ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ

UNIVERSITY OF PIRAEUS

Ανάπτυξη αλγορίθμου κατηγοριοποίησης με αυξητικό τρόπο για ανάλυση συναισθήματος στο twitter

Τμήμα Ψηφιακών Συστημάτων

Δικτυοκεντρικά Πληροφοριακά Συστήματα

Κυριαζής Αλέξανδρος ΜΕ14025

Υπεύθυνη καθηγήτρια Μαρία Χαλκίδη

Abstract

The presented thesis has as a subject the development and experimental apply of an incrementally classification algorithm for solving the problem of the gradually diminished prediction accuracy of emotional imprint in tweet stream of messages.

The environment of the problem and the approach for solving it will be presented analytically with the usage of distributed technologies as well as programmatic norms that appertain to these technologies. Finally, experimental validation of the algorithm will form conclusions on which improvements will be proposed.

Πρόλογος

Η παρούσα εργασία έχει ως αντικείμενο την κατασκευή και πειραματική εφαρμογή incrementally classification αλγορίθμου για την επίλυση του προβλήματος της σταδιακής ελάττωσης της ακρίβειας πρόβλεψης του συναισθήματος σε ροή μηνυμάτων από το twitter.
Πιο συγκεκριμένα, θα παρουσιαστεί αναλυτικά το περιβάλλον του προβλήματος και η προσέγγιση για την αντιμετώπιση του με χρήση κατανεμημένων τεχνολογιών καθώς και προγραμματιστικών νορμών που εμπίπτουν σε αυτές τις τεχνολογίες. Τέλος, θα γίνει πειραματική επιβεβαίωση του αλγορίθμου και εξαγωγή συμπερασμάτων καθώς και βελτίωσης της προσέγγισης αυτής.

Περιεχόμενα

1. Ανάλυση συναισθήματος....6
1.1 Προσεγγίσεις ανάλυσης συναισθήματος: Βασικές Κατηγορίες....7
1.1.1 Document-level....7
1.1.2 Sentence-level analysis....8
1.1.3 Aspect-Based-Sentiment Analysis....9
1.1.4 Comparative Sentiment Analysis....9
1.1.5 Sentiment Lexicon Acquisition....10
1.1.6 Μέθοδος Μηχανικής Μάθησης....11
1.2 Ανάλυση συναισθήματος στο twitter....12
2. Εισαγωγή στη Μηχανική μάθηση....14
2.1 Ορισμός....14
2.2 Κατηγορίες μηχανικής μάθησης....14
2.2.1 Supervised learning....14
2.2.2 Reinforcement learning....15
2.2.3 Semi-supervised....15
2.2.4 Unsupervised....15
2.3 Βασικά βήματα μηχανικής μάθησης [16]....17
2.3.1 Συλλογή δεδομένων....17
2.3.2 Προεπεξεργασία δεδομένων....17
2.3.3 Ανάλυση δεδομένων εισόδου....17
2.3.4 Εκπαίδευση αλγορίθμου....18
2.3.5 Εκτίμηση απόδοσης αλγορίθμου....18
2.3.6 Χρήση....18
2.4 Βασικοί αλγόριθμοι....18
2.4.1 Support Vector Machines (SVM)....18
2.4.2 Decision trees....19
3. Τεχνολογίες επεξεργασίας μεγάλου όγκου δεδομένων....20
3.1 Apache Spark....20
3.1.1 Spark Core, Resilient Distributed Datasets (RDD's)....21
3.1.2 Machine Learning Library (MLlib)....22
3.2 Git....23
3.3 Java....23
4. Naive bayes & αυξητικό μοντέλο κατηγοριοποίησης....25

4.1 Incremental Naive Bayesian Formula....26
4.2 KNN....28
4.3 Εντροπία....29
5. Σύστημα ανάλυσης συναισθήματος σε περιβάλλον microblogging....30
5.1 Εισαγωγή....30
5.2 Συστατικά του συστήματος....33
5.3 Παρουσίαση της προσέγγισης....34
5.4 Εκπαίδευση Naive Bayes Classifier....35
5.5 Κατασκευή μοντέλου KNN....35
5.5 Υπολογισμός Εντροπίας Τεστ Σετ....36
5.6 Αποτελέσματα....38
6. Επίλογος- Συμπεράσματα....41
Βιβλιογραφία....42
Παράρτημα....44
Στιγμιότυπα κώδικα....44
Οδηγίες εγκατάστασης....46
Λεπτομέρειες υλοποίησης....47

1. Ανάλυση συναισθήματος

Η ανθρώπινη ανάγκη για απόκτηση και συλλογή γνώσης έχει οδηγήσει την αύξηση εγγράφων κειμένων που δημοσιεύονται καθημερινά στα μέσα κοινωνικής δικτύωσης αλλά και σε όλες τις ιστοσελίδες. Σημαντικότερης αξίας όμως για τον άνθρωπο έχει η εξαγωγή πληροφορίας από αυτά για κοινωνικά ζητήματα καθώς και για τον ίδιο, εφόσον πρόκειται για δημόσιο πρόσωπο.

Η αλματώδης αύξηση των εγγράφων καθιστά δύσκολη τη συλλογή χρήσιμης πληροφορίας, καθώς υπάρχει πάντα και η άχρηστη πληροφορία ή θόρυβος. Παρόλα αυτά, ο όγκος της πληροφορίας περιέχει επαρκή δεδομένα και μπορεί να χρησιμοποιηθεί, για να διαχωρίσουμε τη χρήσιμη για εμάς πληροφορία από τον θόρυβο.

Η ανάλυση συναισθήματος αφορά τη γενικότερη κατηγοριοποίηση κειμένου, με σκοπό την εξαγωγή συμπερασμάτων συναισθηματικής φόρτισης σχετικά με το περιεχόμενό του. Αφορά τη γενικότερη μεθοδολογία εξαγωγής πολικότητας και υποκειμενικότητας από ένα κείμενο.

Οι τεχνικές συναισθηματικής ανάλυσης ανήκουν σε τρεις κυρίως κλάδους: της τεχνητής νοημοσύνης και μηχανικής μάθησης, της υπολογιστικής γλωσσολογίας (computational liguistics) και της εξόρυξης κειμένου (text mining). Μέσω του ευρύτερου κλάδου της τεχνητής νοημοσύνης στον οποίο συγκαταλέγονται οι τεχνικές της εξόρυξης γνώσης από δεδομένα, η εξόρυξη κειμένου και η μηχανική μάθηση, επιτυγχάνουμε την κατανόηση κειμένων όπως και τη συλλογή της επιθυμητής πληροφορίας.

Ένας άλλος τρόπος ονομασίας της ανάλυσης συναισθήματος που έχει καθιερωθεί είναι και η Εξόρυξη Γνώμης (Opinion Mining), η οποία κάνει χρήση NLP, ανάλυσης κειμένου και υπολογιστικής γλωσσολογίας (computational linguistics) για τον εντοπισμό και την εξαγωγή υποκειμενικής πληροφορίας. Είναι υπολογιστικός κλάδος που μελετά τον σωστό εντοπισμό των απόψεων, αισθημάτων, και αξιολόγησης αυτών που εκφράζονται σε κειμενική μορφή, όπως παράδειγμα στις ειδήσεις, ιστοσελίδες, ιστολόγια και στις συζητήσεις που λαμβάνουν χώρα στα κοινωνικά δίκτυα.

Μερικές από τις κατηγορίες των μεθόδων που αφορά η εξόρυξη κειμένου είναι:

- Γλωσσικός προσδιορισμός (Language Identification)
- Εξαγωγή χαρακτηριστικών γνωρισμάτων (Feature Extraction)
- Περιληπτική παρουσίαση της Πληροφορίας (Summarization)
- Κατηγοριοποίηση, κατάταξη με επίβλεψη (Categorization, Supervised Classification)
- Ομαδοποίηση, μη επιβλεπόμενη κατάταξη (Clustering, Unsupervised Classification)

Η εξόρυξη κειμένου δεν πρέπει να παρερμηνεύεται ως απλή αναζήτηση στον παγκόσμιο ιστό, καθώς με την απλή αναζήτηση ο χρήστης ψάχνει να βρει μια πληροφορία η οποία υπάρχει σίγουρα στον ιστό, σε αντίθεση με την εξόρυξη γνώσης η οποία στόχο έχει να ανακαλύψει άγνωστη πληροφορία χρησιμοποιώντας τα ήδη υπάρχοντα δεδομένα.

Καθώς η κειμενική πληροφορία χωρίζεται κατά κύριο λόγο σε δυο κατηγορίες, τα γεγονότα και τις απόψεις, η ανάγκη για αφουγκρασμό της κοινής γνώμης έχει αυξήσει την ανάγκη για εξαγωγή της σημασιολογίας και των συναισθημάτων που προκύπτουν από τη φυσική γλώσσα. Η ανάλυση συναισθήματος έχει ιδιαίτερη βαρύτητα σε οργανισμούς οι οποίοι επιθυμούν να μάθουν το γενικό συναίσθημα, την κοινή γνώμη σχετικά με μια οντότητα, όπως ένα προϊόν, μια ταινία, ένα άτομο κλπ.

Η προσέγγιση για την κατηγοριοποίηση του κειμένου εμπλέκει την κατασκευή classifiers από κατηγοριοποιημένα δείγματα κειμένου ή προτάσεων, που ανήκουν στην κατηγορία της εκπαιδευόμενης μάθησης (supervised learning). Ένας ακόμα όρος που σχετίζεται με την ανάλυση συναισθήματος είναι η πολικότητα της γνώμης (opinion orientation, sentiment orientation, polarity of opinion, semantic orientation), ο οποίος αναφέρεται στον συναισθηματικό προσανατολισμό ενός κειμένου, μιας πρότασης ή μιας λέξης. Για παράδειγμα η πολικότητα ενός κειμένου μπορεί να είναι «θετική», «αρνητική» ή «ουδέτερη», που σημαίνει ότι επικρατεί ενός είδους συναίσθημα [1].

1.1 Προσεγγίσεις ανάλυσης συναισθήματος: Βασικές Κατηγορίες

1.1.1 Document-level

Είναι η πιο απλή μορφή sentiment analysis στην οποία το έγγραφο θεωρείται ότι περιέχει την άποψη του συγγραφέα πάνω σε ένα κύριο αντικείμενο. Αρκετές έρευνες έχουν ασχοληθεί με το αντικείμενο. Υπάρχουν δυο κύριες προσεγγίσεις επί του θέματος: Προσέγγιση με Επιβλεπόμενη Μάθηση (Supervised learning) και προσέγγιση με Μη Επιβλεπόμενη Μάθηση (Unsupervised learning)

Κατά την επιβλεπόμενη προσέγγιση υποτίθεται ότι υπάρχουν δεδομένα για εκπαίδευση και ένα πεπερασμένο σύνολο κατηγοριών στις οποίες ανήκει το έγγραφο. Η πιο απλή περίπτωση είναι δυο κατηγορίες θετικό και αρνητικό, ενώ η πολυπλοκότητα μπορεί να αυξηθεί προσθέτοντας και ουδέτερη κλάση ή δίνοντας βαθμίδες στο πόσο θετική ή αρνητική μπορεί να είναι μια άποψη. Δεδομένων των training data, το σύστημα χτίζει ένα μοντέλο κατάταξης χρησιμοποιώντας κάποιον

από τους αλγορίθμους κατάταξης όπως SVM, Naive Bayes, Logistic Regression ή KNN. Στη συνέχεια, σε αυτό το μοντέλο βασίζεται η πρόβλεψη κατηγορίας σε νέα έγγραφα. Στην περίπτωση που σαν κατηγορία πρέπει να αποδοθεί αριθμός ως κλάση του εγγράφου, τότε είναι δυνατόν να χρησιμοποιηθεί regression. Πειραματικά έχει αποδειχτεί ότι μέσω της αναπαράστασης ενός εγγράφου ως Bag Of Words είναι δυνατόν να επιτευχθεί καλή ακρίβεια στα αποτελέσματα. Πιο πολύπλοκες και προηγμένες τεχνικές αναπαράστασης εγγράφου χρησιμοποιούν TFIDF, POS (Part-Of-Speech) πληροφορία, sentiment lexicons και ανάλυση της δομής και της μορφολογίας του εγγράφου (parse structures).

Κατά τη Μη Επιβλεπόμενη Μάθηση, η οποία βασίζεται σε SO (Semantic Orientation polarity) συγκεκριμένων φράσεων μέσα στο έγγραφο, το έγγραφο κατηγοριοποιείται ως θετικό όταν ο μέσος όρος του SO των φράσεων αυτών είναι πάνω από κάποιο όριο (threshold), αλλιώς ως αρνητικό. Οι κύριες τεχνικές που ακολουθούνται για την ανίχνευση των φράσεων που θα χρησιμοποιηθούν για τον προσδιορισμό του SO polarity είναι δύο: χρήση συγκεκριμένων POS μοτίβων και χρήση λεξικών αποτελούμενων από λέξεις ομαδοποιημένες ως προς το συναίσθημά τους (sentiment lexicons). Κλασσική μέθοδος υπολογισμού SO κάποιας λέξης ή φράσης είναι ο υπολογισμός της διαφοράς του PMI (Pointwise Mutual Information) μεταξύ δύο λέξεων που εκδηλώνουν συναίσθημα. Το PMI(P,W) μετράει τη στατιστική εξάρτηση μεταξύ μιας φράσης P και της λέξης W, βασιζόμενο στη συμμετοχή τους μέσα σε μια συλλογή λέξεων /φράσεων /εγγράφων ή στο διαδίκτυο με τη χρήση Web search queries [2].

Για την ανάλυση κειμένων που περιέχουν γλώσσες όπως Κινέζικα και Ισπανικά, για τις οποίες δεν υπάρχουν αρκετοί γλωσσικοί πόροι, η συνήθης πρακτική αφορά τη μηχανική μετάφραση για τη μετατροπή του κειμένου πρώτα σε Αγγλικά, για τα οποία υπάρχουν πολλοί πόροι γι΄ αυτή τη διαδικασία και μετά την εφαρμογή τού όποιου αλγόριθμου για την ανάλυση συναισθήματος.

1.1.2 Sentence-level analysis

Η ανάλυση συναισθήματος σε επίπεδο πρότασης είναι κάτι περίπλοκο, διότι η σημασιολογική ερμηνεία των λέξεων εξαρτάται πάρα πολύ από το πλαίσιο στο οποίο αναφέρονται. Η ανάλυση συναισθήματος σε επίπεδο πρότασης δίνει μια πιο αναλυτική οπτική των διαφορετικών απόψεων που μπορεί να εκφράζονται σε ένα κείμενο σχετικά με τις οντότητες που αναφέρονται. Στην περίπτωση αυτή γίνεται η υπόθεση ότι η οντότητα για την οποία εκφράζεται κάποια άποψη σε μια πρόταση είναι γνωστή. Επίσης, γίνεται η υπόθεση ότι η πρόταση εκφράζει ένα κύριο

συναίσθημα μόνο και συνήθως μόνο οι υποκειμενικές προτάσεις αναλύονται, καθώς θεωρείται ότι οι αντικειμενικές προτάσεις δεν έχουν κάποιο συναίσθημα. Κάποιες προσεγγίσεις κάνουν χρήση των αντικειμενικών προτάσεων, η ανάλυση των οποίων όμως αποτελεί αρκετά δυσκολότερο εγχείρημα [2], [3].

1.1.3 Aspect-Based-Sentiment Analysis

Η ανάλυση συναισθήματος σε σχέση με ένα αντικείμενο ή στόχο ονομάζεται Aspect Based Sentiment Analysis (ABSA). Τα συστήματα που έχουν ως στόχο την ABSA λαμβάνουν ως είσοδο ένα σύνολο κειμένων, όπως για παράδειγμα, κριτικές προϊόντων ή μηνύματα από κοινωνικά δίκτυα, τα οποία καταπιάνονται με μια συγκεκριμένη οντότητα, π.χ. ένα προϊόν, ένα κινητό τηλέφωνο κ.α. Στη συνέχεια, προσπαθούν να εντοπίσουν την περιοχή ενδιαφέροντος, για παράδειγμα το πιο συχνό θέμα με τα χαρακτηριστικά του, δηλαδή την οθόνη ενός φορητού υπολογιστή και να υπολογίσουν το συνολικό συναίσθημα προς αυτό. Παρόλο που πολλά συστήματα ABSA έχουν προταθεί, και πρόκειται ως επί το πλείστον για ερευνητικά πρωτότυπα (Liu, 2012), δεν υπάρχει καθιερωμένη διαδικασία για ABSA, ούτε υπάρχουν θεσπισμένα μέτρα αξιολόγησης για τις δευτερεύουσες εργασίες που τα ABSA συστήματα καλούνται να εκτελέσουν [2].

1.1.4 Comparative Sentiment Analysis

Μια συγκριτική πρόταση συνήθως εκφράζει μια σειριακή σχέση μεταξύ δύο συνόλων οντοτήτων σε σχέση με κάποια χαρακτηριστικά ή θέματα. Οι συγκρίσεις σχετίζονται με τις άμεσες απόψεις, αλλά ταυτόχρονα είναι και αρκετά διαφορετικές. Παράδειγμα μιας τυπικής πρότασης που εκφράζει άμεση άποψη είναι «Η ποιότητα του Χ προϊόντος είναι άψογη!». Σε μια συγκριτική πρόταση θα είχαμε το ακόλουθο: «Η ποιότητα του Χ προϊόντος είναι καλύτερη από την ποιότητα του Ψ προϊόντος». Είναι προφανές ότι οι συγκρίσεις χρησιμοποιούν διαφορετικού τύπου εκφράσεις από την έκφραση άποψης. Συνήθως οι συγκρίσεις εκφράζουν μια συγκριτική άποψη για δύο ή περισσότερες οντότητες σχετικά με κοινά χαρακτηριστικά, π.χ. «κατασκευαστική ποιότητα», «τιμή» κλπ. Παραδείγματος χάριν, η συγκριτική πρόταση "Canon's optics are better than those of Sony and Nikon" εκφράζει την συγκριτική σχέση (better, {optics}, {Canon}, {Sony, Nikon}). Οι συγκριτικές προτάσεις έχουν διαφορετικού τύπου γλωσσικές ιδιότητες και γλωσσικές

“κατασκευές” από τις τυπικές προτάσεις που εκφράζουν απόψεις, για παράδειγμα "Cannon's optic is great" [2], [4], [5].

1.1.5 Sentiment Lexicon Acquisition

Η προσέγγιση που βασίζεται στα λεξικά, περιλαμβάνει τον υπολογισμό της συναισθηματικής πολικότητας ενός κειμένου από τον σημασιολογικό προσανατολισμό των λέξεων ή των φράσεων οι οποίες το απαρτίζουν [2], [6].

Τα λεξικά για τη λεξιλογική αυτή προσέγγιση μπορούν να κατασκευαστούν είτε με μη αυτόματο τρόπο (manually) είτε αυτόματα, χρησιμοποιώντας αρχικές λέξεις, ονομαζόμενες ως seed words, ώστε να επεκταθεί η λίστα από λέξεις. Πολλές από τις μεθόδους που βασίζονται σε λεξικά έχουν εστιάσει στη χρήση επιθέτων (adjectives) ως ενδείξεων για τον σημασιολογικό προσανατολισμό ενός κειμένου. Στην αρχή μια λίστα επιθέτων και οι αντίστοιχες SO τιμές συλλέγονται σε ένα λεξικό (dictionary) και στη συνέχεια όλα τα επίθετα ενός κειμένου μαρκάρονται με τα σκορ που υπάρχουν στο λεξικό. Ύστερα, τα σκορ αυτά μετατρέπονται σε μέσο όρο, ο οποίος τελικά θα καθορίσει την πολικότητα (polarity) του κειμένου [7].

Τα Sentiment Lexicons είναι λίστες από λέξεις και εκφράσεις συναισθημάτων ή γνώμης. Εκτός από λέξεις, μπορεί να απαρτίζονται και από φράσεις ή και ιδιωματισμούς. Πολλά από τα λεξικά αυτού του τύπου υπάρχουν στο διαδίκτυο. Συνήθως περιέχουν χιλιάδες όρους και είναι αρκετά χρήσιμα ως συναισθηματικές λέξεις και φράσεις (sentiment words, polar words, opinion bearing words κλπ.). Παραδείγματος χάριν, οι λέξεις όμορφος, υπέροχος, καλός, φανταστικός κατηγοριοποιούνται ως θετικές λέξεις, ενώ οι λέξεις κακός, ελλιπής, απαίσιος ως αρνητικές.

Η μέθοδος χρήσης λεξικών στηρίζεται στη χρήση ειδικών λεξικών τα οποία αποτελούνται από λέξεις οι οποίες έχουν βάρη ανάλογα με το νόημα που έχουν και την ένταση επί της χρήσης της κάθε μιας. Η χρήση των λεξικών κατηγοριοποιείται ως προς τη χρήση η οποία γίνεται :

- Ολιστικά, δηλαδή χρησιμοποιούνται στο σύνολο του κειμένου και υπολογίζεται με την συμμετοχή αλγορίθμου η στάση ως προς τη θεματολογία (document level analysis).
- Τμηματικά, τα οποία κατηγοριοποιούνται με βάση ποιο αποδίδει καλύτερα, αν θέλουμε να εστιάσουμε σε συγκεκριμένη θεματολογία(sentence level analysis).

Η διαφορά στον τρόπο χειρισμού των λεξικών έχει να κάνει με τον τρόπο που μπορούμε να διαχειριστούμε τα δεδομένα. Μπορούμε να ξεχωρίσουμε τα λεξικά και ως

προς τον τρόπο που το καθένα χειρίζεται τα βάρη - τιμές που θα αποδοθούν στις λέξεις του και υπάρχουν οι δύο κατηγορίες:

1. Η πρώτη και πιο απλή κατηγορία είναι αυτή της κατηγοριοποίησης των λέξεων σε αρνητικές και θετικές. Η λέξη “super” κατηγοριοποιείται ως θετική ενώ το “awfull” ως αρνητική. Έτσι διαμορφώνεται μια πολικότητα, και αποδίδεται σε όλο το κείμενο θετική ή αρνητική υποκειμενική άποψη.
2. Στη δεύτερη μέθοδο γίνεται απόδοση τόσο αρνητικό όσο θετικό βάρος, ανάλογα με τη σημασιολογία τους στις προτάσεις. Για παράδειγμα η λέξη “mad” είναι 0,4 θετική και 0,6 αρνητική ενώ η λέξη “angry” είναι 1,0 αρνητική και 0 θετική.

1.1.6 Μέθοδος Μηχανικής Μάθησης

Η μηχανική μάθηση (machine learning) ανήκει στο πεδίο της τεχνητής νοημοσύνης η οποία περιλαμβάνει διάφορους αλγορίθμους όπως και μεθόδους που δίνουν την δυνατότητα στους υπολογιστές να «μαθαίνουν». Η μηχανική μάθηση αποσκοπεί στην κατασκευή προσαρμόσιμων προγραμμάτων που λειτουργούν με βάση την αυτοματοποιημένη ανάλυση συνόλων δεδομένων και όχι τη διαίσθηση των μηχανικών που τα δημιούργησαν. Η μηχανική μάθηση έχει μεγάλες ομοιότητες με την στατιστική, αφού και τα δύο πεδία μελετούν την ανάλυση δεδομένων. Η χρήση των μεθόδων της μηχανικής μάθησης στην ανάλυση συναισθήματος και τη σημασιολογική ανάλυση κειμένων αφορά στον εντοπισμό και στην χρήση του κατάλληλου αλγόριθμου για την εξαγωγή αποτελεσμάτων.

Ωστόσο όμως η εξαγωγή επιθυμητών αποτελεσμάτων απαιτεί πειραματισμούς από τους ερευνητές του πεδίου με διαφορετικού τύπου αλγόριθμους, τους οποίους πρέπει να εκπαιδεύσουν σε πολλά και διαφορετικά σύνολα δεδομένων έτσι ώστε να κατηγοριοποιήσουν τις άγνωστες περιπτώσεις. Ορισμένοι από τους αλγόριθμους θα αναλυθούν περαιτέρω στο κεφάλαιο της μηχανικής μάθησης.

1.2 Ανάλυση συναισθήματος στο twitter

Στη συγκεκριμένη κατηγορία που αφορά η παρούσα εργασία το αντικείμενο ανάλυσης είναι η δομική μονάδα του tweet. Ένα tweet αφορά μορφοποιημένο κείμενο 140 χαρακτήρων με ιδιαιτερότητες όπως, η χρήση emoticons και η δημιουργία hashtags.

Η ανάλυση συναισθήματος σε ένα τέτοιο περιβάλλον έχει αποδειχθεί ότι μπορεί να βελτιστοποιηθεί από αυτούς τους παράγοντες. Πιο συγκεκριμένα, η ύπαρξη emoticons ως features αφορά έναν πολύ αποτελεσματικό τρόπο έκφρασης θετικού ή αρνητικού συναισθήματος [8], [9]. Έρευνες σε αλγορίθμους μηχανικής μάθησης δείχνουν ότι μπορεί να έχουν ακρίβεια πάνω από 80%, όταν εκπαιδεύονται με πλούσια από emoticon δεδομένα [10]. Ενδείξεις χρήσης hashtags και intensifiers όπως η παρουσία λέξεων με κεφαλαία γράμματα και τα σημεία στίξης δείχνουν συσχέτιση μεταξύ της ορθής αναγνώρισης της πολικότητας του κειμένου και των προαναφερόμενων χαρακτηριστικών [11]. Ωστόσο, υπάρχουν έρευνες που αναφέρουν ότι τέτοιου είδους χαρακτηριστικά (features) μπορούν να προσθέσουν αξία στην αναγνώριση συναισθήματος αλλά μόνο οριακά, δηλαδή τα χαρακτηριστικά αυτά παίζουν μικρό ρόλο στη διαδικασία σωστής κατηγοριοποίησης. Χαρακτηριστικά που εμπίπτουν στην κατηγορία natural language, π.χ. Part-Of-Speech tags, και χρήση λεξικών συναισθήματος συμβάλλουν σημαντικά στην ανίχνευση της διάθεσης . Οι Agarwal, Xie, Vovsha, Rambow και Passonneau δείχνουν ότι τα πιο χαρακτηριστικά στοιχεία συνδυάζουν prior polarity των λέξεων μαζί με part-of-speech tags.

Η χρήση τεχνικών σε μεταφορικό λόγο (Sentiment Analysis on Figurative Language) αποτελεί έναν αρκετά απαιτητικό τομέα της ανάλυσης συναισθήματος, λόγω της πολυπλοκότητας που εμπεριέχει η σωστή αναγνώριση του μεταφορικού λόγου και του συναισθήματος που εκφράζεται μέσα από αυτόν. Η αναγνώριση της ειρωνείας και του σαρκασμού αποτελούν επίσης πολύπλοκα μοτίβα, για να αναγνωριστούν λόγω των μεταξύ τους συσχετίσεων που συχνά παρουσιάζονται. Ο σαρκασμός συνήθως στοχεύει στη μείωση του σχολίου και είναι πιο εύκολος στην αναγνώρισή του σε σχέση με τις υπόλοιπες κατηγορίες. Η ειρωνεία λειτουργεί ως άρνηση, αλλοιώνοντας το συναίσθημα, σχεδόν πάντα όμως προς το αρνητικό. Στην περίπτωση, βέβαια, έκφρασης της ειρωνείας μέσω ενός θετικού πλαισίου, η διάκριση του νοήματος είναι πραγματικά δύσκολο εγχείρημα [12], [13].

Έχει παρατηρηθεί συσχέτιση μεταξύ εκφράσεων που περιέχουν μοτίβα όπως “As * As”, “about as * as” και ειρωνικών παρομοιώσεων. Οι [14] εξέτασαν τον σαρκασμό που αποτυπώνεται μέσω των hashtags αν είναι αξιόπιστες πηγές σαρκασμού και κατέληξαν στο συμπέρασμα ότι tweets που ως σαρκαστικά κατηγοριοποιούνται από τους χρήστες ενδέχεται να έχουν “θόρυβο” και ότι αποτελούν την πιο δύσκολη μορφή κατηγοριοποίησης. Επίσης, έχει παρατηρηθεί ότι ο σαρκασμός συχνά συμπεριλαμβάνει αντίθεση μεταξύ ενός θετικού συναισθήματος και μιας

αρνητικής κατάστασης. Έχουν, ακόμη, χρησιμοποιηθεί χαρακτηριστικά που εκφράζουν αυτή την ανισορροπία από τα συμφραζόμενα, τα οποία προκύπτουν από τον τομέα της φυσικής γλώσσας, όπως επίσης και διάφορα μορφολογικά χαρακτηριστικά ενός tweet [12]. Η ανισορροπία αυτή των συμφραζομένων υπολογίζεται ως η σημασιολογική ομοιότητα μεταξύ των λέξεων, αλλά και διαφόρων λεξιλογικών πόρων (Wordnet, Whisel's dictionary) για την πολικότητα λέξεων, τερπνότητα (pleasantness) και επιρρήματα που υπονοούν άρνηση ή εκφράζουν χρονικό παράθυρο και συγχρονισμό.

Επίσης ορισμένα χαρακτηριστικά που αξιοδοτούνται θετικά στον χαρακτηρισμό συναισθήματος είναι τα σημεία στίξης, τα emoticons, οι κεφαλαίες λέξεις, τα n-grams και τα skip-grams [15].

2. Εισαγωγή στη Μηχανική μάθηση

2.1 Ορισμός

Η μηχανική μάθηση αποτελεί την επιστήμη που ασχολείται με την κατασκευή προγραμμάτων τα οποία βελτιώνονται αυτόματα, αποκτώντας εμπειρία. Είναι μια συνδυαστική επιστήμη που βρίσκεται μεταξύ της πληροφορικής, μηχανικής και στατιστικής.

Η μηχανική μάθηση (Machine learning) είναι ο επιστημονικός κλάδος που μελετά την κατασκευή αλγορίθμων οι οποίοι αναδιαμορφώνουν τα αποτελέσματά τους βάση της εισόδου τους. Η βασική λειτουργία ενός γενικού αλγορίθμου περιλαμβάνει την δημιουργία ενός μοντέλου μέσω δεδομένων εισόδου και τη χρήση αυτού για πρόβλεψη ή λήψη αποφάσεων, σε αντίθεση με την εκτέλεση αυστηρών στατικών οδηγιών. Αποτελεί έναν κλάδο ο οποίος είναι στενά συνδεδεμένος με την υπολογιστική στατιστική η οποία ειδικεύεται επίσης στην πρόβλεψη αποφάσεων. Πολλές φορές το αντικείμενό της συγχέεται με την εξόρυξη δεδομένων, παρόλο που αυτή εστιάζει περισσότερο στη διερευνητική ανάλυση δεδομένων. Χρησιμοποιεί στατιστική και μαθηματική βελτιστοποίηση και χρησιμοποιείται σε ένα ευρύ φάσμα εφαρμογών όπου η σχεδίαση ενός προγράμματος με αυστηρούς κανόνες είναι ανέφικτη. Μερικά παραδείγματα χρήσης μηχανικής μάθησης είναι η κατηγοριοποίηση spam mail, η αναγνώριση χαρακτήρων (OCR), οι μηχανές αναζήτησης κλπ.

2.2 Κατηγορίες μηχανικής μάθησης

Μια από τις μεθόδους κατηγοριοποίησης των μεθόδων μηχανικής μάθησης είναι με τον τρόπο που γίνεται η διαδικασία της μάθησης τους, δηλαδή με επίβλεψη(supervised) ή χωρίς επίβλεψη(unsupervised).

2.2.1 Supervised learning

Η επιβλεπόμενη μάθηση αφορά στη μαθησιακή διαδικασία εκείνη όπου ο αλγόριθμος κατασκευάζει μια συνάρτηση που “αντιστοιχεί” σε δεδομένες εισόδους γνωστές, επιθυμητές εξόδους, με απώτερο σκοπό τη γενίκευση της συνάρτησης αυτής και για εισόδους με άγνωστη έξοδο. Σκοπεύει, δηλαδή, στη δημιουργία ενός κανόνα από τα δεδομένα εκπαίδευσης, ώστε να μπορεί να “προβλέψει” άγνωστα δεδομένα με σχετική ακρίβεια.

2.2.2 Reinforcement learning

Το πρόγραμμα αλληλεπιδρά με ένα δυναμικό περιβάλλον μέσα στο οποίο πρέπει να επιτευχθεί ένας στόχος, χωρίς αυτό να γνωρίζει αν πλησιάζει ή όχι στον στόχο του. Ένα παράδειγμα είναι η εκμάθηση ενός παιχνιδιού έχοντας έναν αντίπαλο. Αυτός είναι ένας τρόπος μηχανικής μάθησης όπου το πρόγραμμα μπορεί να μάθει μέσα από μεθόδους επιβράβευσης / τιμωρίας, χωρίς να διευκρινίζεται πως θα επιτευχθεί ο στόχος αυτός.

2.2.3 Semi-supervised

Σε αυτήν την κατηγορία εντάσσονται αλγόριθμοι που τα δεδομένα εκμάθησης είναι ελλιπή, δηλαδή κάποιες από τις εισόδους λείπουν. Μια ειδική περίπτωση της αρχής αυτής είναι η μεταγωγή, όπου το σύνολο των περιπτώσεων του προβλήματος είναι γνωστό στον χρόνο εκμάθησης, αλλά λείπει μέρος των κλάσεων – κατηγοριών. Η κατηγορία αυτή μάθησης βρίσκεται στο ενδιάμεσο μεταξύ supervised και unsupervised learning και συνδυάζει τεχνικές και από τις δυο κατηγορίες .

2.2.4 Unsupervised

Σε αυτή την κατηγορία εντάσσονται αλγόριθμοι οι οποίοι κατασκευάζουν μοντέλα βάση των δεδομένων, χωρίς να γνωρίζουν τις επιθυμητές εξόδους για το σύνολο εκπαίδευσης.

Μια άλλη κατηγοριοποίηση των τεχνικών μηχανικής μάθησης περιλαμβάνει την εξέταση υπό το πρίσμα του επιθυμητού αποτελέσματος. Οι κύριες κατηγορίες περιλαμβάνουν:

- Κατηγοριοποίηση - Classification
- Παλινδρόμηση - Regression
- Συσταδοποίηση - Clustering

<u>Κατηγοριοποίηση – Classification</u>

Στην κατηγοριοποίηση οι είσοδοι του συστήματος χωρίζονται σε δυο η περισσότερες κλάσεις και ο αλγόριθμος προσπαθεί να φτιάξει ένα μοντέλο το οποίο θα μπορεί να κατηγοριοποιεί σε αυτές τις δοθείσες κλάσεις διάφορες άγνωστες μέχρι πρότινος εισόδους σε δυο ή περισσότερες

κλάσεις. Παράδειγμα τέτοιας περίπτωσης είναι και το spam filtering όπου ως είσοδος θεωρούνται τα e-mails ή διάφορα μηνύματα, ενώ ως έξοδος η καταχώρηση ως spam η non spam.

Γενικά η κατηγοριοποίηση θα μπορούσε να κατηγοριοποιηθεί με βάση την πρόβλεψη για την ένταξη των δειγμάτων σε μια από τις παρακάτω κατηγορίες. Στην περίπτωση που οι κατηγορίες-κλάσεις είναι δυο τότε έχουμε Binary classification, ενώ, αν το δείγμα μπορεί να ανήκει σε μία από περισσότερες από δυο, τότε έχουμε Multiclass classification.

Παλινδρόμηση – Regression

Η παλινδρόμηση αφορά supervised πρόβλημα με τη διαφορά ότι οι είσοδοι δεν είναι διακριτές αλλά συνεχείς. Η παλινδρόμηση προϋποθέτει ότι τα σχετικά δεδομένα ταιριάζουν με μερικά γνωστά είδη συνάρτησης και μετά καθορίζει την καλύτερη συνάρτηση αυτού του είδους που μοντελοποιεί τα δεδομένα που έχουν δοθεί[4]. Αποτέλεσμα της παλινδρόμησης, όταν χρησιμοποιείται ως τεχνική εξόρυξης δεδομένων, αποτελεί ένα μοντέλο που χρησιμοποιείται αργότερα για να προβλέψει τις τιμές τής κατηγορίας για τα νέα δεδομένα. Τέτοια παραδείγματα εφαρμογής της παλινδρόμησης αποτελεί η πρόβλεψη της ζήτησης για ένα νέο προϊόν ή υπηρεσία συναρτήσει των δαπανών διαφήμισης ή ο υπολογισμός της ταχύτητας του ανέμου σε σχέση με τη θερμοκρασία, την υγρασία και την ατμοσφαιρική πίεση του περιβάλλοντος.

Συσταδοποιήση – Clustering

Συσταδοποίηση ονομάζεται η διαδικασία κατά την οποία ένα σύνολο από αντικείμενα πρόκειται να χωριστεί σε ομάδες. Σε αντίθεση με τη διαδικασία της κατηγοριοποίησης οι ομάδες δεν είναι γνωστές εκ των προτέρων. Σαν αποτέλεσμα συγκαταλέγεται στους αλγορίθμους χωρίς επίβλεψη. Η καταχώρηση αντικειμένων σε ίδια ομάδα μεταφράζεται ως ομοιότητα βάση κάποιου κριτηρίου, ενώ αντίθετα τα δείγματα αυτά τα οποία ανήκουν σε διαφορετικές ομάδες είναι ανόμοια. Η ομοιότητα ή μη μεταξύ των αντικειμένων στην ουσία εξαρτάται από το πρόβλημα προς επίλυση και τη μορφή τους.

Εκτίμηση πυκνότητας – Density Estimation

Η εκτίμηση πυκνότητας προσπαθεί να διαβλέψει την κατανομή των εισόδων σε κάποιο χώρο και χωρίζεται σε παραμετρική και μη παραμετρική. Η παραμετρική λαμβάνει ως είσοδο ένα

σύνολο από παραμέτρους, ενώ αντίστοιχα η μη παραμετρική λαμβάνει υπόψη το σύνολο των εισόδων.

Μείωση Διάστασης – Dimensionality reduction

Στη μέθοδο αυτή απλοποιούνται οι είσοδοι με τη μετατροπή τους σε χώρο λιγότερων διαστάσεων. Η μοντελοποίηση θεμάτων είναι ένα παρεμφερές πρόβλημα, όπου σε ένα πρόγραμμα δίνεται ως είσοδος μια λίστα από έγγραφα που περιέχουν ανθρώπινη (φυσική) γλώσσα και αποσκοπεί στο να βρει ποια έγγραφα καλύπτουν παρόμοια θέματα.

2.3 Βασικά βήματα μηχανικής μάθησης [16]

2.3.1 Συλλογή δεδομένων

Σε αυτό το βήμα γίνεται μια προσέγγιση της συλλογής των δεδομένων που είναι σχετικά με το πρόβλημα ή την ευρύτερη κατηγορία του προβλήματος που καλούμαστε να λύσουμε. Η ανάκτηση μπορεί να γίνει είτε από τον παγκόσμιο ιστό είτε από μια βάση δεδομένων, ή από διάφορες άλλες πηγές που μπορούν να παραγάγουν δεδομένα ενδιαφέροντος πχ αισθητήρες.

2.3.2 Προεπεξεργασία δεδομένων

Αφού έχουν συλλεχθεί τα δεδομένα θα πρέπει να τροποποιηθούν σε μια μορφή η οποία θα είναι ενιαία και αξιοποιήσιμη. Σπάνια τα δεδομένα έρχονται σε μορφή η οποία μπορεί να μην χρειάζεται μετατροπή. Η διαδικασία της μετατροπής μπορεί να εξαρτάται από διάφορούς αλγορίθμους. Η μορφή που θα μετατραπούν τα δεδομένα εξαρτάται και από τον αλγόριθμο στον οποίο θα διοχετευθούν. Για παράδειγμα κάποιοι αλγόριθμοι δεν μπορούν να χρησιμοποιήσουν αρνητικές τιμές ως είσοδο.

2.3.3 Ανάλυση δεδομένων εισόδου

Σε αυτό το σημείο γίνεται μια πιο αφαιρετική εξέταση των δεδομένων. Σκοπός είναι να διαχωριστούν τα μη σημαντικά δεδομένα συμπεριλαμβανομένου και του θορύβου. Γίνεται μια αναγνώριση μοτίβων ή εμφανών περίεργων σημείων τα οποία είναι τελείως διαφορετικά

συγκριτικά με το υπόλοιπο σύνολο. Σε αυτό το σημείο μπορεί να βοηθήσει διαγραμματική αναπαράσταση αυτών σε μια, δυο, τρεις διαστάσεις.

2.3.4 Εκπαίδευση αλγορίθμου

Αυτό το βήμα αφορά την εκπαίδευση του αλγορίθμου με καλά καθαρισμένα δεδομένα και γίνεται εξαγωγή της γνώσης και της πληροφορίας. Η γνώση αυτή, το μοντέλο αποθηκεύεται προς χρήση με συγκεκριμένο format. Στην περίπτωση του unsupervised learning το βήμα αυτό παραλείπεται.

2.3.5 Εκτίμηση απόδοσης αλγορίθμου

Στο σημείο αυτό χρησιμοποιείται το μοντέλο από το προηγούμενο βήμα, για να επιβεβαιωθεί το κατά πόσο είναι ακριβές σε ένα σύνολο δεδομένων προκαθορισμένου αποτελέσματος. Στην περίπτωση του supervised learning υπάρχουν γνωστές τιμές με τις οποίες μπορεί να αξιολογηθεί η αποτελεσματικότητα του αλγορίθμου ενώ στις unsupervised learning μεθόδους χρησιμοποιούνται άλλες μετρικές, για να αξιολογηθεί η επιτυχία. Γενικά η διαδικασία της εκτίμησης μπορεί να επαναληφθεί γυρνώντας στο βήμα της προεπεξεργασίας των δεδομένων ή ακόμα και της συλλογής τους.

2.3.6 Χρήση

Εδώ γίνεται χρήση του συστήματος για την επίλυση του προβλήματος για το οποίο δημιουργήθηκε. Έχοντας κάνει τις δοκιμές που χρειάζονται, ώστε να είναι κατανοητό πώς λειτουργεί ο αλγόριθμος, γίνεται πλέον η εφαρμογή του σε διαφορετικά δεδομένα.

2.4 Βασικοί αλγόριθμοι

2.4.1 Support Vector Machines (SVM)

Ως Support Vector Machine ορίζεται ο classifier που διαιρεί τον χώρο εισόδου σε δύο περιοχές, οι οποίες χωρίζονται από ένα γραμμικό όριο. Αποτελεί διακριτή προσέγγιση, όχι πιθανολογικό μοντέλο. Ο τρόπος λειτουργίας βασίζεται στο αν ο στόχος που είναι να προβλεφθεί με ακρίβεια σύμφωνα με μια συνάρτηση κόστους. Θα πρέπει να αποτελεί τον πρωταρχικό στόχο

αντί να γίνεται εκτίμηση της κατανομής πιθανοτήτων, το οποίο είναι αρκετά πιο δύσκολο πρόβλημα.

2.4.2 Decision trees

Τα Decision Trees ανήκουν στα πιο δημοφιλή μοντέλα για την επίλυση προβλημάτων κατηγοριοποίησης. Στην κατασκευή μοντέλου ενός Decision Tree, χωρίζεται ο χώρος εισόδου με έναν ιεραρχικό τρόπο αναδρομικά. Η μέθοδος αυτή είναι γνωστή και ως «διαίρει και βασίλευε» και έχει εφαρμοστεί σε διάφορους αλγόριθμους. Υπάρχουν δύο βασικοί αλγόριθμοι που συσχετίζονται με τα Decision Trees. Ο βασικός αλγόριθμος ονομάζεται ID3 [17] και τα βασικά βήματα αναφέρονται ακολούθως. Η επέκταση αυτού ονομάζεται C4.5 αλγόριθμος [18], που μπορεί να διαχειριστεί και αριθμητικά δεδομένα και η συνθήκη τερματισμού είναι πιο "χαλαρή", επιτρέποντας στον αλγόριθμο να διαχειριστεί δεδομένα με "θόρυβο" που μπορεί να περιέχουν συνδυασμό κατηγορικών και αριθμητικών χαρακτηριστικών.

Τα βήματα του ID3:

1. Επιλέγεται ένα χαρακτηριστικό εισόδου ως η «ρίζα» ενός συγκεκριμένου (υπο)δένδρου
2. Διαιρούνται τα δεδομένα αυτής της ρίζας σε υποσύνολα, σε σχέση με την τιμή του κάθε επιλεγμένου χαρακτηριστικού και προστίθεται ένας νέος κόμβος για κάθε υποσύνολο.
3. Εάν κάποιος κόμβος περιέχει παραδείγματα από διαφορετικές κατηγορίες, πηγαίνει στο βήμα 1

Τα χαρακτηριστικά πρέπει να είναι κατηγορικά (διακριτά), εάν δεν είναι πρέπει να γίνουν με προεπεξεργασία. Ο αλγόριθμος τερματίζει, όταν όλα τα παραδείγματα ενός κόμβου βρίσκονται σε μια κλάση, είτε όταν δεν υπάρχουν άλλα χαρακτηριστικά για περαιτέρω τμηματοποίηση, είτε δεν υπάρχουν άλλα δείγματα για κατηγοριοποίηση.

Η ουσία του αλγορίθμου βρίσκεται στον τρόπο με τον οποίον επιλέγεται η διαίρεση του χαρακτηριστικού καθώς ένα κριτήριο που αντιστοιχεί στην ικανότητα να διαχωρίζει σωστά τα instances διαφορετικών κλάσεων υπολογίζεται για κάθε διαθέσιμο χαρακτηριστικό (input attribute) στον κόμβο διαχωρισμού [19].

3. Τεχνολογίες επεξεργασίας μεγάλου όγκου δεδομένων

3.1 Apache Spark

Το Apache Spark είναι μια υπολογιστική πλατφόρμα ειδικά σχεδιασμένη για cluster υπολογιστών (cluster computing platform) υλοποιημένη στη γλώσσα προγραμματισμού Scala [20]. Είναι κατασκευασμένη με σκοπό τη δημιουργία κατανεμημένων εφαρμογών γενικού σκοπού που βασίζονται εν γένει στην επεξεργασία μεγάλου όγκου δεδομένων, με μεγάλο βαθμό αποδοτικότητας και ταχύτητας.

Η βασική ιδέα του Apache Spark είναι το προγραμματιστικό μοντέλο MapReduce, με την κύρια διαφορά ότι υποστηρίζει περισσότερους τύπους υπολογισμών, όπως διαδραστικά ερωτήματα (interactive queries) και επεξεργασία δεδομένων συνεχούς ροής (streaming data processing). Μία ακόμα διαφορά σε σχέση με τις εργασίες (jobs) που μπορούν να ανατεθούν στο Spark σε σχέση με τις υλοποιήσεις του MapReduce σε άλλα συστήματα είναι η δυνατότητα για αποθήκευση δεδομένων στη μνήμη του κάθε κόμβου στο cluster κατά τη διάρκεια της εκτέλεσης του job. Αυτή η τελευταία ιδιότητα παρουσιάζεται ως caching και είναι ίσως ένα από τα πιο σημαντικά χαρακτηριστικά που εισάγει το Spark στην ανάπτυξη κατανεμημένων συστημάτων σε βαθμό μάλιστα να υπερτερεί στην ταχύτητα έναντι του Hadoop MapReduce μέχρι και 100 φορές καλύτερη χρονική απόδοση [21] .

Η δομή του Spark περιγράφεται καλύτερα ως αυτή μιας ενιαίας στοίβας υποσυστημάτων που συνεργάζονται και υποστηρίζουν το καθένα από αυτά ξεχωριστές υπηρεσίες. Συγκεκριμένα, η «καρδιά» του συστήματος είναι το Spark core. Το υποσύστημα core είναι η υπολογιστική μηχανή του συστήματος που είναι υπεύθυνη για τη δρομολόγηση, τον διαμοιρασμό και την επίβλεψη των εφαρμογών οι οποίες αναλύονται σε επιμέρους υπολογιστικές μονάδες (tasks) και αναθέτονται σε κόμβους του cluster. Το core είναι επίσης αυτό που παρέχει τις διεπαφές για τα δομικά προγραμματιστικά στοιχεία του συστήματος όπως είναι για παράδειγμα τα Resilient Distributed Datasets (RDD) που θα αναλύσουμε αργότερα. Αποτελεί επίσης τη βάση πάνω στην οποία στηρίζονται τα υποσυστήματα Spark Sql, Spark Streaming, Mlib και GraphX . Επίσης το Spark Sql προσφέρει δυνατότητα για συνεργασία του Spark με δομημένα δεδομένα μέσω της γλώσσας Sql καθώς και της παραλλαγής αυτής που προσφέρει το Apache Hive. Το Spark Streaming είναι αυτό που επιτρέπει την αποθήκευση και επεξεργασία stream δεδομένων σε πραγματικό χρόνο. Το GraphX είναι μια βιβλιοθήκη ειδικά σχεδιασμένη, για να παρέχει υποστήριξη για jobs και αλγορίθμους που επεξεργάζονται γράφους.

Τέλος η Mllib είναι βιβλιοθήκη η οποία έχει υλοποιημένους τους αλγορίθμους μηχανικής μάθησης εστιασμένους στην κατανεμημένη εφαρμογή τους. Το Spark όπως προαναφέραμε είναι υλοποιημένο εξ ολοκλήρου στη γλώσσα προγραμματισμού Scala, γλώσσα πολλαπλών παραδειγμάτων με ισχυρά στοιχεία συναρτησιακού αλλά και αντικειμενοστραφούς προγραμματισμού. Η scala χρησιμοποιεί το JVM περιβάλλον για την εκτέλεση των προγραμμάτων της, ενώ είναι σχεδιασμένη έτσι, ώστε να μπορεί να εισάγει και να χρησιμοποιεί βιβλιοθήκες της Java. Το Spark υποστηρίζει ανάπτυξη εφαρμογών σε οποιαδήποτε από τις γλώσσες Java και Scala καθώς και τη γλώσσα Python. Τα παραπάνω καθιστούν το Spark ανεξάρτητο πλατφόρμας (platform independent), ενώ παράλληλα με την Python παρέχει μεγάλη ελευθερία επιλογής στους επίδοξους χρήστες του.

3.1.1 Spark Core, Resilient Distributed Datasets (RDD's)

Ένα resilient distributed dataset είναι μια αμετάβλητη κατανεμημένη συλλογή αντικειμένων. Στην ουσία είναι μια δομή που καλύπτει με ένα επίπεδο αφαιρετικότητας (abstraction) τα προς επεξεργασία δεδομένα που χρησιμοποιούνται σε jobs του spark. Με αυτό τον τρόπο προσφέρεται από το σύστημα ένα σύνολο ιδιοτήτων και προγραμματιστικών διεπαφών στα δεδομένα , σε σχέση με το αν αυτά τα μεταχειριζόμασταν στη φυσική τους υπόσταση.

Το σύνολο των αντικειμένων μπορεί να αποτελείται από οποιονδήποτε τύπο δεδομένου υποστηρίζουν οι γλώσσες Python, Scala και Java είτε ακόμα και κλάσεις οριζόμενες από τον χρήστη. Όλα τα jobs που καταθέτονται στο Spark αρχικοποιούνται ορίζοντας ένα RDD από μία πηγή δεδομένων (π.χ HDFS) ή από ένα ήδη υπάρχον RDD. Από τη στιγμή που ορίζεται ένα RDD αυτό πλέον ορίζει μια απεικόνιση των πραγματικών δεδομένων από τα οποία δημιουργήθηκε και διασπάται σε κομμάτια (partitions) που μπορούν να υπολογιστούν σε ξεχωριστούς κόμβους του cluster.

Τα RDD είναι όπως αναφέραμε αμετάβλητα (immutable) με την έννοια, ότι οι μόνες επιτρεπόμενες ενέργειες μετά τη δημιουργία τους είναι οι μετασχηματισμοί (transformations) και οι υπολογισμοί (actions). Τα transformations είναι ουσιαστικά ενέργειες που επεξεργάζονται τα δεδομένα που ορίζονται από το RDD ανά partition, δημιουργώντας νέα RDD. Διαφοροποιούνται όμως από τα actions, γιατί λειτουργούν με οκνηρή αποτίμηση (lazy evaluation). Η έννοια του lazy evaluation, που υπάρχει στον τομέα των γλωσσών προγραμματισμού, υποδηλώνει ότι οποιοδήποτε transformation ορίζουμε σε ένα RDD δε μετασχηματίζει την ίδια στιγμή το υποκείμενο dataset παρά μόνο, όταν ζητηθεί από κάποιο action. Αυτό είναι ένα σημαντικό νέο χαρακτηριστικό που εισάγεται από το Spark και ικανοποιεί πολύ σε περιπτώσεις όπου μια σειρά

μετασχηματισμών που ορίζονται από ένα job και οδηγούν σε ένα αποτέλεσμα μπορεί το ίδιο αποτέλεσμα να παραχθεί αποδοτικότερα χωρίς τη σειριακή τους εκτέλεση. Τα actions από την άλλη πλευρά είναι οι ενέργειες αυτές που εκτελούμε σε ένα RDD και υπολογίζουν ένα αποτέλεσμα. Για τον υπολογισμό ενός αποτελέσματος χρειάζονται να γίνουν υπολογισμοί πάνω στα raw data που αντιπροσωπεύει το RDD, γι΄ αυτό και τα actions ουσιαστικά «πυροδοτούν» μια σειρά μετασχηματισμών που ενδεχομένως έχουν οριστεί για ένα dataset και επιστρέφουν το ζητούμενο αποτέλεσμα είτε στον driver είτε το αποθηκεύουν κάπου (π.χ HDFS).

Ένα ακόμα σημαντικό πλεονέκτημα που παρουσιάζουν τα RDD είναι ο τρόπος που μοντελοποιούνται από το Spark, ώστε να παρέχουν μηχανισμούς ανάκτησής τους σε περίπτωση προβλήματος. Συγκεκριμένα, για κάθε RDD ανά πάσα στιγμή αποθηκεύεται η σειρά των μετασχηματισμών (lineage) που ακολουθήθηκαν, για να κατασκευαστεί καθώς και πληροφορίες για την πηγή των δεδομένων που χρησιμοποιήθηκε, για να κατασκευαστεί το συγκεκριμένο RDD. Αυτή η πληροφορία παρέχει τη δυνατότητα στο Spark να επαναϋπολογίσει οποιοδήποτε partition του RDD χωρίς να επηρεάσει ουσιαστικά την εκτέλεση του job. Τέλος είναι χρήσιμο να αναφέρουμε εδώ ότι το API των RDD δίνει τη δυνατότητα να κατασκευαστούν RDD's από πολλές διαφορετικές πηγές δεδομένων, κάτι που κάνει τον προγραμματισμό εφαρμογών στο Spark ιδιαίτερα απλή διαδικασία. Ανάμεσα στα άλλα παρέχεται και διεπαφή για κατασκευή RDD από το HDFS που ονομάζεται Hadoop RDD. Ένα HadoopRDD στην πιο απλή του μορφή αντιστοιχίζει καθένα block του αρχείου στο HDFS σε ένα partition του RDD, παρόλα αυτά το Spark παρέχει δυνατότητα στον προγραμματιστή να δημιουργήσει RDD από το HDFS με ρυθμιζόμενο αριθμό partitions . Το τελευταίο είναι κάτι που ανά περίσταση μπορεί να ωφελήσει την απόδοση ενός job.

3.1.2 Machine Learning Library (MLlib)

Η βιβλιοθήκη αυτή είναι ένα framework μηχανικής μάθησης κατανεμημένων υπολογιστών και βρίσκεται πάνω από το Spark Core. Λόγω της αρχιτεκτονικής επεξεργασίας στη μνήμη είναι έως 9 φορές γρηγορότερο από την υλοποίηση του Apache Mahout, η οποία γίνεται σε δίσκο. Η βιβλιοθήκη περιλαμβάνει:

- Αλγορίθμους classification και regression: Svm (Support Vector Machines), logistic regression, linear regresssion, desicion trees, naive Bayes classification
- Τεχνικές συνεργατικού διαχωρισμού (collaborative filtering)
- Μεθόδους ανάλυσης cluster οπως kmeans και Latent Dirichlet Allocation
- Τεχνικές μείωσης διαστάσεων όπως singular value decomposition και principal component analysis
- Συναρτήσεις Feature extraction και transformation

3.2 Git

Μια εφαρμογή κατά την ανάπτυξή της χωρίζεται από τον προγραμματιστή ή την ομάδα ανάπτυξής της σε στάδια – κατηγορίες. Το Git είναι ένα σύστημα ελέγχου διανεμόμενης έκδοσης και διαχείρισης κώδικα (SCM) με έμφαση στην ταχύτητα, στην ακεραιότητα των δεδομένων και στην υποστήριξη για κατανεμόμενες μη γραμμικές ροές εργασίας. Το Git σχεδιάστηκε και αναπτύχθηκε αρχικά από τον Λίνους Τόρβαλντς για την ανάπτυξη του πυρήνα του Λίνουξ το 2005 και έχει γίνει από τότε το πιο πλατιά διαδεδομένο σύστημα ελέγχου εκδόσεων για ανάπτυξη λογισμικού.

Όπως τα περισσότερα άλλα διανεμόμενα συστήματα ελέγχου εκδόσεων αναθεώρησης και αντίθετα με τα περισσότερα συστήματα πελάτη-διακομιστή κάθε κατάλογος εργασίας του Git είναι ένα ολοκληρωμένο αποθετήριο με πλήρες ιστορικό και δυνατότητες πλήρους παρακολούθησης της έκδοσης, ανεξάρτητα από την πρόσβαση δικτύου ή ενός κεντρικού διακομιστή. Ο σκοπός του συστήματος διευκολύνει τους προγραμματιστές ώστε να μπορούν να ανακαλέσουν τις συγκεκριμένες εκδόσεις αργότερα.

Από τη γέννησή του το 2005, το Git έχει εξελιχθεί, για να είναι εύκολο στη χρήση αλλά και να διατηρεί τις αρχικές του ιδιότητες. Είναι γρήγορο, πολύ αποτελεσματικό με τα μεγάλα έργα, και έχει ένα σύστημα διακλάδωσης για μη-γραμμική ανάπτυξη και χρησιμοποιείται από μεγάλες εταιρείες και μεγάλα έργα όπως οι Google, Microsoft, gnome, linux, twitter, facebook, LinkedIn, PostegreSQL, android, eclipse.

3.3 Java

Η JAVA [27] είναι μια γλώσσα προγραμματισμού από τη εταιρία Sun Microsystemes. Η Java χαρακτηρίζεται από τα εξής: απλή, αντικειμενοστραφής, συμβατή με δικτυακά πρωτόκολλα, ουδέτερη της υποκείμενης αρχιτεκτονικής, φορητή, ασφαλής, υψηλής απόδοσης, δυναμική, σταθερή, interpreted και multithreaded. Η Java έχει σχεδιαστεί για να υποστηρίζει δικτυακές εφαρμογές. Ένα δίκτυο, όμως, αποτελείται από ποικιλία διαφορετικών συστημάτων, με διαφορετικές CPU και λειτουργικά συστήματα. Για την εκτέλεση Java εφαρμογών στο δίκτυο, το πρόγραμμα Java πρέπει να περάσει από δύο διαδικασίες, ώστε να καταλήξει σε εκτελέσιμη μορφή. Πρώτα ο μεταγλωττιστής, μετατρέπει τον πηγαίο κώδικα του προγράμματος σε μία ενδιάμεση γλώσσα που καλείται Java bytecodes. Τα Java bytecodes είναι ανεξάρτητα της πλατφόρμας και με χρήση του ερμηνευτή (interpreter) κάθε bytecode εντολή μετατρέπεται σε κατάλληλη δυαδική

μορφή, για να τρέξει στον εκάστοτε υπολογιστή. Η μεταγλώττιση (compilation) συμβαίνει μόνο μια φορά για κάθε Java πρόγραμμα και η ερμηνεία (interpretation) γίνεται κάθε φορά που το πρόγραμμα εκτελείται.

Τα Java bytecodes είναι σαν τη γλώσσα μηχανής για τη Java Virtual Machine (JVM). Κάθε Java ερμηνευτής (π.χ. ένας Web server που μπορεί να τρέχει Servlets) είναι μια εφαρμογή του της Java Virtual Machine. Η JVM αναλαμβάνει να μετατρέψει τα bytecodes σε κατάλληλη εκτελέσιμη μορφή, ανάλογα με το υποκείμενο software και hardware.

4. Naive bayes & αυξητικό μοντέλο κατηγοριοποίησης

Ο naive bayes algorithm ανήκει στην κατηγορία τω classifiers που εφαρμόζουν το Naive Bayes θεώρημα. Πιο συγκεκριμένα γίνεται υπολογισμός της εκ των υστέρων πιθανότητας (posterior probability) δοθείσης της εκ των προτέρων πιθανότητας (prior probability). Η εκ των προτέρων πιθανότητα είναι η κατανομή της κάθε κλάσης, δηλαδή αν το tweet είναι θετικό ή αρνητικό στο σύνολο των tweets που έχουμε στη διάθεσή μας. Η εκ των υστέρων πιθανότητα είναι η πιθανότητα κάθε tweet να ανήκει σε μια συγκεκριμένη κλάση (θετική ή αρνητική).

Υποθέτοντας ότι X = {A1, A2,..., An} είναι ένα sample δεδομένων, ένα tweet στη δική μας περίπτωση, με n features, τότε η εκ των υστέρων πιθανότητα να ανήκει το sample X στην κλάση C είναι P(C | X) .

Η Naive Bayesian formula υπολογισμού της εκ των υστέρων πιθανότητας είναι Όπου το P(C) είναι η εκ των προτέρων πιθανότητα ένα sample να ανήκει στην κλάση C. Η δεσμευμένη πιθανότητα P(X|C) αφορά την πιθανότητα ένα sample με χαρακτηριστικά ίδια με του X να ανήκει στην κλάση C.

$$P(C \mid \mathbf{X}) = \frac{P(C)\ P(\mathbf{X} \mid C)}{P(\mathbf{X})}$$

Εικόνα 1: P(c|x)

Έστω C1,C2,...,Cm αναπαριστούν m διαφορετικές κλάσεις. Για κάθε test sample X, η μέθοδος classification υπολογίζει την εκ των υστέρων πιθανότητα P(C j | X) διαμέσου της εκ των προτέρων πιθανότητας P(C = C j) και της P(X | C j). Το εύρος του j είναι από 1 μέχρι m. Το sample X ανήκει στην κλάση της οποίας η εκ των υστέρων πιθανότητα είναι μεγαλύτερη από όλες. Δηλαδή το X κατηγοριοποιείται ως Ci, όταν ικανοποιεί την σχέση:

$P(Ci \mid X) > P(C j \mid X)$, $1 \leq i \neq j \leq m$

Συνδυαστικά με τη formula όπως παρουσιάζεται παραπάνω στην εικόνα προκύπτει

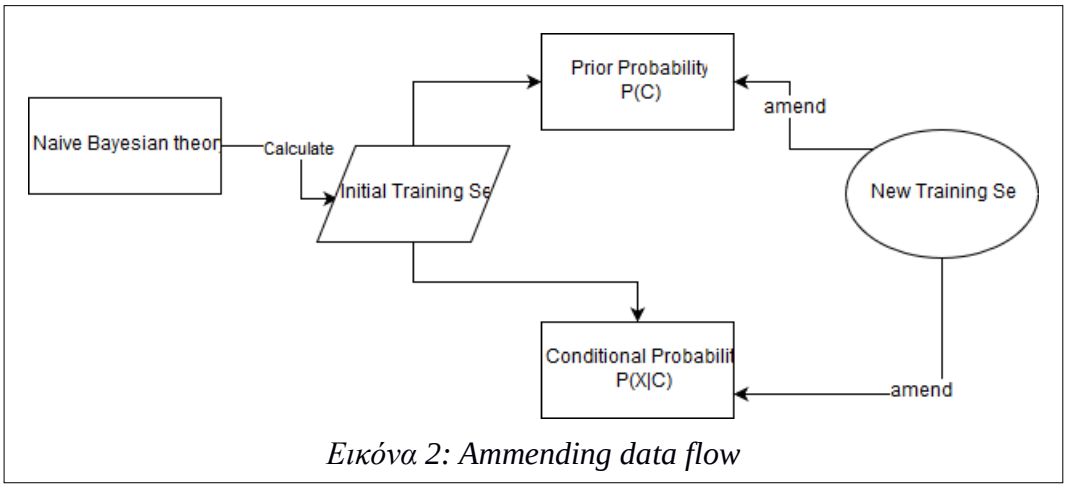
$P(X \mid Ci)P(Ci) > P(X \mid C j)P(C j)$ οπου $1 \leq i \neq j \leq m$

4.1 Incremental Naive Bayesian Formula

Η διαδικασία κατηγοριοποίησης με τον Naive Bayes είναι ως εξής:

Αρχικά, υπολογίζουμε την εκ των προτέρων πιθανότητα του τεστ sample να ανήκει σε κάθε μία από τις δύο κλάσεις με τις δύο παραμέτρους που έχουμε πάρει από το σετ εκπαίδευσης και το αναθέτουμε σε μια από τις δυο κλάσεις, αυτήν που έχει την μεγαλύτερη εκ των υστέρων πιθανότητα.

Οπότε η διαδικασία της αυξητικής μάθησης του classifier αφορά το πώς θα προσδιορίσουμε τη νέα εκ των προτέρων πιθανότητα και την εκ των υστέρων πιθανότητα να ανήκει στην κλάση [22]. Η διαδικασία της ενσωμάτωσης των νέων δεδομένων εκπαίδευσης φαίνεται στην εικόνα 2.



Εικόνα 2: Ammending data flow

Κατά τη διάρκεια της αυξητικής εκπαίδευσης ο αλγόριθμος δε χρειάζεται να προσπελάσει εκ νέου τα δεδομένα εκπαίδευσης, αλλά χρειάζεται μόνο πρόσβαση σε δυο αποθηκευμένες στατιστικές παραμέτρους. Δεδομένης της πληροφορίας στο νέο σετ δεδομένων το σύστημα ενσωματώνει και επαναποθηκεύει αυτές τις δυο στατιστικές παραμέτρους. Έτσι, η φόρμουλα του classifier ελαχιστοποιείται με χρήση στατιστικής.

Έστω ότι D είναι το τωρινό σετ εκπαίδευσης, T είναι το καινούριο σετ προς προσάρτηση ,x_p = (A_1, A_2,...A_i,..., A_n) ∈T είναι το καινούριο sample για ενσωμάτωση και c $_p$ η κατηγορία του.

$$\theta_j' = \begin{cases} \frac{s}{s+1}\theta_j + \frac{1}{1+s} & \text{when } c_p = c_j \\ \frac{s}{s+1}\theta_j & \text{when } c_p \neq c_j \end{cases}$$

Εικόνα 3: εκ των προτέρων πιθανότητα

Ως θ j = P(c = c j) ορίζουμε την εκ των προτέρων πιθανότητα να ανήκει το sample σε αυτήν την κλάσση c $_j$. Τότε η φόρμουλα για την πιθανότητα εκ των προτέρων γίνεται όπως φαίνεται στην εικόνα 3.

όπου s =| D | + | T |. Με | D | συμβολίζεται ο αριθμός των samples στο σετ εκπαίδευσης D, ενώ | T | ο αριθμός των samples στο νέο σετ εκπαίδευσης T.

Έστω θ $_{ik|j}$= P(A_i = a_k | c = c $_j$) είναι η εκ των προτέρων πιθανότητα του χαρακτηριστικού A_i με τιμή a_k να ανήκει στην κλάση c_j . Τότε η φόρμουλα ενσωμάτωσης της δεσμευμένης πιθανότητας γίνεται

$$\theta_{ik|j}' = \begin{cases} \frac{m}{1+m}\theta_{ik|j} + \frac{1}{1+m} & \text{when } c_p = c_j \text{ and } A_i = a_k \\ \frac{m}{1+m}\theta_{ik|j} & \text{when } c_p = c_j \text{ and } A_i \neq a_k \\ \theta_{ik|j} & \text{when } c_p \neq c_j \end{cases}$$

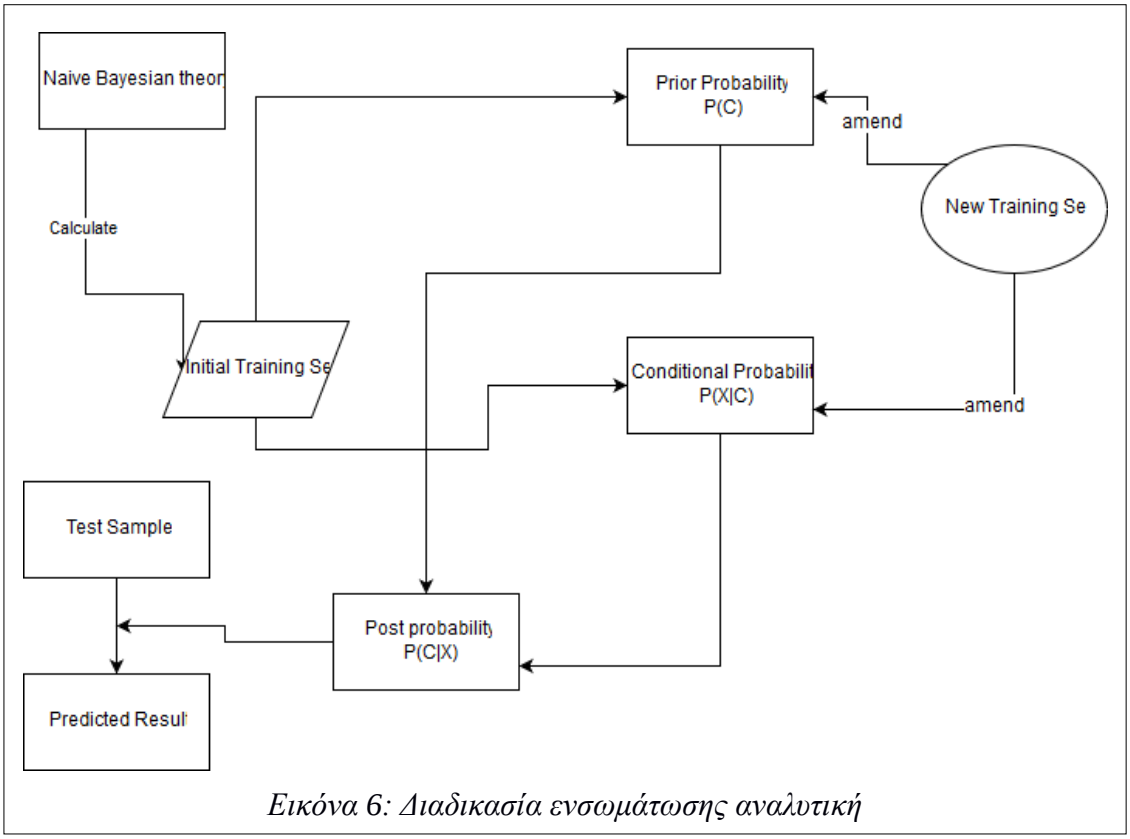
Εικόνα 4: Τυπος ενσωμάτωσης δεσμευμένης πιθανότητας

$$m = |A_i| + count(c_j)$$

Εικόνα 5: Τύπος πλήθους τιμών

Το |Ai| αντιπροσωπεύει των αριθμό των τιμών που παίρνει το χαρακτηριστικό | Ai |, ενώ το count(cj) τον αριθμό των samples των οποίων η κλάση είναι cj.

Έτσι η αυξητική προσάρτηση δεδομένων εκπαίδευσης στο μοντέλο του classifier έχει ολοκληρωθεί μετά την τροποποίηση των δυο στατιστικών παραμέτρων που παρουσιάστηκαν παραπάνω. Η όλη διαδικασία φαίνεται καλύτερα στην εικόνα 6.



Εικόνα 6: Διαδικασία ενσωμάτωσης αναλυτική

4.2 KNN

Ο αλγόριθμος KNN (K nearest neighbors) αφορά μια μη παραμετρική μέθοδο που χρησιμοποιείται για classification regression. Και στις δυο περιπτώσεις η είσοδος αποτελεί τα K κοντινότερα στοιχεία εκπαίδευσης στον χώρο χαρακτηριστικών. Στην περίπτωση του classification η έξοδος του αλγορίθμου είναι η κλάση στην οποία ανήκει το sample, δηλαδή ανήκει από την majority vote των γειτόνων του με το sample να κατηγοριοποιείται στην κλάση που ανήκει η πλειοψηφία των K γειτόνων του (όπου το κ είναι ένας θετικός ακέραιος συνήθως μικρός). Πιο συγκεκριμένα κάθε sample περιγράφεται από ένα διάνυσμα χαρακτηριστικών, συνήθως πολλών διαστάσεων. Δοθέντος τα χαρακτηριστικά ενός αντικειμένου ο στόχος είναι να βρεθεί το κοντινότερο sample-αντικείμενο ως προς τα χαρακτηριστικά με βάση μέτρηση όπως η Ευκλείδεια απόσταση.

Στην πάροδο των ετών, τεχνικές για την επίλυση του ακριβούς και K προσεγγιστικών κ γείτονες έχουν προταθεί από γραμμικές αναζητήσεις των αντικειμένων, σε k-D trees τα οποία κάνουν παράλληλες τομές στα δεδομένα, spill trees και LSH. Δυστυχώς αυτές οι μέθοδοι έχουν σχεδιαστεί να τρέχουν σε έναν υπολογιστή και έτσι δεν μπορούν να είναι αποδοτικές σε κατανεμημένα περιβάλλοντα. Ένας τρόπος για να λυθεί το πρόβλημα είναι η προσπέλαση από τον δίσκο και μεταφορά στη μνήμη, όταν χρειάζεται. Παρόλο που υπάρχουν αρκετά πολύπλοκοι αλγόριθμοι paging δεν αποδίδουν καλά.

Τα spill-trees αποτελούν μια παραλλαγή των metric trees τα οποία επιτρέπουν αποδοτικές προσεγγιστικές knn αναζητήσεις. Ξέχωρα από τα metric trees, τα παιδιά ενός spill tree κόμβου μπορούν να μοιραστούν αντικείμενα. Κανονικά χρησιμοποιείται v για να δηλωθεί ο κόμβος σε ένα spill tree και v.lc, v.rc για τα δεξιά και αριστερά παιδιά. Πρώτα διαλέγονται δυο παιδιά v.lpv , v.rpv και γίνεται εύρεση του ορίου απόφασης L το οποίο είναι το μέσο A μεταξύ αυτών.

4.3 Εντροπία

Με τον όρο εντροπία, εννοούμε, όπως αυτή διατυπώθηκε από τον *Claude Elwood Shannon, την ποσοτικοποίηση της αβεβαιότητας της πληροφορίας σε ένα δειγματοχώρο. Έστω ένα πείραμα τύχης με n πιθανά αποτελέσματα. Θεωρούμε την τυχαία μεταβλητή X και τα απλά ενδεχόμενα x1...xn που πραγματοποιούνται με πιθανότητες p1...pn και άθροισμα τους ίσο με 1.*

Η εντροπία ορίζεται στην εικόνα 6 με τη σύμβαση 0 log 2 0 = 0 .

$$H(X) = -\sum_{i=1}^{n} p_i \log_2 p_i$$

Εικόνα 7: Τύπος υπολογισμού εντροπίας

5. Σύστημα ανάλυσης συναισθήματος σε περιβάλλον microblogging

5.1 Εισαγωγή

Ο σκοπός της εργασίας είναι η ανάπτυξη ενός συστήματος κατηγοριοποίησης το οποίο βασίζεται σε ενα dataset από feature-extracted tweets τα οποία είναι κατηγοριοποιημένα από ανθρώπους ως αρνητικά ή θετικά. Το σύστημα, όπως αυτό προτάθηκε [23] , το οποίο κάνει feauture extraction λαμβάνει υπόψη ορισμένα attributes τα οποία μπορεί να συνεισφέρουν στο τελικό ύφος του κειμένου με τιμές true ή false όπως είναι για παράδειγμα η ύπαρξη άρνησης στο κείμενο ή η ύπαρξη κεφαλαίων γραμμάτων, ενώ άλλα χαρακτηριστικά όπως είναι η συναισθηματική βαρύτητα των λέξεων οι οποίες από μόνες τους μπορούν να ερμηνευτούν ως θετικές, αρνητικές ή ουδέτερες.

Επίσης μια βασική κατηγορία αποτελούν και τα HashTag τα οποία στο σύστημα αυτό, αφού περάσουν από ένα λεξικό για καθαρισμό και διόρθωση στη συνέχεια αξιολογείται το ύφος τους από σκορ του site SentiWord.net και το συνολικό ύφος των HashTags είναι το άθροισμα των θετικών και αρνητικών HashTags ενός tweet. Μια παρόμοια προσέγγιση ακολουθείται και στα emoticon. Τα χαρακτηριστικά των tweets μετά τον παραπάνω καθαρισμό από emoticons, hashtags καθώς και λέξεων λιγότερο από 2 αναπαριστούν κάποια μοτίβα λόγου που αναφέρονται στην αγγλική γλώσσα και μπορούν να μας πληροφορήσουν σχετικά με το γενικότερο ύφος του μηνύματος. Μερικά από αυτά είναι:

- Oh so [True / False]
- Don’t you [True / False]
- As * As [True / False]
- Question mark [True / False]
- Capitals [True / False]
- Reference [True / False]
- RT [True / False]
- Negations [True / False]
- URL [True / False]
- HT_pos [True / False]
- HT_neg[True / False]
- HT_neu[True / False]
- Emoticon Pos [True / False]

● Emoticon Neg[True / False]
● POS-tags “NN”,”VB”,”ADJ”,”RB”
● swnScore “positive” , “somewhat positive” ,neutral” , “somewhat Negative” , “negative”
● swnScoreTotal “positive” , “somewhat positive” ,neutral” , “somewhat negative” , “negative”
● simt(Resink*) Decimal score

Για παράδειγμα ένα tweet με αρχικό κείμενο “ Just spoke with my fam in #Japan via #Skype! Love to see my little 16-mo-old nephew growing! :) “μετατρέπεται μετά την διαδικασία του preprocessing σε επιμέρους features οπως φφαίνονται παρακάτω. Καθώς αυτή είναι η μορφή που έχει το tweet στην preprocessed φάση επιλέγουμε για την κατηγοριοποίηση features τέτοια τα οποία βοηθάνε στην κατηγοριοποίηση και είναι σταθερά για όλα τα tweet όπως φένεται στο κεφάλαιο 5.6 των αποτελεσμάτων.

{'s_word-9': 1.0,
's_word-8': 1.13,
'__POS_SMILEY__': 'True',
u'via': 'IN',
u'love': 'VBP',
'__REFERENCE__': 'False',
'__OH_SO__': 'False',
's_word-1': 1.02,
's_word-0': 1.03,
's_word-3': 1.1,
's_word-2': 1.0,
's_word-5': 1.61,
's_word-4': 1,
's_word-7': 0.73,
's_word-6': 1.03,
'__AS_GROUND_AS_VEHICLE__': 'False',
u'see': 'VB',
'__HT_NEG__': 'False',
u'__lemma_word__via': 'somewhat_positive',
u'skype': 'NN',

'__is_metaphor__': False,
u'nephew': 'NN',
'__hashtag_lexicon_sum__': 2.0,
'__fullstop__': 'False',
u'__lemma_word__spoke': 'neutral',
u'little': 'JJ',
'__HT_POS__': 'False',
u'__lemma_word__nephew': 'neutral',
u'__lemma_word__old': 'somewhat_positive',
u'__lemma_word__japan': 'neutral',
'pos_position_5': 'VBP',
u'old': 'JJ',
'__LINK__': 'False',
'word-10': u'growing',
'__HT__': 'True',
'__CAPITAL__': 'False',
'__exclamation__': 'True',
's_word-10': 0.94,
u'fam': 'NN',
u'__lemma_word__skype': 'neutral',
'pos_position_9': 'NN',
u'__lemma_word__growing': 'somewhat_negative',
'pos_position_7': 'JJ',
'pos_position_6': 'VB',
'word-6': u'see',
'pos_position_4': 'NN',
'pos_position_3': 'IN',
'pos_position_2': 'NN',
'pos_position_1': 'NN',
'pos_position_0': 'VBD',
'__swn_score__': 0.06000000000000005,
u'__lemma_word__love': 'positive',
'__DONT_YOU__': 'False',
'word-1': u'fam',
'word-0': u'spoke',

'word-3': u'via',
'word-2': u'japan',
'word-5': u'love',
'__NEGATION__': 'False',
'word-7': u'little',
'__questionmark__': 'False',
'word-9': u'nephew',
'word-8': u'old',
'pos_position_8': 'JJ',
u'__lemma_word__fam': 'neutral',
u'japan': 'NN',
u'spoke': 'VBD',
u'__lemma_word__little': 'somewhat_negative',
u'__lemma_word__see': 'neutral',
'__LOVE__': 'False',
'__RT__': 'False',
'__NEG_SMILEY__': 'True',
't-similarity': 0.0,
'pos_position_10': 'VBG',
'__punctuation_percentage__': 14.0,
'__LAUGH__': 'False',
u'growing': 'VBG',
'__multiple_chars_in_a_row__': 'False',
'word-4': u'skype'}

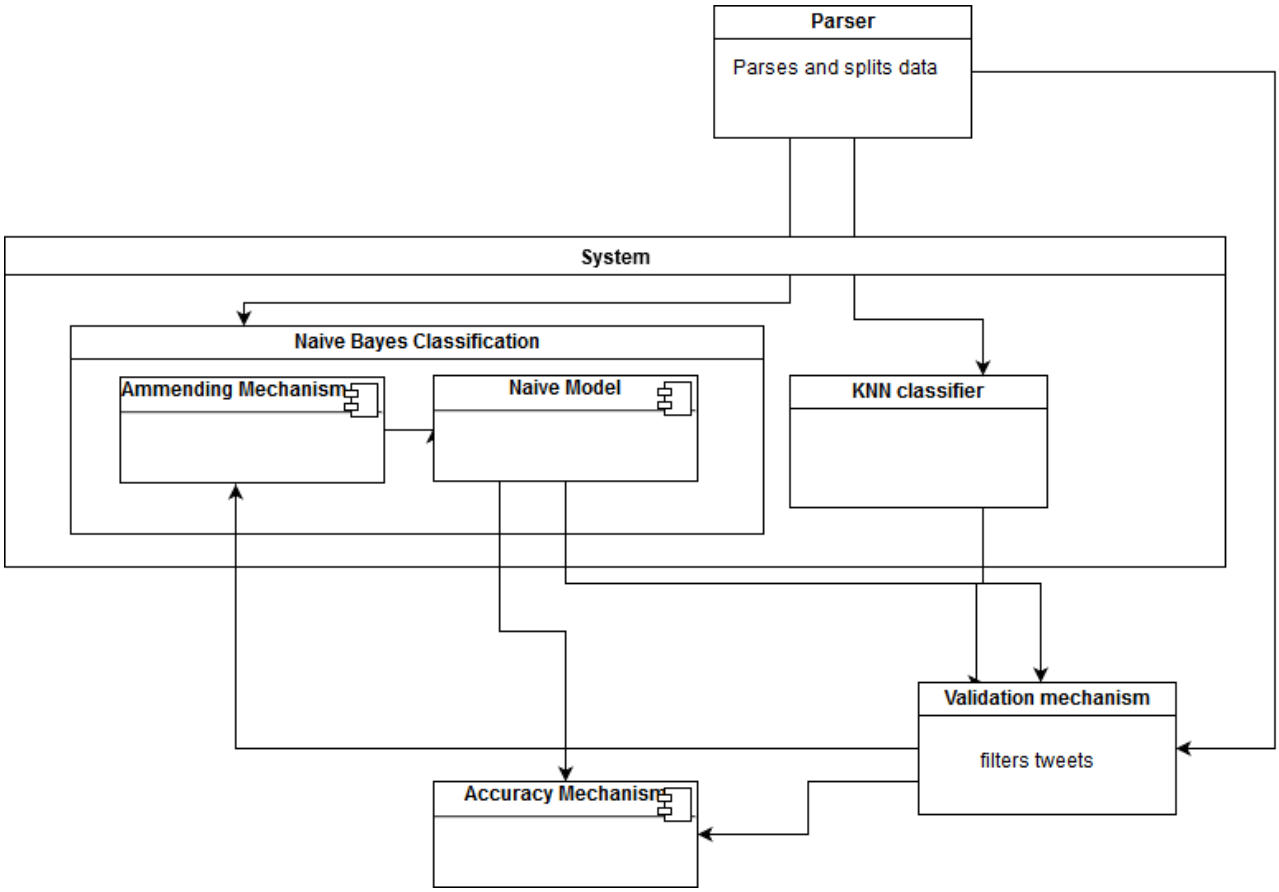
5.2 Συστατικά του συστήματος

Το σύστημα που αναπτύχθηκε βασίστηκε σε τρείς βασικές κατηγορίες. Αυτές είναι:

- Το component το οποίο διαβάζει τα tweets μέσω από την διαδικασία της ανάλυσης csv αρχείου οπου βρίσκονται τα feature- extracted tweets.
- Ο μηχανισμός ο οποίος κατασκεύαζει το μοντέλο, τις πιθανότητες δηλαδή του Naïve bayes classifier.
- Το component που κάνει την εξαγωγή των πιθανοτήτων από τα νέα tweets και την ενσωμάτωσης τους στο μοντέλο του Naïve classifier.

- Το component που κάνει classification ο KNN classifier.

Τα συστατικά φαίνονται καλύτερα στη εικόνα 8. Ωστόσο υπάρχουν και κάποιοι συμπληρωματικοί μηχανισμοί όπως ο μηχανισμός που βρίσκει τις κοινές προβλέψεις μεταξύ του αρχικού μοντέλου και του τελικού.



Εικόνα 8 μέρη του συστήματος

5.3 Παρουσίαση της προσέγγισης

Αφού επιλεγούν κάποια από αυτά με σκοπό την ως επί το πλείστον ορθή πρόβλεψη της πολικότητας των tweet, στη συνέχεια χτίζεται το μοντέλο πρόβλεψης. Κατά τη διάρκεια χρήσης του αναπόφευκτα μειώνεται η ακρίβειά του. Το παρόν σύστημα εφαρμόζει τρόπο εντοπισμού της έλλειψης ακρίβειας του classifier και στη συνέχεια φιλτράρισμα των καθαρών tweets, με σκοπό την ενσωμάτωση τους στο μοντέλο με αυξανόμενο τρόπο για την αύξηση της ακρίβειάς του μέσω ενός συμπληρωματικού classifier KNN. Για αρχικός κατηγοριοποιητής επιλέχθηκε ο Naive Bayes, λόγω της δυνατότητας που προσφέρει στην παρέμβαση της μετέπειτα επεξεργασίας των παραμέτρων

του. Στην πορεία αξιολογείται η απόδοση του συστήματος όσον αφορά την αποτελεσματικότητα του και διατυπώνονται προτάσεις για βελτίωσή του.

Με την πάροδο του χρόνου το διαμορφωμένο μοντέλο κατηγοριοποίησης ενδέχεται να μην παράγει τα επιθυμητά αποτελέσματα – προβλέψεις για τα δεδομένα εισόδου. Λόγω της έλλειψης νομοτέλειας για την κατηγοριοποίηση η στατικότητα των δεδομένων εκπαίδευσης δεν μπορεί να καλύψει τις μελλοντικές μεταβολές των δεδομένων.

Η ανίχνευση της έλλειψης ακρίβειας του αλγορίθμου δεδομένης της απουσίας κλάσης για τα δεδομένα που λαμβάνουμε μπορεί να προσεγγισθεί με διάφορους τρόπους. Για την υλοποίηση της συγκεκριμένης εργασίας χρησιμοποιήθηκε η έννοια της Εντροπίας για την ανίχνευση της αποδοτικότητας του μοντέλου. Άλλες προσεγγίσεις στηρίζονται σε ανθρώπινη παρέμβαση για την εξαγωγή αποτελέσματος από τυχαία δεδομένα εισόδου, καθώς και σε υβριδικά μοντέλα (ανθρώπινου και αυτοματοποιημένου) παράγοντα.

5.4 Εκπαίδευση Naive Bayes Classifier

Ο classifier που χρησιμοποιήθηκε είναι ο Naive Bayes. Ανήκει στην κατηγορία των classifiers που εφαρμόζουν Naive Bayesian formula. Αρχικά γίνεται η κατασκευή του μοντέλου του κατηγοριοποιητή.

Η συλλογή τους γίνεται σε ανάλογα μεγέθη του αρχικού μας σετ δεδομένων. Από πειραματισμούς διαπιστώθηκε ότι 5-10% του πλήθους του σετ εκμάθησης μπορούν να διαχειριστούν εύκολα. Ωστόσο μπορεί να γίνει βελτίωση της αναλογίας αυτής.

Στη συνέχεια για το κάθε ένα από τα καινούρια δεδομένα που φτάνουν, ο αλγόριθμος δίνει κάποια πρόγνωση (πρόβλεψη) πάνω στην οποία εφαρμόζεται ο τύπος της εντροπίας για τα αποτελέσματα του αλγορίθμου.

5.5 Κατασκευή μοντέλου KNN

Τα δεδομένα εκπαίδευσης που χρησιμοποιήθηκαν για το μοντέλο του βασικού μας κατηγοριοποιητή Naive Bayes χρησιμοποιήθηκαν για την αποτύπωση του χώρου πάνω στον οποίο ο nearest neighbor classifier θα κάνει την κατασκευή του αντίστοιχου μοντέλου. Η υλοποίηση του KNN είναι προσαρμοσμένη για κατανεμημένο περιβάλλον χρησιμοποιώντας hybrid spill-trees.

5.5 Υπολογισμός Εντροπίας Τεστ Σετ

Η προσέγγιση της εργασίας αφορά την εύρεση της εντροπίας του τεστ σετ ως εξής :

Έστω ότι Z = {X_1, X_2,..., X_n} το σετ πανω στον οποίο δοκιμάζουμε τον αλγόριθμό μας.
Το σύνολο των αποτελεσμάτων μας {0 , 1 }
Ο τύπος της εντροπίας τροποποιείται όπως φαίνεται στην εικόνα 9

$$H(X) = \mathrm{H_b}(p) = -p \log_2 p - (1-p) \log_2 (1-p)$$

Εικόνα 9: Τροποποιημένος τύπος εντροπίας

Με

$$Pr(X = 1) = p$$

Εικόνα 10: Μεταβλητές εντροπίας

$$Pr(X = 0) = 1 - p$$

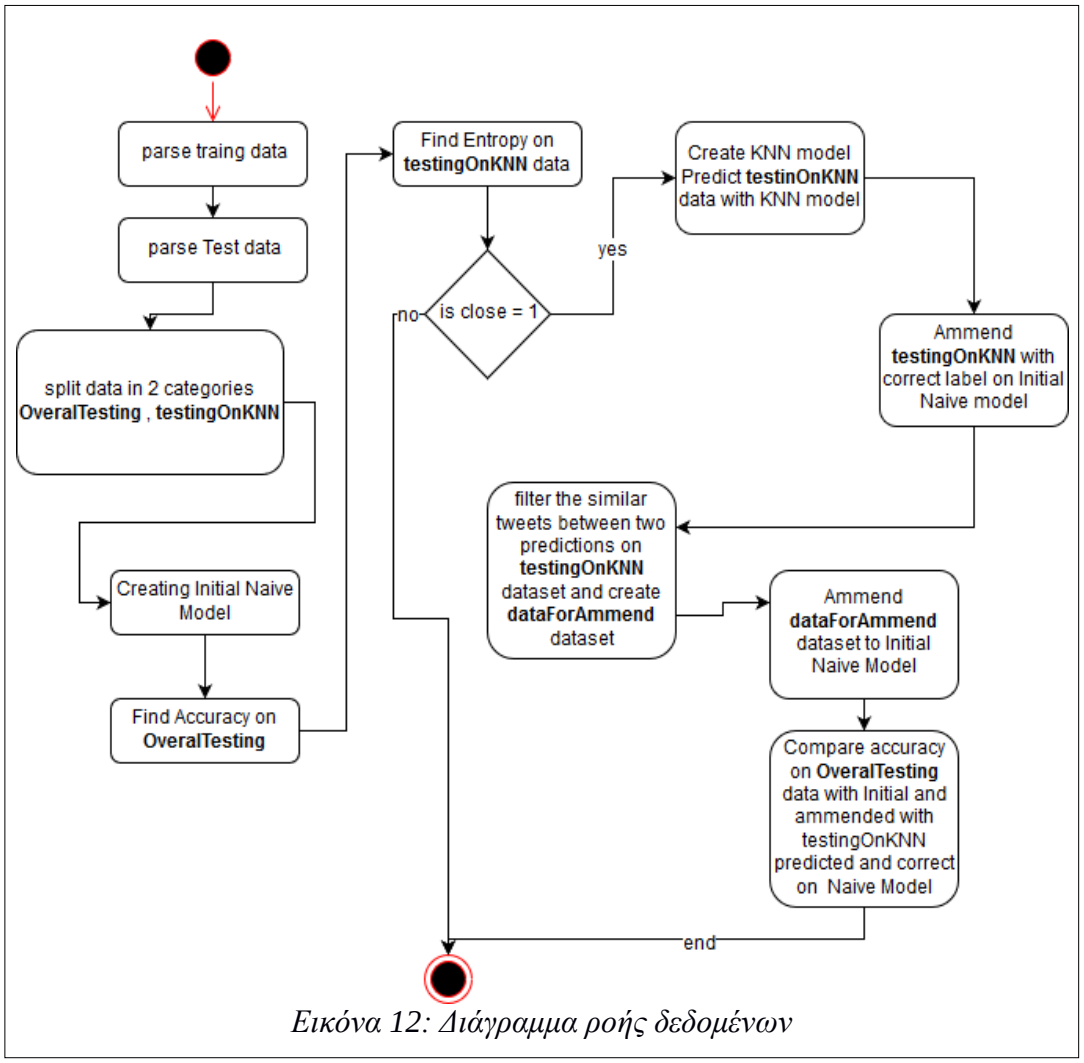
Εικόνα 11: Μεταβλητές εντροπίας

καθώς η ροή των tweets είναι συνεχής δημιουργούνται “κομμάτια” από τα άγνωστα κατηγοριοποιημένα δεδομένα και αφού κατηγοριοποιηθούν από τον αλγόριθμο υπολογίζεται η εντροπία στο κάθε “κομμάτι”. Τα τεστ δεδομένα στα οποία υπολογίζεται η εντροπία είναι περίπου 10% των αρχικών δεδομένων εκπαίδευσης. Όσο η εντροπία πλησιάζει στο 1 τόσο η πιθανότητα ανάθεσης των κλάσεων φτάνει ισόποσα το οποίο αποτελεί ένδειξη εξασθένησης του κατηγοριοποιητή.

Το σετ από αυτά στο οποίο η εντροπία είναι κοντά στο 1 κατηγοριοποιείται από τον KNN και η πρόβλεψη του χρησιμοποιείται ως κλάση του tweet.

Στη συνέχεια με τον τύπο του amend, όπως αυτός έχει οριστεί πιο πάνω, πέρνει ως βάση τις πιθανότητες του αρχικού μοντέλου και κατηγοριοποιεί ένα σετ δοκιμών μετρώντας την ακρίβεια του. Επίσης στο αρχικό μοντέλο ενσωματώνεται και ένα σετ δοκιμής με γνωστές τις κατηγορίες του ούτως, ώστε να βρεθεί το μέγιστο ποσοστό ακρίβειας, για να αξιολογηθεί ο αλγόριθμος. Τα αποτελέσματα της ακρίβειας φαίνονται στο κεφάλαιο 5.6.

Η ροή του συστήματος φαίνεται στο διάγραμμα της Εικόνας 12.



Εικόνα 12: Διάγραμμα ροής δεδομένων

5.6 Αποτελέσματα

Οι δοκιμές στα δεδομένα που έγιναν περιλάμβαναν τις εξής σταθερές
Test data = 10.000 tweets
KNN k = 5 neighbors

Σαν μέθοδος αξιολόγησης χρησιμοποιήθηκε η ακρίβεια του classifier, δηλαδή σε ένα άγνωστο tweet αν το προβλέπει σωστά
accuracy = Αριθμός των σωστά κατηγοριοποιημένων tweets / αριθμό όλων των tweets.

Η *Ακρίβεια του αρχικού μοντέλου* ειναι η πρόβλεψη που δίνει ο Naive κατηγοριοποιητής με τις αρχικές παραμέτρους P(x|c) και P(c) εκπαιδεύοντας τον με τα *Training Data* οπως έχουν οριστει στο κεφάλαιο 4.

Ως *Βέλτιστη προσδοκώμενη ακρίβεια* ορίζεται η ακρίβεια του μοντέλου αφου έχουν ενσωματωθεί τα *Ammended data*, έχοντας επαναυπολογιστεί οι παράμετροι P(x|c) και P(c), σε αυτό tweets με ορθή κλάσση,.

Ως *Πραγματικό αποτέλεσμα* ορίζεται η ακρίβεια του μοντέλου αφού έχουν ενσωματωθεί tweets, τα *Ammended data*, των οποίων η πολικότητα προκύπτει ως η κοινή πρόβλεψη του αρχικού μοντέλου Naive και του κατηγοριοποιητή KNN.

Feature set 1 : *laugh, neg, ohso, capital, isMetaphor, love.*	Ακρίβεια αρχικού μοντέλου	Βέλτιστη προσδοκώμενη ακρίβεια	Πραγματικό αποτέλεσμα
Training Data = 419 Amended data = 8078	0.60002	0.66861	0.33410
Training Data = 3976 Amended data = 4271	0.48023	0.48459	0.48429
Training Data = 16155 Amended data = 3767	0.47741	0.48139	0.47741
Training Data = 36028 Amended data = 3113	0.48003	0.48023	0.48023

Feature set 2 :*capital, isMetaphor, love.*	Ακρίβεια αρχικού μοντέλου	Βέλτιστη προσδοκώμενη ακρίβεια	Πραγματικό αποτέλεσμα
Training Data = 388 Amended data = 10072	0.62042	0.67009	0.32930
Training Data = 4051 Amended data = 1438	0.33353	0.61728	0.33353
Training Data = 11937 Amended data = 71	0.33934	0.33934	0.33934
Training Data = 35973 Amended data = 151	0.33018	0.33018	0.33018

Feature set 3 : posSm, negSm, swn_positive, swn_negative, swn_somewhatPositive, swn_neutural,swn_somewhatNegative, neg, ohso, capital, isMetaphor, love

Οι όλες οι τιμές εκτός απο το __swn_score__ είναι σε μορφή true false ενω το __swn_score__ σε μορφή δεκαδικού. Για να μετατραπεί σε true false, χρησιμοποιήθηκε η κατηγοριοποίηση

swn_score = positive, (> 1.2)

swn_score = somewhat positive, (> 0.05 ≤ 1.2)

swn_score = neutral, (≤ 0.05 ≥ 0.95)

swn_score = somewhat negative, (< 0.95 ≥ 0.2)

swn_score = negative, (< 0.2)

feature set 3 : *posSm, negSm, swn_positive, swn_negative, swn_somewhatPositive, swn_neutural,swn_somewhatNegative, neg, ohso, capital, isMetaphor, love*	Ακρίβεια αρχικού μοντέλου	Βέλτιστη προσδοκώμενη ακρίβεια	Πραγματικό αποτέλεσμα
Training Data = 383 Amended data = 6416	0.64860	0.64860	0.64860
Training Data = 4050 Amended data = 6513	0.64285	0.64285	0.64285
Training Data = 8057 Amended data = 3332	0.32598	0.32598	0.32598

Τα αποτελέσματα δείχνουν μια τάση για σταθεροποίηση του αποτελέσματος οταν ο αριθμός των tweets για εκπαίδευση αρχίζει και γίνεται ίσος με τον αριθμό των tweet που ενσωματώνονται, οπως επίσης και όταν τα features είναι πολλά ή πολύ λίγα Το φαινόμενο αυτό οφείλεται πιθανότατα στο γεγονός οτι οι αρχικές πιθανότητες διαφοροποιούνται ελάχιστα γιατί έχουν πολύ μεγάλύτερο βάρος, στην διάρκεια της ενσωμάτωσης σε τέτοιο βαθμό ωστε απο ένα σημείο και μετά να μήν επηρεάζουν το αποτέλεσμα. Αυτό επιβεβαιώνεται και απο τον πίνακα 1 στον οποίο για λίγα train data βλέπουμε οτι η ακρίβεια με το ιδανικό μοντέλο αυξήθηκε 6% ενώ στην περίπτωση της ανάθεσης κλάσης μέσω της σύγκλισης του δεύτερου κατηγοριοποιητή η ακρίβεια πέφτει κατα πολύ λόγω της λανθασμένης πρόβλεψης της κλάσης. Οι περιπτώσεις στις οποίες η ενσωμάτωση βελτιώνει την ακρίβεια του είναι δυο και η αύξηση ειναι της τάξης του 0.5%. Πιθανότατα η αύξηση αυτή να προκύπτει λόγω της επιλογής των κατάλληλων features (τοσο σε πλήθος, όσο και βαρύτητα πολικότητας).

6. Επίλογος- Συμπεράσματα

Η διαδικασία του εμπλουτισμού του μοντέλου κατηγοριοποίησης προϋποθέτει την επιλογή μοντέλου τέτοιου που το υποστηρίζει. Ένας από αυτούς τους κατηγοριοποιητές είναι ο Naive bayes.

Η διαδικασία αυτή, αν και έχει πολλά οφέλη, κυρίως ως προς την ταχύτητα με την οποία ενημερώνεται το μοντέλο, ωστόσο μπορεί με δεδομένα ή με επιλογή χαρακτηριστικών να μην αποδώσει κατάλληλα.

Διαπιστώνουμε επίσης ότι όσο αυξάνονται σε μέγεθος τόσο τα tweets όσο και τα features των tweet η διαδικασία της ενσωμάτωσης στο αρχικό μοντέλο να μην αποδώσει ιδιαίτερα, λόγω της μικρής εναλλαγής στις πιθανότητες αυτού.

Τα αποτελέσματα του πειράματος δεν προσέφεραν ιδιαίτερη γνώση κυρίως λόγω του συνδυασμού του KNN στο αποτέλεσμα καθώς και της χαμηλής ακρίβειας εξαρχής του Naive bayes.

Μια επέκταση θα μπορούσε να είναι να γίνει τριπλή επαλήθευση της πρόβλεψης του κάθε tweet με έναν τρίτο classifier όπως πχ ο SVN και να δοκιμασθεί ο αλγόριθμος με διαφορετικού είδους δεδομένα.

Βιβλιογραφία

1. N. Indurkhya και F. J. Damerau, «Sentiment Analysis and Subjectivity,» σε Handbook of Natural Language Processing Second Edition, CRC Press.

2. R. Feldman, «Techniques and Applications for Sentiment Analysis,» Communications of the ACM, αρ. 56(4), p. 82–89, 04 2013.

3. N. Farra, E. Challita, R. Assi και H. Hajj, «Sentence-level and Document-level Sentiment Mining for Arabic Texts,» σε Data Mining Workshops (ICDMW), IEEE International Conference, 2010.

4. B. Liu και M. Hu, «Opinion Mining, Sentiment Analysis, and Opinion Spam Detection,» 15 5 2004.

5. M. Ganapathibhotla και B. Liu, σε COLING '08 - Proceedings of the 22nd International Conference on Computational Linguistics, 2008.

6. P. Turney, «Thumbs Up or Thumbs Down? Semantic Orientation Applied to Unsupervised Classification of Reviews,» σε Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL'02), 2002.

7. M. Taboada, J. Brooke, M. Tofiloski, K. Voll και M. Stede, «Lexicon- Based Methods for Sentiment Analysis,» Computational Linguistics, τόμ. 37, αρ. 2, pp. 267-307, 2011.

8. D. Derks, A. E. R. Bos και J. Von Grumbkow, «Emoticons and online message interpretation,» Social Science Computer Review, τόμ. 26, αρ. 3, pp. 379-388, 2007.

9. M. Thelwall, K. Buckley, G. Paltoglou, D. Cai και A. Kappas, «Sentiment Strength Detection in Short Informal Text,» Journal of the American Society for In- formation Science and Technology, τόμ. 61,αρ. 12, p. 2544–2558, 12 2010.

10. A. Go, R. Bhayani και L. Huang, «Twitter sentiment classification using distant supervision,» σε In: Proceeding LSM '11 Proceedings of the Workshop on Languages in Social Media, 2009.

11. E. Kouloumpis, T. Wilson και J. Moore, «Twitter sentiment analysis: The Good the Bad and the OMG!,» σε In: Lada A. Adamic, Ricardo A. Baeza-Yates, and Scott Counts, editors, Proceedings of the Fifth Inter- national Conference on Weblogs and Social Media, ICWSM’ 11, Barcelona, 2010.

12. A. Reyes, P. Rosso και D. Buscaldi, «From Humor Recognition to Irony Detection: The Figurative

Language of Social Media.,» Data & Knowledge Engineering, αρ. 73, pp. 1-12, 2012.

13. Y. Hao και T. Veale, «An Ironic Fist in a Velvet Glove: Creative Mis-Representation in the Construction of Ironic Similes,» Minds and Machines, τόμ. 20, αρ. 4, p. 635–650, 2010.

14. D. Davidov, O. Tsur και A. Rappoport, «Semi-supervised recognition of sarcastic sentences in twitter and amazon.,» σε In Proceedings of the Fourteenth Conference on Computational Natural Language Learning, CoNLL 2010, 2010.

15. P. R. T. V. Antonio Reyes, «A Multidimensional Approach for Detecting Irony in Twitter.,» Languages Resources and Evaluation, αρ. 47(1), pp. 239-268., 2013.
16. P. Harrington, Machine Learning in Action, J. Bleiel, Επιμ., Manning Publications, 2012.
17. J. R. Quinlan, «Induction of Decision Trees.,» Machine Learning, τόμ. 1, αρ. 1, pp. 81-106 , 1986.
18. J. R. Quinlan, «Combining Instance-Based and Model-Based Learning.,» ICML, pp. 236-243, 1993.
19. P. Berka και J. Rauch, «Machine Learning and Association Rules,» σε In 19th Int. Conf. On Computational Statistics COMPSTAT, 2010.
20. "Scala Programming Language", .
21. "Apache Spark", .
22. Shuxia Ren *, Yangyang Lian,Xiaojian Zou, Incremental Naïve Bayesian Learning Algorithm based on Classification Contribution Degree, JOURNAL OF COMPUTERS, VOL. 9, NO. 8, AUGUST 2014.
23. Maria Karanasou, Christos Doulkeridis, Maria Halkidi «DsUniPi: An SVM-based Approach for Sentiment Analysis of Figurative Language on Twitter»,Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015), pages 709–713, Denver, Colorado, June 4-5, 2015

Παράρτημα

Στιγμιότυπα κώδικα

```java
package org.sparkexample.classifiers;

import java.util.ArrayList;
import java.util.List;

import org.apache.spark.SparkConf;
import org.apache.spark.SparkContext;
import org.apache.spark.api.java.JavaPairRDD;
import org.apache.spark.api.java.JavaRDD;
import org.apache.spark.api.java.function.Function;
import org.apache.spark.api.java.function.PairFunction;
import org.apache.spark.sql.SparkSession;
import org.sparkexample.pojo.PojoRow;

import scala.Tuple2;

public class CustomNaiveBayes {
	public static void main(String[] args) {
		SparkConf sparkConf = new SparkConf().setAppName("org.sparkexample.WordCount2").setMaster("local[*]");
		/*
		 * @SuppressWarnings("resource") JavaSparkContext sc = new
		 * JavaSparkContext(sparkConf);
		 */
		SparkContext sc1 = new SparkContext(sparkConf);
		@SuppressWarnings("resource")
		SparkSession sqlSpark = new SparkSession(sc1);

		// Load and parse the data file.
		// Training data 40k tweets
		String trainingDatapath = "data/60000tweetsTrain.csv";
		DataExtraction dataTrain = new DataExtraction(sc1, trainingDatapath);
		// Test and ammending data 20K tweets
		String testDatapath = "data/60000tweetsTest.csv";
		DataExtraction dataTest = new DataExtraction(sc1, testDatapath);

		// Split the data into training and test sets (30% held out for testing)

		JavaRDD<PojoRow> trainingData = dataTrain.getDatalabeledPoint();
		JavaRDD<PojoRow>[] splits = dataTest.getDatalabeledPoint().randomSplit(new double[] { 0.5,  0.3, 0.2,});
```

Εικόνα 13: Main class

```
87                      public PojoRow call(DataFeature dataFeature) throws Exception {
88
89                              double neg;
90                              boolean negation = dataFeature.getData().getBoolean("__NEGATION__");
91                              if (negation) {
92                                      neg = 1;
93                              } else {
94                                      neg = 0;
95                              }
96                              double negSm;
97                              boolean negSmile = dataFeature.getData().getBoolean("__NEG_SMILEY__");
98                              if (negSmile) {
99                                      negSm = 1;
100                             } else {
101                                     negSm = 0;
102                             }
103
104                             double posSm;
105                             boolean pos = dataFeature.getData().getBoolean("__POS_SMILEY__");
106                             if (pos) {
107                                     posSm = 1;
108
109                             } else {
110                                     posSm = 0;
111                             }
112                             double ohso;
113                             boolean ohsoBool = dataFeature.getData().getBoolean("__OH_SO__");
114                             if (ohsoBool) {
115                                     ohso = 1;
116
117                             } else {
118                                     ohso = 0;
119                             }
120                             double capital;
121                             boolean capitalBool = dataFeature.getData().getBoolean("__CAPITAL__");
122                             if (capitalBool) {
123                                     capital = 1;
124
125                             } else {
126                                     capital = 0;
```

Εικόνα 14: Parsing class

```
--
33          private void ammend(List<PojoRow> listOfAmmendingTweets, long countOfTrainingData) {
34                  Long long1 = new Long(countOfTrainingData);
35                  double countOfTrainingDataDouble = long1.doubleValue();
36                  double s = countOfTrainingDataDouble + listOfAmmendingTweets.size();
37                  for (PojoRow tweet : listOfAmmendingTweets) {
38                          if (tweet.label == 1.0) {
39                                  this.numberOfCgood = this.numberOfCgood + 1;
40                                  this.posCgood = ((s / (s + 1)) * this.posCgood) + (1 / (s + 1));
41                                  this.posCbad = (s / (s + 1)) * this.posCbad;
42                                  this.setPXc(tweet);
43                          } else {
44                                  this.numberOfCbad = this.numberOfCbad + 1;
45                                  this.posCgood = ((s / (s + 1)) * this.posCgood);
46                                  this.posCbad = ((s / (s + 1)) * this.posCbad) + (1 / (s + 1));
47                          }
48                  }
49          }
50
51          private void setPXc(PojoRow tweet) {
52                  if (tweet.label == 1.0) {
53                          double m = 2 + this.numberOfCgood;
54                          double[] tweetFeatures = tweet.features.toArray();
55                          for (int i = 0; i < tweetFeatures.length; i++) {
56                                  if (tweetFeatures[i] == 1) {
57                                          // for good tweets with 1
58                                          double tmp1 = this.possibilityOfXEq1givenCgood[i];
59                                          double newTmp1 = ((m / (double)(m + 1)) * tmp1) + (1 / (double)(m + 1));
60                                          this.possibilityOfXEq1givenCgood[i] = newTmp1;
61                                          // for good tweets with 0
62                                          double tmp0 = this.possibilityOfXEq0givenCgood[i];
63                                          double newTmp0 = ((m / (double) (m + 1)) * tmp0);
64                                          this.possibilityOfXEq0givenCgood[i] = newTmp0;
65                                  }
66                          }
67                  } else {
68                          double m = 2 + this.numberOfCbad;
69                          double[] tweetFeatures = tweet.features.toArray();
70                          for (int i = 0; i < tweetFeatures.length; i++) {
71                                  if (tweetFeatures[i] == 1) {
72                                          // for good tweets with 1
```

Εικόνα 15: Ammend procedure

Οδηγίες εγκατάστασης

Ο κώδικας βρίσκεται στο link ://github.com/al3xkyr/Spark_KNN. Για την χρήση του θα χρειαστεί να γίνει import στο IDE και να γίνει build μέσω maven μέσω του Pom.xml. Το spark-knn-0.2.0.jar βρίσκεται μέσα στα αρχεία καθώς και τα data στον αντίστοιχο φάκελο. Στην περίπτωση αλλαγής δεδομένων θα πρέπει να αλλαχτεί το path για αυτά.

Λεπτομέρειες υλοποίησης

Το σύστημα αποτελείται από 2 packages και 8 κλάσεις

- org.sparkexample.classifiers
  1. src/main/java/org/sparkexample/classifiers/AmmendManager.java
  2. src/main/java/org/sparkexample/classifiers/CustomNaiveBayes.java
  3. src/main/java/org/sparkexample/classifiers/DataExtraction.java
  4. src/main/java/org/sparkexample/classifiers/InitialParameters.java
  5. src/main/java/org/sparkexample/classifiers/KNNClassification.java
  6. src/main/java/org/sparkexample/classifiers/NaiveBayesModel.java
- org.sparkexample.pojo
  1. src/main/java/org/sparkexample/pojo/DataFeature.java
  2. src/main/java/org/sparkexample/pojo/PojoRow.java

Η οργάνωση έχει γίνει ως εξής, Η CustomNaiveBayes περιέχει τη main και τη βασική ροή του προγράμματος. Αφού διαβάζονται τα csv με τα δεδομένα αποκτώνται από το apache spark σε local cluster όπου η προσομοίωση των nodes γίνεται με executors. Στη συνέχεια, αφού έχουν δημιουργηθεί JavaRDD από τα δεδομένα, τμηματοποιούνται σε τεστ και τρειν. Οι αναλογίες στα τεστ και train δεδομένα φαίνονται στους πίνακες του κεφαλαίου 7.

Στη συνέχεια δημιουργείται το αρχικό μοντέλο στην κλάση InitialParameters. Λόγω του πιθανού φόρτου χρησιμοποιείται το spark για τον υπολογισμό των πιθανοτήτων P(C) και P (X | C) και αυτά αποθηκεύονται στην μνήμη για περαιτέρω χρήση. Στη συνέχεια εκπαιδεύεται ο KNN με το ίδιο set που εκπαιδεύτηκε ο Naive. Ύστερα, γίνεται ενσωμάτωση στο μοντέλο Naive με το σετ 1, όπως αυτό αναφέρεται παραπάνω, στο οποίο παρέχονται σωστές τιμές και υπολογίζεται η ακρίβειά του μετά την ενσωμάτωση, η οποία γίνεται στην κλάση AmmendManager, στο τεστ σετ, σετ2 . Στη συνέχεια γίνεται πρόβλεψη του σετ 3 και από το αρχικό μοντέλο InitialParameters καθώς και από τον KNN και φιλτράρονται οι κοινές προβλέψεις παράγοντας ένα νέο σετ, σετ 4. Ύστερα ενσωματώνεται το σετ 4 στο αρχικό μοντέλο Naive bayes και γίνεται μέτρηση της ακρίβειας στο σετ 2. Τέλος εκτυπώνονται τα ποσοστά ακρίβειας στο σετ 2 από τα μοντέλα.