ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΤΟΜΕΑΣ ΣΥΣΤΗΜΑΤΩΝ ΜΕΤΑΔΟΣΗΣ ΠΛΗΡΟΦΟΡΙΑΣ ΚΑΙ ΤΕΧΝΟΛΟΓΙΑΣ ΥΛΙΚΩΝ

Συγκριτική μελέτη τεχνικών αποθήκευσης ροών δεδομένων και αλγορίθμων μηχανικής μάθησης για ανάλυση συναισθημάτων βασισμένη σε κείμενο

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

του

ΒΕΡΝΙΚΟΥ ΓΕΩΡΓΙΟΥ

Επιβλέπων : Ιάκωβος Βενιέρης
Καθηγητής Ε.Μ.Π.

Αθήνα, Σεπτέμβριος 2017

ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΤΟΜΕΑΣ ΣΥΣΤΗΜΑΤΩΝ ΜΕΤΑΔΟΣΗΣ ΠΛΗΡΟΦΟΡΙΑΣ ΚΑΙ ΤΕΧΝΟΛΟΓΙΑΣ ΥΛΙΚΩΝ

Συγκριτική μελέτη τεχνικών αποθήκευσης ροών δεδομένων και αλγορίθμων μηχανικής μάθησης για ανάλυση συναισθημάτων βασισμένη σε κείμενο

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

ΒΕΡΝΙΚΟΣ ΓΕΩΡΓΙΟΣ

Επιβλέπων : Ιάκωβος Βενιέρης
Καθηγητής Ε.Μ.Π.

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 26η Σεπτεμβρίου 2017.

(Υπογραφή)	*(Υπογραφή)*	*(Υπογραφή)*
.....	.....	.....
Ιάκωβος Βενιέρης Καθηγητής Ε.Μ.Π.	Δήμητρα-Θεοδώρα Κακλαμάνη Καθηγητής Ε.Μ.Π.	Γιώργος Ματσόπουλος Αναπληρωτής Καθηγητής Ε.Μ.Π

Αθήνα, Σεπτέμβριος 2017

(Υπογραφή)

.....

ΒΕΡΝΙΚΟΣ ΓΕΩΡΓΙΟΣ

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

Περίληψη

Αντικείμενο της παρούσας διπλωματικής εργασίας αποτελεί η εξόρυξη και επεξεργασία δεδομένων σε πραγματικό χρόνο με σκοπό την χρήσιμη εξαγωγή συμπερασμάτων για τα συναισθήματα του κοινού για ορισμένα θέματα . Τα δεδομένα τα οποία χρησιμοποιούνται προέρχονται εξ ολοκλήρου από κοινωνικά δίκτυα και συγκεκριμένα το Twitter το οποίο κάνει διαθέσιμο ένα μέρος των δεδομένων που υποβάλλονται στην πλατφόρμα του σε πραγματικό χρόνο. Τα δεδομένα μετά την εξόρυξή τους αποθηκεύονταν προσωρινά σε μια βάση δεδομένων έτσι ώστε να διευκολυνθεί η επεξεργασία τους ανά δέσμη δεδομένων (batch analysis) για τη συναισθηματική τους ανάλυση.

Για την επεξεργασία των δεδομένων χρησιμοποιήθηκαν δυο εργαλεία που διευκολύνουν και καθιστούν πολύ αποτελεσματικότερη τη διαχείριση και την επεξεργασία μεγάλου όγκου δεδομένων σε πραγματικό χρόνο : το Apache Storm και το Apache Spark.

Τέλος, για την ανάλυση συναισθήματος χρησιμοποιήθηκαν μέθοδοι μηχανικής μάθησης όπως ταξινομητές Naïve Bayes, Μηχανές Διανυσματικής Στήριξης καθώς και Λογιστική Παλινδρόμηση (Logistic Regression).

Λέξεις κλειδιά : Ανάλυση σε πραγματικό χρόνο, Εξόρυξη δεδομένων, Μηχανική Μάθηση, Εφαρμογή, Ανάλυση Συναισθήματος, Ανάλυση Κειμένου, Twitter, Apache Spark, Apache Storm

Abstract

Subject of this thesis is the real-time data mining and processing in order to draw meaningfull conclusions regarding the public opinion about particular topics. The amount of data used derives exclusively from social networks and specifically Twitter, that publicizes part of the data that are submitted to its platform in real-time. This data was temporarily stored in a database in order for its batch analysis, regarding its sentiment, to be facilitated.

For the data processing two tools were used that make the management and analysis of big data in real time faster and more effective : Apache Storm and Apache Spark.

Finally, for the sentiment analysis machine learning algorithms were used, such a Naïve Bayes classifiers, Support Vector Machines and Logistic Regression as well.

Keywords : Real time analysis, Data Mining, Machine Learning, Application, Sentiment Analysis, Text Analysis, Twitter, Apache Spark, Apache Storm

Ευχαριστίες

Αρχικά, θα ήθελα να ευχαριστήσω τον καθηγητή Ε.Μ.Π κύριο Ιάκωβο Βενιέρη για την εμπιστοσύνη που μου έδειξε και τη δυνατότητα που μου έδωσε να εκπονήσω τη διπλωματική μου εργασία σε ένα τόσο ενδιαφέρον και σύγχρονο θέμα σε ένα επιστημονικό αντικείμενο, το οποίο για εμένα προσωπικά παρουσιάζει ιδιαίτερο ερευνητικό ενδιαφέρον. Ακόμα, θέλω να τον ευχαριστήσω για τις συμβουλές του, όχι μόνο όσον αφορά τη διπλωματική εργασία αλλά και για τη μελλοντική επαγγελματική μου πορεία.

Στη συνέχεια θα ήθελα να ευχαριστήσω το διδακτορικό ερευνητή Μανόλη Καραμανή για τη βοήθεια που μου παρείχε όλο αυτό τον καιρό, για την άμεση επικοινωνία που είχαμε και τις καίριες συμβουλές του.

Τελειώνοντας, θα ήθελα να ευχαριστήσω τους φίλους μου εντός και εκτός σχολής για τις εμπειρίες που μοιραστήκαμε αυτά τα έξι χρόνια της φοίτησής μου στη σχολή καθώς και για τη στήριξή τους και ιδιαίτερα το Γιώργο και τη Χαρά, οι οποίοι με τις συμβουλές τους προσέφεραν βοήθεια στην εκπόνηση της παρούσας διπλωματικής. Τέλος, προφανής είναι η ευγνωμοσύνη μου στους ανθρώπους τις οικογένειάς μου για την αμέριστη στήριξη τους και την ανιδιοτελή προσφορά τους στο πρόσωπό μου όλα αυτά τα χρόνια.

Περιεχόμενα

Κατάλογος Σχημάτων

Κατάλογος Πινάκων

ΚΕΦΑΛΑΙΟ 1: ΕΙΣΑΓΩΓΗ

1.1 Γενικά

Η ραγδαία αύξηση της τεχνολογίας και η ανάγκη για καθημερινή επικοινωνία με τους γύρω μας έχει συμβάλει τα μέγιστα στην ανάπτυξη των κοινωνικών δικτύων. Δισεκατομμύρια άνθρωποι ανά τον κόσμο είναι συνδεδεμένοι ή συνομιλούν στο διαδίκτυο κάθε λεπτό δια μέσω αυτών. Χρήστες ανεξαρτήτου ηλικίας, φύλου και καταγωγής μοιράζονται μέσω της συνεχούς πρόσβασης και ενασχόλησης με το διαδίκτυο φωτογραφίες, γνώμες, συναισθήματα και αντιδρούν καθημερινά στα γεγονότα της επικαιρότητας,

Ο όγκος των δεδομένων που παράγονται καθημερινά μέσα από τα κοινωνικά δίκτυα είναι μεγάλος και πολλαπλάσιος σε σχέση με παλαιότερα χρόνια, ενώ προβλέπεται να συνεχίσει να αυξάνεται. Προκύπτει έτσι, άμεσα η ανάγκη διαχείρισης ενός μεγάλου όγκου δεδομένων που παράγονται καθημερινά και η χρησιμοποίησή τους με τρόπο τέτοιο ώστε να μπορεί να εξαχθεί χρήσιμη πληροφορία για τα τεκταινόμενα ανά τον κόσμο, καθώς και για τη γνώμη του κόσμου για τις εξελίξεις. Το Twitter, για παράδειγμα, αριθμεί αυτή την περίοδο 328 εκατομμύρια MAU ενεργούς λογαριασμούς (Monthly Active Users δηλαδή, για το Twitter, χρήστες που ακολουθούν τουλάχιστον 30 άτομα και ακολουθούνται από τουλάχιστον τον ένα τρίτο αυτών τον ατόμων - Active Users και έχουν εισέλθει στην πλατφόρμα του Twitter τουλάχιστον μια φορά τον τελευταίο μήνα) ενώ υπολογίζεται ότι κάθε μέρα υποβάλλονται 500 εκατομμύρια tweets [1].

Αυτό σε συνδυασμό με την ανάγκη για ανάλυση των δεδομένων όσο το δυνατόν ταχύτερα οδήγησε στην δημιουργία εργαλείων ανάλυσης μεγάλου όγκου δεδομένων (big data analysis) σε πραγματικό χρόνο για την επεξεργασία αυτής της πληροφορίας (real time processing) αποτελεσματικά και γρήγορα. Τέτοια εργαλεία χρησιμοποιούνται σε πολλούς τομείς της έρευνας και της βιομηχανίας, όπως στον τραπεζικό τομέα για τον έλεγχο οικονομικών συναλλαγών, στην ανίχνευση οικονομικής απάτης και τη δημιουργία στατιστικών μοντέλων για τη λήψη επενδυτικών αποφάσεων. Στον τομέα των επικοινωνιών και της διασκέδασης. για τη συλλογή δεδομένων των χρηστών με σκοπό την προσφορά ενδεδειγμένου περιεχομένου σε κάθε χρήστη ανάλογα με τις προτιμήσεις του, καθώς και τη μέτρηση

ικανοποίησης από τα προϊόντα τους αλλά και σε άλλους τομείς όπως στη βιολογία, στις μετακινήσεις, στην ενεργειακή διαχείριση κ.α.

Τα εργαλεία αυτά, μερικά από τα οποία θα αναλύσουμε εκτενώς παρακάτω, έχουν τη δυνατότητα να χρησιμοποιούν αποδοτικά τους πόρους που τους δίνονται, είτε «τρέχουν» σε ένα οικιακό πολυπύρηνο σύστημα ή σε μια συστάδα υπολογιστών που συνεργάζονται για την επεξεργασία των δεδομένων. Αυτό το επιτυγχάνουν με τεχνικές διαμοιρασμού είτε των δεδομένων ή των εργασιών μεταξύ των πυρήνων ή των υπολογιστικών μονάδων. Η απαίτηση για παραγωγή αποτελεσμάτων σε πραγματικό ή σχεδόν πραγματικό χρόνο (near real time processing) ωθεί αυτά τα εργαλεία στην εξέλιξη και τη διεύρυνση των δυνατοτήτων τους με αποτέλεσμα η πληροφορία να παράγεται μέσα σε λίγα λεπτά και να υπάρχει δυνατότητα να ανανεώνεται ανά τακτά χρονικά διαστήματα σύμφωνα με τα νέα δεδομένα που εισρέουν στο σύστημα.

Τέτοια, λοιπόν, εργαλεία είναι υπεύθυνα για την εξόρυξη πληροφορίας (data mining) από τις διάφορες πηγές, ανάλογα με το εκάστοτε πρόβλημα, και την επεξεργασία αυτών με σκοπό την απόσπαση του ουσιώδους κομματιού της πληροφορίας και του διαχωρισμού του από το επουσιώδες, καθώς και την παρουσίασή της σε μορφή κατανοητή από τον άνθρωπο.

Επίσης, τα ίδια ή αντίστοιχα εργαλεία είναι συνήθως επιφορτισμένα με το έργο της επεξεργασίας της εξαχθείσας πληροφορίας με μεθόδους στατιστικής ή μηχανικής μάθησης για την εξαγωγή χρήσιμων συμπερασμάτων μέσα από τα δεδομένα. Χωρίς πλέον να είναι απαραίτητη η προσπέλαση αυτού του τεράστιου όγκου δεδομένων οδηγούμαστε έτσι σε μια αυτοματοποιημένη διαδικασία εξαγωγής συμπερασμάτων.

Τέλος, με την ταχεία ανάπτυξη στον τομέα τεχνητής νοημοσύνης και της μηχανικής μάθησης που έχει παρατηρηθεί τα τελευταία χρόνια, σχεδιάζονται συνεχώς βελτιωμένοι αλγόριθμοι τόσο σε απόδοση όσο και σε πολυπλοκότητα. Αυτό έχεις ώς αποτέλεσμα την παραγωγή πιο γρήγορης και πιο αξιόπιστης πληροφορίας, με όλο και μειωμένη τη συμμετοχή του ανθρώπινου παράγοντα ως χειριστή και επιβλέποντα της όλης διαδικασίας εξαγωγής συμπερασμάτων.

1.2 Σκοπός

Σκοπός της παρούσας διπλωματικής εργασίας είναι η δημιουργία ενός συστήματος το οποίο θα συλλέγει σε πραγματικό χρόνο μεγάλο όγκο πληροφοριών από το Twitter και με βάση αυτές θα καθορίζει σε σχεδόν πραγματικό χρόνο το αίσθημα του κόσμου για το συγκεκριμένο θέμα ή είδηση. Συγκεκριμένα, θα συλλέγει πληροφορίες, θα τις προεπεξεργάζεται για να τις φέρει σε κατάλληλη μορφή για περαιτέρω επεξεργασία και θα τις αποθηκεύει σε μια βάση δεδομένων ως ενδιάμεσο σταθμό πριν την τελική τους επεξεργασία. Η επεξεργασία αυτή θα γίνεται ανά δέσμη δεδομένων, δηλαδή ανά τακτά χρονικά διαστήματα με σκοπό τη συγκέντρωση επαρκών δεδομένων.

Τα δεδομένα που θα εισέρχονται από το Twitter θα διοχετεύονται μέσω του λογισμικού διαχείρισης μεγάλου όγκου δεδομένων σε πραγματικό χρόνο Apache Storm, μετά από μια σχετική επεξεργασία τους, στη βάση δεδομένων Mongo DB. Μετά από αυτό το βήμα ένα άλλο εργαλείο επεξεργασίας μεγάλου όγκου δεδομένων, το Apache Spark θα αναλαμβάνει την ανά δέσμη (batch) ανάλυση των δεδομένων με μεθόδους μηχανικής μάθησης.

Αρχικά, στόχος είναι η δημιουργία ενός συστήματος το οποίο θα συλλέγει και θα επεξεργάζεται έναν όγκο δεδομένων σχετικών με κάποιον ή κάποιους όρους κλειδιά και στη συνέχεια θα αξιοποιεί αυτά τα δεδομένα ώστε να δίνει μια εικόνα για τη γνώμη του κοινού σε σχέση με αυτό τον όρο ή όρους.

Πιο συγκεκριμένα στο τελευταίο κομμάτι υλοποιείται μια ανάλυση συναισθήματος για τον όρο-κλειδί με αλγορίθμους μηχανικής μάθησης για κατηγοριοποίηση, με χρήση του εργαλείου Apache Spark, για την εξαγωγή συμπερασμάτων όσον αφορά την γνώμη του κόσμου σχετικά με τον εκάστοτε όρο στην πορεία του χρόνου.

Έτσι, υλοποιείται ένα εργαλείο το οποίο θα μπορεί σε σχεδόν πραγματικό χρόνο να διαχειριστεί μεγάλο όγκο δεδομένων και να παράξει μια μετρική για το πως αντιμετωπίζει ο κόσμος αυτόν τον όρο με την πάροδο του χρόνου, όντας μια διαδικασία δυναμική και όχι στατική, δίνοντας στον ενδιαφερόμενο μια σαφή εικόνα για την γνώμη του κόσμου πάνω στο καθορισμένο θέμα άμεσα και με ικανοποιητική ακρίβεια.

1.3 Οργάνωση Κειμένου

Στο κεφάλαιο 2 παρουσιάζονται οι τεχνολογίες που χρησιμοποιήθηκαν και γίνεται αναφορά στις λειτουργίες τους και στην αρχιτεκτονική τους. Αυτές οι τεχνολογίες συμπεριλαμβάνουν τον ιστότοπο κοινωνικής δικτύωσης Twitter, το διανεμημένο σύστημα υπολογισμού σε πραγματικό χρόνο Apache Storm, το εργαλείο ταχείας επεξεργασίας μεγάλου συνόλου δεδομένων Apache Storm, τη βάση δεδομένων MongoDB καθώς και μια σειρά από εργαλεία που χρησιμοποιήθηκαν στην υλοποίηση του συστήματος. Επίσης, στο ίδιο κεφάλαιο γίνεται μια σύγκριση των δυο εργαλείων επεξεργασίας μεγάλου συνόλου δεδομένων σε πραγματικό χρόνο, του Apache Spark και του Apache Storm. Παρουσιάζονται τα πλεονεκτήματα που έχει το καθένα καθώς και οι περιπτώσεις που ευνοούν την χρησιμοποίηση του ενός αντί του άλλου.

Στο κεφάλαιο 3 πραγματοποιείται μια εισαγωγή στη μηχανική μάθηση και στο πεδίο της ανάλυσης συναισθήματος. Γίνεται παρουσίαση και ανάλυση των επιμέρους στοιχείων των αλγορίθμων μηχανικής μάθησης που χρησιμοποιήθηκαν για την επεξεργασία των δεδομένων.

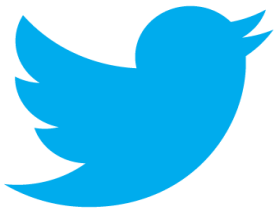
Στο κεφάλαιο 4 δίνεται μια αναλυτική περιγραφή της υλοποίησης του συστήματός μας για την ανάλυση συναισθήματος από την εισροή των δεδομένων μέχρι την εξαγωγή των αποτελεσμάτων. Επίσης παρουσιάζονται οι διάφορες μέθοδοι προεπεξεργασίας των δεδομένων για να είναι στην κατάλληλη μορφή για την εισαγωγή τους στους αλγορίθμους.

Στο κεφάλαιο 5 πραγματοποιείται μια ανασκόπηση των γνώσεων που αποκτήθηκαν από την εκπόνηση της παρούσας διπλωματικής, παρουσιάζονται τα τελικά αποτελέσματα και γίνεται ο σχολιασμός τους, καθώς επίσης παρουσιάζονται σκέψεις για επέκταση της παρούσας δουλειάς σε μελλοντικές εργασίες.

ΚΕΦΑΛΑΙΟ : 2 ΤΕΧΝΟΛΟΓΙΕΣ

2.1 Twitter

Το Twitter ,ως γνωστόν, είναι μια διαδικτυακή υπηρεσία ειδήσεων και κοινωνικής δικτύωσης που επιτρέπει στους χρήστες της να δημοσιεύουν και να αλληλεπιδρούν με μηνύματα “tweets” τα οποία περιλαμβάνουν το πολύ 140 χαρακτήρες.



Σχήμα 2.1: Twitter Logo

Μέσα από το Twitter οι χρήστες έχουν τη δυνατότητα να ενημερωθούν, να μοιραστούν τις ιδέες τους με άλλους ανθρώπους, να σχολιάσουν την επικαιρότητα και γενικότερα να εκφράσουν τα συναισθήματα τους και όλα αυτά σε ταχύτητα σκέψης. Κάθε ενημέρωση κατάστασης ή απλά “tweet” περιέχει εκτός από το ίδιο το κείμενο συνήθως δεδομένα όπως το hashtag(#), δηλαδή το ευρύτερο θέμα ή συζήτηση στην οποία αναφέρεται το κείμενο (ή απλά μια αναφορά), καθώς και το χρήστη στον οποίο απευθύνεται (αν απευθύνεται κάπου ή το χρήστη από τον οποίο προήλθε ως retweet), κάποια διεύθυνση στον ιστό (URL), ή και δεδομένα σχετικά με κάποιο μέρος στο οποίο μπορεί να αναφέρεται ή να συγγράφηκε το tweet.

Ο κάθε χρήστης επιλέγει ποιους χρήστες ή απλά λογαριασμούς θα επιλέξει να ακολουθήσει χωρίς να είναι απαραίτητο ότι θα ακολουθηθεί και από αυτούς. Αυτός ο τρόπος δόμησης του Twitter ξεπερνάει τα όρια ενός κοινωνικού δικτύου που στηρίζεται απλώς στην επικοινωνία μεταξύ χρηστών με στόχο το διαμοιρασμό των νέων τους για ψυχαγωγικούς σκοπούς. Για αυτό το λόγο τα τελευταία χρόνια έχει εξελιχθεί σε πηγή εκτάκτων και μη ειδήσεων, λόγω της περιεκτικότητας κάθε tweet, καθώς και εξαιτίας της ταχύτητας διάδοσής τους μέσα από το πλέγμα των ακολούθων κάθε χρήστη. Μπορεί να χαρακτηριστεί επομένως ως μια πλατφόρμα μικροϊστολογίου (microblogging) δηλαδή ως ένα μέσο όπου οι χρήστες έχουν τη δυνατότητα να ανταλλάσσουν πληροφορίες (κείμενο, εικόνες), περιορισμένες σε όγκο, στα πλαίσια μιας γενικότερης συζήτησης, ή οποιουδήποτε ενημερωτικού ή ψυχαγωγικού περιεχομένου.

Για αυτό το λόγο το twitter χρησιμοποιείται πλέον ως μέσο κοινωνικής δικτύωσης, ως ένα μέσο διαφήμισης για το προϊόντα και μέτρησης της αποδοχής τους από τις επιχειρήσεις καθώς και ως ένα μέσο για το σχολιασμό της επικαιρότητας από δημοσιογράφους, πρακτορεία ειδήσεων, πολιτικούς και απλούς χρήστες που εκφράζουν τη γνώμη τους. Αποτελεί έτσι μιας πρώτης τάξεως πηγή πληροφορίας για την εκτίμηση της κοινής γνώμης για οποιοδήποτε θέμα.

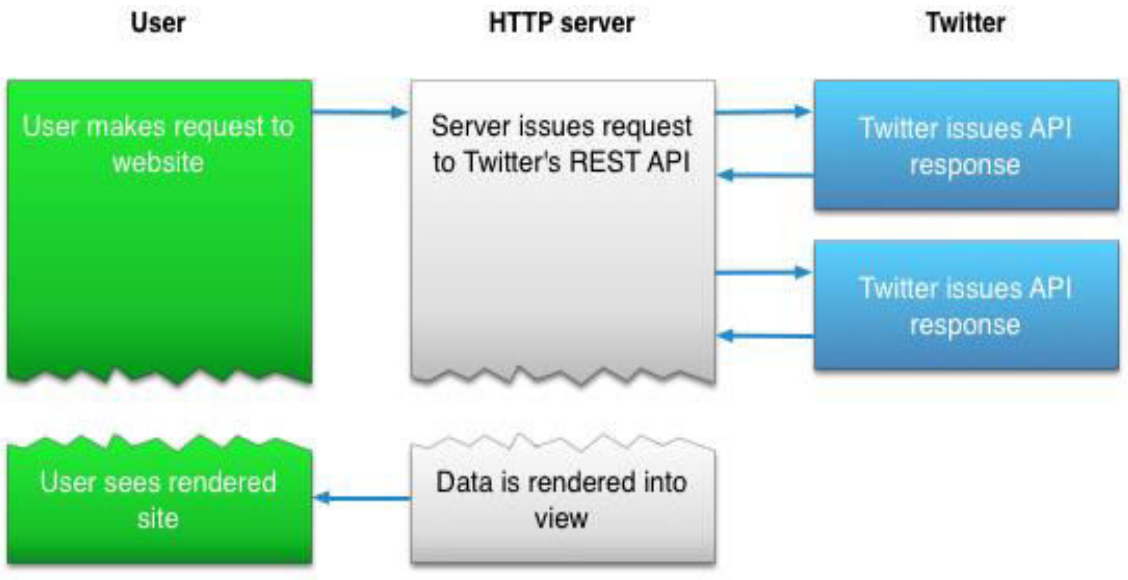
Κάθε χρήστης έχει τη δυνατότητα να δει τα tweets από τους χρήστες που ακολουθεί, ταξινομημένα χρονολογικά, τα οποία εμφανίζονται στην αρχική του σελίδα μόλις συνδεθεί στον προσωπικό του λογαριασμό. Ενώ για να δει τα tweet ενός συγκεκριμένου χρήστη, ταξινομημένα και αυτά χρονολογικά, πρέπει να εισέλθει στο δικό του προφίλ. Επίσης, το Twitter δίνει τη δυνατότητα στους χρήστες του να παρακολουθήσουν ποιά θέματα – συζητήσεις είναι πιο δημοφιλή εκείνη τη στιγμή (trending), διευκολύνοντάς τους στην προσπάθειά τους να ενημερωθούν και να πάρουν θέση για τα θέματα της επικαιρότητας.

Το Twitter κάνει διαθέσιμα, ένα μέρος των δημόσιων tweets που ρέουν καθημερινά σε πραγματικό χρόνο στην πλατφόρμα του, τα οποία μπορούν να αξιοποιηθούν με διάφορους τρόπους από μία εκτίμηση άποψης για ένα συγκεκριμένο θέμα μέχρι ανίχνευση ψευδών ειδήσεων. Με αυτό τον τρόπο, το Twitter αποτελεί μια πηγή μεγάλου όγκου, αξιόπιστων, όσον αφορά τη γενική γνώμη και σε πραγματικό χρόνο πληροφοριών για τους προγραμματιστές. Αυτό το κάνει με τη χρήση δύο Διεπαφών Προγραμματισμού Εφαρμογών (APIs) που διευκολύνουν τους προγραμματιστές να έχουν πρόσβαση στα δεδομένα και να αλληλεπιδρούν με τις υπηρεσίες του, χωρίς να έχουν πρόσβαση στον κώδικα που τις υλοποιεί: το REST API και το Streaming API.

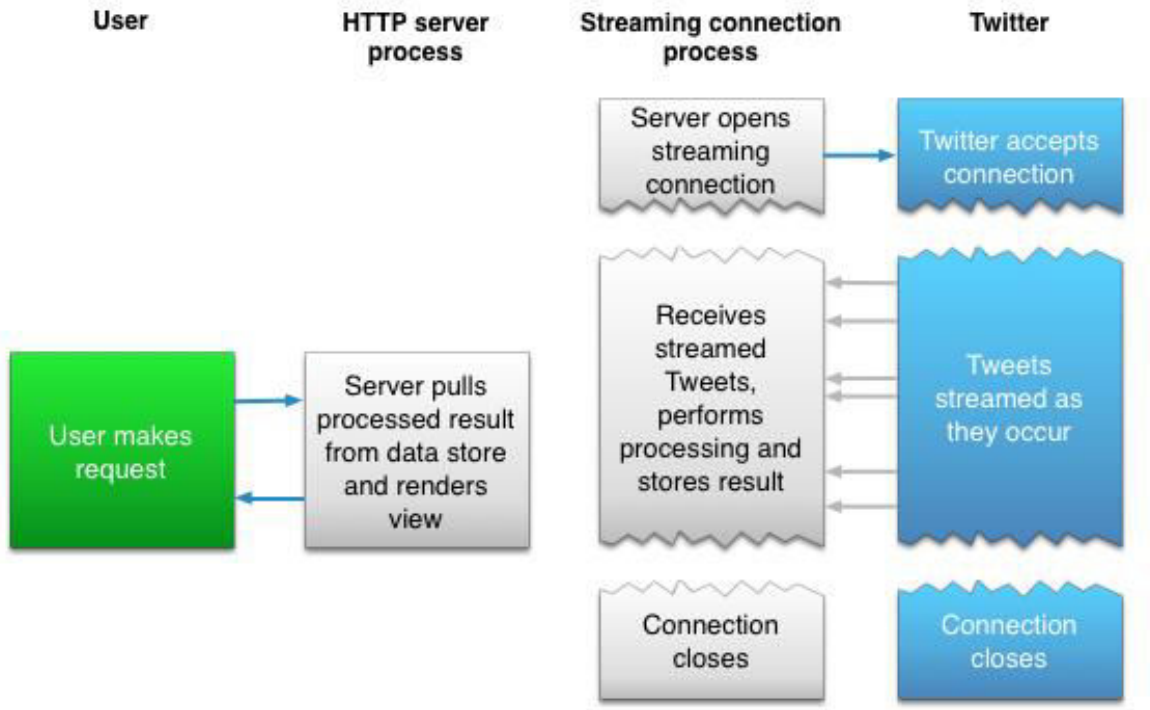
Το πρώτο επιτρέπει στον προγραμματιστή να αλληλεπιδράσει με την εφαρμογή διαβάζοντας και γράφοντας ασύγχρονα δεδομένα. Πιο συγκεκριμένα, με τη χρήση της διεπαφής αυτής έχει τη δυνατότητα να γράψει ένα νέο tweet, να διαβάσει το προφίλ ενός χρήστη και τα δεδομένα των ακολούθων του κ.α. Μέσω αυτού, μπορεί ο χρήστης να αναζητήσει tweets με συγκεκριμένα κριτήρια όπως συγκεκριμένο hashtag, τοποθεσία που γράφηκε, συγκεκριμένη γλώσσα κλπ., έχοντας όμως περιορισμό στον αριθμό των tweets που θα λάβει καθώς και στον αριθμό των αιτήσεων που μπορεί να κάνει ανά δεκαπεντάλεπτο.

Το δεύτερο, το οποίο χρησιμοποιούμε και στην παρούσα διπλωματική, επιτρέπει στον προγραμματιστή να έχει πρόσβαση με μικρή καθυστέρηση στη ροή των δεδομένων του Twitter. Έτσι, η υπηρεσία αυτή προωθεί άμεσα δεδομένα σχετικά με το τι συμβαίνει εκείνη την στιγμή στην πλατφόρμα του Twitter, τα οποία είναι όλα ή όσα ανταποκρίνονται σε κάποια κριτήρια αναζήτησης τα οποία έχει θέσει ο χρήστης. Ο αριθμός των tweets που λαμβάνει τελικά ο χρήστης ποικίλει ανάλογα με την εκάστοτε πολιτική του twitter, την υποδομή του για την εξυπηρέτηση πολλαπλών αιτήσεων και την κυκλοφοριακή κατάσταση εκείνη τη στιγμή. Παρ' όλα αυτά ο αριθμός αυτός υπολογίζεται γύρω στο 1% του πραγματικού όγκου των tweets εκείνη τη στιγμή.

Η ασφαλής επικοινωνία με το Twitter για τη χρήση των παραπάνω APIs εξασφαλίζεται μέσω του πρωτοκόλλου OAuth το οποίο δίνει στους χρήστες πρόσβαση εξ ονόματος ενός κατόχου πόρων. Ορίζει μια διαδικασία για τους κατόχους πόρων που επιτρέπει την πρόσβαση τρίτων στους πόρους του διακομιστή χωρίς να μοιράζονται τα διαπιστευτήρια τους. Έτσι, οι χρήστες έχουν πρόσβαση στη διεπαφή χωρίς να χρειάζεται να μοιραστούν τους κωδικούς τους Παρακάτω υπάρχουν δύο εικόνες που τονίζουν τη διαφορά ανάμεσα στον τρόπο που λειτουργούν τα δύο αυτά APIs [2] :



Σχήμα 2.2 : Rest API



Σχήμα 2.3 : Streaming API

2.2 Python

Ένα μεγάλο κομμάτι της εργασίας υλοποιήθηκε στη γλώσσα προγραμματισμού Python και περιλαμβάνει τόσο παραγωγή ιδίου κώδικα όσο χρήση μεθόδων διαθέσιμων πακέτων λογισμικού. Η python είναι μια υψηλού επιπέδου αντικειμενοστραφής γλώσσα προγραμματισμού, της οποίας κύρια χαρακτηριστικά είναι η ευκολία στην ανάγνωση κώδικα που έχει παραχθεί μέσω αυτής, η ομαλή καμπύλη εκμάθησης που προσφέρει καθώς και η ύπαρξη πληθώρας βιβλιοθηκών για τη διευκόλυνση αρκετών εργασιών του προγραμματισμού [3]. Επίσης, το συντακτικό της επιτρέπει στο χρήστη να εκφράσει προγραμματιστικά έννοιες που είναι κοντά στη σκέψη του με αποτέλεσμα να επιταχύνει τη συγγραφή προγραμμάτων καθώς και να οδηγεί σε μικρότερα σε όγκο προγράμματα σε σχέση με άλλες γλώσσες προγραμματισμού υψηλού επιπέδου. Για την υλοποίηση του κώδικα χρησιμοποιήθηκαν οι βιβλιοθήκες NLTK και PySpark. Το πακέτο NLTK [4] είναι ένα από τα πιο διαδεδομένα πακέτα επεξεργασίας φυσικής γλώσσας και προσφέρει βιβλιοθήκες για ταξινόμηση, κατηγοριοποίηση και επεξεργασία δεδομένων, ενώ η βιβλιοθήκη PySpark αποτελεί ουσιαστικά τη διεπαφή για την εκτέλεση και τον προγραμματισμό του Apache Spark μέσω της γλώσσας Python.

2.3 Java

Κατά την υλοποίηση του συστήματός μας έγινε χρήση και της γλώσσας προγραμματισμού Java, καθώς και πακέτων λογισμικού που προσφέρονται σε αυτή τη γλώσσα. Η Java αποτελεί μια αντικειμενοστραφή γλώσσα προγραμματισμού που ως κύριο χαρακτηριστικό της έχει τη μεταφερσιμότητα του κώδικά της, με αποτέλεσμα να μπορεί να τρέξει σε όλες τις πλατφόρμες ανεξαρτήτου λειτουργικού συστήματος. Για να συμβεί αυτό κάθε φορά που πρόκειται να εκτελεστεί ένας κώδικας Java, η εικονική μηχανή (JVM) αναλαμβάνει να μετατρέψει τον ενδιάμεσο κώδικα που προκύπτει από το μεταγλωττιστή της Java σε κώδικα μηχανής που να υποστηρίζεται από το λειτουργικό και τον επεξεργαστή. Αυτό το χαρακτηριστικό, αν και προσφέρει το πλεονέκτημα της φορητότητας, κάνει τη Java να υστερεί σε ταχύτητα σε σχέση με άλλες γλώσσες που με τη μεταγλώττιση τους παράγεται απευθείας κώδικας μηχανής. Στο πλαίσιο της παρούσας εργασίας χρησιμοποιήθηκε και η βιβλιοθήκη Twitter4j η οποία βοηθάει στην ενσωμάτωση των υπηρεσιών του Twitter (Rest API, Streaming API) σε οποιαδήποτε εφαρμογή σε Java ή οποιαδήποτε JVM γλώσσα [5]. Επίσης, χρησιμοποιήθηκε το Apache Maven το οποίο είναι ένα εργαλείο για το χτίσιμο και τη διαχείριση projects σε Java που χρησιμοποιήθηκε για την ενσωμάτωση των πακέτων του Storm κατά την δημιουργία των τοπολογιών του σε Java [6].

2.4 MongoDB

Η MongoDB είναι μια ανοιχτού κώδικα μη σχεσιακή βάση δεδομένων (NoSQL database) η οποία είναι γραμμένη σε C++. Είναι μια βάση προσανατολισμένη στην αποθήκευση, τη διαχείριση και ανάκτηση δεδομένων τύπου εγγράφου και προσφέρει υψηλή απόδοση, διαθεσιμότητα και επεκτασιμότητα. Η βάση αυτή (όπως κάθε NoSQL βάση) δεν χρησιμοποιεί κάποιο δομημένο σύστημα για τα στοιχεία που περιλαμβάνει, όπως πχ. πίνακες, ούτε χρησιμοποιεί κάποια Structured Query Language (SQL) για την διαχείριση των δεδομένων, αντιθέτως έχει τη δυνατότητα να αποθηκεύει και να ανακτά μεγάλο όγκο δεδομένων χωρίς να ενδιαφέρεται για τις σχέσεις που έχουν μεταξύ τους. Ακόμα, η MongoDB αποθηκεύει τα δεδομένα σε μορφή BSON, η οποία αποτελεί τη δυαδική αναπαράσταση της μορφής JSON (JavaScript Object Notation).

Σχήμα 2.4 : MongoDB Logo

Οι δύο βασικές έννοιες στην MongoDB είναι η συλλογή (collection) και το έγγραφο (document). Η συλλογή, όπως προδίδει και το όνομά της, αποτελεί ουσιαστικά μια συλλογή δεδομένων και είναι για τη MongoDB το αντίστοιχο του πίνακα (table) για τις σχεσιακές βάσεις (Relational Database Management System). Η συλλογή υπάρχει εντός της βάση δεδομένων και δεν επιβάλλει απαραίτητα κάποια δομή στα διάφορα έγγραφα που βρίσκονται σε αυτή, τα οποία μπορεί να έχουν διαφορετικά πεδία (σε αριθμό και δομή). Τυπικά, τα έγγραφα που βρίσκονται στην ίδια συλλογή έχουν κάποιο κοινό χαρακτηριστικό. Το έγγραφο είναι ένα σύνολο ζευγών μεταβλητών (ή πεδίων) με την ανάλογη τιμή. Όπως προαναφέραμε τα έγγραφα σε μία βάση δεν είναι απαραίτητο να έχουν την ίδια δομή, δηλαδή τα ίδια πλαίσια, καθώς επίσης υπάρχει η δυνατότητα σε κοινά πλαίσια να φιλοξενούνται διαφορετικοί τύποι μεταβλητής μεταξύ των εγγράφων, κάτι το οποίο δε συμβαίνει σε σχετικιστικές βάσεις.

Η βάση MongoDB λόγω της δομή της έχει το πλεονέκτημα ότι δίνει ελευθερία στο χρήστη στη δημιουργία και τη διαχείριση συλλογών καθώς δεν τον περιορίζει με την επιβολή κοινής δομής για τα περιεχόμενα έγγραφα. Επίσης, προσφέρει βαθιά αναζήτηση εγγράφων με τη χρήση μιας βασισμένης σε έγγραφα γλώσσας αναζήτησης η οποία είναι σχεδόν τόσο αποτελεσματική όσο και η SQL. Ακόμα, υποστηρίζει την αποθήκευση και τη διαχείριση μεγάλου όγκου δεδομένων. Επιπροσθέτως, η MongoDB εξασφαλίζει την υψηλή διαθεσιμότητα με τη χρήση αντιγράφων των συνόλων δεδομένων(replica sets) έτσι ώστε εάν κάποιο σετ δεδομένων χαθεί να μπορεί να ανακτηθεί εύκολα. Τέλος, η MongoDB είναι εύκολα επεκτάσιμη, γεγονός που την καθιστά ιδανική για την υλοποίηση συστημάτων όπως το δικό μας [7].

2.4 Apache Storm

2.4.1 Διαμοιρασμένος υπολογισμός

Πριν προχωρήσουμε στην ανάλυση των δύο εργαλείων Apache Spark και Apache Storm είναι ορθό να ορίσουμε τι εννοούμε κατανεμημένη ή διαμοιρασμένη επεξεργασία. Η διαμοιρασμένη επεξεργασία είναι η δυνατότητα της εκτέλεσης εφαρμογών σε πολλαπλές μηχανές – συστάδες, οι οποίες επικοινωνούν και συνεργάζονται μεταξύ τους για να μπορούν να τις εκτελέσουν με τρόπο αποτελεσματικό και γρήγορο, μειώνοντας έτσι δραματικά το χρόνο της εκτέλεσης προγραμμάτων. Η μέθοδος υπολογισμού αυτή προσφέρει αξιοπιστία (reliability), δεδομένου ότι εάν αστοχήσει ένας κόμβος (μηχάνημα) δεν αστοχεί όλο το σύστημα, επεκτασιμότητα (scalability), αυξάνοντας την επίδοση του συστήματος προσθέτοντας περισσότερα μηχανήματα και διαφάνεια (transparency) αποκρύπτοντας την πολυπλοκότητα του μοντέλου παίρνοντας τη μορφή (θεωρητικά) μιας ενιαίας οντότητας.

2.4.2 Γενικά

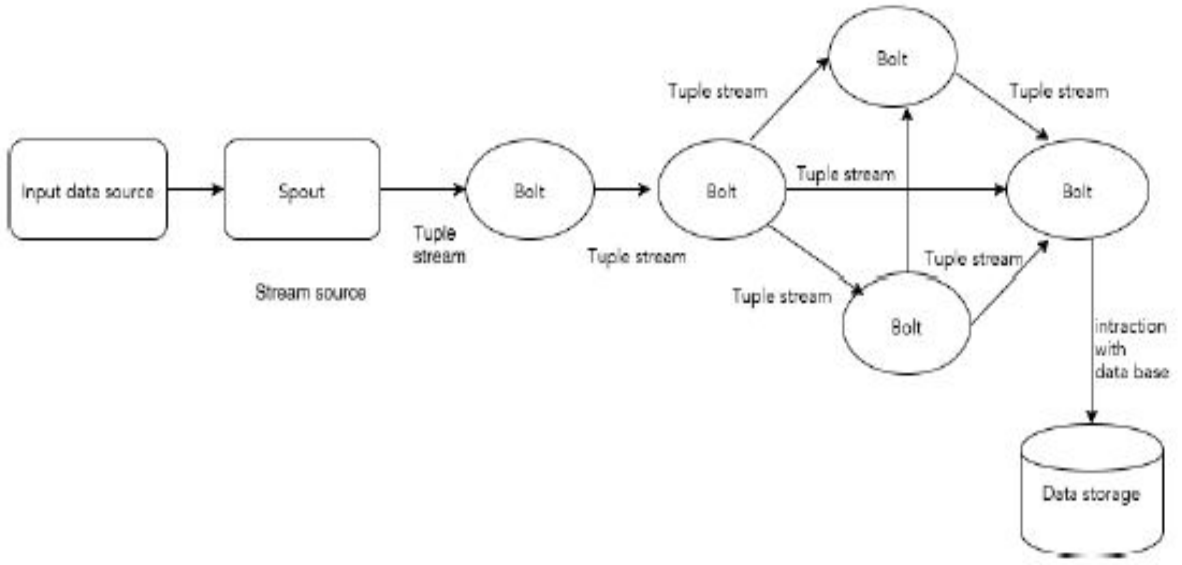
Το Apache Storm είναι ένα εργαλείο διαμοιρασμένης επεξεργασίας σε πραγματικό χρόνο, ανοιχτού κώδικα (open source) σε συστάδες υπολογιστών (clusters). Το Storm διευκολύνει την επεξεργασία απεριόριστου όγκου δεδομένων που ρέουν σε αυτό σε πραγματικό χρόνο, κάνοντας σε πραγματικό χρόνο ότι το Hadoop (ένα άλλο εργαλείο επεξεργασίας μεγάλου συνόλου δεδομένων) κάνει για ανά δέσμη επεξεργασία δεδομένων. Είναι γραμμένο στις γλώσσες προγραμματισμού Java και Clojure αν και παρέχει τη δυνατότητα προγραμματισμού σε περισσότερες γλώσσες.

Σχήμα 2.5 : Apache Storm Logo

Το Storm είναι έτσι σχεδιασμένο ώστε να επεξεργάζεται έναν αχανή όγκο δεδομένων με μέθοδο κλιμακωτή και ανεκτική σε σφάλματα (fault tolerant) που εγγυάται ότι τα δεδομένα θα επεξεργαστούν σίγουρα, κάνοντας παράλληλη επεξεργασία. Είναι εξαιρετικά γρήγορο έχοντας τη δυνατότητα να επεξεργαστεί ένα εκατομμύριο πλειάδες (tuples), που είναι και η δομική του μονάδα, ανά δευτερόλεπτο σε ένα κόμβο. Είναι πολύ διαδεδομένο όσον αφορά την επεξεργασία μεγάλου όγκου δεδομένων σε πραγματικό χρόνο και χρησιμοποιείται αρκετά στη βιομηχανία από εταιρίες όλων των ειδών, που χρειάζονται να προσπελάσουν πολλά δεδομένα σε μικρό χρόνο, όπως Yahoo, Spotify, The Weather Channel καθώς και το ίδιο το Twitter που μέχρι πριν από λίγα χρόνια (έως το 2015 όπου άρχισε να χρησιμοποιεί το Heron) το χρησιμοποιούσε ως ένα βασικό εργαλείο της υποδομής του.

2.4.3 Δομικά Στοιχεία

Το Storm διαβάζει μια ροή αδόμητων δεδομένων από το ένα άκρο και περνώντας την από μια ακολουθία επεξεργαστικών μονάδων και παράγει ως έξοδο τα επεξεργασμένα δεδομένα. Ο τρόπος με τον οποίο γίνεται αυτό είναι με τον ορισμό μιας τοπολογίας, δηλαδή ενός κατευθυνόμενου γράφου υπολογισμού όπου κάθε κόμβος αποτελεί μια λογική υπολογιστική μονάδα ενώ οι ακμές δείχνουν πως διαμοιράζονται τα δεδομένα από το ένα στάδιο στο άλλο. Η τοπολογία αυτή θα εκτελείται μέχρις ότου διακοπεί (killed) από το χρήστη, ενώ το Storm έχει τη δυνατότητα να εκτελεί παραπάνω από μια τοπολογίες ταυτόχρονα. Μια βασική τοπολογία που επιδεικνύει τα βασικά συστατικά στοιχεία και τις επεξεργαστικές μονάδες του εργαλείου τα οποία θα αναλύσουμε παρακάτω είναι η εξής:



Σχήμα 2.6 : Storm Topology

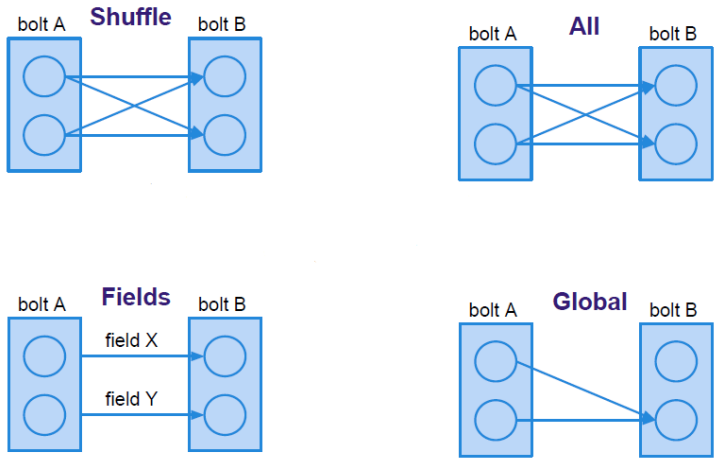
Σε αυτή τη φωτογραφία διακρίνονται τα στοιχεία του Storm τα οποία είναι χρήσιμα για την επεξεργασία : οι πλειάδες (Tuples), τα στόμια (Spouts), οι κεραυνοί (Bolts) και η ροή δεδομένων (Stream). Παρακάτω θα τα αναλύσουμε εκτενώς.

- **Tuples** : είναι η βασική δομή δεδομένων και αποτελείται από μια λίστα από διατεταγμένα στοιχεία. Μπορεί να είναι κάθε τύπος δεδομένων (αριθμός, λέξη) και στην πράξη αποτελείται από διαφορετικές τιμές που διαχωρίζονται με κόμμα η μια από την άλλη και «ρέουν» από τη μια επεξεργαστική μονάδα στην επόμενη.
- **Stream** : ουσιαστικά αποτελεί την αδόμητη ακολουθία από πλειάδες (tuples) που εισέρχονται στο Storm για επεξεργασία ή φιλτράρισμα.
- **Spout** : αποτελεί μια πηγή δεδομένων. Το Storm έχει τη δυνατότητα να συνδεθεί κατευθείαν σε αδόμητες πηγές ροής δεδομένων όπως είναι το Twitter. Όμως με τη χρήση των Spouts υπάρχει η δυνατότητα μιας ενδιάμεσης σύνδεσης μεταξύ της πηγής και των μονάδων επεξεργασίας, ώστε να γίνεται εκροή πιο συγκεκριμένης πληροφορίας και δομημένης σε μορφή που μπορεί να αξιοποιήσει καλύτερα το Storm.
- **Bolt** : αποτελεί μια λογική μονάδα επεξεργασίας. Τα Spouts εκπέμπουν δεδομένα (tuples) στα Bolts και αυτά αναλαμβάνουν την επεξεργασία δημιουργώντας μια νέα ροή δεδομένων στα επόμενα επίπεδα. Μπορούν να πραγματοποιήσουν λειτουργίες

όπως φιλτράρισμα, συνάθροιση (aggregation), συνένωση (joining) καθώς και αλληλεπίδραση με βάσεις και πηγές δεδομένων. Τέλος, η έξοδός τους μπορεί είτε να προωθηθεί εκ νέου σε ένα Bolt ή να εμφανιστεί στην συσκευή εξόδου ή να καταχωρηθεί σε κάποια βάση δεδομένων.

Μια βασική τοπολογία αποτελείται συνήθως από ένα Spout και ένα σύνολο από Bolts που δουλεύουν είτε ακολουθιακά ή παράλληλα Για να δουλέψει σωστά η εκάστοτε τοπολογία θα πρέπει η εκτέλεση αυτών των κόμβων να γίνει με τη σωστή σειρά. Η εκτέλεση αυτή της βασικής λογικής μονάδας που είτε είναι Bolt ή Spout ονομάζεται έργο (Task).

Τα δεδομένα μπορεί να μετακινηθούν από Spouts σε έναν αριθμό Bolts ή από κάποιο Bolt σε κάποιο άλλο. Ο τρόπος με τον οποίο γίνεται αυτή η ροή δεδομένων (tuples) ανάμεσα στις βασικές λογικές μονάδες ονομάζεται ομαδοποίηση ροής (stream grouping). Υπάρχουν τέσσερις διαφορετικοί τρόποι ομαδοποίησης ροής που διακρίνονται και στην εικόνα: **Τυχαία Ομαδοποίηση (Shuffle Grouping)** όπου οι πλειάδες (tuples) διαμοιράζονται τυχαία και ομοιόμορφα στα Bolts, **Ομαδοποίηση με βάση το Πεδίο (Fields Grouping)** όπου κάθε πλειάδα προωθείται σε συγκεκριμένο κόμβο (bolt) ανάλογα με την τιμή της σε κάποιο πεδίο έτσι ώστε πλειάδες με ίδια τιμή πεδίου να προωθούνται πάντα στον ίδιο κόμβο**, Καθολική Ομαδοποίηση (Global Grouping)** όπου όλες οι ροές δεδομένων προωθούνται σε έναν συγκεκριμένο Bolt και τέλος **Ολική Ομαδοποίηση (All Grouping)** όπου κάθε πλειάδα στέλνεται σε όλους τους διαθέσιμους κόμβους (bolts).



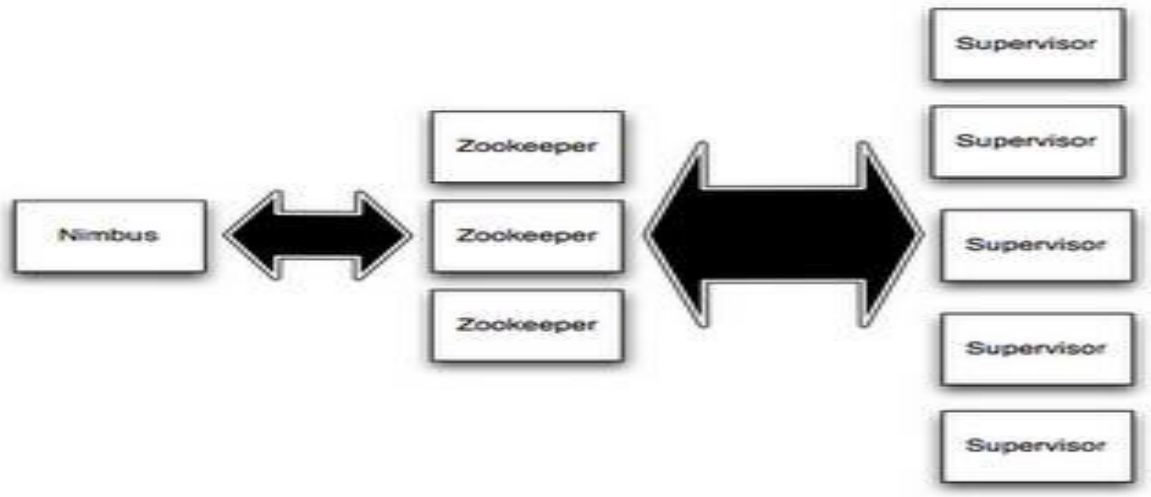
Σχήμα 2.7 : Είδη Grouping

2.4.4 Αρχιτεκτονική

Το Storm είναι φτιαγμένο για να «τρέχει» πάνω σε μια συστάδα από υπολογιστές, μπορεί όμως να εκτελεστεί και τοπικά σε έναν προσωπικό υπολογιστή πάνω σε JVM (Java Virtual Machine). Το Storm χρησιμοποιεί μια master/slave αρχιτεκτονική αποτελούμενη από

έναν κύριο κόμβο (master) και πολλούς κόμβους-εργάτες (worker nodes). Ο κύριος κόμβος που ονομάζεται Nimbus είναι υπεύθυνος για το διαμοιρασμό του κώδικα στη συστάδα, την ανάθεση έργων (tasks) στις υπολογιστικές μηχανές και στην επίβλεψη τυχών αστοχιών. Οι κόμβοι-εργάτες εκτελούν ένα υποσύνολο της εκάστοτε τοπολογίας η οποία αποτελείται από πολλές διεργασίες (worker processes) κατανεμημένες ανάμεσα στις μηχανές της συστάδας. Οι κόμβοι αυτοί τρέχουν μια διεργασία παρασκηνίου (daemon) που ονομάζεται “Supervisor” και έργο της είναι να παρακολουθεί το έργο που έχει ανατεθεί στον αντίστοιχο κόμβο που την καλεί και να δημιουργεί ή να διακόπτει εργασίες ανάλογα με το έργο που της έχει ανατεθεί από τον κύριο κόμβο.

Ο κόμβος Nimbus και κατ’ επέκταση το Storm είναι από τη φύση του «άστατο» (stateless), δηλαδή δεν αποθηκεύει την προσωρινή κατάστασή του. Παρ΄ όλο που αυτό φαινομενικά μοιάζει με μειονέκτημα, στην πράξη βοηθάει το Storm να επεξεργαστεί σε πραγματικό χρόνο τα δεδομένα με τον βέλτιστο και τάχιστο τρόπο. Για να παρακάμψει αυτή την «άστατη» φύση του χρησιμοποιεί την υπηρεσία Apache Zookeeper για τον συντονισμό μεταξύ του Nimbus κόμβου και των Workers κόμβων. Αυτή η υπηρεσία είναι υπεύθυνη για τον συντονισμό μεταξύ των διαφορετικών κόμβων μιας συστάδας εξασφαλίζοντας την ομαλή συνεργασία τους, διατηρώντας τα κοινά τους δεδομένα (αυτά που μοιράζονται μεταξύ τους) με εύρωστες τεχνικές συγχρονισμού. Έτσι, το Storm αποθηκεύει την κατάστασή του στο Zookeeper (ή στο φυσικό δίσκο του υπολογιστή εάν εκτελείται τοπικά) , δηλαδή την κατάσταση των κόμβων του κάθε στιγμή, έτσι ώστε εάν ένας κόμβος (ακόμα και ο Nimbus) αστοχήσει, να μπορεί να επανεκκινήσει και να συνεχίσει την επεξεργασία από το σημείο που την άφησε. Για την εκτέλεση του Storm σε συστάδα είναι λοιπόν απαραίτητο να υπάρχει ένας μηχάνημα για το κόμβο Nimbus, ένα ή περισσότερα μηχανήματα για τους Supervisors και ένα μηχάνημα για το Zookeeper για το συντονισμό τους [8].



Σχήμα 2.8 : Storm και Zookeeper

Έχοντας αναλύσει τη βασική δομή του Apache Storm μπορούμε να κάνουμε μια περιγραφή της ροής εργασιών (workflow) κατά τη χρήση του για την εκτέλεση ενός έργου:

- Αρχικά ο κόμβος Nimbus αναμένει την υποβολή σε αυτόν μίας τοπολογίας. Μόλις αυτή του ανατεθεί την επεξεργάζεται και τη χωρίζει σε επιμέρους εργασίες (tasks) οι οποίες πρέπει να εκτελεστούν.
- Διαμοιράζει ομοιόμορφα τις εργασίες στους διαθέσιμους Supervisors που τρέχουν στους κόμβους εργάτες (worker nodes). Αυτοί είναι υπεύθυνοι να στέλνουν ανά τακτά χρονικά διαστήματα έναν παλμό (heartbeat) στον κόμβο Nimbus ώστε να βεβαιωθεί ότι ο κάθε κόμβος λειτουργεί.
- Εάν ένας κόμβος στην πορεία πεθάνει και δε στείλει παλμό τότε ο κόμβος Nimbus αναθέτει τις εργασίες που είχε αναλάβει σε άλλο κόμβο.
- Εάν πεθάνει ο κόμβος Nimbus τότε οι κόμβοι-εργάτες θα συνεχίσουν μέχρι να φέρουν εις πέρας τις εργασίες που είχαν αναλάβει και όταν αυτό συμβεί θα αναζητήσουν νέες. Στο διάστημα αυτό το κόμβος Nimbus θα έχει επανεκκινήσει και η επεξεργασία θα συνεχιστεί. Με αυτό τον τρόπο το Storm εγγυάται ότι θα επεξεργαστεί όλα τα δεδομένα τουλάχιστον μία φορά [9].

Στο σημείο αυτό αξίζει να κάνουμε ιδιαίτερη μνεία στον τρόπο με τον οποίο το Storm υλοποιεί την παραλληλία. Όπως έχουμε αναφέρει κάθε τοπολογία αποτελείται από πολλές

διεργασίες (worker processes), ένα υποσύνολο των οποίων εκτελείται από κάθε κόμβο-εργάτη. Κάθε διεργασία από αυτές αποτελείται από επιμέρους threads (executors), καθένα από τα οποία τρέχει στον ίδιο κόμβο που τρέχει και η διεργασία. Κάθε τέτοιο thread υλοποιεί ένα ή περισσότερα έργα (tasks) ίδιου τύπου (Spouts ή Bolts) τα οποία εκτελούνται ακολουθιακά μέσα στο thread. Το έργο όπως έχουμε ήδη αναφέρει είναι η βασική λογική μονάδα και αποτελείται από ένα Spout ή Boit. Κάθε τέτοια μονάδα που ορίζουμε μέσα στο πρόγραμμα αντιστοιχεί σε ένα ακριβώς έργο (Task) και ο αριθμός τους παραμένει σταθερός όσο εκτελείται η τοπολογία σε αντίθεση με τον αριθμό των executors που μπορεί να αλλάξει, ενώ η εξ ορισμού ρύθμιση αντιστοιχεί σε ένα έργο ανά thread. Έτσι, για να αυξήσουμε την παραλληλία των εργασιών μας μπορούμε να αυξήσουμε τα threads (executors), έτσι ώστε ένα έργο να τρέχει παράλληλα σε παραπάνω threads, να αυξήσουμε τα tasks, υλοποιώντας παραπάνω από ένα ανά thread και τέλος να αυξήσουμε τις διεργασίες (worker processes) σε συνδυασμό με τις παραπάνω επιλογές ώστε να εκμεταλλευτούμε στο μέγιστο τις δυνατότητες των μηχανημάτων μας.

2.5 Apache Spark

2.5.1 Γενικά

Το Apache Spark είναι ένα ανοιχτού κώδικα εργαλείο διαμοιρασμένου υπολογισμού μεγάλου όγκου δεδομένων το οποίο έχει ως βασικά του χαρακτηριστικά την ταχύτητα, την ευκολία στη χρήση και τη δυνατότητα εκτενούς ανάλυσης των δεδομένων. Είναι και αυτό βασισμένο ,όπως και το Storm, στο Hadoop, όσον αφορά το διαμοιρασμό και την επεξεργασία των δεδομένων. Το βασικό του χαρακτηριστικό είναι ο τρόπος που χρησιμοποιεί τη μνήμη για να αποθηκεύσει τα δεδομένα και τα ενδιάμεσα βήματα της επεξεργασίας του, γεγονός που το καθιστά πολύ πιο αποτελεσματικό από το Hadoop, ειδικά σε επαναληπτικούς αλγορίθμους. Επίσης το Spark προσφέρει ένα σύνολο εργαλείων τα οποία δίνουν πολλές δυνατότητες όσον αφορά την πηγή των δεδομένων (ανάλυση ανά δέσμη αλλά και σε πραγματικό χρόνο), καθώς και τον τρόπο επεξεργασίας τους. Είναι γραμμένο στην προγραμματιστική γλώσσα Scala αλλά μπορεί να υποστηρίξει και άλλες γλώσσες προγραμματισμού όπως Python, Java, Clojure και R [10].

Σχήμα 2.9 : Apache Spark Logo

Το Apache Spark είναι ένα σύστημα υπολογισμού μεγάλου όγκου δεδομένων το οποίο αποτελείται από τέσσερα επιμέρους εργαλεία:

- **Apache Spark Core** : είναι η βάση του Apache Spark και ουσιαστικά ο πυρήνας πάνω στον οποίο βασίζονται όλα τα υπόλοιπα εργαλεία, προσφέροντας βασικές λειτουργίες όπως in-memory computing (δηλαδή η χρήση της μνήμης και όχι του δίσκου για την αποθήκευση δεδομένων) και τη δυνατότητα αναφοράς σε εξωτερικά συστήματα αποθήκευσης (πχ βάσεις).
- **Spark SQL** : είναι ένα εργαλείο που λειτουργεί πάνω από τον πυρήνα και αφορά δομημένους τύπους δεδομένων. Δίνει τη δυνατότητα στο χρήστη να αλληλεπιδράσει με σύνολα δεδομένων πολλών μορφών (JSON, Parquet, CSV, βάσεις δεδομένων) επιτρέποντάς του να τα φορτώσει, να κάνει αναζητήσεις και γενικά να τα επεξεργαστεί άμεσα.
- **Spark Streaming :** το εργαλείο του Spark για την επεξεργασία δεδομένων που ρέουν σε πραγματικό χρόνο. Αξιοποιεί τη δυνατότητα του Spark να επεξεργαστεί δεδομένα με πολύ μεγάλη ταχύτητα, τα οποία δεδομένα εισρέουν στο σύστημά σε μικρές δέσμες (RDDs).
- **Spark MLib :** είναι μια διεπαφή μηχανικής μάθησης διαμοιρασμένου υπολογισμού που λειτουργεί πάνω στην αρχιτεκτονική διαμοιρασμένης μνήμης του Spark. Ουσιαστικά είναι μια βιβλιοθήκη που περιλαμβάνει τους κοινούς αλγορίθμους μηχανικής μάθησης για ταξινόμηση, ομαδοποίηση, φιλτράρισμα και προεπεξεργασία σε δεδομένα τα οποία είναι κατανεμημένα στους διάφορους πόρους του εκάστοτε συστήματος.

- **Spark GraphX :** αποτελεί ένα εργαλείο για την μοντελοποίηση και επεξεργασία γράφων μέσω του διαμοιρασμένου υπολογισμού του Spark.

2.5.2 Δομικά Στοιχεία

Η βασική δομή δεδομένων του Spark ονομάζεται RDD (Resilient Distributed Dataset) και αποτελεί μια αμετάβλητη συλλογή αντικειμένων στα οποία μπορούν να εφαρμοστούν οι διάφορες λειτουργίες εν παραλλήλω. Κάθε σετ δεδομένων RDD διαιρείται σε κομμάτια (partitions) τα οποία μπορεί να βρίσκονται και να επεξεργάζονται σε διαφορετικούς κόμβους και μπορεί να αποτελείται από κάθε είδους αντικείμενα των γνωστών προγραμματιστικών γλωσσών. Η συλλογή αυτή δεδομένων (RDD) είναι αμετάβλητη και μόνο για ανάγνωση (read only). Αυτό συνεπάγεται ότι μια τέτοια διαμοιρασμένη συλλογή δημιουργείται είτε μετασχηματίζοντας δεδομένα που είναι αποθηκευμένα στο δίσκο, ή αλλού, σε μορφή RDD παραλληλλίζοντάς τα, δηλαδή μοιράζοντάς τα στους κόμβους ή μετασχηματίζοντας ένα ήδη υπάρχων RDD. Τα RDDs είναι αμετάβλητα όπως είπαμε και αυτό συνεπάγεται ότι κάθε μετασχηματισμός ενός υπάρχων RDD δημιουργεί ένα καινούργιο, αφήνοντας το αρχικό αναλλοίωτο. Επίσης λόγω της αμετάβλητης φύσης τους, τα RDDs είναι ανεκτικά στα σφάλματα μιας και σε κάθε πιθανή αστοχία υπάρχει η δυνατότητα να επαναληφθεί η διαδικασία για τη δημιουργία του από την αρχή με την διαδοχική εφαρμογή των ενεργειών που εφαρμόστηκαν στο αρχικό RDD και έτσι να επανακτηθεί η πληροφορία.

Τα RDDs υποστηρίζουν δυο διαφορετικούς τύπους λειτουργιών τις δράσεις (actions) και τους μετασχηματισμούς (transformations). Οι μετασχηματισμοί (transformations) εφαρμόζονται επί των RDDs και επιστρέφουν ένα νέο RDD (map, filter, aggregate) ενώ οι δράσεις επιστρέφουν μια τιμή έπειτα από κάποιο υπολογισμό επί του RDD (collect, count, take). Οι μετασχηματισμοί στο Spark είναι «οκνηροί» (lazy) με την έννοια ότι δεν υπολογίζονται ακριβώς μόλις εφαρμόζονται αλλά αντ' αυτού το ιστορικό της εφαρμογής τους αποθηκεύεται σε ένα αρχείο και οι ίδιοι υπολογίζονται μόνο όταν κάποια δράση απαιτήσει την επιστροφή κάποιου αποτελέσματος στο πρόγραμμα. Αυτό επιτρέπει στο Spark να λειτουργεί πιο αποδοτικά καθώς επίσης εξασφαλίζει και την ανοχή του στα σφάλματα δίνοντας τη δυνατότητα να επανακτηθεί ένα RDD ακολουθώντας όλους τους μετασχηματισμούς που έχουν εφαρμοστεί σε αυτό με βάση το ιστορικό που βρίσκεται στο αρχείο [11].

Το Spark υποστηρίζει δύο ακόμα δομές δεδομένων, τα Dataframes και τα Datasets. Τα Dataframes είναι επίσης μια αμετάβλητη συλλογή διαμοιρασμένων δεδομένων, τα οποία όμως είναι οργανωμένα με σε στήλες, όπως ένας πίνακας από μια συσχετιστική βάση. Είναι σχεδιασμένα για την γρήγορη διαχείριση και επεξεργασία μεγάλου όγκου κατανεμημένων δεδομένων. Τα Datasets είναι μια δομή δεδομένων που εισάχθηκε πρόσφατα στο Apache Spark και αποτελεί μια μείξη των δύο προηγούμενων τύπων που στόχο έχει τη χρήση ομοιόμορφων βιβλιοθηκών για όλους τους τύπους εργασιών στο Spark, αν και ακόμα δεν προσφέρεται σε όλες τις γλώσσες που υποστηρίζει το Spark [12].

2.5.3 Αρχιτεκτονική

Τέλος, ο τρόπος με τον οποίο διαμορφώνεται η αρχιτεκτονική του Spark όταν τρέχει σε μια συστάδα υπολογιστών είναι παρόμοιος με αυτόν του Storm. Οι εφαρμογές του Spark εκτελούνται ως ανεξάρτητες διεργασίες στη συστάδα και συντονίζονται από το αντικείμενο (object) SparkContext το οποίο ορίζεται στο κυρίως πρόγραμμα. Πιο συγκεκριμένα, το Spark συνδέεται με τη συστάδα και μόλις λάβει τους πόρους από τους worker κόμβους στέλνει σε αυτούς τον κώδικα της εκάστοτε εφαρμογής, η οποία αντιστοιχεί σε πολλές διεργασίες executors, σε tasks. Κάθε εφαρμογή έχει τις δικές της διεργασίες executors και αυτό επιτρέπει στις εφαρμογές να εκτελούνται απομονωμένα η μία από την άλλη δυσκολεύοντας όμως έτσι το διαμοιρασμό δεδομένων μεταξύ τους.

2.6 Σύγκριση μεταξύ Spark και Storm

2.6.1 Εισαγωγή

Το Spark και το Storm αποτελούν δύο διαφορετικά προγράμματα ανοιχτού κώδικα διαμοιρασμένου υπολογισμού για μεγάλα σύνολα δεδομένων, με το Storm να είναι το παλαιότερο στον τομέα από τα δύο. Πρέπει σε αυτό το σημείο να τονιστεί ότι σκοπός αυτού του κεφαλαίου δεν είναι να καταλήξουμε στην υπεροχή του ενός επί του άλλου. Αντιθέτως, στόχος είναι να επισημανθούν και να αποσαφηνιστούν οι ομοιότητες και οι διαφορές που έχουν τα δύο αυτά εργαλεία, έτσι ώστε να προβληθούν οι περιπτώσεις και οι εφαρμογές που θα ευνοήσουν τη χρησιμοποίηση το ενός αντί του άλλου.

Βασικά, το Spark όπως έχουμε αναφέρει αποτελείται ουσιαστικά από μια ομάδα εργαλείων που εκτελούν πολύ διαφορετικές λειτουργίες το καθένα, οπότε θα ήταν άτοπο να τεθεί σε ένα-προς-ένα σύγκριση με το Storm. Το πιο σωστό και το πιο δόκιμο είναι να συγκρίνουμε το Storm με το αντίστοιχο εργαλείο του Spark το οποίο αναλαμβάνει την επεξεργασία δεδομένων σε πραγματικό χρόνο, το Spark Streaming, κάνοντας έτσι αντικείμενο του κεφαλαίου τη σύγκριση του Storm με το Spark Streaming.

2.6.2 Βασικά χαρακτηριστικά

Τα δύο αυτά εργαλεία έχουν διαφορετικές προσεγγίσεις όσον αφορά τον τρόπο με τον οποίο επιτυγχάνουν υψηλές αποδόσεις και κυρίως τον τρόπο με τον οποίο υλοποιούν την παράλληλη υλοποίηση των εφαρμογών τους. Το Apache Storm χρησιμοποιεί όπως έχουμε αναφέρει παραλληλισμό εργασιών μοιράζοντας τα επιμέρους κομμάτια του κώδικα (spouts, bolts) στους διάφορους κόμβους και υλοποιώντας παραπάνω από μια λογικές μονάδες για την ίδια δουλειά. Αντιθέτως το Spark Streaming υλοποιεί παραλληλισμό σε επίπεδο δεδομένων μοιράζοντας τα δεδομένα σε κομμάτια (partitions), έτσι ώστε πολλοί κόμβοι να εκτελούν την ίδια διεργασία σε διαφορετικά υποσύνολα των δεδομένων.

Επίσης, βασική μονάδα δεδομένων στο Storm είναι οι πλειάδες (tuples) στις οποίες εφαρμόζονται όλες οι συναρτήσεις, ενώ το Spark Streaming λειτουργεί με τα RDDs στα οποία όπως έχουμε αναφέρει εφαρμόζονται οι δράσεις και οι μετασχηματισμοί, καθώς επίσης προσφέρεται και η δυνατότητα εγγραφής των αποτελεσμάτων σε εξωτερικές πηγές με δεδομένες συναρτήσεις.

Τέλος, τα δύο αυτά εργαλεία είναι γραμμένα σε γλώσσες βασισμένες σε JVM, το Spark σε Scala και το Storm σε Clojure. Η Scala αποτελεί μια μείξη συναρτησιακής και αντικειμενοστραφούς γλώσσας προγραμματισμού, ενώ η Clojure, η οποία είναι μια διάλεκτος της γλώσσας προγραμματισμού Lisp, είναι γενικού σκοπού γλώσσα με έμφαση στο συναρτησιακό προγραμματισμό. Παρ’ όλα αυτά και τα δύο εργαλεία υποστηρίζουν μερικώς ή ολικώς τη χρήση άλλων γλωσσών για τον προγραμματισμό τους. Σε αυτό το σημείο αξίζει να αναφέρουμε ότι η δημιουργία μιας εφαρμογής στο Spark είναι πιο κοντά στην μορφή ενός τυπικού προγράμματος απ’ ότι στο Storm το οποίο απαιτεί κάποια περίοδο εξοικείωσης στη σύνταξη των τοπολογιών καθώς και των Bolts, Spouts. Θα μπορούσαμε δηλαδή να πούμε ότι η

δημιουργία μιας εφαρμογής στο Spark είναι πιο εύκολη για τον προγραμματιστή καθώς δεν απέχει πολύ από τη συγγραφή ενός προγράμματος σε μια οποιαδήποτε γλώσσα προγραμματισμού, σε αντίθεση με το Storm που έχει ένα σαφώς πιο πολύπλοκο μοντέλο [13].

2.6.3 Απόδοση

Αρχικά, το Spark Streaming εκτελεί near real time event processing που σημαίνει ότι δεν επεξεργάζεται κάθε δεδομένο μόλις αυτό εισέρχεται στο δίαυλο, αλλά αντ' αυτού εκτελεί micro-batch processing (επεξεργασία ανά μικρή δέσμη) υλοποιώντας μικρά σύνολα δεδομένων τα οποία και επεξεργάζεται γρήγορα. Αντιθέτως, το Storm που επεξεργάζεται κάθε δεδομένο με το που ρέει στο δίαυλο (one at a time). Αυτό έχει ως αποτέλεσμα το Storm να εμφανίζει καθυστέρηση μικρότερη του δευτερολέπτου και της τάξης των milliseconds [14], ενώ το Spark Streaming εμφανίζει καθυστέρηση της τάξης των 1-2 δευτερολέπτων κάνοντας το ακατάλληλο σε εφαρμογές όπου και η ελάχιστη καθυστέρηση (latency) παίζει ρόλο.

Μπορεί το Storm να έχει μικρότερη καθυστέρηση στην επεξεργασία του κάθε δεδομένου που εισέρχεται στο δίαυλο (latency), αλλά το Spark επιτυγχάνει μεγαλύτερη απόδοση όσον αφορά την επεξεργασία ενός όγκου δεδομένων ανά μονάδα χρόνου. Αυτό το πετυχαίνει εκμεταλλευοντας το διαμοιρασμό των δεδομένων, γεγονός που ευνοεί την εισροή τεράστιου όγκου δεδομένων από διάφορες πηγές. Επίσης, η το Spark χρησιμοποιεί δυναμική παραχώρηση εργασιών στους κόμβους του, έτσι ώστε η επεξεργασία αυτών των μικρών δεσμών δεδομένων να γίνεται με βάση την τοπικότητα των δεδομένων και το φόρτο εργασίας κάθε κόμβου δημιουργώντας έτσι ένα μοντέλο εξισορρόπησης φόρτου εργασίας (load balancing) μεταξύ των κόμβων [15].

Ένα σημαντικό σημείο αναφοράς όταν ασχολούμαστε με εργαλεία που τρέχουν σε συστάδες υπολογιστών και επεξεργάζονται μεγάλο όγκο δεδομένων είναι η ανοχή στα σφάλματα. Το Storm για να μπορεί να έχει μεγάλη ανοχή στα σφάλματα πρέπει να μπορεί να παρακολουθεί κάθε δεδομένο που εισρέει σε αυτό και διαθέτει 3 τρόπους εγγύησης επεξεργασίας μηνύματος (message processing guarantee, ένα σύστημα που χρησιμοποιείται κατά κόρον στο διαμοιρασμένο υπολογισμό για την επικοινωνία και την αποστολή αιτήσεων επεξεργασίας μεταξύ των κόμβων) : τουλάχιστον μια φορά, το πολύ μια φορά και ακριβώς μια φορά. Το Storm, όπως έχουμε αναφέρει φροντίζει έτσι ώστε εάν κάποιος worker κόμβος

αστοχήσει να αναθέσει όλα τα tasks για το οποία ήταν υπεύθυνος σε έναν άλλο (μέσω του εργαλείου που είναι υπεύθυνο για την αποθήκευση της κατάστασης του Storm, του Zookeeper) κατευθύνοντας σε αυτόν πλέον τις αντίστοιχες πλειάδες δεδομένων, με αποτέλεσμα να μπορεί να ανακάμψει πολύ γρήγορα από μια αστοχία σε οποιοδήποτε κόμβο. Βέβαια, υπάρχει η πιθανότητα μέσω αυτής της μεθόδου να εμφανιστούν αντίγραφα δεδομένων καθώς μια πλειάδα (tuple) μπορεί να υποστεί επεξεργασία δύο φορές κατά την αστοχία ενός κόμβου (node failure). Το Spark από την άλλη στην περίπτωση που κάποιος worker κόμβος αστοχήσει το σύστημα συνεχίζει εκ νέου τον υπολογισμό από τα δεδομένα εισόδου, που όπως έχουμε τονίσει παραμένουν αμετάβλητα (RDDs), ενώ αν ο master κόμβος αστοχήσει αποτελεί Single Point Of Failure (μοναδικό σημείο αποτυχίας). Για να αποφευχθεί αυτή η αστοχία και η εφαρμογή να επανεκκινήσει θα πρέπει να χρησιμοποιηθούν ενδιάμεσα checkpoints για τα δεδομένα ώστε να υπάρξει συνοχή στην επεξεργασία. Συμπερασματικά, το σύστημα του Storm είναι πιο εύρωστο όσον αφορά το χειρισμό των σφαλμάτων, ελαχιστοποιώντας την πιθανότητα απώλειας δεδομένων, το οποίο όμως μπορεί να οδηγήσει στην επεξεργασία κάποιων δεδομένων παραπάνω από μία φορά και στην εμφάνιση αντιγράφων (duplicates), δηλαδή κάποιων διπλών αποτελεσμάτων στο τελικό στάδιο [16].

2.6.4 Συμπεράσματα

Εν κατακλείδι όπως αναφέραμε και στον πρόλογο σκοπός του κεφαλαίου δεν ήταν να συμπεράνουμε πιο εργαλείο είναι καλύτερο παρά μόνο να εξηγηθούν οι ομοιότητες και οι διαφορές τους. Το συμπέρασμα από αυτή την ανάλυση είναι ότι οι απαιτήσεις της εφαρμογής και του προγραμματιστή είναι αυτές που θα καθορίσουν τελικά την επιλογή του καταλληλότερου εργαλείου. Εάν η εφαρμογή έχει μεγάλες απαιτήσεις στην καθυστέρηση (latency) και απαιτεί γρήγορα κάθε αποτέλεσμα σε ροές δεδομένων (stream processing), το Storm θα προτιμηθεί έναντι του Spark το οποίο είναι ιδανικό όταν μας ενδιαφέρει περισσότερο ο υψηλός ρυθμός διεκπεραίωσης (throughput). Επίσης, θα πρέπει να ληφθεί υπόψιν η σημασία της πιθανής απώλειας δεδομένων από κάποια αστοχία, όπου μάλλον υπερέχει το Storm ή εάν θα υπάρξει πρόβλημα από κάποια διπλά αποτελέσματα. Ακόμα, δε μπορούμε να μιλήσουμε με σιγουριά για την απόδοση του ενός έναντι του άλλου με αριθμούς, μιας και η απόδοση είναι ζήτημα σωστής ρύθμισης και διαμοιρασμού τον πόρων για κάθε

εφαρμογή. Δεν πρέπει να ξεχνάμε ότι το Spark, σαν εργαλείο, προσφέρει περισσότερες επιλογές και δυνατότητες πέρα από το Spark Streaming, κάνοντας το κατάλληλο για μεγαλύτερη γκάμα εφαρμογών (batch processing, stream processing, interactive processing) με τη χρήση πληθώρας έτοιμων συναρτήσεων των βιβλιοθηκών του και ιδανικό για εφαρμογές όπου χρειάζεται να εκτελεστούν επαναληπτικοί αλγόριθμοι σε μεγάλο όγκο δεδομένων, όπως πχ. στην εκπαίδευση μοντέλων μηχανικής μάθησης. Συμπερασματικά το Storm και το Spark αποτελούν δύο πολύ ισχυρά και χρήσιμα εργαλεία όσων αφορά την επεξεργασία μεγάλου όγκου δεδομένων με μεγάλη ταχύτητα [17] [18].

ΚΕΦΑΛΑΙΟ 3: ΘΕΩΡΗΤΙΚΟ ΥΠΟΒΑΘΡΟ

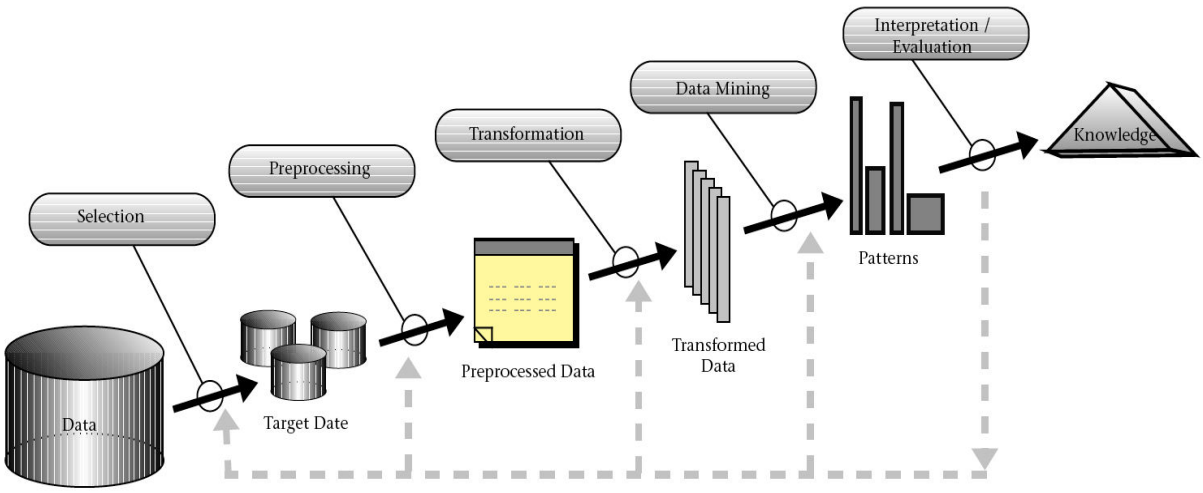
3.1 Εξόρυξη Δεδομένων

Ο όλο και αυξανόμενος όγκος πληροφοριών που συγκεντρώνονται σε επιχειρήσεις και οργανισμούς και η ανάγκη αξιοποίησής τους για την εξαγωγή γνώσης έχει οδηγήσει στην μεγάλη ανάπτυξη το τομέα της εξόρυξης δεδομένων. Πριν προχωρήσουμε όμως στη ανάλυση αυτού του βασικού σταδίου επεξεργασίας δεδομένων, καλό θα ήταν να αποσαφηνίσουμε τον όρο «Εξόρυξη Δεδομένων». Εξόρυξη δεδομένων (ή ανακάλυψη γνώσης από βάσεις δεδομένων) είναι η εξεύρεση μιας (ενδιαφέρουσας, αυτονόητης, μη προφανούς και πιθανόν χρήσιμης) πληροφορίας ή προτύπων από μεγάλες βάσεις δεδομένων με χρήση αλγορίθμων ομαδοποίησης ή κατηγοριοποίησης και των αρχών της στατιστικής, της τεχνητής νοημοσύνης, της μηχανικής μάθησης και των συστημάτων βάσεων δεδομένων. Στόχος της εξόρυξης δεδομένων είναι η πληροφορία που θα εξαχθεί και τα πρότυπα που θα προκύψουν να έχουν δομή κατανοητή προς τον άνθρωπο έτσι ώστε να τον βοηθήσουν να πάρει τις κατάλληλες αποφάσεις.

Ο πραγματικός στόχος της εξόρυξης δεδομένων είναι η αυτόματη ή ημιαυτόματη ανάλυση μεγάλων ποσοτήτων δεδομένων για την εξαγωγή κάποιου ενδιαφέροντος προτύπου που ήταν άγνωστο μέχρι εκείνη τη στιγμή, όπως ομάδες από εγγραφές δεδομένων (συσταδοποίηση), ασυνήθιστες εγγραφές (anomaly detection) και εξαρτήσεις (κανόνες συσχετίσεων). Αυτό συνήθως συμπεριλαμβάνει τη χρήση μιας βάσης δεδομένων, όπως και χωρικά ευρετήρια. Αυτά τα πρότυπα ύστερα μπορούν να θεωρηθούν ως μία περιγραφή των δεδομένων εισαγωγής και να χρησιμοποιηθούν για περαιτέρω ανάλυση ή ως παράδειγμα στην εκμάθηση μηχανής και στην προγνωστική ανάλυση. Για παράδειγμα, οι διακυμάνσεις μιας μετοχής σε συνδυασμό με κάποιους οικονομικούς δείκτες μπορούν να χρησιμοποιηθούν να χρησιμοποιηθούν για την πρόβλεψη της μακροχρόνιας τιμής αυτής της μετοχής. Τα δεδομένα αυτά, μετά την εξόρυξή τους θα αναλυθούν και με μεθόδους στατιστικής και μηχανικής μάθησης θα εξαχθεί μια πρόγνωση [19].

Η εξόρυξη δεδομένων δεν αποτελεί παρά έναν κρίκο στην αλυσίδα που λέγεται ανακάλυψη γνώσης από βάσεις δεδομένων (Knowledge Discovery from Data) και έχει τα εξής στάδια τα οποία εικονίζονται και παρακάτω:

- Συλλογή δεδομένων ή συγκέντρωση δεδομένων εστιάζοντας σε ένα υποσύνολο από μεταβλητές η δείγματα δεδομένων από τα οποία θα γίνει η ανακάλυψη γνώσης.
- Προεπεξεργασία , δηλαδή απομάκρυνση θορύβου, επιλογή στρατηγικής για το χειρισμό ελλιπών δεδομένων.
- Μετασχηματισμός των δεδομένων σε μορφή κατάλληλη για την εξόρυξη εκτελώντας λειτουργίες σύνοψης ή συνάθροισης.
- Εξόρυξη δεδομένων με χρήση ευφυών μεθόδων με σκοπό την εξαγωγή προτύπων.
- Ερμηνεία/ Αξιολόγηση της εξαχθείσας πληροφορίας και πιθανώς απεικόνισή της [20].



Σχήμα 3.1 : Στάδια του Data Mining

3.2 Ανάλυση Συναισθήματος

3.2.1 Γενικά

Η ανάλυση συναισθήματος από γραπτό λόγο (sentiment analysis), ή αλλιώς εξόρυξη γνώμης-άποψης, είναι μια υποπεριοχή της εξόρυξης δεδομένων από κειμένο (text data mining) που ασχολείται με το συστηματικό προσδιορισμό, την εξαγωγή, την ποσοτικοποίηση και την μελέτη των συναισθηματικών καταστάσεων και υποκειμενικών πληροφοριών των χρηστών μέσα από το γραπτό τους λόγο. Πιο συγκεκριμένα, η ανάλυση συναισθήματος σύμφωνα με τον ορισμό του λεξικού της Οξφόρδης, είναι : «Η διαδικασία υπολογιστικής ταυτοποίησης και κατηγοριοποίησης των απόψεων που εκφράζονται σε ένα κομμάτι κειμένου, ειδικά για να προσδιοριστεί εάν η στάση του συγγραφέα έναντι ενός συγκεκριμένου θέματος, προϊόντος κλπ. είναι θετική, αρνητική ή ουδέτερη» [21]. Χρησιμοποιεί τεχνικές Επεξεργασίας Φυσικής Γλώσσας (Natural Language Processing), στατιστικές μεθόδους και μεθόδους μηχανικής μάθησης για την ταξινόμηση ενός κειμένου σε κλάσεις που εκφράζου συναισθήματα.

Ως Επεξεργασία Φυσικής Γλώσσας (NLP) ορίζουμε το πεδίο της Τεχνητής Νοημοσύνης (Artificial Intelligence) το οποίο ασχολείται με την αλληλεπίδραση μεταξύ των υπολογιστών και της ανθρώπινης (φυσικής) γλώσσας και είναι ουσιαστικά υπεύθυνο για τη κατανόηση από τον υπολογιστή του ανθρώπινου αδόμητου κειμένου. Έτσι, το πεδίο αυτό προσπαθεί να γεφυρώσει το χάσμα μεταξύ των αυστηρά δομημένων γλωσσών προγραμματισμού, με τις οποίες γίνεται ουσιαστικά η επικοινωνία ανθρώπου-μηχανής και την πιο χαλαρά δομημένη ανθρώπινη γλώσσα με την οποία γίνεται η επικοινωνία μεταξύ των ανθρώπων.

Γενικά, η Ανάλυση Συναισθήματος έχει ως σκοπό να προσδιορίσει είτε τη γενικότερη συναισθηματική κατάσταση ενός συγγραφέα κατά τη συγγραφή του κειμένου, τη στάση ή την άποψή του σχετικά με ένα θέμα ή γεγονός ή κατάσταση, η οποία μεταδίδεται ηθελημένα ή μη μέσω του κειμένου. Χρησιμοποιείται δηλαδή για την ανίχνευση γενικότερων συναισθημάτων όπως χαρά, λύπη, απογοήτευση και καταστάσεων όπως ειρωνεία, χιούμορ αλλά και στον προσδιορισμό μιας έκφρασης ως προς το αν είναι θετική, αρνητική ή ουδέτερη. Στην τελευταία περίπτωση η ταξινόμηση γίνεται μέσω κατηγοριών συναισθημάτων ή οποίες συνήθως κυμαίνονται από δύο (θετικό, αρνητικό) μέχρι πέντε (από πολύ αρνητικό έως πολύ θετικό).

3.2.2 Σημασία

Η Ανάλυση Συναισθήματος είναι κριτικής σημασίας, ειδικά στον τομέα των επιχειρήσεων, διότι βοηθάει μια επιχείρηση να εκτιμήσει τί αρέσει στους καταναλωτές και τί όχι σχετικά με τις υπηρεσίες που προσφέρει. Στην πράξη, η ανατροφοδότηση που δέχεται μέσω των σχολίων των πελατών της στα κοινωνικά δίκτυα, στην ιστοσελίδα της ή σε άλλες πλατφόρμες αποτελεί μιας πρώτης τάξεως πηγή πληροφοριών για την επιχείρηση καθώς μπορεί να εξορύξει πληροφορία, όχι μόνο για το θέμα που συζητούν αλλά και για τη γνώμη τους πάνω στο θέμα. Έτσι, έχει στη διάθεσή της την άποψη του κοινού ανά πάσα στιγμή, για τη ίδια την επιχείρηση ή για τα επιμέρους προϊόντα της και μπορεί ακολούθως να χαράξει τη διαφημιστική της πολιτική ή να βελτιώσει τις υπηρεσίες της. Η Ανάλυση Συναισθήματος, όντας μια διαδικασία δυναμική, ανανεώνει συνεχώς τα ευρήματά παράγοντας ένα μεγάλο όγκο δεδομένων που, εάν χρησιμοποιηθεί ορθά και με αποδοτικούς αλγορίθμους, μπορεί να οδηγήσει στη βελτίωση της εκάστοτε επιχείρησης σε επίπεδο ποιότητας προϊόντων, πωλήσεων, διαφήμισης καθώς και προσωπικού το οποίο ανταποκρίνεται στις ανάγκες των καταναλωτών.

Εξαιρετικής χρησιμότητας είναι η χρήση της Ανάλυσης Συναισθήματος στα κοινωνικά δίκτυα, μιας και μας παρέχει μια γενική εικόνα σχετικά με την ευρεία γνώμη του κοινού για κάποιο θέμα. Πολλοί οργανισμοί στις μέρες μας χρησιμοποιούν αυτή τη δυνατότητα της εξαγωγής χρήσιμης πληροφορίας από τα κοινωνικά δίκτυα για δική τους εκμετάλλευση. Για παράδειγμα έχει παρατηρηθεί ότι οι αλλαγές της διάθεσης στα κοινωνικά δίκτυα σχετίζονται με τις διακυμάνσεις των μετοχών του χρηματιστηρίου. Επίσης, πολιτικές καμπάνιες ήδη από τις προεδρικές εκλογές του 2012 στην Αμερική χρησιμοποιούν την Ανάλυση Συναισθήματος για να αφουγκραστούν τις ανάγκες, τις προσδοκίες και τις προτιμήσεις των ψηφοφόρων με σκοπό να τους προσελκύσουν με μεγαλύτερη αποτελεσματικότητα. Ακόμα, χρησιμοποιείται κατά κόρον στην επιχειρησιακή έρευνα για την ανίχνευση τάσεων και τη δημιουργία επιτυχημένων προϊόντων καθώς και στην βελτίωση της αντίληψης μια επιχείρησης όσον αφορά την εικόνα που έχουν για αυτή οι καταναλωτές της. Τέλος, αλγόριθμοι Ανάλυσης Συναισθήματος χρησιμοποιούνται και σε μοντέλα πρόβλεψης για τη λήψη αποφάσεων και τη

χάραξη στρατηγικών έχοντας ως αποτέλεσμα τη δημιουργία όλο και πιο πολύπλοκων αλγορίθμων και την υλοποίηση όσο το δυνατόν πιο αξιόπιστων συστημάτων [22].

3.2.3 Προβλήματα

Παρ' όλα αυτά η Ανάλυση Συναισθήματος από κείμενο αντιμετωπίζει δυσκολίες και συχνά δε μπορεί να παράξει ακριβή αποτελέσματα. Τα πιο βασικά προβλήματα στην Ανάλυση Συναισθήματος είναι :

- Ο χειρισμός της άρνησης (negation handling), η οποία επηρεάζει την πολικότητα και πρέπει να ληφθεί υπόψιν. Αρχικά, η άρνηση δεν εκφράζεται μόνο με απλές λέξεις όπως όχι, δεν, μη αλλά με διάφορα μέρη του λόγου που χρησιμοποιούμε καθημερινά με αρνητική χροιά (μισώ, ανεπαρκής, δυστυχώς), καθώς και μέσω συγκρίσεων που φανερώνουν μια αρνητική γνώμη. Ακόμα, πολλές φορές η χρήση αρνητικών λέξεων, όχι μόνο δεν αντιστρέφει την πολικότητα αλλά χρησιμοποιείται για να δώσει έμφαση και να την ενισχύσει (όχι απλά καλός).
- Η αμφισημία (ambiguity), δηλαδή η ύπαρξη πολλαπλών ερμηνειών μιας λέξης ή μιας φράσης ανάλογα με το πλαίσιο στο οποίο εντάσσεται. Αυτό έχει ως αποτέλεσμα φαινομενικά απλές φράσεις να έχουν διαφορετικό συναίσθημα ανάλογα τον τρόπο ερμηνείας τους (πχ. η φράση «η αναμόρφωση του σταδίου τους πήρε πέντε χρόνια» έχει διαφορετική πολικότητα ανάλογα με την έκταση και την πραγματική δυσκολία του έργου) και την προσωπική πεποίθηση του συγγραφέα (πχ. η φράση «ο Παναθηναϊκός διέλυσε τον Ολυμπιακό» έχει διαφορετική πολικότητα ανάλογα με την ομάδα που υποστηρίζει αυτός που το γράφει).
- Οι υπαινιγμοί, κατηγορία που αφορά δηλώσεις που αφορούν ειρωνεία, χιούμορ ή σαρκασμό. Έννοιες όπως αυτές είναι συχνά δύσκολο να ανιχνευθούν από τον άνθρωπο κατά πόσο μάλλον από μια μηχανή. Το βασικό χαρακτηριστικό αυτών των δηλώσεων είναι ότι για την ανίχνευση και την αποκωδικοποίησή τους απαιτείται γνώση του κοινωνικοπολιτικού υποβάθρου του ατόμου που τις κάνει (πχ. η δήλωση «λεφτά υπάρχουν» μπορεί να χρησιμοποιηθεί για να δηλώσει διαφορετικό συναίσθημα ανάλογα με το πλαίσιο).

- Η Αναγνώριση Ονομαστικών Οντοτήτων (Named Entity Recognition), δηλαδή ο εντοπισμός των οντοτήτων που αναφέρονται στο κείμενο όπως πρόσωπα, προϊόντα και τοποθεσίες. Είναι πολύ σημαντικό για τον εντοπισμό των αντικειμένων για τα οποία εκφράζεται η άποψη για την αποφυγή λάθος ερμηνειών (πχ. η δήλωση «έγραψα το διαγώνισμα και μετά πήγα για φαγητό και ήταν απαίσιο» είναι αρνητική μόνο όσον αφορά το φαγητό).

  Σε αυτό το σημείο είναι χρήσιμο να αναφέρουμε ότι η Ανάλυση Συναισθήματος σε κοινωνικά δίκτυα και συγκεκριμένα στο Twitter πάσχει και από άλλα προβλήματα όπως είναι η χρήση αργκό του διαδικτύου με συντομογραφίες, αρκτικόλεξα και διάφορες ορθογραφίες για την ίδια λέξη, ειδικούς χαρακτήρες όπως emoticons (που συχνά βοηθούν την Ανάλυση Συναισθήματος καθώς από μόνοι τους εκφράζουν συναίσθημα) καθώς και από τη χρήση συχνά περισσότερων από μια γλωσσών στην ίδια πρόταση.

3.2.4 Κατηγορίες

Η Ανάλυση Συναισθήματος μπορεί να χωριστεί σε κατηγορίες ανάλογα με το επίπεδο στο οποίο υλοποιείται. Υπάρχουν τρία βασικά επίπεδα υλοποίησης της Ανάλυσης Συναισθήματος:

1. Επίπεδο Εγγράφου (Document Level). Σε αυτή την περίπτωση γίνεται μελέτη ενός εγγράφου για την κατηγοριοποίηση του ως έγγραφο που εκφράζει θετική ή αρνητική άποψη, η οποία γίνεται με την παραδοχή ότι αυτό το έγγραφο εκφράζει ένα μόνο συναίσθημα για μια μόνο οντότητα όπως πχ. κριτική για μια ταινία ή προϊόν.
2. Επίπεδο Πρότασης (Sentence Level). Σε αυτό το επίπεδο πραγματοποιείται ανάλυση κάθε πρότασης για την ανάλυση του συναισθήματος που εκφράζει ως θετικό ή αρνητικό. Θεωρείται ότι κάθε πρόταση εκφράζει ξεχωριστό συναίσθημα.
3. Επίπεδο Οντότητας (Entity Level). Εδώ απομονώνονται οι οντότητες για τις οποίες εκφράζεται κάποια γνώμη και γίνεται ανάλυση για κάθε διαφορετική οντότητα. Πολλές οντότητες μπορεί να συνυπάρχουν στο ίδιο κείμενο, ακόμα και στην ίδια πρόταση και να εκφέρεται διαφορετική άποψη για την κάθε μια.

Στην παρούσα διπλωματική εργασία θα χρησιμοποιηθεί μια μείξη των πρώτων δύο προσεγγίσεων καθώς αντικείμενο της Ανάλυσης Συναισθήματος είναι διάφορα «τιτιβίσματα» (tweets), καθένα από τα οποία θεωρείται έγγραφο με μία πολικότητα, αν και πολλές φορές λόγω του περιορισμού των λέξεων και της ανάγκης για λακωνικότητα το μέγεθος τους δεν ξεπερνά τη μία πρόταση.

Για την υλοποίηση της Ανάλυσης Συναισθήματος υπάρχουν τρεις προσεγγίσεις που ακολουθούνται κατά κόρον οι οποίες είναι οι εξής:

1. Μέθοδοι με χρήση λεξικού (lexicon-based approaches) όπου η κάθε πρόταση χωρίζεται σε λέξεις, μερικές από τις οποίες αντιστοιχούν σε λέξεις που εκφράζουν συναίσθημα και το συναίσθημα αυτό καθώς και η ένταση του υπολογίζονται με βάση ενός υπάρχοντος λεξικού (συνήθως είναι θετικό για θετικό συναίσθημα και αντίστροφα για αρνητικό) ενώ γίνεται και χειρισμό της άρνησης. Έτσι, για την εκτίμηση του συναισθήματος κάθε πρότασης συνδυάζονται οι επιμέρους βαθμολογίες των λέξεων που αντιστοιχούν σε συναισθηματικές λέξεις στο λεξικό (πολλές φορές αθροίζοντάς τες). Αν και φαινομενικά αφελής μέθοδος, η χρήση της δίνει αξιόπιστα αποτελέσματα.
2. Μέθοδοι Μηχανικής Μάθησης (machine learning approaches). Είναι και η μέθοδος που χρησιμοποιείται περισσότερο εξαιτίας της προσαρμοστικότητας και της ακρίβειάς της. Χρησιμοποιεί τη βασική ακολουθία εργασιών της μηχανικής μάθησης, δηλαδή συλλογή δεδομένων, προεπεξεργασία, εκπαίδευση και ταξινόμηση με τη χρήση διάφορων διανυσμάτων χαρακτηριστικών.
3. Υβριδικές μέθοδοι (hybrid approaches). Αποτελούν μια μείξη των δύο παραπάνω μεθόδων και επιχειρούν να συνδυάσουν την ακρίβεια των μεθόδων μηχανικής μάθησης με την ταχύτητα των μεθόδων με χρήση λεξικού [23].

   Συμπερασματικά και οι δύο μέθοδοι έχουν πλεονεκτήματα και μειονεκτήματα με την πρώτη μέθοδο να είναι πιο γρήγορη αλλά και λιγότερο ακριβής ενώ η δεύτερη είναι πιο ακριβής αλλά απαιτεί μεγάλο όγκο δεδομένων στο στάδιο της εκπαίδευσης .

Στην παρούσα διπλωματική εργασία χρησιμοποιήθηκε αποκλειστικά η προσέγγιση της Ανάλυσης Συναισθήματος με μεθόδους μηχανικής μάθησης τα χαρακτηριστικά και οι διάφορες υλοποιήσεις της οποία θα αναλυθούν εκτενώς στην επόμενη ενότητα

3.2.5 Κορυφαίες Μέθοδοι (State of the Art)

Αν και ο τομέας της ανάλυσης συναισθήματος και της εξόρυξης γνώμης είναι σχετικά καινούργιος, έχει πραγματοποιηθεί εκτεταμένη έρευνα πάνω στο αντικείμενο. Στη διάρκεια της έρευνας πάνω στον τομέα της ανάλυσης συναισθήματος τα τελευταία χρόνια έχουν προταθεί διάφορες προσεγγίσεις. Από την ταξινόμηση, αρχικά, επιπέδου εγγράφων (Pang and Lee [24]), στην εκμάθηση της πολικότητας λέξεων και φράσεων (πχ. Hatzivassiloglou and McKeown [25]; Esuli and Sebastiani [26]). Με δεδομένο τον περιορισμό χαρακτήρων στα tweets, η ανάλυση συναισθήματος στο Twitter, προσεγγίζεται σωστότερα από ταξινόμηση επιπέδου πρότασης (πχ. Kim and Hovy [27]; Wilson, Wiebe and Hoffman [28]) αν και η ιδιαιτερότητα της χρησιμοποιούμενης γλώσσας καθώς και η ίδια η δομή των μικροϊστολογίων δυσχεραίνουν κατά πολύ την εργασία της ανάλυσης συναισθήματος. Επίσης, έχει πραγματοποιηθεί αρκετή έρευνα τα τελευταία χρόνια στο πεδίο της ανάλυσης συναισθήματος και επικαιρότητας στο Twitter (πχ. Pak and Paroubek [29]; O' Connor et al. [30]; Tumasjan et al. [31]; Bifet and Frank [32]; Barbosa and Feng [33]; Davidov, Tsur and Rappoport [34]). Ακόμα, ο Erik Cambria έχει κάνει μια εισαγωγή στην ανάλυση συναισθήματος σε επίπεδο έννοιας, όπου εστιάζει στη σημασιολογική ανάλυση (semantic analysis) του κειμένου.

Πολλοί ερευνητές υιοθετούν την προσέγγιση της χρήσης των διάφορων μερών του λόγου ως χαρακτηριστικών (part-of-speech features) με τα αποτελέσματα να παραμένουν όχι ικανοποιητικά. Ακόμα, πολλές φορές χρησιμοποιούνται ως χαρακτηριστικά στην ανάλυση συναισθήματος, ιδιαίτερα στην περίπτωση των μικροϊστολογίων, τα emoticons, ενώ για τη γενική περίπτωση υπάρχουν ανεπαρκή δείγματα για το εάν αυτή η προσέγγιση είναι χρήσιμη.

3.3 Μηχανική Μάθηση

Η έννοια της μάθησης σε ένα γνωστικό σύστημα μπορεί να προσδιοριστεί με δύο βασικές ιδιότητες : την ικανότητά του να αποκτά επιπλέον γνώση μέσω της αλληλεπίδρασης με το

περιβάλλον στο οποίο δραστηριοποιείται, καθώς και την ικανότητά του να βελτιώνει με την επανάληψη τον τρόπο που εκτελεί μια ενέργεια. Η Μηχανική Μάθηση αποτελεί μια υποπεριοχή της επιστήμης των υπολογιστών που αναπτύχθηκε από την μελέτη της αναγνώριση προτύπων και της υπολογιστικής θεωρίας μάθησης στην τεχνητή νοημοσύνη. Ένας ορισμός της Μηχανικής Μάθησης δόθηκε το 1959 από τον Arthur Samuel ο οποίος αναφέρει : «Πεδίο μελέτης που δίνει στους υπολογιστές την ικανότητα να μαθαίνουν, χωρίς να έχουν ρητά προγραμματιστεί». Η γνώση αυτή προκύπτει από αλγορίθμους οι οποίοι εκτελούνται πάνω σε κάποια δεδομένα με αποτέλεσμα να μαθαίνουν από αυτά και να μπορούν να κάνουν προβλέψεις ή να λάβουν αποφάσεις με βάση αυτά.

Ο τομέας της τεχνητής νοημοσύνης δημιουργήθηκε στα μέσα του 20^{ου} αιώνα και γνωρίζει όλο και αυξανόμενη δημοτικότητα στις σημερινές κοινωνίες. Από τον Alana Turing και το παιχνίδι της μίμησης μέχρι σήμερα έχουν αλλάξει πολλά στην προσέγγιση της τεχνητής νοημοσύνης και κυρίως της έννοιας της μηχανικής μάθησης. Αρχικά, η παραδοσιακή προσέγγιση όριζε τη δημιουργία πληθώρας κανόνων λογικού συλλογισμού σε συνδυασμό με πιθανά ενδεχόμενα, σύμφωνα με τα οποία θα αντιδρούσε η μηχανή. Αυτή η μέθοδος όμως ήταν αφενός χρονοβόρα για τους προγραμματιστές και αφετέρου η αποτελεσματικότητά της εξαρτιόταν από την σαφήνεια των κανόνων. Η σύγχρονη προσέγγιση έφερε στο προσκήνιο το τομέα της μηχανικής μάθησης, υιοθετώντας την αρχή ότι για την εκπαίδευση μιας μηχανής για να εκτελεί ορισμένες εργασίες απαιτείται παρόμοια αντιμετώπιση με αυτή που συναντιέται στη διδασκαλία ενός παιδιού. Δεν χρειάζεται πλέον να ορίζονται εκ των προτέρων σαφείς και πολύπλοκοι κανόνες για κάθε λειτουργία, αλλά αρκεί η ύπαρξη επαρκών υπολογιστικών πόρων στους οποίους διοχετεύονται αρκετά παραδείγματα, σε συνδυασμό με τη σχεδίαση αποτελεσματικών αλγορίθμων που ορίζουν τη διαδικασία της μάθησης. Με αυτό τον τρόπο βελτιώνεται η απόδοση της μηχανής μέσω συνεχών δοκιμών και σφαλμάτων, εξαγωγής προτύπων και ελέγχου των αρχικών υποθέσεων. Βέβαια, σε αντίθεση με την διδασκαλία ενός ανθρώπου η μηχανή υστερεί στην ικανότητα γενίκευσης δηλαδή στη δυνατότητα να παράξει νέα γνώση με τη χρήση των ήδη υπαρχόντων εκπαιδευτικών παραδειγμάτων, ικανότητα η οποία ουσιαστικά χαρακτηρίζει και την απόδοση του συστήματος.

Με αυτό τον τρόπο η μηχανική μάθηση έχει καταφέρει να καθιερωθεί ως η πλέον κατάλληλη μέθοδος στην ανάλυση δεδομένων με αμέτρητες εφαρμογές στην καθημερινότητα και μεγάλο ερευνητικό ενδιαφέρον. Η ραγδαία αυτή ανάπτυξη είναι εν μέρει απόρροια της ύπαρξης πολλών τεχνικών μηχανικής μάθησης οι οποίες μπορούν να ταξινομηθούν σε τέσσερις κατηγορίες ανάλογα με την φύση του εκπαιδευτικού συστήματος ή την ανατροφοδότηση που είναι διαθέσιμη στο σύστημα εκμάθησης :

- Μάθηση με επίβλεψη (supervised learning) : σε αυτές τις μεθόδους μάθησης το υπολογιστικό σύστημα δημιουργεί μια συνάρτηση που αντιστοιχίζει τις εισόδους στις επιθυμητές εξόδους, με τη χρήση παραδειγματικών εισόδων με γνωστά τα επιθυμητά αποτελέσματα. Ουσιαστικά το σύστημα προσομοιώνει μια συνάρτηση που περιγράφει τα δεδομένα με τον καλύτερο δυνατό τρόπο. Τα δύο κύρια είδη προβλημάτων που χρησιμοποιείται είναι η ταξινόμηση και η παρεμβολή.
- Μάθηση χωρίς επίβλεψη (unsupervised learning): στις μεθόδους μάθησης χωρίς επίβλεψη η μάθηση γίνεται χωρίς να παρέχεται κάποια εμπειρία επί των δεδομένων. Το σύστημα ανακαλύπτει συσχετίσεις και ομάδες από δεδομένα χωρίς να γνωρίζει εκ των προτέρων αν υπάρχουν, πόσες και ποιες είναι. Χρησιμοποιείται κυρίως στη δημιουργία κανόνων συσχέτισης και στην ομαδοποίηση και πολλές φορές αποτελεί ένα ενδιάμεσο στάδιο της επεξεργασίας.
- Μάθηση με μερική επίβλεψη (semi-supervised learning) : αποτελεί το ενδιάμεσο στάδιο των δύο προηγούμενων μεθόδων καθώς το σύστημα έχει στη διάθεση του κάποια δεδομένα εκπαίδευσης τα οποία όμως είναι ελλιπή.
- Ενισχυτική μάθηση (reinforcement learning) : κατά την εφαρμογή τέτοιων μεθόδων μάθησης το σύστημα χρησιμοποιεί δεδομένα και παρατηρήσεις τα οποία συλλέγει από την αλληλεπίδραση με το περιβάλλον του με στόχο να μεγιστοποιήσει το κέρδος ή να ελαχιστοποιήσει το λάθος. Δεν παρέχονται ακριβής δεδομένα όσον αφορά το στόχο του συστήματος παρά μόνο ένα σύστημα επιβράβευσης σύμφωνα με το οποία το σύστημα μεγιστοποιεί την απόδοσή του [35].

Στη συνέχεια θα παρουσιαστούν σύντομα, αλλά και περιεκτικά, οι αλγόριθμοι μηχανικής μάθησης που χρησιμοποιήθηκαν κατά την υλοποίηση της παρούσας διπλωματικής εργασίας. Οι αλγόριθμοι που θα παρουσιαστούν είναι αλγόριθμοι μηχανικής μάθησης που λύνουν ουσιαστικά ένα πρόβλημα ταξινόμησης, αυτό της Ανάλυσης Συναισθήματος. Πριν όμως περάσουμε στην περιγραφή των επιμέρους αλγορίθμων καλό θα ήταν να παρουσιάσουμε τα βασικά συστατικά ενός προβλήματος ταξινόμησης.

Οι μετρήσεις που χρησιμοποιούνται για την ταξινόμηση, ονομάζονται χαρακτηριστικά $x_i,\ i = 1,2,\dots,n$ και σχηματίζουν το διάνυσμα χαρακτηριστικών $x = [x_1,\ x_2,\dots,x_n]$, όπου κάθε ένα ορίζει ένα διαφορετικό πρότυπο. Τα χαρακτηριστικά και τα διανύσματα θεωρούνται τυχαίες μεταβλητές διότι θεωρείται ότι προκύπτουν από μετρήσεις που παρουσιάζουν τυχαία διακύμανση. Το πρόβλημα της ταξινόμησης αφορά την επιτυχή ταξινόμηση κάθε διανύσματος χαρακτηριστικών σε μία από τις k πιθανές κλάσεις $C_1, C_2,\ C_3, \dots., C_k\ \in\ C$, όπου C το σύνολο των κλάσεων.

Τα πρότυπα των οποίων η κλάση είναι γνωστή ονομάζονται δεδομένα με ετικέτα και αυτά που χρησιμοποιούνται κατά την εκπαίδευση και το σχεδιασμό του ταξινομητή ονομάζονται δεδομένα (ή πρότυπα) εκπαίδευσης. Για την σωστή εκπαίδευση του μοντέλου πρέπει να του παρέχουμε πληθώρα προτύπων εκπαίδευσης (training set) που να καλύπτουν τις περισσότερες αν όχι όλες τις περιπτώσεις του προβλήματός ώστε να μπορεί να έχει καλές επιδόσεις στα νέα δεδομένα. Επίσης, είναι εξίσου χρήσιμο να υπάρχει μια σχετική αξιοπιστία στα δεδομένα εκπαίδευσης, υπό την έννοια ότι όλες οι κατηγορίες - κλάσεις θα πρέπει να αντιπροσωπεύονται στο βαθμό που συναντώνται στο πραγματικό πρόβλημα. Για αυτό το λόγο θα πρέπει να γίνεται με ιδιαίτερη προσοχή η συγκέντρωση δεδομένων εκπαίδευσης καθώς και η κατηγοριοποίησή τους με τη χρήση ετικετών για την εκπαίδευση των μοντέλων. Βέβαια, θα πρέπει να φροντίσουμε έτσι ώστε το μοντέλο μας να μην υπερεκπαιδευτεί στα δεδομένα εκπαίδευσης (overfitting) , δηλαδή το ενδεχόμενο να προσαρμοστεί πολύ καλά εμφανίζοντας μεγάλη ακρίβεια σε αυτά μειώνοντας όμως την απόδοσή του σε δεδομένα που θα διαφέρουν από τα δεδομένα εκπαίδευσης. Ο προγραμματιστής πρέπει να λάβει υπόψιν του αυτόν τον κίνδυνο και να λάβει μέτρα για την αντιμετώπισή του κατά την παραγωγή του μοντέλου.

Επίσης, χρειάζεται ένα σετ δεδομένων επαλήθευσης (validation set) το οποίο χρησιμοποιείται κατά την παραγωγή του μοντέλου για τον έλεγχο της αξιοπιστίας του.

Στόχος λοιπόν του προβλήματος ταξινόμησης είναι η χρήση της γνώσης από τα δεδομένα εκπαίδευσης για την ταξινόμηση των δεδομένων χωρίς ετικέτα στη σωστή κλάση. Για την επίτευξη αυτού το στόχου έχουν αναπτυχθεί διάφοροι ταξινομητές οι οποίοι προσεγγίζουν διαφορετικά το πρόβλημα (γεωμετρικά, πιθανοτικά) και θα αναλυθούν παρακάτω.

Αλγόριθμοι ταξινόμησης Bayes

Οι αλγόριθμοι ταξινόμησης Bayes βασίζονται στη θεωρία αποφάσεων κατά Bayes. Έτσι, όταν έχουμε ένα πρόβλημα ταξινόμησης σε κάποια από τις k κλάσεις $C_1, C_2, C_3, \dots, C_k$ και ένα πρότυπο, που αντιστοιχεί σε ένα διάνυσμα χαρακτηριστικών x, που πρέπει να ταξινομηθεί, τότε για την ταξινόμηση αυτή υπολογίζουμε τις πιθανότητες $P(C_i|x) = P(C_i|x_1, x_2, \dots, x_n)\ i = 1,2, \dots, k$, οι οποίες ονομάζονται εκ των υστέρων ή αλλιώς a posteriori πιθανότητες. Κάθε μια από αυτές εκφράζει την πιθανότητα το διάνυσμα να ανήκει στη συγκεκριμένη κλάση και για την ταξινόμηση του διανύσματος επιλέγουμε την κλάση η οποία εμφανίζει τη μεγαλύτερη a posteriori πιθανότητα για αυτό $\hat{y} = argmax_{\mathrm{i}\,\in(1,2,\dots,\mathrm{k})} P(C_i|x)$, όπου $\hat{y}$ είναι η απόφαση του ταξινομητή. Η παραπάνω διαδικασία είναι ίδια για όλους τους στατιστικούς ταξινομητές, παρ' όλα αυτά οι Μπευζιανοί ταξινομητές χρησιμοποιούν το θεώρημα του Bayes για τον υπολογισμό των παραπάνω πιθανοτήτων που συνδέει την prior με την posterior πιθανότητα. Σύμφωνα με αυτό για την posterior πιθανότητα $P(C_i|x)$ έχουμε:

$$P(C_i|x) = \frac{P(x|C_i)P(C_i)}{P(x)}$$

όπου $P(x)$ είναι η συνάρτηση πυκνότητας πιθανότητας του διανύσματος χαρακτηριστικών x, η οποία όμως είναι κοινή για όλες τις κλάσεις (ξεχωριστή για κάθε διάνυσμα χαρακτηριστικών) και άρα μπορεί να παραληφθεί στον υπολογισμό, οπότε η απόφαση του ταξινομητή πλέον δίνεται από τον τύπο $\hat{y} = argmax_{\mathrm{i}\,\in(1,2,\dots,\mathrm{k})}\, P(x|C_i)P(C_i)$, όπου ισχύει ότι $P(x|C_i)P(C_i) =$

$P(x, C_i) = P(x_1, x_2, \dots, x_n, C_i)$. Κάνοντας χρήση αυτού και σύμφωνα με τον κανόνα της αλυσίδας έχουμε:

$P(x_1, x_2, \dots, x_n, C_i) = P(x_1|x_2, \dots, x_n, C_i)P(x_2|x_3, \dots, x_n, C_i) \dots P(x_{n-1}|x_n, C_i)P(x_n| C_i)P(C_i)$.

Μένει να υπολογιστούν οι παραπάνω πιθανότητες, οι οποίες εκτιμώνται από τα δεδομένα εκπαίδευσης. μια διαδικασία που μπορεί να αποδειχθεί ακριβή υπολογιστικά για αυτό συχνά γίνεται η υπόθεση της στατιστικής ανεξαρτησίας μεταξύ των δειγμάτων, ότι δηλαδή ισχύει $P(x_1|x_2, \dots, x_n, C_i) = P(x_1| C_i)$. Ακολούθως, η παραπάνω σχέση μετασχηματίζεται ως εξής :

$$P(x_1, x_2, \dots, x_n, C_i) = P(x_1| C_i)P(x_2| C_i) \dots P(x_{n-1}| C_i)P(x_n| C_i)P(C_i) = P(C_i)\prod_{j=1}^{n} P(x_j| C_i)$$

Άρα τελικά η εκτίμηση ταξινόμηση γίνεται με τη σχέση $\hat{y} = argmax_{\text{i} \in (1,2,\dots,\text{k})} P(C_i)\prod_{j=1}^{n} P(x_j| C_i)$ και έτσι προκύπτει ο **Απλοϊκός (ή αφελής) Ταξινομητής Bayes (Naïve Bayes Classifier**), ο οποίος αν και κάνει μια φαινομενικά αφελή παραδοχή, είναι εξαιρετικά χρήσιμος και δημοφιλής, εξαιτίας της μεγάλης αποτελεσματικότητάς του σε πραγματικά προβλήματα καθώς και της ταχύτητας στην εκπαίδευσή του λόγω της απλότητάς του. Η εκπαίδευση του μοντέλου, όπως φαίνεται, ουσιαστικά ισοδυναμεί με τον υπολογισμό των πιθανοτήτων $P(x|C_i) = \prod_{j=1}^{n} P(x_j| C_i)$, ο οποίος βασίζεται στο γεγονός ότι τα δεδομένα προκύπτουν από κάποια κατανομή, της οποίας οι παράμετροι θα εκτιμηθούν με τη μέθοδο της μέγιστης πιθανοφάνειας (maximum likelihood). Όσον αφορά τις a priori πιθανότητες των κλάσεων, συνήθως υπολογίζονται ως εξής :

$$P(C_i) = \frac{αριθμός\ δεδομένων\ στην\ κλάση\ C_i}{αριθμός\ συνολικών\ δεδομένων}, \qquad i = 1,2,..,k$$

Επίσης όταν ασχολούμαστε με συνεχή δεδομένα, μια τυπική παραδοχή είναι ότι οι συνεχείς τιμές που σχετίζονται με κάθε τάξη κατανέμονται σύμφωνα με Gaussian κατανομή.

Multinomial Naive Bayes

Οι Multinomial ταξινομητές εφαρμόζονται σε περιπτώσεις όπου τα δεδομένα δεν είναι συνεχή, αλλά αντιπροσωπεύουν συχνότητες με τις οποίες τα δεδομένα παράγονται από μια multinomial κατανομή $(p_1, p_2, \dots, p_n)$. Σε αυτές τις περιπτώσεις το διάνυσμα χαρακτηριστικών δεν είναι τίποτα παραπάνω από ένα ιστόγραμμα όπου η τιμή κάθε

χαρακτηριστικού x_ι είναι ουσιαστικά ο αριθμός των εμφανίσεών του στο κάθε διάνυσμα. Αυτού του είδους οι ταξινομητές χρησιμοποιούνται συχνά στην ταξινόμηση κειμένου και ειδικά σε συνδυασμό με την τεχνική εξαγωγής χαρακτηριστικών Bag-of-Words όπου κάθε διάνυσμα χαρακτηριστικών, έστω πρόταση, αποτελείται από τον αριθμό των εμφανίσεων κάθε λέξης του λεξιλογίου στη συγκεκριμένη πρόταση. Η πιθανότητα εμφάνισης ενός διανύσματος x στην κλάση C_i είναι : $P(x|\, C_i) = \frac{(\sum_j x_j)!}{\prod_j x_j} \prod_j p_{ij}^{x_j}$, όπου p_{ij} είναι η πιθανότητα στην κλάση C_i να εμφανιστεί το χαρακτηριστικό x_j. Με αυτό τον τρόπο εκτίμησης της πιθανότητας, εάν μια δεδομένη τιμή χαρακτηριστικού δεν εμφανιστεί ποτέ σε μια κλάση στα δεδομένα εκπαίδευσης η πιθανότητα p_{ij} θα είναι μηδέν, με αποτέλεσμα λόγω του πολλαπλασιασμού των τιμών να εξαλειφθούν και οι άλλες τιμές. Για να αποφευχθεί αυτό χρησιμοποιείται μια ψευτομέτρηση (pseudocount) έτσι ώστε ποτέ καμία πιθανότητα να μην ορίζεται μηδέν. Αυτός ο τρόπος διόρθωσης αποτελέσματος ονομάζεται εξομάλυνση Laplace (Laplace smoothing) και ο συντελεστής με τον οποίο επιτυγχάνεται ονομάζεται συντελεστής εξομάλυνσης.

Bernoulli Naïve Bayes

Όπως και στην προηγούμενη κατηγορία ταξινομητών, οι ταξινομητές αυτοί χρησιμοποιούνται ευρέως στην ταξινόμηση κειμένου και ιδιαίτερα μικρού κειμένου όπου οι διάφορες λέξεις θα εμφανίζονται ελάχιστες φορές στο ίδιο διάνυσμα. Σε αυτή την περίπτωση όμως τα χαρακτηριστικά όχι μόνο λαμβάνουν ακέραιες τιμές αλλά είναι και δυαδικά δηλαδή παίρνουν μόνο τις τιμές 0 και 1 ακολουθώντας κατανομή Bernoulli. Το διάνυσμα χαρακτηριστικών πλέον αποτελείται από δυαδικές τιμές ανάλογα με την ύπαρξη ή όχι του χαρακτηριστικού στο συγκεκριμένο πρότυπο και η πιθανότητα εμφάνισης ενός διανύσματος x στην κλάση C_i είναι : $P(x|\, C_i) = \prod_{i=1}^{n} p_{ij}^{x_i} (1 - p_{ij})^{1-x_i}$, όπου p_{ij} είναι η πιθανότητα στην κλάση C_i να εμφανιστεί το χαρακτηριστικό x_j.

Παλινδρόμηση

Παλινδρόμηση (regression) στη στατιστική μοντελοποίηση ονομάζεται ένα σύνολο διαδικασιών για την εκτίμηση των σχέσεων μεταξύ μεταβλητών. Πιο συγκεκριμένα, συνήθως ερευνάται η σχέση που έχει μια (ή περισσότερες) ανεξάρτητη μεταβλητή εισόδου με μία ή περισσότερες μεταβλητές εξόδου (εξαρτημένες μεταβλητές). Στόχος της παλινδρόμησης είναι

να εκτιμήσει την αναμενόμενη τιμή της μεταβλητής εξόδου, δεδομένων των μεταβλητών εισόδου. Το πιο απλό και το πιο διαδεδομένο μοντέλο είναι το γραμμικό, όπου η αναμενόμενη τιμή της κάθε εξόδου μοντελοποιείται ως μια γραμμική συνάρτηση ή σταθμισμένο άθροισμα των εισόδων (μιας μεταβλητής n διαστάσεων) $y_i = w_0 + w_1 x_{1i} + w_2 x_{2i} + \cdots + w_n x_{ni}$, $i = 1,2, \dots, k-1$. Οι γραμμικοί ταξινομητές έχουν το πλεονέκτημα πέραν του ότι είναι απλοί στην υλοποίησή τους, ότι δεν είναι υπολογιστικά απαιτητικοί. Βέβαια, αυτή τους η απλότητα τους αναγκάζει να περιορίζουν την αποδοτικότητά τους κατά κύριο λόγο σε περιπτώσεις που τα δεδομένα είναι γραμμικώς διαχωρίσιμα.

Στη μαθηματική θεμελίωση της παλινδρόμησης θα θεωρήσουμε ότι υπάρχουν μόνο δύο κλάσεις για ευκολία στο συμβολισμό, που όμως γενικεύεται εύκολα για k κλάσεις. Όπως, αναφέραμε η έξοδος ενός γραμμικού ταξινομητή, δοθέντος ενός διανύσματος x είναι $y = w_0 + w^T x$, όπου $w = (w_1, w_2, \dots, w_n)$ το διάνυσμα βαρών ή ισοδύναμα $y = w^T x$ επαυξάνοντας το διάνυσμα βαρών σε $w = (w_0, w_1, w_2, \dots, w_n)$ και το διάνυσμα εισόδου σε $x = (1, x_1, x_2, \dots, x_n)$.

Το πιο απλό μοντέλο της παλινδρόμησης είναι το μοντέλο **Γραμμικής Παλινδρόμησης (Linear Regression)** όπου αναζητείται η εξίσωση του υπερεπιπέδου που να ελαχιστοποιεί το μέσο τετραγωνικό σφάλμα (mean square error) μεταξύ της εκτιμώμενης και της πραγματικής τιμής της εξόδου. Σύμφωνα με τα παραπάνω αναζητείται το διάνυσμα βαρών $\hat{w} = argmin_w J(w)$, όπου $J(w)$ το μέσο τετραγωνικό σφάλμα $J(w) = E\left[||y - \hat{y}||^2\right]$ ή ισοδύναμα $J(w) = E\left[||y - w^T x||^2\right]$. Ο ταξινομητής αυτός δεν χρησιμοποιήθηκε στην παρούσα διπλωματική εργασία γιατί αντ' αυτού χρησιμοποιήθηκε το μοντέλο διανυσμάτων στήριξης (το οποίο θα παρουσιαστεί παρακάτω), το οποίο προσεγγίζει το πρόβλημα ταξινόμησης με τον ίδιο τρόπο αναζητώντας όμως βέλτιστη λύση, δηλαδή κατασκευάζοντας το υπερεπίπεδο με το βέλτιστο περιθώριο από τα δεδομένα.

Logistic Regression

Ως γνωστόν το υπερεπίπεδο απόφασης για την περίπτωση της ελάχιστης πιθανότητας σφάλματος περιγράφεται από τη σχέση $P(C_i|x) - P(C_j|x) = 0$ και επιλέγοντας τη γνησίως αύξουσα συνάρτηση ln ως συνάρτηση διάκρισης έχουμε $y_{ij} \equiv \ln P(C_i|x) - \ln P(C_j|x) =$

$\ln\frac{P(C_i|x)}{P(C_j|x)} = w_{ij}^T x$, $i = 1,2,\dots,k-1$. Τα βάρη θα πρέπει να επιλεχθούν έτσι ώστε το άθροισμα όλων των πιθανοτήτων να είναι μονάδα, δηλαδή $\sum_{i=1}^{k} P(C_i|x) = 1$. Με βάση αυτές τις δύο εξισώσεις μπορούμε να δούμε ότι οι εκ των υστέρων πιθανότητες μπορούν να μοντελοποιηθούν με τη χρήση της λογιστικής συνάρτησης (logistic function): $f(x) = \frac{1}{1+e^{-x}}$, η οποία αντιστοιχίζει διανύσματα χαρακτηριστικών x στο διάστημα [0,1] (για αυτό χρησιμοποιείται συχνά στη μοντελοποίηση πιθανοτήτων) και δανείζει και το όνομά της στη συγκεκριμένη μέθοδο ταξινόμησης. Έτσι, για την περίπτωση των δύο κλάσεων όπου έχουμε ένα υπερεπίπεδο απόφασης y και δύο πιθανά αποτελέσματα $y = 0$ και $y = 1$ έχουμε :

$$P(y = 1|x) = P(y = 1|x, w) = \frac{1}{1 + e^{-wx}}$$

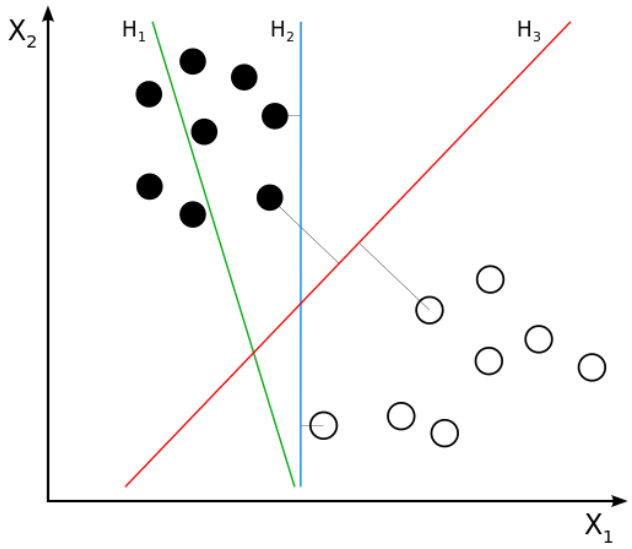
$$\kappa\alpha\iota\ P(y = 0|x) = P(y = 0|x, w) = 1 - P(y = 1|x, w) = \frac{e^{-wx}}{1 + e^{-wx}}$$

Ο υπολογισμός των βαρών είναι ένα κυρτό πρόβλημα βελτιστοποίησης όπου προσπαθούμε να ελαχιστοποιήσουμε μια συνάρτηση f η οποία εξαρτάται από τα βάρη w η οποία είναι της μορφής : $f(w) = \lambda R(w) + \frac{1}{n}\sum_{i=1}^{N} L(w; x_i, y_i)$ όπου (x_i, y_i) , $i = 1,2,..,N$ είναι το σύνολο των δειγμάτων εκπαίδευσης. Η συνάρτηση έχει δύο μέρη, τον παράγοντα ομαλοποίησης που φροντίζει για την πολυπλοκότητα του μοντέλου και τη συνάρτηση κόστους που μετράει το μέγεθος του σφάλματος στα δεδομένα εκπαίδευσης, η οποία είναι κυρτή στο w. Ο συντελεστής λ είναι υπεύθυνος για τον συμβιβασμό μεταξύ πολυπλοκότητας και μεγέθους σφάλματος. Τείνει να μειώσει την πολυπλοκότητα του μοντέλου δηλαδή την μεγάλη αύξηση των βαρών για την αποφυγή του φαινομένου της υπερεκπαίδευσης (overfitting) στα δεδομένα εκπαίδευσης. Στην λογιστική παλινδρόμηση η συνάρτηση σφάλματος είναι $L(w; x, y) = \log(1 + e^{-yw^T x})$.

Για την επίλυση αυτού του προβλήματος βελτιστοποίησης χρησιμοποιείται η μέθοδος καθόδου (Gradient Descent) ή η μέθοδος L-BFGS, η οποία προσφέρει ταχύτερη σύγκλιση.

Μηχανές Διανυσμάτων Στήριξης

Η **Μηχανή Διανυσμάτων Στήριξης (Support Vector Machine)** είναι ένας γραμμικός ταξινομητής που διαχωρίζει τα δεδομένα δύο κλάσεων. Η διαφορά τους από το γραμμικό ταξινομητή Γραμμικής Παλινδρόμησης που αναφέρθηκε πιο πάνω είναι ότι κατασκευάζει το επίπεδο διαχωρισμού με τρόπο τέτοιο, ώστε να βρίσκεται στη μεγαλύτερη δυνατή απόσταση από τα κοντινότερα σημεία εκπαίδευσης και των δύο κλάσεων με σκοπό



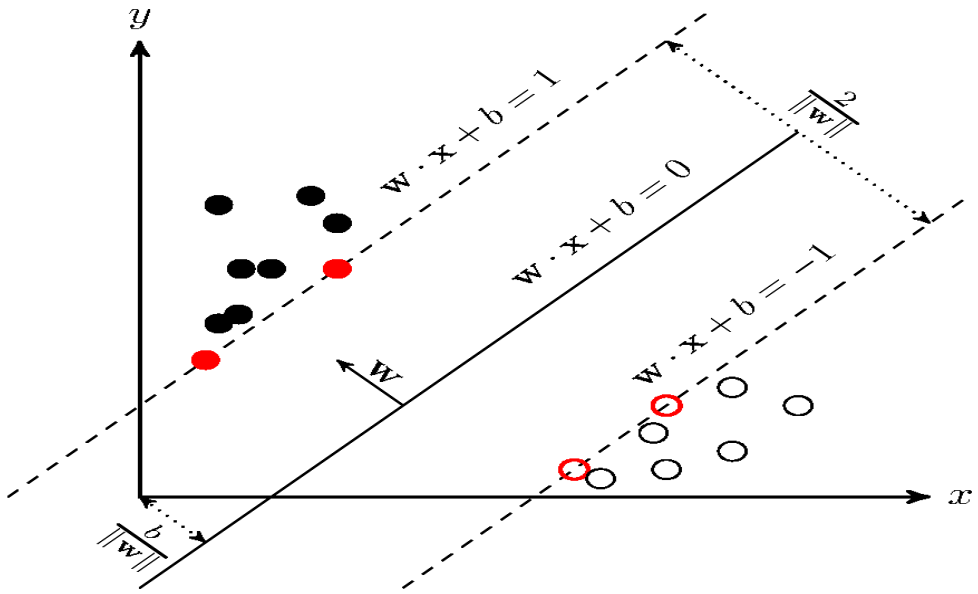
Σχήμα 3.2 : SVMs

τον βέλτιστο διαχωρισμό των κλάσεων. Δηλαδή, από τα άπειρα υπερεπίπεδα που λύνουν το πρόβλημα της ταξινόμησης γραμμικά διαχωρίσιμων σημείων επιλέγει αυτό το οποίο έχει το μεγαλύτερο περιθώριο και ως προς τις δύο κλάσεις. Παραπάνω εικονίζεται μια περίπτωση γραμμικά διαχωρίσιμων σημείων σε ένα δισδιάστατο χώρο. Η πράσινη γραμμή δεν διαχωρίζει τα σημεία, η μπλε τα διαχωρίζει αλλά όχι βέλτιστα και η κόκκινη τα διαχωρίζει αφήνοντας το μεγαλύτερο δυνατό περιθώριο μεταξύ των σημείων των κλάσεων. Αυτό, όχι μόνο επιτρέπει τον καλύτερο διαχωρισμό των σημείων, αλλά μειώνει την πιθανότητα της λάθος ταξινόμησης ενός σημείου που δεν υπάρχει στα δεδομένα εκπαίδευσης. Ταξινομητές σαν και αυτόν που διαχωρίζουν τα υπάρχοντα δεδομένα με τέτοιο τρόπο ώστε να μπορέσουν να ταξινομήσουν καλύτερα τα νέα δεδομένα λέμε ότι έχουν καλή δυνατότητα γενίκευσης, δηλαδή είναι πιο αξιόπιστοι στην ταξινόμηση άγνωστων δεδομένων. Τα πλησιέστερα σημεία και από τις δύο κλάσεις στο υπερεπίπεδο, η απόσταση των οποίων από αυτό θέλουμε να είναι μέγιστη και εικονίζεται με γκρι γραμμή, ονομάζονται διανύσματα στήριξης (support vectors) και είναι πολύ χρήσιμα στο σχεδιασμό του ταξινομητή.

Η μέθοδος των Μηχανών Διανυσματικής Στήριξης γενικεύεται και για δεδομένα που δεν είναι γραμμικά διαχωρίσιμα με τη μεταφορά τους σε ένα χώρο Hilbert υψηλότερης διάστασης στον οποίο είναι γραμμικά διαχωρίσιμα και την ταξινόμηση τους σε αυτό το χώρο ή τη χρήση μεταβλητών χαλάρωσης (slack variables). Επίσης, η μέθοδος μπορεί να

χρησιμοποιηθεί για την ταξινόμηση με περισσότερες από δύο κλάσεις με συνδυασμό πολλών SVM ταξινομητών.

Έχουμε το κλασικό πλέον πρόβλημα της ταξινόμησης σε δύο κλάσεις. Υποθέτουμε ότι έχουμε ένα σύνολο δειγμάτων εκπαίδευσης (x_i, y_i) , $i = 1,2,..,N$ για τα οποία ισχύει $y_i \in \{-1,1\}$. Στόχος είναι να βρούμε το υπερεπίπεδο $y = g(x) = w_0 + w^T x = 0$ που δίνει το μέγιστο δυνατό περιθώριο. Παρακάτω εικονίζεται το πρόβλημα στην περίπτωση των δύο διαστάσεων :



Σχήμα 3.3 : SVM στην περίπτωση των δύο κλάσεων

Τα διανύσματα στήριξης είναι τα σημεία που βρίσκονται πλησιέστερα στο επίπεδο. Όπως γνωρίζουμε η απόσταση ενός σημείου από ένα υπερεπίπεδο δίνεται από τον τύπο $z = \frac{|g(x)|}{[\![w]\!]}$. Τώρα, μπορούμε να υπολογίσουμε τα βάρη w, w_0 έτσι ώστε για αυτά να ισχύει $w_0 + w^T x = 1$ και $w_0 + w^T x = -1$ ανάλογα την κλάση. Συνεπώς για όλα τα υπόλοιπα σημεία θα ισχύει $w_0 + w^T x \geq 1$, για τη μία κλάση και $w_0 + w^T x \leq -1$ για την άλλη. Άρα τα διανύσματα στήριξης θέλουμε να έχουν απόσταση $\frac{1}{[\![w]\!]}$ το καθένα από το υπερεπίπεδο και άρα τελικά απαιτούμε να έχουμε περιθώριο $\frac{1}{[\![w]\!]} + \frac{1}{[\![w]\!]} = \frac{2}{[\![w]\!]}$ (το οποίο μεγιστοποιείται όταν ελαχιστοποιείται η νόρμα) και να ισχύει $w_0 + w^T x \geq 1$ για τη μία κλάση και $w_0 + w^T x \leq -1$

για την άλλη. Οπότε το πρόβλημα μας, είναι το εξής πρόβλημα βελτιστοποίησης ως προς ένα σύνολο περιορισμών γραμμικών ανισοτήτων:

$$\begin{cases} \text{ελαχιστοποίηση της συνάρτησης } J(w, w_0) = \frac{1}{2}\|w\|^2 \\ \text{υπό τους περιορισμούς } y_i(w_0 + w^T x_\iota) \geq 1 \ , y_i \in \{-1,1\}, i = 1,2,..,N \end{cases}$$

το πρόβλημα αυτό λύνεται κατά τα γνωστά ορίζοντας μια συνάρτηση Lagrange

$$L(w, w_0, \lambda) = \frac{1}{2} w^T w - \sum_{i=1}^{N} \lambda_\iota [y_i(w_0 + w^T x_\iota) - 1] , \qquad y_i \in \{-1,1\}, \qquad i = 1,2,..,N$$

εφαρμόζοντας τις συνθήκες Karush-Kuhn-Tucker(KKT) έχουμε :

$$\frac{\partial}{\partial w} L(w, w_0, \lambda) = 0 , \qquad \frac{\partial}{\partial w_0} L(w, w_0, \lambda) = 0 ,$$

$$\lambda_i \geq 0 , i = 1,2,\ldots,N \ \kappa\alpha\iota \ \lambda_\iota [y_i(w_0 + w^T x) - 1] = 0 \ , i = 1,2,\ldots,N$$

Συνδυάζοντας τις παραπάνω εξισώσεις προκύπτει η λύση του προβλήματος βελτιστοποίησης.

Στην περίπτωση ,όμως, που οι κλάσεις δεν είναι διαχωρίσιμες δεν μπορούμε να εφαρμόσουμε τα παραπάνω οπότε δεν μπορεί να σχεδιαστεί υπερεπίπεδο το οποίο να δημιουργεί μια ζώνη διαχωρισμού χωρίς σημεία στο εσωτερικό της, όπως γινόταν παραπάνω. Για να αντιμετωπιστεί αυτό το πρόβλημα εισάγουμε ένα νέο περιορισμό $y_i(w_0 + w^T x) \geq 1 - \xi_i$ ή αλλιώς $\xi_\iota = \max(0, 1 - y_i(w_0 + w^T x))$, όπου για $\xi_i = 0$ προκύπτουν τα σημεία που είναι σωστά ταξινομημένα, για $0 < \xi_i < 1$ τα σημεία που βρίσκονται εντός της ζώνης διαχωρισμού και είναι σωστά ταξινομημένα και για $\xi_i > 1$ τα σημεία που βρίσκονται στη ζώνη διαχωρισμού και είναι λάθος ταξινομημένα. Οι μεταβλητές ξ_i ονομάζονται μεταβλητές χαλάρωσης (slack variables) και πλέον στόχος είναι να έχουμε όσο το δυνατόν μεγαλύτερο περιθώριο κατά την ταξινόμηση, αλλά και να μειώσουμε τον αριθμό των σημείων για τα οποία ισχύει $\xi_i > 0$. Συνεπώς, το πρόβλημα πλέον γίνεται:

$$\begin{cases} \text{ελαχιστοποίηση της συνάρτησης } J(w, w_0, \xi) = \frac{1}{2}\|w\|^2 + C \sum_{\iota=1}^{N} \xi_i \\ \text{υπό τους περιορισμούς } y_i(w_0 + w^T x) \geq 1 - \xi_i , \qquad \xi_i \geq 0 , \ y_i \in \{-1,1\}, i = 1,2,..,N \end{cases}$$

Όπου ο παράγοντας C ουσιαστικά είναι η σταθερά που ελέγχει τη σχετική επιρροή δύο ανταγωνιστικών όρων, έτσι για μικρές τιμές το πρόβλημα τείνει να προσεγγίσει το προηγούμενο.

Το πρόβλημα αυτό λύνεται με τον ίδιο τρόπο ορίζοντας μια συνάρτηση Lagrange

$$L(w, w_0, \xi, \lambda, \mu) = \frac{1}{2} w^T w + C \sum_{\iota=1}^{N} \xi_i - \sum_{\iota=1}^{N} \mu_i \xi_i \ - \sum_{i=1}^{N} \lambda_\iota [y_i (w_0 + \ w^T x_\iota) - 1 + \xi_i]$$

με τις αντίστοιχες συνθήκες Karush-Kuhn-Tucker(KKT) :

$$\frac{\partial}{\partial w} L = 0 \, , \qquad \frac{\partial}{\partial w_0} L = 0 \, , \qquad \frac{\partial}{\partial \xi_i} L = 0$$

$$\mu_i \xi_i = 0 \, , \qquad \lambda_\iota [y_i (w_0 + \ w^T x_\iota) - 1 + \xi_i] = 0 \ \ \kappa\alpha\iota \ \ \mu_i \geq 0 \, , i = 1,2, \dots , N$$

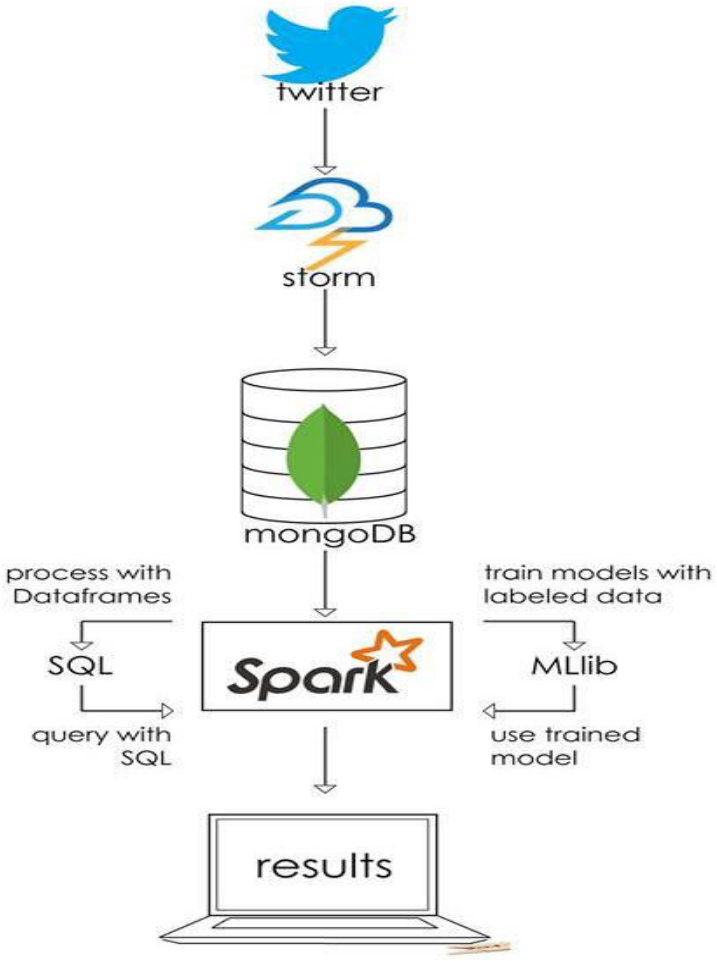
η λύση του οποίου μας δίνει τις επιθυμητές τιμές για όλες τις παραμέτρους.

Περισσότερες πληροφορίες για την θεμελίωση των ανωτέρω αλγορίθμων καθώς και για το μαθηματικό τους υπόβαθρο μπορούν να βρεθούν στα [36] και [37].

ΚΕΦΑΛΑΙΟ 4: ΥΛΟΠΟΙΗΣΗ ΤΟΥ ΣΥΣΤΗΜΑΤΟΣ

4.1 Εισαγωγή

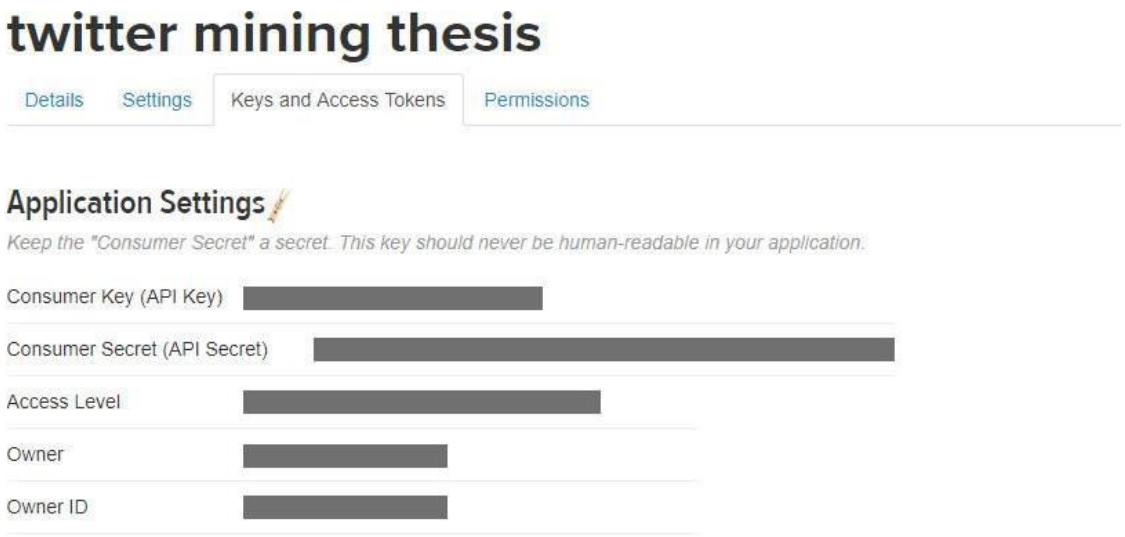
Στην παρούσα διπλωματική υλοποιήθηκε ένα σύστημα το οποίο εξάγει πραγματικού χρόνου δεδομένα από το Twitter, τα οποία έχουν γραφεί στην Αγγλική γλώσσα και με τη χρήση του Apache Storm, τα μορφοποιεί και τα διοχετεύει στη βάση δεδομένων MongoDB. Από αυτή τη βάση τα ανακτά το Apache Spark, σύμφωνα με μια ή περισσότερες λέξεις κλειδιά ή και άλλες παραμέτρους (πχ. χρόνο συγγραφής του tweet) και με βάση κάποιους προεκπαιδευμένους αλγορίθμους μηχανικής μάθησης τα επεξεργάζεται έτσι ώστε να παρέχει πληροφορία για το αν οι χρήστες εκφράζονται θετικά η αρνητικά και σε τι ποσοστό σχετικά με το επιλεγμένο θέμα-λέξη κλειδί. Στο παρακάτω γράφημα εικονίζεται η ροή δεδομένων στο σύστημά μας:



Σχήμα 4.1 : Το σύστημα που υλοποιήθηκε

4.2 Εξαγωγή Δεδομένων

Αρχικά για να είναι δυνατή η επικοινωνία με το Twitter για πρόσβαση στην πραγματικού χρόνου ροή των tweets έγινε χρήση της διεπαφής Streaming API που επιτρέπει στον προγραμματιστή να αλληλεπιδρά με τις υπηρεσίες του Twitter. Για να συμβεί αυτό πρέπει να δημιουργηθεί ένα twitter application έτσι ώστε να παραχθούν τα διαπιστευτήρια (credentials) Consumer Key, Consumer Secret, Access Token και Access Token Secret με τα οποία μπορεί ο κάθε χρήστης-προγραμματιστής να έχει πρόσβαση στις προαναφερθείσες υπηρεσίες, όπως φαίνεται στις παρακάτω εικόνες:

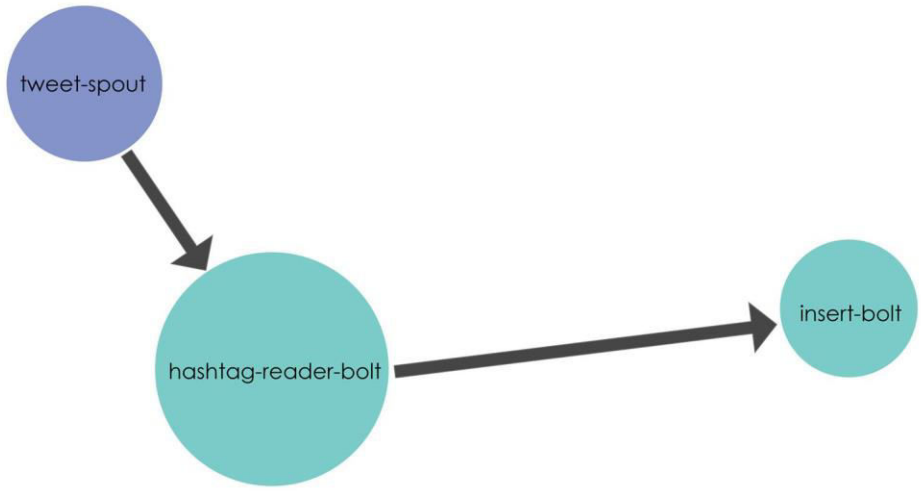


Σχήμα 4.2 : Credentials

Σχήμα 4.3 : Credentials

Για τη διαχείριση, την προεπεξεργασία και τη διοχέτευση αυτής της ροής δεδομένων από το Twitter στη βάση δεδομένων MongoDB χρησιμοποιήθηκε το εργαλείο επεξεργασίας μεγάλου όγκου δεδομένων πραγματικού χρόνου Apache Storm. Το Storm επιλέχθηκε εξαιτίας των πολύ

καλών χαρακτηριστικών του, όπως αναφέρθηκε στο κεφάλαιο 2, δηλαδή της μεγάλης αξιοπιστίας του όσον αφορά την μικρή πιθανότητα απώλειας δεδομένων, καθώς και της ταχύτητας του στην επεξεργασία μεγάλου όγκου ροών δεδομένων (streaming data). Για την υλοποίηση αυτού του έργου δημιουργήθηκε η κάτωθι τοπολογία:



Σχήμα 4.4 : Η τοπολογία του συστήματος

Η τοπολογία αυτή δείχνει τη ροή των δεδομένων και την ακολουθία βημάτων για την επεξεργασία τους. Αποτελείται από τρία μέρη : ενός Spout που επιτελεί ουσιαστικά το έργο της σύνδεσης του Apache Storm με το Twitter και κατευθύνει τη ροή των Tweets στο επόμενο βήμα επεξεργασίας, ενός Bolt που είναι υπεύθυνο για το φιλτράρισμα και την μορφοποίηση των δεδομένων, τα οποία μεταφέρει σε ένα ακόμα Bolt που αποτελεί τη σύνδεση του Storm με τη βάση δεδομένων Mongo για την αποθήκευση των δεδομένων με την καθορισμένη μορφή.

Ο ορισμός του Spout έγινε χρησιμοποιώντας τη διεπαφή TwitterSampleSpout, με τα αντίστοιχα credentials, που προσφέρει το Storm για σύνδεση με το Streaming API του Twitter ενώ η εισαγωγή του στην τοπολογία έγινε με χρήση της μεθόδου setSpout όπου ,όπως βλέπουμε, ορίστηκε και ο βαθμός παραλληλίας αυτού του Spout (η τελευταία παράμετρος της μεθόδου) :

```
TwitterSampleSpout tweetSpout = new TwitterSampleSpout(
        "Consumer Key",
        "Consumer Secret",
        "Access Token",
```

```
        "Access Token Secret"
    );
builder.setSpout("tweet-spout", tweetSpout, 1);
```

Ο ορισμός του πρώτου Bolt, όπως και κάθε ορισμός Bolt στο Storm γίνεται μέσω της διεπαφής IRichBolt η οποία περιλαμβάνει τις εξής μεθόδους : prepare που χρησιμοποιείται για την αρχικοποίηση του Bolt, execute που υποδεικνύει στο Bolt πως να επεξεργαστεί την κάθε πλειάδα tuple και ουσιαστικά αποτελεί το κυρίως σώμα του, cleanup που καλείται όταν ένας Bolt πρόκειται να τερματίσει τη λειτουργία του και declareOutputFields που καθορίζει τη μορφή που θα έχει κάθε tuple μετά από την επεξεργασία του από το Bolt. Τα βασικά σημεία του κώδικα για την υλοποίησή του φαίνονται παρακάτω :

```
public void prepare(Map conf, TopologyContext context, OutputCollector
collector) {
      this.collector = collector;

public void execute(Tuple tuple) {
      Status tweet = (Status) tuple.getValueByField("tweet");
      if (tweet.getLang().equals("en") == true ) {
            ArrayList<String> hashtags = new ArrayList<String>();
            for(HashtagEntity hashtage : tweet.getHashtagEntities()) {
                hashtags.add(hashtage.getText());
            }
            this.collector.emit(new Values(tweet.getText() , hashtags,
            tweet.getFavoriteCount(), tweet.getRetweetCount(),
            tweet.getUser().getScreenName(),
            tweet.getCreatedAt().getTime()/1000,
            tweet.getUser().getLocation(),
            tweet.getUser().getFollowersCount()));
        }

}
```

Είναι εμφανές ότι αυτό το Bolt φιλτράρει τα δεδομένα με βάση την γλώσσα συγγραφής (Αγγλικά), συλλέγει τα hashtags και μορφοποιεί κάθε Tweet με χρήση μεθόδων της βιβλιοθήκης Twitter4J. Ο ορισμός του γίνεται με χρήση της μεθόδου setBolt, όπου καθορίζεται η σύνδεση του με το Spout που γίνεται με τη μέθοδο της τυχαίας ομαδοποίησης καθώς και το επίπεδο παραλληλίας (10) όπως φαίνεται παρακάτω :

```
builder.setBolt("hashtag-reader-bolt",
new HashtagReaderBolt(),10).shuffleGrouping("tweet-spout");
```

Τέλος, χρησιμοποιείται και ένα Bolt που αποτελεί τη σύνδεση μεταξύ του προηγούμενου Bolt και της βάσης δεδομένων MongoDB αποθηκεύοντας τα δεδομένα με την μορφοποίηση και τα πεδία που έχουν οριστεί από αυτό:

```
builder.setBolt("insert-bolt", new
MongoInsertBolt("mongodb://127.0.0.1:27017/test","hashtagcount", new
SimpleMongoMapper().withFields("status", "hashtags", "favourited",
"retweeted", "name", "time", "location",
"followers")),1).shuffleGrouping("hashtag-reader-bolt");
```

4.3 Αποθήκευση Δεδομένων

Στη συνέχεια τα δεδομένα (tuples) αποθηκεύονται στη βάση δεδομένων MongoDB στη συλλογή (collection) “hashtagcount” και έχουν τα εξής πεδία :

- Status, το κείμενο του tweet αυτούσιο, χωρίς καμία επεξεργασία
- Hashtags, μια λίστα με όλα τα hashtags που αναφέρονται στο tweet
- Favourited, ο αριθμός των χριστών που έχουν κάνει favourite το συγκεκριμένο tweet
- Retweeted, ο αριθμό των χρηστών που έχουν κάνει retweet το συγκεκριμένο tweet
- Name, το όνομα του χρήστη που έχει συντάξει το συγκεκριμένο tweet
- Time, η ώρα δημιουργίας του tweet σε μορφή που καταλαβαίνει το Unix σύστημα
- Location, ο τόπος που έχει θέση ο κάθε χρήστης στο προφίλ του ως τόπος διαμονής
- Followers, ο αριθμός των followers που έχει κάθε χρήστης

Το πεδίο της τοποθεσίας που αποθηκεύεται στη βάση δεν είναι γενικά μια αξιόπιστη πηγή πληροφοριών καθώς είναι ένα πεδίο που συμπληρώνει ο ίδιος ο χρήστης με αποτέλεσμα τις περισσότερες φορές η πληροφορία να είναι ασαφής ή αναξιόπιστη. Τα μόνα tweets τα οποία φέρουν κάποιας μορφής πληροφορία για την τοποθεσία είναι τα λεγόμενα Geolocated tweets τα οποία συνδέονται με μια τοποθεσία , συνήθως αυτή της συγγραφής τους. Δυστυχώς, όμως η επιλογή είναι by-default απενεργοποιημένη και έτσι είναι ελάχιστα τα δεδομένα και όχι αρκετά για να εξάγουμε κάποια πληροφορία.

Τα δεδομένα αυτά αποθηκεύονται στη βάση δεδομένων σε μορφή BSON με σκοπό τη χρήση τους στην επεξεργασία. Καθ’ όλη τη διάρκεια της διπλωματικής συλλέχθηκαν αρκετά δεδομένα από διαφορετικές περιόδους και έγινε πολλές φορές εκκαθάριση της βάσης και ανανέωση της με καινούργια δεδομένα ή ενημέρωσή της διατηρώντας τα υπάρχοντα. Συνήθως ο όγκος των δεδομένων διατηρούνταν κοντά στα 250.000 tweets.

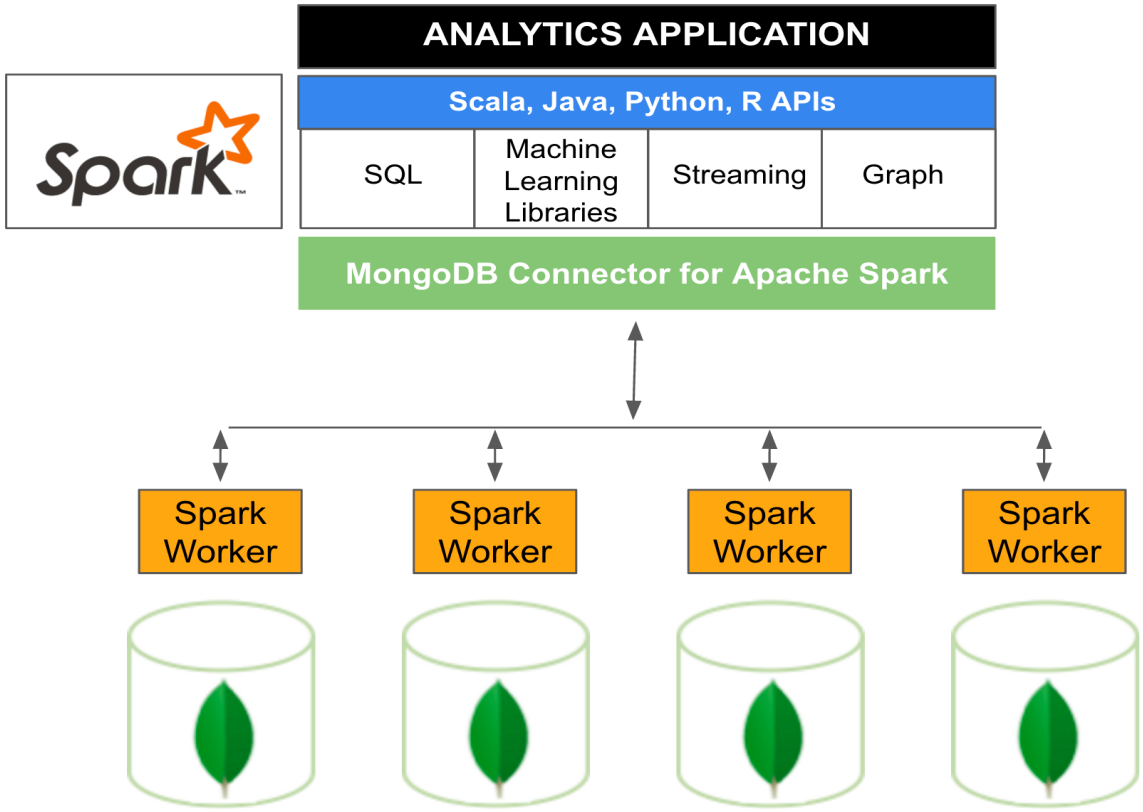
Αξίζει εδώ να αναφέρουμε ότι στην υλοποίησή μας τα δεδομένα ρέουν συνεχώς στη βάση δεδομένων από το Storm ενημερώνοντάς την, υλοποιώντας έτσι ένα σύστημα επεξεργασίας και αποθήκευσης πραγματικού χρόνου το οποίο μας επιτρέπει να κάνουμε

queries σε παλιά και νέα δεδομένα για να αποκτήσουμε μια πιο ειδική ή πιο γενική εικόνα για την γνώμη του κόσμου για οποιοδήποτε θέμα. Η σημασία της βάσης δεδομένων στο σύστημά μας είναι ακριβώς αυτή η ευελιξία που μας παρέχει για εναλλαγή μεταξύ σύγχρονης και ασύγχρονης επεξεργασίας των δεδομένων.

4.4 Επεξεργασία Δεδομένων

4.4.1 Ανάκτηση Δεδομένων

Πριν ξεκινήσει η επεξεργασία των δεδομένων πρέπει να γίνει η ανάκτηση των δεδομένων από τη βάση μέσω του Apache Spark το οποίο θα αναλάβει να τα επεξεργαστεί. Αυτό έγινε με το πακέτο *MongoDB connector for Apache Spark v2.2.0* το οποίο επιτρέπει την αλληλεπίδραση της βάσης Mongo με το Spark για την αμφίδρομη ροή δεδομένων.



Σχήμα 4.5 : Αλληλεπίδραση Spark με MongoDB

Η σύνδεση αυτή γίνεται με το πέρασμα του πακέτου ως μεταβλητή κατά την εκτέλεση του προγράμματος καθώς και με τη χρήση του αντικειμένου SparkSession από το Spark για την αλληλεπίδραση με τη βάση μας δίνοντας του ως είσοδο την κατάλληλη διεύθυνση.

Για την επεξεργασία των δεδομένων επιλέχθηκε το Spark για την απόδοσή του στην επεξεργασία μεγάλου όγκου δεδομένων, ειδικά κατά την εκτέλεση επαναληπτικών αλγορίθμων όπως είναι αυτοί της εκπαίδευσης των μοντέλων ταξινομητών που χρησιμοποιήσαμε και στους οποίους όντως παρατηρήθηκε τεράστια διαφορά από τις αντίστοιχες εκτελέσεις τους σε ένα απλό πρόγραμμα (πχ. script σε python).

Τα δεδομένα φορτώνονται στο Spark με τη μορφή Dataframes και τη χρήση του εργαλείου Spark SQL που χρησιμοποιείται για την επεξεργασία δομημένου τύπου δεδομένων όπως είναι αυτά που είναι αποθηκευμένα σε μία βάση δεδομένων. Το Spark SQL διαθέτει τις κατάλληλες βιβλιοθήκες για την αναζήτηση, την αποθήκευση και την επεξεργασία δομημένου τύπου δεδομένων. Μετά την ανάκτηση των δεδομένων από τη βάση ξεκινάει η διαδικασία της προεπεξεργασίας τους, η οποία προηγείται της ανάλυσης με τη χρήση μεθόδων μηχανικής μάθησης. Η προεπεξεργασία των δεδομένων είναι ίδια τόσο κατά τη διαδικασία της εκπαίδευσης όσο και κατά τη διαδικασία της ανάλυσης των δεδομένων και αποτελείται από επιμέρους στάδια.

4.4.2 Προεπεξεργασία δεδομένων

Φιλτράρισμα

Το πρώτο βήμα της προεπεξεργασίας έχει ως στόχο την απαλοιφή των περιττών όρων και την μορφοποίηση των δεδομένων πριν αυτά χρησιμοποιηθούν για την εξαγωγή χαρακτηριστικών. Στην περίπτωσή μας, αρχικά τα κεφαλαία γράμματα σε κάθε tweet μετατρέπονται σε πεζά, έτσι ώστε να μην υπάρχει διάκριση όρων ανάλογα με τον αν αυτοί είναι γραμμένοι σε πεζά ή κεφαλαία γράμματα. Στη συνέχεια, όλα τα links (www. , http: , https:) που υπάρχουν στο tweet ,αν υπάρχουν, αντικαθίστανται από το αρκτικόλεξο URL ενώ οι αναφορές σε κάποιον χρήστη (@username) από τη φράση AT_USER. Ακόμα, αφαιρούνται τα παραπανίσια κενά, ο χαρακτήρας της δίεσης που προηγείται των hashtags (δηλ. το #nofilter γίνεται nofilter) καθώς και τυχόν εισαγωγικά, απλά ή διπλά, ενδιάμεσα από τις λέξεις (δηλ. το ¨funny” γίνεται funny). Επίσης, αν υπάρχουν δύο ή παραπάνω συνεχόμενες εμφανίσεις ενός γράμματος σε μία λέξη,

αυτές περιορίζονται στις δύο μιας και αυτό αποδίδει απλά έμφαση (δηλ. το happyyyyy γίνεται happy). Δεν μπορούμε να τις περιορίσουμε στη μία διότι υπάρχει η πιθανότητα να καταπατηθούν κανόνες ορθογραφίας της αγγλικής γλώσσας (δηλ. το happppyyyyy θα γινόταν hapy). Επιπροσθέτως, αφαιρούνται τα περισσότερα σημεία στίξης (?.,!&*) εκτός από αυτά που μπορεί να σχηματίζουν emoticons (:()-;), τα οποία είναι πολύ χρήσιμα στην ανάλυσή μας. Τέλος, κάθε λέξη ελέγχεται σύμφωνα με μια λίστα από λέξεις που ονομάζονται stopwords που παρέχεται από τη βιβλιοθήκη NLTK. Αυτή η λίστα αποτελείται από λέξεις τις αγγλικής γλώσσας οι οποίες δεν συνεισφέρουν στο νόημα και δε φέρουν καμία απολύτως πληροφορία, όπως 'a', 'the' και σε αυτές προστίθενται επίσης οι όροι 'URL' , 'AT_USER' , 'rt' , 'via' και ΄u', ενώ ακόμα αφαιρέθηκαν και οι λέξεις που αποτελούνται από ένα και δύο γράμματα για τον ίδιο ακριβώς λόγο.

Εξαγωγή Χαρακτηριστικών

Οι ταξινομητές, όπως έχουμε αναφέρει, αποτελούν αλγορίθμους μηχανικής μάθησης που εκπαιδεύονται σε ένα σύνολο δεδομένων έτσι ώστε να μπορούν να ταξινομούν επιτυχώς νέα δεδομένα σε κλάσεις. Η ταξινόμηση αυτή μαθηματικά ισοδυναμεί με το διαμερισμό του χώρου σε περιοχές, όπου κάθε περιοχή αντιστοιχεί σε μια κλάση δεδομένων και χωρίζονται μεταξύ τους με επιφάνειες (ή υπερεπιφάνειες) απόφασης.

Για την υλοποίηση αυτού διαμερισμού είναι απαραίτητη η αναπαράσταση των αντικειμένων που επιθυμούμε να ταξινομήσουμε (εικόνες, κείμενα) σε διανύσματα ενός n-διάστατου χώρου χαρακτηριστικών. Η εργασία αυτή ονομάζεται εξαγωγή χαρακτηριστικών (feature extraction) και μπορεί να αποδειχθεί σύνθετη, ιδιαίτερα σε περιπτώσεις όπου δεν αρκεί η χρήση raw δεδομένων αλλά γίνεται προσπάθεια ανίχνευσης κρυφών δομών στα δεδομένα, κάτι το οποίο δε συμβαίνει στην ανάλυσή μας. Στην περίπτωση μας στόχος είναι η αντιστοίχιση κάθε tweet σε ένα n-διάστατο διάνυσμα χαρακτηριστικών

Στην παρούσα διπλωματική εργασία για την εξαγωγή χαρακτηριστικών χρησιμοποιήθηκε το μοντέλο Bag-of-words το οποίο αντιστοιχίζει φράσεις ή προτάσεις σε αραιά διανύσματα χαρακτηριστικών (sparse vectors) μεγάλων διαστάσεων. Το μοντέλο αυτό χρησιμοποιείται ευρέως στην επεξεργασία κειμένου και στην ταξινόμηση εγγράφων. Όπως δηλώνει και το όνομά του, αντιμετωπίζει κάθε πρόταση σαν ένα σάκο με λέξεις ο οποίος

περιλαμβάνει τις λέξεις που εμφανίζονται σε αυτήν χωρίς να ασχολείται με τη σειρά που εμφανίζονται. Είναι εμφανές ότι η προσέγγιση αυτή δεν είναι απολύτως ορθή, μιας και πολλές φορές η σειρά των λέξεων σε ένα κείμενο είναι καθοριστικής σημασίας. Παρ' όλα αυτά χρησιμοποιείται συχνά εξαιτίας της απλότητάς της και των καλών αποτελεσμάτων που δίνει στην Ανάλυση Συναισθήματος και την Μοντελοποίηση Θέματος (topic modelling).

Το πρώτο βήμα στην υλοποίηση αυτού του μοντέλου είναι ο χωρισμός του κειμένου, στην περίπτωση μας tweet, σε επιμέρους λέξεις (tokenization). Έτσι κάθε δεδομένο αντικαθίσταται πλέον από μια λίστα από λέξεις, το πλήθος των οποίων εξαρτάται από τον όγκο του κειμένου.

Το επόμενο βήμα είναι η δημιουργία ενός λεξικού (vocabulary) το οποίο αποτελείται από όλες τις λέξεις που υπάρχουν στα δεδομένα. Έπειτα, κάθε πρόταση ή φράση αναπαρίσταται σύμφωνα με το μοντέλο bag-of-words από ένα διάνυσμα ίσων διαστάσεων με το πλήθος των λέξεων του λεξιλογίου, το οποίο σε κάθε θέση λαμβάνει μια τιμή ανάλογα με την παρουσία ή τη συχνότητα της εκάστοτε λέξης στο κείμενο. Για την αποφυγή της δημιουργίας διανυσμάτων τεραστίων διαστάσεων, στην πράξη το λεξικό αυτό δεν περιέχει όλες τις λέξεις που βρίσκονται σε όλα τα δεδομένα, αλλά μόνο τους n πιο συχνά χρησιμοποιούμενους όρους από το σύνολο των εγγράφων (text corpus). Το n παίρνει τιμές ανάλογα με το πλήθος των δεδομένων εκπαίδευσης αλλά και με βάση τα αποτελέσματα τα οποία δίνει η αύξηση ή η μείωσή του.

Σε κάθε θέση του διανύσματος χαρακτηριστικών δηλαδή αντιστοιχεί μια ξεχωριστή λέξη. Έτσι το κάθε χαρακτηριστικό μπορεί να λάβει τις τιμές 1 ή 0 υποδηλώνοντας την παρουσία ή μη της λέξης στο συγκεκριμένο πρότυπο (φράση ή πρόταση) ή έναν φυσικό αριθμό που ισοδυναμεί με τον αριθμό των εμφανίσεων της λέξης στο πρότυπο. Η πρώτη προσέγγιση ονομάζεται term occurrence ενώ η δεύτερη term frequency. Παρακάτω εικονίζονται ορισμένα tweets από το σύνολο εκπαίδευσης που χρησιμοποιήσαμε, για την κατανόηση των διαφορών των δύο αυτών μεθόδων:

- *ok that's it you win.*
- *i must think about positive..*
- *i think i need a drink*
- *fed up....*

Το λεξικό σε αυτή την περίπτωση είναι το σύνολο όλων των διαφορετικών λέξεων που εμφανίζονται στα έγγραφα, δηλαδή:

V = [‘ok’, ‘that’s’, ‘it’, ‘you’, ‘win’, ‘i’, ‘must’, ‘think’, ‘about’, ‘positive’, ‘need’, ‘a’, ‘drink’, ‘fed’, ‘up’, ‘.’]

Τα διαφορετικά διανύσματα σύμφωνα με την προσέγγιση term occurrence διαμορφώνονται ως εξής:

$x_1 = [1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0,0,0,1]$

$x_2 = [0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 0, 0, 0, 0,0,0,0]$

$x_3 = [0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 1, 1, 1, 0,0,0,0]$

$x_4 = [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,1,1,1]$

Ενώ σύμφωνα με την προσέγγιση term frequency:

$x_1 = [1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0,0,0,2]$

$x_2 = [0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 0, 0, 0, 0,0,0,0]$

$x_3 = [0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 1, 1, 1, 0,0,0,0]$

$x_4 = [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,1,1,4]$

Παρατηρούμε ότι τα πρώτα τρία διανύσματα είναι ίδια και με τις δύο αναπαραστάσεις μιας και σε αυτά τα έγγραφα (tweets) δεν υπάρχει επανάληψη κάποιου όρου. Στην περίπτωση του τέταρτου εγγράφου παρατηρείται η διαφορά στον τελευταίο όρο καθώς υπάρχει επανάληψη του χαρακτήρα ‘.’ τέσσερις φορές μέσα στο έγγραφο. Στις περιπτώσεις ανάλυσης μικρών κειμένων (short text analysis) οι δύο αναπαραστάσεις δεν παράγουν μεγάλες διαφορές, όπως γίνεται εμφανές, καθώς είναι λίγες οι φορές που επαναλαμβάνεται κάποιος όρος στο έγγραφο. Η ανάλυση συναισθήματος από tweets είναι μια από αυτές τις περιπτώσεις και στην παρούσα εργασία χρησιμοποιήθηκαν και οι δύο προσεγγίσεις.

Μια εναλλακτική προσέγγιση για την εξαγωγή των χαρακτηριστικών στην ταξινόμηση κειμένου, η οποία επίσης χρησιμοποιήθηκε στην παρούσα εργασία, είναι η μέθοδος Term Frequency - Inverse Document Frequency (tf-idf). Η μέθοδος αυτή αντιστοιχίζει σε κάθε όρο-λέξη μια τιμή που ισοδυναμεί με την αξία της, η οποία τιμή αυξάνει αναλογικά με τις φορές που εμφανίζεται στο έγγραφο, αλλά η αύξηση αυτή αντισταθμίζεται από τις φορές που εμφανίζεται η λέξη αυτή στο σύνολο των εγγράφων (corpus). Έτσι, δε δίνεται αξία απλά στις λέξεις που εμφανίζονται συχνότερα σε ένα έγγραφο, αλλά σε αυτές που μεταφέρουν

πραγματική πληροφορία και δεν αποτελούν λέξεις που εμφανίζονται όλα τα έγγραφα. Για την εφαρμογή της μεθόδου αυτής υπολογίζεται κατά τα γνωστά η συχνότητα της λέξης (term frequency) $TF(t,d)$, όπου t η λέξη και d το έγγραφο, ως document frequency $DF(t,D)$ ορίζεται ο αριθμός των εμφανίσεων της λέξης t στο σύνολο των εγγράφων D ενώ ως inverse document frequency: $IDF(t,D) = \log\frac{|D|+1}{DF(t,D)+1}$, όπου $|D|$ είναι ο συνολικός αριθμός των εγγράφων ενώ ο όρος 1 που προστίθεται σε αριθμητή και παρονομαστή είναι ένας συντελεστής ομαλοποίησης για την περίπτωση όπου κάποιος όρος δεν εμφανίζεται σε κανένα έγγραφο. Τελικά, ο συντελεστής tf-idf υπολογίζεται ως εξής : $TFIDF(t,d,D) = TF(t,d) \cdot IDF(t,D)$. Είναι, εμφανές ότι η μέθοδος αυτή τείνει να μειώσει πολύ την αξία λέξεων που εμφανίζονται αρκετά συχνά στο σώμα των εγγράφων, όπως είναι οι λέξεις stopwords με αποτέλεσμα να χρησιμοποιείται και με αυτό τον σκοπό αντί της χρήσης έτοιμης λίστας λέξεων.

Στην ανάλυση των βημάτων της προεπεξεργασίας μέχρι τώρα υιοθετήθηκε η προσέγγιση ότι κάθε όρος-χαρακτηριστικό αντιστοιχεί σε μία μόνο λέξη. Ωστόσο, αυτό δεν είναι πάντα αλήθεια, καθώς υπάρχει η δυνατότητα να χρησιμοποιήσουμε ακολουθίες δύο ή και περισσότερων λέξεων σαν όρο. Οι ακολουθίες αυτές ονομάζονται n-grams, όπου το n συμβολίζει το μέγεθος κάθε ακολουθίας. Στην ανάλυση κειμένου συχνά χρησιμοποιούνται ακολουθίες δύο λέξεων, γνωστές και ως bigrams και σπανιότερα οι ακολουθίες τριών λέξεων, γνωστές και ως trigrams. Στην παρούσα διπλωματική εργασία χρησιμοποιήθηκαν ως χαρακτηριστικά απλές λέξεις ή αλλιώς unigrams καθώς και ακολουθίες των δύο λέξεων ή bigrams. Η φιλοσοφία για τη εξαγωγή χαρακτηριστικών είναι η ίδια όσον αφορά τα n-grams : κατασκευάζεται ένα λεξικό από όλα τα n-grams που εμφανίστηκαν στο σύνολο των δεδομένων και από εκεί επιλέγονται οι πιο συχνά χρησιμοποιούμενοι όροι. Στη συνέχεια, σχηματίζονται τα διανύσματα χαρακτηριστικών με παρόμοιο τρόπο, όπως με τα unigrams. Σε κάθε θέση του διανύσματος αντιστοιχεί μια τιμή που εκφράζει την παρουσία ή τη συχνότητα του n-gram στο συγκεκριμένο έγγραφο. Πρέπει να επισημανθεί ότι τα n-grams σχηματίζονται από ακολουθίες λέξεων που υπάρχουν στο κείμενο και όχι από όλες τις πιθανές ακολουθίες που θα μπορούσαν να σχηματιστούν με τις λέξεις του κειμένου σε οποιαδήποτε σειρά. Έτσι, για τα παρακάτω tweets:

- *ok that's it you win.*

- *I must think about positive..*

το λεξικό των bigrams είναι το εξής:

V = [‘ok that’s’, ‘that’s it’, ‘it you’, ‘you win’, ‘win .’, ‘I must’, ‘must think’, ‘think about’, ‘about positive’, ‘positive .’, ‘. .’]

Τα διανύσματα χαρακτηριστικών σχηματίζονται κατά τα γνωστά:

$x_1 = [1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0]$

και $x_2 = [0, 0, 0, 0, 0, 1, 1, 1, 1, 1,1,1]$

Πολλές φορές στην ανάλυση κειμένου χρησιμοποιείται συνδυασμός unigrams και bigrams ως χαρακτηριστικά για μεγαλύτερη ακρίβεια (συμπερίληψη όρων που αποτελούνται από δύο λέξεις, χειρισμός της άρνησης κ.α.).

Επιπρόσθετα, κατά την εκπαίδευση και την ταξινόμηση των δεδομένων τα δεδομένα (tweets) που εξαλείφονται εντελώς, είτε εξαιτίας της μη τήρησης των περιορισμών κατά το φιλτράρισμα ή επειδή οι λέξεις που περιείχαν δεν ανήκουν στο προεκπαιδευμένο λεξιλόγιο κατά την εξαγωγή χαρακτηριστικών (λόγω περιορισμένου μεγέθους του λεξιλογίου ή επειδή δεν είναι γραμμένες με το σωστό τρόπο) δεν λαμβάνονται υπόψιν ούτε στην εκπαίδευση ούτε στην ταξινόμηση.

Σε αυτό το σημείο κρίνεται σκόπιμο να αναφθεί ότι μερικές συναρτήσεις για τα στάδια της προεπεξεργασίας, του φιλτραρίσματος και της εξαγωγής χαρακτηριστικών δεν υπήρχαν ενσωματωμένες στη βιβλιοθήκη του Spark SQL ή δεν ήταν ακριβής υλοποίηση της προσέγγισής μας οπότε και χρειάστηκε να ορίσουμε ορισμένες User Defined Functions (UDFs). Οι UDFs είναι συναρτήσεις που ορίζει ο χρήστης για δομημένα δεδομένα, οι οποίες ναι μεν παρέχουν ελευθερία αλλά δε προτείνεται η χρήση τους, διότι η δομή δεδομένων Dataframe του Spark είναι στην ουσία μια JVM δομή και η πρόσβαση σε αυτό γίνεται με κλήσεις του API της Java. Έτσι, UDFs γραμμένες σε Python ουσιαστικά απαιτούν τη μετακίνηση δεδομένων μπρος πίσω καθώς και τη σειριοποίησή τους, σε αντίθεση με τις έτοιμες συναρτήσεις του Spark SQL οι οποίες λειτουργούν απευθείας στο JVM. Αυτό οδηγεί στη μείωση της απόδοσης του προγράμματος και για αυτό δε συστήνεται η χρήση τους. Στην περίπτωση μας, θυσιάσαμε λίγη από την απόδοση του Spark για την αύξηση της ακρίβειας των αλγορίθμων μας με καλύτερη προεπεξεργασία των δεδομένων.

4.4.3 Εκπαίδευση Μοντέλων

Η επεξεργασία των δεδομένων έγινε με χρήση προεκπαιδευμένων σε labeled δεδομένα αλγορίθμων μηχανικής μάθησης με χρήση του Apache Spark Mlib. Το Apache Spark Mlib διαθέτει τις απαραίτητες βιβλιοθήκες για το φιλτράρισμα, την προεπεξεργασία και την εκπαίδευση των μοντέλων καθώς και για την αποθήκευσή τους για τη χρήση τους αργότερα στην ταξινόμηση. Επίσης, προσφέρει μια πληθώρα μεθόδων εξαγωγής χαρακτηριστικών καθώς και αλγορίθμων μηχανικής μάθησης οι οποίοι είναι διαθέσιμοι τόσο για δομημένα δεδομένα (Dataframes) που χρησιμοποιήσαμε εμείς όσο και για RDDs. Οι αλγόριθμοι που χρησιμοποιήθηκαν είναι οι εξής τέσσερις: ο Multinomial Naive Bayes, ο Bernoulli Naive Bayes, ο Logistic Regression και ο Linear Support Vector Machine. Όλοι οι αλγόριθμοι εκπαιδεύτηκαν πάνω στα ίδια δεδομένα και έπειτα από μια διαδικασία δοκιμών αλγορίθμων και μεθόδων προεπεξεργασίας επιλέχθηκε για την ταξινόμηση των δεδομένων της βάσης ο συνδυασμός που έδωσε τα καλύτερα αποτελέσματα.

4.4.4 Ταξινόμηση δεδομένων

Τα δεδομένα μετά την ανάκτηση τους από τη βάση MongoDB φιλτράρονται σύμφωνα με τις επιλογές που δίνονται ως είσοδοι στο πρόγραμμα. Τέτοιες επιλογές είναι η αναζήτηση με βάση κάποιο θέμα ή λέξη κλειδί (ή και παραπάνω από ένα) που να αναφέρεται στο κείμενο του tweet καθώς και το χρονικό περιθώριο συγγραφής του tweet. Αφού συγκεντρωθούν τα δεδομένα που ανταποκρίνονται στις παραπάνω απαιτήσεις, αρχικά ακολουθούν τα βήματα τις προεπεξεργασίας που αναφέρθηκαν πιο πάνω και στη συνέχεια υλοποιείται η ανάλυση συναισθήματος χρησιμοποιώντας τον προεπιλεγμένο αλγόριθμο για την ταξινόμηση των δεδομένων στις δύο κατηγορίες (θετικό ή αρνητικό συναίσθημα) και παρουσιάζονται τα ποσοστιαία αποτελέσματα. Ο αλγόριθμος μηχανικής μάθησης που χρησιμοποιείται τελικά για την ανάλυση συναισθήματος των δεδομένων της βάσης που έχουν συλλεχθεί καθώς και η μέθοδος προεπεξεργασίας (το λεξιλόγιο είναι αυτό που πρέπει να έχει οριστεί από πριν για την εξαγωγή των χαρακτηριστικών) έχουν αποθηκευτεί στη φάση της εκπαίδευσης και είναι έτοιμοι για χρήση.

ΚΕΦΑΛΑΙΟ 5 : ΑΠΟΤΕΛΕΣΜΑΤΑ ΚΑΙ ΣΥΜΠΕΡΑΣΜΑΤΑ

5.1 Γενικά

Μετά από την περιγραφή και ανάλυση του συστήματός μας είμαστε σε θέση να δώσουμε την αναλυτική υλοποίησή μας για την ανάλυση συναισθημάτων βασισμένη σε κείμενο που πραγματοποιήσαμε καθώς και να παρουσιάσουμε τα αριθμητικά αποτελέσματα για τις διαφορετικές μεθόδους κατά την ταξινόμηση των δεδομένων.

5.2 Υλοποίηση

Αρχικά, για την εκπαίδευση των αλγορίθμων μηχανικής μάθησης χρησιμοποιήθηκε το σετ δεδομένων Twitter Sentiment Analysis Dataset το οποίο αποτελείται από 1,578,627 labeled δεδομένα από το Twitter τα οποία φέρουν δύο ετικέτες (labels) : 0 για αρνητικό και 1 για θετικό συναίσθημα. Το σετ δεδομένων χωριζόταν κάθε φορά σε training και test δεδομένα, με το 80% να αποτελεί τα δεδομένα εκπαίδευσης και το υπόλοιπο 20% να αποτελεί τα test δεδομένα. Ο χωρισμός αυτός έγινε με χρήση της εντολής randomSplit που επιλέγει κάθε τυχαία κάθε φορά τα δεδομένα για τον χωρισμό. Επίσης, έγινε έλεγχος των μοντέλων και με άλλα σετ δεδομένων για επαλήθευση. Όλοι οι αλγόριθμοι εκπαιδεύτηκαν στο ίδιο σύνολο δεδομένων για να υπάρχει συνοχή στα αποτελέσματα.

Το πρώτο στάδιο ήταν το φιλτράρισμα των δεδομένων εκπαίδευσης και στη συνέχεια υλοποιήθηκε η εξαγωγή χαρακτηριστικών τους σύμφωνα με τις μεθόδους που αναλύσαμε στο προηγούμενο κεφάλαιο. Επίσης, ένα σημαντικό στοιχείο στην προεπεξεργασία των δεδομένων εκπαίδευσης ήταν η επιλογή του μεγέθους του λεξιλογίου το οποίο ουσιαστικά καθορίζει το πλήθος των διαστάσεων του χώρου χαρακτηριστικών και άρα και των διανυσμάτων. Για το μέγεθος του λεξιλογίου δοκιμάστηκαν πολλές τιμές για την εύρεση της καλύτερης, βάσει αποτελεσμάτων. Τέλος, για ως όροι – χαρακτηριστικά επιλέχθηκαν λέξεις καθώς και ακολουθίες δύο λέξεων, τα γνωστά bigrams.

Για την αξιολόγηση των αλγορίθμων χρησιμοποιήθηκε ως μετρική η ακρίβεια (accuracy) κατά την ταξινόμηση του συνόλου δοκιμής (test set) η οποία ορίζεται ως εξής:

$$ακρίβεια = \frac{αριθμός\ σωστά\ ταξινομημένων\ δεδομένων\ στο\ σύνολο\ δοκιμής}{συνολικός\ αριθμός\ δεδομένων\ στο\ σύνολο\ δοκιμής}$$

Αναλυτικά, οι αλγόριθμοι που χρησιμοποιήθηκαν για την ανάλυση συναισθήματος είναι οι εξής:

Multinomial Naïve Bayes

Ο Naïve Bayes ταξινομητής με τη θεώρηση της multinomial κατανομής για τα δεδομένα. Θεωρητικά δέχεται ως είσοδο διακριτά διανύσματα αλλά στην πράξη μπορεί να εκπαιδευτεί και με συνεχή. Επίσης, υπάρχει και ένας συντελεστής εξομάλυνσης (laplace smoothing) για την αποφυγή εμφανίσεων μηδενικών πιθανοτήτων μετά το στάδιο εκπαίδευσης.

```
nb = NaiveBayes(featuresCol="features", labelCol="Sentiment", smoothing=1.0, modelType="multinomial")
```

Bernoulli Naïve Bayes

Ο ταξινομητής Naïve Bayes με τη θεώρηση, αυτή τη φορά, Bernoulli κατανομής για τα δεδομένα. Απαιτεί, όπως είναι φυσικό, διανύσματα που τα χαρακτηριστικά τους λαμβάνουν δυαδικές τιμές (0 ή 1). Ο συντελεστής εξομάλυνσης παραμένει ο ίδιος. Η οικογένεια αυτών των ταξινομητών είναι, όπως ήταν αναμενόμενο, η πιο ταχεία στην εκπαίδευση.

```
nb = NaiveBayes(featuresCol="features", labelCol="Sentiment", smoothing=1.0, modelType="bernoulli")
```

Logistic Regression

Ο ταξινομητής λογιστικής παλινδρόμησης, όπου για το συντελεστή κανονικοποίησης λ βρέθηκε ιδανική η τιμή 0,1, ενώ οι επαναλήψεις του αλγορίθμου για τη βελτιστοποίηση της συνάρτησης L(w;x,y) ορίστηκαν ως 10.

```
lr = LogisticRegression(featuresCol ="features", labelCol = "Sentiment",maxIter=10, regParam= 0.1)
```

Linear SVC

Πρόκειται για τον γραμμικό ταξινομητή SVM ο οποίος επίσης επιλέχθηκε με συντελεστή κανονικοποίησης λ = 0,1 και μέγιστο αριθμό 10 επαναλήψεων για τη σύγκληση. Παρατηρήθηκε ότι είναι ο πιο χρονοβόρος αλγόριθμος όσων αφορά την εκπαίδευση εξαιτίας της ύπαρξης πολλών δεδομένων που δημιουργούν πολλές μηχανές διανυσματικής στήριξης.

```
lsvc = LinearSVC(featuresCol ="features", labelCol = "Sentiment", maxIter=10, regParam=0.1)
```

5.3 Αποτελέσματα

Σε αυτή την ενότητα παρουσιάζονται τα αποτελέσματα από την εφαρμογή των παραπάνω αλγορίθμων. Για την εξαγωγή χαρακτηριστικών χρησιμοποιήθηκαν οι μέθοδοι της παρουσίας λέξης (term occurrence), της συχνότητας λέξης (term frequency) καθώς και η μέθοδος Term Frequency – Inverse Document Frequency (tf-idf). Πρέπει να σημειωθεί ότι κατά τη χρήση της μεθόδου tf-idf δε χρησιμοποιήθηκε λίστα για την απαλοιφή των stopwords (σε αντίθεση με τις υπόλοιπες μεθόδους), αλλά επιλέχθηκαν οι n λέξεις με το υψηλότερο σκορ για αυτή τη μέθοδο. Όπως έχουμε αναφέρει στο προηγούμενο κεφάλαιο η ίδια η φύση της μεθόδου δίνει βάρος στις σημαντικές από άποψη πληροφορία λέξεις. Σαν χαρακτηριστικά (n) επιλέχθηκαν 5,000 , 15,000, 30,000 και 40,000 λέξεις (unigrams), καθώς και 5,000, 10,000, 15,000 και 30,000 ακολουθίες των δύο λέξεων (bigrams). Συγκεντρωτικά, τα αποτελέσματα για όλους του συνδυασμούς μεθόδων, χαρακτηριστικών και αλγορίθμων φαίνονται στους παρακάτω πίνακες :

Algorithms \ Feature extraction	Term Occurrence		Term Frequency		TF-IDF	
	5,000u	*15,000u*	*5,000u*	*15,000u*	*5,000u*	*15,000u*
Multinomial Naïve Bayes	76.39	77.07	76.4	77.11	74.38	75.49
Binomial Naïve Bayes	76.58	77.21	-	-	-	-
Logistic Regression	**77.32**	77.94	77.28	77.89	76.04	77.58
Linear SVC	77.29	**77.96**	**77.31**	**78.05**	**76.1**	**77.67**

Πίνακας 5.1: Αποτελέσματα για 5,000-15,000 unigrams

Παρατηρούμε αρχικά ότι όλοι οι αλγόριθμοι έχουν συγκρίσιμες αποδόσεις, με τους Bayesian ταξινομητές να είναι οι λιγότερο ακριβής. Παρά την απλότητα και την ταχύτητα τους, η «αφελής» τους προσέγγιση όσον αφορά τη στατιστική ανεξαρτησία των λέξεων οδηγεί σε χειρότερα αποτελέσματα σε σχέση με τους άλλους αλγορίθμους. Οι αλγόριθμοι της λογιστικής παλινδρόμησης και των μηχανών διανυσματικής στήριξης έχουν εξίσου καλές αποδόσεις με τον τελευταίο να είναι οριακά καλύτερος στις περισσότερες κατηγορίες. Επίσης, οι μέθοδοι term frequency και term occurrence παρουσιάζουν καλύτερες επιδόσεις από τη μέθοδο tf-idf. Όπως, έχουμε αναφέρει και στην ανάλυση των δύο αυτών μεθόδων, στην περίπτωση της ανάλυσης μικρού κειμένου όπως είναι τα tweets, η ακρίβεια των δύο προσεγγίσεων

αναμένεται να είναι παρόμοια. Τέλος, όσο αυξάνεται η διαστατικότητα των χαρακτηριστικών, αυξάνεται και η ακρίβεια, πράγμα λογικό και αναμενόμενο.

Algorithms \ Feature extraction	Term Occurrence		Term Frequency		TF-IDF	
	30,000u	*40,000u*	*30,000u*	*40,000u*	*30,000u*	*40,000u*
Multinomial Naïve Bayes	77.21	77.3	77.27	77.38	75.62	75.51
Binomial Naïve Bayes	77.39	77.42	-	-	-	-
Logistic Regression	**78.05**	**78.01**	78.01	**78.05**	77.94	77.93
Linear SVC	77.88	77.87	**78.1**	77.96	**78.05**	**77.98**

Πίνακας 5.2: Αποτελέσματα για 30,000-40,000 unigrams

Οι προηγούμενες παρατηρήσεις ισχύουν πάνω κάτω και για την περίπτωση των μεγαλύτερων διανυσμάτων χαρακτηριστικών. Οι αλγόριθμοι Bayes συνεχίζουν να είναι αισθητά πιο ανακριβείς από τους άλλους δύο, οι οποίοι είναι πολύ κοντά σε ακρίβεια με αποτέλεσμα να μην υπάρχει ξεκάθαρος νικητής σε όλες τις κατηγορίες. Η μέθοδος tf-idf συνεχίζει να είναι η χειρότερη σε απόδοση και μάλιστα παρατηρούμε ότι πλήττεται από την αύξηση της διαστατικότητας, αποδίδοντας χειρότερα κατά την αύξηση του αριθμού των χαρακτηριστικών – λέξεων από 30,000 σε 40,000. Γενικά η περαιτέρω αύξηση των διαστάσεων (από 30,000 σε 40,000 λέξεις) δε δίνει πάντα καλύτερα αποτελέσματα, η τουλάχιστον όχι αισθητά. Συμπερασματικά, ως καλύτερη επιλογή της διάστασης για την εργασία μας θεωρούνται οι 30,000 λέξεις καθώς δεν υπάρχει λόγος αύξησης της διάστασης, η οποία επιβαρύνει το σύστημα μας σε πολυπλοκότητα και άρα σε απόδοση, ενώ τα οφέλη από άποψη ακρίβειας δεν είναι αντίστοιχα.

Στην περίπτωση των bigrams υιοθετήθηκε για την εξαγωγή χαρακτηριστικών μόνο η μέθοδος term frequency και στην περίπτωση του Bernoulli Naive Bayes η term frequency (δέχεται μόνο δυαδικά δεδομένα). Βέβαια, οι δύο αυτές μέθοδοι είναι πρακτικά ισοδύναμες, ειδικά στην περίπτωση των bigrams που έχουμε εδώ, καθώς θεωρείται σχεδόν απίθανο να εμφανιστεί η ίδια ακολουθία δύο λέξεων παραπάνω από μία φορά σε ένα μικρό κείμενο (tweet).

Algorithms	Term Frequency			
	5,000b	*10,000b*	*15,000b*	*30,000b*
Multinomial Naïve Bayes	**69.16**	**71.47**	**72.47**	**74.27**
Binomial Naïve Bayes	63.26	65.47	66.67	68.74
Logistic Regression	63.83	65.94	67.17	69.05
Linear SVC	63.7	65.26	67.02	68.96

Πίνακας 5.3: Αποτελέσματα για bigrams

Με τη χρήση των bigrams παρατηρούμε ότι ο ταξινομητής Multinomial Naïve Bayes έχει αισθητά καλύτερα αποτελέσματα από τους άλλους. Αυτό πιθανόν να οφείλεται στο γεγονός ότι η χρήση των ακολουθιών λέξεων αντισταθμίζει την «αφελή» προσέγγιση αυτού του ταξινομητή καθώς διατηρεί τη σειρά σε κάθε ακολουθία λέξεων. Παρ' όλα αυτά η απόδοση των αλγορίθμων με τη χρήση bigrams είναι πολύ χειρότερη από αυτή των ίδιων αλγορίθμων με τη χρήση απλών λέξεων. Κατά τα άλλα η αύξηση της διάστασης βελτιώνει αισθητά τα αποτελέσματα, όπως αναμένεται καθώς η τελευταία κατηγορία αποτελείται από 30,000 bigrams και όχι 40,000.

Συμπερασματικά, σύμφωνα με τα πειραματικά αποτελέσματα ο πιο αποδοτικός αλγόριθμος φαίνεται να είναι ο Linear SVC, με τον Logistic Regression να είναι λίγο χειρότερος και τους μπεϊζιανούς ταξινομητές να ακολουθούν. Τα καλύτερα αποτελέσματα παρατηρούνται κατά τη χρήση της μεθόδου term frequency για την εξαγωγή χαρακτηριστικών. Συνεπώς, το μοντέλο Bag-of-words με τη χρήση αυτών των ταξινομητών παρέχει αξιόπιστα αποτελέσματα γεγονός που επιβεβαιώνεται και από άλλες ερευνητικές εργασίες όπως αυτή των Pang και Lee [38] καθώς και πιο πρόσφατες όπως αυτή των Wang και Manning [39].Αυτός ο συνδυασμός (Linear SVC και term frequency) ήταν και αυτός που τελικά χρησιμοποιήθηκε στο σύστημά μας για την ταξινόμηση των δεδομένων μας. Ένας ακόμα λόγος που συντέλεσε στην επιλογή αυτή, πέραν των καλών πειραματικών αποτελεσμάτων είναι η καλή ικανότητα γενίκευσης που έχει αυτός ο αλγόριθμος για δεδομένα που δεν εμφανίστηκαν στο σετ εκπαίδευσης, όπως είναι πολύ πιθανό να συμβαίνει με τα δεδομένα που λαμβάνει το σύστημά μας.

5.4 Σύνοψη - Συμπεράσματα

Στην παρούσα διπλωματική εργασία υλοποιήθηκε ο σχεδιασμός ενός συστήματος που ως σκοπό έχει την εξόρυξη γνώσης από μεγάλο όγκο δεδομένων σε πραγματικό χρόνο. Η εξόρυξη γνώσης αφορά στην ανάλυση συναισθήματος δεδομένων που προέρχονται από την πλατφόρμα του Twitter. Στόχος της διπλωματικής ήταν η σχεδίαση ενός αποδοτικού, γρήγορου και επεκτάσιμου συστήματος που θα ταξινομεί τα δεδομένα με βάσει το συναίσθημα που εκφράζουν σε πραγματικό χρόνο. Συγκεκριμένα, το σύστημα έχει τη δυνατότητα με τον καθορισμό μιας λέξης-κλειδιού από το χρήστη να συγκεντρώσει τα δεδομένα στα οποία αναφέρεται αυτή η λέξη ή λέξεις, να τα ταξινομήσει με βάση το συναίσθημα και να παρουσιάσει τα ποσοστιαία αποτελέσματα για τα διάφορα συναισθήματα. Η διαφορά με προηγούμενες διπλωματικές εργασίες πάνω στον ίδιο τομέα είναι η σχεδίαση αυτού του ενιαίου συστήματος το οποίο επιτελεί όλες τις λειτουργίες από την εξόρυξη των δεδομένων μέχρι την εξαγωγή των αποτελεσμάτων.

Για την εξόρυξη των δεδομένων επιλέχθηκε το εργαλείο Apache Storm, το οποίο είναι πολύ αποδοτικό στην επεξεργασία ροών δεδομένων (streaming data) ενώ για την επεξεργασία των δεδομένων και την υλοποίηση της ανάλυσης συναισθήματος χρησιμοποιήθηκε το εργαλείο Apache Spark το οποίο είναι εξαιρετικά γρήγορο στην επεξεργασία μεγάλου όγκου δεδομένων και προσφέρει τη δυνατότητα εφαρμογής αλγορίθμων μηχανικής μάθησης. Δυστυχώς, λόγω του περιορισμού στον όγκο των δεδομένων (ο χρήστης έχει πρόσβαση μόνο στο 1% του πραγματικού όγκου δεδομένων του Twitter) δεν έγιναν εμφανείς οι πραγματικές δυνατότητες των δύο παραπάνω εργαλείων, καθώς τα δεδομένα ναι μεν ήταν πολλά, αλλά δεν ανταποκρίνονται στα πραγματικά δεδομένα που μπορεί να επεξεργαστεί αυτό το σύστημα.

Κατά την υλοποίηση του συστήματος δοκιμάστηκαν και συγκρίθηκαν διάφορες μέθοδοι για την προεπεξεργασία των δεδομένων καθώς και μια σειρά αλγορίθμων μηχανικής μάθησης για την ανάλυση συναισθήματος. Στην προηγούμενη ενότητα παρουσιάστηκαν τα αποτελέσματα που προέκυψαν από τη χρήση αυτών των αλγορίθμων που αναλύθηκαν στο κεφάλαιο 3, για την υλοποίηση αυτού του συστήματος που παρουσιάστηκε στο κεφάλαιο 4. Συνοψίζοντας τις παραπάνω ενότητες καταλήξαμε σε διάφορα συμπεράσματα που προέκυψαν κατά το σχεδιασμό του συστήματος:

- Το Apache Storm είναι πιο κατάλληλο για την εξόρυξη των δεδομένων σε εφαρμογές όπου απαιτείται μικρή καθυστέρηση και ανοχή στα σφάλματα.
- Το Apache Spark είναι ένα εργαλείο που συνδυάζει ιδανικά την επεξεργασία μεγάλου όγκου δεδομένων και μηχανικής μάθησης, για αυτό το λόγο χρησιμοποιείται ήδη ευρέως στη βιομηχανία, καθώς η επιτάχυνση που προσφέρει στην εκπαίδευση και ανάλυση των δεδομένων είναι σημαντική.
- Οι διάφοροι αλγόριθμοι που χρησιμοποιήθηκαν για την ανάλυση συναισθήματος παρουσιάζουν ακρίβεια 75% - 78% στο σετ εκπαίδευσης.
- Η προεπεξεργασία των δεδομένων, όπως φαίνεται και από τα αποτελέσματα είναι μια ιδιαίτερα σημαντική εργασία και ο τρόπος υλοποίησής της επηρεάζει κατά πολύ την ποιότητα των εξαγόμενων συμπερασμάτων.
- Το μοντέλο Bag-of-words δουλεύει ικανοποιητικά και η υλοποίηση του είναι ιδιαίτερα απλή και γρήγορη. Το κατάλληλο μέγεθος του λεξιλογίου για αυτή την περίπτωση αποτελεί έναν συμβιβασμό ακρίβειας και πολυπλοκότητας. Επίσης, η απόδοση του μοντέλου δε φαίνεται να επηρεάζεται θετικά, τουλάχιστον στη δική μας περίπτωση από τη χρήση bigrams,
- Οι bayesian ταξινομητές πέρα από την απλότητά τους παρέχουν επαρκώς ακριβή αποτελέσματα για αυτό το λόγο και χρησιμοποιούνται κατά κόρον στην ανάλυση κειμένου.
- Οι μηχανές διανυσματικής στήριξης πετυχαίνουν μαζί με τον αλγόριθμο Logistic Regression, στο πείραμά μας, τα καλύτερα αποτελέσματα.

5.5 Μελλοντικοί προσανατολισμοί έρευνας

Το συγκεκριμένο πεδίο έρευνας γίνεται ολοένα και πιο δημοφιλές, καθώς ο όγκος των δεδομένων αυξάνει με την πάροδο του χρόνου, ενώ και ο τομέας της μηχανικής μάθησης γνωρίζει μεγάλη άνθηση τα τελευταία χρόνια. Στόχος, μελλοντικών εργασιών θα μπορούσε να είναι η βελτίωση του συστήματος από άποψη ακρίβειας, για την βελτίωση των παραγόμενων αποτελεσμάτων καθώς και βελτιστοποίηση του από άποψη απόδοσης με τη χρήση συστάδων υπολογιστών για την πλήρη αξιοποίηση των δυνατοτήτων του.

Θετική θα ήταν η συμπερίληψη και των ελληνικών tweets για την ανάλυση συναισθήματος του ελληνικού Twitter και την εξαγωγή συμπερασμάτων για την ελληνική πραγματικότητα. Αυτό το εγχείρημα απαιτεί τη δημιουργία συνόλου εκπαίδευσης που θα αποτελείται από ελληνικά tweets τα οποία πρέπει να χαρακτηριστούν ως προς το συναίσθημα τους, μια εργασία που απαιτεί πολύ χρόνο. Επιπροσθέτως, για την επεξεργασία των ελληνικών tweets πρέπει να ληφθεί υπόψιν η χρήση greeklish και να χρησιμοποιηθεί μια μέθοδος που να τα συμπεριλαμβάνει.

Επίσης, η πληροφορία που μας είναι διαθέσιμη μέσω του twitter θα μπορούσε να αξιοποιηθεί καλύτερα με τη χρήση στοιχείων όπως ο αριθμός των followers ή οι φορές που έγινε favorite ένα tweet, ακόμη και η τοποθεσία, πληροφορία την οποία προσπαθήσαμε να χρησιμοποιήσουμε και σε αυτή την εργασία αλλά τα δεδομένα ήταν ανεπαρκή. Με αυτό τον τρόπο θα μπορούσαμε να παράγουμε πληθώρα αποτελεσμάτων όσον αφορά την ανάλυση, καθώς επίσης και να βελτιώσουμε την εξαγωγή συμπερασμάτων με τη χρήση βαρών για λογαριασμούς οι οποίοι φαίνεται να επηρεάζουν περισσότερο την κοινή γνώμη, στα πλαίσια του Twitter.

Η χρήση αλγορίθμων εντοπισμού γεγονότων (event detection) θα ήταν μια λογική επέκταση αυτής της εργασίας για τη δημιουργία ενός συστήματος που όχι μόνο θα αναλύει τη γνώμη του κόσμου για γεγονότα και θέματα που υπάρχουν ήδη στην επικαιρότητα, αλλά θα είναι αυτό που θα τα εντοπίζει και θα παρέχει πληροφορία σχετικά με έκτακτες ειδήσεις, όπως ακραία καιρικά φαινόμενα, τρομοκρατικά χτυπήματα, ψήφιση νομοσχεδίων κ.α. Έτσι, το σύστημα θα μπορούσε πλήρως αυτοματοποιημένα να ανακαλύπτει ειδήσεις καθώς και τη γνώμη του κόσμου για αυτές.

Όσον αφορά την ακρίβεια η κυρίαρχη ερευνητική τάση στον τομέα της μηχανικής μάθησης αυτή τη στιγμή είναι η λεγόμενη βαθιά μάθηση (deep learning), η οποία χρησιμοποιείται ήδη σε τομείς όπως η όραση υπολογιστών, η τεχνητή νοημοσύνη και η επεξεργασία φυσικής γλώσσας. Η τάση αυτή ουσιαστικά περιλαμβάνει τη χρήση πολύπλοκων νευρωνικών δικτύων με το μοντέλο αναπαράστασης λέξεων σε χώρους μικρής διάστασης word2vec. Πρόκειται για ένα νευρωνικό γλωσσικό μοντέλο το οποίο προτάθηκε το 2013 από τους Mikolov et al. [40] και χρησιμοποιείται ήδη από τους ερευνητές εξαιτίας των πολύ καλών αποτελεσμάτων που παράγει. Το μοντέλο αυτό προκύπτει από ένα «ρηχό» νευρωνικό δίκτυο δύο επιπέδων και έχει τη δυνατότητα να ανακαλύπτει σημασιολογικές και συντακτικές σχέσεις μεταξύ των λέξεων και να τις απεικονίζει ως γραμμικές ιδιότητες ενός διανυσματικού χώρου. Με αυτό τον τρόπο λέξεις που έχουν παρόμοιο νόημα θα απεικονίζονται σχετικά κοντά στο διανυσματικό χώρο.

Τα νευρωνικά δίκτυα τα οποία έχουν γνωρίσει μεγάλη άνθιση τα τελευταία χρόνια επίσης μπορούν να συμβάλλουν στη βελτίωση της ακρίβειας συστημάτων όπως αυτό που σχεδιάστηκε στα πλαίσια της παρούσας εργασίας. Τέτοια νευρωνικά δίκτυα θα μπορούσαν να είναι τα συνελικτικά δίκτυα τα οποία εμπνεύστηκαν από τον τρόπο και τα στάδια επεξεργασίας πληροφοριών στον οπτικό φλοιό οργανισμών. Τα δίκτυα αυτά χρησιμοποιούνται κυρίως στην επεξεργασία εικόνας και αποτελούνται από πολλά κρυφά επίπεδα καθένα από τα οποία προσπαθεί να ανακαλύψει μια πιο πολύπλοκη δομή στην εικόνα από το προηγούμενο, ξεκινώντας από τις πιο απλές όπως ακμές και καταλήγοντας σε κάτι πιο σύνθετο. Νευρωνικά δίκτυα που είναι αρκετά δημοφιλή στην επεξεργασία φυσικής γλώσσας είναι τα αναδρομικά νευρωνικά δίκυτα (recursive neural networks) καθώς και τα δίκτυα με επανατροφοδότηση (recurrent neural networks). Οι δύο αυτοί τύποι νευρωνικών δικτύων χρησιμοποιούν συνδέσεις προς τα προηγούμενα επίπεδα νευρώνων οι οποίοι νευρώνες διαθέτουν κάποιο είδος μνήμης (στην πραγματικότητα είναι μια εσωτερική κατάσταση που τους επιτρέπει να επεξεργάζονται δεδομένα με ακολουθιακή συσχέτιση όπως είναι η φυσική γλώσσα.

Ακόμα για τη βελτίωση της ανάλυσης συναισθήματος θα ήταν χρήσιμη η κατηγοριοποίηση σε περισσότερες από δύο συναισθηματικές καταστάσεις, έτσι ώστε να έχουμε μια πιο σαφή εικόνα για τα συναισθήματα που εκφράζονται σε ένα tweet. Επίσης,

πέρα από τη χρήση περισσότερων κατηγοριών για τη διακύμανση των συναισθημάτων, θα μπορούσαν να οριστούν πιο συγκεκριμένες κατηγορίες που αφορούν την έκφραση συναισθήματος όπως, ειρωνεία, σαρκασμός, οργή κ.α. Ακόμα, ενδιαφέρον θα είχε η ενσωμάτωση της υποκειμενικότητας κατά την ανάλυση συναισθήματος όπως έκαναν οι Pang και Lee στην εργασία τους [41].

Συμπερασματικά, μπορούμε να πούμε ότι αυτή η εργασία μπορεί να αναπτυχθεί και να επεκταθεί και σε άλλους τομείς με τη χρήση βελτιωμένων τεχνικών και τη συμπερίληψη περισσότερων παραγόντων στην εξαγωγή συμπερασμάτων. Βεβαίως, το πρόβλημα αυτό είναι δυσκολότερο και απαιτεί το συνδυασμό πολλών ερευνητικών τομέων.

Βιβλιογραφία

[1] «Omnicore,» [Ηλεκτρονικό]. Available: https://www.omnicoreagency.com/twitter-statistics/.

[2] «Twitter Developer,» [Ηλεκτρονικό]. Available: https://developer.twitter.com/en/docs.

[3] «Python,» [Ηλεκτρονικό]. Available: https://www.python.org/doc/.

[4] «NLTK 3.2.5 documentation,» [Ηλεκτρονικό]. Available: http://www.nltk.org/.

[5] «Twitter4J,» [Ηλεκτρονικό]. Available: http://twitter4j.org/en/index.html.

[6] «Apache Maven,» [Ηλεκτρονικό]. Available: https://maven.apache.org/.

[7] «Tutorialspoint,» [Ηλεκτρονικό]. Available: https://www.tutorialspoint.com/mongodb/mongodb_advantages.htm.

[8] «Apache Storm,» [Ηλεκτρονικό]. Available: http://storm.apache.org/releases/1.1.0/Tutorial.html .

[9] «Tutorialspoint,» [Ηλεκτρονικό]. Available: http://www.tutorialspoint.com/apache_storm/.

[10] «Tutorialspoint,» [Ηλεκτρονικό]. Available: https://www.tutorialspoint.com/apache_spark/apache_spark_introduction.htm.

[11] «Apache Spark,» [Ηλεκτρονικό]. Available: https://spark.apache.org/docs/latest/quick-start.html.

[12] «Databricks,» [Ηλεκτρονικό]. Available: https://databricks.com/blog/2016/07/14/a-tale-of-three-apache-spark-apis-rdds-dataframes-and-datasets.html.

[13] «Ericcson Research Blog,» [Ηλεκτρονικό]. Available: https://www.ericsson.com/research-blog/apache-storm-vs-spark-streaming/.

[14] «Hortonworks,» [Ηλεκτρονικό]. Available: https://hortonworks.com/blog/microbenchmarking-storm-1-0-performance/.

[15] «Xinh's Tech Blog,» [Ηλεκτρονικό]. Available: http://xinhstechblog.blogspot.gr/2014/06/storm-vs-spark-streaming-side-by-side.html.

[16] «Data Flair,» [Ηλεκτρονικό]. Available: http://data-flair.training/blogs/apache-storm-vs-spark-streaming/.

[17] «Linkedin,» [Ηλεκτρονικό]. Available: https://www.linkedin.com/pulse/comprehensive-analysis-data-processing-part-deux-apache-fazelat/.

[18] «Langrangian Points Blog,» [Ηλεκτρονικό]. Available: http://lagrangianpoints.com/2016/02/moving-from-apache-storm-to-apach-spark-streaming/6/.

[19] M. Kantardzic, Data Mining: Concepts, Models, Methods, and Algorithms, 2nd Edition, Wiley-IEEE Press, 2011.

[20] U. Fayyad, G. Piatetsky-Shapiro και P. Smyth, «From Data Mining to Knowledge Discovery in Databases,» *AI Magazine,* 1996.

[21] «English Oxford Living Dictionaries,» [Ηλεκτρονικό]. Available: https://en.oxforddictionaries.com/definition/sentiment_analysis.

[22] «Linkedin,» [Ηλεκτρονικό]. Available: https://www.linkedin.com/pulse/importance-sentiment-analysis-social-media-christine-day/.

[23] P. Patil και P. Yalagi, «Sentiment Analysis Levels and Techniques: A Survey,» *International Journal of Innovations in Engineering and Technology (IJIET),* 2016.

[24] B. Pang και L. Lee, «Opinion Mining and Sentiment Analysis,» *Foundations and Trends in Information Retrieval archive,* τόμ. 2, αρ. 1-2, 2008.

[25] V. Hatzivassiloglou και K. R. McKeown, «Predicting the semantic orientation of adjectives,» σε *ACL '98 Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics and Eighth Conference of the European Chapter of the Association for Computational Linguistics*, Madrid, 1997.

[26] A. Esuli και F. Sebastiani, «SentiWordNet: A High-Coverage Lexical Resource,» σε *5th Conference on Language Resources and Evaluation* , 2006.

[27] S.-M. Kim και E. Hovy, «Determining the sentiment of opinions,» σε *COLING '04 Proceedings of the 20th international conference on Computational Linguistics*, Geneva, 2004.

[28] T. Wilson, J. Wiebe και P. Hoffman, «Recognizing contextual polarity in phrase-level sentiment analysis,» σε *HLT '05 Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, Vancouver, 2005.

[29] A. Pak και P. Paroubek, «Twitter based system: Using Twitter for disambiguating sentiment ambiguous adjectives,» σε *SemEval '10 Proceedings of the 5th International Workshop on Semantic Evaluation*, Los Angeles, 2010.

[30] J. Eisenstein, B. O'Connor, N. A. Smith και E. P. Xing, «A latent variable model for geographic lexical variation,» σε *EMNLP '10 Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, Cambridge, 2010.

[31] A. Tumasjan, T. O. Sprenger, P. G. Sandner και I. M. Welpe, «Predicting Elections with Twitter: What 140 Characters Reveal about Political Sentiment,» σε *Fourth International AAAI Conference on Weblogs and Social Media*, 2010.

[32] A. Bifet και E. Frank, «Sentiment knowledge discovery in twitter streaming data,» σε *DS'10 Proceedings of the 13th international conference on Discovery science*, Canberra, 2010.

[33] L. Barbosa και J. Feng, «Robust sentiment detection on Twitter from biased and noisy data,» σε *COLING '10 Proceedings of the 23rd International Conference on Computational Linguistics: Posters*, Beijing, 2010.

[34] D. Davidov, O. Tsur και A. Rappoport, «Semi-supervised recognition of sarcastic sentences in Twitter and Amazon,» σε *CoNLL '10 Proceedings of the Fourteenth Conference on Computational Natural Language Learning*, Uppsala, 2010.

[35] «Medium,» [Ηλεκτρονικό]. Available: https://medium.com/towards-data-science/types-of-machine-learning-algorithms-you-should-know-953a08248861.

[36] S. Theodoridis και K. Koutroumpas, Pattern Recognition, 2008: Academic Press.

[37] C. M. Bishop, Pattern Recognition and Machine Learning, Springer, 2006.

[38] B. Pang και L. Lee, «Thumbs up? Sentiment Classification using Machine Learning,» *Proceedings of the ACL-02 conference on Empirical methods in natural language processing,* τόμ. 10, 2002.

[39] S. Wang και C. D. Manning, «Baselines and Bigrams: Simple, Good Sentiment and Topic Classification,» 2013.

[40] T. Mikolov, K. Chen, G. Corrado και J. Dean, «Efficient Estimation of Word Representations in Vector Space,» 2013.

[41] B. Pang και L. Lee, «A Sentimental Education: Sentiment Analysis Using Subjectivity,» 2004.

[42] M. Bonzanini, Mastering Social Media Mining with Python, Pack Publishing, 2016.

[43] M. Zaharia, H. Karau, A. Konwinski και P. Wendell, Learning Spark; Lightning-Fast Big Data Analysis, O'Reilly Media, 2015.