 Διαδοχική Ελάχιστη Βελτιστοποίηση (SMO)

Ο επαναληπτικός αλγόριθμος (Sequential Minimal Optimization – SMO) εφευρέθηκε από τον John Platt το 1998 και αναπτύχθηκε ως βελτίωση των SVM [44], [45], [46]. Για τον υπολογισμό των υπερεπιπέδων στα SVM, θα πρέπει να λυθεί το πρόβλημα ελαχιστοποίησης του κόστους, που αποτελεί ένα υπολογιστικά μεγάλο πρόβλημα τετραγωνικού προγραμματισμού (Quadratic Programming – QP). Ο SMO διασπά το

αρχικό πρόβλημα QP σε επιμέρους μικρότερα QP που επιλύονται πιο εύκολα. Λόγω του περιορισμού γραμμικής ισότητας που περιλαμβάνει τους πολλαπλασιαστές Lagrange λ_i, το μικρότερο δυνατό πρόβλημα περιλαμβάνει δύο τέτοιους πολλαπλασιαστές, έστω λ_1 και λ_2. Οπότε οι περιορισμοί γίνονται ως εξής:

Στόχος είναι η εύρεση ενός ελάχιστου μιας μονοδιάστατης τετραγωνικής συνάρτησης. Το k είναι το αρνητικό του αθροίσματος έναντι των υπολοίπων όρων στον περιορισμό ισότητας, που καθορίζεται σε κάθε επανάληψη.

Ο αλγόριθμος εκτελείται ως εξής:

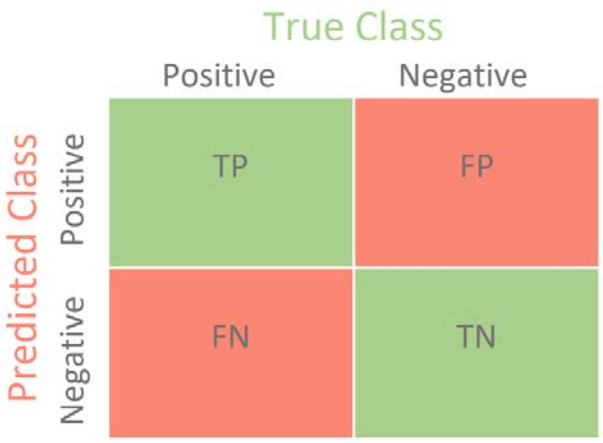
1. Βρείτε έναν πολλαπλασιαστή Lagrange λ_1 που παραβιάζει τις συνθήκες Karush – Kuhn – Tucker (KKT) για το πρόβλημα βελτιστοποίησης.
2. Επιλέξτε έναν δεύτερο πολλαπλασιαστή λ_2 και βελτιστοποιήστε το ζεύγος (λ_1, λ_2).
3. Επαναλάβετε τα βήματα 1 και 2 έως τη σύγκλιση.

Όταν όλοι οι πολλαπλασιαστές Lagrange ικανοποιούν τις συνθήκες KKT, το πρόβλημα έχει επιλυθεί. Παρόλο που αυτός ο αλγόριθμος εγγυάται ότι συγκλίνει, οι ευρετικές μέθοδοι χρησιμοποιούνται για την επιλογή του ζεύγους των πολλαπλασιαστών, έτσι ώστε να επιταχυνθεί ο ρυθμός σύγκλισης. Αυτό είναι κρίσιμο για μεγάλα σύνολα δεδομένων, αφού υπάρχουν $\frac{n(n-1)}{2}$ πιθανές επιλογές για λ_i και λ_j.

2.7 Αξιολόγηση Μοντέλων Ταξινόμησης

2.7.1 Μετρικές Αξιολόγησης

Διάφορες μετρικές χρησιμοποιούνται για την αξιολόγηση των μοντέλων και την επίδοση ενός ταξινομητή [32], [47]. Παρακάτω θα περιγράψουμε μερικές μετρικές που είναι χρήσιμες για την αξιολόγηση των μοντέλων ταξινόμησης που αναπτύσσουμε στην παρούσα εργασία.



Εικόνα 13: Πίνακας Σύγχυσης[7]

2.7.1.1 Accuracy

Σε ένα πρόβλημα δυαδικής ταξινόμησης, η ακρίβεια (Accuracy) εκφράζεται ως το ποσοστό των στιγμιότυπων του συνόλου ελέγχου (Test Set) που έλαβαν σωστή ετικέτα και ταξινομήθηκαν στην σωστή κλάση που ανήκουν. Υπολογίζεται ως:

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN}$$

(62)

Όπου:

- **TP (True Positive):** Ανήκουν όντως στην κλάση “Positive” και ταξινομήθηκαν ως “Positive”.
- **TN (True Negative):** Ανήκουν όντως στην κλάση “Negative” και ταξινομήθηκαν ως “Negative”.
- **FP (False Positive):** Ανήκουν όντως στην κλάση “Negative” και ταξινομήθηκαν ως “Positive”.
- **FN (False Negative):** Ανήκουν όντως στην κλάση “Positive” και ταξινομήθηκαν ως “Negative”.

[7]https://www.researchgate.net/publication/350487701_Explaining_what_learned_models_predict_In_which_cases_can_we_trust_machine_learning_models_and_when_is_caution_required/figures?lo=1

2.7.1.2 Precision and Recall

Η ανάκληση (Recall) είναι η ικανότητα του ταξινομητή να βρίσκει όλα τα σωστά κατηγοριοποιημένα ως θετικά στιγμιότυπα. Ορίζεται ως:

$$Recall = \frac{TP}{TP + FN}$$

(63)

Η ακρίβεια (Precision) είναι η ικανότητα του αλγορίθμου ταξινόμησης να μην κατατάσσει ως θετικό ένα στιγμιότυπο που είναι αρνητικό. Ορίζεται ως:

$$Precision = \frac{TP}{TP + FP}$$

(64)

2.7.1.3 F – Measure - F1 – Score

Η μετρική F – Measure συνδυάζει τις δύο προηγούμενες μετρικές και παρέχει μια συνολική εκτίμηση του μοντέλου. Στην περίπτωση του F1 – Score, η συμβολή της ανάκλησης και της ακρίβειας είναι ίσες. Υπολογίζεται ως ο αρμονικός μέσος όρος της ανάκλησης και της ακρίβειας:

$$F - Measure = \frac{2 \cdot Recall \cdot Precision}{Recall + Precision}$$

(65)

2.7.2 Μέθοδοι Αξιολόγησης

2.7.2.1 Hold Out Μέθοδος

Στη μέθοδο αυτή χωρίζουμε το σύνολο των δεδομένων σε δύο διαφορετικά μεταξύ τους σύνολα, για εκπαίδευση (Train Set) και έλεγχο (Test Set) αντίστοιχα, στη συνέχεια υπολογίζουμε την ακρίβεια του μοντέλου στο σύνολο ελέγχου. Γενικά, το σύνολο εκπαίδευσης σχηματίζεται με τυχαία δειγματοληψία από το σύνολο δεδομένων και οτιδήποτε απομείνει από αυτό, αποτελεί το σύνολο ελέγχου. Η αναλογία του συνόλου ελέγχου σε σχέση με το σύνολο δεδομένων συνήθως είναι το μισό έως 2/3 του αρχικού συνόλου.

2.7.2.2 K – Fold Cross – Validation

Η k – πλάσια διασταυρωμένη επικύρωση (k – Fold Cross – Validation) ξεκινά με τον διαχωρισμό των δεδομένων σε k ομάδες (ίσου ή περίπου ίσου μήκους). Συνήθως πριν από αυτό πραγματοποιείται ανακατανομή των δεδομένων, ώστε κάθε ομάδα να περιέχει αντιπροσωπευτικό μέρος του συνόλου. Έπειτα, για κάθε επανάληψη εκπαίδευσης και ελέγχου, οι k – 1 χρησιμοποιούνται για εκπαίδευση και ο έλεγχος πραγματοποιείται στην k – οστή ομάδα, ώστε όλες οι ομάδες να χρησιμοποιηθούν από μία φορά ως σύνολο ελέγχου. Με αυτό τον τρόπο περιορίζεται η υπερπροσαρμογή.

3 Ανάλυση Συγγραφέα (Authorship Analysis)

3.1 Εισαγωγή

Τις τελευταίες δεκαετίες που διανύουμε, ένα νέο ερευνητικό πεδίο έχει κάνει την εμφάνισή του στον τομέα της μηχανικής μάθησης, η λεγόμενη "Ανάλυση Συγγραφέα" (Authorship Analysis). Κι αυτό εξαιτίας της συνεχούς αυξητικής τάσης εξόρυξης πληροφοριών για οποιονδήποτε συγγραφέα, τόσο σε επίπεδο γραπτού λόγου όσο και σε επίπεδο προφορικού, κυρίως όσον αφορά το πεδίο του διαδικτύου. Στην ανάλυση συγγραφέα μελετώνται στατιστικά τα γλωσσικά και υπολογιστικά χαρακτηριστικά των γραπτών εγγράφων του κάθε ατόμου. Το πρόβλημα του Authorship Analysis μπορεί να διακριθεί σε 3 τρεις υποκατηγορίες:

- **Author Attribution ή Identification:** Προσδιορίζει την πιθανότητα ένα κείμενο να ανήκει σε συγκεκριμένο συγγραφέα, έχοντας ως δεδομένα άλλα διαθέσιμα έγγραφά του.
- **Author Profiling ή Characterization:** Εξάγει χαρακτηριστικά του δημιουργού ενός εγγράφου και συνθέτει για αυτόν ένα προφίλ. Αυτά τα χαρακτηριστικά περιλαμβάνουν το φύλο του, τα δημογραφικά στοιχεία του, το εκπαιδευτικό του υπόβαθρο, την προσωπικότητά του, κλπ.
- **Similarity Detection:** Συγκρίνει πολλαπλά κομμάτια εγγράφων και καθορίζει αν προέρχονται από έναν ή πολλούς συγγραφείς, χωρίς απαραίτητα να τους προσδιορίζει. Αυτή η υποκατηγορία χρησιμοποιείται κυρίως σε προβλήματα ανίχνευσης λογοκλοπής.

3.2 Αναγνώριση Προφίλ Συγγραφέα (Author Profiling)

Η Αναγνώριση Προφίλ Συγγραφέα (Author Profiling) αποτελεί έναν υποτομέα της τεχνητής νοημοσύνης, ο οποίος ασχολείται με την ανίχνευση του προφίλ ενός συγγραφέα. Σχετικά με αυτό, γίνεται η εκπαίδευση ενός μοντέλου πάνω σε έγγραφα, αποθηκεύοντας πληροφορίες για τον συγγραφέα του κάθε εγγράφου. Δεδομένου οποιουδήποτε εγγράφου σαν είσοδο, μαθαίνει να αναζητά και να ανακτά πληροφορίες για το προφίλ του συγγραφέα. Αυτό αποτελεί ένα πρόβλημα ταξινόμησης και υπάρχουν αρκετές προσεγγίσεις του. Δημογραφικά στοιχεία, προσωπικότητα, μορφωτικό επίπεδο και γλωσσική ευχέρεια είναι πληροφορίες προς ανάκτηση για αυτά τα προβλήματα.

3.3 Εφαρμογές του Author Profiling

Το Author Profiling εφαρμόζεται στα παρακάτω:

- **Περιπτώσεις Chat & Social Media:** Όπως γνωρίζουμε, υπάρχει σημαντική αύξηση της εγκληματικότητας εντός του κυβερνοχώρου, η οποία επηρεάζεται από την ραγδαία εξέλιξη του διαδικτύου. Συνεπώς, παρατηρείται και αύξηση σε περιπτώσεις επιθέσεων και παραπλανήσεων ατόμων, είτε ανήλικων είτε ενήλικων. Έτσι λοιπόν, ένα αυτοματοποιημένο σύστημα εξόρυξης πληροφοριών για οποιονδήποτε συγγραφέα μπορεί να βοηθήσει σημαντικά στην μείωση θυμάτων.
- **Αρχαιολογικές και επιστημονικές εφαρμογές:** Είναι χρήσιμο για έγγραφα στα οποία δεν είναι γνωστός ο συγγραφέας. Υπάρχει πληθώρα αρχαίων εγγράφων, τα οποία δε διαθέτουν πληροφορίες για τον συγγραφέα, με αποτέλεσμα οι ερευνητές να καθυστερούν, βασιζόμενοι μόνο στις γνώσεις και εμπειρίες που κατέχουν. Πλέον, με ένα αυτοματοποιημένο λογισμικό, η διαδικασία αυτή μπορεί να μειωθεί κατά πολύ.
- **Μάρκετινγκ και πελατειακές σχέσεις:** Στη σημερινή εποχή, ένα μεγάλο ποσοστό των οργανισμών και των επιχειρήσεων ανακτούν πληροφορίες απ’

τους πελάτες τους, ώστε να δημιουργήσουν ολοκληρωμένα προφίλ αυτών στις βάσεις δεδομένων τους και να προσαρμόζουν την συμπεριφορά τους ανάλογα.

3.4 Περιγραφή Δεδομένων

Στον τομέα του Author Profiling, μεγάλο πλήθος ερευνών επικεντρώνεται σε συλλογές εγγράφων, όπως Emails, Essays, Blogs και λιγότερο σε επίσημα έγγραφα εξαιτίας της ραγδαίας εξέλιξης των Social Media. Κι αυτό επειδή τέτοιες συλλογές διαθέτουν διαφορετικά γλωσσικά και μορφολογικά χαρακτηριστικά και περιβάλλονται από γλώσσα λιγότερο δομημένη και πιο άτυπη. Στην παρούσα εργασία, έγινε χρήση της συλλογής CMCC Corpus [3] ως συλλογή εκπαίδευσης και ελέγχου των μοντέλων. Παρακάτω γίνεται αναφορά των κύριων σημείων αυτής της συλλογής.

3.5 Ο Διαγωνισμός PAN για το Author Profiling

Το 2013 για την αναγνώριση προφίλ συγγραφέα εισήχθη ο διαγωνισμός του PAN [48], παρέχοντας μια συλλογή εκπαίδευσης βασισμένη σε Blogs. Ακολούθησε το PAN 2014, όπου δημιουργήθηκε ένα σώμα από τέσσερις διαφορετικές συλλογές: Twitter, Blogs, Social Media and Hotel Reviews. Το σώμα εκπαίδευσης διατάχθηκε σε αρχεία XML, ένα για κάθε συγγραφέα, του οποίου σημειώθηκαν πληροφορίες για την ηλικία και το φύλο. Στο PAN 2015 εξετάστηκαν μέθοδοι δημιουργίας προφίλ που μπορούσαν να εφαρμοστούν σε πολλές γλώσσες. Γι’ αυτό τον λόγο δημιουργήθηκε ένα σώμα τεσσάρων διαφορετικών γλωσσών: Αγγλικά, Ολλανδικά, Ιταλικά και Ισπανικά. Εκτός της αναγνώρισης της ηλικίας και του φύλου, πραγματοποιήθηκε αναγνώριση πέντε χαρακτηριστικών γνωρισμάτων προσωπικότητας (Outgoing, Stable, Friendly, Conscientious and Opened). Ο κύριος στόχος του PAN 2016 δόθηκε στην εφαρμογή των αναπτυσσόμενων μοντέλων σε είδη εγγράφων διαφορετικών από την συλλογή εκπαίδευσης. Οι διαγωνιζόμενοι κλήθηκαν να αναγνωρίσουν το φύλο και την ηλικία με την εκπαίδευση να γίνεται σε ένα είδος (π.χ. Twitter, Blogs, Social Media) και την αξιολόγηση να γίνεται σε ένα διαφορετικό είδος από αυτό της εκπαίδευσης. Τέλος, στο PAN 2017 προσδιορίστηκαν το φύλο και η γλωσσική ποικιλία στο Twitter. Έτσι, το

σύνολο των δεδομένων εκπαίδευσης λαμβάνεται από το Twitter και περιέχει tweets ισάριθμα κατανεμημένα μεταξύ των δύο φύλων σε τέσσερις διαφορετικές γλώσσες: Αγγλικά, Ισπανικά, Πορτογαλικά και Αραβικά.

3.6 Πεδίο Έρευνας

Το ζήτημα της αναγνώρισης προφίλ συγγραφέα έχει μελετηθεί και σημειώσει σημαντική πρόοδο τα τελευταία χρόνια. Έτσι, έχουν προκύψει σημαντικά δεδομένα για τις τεχνικές αναπαράστασης κειμένου, προεπεξεργασίας, καθώς και τα μοντέλα ταξινομητών.

Στη διαδικασία αναπαράστασης του κειμένου, σύμφωνα με τα δεδομένα του διαγωνισμού PAN ανά τα χρόνια [49], [50], [51], [4], [2], οι περισσότερες προσεγγίσεις κινούνται γύρω από μοτίβα σε επίπεδο στυλ (συχνότητα όρων, POS, στίξη, πεζά/κεφαλαία γράμματα, ειδικοί χαρακτήρες του Twitter, Emoticons, Character Floods – N ίδιοι χαρακτήρες στη σειρά) αλλά και περιεχομένου (BOW, Word N – Gram, λέξεις από λεξικά, αργκό, συναισθηματικά φορτισμένες λέξεις, θεματικές λίστες λέξεων). Παρατηρώντας τις προσεγγίσεις των ομάδων με τις καλύτερες επιδόσεις, φαίνεται πως ο συνδυασμός στιλιστικών και χαρακτηριστικών περιεχομένου επιφέρει το καλύτερο αποτέλεσμα. Γενικά πιο διαδεδομένες φαίνονται οι μέθοδοι N – Gram (Word, Character, TF – IDF, POS) και η επιλογή N συχνότερων λέξεων. Οι μέθοδοι Word και POS N – Gram δίνουν ικανοποιητικά αποτελέσματα τις περισσότερες φορές, ωστόσο φαίνεται να αποδίδουν καλύτερα σε πυκνά κείμενα [52], ενώ τα Character N – Gram χρησιμοποιούνται για να εξάγουν όσο το δυνατό περισσότερη πληροφορία από λιγότερο πυκνά είδη, καθώς μπορούν να εκμεταλλευτούν και χαρακτήρες, όπως emoticons, στίξη, κλπ. που συναντώνται κυρίως στα μέσα κοινωνικής δικτύωσης [4].

Στο [34], μέσα από έρευνα σε υπάρχουσες υλοποιήσεις, αναφέρεται για τη φάση προεπεξεργασίας ότι οι μετρικές εξαγωγής χαρακτηριστικών που φαίνεται να δίνουν τα πιο χρήσιμα στοιχεία και τείνουν να επικρατούν είναι το κέρδος πληροφορίας (Information Gain) και το X^2 (Chi – Squared). Στις περισσότερες περιπτώσεις, επιλέγεται να αφαιρεθούν οι Stop Words. Όσον αφορά τους ταξινομητές, ενώ δεν υπάρχει μία ενιαία αποτελεσματική λύση για όλα τα είδη δεδομένων, ο γραμμικός

SVM είναι ο πιο διαδεδομένος, όπως επιβεβαιώνεται και από τα δεδομένα του διαγωνισμού PAN. Αρκετά δημοφιλή είναι και τα Δένδρα Απόφασης, καθώς και η υποκατηγορία Random Forests. Τα τελευταία χρόνια παρατηρείται αυξανόμενο ενδιαφέρον για τη χρήση νευρωνικών δικτύων για ταξινόμηση, ειδικά για Αναδρομικά Νευρωνικά Δίκτυα (Recurrent Neural Networks – RNN), καθώς είναι τα καταλληλότερα για αναγνώριση ακολουθιακών χαρακτηριστικών στα δεδομένα και χρησιμοποιούνται ήδη σε εφαρμογές αναγνώρισης ομιλίας και επεξεργασίας φυσικής γλώσσας. Τα δεδομένα από τον διαγωνισμό PAN δε δείχνουν κάποια αύξηση στην ακρίβεια με χρήση νευρωνικών δικτύων έναντι των παραδοσιακών ταξινομητών [53]. Επίσης, στο [54] αναφέρεται ότι η ακρίβεια του νευρωνικού δικτύου που αναπτύχθηκε ήταν χαμηλή και η διαδικασία εκπαίδευσης αποδείχθηκε πρόκληση, καθώς το μοντέλο οδηγούταν σε υπερεκπαίδευση ήδη από τις πρώτες εποχές.

4 Εργαλεία και Βιβλιοθήκες

4.1 Κώδικας PAN 2018

Ως βάση για την υλοποίηση, αξιοποιήθηκαν συναρτήσεις από το εξής αποθετήριο στο GitHub. Ο συγκεκριμένος κώδικας προορίζεται για το Dataset του διαγωνισμού PAN 2018, οπότε έγιναν οι απαραίτητες τροποποιήσεις, καθώς και προσθήκη νέων συναρτήσεων (Βλ. Παράρτημα Ι) με χρήση βιβλιοθηκών Python, όπως θα αναφερθεί στη συνέχεια.

4.2 Βιβλιοθήκες Python

4.2.1. Διαχείριση Δεδομένων

Αρχικά, για την οργάνωση των δεδομένων από το αρχείο **(.csv)** σε μια μορφή εύκολα διαχειρίσιμη από το μοντέλο κατηγοριοποίησης, χρησιμοποιήθηκε η βιβλιοθήκη **Pandas** της **Python** και πιο συγκεκριμένα η δομή **Dataframe**. Για την εξαγωγή συμπερασμάτων σε σχέση με την απόδοση του μοντέλου, αξιοποιήθηκε η βιβλιοθήκη **Matplotlib** και το **Pyplot**.

4.2.2. Επεξεργασία Κειμένου

Για την αναπαράσταση των κειμένων σε Tokens, έγινε χρήση της εργαλειοθήκης **Natural Language Toolkit (NLTK)** της **Python**, μία συλλογής βιβλιοθηκών για επεξεργασία φυσικής γλώσσας. Με συναρτήσεις, όπως η **word_tokenize** πραγματοποιήθηκε ο διαχωρισμός του κειμένου σε λέξεις για το μοντέλο Word N – Gram. Ακόμη, με τη συνάρτηση **pos_tag** επιτεύχθηκε ο διαχωρισμός των λέξεων σε κατηγορίες, με βάση με το μέρος του λόγου στο οποίο ανήκουν, τη σημασία τους, κλπ.

Όσον αφορά την προεπεξεργασία των δεδομένων πριν την κατηγοριοποίηση, για τη μέτρηση συχνότητας εμφάνισης κάθε Token, χρησιμοποιήθηκε η κλάση **TfidfVectorizer** της βιβλιοθήκης **Scikit – learn**. Δεδομένου ενός υπολογισμένου λεξιλογίου, επιστρέφει σε μορφή πίνακα για κάθε Token τον δείκτη του, το δείκτη του κειμένου εμφάνισής του και την TF – IDF συχνότητά του (Term Frequency – Συχνότητα Όρου, Inverse Document Frequency – Αντίστροφη Συχνότητα Εγγράφου). Η αλλαγή κλίμακας των δεδομένων (Scaling) πραγματοποιήθηκε με την κλάση **MaxAbsScaler** της **Scikit – learn**, η οποία μετατρέπει την κλίμακα του κάθε χαρακτηριστικού του δείγματος εκπαίδευσης με βάση τη μέγιστη απόλυτη τιμή του.

4.2.3. Ταξινόμηση Συγγραφέων

Για τη διαδικασία ταξινόμησης των συγγραφέων με βάση το φύλο, έγινε χρήση της βιβλιοθήκης **Scikit – learn** της **Python**, η οποία περιέχει μοντέλα για αλγορίθμους Machine Learning. Τα μοντέλα που χρησιμοποιήθηκαν είναι τα εξής: **LinearSVC** (Γραμμική SVM), **SVC** (Kernel = "poly") (Πολυωνυμική SVM), **MultinomialNB** (Πολυωνυμικός Naïve Bayes)**, DecisionTreeClassifier** (Δέντρο Απόφασης) και **RandomForestClassifier** (Τυχαίο Δάσος).

5 Περιγραφή Δεδομένων

5.1 CMCC Corpus

Το Dataset που χρησιμοποιήθηκε για την υλοποίηση ενός συστήματος Author Profiling είναι το CMCC Corpus. Η συλλογή περιέχει κείμενα από Email, Essay, Phone Interview, Blog, Chat και Discussion. Οι Discussion και Phone Interview αρχικά ηχογραφήθηκαν και στη συνέχεια απομαγνητοφωνήθηκαν. Τα στοιχεία συλλέχθηκαν από 21 φοιτητές και φοιτήτριες και ταξινομούνται σε έξι θέματα. Τα θέματα είναι τα εξής: "Church", "Gay Marriage", "Privacy", "Marijuana", "Iraq War" και "Sex Discrimination". Επιλέχθηκαν αμφιλεγόμενα θέματα, ώστε οι συμμετέχοντες να έχουν ώθηση να εκφράσουν την άποψή τους και ως συνέπεια, να συγκεντρωθούν αποσπάσματα κειμένων με μεγαλύτερη έκταση. Κατά μέσο όρο, από τον κάθε συμμετέχοντα συλλέχθηκαν συνολικά 22.000 λέξεις και για τα 6 θέματα. Συνοψίζοντας, η συλλογή περιέχει κείμενα από 6 διαφορετικά είδη του λόγου, τα οποία ανήκουν σε 6 διαφορετικά θέματα και προέρχονται από 21 φοιτητές/φοιτήτριες.

5.2 Προεπεξεργασία Δεδομένων

Αρχικά, τα δεδομένα βρίσκονταν σε μορφή αρχείων κειμένου **(.txt)**. Το κάθε αρχείο περιείχε την άποψη ενός συμμετέχοντα για ένα θέμα (π.χ. "Sex Discrimination"), η οποία έχει προέλθει από ένα είδος γραπτού/προφορικού λόγου (π.χ. Email). Για την ευκολότερη χρήση των δεδομένων στον κώδικα, έγινε η μετατροπή τους σε ένα αρχείο της μορφής **(.csv)**. Σαν δεύτερο βήμα, ήταν απαραίτητη η ανάλυση των κειμένων για την εξαγωγή των φύλων των συμμετεχόντων. Η εξαγωγή του φύλου έγινε με μελέτη της χρήσης των αντωνυμιών ή/και από την αυτούσια παραδοχή των συμμετεχόντων μέσα στα αποσπάσματα. Το εμπλουτισμένο αρχείο **(.csv)**, που περιέχει στήλες για την ταυτότητα του συγγραφέα, το απόσπασμα του κειμένου, το θέμα του κειμένου, το είδος από το οποίο αποσπάστηκε το κείμενο και το φύλο του συγγραφέα, δίνεται σαν είσοδος στο σύστημα Author Profiling που αναπτύχθηκε.

Από τις μεθόδους προεπεξεργασίας δεδομένων που αναφέρθηκαν στο Κεφάλαιο 1.5, εφαρμόστηκε το POS – Tagging στα μοντέλα POS N – Gram, καθώς και ένα είδος μείωσης διαστασιμότητας. Δηλαδή, σε ορισμένα πειράματα χρησιμοποιήθηκε μόνο ένα τμήμα του υπολογισμένου λεξιλογίου, ώστε να παραληφθούν οι σπάνιοι όροι.

6 Πειράματα

6.1 Διαχείριση Dataset

Το Dataset των 21 συγγραφέων αποτελείται από δείγματα κειμένου 11 ανδρών (M – Male) και 10 γυναικών (F – Female). Για την εκπαίδευση του συστήματος Author Profiling, το Dataset χωρίστηκε σε 11 συγγραφείς (6 M, 5 F) που αποτελούν το δείγμα εκπαίδευσης (Train Set) και στους υπόλοιπους 10 (5 M, 5 F) που αποτελούν το δείγμα ελέγχου (Test Set).

6.2 Cross – Validation

Για εξασφάλιση της καλύτερης εκπαίδευσης και δυνατότητας γενίκευσης του μοντέλου, εφαρμόστηκε τεχνική Διασταυρωμένης Επικύρωσης (Cross – Validation). Συγκεκριμένα, για κάθε είδος κειμένου χρησιμοποιούνται τα 5 από τα 6 θέματα του Train Set για εκπαίδευση και το εκπαιδευμένο μοντέλο ελέγχεται στο 6ο θέμα από το Test Set. Η διαδικασία αυτή, γνωστή και ως **Leave – One – Out Cross – Validation**, επαναλαμβάνεται 6 φορές για κάθε είδος κειμένου, ώστε κάθε θέμα να έχει χρησιμοποιηθεί ως Test Set.

6.3 Μοντέλα N – Gram

Οι αναπαραστάσεις N – Gram που δοκιμάστηκαν είναι οι εξής:

- Word N – Gram για N = 1, 2, 3
- POS N – Gram για N = 1, 2, 3
- Character N – Gram για N = 3, 4, 5

Τα 9 αυτά μοντέλα εφαρμόζονται σε καθένα από τα 6 είδη κειμένου, επομένως συνολικά θα προκύψουν 54 διαγράμματα μέσης ακρίβειας, 9 διαγράμματα Macro – Average ακρίβειας και 9 διαγράμματα Micro – Average ακρίβειας για αξιολόγηση του συστήματος. Επίσης, για κάθε τριάδα των Word, Character και POS N – Gram προκύπτει ένα διάγραμμα μέσης ακρίβειας, κάνοντας χρήση μόνο N συχνότερων όρων του λεξιλογίου.

6.4 Χρήση Ν συχνότερων λέξεων ως λεξιλόγιο

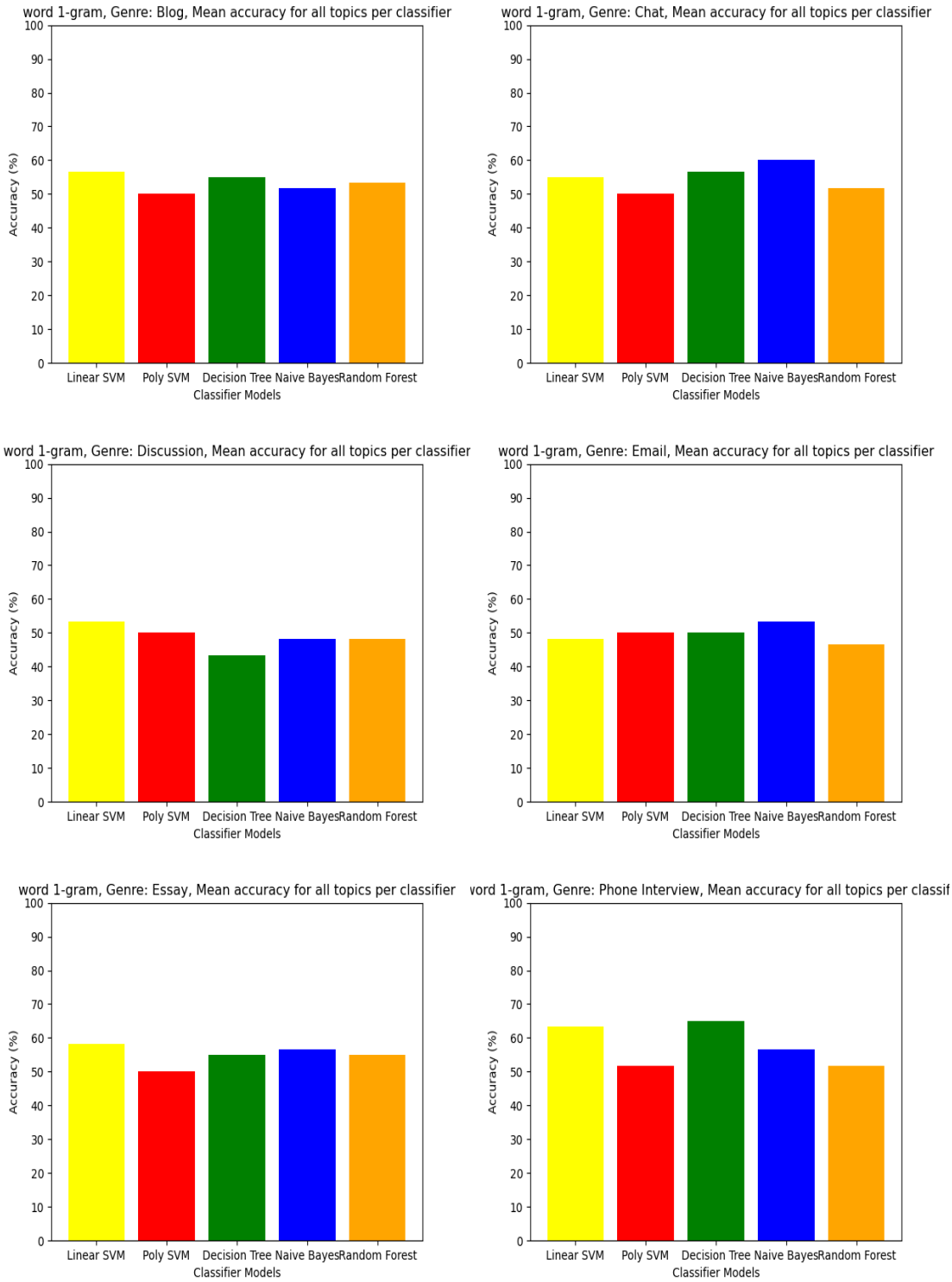
Όπως αναφέρθηκε ήδη, οι σπανιότεροι όροι σε ένα κείμενο δεν συνεισφέρουν σημαντικά στη διαδικασία ταξινόμησής του. Έτσι, σε κάποιες από τις δοκιμές συμπεριλήφθηκαν στο λεξιλόγιο του εκάστοτε είδους κειμένου μόνο οι Ν συχνότεροι όροι, με το Ν να κυμαίνεται από **500** μέχρι **3000**.

7 Αξιολόγηση

7.1 Διαχωρισμός σε Word N – Gram

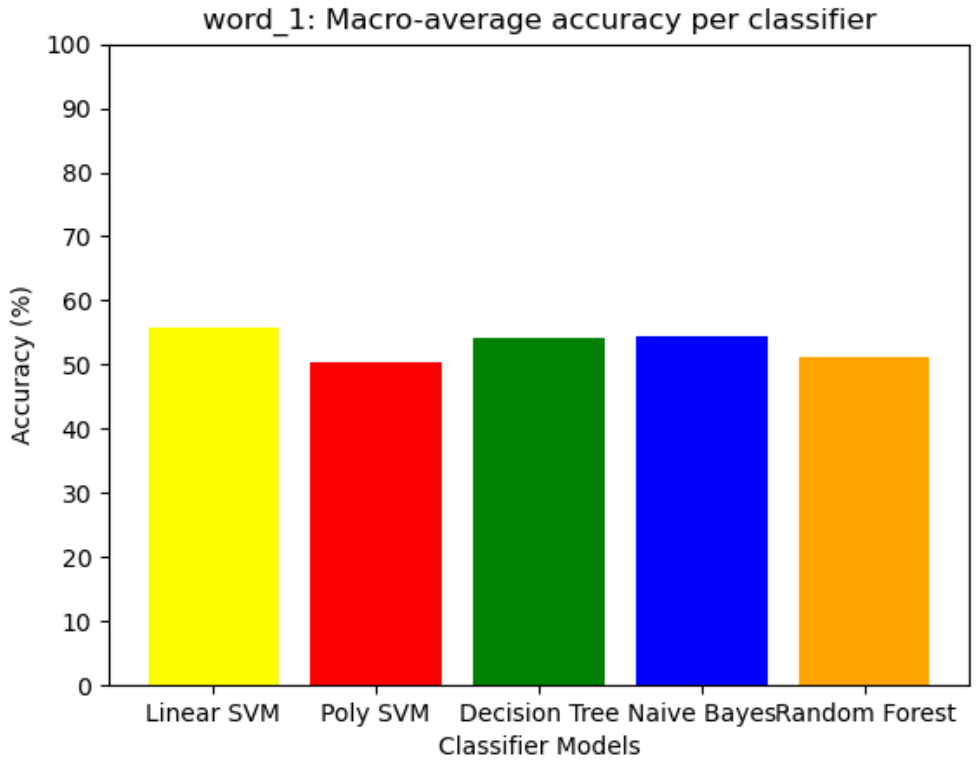
Παρακάτω φαίνονται διαγράμματα της μέσης ακρίβειας (Mean Accuracy) του συστήματος από τα 6 θέματα για κάθε είδος κειμένου που αναπαραστάθηκε ως Word N – Gram, με δοκιμές σε 5 διαφορετικά μοντέλα ταξινόμησης.

7.1.1 Word N – Gram, N = 1



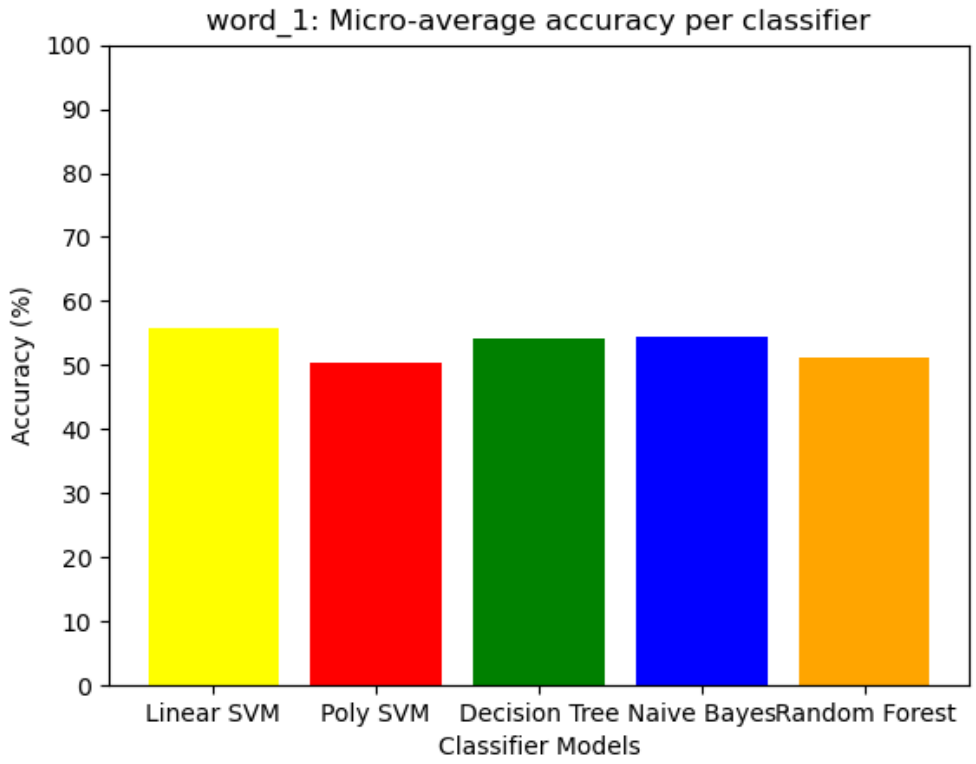
Εικόνα 14: Mean Accuracy ανά είδος για Word N – Gram, N = 1

Η μέγιστη μέση απόδοση **65%** παρατηρείται στο είδος **Phone Interview** για το μοντέλο **Decision Tree**. Στην Εικόνα 15, παρουσιάζονται οι Macro – Average τιμές ακρίβειας (Macro – Average Accuracy) ανά μοντέλο ταξινομητή. Οι τιμές υπολογίστηκαν ως εξής: για κάθε είδος κειμένου υπολογίστηκαν οι μέσες ακρίβειες για όλα τα θέματα για κάθε ταξινομητή και στη συνέχεια λήφθηκε ο μέσος όρος και από τα 6 είδη κειμένων ανά ταξινομητή.



Εικόνα 15: Macro – Average Accuracy για Word N – Gram, N = 1

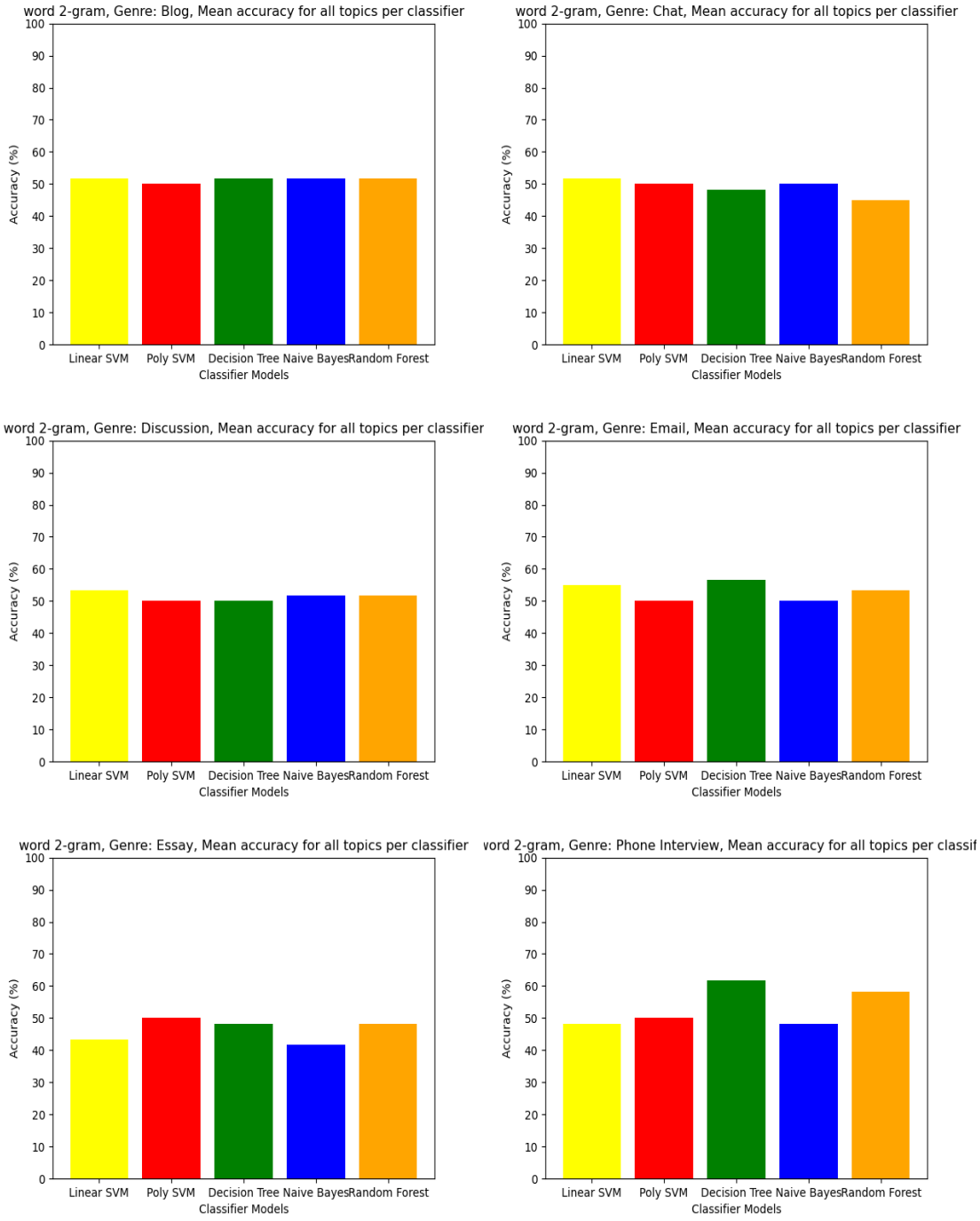
Η μέγιστη Macro – Average τιμή για το μοντέλο Word N – Gram, N = 1 καταγράφεται στο **56.1%** για το **Linear SVM**. Στην Εικόνα 16, παρουσιάζονται οι Micro – Average τιμές ακρίβειας (Micro – Average Accuracy) για το Word N – Gram, N = 1 για κάθε ταξινομητή, που υπολογίστηκαν λαμβάνοντας το μέσο όρο συνολικά από όλα τα θέματα και όλα τα είδη κειμένου.



Εικόνα 16: Micro – Average Accuracy για Word N – Gram, N = 1

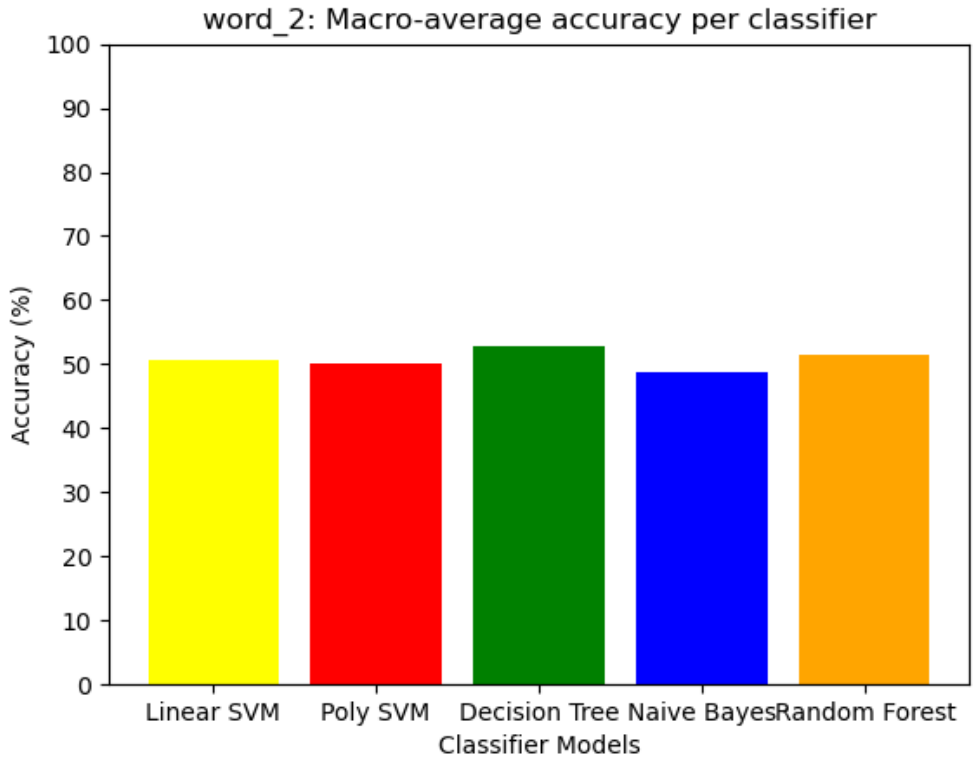
Η μέγιστη Micro – Average τιμή για το μοντέλο Word N – Gram, N = 1 καταγράφεται στο **56%** για το **Linear SVM**.

7.1.2 Word N – Gram, N = 2



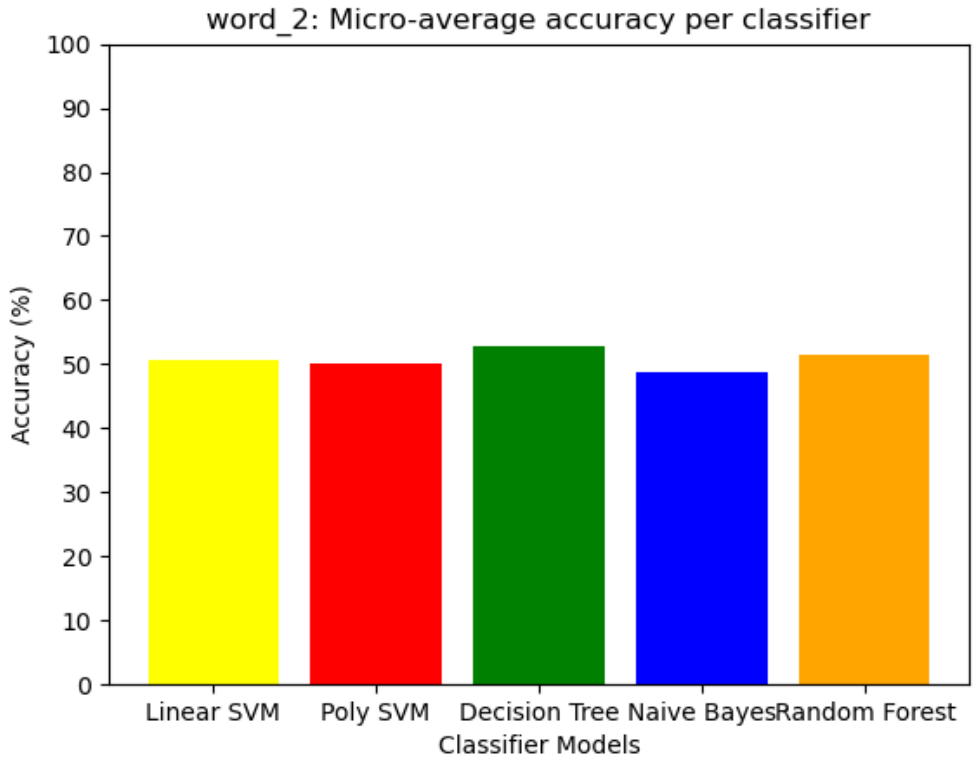
Εικόνα 17: Mean Accuracy για Word N – Gram, N = 2

Η μέγιστη μέση απόδοση **62%** παρατηρείται στο είδος **Phone Interview** για το μοντέλο **Decision Tree** αντίστοιχα. Στην Εικόνα 18, παρουσιάζονται οι Macro – Average τιμές ακρίβειας ανά μοντέλο ταξινομητή, με τον τρόπο που υπολογίστηκαν και πριν.



Εικόνα 18: Macro – Average Accuracy για Word N – Gram, N = 2

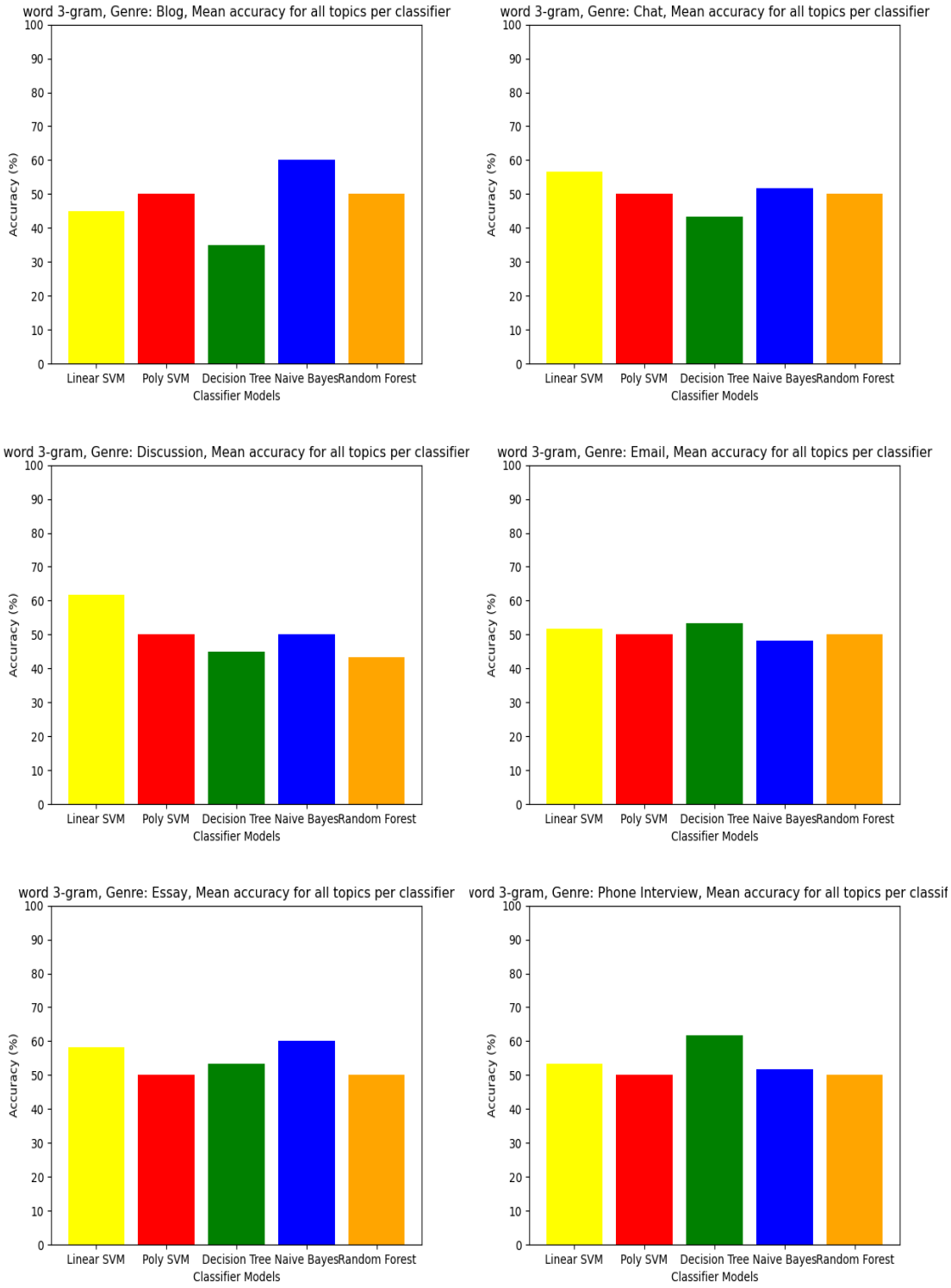
Η μέγιστη Macro – Average τιμή για το μοντέλο Word N – Gram, N = 2 καταγράφεται στο **53%** για το **Decision Tree**. Στην Εικόνα 19, παρουσιάζονται οι Micro – Average τιμές ακρίβειας για το Word N – Gram, N = 2 για κάθε ταξινομητή, που υπολογίστηκαν όπως αναφέρθηκε νωρίτερα.



Εικόνα 19: Micro – Average Accuracy για Word N – Gram, N = 2

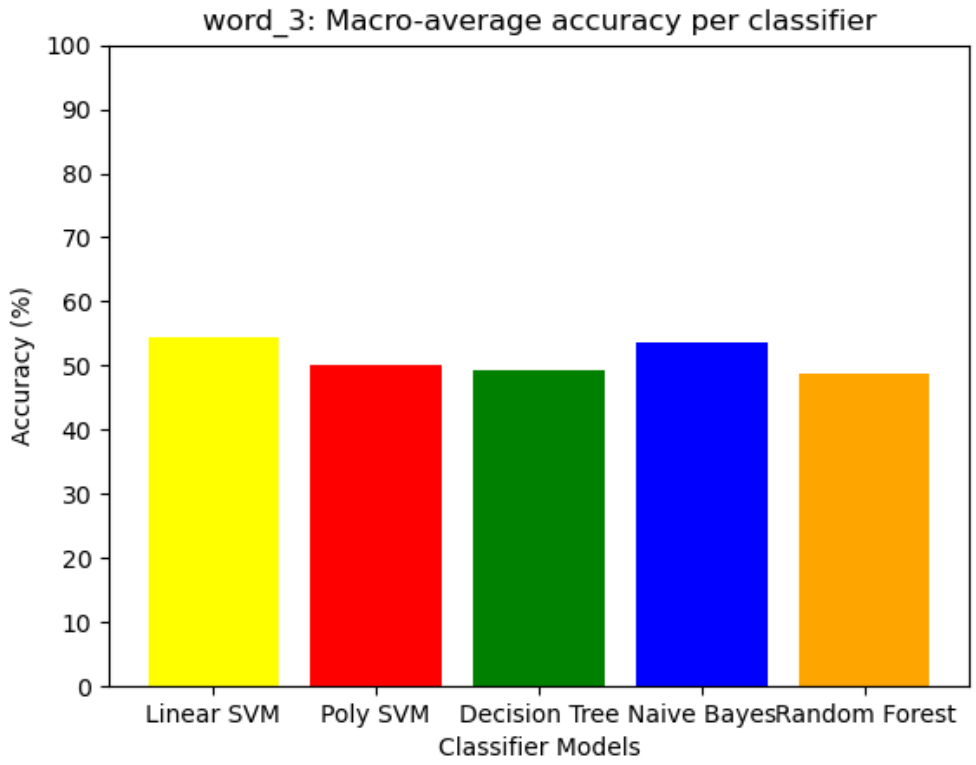
Η μέγιστη Micro – Average τιμή για το μοντέλο Word N – Gram, N = 2 καταγράφεται στο **53%** για το **Decision Tree**.

7.1.3 Word N – Gram, N = 3



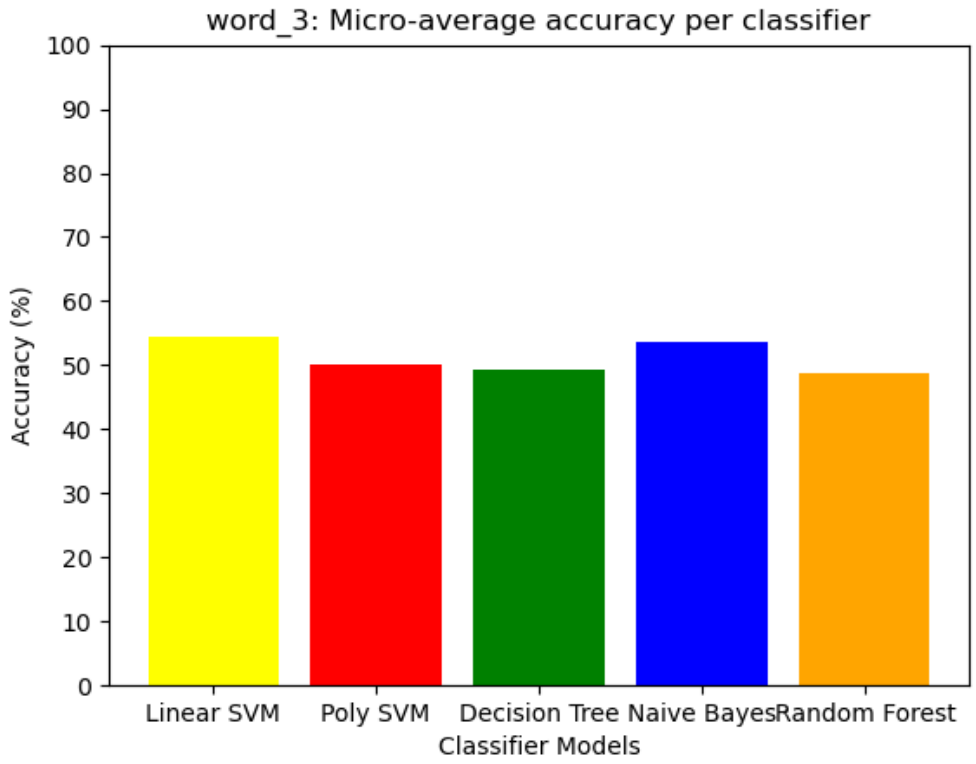
Εικόνα 20: Mean Accuracy για Word N – Gram, N = 3

Η μέγιστη μέση απόδοση **62%** παρατηρείται στα είδη **Discussion** και **Phone Interview** για τα μοντέλα ταξινομητών **Linear SVM** και **Decision Tree**. Στην Εικόνα 21, παρουσιάζονται οι Macro – Average τιμές ακρίβειας ανά μοντέλο ταξινομητή, με τον τρόπο που υπολογίστηκαν και πριν.



Εικόνα 21: Macro – Average Accuracy για Word N – Gram, N = 3

Η μέγιστη Macro – Average τιμή για το μοντέλο Word N – Gram, N = 3 καταγράφεται στο **55%** για το **Linear SVM**. Στην Εικόνα 22, παρουσιάζονται οι Micro – Average τιμές ακρίβειας για το Word N – Gram, N = 3 για κάθε ταξινομητή, που υπολογίστηκαν όπως αναφέρθηκε νωρίτερα.

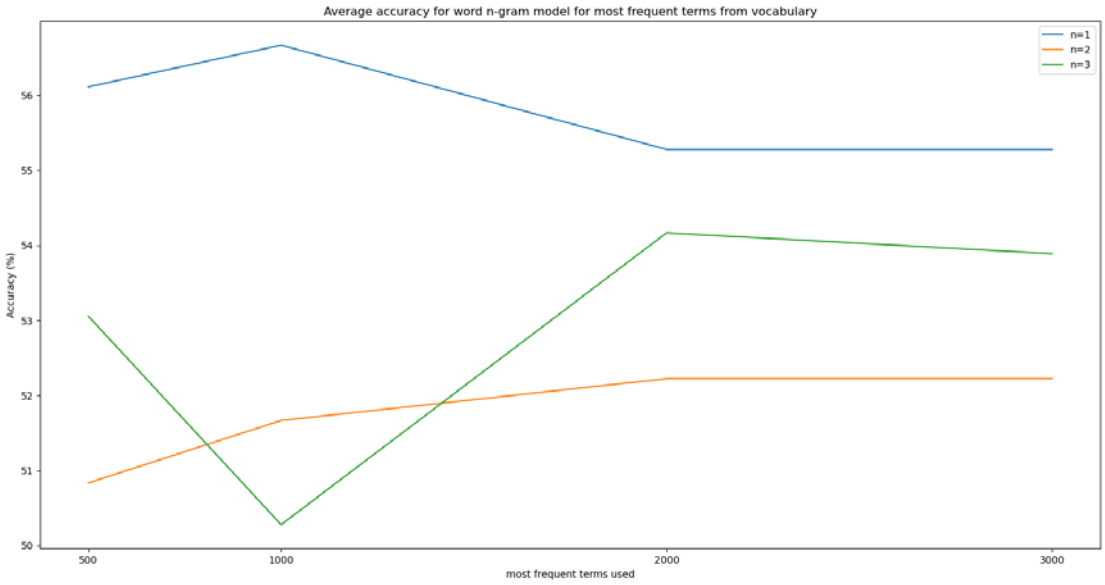


Εικόνα 22: Micro – Average Accuracy για Word N – Gram, N = 3

Η μέγιστη Micro – Average τιμή για το μοντέλο Word N – Gram, N = 3 καταγράφεται στο **55%** για το **Linear SVM**.

7.1.4 Χρήση Ν συχνότερων όρων

Για τις αναπαραστάσεις με Word N – Gram (N = 1, 2, 3), δοκιμάστηκε ο περιορισμός του λεξιλογίου και η χρήση των συχνότερων όρων. Τα διαφορετικά λεξιλόγια που χρησιμοποιήθηκαν για τις δοκιμές αποτελούνται από τους **500**, **1000**, **2000** και **3000** συχνότερους όρους. Παρακάτω παρουσιάζεται η Macro – Average ακρίβεια για κάθε μοντέλο σε σχέση με το πλήθος των συχνότερων όρων. Για ευκολία στη σύγκριση χρησιμοποιήθηκε η Macro – Average ακρίβεια του **Linear SVM**.



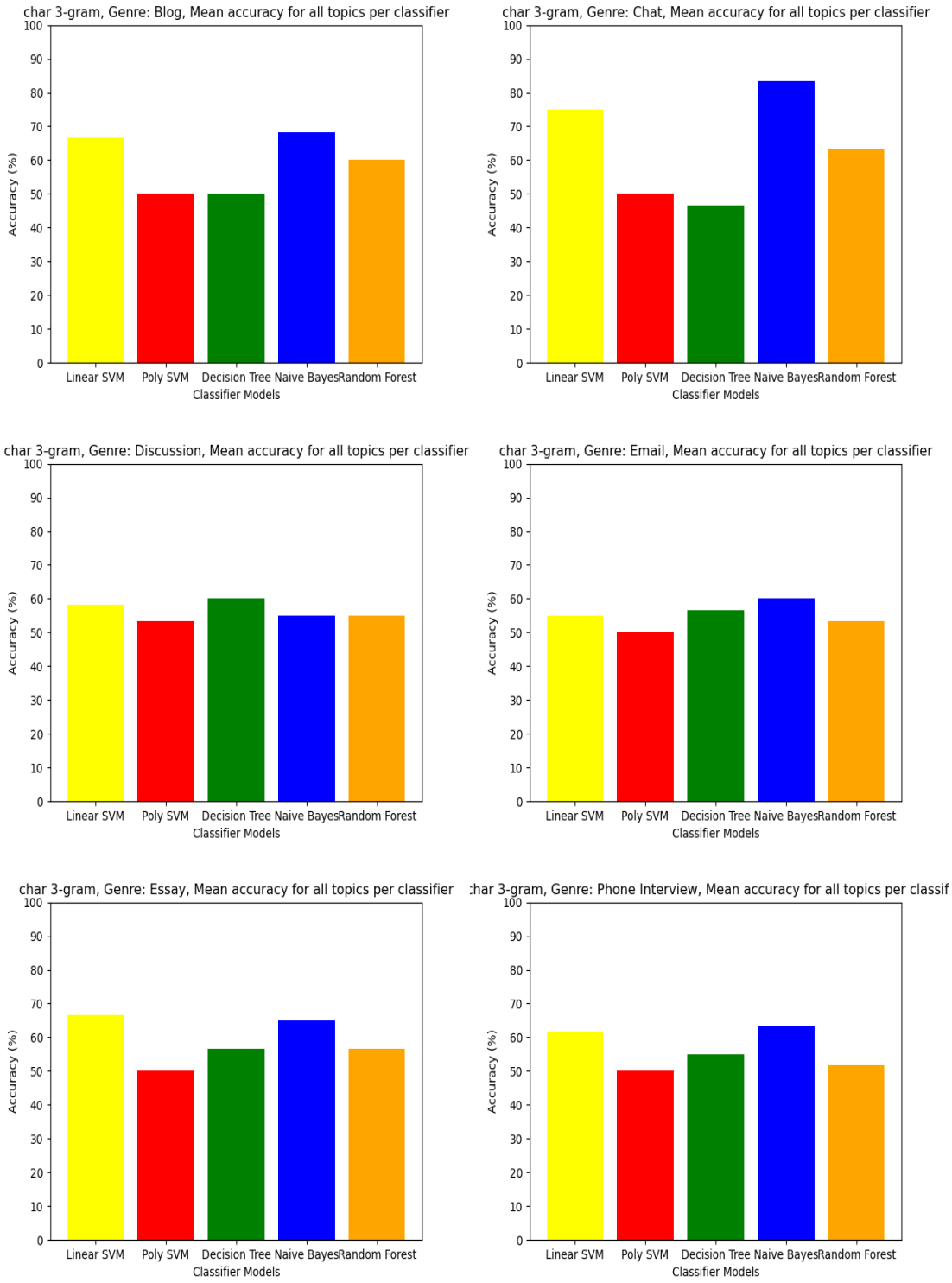
Εικόνα 23 : Macro – Average Accuracy για Word N – Gram με χρήση συχνότερων όρων

Η χρήση των συχνότερων όρων του λεξιλογίου φαίνεται να αυξάνει την ακρίβεια σε σχέση με τη χρήση όλου του λεξιλογίου στις περιπτώσεις **N = 2** και **N = 3**. Πιο συγκεκριμένα **από τους 500 μέχρι 2000** συχνότερους όρους, για την πρώτη περίπτωση, παρατηρείται αύξηση μιας μονάδας και στη συνέχεια μια σταθερότητα. Αντίστοιχα στην δεύτερη περίπτωση, σημειώνεται μια πτώση **μέχρι τους 1000** συχνότερους όρους κι άλλη μία μηδαμινή **μετά τους 2000**, ενώ **μεταξύ 1000 και 2000** μια σημαντική αύξηση 4 μονάδων. Στην περίπτωση όπου **N = 1**, μετά τους 1000 όρους υπάρχει μία πτώση της τάξης της σχεδόν μίας ποσοστιαίας μονάδας.

7.2 Διαχωρισμός σε Character N – Gram

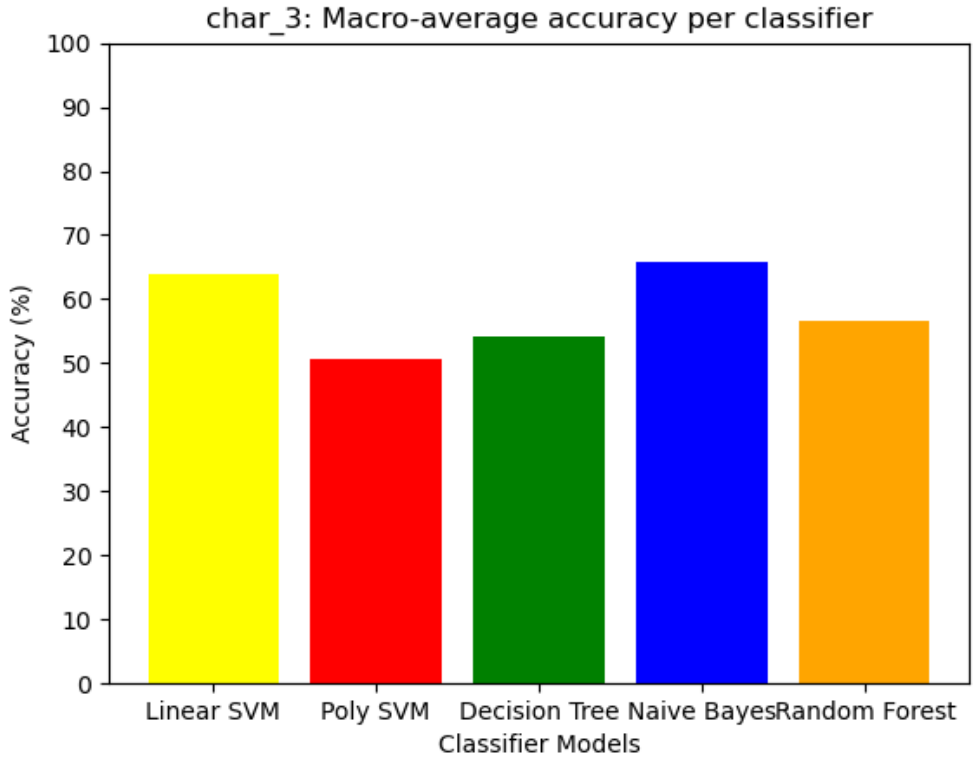
Παρακάτω φαίνονται διαγράμματα της μέσης ακρίβειας (Mean Accuracy) του συστήματος από τα 6 θέματα για κάθε είδος κειμένου που αναπαραστάθηκε ως Character N – Gram, με δοκιμές σε 5 διαφορετικά μοντέλα ταξινόμησης.

7.2.1 Character N – Gram, N = 3



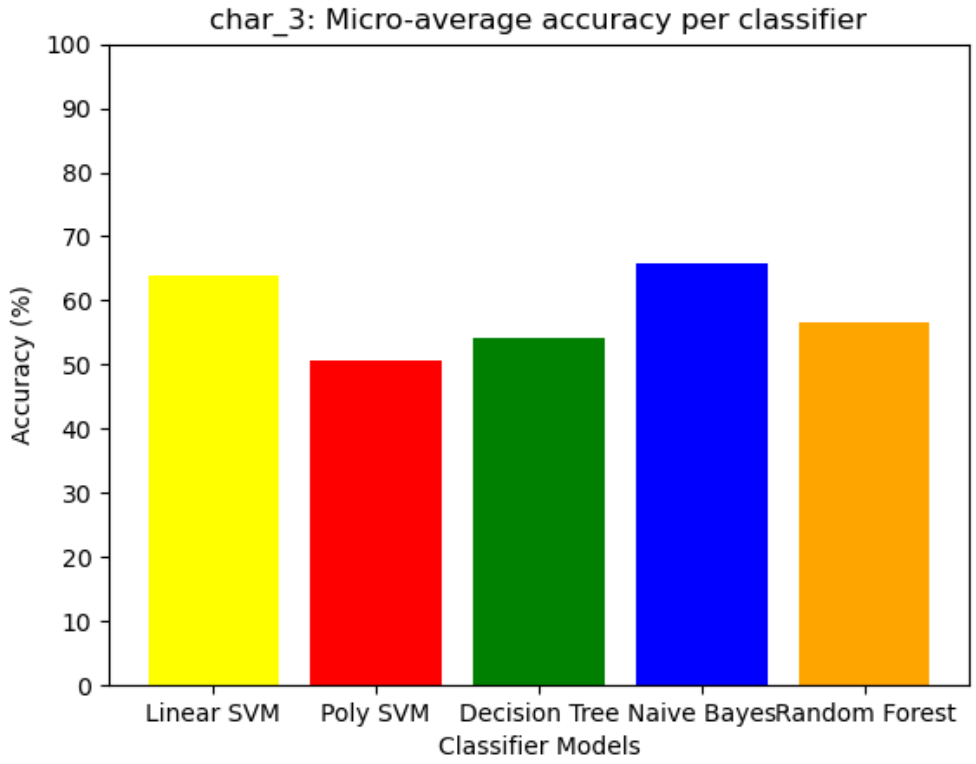
Εικόνα 24: Mean Accuracy για Character N – Gram , N = 3

Η μέγιστη μέση απόδοση **83.5%** παρατηρείται στο είδος **Chat** για το μοντέλο ταξινομητή **Naïve Bayes**. Στην Εικόνα 25, παρουσιάζονται οι Macro – Average τιμές ακρίβειας ανά μοντέλο ταξινομητή, με τον τρόπο που υπολογίστηκαν και πριν.



Εικόνα 25: Macro – Average Accuracy για Character N – Gram, N = 3

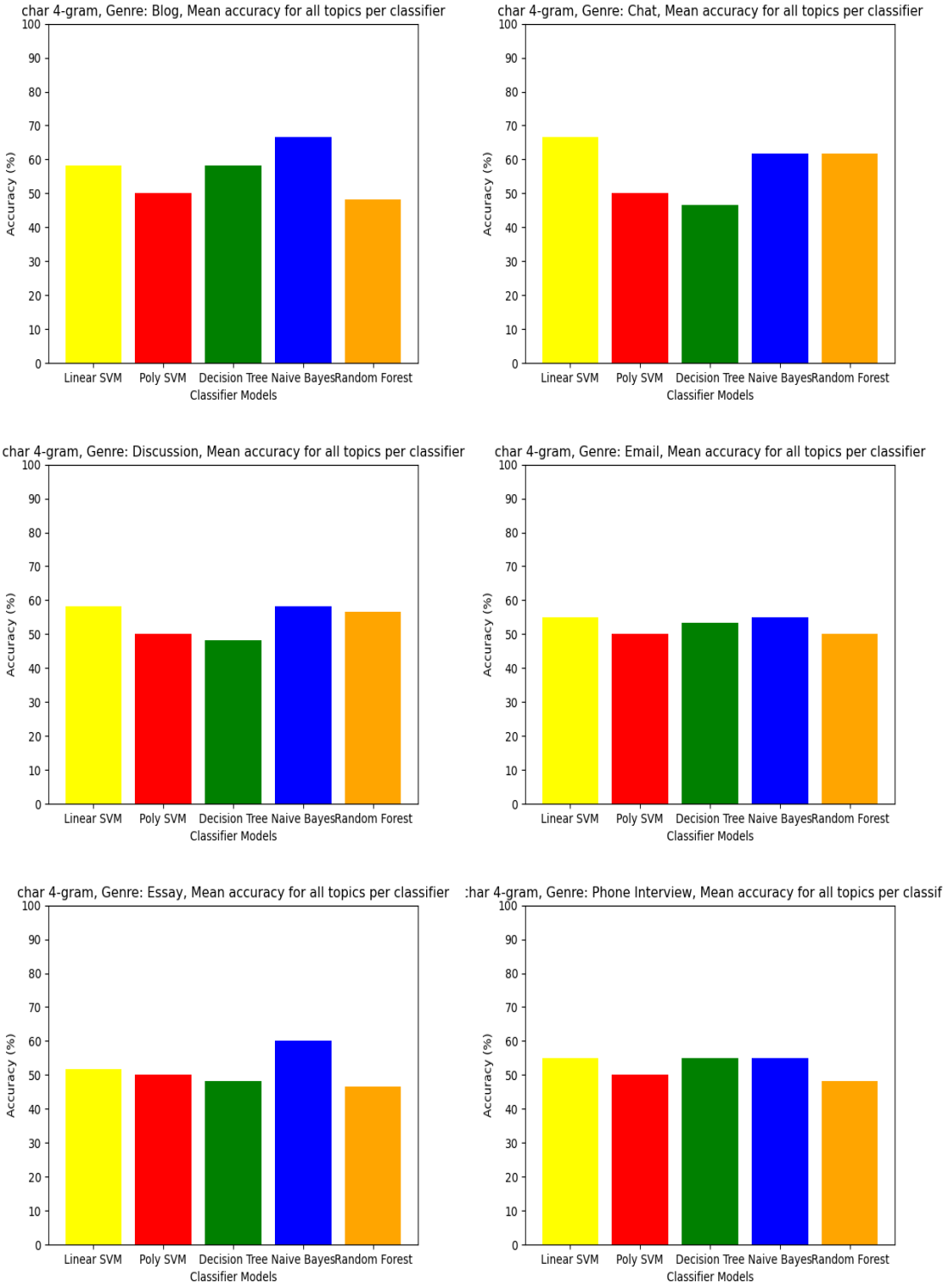
Η μέγιστη Macro – Average τιμή για το μοντέλο Character N – Gram, N = 3 καταγράφεται στο **66%** για το μοντέλο ταξινομητή **Naïve Bayes**. Στην Εικόνα 26, παρουσιάζονται οι Micro – Average τιμές ακρίβειας για το Character N – Gram, N = 3 για κάθε ταξινομητή, που υπολογίστηκαν όπως αναφέρθηκε νωρίτερα.



Εικόνα 26: Micro – Average Accuracy για Character N – Gram, N = 3

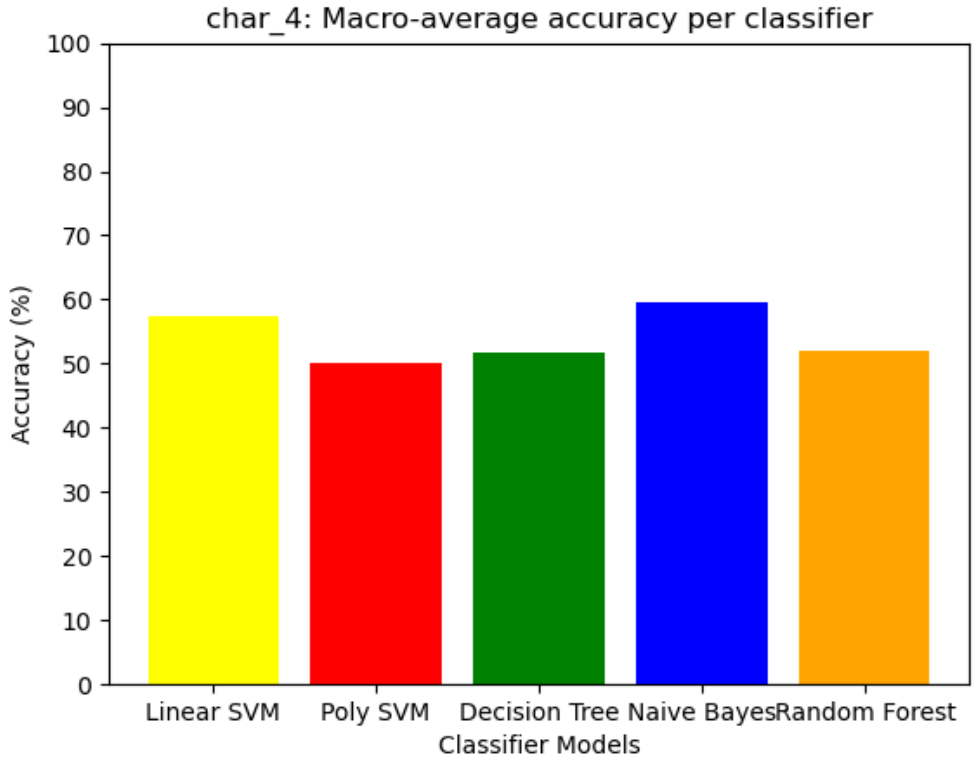
Η μέγιστη Micro – Average τιμή για το μοντέλο Character N – Gram, N = 3 καταγράφεται στο **66%** για το μοντέλο ταξινομητή **Naïve Bayes**.

7.2.2 Character N – Gram, N = 4



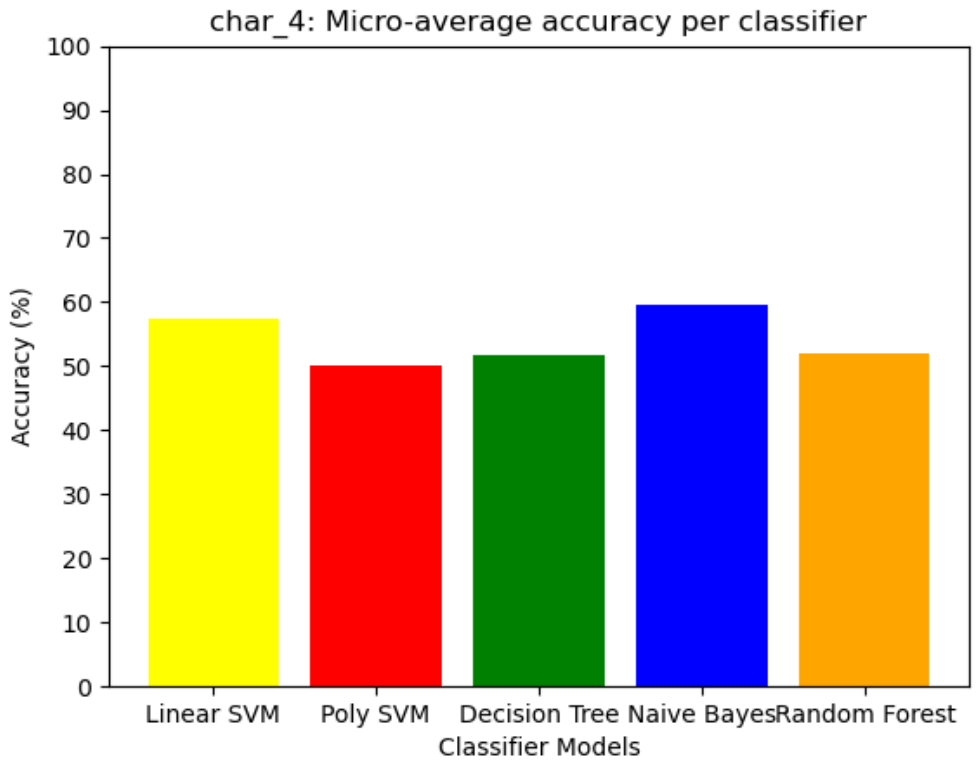
Εικόνα 27: Mean Accuracy για Character N – Gram, N = 4

Η μέγιστη μέση απόδοση **67%** παρατηρείται στα είδη **Blog** και **Chat** για τα μοντέλα ταξινομητών **Naïve Bayes** και **Linear SVM** αντίστοιχα. Στην Εικόνα 28, παρουσιάζονται οι Macro – Average τιμές ακρίβειας ανά μοντέλο ταξινομητή, με τον τρόπο που υπολογίστηκαν και πριν.



Εικόνα 28: Macro – Average Accuracy για Character N – Gram, N = 4

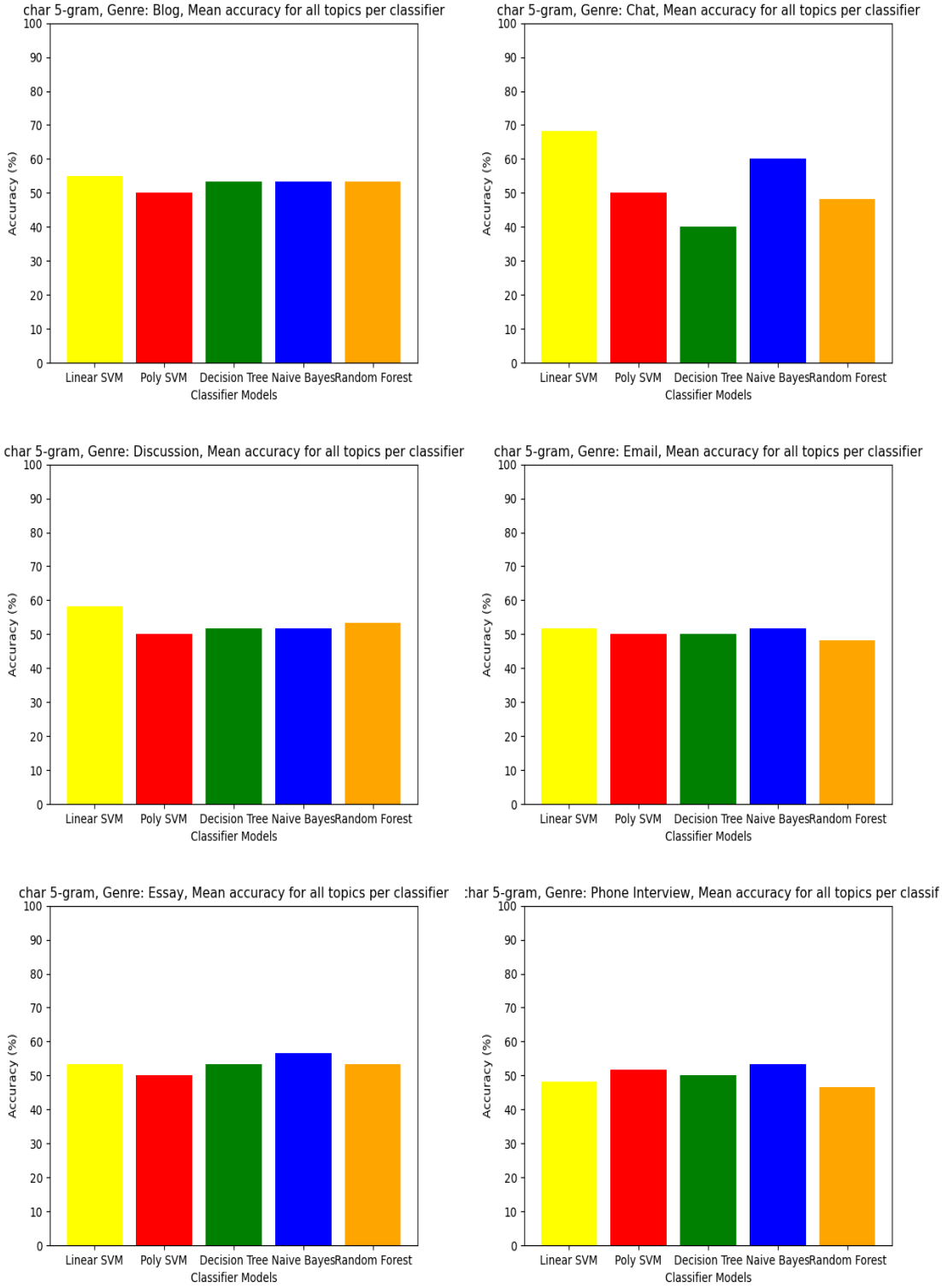
Η μέγιστη Macro – Average τιμή για το μοντέλο Character N – Gram, N = 4 καταγράφεται στο **60%** για το **Naïve Bayes**. Στην Εικόνα 29, παρουσιάζονται οι Macro – Average τιμές ακρίβειας ανά μοντέλο ταξινομητή, με τον τρόπο που υπολογίστηκαν και πριν.



Εικόνα 29: Micro – Average Accuracy για Character N – Gram, N = 4

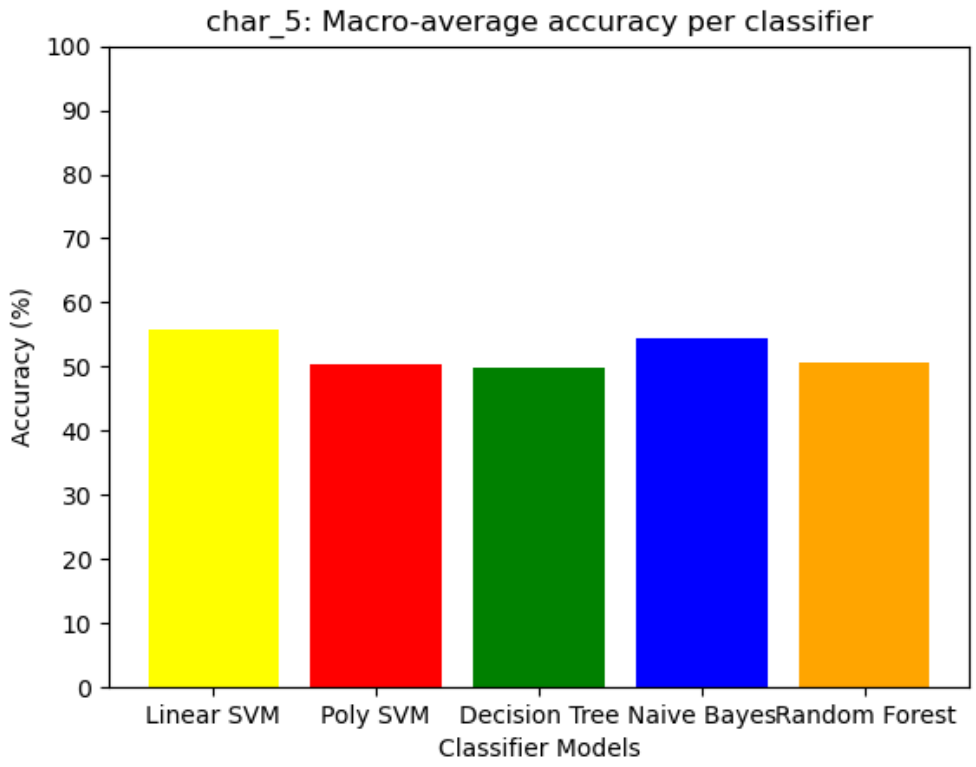
Η μέγιστη Micro – Average τιμή για το μοντέλο Character N – Gram, N = 4 καταγράφεται στο **60%** για το **Naïve Bayes**.

7.2.3 Character N – Gram, N = 5



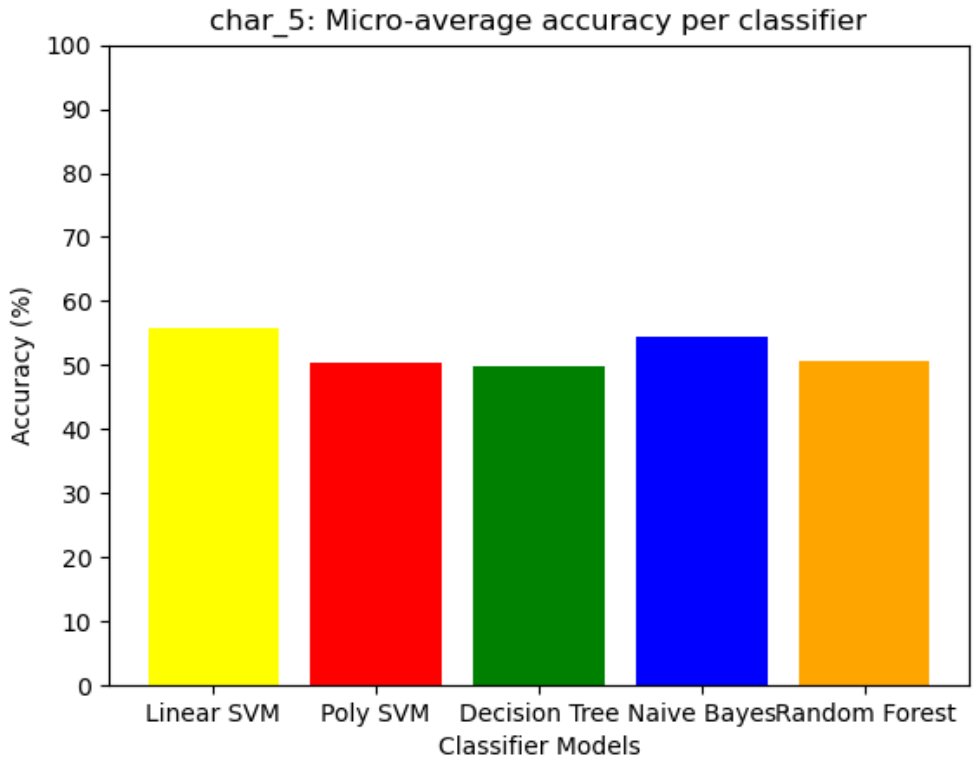
Εικόνα 30: Mean Accuracy για Character N – Gram, N = 5

Η μέγιστη μέση απόδοση **68.5%** παρατηρείται στο είδος **Chat** για το μοντέλο ταξινομητή **Linear SVM**. Στην Εικόνα 31, παρουσιάζονται οι Macro – Average τιμές ακρίβειας ανά μοντέλο ταξινομητή, με τον τρόπο που υπολογίστηκαν και πριν.



Εικόνα 31: Macro – Average Accuracy για Character N – Gram, N = 5

Η μέγιστη Macro – Average τιμή για το μοντέλο Character N – Gram, N = 5 καταγράφεται στο **56%** για το **Linear SVM**. Στην Εικόνα 32, παρουσιάζονται οι Micro – Average τιμές ακρίβειας ανά μοντέλο ταξινομητή, με τον τρόπο που υπολογίστηκαν και πριν.

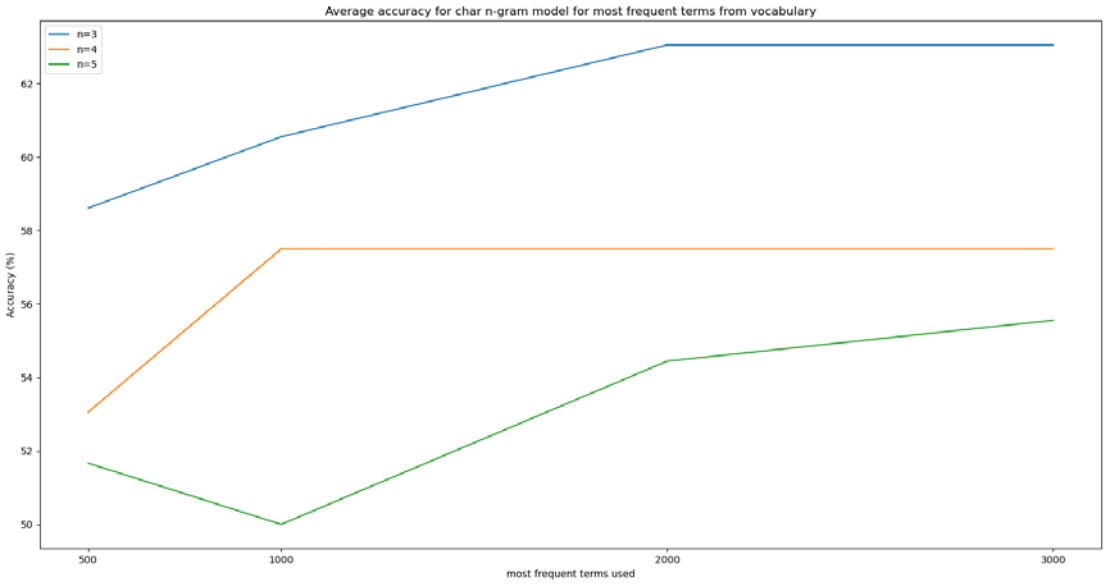


Εικόνα 32: Micro – Average Accuracy για Character N – Gram, N = 5

Η μέγιστη Micro – Average τιμή για το μοντέλο Character N – Gram, N = 5 καταγράφεται στο **56%** για το **Linear SVM**.

7.2.4 Χρήση Ν συχνότερων όρων

Για τις αναπαραστάσεις με Character N – Gram (N = 3, 4, 5), δοκιμάστηκε ο περιορισμός του λεξιλογίου και η χρήση των συχνότερων όρων. Τα διαφορετικά λεξιλόγια που χρησιμοποιήθηκαν για τις δοκιμές αποτελούνται από τους **500**, **1000**, **2000** και **3000** συχνότερους όρους. Παρακάτω παρουσιάζεται η Macro – Average ακρίβεια για κάθε μοντέλο σε σχέση με το πλήθος των συχνότερων όρων. Για ευκολία στη σύγκριση χρησιμοποιήθηκε η Macro – Average ακρίβεια του ταξινομητή **Linear SVM**.



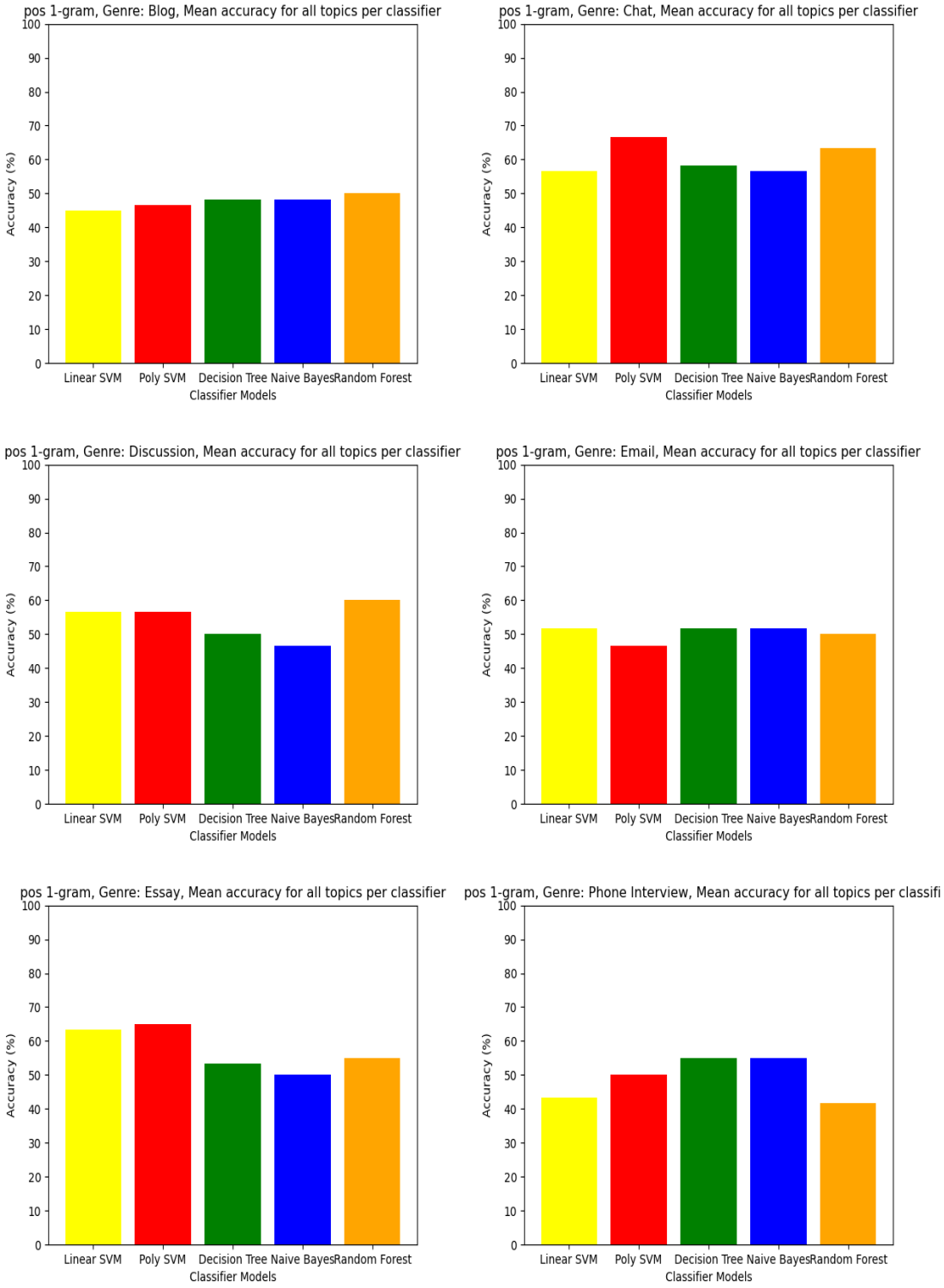
Εικόνα 33 : Macro – Average Accuracy για Character N – Gram με χρήση συχνότερων όρων

Παρατηρείται ότι η χρήση μόνο των συχνότερων όρων του λεξιλογίου αυξάνει την ακρίβεια για όλες τις περιπτώσεις. Για **N = 3** και **N = 5**, η αύξηση παρατηρείται κυρίως για τους **1000 και πάνω** συχνότερους όρους. Στο μοντέλο **Character N – Gram, N = 4**, η ακρίβεια αυξάνεται μέχρι τους 1000 όρους, ενώ έπειτα παραμένει σταθερή.

7.3 Διαχωρισμός σε POS N – Gram

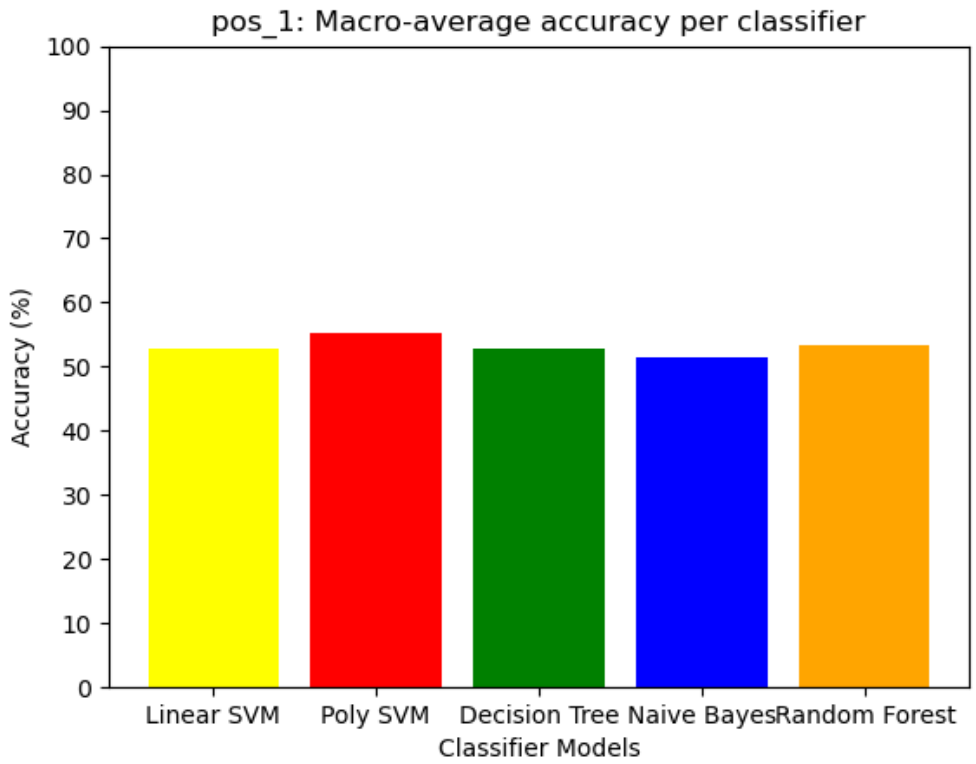
Παρακάτω φαίνονται διαγράμματα της μέσης ακρίβειας (Mean Accuracy) του συστήματος από τα 6 θέματα για κάθε είδος κειμένου που αναπαραστάθηκε ως POS N – Gram, με δοκιμές σε 5 διαφορετικά μοντέλα ταξινόμησης.

7.3.1 POS N – Gram, N = 1



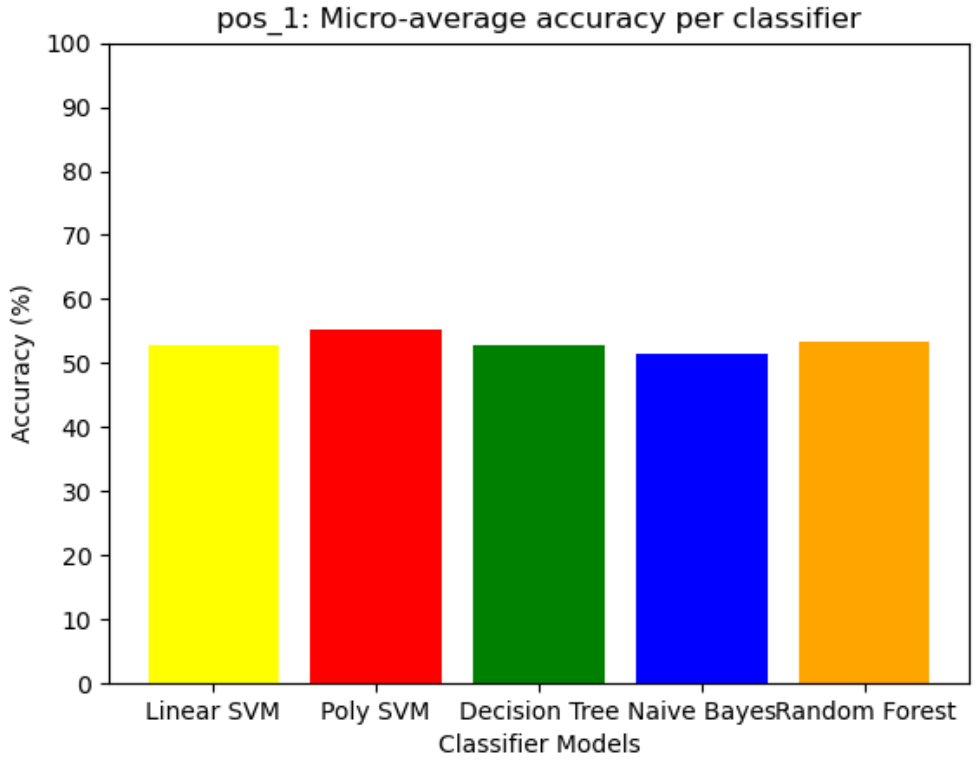
Εικόνα 34: Mean Accuracy για POS N – Gram, N = 1

Η μέγιστη μέση απόδοση **67%** παρατηρείται στο είδος **Chat** για το μοντέλο ταξινομητή **Poly SVM**. Στην Εικόνα 35, παρουσιάζονται οι Macro – Average τιμές ακρίβειας ανά μοντέλο ταξινομητή, με τον τρόπο που υπολογίστηκαν και πριν.



Εικόνα 35: Macro – Average Accuracy για POS N – Gram, N = 1

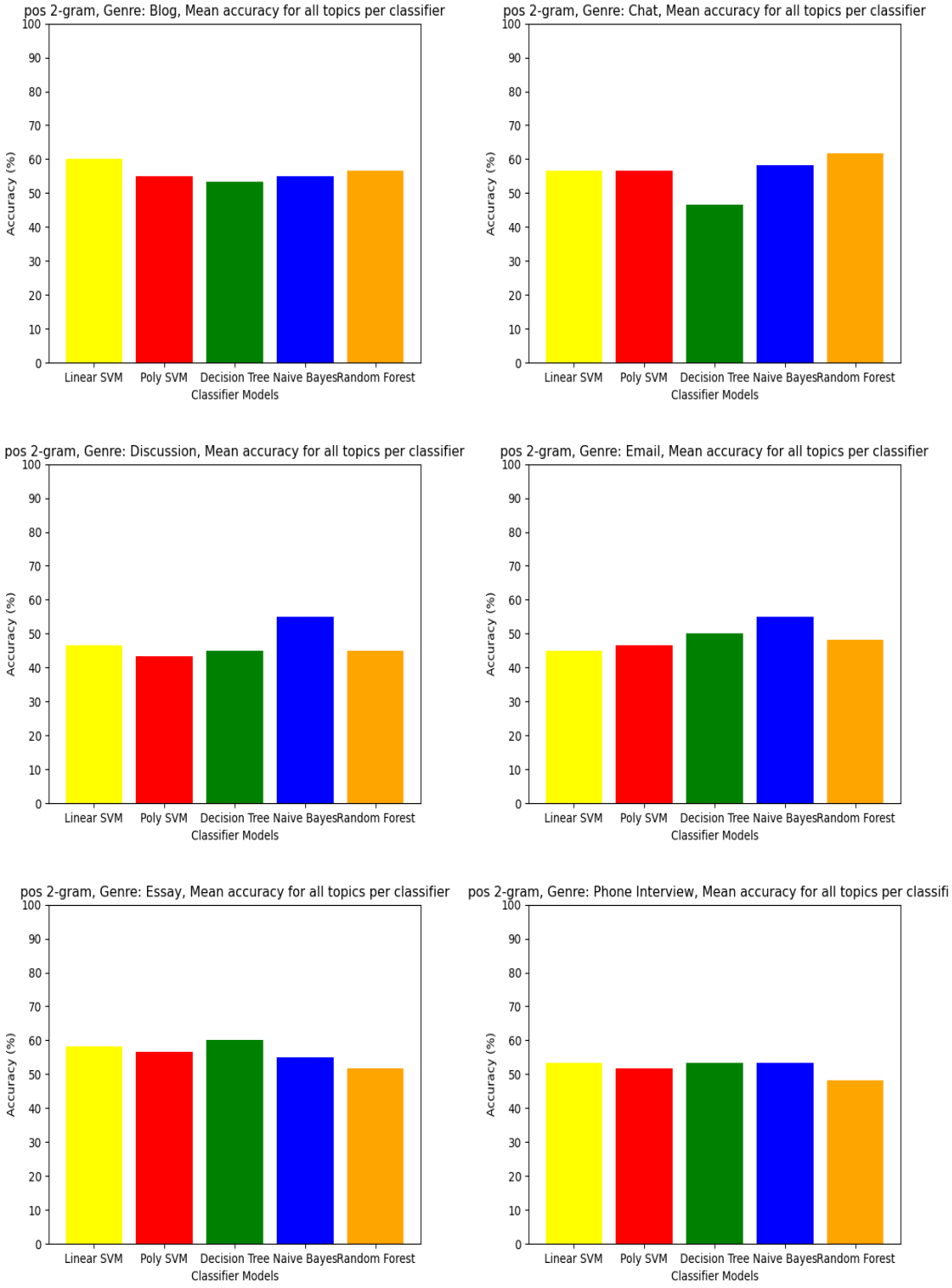
Η μέγιστη Macro – Average τιμή για το μοντέλο POS N – Gram, N = 1 καταγράφεται στο **55.5%** για το **Poly SVM**. Στην Εικόνα 36, παρουσιάζονται οι Micro – Average τιμές ακρίβειας ανά μοντέλο ταξινομητή, με τον τρόπο που υπολογίστηκαν και πριν.



Εικόνα 36: Micro – Average Accuracy για POS N – Gram, N = 1

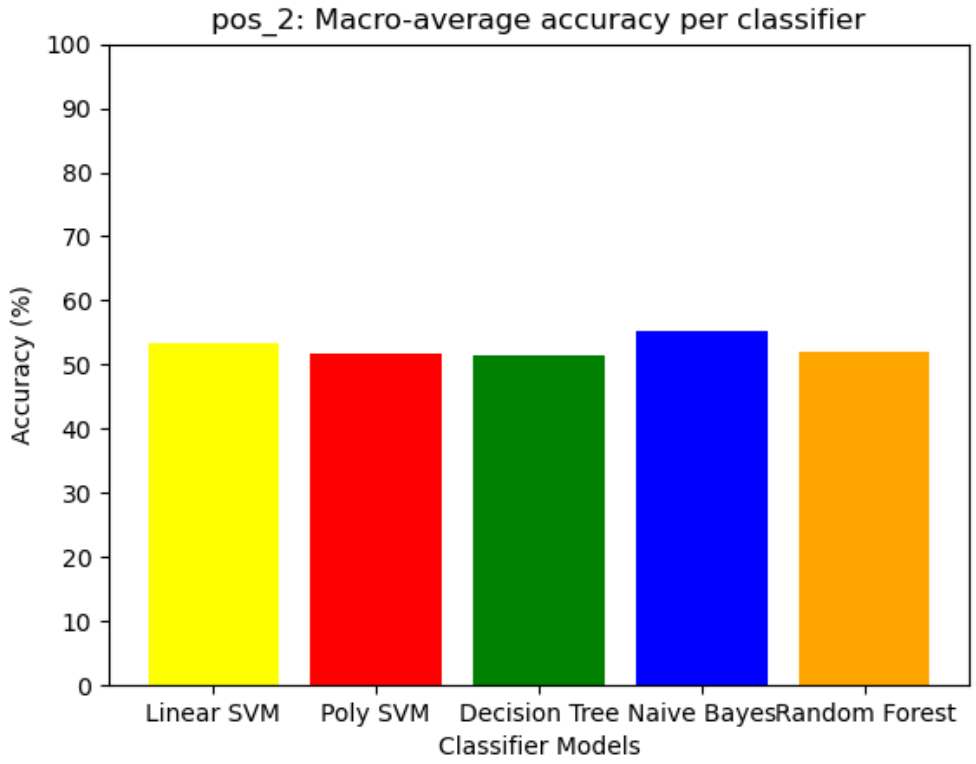
Η μέγιστη Micro – Average τιμή για το μοντέλο POS N – Gram, N = 1 καταγράφεται στο **55.5%** για το **Poly SVM**.

7.3.2 POS N – Gram, N = 2



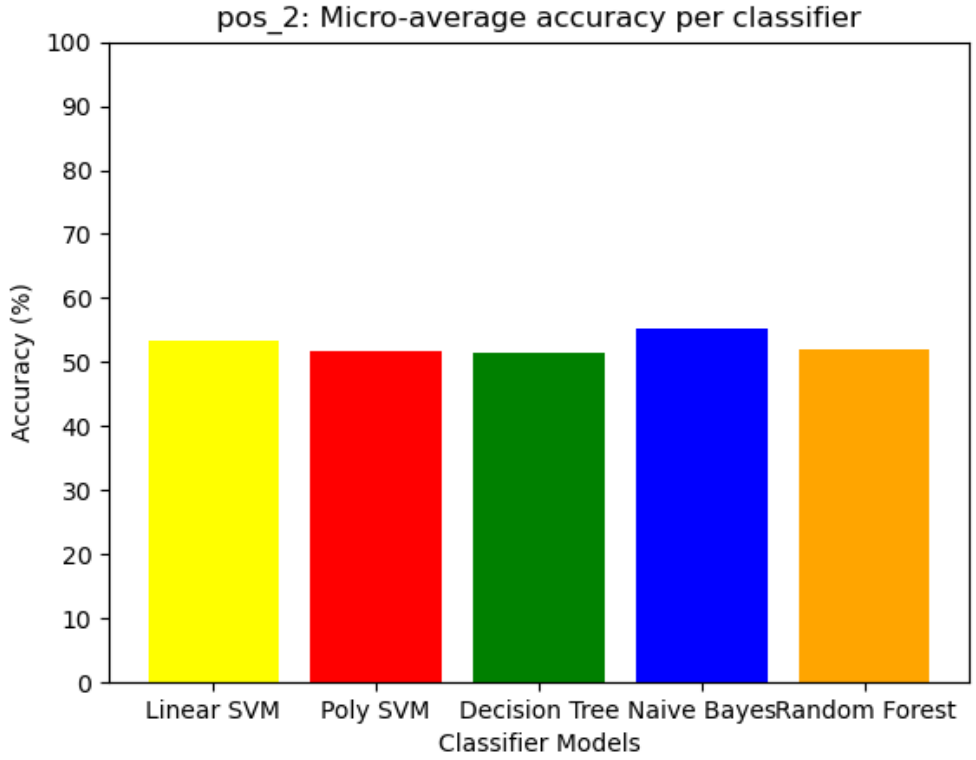
Εικόνα 37: Mean Accuracy για POS N – Gram, N = 2

Η μέγιστη μέση απόδοση **60%** παρατηρείται στα είδη **Blog** και **Essay** για τα μοντέλα ταξινομητών **Linear SVM** και **Decision Tree**. Στην Εικόνα 38, παρουσιάζονται οι Macro – Average τιμές ακρίβειας ανά μοντέλο ταξινομητή, με τον τρόπο που υπολογίστηκαν και πριν.



Εικόνα 38: Macro – Average Accuracy για POS N – Gram, N = 2

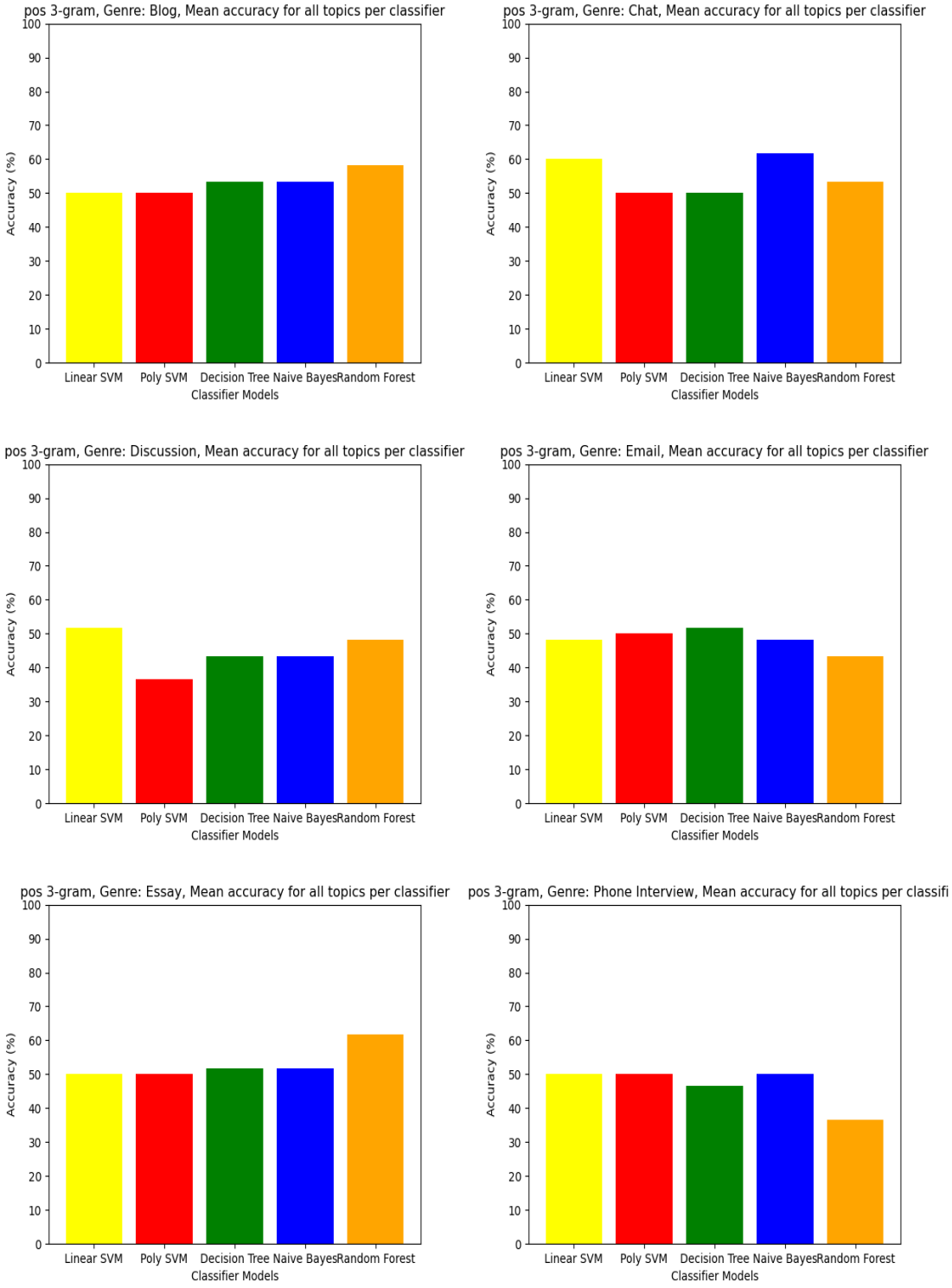
Η μέγιστη Macro – Average τιμή για το μοντέλο POS N – Gram, N = 2 καταγράφεται στο **55.5%** για το **Naïve Bayes**. Στην Εικόνα 39, παρουσιάζονται οι Micro – Average τιμές ακρίβειας ανά μοντέλο ταξινομητή, με τον τρόπο που υπολογίστηκαν και πριν.



Εικόνα 39: Micro – Average Accuracy για POS N – Gram, N = 2

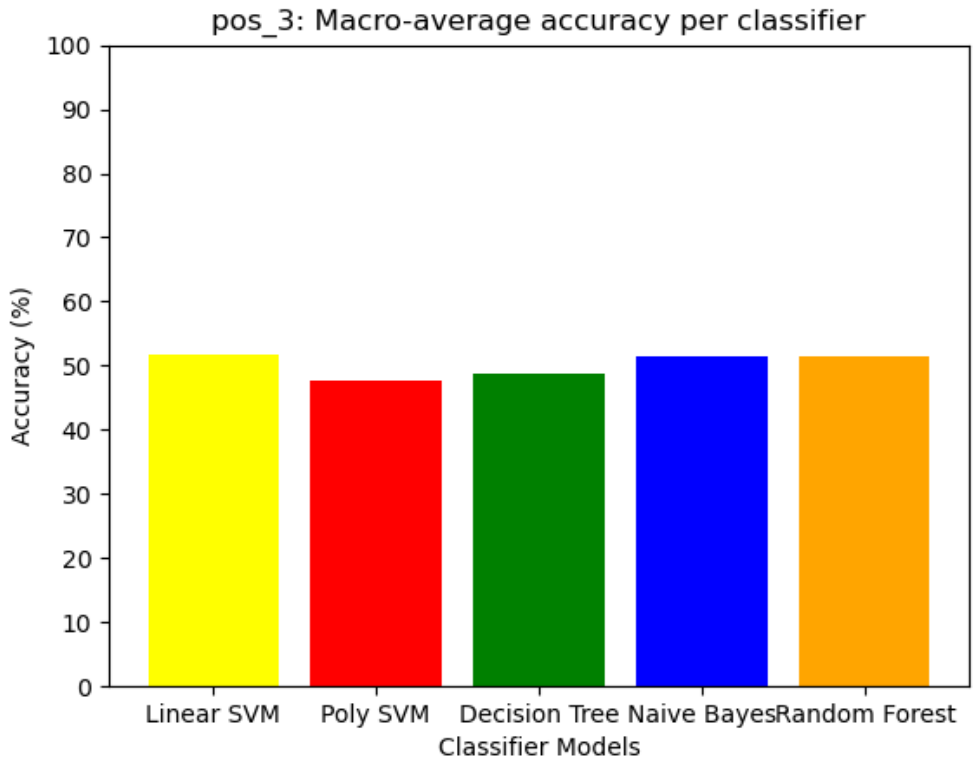
Η μέγιστη Micro – Average τιμή για το μοντέλο POS N – Gram, N = 2 καταγράφεται στο **55.5%** για το **Naïve Bayes**.

7.3.3 POS N – Gram, N = 3



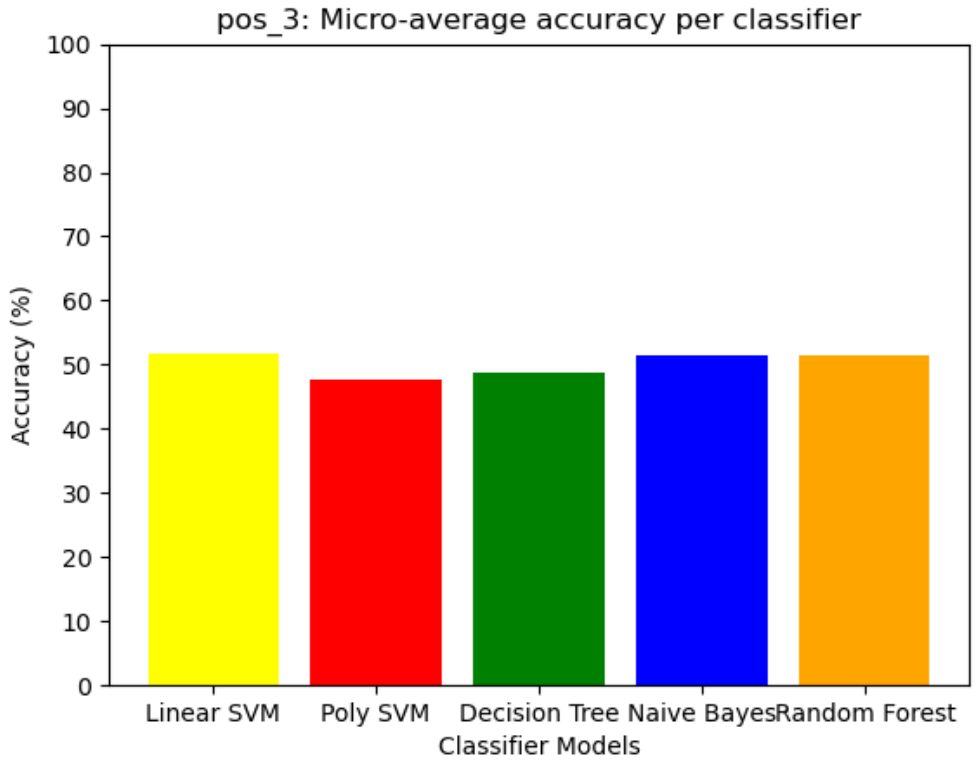
Εικόνα 40: Mean Accuracy για POS N – Gram, N = 3

Η μέγιστη μέση απόδοση **62%** παρατηρείται στα είδη **Chat** και **Essay** για τα μοντέλα ταξινομητών **Naïve Bayes** και **Random Forest**. Στην Εικόνα 41, παρουσιάζονται οι Macro – Average τιμές ακρίβειας ανά μοντέλο ταξινομητή, με τον τρόπο που υπολογίστηκαν και πριν.



Εικόνα 41: Macro – Average Accuracy για POS N – Gram, N = 3

Η μέγιστη Macro – Average τιμή για το μοντέλο POS N – Gram, N = 3 καταγράφεται στο **52%** για το **Linear SVM**. Στην Εικόνα 42, παρουσιάζονται οι Micro – Average τιμές ακρίβειας ανά μοντέλο ταξινομητή, με τον τρόπο που υπολογίστηκαν και πριν.

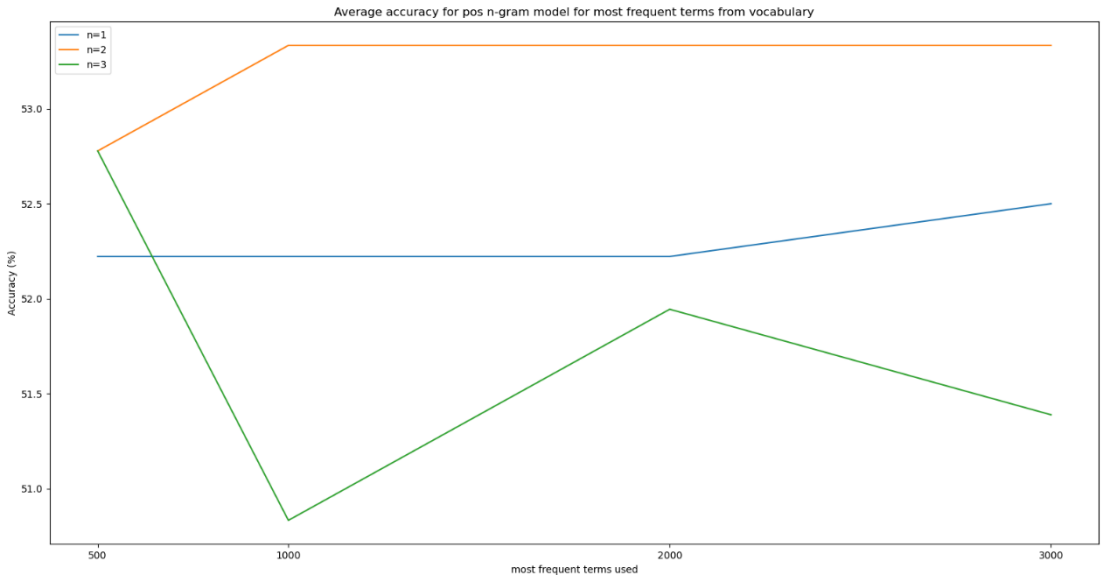


Εικόνα 42: Micro – Average Accuracy για POS N – Gram, N = 3

Η μέγιστη Micro – Average τιμή για το μοντέλο POS N – Gram, N = 3 καταγράφεται στο **52%** για το **Linear SVM**.

7.3.4 Χρήση Ν συχνότερων όρων

Για τις αναπαραστάσεις με POS N – Gram (N = 1, 2, 3), δοκιμάστηκε ο περιορισμός του λεξιλογίου και η χρήση των συχνότερων όρων. Τα διαφορετικά λεξιλόγια που χρησιμοποιήθηκαν για τις δοκιμές αποτελούνται από τους **500**, **1000**, **2000** και **3000** συχνότερους όρους. Παρακάτω παρουσιάζεται η Macro – Average ακρίβεια για όλα τα μοντέλα σε σχέση με το πλήθος των συχνότερων όρων. Για ευκολία στη σύγκριση χρησιμοποιήθηκε ο ταξινομητής **Linear SVM**.



Εικόνα 43: Macro – Average Accuracy για POS N – Gram με χρήση συχνότερων όρων

Παρατηρείται ότι η χρήση μόνο των συχνότερων όρων του λεξιλογίου αυξάνει την ακρίβεια για όλες τις περιπτώσεις εκτός από **N = 3**, όπου σημειώνονται αυξομειώσεις. Συγκεκριμένα, για τις περιπτώσεις όπου **N = 1** και **N = 2**, **κυρίως μεταξύ 1000 και 2000** συχνότερων όρων, υπάρχει σταθερότητα. Αντιθέτως, στην περίπτωση όπου **N = 3**, φαίνεται να υπάρχει μια αύξηση της ακρίβειας για το συγκεκριμένο εύρος συχνότερων όρων.

7.4 Σύγκριση όλων των μοντέλων

Στον Πίνακα 1, παρουσιάζονται οι Macro – Average και Micro – Average ακρίβειες για όλα τα είδη κειμένου για τα 9 μοντέλα που δοκιμάστηκαν, λαμβάνοντας τον Linear SVM ταξινομητή για το καθένα:

Model		Macro – Average Accuracy (%)	Micro – Average Accuracy (%)
Word N – Gram	**N = 1**	56.1	56
	N = 2	50.5	50.5
	N = 3	55	55
Character N – Gram	**N = 3**	**64.2**	**64**
	N = 4	58	58
	N = 5	56	56
POS N – Gram	**N = 1**	53	53
	N = 2	53.5	53.5
	N = 3	52	52

Πίνακας 1: Macro/Micro – Average Accuracy (SVM Classifier) ανά μοντέλο N – Gram

Οι μέγιστες τιμές των Macro και Micro – Average ακριβειών (**64.2%** και **64%** αντίστοιχα) σημειώνονται για το μοντέλο **Character N – Gram, N = 3**. Συνολικά, τα μοντέλα **Character N – Gram** παρουσιάζουν τις υψηλότερες τιμές μέσης ακρίβειας.

Στον Πίνακα 2, παρουσιάζεται η μέση ακρίβεια ανά είδος κειμένου για τα 9 μοντέλα που δοκιμάστηκαν, λαμβάνοντας τον Linear SVM ταξινομητή για το κάθε είδος:

		Mean Accuracy (%)					
Model		**Blog**	**Chat**	**Discussion**	**Email**	**Essay**	**Phone Interview**
Word N – Gram	**N = 1**	57	55	53.5	48.5	58.5	**63.5**
	N = 2	52	52	53.5	**55.5**	43.5	49
	N = 3	45.5	57	**62**	52	58.5	54
Character N – Gram	**N = 3**	**67**	**75**	58.5	55	**67**	62
	N = 4	58.5	67	58.5	**55.5**	52	55.5
	N = 5	55	68.5	58.5	52	53.5	48.5
POS N – Gram	**N = 1**	45	57	57	52	64	43.5
	N = 2	60	57	47	45	58.5	53
	N = 3	50.5	60	52	48.5	50.5	50

Πίνακας 2: Mean Accuracy ανά Genre κειμένου (SVM Classifier) ανά μοντέλο N – Gram

Η μέγιστη τιμή μέσης ακρίβειας **(75%)** σημειώνεται για το είδος **Chat** για το μοντέλο **Character N – Gram, N = 3**. Στις περισσότερες περιπτώσεις, οι μέγιστες τιμές ανά είδος κειμένου καταγράφονται για το μοντέλο **Character N – Gram, N = 3**.

Στον Πίνακα 3, παρουσιάζονται οι **Macro – Average** και **Micro – Average** ακρίβειες ανά είδος κειμένου για τα 9 μοντέλα που δοκιμάστηκαν, λαμβάνοντας υπόψιν μόνο τους N συχνότερους όρους **μεταξύ 500 και 3000**:

Genres		Blog		Chat		Discussion		Email		Essay		Phone Interview	
Model	**N**	**Macro (%)**	**Micro (%)**	**Macro (%)**	**Micro (%)**	**Macro (%)**	**Micro (%)**	**Macro (%)**	**Micro (%)**	**Macro (%)**	**Micro (%)**	**Macro (%)**	**Micro (%)**
Word 1 – Gram	**500**	49.7	49.6	57.1	57	**54.8**	54.6	**53.5**	53.3	53.7	53.3	59.2	59
	1000	51.4	51.3	58	57.6	53.3	53	50.4	50.3	**53.8**	53.6	**60.5**	60.3
	2000	**51.6**	51.3	**59**	58.6	49.2	49	52.6	52.3	51.9	51.6	58.8	58.3
	3000	51.2	51	53.7	53.6	47.3	47	52	51.6	53.7	53.3	57.6	57.3
Word 2 – Gram	**500**	52.3	52	45.5	45.3	50.8	50.6	**56.2**	56	46.3	46	**55.9**	55.6
	1000	48.1	48	46.8	46.6	**51.2**	51	52.8	52.6	**48.8**	48.6	55.4	55
	2000	50.3	50	**48.6**	48.3	49	48.6	55.3	55	48.2	48	52.9	52.6
	3000	**54**	53.6	45.3	45	50.7	50.3	51.4	51	47	46.6	54	53.6
Word 3 – Gram	**500**	48.5	48.3	45.2	45	53.8	53.6	52.2	52	**58.1**	58	47.4	47.3
	1000	46.1	46	**47.2**	47	50.4	50	**55.5**	55.3	54.1	54	**49**	48.6
	2000	**49.7**	49.3	47	46.6	53.2	53	54	53.6	56	55.6	48.7	48.3
	3000	49.3	49	46.7	46.3	**54.3**	54	51.8	51.3	57.6	57.3	48.4	48
Char 3 – Gram	**500**	56.5	56.3	62.5	62.3	57.2	57	52.1	52	55.1	55	**62.8**	62.6
	1000	59.3	59	63.5	63.3	56.3	56	55.4	55	60.6	60.3	59.4	59
	2000	58.6	58.3	**67.6**	67.3	**58.6**	58.3	53.7	53.3	**65**	64.6	58.3	58
	3000	**60.2**	60	67.3	67	57.9	57.6	**58.3**	58	58.5	58.3	59.2	59
Char 4 – Gram	**500**	53.2	53	59.5	59.3	51.5	51.3	47.2	47	49.5	49.3	58.9	58.6
	1000	**56.9**	56.6	**61.4**	61	52.3	52	50.4	50	52.3	52	**59.7**	59.3
	2000	53.2	53	60	59.6	**57.7**	57.3	**54.1**	53.6	**55**	54.6	55.7	55.3
	3000	56	55.6	58.9	58.6	56.6	56.3	51.6	51.3	**55**	54.6	57	56.6
Char 5 – Gram	**500**	**52.2**	52	**58.1**	58	47.5	47.3	51.2	51	46.5	46.3	**53.9**	53.6
	1000	46.9	46.6	54.3	54	51.2	51	45.3	45	50.7	50.3	51.4	51
	2000	49	48.6	**58.1**	57.6	**54.2**	54	**52.1**	51.6	51.9	51.6	49.3	49
	3000	52	51.6	56.6	56.3	52.3	52	51.7	51.3	**53.7**	53.3	47.3	47
POS 1 – Gram	**500**	48.5	48.3	60.3	60	54.3	54	51	50.6	59.1	58.6	49.6	49.3
	1000	48.3	48	**61.3**	61	55	54.6	50.6	50.3	57.5	57.3	**52.1**	52
	2000	**50**	49.6	60.6	60.3	**55.9**	55.6	50.6	50.3	**61**	60.6	50.2	50
	3000	49.3	49	60.2	60	53	52.6	**51.7**	51.6	59	58.6	51.8	51.3
POS 2 – Gram	**500**	55.5	55.3	**56.5**	56.3	**51.6**	51.3	50.2	50	53.5	53.3	51.2	51
	1000	55.7	55.6	53.2	53	48.3	48	50.2	50	56.2	56	51.6	51.3
	2000	54.1	54	52.7	52.3	47.5	47.3	49.3	49	56.2	56	**53.7**	53.3
	3000	**55.8**	55.6	53.3	53	46.6	46.3	**50.9**	50.6	**57.9**	57.6	53.5	53.3
POS 3 – Gram	**500**	51.9	51.6	54.7	54.3	**58.8**	58.6	46.5	46.3	**57.8**	57.6	50.1	50
	1000	**53.5**	53.3	53.9	53.6	51.2	51	50.8	50.6	52.6	52.3	48.4	48
	2000	52	51.6	51.5	51.3	47.5	47.3	48.6	48.3	55.3	55	**52**	51.6
	3000	50	49.6	**54.9**	54.6	46.7	46.3	**50.9**	50.6	52.3	52	48.2	48

Πίνακας 3: Macro/Micro – Average Accuracy ανά Genre κειμένου ανά μοντέλο N – Gram για N συχνότερους όρους

Με βάση τον παραπάνω πίνακα, οι Macro – Average ακρίβειες προκύπτουν παίρνοντας πρώτα τον μέσο όρο για όλα τα θέματα κι έπειτα τον μέσο όρο των ταξινομητών για κάθε είδος κειμένου. Αντίστοιχα, οι Micro – Average ακρίβειες προκύπτουν απ' τον μέσο όρο θεμάτων και ταξινομητών μαζί. Συγκρίνοντας τις Macro – Average ακρίβειες των 9 μοντέλων παρατηρείται ότι όσο μικρό είναι το N ενός μοντέλου, υπάρχει αύξηση της ακρίβειας για πάνω από 1000 συχνότερους όρους. Συγκεκριμένα, τα μοντέλα Character N – Gram πετυχαίνουν μεγάλες ακρίβειες στο είδος Chat άνω των 1000 συχνότερων όρων. Επίσης, σχεδόν σε όλες τις περιπτώσεις, όσον αφορά τα είδη κειμένου, καλύτερες ακρίβειες επιτυγχάνονται για το μοντέλο Character N – Gram, N = 3, κυρίως μεταξύ 2000 και 3000 συχνότερων όρων. Τώρα ξεχωριστά για 500, 1000, 2000 και 3000 συχνότερους όρους, σχεδόν σε όλες τις περιπτώσεις των μοντέλων υπερτερεί το είδος Chat, εκτός από τους 2000 όπου υπάρχει μια ισορροπία μεταξύ των ειδών Chat και Essay. Πιο συγκεκριμένα, στα μοντέλα Character N – Gram επιτυγχάνονται μεγάλες ακρίβειες στο είδος Chat, ενώ στα μοντέλα POS N – Gram στο είδος Essay. Αντίστοιχα, οι ίδιες παρατηρήσεις ισχύουν και για τις Micro – Average ακρίβειες. Ωστόσο, οι Macro – Average ακρίβειες παρουσιάζουν καλύτερα αποτελέσματα απ' τις Micro.

8 Συμπεράσματα & Μελλοντικές Επεκτάσεις

8.1 Συμπεράσματα

Συνοψίζοντας, στη συγκεκριμένη εργασία έγινε προσπάθεια για αναγνώριση του συγγραφικού προφίλ και κατηγοριοποίηση συγγραφέων με βάση το φύλο, με εφαρμογή τεχνικών αναπαράστασης εγγράφου και ταξινόμησης κειμένου με μηχανική μάθηση. Η αναπαράσταση έγινε με τον χωρισμό των κειμένων σε βασικές μονάδες (Tokens) και με εφαρμογή της θεωρίας N – Gram για λέξεις, χαρακτήρες και όρους ως συντακτικά μέρη.

Εξετάζοντας τα αποτελέσματα ακρίβειας για τα 3 είδη N – Gram (Word, Character, POS), **μέγιστη Macro – Average ακρίβεια (64.2%)** φαίνεται να επιτυγχάνεται για το μοντέλο **Character N – Gram, N = 3**. Επίσης, **μέγιστη Micro – Average ακρίβεια (64%)** καταγράφηκε για το ίδιο μοντέλο. Γενικά, παρατηρήθηκε ότι η ακρίβεια φθίνει όσο αυξάνεται το N. Σχετικά με την επιλογή των συχνότερων όρων έναντι ολόκληρου του λεξιλογίου, στις περισσότερες περιπτώσεις φαίνεται να αυξάνει την ακρίβεια. Οι πιο σημαντικές αυξήσεις παρατηρήθηκαν στα μοντέλα **Character N – Gram**, όταν οι συχνότεροι όροι ξεπερνούσαν κυρίως τους **1000**.

Αν εξετάσουμε την ακρίβεια ως προς το είδος του κειμένου, σύμφωνα με τα αναλυτικά διαγράμματα στο Παράρτημα ΙΙ, τα καλύτερα αποτελέσματα παρατηρούνται σχεδόν πάντα για το είδος **Chat**. Το γεγονός αυτό μπορεί να οφείλεται στην μεγαλύτερη αμεσότητα και αυθορμητισμό που χαρακτηρίζει αυτές τις μορφές λόγου, επιτρέποντας στα άτομα να εκφραστούν απεριόριστα και εξασφαλίζοντας, έτσι, περισσότερο και πιο αντιπροσωπευτικό για τον χαρακτήρα του ατόμου κείμενο. Τα χαμηλότερα ποσοστά συνήθως παρατηρούνται στο είδος **Email**.

Παρατηρώντας τα αποτελέσματα σε επίπεδο θέματος κειμένου, όπως φαίνεται στα αναλυτικά διαγράμματα στο Παράρτημα ΙΙ, στα περισσότερα από τα 9 μοντέλα, οι μέγιστες ακρίβειες καταγράφονται για τα θέματα **“Gay Marriage”** και **“Privacy”**.

Θα πρέπει να σημειωθεί ότι η ακρίβεια που παρουσιάστηκε ως μέσος όρος όλων των θεμάτων για κάθε μοντέλο παρουσιάζει σχετικά χαμηλές τιμές **(περίπου μέχρι 70%)**. Αυτό οφείλεται στο γεγονός ότι, παρατηρώντας το Παράρτημα ΙΙ, για κάθε μοντέλο ταξινομητή η ακρίβεια ανά θέμα παρουσίαζε σημαντικές διακυμάνσεις. Για παράδειγμα, για ένα θέμα μπορεί να σημείωνε ποσοστό 90%, αλλά για άλλο μόλις 40%, οπότε η μέση τιμή που χρησιμοποιήσαμε για όλα τα θέματα μειώνεται σημαντικά. Δοκιμάστηκαν αρκετές διαφορετικές παράμετροι στα μοντέλα ταξινόμησης για να βελτιωθούν οι τιμές. Φαίνεται, τελικά, πως κάποια θέματα δεν παρέχουν τόσο χαρακτηριστική πληροφορία όσο άλλα, ώστε το μοντέλο να αναγνωρίζει επιτυχώς το φύλο.

8.2 Μελλοντικές Επεκτάσεις

Ως εξέλιξη της υλοποίησης που πραγματοποιήθηκε στο πλαίσιο της συγκεκριμένης εργασίας, θα μπορούσε να γίνει κατηγοριοποίηση των συγγραφέων ως προς άλλα χαρακτηριστικά εκτός από το φύλο, όπως η ηλικία, εφόσον υπάρχουν διαθέσιμα δεδομένα. Σχετικά με τη φάση της προεπεξεργασίας, θα μπορούσε να ενσωματωθεί μία διαδικασία λημματοποίησης των όρων, ώστε να εκμεταλλευτεί το μοντέλο την κοινή ρίζα κάποιων όρων του λεξιλογίου και να εκπαιδευτεί καλύτερα. Επίσης, θα μπορούσαν να συνδυαστούν τα διαφορετικά μοντέλα αναπαράστασης κειμένου (Word/POS/Character) για την εκπαίδευση του ίδιου μοντέλου κατηγοριοποίησης.

Βιβλιογραφία

[1] K Santosh, Romil Bansal, Mihir Shekhar, and Vasudeva Varma, “Author Profiling: Predicting Age and Gender from Blogs”, Notebook for PAN at CLEF 2013.

[2] Francisco Rangel, Paolo Rosso, Martin Potthast, Benno Stein. “Overview of the 5th Author Profiling Task at PAN 2017: Gender and Language Variety Identification in Twitter”.

[3] Goldstein-Stewart J, Winder R, Sabin RE (2009) Person identification from text and speech genre samples. In: Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics. Association for Computational Linguistics, pp 336–344

[4] Francisco Rangel, Fabio Celli, Paolo Rosso, Martin Potthast, Benno Stein, Walter Daelemans “Overview of the 3rd Author Profiling Task at PAN 2015”.

[5] Vallikannu Ramanathan, T. Meyyappan "Survey of Text Mining", International Conference on Technology and Business and Management, March 2013, pp. 508-514.

[6] Vishal Gupta and Gurpreet S. Lehal, A Survey of Text Mining Techniques and Applications, JOURNAL OF EMERGING TECHNOLOGIES IN WEB INTELLIGENCE, VOL. 1, NO. 1, AUGUST 2009.

[7] R.Sagayam, S.Srinivasan, S.Roshini, “A Survey of Text Mining: Retrieval, Extraction and Indexing Techniques”. Internaltional Journal of Computational Engineering Research (ijceronline.com) Vol.2 Issue.5

[8] Mr. Rahul Patel, Mr. Gaurav Sharma, "A survey on text mining techniques", International Journal Of Engineering And Computer Science ISSN:2319- 7242, Vol 3 Issue 5, May 2014, pp.5621-5625

[9] Rashmi Agrawal, Mridula Batra, "A Detailed Study on Text Mining Techniques", IJSCE, ISSN: 2231- 2307, Vol. 2, Issue-6, January 2013.

[10] Dunham Margaret H., “Data Mining Introductory and Andanced Topics”, Pearson Education Inc, (2003).

[11] Raghavan, V. V. and Wong, S. K. M. A critical analysis of vector space model for information retrieval. Journal of the American Society for Information Science, Vol.37 (5), p. 279-87, 1986

[12] Salton, Gerard. Introduction to Modern Information Retrieval. McGraw-Hill, 1983.

[13] Luhn, H. P. The Automatic Creation of Literature Abstracts. IBM Journal of Research and Development 2 (2), p. 159-165 and 317, April 1958.

[14] Van Rijsbergen, C. J. Information retrieval. Butterworths, 1979

[15] Salton, G. and Buckley, C. (1988) “Term weighting approaches In automatic text retrieval, Information Processing and Management”,Vol. 24, No.5, Pp. 513 - 523.

[16] Manning, C.D.; Raghavan, P.; Schutze, H. (2008). "Scoring, term weighting, and the vector space model" (PDF). *Introduction to Information Retrieval*. p. 100. doi:10.1017/CBO9780511809071.007. ISBN 978-0-511-80907-1.

[17] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze.“Introduction to Information Retrieval”, Cambridge University Press, New York, NY, USA, 2008.

[18] C. C. Aggarwal and C.-X. Zhai, “Mining Text Data”, New York, NY, USA: Springer, 2012.

[19] Porter, M. (1980) “An algorithm for suffix stripping, Program”,Vol. 14, No. 3, Pp. 130–137

[20] http://www.britannica.com/EBchecked/topic/1116194/machine-learning

[21] Phil Simon (March 18, 2013). Too Big to Ignore: The Business Case for Big Data. Wiley, σελ. 89. ISBN 978-1-118-63817-0

[22] Ron Kohavi; Foster Provost (1998). «Glossary of terms». Machine Learning 30: 271–274.

[23] Machine learning and pattern recognition "can be viewed as two facets of the same field."

[24] Mitchell, T. (1997). *Machine Learning*, McGraw Hill, *Machine Learning*, McGraw Hill, p.2

[25] Harnad, Stevan (2008), «The Annotation Game: On Turing (1950) on Computing, Machinery, and Intelligence», στο: Epstein, Robert; Peters, Grace, επιμ., *The Turing Test Sourcebook: Philosophical and Methodological Issues in the Quest for the Thinking Computer*, Kluwer

[26] **Σφάλμα! Η αναφορά της υπερ-σύνδεσης δεν είναι έγκυρη.**. *www.sas.com*. Ανακτήθηκε στις 29 Μαρτίου 2016.

[27] Russell, Stuart· Norvig, Peter (2003). *Artificial Intelligence: A Modern Approach* (2nd έκδοση). Prentice Hall. ISBN 978-0137903955.

[28] Mehryar Mohri, Afshin Rostamizadeh, Ameet Talwalkar (2012) Foundations of Machine Learning, The MIT Press ISBN 9780262018258

[29] https://repository.kallipos.gr/bitstream/11419/3382/1/02_chapter_04.pdf

[30] http://blogs.sas.com/content/sascom/2015/08/11/an-introduction-to-machine-learning/

[31] Chris Manning and Hinrich Schutze “Foundations of Statistical Natural Language Processing”, MIT Press. Cambridge, MA: May 1999.

[32] Sebastiani F. “Machine learning in automated text categorization”, computing surveys (CSUR), 34(1), 1-47, 2002.

[33] Koppel M., Schler J. & Argamon S. “Computational Methods in Authorship Attribution”, Journal of the American Society for Information Science and Technology, 60(1), 9-16, 2008.

[34] Vandana Korde, C. Namrata Mahender, “Text Classification and classifiers: a survey”, International Journal of Artificial Intelligence & Applications (IJAIA), Vol.3, No.2, 2012.

[35] Ng, A.Y. and Jordan, M.I. “discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes”. Neural Information Processing Systems, 14, 841, 2002.

[36] Joachims T. “Text categorization with Support Vector Machines: Learning with many relevant features”, Machine Learning: ECML1998.

[37] https://en.wikipedia.org/wiki/Multi-labelclassification

[38] Collobert - Weston - Bottou - Karlen - Kavukcuoglu και Kuksa. Natural Language Processing (Almost) from Scratch, Journal of Machine Learning Research 12, σελίδες: 2493 - 2537, 2011.

[39] Κωτσιαντής Σωτήριος, Διδακτορική Διατριβή, Ομάδες Ταξινομητών για την αύξηση της ακρίβειας των μεθόδων μηχανικής μάθησης και εξόρυξης γνώσης, 2005.

[40] Pang-Ning Tan, Michael Steinbach, Vipin Kumar, Εισαγωγή στην εξόρυξη δεδομένων.

[41] Wagner Meira Jr. Mohammed J.Zaki, Εξόρυξη και ανάλυση δεδομένων, βασικές έννοιες και αλγόριθμοι.

[42] Vapnik, V.: Statistical Learning Theory. John Wiley, New York (1998).

[43] Παναγιώτης Καραφώτης, Αναγνώριση μη γλωσσολογικών ήχων από ομιλία, 2017

[44] Platt, John (1998), Sequential Minimal Optimization: A Fast Algorithm for Training Support Vector Machines, CiteSeerX 10.1.1.43.4376

[45] Chang, Chih-Chung; Lin, Chih-Jen (2011). "LIBSVM: A library for support vector machines". *ACM Transactions on Intelligent Systems and Technology*. **2** (3).

[46] Zanni, Luca (2006). "Parallel Software for Training Large Scale Support Vector Machines on Multiprocessor Systems"

[47] Daniel Jurafsky, James H. Martin, “Speech and Language Processing”

[48] https://pan.webis.de/

[49] Stamatatos E., Fakotakis N. & Kokkinakis G., “Automatic text categorization in terms of genre and author”. Computational Linguistics, 26(4), 471–495, 2000.

[50] Francisco Rangel, Paolo Rosso, Moshe Koppel, Efstathios Stamatatos, Giacomo Inches “Overview of the Author Profiling Task at PAN 2013”.

[51] Francisco Rangel, Paolo Rosso, Irina Chugur, Martin Potthast, Martin Trenkmann, Benno Stein, Ben Verhoeven, Walter Daelemans. “Overview of the 2nd Author Profiling Task at PAN 2014”.

[52] Angelo Basile, Gareth Dwyer, Maria Medvedeva, Josine Rawee, Hessel Haagsma, and Malvina Nissim, “N-GrAM: New Groningen Author-profiling Model”, Notebook for PAN at CLEF 2017.

[53] Mart Busger op Vollenbroek, Talvany Carlotto, Tim Kreutz, Maria Medvedeva, Chris Pool, Johannes Bjerva, Hessel Haagsma, and Malvina Nissim, “GronUP: Groningen User Profiling” Notebook for PAN at CLEF 2016.

[54] Sebastian Sierra, Manuel Montes y Gómez, Thamar Solorio, and Fabio A. González. Convolutional neural networks for author profiling in pan 2017. In Cappellato et al. [13].

Παράρτημα I (Κώδικας I)

```
import argparse
import csv
import random
from pathlib import Path

"Create a csv file from CMCC-Corpus file."
#col1: counter
#col2: author id
#col3: text
#col4: filename
#col5: author
#col6: genre
#col7: genre_number
#col8: topic
#col9: topic_number

CORPUS = {
    'CMCC': ['counter',
             'author_id',
             'text',
             'filename',
             'author',
             'genre',
             'genre_number',
             'topic',
             'topic_number']
    }
```

```
def cmcc(corpus_path):
    i = 0
    for txtfile in corpus_path.glob('**/*.txt'):
        with txtfile.open(encoding='ISO-8859-1') as fr:
            filename = txtfile.name
            i += 1
            author = filename[:3] if filename[2].isdigit() else filename[:2]
            ret = []
            ret.append(i)
            ret.append(author[1:])  # author_id
            ret.append(fr.read().replace('\n', ' '))  # text
            ret.append(filename) # filename
            ret.append(author)  # author
            ret.append(filename[-8])  # genre
            ret.append(filename[-7])  # genre_number
            ret.append(filename[-6])  # topic
            ret.append(filename[-5])  # topic_number
            yield ret


def create_csv(corpus, corpus_path, output):
    with output.open('w', newline='', encoding='ISO-8859-1') as fw:
        writer = csv.writer(fw)
        writer.writerow(CORPUS[corpus])
        for instance in get_generator(corpus, corpus_path):
            writer.writerow(instance)


    print('')
    print("Created '%s' from '%s' folder." % (output, corpus_path))
    print("Header: '%s'." % ','.join(CORPUS[corpus]))
    print('Done!')
```

```
    print('')

def get_generator(corpus, corpus_path):
    cmcc = corpus.lower().replace('-', '_')
    generator = globals()[cmcc](corpus_path)
    return generator

def main():
    parser = argparse.ArgumentParser()
    parser.add_argument('corpus', choices=CORPUS.keys(), help="Choose CMCC.")
    parser.add_argument('corpus_path', help="Where are the data?")
    parser.add_argument('csv_filename', help="The filename of the output csv.")
    args = parser.parse_args()
    args.corpus_path = Path(args.corpus_path).resolve()
    args.csv_filename = Path(args.csv_filename).with_suffix('.csv')
    create_csv(args.corpus, args.corpus_path, args.csv_filename)

if __name__ == '__main__':
    main()
```

Παράρτημα I (Κώδικας II)

```
import argparse
import nltk
import pandas as pd
from collections import defaultdict
from collections import Counter
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn import preprocessing
from sklearn.svm import LinearSVC, SVC
from sklearn.naive_bayes import MultinomialNB
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score, confusion_matrix, classification_report
from nltk import word_tokenize, pos_tag
from matplotlib import pyplot as plt
from statistics import mean

nltk.download('averaged_perceptron_tagger')
nltk.download('tagsets')
nltk.download('punkt')
pd.options.mode.chained_assignment = None

topics_full_names = {
    'C': 'Church',
    'G': 'Gay Marriage',
    'I': 'Iraq War',
    'M': 'Marijuana',
    'P': 'Privacy',
    'S': 'Sex Discrimination'
```

```
}

genres_full_names = {
    'B': 'Blog',
    'C': 'Chat',
    'D': 'Discussion',
    'E': 'Email',
    'S': 'Essay',
    'P': 'Phone Interview'
}

classifiers = {
    'Linear SVM': LinearSVC(C = 10, max_iter = 10000),
    'Poly SVM': SVC(kernel = 'poly'),
    'Decision Tree': DecisionTreeClassifier(),
    'Naive Bayes': MultinomialNB(),
    'Random Forest': RandomForestClassifier()
}

def generate_word_n_grams(text, n):
    "Extracts all word n-grams from a text."
    words = word_tokenize(text)
    temp = zip(*[words[i:] for i in range(0, n)])
    ngrams = [' '.join(ngram) for ngram in temp]
    frequency = defaultdict(int)
    for token in ngrams:
        frequency[token] += 1
    return frequency

def generate_char_n_grams(text, n):
```

```
    "Extracts all character n-grams from a text."
    if n > 0:
        tokens = [text[i: i + n] for i in range(len(text) - n + 1)]
    frequency = defaultdict(int)
    for token in tokens:
        frequency[token] += 1
    return frequency


def generate_pos_n_grams(text, n):
    "Extracts all POS n-grams from a text."
    tokens = word_tokenize(text)
    tagged_tokens = pos_tag(tokens)
    pos_tagged_tokens = [token[1] for token in tagged_tokens]
    temp = zip(*[pos_tagged_tokens[i:] for i in range(0, n)])
    ngrams = [' '.join(ngram) for ngram in temp]
    frequency = defaultdict(int)
    for token in ngrams:
        frequency[token] += 1
    return frequency


def represent_texts_as_pos_tags(texts):
    "Converts text into word tokens and then applies pos tagging so that each text is
    represented as a sequence of pos tags."
    pos_tagged_texts = []
    for text in texts:
        tokens = word_tokenize(text)
        pos_tagged_text = pos_tag(tokens)
        only_tags = [tagged_token[1] for tagged_token in pos_tagged_text]
        pos_tagged_texts.append(' '.join(only_tags))
    return pos_tagged_texts
```

```
def extract_vocabulary(texts, n, option, n_most_frequent):
    "Calls the right function to generate the vocabulary for each model. Extracts all n-grams in the vocabulary. Calculates each vocabulary term's frequency and only keeps the n_most_frequent terms (used in some experiments)."
    occurrences = defaultdict(int)
    for text in texts:
        if option == 'char':
            text_occurrences = generate_char_n_grams(text, n)
        elif option == 'word':
            text_occurrences = generate_word_n_grams(text, n)
        elif option == 'pos':
            text_occurrences = generate_pos_n_grams(text, n)
        for ngram in text_occurrences:
            if ngram in occurrences:
                occurrences[ngram] += text_occurrences[ngram]
            else:
                occurrences[ngram] = text_occurrences[ngram]
    vocabulary = []
    if n_most_frequent is None:
        # keep all vocabulary
        for i in occurrences.keys():
            vocabulary.append(i)
    else:
        # pick most frequent tokens
        c = Counter(occurrences)
        most_frequent = (c.most_common(n_most_frequent))
        for k in range(len(most_frequent)):
            vocabulary.append(most_frequent[k][0])
    return vocabulary


def create_labels(dataset):
```

```
    "Appends a gender column to the dataset that is used as a label for the model
    training."
    dataset['gender'] = ['M' if a in [2, 4, 6, 8, 10, 12, 14, 16, 18, 20]
    else 'F' for a in dataset['author_id'].values]
    return dataset


def split_dataset(dataset):
    "Splits dataset into train and test set."
    train_set=dataset[dataset['author_id'] <= 11]
    test_set=dataset[dataset['author_id'] >= 12]
    return train_set, test_set


def split_dataset_into_topics(dataset):
    "Splits dataset into an array of datasets based on the text's topic."
    c = dataset[dataset['topic'] == 'C']
    g = dataset[dataset['topic'] == 'G']
    i = dataset[dataset['topic'] == 'I']
    m = dataset[dataset['topic'] == 'M']
    p = dataset[dataset['topic'] == 'P']
    s = dataset[dataset['topic'] == 'S']
    return [c, g, i, m, p, s]


def split_dataset_into_genres(dataset):
    "Splits dataset into an array of datasets based on the texts' genre."
    blog = dataset[dataset['genre'] == 'B']
    chat = dataset[dataset['genre'] == 'C']
    discussion = dataset[dataset['genre'] == 'D']
    email = dataset[dataset['genre'] == 'E']
    essay = dataset[dataset['genre'] == 'S']
    interview = dataset[dataset['genre'] == 'P']
    return [blog, chat, discussion, email, essay, interview]
```

```
def prepare_data(train_set, test_set, topic_to_leave_out, option, n, n_most_frequent = None):
    "Apply data modifications to prepare them to be fed into the classification model."
    # keep the selected train and test topics
    selected_train_topics = train_set[train_set['topic'] != topic_to_leave_out]
    selected_test_topic = test_set[test_set['topic'] == topic_to_leave_out]
    # get texts vocabulary: tokens appearing
    vocabulary = extract_vocabulary(selected_train_topics['text'], n, option, n_most_frequent)
    # text vectorizer
    if option == 'word' or option == 'char':
        vectorizer = TfidfVectorizer(analyzer = option, ngram_range = (n, n), lowercase = False, vocabulary = vocabulary)
    elif option == 'pos':
        vectorizer = TfidfVectorizer(analyzer = 'word', ngram_range = (n, n), lowercase =False, vocabulary = vocabulary)
    # prepare train and test data
    if option == 'pos':
        # represent texts as pos tags so that they can be fitted into the vectorizer
        pos_tagged_train_texts=
        represent_texts_as_pos_tags(selected_train_topics['text'].values)
        pos_tagged_test_texts =
        represent_texts_as_pos_tags(selected_test_topic['text'].values)
        train_data = vectorizer.fit_transform(pos_tagged_train_texts)
        test_data = vectorizer.transform(pos_tagged_test_texts)
    else:
        # fit vectorizer in text as it is
        train_data = vectorizer.fit_transform(selected_train_topics['text'].values)
        test_data = vectorizer.transform(selected_test_topic['text'].values)
    train_data = train_data.astype(float)
    test_data = test_data.astype(float)
    # scale data
```

```
    max_abs_scaler = preprocessing.MaxAbsScaler()
    scaled_train_data = max_abs_scaler.fit_transform(train_data)
    scaled_test_data = max_abs_scaler.transform(test_data)
    return scaled_train_data, scaled_test_data, selected_train_topics,
selected_test_topic


def implement_classification(train_data, test_data, selected_train_topics,
selected_test_topic):
    "Implements all classification models on texts from a certain topic
    to predict the author's gender and calculates the model's accuracy."
    topic_accuracies = []
    for model_name, model in classifiers.items():
        clf = model.fit(train_data, selected_train_topics['gender'])
        topic_accuracies.append(clf.score(test_data, selected_test_topic['gender'].values)
* 100)
        # metrics
        results = clf.predict(test_data)
        print(model_name)
        predicted_gender = results
        print('Predicted gender: ', predicted_gender)
        actual_gender = selected_test_topic['gender'].values
        print('Actual gender: ', actual_gender)
        print('Confusion matrix:')
        print(confusion_matrix(selected_test_topic['gender'].values, results, labels = ['M',
'F']))
        print("Classification report:")
        print(classification_report(selected_test_topic['gender'].values, results, labels =
['M', 'F'], zero_division = 1))
    return topic_accuracies, clf


def implement_loocv(train_genres, test_genres, option, n):
```

```
    "Implements leave-one-topic-out cross validation for all topics of each genre
    and for all classification models and calls plotting functions for visualising
    accuracy results."
    topics = ['C', 'G', 'I', 'M', 'P', 'S']
    all_genres_accuracies = []
    macro_average = []
    for i in range(len(train_genres)):
        all_topics_accuracies = []
        genre_name = str(genres_full_names.get(train_genres[i]['genre'].values[0]))
        for k, topic in enumerate(topics):
            scaled_train_data, scaled_test_data, selected_train_topics, selected_test_topic
= prepare_data(train_genres[i], test_genres[i], topic, option, n)
            results, clf = implement_classification(scaled_train_data, scaled_test_data,
selected_train_topics, selected_test_topic)
            all_topics_accuracies.append(results)
            all_genres_accuracies.append(results)
        plot_detailed_results(all_topics_accuracies, genre_name, n, option)
        plot_average_per_genre(all_topics_accuracies, genre_name, n, option,
macro_average)
    plot_micro_average_per_ngram(all_genres_accuracies, n, option)
    plot_macro_average_per_ngram(macro_average, n, option)


def test_all_n_most_frequent(train_genres, test_genres, option, n, most_frequent):
    topics = ['C', 'G', 'I', 'M', 'P', 'S']
    accuracies_for_n_most_frequent = []
    for n_most_frequent in most_frequent:
        all_accuracies = []
        for i in range(len(train_genres)):
            for k, topic in enumerate(topics):
                scaled_train_data, scaled_test_data, selected_train_topics,
selected_test_topic =
prepare_data(train_genres[i], test_genres[i], topic, option, n, n_most_frequent)
```

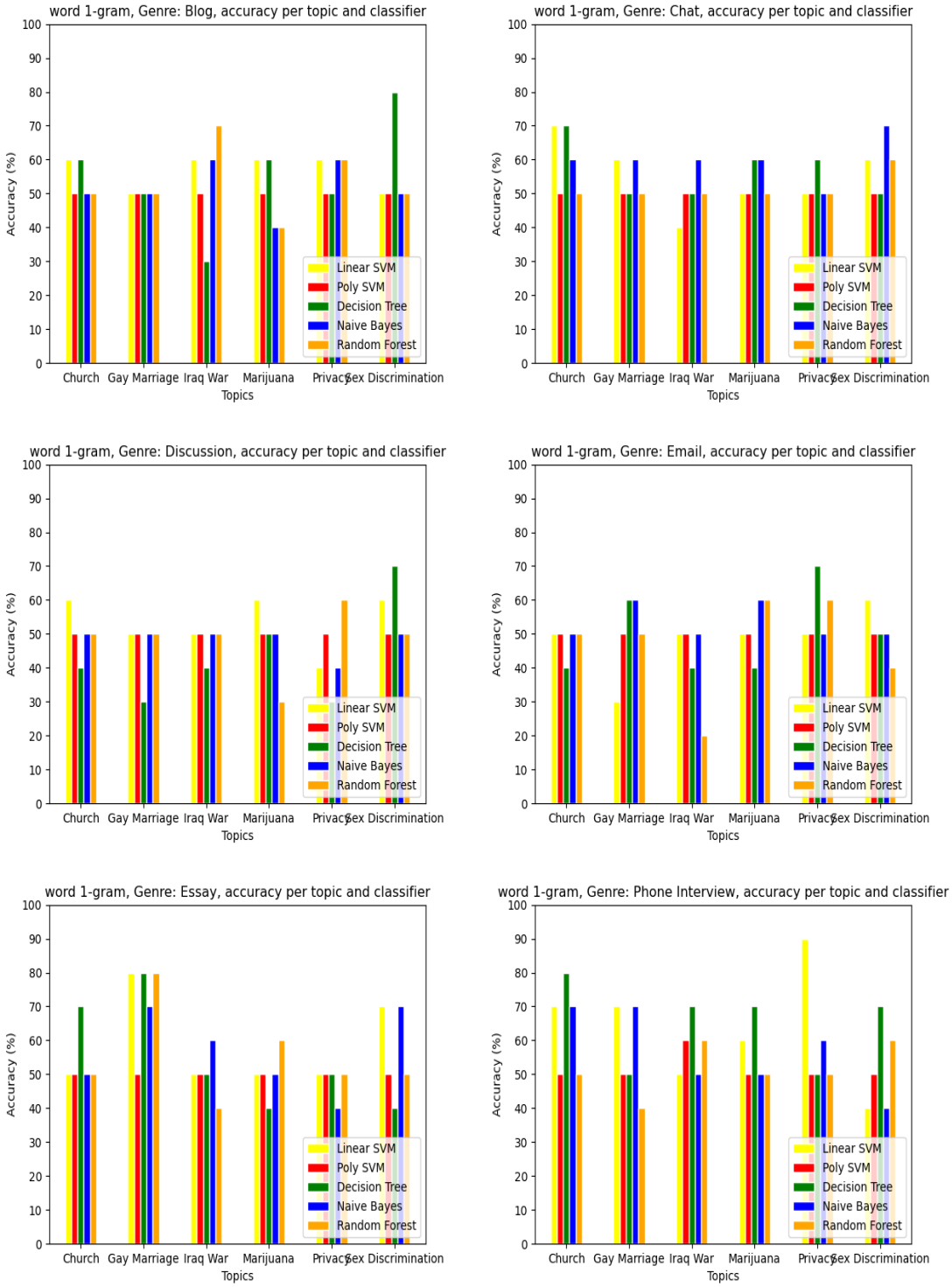
```
            clf = LinearSVC(C = 10).fit(scaled_train_data,
selected_train_topics['gender'])

            all_accuracies.append(clf.score(scaled_test_data,
selected_test_topic['gender'].values) * 100)

        accuracies_for_n_most_frequent.append(mean(all_accuracies))

    return accuracies_for_n_most_frequent


def plot_detailed_results(accuracies, genre_name, n, option):

    "Gets all topics/classifiers accuracy results for each genre and generates bar charts.
    Uses a dataframe to group each classifier's results per topic, so that it's easier to plot
    in bars and visualise."

    accuracies_per_topic = {'Linear SVM': [accuracies[i][0] for i in range(6)],

                            'Poly SVM': [accuracies[i][1] for i in range(6)],

                            'Decision Tree': [accuracies[i][2] for i in range(6)],

                            'Naive Bayes': [accuracies[i][3] for i in range(6)],

                            'Random Forest': [accuracies[i][4] for i in range(6)]}

    df = pd.DataFrame(accuracies_per_topic, index = topics_full_names.values())

    df.plot(kind = 'bar', edgecolor = 'white', linewidth = 1, rot = 0, xlabel = 'Topics',
ylabel = 'Accuracy (%)', yticks = ([w * 10 for w in range(11)]), ylim = (0, 100), color
= ['yellow','red','green','blue','orange'])

    plt.legend(loc = 'lower right')

    plt.title(str(option) + ' ' + str(n) + '-gram, Genre: ' + genre_name + ', accuracy per
topic and classifier')

    plt.savefig('Answers/Genre_' + genre_name + '_' + option + '_' + str(n) +
'_gram.png')

    plt.show()


def plot_average_per_genre(accuracies, genre_name, n, option, macro_average):

    "Gets the mean accuracy for all 6 topics in a genre, for 5 different classification
    models."

    mean_accuracy_per_model = {'Linear SVM': mean([accuracies[i][0] for i in
range(6)]),

                               'Poly SVM': mean([accuracies[i][1] for i in range(6)]),

                               'Decision Tree': mean([accuracies[i][2] for i in range(6)]),
```

```
                'Naive Bayes': mean([accuracies[i][3] for i in range(6)]),
                'Random Forest': mean([accuracies[i][4] for i in range(6)])}
    macro_average.append(list(mean_accuracy_per_model.values()))
    plt.bar(mean_accuracy_per_model.keys(), mean_accuracy_per_model.values(),
color = ['yellow','red','green','blue','orange'])
    plt.ylabel('Accuracy (%)')
    plt.xlabel('Classifier Models')
    plt.yticks([w * 10 for w in range(11)])
    plt.title(str(option) + ' ' + str(n) + '-gram, Genre: ' + genre_name + ', Mean accuracy
for all topics per classifier')
    plt.savefig('Answers/Genre_' + genre_name + '_' + option + '_' + str(n) +
'_gram_macro_average.png')
    plt.show()


def plot_macro_average_per_ngram(accuracies, n, option):
    "Plots the macro average of the accuracy for 5 different classifiers for all instances
    of an n-gram model."
    macro_avg_accuracy_per_model = {'Linear SVM': mean([accuracies[i][0] for i in
range(6)]),
                'Poly SVM': mean([accuracies[i][1] for i in range(6)]),
                'Decision Tree': mean([accuracies[i][2] for i in range(6)]),
                'Naive Bayes': mean([accuracies[i][3] for i in range(6)]),
                'Random Forest': mean([accuracies[i][4] for i in range(6)])}
    plt.bar(macro_avg_accuracy_per_model.keys(),
macro_avg_accuracy_per_model.values(), color =
['yellow','red','green','blue','orange'])
    plt.ylabel('Accuracy (%)')
    plt.xlabel('Classifier Models')
    plt.yticks([w * 10 for w in range(11)])
    plt.title(option + '_' + str(n) + ': Macro-average accuracy per classifier')
    plt.savefig('Answers/'+ option + '_' + str(n) + 'macro.png')
    plt.show()
def plot_micro_average_per_ngram(micro_average, n, option):
```

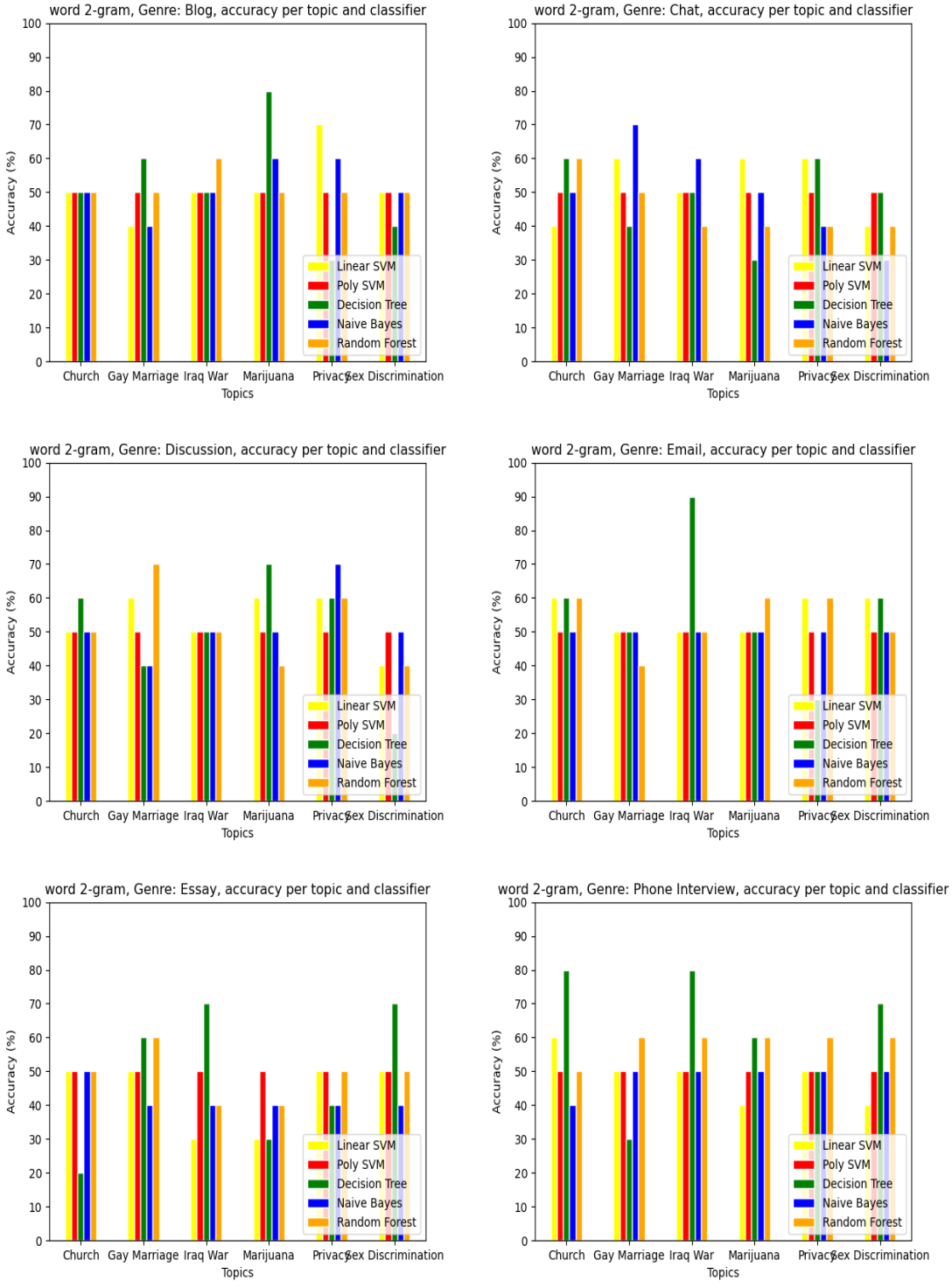
```
    "Plots the micro average of the accuracy for 5 different classifiers for all instances
    of an n-gram model."

    micro_average_classifier1 = [micro_average[i][0] for i in range(36)]

    micro_average_classifier2 = [micro_average[i][1] for i in range(36)]

    micro_average_classifier3 = [micro_average[i][2] for i in range(36)]

    micro_average_classifier4 = [micro_average[i][3] for i in range(36)]

    micro_average_classifier5 = [micro_average[i][4] for i in range(36)]

    micro_average_per_classifier = [mean(micro_average_classifier1),
mean(micro_average_classifier2),
                                    mean(micro_average_classifier3),
mean(micro_average_classifier4), mean(micro_average_classifier5)]

    plt.bar(classifiers.keys(), micro_average_per_classifier, color =
['yellow','red','green','blue','orange'])

    plt.ylabel('Accuracy (%)')

    plt.xlabel('Classifier Models')

    plt.yticks([w * 10 for w in range(11)])

    plt.title(option + '_' + str(n) + ': Micro-average accuracy per classifier')

    plt.savefig('Answers/'+ option + '_' + str(n) + 'micro.png')

    plt.show()


def plot_results_for_all_n_most_frequent(train_genres, test_genres, option, n_range):

    most_frequent = [500, 1000, 2000, 3000]

    results_for_all_n = []

    for n in n_range:

        results_for_all_n.append(test_all_n_most_frequent(train_genres, test_genres,
option, n, most_frequent))

    plt.figure(figsize = (12, 8))

    plt.plot(most_frequent, results_for_all_n[0])

    plt.plot(most_frequent, results_for_all_n[1])

    plt.plot(most_frequent, results_for_all_n[2])

    plt.legend(['n=' + str(n_range[0]), 'n=' + str(n_range[1]), 'n=' + str(n_range[2])])
```

```
    plt.title('Average accuracy for ' + option + ' n-gram model for most frequent terms from vocabulary')
    plt.ylabel('Accuracy (%)')
    plt.xlabel('most frequent terms used')
    plt.xticks(most_frequent)
    plt.savefig('Answers/most_frequent '+ option +' n-gram.jpg')
    plt.show()


def main():
    # get input parameters
    parser = argparse.ArgumentParser(description = 'CMCC-Corpus')
    parser.add_argument('-i', type = str, help = 'Absolute Path to the dataset csv file')
    args = parser.parse_args()
    if not args.i:
        print('ERROR: The input folder is required')
        parser.exit(1)
    dataset = pd.read_csv(args.i, encoding = 'ISO-8859-1')
    dataset = create_labels(dataset)
    train_set, test_set = split_dataset(dataset)
    train_genres = split_dataset_into_genres(train_set)
    test_genres = split_dataset_into_genres(test_set)
    for i in range(1, 4):
        implement_loocv(train_genres, test_genres, 'word', i)
        implement_loocv(train_genres, test_genres, 'char', i + 2)
        implement_loocv(train_genres, test_genres, 'pos', i)
    plot_results_for_all_n_most_frequent(train_genres, test_genres, 'word', [1, 2, 3])
    plot_results_for_all_n_most_frequent(train_genres, test_genres, 'char', [3, 4, 5])
    plot_results_for_all_n_most_frequent(train_genres, test_genres, 'pos', [1, 2, 3])


if __name__ == '__main__':
    main()
```

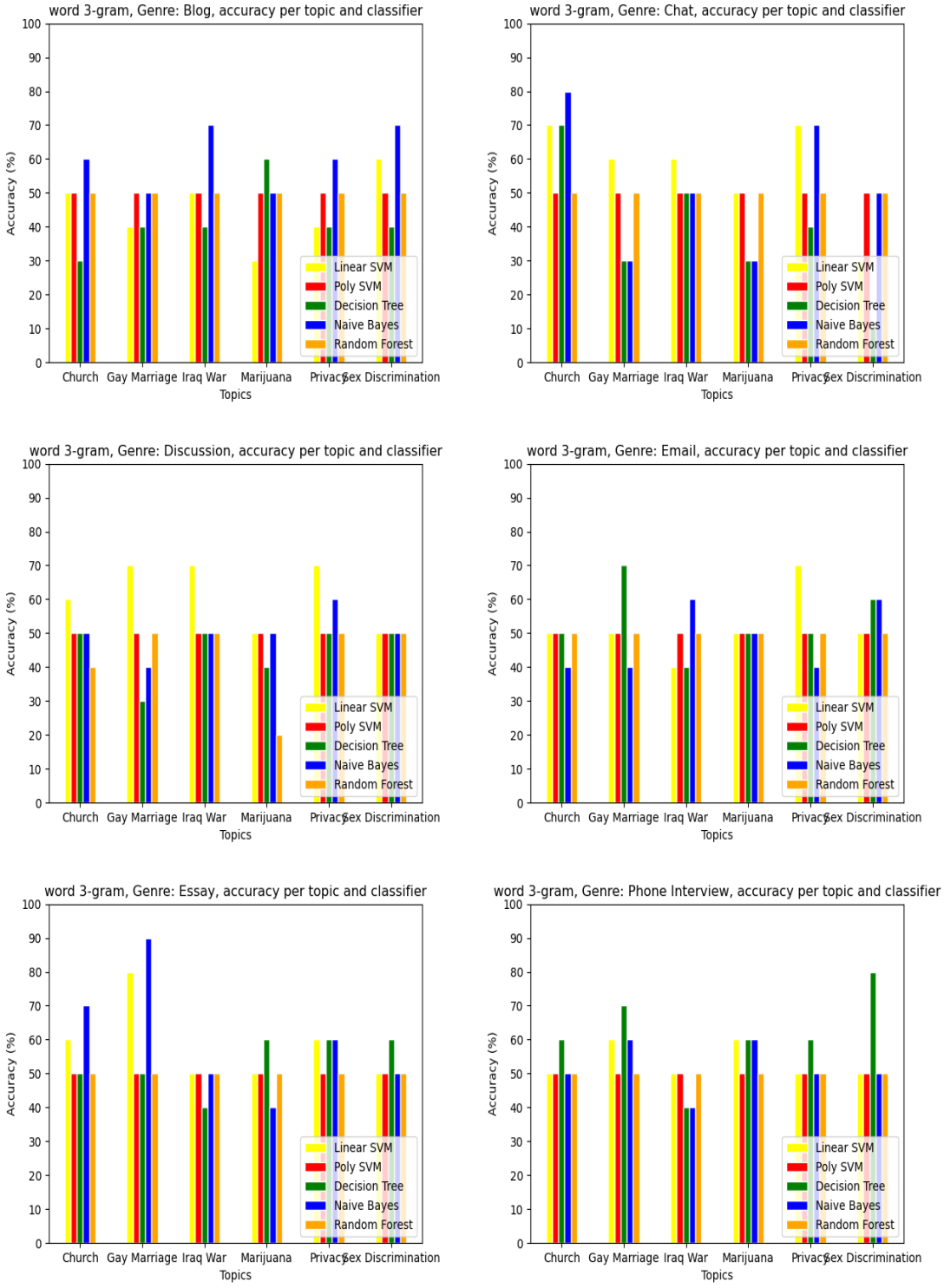
Παράρτημα ΙΙ (Διαγράμματα Accuracy ανά Topic)



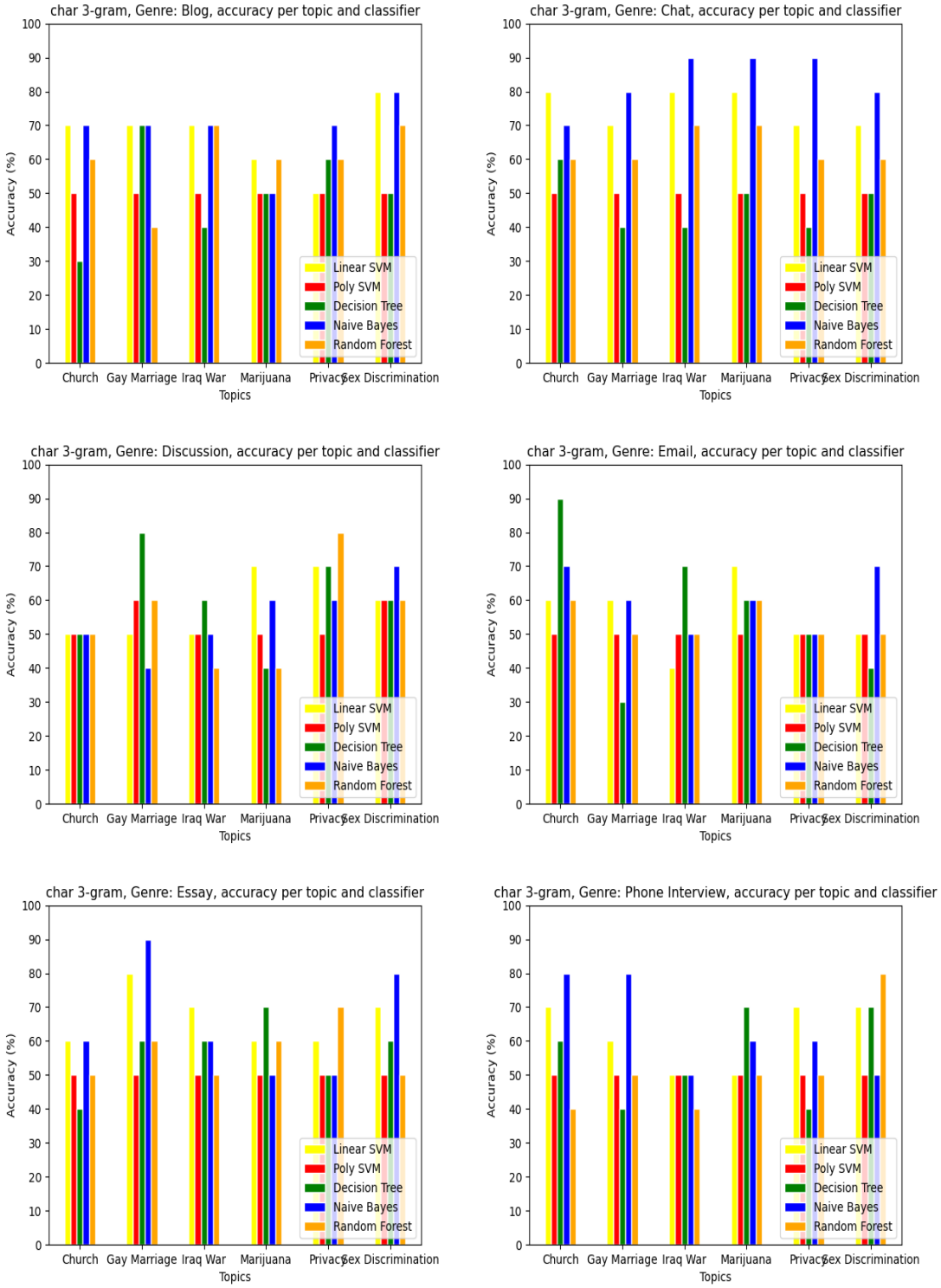
Εικόνα 44: Accuracy ανά Topic για Word N – Gram, N = 1



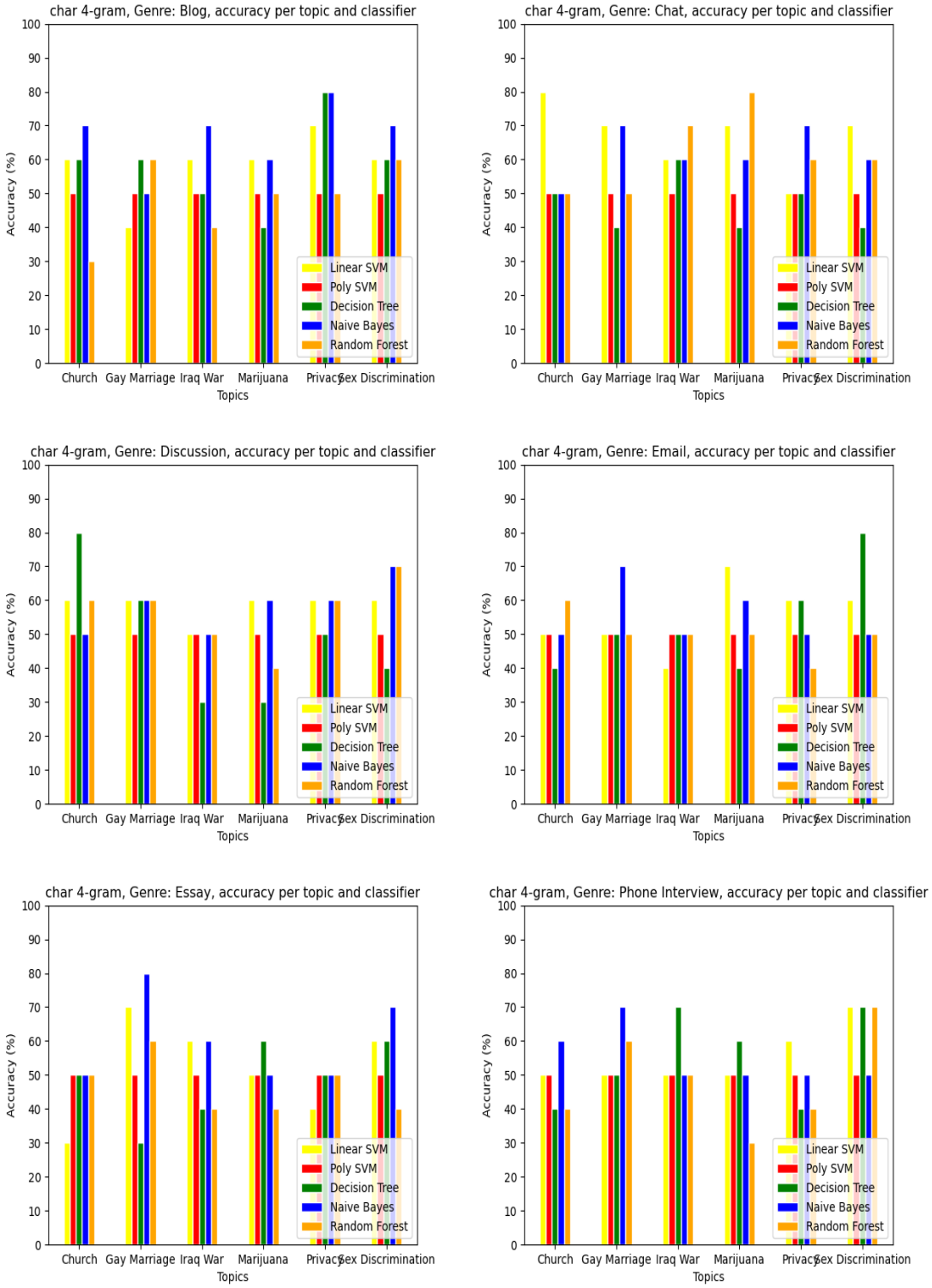
Εικόνα 45: Accuracy ανά Topic για Word N – Gram, N = 2



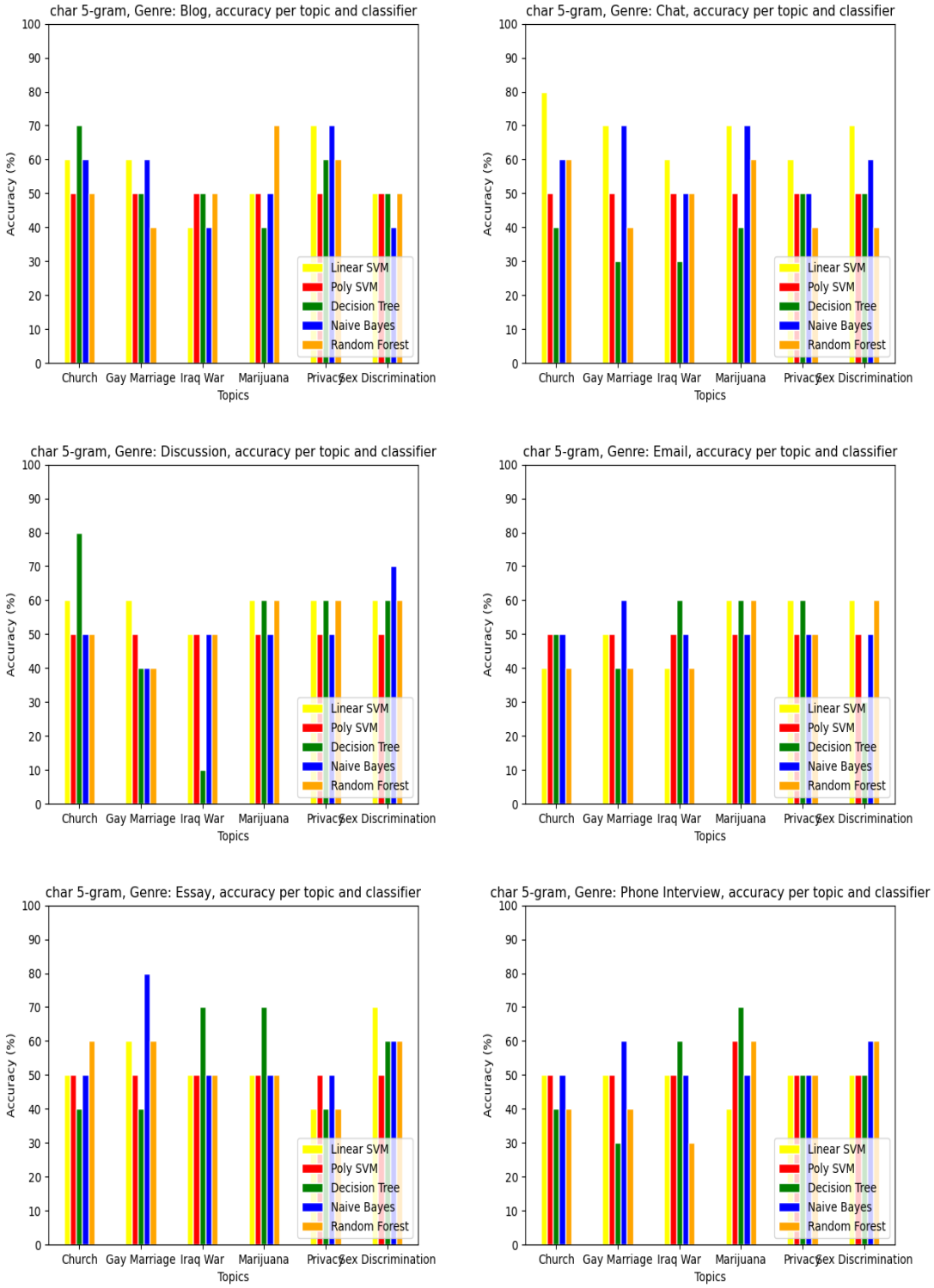
Εικόνα 46: Accuracy ανά Topic για Word N – Gram, N = 3



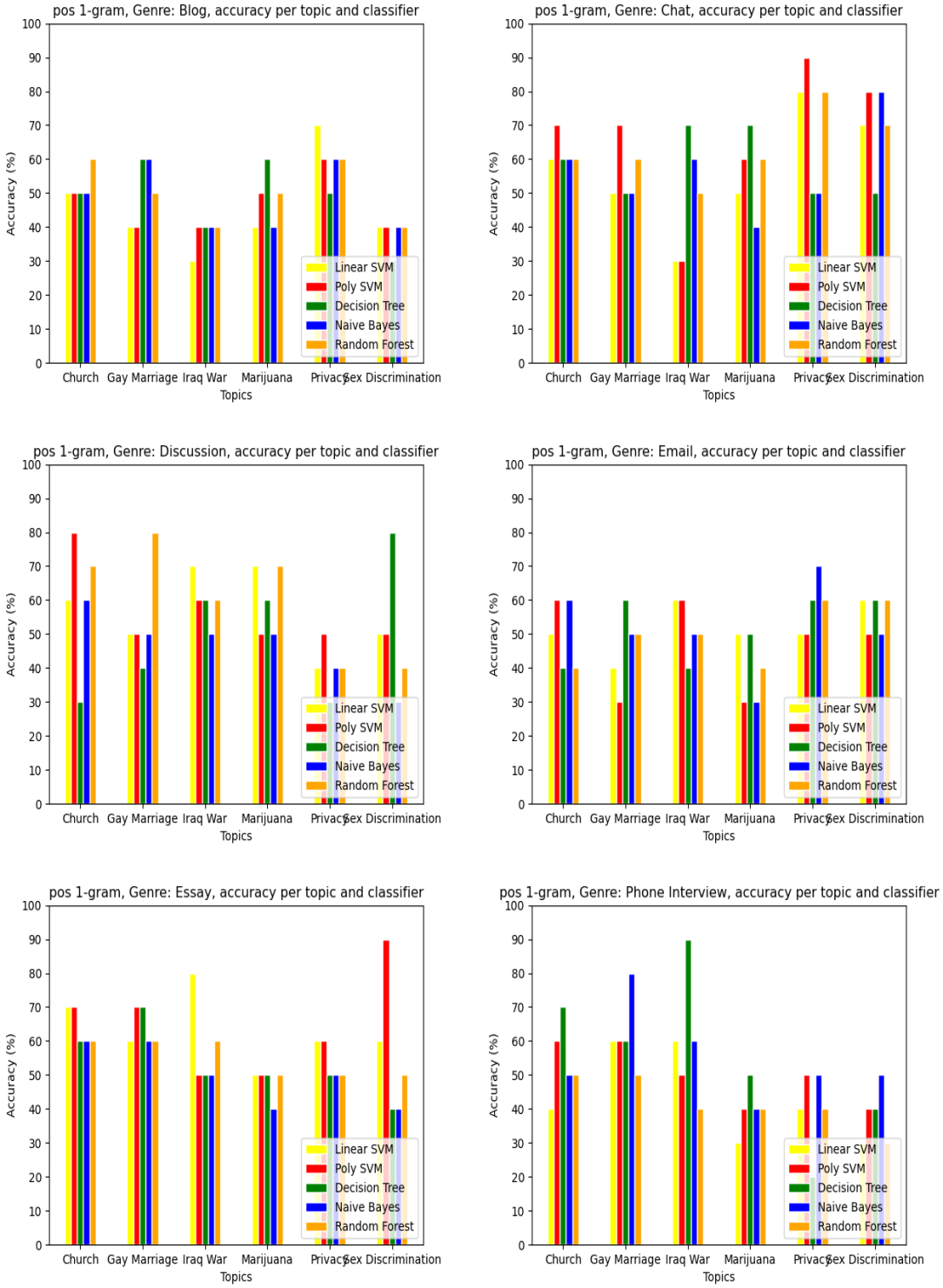
Εικόνα 47: Accuracy ανά Topic για Character N – Gram, N = 3



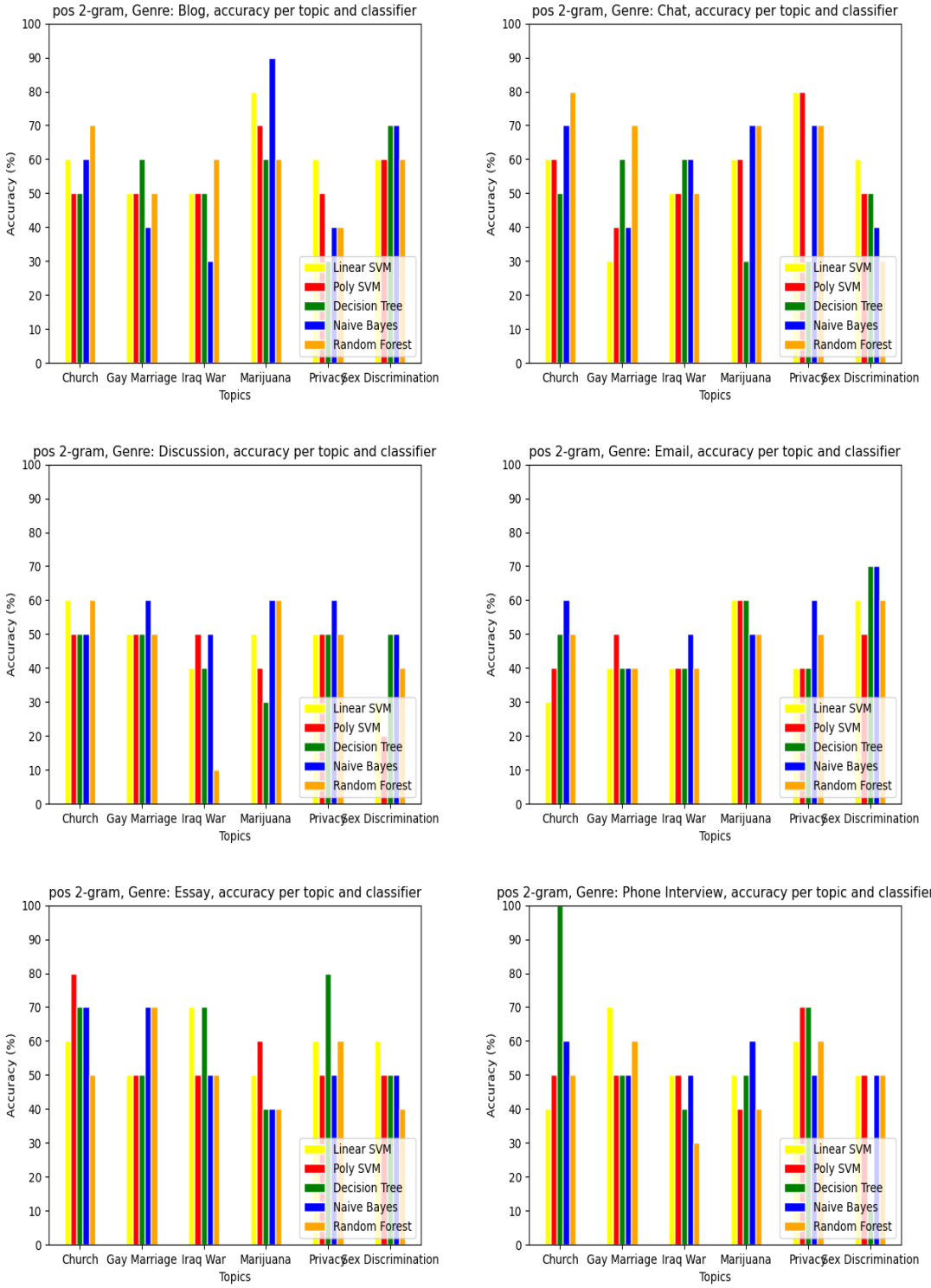
Εικόνα 48: Accuracy ανά Topic για Character N – Gram, N = 4



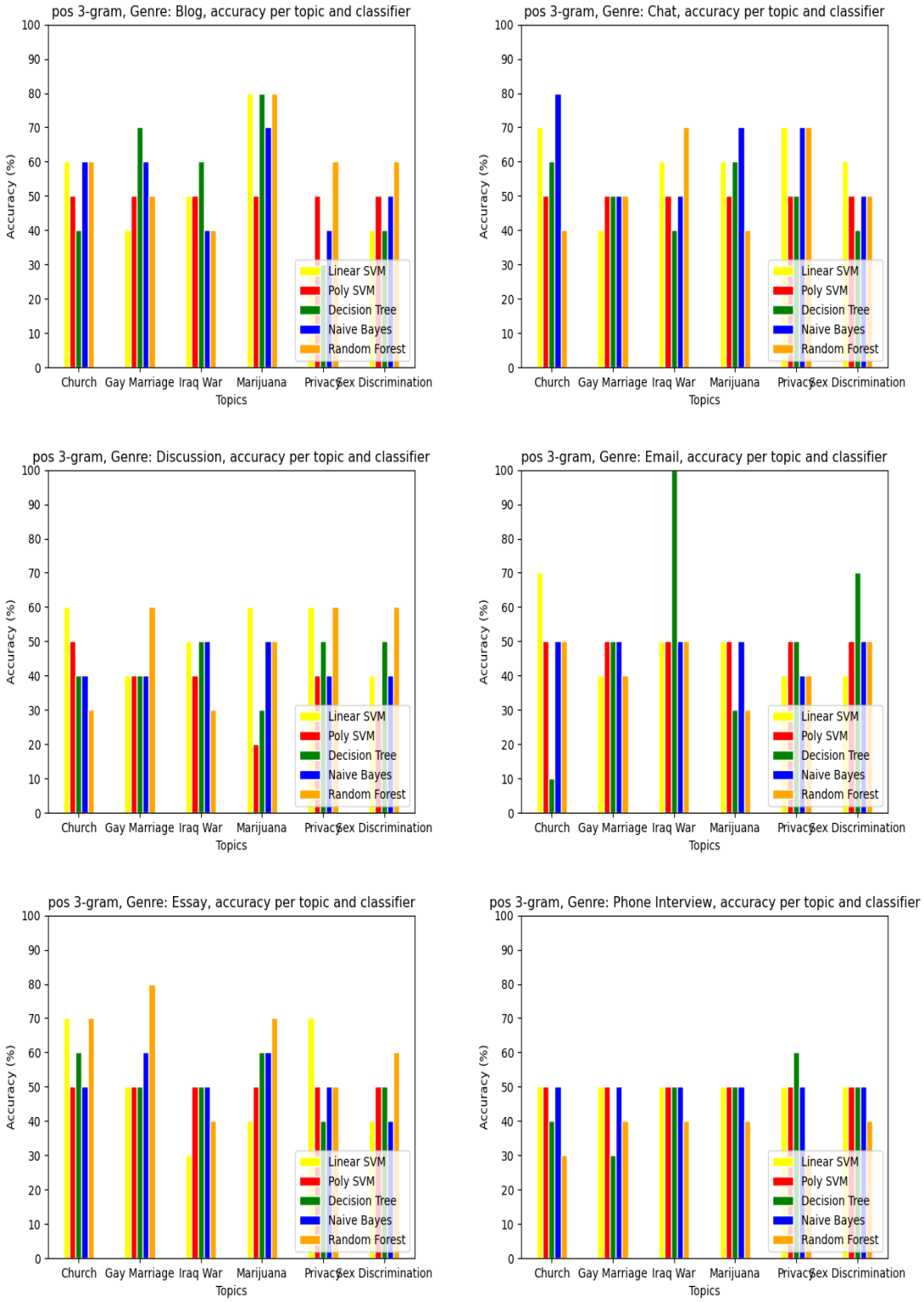
Εικόνα 49: Accuracy ανά Topic για Character N – Gram, N = 5



Εικόνα 50: Accuracy ανά Topic για POS N – Gram, N = 1



Εικόνα 51: Accuracy ανά Topic για POS N – Gram, N = 2



Εικόνα 52: Accuracy ανά Topic για POS N – Gram, N = 3