ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΤΟΜΕΑΣ ΤΕΧΝΟΛΟΓΙΑΣ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΥΠΟΛΟΓΙΣΤΩΝ

Ανάλυση Συναισθήματος Σε Μεγάλο Όγκο Δεδομένων Κειμένου Με Χρήση Κατανεμημένων Τεχνικών Μηχανικής Εκμάθησης

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

του

ΙΩΑΝΝΗ ΙΩΑΝΝΟΥ

Επιβλέπων : Νεκτάριος Κοζύρης
Καθηγητής Ε.Μ.Π.

Αθήνα, Ιούλιος 2014

ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΤΟΜΕΑΣ ΤΕΧΝΟΛΟΓΙΑΣ ΠΛΗΡΟΦΟΡΙΚΗΣ
ΚΑΙ ΥΠΟΛΟΓΙΣΤΩΝ

Ανάλυση Συναισθήματος Σε Μεγάλο Όγκο Δεδομένων Κειμένου Με Χρήση Κατανεμημένων Τεχνικών Μηχανικής Εκμάθησης

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

του

ΙΩΑΝΝΗ ΙΩΑΝΝΟΥ

Επιβλέπων : Νεκτάριος Κοζύρης
Καθηγητής Ε.Μ.Π.

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 3η Ιουλίου 2014.

(Υπογραφή)	*(Υπογραφή)*	*(Υπογραφή)*
.....	.....	.....
Νεκτάριος Κοζύρης Καθηγητής Ε.Μ.Π.	Παναγιώτης Τσανάκας Καθηγητής Ε.Μ.Π.	Δημήτριος Τσουμάκος Επίκουρος Καθηγητής Ιόνιο Πανεπιστήμιο

Αθήνα, Ιούλιος 2014

(Υπογραφή)

.....

ΙΩΑΝΝΗΣ ΙΩΑΝΝΟΥ

Διπλωματούχος Ηλεκτρολόγος Μηχανικός και Μηχανικός Υπολογιστών Ε.Μ.Π.

Copyright © Ιωάννης Ιωάννου, 2014
Με επιφύλαξη παντός δικαιώματος. All rights reserved.

Απαγορεύεται η αντιγραφή, αποθήκευση και διανομή της παρούσας εργασίας, εξ' ολοκλήρου ή τμήματος αυτής, για εμπορικό σκοπό. Επιτρέπεται η ανατύπωση, αποθήκευση και διανομή για σκοπό μη κερδοσκοπικό, εκπαιδευτικής ή ερευνητικής φύσης, υπό την προϋπόθεση να αναφέρεται η πηγή προέλευσης και να διατηρείται το παρόν μήνυμα. Ερωτήματα που αφορούν τη χρήση της εργασίας για κερδοσκοπικό σκοπό πρέπει να απευθύνονται προς τον συγγραφέα.

Οι απόψεις και τα συμπεράσματα που περιέχονται σ' αυτό το έγγραφο εκφράζουν τον συγγραφέα και δεν πρέπει να ερμηνευθεί ότι αντιπροσωπεύουν τις επίσημες θέσεις του Εθνικού Μετσόβιου Πολυτεχνείου.

Περίληψη

Αντικείμενο της συγκεκριμένης διπλωματικής εργασίας είναι η ανάλυση συναισθήματος σε μεγάλο όγκο δεδομένων κειμένου με τη χρήση κατανεμημένων τεχνικών μηχανικής εκμάθησης. Για το σκοπό αυτό μελετάται το Hadoop, το οποίο αποτελεί το πλέον διαδεδομένο framework για κατανεμημένη επεξεργασία και το Mahout, το οποίο προσφέρει μια βιβλιοθήκη για αλγόριθμους μηχανικής εκμάθησης πάνω από το Hadoop. Στη συνέχεια μελετούνται οι κατανεμημένοι αλγόριθμοι που προσφέρονται από το Mahout, ο Naive Bayes και ο Random Forests, και επιλέγεται ο καταλληλότερος γι' αυτήν την εργασία. Κατόπιν, γνωρίζοντας πως η ακρίβεια εξαγωγής του συναισθήματος διαδραματίζει καθοριστικό παράγοντα σε μια εφαρμογή, μελετούνται πολλές διαδεδομένες μέθοδοι βελτίωσης της ακρίβειας. Ακολούθως, γίνεται παρουσίαση της εφαρμογή εξαγωγής συναισθήματος που μελετάται στην εργασία αυτήν, καθώς επίσης και εφαρμογή του επιλεγμένου αλγορίθμου και των διαφορετικών μεθόδων βελτίωσης της ακρίβειας, με σκοπό την επίτευξη της καλύτερης δυνατής ακρίβειας ταξινόμησης και την εξαγωγή συμπερασμάτων. Έπειτα, εξηγείται η απαίτηση των σύγχρονων εφαρμογών για χρήση μεγάλου όγκου δεδομένων και πως αυτή καθιστά υποχρεωτική τη χρήση κατανεμημένων αλγορίθμων. Στη συνέχεια, θέλοντας να επιβεβαιώσουμε τη χρησιμότητα των κατανεμημένων αλγορίθμων ταξινόμησης, γίνεται αξιολόγηση της χρήσης τους όταν υπάρχει μεγάλος όγκος δεδομένων. Τέλος, μελετάται η επίδραση του αριθμού των εκπαιδευτικών εγγράφων (από μερικές εκατοντάδες, μέχρι μερικά εκατομμύρια) στην ακρίβεια της εφαρμογής που μελετήθηκε, με σκοπό την περαιτέρω βελτίωση της ακρίβειας ταξινόμησης.

Λέξεις Κλειδιά

ανάλυση συναισθήματος, μεγάλος όγκος δεδομένων, mahout, hadoop, κατανεμημένοι ταξινομητές, ταξινόμηση, naive bayes, random forests, βελτίωση ακρίβειας, προεπεξεργασία δεδομένων, επιλογή χαρακτηριστικών, βελτιστοποιήσεις, κλιμακωσιμότητα

Abstract

The scope of this particular thesis is the sentiment analysis within big data, using distributed techniques of machine learning. For this purpose, Hadoop is studied, which is the most widespread framework for distributed processing, as well as Mahout, which offers a library for algorithms of machine learning. The distributed algorithms offered by Mahout – Naive Bayes and Random Forests – are also studied, and the one most suitable for this project is chosen. Afterwards, knowing that the accuracy in sentiment analysis applications is a determining factor, various widespread methods of accuracy improvement are studied. The sentiment analysis application that this thesis studies is then presented, and the chosen algorithm and various methods of accuracy improvement are applied, in order to achieve the best possible accuracy in classification and to reach conclusions. I go on to explain the demands of contemporary applications for big data use and how this renders the use of distributed algorithms necessary. Wishing to confirm the usefulness of distributed algorithms for classification, their use when there is big data is evaluated. Finally, the effect that the number of training documents (from hundreds to millions) has on the accuracy of the studied application is analyzed, so that accuracy in classification is further improved.

Keywords

sentiment analysis, big data, mahout, hadoop, distributed classifiers, classification, naive bayes, random forests, accuracy improvement, data pre-processing, feature selection, optimizations, scalability

Πίνακας Περιεχομένων

1 Εισαγωγή 13

1.1 Ανάλυση συναισθήματος 13

1.2 Αντικείμενο διπλωματικής 14

1.3 Οργάνωση κειμένου 14

2 Hadoop και Mahout 16

2.1 Hadoop 16

2.1.1 Map Reduce 16

2.1.2 HDFS 18

2.1.3 Αρχιτεκτονική Hadoop 18

2.2 Mahout 19

2.2.1 Ταξινόμηση κειμένου 19

3 Κατανεμημένοι αλγόριθμοι ταξινόμησης 21

3.1 Naive Bayes Classifier 21

3.1.1 Μαθηματικό μοντέλο 21

3.1.2 «Naive» υπόθεση 23

3.1.3 Παράδειγμα χρήσης Naive Bayes 24

3.1.4 Naive Bayes στο Mahout 25

3.2 Random Forests 27

3.2.1 Decision Tree 27

3.2.2 Αλγόριθμος Random Forests 29

4 Μέθοδοι βελτίωσης της ακρίβειας του ταξινομητή 31

4.1 Επιλογή κατάλληλου training set 31

4.1.1 Μέγεθος training set 31

4.1.2 Περιεχόμενο training set 32

4.1.3 Αριθμός εκπαιδευτικών εγγράφων ανά κατηγορία 32

4.2 Τεχνικές βελτίωσης ακρίβειας 33

4.2.1 Μετατροπή γραμμάτων σε πεζά και αφαίρεση συμβόλων 33

4.2.2 Αφαίρεση αριθμών 34

4.2.3 Αφαίρεση stopwords με χρήση λιστών 35

4.2.4 Tf-idf 36

4.2.5 Κανονικοποίηση 37

4.2.6 N-gram 37

4.2.7 Stemming 38

4.2.8 Ελάχιστες εμφανίσεις 39
4.2.9 Ελάχιστες εμφανίσεις σε διαφορετικά έγγραφα 40
4.2.10 Μέγιστες εμφανίσεις σε διαφορετικά έγγραφα 40
4.2.11 Information Gain, Chi-Squared και Bi-Normal Separation 41
5 Εφαρμογή 44
5.1 Σκοπός της εφαρμογής 44
5.2 Δεδομένα της εφαρμογής 44
5.3 Επιλογή κατάλληλου training set 46
5.3.1 Βαθμολογίες ανά κατηγορία 46
5.3.2 Καταλληλότερα δεδομένα 48
5.3.3 Αναλογία εγγράφων ανά κατηγορία και βαθμολογία 51
5.4 Εφαρμογή τεχνικών βελτίωσης της ακρίβειας 52
5.4.1 Εφαρμογή τεχνικής 1 (Μετατροπή γραμμάτων σε πεζά και αφαίρεση συμβόλων) 52
5.4.2 Εφαρμογή τεχνικής 2 (Αφαίρεση αριθμών) 53
5.4.3 Εφαρμογή τεχνικής 3 (Tf-idf) 53
5.4.4 Εφαρμογή τεχνικής 4 (Stemming) 54
5.4.5 Εφαρμογή τεχνικής 5 (Αφαίρεση stopwords με χρήση λιστών) 54
5.4.6 Εφαρμογή τεχνικής 6 (Μέγιστες εμφανίσεις σε διαφορετικά έγγραφα) 55
5.4.7 Εφαρμογή τεχνικής 7 (Ελάχιστες εμφανίσεις) 56
5.4.8 Εφαρμογή τεχνικής 8 (Ελάχιστες εμφανίσεις σε διαφορετικά έγγραφα) 58
5.4.9 Εφαρμογή τεχνικής 9 (Κανονικοποίηση) 58
5.4.10 Εφαρμογή τεχνικής 10 (Information Gain/Chi-Squared/Bi-Normal Separation) 59
5.4.11 Εφαρμογή τεχνικής 11 (N-gram) 61
5.5 Μια ακόμη δοκιμή 63
5.6 Σύνοψη ενότητας 63
6 Big Data 65
6.1 Χρησιμότητα μεγάλου όγκου δεδομένων και αναγκαιότητα χρήσης κατανεμημένων αλγορίθμων ταξινόμησης 65
6.2 Αξιολόγηση κατανεμημένων ταξινομητών 67
6.3 Εφαρμογή (Συνέχεια) 74
6.4 Σύνοψη ενότητας 76
7 Επίλογος 78
7.1 Σύντομη ανακεφαλαίωση 78
7.2 Γενικά συμπεράσματα 79
7.3 Επεκτάσεις 80
8 Βιβλιογραφία 81

Πίνακας Εικόνων

Εικόνα 1: Αρχιτεκτονική Hadoop 18
Εικόνα 2: Εκπαιδευμένο Decision tree 28
Εικόνα 3: Διαχωρισμός training data 29
Εικόνα 4: Πληροφορίες που υπάρχουν για ένα εστιατόριο 45
Εικόνα 5: Πληροφορίες που υπάρχουν για μια κριτική 45
Εικόνα 6: Χρόνος εκτέλεσης Naive Bayes σε συνάρτηση με το μέγεθος της εισόδου και τους χρησιμοποιούμενους κόμβους 69
Εικόνα 7: Εστίαση στην περιοχή της εικόνας 6, όπου ο αριθμός των εγγράφων που χρησιμοποιείται είναι μικρός 70
Εικόνα 8: Χρόνος εκπαίδευσης και αξιολόγησης για 8 εκατομμύρια training data και 8 εκατομμύρια test data σε συνάρτηση με τους χρησιμοποιούμενους κόμβους 71
Εικόνα 9: Χρόνος εκπαίδευσης και αξιολόγησης για 100 χιλιάδες training data και 100 χιλιάδες test data σε συνάρτηση με τους χρησιμοποιούμενους κόμβους 72
Εικόνα 10: Πως επηρεάζουν οι τεχνικές βελτίωσης της ακρίβειας το χρόνο εκτέλεσης του Naive Bayes όταν εκτελείται σειριακά 73
Εικόνα 11: Πως επηρεάζουν οι τεχνικές βελτίωσης της ακρίβειας το χρόνο εκτέλεσης του Naive Bayes όταν εκτελείται σε 4 κόμβους 73
Εικόνα 12: Ακρίβεια ταξινομητή σε συνάρτηση με τον αριθμό των εκπαιδευτικών εγγράφων 75
Εικόνα 13: Ακρίβεια ταξινομητή σε συνάρτηση με τον αριθμό των εκπαιδευτικών εγγράφων όταν χρησιμοποιούνται ή όχι οι τεχνικές βελτίωσης της ακρίβειας 75

Πίνακας Πινάκων

Πίνακας 1: Ψευδοκώδικας MapReduce παραδείγματος 17
Πίνακας 2: Mahout έναντι άλλων προσεγγίσεων 19
Πίνακας 3: Μέγιστος συνιστώμενος αριθμός εκπαιδευτικών εγγράφων για τους κατανεμημένους αλγορίθμους που προσφέρει το Mahout 20
Πίνακας 4: Training set παραδείγματος Naive Bayes 24
Πίνακας 5: Training data για την εκπαίδευση του decision tree 27
Πίνακας 6: Training set παραδείγματος χρήσης τεχνικής Chi-Squared 42
Πίνακας 7: Βαθμολογία 3 ως Αρνητική και ως Θετική 47
Πίνακας 8: Σύγκριση βαθμολογίας 3 με τις υπόλοιπες βαθμολογίες 47
Πίνακας 9: Σύγκριση των βαθμολογιών 1, 2, 4 και 5 48
Πίνακας 10: Χρήση τίτλων σχολίων στα training data 50
Πίνακας 11: Δοκιμή διαφορετικών συνδυασμών τίτλων και σχολίων στα training και test data 51
Πίνακας 12: Εφαρμογή τεχνικής 1 (μετατροπή γραμμάτων σε πεζά και αφαίρεση συμβόλων) 52
Πίνακας 13: Εφαρμογή τεχνικής 2 (αφαίρεση αριθμών) 53
Πίνακας 14: Εφαρμογή τεχνικής 3 (tf-idf) 53
Πίνακας 15: Εφαρμογή τεχνικής 4 (stemming) 54
Πίνακας 16: Εφαρμογή τεχνικής 5 (αφαίρεση stopwords με χρήση λιστών) 54
Πίνακας 17: Εφαρμογή τεχνικής 6 (μέγιστες εμφανίσεις σε διαφορετικά έγγραφα) 55
Πίνακας 18: Εφαρμογή τεχνικής 7 (ελάχιστες εμφανίσεις) 56
Πίνακας 19: Σύγκριση μεταξύ τεχνικών 5 και 6 57
Πίνακας 20: Εφαρμογή τεχνικής 8 (ελάχιστες εμφανίσεις σε διαφορετικά έγγραφα) 58
Πίνακας 21: Εφαρμογή τεχνικής 9 (κανονικοποίηση) 59
Πίνακας 22: Εφαρμογή τεχνικής 10 (information gain, chi-squared και bi-normal separation) 60
Πίνακας 23: Πέντε πιο σημαντικές λέξεις στα training data όταν γίνεται ή όχι εφαρμογή της τεχνικής Bi-Normal Separation 61
Πίνακας 24: Πέντε πιο σημαντικές λέξεις στα training data, όταν γίνεται εφαρμογή των διαφορετικών τεχνικών της τεχνικής 10 61
Πίνακας 25: Εφαρμογή τεχνικής 11 (n-gram) 62
Πίνακας 26: Χρήση μόνο βαθμολογιών 1 και 5 στα test data 63
Πίνακας 27: Speedup όταν γίνεται χρήση 64 εκατομμυρίων εγγράφων 70
Πίνακας 28: Αξιολόγηση ταξινομητή σε μεγάλο όγκο δεδομένων 76

1

Εισαγωγή

1.1 Ανάλυση συναισθήματος

Η ραγδαία ανάπτυξη της τεχνολογίας τα τελευταία χρόνια έχει οδηγήσει σε ευρεία χρήση του διαδικτύου. Καθημερινά, όλο και περισσότεροι χρήστες χρησιμοποιούν το διαδίκτυο για να εκφράσουν τη γνώμη τους και για να μοιραστούν τις εμπειρίες και τα συναισθήματά τους με άλλους χρήστες.

Τα συναισθήματα που κρύβουν τα δεδομένα αυτά έχουν τεράστια σημασία τόσο σε παρόχους προϊόντων και υπηρεσιών, όσο και σε καταναλωτές. Για παράδειγμα, επιχειρήσεις που προσφέρουν υπηρεσίες για ταξίδια μπορούν να εκμεταλλευτούν τις ήδη υπάρχουσες κριτικές χρηστών με σκοπό να προτείνουν ονειρεμένα ταξίδια στους πελάτες τους. Αντίστοιχα, οι καταναλωτές μπορούν να αποφύγουν να αγοράσουν ένα προϊόν το οποίο έχει λάβει πολλές αρνητικές κριτικές στα κοινωνικά δίκτυα π.χ. Twitter.

Η εξαγωγή του συναισθήματος από τα δεδομένα γίνεται με τη χρήση αλγορίθμων ταξινόμησης, οι οποίοι κατατάσσουν αυτόματα έγγραφα σε κατηγορίες. Ωστόσο για να μπορέσουν οι ταξινομητές να πάρουν αυτές τις αποφάσεις, θα πρέπει πρώτα να εκπαιδευτούν, να τους δοθούν δηλαδή παραδείγματα εγγράφων καταταγμένα στη σωστή κατηγορία, έτσι ώστε με τη σειρά τους να είναι σε θέση να κατατάξουν νέα έγγραφα στις κατηγορίες αυτές. Μ' άλλα λόγια, χρησιμοποιούν τη γνώση που τους δίνεται για να μπορούν να εξάγουν το συναίσθημα από νέα έγγραφα.

Μια πολύ σημαντική απαίτηση που υπάρχει κατά τη χρήση ταξινομητών σε μια εφαρμογή είναι η εξαγωγή του συναισθήματος με μεγάλη ακρίβεια. Ένας ταξινομητής που κάνει λάθος εξαγωγή συναισθήματος τις περισσότερες φορές θα οδηγούσε στη λήψη λανθασμένων αποφάσεων, γεγονός το οποίο θα είχε ως αποτέλεσμα την αποτυχία της εφαρμογής, με όποιες συνέπειες αυτό συνεπάγεται. Κρίνεται σκόπιμο να αναφερθεί, ότι η χρήση ενός ταξινομητή δεν είναι ικανή από μόνη της να πετύχει την καλύτερη δυνατή ακρίβεια.

Μια εξίσου σημαντική απαίτηση των σύγχρονων εφαρμογών είναι η χρήση μεγάλου όγκου δεδομένων τόσο κατά την εκπαίδευση, όσο και κατά τη χρήση του ταξινομητή στην κατάταξη νέων εγγράφων σε κατηγορίες. Η απαίτηση αυτή οφείλεται στη ραγδαία αύξηση

των δεδομένων που υπάρχουν στο διαδίκτυο. Με την εμφάνιση του Web 2 δίνεται η δυνατότητα στο χρήστη, εκτός από καταναλωτής, να είναι και παραγωγός πληροφορίας. Μ' άλλα λόγια, ο χρήστης έχει τη δυνατότητα να αλληλεπιδρά και να μοιράζεται δεδομένα μ' άλλους χρήστες. Το πιο αντιπροσωπευτικό παράδειγμα εφαρμογής του Web 2 είναι τα κοινωνικά δίκτυα όπως το Facebook και το Twitter. Σύμφωνα με επίσημα στατιστικά [1] [2], το Facebook διαθέτει πάνω από 1.31 δισεκατομμύρια ενεργούς χρήστες που δημοσιεύουν περισσότερα από 1 δισεκατομμύρια posts συνολικά την ημέρα. Αντίστοιχα, το Twitter διαθέτει πάνω από 255 εκατομμύρια ενεργούς χρήστες που δημοσιεύουν περισσότερα από 500 εκατομμύρια tweets συνολικά την ημέρα. Κρίνεται σκόπιμο να αναφερθεί, πως το Facebook και το Twitter αποτελούν μόνο ένα μικρό μέρος των Web 2 εφαρμογών, γεγονός που υποδηλώνει τον τεράστιο όγκο δεδομένων που διακινείται στο διαδίκτυο. Η πληροφορία που κρύβει ο τεράστιος αυτός όγκος δεδομένων είναι καθοριστικής σημασίας για τις σύγχρονες εφαρμογές.

Ωστόσο, η απαίτηση των σύγχρονων εφαρμογών για χρήση μεγάλου όγκου δεδομένων είναι αδύνατο να ικανοποιηθεί με τη χρήση κλασικών (μη κλιμακώσιμων) αλγορίθμων ταξινόμησης. Συγκεκριμένα, η χρήση μεγάλου όγκου δεδομένων από κλασικούς ταξινομητές έχει ως αποτέλεσμα απαγορευτικά μεγάλους χρόνους εκπαίδευσης και κατάταξης νέων εγγράφων σε κατηγορίες, με αποτέλεσμα να μην ικανοποιούνται οι χρονικές απαιτήσεις που έχουν οι σύγχρονες εφαρμογές [3]. Επίσης, η απαίτηση για μνήμη από τους κλασικούς αλγόριθμους ταξινόμησης αυξάνεται με πολύ γρήγορους ρυθμούς, με αποτέλεσμα την έλλειψη της μνήμης όταν γίνεται χρήση μεγάλου όγκου δεδομένων [3] [4].

1.2 Αντικείμενο διπλωματικής

Αντικείμενο της παρούσας διπλωματικής εργασίας είναι η ανάλυση συναισθήματος σε μεγάλο όγκο δεδομένων. Για το σκοπό αυτό γίνεται χρήση κατανεμημένων αλγορίθμων ταξινόμησης, οι οποίοι κάνουν εφικτή τη χρήση μεγάλου όγκου δεδομένων. Συγκεκριμένα, μελετούνται οι δύο κατανεμημένοι αλγόριθμοι που προσφέρει το Mahout και επιλέγεται ο καταλληλότερος. Στη συνέχεια, λόγω της καθοριστικής σημασίας που έχει η ακρίβεια σε μια εφαρμογή μελετούνται πολλές μέθοδοι, οι οποίες έχουν ως σκοπό την επίτευξη της καλύτερης δυνατής ακρίβειας. Ακολούθως, γίνεται εφαρμογή του επιλεγμένου αλγορίθμου και των διαφορετικών μεθόδων βελτίωσης της ακρίβειας στην εφαρμογή που μελετάται σ' αυτήν την εργασία, με σκοπό την επίτευξη της καλύτερης δυνατής ακρίβειας. Έπειτα, θέλοντας να επιβεβαιώσουμε τη χρησιμότητα των κατανεμημένων αλγορίθμων ταξινόμησης γίνεται αξιολόγηση της χρήσης τους, όταν υπάρχει μεγάλος όγκος δεδομένων. Τέλος, γίνεται μελέτη της επίδρασης του αριθμού των εκπαιδευτικών εγγράφων στην ακρίβεια της εφαρμογής που παρουσιάζεται στην εργασία αυτή, με σκοπό την περαιτέρω βελτίωση της ακρίβειας ταξινόμησης.

1.3 Οργάνωση κειμένου

Το υπόλοιπο αυτής της εργασίας οργανώνεται ως εξής: Στην ενότητα 2 παρουσιάζονται τα 2 κατανεμημένα εργαλεία που θα χρησιμοποιηθούν, το Hadoop και το Mahout. Στη

συνέχεια, στην ενότητα 3 μελετούνται οι 2 κατανεμημένοι αλγόριθμοι ταξινόμησης που υπάρχουν στο Mahout, ο Naive Bayes και ο Random Forests. Ακολούθως, στην ενότητα 4 γίνεται μελέτη των μεθόδων αύξησης της ακρίβειας του ταξινομητή. Στην ενότητα 5 γίνεται παρουσίαση της εφαρμογής εξαγωγής συναισθήματος που μελετάται στην εργασία αυτή, καθώς επίσης και εφαρμογή του Naive Bayes και των διαφορετικών μεθόδων βελτίωσης της ακρίβειας. Η απαίτηση των σύγχρονων εφαρμογών για χρήση μεγάλου όγκου δεδομένων και η αναγκαιότητα χρήσης κατανεμημένων αλγορίθμων ταξινόμησης απασχολούν την ενότητα 6. Επιπλέον, στην ενότητα αυτή γίνεται και αξιολόγηση των κατανεμημένων αλγορίθμων ταξινόμησης όταν υπάρχει μεγάλος όγκος δεδομένων, καθώς επίσης μελετάται και η επίδραση του αριθμού των εκπαιδευτικών εγγράφων στην εφαρμογή που μελετάται στην εργασία αυτή. Κατόπιν, στην ενότητα 7 παρουσιάζεται σύντομη σύνοψη της εργασίας, γενικά συμπεράσματα και τυχόν επεκτάσεις της εργασίας αυτής. Τέλος, στην ενότητα 8 αναγράφεται η βιβλιογραφία.

2

Hadoop και Mahout

Στην ενότητα αυτή, μελετούνται τα δύο κατανεμημένα εργαλεία που θα χρησιμοποιηθούν στη συνέχεια αυτής της εργασίας.

2.1 Hadoop

Το Hadoop είναι μια ανοιχτού κώδικα πλατφόρμα λογισμικού, η οποία μπορεί να χρησιμοποιηθεί για την ανάλυση και επεξεργασία μεγάλου όγκου δεδομένων [5]. Συγκεκριμένα, το Hadoop κατανέμει τα δεδομένα και τη διαδικασία ανάλυσης τους σε ομάδες υπολογιστών (cluster), ώστε να επεξεργαστούν παράλληλα και συνεπώς να επιταχυνθεί η όλη διαδικασία.

Τεχνικά το Hadoop παρέχει δύο βασικές υπηρεσίες: την αξιόπιστη αποθήκευση δεδομένων με τη χρήση του Hadoop Distributed File System (HDFS) και την παράλληλη επεξεργασία των δεδομένων χρησιμοποιώντας το προγραμματιστικό μοντέλο Map Reduce. Στη συνέχεια, θα αναλυθεί η λειτουργία των δύο αυτών βασικών υπηρεσιών, καθώς επίσης θα παρουσιαστεί και η αρχιτεκτονική που ακολουθεί το Hadoop.

2.1.1 Map Reduce

Το Map Reduce είναι ένα προγραμματιστικό μοντέλο για παράλληλη επεξεργασία μεγάλου όγκου δεδομένων με τη χρήση ομάδων υπολογιστών (cluster). Το μοντέλο είναι αρκετά απλό στη χρήση και ευρέως διαδεδομένο.

Σύμφωνα με αυτό το προγραμματιστικό μοντέλο, το πρόβλημα χωρίζεται σε δύο φάσεις, την map και την reduce. Κατά τη διάρκεια της πρώτης φάσης, δηλαδή της map φάσης, ξεχωριστά κομμάτια της εισόδου ανατίθενται σε διαφορετικές διεργασίες (mappers), οι οποίες μετά από επεξεργασία των δεδομένων εισόδου εξάγουν ένα σετ από ενδιάμεσα αποτελέσματα. Η είσοδος δίνεται στη μορφή ζευγών <κλειδί, τιμή> και τα ενδιάμεσα αποτελέσματα έχουν και αυτά την ίδια μορφή. Στη συνέχεια εκτελείται η reduce φάση, κατά την οποία δίνονται τα αποτελέσματα της map φάσης σ' ένα συνήθως μικρότερο αριθμό διεργασιών (reducers), οι οποίες μετά από επεξεργασία εξάγουν ζεύγη της μορφής <κλειδί, τιμή>. Ουσιαστικά, για την επίλυση ενός προβλήματος πρέπει να προγραμματιστούν οι

συναρτήσεις map και reduce. Ωστόσο, ο προγραμματισμός της συνάρτησης reduce δεν είναι υποχρεωτικός και μπορεί να αγνοηθεί, όταν δεν είναι απαραίτητος για την επίλυση του προβλήματος.

Για την καλύτερη κατανόηση του προγραμματιστικού μοντέλου Map Reduce, στον πίνακα 1 δίνεται ο ψευδοκώδικας ενός απλού παραδείγματος [6] [7]. Το παράδειγμα αυτό παίρνει ως είσοδο διάφορα αρχεία κειμένου και μετρά τις εμφανίσεις κάθε διαφορετικής λέξης που υπάρχει στα αρχεία αυτά.

Πίνακας 1: ***Ψευδοκώδικας MapReduce παραδείγματος***

Map συνάρτηση	**Reduce συνάρτηση**
Input: (key, value) = (name of document, text of document) Output: (key, value) = (word, 1)	Input: (key, intermediate values) = (word, list of values) Output: (word, occurrences)

Map(input) {
for each word w in input
EmitIntermediate(w, 1);
}
	Reduce(String key, Iterator values) {
	int result = 0;
	for each v in values
	result += v;
	Emit(w , result);
	}

Η εκτέλεση του προγράμματος αυτού έχει ως εξής: Αρχικά, η είσοδος μοιράζεται σε αρκετά κομμάτια, όσες είναι και οι map διεργασίες. Κάθε μια από αυτές τις διεργασίες εκτελεί τη συνάρτηση map πάνω στα δικά της δεδομένα. Η συνάρτηση map παίρνει ως είσοδο ένα κείμενο και βγάζει ως έξοδο ένα ζεύγος <λέξη, 1> για κάθε λέξη που υπάρχει στο κείμενο της εισόδου. Μετά το τέλος της map φάσης, όλα τα ζεύγη ταξινομούνται σύμφωνα με τη λέξη-κλειδί και στη συνέχεια, ομαδοποιούνται με βάση αυτήν. Για παράδειγμα, αν η έξοδος της map φάσης είναι τα ζεύγη <ωραίο, 1>, <ξενοδοχείο, 1>, <ωραίο, 1> και <εστιατόριο, 1>, μετά την ταξινόμηση και ομαδοποίηση, τα ζεύγη θα έχουν τη μορφή <εστιατόριο, {1}>, <ξενοδοχείο, {1}> και <ωραίο, {1,1}>. Η ταξινόμηση και η ομαδοποίηση γίνονται από το σύστημα, χωρίς να χρειάζεται να τις υλοποιήσει ο χρήστης. Στη συνέχεια, αυτά τα ζεύγη μοιράζονται στις reduce διεργασίες. Κάθε μια από τις reduce διεργασίες εκτελεί την reduce συνάρτηση στα ζεύγη που της μοιράστηκαν. Η reduce συνάρτηση είναι υπεύθυνη για να αθροίσει όλες τις εμφανίσεις κάθε λέξης-κλειδιού και να δώσει ως έξοδο το ζεύγος <κλειδί, άθροισμα των τιμών>. Για παράδειγμα, αν της δινόταν το ζεύγος <ωραίο, {1,1}> ως είσοδος, θα έδινε το ζεύγος <ωραίο, 2> ως έξοδο. Μετά το τέλος της reduce φάσης, τα ζεύγη που εξάγονται αποτελούν και τη λύση του προβλήματος.

Για την εκτέλεση των map και reduce εργασιών χρησιμοποιούνται ο jobtracker και οι tasktrackers. Ο jobtracker είναι υπεύθυνος για το διαμοιρασμό των διαθέσιμων map και reduce εργασιών στους tasktrackers, οι οποίοι είναι υπεύθυνοι με τη σειρά τους για την εκτέλεση των εργασιών αυτών. Ο jobtracker επίσης, ενημερώνει τους tasktrackers για το που βρίσκονται τα δεδομένα που θα χρησιμοποιήσουν στην εργασία που τους ανατέθηκε.

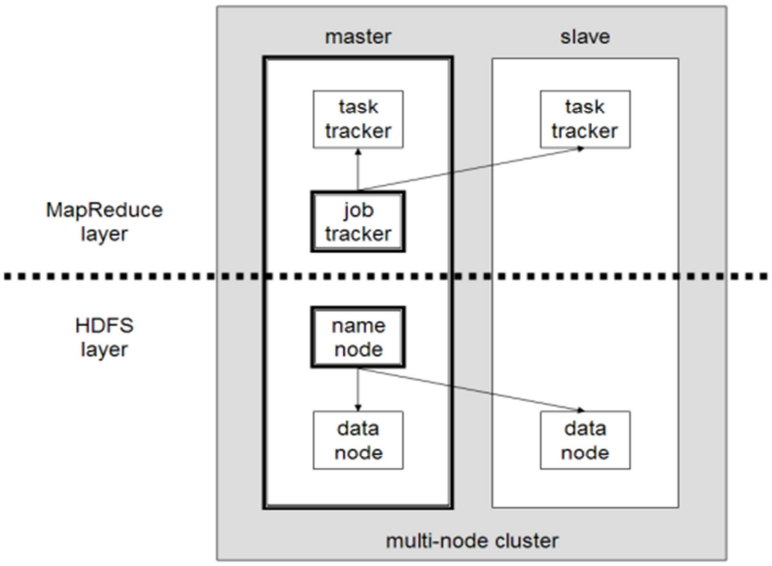
2.1.2 HDFS

Το HDFS είναι ένα κατανεμημένο σύστημα αρχείων και χρησιμοποιείται για την αποθήκευση μεγάλου όγκου δεδομένων. Με άλλα λόγια, για την αποθήκευση των δεδομένων χρησιμοποιούνται πολλοί κόμβοι, όπου καθένας απ' αυτούς έχει αποθηκευμένο μόνο ένα κομμάτι όλων των δεδομένων που υπάρχουν. Επιπλέον, επιτυγχάνει να είναι αξιόπιστο αποθηκεύοντας τα δεδομένα σε περισσότερους από ένα κόμβους. Δηλαδή αν κάποια δεδομένα χαθούν από ένα κόμβο, υπάρχουν τα ίδια δεδομένα και σε άλλο κόμβο.

Το σύστημα αρχείων HDFS χρησιμοποιεί ένα κεντρικό κόμβο, τον namenode, ο οποίος είναι και το μοναδικό σημείο αποτυχίας, που κρατά τις πληροφορίες για το που βρίσκεται κάθε δεδομένο στο σύστημα αρχείων. Αν αυτός δεν είναι διαθέσιμος, τότε δεν υπάρχει πρόσβαση στο σύστημα αρχείων. Επιπλέον, χρησιμοποιεί ένα ακόμη κόμβο, τον secondary namenode, ο οποίος κρατά snapshots των δεδομένων του namenode και μαζί με τα αρχεία ιστορικού του namenode επαναφέρει το σύστημα αρχείων μετά από αποτυχία. Οι υπόλοιποι κόμβοι ονομάζονται datanodes και απλά αποθηκεύουν δεδομένα.

2.1.3 Αρχιτεκτονική Hadoop

Το Hadoop ακολουθεί το μοντέλο αρχιτεκτονικής master/slave. Στην εικόνα 1 παρουσιάζεται η αρχιτεκτονική του.



Εικόνα 1: ***Αρχιτεκτονική Hadoop***

Κατά τη διάρκεια εγκατάστασης του Hadoop, ένας κόμβος από ολόκληρη τη συστοιχία υπολογιστών τίθεται ως master και οι υπόλοιποι ως slaves. Όπως φαίνεται και από την εικόνα 1, ο jobtracker δηλαδή αυτός που διαμοιράζει τις διαθέσιμες εργασίες και ο namenode δηλαδή αυτός που κρατάει τις πληροφορίες για το που βρίσκεται το κάθε δεδομένο στο κατανεμημένο σύστημα αρχείων, βρίσκονται μόνο στον master node. Οι tasktrackers, δηλαδή αυτοί που εκτελούν τις map και reduce συναρτήσεις, και οι datanodes δηλαδή αυτοί που φυλάνε ένα κομμάτι απ' όλα τα δεδομένα που είναι διαθέσιμα στο HDFS, βρίσκονται μόνο στους slave nodes. Το Hadoop δίνει τη δυνατότητα ο master node να

συμπεριφέρεται και ως slave node, με σκοπό την πλήρη υπολογιστική εκμετάλλευση και αυτού του κόμβου. Στην εικόνα 1 αυτό ισχύει, μιας και στον master node τρέχει tasktracker και datanode.

Όπως αναφέρθηκε, ο jobtracker είναι και υπεύθυνος για την ενημέρωση των tasktrackers για το που βρίσκονται τα δεδομένα που θα χρησιμοποιήσουν στην εργασία που τους ανατέθηκε. Αυτό το επιτυγχάνει με το να συμβουλεύεται τον namenode, ο οποίος βρίσκεται στο ίδιο μηχάνημα, στον master node, και ο οποίος διαθέτει όλη την πληροφορία για το που βρίσκονται όλα τα δεδομένα που υπάρχουν στο κατανεμημένο σύστημα αρχείων. Επίσης, ο jobtracker προσπαθεί να κατανέμει εργασίες στους tasktrackers, οι οποίες για την εκτέλεσή τους χρειάζονται δεδομένα, τα οποία βρίσκονται στον datanode που βρίσκεται στον κόμβο τους. Αυτό γίνεται με σκοπό να μη χρειάζεται να γίνει μεταφορά δεδομένων από ένα κόμβο σ' ένα άλλο, μιας και αυτό θα σπαταλούσε πολύτιμο χρόνο.

2.2 Mahout

Το Mahout είναι μια βιβλιοθήκη αλγορίθμων μηχανικής εκμάθησης [8]. Περιέχει αλγόριθμους που χρησιμοποιούνται για clustering, classification, recommendation και για πολλούς άλλους σκοπούς. Αυτό που κάνει το Mahout διαφορετικό από άλλες βιβλιοθήκες αλγορίθμων μηχανικής εκμάθησης είναι το γεγονός ότι οι περισσότεροι αλγόριθμοι μηχανικής εκμάθησης που προσφέρει είναι κατανεμημένοι και κλιμακώσιμοι, γεγονός που τους καθιστά κατάλληλους για μεγάλο όγκο δεδομένων. Για να γίνει αυτό εφικτό, οι αλγόριθμοι αυτοί έχουν υλοποιηθεί σύμφωνα με το προγραμματιστικό μοντέλο MapReduce και εκτελούνται πάνω από το Hadoop. Πιο κάτω γίνεται εμβάθυνση στη χρήση του Mahout στην ταξινόμηση κειμένων, μιας και μόνο αυτό το κομμάτι του Mahout αφορά τη συγκεκριμένη εργασία.

2.2.1 Ταξινόμηση κειμένου

Το Mahout μπορεί να χρησιμοποιηθεί σε πολλές διαφορετικές εφαρμογές ταξινόμησης κειμένου παρουσιάζοντας συγκριτικό πλεονέκτημα έναντι των υπόλοιπων προσεγγίσεων, όταν απαιτείται να χρησιμοποιηθεί μεγάλος όγκος δεδομένων. Λόγω των κατανεμημένων και κλιμακώσιμων αλγορίθμων που είναι διαθέσιμοι, η χρήση πολλών εγγράφων δεν απαιτεί απαγορευτικά πολύ χρόνο και μνήμη. Ο πίνακας 2 παρουσιάζει σε ποιες περιπτώσεις το Mahout είναι καταλληλότερο από τις υπόλοιπες προσεγγίσεις [3]:

Πίνακας 2: ***Mahout έναντι άλλων προσεγγίσεων***

Αριθμός εγγράφων	Σχόλια
<100 χιλιάδες	Δε συνιστάται η χρήση του Mahout. Οι υπόλοιπες προσεγγίσεις υπερτερούν σε ταχύτητα.
100 χιλιάδες μέχρι 1 εκατομμύριο	Το Mahout αρχίζει να αποτελεί καλή λύση. Τα εργαλεία που προσφέρει κάνουν το Mahout επιθυμητό, αν και δεν υπερτερεί σε ταχύτητα.
1 μέχρι 10 εκατομμύρια	Το Mahout υπερτερεί σε ταχύτητα.
>10 εκατομμύρια	Το Mahout αποτελεί ιδανική λύση. Οι περισσότερες άλλες προσεγγίσεις αποτυγχάνουν.

Όπως παρουσιάζεται στον πίνακα 2, η μόνη περίπτωση που δεν προτείνεται χρήση του Mahout είναι όταν η εφαρμογή απαιτεί λιγότερα από 100 χιλιάδες έγγραφα. Σε κάθε άλλη περίπτωση, η χρήση του Mahout συνιστάται. Ειδικότερα στις περιπτώσεις όπου ο αριθμός των εγγράφων που απαιτείται είναι πάνω από 1 εκατομμύριο, η χρήση του Mahout αποτελεί μια από τις λιγοστές λύσεις [3].

Δυστυχώς, λόγω της πρόσφατης δημιουργίας του Mahout, οι μόνοι κατανεμημένοι αλγόριθμοι που προσφέρει είναι ο Naive Bayes, ο Complementary Naive Bayes και ο Random Forests. Προσφέρονται επίσης, ο Support Vector Machines (SVM) και ο Stochastic Gradient Descent (SGD), αλλά αυτοί είναι σειριακοί και συνεπώς δε θα μελετηθούν σ' αυτήν την εργασία.

Ο πίνακας 3 παρουσιάζει μέχρι ποιο αριθμό εκπαιδευτικών εγγράφων συστήνεται να χρησιμοποιηθεί ο Naive Bayes, ο Complementary Naive Bayes και ο Random Forests, έτσι ώστε η εκπαίδευση τους να μη χρειάζεται αρκετό χρόνο [3].

Πίνακας 3: ***Μέγιστος συνιστώμενος αριθμός εκπαιδευτικών εγγράφων για τους κατανεμημένους αλγορίθμους που προσφέρει το Mahout***

Αλγόριθμος	**Μέγιστος συνιστώμενος αριθμός εκπαιδευτικών παραδειγμάτων**
Naive Bayes	<100 εκατομμύρια
Complementary Naive Bayes	<100 εκατομμύρια
Random Forests	<10 εκατομμύρια

Όπως παρουσιάζεται στον πίνακα 3, η χρήση του Random Forests δε συνίσταται στην περίπτωση που ο αριθμός των εκπαιδευτικών εγγράφων ξεπερνά τα 10 εκατομμύρια. Αυτό οφείλεται στο γεγονός ότι ο αλγόριθμος έχει μεγάλη υπολογιστική πολυπλοκότητα και επίσης βρίσκεται στα πρώτα στάδια της υλοποίησης του στο Mahout [9]. Το γεγονός αυτό όμως, δεν καθιστά τον Random Forests ως κακή επιλογή. Ο Random Forests αποτελεί ένα από τους πιο ακριβής ταξινομητές και μπορεί να χρησιμοποιηθεί και σε περιπτώσεις όπου οι σειριακοί αλγόριθμοι δεν είναι τόσο αποδοτικοί (περισσότερο από 1 εκατομμύριο έγγραφα). Από την άλλη, ο Naive Bayes και ο Complementary Naive Bayes μπορούν να χρησιμοποιηθούν ακόμη και στην περίπτωση που τα διαθέσιμα έγγραφα κοντεύουν τα 100 εκατομμύρια.

Στην εργασία αυτή, ο αριθμός των εκπαιδευτικών εγγράφων που θα χρησιμοποιηθεί θα ξεπερνά τα 10 εκατομμύρια και συνεπώς η χρήση του Naive Bayes ή του Complementary Naive Bayes αποτελεί μονόδρομο. Οι δύο αυτοί αλγόριθμοι είναι σχεδόν πανομοιότυποι, γι' αυτό σ' αυτήν την εργασία γίνεται ανάλυση και χρήση μόνο του Naive Bayes. Η διαφορά τους ωστόσο επισημαίνεται. Ο Naive Bayes θα μελετηθεί στη λεπτομέρειά του στην επόμενη ενότητα. Τέλος, κρίνεται σκόπιμο να μελετηθεί και ο Random Forests, ο οποίος φαίνεται να παρουσιάζει αρκετό ενδιαφέρον, ακόμη και αν δεν αποτελεί την καλύτερη λύση, όταν τα διαθέσιμα εκπαιδευτικά έγγραφα είναι περισσότερα από 10 εκατομμύρια.

3

Κατανεμημένοι αλγόριθμοι ταξινόμησης

Στην ενότητα αυτή, μελετούνται οι κατανεμημένοι αλγόριθμοι που προσφέρει το Mahout. Αρχικά, παρουσιάζεται ο Naive Bayes με πλήρη λεπτομέρεια και στη συνέχεια παρουσιάζεται εν συντομία ο Random Forests.

3.1 Naive Bayes Classifier

Ο Naive Bayes είναι ένας από τους πιο δημοφιλής ταξινομητές κειμένου. Αυτό που τον κάνει τόσο δημοφιλή είναι η απλότητα, η γραμμική χρονική πολυπλοκότητα και οι καλές του επιδόσεις σε προβλήματα ταξινόμησης [10].

Ο Naive Bayes είναι πιθανοτικός αλγόριθμος και βασίζεται στο θεώρημα του Bayes. Για να μπορέσει να χρησιμοποιηθεί σε προβλήματα ταξινόμησης πρέπει πρώτα να εκπαιδευτεί, έτσι ώστε να αρχικοποιηθούν οι παράμετροί του. Η εκπαίδευση γίνεται με τα εκπαιδευτικά έγγραφα, δηλαδή έγγραφα τα οποία περιέχουν και την κατηγορία στην οποία ανήκουν. Με το τέλος της εκπαίδευσης, ο Naive Bayes είναι σε θέση να υπολογίσει την πιθανή κατηγορία στην οποία ανήκει ένα νέο έγγραφο.

Συγκεκριμένα, ο Naive Bayes υπολογίζει ξεχωριστά τις πιθανότητες το νέο αυτό έγγραφο να ανήκει σε κάθε μια από τις δοσμένες κατηγορίες και στη συνέχεια δίνει ως κατηγορία κατάταξης, αυτή με τη μεγαλύτερη πιθανότητα. Για τον υπολογισμό των πιο πάνω πιθανοτήτων λαμβάνεται υπόψη τόσο η προγενέστερη πιθανότητα της κάθε κατηγορίας, όσο και η κάθε λέξη που περιέχεται στο νέο έγγραφο.

3.1.1 Μαθηματικό μοντέλο

Σύμφωνα με τον αλγόριθμο του Naive Bayes, η πιο κατάλληλη κατηγορία c για ένα νέο έγγραφο d είναι η κατηγορία που δίνει τη μεγαλύτερη πιθανότητα το έγγραφο d να ανήκει σ' αυτή (maximum a posteriori):

$$l(d) = argmax_{c \in C}\ P\left(c/d\right) \quad (1)$$

Χρησιμοποιώντας το νόμο του Bayes στη σχέση 1, αυτή παίρνει τη μορφή:

$$l(d) = argmax_{c \in C} \frac{P\left(d/c\right) * P(c)}{P(d)} \quad (2)$$

όπου στη σχέση 2 μπορεί να απλοποιηθεί ο παρονομαστής, μιας και επηρεάζεται μόνο από το έγγραφο d και είναι ο ίδιος για κάθε κατηγορία. Συνεπώς, η σχέση 2 παίρνει τη μορφή:

$$l(d) = argmax_{c \in C} \; P\left(d/c\right) * P(c) \quad (3)$$

Το έγγραφο d μπορεί να αναπαρασταθεί ως ένα διάνυσμα από n_d λέξεις χωρισμένες μεταξύ τους με κενό και κατά συνέπεια η σχέση 3 παίρνει τη μορφή:

$$l(d) = argmax_{c \in C} \, P\left(\left(t_1, t_2, \ldots., t_{n_d}\right)/c\right) * P(c) \quad (4)$$

όπου t_i είναι οι λέξεις που παρουσιάζονται στο έγγραφο d. Ο υπολογισμός της πρώτης πιθανότητας στη σχέση 4 είναι πρακτικά αδύνατος, αφού απαιτεί τον υπολογισμό υπερβολικών παραμέτρων. Στην πραγματικότητα, αν το μέγεθος των νέων εγγράφων δεν είναι φραγμένο, ο αριθμός των παραμέτρων που πρέπει να υπολογιστούν είναι άπειρος [10]. Για το λόγο αυτό, εφαρμόζεται η «naive» υπόθεση που κάνει ο αλγόριθμος του Naive Bayes. Η υπόθεση αυτή υποστηρίζει ότι κάθε λέξη είναι ανεξάρτητη από κάθε άλλη λέξη, δοσμένης της κατηγορίας c. Η υπόθεση αυτή, θα αναλυθεί στη λεπτομέρειά της στη συνέχεια. Εφαρμόζοντας λοιπόν την υπόθεση αυτή μαζί με τον κανόνα της αλυσίδας, η σχέση 4 παίρνει την μορφή:

$$l(d) = \; argmax_{c \in C} \left(P(c) * \prod_{1 \le i \le n_d} P\left(t_i/c\right) \right) \quad (5)$$

όπου P(c) είναι η προγενέστερη πιθανότητα της κατηγορίας c και $P(t_i/c)$ είναι η δεσμευμένη πιθανότητα η λέξη t_i να ανήκει στη κατηγορία c. Ουσιαστικά, ο όρος $P(t_i/c)$ είναι μια ένδειξη της συνεισφοράς της λέξης t_i στο να κατηγοριοποιηθεί το έγγραφο d στην κατηγορία c και ο όρος P(c) φανερώνει πόσο πιθανή είναι η κατηγορία c ανάμεσα στις υπόλοιπες κατηγορίες.

Λόγω υπερχειλίσεων στους πολλαπλασιασμούς, χρησιμοποιούνται λογάριθμοι και έτσι η σχέση 5 παίρνει την εξής μορφή:

$$l(d) = argmax_{c \in C} \left(\log P(c) + \sum_{1 \le i \le n_d} \log P\left(t_i/c\right) \right) \quad (6)$$

Για τον υπολογισμό των εν λόγω πιθανοτήτων γίνεται χρήση του training set. Η πιθανότητα P(c) υπολογίζεται ως εξής:

$$P(c) = \frac{training \; documents \; of \; category \; c}{all \; documents} \quad (7)$$

και η $P(t_i/c)$:

$$P\left(t_i/_c\right) = \frac{Tct_i}{\sum_{t' \in V} Tct'} \quad (8)$$

όπου το Tct_i στη σχέση 8 είναι ο αριθμός των εμφανίσεων της λέξης t_i στην κατηγορία c και V είναι το λεξιλόγιο, δηλαδή όλες οι διαφορετικές λέξεις που υπάρχουν στο training set, ανεξαρτήτου κατηγορίας.

Παρατηρώντας τη σχέση 5 και τη σχέση 8, γίνεται φανερό ότι αν μια λέξη που υπάρχει στο έγγραφο d δεν υπάρχει στην κατηγορία c, τότε η δεσμευμένη πιθανότητα $P(t_i/c)$ θα πάρει τιμή 0 και συνεπώς και η πιθανότητα της κατηγορίας θα πάρει τιμή 0. Αυτό όμως δεν είναι κατά ανάγκη σωστό, αφού οι υπόλοιπες λέξεις που υπάρχουν στο έγγραφο d μπορεί να υπάρχουν στην κατηγορία c. Για να επιλυθεί αυτό το μικρό πρόβλημα, η σχέση 8 παίρνει τη μορφή:

$$P\left(t_i/_c\right) = \frac{Tct_i + a_i}{\sum_{t' \in V}(Tct' + a_i)} = \frac{Tct_i + a_i}{\sum_{t' \in V} Tct' + a} \quad (9)$$

όπου το α_i είναι μια σταθερά. Κάθε λέξη μπορεί να έχει διαφορετική τιμή σταθεράς, αλλά συνήθως για λόγους απλότητας χρησιμοποιείται η σταθερά $\alpha_i = 1$ για όλες τις λέξεις. Έτσι, η σχέση 9 παίρνει την εξής μορφή:

$$P\left(t_i/_c\right) = \frac{Tct_i + 1}{\sum_{t' \in V} Tct' + |V|} \quad (10)$$

όπου το $|V|$ είναι το μέγεθος του λεξιλογίου.

3.1.2 «Naive» υπόθεση

Όπως αναφέρθηκε κατά τη μαθηματική ανάλυση του Naive Bayes, ο υπολογισμός της πιθανότητας $P(t_1,t_2,\ldots,t_{n_d}/c)$ είναι πρακτικά αδύνατος και γι' αυτό χρειάστηκε να εφαρμοστεί η «naive» υπόθεση που κάνει ο αλγόριθμος του Naive Bayes.

Η υπόθεση αυτή υποστηρίζει ότι κάθε λέξη είναι ανεξάρτητη από κάθε άλλη, δοσμένης της κατηγορίας. Μ' άλλα λόγια, κάθε λέξη συνεισφέρει ανεξάρτητα από κάθε άλλη λέξη στην πιθανότητα ένα νέο έγγραφο να ανήκει σε μια κατηγορία. Για παράδειγμα, το γεγονός ότι ένα έγγραφο που ανήκει στην κατηγορία «αναψυκτικό» περιέχει τη λέξη «coca», δεν το κάνει λιγότερο ή περισσότερο πιθανό να περιέχει και τη λέξη «cola». Στην πραγματικότητα όμως, η υπόθεση αυτή δεν ισχύει. Πολλές λέξεις είναι εξαρτημένες μεταξύ τους, όπως η «coca» και η «cola», και η εμφάνιση της μιας κάνει πολύ πιθανή και την εμφάνιση της άλλης.

Παρόλο του ότι ο αλγόριθμος του Naive Bayes εφαρμόζει αυτή τη «naive» υπόθεση, στην πράξη καταφέρνει να επιτυγχάνει πολύ καλή ακρίβεια ταξινόμησης [10] [11]. Μπορεί να μην υπολογίζει σωστά τις πιθανότητες για κάθε κατηγορία, αλλά καταφέρνει να πάρει τη σωστή απόφαση.

3.1.3 Παράδειγμα χρήσης Naive Bayes

Στο παράδειγμα αυτό, ζητείται να ταξινομηθεί το έγγραφο «μοναδική εμπειρία συνιστώ» στην κατηγορία «Θετικό» ή στην κατηγορία «Αρνητικό», ανάλογα με το συναίσθημα που κρύβει. Το training set για την εκπαίδευση του Naive Bayes δίνεται στον πίνακα 4.

Πίνακας 4: ***Training set παραδείγματος Naive Bayes***

Έγγραφο	**Δοσμένη κατηγορία**
εξαιρετικό ξενοδοχείο	Θετικό
αξέχαστη εμπειρία συνιστώ	Θετικό
συνιστώ ανεπιφύλαχτα	Θετικό
χειρότερη εμπειρία	Αρνητικό

Με βάση το training set που δίνεται στον πίνακα 4, το λεξιλόγιο είναι:

$$V = \{εξαιρετικό, ξενοδοχείο, αξέχαστη, εμπειρία, συνιστώ, ανεπιφύλαχτα, χειρότερη\}$$

και έχει μέγεθος |V| = 7. Στη συνέχεια υπολογίζονται οι πιθανότητες των κατηγοριών χρησιμοποιώντας τη σχέση 7:

$$P(Θετικό) = \frac{3}{4}, \qquad P(Αρνητικό) = \frac{1}{4}$$

Ακολούθως, υπολογίζεται η πιθανότητα της κάθε λέξης του νέου εγγράφου και για τις δύο κατηγορίες, σύμφωνα με τη σχέση 10:

$$P\left(μοναδική/_{Θετικό}\right) = \frac{0+1}{7+7} = \frac{1}{14}, \qquad P\left(μοναδική/_{Αρνητικό}\right) = \frac{0+1}{2+7} = \frac{1}{9},$$

$$P\left(εμπειρία/_{Θετικό}\right) = \frac{1+1}{7+7} = \frac{1}{7}, \qquad P\left(εμπειρία/_{Αρνητικό}\right) = \frac{1+1}{2+7} = \frac{2}{9},$$

$$P\left(συνιστώ/_{Θετικό}\right) = \frac{2+1}{7+7} = \frac{3}{14}, \qquad P\left(συνιστώ/_{Αρνητικό}\right) = \frac{0+1}{2+7} = \frac{1}{9}$$

Τέλος, υπολογίζεται η πιθανότητα το έγγραφο d να ανήκει στις δύο κατηγορίες με χρήση της σχέσης 5:

$$P\left(Θετικό/_{d}\right) = \frac{3}{4} * \frac{1}{14} * \frac{1}{7} * \frac{3}{14} = 1.64 * 10^{-3}$$

$$P\left(Αρνητικό/_{d}\right) = \frac{1}{4} * \frac{1}{9} * \frac{2}{9} * \frac{1}{9} = 6.86 * 10^{-4}$$

Συνεπώς, σύμφωνα με τη σχέση αυτή, ο Naive Bayes θα κατατάξει το έγγραφο d ως «Θετικό». Όπως φαίνεται από αυτό το παράδειγμα, η χρήση της τροποποιημένης εξίσωσης

(10 αντί για 8) για τον υπολογισμό της πιθανότητας της κάθε λέξης του νέου εγγράφου είναι απαραίτητη.

3.1.4 Naive Bayes στο Mahout

Ο Naive Bayes είναι από τους πρώτους αλγορίθμους που υποστηρίχθηκε από το Mahout και αυτό οφείλεται στην ευκολία υλοποίησής του. Η υλοποίηση του Naive Bayes στο Mahout είναι κατανεμημένη, γεγονός που καθιστά τον αλγόριθμο κατάλληλο για έργα όπου απαιτείται να γίνει χρήση μεγάλου όγκου δεδομένων. Το Mahout παρέχει τη δυνατότητα εκτέλεσης του Naive Bayes τόσο από το τερματικό, όσο και από κώδικα [3] [12] [13].

3.1.4.1 Υλοποίηση

Η υλοποίηση του Naive Bayes στο Mahout [10] [14] [15] διαφέρει λίγο από αυτό που αναφέρθηκε πιο πάνω και αυτό οφείλεται στον τρόπο αναπαράστασης των δεδομένων που ακολουθείται στο Mahout. Συγκεκριμένα, κάθε έγγραφο αναπαρίσταται ως ένα διάνυσμα από λέξεις, σε συνδυασμό με ένα βάρος, το οποίο δείχνει πόσο αντιπροσωπευτική είναι κάθε λέξη. Ο προκαθορισμένος τρόπος υπολογισμού του βάρους κάθε λέξης ισούται με τις εμφανίσεις κάθε λέξης στο αρχικό έγγραφο (tf). Για παράδειγμα, το έγγραφο «εξαιρετικό το ξενοδοχείο το συνιστώ ανεπιφύλαχτα» αναπαριστάται ως το διάνυσμα {εξαιρετικό:1.0, το:2.0, ξενοδοχείο:1.0, συνιστώ:1.0, ανεπιφύλαχτα:1.0}. Ο υπολογισμός της πιο πιθανής κατηγορίας κατάταξης ενός νέου εγγράφου παίρνει τώρα τη μορφή:

$$l(d) = argmax_{c \in C} \left(\log P(c) + \sum_{1 \le i \le |V|} \log \left(P\left({}^{t_i}/_{c} \right) \right)^{tf_{t_i,d}} \right) \quad (11)$$

$$\Leftrightarrow \; l(d) = argmax_{c \in C} \left(\log P(c) + \sum_{1 \le i \le |V|} \mathrm{tf}_{\mathrm{t_i,d}} * \log P\left({}^{t_i}/_{c} \right) \right) \quad (12)$$

όπου το |V| είναι το μέγεθος του λεξιλογίου και $\mathrm{tft}_{\mathrm{i,d}}$ είναι ο αριθμός των εμφανίσεων της λέξης t_i στο έγγραφο d. Οι πιθανότητες P(c) και $P(t_i/c)$ υπολογίζονται όπως και πριν. Ουσιαστικά, δεν υπάρχει καμιά διαφορά. Τώρα το άθροισμα εκτελείται πάνω σ' όλο το λεξιλόγιο, αντί μόνο πάνω στις λέξεις που περιέχονται στο προς εξέταση έγγραφο. Αυτό φυσικά δεν επιφέρει κάποια διαφορά, επειδή ο όρος $P(t_i/c)$ πολλαπλασιάζεται με τις εμφανίσεις της λέξης t_i στο έγγραφο. Δηλαδή αν η λέξη t_i δεν υπάρχει στο έγγραφο, ο εν λόγω όρος δε θα συνεισφέρει τίποτα στο άθροισμα.

3.1.4.2 Χρήση Naive Bayes στο Mahout

Ο Naive Bayes στο Mahout μπορεί να χρησιμοποιηθεί είτε από τη γραμμή εντολών είτε από κώδικα. Για να μπορέσει να χρησιμοποιηθεί ο Naive Bayes στην ταξινόμηση νέων εγγράφων, θα πρέπει αρχικά να εκπαιδευτεί. Η εκπαίδευση του Naive Bayes γίνεται με το training set που του δίνεται ως είσοδος. Ο Naive Bayes, όπως και οι περισσότεροι αλγόριθμοι που προσφέρονται από το Mahout, απαιτεί τα έγγραφα που του δίνονται ως είσοδος να είναι σε μορφή διανυσμάτων.

Το Mahout προσφέρει εργαλεία για τη μετατροπή των εγγράφων σε διανύσματα που είναι κατανοητά απ' τους αλγορίθμους. Η μετατροπή των εγγράφων γίνεται σε δύο στάδια:

1. Τα έγγραφα μετατρέπονται σε SequenceFile. Το SequenceFile είναι τύπος του Hadoop που αποτελείται από keys και values. Κάθε key αντιστοιχεί σ' ένα συγκεκριμένο έγγραφο και κάθε value αντιστοιχεί στο κείμενο που περιέχει το κάθε έγγραφο. Για παράδειγμα δηλαδή, όλο το training set μετατρέπεται σ' ένα έγγραφο το οποίο περιέχει keys και values. Κάθε έγγραφο του training set αντιστοιχεί σ' ένα μοναδικό key και το κείμενο που περιέχει το έγγραφο αυτό τοποθετείται στο αντίστοιχο value πεδίο.
2. Στη συνέχεια, το SequenceFile μετατρέπεται σε vectors. Όπως αναφέρθηκε, τα vectors είναι διανύσματα από λέξεις που περιέχουν και τις εμφανίσεις των λέξεων στο κάθε έγγραφο (προκαθορισμένο βάρος). Εν ολίγοις, για κάθε έγγραφο (ένα key/value στο SequenceFile) δημιουργείται ένα διάνυσμα, το οποίο περιέχει όλες τις λέξεις που συμπεριλαμβάνονται στο συγκεκριμένο έγγραφο, μαζί με τον αριθμό των εμφανίσεων τους στο εν λόγω έγγραφο. Κατά το στάδιο αυτό, το Mahout δίνει τη δυνατότητα εφαρμογής πολλών τεχνικών βελτίωσης της ακρίβειας. Οι τεχνικές αυτές, θα αναλυθούν σε επόμενη ενότητα.

Έχοντας μετατρέψει το training set σε διανύσματα, είμαστε πλέον σε θέση να εκπαιδεύσουμε το Naive Bayes. Μετά την εκπαίδευση του Naive Bayes, εξάγεται ένα μοντέλο, το οποίο μπορεί να χρησιμοποιηθεί για την κατάταξη νέων εγγράφων σε κατηγορίες.

Στη συνέχεια, μπορούν να χρησιμοποιηθούν μέθοδοι αξιολόγησης του μοντέλου που παράχθηκε με σκοπό τον υπολογισμό της πιθανής ακρίβειας ταξινόμησης σε νέα έγγραφα. Για το σκοπό αυτό, το Mahout προσφέρει δύο διαφορετικές μεθόδους:

1. Cross-validation: Σ' αυτή την τεχνική, το training set χωρίζεται σε δύο set πριν την εκπαίδευση του Naive Bayes. Το ένα set είναι τα training data, δηλαδή τα έγγραφα που θα χρησιμοποιηθούν για την εκπαίδευση του Naive Bayes και το άλλο τα test data, δηλαδή τα έγγραφα που θα χρησιμοποιηθούν για την αξιολόγηση του μοντέλου που θα παραχθεί μετά την εκπαίδευση. Ο πιο συνηθισμένος χωρισμός του training set είναι ο 80%-20%, δηλαδή χρησιμοποιούνται 80% όλων των εγγράφων στην εκπαίδευση του ταξινομητή και 20% στην αξιολόγηση του μοντέλου. Η συγκεκριμένη τεχνική στο Mahout επιλέγει τυχαία τα training data και τα test data από το training set και αυτό έχει ως αποτέλεσμα να μην μπορούν να γίνουν συγκρίσεις μεταξύ διαφορετικών εκτελέσεων. Κρίνουμε σκόπιμο να αναφέρουμε, πως το Mahout προσφέρει και δυνατότητα μη τυχαίας επιλογής των εγγράφων, αλλά στη συγκεκριμένη έκδοση του Mahout η συγκεκριμένη δυνατότητα δε λειτουργεί σωστά.
2. Με ξεχωριστό test set: Σ' αυτή την τεχνική, πρέπει να υπάρχει διαθέσιμο ένα δεύτερο set εγγράφων, όπου στα έγγραφα να περιέχεται και η πληροφορία της κατηγορίας όπου ανήκουν. Για να εφαρμοστεί αυτή η τεχνική, πρέπει να έχει ήδη εξαχθεί το μοντέλο με τη χρήση διαφορετικού set εγγράφων. Έχοντας το μοντέλο, χρησιμοποιείται το ξεχωριστό test set και αξιολογείται η επίδοση του μοντέλου πάνω σ' αυτό. Δυστυχώς, αν και αυτή η μέθοδος είναι πολύ χρήσιμη στη σύγκριση αποτελεσμάτων μεταξύ

διαφορετικών εκτελέσεων, στη συγκεκριμένη έκδοση του Mahout δε λειτουργεί σωστά και συνεπώς πρέπει να υλοποιηθεί από το χρήστη.

Με το πέρας της διαδικασίας αξιολόγησης του μοντέλου, αν η ακρίβεια του ταξινομητή είναι η επιθυμητή, τότε μπορεί να χρησιμοποιηθεί για την επίτευξη των στόχων μας. Σε αντίθετη περίπτωση, θα πρέπει να ακολουθηθούν διαδεδομένες μέθοδοι βελτίωσης της ακρίβειας του ταξινομητή.

3.2 Random Forests

Ο αλγόριθμος Random Forests στηρίζεται στα decision trees. Σ' ένα πρώτο στάδιο λοιπόν, κρίνεται σκόπιμη η παρουσίασή τους. Στη συνέχεια, παρουσιάζεται πως ο συνδυασμός πολλών decision tree οδηγεί τελικά στον αλγόριθμο του Random Forests.

3.2.1 Decision Tree

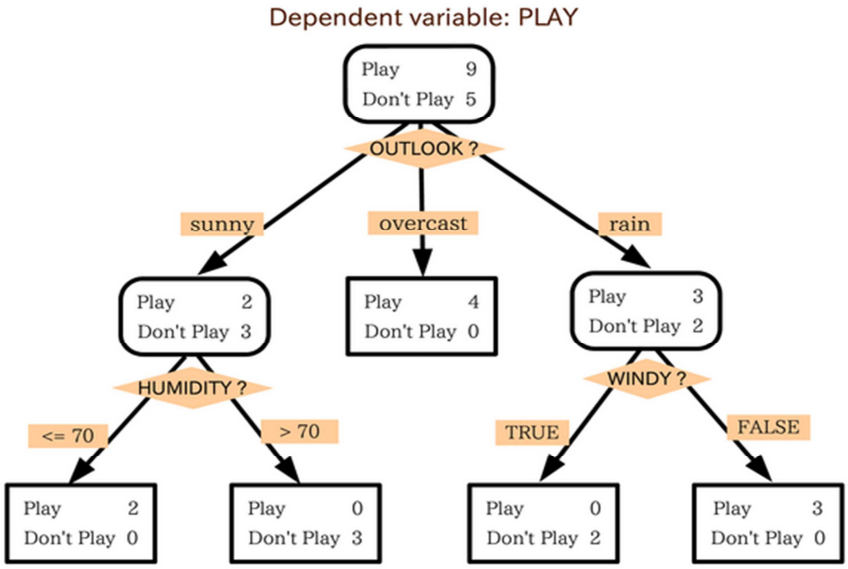
Ένα decision tree είναι ένα δέντρο το οποίο χρησιμοποιείται για ταξινόμηση. Λόγω της μη ικανοποιητικής ακρίβειας που προσφέρει δε χρησιμοποιείται πολύ συχνά ως ταξινομητής. Η χρήση όμως πολλών decision trees μαζί οδηγεί σε πολύ ισχυρούς ταξινομητές, ένας εκ των οποίων είναι και ο Random Forests.

Ένα decision tree, όπως και κάθε δέντρο, αποτελείται από ενδιαμέσους κόμβους και φύλλα. Οι ενδιάμεσοι κόμβοι αποτελούνται από ερωτήσεις, οι οποίες έχουν ως σκοπό να ξεχωρίσουν τα δεδομένα των διαφορετικών κατηγοριών και τα φύλλα να καθορίσουν την κατηγορία στην οποία ανήκουν τα δεδομένα, τα οποία κατέληξαν εκεί ακολουθώντας τις ερωτήσεις. Για να μπορέσει να γίνει κατανοητό αυτό, πιο κάτω δίνεται ένα παράδειγμα [16]. Στο παράδειγμα αυτό, θέλουμε να αποφασίσουμε αν ο καιρός είναι κατάλληλος για να παίξουμε τένις. Για την εκπαίδευση του decision tree χρησιμοποιήθηκαν 14 training data τα οποία παρουσιάζονται στον πίνακα 5.

Πίνακας 5: ***Training data για την εκπαίδευση του decision tree***

Training data	Outlook	Humidity	Wind	Play Tennis
1	Sunny	76	false	No
2	Sunny	82	true	No
3	Overcast	87	false	Yes
4	Rain	85	false	Yes
5	Rain	57	false	Yes
6	Rain	49	true	No
7	Overcast	70	false	Yes
8	Sunny	80	false	No
9	Sunny	56	false	Yes
10	Rain	62	false	Yes
11	Sunny	63	true	Yes
12	Overcast	91	true	Yes
13	Overcast	64	false	Yes
14	Rain	89	true	No

Όπως φαίνεται στον πίνακα 5, για να πάρουμε την απόφαση αν θα παίξουμε τένις ή όχι κοιτάζουμε 3 διαφορετικά χαρακτηριστικά του καιρού, την εμφάνιση του καιρού, την υγρασία και τον άνεμο. Καθένα απ' αυτά τα χαρακτηριστικά παίρνει διαφορετικές τιμές. Για παράδειγμα, ο άνεμος παίρνει τις τιμές αληθές και ψευδές. Στην εικόνα 2 φαίνεται το decision tree μετά την εκπαίδευση.

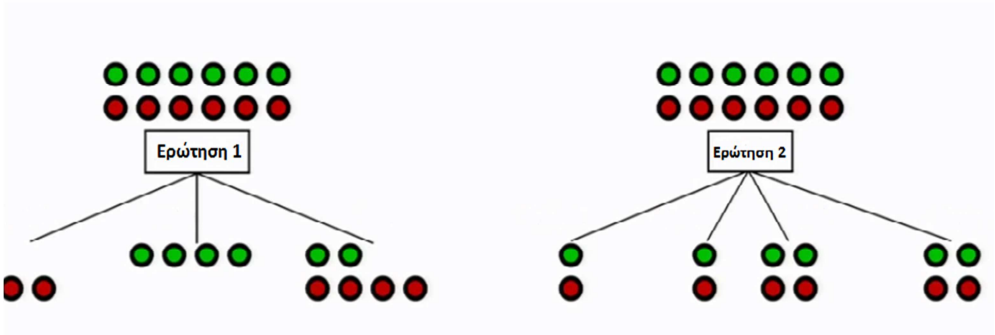


Εικόνα 2: *Εκπαιδευμένο Decision tree*

Όπως φαίνεται, οι ενδιάμεσοι κόμβοι περιέχουν ερωτήσεις οι οποίες χωρίζουν τα training data. Στην αρχή υπάρχουν συνολικά 14 training data, 9 από τα οποία αντιστοιχούν στο να παίξουμε τένις και 5 από τα οποία αντιστοιχούν στο να μην παίξουμε. Αρχικά, η ρίζα του δέντρου θέτει το ερώτημα για την εμφάνιση του καιρού, γεγονός που έχει ως αποτέλεσμα τα training data να χωριστούν σε 3 διαφορετικά υποσύνολα. Το πρώτο υποσύνολο περιέχει τα training data στα οποία ο καιρός είναι αίθριος, το δεύτερο τα training data όπου είναι συννεφιασμένος, ενώ το τρίτο τα training data όπου ο καιρός είναι βροχερός. Όπως φαίνεται από τον πίνακα 5, μόνο σε 5 από τα training data ο καιρός είναι αίθριος, μόνο σε 4 είναι νεφώδης και μόνο σε 5 είναι βροχερός. Στη συνέχεια, χρησιμοποιούνται και τα άλλα χαρακτηριστικά του καιρού για να γίνουν και άλλες ερωτήσεις και συνεπώς να χωριστούν περεταίρω τα training data. Σκοπός του αλγορίθμου εκπαίδευσης είναι να οδηγηθεί σε φύλλα, τα οποία περιέχουν training data που ανήκουν μόνο σε μια κατηγορία. Έτσι, μ' αυτό επιτυγχάνει πλήρη διαχωρισμό. Ωστόσο, υπάρχουν και περιπτώσεις όπου δεν καταφέρνει ο αλγόριθμος να διαχωρίσει τα training data πλήρως. Σ' αυτές τις περιπτώσεις, η κατηγορία με τα περισσότερα training data είναι και η πιο αντιπροσωπευτική κατηγορία. Στην εικόνα 2 παρατηρούμε ότι έγινε πλήρης διαχωρισμός.

Όταν θέλουμε να κατατάξουμε ένα νέο data στην κατηγορία στην οποία ανήκει, αυτό που έχουμε να κάνουμε είναι να ξεκινήσουμε απ' τη ρίζα του δέντρου και ανάλογα με την απάντηση που δίνεται σε κάθε ερώτηση, να ακολουθήσουμε τους κόμβους μέχρι να φτάσουμε σε φύλλο. Εκεί, καταλήγουμε στο συμπέρασμα ότι το νέο data ανήκει στην κατηγορία με τα περισσότερα training data στο φύλλο αυτό.

Εύκολα γίνεται κατανοητό πως υπάρχουν πολλά διαφορετικά δέντρα, μιας και οι ερωτήσεις μπορούν να γίνουν με οποιαδήποτε σειρά. Μερικά απ' αυτά τα δέντρα έχουν καλό διαχωρισμό και άλλα όχι. Προκειμένου λοιπόν να δημιουργείται πάντα ένα δέντρο με καλό διαχωρισμό (πιθανώς όχι τον καλύτερο, επειδή η μέθοδος που ακολουθείται είναι greedy), ο αλγόριθμος σε κάθε κόμβο κάνει όλες τις δυνατές ερωτήσεις και επιλέγει να χρησιμοποιήσει την ερώτηση που δίνει την καλύτερη διαχωρισιμότητα, δηλαδή αυτή που καταφέρνει να οδηγήσει σε όσον το δυνατό μεγαλύτερη αναλογία training data μιας κατηγορίας σε σχέση με τις υπόλοιπες. Αυτό επιτυγχάνεται υπολογίζοντας το Information Gain που προσφέρει κάθε ερώτηση [17]. Το Information Gain είναι μια τεχνική η οποία δίνει μεγάλη τιμή, όταν υπάρχει υψηλή διαχωρισιμότητα και μικρή, όταν υπάρχει χαμηλή διαχωρισιμότητα. Για παράδειγμα, στην εικόνα 3 υπάρχουν 12 διαφορετικά training data, 6 για κάθε κατηγορία (κόκκινα και πράσινα). Η ερώτηση 1 καταφέρνει να διαχωρίσει τα training data πολύ καλά, ενώ η ερώτηση 2 δεν καταφέρνει να τα διαχωρίσει καθόλου, αφού σε κάθε νέο κόμβο υπάρχει ίσος αριθμός training data από κάθε κατηγορία. Συνεπώς, ο αλγόριθμος σ' αυτό το σημείο θα επέλεγε την ερώτηση 1.



Εικόνα 3: *Διαχωρισμός training data*

3.2.2 Αλγόριθμος Random Forests

Όπως αναφέρθηκε ένα decision tree δεν αποτελεί καλό ταξινομητή. Για το λόγο αυτό, ο αλγόριθμος Random Forests κάνει χρήση πολλών δέντρων. Κάθε δέντρο μπορεί να κατασκευαστεί σε διαφορετικό υπολογιστή και συνεπώς ο αλγόριθμος είναι πλήρως κατανεμημένος. Κάθε decision tree χρησιμοποιεί μόνο ένα κομμάτι απ' τα training data (συγκεκριμένο μέγεθος που δίνεται από το χρήστη) για την εκπαίδευσή του. Τα training data όμως δε μοιράζονται στα δέντρα. Κάθε δέντρο παίρνει ένα μέρος των training data, αλλά αυτό επιλέγεται τυχαία. Μ' άλλα λόγια, μέρος των training data ενός δέντρου, ενδεχομένως να υπάρχει και σ' άλλο δέντρο. Στη συνέχεια, για την κατασκευή του δέντρου χρησιμοποιείται μόνο ένα μέρος απ' όλα τα χαρακτηριστικά των training data (π.χ. μόνο το outlook και humidity στο πιο πάνω παράδειγμα). Συγκεκριμένα, κατά τη διάρκεια κατασκευής κάθε κόμβου στο δέντρο επιλέγονται τυχαία μερικά χαρακτηριστικά (συγκεκριμένος αριθμός που δίνεται απ' το χρήστη) και απ' αυτά επιλέγεται αυτό που προσφέρει την καλύτερη διαχωρισιμότητα με βάση το υπολογιζόμενο information gain. Μετά την εκπαίδευση των δέντρων, ένα νέο data διατρέχει όλα τα δέντρα. Κάθε δέντρο δίνει μια πιθανή κατηγορία. Η κατηγορία με τις περισσότερες ψήφους είναι και η πιθανή κατηγορία του data αυτού.

Αν και είναι δύσκολο στην κατανόηση, η τυχαία επιλογή του training υποσυνόλου που θα χρησιμοποιηθεί για την κατασκευή κάθε δέντρου και η τυχαία επιλογή του υποσυνόλου των χαρακτηριστικών που θα ελεγχθεί σε κάθε κόμβο του δέντρου, έχει αποδειχθεί [18] ότι συμβάλουν στη βελτίωση της ακρίβειας. Οι τεχνικές αυτές ονομάζονται bootstrap aggregating και random subspace method αντίστοιχα και χρησιμοποιούνται πολύ στο machine learning.

Παρά το γεγονός ότι χρησιμοποιείται μόνο ένα υποσύνολο των χαρακτηριστικών της εισόδου για εύρεση του καλύτερου χωρίσματος σε κάθε κόμβο, αυτό οδηγεί σε μεγάλη υπολογιστική πολυπλοκότητα, μιας και σε κάθε κόμβο υπολογίζεται το information gain για κάθε τέτοιο χαρακτηριστικό. Αν και πλήρως κατανεμημένος αλγόριθμος, η χρήση του με πάνω από 10 εκατομμύρια εκπαιδευτικά παραδείγματα, έχει ως αποτέλεσμα μεγάλους χρόνους εκπαίδευσης.

4

Μέθοδοι βελτίωσης της ακρίβειας του ταξινομητή

Όπως αναφέρθηκε και στην εισαγωγή αυτής της εργασίας, η ακρίβεια σε μια εφαρμογή αποτελεί καθοριστικό παράγοντα. Η επιλογή ενός καλού ταξινομητή και μόνο, δεν προσφέρει την επιθυμητή ακρίβεια. Συγκεκριμένα, για να επιτύχουμε την ακρίβεια που επιθυμούμε, θα πρέπει αρχικά να επιλέξουμε το κατάλληλο training set και στη συνέχεια να εφαρμόσουμε γνωστές τεχνικές βελτίωσης της ακρίβειας. Η επιλογή του κατάλληλου training set είναι καθοριστική και πρέπει να γίνεται με προσοχή. Χωρίς ένα κατάλληλο training set, οι τεχνικές βελτίωσης της ακρίβειας δεν προσφέρουν τα αναμενόμενα αποτελέσματα.

4.1 Επιλογή κατάλληλου training set

Για την επιλογή κατάλληλου training set πρέπει να προσεχτεί ιδιαίτερα το μέγεθος, το περιεχόμενο και η ποσότητα των εγγράφων που υπάρχουν στο training set ανά κατηγορία ταξινόμησης.

4.1.1 Μέγεθος training set

Το μέγεθος του training set παίζει καθοριστικό παράγοντα στην ακρίβεια του ταξινομητή. Συγκεκριμένα, όσα περισσότερα εκπαιδευτικά έγγραφα χρησιμοποιηθούν κατά τη διάρκεια εκπαίδευσης του ταξινομητή, τόσο μεγαλύτερη γνώση αποκτά για τη συγκεκριμένη εφαρμογή, με αποτέλεσμα τα νέα έγγραφα να κατατάσσονται από τον ταξινομητή στη σωστή κατηγορία με περισσότερη ακρίβεια. Επίσης, λόγω της διαφορετικότητας που παρουσιάζει κάθε εκπαιδευτικό έγγραφο η γνώση του ταξινομητή διευρύνεται, με αποτέλεσμα να μπορεί να κατατάξει ένα μεγαλύτερο φάσμα από νέα έγγραφα στη σωστή κατηγορία. Ωστόσο, η χρήση περισσότερων εκπαιδευτικών εγγράφων δεν οδηγεί πάντα σε καλύτερη ακρίβεια. Μπορούν να οδηγήσουν σε υπερεκπαίδευση του ταξινομητή, δηλαδή να υπολογιστούν με τέτοιο τρόπο οι παράμετροί του που και η παραμικρή διαφορά ανάμεσα στα training και test data να οδηγήσει σε λάθος ταξινόμηση.

Δυστυχώς, δεν υπάρχει ειδική μεθοδολογία για την επιλογή του μεγέθους του training set. Εκτός από μερικές προφανείς περιπτώσεις, στις περισσότερες περιπτώσεις απαιτείται να γίνουν πολλές μετρήσεις για την επιλογή του κατάλληλου μεγέθους.

4.1.2 Περιεχόμενο training set

Ένας εξίσου σημαντικός παράγοντας είναι και η επιλογή ενός καλού training set. Το training set που διαλέγεται πρέπει να είναι αντιπροσωπευτικό για την εφαρμογή, έτσι ώστε ο ταξινομητής να αρχικοποιήσει κατάλληλα τα βάρη του και να είναι σε θέση να γενικεύσει ικανοποιητικά καλά. Αν το training set δεν είναι αντιπροσωπευτικό, τότε θα είναι αδύνατο για τον ταξινομητή να μπορέσει να ταξινομήσει ένα νέο έγγραφο με ακρίβεια, μιας και η γνώση του δεν είναι η κατάλληλη για να μπορεί να κάνει μια τέτοια ταξινόμηση.

Φανταστείτε τον εαυτό σας να προσπαθεί να ταξινομήσει έγγραφα σε κατηγορίες, χωρίς να έχει την κατάλληλη γνώση για το θέμα το οποίο αναφέρουν τα έγγραφα αυτά. Αυτό θα είχε ως αποτέλεσμα, πολλά έγγραφα να καταταχτούν σε λάθος κατηγορία. Έτσι και οι ταξινομητές αν δεν εκπαιδευτούν με αντιπροσωπευτικά εκπαιδευτικά έγγραφα για να αρχικοποιηθούν κατάλληλα τα βάρη τους, δε θα είναι σε θέση να ταξινομήσουν με ικανοποιητική ακρίβεια τα νέα έγγραφα που τους δίνονται.

Για την επιλογή ενός καλού training set πρέπει να μελετηθεί σε μεγάλο βαθμό η εφαρμογή που θέλουμε να χρησιμοποιήσουμε το ταξινομητή, έτσι ώστε να επιλεγούν τα πιο αντιπροσωπευτικά training data από τα δεδομένα που υπάρχουν για την εφαρμογή. Συνήθως σε μια εφαρμογή υπάρχουν πολλά δεδομένα και απ' αυτά διαλέγονται μόνο μερικά ως τα πιο αντιπροσωπευτικά. Αν διαλέγονταν όλα, αυτό θα ήταν σπατάλη χώρου, χρόνου και ακρίβειας.

4.1.3 Αριθμός εκπαιδευτικών εγγράφων ανά κατηγορία

Όπως παρουσιάστηκε πιο πάνω, ο υπολογισμός της κατηγορίας που θα καταταχθεί ένα νέο έγγραφο δίνεται από τη σχέση:

$$l(d) = argmax_{c \in C} \left(\log P(c) + \sum_{1 \le i \le |V|} \mathrm{tf}_{\mathrm{t_i,d}} * \log P(\,{}^{t_i}/_{c}) \right) \quad (13)$$

Σύμφωνα με τη σχέση 13, σημαντικό ρόλο στην ταξινόμηση νέων εγγράφων διαδραματίζει η προγενέστερη πιθανότητα της κατηγορίας. Η προγενέστερη πιθανότητα της κατηγορίας δίνεται από τη σχέση:

$$P(c) = \frac{training\ documents\ of\ category\ c}{all\ documents} \quad (14)$$

Ουσιαστικά, η προγενέστερη πιθανότητα της κατηγορίας δίνει ένα βάρος στην κάθε κατηγορία και αυτό το βάρος αντιστοιχεί στο πόσο πιθανή είναι η κατηγορία αυτή. Συνεπώς, ο αριθμός των training data που δίνονται ανά κατηγορία πρέπει να διαλέγεται με προσοχή, ώστε να αντιστοιχεί στην πιθανότητα που θέλουμε να έχει κάθε κατηγορία.

Για παράδειγμα, έχουμε τις κατηγορίες «Θετικό» και «Αρνητικό» και επίσης πιστεύουμε πως κάθε νέο έγγραφο είναι ισοπίθανο να είναι Θετικό ή Αρνητικό. Αυτό σημαίνει ότι πρέπει να δώσουμε ίσα training data για την κάθε κατηγορία, ώστε η προγενέστερη πιθανότητα της κατηγορίας «Θετικό» να είναι ίση με τη προγενέστερη πιθανότητα της κατηγορίας «Αρνητικό». Αν για κάποιο λόγο δίνονταν περισσότερα training data της κατηγορίας «Θετικό», τότε η προγενέστερη πιθανότητα της κατηγορίας «Θετικό» θα ήταν μεγαλύτερη από αυτήν της κατηγορίας «Αρνητικό» και σύμφωνα με τη σχέση 13, ο Naive Bayes θα έτεινε στο να διαλέξει την κατηγορία «Θετικό», γεγονός που θα ήταν ανεπιθύμητο σύμφωνα με τα στοιχεία του παραδείγματος.

Αν για κάποιο λόγο δεν μπορούν να υπάρχουν ικανοποιητικά πολλά και με τη σωστή αναλογία training data για τις κατηγορίες, τότε θα πρέπει να ανατρέξουμε στο Complementary Naive Bayes [14] ή σε άλλους δημοφιλής τρόπους επιλύσεις του προβλήματος αυτού [19] [20].

4.2 Τεχνικές βελτίωσης ακρίβειας

Έχοντας επιλέξει κατάλληλο training set μπορούμε να προχωρήσουμε στην εφαρμογή τεχνικών βελτίωσης της ακρίβειας του ταξινομητή. Οι τεχνικές που παρουσιάζονται παρακάτω είναι οι πιο συνηθισμένες και εφαρμόζονται κατά κόρο στην ταξινόμηση κειμένων [3] [10] [14] [21]. Κρίνουμε σκόπιμο να αναφέρουμε ότι το Mahout προσφέρει πολλές από αυτές, γεγονός που διευκολύνει πολύ το έργο μας, αφού δε χρειάζεται να σπαταλήσουμε πολύτιμο χρόνο στην υλοποίησή τους [3]. Η εφαρμογή των τεχνικών αυτών δεν εγγυάται τη βελτίωση της ακρίβειας, αλλά τις περισσότερες φορές το επιτυγχάνει. Ο λόγος που δε δίνουν πάντα τα καλύτερα αποτελέσματα είναι επειδή εξαρτούνται άμεσα από την κατάσταση του training set.

4.2.1 Μετατροπή γραμμάτων σε πεζά και αφαίρεση συμβόλων

Γενικά, αυτές οι δύο τεχνικές είναι πολύ διαδεδομένες και χρησιμοποιούνται κατά κόρο στο machine learning. Εφαρμόζονται τόσο στα training data, όσο και στα test data και έχουν ως σκοπό να κάνουν τις λέξεις συγκρίσιμες μεταξύ τους.

Η πρώτη τεχνική αφορά τα πεζά γράμματα, δηλαδή όλα τα γράμματα στα training και test data μετατρέπονται σε πεζά, ώστε να μπορούν να συγκρίνονται οι λέξεις μεταξύ τους. Για παράδειγμα, αν δε μετατρέπονταν όλα τα γράμματα σε πεζά, οι λέξεις «Γιάννης» και «γιάννης» θα ήταν διαφορετικές. Αυτό στις περισσότερες περιπτώσεις θα οδηγούσε σε χαμηλότερη ακρίβεια, αφού κατά την ταξινόμηση ενός νέου εγγράφου δε θα λαμβάνονταν υπόψη λέξεις που είναι ίδιες, αλλά διαφέρουν λόγω διαφοράς στα πεζά/κεφαλαία.

Η δεύτερη τεχνική αναφέρεται στα σύμβολα, τα οποία ενδεχομένως να δυσκολεύουν τη σύγκριση μεταξύ των λέξεων. Για παράδειγμα, η πρόταση «ένα δύο ένα!» θα μετατρεπόταν στο διάνυσμα {ένα:1.0, δύο:1.0, ένα!:1.0} που είναι λάθος, μιας και το «ένα» και «ένα!» είναι η ίδια λέξη. Ωστόσο, κρίνουμε σκόπιμο να αναφέρουμε πως σε συγκεκριμένες περιπτώσεις μερικά σύμβολα παίζουν σημαντικό ρόλο και δεν πρέπει να αφαιρούνται. Μια τέτοια

περίπτωση είναι τα emoticons που χρησιμοποιούνται ευρύτατα στα κοινωνικά δίκτυα και τα οποία παρέχουν μεγάλη πληροφορία στην ταξινόμηση [22].

Γενικά, ακόμη και να μη βοηθάνε οι συγκεκριμένες τεχνικές στη βελτίωση της ακρίβειας του ταξινομητή πρέπει να χρησιμοποιούνται, μιας και δε γνωρίζουμε τη μορφή των νέων εγγράφων (με εξαίρεση την αφαίρεση μερικών συμβόλων). Με τις τεχνικές αυτές, ότι μορφή και να έχουν τα νέα έγγραφα, θα είναι συγκρίσιμα με τα έγγραφα που χρησιμοποιήθηκαν για την εκπαίδευση του ταξινομητή, αφού τόσο τα νέα έγγραφα, όσο και τα training data υπόκεινται των συγκεκριμένων επεξεργασιών. Μ' άλλα λόγια, αν δε χρησιμοποιηθούν οι συγκεκριμένες τεχνικές τα αποτελέσματα που θα παρθούν από την αξιολόγηση θα είναι παραπλανητικά, μιας και τα πραγματικά δεδομένα μπορεί να έχουν διαφορετική μορφή και συνεπώς στην πράξη το μοντέλο δε θα αποδίδει τόσο καλά. Εφαρμόζοντας τις τεχνικές αυτές τόσο στα training, όσο και στα test data, δίνεται η δυνατότητα για δημιουργία μιας καλύτερης εικόνας της ακρίβειας του ταξινομητή σε νέα άγνωστα δεδομένα.

Οι συγκεκριμένες τεχνικές υποστηρίζονται από το Mahout και χρησιμοποιούνται κατά τη μετατροπή των εγγράφων σε διανύσματα. Το Mahout υποστηρίζει αρκετούς αναλυτές κειμένου και ο καθένας επεξεργάζεται το κείμενο διαφορετικά. Για παράδειγμα, ο StandardAnalyzer μετατρέπει τα γράμματα σε πεζά και αφαιρεί και όλα τα σύμβολα, με εξαίρεση αυτών που βρίσκονται ανάμεσα σε γράμματα. Αντίθετα, ο WhitespaceAnalyzer δεν κάνει τίποτα από αυτά.

Κρίνεται σκόπιμο να αναφερθεί ότι στις περιπτώσεις που δε μας ικανοποιούν οι αναλυτές κειμένου που προσφέρει το Mahout, μπορούμε να προγραμματίσουμε το δικό μας αναλυτή. Η εφαρμογή του όμως στα έγγραφα πρέπει να γίνει πριν τα έγγραφα να δοθούν στο Mahout για μετατροπή σε διανύσματα.

4.2.2 Αφαίρεση αριθμών

Ένα συχνό χαρακτηριστικό των εγγράφων είναι ότι περιέχουν πολλούς αριθμούς. Στο κάθε έγγραφο όμως, ο κάθε αριθμός μπορεί να έχει διαφορετική σημασία. Για παράδειγμα, σ' ένα έγγραφο ο αριθμός 5 μπορεί να εννοεί την ώρα, ενώ σε κάποιο άλλο μπορεί να εννοεί το βαθμό που δίνει στο ξενοδοχείο ένας χρήστης. Ο ταξινομητής όμως δε μπορεί να καταλάβει το νόημα που κρύβεται πίσω απ' τους αριθμούς και έτσι όταν προσπαθεί να ταξινομήσει ένα νέο έγγραφο, αυτό πιθανώς να οδηγήσει σε λάθος συμπεράσματα, αφού θα ληφθούν υπόψη αριθμοί που δεν έχουν το ίδιο νόημα.

Αυτή η τεχνική υποστηρίζεται από το Mahout και μπορεί να χρησιμοποιηθεί κατά τη μετατροπή των εγγράφων σε διανύσματα. Φυσικά, αν θα γίνει ή όχι αφαίρεση των αριθμών, αυτό θα εξαρτηθεί από τον αναλυτή κειμένου που θα χρησιμοποιηθεί.

Όπως προαναφέρθηκε, αν για κάποιο λόγο δε μας ικανοποιούν οι αναλυτές κειμένου που προσφέρει το Mahout, μπορούμε να προγραμματίσουμε το δικό μας αναλυτή, αλλά η εφαρμογή του πρέπει να γίνει πριν τα έγγραφα δοθούν στο Mahout για μετατροπή σε διανύσματα.

Η συγκεκριμένη τεχνική εφαρμόζεται στα training data. Κρίνεται όμως σκόπιμο να αναφερθεί ότι στις περιπτώσεις στις οποίες γίνεται αφαίρεση αριθμών που περιέχονται σε λέξεις, όπως για παράδειγμα το «test1» μετατρέπεται στο «test», η συγκεκριμένη τεχνική θα πρέπει να εφαρμόζεται και στα test data προκειμένου να έχει νόημα η αφαίρεση.

4.2.3 Αφαίρεση stopwords με χρήση λιστών

Ως stopwords ορίζονται όλες οι πολύ συχνά χρησιμοποιούμενες λέξεις που δεν προσφέρουν όμως κάποια πληροφορία στην ταξινόμηση. Μερικές απ' αυτές τις λέξεις είναι οι «and», «or», «the» και «of». Ο ταξινομητής δεν πρέπει να βασίζεται πάνω σ' αυτές για να βγάλει το συμπέρασμα της κατηγορίας, αφού αυτές οι λέξεις δε δίνουν ένδειξη για την κατηγορία και οδηγούν τον ταξινομητή σε λάθη. Επίσης, λόγω της μεγάλης συχνότητάς τους, κατέχουν μεγάλο χώρο στα training data και αυτό οδηγεί τόσο σε σπατάλη χώρου, όσο και χρόνου. Αν και μερικές από αυτές τις λέξεις είναι γνωστές (όπως αυτές που αναφέρθηκαν πιο πάνω), δεν υπάρχει μια λίστα που να περιέχει όλα τα stopwords. Κι αυτό γιατί τα stopwords δεν εξαρτούνται μόνο από τη συχνότητα χρήσης κάποιων λέξεων στην κάθε γλώσσα, αλλά εξαρτούνται και από την εφαρμογή. Για παράδειγμα, όταν έχουμε να ταξινομήσουμε έγγραφα που αναφέρονται σε αυτοκίνητα σε διαφορετικές μάρκες αυτοκινήτων, η λέξη «αυτοκίνητο» θα συμπεριφέρεται ως stopword, μιας και θα εμφανίζεται σχεδόν παντού και δε θα δίνει κάποια πληροφορία στην ταξινόμηση.

Παρόλο που δεν υπάρχει μια λίστα που να περιέχει όλα τα stopwords, συνηθίζεται να χρησιμοποιούνται λίστες που περιέχουν αρκετά γνωστά stopwords. Η εφαρμογή τέτοιων λιστών πρέπει να γίνεται με προσοχή, αφού στην εφαρμογή μας κάποιες από τις λέξεις που υπάρχουν στις λίστες αυτές, μπορεί να μη συμπεριφέρονται ως stopwords και αυτό να έχει ως αποτέλεσμα τη μείωση της ακρίβειας.

Το Mahout υποστηρίζει αυτή την τεχνική και εφαρμόζεται κατά τη μετατροπή των εγγράφων σε διανύσματα. Δεν υποστηρίζεται όμως απ' όλους τους αναλυτές κειμένου που υπάρχουν στο Mahout. Για παράδειγμα, ο StandardAnalyzer περιέχει μια προκαθορισμένη λίστα με stopwords, η οποία μπορεί να επεκταθεί ή να αλλαχτεί μόνο προγραμματιστικά, ενώ ο WhitespaceAnalyzer δεν υποστηρίζει κάτι τέτοιο.

Στην περίπτωση που επιθυμούμε την εφαρμογή της δικής μας λίστας, μπορούμε είτε να χρησιμοποιήσουμε τον StandardAnalyzer στο κώδικα μας δίνοντας του τη λίστα με τα stopwords που θέλουμε να αφαιρέσει, είτε να προγραμματίσουμε το δικό μας αναλυτή. Στη δεύτερη περίπτωση, αυτό πρέπει να γίνει πριν τα έγγραφα δοθούν στο Mahout για μετατροπή σε διανύσματα.

Η τεχνική αυτή εφαρμόζεται στα training data. Δεν είναι υποχρεωτικό να εφαρμοστεί αυτή η τεχνική και στα test data, μιας και αν ο εκπαιδευμένος ταξινομητής δε γνωρίζει γι' αυτές τις λέξεις, τότε απλά είναι σαν και τις αγνοεί κατά τη διάρκεια της ταξινόμησης νέων εγγράφων.

4.2.4 Tf-idf

Όταν γίνεται μετατροπή των εγγράφων σε διανύσματα, το πιο συνηθισμένο είναι το βάρος της κάθε λέξης σε κάθε έγγραφο να ισούται με τις εμφανίσεις της λέξης στο αντίστοιχο έγγραφο (term frequency (tf)). Αυτό έχει ως αποτέλεσμα, οι πολύ χρησιμοποιούμενες λέξεις, δηλαδή τα stopwords, να έχουν το μεγαλύτερο βάρος και να θεωρούνται έτσι ως οι πιο αντιπροσωπευτικές λέξεις των εγγράφων. Μια λύση σε αυτό το πρόβλημα είναι αντί το βάρος της κάθε λέξης να είναι το term frequency (tf), να είναι το term frequency – inverse document frequency (tf-idf). Τώρα το βάρος της κάθε λέξης υπολογίζεται σύμφωνα με την εξής σχέση:

$$W_i = tf_i * idf_i = tf_i * \log \frac{all\ documents}{df_i} \quad (15)$$

όπου το tf_i είναι η συχνότητα της λέξης στο έγγραφο και df_i είναι ο αριθμός των διαφορετικών εγγράφων του training set που υπάρχει η συγκεκριμένη λέξη. Αυτό που κάνει το tf-idf είναι να πολλαπλασιάζει τον όρο tf με το όρο idf. Ο όρος idf παίρνει μεγάλες τιμές, όταν μια λέξη υπάρχει σε λίγα έγγραφα και μικρές τιμές, όταν η λέξη υπάρχει σε πολλά έγγραφα. Ουσιαστικά δηλαδή, λέξεις που παρουσιάζουν μεγάλη συχνότητα πολλαπλασιάζονται με μικρό idf, ώστε να μικρύνει το βάρος τους και λέξεις που είναι σπάνιες πολλαπλασιάζονται με μεγάλο idf, ώστε να αυξηθεί το βάρος τους. Αυτό έχει ως αποτέλεσμα, τα stopwords να παίρνουν σχετικά μικρό βάρος και να μην αποτελούν πλέον τις πιο αντιπροσωπευτικές λέξεις στα έγγραφα. Αν και η τεχνική μπορεί να δώσει καλύτερη ακρίβεια στις περισσότερες περιπτώσεις, υπάρχουν και περιπτώσεις που τη χειροτερεύει. Μια τέτοια περίπτωση είναι όταν οι λέξεις με τις λιγότερες εμφανίσεις σε διαφορετικά έγγραφα δεν είναι αντιπροσωπευτικές των κατηγοριών. Αυτό οδηγεί στο να θεωρούνται σημαντικές μη αντιπροσωπευτικές λέξεις. Επίσης, μια άλλη περίπτωση είναι όταν οι αντιπροσωπευτικές λέξεις των κατηγοριών εμφανίζονται παρόμοιες φορές σε διαφορετικά έγγραφα με τα stopwords (όρος df). Αν συμβαίνει αυτό, οι αντιπροσωπευτικές λέξεις θα πάρουν πάλι χαμηλότερη τιμή από τα stopwords, μιας και τις περισσότερες φορές τα stopwords έχουν περισσότερες εμφανίσεις σ' όλα τα έγγραφα από τις αντιπροσωπευτικές λέξεις (όρος tf).

Αυτή η τεχνική υποστηρίζεται από το Mahout. Γενικά, το Mahout δίνει τη δυνατότητα επιλογής μεταξύ tf και tf-idf βαρών κατά τη διάρκεια μετατροπής των εγγράφων σε διανύσματα. Η τεχνική αυτή μπορεί να εφαρμοστεί και στα training και στα test data. Δεν είναι όμως υποχρεωτικό όταν εφαρμοστεί στα training data, να εφαρμοστεί και στα test data και αντίστροφα. Για παράδειγμα, τα test data μπορεί να χρησιμοποιούν την τεχνική tf για τον υπολογισμό του βάρους των λέξεων, ενώ τα training data την τεχνική tf-idf. Στις περισσότερες όμως περιπτώσεις, εφαρμόζεται η ίδια τεχνική και στα training και στα test data, ώστε να μη χάνει το νόημά της η τεχνική.

4.2.5 Κανονικοποίηση

Η κανονικοποίηση εφαρμόζεται όταν υπάρχουν έγγραφα διαφορετικού μεγέθους. Το να εμφανίζεται μια φορά μια λέξη σε ένα μεγάλο έγγραφο, δεν είναι το ίδιο με το να εμφανίζεται μια φορά μια λέξη σ' ένα μικρό έγγραφο. Όταν εμφανίζεται μια φορά μια λέξη σε ένα μεγάλο έγγραφο, αυτό δεν την κάνει τόσο αντιπροσωπευτική, όσο το να εμφανίζεται μια φορά μια λέξη σε ένα μικρό έγγραφο. Με άλλα λόγια, πρέπει να παίζει ρόλο το μέγεθος του εγγράφου, όταν υπολογίζεται το βάρος μιας λέξης και υπάρχουν διαφορετικού μεγέθους έγγραφα.

Αυτή η τεχνική υποστηρίζεται από το Mahout. Το Mahout προσφέρει την p-νόρμα. Για παράδειγμα, η p-νόρμα ενός διανύσματος με τρία στοιχεία x, y, z δίνεται από τον πιο κάτω τύπο:

$$\frac{x}{(x^p + y^p + z^p)^{1/p}}, \frac{y}{(x^p + y^p + z^p)^{1/p}}, \frac{z}{(x^p + y^p + z^p)^{1/p}} \quad (16)$$

Πολύ γνωστή νόρμα είναι η Manhattan όπου είναι η 1-νόρμα και υπολογίζεται ως εξής:

$$\frac{x}{|x| + |y| + |z|}, \frac{y}{|x| + |y| + |z|}, \frac{z}{|x| + |y| + |z|} \quad (17)$$

Επίσης πολύ γνωστή νόρμα είναι και η Ευκλείδεια, όπου είναι η 2-νόρμα και υπολογίζεται ως εξής:

$$\frac{x}{\sqrt{x^2 + y^2 + z^2}}, \frac{y}{\sqrt{x^2 + y^2 + z^2}}, \frac{z}{\sqrt{x^2 + y^2 + z^2}} \quad (18)$$

Τέλος η infinite norm είναι εξίσου γνωστή νόρμα και υπολογίζεται ως εξής:

$$\frac{x}{\max(|x|, |y|, |z|)}, \frac{y}{\max(|x|, |y|, |z|)}, \frac{z}{\max(|x|, |y|, |z|)} \quad (19)$$

Αυτή η τεχνική εφαρμόζεται κατά τη διάρκεια μετατροπής των εγγράφων σε διανύσματα και εφαρμόζεται μόνο στα training data. Δεν έχει νόημα να εφαρμοστεί στα test data. Εφαρμόζεται μόνο όταν χρησιμοποιείται η τεχνική tf-idf (περιορισμός του Mahout) και εφαρμόζεται μετά την εφαρμογή της.

4.2.6 N-gram

Ένα συνηθισμένο λάθος που γίνεται κατά την κατασκευή των διανυσμάτων είναι να θεωρείται πως κάθε λέξη είναι ανεξάρτητη από τις υπόλοιπες λέξεις. Αυτό όμως στη πραγματικότητα δεν είναι αληθές, μιας και οι περισσότερες λέξεις εξαρτούνται από τις άλλες. Για παράδειγμα, η «Coca Cola» θα θεωρείται ως δύο λέξεις, τη λέξη «Coca» και τη λέξη «Cola». Στην πραγματικότητα όμως, η μια λέξη εξαρτάται απ' την άλλη και είναι σχεδόν αδύνατο να παρουσιαστεί η μια χωρίς την άλλη. Όταν θεωρείται πως η κάθε λέξη είναι ανεξάρτητη από τις υπόλοιπες, τα διανύσματα που κατασκευάζονται δεν αντιπροσωπεύουν ικανοποιητικά τα έγγραφα απ' τα οποία προήλθαν. Για παράδειγμα, θέλουμε να ταξινομήσουμε έγγραφα ως

«Θετικά» και «Αρνητικά» και έστω το Θετικό έγγραφο «πολύ ωραίο ξενοδοχείο» είναι ένα από τα έγγραφα που θα χρησιμοποιηθούν για την εκπαίδευση του ταξινομητή. Αν για την κατασκευή των διανυσμάτων θεωρηθεί πως η κάθε λέξη είναι ανεξάρτητη από την άλλη, τότε θα πάρουμε το εξής διάνυσμα {πολύ:1.0, ωραίο:1.0, ξενοδοχείο:1.0} για το συγκεκριμένο έγγραφο. Η μόνη λέξη που μπορεί προσφέρει κάποια πληροφορία για την κατηγορία είναι η λέξη «ωραίο». Οι λέξεις «πολύ» και «ξενοδοχείο» από μόνες τους δεν προσφέρουν καμιά πληροφορία για την κατηγορία και πολύ πιθανό να υπάρχουν και στα έγγραφα εκπαίδευσης της «Αρνητικής» κατηγορίας. Αυτό θα έχει ως αποτέλεσμα, ο ταξινομητής να μην είναι τόσο ακριβής, μιας και κατά τη διάρκεια ταξινόμησης ενός νέου εγγράφου, λέξεις όπως το «πολύ» και «ξενοδοχείο», θα τον οδηγήσουν σε πιθανό λάθος. Από την άλλη, αν για την κατασκευή των διανυσμάτων χρησιμοποιηθούν 2-grams, τότε το διάνυσμα θα είναι το {πολύ καλό:1.0, καλό ξενοδοχείο:1.0}, όπου και τα δύο 2-grams δίνουν την πληροφορία της κατηγορίας και συνεπώς δε θα οδηγούσαν τον ταξινομητή σε λάθος απόφαση.

Το Mahout δίνει τη δυνατότητα κατασκευής n-grams οποιουδήποτε μεγέθους. Αυτό μπορεί να γίνει τόσο προγραμματιστικά, όσο και από τη γραμμή εντολών κατά την κατασκευή των διανυσμάτων. Το Mahout συγκεκριμένα κατασκευάζει n-grams μέχρι το όρισμα που του δίνεται ως είσοδος. Για παράδειγμα αν του δοθεί το όρισμα 3, τότε τα διανύσματα που θα κατασκευαστούν από τα έγγραφα, θα περιέχουν 1-gram, 2-grams και 3-grams. Δηλαδή, θα περιέχουν τις λέξεις των εγγράφων μία προς μία, τις λέξεις των εγγράφων ανά δύο και τις λέξεις των εγγράφων ανά τρεις. Αυτό βοηθάει, γιατί δεν είναι όλες οι λέξεις εξαρτημένες μεταξύ τους ανά ν-άδες. Άλλες λέξεις είναι ανεξάρτητες από τις υπόλοιπες και αντιπροσωπεύουν την κατηγορία από μόνες τους και άλλες είναι εξαρτημένες και πρέπει να είναι σε ν-άδες για να αντιπροσωπεύουν την κατηγορία. Φυσικά, περνώντας όλα τα n-grams μέχρι ένα συγκεκριμένο n, θα ήταν σπατάλη χώρου και ακρίβειας, αφού δεν προσφέρουν όλα τα n-grams πληροφορία. Για το λόγο αυτό, το Mahout προσφέρει μια τεχνική τη λεγόμενη log-likelihood ratio (LLR), όπου κρατάει τα πιο σημαντικά grams μόνο, δηλαδή αυτά που προσφέρουν την περισσότερη πληροφορία. Ανάλογα με τον αριθμό που της δίνεται ως είσοδος, διαγράφει και τα ανάλογα grams. Αν της δοθεί μεγάλος αριθμός (πάνω από 100) αφήνει μόνο τα σημαντικότερα, ενώ αν της δοθεί μικρότερος, αφήνει και μη σημαντικά grams στα διανύσματα.

Η συγκεκριμένη τεχνική εφαρμόζεται κατά τη διάρκεια μετατροπής των εγγράφων σε διανύσματα. Πρέπει να εφαρμόζεται τόσο στα training, όσο και στα test data, ώστε να μπορεί να υπάρχει η απαιτούμενη σύγκριση κατά τη διάρκεια της εφαρμογής του ταξινομητή σε νέα έγγραφα.

4.2.7 Stemming

Μια άλλη γνωστή τεχνική που χρησιμοποιείται ευρύτατα είναι το stemming. Σκοπός της τεχνικής αυτής είναι να κάνει τις λέξεις συγκρίσιμες, αφήνοντας από κάθε λέξη μόνο τη ρίζα της. Για παράδειγμα, οι λέξεις «ψάρι», «ψαράκι» και «ψαράς» προέρχονται απ' την ίδια λέξη, τη λέξη «ψάρι». Αν δε χρησιμοποιηθεί κάποιος αλγόριθμος stemming, οι πιο πάνω λέξεις θα θεωρούνταν διαφορετικές και έτσι δε θα μπορούσε να γίνει σύγκριση μεταξύ τους. Το stemming σε γενικές γραμμές βοηθάει στη βελτίωση της ακρίβειας, αλλά υπάρχουν και

περιπτώσεις στις οποίες αυτή χειροτερεύει. Ένας σημαντικός λόγος που μπορεί να οδηγήσει σε μείωση της ακρίβειας είναι διότι οι αλγόριθμοι που κάνουν stemming μετατρέπουν λέξεις με διαφορετικές ρίζες στην ίδια λέξη και αυτό οδηγεί τους ταξινομητές στο να πάρουν λάθος αποφάσεις. Επίσης, μπορεί να οδηγήσει σε υπερεκπαίδευση του ταξινομητή.

Δυστυχώς, η πιο πάνω τεχνική δεν υποστηρίζεται από το Mahout. Αυτό δεν είναι πρόβλημα όμως, μιας και υπάρχουν πολλοί αλγόριθμοι που κάνουν stemming και διατίθεται ο κώδικας τους ελεύθερα. Ένας από τους πιο γνωστούς είναι ο Porter stemming algorithm. Υπάρχουν πολλές υλοποιήσεις του συγκεκριμένου αλγορίθμου που διανέμονται ελεύθερα. Ο Martin Porter, ο άνθρωπος που έγραψε τον αλγόριθμο αυτό, διαθέτει στην προσωπική του ιστοσελίδα υλοποίηση του αλγορίθμου του σε πολλές γλώσσες προγραμματισμού και ο κώδικας είναι ελεύθερος προς χρήση από οποιονδήποτε [23].

Η συγκεκριμένη τεχνική εφαρμόζεται πριν τα έγγραφα δοθούν στο Mahout για μετατροπή σε διανύσματα. Πρέπει να εφαρμόζεται τόσο στα training, όσο και στα test data, ώστε να μπορεί να υπάρχει η απαιτούμενη σύγκριση κατά τη διάρκεια της εφαρμογής του ταξινομητή σε νέα έγγραφα.

4.2.8 Ελάχιστες εμφανίσεις

Ένα πρόβλημα που δημιουργείται είναι ότι πολλές φορές τα training data περιέχουν πολλές λέξεις οι οποίες εμφανίζονται πολύ λίγες φορές, παρά το μεγάλο μέγεθος των training data. Το πιο πιθανό, αυτές οι λέξεις να αποτελούν ορθογραφικά λάθη ή να είναι λέξεις που δεν χρησιμοποιούνται πολύ. Οι λέξεις αυτές, αποτελούν σπατάλη χώρου και επίσης, η χρήση τους στα test data μπορεί να οδηγήσει σε λάθος συμπεράσματα από το ταξινομητή, αφού οι συγκεκριμένες λέξεις λόγω των λίγων εμφανίσεων τους στα training data παίρνουν πολύ μεγάλο βάρος, αν χρησιμοποιείται η τεχνική tf-idf. Η τεχνική αυτή έχει ως σκοπό τον εντοπισμό και τη διαγραφή των λέξεων αυτών με τον έλεγχο του αριθμού των εμφανίσεων κάθε λέξης στα έγγραφα της εισόδου.

Η συγκεκριμένη τεχνική υποστηρίζεται από το Mahout και εφαρμόζεται κατά τη διάρκεια μετατροπής των εγγράφων σε διανύσματα. Το Mahout δίνει τη δυνατότητα εισαγωγής οποιουδήποτε αριθμού και διαγράφει τις λέξεις με λιγότερες εμφανίσεις από το δοσμένο αριθμό σε όλα τα έγγραφα της εισόδου.

Η τεχνική αυτή εφαρμόζεται στα training data. Εφαρμογή της στα test data δεν έχει νόημα. Επίσης, η τεχνική αυτή έχει νόημα να εφαρμόζεται μόνο όταν χρησιμοποιείται η τεχνική tf-idf, αφού μόνο τότε οι συγκεκριμένες λέξεις παίρνουν μεγάλο βάρος. Όταν χρησιμοποιείται η τεχνική tf, τότε οι συγκεκριμένες λέξεις παίρνουν αμελητέο βάρος και η διαγραφή τους δε θα επιφέρει σχεδόν καμιά διαφορά στη βελτίωση της ακρίβειας. Φυσικά το γεγονός αυτό, θα έχει ως αποτέλεσμα τη μείωση του μεγέθους του dataset.

4.2.9 Ελάχιστες εμφανίσεις σε διαφορετικά έγγραφα

Έχει ακριβώς την ίδια χρησιμότητα με την πιο πάνω τεχνική, δηλαδή να διαγράφει σπάνιες λέξεις. Η διαφορά αυτής της τεχνικής όμως είναι ότι διαγράφει μια λέξη όταν αυτή δεν εμφανίζεται σ' ένα συγκεκριμένο αριθμό διαφορετικών εγγράφων. Στην προηγούμενη τεχνική μια λέξη μπορεί να έμενε στα έγγραφα ακόμη και αν εμφανιζόταν ικανοποιητικές φορές σε ένα και μόνο έγγραφο. Κανονικά όμως θα έπρεπε να διαγραφεί, μιας και όντως είναι σπάνια για να εμφανίζεται σε ένα και μόνο έγγραφο, ανεξάρτητα από τη συχνότητα εμφάνισής της σ' αυτό. Αν δε διαγραφεί, εκτός του ότι σπαταλά χώρο, θα πάρει και πολύ μεγάλο βάρος, στην περίπτωση χρήσης της τεχνικής tf-idf, γεγονός που ενδεχομένως να οδηγήσει σε λανθασμένες ταξινομήσεις. Η τεχνική όμως που παρουσιάζεται εδώ εγγυάται πως πραγματικά σπάνιες λέξεις, δηλαδή που εμφανίζονται σε λίγα έγγραφα, διαγράφονται.

Η συγκεκριμένη τεχνική υποστηρίζεται απ' το Mahout και εφαρμόζεται κατά τη διάρκεια μετατροπής των εγγράφων σε διανύσματα. Το Mahout δίνει τη δυνατότητα εισαγωγής οποιουδήποτε αριθμού και διαγράφει τις λέξεις που εμφανίζονται σε λιγότερα από τόσα διαφορετικά έγγραφα της εισόδου.

Η τεχνική αυτή εφαρμόζεται στα training data. Εφαρμογή της στα test data δεν έχει νόημα. Επίσης, όπως και πριν, η τεχνική αυτή έχει νόημα να εφαρμόζεται μόνο όταν χρησιμοποιείται η τεχνική tf-idf, μιας και μόνο τότε οι συγκεκριμένες λέξεις παίρνουν μεγάλο βάρος. Όταν χρησιμοποιείται η τεχνική tf, τότε οι λέξεις αυτές παίρνουν αμελητέο βάρος και η διαγραφή τους δε θα επιφέρει σχεδόν καμιά διαφορά στη βελτίωση της ακρίβειας. Φυσικά, το γεγονός αυτό θα έχει ως αποτέλεσμα, τη μείωση του μεγέθους του dataset.

4.2.10 Μέγιστες εμφανίσεις σε διαφορετικά έγγραφα

Αυτή η τεχνική χρησιμοποιείται για την αφαίρεση των πολύ χρησιμοποιούμενων λέξεων (stopwords) που υπάρχουν στα έγγραφα. Αυτές οι λέξεις, όπως το «and», «or» και «of» τείνουν να υπάρχουν σ' όλα τα έγγραφα. Η τεχνική αυτή διαγράφει τις λέξεις που υπάρχουν στο μεγαλύτερο ποσοστό των εγγράφων της εισόδου και έτσι καταφέρνει να μας απαλλάξει απ' τα περισσότερα stopwords.

Η συγκεκριμένη τεχνική υποστηρίζεται από το Mahout και εφαρμόζεται κατά τη διάρκεια μετατροπής των εγγράφων σε διανύσματα. Το Mahout δίνει τη δυνατότητα εισαγωγής του ποσοστού των εγγράφων και αν μια λέξη υπάρχει σε περισσότερα έγγραφα απ' αυτό το ποσοστό, τότε διαγράφεται απ' τα έγγραφα της εισόδου. Για παράδειγμα, δίνουμε ως είσοδο 100 έγγραφα και δίνουμε ως ποσοστό το 90%. Αν μια λέξη υπάρχει σε 91 διαφορετικά έγγραφα από τα 100 έγγραφα της εισόδου, τότε διαγράφεται απ' όλα τα έγγραφα της εισόδου.

Η τεχνική αυτή εφαρμόζεται στα training data. Εφαρμογή της στα test data δεν έχει νόημα.

4.2.11 Information Gain, Chi-Squared και Bi-Normal Separation

Οι τρεις αυτές τεχνικές έχουν τον ίδιο σκοπό, να επιλέξουν δηλαδή τις πιο σημαντικές λέξεις των εγγράφων που τους δίνονται ως είσοδος και να διαγράψουν τις υπόλοιπες. Η διαφορά τους έγκειται στον τρόπο υπολογισμού των πιο σημαντικών λέξεων των εγγράφων [21] [24]. Αυτές οι μέθοδοι διαγράφουν όλες τις λέξεις που δεν είναι σημαντικές, δηλαδή τις λέξεις που δεν προσφέρουν μεγάλη πληροφορία για την κατηγορία στην οποία ανήκουν και αφήνουν τις πιο σημαντικές λέξεις, δηλαδή αυτές που προσφέρουν τη μεγαλύτερη πληροφορία για την κατηγορία στην οποία ανήκουν. Ως αποτέλεσμα της εφαρμογής των συγκεκριμένων τεχνικών είναι η μείωση του μεγέθους του set που τους δίνεται ως είσοδος και η βελτίωση της ακρίβειας. Χρησιμοποιούνται κατά κόρο όταν χρειάζεται να γίνει μείωση του μεγέθους του training set. Ένα μειονέκτημά τους όμως είναι ότι δε δίνουν καλύτερη ακρίβεια όταν χρησιμοποιηθούν σε συνδυασμό μ' άλλες τεχνικές.

Για τον υπολογισμό των πιο αντιπροσωπευτικών λέξεων χρησιμοποιούν τους πιο κάτω τύπους:

$$Information\ Gain = -\left(\frac{gPos}{n}\right) * log_2 \frac{gPos}{n} - \left(\frac{gNeg}{n}\right) * log_2 \frac{gNeg}{n} +$$

$$\left(\frac{tp}{n}\right) * log_2 \frac{tp}{fPos} + \left(\frac{fp}{n}\right) * log_2 \frac{fp}{fPos} + \left(\frac{fn}{n}\right) * log_2 \frac{fn}{fNeg} + \left(\frac{tn}{n}\right) * log_2 \frac{tn}{fNeg} \quad (20)$$

$$Chi - Squared = \frac{n * (tp * tn - fn * fp)^2}{(tp + fp) * (tp + fn) * (fn + tn) * (fp + tn)} \quad (21)$$

$$Bi - Normal\ Separation = \left| F^{-1}\left(\frac{tp}{gPos}\right) - F^{-1}\left(\frac{fp}{fNeg}\right) \right| \quad (22)$$

Όπου στους πιο πάνω τύπους:

tp = true positives = correctly identified = αριθμός εγγράφων που περιέχουν τη λέξη t και ανήκουν στην κλάση c

fp = false positives = incorrectly identified = αριθμός εγγράφων που περιέχουν τη λέξη t αλλά δεν ανήκουν στη κλάση c

tn = true negatives = correctly rejected = αριθμός εγγράφων που δεν περιέχουν τη λέξη t και δεν ανήκουν στη κλάση c

fn = false negatives = incorrectly rejected = αριθμός εγγράφων που δεν περιέχουν τη λέξη t και ανήκουν στη κλάση c

gPos = tp+ fn

gNeg = fp + tn

fPos = tp + fp

fNeg = fn + tn

n = tp + fn + fp + tn

και F-1 είναι η αντίστροφη της Normal CDF

Οι πιο πάνω τεχνικές εφαρμόζονται σε κάθε διαφορετική λέξη που υπάρχει στα έγγραφα της εισόδου, δηλαδή εφαρμόζονται σ' όλο το λεξιλόγιο. Οι λέξεις που είναι σημαντικές παίρνουν μεγάλες τιμές, ενώ οι λέξεις που δεν είναι σημαντικές παίρνουν μικρές τιμές. Για παράδειγμα, θεωρούμε πως θέλουμε να ταξινομήσουμε έγγραφα σε δύο κατηγορίες, τις κατηγορίες «Θετικό» και «Αρνητικό». Για την εκπαίδευση του ταξινομητή χρησιμοποιούνται τα πιο κάτω έγγραφα:

Πίνακας 6: ***Training set παραδείγματος χρήσης τεχνικής Chi-Squared***

Έγγραφο	**Δοσμένη κατηγορία**
λέξη1 λέξη2	Θετικό
λέξη1 λέξη1 λέξη2	Θετικό
λέξη1	Θετικό
λέξη2 λέξη3	Αρνητικό
λέξη2	Αρνητικό
λέξη4 λέξη4	Αρνητικό

Επιλέγουμε να χρησιμοποιήσουμε την τεχνική Chi-Squared για την επιλογή των πιο αντιπροσωπευτικών λέξεων. Εκτελώντας την τεχνική αυτή στα εκπαιδευτικά έγγραφα που παρουσιάζονται στον πίνακα 6, πήραμε τα πιο κάτω αποτελέσματα:

(λέξη1, 6.0), (λέξη2, 0.0), (λέξη3, 1.2), (λέξη4, 1.2)

Στα πιο πάνω αποτελέσματα βλέπουμε πως η «λέξη1» είναι η πιο σημαντική. Αυτό φαίνεται πολύ εύκολα από τα έγγραφα στον πίνακα 6, αφού η «λέξη1» υπάρχει σε όλα τα έγγραφα της κατηγορίας «Θετικό», αλλά δεν υπάρχει σε κανένα έγγραφο της κατηγορίας «Αρνητικό». Στη συνέχεια, ακολουθεί η «λέξη3» και η «λέξη4». Όπως παρατηρούμε, αυτές βρίσκονται σ' ένα έγγραφο η κάθε μια, στην «Αρνητική» κατηγορία, αλλά δεν υπάρχουν καθόλου σε έγγραφα στη «Θετική» κατηγορία. Παρατηρούμε επίσης, πως παρόλο που η «λέξη4» υπάρχει 2 φορές παίρνει την ίδια τιμή με τη «λέξη3» που υπάρχει 1 φορά. Αυτό οφείλεται στο γεγονός, ότι οι τεχνικές αυτές ελέγχουν σε πόσα διαφορετικά έγγραφα υπάρχουν οι λέξεις και όχι πόσες είναι οι διαφορετικές τους εμφανίσεις. Τέλος, ακολουθεί η «λέξη2». Αν και υπάρχει σε δύο διαφορετικά έγγραφα της κατηγορίας «Θετικό» και δύο διαφορετικά έγγραφα της κατηγορίας «Αρνητικό» παίρνει την τιμή 0. Αυτό οφείλεται στο γεγονός, ότι η εν λόγω λέξη δεν προσφέρει καμιά πληροφορία, αφού βρίσκεται σε ίδιο ποσοστό εγγράφων και στις δύο κατηγορίες και συνεπώς δε θα βοηθούσε στην ταξινόμηση νέων εγγράφων.

Ουσιαστικά, αυτό που κάνουν οι συγκεκριμένες τεχνικές είναι να επιλέγουν ως σημαντικές τις λέξεις που βρίσκονται σε μεγαλύτερο ποσοστό εγγράφων στη μια κατηγορία σε σχέση με την άλλη. Λέξεις που βρίσκονται σε παρόμοια ποσοστά στις δύο κατηγορίες παίρνουν χαμηλή τιμή και συνεπώς δεν επιλέγονται. Όπως γίνεται φανερό, αυτές οι τεχνικές μπορούν να διαγράψουν τόσο τα stopwords που υπάρχουν σε μεγάλα ποσοστά σε κάθε κατηγορία, όσο και τις σπάνιες λέξεις που βρίσκονται μόνο σε ελάχιστα ποσοστά.

Ως είσοδο, οι συγκεκριμένες τεχνικές παίρνουν τον αριθμό των πιο σημαντικών λέξεων που θέλουμε να κρατήσουν στα έγγραφα της εισόδου. Δεν υπάρχει κάποιος τρόπος για την επιλογή του αριθμού αυτού. Για την εύρεση του αριθμού που δίνει την καλύτερη ακρίβεια, πρέπει να γίνουν αρκετές μετρήσεις.

Δυστυχώς αυτές οι τεχνικές δεν προσφέρονται από το Mahout. Ωστόσο, η υλοποίησή τους δεν είναι πολύ δύσκολη. Οι συγκεκριμένες τεχνικές εφαρμόζονται πριν τα έγγραφα δοθούν στο Mahout για μετατροπή σε διανύσματα. Εφαρμόζονται μόνο στα training data. Η εφαρμογή των πιο πάνω τεχνικών δεν πρέπει να γίνεται στα test data, μιας και δεν μπορούν οι τεχνικές αυτές να εφαρμοστούν σε πραγματικά άγνωστα δεδομένα. Αν τις εφαρμόσουμε στα test data, θα οδηγηθούμε σε αποτελέσματα που δε θα είναι αντιπροσωπευτικά για την εφαρμογή του ταξινομητή σε πραγματικά άγνωστα δεδομένα.

5

Εφαρμογή

Στην ενότητα αυτή, αρχικά θα παρουσιαστεί η εφαρμογή εξαγωγής συναισθήματος που μελετάται σ' αυτή την εργασία και στη συνέχεια, θα γίνει εφαρμογή του Naive Bayes και των διαφορετικών μεθόδων βελτίωσης της ακρίβειας που παρουσιάστηκαν στην προηγούμενη ενότητα. Για τις δοκιμές, θα γίνει χρήση ενός μεσαίου μεγέθους dataset, έτσι ώστε να είναι εφικτός ο πειραματισμός με τις μεθόδους αυτές. Επίσης, ο αλγόριθμος σ' αυτήν την περίπτωση θα εκτελεστεί σειριακά (τοπικά). Τα συμπεράσματα που θα εξαχθούν θα εφαρμοστούν σε αρκετά μεγάλα dataset της συγκεκριμένης εφαρμογής, με τη χρήση πολλών κόμβων (πάνω από το Hadoop) στην επόμενη ενότητα.

5.1 Σκοπός της εφαρμογής

Σκοπός της εφαρμογής είναι η εξαγωγή του συναισθήματος που κρύβουν κριτικές χρηστών και η κατάταξή τους σε «Θετικές» και «Αρνητικές». Οι κριτικές αυτές μπορούν να αφορούν εστιατόρια, ξενοδοχεία, αξιοθέατα κλπ. Η συγκεκριμένη εφαρμογή μπορεί να φανεί πολύ χρήσιμη σε ξενοδόχους, εστιάτορες, επιχειρήσεις που προσφέρουν συμβουλές για ταξίδια και σε πολλούς άλλους. Κι αυτό γιατί μπορεί να αυτοματοποιήσει την ταξινόμηση των κριτικών διαφορετικών χρηστών σε «Θετικές» και «Αρνητικές» και με βάση τα αποτελέσματα αυτά, να εξάγουν τα συμπεράσματα που τους ενδιαφέρουν. Η ταξινόμηση τέτοιων κριτικών στο χέρι θα ήταν πρακτικά αδύνατη, μιας και θα απαιτούσε πάρα πολύ χρόνο και κόπο. Ωστόσο, για να είναι χρήσιμη η συγκεκριμένη εφαρμογή, θα πρέπει να δίνει ακριβή αποτελέσματα. Αν η ακρίβεια της εφαρμογής είναι σχετικά μικρή, τότε αυτό θα οδηγήσει σε λάθος ταξινομήσεις και σε λάθος συμπεράσματα. Λογικό είναι να υπάρχουν και λάθος ταξινομήσεις, αλλά απαιτείται να είναι όσον το δυνατό λιγότερες.

5.2 Δεδομένα της εφαρμογής

Τα δεδομένα που θα χρησιμοποιηθούν στη συγκεκριμένη εφαρμογή προέρχονται από το TripAdvisor [25]. Το TripAdvisor είναι μια ιστοσελίδα, η οποία δίνει τη δυνατότητα στους χρήστες της, να βλέπουν κριτικές χρηστών για ξενοδοχεία, εστιατόρια, αξιοθέατα κλπ. και με βάση αυτές να οργανώνουν τα ταξίδια τους. Τους δίνεται επίσης η δυνατότητα να

μοιράζονται τις δικές τους εμπειρίες με τους υπόλοιπους χρήστες, γράφοντας τις δικές τους κριτικές.

Η συγκεκριμένη ιστοσελίδα περιέχει πάρα πολλές κριτικές διαφορετικών χρηστών για πολλούς διαφορετικούς τόπους και αυτό την κάνει ιδανική για την εφαρμογή μας. Τα δεδομένα αυτής της ιστοσελίδας μπορούν να αποτελέσουν πολύ καλά εκπαιδευτικά έγγραφα για τον ταξινομητή. Λόγω της μεγάλης ποικιλίας διαφορετικών κριτικών που υπάρχουν, ο ταξινομητής θα είναι ικανός να κατατάξει ένα ευρύ φάσμα από κριτικές χρηστών.

Για να γίνουν καλύτερα κατανοητά τα δεδομένα που προσφέρονται απ' αυτήν την ιστοσελίδα, παρουσιάζονται πιο κάτω οι πληροφορίες που παρέχονται για ένα εστιατόριο.

Εικόνα 4: ***Πληροφορίες που υπάρχουν για ένα εστιατόριο***

Όπως φαίνεται από την εικόνα 4, παρέχονται πληροφορίες όπως το όνομα, η διεύθυνση, η ιστοσελίδα του εστιατορίου, το τηλέφωνο και το e-mail επικοινωνίας, η γενική βαθμολογία του εστιατορίου η οποία προέρχεται από τους χρήστες (με κλίμακα 1 έως 5, όπου 1 η χαμηλότερη και 5 η υψηλότερη βαθμολογία), κάποιες άλλες γενικές πληροφορίες οι οποίες για κάθε είδος τοποθεσίας είναι διαφορετικές όπως τιμές και το είδος του φαγητού που προσφέρει, καθώς επίσης, παρουσιάζονται και φωτογραφίες του ίδιου του εστιατορίου.

Επίσης για το εστιατόριο δίνονται και οι κριτικές των χρηστών.

Εικόνα 5: ***Πληροφορίες που υπάρχουν για μια κριτική***

Όπως φαίνεται από την εικόνα 5, η κριτική αποτελείται από ένα σχόλιο, τον τίτλο του σχολίου, τη βαθμολογία που βάζει ο εκάστοτε χρήστης (με κλίμακα 1 έως 5, όπου 1 η χαμηλότερη και 5 η υψηλότερη βαθμολογία), την ημερομηνία εισαγωγής της κριτικής, το όνομα, την καταγωγή και το συνολικό αριθμό των κριτικών του εν λόγω χρήστη. Παρόμοια δεδομένα δίνονται και για τα ξενοδοχεία, αξιοθέατα κλπ.

Μιας και σκοπός της εφαρμογής μας είναι η ταξινόμηση των κριτικών χρηστών σε «Θετικές» και «Αρνητικές», γίνεται λοιπόν φανερό πως τα training έγγραφα θα αποτελούνται σίγουρα από τις κριτικές των χρηστών. Το γεγονός ότι στα δεδομένα που έχουν συλλεχθεί, οι κριτικές των χρηστών περιέχουν πολλές πληροφορίες (σχόλιο, τίτλος σχολίου, όνομα χρήστη κλπ.) δημιουργείται εύλογα το ερώτημα για το ποιες είναι οι πιο κατάλληλες. Επίσης, στα εν λόγω δεδομένα υπάρχουν και πληροφορίες για τον τόπο όπου έγιναν οι κριτικές αυτές, γεγονός που δημιουργεί και το ερώτημα του κατά πόσο μπορούν να χρησιμοποιηθούν και οι συγκεκριμένες πληροφορίες στα training έγγραφα.

Επιπλέον, είναι γνωστό ότι τα training data πρέπει να αποτελούνται τόσο από τα έγγραφα, όσο και από τις κατηγορίες στις οποίες ανήκουν τα έγγραφα αυτά. Η αντιστοίχιση των εγγράφων με την κατηγορία στην οποία ανήκουν, πρέπει να γίνεται κανονικά χειροκίνητα, αλλά ευτυχώς οι κριτικές περιέχουν και τη βαθμολογία του χρήστη, η οποία μπορεί να χρησιμοποιηθεί για την αντιστοίχιση ενός εγγράφου (κριτικής και οτιδήποτε άλλο χρήσιμο) με την κατηγορία στην οποία ανήκει (Θετική ή Αρνητική). Ωστόσο, γνωρίζοντας την κλίμακα των βαθμολογιών, δημιουργείται ένα ακόμη ερώτημα το οποίο αφορά τις βαθμολογίες και ποιες από αυτές αντιστοιχούν σε ποια κατηγορία. Στα ερωτήματα αυτά, θα δοθεί απάντηση στην επόμενη υποενότητα, όπου θα γίνει η επιλογή κατάλληλου training set από τα διαθέσιμα δεδομένα.

5.3 Επιλογή κατάλληλου training set

Όπως αναφέρθηκε και σε προηγούμενη ενότητα, η επιλογή κατάλληλου training set είναι πολύ σημαντική και πρέπει να γίνεται με προσοχή. Σ' αυτή την υποενότητα εξετάζονται τα θέματα: ποιες βαθμολογίες αντιστοιχούν σε ποιες κατηγορίες, ποια δεδομένα είναι τα πιο κατάλληλα για τα training έγγραφα και ποια πρέπει να είναι η αναλογία των εκπαιδευτικών εγγράφων ανά κατηγορία.

5.3.1 Βαθμολογίες ανά κατηγορία

Ένα πρόβλημα που συναντάται είναι ποιες βαθμολογίες πρέπει να χρησιμοποιηθούν ανά κατηγορία, ώστε να καταταχτούν τα εκπαιδευτικά έγγραφα σε κατηγορίες.

Γνωρίζουμε πως οι βαθμολογίες των κριτικών ακολουθούν την κλίμακα 1 έως 5, με χαμηλότερη βαθμολογία το 1 και υψηλότερη το 5. Ένα πρόβλημα που υπάρχει είναι η υποκειμενικότητα όσον αφορά τον τρόπο βαθμολόγησης των χρηστών. Μ' άλλα λόγια, η ίδια βαθμολογία μπορεί να αντιστοιχεί σε κάτι θετικό για κάποιους χρήστες, ενώ για κάποιους άλλους σε κάτι αρνητικό. Αυτό που μπορούμε όμως να υποθέσουμε είναι πως όσο πιο ψηλή βαθμολογία έχει μια κριτική, τόσο πιο «Θετική» είναι, ενώ όσο πιο χαμηλή βαθμολογία έχει, τόσο πιο «Αρνητική» είναι. Συνεπώς, για να μπορέσουμε να κατανοήσουμε πως αντιστοιχεί η πλειοψηφία των χρηστών τις βαθμολογίες στις κατηγορίες πρέπει να κάνουμε αρκετές μετρήσεις. Είναι αδύνατο να μπορέσουμε να κατατάξουμε όλες τις κριτικές σε «Θετικές» και «Αρνητικές» σωστά, λόγω αυτού που αναφέρθηκε πιο πάνω, αλλά σκοπός μας είναι να κατατάξουμε την πλειοψηφία αυτών. Κανονικά όμως, δεν πρέπει να υπάρχουν λάθος ταξινομημένα training data, αφού αυτό θα οδηγήσει σε ανακριβή εκπαίδευση και

συνεπώς σε μειωμένη ακρίβεια σε πραγματική χρήση. Λόγω της ποσότητας των δεδομένων, η χειροκίνητη αντιστοίχιση δεν αποτελεί λύση και συνεπώς δεχόμαστε αυτή την απώλεια ακρίβειας.

Όπως αναφέρθηκε ήδη, η κλίμακα των βαθμολογιών κυμαίνεται από το 1 έως το 5. Θα μπορούσαμε λοιπόν να υποθέσουμε ότι οι βαθμολογίες 1 και 2 αντιστοιχούν σε «Αρνητικές» κριτικές, ενώ οι βαθμολογίες 4 και 5 σε «Θετικές». Η βαθμολογία 3, η οποία βρίσκεται στο μέσο είναι πολύ αμφιλεγόμενη και έτσι το πρώτο πράγμα που θα εξεταστεί είναι το κατά πόσο αυτή μοιάζει περισσότερο με τις βαθμολογίες 1 και 2 ή με τις βαθμολογίες 4 και 5.

Πίνακας 7: ***Βαθμολογία 3 ως Αρνητική και ως Θετική***

Training data για Αρνητική κατηγορία	**Training data για Θετική κατηγορία**	**Ακρίβεια ταξινόμησης**
1(70419), 2(70419)	3(46946), 4(46946), 5(46946)	82.9826%
1(46946), 2(46946), 3(46946)	4(70419), 5(70419)	84.0524%
1(70419), 2(70419)	4(70419), 5(70419)	86.6746%

Πίνακας 8: ***Σύγκριση βαθμολογίας 3 με τις υπόλοιπες βαθμολογίες***

Training data για 1η κατηγορία	**Training data για 2η κατηγορία**	**Ακρίβεια ταξινόμησης**
1(70419), 2(70419)	3(140838)	73.2963%
4(70419), 5(70419)	3(140838)	78.6034%

Ο πίνακας 7 και ο πίνακας 8 παρουσιάζουν ποιες βαθμολογίες χρησιμοποιήθηκαν ανά κατηγορία για την εκπαίδευση του ταξινομητή, καθώς και την ακρίβεια του ταξινομητή σε κάθε περίπτωση μετά από αξιολόγηση. Επίσης, παρουσιάζουν τον αριθμό των εγγράφων τα οποία χρησιμοποιήθηκαν από κάθε βαθμολογία. Παρατηρούμε πως χρησιμοποιούνται ίσα έγγραφα για την εκπαίδευση κάθε κατηγορίας, γεγονός το οποίο θα εξηγηθεί στην αντίστοιχη υποενότητα. Τα έγγραφα περιείχαν μόνο τα σχόλια της κάθε κριτικής. Για την αξιολόγηση χρησιμοποιήθηκε η τεχνική cross-validation με ποσοστό 80% του training set ως training data και 20% ως test data.

Με βάση τα αποτελέσματα, παρατηρούμε ότι η βαθμολογία 3 μοιάζει περισσότερο στις βαθμολογίες 1 και 2, παρά στις βαθμολογίες 4 και 5. Όπως φαίνεται από τον πίνακα 7, όταν η βαθμολογία 3 χρησιμοποιήθηκε ως «Αρνητική», η ακρίβεια ταξινόμησης είναι ψηλότερη από όταν αυτή χρησιμοποιήθηκε ως «Θετική». Παράλληλα, όπως φαίνεται από τον πίνακα 8, η ακρίβεια ταξινόμησης είναι μεγαλύτερη, όταν η βαθμολογία 3 συγκρίνεται με τις βαθμολογίες 4 και 5, από όταν συγκρίνεται με βαθμολογίες 1 και 2. Υψηλότερη ακρίβεια σημαίνει ότι τα έγγραφα των δύο κατηγοριών είναι διαφορετικά μεταξύ τους.

Σύμφωνα με τον πίνακα 8, η βαθμολογία 3 περιέχει σε μεγάλο βαθμό τόσο «Αρνητικές», όσο και «Θετικές» κριτικές. Το γεγονός αυτό αποδεικνύεται από την ακρίβεια του ταξινομητή, η οποία σε καμιά από τις δυο περιπτώσεις δεν είναι κοντά στο 50%. Σ' αντίθετη περίπτωση, αυτό θα υποδήλωνε πως οι κατηγορίες που συγκρίνονται μεταξύ τους είναι αδύνατο να ξεχωριστούν και ο ταξινομητής κατηγοριοποιεί στη τύχη.

Επιπλέον, είναι φανερό ότι στην περίπτωση που δε χρησιμοποιείται η βαθμολογία 3, η ακρίβεια είναι μεγαλύτερη (πίνακας 7). Για να μην οδηγηθούμε λοιπόν σε μια ανακριβή εκπαίδευση, αποφασίστηκε να μη δοθούν προς εκπαίδευση οι κριτικές της βαθμολογίας 3.

Ακολούθως, κρίθηκε σκόπιμο να συγκριθούν και οι υπόλοιπες βαθμολογίες, κάνοντας χρήση της τεχνικής cross-validation με αναλογία 80%-20% και εγγράφων, τα οποία να περιέχουν μόνο τα σχόλια της κάθε κριτικής.

Πίνακας 9: ***Σύγκριση των βαθμολογιών 1, 2, 4 και 5***

Training data για 1η κατηγορία	**Training data για 2η κατηγορία**	**Ακρίβεια ταξινόμησης**
2(70419)	4(70419)	84.6265%
1(70419)	5(70419)	91.2379%
1(70419)	4(70419)	89.6535%
2(70419)	5(70419)	89.4903%
1(70419)	2(70419)	66.0005%
4(70419)	5(70419)	67.6015%

Σύμφωνα με τα αποτελέσματα που παρουσιάζονται στον πίνακα 9, παρατηρούμε ότι οι καλύτερα διαχωρίσιμες βαθμολογίες είναι οι βαθμολογίες 1 και 5. Οι βαθμολογίες 2 και 4, αν και είναι κατά ένα πολύ μεγάλο ποσοστό διαχωρίσιμες, σίγουρα περιέχουν έγγραφα τα οποία θα συγχύσουν τον ταξινομητή και θα τον οδηγήσουν σε λάθη. Επίσης, παρατηρούμε ότι οι βαθμολογίες 1 και 2 είναι κατά ένα μικρό βαθμό διαχωρίσιμες, μιας και η ακρίβεια είναι μεγαλύτερη από το 50%. Το ίδιο ισχύει και για τις βαθμολογίες 4 και 5.

Κανονικά για την εκπαίδευση του ταξινομητή θα πρέπει να χρησιμοποιηθούν μόνο οι βαθμολογίες 1 και 5, μιας και είναι οι πιο διαχωρίσιμες μεταξύ τους και συνεπώς θα αποτελούσαν τις πιο αντιπροσωπευτικές βαθμολογίες για τις κατηγορίες «Θετικό» και «Αρνητικό». Αυτό φυσικά ήταν αναμενόμενο, αφού η βαθμολογία 1 αποτελεί τη χαμηλότερη βαθμολογία που μπορεί να δώσει ένας χρήστης και η βαθμολογία 5 την υψηλότερη. Έτσι, όσο διαφορετικά και να αντιλαμβάνεται ο κάθε χρήστης την κάθε βαθμολογία, οι βαθμολογίες 1 και 5 αποτελούν για τον κάθε χρήστη το ίδιο πράγμα. Δυστυχώς όμως, αν χρησιμοποιηθούν μόνο οι βαθμολογίες 1 και 5, αυτό θα έχει ως αποτέλεσμα τη χρήση μόνο ενός μικρού ποσοστού του dataset, γεγονός το οποίο θα περιορίσει αρκετά τις συγκρίσεις μεταξύ διαφορετικών μεγεθών dataset στη συνέχεια. Έτσι λοιπόν κρίθηκε αναγκαία η χρήση των βαθμολογιών 1-2 ως «Αρνητικές» και 4-5 ως «Θετικές». Η αντιστοίχιση αυτή δεν περιορίζει το μέγεθος του dataset, καθώς επίσης έχει και ένα αρκετά καλό διαχωρισμό, παρόλο που μπορεί να περιέχει κάποια έγγραφα τα οποία θα είναι παραπλανητικά για τον ταξινομητή.

5.3.2 Καταλληλότερα δεδομένα

Έχοντας επιλέξει ποιες βαθμολογίες αντιστοιχούν σε ποιες κατηγορίες, θα πρέπει να γίνει και η επιλογή των καταλληλότερων δεδομένων για τα εκπαιδευτικά έγγραφα. Η

χρησιμοποίηση όλων των δεδομένων που συλλέχτηκαν στα εκπαιδευτικά έγγραφα δεν αποτελεί λύση, μιας και αυτό θα οδηγήσει σε σπατάλη χώρου, χρόνου και ακρίβειας.

Πέραν των σχολίων των χρηστών που περιέχονται στις κριτικές, μπορούν και άλλες πληροφορίες να φανούν χρήσιμες στο διαχωρισμό εγγράφων ανάμεσα στις δύο κατηγορίες. Έτσι μελετώντας τα δεδομένα που έχουμε, τόσο δηλαδή τις πληροφορίες που περιέχονται στις κριτικές των χρηστών, όσο και τις πληροφορίες που περιέχονται για τους τόπους που αφορούν οι κριτικές, καταλήξαμε στα πιο κάτω συμπεράσματα:

- Όνομα τόπου όπου έγινε η κριτική: Αν τοποθετούσαμε τη συγκεκριμένη πληροφορία στα training έγγραφα, τότε το όνομα της τοποθεσίας θα συνδεόταν μ' ένα αριθμό από Θετικά και Αρνητικά έγγραφα. Ουσιαστικά, το όνομα της τοποθεσίας θα συνδεόταν με το μέσο όρο των βαθμολογιών όλων των χρηστών. Αυτό θα έχει ως αποτέλεσμα, αν εμφανιζόταν το όνομα της συγκεκριμένης τοποθεσίας σ' ένα νέο έγγραφο, αυτή η λέξη να έτεινε να ταξινομήσει το νέο έγγραφο ως προς τη μέση τιμή όλων των βαθμολογιών των χρηστών. Αυτή η ιδέα βασίζεται στο γεγονός πως αν μια τοποθεσία έχει βαθμολογηθεί από πολλούς χρήστες ως «Θετική» για παράδειγμα, τότε είναι πιο πιθανό να εμφανιστεί μια νέα θετική κριτική αντί για αρνητική. Εκ πρώτης όψεως, η συγκεκριμένη πληροφορία θα μπορούσε να βοηθήσει στην ταξινόμηση. Ωστόσο, αν η συγκεκριμένη τοποθεσία υποστεί αλλαγές π.χ. γίνει κάποια ανακαίνιση, αυτό θα είχε ως αποτέλεσμα η συγκεκριμένη λέξη να οδηγούσε σε λάθος αποτελέσματα, αφού θα έπαιρνε μια τιμή η οποία πιθανό να μην ισχύει μετά την ανακαίνιση. Για παράδειγμα, έχουμε ένα εστιατόριο του οποίου η γενική του εικόνα είναι αρνητική. Προσθέτοντας την ονομασία του στα έγγραφα, η ονομασία του θα έτεινε να ταξινομήσει ένα νέο έγγραφο που περιέχει το όνομα του εστιατορίου αυτού ως προς την «Αρνητική» κατηγορία. Αν όμως το εστιατόριο βλέποντας την αρνητική του εικόνα προσλάμβανε νέους μάγειρες, η εικόνα του μπορεί να γινόταν «Θετική». Λόγω του ότι όμως το παραγμένο μοντέλο του ταξινομητή δεν ενημερώνεται συχνά με νέα έγγραφα, θα έτεινε να ταξινομήσει και πάλι ένα νέο έγγραφο που περιέχει την ονομασία της τοποθεσίας ως «Αρνητικό», ενώ τη στιγμή αυτή αφού η εικόνα του είναι «Θετική» είναι πιο πιθανό να εμφανιστεί μια θετική, αντί για αρνητική κριτική. Για το λόγο αυτό η συγκεκριμένη πληροφορία δεν τοποθετήθηκε στα training data.
- Τηλέφωνο επικοινωνίας, ιστοσελίδα του εστιατορίου και email επικοινωνίας: Ισχύει ακριβώς το ίδιο με το όνομα της τοποθεσίας, έτσι οι συγκεκριμένες πληροφορίες δεν τοποθετήθηκαν στα training data.
- Διεύθυνση τοποθεσίας: Ισχύει και πάλι το ίδιο με πιο πάνω, έτσι δεν τοποθετείται στα training data. Επίσης αυτό, θα μπορούσε να οδηγήσει και σε εξαρτήσεις μεταξύ εγγράφων που περιέχουν κριτικές για τόπους, όπου έγιναν σε παρόμοιες τοποθεσίες, λόγω του ότι ο Naive Bayes θεωρεί πως η κάθε λέξη συνεισφέρει ανεξάρτητα στο να καταταχθεί ένα νέο έγγραφο σε μια κατηγορία. Για παράδειγμα, μπορεί δυο εστιατόρια να μην έχουν ακριβώς τις ίδιες διευθύνσεις, αλλά μπορεί να βρίσκονται στην ίδια χώρα. Η χώρα στη διεύθυνση τους θα είναι η ίδια και θα προκαλεί εξαρτήσεις μεταξύ τους. Έτσι ένα νέο έγγραφο το οποίο περιέχει τη διεύθυνση ενός νέου εστιατόριου που βρίσκεται και αυτό στην ίδια χώρα, θα είχε ως αποτέλεσμα η κατάταξη του σε

κατηγορία να επηρεαζόταν από τη χώρα όλων των εστιατορίων που βρίσκονται στην ίδια χώρα.

- Τίτλοι σχολίων: Οι τίτλοι των σχολίων που έδωσε ο χρήστης είναι εξίσου καλοί για τα training data με τα σχόλια. Αυτό οφείλεται στο γεγονός ότι οι χρήστες τείνουν να γράφουν το συναίσθημα τους στο τίτλο με λίγα λόγια. Για παράδειγμα μερικοί τίτλοι είναι οι: «Absolutely amazing», «Outstanding in Every way» και «disappointment». Όπως γίνεται φανερό, όλοι οι τίτλοι περιέχουν λέξεις οι οποίες βοηθάνε στην κατάταξη ενός εγγράφου στη σωστή κατηγορία.
- Όνομα χρήστη: Δεν προσφέρει καμιά πληροφορία για τις κατηγορίες.
- Καταγωγή χρήση: Δεν προσφέρει καμιά πληροφορία για τις κατηγορίες.
- Ημερομηνία εισαγωγής της κριτικής: Δεν προσφέρει καμιά πληροφορία για τις κατηγορίες.
- Αριθμός κριτικών χρήστη: Δεν προσφέρει καμιά πληροφορία για τις κατηγορίες.

Συνεπώς, με βάση τα πιο πάνω συμπεράσματα, το μόνο που θα μπορούσε να φανεί χρήσιμο, πέραν από τα σχόλια των χρηστών, είναι οι τίτλοι των σχολίων που έδωσαν οι χρήστες. Για να μπορέσει όμως να επιβεβαιωθεί πως τα συγκεκριμένα δεδομένα είναι χρήσιμα, πραγματοποιήθηκαν μετρήσεις.

Πίνακας 10: ***Χρήση τίτλων σχολίων στα training data***

Training και test data για κατηγορίες	**Ακρίβεια ταξινόμησης**
Μόνο σχόλια	86.6746%
Τίτλοι και σχόλια	90.1537%

Για την αξιολόγηση έγινε εκ των προτέρων μοίρασμα του dataset σε training και test set. Το μοίρασμα έγινε 80% training set και 20% test set. Δεν έγινε χρήση της τεχνικής cross-validation, αφού σκοπός είναι η πραγματοποίηση συγκρίσεων μεταξύ των αξιολογήσεων. Για το λόγο αυτό, όλες οι αξιολογήσεις πρέπει να γίνονται με το ίδιο training και test set, με πιθανές τροποποιήσεις, ανάλογα με την περίπτωση. Για παράδειγμα, στις δύο περιπτώσεις που παρουσιάζονται στον πίνακα 10, τα σχόλια στα training και test sets είναι τα ίδια, μόνο που στη δεύτερη περίπτωση προσθέτονται και οι τίτλοι των σχολίων μαζί με τα αντίστοιχα σχόλια. Επίσης, το dataset (πριν το μοίρασμα) αποτελείται από 281676 έγγραφα, 70419 ανά βαθμολογία και συνεπώς 140838 ανά κατηγορία, μιας και κάθε κατηγορία αποτελείται από δύο βαθμολογίες (1 και 2 θεωρούνται ως «Αρνητική» και 4 και 5 ως «Θετική»). Το μοίρασμα σε training και test set έγινε προσεχτικά, έτσι ώστε η κάθε βαθμολογία και συνεπώς και η κάθε κατηγορία να αποτελείται από τα ίσα έγγραφα.

Σύμφωνα με τα αποτελέσματα που παρουσιάζονται στον πίνακα 10, παρατηρούμε πως προσθέτοντας τους τίτλους των σχολίων αυξάνεται αρκετά η ακρίβεια του ταξινομητή. Κρίναμε σκόπιμο όμως, να εξετάσουμε όλους τους πιθανούς συνδυασμούς των τίτλων και των σχολίων στα training και test data, με στόχο την επίτευξη της καλύτερης δυνατής ακρίβειας.

Πίνακας 11: ***Δοκιμή διαφορετικών συνδυασμών τίτλων και σχολίων στα training και test data***

Training data για κατηγορίες	Test data για κατηγορίες	Ακρίβεια ταξινόμησης
Τίτλοι και σχόλια	Τίτλοι και σχόλια	90.1537%
Τίτλοι και σχόλια	Μόνο Τίτλοι	88.6272%
Τίτλοι και σχόλια	Μόνο Σχόλια	86.6125%
Μόνο Τίτλοι	Τίτλοι και σχόλια	83.2913%
Μόνο Τίτλοι	Μόνο Τίτλοι	88.7088%
Μόνο Τίτλοι	Μόνο Σχόλια	77.9981%
Μόνο Σχόλια	Τίτλοι και σχόλια	89.9904%
Μόνο Σχόλια	Μόνο Τίτλοι	87.6757%
Μόνο Σχόλια	Μόνο Σχόλια	86.6746%

Σύμφωνα με τα αποτελέσματα που παρουσιάζονται στον πίνακα 11, παρατηρούμε πως ο συνδυασμός τίτλοι και σχόλια τόσο στα training, όσο και στα test data δίνει όντως τα καλύτερα αποτελέσματα. Επίσης, παρατηρούμε πως όταν χρησιμοποιούνται μόνο οι τίτλοι των σχολίων και στα training και στα test data, η ακρίβεια ταξινόμησης είναι καλύτερη από όταν χρησιμοποιούνται μόνο σχόλια και στα training και στα test data. Επιπλέον, οι τίτλοι μπορούν να ταξινομήσουν τα σχόλια πολύ καλά και αντίστροφα. Αυτό φαίνεται από το συνδυασμό τίτλοι στα training data και σχόλια στα test data και αντίστροφα.

Συμπερασματικά, η πολύ καλή διαχωριστικότητα μεταξύ των τίτλων και η στενή σχέση μεταξύ των τίτλων και των σχολίων συνέτεινε στην επίτευξη της καλύτερης δυνατής ακρίβειας από το συνδυασμό τίτλοι και σχόλια τόσο στα training, όσο και στα test data.

5.3.3 Αναλογία εγγράφων ανά κατηγορία και βαθμολογία

Όπως παρατηρήθηκε, σ' όλες τις προηγούμενες μετρήσεις που έγιναν, ο αριθμός των εγγράφων που χρησιμοποιήθηκαν ανά κατηγορία είναι ο ίδιος. Ένα καινούργιο έγγραφο είναι εξίσου πιθανό να είναι «Θετικό» ή «Αρνητικό» και συνεπώς πρέπει να χρησιμοποιηθεί ίσος αριθμός εγγράφων ανά κατηγορία, έτσι ώστε η προγενέστερη πιθανότητα στην εξίσωση του Naive Bayes να είναι ίση με ½ και για τις δύο κατηγορίες. Χρησιμοποίηση διαφορετικής αναλογίας εγγράφων θα οδηγούσε σε άνισες πιθανότητες και συνεπώς η μια κατηγορία θα υπερτερούσε της άλλης, γεγονός που δεν είναι επιθυμητό.

Επίσης, έγινε χρήση ίσων εγγράφων ανά βαθμολογία. Σε αντίθετη περίπτωση, δηλαδή η χρήση περισσότερων εγγράφων από τις βαθμολογίες 1 και 5, θα είχε ως αποτέλεσμα τη βελτίωση της ακρίβειας, ενώ η χρήση περισσότερων εγγράφων από τις βαθμολογίες 2 και 4, θα είχε ως αποτέλεσμα τη μείωση της ακρίβειας. Φυσικά, η βελτίωση της ακρίβειας από τη χρήση περισσότερων εγγράφων από τις βαθμολογίες 1 και 5, δεν είναι επιθυμητή λόγω μεγέθους του dataset, γεγονός που αναλύθηκε σε προηγούμενο στάδιο. Από την άλλη όμως, δε θέλουμε να χάσουμε την ακρίβεια που προσφέρουν οι βαθμολογίες 1 και 5, και έτσι δε γίνεται χρήση περισσότερων εγγράφων από τις βαθμολογίες 2 και 4.

5.4 Εφαρμογή τεχνικών βελτίωσης της ακρίβειας

Έχοντας επιλέξει κατάλληλο training set, τώρα μπορούν να χρησιμοποιηθούν οι διαφορετικές τεχνικές βελτίωσης της ακρίβειας που παρουσιάστηκαν, με σκοπό την επίτευξη όσον το δυνατό καλύτερης ακρίβειας και την κατανόηση της συμπεριφοράς των τεχνικών αυτών, στην εφαρμογή που αναλύθηκε πιο πάνω. Θα χρησιμοποιηθούν όλες οι τεχνικές που παρουσιάστηκαν, μια προς μια, και θα συνοδεύονται από αξιολόγηση και συμπεράσματα.

Όπως αναφέρθηκε και στην προηγούμενη υποενότητα, για την εκπαίδευση και την αξιολόγηση του ταξινομητή χρησιμοποιούνται συνεχώς τα ίδια training και test set. Απλά κάθε φορά υπόκεινται στην επεξεργασία που απαιτεί η προς αξιολόγηση τεχνική. Ο σκοπός που γίνεται αυτό είναι για να μπορούν να γίνουν συγκρίσεις μεταξύ των αποτελεσμάτων.

5.4.1 Εφαρμογή τεχνικής 1 (Μετατροπή γραμμάτων σε πεζά και αφαίρεση συμβόλων)

Η πρώτη τεχνική που θα εφαρμοστεί είναι η μετατροπή των γραμμάτων σε πεζά και η αφαίρεση των συμβόλων απ' αυτά. Η τεχνική αυτή θα εφαρμοστεί τόσο στα training, όσο και στα test data.

Για την αξιολόγηση των συγκεκριμένων τεχνικών πραγματοποιήθηκαν τέσσερις διαφορετικές μετρήσεις. Η πρώτη μέτρηση έγινε με τη χρήση του WhitespaceAnalyzer αναλυτή που προσφέρει το Mahout. Ο συγκεκριμένος αναλυτής κειμένου δε μετατρέπει τα γράμματα σε πεζά, ούτε αφαιρεί τα σύμβολα που υπάρχουν στα έγγραφα. Οι υπόλοιπες τρεις μετρήσεις έγιναν με τη χρήση δικού μας αναλυτή κειμένου. Στη δεύτερη μέτρηση μετατράπηκαν όλα τα γράμματα σε πεζά, στην τρίτη αφαιρέθηκαν μόνο τα σύμβολα και στην τέταρτη διαγράφηκαν όλα τα σύμβολα και έγιναν και όλα τα γράμματα πεζά.

Πίνακας 12: ***Εφαρμογή τεχνικής 1 (μετατροπή γραμμάτων σε πεζά και αφαίρεση συμβόλων)***

Αλλαγές στα έγγραφα	Ακρίβεια ταξινομητή
Καμιά αλλαγή	90.1537%
Μετατροπή γραμμάτων σε πεζά	90.9543%
Διαγραφή συμβόλων	90.7395%
Μετατροπή γραμμάτων σε πεζά και διαγραφή συμβόλων (1)	91.6288%

Σύμφωνα με τα αποτελέσματα που παρουσιάζονται στον πίνακα 12, παρατηρούμε πως η μετατροπή των γραμμάτων σε πεζά και η διαγραφή των συμβόλων συμβάλουν στη βελτίωση της ακρίβειας του ταξινομητή.

Όπως προαναφέρθηκε, οι συγκεκριμένες τεχνικές θα πρέπει να χρησιμοποιηθούν ακόμη κι αν δε συνεισφέρουν στη βελτίωση της ακρίβειας, έτσι ώστε να έχουμε μια καλή εικόνα της ακρίβειας του ταξινομητή σε νέα έγγραφα.

Ο αριθμός που φαίνεται στον πίνακα 12 χρησιμοποιείται για να μπορούμε να αναφερόμαστε στο συγκεκριμένο dataset στη συνέχεια. Για παράδειγμα, αν πιο κάτω αναφερθεί πως θα

γίνει χρήση του dataset 1, σημαίνει πως τα δεδομένα που θα χρησιμοποιηθούν έχουν όλα τα γράμματα πεζά και δεν περιέχουν σύμβολα.

5.4.2 Εφαρμογή τεχνικής 2 (Αφαίρεση αριθμών)

Η τεχνική 2 αναφέρεται στην αφαίρεση των αριθμών τόσο στα training, όσο και στα test data. Για την αξιολόγηση της τεχνικής αυτής χρησιμοποιήθηκε το dataset 1, δηλαδή το dataset που πήραμε μετά την εφαρμογή της τεχνικής 1.

Πίνακας 13: ***Εφαρμογή τεχνικής 2 (αφαίρεση αριθμών)***

Αφαίρεση αριθμών	Ακρίβεια ταξινομητή
Όχι	91.6288%
Ναι (2)	91.7158%

Εφαρμόζοντας λοιπόν τη συγκεκριμένη τεχνική, παρατηρήσαμε ότι με την αφαίρεση των αριθμών που υπάρχουν, η ακρίβεια του ταξινομητή παρουσίασε μια μικρή βελτίωση (πίνακας 13).

5.4.3 Εφαρμογή τεχνικής 3 (Tf-idf)

Μέχρι στιγμής το βάρος μιας λέξης σ' ένα διάνυσμα ισούταν με τις εμφανίσεις της λέξης στο έγγραφο που χρησιμοποιήθηκε για την κατασκευή του διανύσματος (tf). Σ' αυτό το κομμάτι θα γίνει αξιολόγηση της τεχνικής tf-idf, δηλαδή το βάρος της κάθε λέξης αντί να υπολογίζεται με τη χρήση της τεχνικής tf, θα υπολογίζεται με τη χρήση της τεχνικής tf-idf.

Για την αξιολόγηση της τεχνικής αυτής χρησιμοποιήθηκε το dataset 2, δηλαδή το dataset που πήραμε μετά την εφαρμογή της τεχνικής 1 και 2 στο αρχικό dataset. Εφαρμόζοντας λοιπόν τη συγκεκριμένη τεχνική τόσο στα training όσο και στα test data, η ακρίβεια του ταξινομητή μειώθηκε (πίνακας 14).

Πίνακας 14: ***Εφαρμογή τεχνικής 3 (tf-idf)***

Τρόπος υπολογισμού βάρους κάθε λέξης	Ακρίβεια ταξινομητή
tf	91.7158%
tf-idf	90.2087%

Είναι γνωστό ότι με την εφαρμογή της τεχνικής tf-idf, το βάρος των λέξεων που εμφανίζονται σε λίγα διαφορετικά έγγραφα αυξάνεται, ενώ το βάρος των λέξεων που εμφανίζονται σε πολλά διαφορετικά έγγραφα μειώνεται. Παρατηρώντας τις εμφανίσεις των λέξεων, καταλήξαμε στο συμπέρασμα ότι η τεχνική tf-idf δεν ήταν αποτελεσματική για δυο λόγους. Ο πρώτος λόγος είναι ότι οι λέξεις με τις λιγότερες εμφανίσεις σε διαφορετικά έγγραφα αποτελούσαν μη αντιπροσωπευτικές λέξεις και ο δεύτερος λόγος είναι ότι οι πιο αντιπροσωπευτικές λέξεις βρίσκονταν σε παρόμοιο αριθμό εγγράφων με πολλά stopwords.

5.4.4 Εφαρμογή τεχνικής 4 (Stemming)

Η εφαρμογή της τεχνικής αυτής θα γίνει τόσο όταν τα βάρη των λέξεων υπολογίζονται με την τεχνική tf, όσο και όταν υπολογίζονται με την τεχνική tf-idf. Με την εφαρμογή της τεχνικής αυτής στα training και test data του dataset 2, η ακρίβεια του ταξινομητή μειώθηκε (πίνακας 15).

Πίνακας 15: ***Εφαρμογή τεχνικής 4 (stemming)***

Χρήση stemming	Ακρίβεια ταξινομητή (tf)	Ακρίβεια ταξινομητή (tf-idf)
χωρίς stemming	91.7158%	90.2087%
με stemming	91.3057%	89.8395%

Η μείωση αυτή οφείλεται κυρίως στα λάθη του αλγορίθμου stemming, δηλαδή σε μετατροπή λέξεων με διαφορετική ρίζα στην ίδια λέξη. Λόγω της μείωσης της ακρίβειας, η τεχνική αυτή δε χρησιμοποιήθηκε σε νέες μετρήσεις.

5.4.5 Εφαρμογή τεχνικής 5 (Αφαίρεση stopwords με χρήση λιστών)

Αυτή η τεχνική θα εφαρμοστεί τόσο όταν χρησιμοποιείται η τεχνική tf για τον υπολογισμό των βαρών των λέξεων, όσο και όταν χρησιμοποιείται η τεχνική tf-idf.

Για την αξιολόγηση της τεχνικής αυτής, θα χρησιμοποιηθούν τρεις διαφορετικές λίστες που περιέχουν stopwords. Η πρώτη λίστα προέρχεται από τη χρήση του αναλυτή StandardAnalyzer στο Mahout. Ο συγκεκριμένος αναλυτής αφαιρεί 33 stopwords. Η δεύτερη λίστα αποτελείται από 183 stopwords και προέρχεται από το διαδίκτυο. Η τρίτη λίστα περιέχει 40 stopwords, τα οποία διαλέχτηκαν προσεχτικά από τη δεύτερη λίστα. Για την εφαρμογή της δεύτερης και τρίτης λίστας κατασκευάστηκε πρόγραμμα που αφαιρεί τα stopwords, μιας και δεν παρέχεται κάτι τέτοιο από τη γραμμή εντολών του Mahout.

Η συγκεκριμένη τεχνική εφαρμόστηκε μόνο στα training data και για την αξιολόγηση της χρησιμοποιήθηκε το dataset 2.

Πίνακας 16: ***Εφαρμογή τεχνικής 5 (αφαίρεση stopwords με χρήση λιστών)***

Λίστα που χρησιμοποιήθηκε	Ακρίβεια ταξινομητή (tf)	Ακρίβεια ταξινομητή (tf-idf)
δεν έγινε χρήση	91.7158%	90.2087%
λίστα 1	91.7566%	90.1182%
λίστα 2	92.0655%	90.0845%
λίστα 3 (3)	92.1294%	90.2336%

Με βάση τα αποτελέσματα του πίνακα 16, παρατηρούμε ότι με την αφαίρεση μερικών stopwords (χρησιμοποιώντας λίστες), βελτιώθηκε αρκετά η ακρίβεια του ταξινομητή στην περίπτωση χρήσης της τεχνικής tf, γεγονός που δε συμβαίνει όμως και με τη χρήση της τεχνική tf-idf.

Συγκεκριμένα, με τη χρήση της τεχνικής tf παρατηρούμε ότι την καλύτερη ακρίβεια την προσφέρει η αφαίρεση των stopwords της λίστας 3. Αυτή η λίστα όπως αναφέρθηκε, έχει

μόνο 40 stopwords, τα οποία προέρχονται από τα stopwords που περιέχει η λίστα 2. Ο λόγος που δίνει καλύτερα αποτελέσματα από την αφαίρεση των stopwords της λίστας 2, οφείλεται στο γεγονός ότι η λίστα 2 περιέχει λέξεις που δεν αποτελούν stopwords στη δική μας εφαρμογή. Μερικές από αυτές τις λέξεις είναι οι «no» και «not». Τα stopwords της λίστας 3 επιλέχθηκαν πολύ προσεχτικά, ώστε να αποτελούν πραγματικά stopwords στην εφαρμογή μας. Αυτό φάνηκε και μέσα από τα αποτελέσματα, αφού με την αφαίρεση 40 και μόνο stopwords, η ακρίβεια στον ταξινομητή ήταν συγκριτικά καλύτερη από την αφαίρεση 183 stopwords. Επίσης, παρατηρούμε πως η προκαθορισμένη λίστα που προσφέρει το Mahout (λίστα 1) δεν προσφέρει σχεδόν καμιά βελτίωση. Αυτό οφείλεται στο γεγονός ότι και αυτή η λίστα περιέχει stopwords που δεν αποτελούν stopwords στην εφαρμογή μας.

Όταν χρησιμοποιείται η τεχνική tf-idf, τα stopwords παίρνουν μικρό βάρος και συνεπώς δεν αποτελούν μεγάλο πρόβλημα. Έτσι όταν αφαιρούνται, δεν παρουσιάζεται μεγάλη αλλαγή στην ακρίβεια (πίνακας 16). Η μείωση στην ακρίβεια όταν χρησιμοποιούνται οι λίστες 1 και 2, οφείλεται στο γεγονός ότι περιέχουν και αντιπροσωπευτικές λέξεις των εγγράφων. Η λίστα 3 που περιέχει προσεχτικά επιλεγμένα stopwords παρουσιάζει μια μικρή βελτίωση στην ακρίβεια.

5.4.6 Εφαρμογή τεχνικής 6 (Μέγιστες εμφανίσεις σε διαφορετικά έγγραφα)

Όπως αναφέρθηκε και στην παρουσίαση των διαφορετικών τεχνικών, ένας διαφορετικός τρόπος διαγραφής των stopwords είναι με τη διαγραφή των λέξεων που υπάρχουν στο μεγαλύτερο ποσοστό των training εγγράφων.

Η εφαρμογή της τεχνικής αυτής, θα γίνει τόσο όταν τα βάρη των λέξεων υπολογίζονται με την τεχνική tf, όσο και όταν υπολογίζονται με την τεχνική tf-idf.

Πίνακας 17: ***Εφαρμογή τεχνικής 6 (μέγιστες εμφανίσεις σε διαφορετικά έγγραφα)***

Διαγραφή λέξεων που ξεπερνούν αυτό το ποσοστό εμφανίσεων	**Ακρίβεια ταξινομητή (tf)**	**Ακρίβεια ταξινομητή (tf-idf)**	**Διαγραφή λέξεων που ξεπερνούν αυτό το ποσοστό εμφανίσεων (συνέχεια)**	**Ακρίβεια ταξινομητή (tf)**	**Ακρίβεια ταξινομητή (tf-idf)**
100%	91.7158%	90.2087%	50%	91.6394%	90.207%
95%	91.7158%	90.2087%	45% (4)	91.7637%	90.2531%
90%	91.7158%	90.2087%	40%	91.7673%	90.2513%
85%	91.7229%	90.1999%	35%	91.6998%	90.2478%
80%	91.7229%	90.191%	30%	91.1886%	90.1271%
75%	91.7247%	90.1892%	25%	91.0537%	90.1324%
70%	91.7247%	90.1892%	20%	90.3614%	90.0241%
65%	91.7442%	90.2034%	15%	90.2833%	90.017%
60%	91.7442%	90.2034%	10%	89.9567%	89.891%
55%	91.7886%	90.2212%	5%	87.8089%	88.8615%

Εφαρμόζοντας λοιπόν την τεχνική αυτή, μόνο στα training data του dataset 2, παρατηρήθηκε ότι η ακρίβεια του ταξινομητή δε βελτιώνεται σημαντικά (πίνακας 17). Από αυτό συμπεραίνουμε ότι ο αριθμός των εγγράφων στα οποία βρίσκονται τα stopwords είναι παρόμοιος με αυτόν των σημαντικών λέξεων.

Στη συνέχεια, θελήσαμε να προβούμε σε συγκρίσεις μεταξύ της τεχνικής 5 και της τεχνικής 6. Παρατηρούμε λοιπόν, ότι η τεχνική 5 οδηγεί σε πολύ καλύτερη ακρίβεια από την τεχνική 6, όταν τα βάρη των διανυσμάτων υπολογίζονται με την τεχνική tf. Το ίδιο όμως δε συμβαίνει, όταν τα βάρη των διανυσμάτων υπολογίζονται με την τεχνική tf-idf, όπου η ακρίβεια που επιτυγχάνεται με τη χρήση της τεχνικής 5, είναι ελαφρώς χειρότερη. Όμως, ακόμη και στην περίπτωση αυτή, θα προτιμηθεί η χρήση της τεχνικής 5. Κι αυτό γιατί η χρήση της τεχνικής 6 οδηγεί στη διαγραφή ενός μεγάλου ποσοστού των λέξεων του λεξιλογίου, προκειμένου να επιτευχθεί η καλύτερη ακρίβεια. Σ' ένα επόμενο στάδιο, θα αποδειχθεί και πειραματικά, ότι αυτό δεν είναι επιθυμητό, αφού θα οδηγεί σε χειρότερα αποτελέσματα (πίνακας 19).

5.4.7 Εφαρμογή τεχνικής 7 (Ελάχιστες εμφανίσεις)

Η εφαρμογή της τεχνικής αυτής θα γίνει τόσο όταν τα βάρη των διανυσμάτων υπολογίζονται με την τεχνική tf, όσο και όταν υπολογίζονται με την τεχνική tf-idf. Κρίνεται σκόπιμο να αναφερθεί, ότι η εφαρμογή της τεχνικής 7, θα περιοριστεί μόνο στα training data του dataset 3.

Πίνακας 18: ***Εφαρμογή τεχνικής 7 (ελάχιστες εμφανίσεις)***

Ελάχιστες εμφανίσεις	**Ακρίβεια ταξινομητή (tf)**	**Ακρίβεια ταξινομητή (tf-idf)**	**Ελάχιστες εμφανίσεις (συνέχεια)**	**Ακρίβεια ταξινομητή (tf)**	**Ακρίβεια ταξινομητή (tf-idf)**
1	92.1294%	90.2336%	60	91.8968%	91.3324%
2	92.101%	90.5868%	70	91.8507%	91.304%
3	92.1081%	90.7803%	80	91.8223%	91.3199%
4	92.1116%	90.8655%	90	91.8258%	91.3057%
5	92.1063%	90.933%	100	91.8312%	91.2933%
10	92.0548%	91.1957%	120	91.7229%	91.2081%
15	92.0424%	91.1957%	140	91.6572%	91.1637%
20	92.0495%	91.2844%	160	91.572%	91.0963%
25	92.0264%	91.2702%	180	91.5241%	91.0714%
30	91.9838%	91.2933%	200	91.4371%	91.0768%
35	91.9749%	91.3359%	250	91.2915%	90.885%
40	91.9448%	91.3306%	300	91.1833%	90.8087%
45 (5)	91.9306%	91.3661%	350	91.0785%	90.7146%
50	91.9217%	91.3608%	400	90.9401%	90.6135%

Σύμφωνα με τα αποτελέσματα που παρουσιάζονται στον πίνακα 18, παρατηρούμε ότι όταν χρησιμοποιείται η τεχνική tf για τον υπολογισμό των βαρών των διανυσμάτων, η εφαρμογή

της τεχνικής αυτής δεν πρόσφερε καμιά αύξηση στην ακρίβεια του ταξινομητή. Συγκεκριμένα, παρατηρούμε ότι με την αφαίρεση των λέξεων με λίγες εμφανίσεις, η ακρίβεια του ταξινομητή παραμένει σχεδόν σταθερή. Στην αντίθετη περίπτωση, όταν δηλαδή αφαιρέθηκαν λέξεις με αρκετές εμφανίσεις, αυτό οδήγησε σε μείωση της ακρίβειας. Αυτό οφείλεται στο γεγονός, ότι με τη χρήση της τεχνικής tf για τον υπολογισμό των βαρών, οι λέξεις με λίγες εμφανίσεις παίρνουν πολύ μικρό βάρος και κατά συνέπεια η αφαίρεσή τους δεν παίζει κανένα ρόλο στην ακρίβεια του ταξινομητή. Αντίθετα, οι λέξεις με αρκετές εμφανίσεις παίρνουν μεγάλο βάρος και αποτελούν σημαντικές λέξεις ως προς την ταξινόμηση νέων εγγράφων και συνεπώς η αφαίρεσή τους οδηγεί σε μείωση της ακρίβειας. Παρόλο που η αφαίρεση λέξεων με λίγες εμφανίσεις δεν προσφέρει αύξηση στην ακρίβεια του ταξινομητή, έχει ως αποτέλεσμα τη μείωση του μεγέθους του dataset.

Ωστόσο, όταν χρησιμοποιείται η τεχνική tf-idf για τον υπολογισμό των βαρών των λέξεων, η εφαρμογή της τεχνικής αυτής έχει ως αποτέλεσμα την αύξηση της ακρίβειας του ταξινομητή. Λέξεις με λίγες εμφανίσεις παίρνουν μεγάλο βάρος όταν χρησιμοποιείται η τεχνική tf-idf και συνεπώς, η αφαίρεση τους οδηγεί σε καλύτερα αποτελέσματα. Κι αυτό γιατί τέτοιες λέξεις στην πραγματικότητα αποτελούν λέξεις με ορθογραφικά λάθη ή σπάνιες λέξεις, γεγονός που αποδεικνύει ότι λανθασμένα δόθηκε σ' αυτές μεγάλο βάρος. Όταν αφαιρούνται λέξεις με αρκετές εμφανίσεις παρατηρούμε μείωση στην αύξηση της ακρίβειας που πετύχαμε και αυτό οφείλεται στο γεγονός ότι αυτές οι λέξεις όντως αποτελούν σημαντικές λέξεις και δεν πρέπει να αφαιρούνται.

Στη συνέχεια θέλοντας να συγκρίνουμε τις τεχνικές 5 και 6, χρησιμοποιήσαμε και το dataset 4 για την αξιολόγηση αυτής της τεχνικής.

Πίνακας 19: ***Σύγκριση μεταξύ τεχνικών 5 και 6***

Ελάχιστες εμφανίσεις	**Ακρίβεια ταξινομητή με dataset 3 (tf-idf)**	**Ακρίβεια ταξινομητή με dataset 4 (tf-idf)**	**Ελάχιστες εμφανίσεις (συνέχεια)**	**Ακρίβεια ταξινομητή με dataset 3 (tf-idf)**	**Ακρίβεια ταξινομητή με dataset 4 (tf-idf)**
1	90.2336%	90.2531%	60	91.3324%	91.2809%
2	90.5868%	90.5584%	70	91.304%	91.2489%
3	90.7803%	90.7643%	80	91.3199%	91.256%
4	90.8655%	90.8744%	90	91.3057%	91.2241%
5	90.933%	90.9489%	100	91.2933%	91.2152%
10	91.1957%	91.1904%	120	91.2081%	91.1389%
15	91.1957%	91.162%	140	91.1637%	91.0998%
20	91.2844%	91.2188%	160	91.0963%	91.0697%
25	91.2702%	91.2028%	180	91.0714%	91.0501%
30	91.2933%	91.2081%	200	91.0768%	91.0058%
35	91.3359%	91.2489%	250	90.885%	90.8584%
40	91.3306%	91.2702%	300	90.8087%	90.775%
45	91.3661%	91.2933%	350	90.7146%	90.6454%
50	91.3608%	91.2898%	400	90.6135%	90.5283%

Σύμφωνα με τα αποτελέσματα που παρουσιάζονται στον πίνακα 19, επιβεβαιώνουμε το συμπέρασμα ότι αν και η τεχνική 6 επιτυγχάνει καλύτερη ακρίβεια από την τεχνική 5 όταν χρησιμοποιείται η τεχνική tf-idf, δεν πρέπει να χρησιμοποιηθεί, μιας και στη συνέχεια οδηγεί σε χειρότερα αποτελέσματα.

5.4.8 Εφαρμογή τεχνικής 8 (Ελάχιστες εμφανίσεις σε διαφορετικά έγγραφα)

Η εφαρμογή της τεχνικής αυτής θα γίνει τόσο όταν τα βάρη των διανυσμάτων υπολογίζονται με την τεχνική tf, όσο και όταν υπολογίζονται με την τεχνική tf-idf. Η τεχνική αυτή εφαρμόστηκε μόνο στα training data του dataset 3.

Πίνακας 20: ***Εφαρμογή τεχνικής 8 (ελάχιστες εμφανίσεις σε διαφορετικά έγγραφα)***

Ελάχιστες εμφανίσεις σε διαφορετικά έγγραφα	Ακρίβεια ταξινομητή (tf)	Ακρίβεια ταξινομητή (tf-idf)	Ελάχιστες εμφανίσεις σε διαφορετικά έγγραφα (συνέχεια)	Ακρίβεια ταξινομητή (tf)	Ακρίβεια ταξινομητή (tf-idf)
1	92.1294%	90.2336%	60	91.8844%	91.2418%
2	92.101%	90.6135%	70	91.8223%	91.201%
3	92.1223%	90.8016%	80	91.8258%	91.2276%
4	92.1223%	90.9063%	90	91.7992%	91.1726%
5	92.1081%	90.9916%	100	91.7779%	91.162%
10	92.0548%	91.2063%	120	91.6679%	91.0448%
15	92.0246%	91.2081%	140	91.6004%	91.004%
20	92.0459%	91.2578%	160	91.5205%	90.9294%
25	92.0104%	91.2436%	180	91.4548%	90.8886%
30	91.9607%	91.2489%	200	91.3554%	90.7945%
35	91.9661%	91.2898%	250	91.2347%	90.7075%
40	91.9164%	91.2986%	300	91.0608%	90.5016%
45	91.9164%	91.3057%	350	90.9152%	90.3437%
50	91.9075%	91.2791%	400	90.846%	90.1945%

Τα συμπεράσματα που εξάγουμε για την τεχνική αυτή είναι τα ίδια με αυτά της τεχνικής 7. Ωστόσο, η τεχνική αυτή παρουσιάζει ελαφρώς μικρότερη αύξηση στην ακρίβεια, γεγονός που απέτρεψε τη χρήση στη συνέχεια.

5.4.9 Εφαρμογή τεχνικής 9 (Κανονικοποίηση)

Η εφαρμογή της συγκεκριμένης τεχνικής θα γίνει μόνο όταν τα βάρη των διανυσμάτων υπολογίζονται με τη χρήση της τεχνικής tf-idf. Κι αυτό γιατί, η συγκεκριμένη τεχνική δεν υποστηρίζεται από το Mahout, όταν τα βάρη των διανυσμάτων υπολογίζονται με τη χρήση της τεχνικής tf.

Πίνακας 21: ***Εφαρμογή τεχνικής 9 (κανονικοποίηση)***

Νόρμα που εφαρμόστηκε	**Ακρίβεια ταξινομητή**	**Νόρμα που εφαρμόστηκε (συνέχεια)**	**Ακρίβεια ταξινομητή**
Χωρίς κανονικοποίηση	91.3661%	4-norm	91.7158%
1-norm (6)	91.8809%	5-norm	91.7122%
2-norm	91.769%	6-norm	91.7193%
3-norm	91.73%	infinite-norm	91.7069%

Με την εφαρμογή λοιπόν της τεχνικής 9 μόνο στα training data του dataset 5, παρατηρείται αύξηση της ακρίβειας του ταξινομητή (πίνακας 21). Αυτό οφείλεται στο γεγονός ότι τα έγγραφα στα training data έχουν διαφορετικά μεγέθη. Παρατηρούμε επίσης, ότι η χρήση της 1-norm οδήγησε στα καλύτερα αποτελέσματα. Από αυτό συμπεραίνουμε ότι, αν και τα έγγραφα στα training data έχουν διαφορετικά μεγέθη, η διαφορά τους δεν είναι πολύ μεγάλη. Με άλλα λόγια, το μεγαλύτερο ποσοστό των training εγγράφων έχει παρόμοιο μέγεθος.

5.4.10 Εφαρμογή τεχνικής 10 (Information Gain/Chi-Squared/Bi-Normal Separation)

Η εφαρμογή των τεχνικών αυτών θα γίνει μόνο όταν χρησιμοποιείται η τεχνική tf για τον υπολογισμό των βαρών των διανυσμάτων. Οι συγκεκριμένες τεχνικές αφήνουν μόνο τις πιο σημαντικές λέξεις στα έγγραφα και συνεπώς, η χρήση της τεχνικής tf-idf δεν έχει νόημα. Με άλλα λόγια, οι λέξεις που εμφανίζονται πιο συχνά στα έγγραφα μετά τη χρήση των τεχνικών αυτών είναι και οι πιο αντιπροσωπευτικές των εγγράφων και συνεπώς, δεν πρέπει να γίνεται μείωση του βάρους τους (τεχνική tf-idf).

Το dataset που χρησιμοποιήθηκε για την αξιολόγηση των τεχνικών αυτών είναι το dataset 2. Ο λόγος που χρησιμοποιήθηκε αυτό το dataset, αντί του dataset 3, είναι γιατί αυτές οι τεχνικές μπορούν να αναγνωρίσουν αν μια λέξη είναι stopword ή όχι. Λόγω του ότι δεν υπάρχει τρόπος επιλογής του αριθμού των λέξεων που πρέπει να αφήσουν στο λεξιλόγιο, οι συγκεκριμένες τεχνικές εφαρμόστηκαν για πολλά διαφορετικά μεγέθη.

Με βάση τα αποτελέσματα που παρουσιάζονται στον πίνακα 22, παρατηρούμε ότι οι τεχνικές Information Gain και Chi-Squared δεν κατάφεραν να βελτιώσουν αισθητά την ακρίβεια του ταξινομητή. Ωστόσο, κατάφεραν με μόνο λίγες διαφορετικές λέξεις να πετύχουν αντίστοιχη ακρίβεια μ' αυτή, στην οποία γίνεται χρήση όλου του λεξιλογίου. Είναι εντυπωσιακό το γεγονός, πως με μόνο 500 διαφορετικές λέξεις από τις 84324 που περιέχονται στο λεξιλόγιο, οι δύο αυτές τεχνικές κατάφεραν να πετύχουν πάνω από 90% ακρίβεια στα test data. Επιπρόσθετα, κρίνεται σκόπιμο να αναφερθεί ότι με τη χρήση της τεχνικής Bi-Normal Separation, η ακρίβεια του ταξινομητή βελτιώθηκε σημαντικά. Για να πετύχει τα συγκεκριμένα αποτελέσματα η τεχνική αυτή, έκανε χρήση 6000 διαφορετικών λέξεων. Ωστόσο όμως, όταν η τεχνική Bi-Normal Separation κάνει χρήση λιγότερων από 1000 διαφορετικές λέξεις, τότε η απόδοσή της είναι συγκριτικά χαμηλότερη, σε σχέση με τις άλλες δύο τεχνικές.

Πίνακας 22: ***Εφαρμογή τεχνικής 10 (information gain, chi-squared και bi-normal separation)***

Λέξεις που απέμειναν στο λεξιλόγιο	Ακρίβεια ταξινομητή (Information Gain)	Ακρίβεια ταξινομητή (Chi-Squared)	Ακρίβεια ταξινομητή (Bi-Normal Separation)
100	88.0503%	87.9793%	63.3911%
500	90.4377%	90.3738%	83.7599%
1000	91.0377%	91.0324%	88.0964%
2000	91.4566%	91.4637%	91.4744%
4000	91.6252%	91.6643%	91.9412%
6000	91.6039%	91.6696%	92.3743%
8000	91.6554%	91.6998%	91.9288%
10000	91.6607%	91.7229%	91.8187%
12000	91.6519%	91.7176%	91.7584%
14000	91.6394%	91.7069%	91.7424%
16000	91.6448%	91.7477%	91.7318%
18000	91.6128%	91.7176%	91.6448%
20000	91.6252%	91.7353%	91.5951%
25000	91.6572%	91.746%	91.6412%
30000	91.6377%	91.7264%	91.6377%
35000	91.7087%	91.7264%	91.7087%
40000	91.698%	91.73%	91.698%
45000	91.7193%	91.7335%	91.7193%
50000	91.7176%	91.7353%	91.7176%
60000	91.7318%	91.7424%	91.7318%
70000	91.7229%	91.6945%	91.7229%
80000	91.7264%	91.714%	91.7264%
84324 (όλες)	91.7158%	91.7158%	91.7158%

Γενικά στην εφαρμογή μας, οι τεχνικές αυτές μπορούν να μειώσουν το μέγεθος του training set χωρίς αυτό να συνεπάγεται και μείωση της ακρίβειας. Επίσης, όπως παρατηρήθηκε αυξάνουν και την ακρίβεια ταξινόμησης σε κάποιες περιπτώσεις. Παρόλο αυτά όμως, οι εν λόγω τεχνικές όταν συνδυαστούν με άλλες τεχνικές, τότε δεν παρουσιάζεται περεταίρω αύξηση στην ακρίβεια λόγω της μεγάλης μείωσης του μεγέθους των training data.

Στη συνέχεια, θέλοντας να δούμε ποιες λέξεις θεωρούνται πιο σημαντικές απ' αυτές τις μεθόδους, κατασκευάσαμε πρόγραμμα που παρουσιάζει τις εμφανίσεις όλων των λέξεων του dataset της εισόδου. Στον πίνακα 23 παρουσιάζονται οι 5 πιο σημαντικές λέξεις που υπάρχουν στα training data του dataset 2 και οι 5 πιο σημαντικές λέξεις που υπάρχουν στα training data του dataset 2 μετά την εφαρμογή της Bi-Normal Separation για επιλογή των 6000 καλύτερων διαφορετικών λέξεων.

Πίνακας 23: ***Πέντε πιο σημαντικές λέξεις στα training data όταν γίνεται ή όχι εφαρμογή της τεχνικής Bi-Normal Separation***

Με χρήση της τεχνικής	Χωρίς τη χρήση της τεχνικής
to	the
was	and
good	a
not	to
great	was

Από τον πίνακα 23 φαίνεται, ότι ανάμεσα στις πιο σημαντικές λέξεις μετά την εφαρμογή της Bi-Normal Separation, είναι και λέξεις που εύκολα γίνεται αντιληπτό, ότι αποτελούν αντιπροσωπευτικές λέξεις για τις κατηγορίες «Θετικό» και «Αρνητικό». Οι λέξεις αυτές είναι το «good», «not» και «great» και οι οποίες βρίσκονται πολύ πιο κάτω στη λίστα, όταν δε χρησιμοποιείται η τεχνική αυτή.

Στη συνέχεια, έγινε σύγκριση μεταξύ των 5 πιο σημαντικών λέξεων που υπολογίστηκαν από την κάθε τεχνική, επιλέγοντας 6000 διαφορετικές λέξεις από το λεξιλόγιο.

Πίνακας 24: ***Πέντε πιο σημαντικές λέξεις στα training data όταν γίνεται εφαρμογή των διαφορετικών τεχνικών της τεχνικής 10***

Bi-Normal Separation	Information Gain	Chi-Squared
to	the	the
was	and	and
good	to	to
not	was	was
great	we	we

Σύμφωνα με τα αποτελέσματα του πίνακα 24, παρατηρούμε ότι οι τεχνικές Information Gain και Chi-Squared επιλέγουν τις ίδιες 5 πιο σημαντικές λέξεις. Οι λέξεις αυτές, στην περίπτωση που δε χρησιμοποιείται καμιά απ' αυτές τις τεχνικές, παρουσιάζουν μια και μόνο διαφορά, τη λέξη «a». Παρατηρούμε επίσης πως καμία από τις λέξεις που υπάρχουν δε φαίνεται να είναι αντιπροσωπευτική για τις κατηγορίες «Θετικό» και «Αρνητικό». Οι λέξεις «good», «not» και «great» βρίσκονται λίγο πιο κάτω στην λίστα τους, αλλά όχι τόσο κάτω, όσο όταν δε χρησιμοποιείται καμία από αυτές τις τεχνικές. Από αυτή τη σύγκριση γίνεται αντιληπτό γιατί η Bi-Normal Separation έχει καλύτερη ακρίβεια από τις άλλες δύο μεθόδους. Η Bi-Normal Separation καταφέρνει να εντοπίσει καλύτερα τις πιο αντιπροσωπευτικές λέξεις που περιέχονται στα έγγραφα στην περίπτωσή μας.

5.4.11 Εφαρμογή τεχνικής 11 (N-gram)

Αυτή είναι η τελευταία τεχνική που θα δοκιμάσουμε. Η εφαρμογή της τεχνικής αυτής θα γίνει τόσο όταν τα βάρη των λέξεων υπολογίζονται με την τεχνική tf, όσο και όταν υπολογίζονται με την τεχνική tf-idf. Έγινε λοιπόν εφαρμογή της τεχνικής αυτής στα training και test data του dataset 3 και 6 για την τεχνική tf και tf-idf αντίστοιχα.

Πίνακας 25: ***Εφαρμογή τεχνικής 11 (n-gram)***

Maximum grams	LLR (log-likelihood ratio)	Ακρίβεια ταξινομητή (tf)	Ακρίβεια ταξινομητή (tf-idf)
1	δεν λειτουργεί	92.1294%	91.8809%
2	5	93.5281%	93.5885%
2	10	93.5547%	93.5814%
2	25	93.4944%	93.5938%
2	50	93.45%	93.6133%
2	100	93.3364%	93.5228%
3	5	92.8642%	93.7589%
3	10 (7)	93.1057%	93.7749%
3	25	93.4571%	93.7695%
3	50	93.4021%	93.766%
3	100	93.2725%	93.7305%
4	5	92.7856%	93.7731%
4	10	93.0029%	93.7482%
4	25	93.3531%	93.7092%
4	50	93.2992%	93.7038%
4	100	93.1696%	93.6861%
5	5	92.8156%	93.8157%
5	10	93.1346%	93.8121%
5	25	93.373%	93.7624%
5	50	93.3242%	93.7713%
5	100	93.1461%	93.734%

Με βάση τα αποτελέσματα που παρουσιάζονται στον πίνακα 25, παρατηρούμε ότι η τεχνική αυτή, βελτίωσε πάρα πολύ την ακρίβεια του ταξινομητή τόσο όταν χρησιμοποιείται η τεχνική tf για τον υπολογισμό των βαρών των διανυσμάτων, όσο και όταν χρησιμοποιείται η τεχνική tf-idf. Από αυτό συμπεραίνουμε ότι στα έγγραφα υπάρχουν πολλές λέξεις, οι οποίες είναι εξαρτημένες μεταξύ τους.

Στην περίπτωση χρήσης της τεχνικής τεχνική tf για τον υπολογισμό των βαρών των διανυσμάτων, παρατηρούμε ότι η καλύτερη ακρίβεια επιτυγχάνεται όταν γίνεται χρήση μέχρι και 2-grams για την εκπαίδευση του ταξινομητή. Επίσης, παρατηρούμε ότι η τεχνική LLR επηρεάζει αρκετά τα αποτελέσματα. Η καλύτερη ακρίβεια επιτυγχάνεται όταν διαγράφονται τα πιο ασήμαντα grams από τα training data (τιμή>5). Όταν χρησιμοποιούνται μόνο τα πιο σημαντικά grams, παρατηρούμε ότι η ακρίβεια μειώνεται (τιμή=100). Αυτό μας οδηγεί στο συμπέρασμα ότι και τα μεσαίας σημαντικότητας grams συμβάλουν στη βελτίωση της ακρίβειας.

Όσον αφορά τη χρήση της τεχνικής tf-idf για τον υπολογισμό των βαρών των διανυσμάτων, παρατηρούμε ότι η καλύτερη ακρίβεια επιτυγχάνεται, όταν γίνεται χρήση μέχρι και 5 grams για την εκπαίδευση του ταξινομητή. Περιορισμός όμως στην επίτευξη της εν λόγω ακρίβειας είναι ο χρόνος και ο χώρος. Οπόταν, θα προτιμήσουμε να χρησιμοποιήσουμε μέχρι και 3-

grams που προσφέρουν παρόμοια ακρίβεια, αλλά σε πιο λογικό χρόνο και χώρο. Επίσης παρατηρούμε ότι η ακρίβεια στον ταξινομητή δεν επηρεάζεται πολύ από την εφαρμογή της τεχνικής LLR (δεν παρατηρούνται τόσες διακυμάνσεις όσες πριν). Ωστόσο, όμοια με την εφαρμογή της τεχνικής tf, η καλύτερη ακρίβεια στον ταξινομητή επιτυγχάνεται με τη διαγραφή των πιο ασήμαντων grams από τα training data.

5.5 Μια ακόμη δοκιμή

Σκεφτήκαμε πως θα ήταν καλό να δοκιμάσουμε τον ταξινομητή σε test data που περιέχουν μόνο τις βαθμολογίες 1 και 5, παρά τις βαθμολογίες 1, 2, 4 και 5. Αυτή η δοκιμή θα μας δώσει μια πιο αντιπροσωπευτική ακρίβεια για τον ταξινομητή, μιας και οι βαθμολογίες 1 και 5 είναι πιο πιθανό να είναι καταταγμένες σωστά στις κατηγορίες «Θετικό» και «Αρνητικό» από τις βαθμολογίες 2 και 4 (υποενότητα 5.3.2). Ο ταξινομητής μπορεί να κατατάσσει σωστά μια κριτική, αλλά λόγω της λάθος ετικέτας που έχει αυτή, να θεωρείται πως ο ταξινομητής έκανε λάθος.

Για την αξιολόγηση αυτής της ιδέας, χρησιμοποιήθηκε το dataset 7, δηλαδή το dataset που έχει προσφέρει τη μεγαλύτερη δυνατή ακρίβεια μέχρι στιγμής.

Πίνακας 26: ***Χρήση μόνο βαθμολογιών 1 και 5 στα test data***

Βαθμολογίες που χρησιμοποιήθηκαν στα training data	**Βαθμολογίες που χρησιμοποιήθηκαν στα test data**	**Ακρίβεια ταξινόμησης**
1, 2, 4, 5	1, 2, 4, 5	93.7749%
1, 2, 4, 5	1, 5	96.3464%

Με βάση τα αποτελέσματα που παρουσιάζονται στον πίνακα 26, καταλήγουμε στο συμπέρασμα πως ο ταξινομητής θα είναι ακόμη καλύτερος στην πράξη, αφού κατάφερε να ταξινομήσει σωστά 96.3464% των test εγγράφων σε αυτή την περίπτωση.

5.6 Σύνοψη ενότητας

Στην ενότητα αυτή, έγινε παρουσίαση της εφαρμογής εξαγωγής συναισθήματος που μελετάται σ' αυτήν την εργασία. Σκοπός της συγκεκριμένης εφαρμογής είναι η εξαγωγή του συναισθήματος που κρύβουν κριτικές χρηστών και η κατάταξή τους σε «Θετικές» και «Αρνητικές». Στη συνέχεια, έγινε εφαρμογή του αλγορίθμου Naive Bayes και των διάφορων μεθόδων βελτίωσης της ακρίβειας που παρουσιάστηκαν, με σκοπό την επίτευξη όσον το δυνατό μεγαλύτερης ακρίβειας και την εξαγωγή συμπερασμάτων. Αρχικά, έγινε επιλογή κατάλληλων δεδομένων για την εκπαίδευση του ταξινομητή και στη συνέχεια εφαρμογή γνωστών τεχνικών για την αύξηση της ακρίβειάς του. Συγκεκριμένα, σε ένα πρώτο στάδιο έγινε επιλογή κατάλληλων εκπαιδευτικών εγγράφων μέσα από πειραματική διαδικασία. Η επιλογή αυτή βελτίωσε την ακρίβεια ταξινόμησης νέων κριτικών από περίπου 83% σε 90.1537%. Σ' ένα δεύτερο στάδιο, έγινε χρήση αρκετών τεχνικών με στόχο την περαιτέρω βελτίωση της ακρίβειας. Συγκεκριμένα, επιτεύχθηκε ακρίβεια 93.7749%, στον ταξινομητή, όταν έγινε μετατροπή των γραμμάτων σε πεζά, αφαίρεση των σύμβολων, των αριθμών, των stopwords με τη χρήση λιστών (λίστα 3) και των λέξεων με τις ελάχιστες εμφανίσεις

(τουλάχιστον 45 εμφανίσεις). Σε συνδυασμό με τη χρήση της κανονικοποίησης (1-norm), της τεχνικής n-gram (3-grams και LLR 10) και της τεχνικής tf-idf για τον υπολογισμό των βαρών των διανυσμάτων. Ωστόσο, αν και η συγκεκριμένη ακρίβεια ταξινόμησης είναι πολύ ικανοποιητική, στην επόμενη ενότητα μελετάται η επίδραση του αριθμού των εκπαιδευτικών εγγράφων στην εφαρμογή αυτή, με σκοπό την επίτευξη ακόμη μεγαλύτερης ακρίβειας ταξινόμησης.

6

Big Data

Όπως αναφέρθηκε και στην εισαγωγή αυτής της διπλωματικής εργασίας, η ραγδαία αύξηση των δεδομένων που υπάρχουν στο διαδίκτυο έχει οδηγήσει πολλές εφαρμογές στο να απαιτούν να γίνει χρήση μεγάλου όγκου δεδομένων τόσο κατά την εκπαίδευση, όσο και κατά τη χρήση του ταξινομητή στην κατάταξη νέων εγγράφων σε κατηγορίες.

Στην ενότητα αυτή, αρχικά εξηγείται γιατί η χρήση μεγάλου όγκου δεδομένων είναι επιθυμητή από τις σύγχρονες εφαρμογές και πως η χρήση τέτοιου όγκου δεδομένων κάνει αναγκαία τη χρήση κατανεμημένων ταξινομητών. Στη συνέχεια, προκειμένου να επιβεβαιωθεί η χρησιμότητα των κατανεμημένων ταξινομητών, αξιολογείται η χρήση του κατανεμημένου αλγορίθμου Naive Bayes που προσφέρει το Mahout και τέλος μελετάται πως ο αριθμός των εκπαιδευτικών εγγράφων επηρεάζει την ακρίβεια ταξινόμησης της εφαρμογής που μελετήθηκε στην ενότητα 5.

6.1 Χρησιμότητα μεγάλου όγκου δεδομένων και αναγκαιότητα χρήσης κατανεμημένων αλγορίθμων ταξινόμησης

Η χρήση μεγάλου όγκου δεδομένων είναι επιθυμητή και σε αρκετές εφαρμογές αναγκαία τόσο κατά την εκπαίδευση, όσο και κατά τη χρήση του ταξινομητή στην κατάταξη νέων εγγράφων σε κατηγορίες. Συγκεκριμένα, όσα περισσότερα εκπαιδευτικά έγγραφα χρησιμοποιηθούν κατά τη διάρκεια εκπαίδευσης του ταξινομητή, τόσο μεγαλύτερη γνώση αποκτά για τη συγκεκριμένη εφαρμογή και αυτό έχει ως αποτέλεσμα να μπορεί να κατατάσσει νέα έγγραφα στη σωστή κατηγορία με περισσότερη ακρίβεια. Επίσης, λόγω της διαφορετικότητας που παρουσιάζει κάθε εκπαιδευτικό έγγραφο, η γνώση του ταξινομητή διευρύνεται με αποτέλεσμα να μπορεί να κατατάξει ένα μεγαλύτερο φάσμα από νέα έγγραφα στη σωστή κατηγορία. Ωστόσο, κρίνουμε σκόπιμο να αναφέρουμε πως η χρήση περισσότερων εγγράφων δεν οδηγεί πάντα σε καλύτερη ακρίβεια. Μπορούν να οδηγήσουν σε υπερεκπαίδευση του ταξινομητή, δηλαδή να υπολογιστούν με τέτοιο τρόπο οι παράμετροί του, που και η παραμικρή διαφορά ανάμεσα στα training και test data να οδηγήσει σε λάθος ταξινόμηση. Ο ιδανικός αριθμός των εκπαιδευτικών εγγράφων που

απαιτείται για την εκπαίδευση του ταξινομητή εξαρτάται καθαρά από την εκάστοτε εφαρμογή.

Αντίστοιχα, η χρήση του εκπαιδευμένου ταξινομητή στην κατάταξη περισσότερων νέων έγγραφων οδηγεί σε πιο ακριβή συμπεράσματα. Για παράδειγμα κατατάσσοντας περισσότερες κριτικές χρηστών για ένα ξενοδοχείο στις κατηγορίες «Θετικό» και «Αρνητικό», αποκτάται μια καλύτερη εικόνα για την άποψη όλων των επισκεπτών του ξενοδοχείου. Αν κατατάσσαμε μόνο λίγες κριτικές χρηστών, αυτό θα είχε ως αποτέλεσμα να μην έχουμε μια καλή εικόνα για το ξενοδοχείο, μιας και οι υπόλοιποι επισκέπτες μπορεί να είχαν μια εντελώς διαφορετική άποψη, η οποία δε λήφθηκε υπόψη.

Όπως έγινε κατανοητό, η χρήση μεγάλου όγκου δεδομένων είναι πολύ χρήσιμη και σε αρκετές περιπτώσεις απαραίτητη για την επιτυχία της εφαρμογής. Ωστόσο, η απαίτηση για μνήμη από τους κλασικούς (μη κλιμακώσιμους) αλγόριθμους ταξινόμησης αυξάνεται με πολύ γρήγορους ρυθμούς, με αποτέλεσμα την έλλειψη της μνήμης όταν γίνεται χρήση μεγάλου όγκου δεδομένων. Επίσης, η χρήση κλασικών ταξινομητών οδηγεί σε πάρα πολύ μεγάλους χρόνους εκτέλεσης. Μ' άλλα λόγια οι κλασικοί ταξινομητές χρειάζονται πάρα πολύ χρόνο για να μπορέσουν να εκπαιδευτούν με πολλά δεδομένα και επίσης χρειάζονται πάρα πολύ χρόνο για να ταξινομήσουν πολλά νέα έγγραφα. Και στις δύο όμως περιπτώσεις, στην εκπαίδευση δηλαδή αλλά και στη χρήση του ταξινομητή, ο χρόνος παίζει καθοριστικό παράγοντα. Είναι πολύ σημαντικό ο χρόνος εκπαίδευσης να είναι μικρός, γιατί όπως έγινε κατανοητό και από την ενότητα 5, η διαδικασία αυτή πρέπει να γίνει πάρα πολλές φορές μέχρι να βρεθούν οι ρυθμίσεις που δίνουν την καλύτερη ακρίβεια ταξινόμησης. Αντίστοιχα, και ο χρόνος κατάταξης νέων εγγράφων σε κατηγορίες, πρέπει να είναι όσον το δυνατό μικρότερος. Κι αυτό γιατί ολοένα και περισσότερες εφαρμογές καθημερινά απαιτούν ανάλυση σε πραγματικό χρόνο. Μόλις λοιπόν παρουσιαστούν νέα έγγραφα πρέπει να ταξινομούνται χωρίς καμιά καθυστέρηση, ώστε να πάρουν σειρά τα επόμενα, προκειμένου να εξαχθούν άμεσα τα συμπεράσματα, χωρίς καθυστερήσεις.

Κρίνουμε σκόπιμο σε αυτό το σημείο, να αναφέρουμε μερικά παραδείγματα σύγχρονων εφαρμογών που κάνουν χρήση μεγάλου όγκου δεδομένων και ο χρόνος εκπαίδευσης και χρήσης του ταξινομητή έχει καθοριστική σημασία. Ένα πρώτο τέτοιο παράδειγμα είναι όταν τα τηλεοπτικά κανάλια σε καιρό προεδρικών εκλογών, εξάγουν το συναίσθημα που κρύβουν τα μηνύματα των χρηστών στο Twitter για τους υποψήφιους προέδρους, με σκοπό να σχηματίσουν μια καλή εικόνα της άποψης του πληθυσμού για κάθε υποψήφιο [26]. Σύμφωνα με στατιστικά ([26]) όσο πλησιάζουν οι μέρες των προεδρικών εκλογών, τόσο και περισσότερα συμβάντα διαδραματίζονται. Αυτό έχει ως αποτέλεσμα, τα σχετικά μηνύματα να αυξάνονται πάρα πολύ και να φτάνουν τα μερικά εκατομμύρια καθημερινώς. Σ' αυτή την εφαρμογή ο χρόνος εξαγωγής του συναισθήματος από τα μηνύματα είναι πολύ κρίσιμος, καθώς τα τηλεοπτικά κανάλια θέλουν να μεταδίδουν αμέσως οποιαδήποτε εξέλιξη και αλλαγή στην άποψη του πληθυσμού. Λόγω του μεγάλου αριθμού όμως μηνυμάτων που λαμβάνονται συνεχώς, η χρήση κλασικών αλγορίθμων ταξινόμησης θα απαιτούσε πάρα πολύ χρόνο προκειμένου να κάνει την ανάλυση αυτή. Επίσης, λόγω των πολλών διαφορετικών τρόπων που συνηθίζουν οι χρήστες να εκφράζουν την άποψή τους στο Twitter (emoticons, συντομογραφίες κλπ) η χρήση μεγάλου όγκου εκπαιδευτικών εγγράφων είναι απαραίτητη

κατά την εκπαίδευση του ταξινομητή. Συνεπώς, αν δε γίνει χρήση κατανεμημένων αλγορίθμων ταξινόμησης, η βελτιστοποίηση της ακρίβειας του ταξινομητή θα χρειαστεί πάρα πολλές μέρες.

Παρόμοια, τα τηλεοπτικά κανάλια συνηθίζουν να αναλύουν τα συναισθήματα του πληθυσμού από μηνύματα στο Tweeter κατά τη διάρκεια ενός συνταρακτικού γεγονότος, με σκοπό να μεταδώσουν πως αυτό το γεγονός επηρέασε τον πληθυσμό. Και σ' αυτή την εφαρμογή, η χρήση μεγάλου όγκου δεδομένων είναι απαραίτητη και ο χρόνος παίζει καθοριστικό παράγοντα. Ένα τελευταίο παράδειγμα [27] είναι η περίπτωση των επιχειρήσεων που αναλύουν τα μηνύματα που προέρχονται από το Tweeter. Συγκεκριμένα, οι επιχειρήσεις αυτές έχουν ως σκοπό να μαθαίνουν την άποψη του κόσμου για τα προϊόντα τους αλλά και για τα προϊόντα αντίπαλων επιχειρήσεων, έτσι ώστε να προβαίνουν σε τυχόν άμεσες αλλαγές, με απώτερο στόχο την αύξηση των εισοδημάτων τους.

6.2 Αξιολόγηση κατανεμημένων ταξινομητών

Σ' αυτήν την υποενότητα θα γίνει χρήση του κατανεμημένου αλγορίθμου Naive Bayes που προσφέρει το Mahout όταν υπάρχει μεγάλος όγκος δεδομένων, με σκοπό να επιβεβαιωθεί η χρησιμότητα των κατανεμημένων αλγορίθμων ταξινόμησης του Mahout και η χρησιμότητα των κατανεμημένων αλγορίθμων ταξινόμησης γενικότερα. Για το σκοπό αυτό έγιναν αρκετές μετρήσεις με διαφορετικά μεγέθη δεδομένων και διαφορετικό αριθμό κόμβων. Στη διάθεσή μας είχαμε συνολικά 4 υπολογιστές (κόμβους) συνδεδεμένους μεταξύ τους σε τοπικό δίκτυο. Ο κάθε κόμβος αποτελείται από 8 πυρήνες (2 τετραπύρηνους Intel Xeon E5405 χρονισμένους στα 2GHz), 8GB μνήμη ram και περισσότερα από 200GB σκληρό δίσκο.

Όπως αναφέρθηκε, οι κατανεμημένοι αλγόριθμοι που προσφέρει το Mahout κάνουν χρήση του προγραμματιστικού μοντέλου Map Reduce. Ένα πρόβλημα που προκύπτει στην εκτέλεση αλγορίθμων που χρησιμοποιούν αυτό το προγραμματιστικό μοντέλο είναι ο αριθμός των map και reduce διεργασιών που μπορούν να εκτελεστούν παράλληλα, ώστε να επιτυγχάνεται όσο το δυνατό καλύτερη χρήση του υλικού και συνεπώς οι καλύτερες χρονικές επιδόσεις. Μ' άλλα λόγια πρέπει να υπάρχουν ταυτόχρονα αρκετές διεργασίες, ώστε όλο το διαθέσιμο υλικό να είναι σε λειτουργία, όχι όμως τόσες πολλές που να απαιτούν περισσότερο υλικό γιατί αυτό οδηγεί σε σημαντική μείωση της επίδοσης.

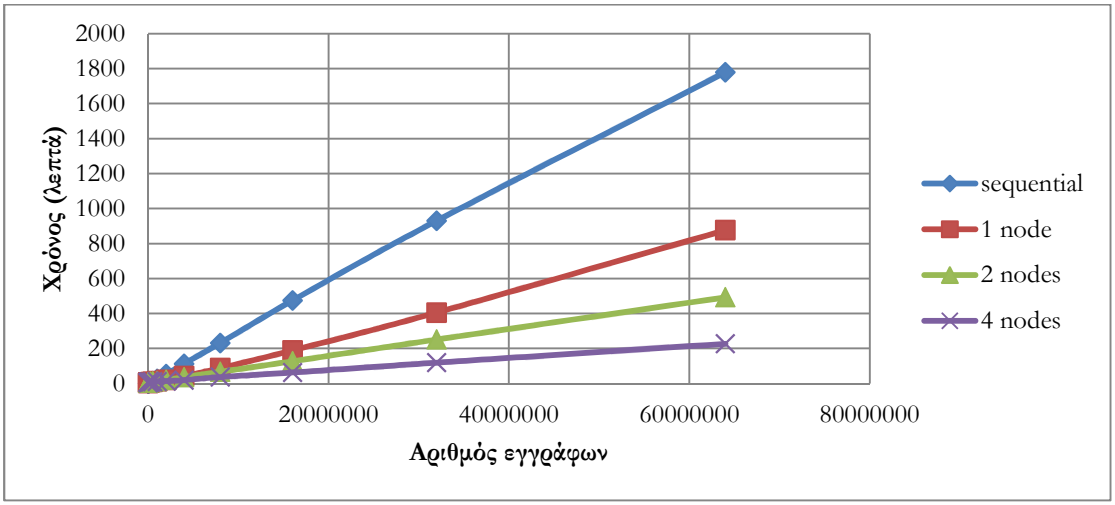
Η επιλογή του ιδανικού αριθμού map και reduce διεργασιών που μπορούν να εκτελούνται παράλληλα είναι πολύ δύσκολη, καθώς εξαρτάται από πολλούς παράγοντες. Εξαρτάται από τους διαθέσιμους πυρήνες, την ταχύτητα και τη διάταξη των δίσκων, τη διαθέσιμη μνήμη ram, το μέγεθος των δεδομένων εισόδου, το δίκτυο διασύνδεσης, το είδος των εργασιών, αλλά και από πολλούς άλλους παράγοντες. Όπως είναι φανερό, είναι σχεδόν αδύνατο να εξεταστούν όλοι αυτοί οι παράγοντες για να επιλεγεί ο ιδανικός αριθμός map και reduce διεργασιών που μπορούν να εκτελούνται παράλληλα, έτσι θα ληφθεί υπόψη μόνο ένα υποσύνολό τους, καθώς επίσης και γενικές κατευθυντήριες συμβουλές.

Για την επιλογή του αριθμού των map και reduce διεργασιών που θα εκτελούνται παράλληλα λήφθηκαν υπόψη τα εξής:

1. Κάθε κόμβος αποτελείται από 8GB μνήμη ram. Σ' κάθε slave node, η μνήμη αυτή θα χρησιμοποιείται από το λειτουργικό, τον datanode, τον tasktracker, τις map και reduce διεργασίες και από τυχόν άλλες διεργασίες που τρέχουν στο παρασκήνιο. Στον master node, η μνήμη ram θα χρησιμοποιείται επιπλέον από τον jobtracker, τον namenode και τον secondary namenode, μιας και αποφασίστηκε πως ο master node θα συμπεριφέρεται και ως slave node.
2. Ο datanode, ο tasktracker, ο jobtracker, ο namenode και ο secondary namenode είναι προκαθορισμένο να χρησιμοποιούν μέχρι και 1GB μνήμης ram ο καθένας. Μ' άλλα λόγια, στους slave nodes πρέπει να υπάρχουν ελεύθερα 2GB μνήμης ram, ενώ στον master node είναι απαραίτητη η ύπαρξη 5GB ελεύθερης μνήμης ram. Για τους slave nodes αυτό δεν αποτελεί πρόβλημα, γεγονός που δεν ισχύει και για τον master node. Ωστόσο, μετά την εκτέλεση παρόμοιων εργασιών μ' αυτές που θα εκτελεστούν στη συνέχεια, παρατηρήθηκε ότι όλες αυτές οι υπηρεσίες μαζί δε χρησιμοποιούν πάνω από 1.5GB μνήμης ram. Αν χρησιμοποιούσαν αρκετά μεγάλο ποσοστό της ram, θα υποχρεωνόμαστε να μη θεωρήσουμε τον master node και ως slave node.
3. Κάθε κόμβος αποτελείται από 8 πυρήνες, έτσι προτείνεται να μην εκτελούνται πολύ περισσότερες από 8 διεργασίες ταυτόχρονα.
4. Η reduce φάση κανονικά ξεκινά μετά το τέλος της map φάσης. Για λόγους ταχύτητας όμως, η reduce φάση αρκετές φορές ξεκινά μόλις παραχθούν τα πρώτα ενδιάμεσα αποτελέσματα της map φάσης. Συγκεκριμένα, ξεκινά η sort φάση, η οποία περιλαμβάνεται στην reduce φάση.
5. Συνήθως οι reduce διεργασίες κάνουν πολύ περισσότερες εγγραφές στο δίσκο από τις map διεργασίες. Για να αποφευχθεί λοιπόν το disk overhead (υπερχρησιμοποίηση του δίσκου) προτείνεται οι map διεργασίες να είναι περισσότερες από τις reduce διεργασίες.
6. Για να αποφευχθεί το disk overhead μπορεί επίσης να χρησιμοποιηθεί κάποια μέθοδος συμπίεσης των δεδομένων. Αυτό βοηθάει επίσης στο να μειωθεί το network overhead, δηλαδή ο επιπλέον χρόνος που χρειάζεται για τη μετακίνηση των δεδομένων από ένα κόμβο σ' ένα άλλο. Σε αντίθεση με το σημείο 5, εδώ απαιτείται οι map και reduce διεργασίες να είναι όσον το δυνατό περισσότερες, έτσι ώστε να μειωθεί όσο το δυνατό περισσότερο ο χρόνος που απαιτείται για τη συμπίεση και αποσυμπίεση των δεδομένων.
7. Κάθε διεργασία προτείνεται να έχει στη διάθεσή της μέχρι 512MB μνήμης ram.

Μετά από αρκετούς πειραματισμούς, και λαμβάνοντας υπόψη τα παραπάνω σημεία, καταλήξαμε στο συμπέρασμα ότι για τη χρήση ενός μόνο κόμβου, τα καλύτερα αποτελέσματα επιτυγχάνονται με τη χρήση μέχρι 8 map και 4 reduce διεργασίες ταυτόχρονα. Παρατηρήσαμε επίσης, ότι η συμπίεση των δεδομένων δε βοηθάει καθόλου, αντίθετα χειροτερεύει το χρόνο εκτέλεσης, και γι' αυτό δε χρησιμοποιήθηκε. Παράλληλα, όταν γίνεται χρήση περισσότερων από ένα κόμβους, τα καλύτερα αποτελέσματα επιτυγχάνονται με τη χρήση μέχρι 8 map και 8 reduce διεργασίες ανά κόμβο ταυτόχρονα. Στην περίπτωση αυτή, η συμπίεση των δεδομένων βοηθάει πάρα πολύ και χρησιμοποιήθηκε. Αυτό ήταν αναμενόμενο, αφού μειώνει τη μεγάλη χρονική επιβάρυνση που προκαλείται από την επικοινωνία των κόμβων μεταξύ τους.

Στη συνέχεια, θέλοντας να αξιολογήσουμε τις κατανεμημένες λύσεις, εκπαιδεύσαμε και δοκιμάσαμε τον Naive Bayes με τη χρήση της τεχνικής cross-validation (80%-20%). Για το σκοπό αυτό χρησιμοποιήθηκαν μέχρι 64 εκατομμύρια έγγραφα και καταγράφηκε ο ολικός χρόνος που χρειάστηκε ο αλγόριθμος για την εκπόνηση της εργασίας που του ανατέθηκε όταν εκτελέστηκε σειριακά (1 πυρήνας), σε 1 κόμβο (8 πυρήνες), σε 2 κόμβους (16 πυρήνες) και σε 4 κόμβους (32 πυρήνες). Στις 3 τελευταίες περιπτώσεις ο αλγόριθμος εκτελέστηκε πάνω από το Hadoop, ενώ στην πρώτη περίπτωση εκτελέστηκε τοπικά. Κρίνουμε σκόπιμο εδώ να αναφέρουμε πως ο ολικός χρόνος αποτελείται απ' το χρόνο μετατροπής των εγγράφων σε διανύσματα, το χρόνο χωρισμού των δεδομένων σε training και test set, το χρόνο εκπαίδευσης και το χρόνο αξιολόγησης. Η μετατροπή των εγγράφων σε διανύσματα κατέχει το μεγαλύτερο ποσοστό του ολικού χρόνου, η εκπαίδευση του ταξινομητή κατέχει το δεύτερο μεγαλύτερο, η αξιολόγηση το τρίτο και τέλος ο χωρισμός των δεδομένων το μικρότερο ποσοστό του ολικού χρόνου. Η μόνη προεπεξεργασία που υπέστηκαν τα δεδομένα είναι ανάλυση με τη χρήση του StandardAnalyzer αναλυτή που προσφέρει το Mahout. Καμιά άλλη τεχνική δεν εφαρμόστηκε.



Εικόνα 6: ***Χρόνος εκτέλεσης Naive Bayes σε συνάρτηση με το μέγεθος της εισόδου και τους χρησιμοποιούμενους κόμβους***

Όπως παρατηρείται από την εικόνα 6, όταν αυξάνονται οι πυρήνες που χρησιμοποιούνται για την εκτέλεση του Naive Bayes, ο χρόνος εκτέλεσης του μειώνεται σημαντικά. Συγκεκριμένα όταν χρησιμοποιείται μόνο ένας πυρήνας, ο χρόνος εκτέλεσης του Naive Bayes στις περιπτώσεις όπου η είσοδος είναι μεγαλύτερη από 1 εκατομμύριο έγγραφα είναι απαγορευτικά μεγάλος. Αντίθετα, όταν χρησιμοποιούνται περισσότεροι πυρήνες ο χρόνος εκτέλεσης αρχίζει να γίνεται αποδεκτός. Για παράδειγμα, ο χρόνος εκτέλεσης του Naive Bayes όταν εκτελείται σειριακά και η είσοδος είναι 64 εκατομμύρια έγγραφα είναι περίπου 1800 λεπτά, δηλαδή 30 ώρες, ενώ όταν εκτελείται σε 4 κόμβους είναι μόλις 200 λεπτά, δηλαδή 3 ώρες και 20 λεπτά.

Ωστόσο, η μείωση στο χρόνο δεν είναι ικανοποιητική για το υλικό που παρέχεται. Για παράδειγμα στον πίνακα 27 παρουσιάζεται το speedup για κάθε διαφορετική περίπτωση, όταν η είσοδος είναι 64 εκατομμύρια έγγραφα.

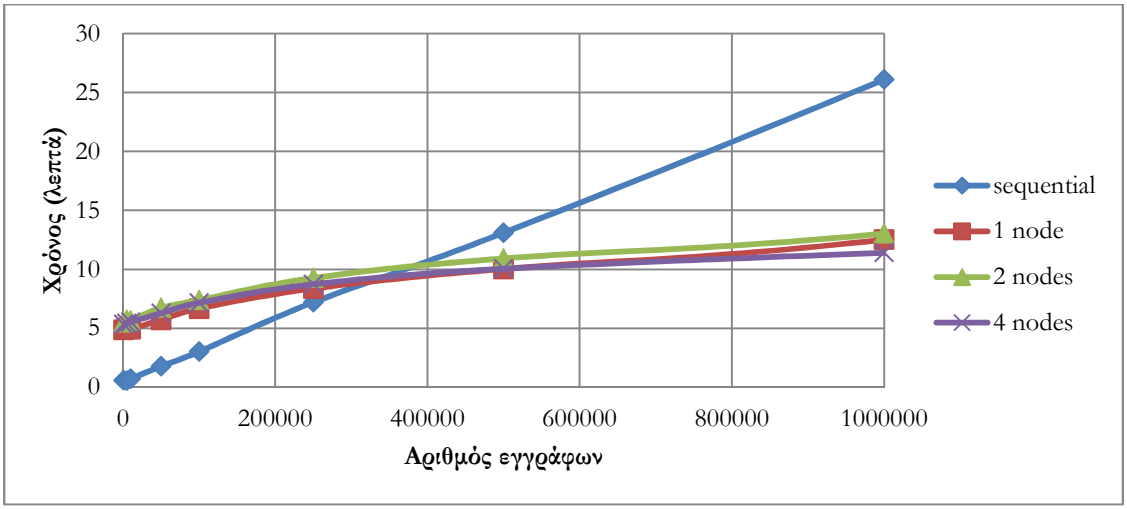
Πίνακας 27: ***Speedup όταν γίνεται χρήση 64 εκατομμυρίων εγγράφων***

Αλγόριθμος εκτελείται	Sequential	1 κόμβος	2 κόμβοι	4 κόμβοι
Speedup	1	2.03	3.61	7.85

Παρατηρούμε λοιπόν, ότι το speedup είναι μόλις 2.03 όταν ο αλγόριθμος εκτελείται σ' ένα κόμβο (8 πυρήνες), ενώ το αναμενόμενο είναι 8, 3.61 όταν ο αλγόριθμος εκτελείται σε 2 κόμβους, ενώ το αναμενόμενο είναι 16 και 7.85 όταν ο αλγόριθμος εκτελείται σε 4 κόμβους, ενώ το αναμενόμενο είναι 32. Αυτό οφείλεται στο γεγονός ότι η χρήση του Hadoop εισάγει μεγάλο overhead (επιπλέον καθυστερήσεις). Η όλη διαδικασία χωρισμού των δεδομένων και της δουλείας σε διεργασίες εισάγει μεγάλες καθυστερήσεις. Ειδικότερα, στην περίπτωση που ο αλγόριθμος εκτελείται σε περισσότερους από ένα κόμβους, εισάγεται και μεγάλη καθυστέρηση λόγω της επικοινωνίας μεταξύ των κόμβων. Ένας άλλος λόγος είναι ότι στους σειριακούς αλγόριθμους συνηθίζεται η εφαρμογή βελτιστοποιήσεων, οι οποίες δεν μπορούν να εφαρμοστούν στους κατανεμημένους αλγορίθμους.

Επίσης, από την εικόνα 6 φαίνεται πως παρά τους μεγάλους χρόνους εκτέλεσης που παρουσιάζει ο Naive Bayes όταν εκτελείται σειριακά, είναι γραμμικός σε σχέση με την είσοδο. Μ' άλλα λόγια όταν διπλασιάζεται η είσοδος, διπλασιάζεται και ο χρόνος εκτέλεσης. Αυτό είναι ένα πολύ καλό χαρακτηριστικό του συγκεκριμένου αλγορίθμου [10]. Στους περισσότερους αλγόριθμους όταν διπλασιάζεται η είσοδος, ο χρόνος εκτέλεσής τους αυξάνεται πολύ περισσότερο από 2 φορές [3]. Ο λόγος αυτός, σε συνδυασμό με την καλή κλιμακωσιμότητα που παρουσιάζει σε σχέση μ' άλλους αλγορίθμους, κάνουν τον αλγόριθμο κατάλληλο όταν τα δεδομένα είναι πάρα πολλά.

Στην εικόνα 6, δε διακρίνονται οι χρόνοι εκτέλεσης όταν τα έγγραφα που χρησιμοποιήθηκαν είναι λιγότερα από 1 εκατομμύριο, έτσι στην εικόνα 7 παρουσιάζεται η συγκεκριμένη περιοχή εστιασμένη.

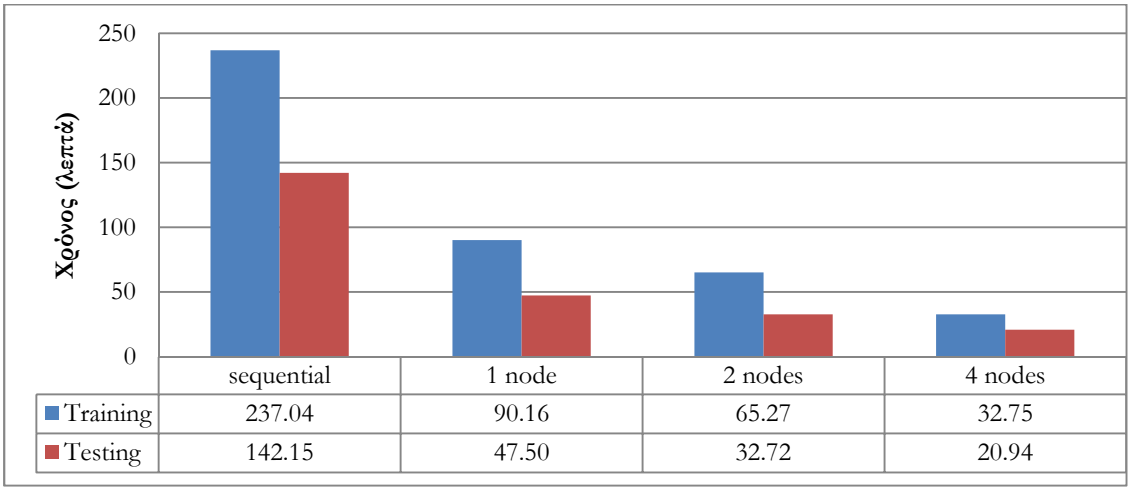


Εικόνα 7: ***Εστίαση στην περιοχή της εικόνας 6, όπου ο αριθμός των εγγράφων που χρησιμοποιείται είναι μικρός***

Στην εικόνα 7, παρατηρούμε κάτι το εντελώς διαφορετικό. Όταν ο αριθμός των εγγράφων που χρησιμοποιείται είναι μικρότερος από 400 χιλιάδες (≈200MB δεδομένα), η σειριακή

εκτέλεση του αλγορίθμου έχει το μικρότερο χρόνο εκτέλεσης. Αυτό οφείλεται στο γεγονός ότι η χρήση κατανεμημένων αλγορίθμων (πάνω από Hadoop) εισάγει μια αρχική καθυστέρηση, ανεξάρτητα από το πόσο μικρή είναι η είσοδος. Αυτή η καθυστέρηση όπως αναφέρθηκε και πριν, οφείλεται στο μοίρασμα των δεδομένων και της δουλείας στις διεργασίες και στην επικοινωνία μεταξύ των κόμβων. Ωστόσο, όταν τα έγγραφα ξεπεράσουν τις 400 χιλιάδες, η χρήση κατανεμημένων λύσεων υπερτερεί της σειριακής. Το σημείο αυτό, είναι το σημείο όπου παρά το αρχικό overhead που παρουσιάζουν οι κατανεμημένοι αλγόριθμοι, καταφέρνουν να υπερκαλύψουν αυτή τη διαφορά λόγω της μεγαλύτερης επεξεργαστικής ισχύς που κατέχουν. Ένα άλλο σημείο που θέλουμε να τονίσουμε είναι ότι αν και κανείς θα περίμενε η χρήση περισσότερων κόμβων να παρουσίαζε περισσότερο αρχικό overhead από αυτό που παρουσιάζουν λιγότεροι κόμβοι (π.χ. 4 κόμβοι και 2 κόμβοι) αυτό δε φαίνεται στην εικόνα 7. Αυτό οφείλεται στο γεγονός ότι το Hadoop θέτει τον αριθμό των map διεργασιών σύμφωνα με το μέγεθος της εισόδου και μόνο οι reduce διεργασίες μπορούν να επιλεχτούν από το χρήστη (και οι map μπορούν αν επιλεχθεί μικρότερο split size της εισόδου, αλλά δεν πειραματιστήκαμε μ' αυτή τη ρύθμιση) και συνεπώς η χρήση όλων των κόμβων που ανατέθηκαν δε γίνεται συνεχώς για μικρά μεγέθη.

Στη συνέχεια, θελήσαμε να εξετάσουμε το χρόνο που χρειάζεται ο ταξινομητής για την εκπαίδευση και την κατάταξη νέων εγγράφων σε κατηγορίες. Για το σκοπό αυτό έγιναν καινούργιες μετρήσεις στις οποίες χρησιμοποιήθηκαν 8 εκατομμύρια έγγραφα για την εκπαίδευση του ταξινομητή και 8 εκατομμύρια νέα έγγραφα για την αξιολόγησή του. Τα αποτελέσματα για κάθε περίπτωση φαίνονται στην εικόνα 8, η οποία συμπεριλαμβάνει και το χρόνο μετατροπής των εγγράφων σε διανύσματα σε κάθε περίπτωση, καθώς αυτή η διαδικασία αποτελεί τόσο μέρος της εκπαίδευσης (μετατροπή training data σε διανύσματα), όσο και της αξιολόγησης (μετατροπή test data σε διανύσματα).



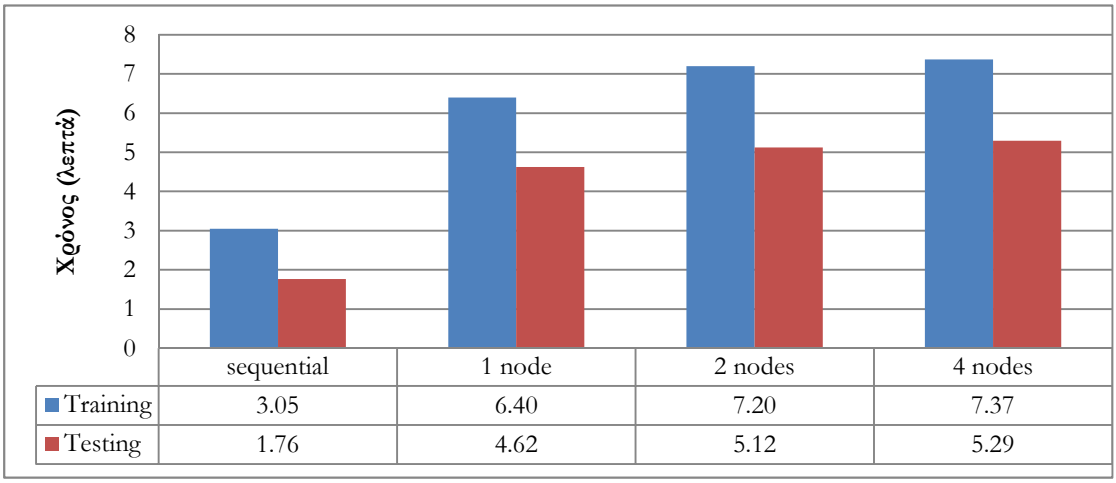
	sequential	1 node	2 nodes	4 nodes
Training	237.04	90.16	65.27	32.75
Testing	142.15	47.50	32.72	20.94

Εικόνα 8: ***Χρόνος εκπαίδευσης και αξιολόγησης για 8 εκατομμύρια training data και 8 εκατομμύρια test data σε συνάρτηση με τους χρησιμοποιούμενους κόμβους***

Σύμφωνα με τα αποτελέσματα που παρουσιάζονται στην εικόνα 8, όταν αυξάνονται οι κόμβοι που χρησιμοποιούνται μειώνεται τόσο ο χρόνος εκπαίδευσης, όσο και ο χρόνος κατάταξης νέων εγγράφων σε κατηγορίες. Μ' άλλα λόγια, η χρήση κατανεμημένων λύσεων σε αρκετούς κόμβους κάνει δυνατή τη χρήση μεγάλου όγκου δεδομένων και ικανοποιεί τους

χρονικούς περιορισμούς που τίθενται (υποενότητα 6.1). Επίσης, παρατηρούμε ότι η εκπαίδευση του ταξινομητή αποτελεί μια πιο χρονοβόρα εργασία από τη χρήση του ταξινομητή στην κατάταξη νέων εγγράφων σε κατηγορίες.

Επιπλέον, εκτελέσαμε και μετρήσεις όπου χρησιμοποιήθηκαν 100 χιλιάδες εκπαιδευτικά έγγραφα για την εκπαίδευση του ταξινομητή και 100 χιλιάδες νέα έγγραφα για την αξιολόγησή του.

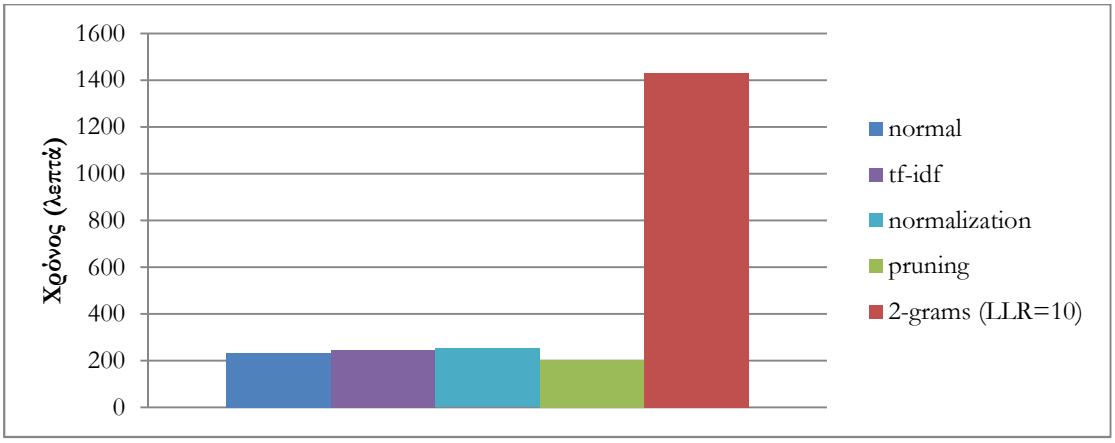


	sequential	1 node	2 nodes	4 nodes
Training	3.05	6.40	7.20	7.37
Testing	1.76	4.62	5.12	5.29

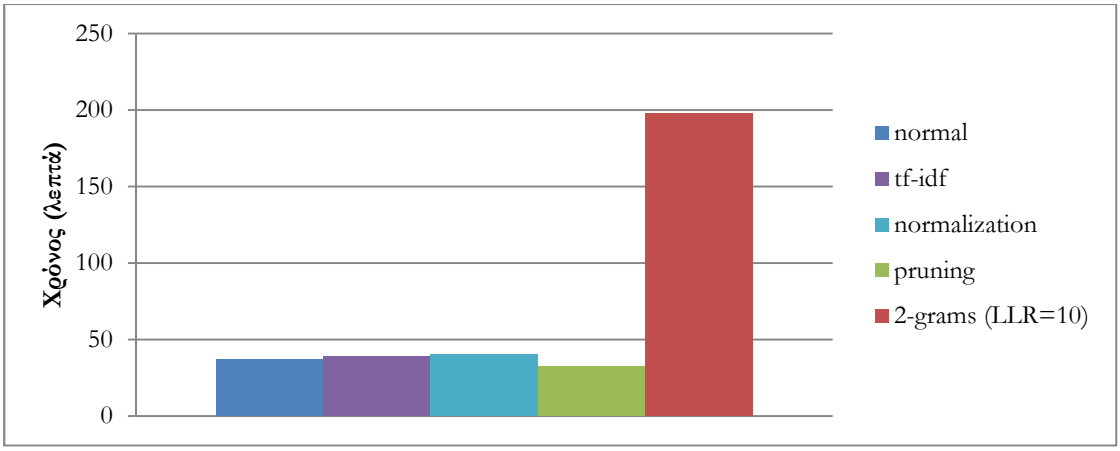
Εικόνα 9: ***Χρόνος εκπαίδευσης και αξιολόγησης για 100 χιλιάδες training data και 100 χιλιάδες test data σε συνάρτηση με τους χρησιμοποιούμενους κόμβους***

Σύμφωνα με τα αποτελέσματα που παρουσιάζονται στην εικόνα 9, επιβεβαιώνεται πως η χρήση κατανεμημένων αλγορίθμων ταξινόμησης δε συνίσταται όταν τα έγγραφα δεν είναι πολλά (κάτω από περίπου 500 χιλιάδες).

Στη συνέχεια, γνωρίζοντας πως κατά τη διάρκεια εκπαίδευσης του ταξινομητή χρησιμοποιούνται συνήθως αρκετές τεχνικές βελτίωσης της ακρίβειας, έγινε χρήση των τεχνικών βελτίωσης της ακρίβειας που προσφέρονται από το Mahout, ώστε να παρατηρηθεί πως αυτές επηρεάζουν τις χρονικές επιδόσεις. Έγινε εφαρμογή των τεχνικών tf-idf, κανονικοποίησης (Manhattan), 2-grams (LLR=10) και pruning (μέγιστες επιτρεπτές εμφανίσεις σε διαφορετικά έγγραφα (85%) και ελάχιστες εμφανίσεις σε διαφορετικά έγγραφα (50)). Για το σκοπό αυτό χρησιμοποιήθηκαν 8 εκατομμύρια έγγραφα και έγινε χρήση της τεχνικής cross-validation (80%-20%). Έγιναν μετρήσεις μόνο όταν ο αλγόριθμος εκτελέστηκε σειριακά και σε 4 κόμβους. Τα αποτελέσματα παρουσιάζονται στην εικόνα 10 και στην εικόνα 11, αντίστοιχα.



Εικόνα 10: ***Πως επηρεάζουν οι τεχνικές βελτίωσης της ακρίβειας το χρόνο εκτέλεσης του Naive Bayes όταν εκτελείται σειριακά***



Εικόνα 11: ***Πως επηρεάζουν οι τεχνικές βελτίωσης της ακρίβειας το χρόνο εκτέλεσης του Naive Bayes όταν εκτελείται σε 4 κόμβους***

Σύμφωνα με τα αποτελέσματα που παρουσιάζονται στην εικόνα 10 και στην εικόνα 11, παρατηρούμε πως οι τεχνικές tf-idf, κανονικοποίηση και pruning επηρεάζουν το χρόνο εκτέλεσης σε πολύ μικρό βαθμό. Αν και κάποιος θα ανέμενε πως το pruning θα μείωνε αρκετά το χρόνο, στην πραγματικότητα αυτό δε συμβαίνει. Αυτό οφείλεται στο γεγονός, πως ο χρόνος μετατροπής των εγγράφων σε διανύσματα που αποτελεί το μεγαλύτερο ποσοστό του ολικού χρόνου δε μειώνεται καθόλου, καθώς η εργασία αυτού του κομματιού δε μειώνεται. Αντίθετα, με τη χρήση 2-grams παρατηρούμε πως ο χρόνος εκτέλεσης αυξάνεται κατά πολύ. Αυτό οφείλεται στο γεγονός ότι το λεξιλόγιο σ' αυτή την περίπτωση δεν περιέχει μόνο λέξεις, αλλά και ζευγάρια λέξεων. Φυσικά δεν περιέχονται όλα τα ζευγάρια, καθώς επιλέχθηκε LLR=10 (διαγραφή των ασήμαντων ζευγαριών), αλλά περιέχονται αρκετά. Κρίνουμε σκόπιμο να αναφέρουμε εδώ, πως όταν γίνεται χρήση 2-grams ο χρόνος εκτέλεσης του Naive Bayes όταν εκτελείται σειριακά αυξάνεται κατά 7 φορές, ενώ όταν εκτελείται σε 4 κόμβους κατά 5 φορές. Η αύξηση της απαιτούμενης δουλείας κατανέμεται σε αρκετούς πυρήνες, όταν υπάρχουν περισσότεροι από ένα διαθέσιμοι και αυτό έχει ως αποτέλεσμα να μην αυξάνεται ο χρόνος εκτέλεσης σ' αυτή την περίπτωση με τον ίδιο ρυθμό που αυξάνεται ο χρόνος της σειριακής εκτέλεσης (1 πυρήνας).

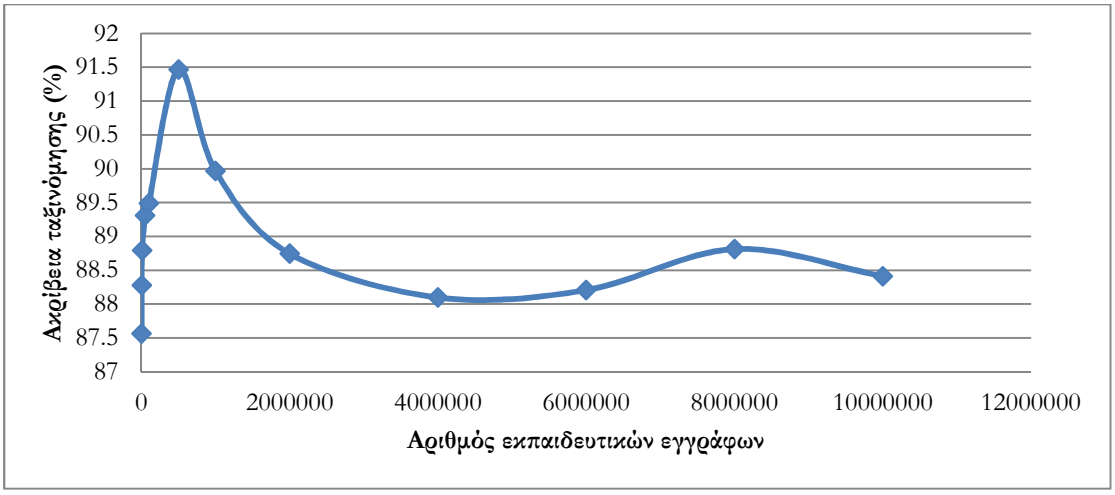
Όπως παρατηρήθηκε από την ενότητα 5, η τεχνική n-gram συμβάλει στη μεγαλύτερη αύξηση της ακρίβειας και συνεπώς η χρήση της είναι σχεδόν σίγουρη σε μια εφαρμογή. Όταν δε χρησιμοποιούνται όμως κατανεμημένες λύσεις, ακόμη και αν το μέγεθος των δεδομένων δεν είναι πάρα πολύ μεγάλο (8 εκατομμύρια στην πιο πάνω περίπτωση) ο χρόνος γίνεται απαγορευτικά μεγάλος ακόμη και με τη χρήση μόνο 2-grams. Παρατηρούμε ότι γίνεται συγκρίσιμος με την περίπτωση όπου έγινε μόνο ανάλυση των δεδομένων εισόδου με τη χρήση του αναλυτή StandardAnalyzer που προσφέρει το Mahout και τα έγγραφα εισόδου είναι 32-64 εκατομμύρια (εικόνα 6).

6.3 Εφαρμογή (Συνέχεια)

Σ' αυτήν την υποενότητα, θα μελετήσουμε πως ο αριθμός των εκπαιδευτικών εγγράφων επηρεάζει την ακρίβεια ταξινόμησης της εφαρμογής που μελετήσαμε στην ενότητα 5. Όπως αναφέρθηκε στην υποενότητα 4.1.1 και στην υποενότητα 6.1, το μέγεθος των δεδομένων εκπαίδευσης παίζει σημαντικό ρόλο στην ακρίβεια ταξινόμησης σε μια εφαρμογή. Αν και τις περισσότερες φορές, η χρήση περισσότερων εκπαιδευτικών εγγράφων οδηγεί σε αύξηση της ακρίβειας ταξινόμησης, αυτό δεν είναι εντελώς σίγουρο και εξαρτάται καθαρά από την εκάστοτε εφαρμογή. Η χρήση κατανεμημένων αλγορίθμων ταξινόμησης, δίνει τη δυνατότητα εξέτασης της συμπεριφοράς της εφαρμογής μας, ακόμη και στην περίπτωση που το μέγεθος των δεδομένων εκπαίδευσης είναι πολύ μεγάλο.

Για το σκοπό αυτό συλλέχτηκαν περίπου 44 εκατομμύρια κριτικές (με υλοποίηση crawler με τη χρήση του scrapy framework), από τις οποίες δυστυχώς μόνο 10 εκατομμύρια μπορούν να χρησιμοποιηθούν για εκπαιδευτικά έγγραφα. Ο λόγος που δεν μπορούν να χρησιμοποιηθούν περισσότερες κριτικές είναι γιατί στα 44 εκατομμύρια κριτικές περιλαμβάνονται περίπου 5 εκατομμύρια κριτικές της βαθμολογίας 3, και επίσης υπάρχει μεγάλη ανισοκατανομή μεταξύ Θετικών (βαθμολογίες 4 και 5) και Αρνητικών κριτικών (βαθμολογίες 1 και 2). Όπως αναφέρθηκε η βαθμολογία 3 δε χρησιμοποιείται στην εφαρμογή μας (υποενότητα 5.3.1) και επίσης πρέπει να υπάρχει ίσος αριθμός Αρνητικών και Θετικών κριτικών κατά τη διάρκεια εκπαίδευσης του ταξινομητή (υποενότητα 4.1.3).

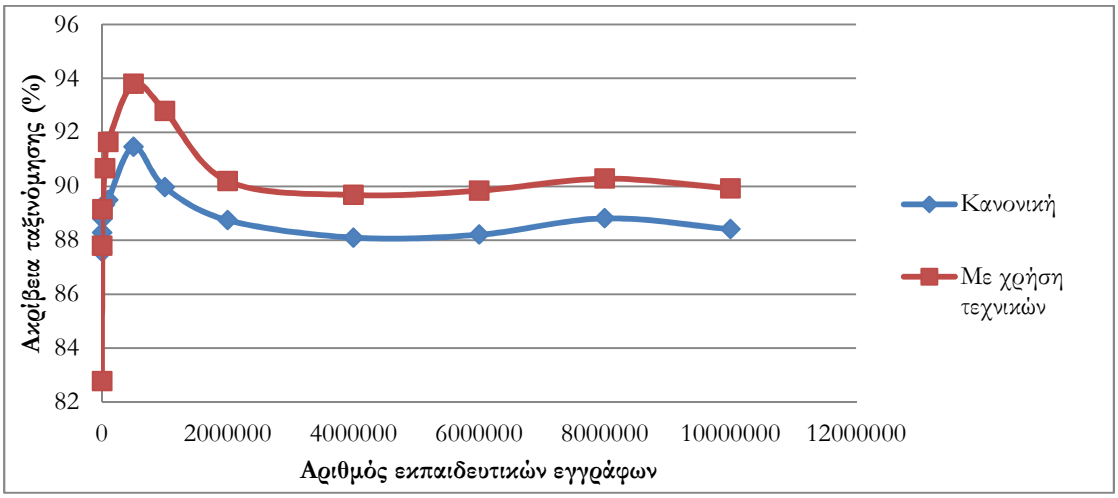
Στην εικόνα 12, παρουσιάζεται η ακρίβεια του ταξινομητή σε συνάρτηση με τα έγγραφα που χρησιμοποιήθηκαν για την εκπαίδευσή του. Για την αξιολόγηση του χρησιμοποιήθηκε ένα ξεχωριστό test set το οποίο αποτελείται από 1 εκατομμύρια νέα έγγραφα, 500 χιλιάδες ανά κατηγορία (επιλέχτηκαν τυχαία ύστερα από ανακάτεμα όλου του dataset). Η μόνη προεπεξεργασία που υπέστηκαν τα δεδομένα είναι η ανάλυση με τη χρήση του StandardAnalyzer αναλυτή που προσφέρει το Mahout. Καμιά άλλη τεχνική δεν εφαρμόστηκε. Τα εκπαιδευτικά έγγραφα περιείχαν τόσο τα σχόλια των κριτικών, όσο και τους τίτλους τους, δηλαδή περιείχαν τις πληροφορίες που δίνουν τη μεγαλύτερη ακρίβεια ταξινόμησης (υποενότητα 5.3.2).



Εικόνα 12: ***Ακρίβεια ταξινομητή σε συνάρτηση με τον αριθμό των εκπαιδευτικών εγγράφων***

Όπως φαίνεται στην εικόνα 12 η ακρίβεια του ταξινομητή μειώνεται, όταν τα εκπαιδευτικά έγγραφα ξεπεράσουν τις 500 χιλιάδες. Μ' άλλα λόγια, η χρήση μεγάλου όγκου δεδομένων στη συγκεκριμένη εφαρμογή όχι μόνο δεν αυξάνει την ακρίβεια, αντιθέτως τη μειώνει. Αυτό οφείλεται στο γεγονός ότι το αναγκαίο λεξιλόγιο που χρειάζεται η συγκεκριμένη εφαρμογή για να μπορεί να κατατάξει με ακρίβεια νέα έγγραφα στη σωστή κατηγορία μπορεί να παρθεί με μόνο 500 χιλιάδες διαφορετικά εκπαιδευτικά έγγραφα. Η περεταίρω χρήση εκπαιδευτικών εγγράφων έχει ως αποτέλεσμα ο ταξινομητής να υπερεκπαιδευτεί, δηλαδή να υπολογίσει με τέτοιο τρόπο τις παραμέτρους του που και η παραμικρή διαφορά ανάμεσα στα training και test data οδηγεί σε λάθος ταξινόμηση. Δηλαδή, ουσιαστικά χάνει τη δυνατότητα γενίκευσής του.

Παρόλο, που είναι πλέον φανερό πως η συγκεκριμένη εφαρμογή δεν απαιτεί μεγάλο όγκο δεδομένων κατά τη διάρκεια εκπαίδευσης του ταξινομητή, εφαρμόσαμε μερικές τεχνικές βελτίωσης της ακρίβειας (διαγραφή λέξεων με λιγότερες από 50 εμφανίσεις, διαγραφή stopwords, tf-idf, κανονικοποίηση Manhattan και 3-grams με llr=10), ώστε να δούμε πως διαφοροποιείται η ακρίβεια σε κάθε περίπτωση.



Εικόνα 13: ***Ακρίβεια ταξινομητή σε συνάρτηση με τον αριθμό των εκπαιδευτικών εγγράφων όταν χρησιμοποιούνται ή όχι οι τεχνικές βελτίωσης της ακρίβειας***

Σύμφωνα με τα αποτελέσματα που παρουσιάζονται στην εικόνα 13, οι τεχνικές βελτίωσης της ακρίβειας αυξάνουν την ακρίβεια ταξινόμησης, όταν ο αριθμός των εκπαιδευτικών εγγράφων είναι μεγαλύτερος από 5000. Για τις δύο περιπτώσεις, 1000 και 5000, η ακρίβεια ταξινόμησης μειώνεται. Αυτό οφείλεται στο γεγονός ότι αφαιρέθηκαν οι λέξεις με λιγότερες από 50 εμφανίσεις στα εκπαιδευτικά έγγραφα και λόγω του μικρού αριθμού εκπαιδευτικών εγγράφων σ' αυτές τις περιπτώσεις, διαγράφηκε το μεγαλύτερο ποσοστό των λέξεων (ακόμη και οι σημαντικές) που περιείχαν τα έγγραφα αυτά. Επίσης παρατηρούμε, πως η καλύτερη ακρίβεια ταξινόμησης παρατηρείται και πάλι όταν τα εκπαιδευτικά έγγραφα είναι 500 χιλιάδες.

Τέλος, θα χρησιμοποιηθούν όλα τα δεδομένα που δε χρησιμοποιήθηκαν ήδη για την αξιολόγηση του ταξινομητή. Συγκεκριμένα θα γίνουν δύο μετρήσεις. Στη πρώτη θα χρησιμοποιηθούν περίπου 37 εκατομμύρια κριτικές για την αξιολόγηση του ταξινομητή, οι οποίες προέρχονται από τις βαθμολογίες 1, 2, 4 και 5 και στη δεύτερη θα χρησιμοποιηθούν περίπου 22 εκατομμύρια κριτικές, οι οποίες προέρχονται από τις βαθμολογίες 1 και 5 (μιας και αυτές όπως αναφέρθηκε είναι πιο πιθανό να ανήκουν στη σωστή κατηγορία). Ο ταξινομητής εκπαιδεύτηκε με μόλις 500 χιλιάδες εκπαιδευτικά έγγραφα και εφαρμόστηκαν όλες οι τεχνικές που χρησιμοποιήθηκαν και πιο πάνω.

Πίνακας 28: ***Αξιολόγηση ταξινομητή σε μεγάλο όγκο δεδομένων***

Βαθμολογίες που χρησιμοποιήθηκαν στα test data	**Ακρίβεια ταξινόμησης**
1, 2, 4, 5	93.1213%
1, 5	95.4141%

Συγκρίνοντας τα αποτελέσματα του πίνακα 28 με αυτά του πίνακα 26 (υποενότητα 5.5) και της εικόνας 13, παρατηρούμε ότι η χρήση μεγάλου όγκου δεδομένων κατά την αξιολόγηση οδηγεί σε ελαφρώς χαμηλότερη ακρίβεια. Παρόλα αυτά, η εν λόγω αξιολόγηση παρέχει τη βεβαιότητα ότι η χρήση του ταξινομητή σε νέα δεδομένα θα έχει τα επιθυμητά αποτελέσματα, αφού η ακρίβεια ήταν πολύ υψηλή ακόμη και όταν αξιολογήθηκε σε μεγάλο όγκο νέων δεδομένων.

6.4 Σύνοψη ενότητας

Στην ενότητα αυτήν, εξηγήθηκε αρχικά γιατί η χρήση μεγάλου όγκου δεδομένων είναι επιθυμητή και σε αρκετές περιπτώσεις αναγκαία και πως η χρήση τέτοιου όγκου δεδομένων κάνει απαραίτητη τη χρήση κατανεμημένων αλγορίθμων ταξινόμησης. Επίσης, δόθηκαν παραδείγματα σύγχρονων εφαρμογών που απαιτούν τη χρήση μεγάλου όγκου δεδομένων και σύμφωνα και με τις υπόλοιπες απαιτήσεις που έχουν, κάνουν υποχρεωτική τη χρήση κατανεμημένων αλγορίθμων ταξινόμησης. Στη συνέχεια, θέλοντας να επιβεβαιώσουμε τη χρησιμότητα των κατανεμημένων λύσεων, εκτελέστηκαν πολλά πειράματα σε μεγάλο όγκο δεδομένων και έγινε χρήση διαφορετικού αριθμού κόμβων. Σύμφωνα λοιπόν με τα αποτελέσματα που λήφθηκαν και τις συγκρίσεις που έγιναν, έγινε κατανοητό πως η χρήση κατανεμημένων αλγορίθμων ταξινόμησης είναι αναγκαία τόσο κατά την εκπαίδευση όσο και κατά τη χρήση του ταξινομητή όταν τα δεδομένα που χρησιμοποιούνται είναι πάρα πολλά,

καθώς μόνο με τη χρήση τους ικανοποιούνται οι απαιτήσεις που έχουν πολλές σύγχρονες εφαρμογές. Ωστόσο, διαπιστώθηκε πως η χρήση κατανεμημένων αλγορίθμων ταξινόμησης όταν ο αριθμός των δεδομένων δεν είναι πολύ μεγάλος (μικρότερος από περίπου 500 χιλιάδες έγγραφα (≈200MB δεδομένα)) έχει ως αποτέλεσμα χειρότερους χρόνους εκτέλεσης. Αυτό, όπως αναφέρθηκε οφείλεται στο γεγονός ότι το Hadoop προσθέτει ένα αρχικό overhead, γεγονός που καθιστά την κατανεμημένη εκτέλεση πιο αργή από τη σειριακή σ' αυτή την περίπτωση. Τέλος, μελετήθηκε πως ο αριθμός των εγγράφων εκπαίδευσης επηρεάζει την ακρίβεια ταξινόμησης της εφαρμογής που μελετήσαμε στην ενότητα 5. Όπως παρατηρήθηκε, η χρήση μεγάλου όγκου δεδομένων κατά την εκπαίδευση, δεν επέφερε καμιά αύξηση στην ακρίβεια ταξινόμησης. Αντιθέτως, η ακρίβεια μειώθηκε. Η καλύτερη ακρίβεια παρατηρήθηκε όταν τα εκπαιδευτικά έγγραφα ήταν μόνο 500 χιλιάδες, που αυτό σημαίνει πως η χρήση κατανεμημένων αλγορίθμων ταξινόμησης σ' αυτήν την εφαρμογή δεν προτείνεται. Παρόλα αυτά, η χρήση κατανεμημένων ταξινομητών μπορεί να βοηθήσει στη συγκεκριμένη εφαρμογή στην κατάταξη μεγάλου όγκου δεδομένων σε κατηγορίες σε μικρό χρονικό διάστημα, αν αυτό φυσικά απαιτείται από τη χρήση της εφαρμογής. Επίσης, δόθηκε η δυνατότητα για πειραματισμό σε μεγάλο όγκο δεδομένων, αφού δεν ήταν εξ' αρχής γνωστό πως η χρήση μεγάλου όγκου δεδομένων δε θα επέφερε τα επιθυμητά αποτελέσματα.

7

Επίλογος

7.1 Σύντομη ανακεφαλαίωση

Σκοπός της συγκεκριμένης διπλωματικής ήταν η ανάλυση συναισθήματος σε μεγάλο όγκο δεδομένων με τη χρήση κατανεμημένων τεχνικών μηχανικής εκμάθησης. Για το σκοπό αυτό, στην ενότητα 2 μελετήθηκε το Hadoop, το οποίο αποτελεί το πλέον διαδεδομένο framework για κατανεμημένη επεξεργασία και το Mahout, το οποίο προσφέρει μια βιβλιοθήκη για αλγορίθμους μηχανικής εκμάθησης. Αυτό που ξεχωρίζει το Mahout από άλλες βιβλιοθήκες αλγορίθμων μηχανικής εκμάθησης είναι το γεγονός ότι οι περισσότεροι αλγόριθμοι που προσφέρει είναι κατανεμημένοι και κλιμακώσιμοι. Μ' άλλα λόγια δηλαδή, μπορούν να χρησιμοποιηθούν και στην περίπτωση όπου τα διαθέσιμα έγγραφα είναι πάρα πολλά. Για να γίνει αυτό εφικτό, οι συγκεκριμένοι αλγόριθμοι υλοποιήθηκαν σύμφωνα με το προγραμματιστικό μοντέλο Map Reduce και εκτελούνται πάνω από το Hadoop. Στη συνέχεια, στην ενότητα 3 μελετήθηκαν οι κατανεμημένοι αλγόριθμοι που προσφέρονται από το Mahout, ο Naive Bayes και ο Random Forests, και δόθηκε ιδιαίτερη έμφαση στο Naive Bayes καθώς αυτός επιλέχτηκε να χρησιμοποιηθεί στη συνέχεια. Παρόλο που ο Random Forests αποτελεί ένα από τους καλύτερους αλγόριθμους ταξινόμησης, η χρήση του στο Mahout δε συστήνεται όταν τα διαθέσιμα έγγραφα ξεπερνούν τα 10 εκατομμύρια. Αντίθετα, ο Naive Bayes μπορεί να χρησιμοποιηθεί ακόμη και όταν τα διαθέσιμα έγγραφα φτάνουν τα 100 εκατομμύρια. Κατόπιν, γνωρίζοντας πως η ακρίβεια εξαγωγής του συναισθήματος παίζει καθοριστικό παράγοντα, στην ενότητα 4 μελετήσαμε γνωστές μεθόδους βελτίωσης της ακρίβειας. Οι μέθοδοι αυτοί χωρίζονται σε δύο κατηγορίες, στην επιλογή των κατάλληλων εκπαιδευτικών εγγράφων και στην εφαρμογή τεχνικών βελτίωσης της ακρίβειας. Οι τεχνικές αυτές έχουν ως σκοπό να προεπεξεργαστούν κατάλληλα τα επιλεγμένα εκπαιδευτικά έγγραφα, ώστε να τα φέρουν σε μια καλύτερη μορφή και επίσης, να βελτιώσουν τη διαδικασία υπολογισμού του βάρους κάθε λέξης.

Με το πέρας του θεωρητικού μέρους, στην ενότητα 5 έγινε παρουσίαση της εφαρμογής εξαγωγής συναισθήματος που μελετάται σ' αυτήν την εργασία. Σκοπός της συγκεκριμένης εφαρμογής ήταν η εξαγωγή του συναισθήματος από κριτικές χρηστών και η κατάταξή τους σε «Θετικές» ή «Αρνητικές». Στη συνέχεια, έγινε εφαρμογή του Naive Bayes και των

διαφόρων μεθόδων βελτίωσης της ακρίβειας που παρουσιάστηκαν, με σκοπό την εξαγωγή συμπερασμάτων και την επίτευξη όσον το δυνατό μεγαλύτερης ακρίβειας. Η αρχική ακρίβεια ταξινόμησης του Naive Bayes ήταν περίπου 83%. Μετά την προσεχτική μελέτη της εφαρμογής και την εφαρμογή των μεθόδων βελτίωσης της ακρίβειας, η ακρίβεια έφτασε περίπου το 94%. Συγκεκριμένα, η επιλογή κατάλληλων εκπαιδευτικών εγγράφων οδήγησε σε ακρίβεια 90.1537% και η εφαρμογή των τεχνικών σε ακρίβεια 93.7749%.

Κατόπιν, στην ενότητα 6 αρχικά εξηγήθηκε, γιατί η χρήση μεγάλου όγκου δεδομένων είναι επιθυμητή και σε αρκετές περιπτώσεις αναγκαία και πως η χρήση τέτοιου όγκου δεδομένων κάνει απαραίτητη τη χρήση κατανεμημένων αλγορίθμων ταξινόμησης. Στη συνέχεια, επιβεβαιώθηκε η χρησιμότητα των κατανεμημένων αλγορίθμων ταξινόμησης, όταν γίνεται χρήση μεγάλου όγκου δεδομένων. Συγκεκριμένα, φάνηκε ότι όταν ο αριθμός των εγγράφων ξεπερνά τις 500χιλιάδες (≈200MB δεδομένα), η χρήση κατανεμημένων λύσεων μειώνει το χρόνο εκτέλεσης σε μεγάλο βαθμό και ικανοποιεί τις χρονικές απαιτήσεις που θέτουν οι σύγχρονες εφαρμογές. Τέλος, μελετήθηκε η επίδραση του αριθμού των εκπαιδευτικών εγγράφων στην εφαρμογή που μελετήσαμε, με σκοπό την επίτευξη ακόμη μεγαλύτερης ακρίβειας ταξινόμησης. Όπως φάνηκε, η χρήση μεγάλου όγκου δεδομένων στη συγκεκριμένη εφαρμογή δεν ενδείκνυται, μιας και υπερεκπαιδεύεται ο ταξινομητής και συνεπώς μειώνεται η ακρίβεια ταξινόμησης.

7.2 Γενικά συμπεράσματα

Ολοκληρώνοντας την παρούσα εργασία, θα θέλαμε να επικεντρωθούμε σε μερικά σημεία. Όπως έγινε αντιληπτό, η χρήση του Mahout βοήθησε πάρα πολύ στην επίλυση του προβλήματός μας. Συγκεκριμένα, χρησιμοποιώντας την πλούσια βιβλιοθήκη αλγορίθμων μηχανικής εκμάθησης που προσφέρει το Mahout, εξοικονομήσαμε πολύ χρόνο και κόπο, αφού πολλά πράγματα παρέχονταν έτοιμα (κατανεμημένος αλγόριθμος Naive Bayes και κάποιες τεχνικές βελτίωσης της ακρίβειας) και δε χρειάστηκε να υλοποιηθούν. Ωστόσο, κρίνουμε σκόπιμο να αναφέρουμε πως το Mahout δεν αποτελεί την ιδανική λύση. Λόγω της πρόσφατης δημιουργίας του, αρκετά από τα παρεχόμενα εργαλεία δε συμπεριφέρονται όπως θα έπρεπε και αυτό έχει ως αποτέλεσμα να επιβαρύνεται ο χρήστης με την υλοποίηση/διόρθωση τους.

Ένα άλλο σημείο που θα θέλαμε να τονίσουμε είναι η σπουδαιότητα των μεθόδων βελτίωσης της ακρίβειας. Όπως φάνηκε, η χρήση του ταξινομητή αυτή καθαυτή δεν έδωσε την επιθυμητή ακρίβεια ταξινόμησης. Μόνο μετά τη χρήση των μεθόδων βελτίωσης της ακρίβειας επιτεύχθηκε η επιθυμητή ακρίβεια. Μ' άλλα λόγια, αν δε χρησιμοποιηθούν διαδεδομένες μέθοδοι βελτίωσης της ακρίβειας, μια εφαρμογή πολύ πιθανό να αποτύχει.

Επίσης, κρίνεται σκόπιμο να τονιστεί η αναγκαιότητα χρήσης κατανεμημένων αλγορίθμων ταξινόμησης όταν τα διαθέσιμα έγγραφα είναι πάρα πολλά (>500χιλιάδες (≈200MB δεδομένα)). Όπως παρατηρήθηκε, μειώνουν τόσο το χρόνο εκπαίδευσης, όσο και το χρόνο κατάταξης νέων εγγράφων σε κατηγορίες, με αποτέλεσμα να καταφέρνουν να ικανοποιήσουν τις χρονικές απαιτήσεις των σύγχρονων εφαρμογών. Επιπλέον, θα θέλαμε να επισημάνουμε πως η χρήση των κατανεμημένων αλγορίθμων ταξινόμησης δεν προτείνεται για όλες τις

εφαρμογές. Εφαρμογές που απαιτούν τη χρήση λιγότερων από περίπου 500χιλιάδες έγγραφα, προτείνεται να χρησιμοποιούν σειριακούς αλγόριθμους ταξινόμησης. Κι αυτό γιατί οι σειριακοί αλγόριθμοι ταξινόμησης απαιτούν μικρότερο χρόνο εκτέλεσης, όταν τα διαθέσιμα έγγραφα δε ξεπερνούν τις 500χιλιάδες.

Τέλος, θα θέλαμε να τονίσουμε πως η χρήση μεγάλου όγκου εκπαιδευτικών εγγράφων δε συνεπάγεται την αύξηση της ακρίβειας. Αυτό φάνηκε και από την εφαρμογή που μελετήσαμε σ' αυτήν την εργασία.

7.3 Επεκτάσεις

Η περαιτέρω διερεύνηση του αντικειμένου της εργασίας θα μπορούσε να περιλαμβάνει συγκρίσεις μεταξύ των 2 κατανεμημένων αλγορίθμων που προσφέρει το Mahout. Συγκεκριμένα, οι συγκρίσεις αυτές θα μπορούσαν να λάβουν υπόψη τόσο την ακρίβεια ταξινόμησης, όσο και το χρόνο εκτέλεσης. Γενικά, οι συγκρίσεις αυτές μπορούν να προσφέρουν πολύ χρήσιμα συμπεράσματα. Για παράδειγμα, αν και ο Random Forests έχει μεγαλύτερη χρονική πολυπλοκότητα από το Naive Bayes, αυτό δε σημαίνει πως θα χρειάζεται μεγαλύτερο χρόνο από το Naive Bayes στην περίπτωση μιας εφαρμογής. Ο λόγος που μπορεί να συμβαίνει αυτό είναι γιατί ο Random Forests μπορεί να απαιτεί πολύ λιγότερα εκπαιδευτικά έγγραφα από τον Naive Bayes για να πετύχει παρόμοια ακρίβεια ή ακόμη και μεγαλύτερη.

8

Βιβλιογραφία

[1] "Twitter Statistics," [Online]. Available: https://about.twitter.com/company. [Accessed June 2014].

[2] "Facebook Statistics," [Online]. Available: http://www.statisticbrain.com/facebook-statistics/. [Accessed June 2014].

[3] S. Owen, R. Anil, T. Dunning and E. Friedman, Mahout In Action, Manning Publications Co, 2012.

[4] B. Liu, E. Blasch, Y. Chen, D. Shen and G. Chen, "Scalable Sentiment Classification for Big Data Analysis Using Naıve Bayes Classifier," IEEE International Conference on Big Data, pp. 99-104, 2013.

[5] "Apache Hadoop," [Online]. Available: http://hadoop.apache.org/. [Accessed June 2014].

[6] Π. Ανδρέας, "Hadoop," University of Cyprus, 2012.

[7] "Map Reduce," [Online]. Available: http://hadoop.apache.org/docs/r1.2.1/mapred_tutorial.html. [Accessed June 2014].

[8] "Apache Mahout," [Online]. Available: http://mahout.apache.org/. [Accessed June 2014].

[9] "Random Forest," [Online]. Available: http://mahout.apache.org/users/classification/partial-implementation.html. [Accessed June 2014].

[10] C. D. Manning, P. Raghavan and H. Schütze, "Text classification and Naive Bayes," in An Introduction to Information Retrieval, Cambridge University Press, 2009, pp. 253-288.

[11] H. Zhang, "The Optimality of Naive Bayes," Proceedings of the Seventeenth Florida Artificial Intelligence Research Society Conference, p. 562–567, 2004.

[12] P. Giacomelli, Apache Mahout Cookbook, Packt Publishing, 2013.

[13] J. R. Owens, B. Femiano and J. Lentz, Hadoop Real-World Solutions Cookbook, Packt Publishing, 2013.

[14] J. D. M. Rennie, L. Shih, J. Teevan and D. R. Karger, "Tackling the Poor Assumptions of Naive Bayes Text Classifiers," Proceedings of the Twentieth International Conference on Machine Learning (ICML-2003), p. 616–623, 2003.

[15] "Naive Bayes," [Online]. Available: http://mahout.apache.org/users/classification/bayesian.html. [Accessed June 2014].

[16] M. Poesio, "Text categorization with Decision Trees and Naïve Bayes," Universita' di Trento, 2013.

[17] T. M. Mitchell, "Decision Tree Learning," in Machine Learning, McGraw-Hill Science/Engineering/Math, 1997, pp. 52-80.

[18] P. Latinne, O. Debeir and C. Decaestecker, "Mixing bagging and multiple feature subsets to improve classification accuracy of decision tree combination," Proceedings of the Tenth Belgian-Dutch Conference on Machine Learning, pp. 15-22, 2000.

[19] V. Ganganwar, "An overview of classification algorithms for imbalanced datasets," International Journal of Emerging Technology and Advanced Engineering, 2012.

[20] E. Frank and R. R. Bouckaert, "Naive Bayes for text classification with unbalanced classes," Proceedings 10th European Conference on Principles and Practice of Knowledge Discovery in Databases, pp. 503-510, 2006.

[21] G. Forman, Feature Selection for Text Classification, Taylor & Francis Group, LLC, 2007.

[22] A. Hogenboom, D. Bal and F. Frasincar, "Exploiting Emoticons in Sentiment Analysis," Twenty-Eighth Symposium on Applied Computing, p. 703–710, 2013.

[23] "The Porter Stemming Algorithm," [Online]. Available: http://tartarus.org/martin/PorterStemmer/. [Accessed June 2014].

[24] G. Forman, "An Extensive Empirical Study of Feature Selection Metrics for Text Classification," Journal of Machine Learning Research 3, pp. 1289-1305, 2003.

[25] "TripAdvisor," [Online]. Available: http://www.tripadvisor.com/. [Accessed June 2014].

[26] H. Wang, D. Can, A. Kazemzadeh, F. Bar and S. Narayanan, "A System for Real-time Twitter Sentiment Analysis of 2012 U.S. Presidential Election Cycle," Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics, p. 115–120, 2012.

[27] A. Agarwal, B. Xie, I. Vovsha, O. Rambow and R. Passonneau, "Sentiment Analysis of Twitter Data," Proceedings of the Workshop on Language in Social Media (LSM 2011), p. 30–38, 2011.