ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΑΤΡΩΝ
UNIVERSITY OF PATRAS

Πολυτεχνική Σχολή
Τμήμα Μηχανικών Η/Υ & Πληροφορικής

Διπλωματική Εργασία

Συναισθηματική Ανάλυση σε Στίχους και Μουσική χρησιμοποιώντας Μηχανική Μάθηση

Νικόλαος Μαυροφόρος
Α.Μ. 4770
Επιβλέποντες
Ιωάννης Χατζηλυγερούδης, Ισίδωρος Περίκος

Πάτρα, 2019

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΑΤΡΩΝ
UNIVERSITY OF PATRAS

School of Engineering
Computer Engineering and Informatics Department

Diploma Thesis

Sentiment Analysis on Lyrics and Music using Machine Learning

---

Nikolaos Mavroforos
A.M. 4770

Patras, 2019

Ευχαριστίες

Θα ήθελα να ευχαριστήσω την οικογένεια μου,
τον επιβλέποντα κ. Ισίδωρο Περίκο και τον καθηγητή μου κ. Ιωάννη Χατζηλυγερούδη.

ΠΡΟΛΟΓΟΣ

Οι άνθρωποι έχουν την ικανότητα να σκέφτονται, να αναλύουν και να παίρνουν αποφάσεις με βάση τη λογική. Ταυτόχρονα όμως, οι άνθρωποι μπορούν να επηρεαστούν συναισθηματικά από τα πάντα, θετικά ή αρνητικά. Η μελέτη των συναισθημάτων αυτών αναλύεται υπό το πρίσμα της Μηχανικής Μάθησης.

Από την αρχή της, η ανθρωπότητα χρησιμοποιούσε ήχους, σαν πρωτόλεια μορφή επικοινωνίας. Η επικοινωνία αυτή, για να γίνει βασικό εξελικτικό εργαλείο του ανθρώπου έπρεπε να αποκοπεί σε μεγάλο βαθμό από την συναισθηματική πληροφορία που έφερνε (όπως τόνος, ταχύτητα έκφρασης κτλ) και να διαφοροποιηθεί αυστηρά από τις υποκειμενικές της προσκείμενες. Με τον τρόπο αυτό, αυτή η πρωτόγονη μορφή επικοινωνίας εξελίχθηκε στη μορφή της γλώσσας που καταλαβαίνουμε σήμερα. Αντίθετα, η μορφή επικοινωνίας που διατηρούσε τη συναισθηματική της βάση, εξελίχθηκε στη μουσική [1]. Μπορούμε συνεπώς να αντιληφθούμε ότι η μουσική, με την έννοια της δομημένης ανθρώπινης δημιουργίας, υπάρχει και επιβίωσε για να καλύψει και να συμπληρώσει ψυχοσυναισθηματικές διαστάσεις της ανθρώπινης ύπαρξης. Έτσι, η μουσική είναι μία από τις κυρίαρχες μορφές τέχνης που εξερευνούν και αποτυπώνουν το ανθρώπινο συναίσθημα. Η μουσική αναφέρεται πολλές φορές ως η «γλώσσα του συναισθήματος» [2], και είναι φυσικό για εμάς να κατηγοριοποιούμε τη μουσική με βάση τις συναισθηματικές διασυνδέσεις που κάνουμε.

Διαφορετικά μουσικά δομικά στοιχεία, όπως η μελωδία, ο ρυθμός, η αρμονία, επιδρούν διαφορετικά στο συναίσθημα. Αν και το ερέθισμα της μουσικής έχει αδιαμφισβήτητα κοινωνική και υλιστική διάσταση, διαφορετικοί λαοί σε διαφορετικούς χρόνους χρησιμοποίησαν τη μουσική, όπως ακριβώς χρησιμοποιούσαν τη γλώσσα, για να καλύψουν συγκεκριμένες ανάγκες. Μελαγχολία, χαρά, νοσταλγία, φόβος, οργή είναι μόνο από τα μερικά συναισθήματα που μπορούν να μας δημιουργηθούν, ακούγοντας τη μουσική και αποκωδικοποιώντας τους στίχους. Η γέννηση των συναισθημάτων και η διαφορετικότητα της, γνωρίζουμε πως οφείλεται στην μουσική καθαυτή, στον άνθρωπο που την ακούει και στο περιβάλλον όπου βρίσκεται [3].

Στην ιστορία της μελέτης της μουσικολογίας, κάθε έρευνα περιοριζόταν από την έλλειψη διαθέσιμων δεδομένων. Πλέον, με το κόστος ηχογράφησης μουσικής να είναι πολύ προσιτό, και με την απομάκρυνση από την ανάγκη αποθήκευσης σε φυσικά μέσα, έχουμε τεράστιο όγκο μουσικής και στιχουργικής πληροφορίας διαθέσιμο για μελέτη και ανάλυση.
Για το λόγο αυτό, λοιπόν, τα τελευταία χρόνια έχει ενταθεί η έρευνα και μελέτη στη κατηγοριοποίηση σε όλα τα συστατικά στοιχεία της μουσικής: Την αρμονία, τον ρυθμό και τους στίχους. Τέτοια μοντέλα ταξινόμησης χρησιμοποιούνται ήδη στη βιομηχανία της μουσικής. Τεράστιες πλατφόρμες streaming αξιοποιούν στιχουργική και μουσική συναισθηματική ανάλυση χρησιμοποιώντας Αλγόριθμους Μάθησης, σαν αυτούς που παρουσιάζουμε στην εργασία αυτή. Σκοπός τους είναι η αναγνώριση της συναισθηματικής κατάστασης της μουσικής που ακούει ο ακροατής, ώστε να προτείνουν παραπλήσια μουσική, αλλά και για να ανοίξει έναν τρομακτικά εύφορο χώρο για διαφημίσεις.
Με βάση έρευνες που έχουν γίνει με μαγνητικούς τομογράφους η γλωσσική πληροφορία και ο ήχος επεξεργάζονται από ανεξάρτητα μέρη του εγκεφάλου, ακόμα και αν οι πηγές αυτές είναι συναφείς [4]. Συνεπώς, αναλύοντας ξεχωριστά τους στίχους

και τη μουσική, μπορούμε να πάρουμε σημαντικές πληροφορίες για το τραγούδι, όπως το είδος, το συναίσθημα και το θέμα του τραγουδιού.

Στην εργασία αυτή, εξερευνούμε διαφορετικούς τρόπους στην συναισθηματική ανάλυση της μουσικής, βασιζόμενοι σε κειμενική πληροφορία μέσω των στίχων, καθώς και σε μουσικά χαρακτηριστικά, ξεχωριστά. Στόχος είναι να βρούμε ποιες λέξεις και ποια μουσικά στοιχεία είναι τα κατάλληλα ώστε να δημιουργήσουμε αποτελεσματικά μοντέλα ταξινόμησης ως προς τον αναγνώριση συναισθήματος από τη μουσική και το στίχο, ως θετικό ή αρνητικό συναίσθημα

ΠΕΡΙΕΧΟΜΕΝΑ

Λίστα Εικόνων

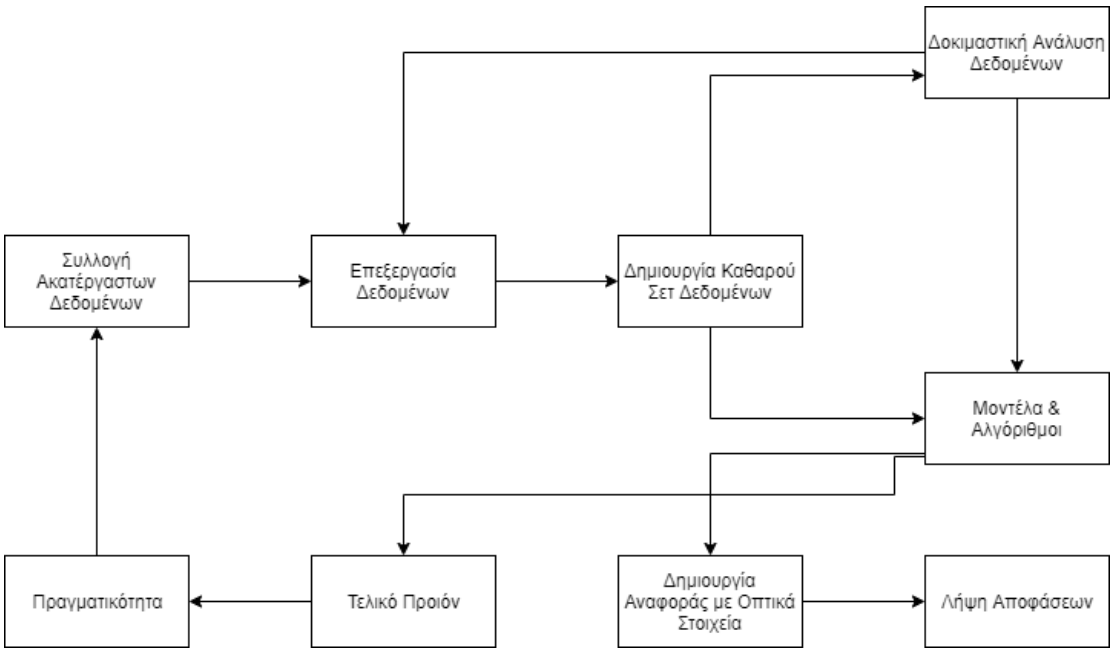
Λίστα Πινάκων

.

ΚΕΦΑΛΑΙΟ 1. ΕΙΣΑΓΩΓΗ ΣΤΗΝ ΣΥΝΑΙΣΘΗΜΑΤΙΚΗ ΑΝΑΛΥΣΗ

1.1 Εισαγωγή στην Ανάλυση Δεδομένων

Η Ανάλυση Δεδομένων (data analysis), αποτελεί τη διαδικασία συλλογής, επεξεργασίας και μοντελοποίησης δεδομένων, με στόχο την εύρεση και άντληση χρήσιμης πληροφορίας για την υποστήριξη λήψεων αποφάσεων. Η «εξόρυξη γνώσης» είναι μια τεχνική ανάλυσης που επικεντρώνεται στην εύρεση μοντέλων-προτύπων και την αναζήτηση γνώσης με σκοπό να προβλέψουμε συμπεριφορές. Η διαδικασία αυτή επικεντρώνεται κυρίως στην πρόβλεψη, παρά την περιγραφή των φαινομένων και συμπεριφορών. Η προγνωστική ανάλυση (predictive analytics), στοχεύει στην εφαρμογή στατιστικών μοντέλων για την πρόβλεψη ή κατηγοριοποίηση δεδομένων, καθώς επίσης και η ανάλυση κειμένου (text analytics) επιτυγχάνεται με την εφαρμογή στατιστικών εργαλείων σε συνδυασμό με γλωσσολογικές τεχνικές ώστε να εξαχθεί και να κατηγοριοποιηθεί πληροφορία από πηγές με δεδομένα χωρίς δομή (unstructured data). Η διαδικασία της ανάλυσης δεδομένων είναι η ακόλουθη:



Εικόνα 1: *Διαδικασία Ανάλυσης Δεδομένων*

1.2 Εισαγωγή στο Big Data

Η ανάλυση μεγάλου όγκου δεδομένων έχει εξελιχθεί ραγδαία τα τελευταία χρόνια και ενώ παλιότερα γινόταν αναφορά σε αυτή απλά με την ορολογία analytics ή business intelligence, έφτασε κάποια στιγμή που η αγορά ένιωσε την ανάγκη για την χρήση ενός πιο ελκυστικού όρου, γνωστό και ως big data.

Επειδή ο όγκος των δεδομένων προς ανάλυση είναι τεράστιος, κρίνεται σκόπιμο να αλλάξουμε δομικά τον τρόπο με τον οποίο γίνεται η ανάλυση δεδομένων. Για παράδειγμα, η δυνατότητα να εξαχθεί γνώση για το ποιόν μίας εταιρείας, από τις γνώμες που κυκλοφορούν στα social media, είναι τρομακτικά πολύτιμη: Η εταιρεία έχει τη δυνατότητα να καταλάβει τα χαρακτηριστικά του προϊόντος της που το κάνουν

περιζήτητο, καθώς και τα χαρακτηριστικά εκείνα για τα οποία ο κόσμος αποφεύγει την αγορά του προϊόντος αυτού.

Για να επιλυθεί η ανάγκη αυτή, λοιπόν, πρέπει να ξεπεραστεί η δυσκολία του τεράστιου όγκου δεδομένου, ο διαχωρισμός των πηγών, και κυρίως, στο να γίνει επιτυχής ανάλυση του συναισθήματος που εκφράζεται, ώστε να μοντελοποιηθεί κατάλληλα.

1.3 Συναισθηματική Ανάλυση

Η συναισθηματική ανάλυση είναι ένα πεδίο της Επεξεργασίας Φυσικής Γλώσσας (Natural Language Processing – NLP) που αφορά κυρίως την εξόρυξη γνώσης από κείμενα (Text Mining). Πρόκειται για την εύρεση και συλλογή γνωμών, συναισθημάτων και γενικότερα κάθε υποκειμενικότητας που μπορεί να φέρει ένα κείμενο. Αποτελείται από την συστηματική εύρεση, την αναγνώριση, ποσοτικοποίηση και την μελέτη των υποκειμενικών στοιχείων από το αρχικό υλικό. Ο ρόλος των συναισθημάτων στην αλληλεπίδραση ανθρώπου υπολογιστή, αρχικά ερευνήθηκε από την Picard η οποία πρότεινε το μοντέλο Συναισθηματικής Υπολογιστικής (Affective Computing) [5], αναδεικνύοντας την σημασία των συναισθημάτων στην αλληλεπίδραση ανθρώπου υπολογιστή, ανοίγοντας τον δρόμο για έρευνα από τομείς πληροφορικής ψυχολογίας και άλλων. Σκοπός την Συναισθηματικής Υπολογιστικής είναι να εφοδιαστούν οι υπολογιστές με την ικανότητα να αναγνωρίζουν και να εκφράζουν συναισθήματα, γεφυρώνοντας το κενό μεταξύ της συναισθηματικής φύσης του ανθρώπου και του υπολογιστή, αναπτύσσοντας υπολογιστικά συστήματα που αναγνωρίζουν και προσαρμόζονται στη συναισθηματική κατάσταση του χρήστη [6].

Σε αυτό βοήθησε πολύ το μοντέλο συναισθηματικών κατηγοριών του Ekman [7] το οποίο προσδιόριζε έξι βασικά ανθρώπινα συναισθήματα. Θυμός, αηδία, φόβος, χαρά, λύπη, έκπληξη. Έχει χρησιμοποιηθεί σε πολλές μελέτες και συστήματα που αναγνωρίζουν συναισθηματικά κείμενα και εκφράσεις προσώπου, ως προς την σχέση τους με την συναισθηματική κατάσταση. Αρκετά μοντέλα συναισθημάτων χρησιμοποιήθηκαν με άλλα συναισθήματα και άλλη δομή, όπως το μοντέλο OCC [8] που πρότεινε μια ιεραρχία ταξινόμησης 22 τύπων συναισθημάτων, σαν ένα δέντρο αποφάσεων.

Ο τεράστιος όγκος κειμενικής πληροφορίας που υπάρχει στο διαδίκτυο, δείχνει πως η αποτελεσματική Συναισθηματική Ανάλυση είναι απαραίτητη. Η συναισθηματική ανάλυση ήδη χρησιμοποιείται ευρέως σε προϊόντα και υπηρεσίες που εκφράζουν τη γνώμη τους οι καταναλωτές, όπως κριτικές και αποτελέσματα δημοσκοπήσεων, στα κοινωνικά δίκτυα, καθώς και σε φορείς υγείας, από μάρκετινγκ, εξυπηρέτηση πελατών, μέχρι και αυτοματοποιημένη υγειονομική διάγνωση. Τα δεδομένα από το Twitter , για παράδειγμα, χρησιμοποιούνται πολύ συχνά για προβλήματα Επεξεργασίας Φυσικής Γλώσσας και Συναισθηματικής Ανάλυσης.[9,10,11,12,13].

Είναι σημαντικό να αναφέρουμε πως έχει αναπτυχθεί μοντέλο συναισθηματικής ανάλυσης και για την Ελληνική Γλώσσα [14].

1.3.1 Συναισθηματική Ανάλυση σε κειμενικά δεδομένα

Χρησιμοποιώντας τεχνικές Επεξεργασίας Φυσικής Γλώσσας (Natural Language Processing – NLP) μπορούμε να κατατάξουμε κείμενα σε κάποια συναισθηματική

κατηγορία. Γενικά, η Συναισθηματική Ανάλυση προσπαθεί να αντλήσει τη θέση του συγγραφέα ως προς το θέμα, ή το γενικό συναίσθημα που κυριαρχεί στο κείμενο.

Με βάση το μέγεθος του κειμένου, μπορούμε να χωρίσουμε την συναισθηματική ανάλυση σε διαφορετικές κατηγορίες.

1.3.1.1 Κατηγορίες Συναισθηματικής Ανάλυσης

Μπορούμε να διακρίνουμε την συναισθηματική ανάλυση σε διαφορετικές κατηγορίες, ανάλογα με τον τύπο των συναισθημάτων που εξετάζουμε, ή τον όγκο του εκάστοτε κειμένου.

1.3.1.1.1 Συναισθηματική ανάλυση με βάση τον τύπο του συναισθήματος

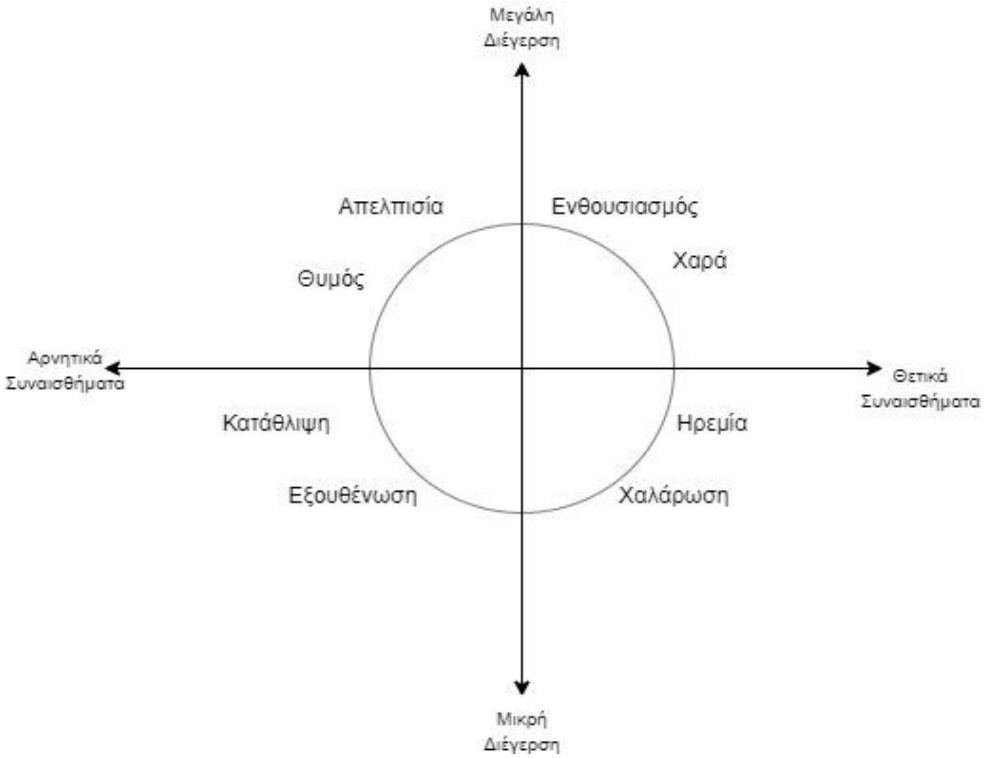
Απλή ανάλυση

Ο τύπος της ανάλυσης αυτής αναγνωρίζει αν έχουμε συναίσθημα θετικό ή αρνητικό. Σε κάποιες περιπτώσεις, οι κατηγορίες μπορεί να είναι 3 (θετικό, αρνητικό, ουδέτερο) ή 5 (πολύ θετικό, θετικό, ουδέτερο, αρνητικό, πολύ αρνητικό).

Ανάλυση συναισθηματικής κατάστασης

Στην ανάλυση αυτή, το σύστημα προσπαθεί να αναγνωρίσει την συναισθηματική κατάσταση των δεδομένων προέλευσης. Τέτοιες καταστάσεις μπορεί να είναι: χαρούμενη, στενάχωρη, θυμωμένη κτλ.

Η γραφική παράσταση Διάθεσης – Διέγερσης (Valence-Arousal) περιγράφει σε 2 διαστάσεις την κατηγοριοποίηση διαφορετικών συναισθημάτων. Στον άξονα X περιγράφεται αν και πόσο θετικό ή αρνητικό είναι ένα συναίσθημα, ενώ στον άξονα y η διέγερση περιγράφει την πόσο διεγερτικό είναι το συναίσθημα[15].



***Εικόνα 2**: Το διάγραμμα Διέγερσης - Συναισθήματος (Valance - Arousal)*

1.3.1.1.2 Συναισθηματική ανάλυση με βάση το μέγεθος του κειμένου

Σε επίπεδο κειμένου

Η συναισθηματική ανάλυση σε επίπεδο κειμένου κατηγοριοποιεί το κείμενο που δεχτήκαμε σαν είσοδο σαν θετικό η αρνητικό. Για παράδειγμα, όταν γίνεται κριτική για ένα προϊόν από χρήστες, το σύστημα αναγνωρίζει μία γενική εικόνα για το συναίσθημα του χρήστη και παρουσιάζει τα αποτελέσματα. Όλες οι γνώμες που εκφράζονται σε ένα κείμενο, αποτελούν μία οντότητα. Σε περίπτωση που έχουμε παραπάνω οντότητες, ή το αποτέλεσμα υπολογίζεται στο τέλος ή αν γίνονται συγκρίσεις μέσα στο κείμενο, τότε δεν μπορούμε να εφαρμόσουμε συναισθηματική ανάλυση σε επίπεδο κειμένου [16].

Σε επίπεδο προτάσεων

Όπως φαίνεται και από το όνομα, η συναισθηματική ανάλυση γίνεται σε επίπεδο προτάσεων και όχι πάνω σε ολόκληρο κείμενο. Στο επίπεδο αυτό, κάθε πρόταση χαρακτηρίζεται από θετικό, αρνητικό, ή ουδέτερο συναίσθημα. Στο επίπεδο αυτό εφαρμόζεται αναγνώριση υποκειμενικότητας και συναισθηματική ανάλυση [17].

Σε επίπεδο χαρακτηριστικών

Όταν το κείμενο που παρέχεται προς ανάλυση, αναλύει με διαφορετικό τρόπο τα συστατικά του χαρακτηριστικά. Η ανάλυση αυτή μπορεί να είναι χρήσιμη σε εταιρείες παροχής προϊόντων ή υπηρεσιών που αποτελούνται από διαφορετικά χαρακτηριστικά [18].

1.3.1.2 Τεχνικές υλοποίησης Συναισθηματικής Ανάλυσης σε κειμενικά δεδομένα

Υπάρχουν δύο κύριες τεχνικές συναισθηματικής ανάλυσης: Αυτές που βασίζονται σε μηχανική μάθηση και αυτές που βασίζονται σε λεξικό

Τεχνικές που βασίζονται σε Μηχανική Μάθηση

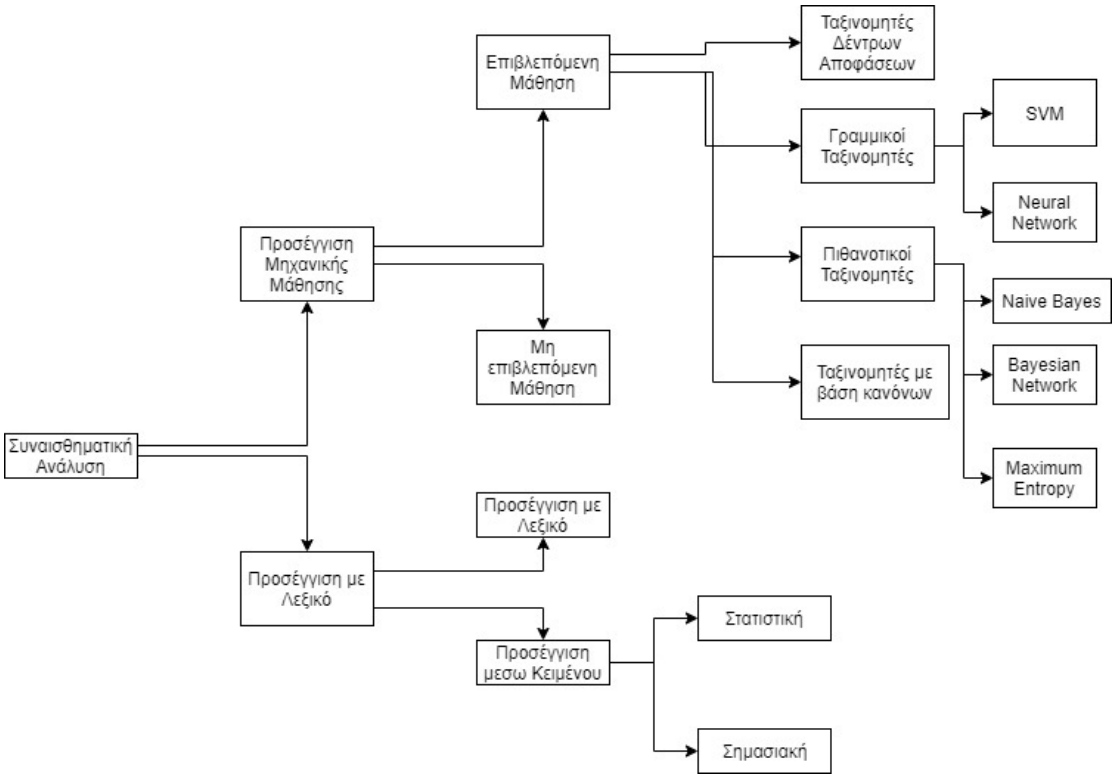
Η προσέγγιση με μηχανική μάθηση χρησιμοποιείται κυρίως σε συναισθηματική ανάλυση κατηγοριοποίησης. Στην τεχνική αυτή, χρησιμοποιούμε 2 σετ δεδομένων: Το σετ εκπαίδευσης (σετ εκπαίδευσης) και το τεστ ελέγχου (σετ ελέγχου). Το σετ εκπαίδευσης χρησιμοποιείται από έναν αυτόματο ταξινομητή ώστε να μάθει τα διαφορετικά χαρακτηριστικά των κειμένων μας, και το σετ ελέγχου για να ελέγχει την ακρίβεια του ταξινομητή αυτού. Η μηχανική μάθηση ξεκινά συλλέγοντας τα δεδομένα μας. Το επόμενο βήμα είναι η εκπαίδευση του ταξινομητή μας με το σετ δεδομένων εκπαίδευσης. Όταν έχουμε επιλέξει μια προσέγγιση επιβλεπόμενης μάθησης, ένα σημαντικό βήμα είναι να διαλέξουμε τα γλωσσολογικά στοιχεία που θα διαμορφώσουν το μοντέλο μας. Με τον τρόπο αυτό, θα αναπαρασταθούν τα κείμενα μας.

Καθώς κάθε ταξινομητής φέρει διαφορετική υλοποίηση, με διαφορετικά χαρακτηριστικά, μπορούμε να δημιουργήσουμε πολλά διαφορετικά μοντέλα, με

διαφορετικά χαρακτηριστικά, διαφορετικούς ταξινομητές ή και συνδυασμό τους, ώστε κάθε ένας να εφαρμόζεται πάνω σε δεδομένα τύπου που αξιοποιεί καλυτέρα [19]

Τεχνικές που βασίζονται σε λεξικά

Στην προσέγγιση αυτή χρησιμοποιούμε λεξικά που περιέχουν λίστες με τα διαφορετικά συναισθήματα και γνώμες που μπορεί να εκφράζουν διαφορετικές λέξεις. Οι αντιστοιχίες αυτές προϋπάρχουν. Για παράδειγμα, χρησιμοποιώντας λεξικά με θετικά και αρνητικά συναισθήματα, μπορούμε να ποσοτικοποιήσουμε το συναίσθημα ενός κειμένου, αθροίζοντας το σκορ κάθε λέξεις. Στο τέλος, αν το θετικό συναισθηματικό σκορ είναι μεγαλύτερο από το αρνητικό, τότε το κείμενο μας έχει θετική συναισθηματική χροιά.
Η τεχνικές που βασίζονται σε λεξικό αποτελούν μη επιβλεπόμενη μάθηση, καθώς δεν χρειαζόμαστε προηγούμενη εκπαίδευση του μοντέλου μας ώστε να ταξινομήσει τα δεδομένα. Μοντέλα που δημιουργήθηκαν για την αναγνώριση συναισθήματος σε προτάσεις Φυσικής Γλώσσας βασισμένες σε λεξικά, έχουν δείξει ότι μπορούν να έχουν ικανοποιητικά αποτελέσματα [20].



***Εικόνα 3**: Διαφορετικοί Τρόποι Συναισθηματικής Ανάλυσης*

1.3.2 Συναισθηματική Ανάλυση σε στίχους

Η διαδικασία συναισθηματικής ανάλυσης σε στίχους είναι η ίδια όπως με τα υπόλοιπα κειμενικά δεδομένα. Σε σχέση με την συναισθηματική ανάλυση σε κριτικές από χρήστες, η συναισθηματική ανάλυση σε στίχους είναι πιο δύσκολη. Οι κριτικές

περιλαμβάνουν συνήθως συναισθηματικά φορτισμένες λέξεις, με τις οποίες εύκολα μπορούμε να αντλήσουμε χρήσιμες πληροφορίες. Παρόλα αυτά, λόγω της διαφορετικής τους φύσης, στους στίχους, δεν είναι πάντα έτσι τα πράγματα. Συγκεκριμένα, υπάρχουν οι εξής δυσκολίες:

1. Οι στίχοι μπορεί να περιέχουν αρνητικά φορτισμένο περιεχόμενο, και να καταλήγουν σε ένα αντιθετικό, θετικό συναίσθημα, ή και το αντίστροφο. Το τελικό συναίσθημα που θα καταλάβαινε ένας άνθρωπος, θα ήταν το τελευταίο, αντίθετα με την συναισθηματική ανάλυση, που σε κανονικές καταστάσεις θα συνυπολόγιζε με το ίδιο βάρος όλα τα μέρη του τραγουδιού.

2. Οι στίχοι μπορεί να μην περιέχουν κάποια συναισθηματική πληροφορία σαν λέξη, αλλά να εκπέμπουν συγκεκριμένο συναίσθημα. Για παράδειγμα το ”It’s Still Rock And Roll To Me” από Billy Joel περιέχει το ακόλουθο: «What’s the matter with the clothes I’m wearing? Can’t you tell that your tie’s too wide? Maybe I should buy some old tab collars? Welcome back to the age of jive. » Αν αντιστοιχούσαμε τις λέξεις αυτές σε κάποιο λεξικό που περιλαμβάνουν θετικό περιεχόμενο, δεν θα βρίσκαμε κάτι χρήσιμο. Παρόλα αυτά, σαν σύνολο το τραγούδι εκπέμπει θετικό συναίσθημα.

3.Οι στίχοι μπορούν να εκπέμπουν θετικό συναίσθημα για κάποιο αρνητικό γεγονός, και το αντίστροφο. Αυτό έγκειται στην κοινωνική φύση της γλώσσας. Για παράδειγμα, Η ραπ μουσική περιέχει θετικά συναισθήματα για ληστείες και ναρκωτικά.

4.Η φύση των στίχων αποτελεί διαφορετική περίπτωση από τις άλλες μορφές κειμένου προς συναισθηματική ανάλυση. Η προσοχή που δίνεται στο ρεφραίν, τόσο λόγω της επαναληπτικότητας του αλλά και αρμονικά, σε αντίθεση με το κουπλέ, υποδεικνύει πως πρέπει να δομήσουμε το ερώτημα μας με τον καλύτερο τρόπο, για να πάρουμε την καλύτερη απάντηση. Θα χειριστούμε τα δεδομένα μας σαν στίχους, ή σαν κείμενο? Κάθε απάντηση κρύβει παγίδες και λάθη, ανάλογα με το σετ δεδομένων μας

1.3.3 Συναισθήματα και Διάθεση στη Μουσική

Είναι σημαντικό να αναφέρουμε τη διαφορά που υπάρχει στην ορολογία διαφορετικών πεδίων επιστήμης. Ερευνητές της μουσικής ψυχολογίας από τα αρχικά στάδια της και ερευνητών Εξόρυξης Μουσικής πληροφορίας, χρησιμοποιούν παραπλήσιους όρους, όμως διαφορετικούς για να απευθυνθούν στη συναισθηματική κατάσταση που δημιουργείται στον ακροατή. Ο πρώτος επίσημος διαχωρισμός έγινε από τον Meyer, ο οποίος χρησιμοποίησε τη λέξη «συναίσθημα» για να περιγράψει κάτι «προσωρινό και ασταθές» [21] και τη λέξη «διάθεση» για να περιγράψει κάτι «σχετικά διαρκές και σταθερό». 50 χρόνια μετά σε έρευνα των Sloboda και Juslin [22] επικυρώθηκε μόλις τελείωσαν σχετικές έρευνες. Στη μουσική ψυχολογία, και οι 2 λέξεις αναφέρονται στα αποτελέσματα της μουσικής, αν και ο όρος συναισθήματα φαίνεται να είναι πιο διαδεδομένος. Αντίθετα, στην Εξόρυξη Μουσικής Πληροφορίας – MIR όλοι οι ερευνητές ακολουθούν τη λέξη διάθεση.

Αν και οι ψυχολόγοι επικεντρώνονται στις ανθρώπινες αντιδράσεις σε διάφορα ερεθίσματα συναισθημάτων, οι ερευνητές επικεντρώνονται σε συναισθήματα που δημιουργεί η μουσική. Με άλλα λόγια, οι ψυχολόγοι στοχεύουν στις πολύ υποκειμενικές αντιδράσεις που μπορεί να προκαλέσει η προσωρινή, ακραία μεταβαλλόμενη οξεία μουσική πληροφορία, ενώ οι ερευνητές της εξόρυξης μουσικής πληροφορίας επικεντρώνονται σε κοινά μοτίβα μουσικής που αναγνωρίζονται από πολύ κόσμο και είναι περισσότερο σταθερά. Για το λόγο αυτό, βλέπουμε διαφορετική

ορολογία σε επιστημονικές δημοσιεύσεις ίδιου κλάδου, αυτού της συναισθηματικής ανάλυσης σε μουσική και στίχους.

1.3.4 Συναισθηματική Ανάλυση σε Μουσική

Η συναισθηματική ανάλυση σε μουσική, γίνεται αντλώντας τα μουσικά στοιχεία ενός τραγουδιού. Η διαδικασία αυτή γίνεται είτε σε κάθε ανάλυση, παίρνοντας ένα μικρό μέρος του τραγουδιού εξάγοντας μουσική πληροφορία, είτε παίρνοντας έτοιμες μουσικές πληροφορίες από τραγούδια. Αφού έχουμε μηχανική μάθηση με επίβλεψη, είναι σημαντικό να έχουμε και τις συναισθηματικές κλάσεις των τραγουδιών αυτών. Τα μουσικά χαρακτηριστικά αυτά μπορούν να αφορούν διαφορετικά ποιοτικά στοιχεία ενός τραγουδιού, όπως ο ρυθμός, η τονικότητα, η κλίμακα στην οποία είναι γραμμένη το τραγούδι, το τέμπο κτλ. Τα δεδομένα αυτά πιθανόν να χρειάζονται κανονικοποίηση, καθώς κάθε ένα από αυτά έχει διαφορετική μέτρηση. (η κλίμακα είναι ποιοτικό χαρακτηριστικό, το τέμπο αποτελεί μία συνεχή τιμή συνήθως έως 220 χτύπους/δευτερόλεπτο, η τονικότητα μπορεί να εκφραστεί σε Hz). Στην συνέχεια, μπορούμε να υλοποιήσουμε μηχανική μάθηση χρησιμοποιώντας διαφορετικούς ταξινομητές, ώστε να εξάγουμε πληροφορία.

.

1.4 Προηγούμενη Δουλειά

1.4.1 Αναγνώριση συναισθήματος σε Κείμενο

Η αναγνώριση συναισθήματος σε κείμενο είναι ένα διαχρονικό πεδίο μελέτης. Ερευνητές προσπαθούν να επιλύσουν ικανοποιητικά το πρόβλημα ακολουθώντας διαφορετικές προσεγγίσεις, πάνω σε διαφορετικού είδους δεδομένα. Η συναισθηματική ανάλυση πάνω σε κριτικές προϊόντων / υπηρεσιών , αξιοποιούνται χρόνια από την αγορά, όπως και η συναισθηματική ανάλυση πάνω σε δεδομένα κοινωνικών δικτύων.

Άξια μνείας είναι η υλοποίηση συναισθηματικής ανάλυσης σε κριτικές ξενοδοχείων στο «A System for Aspect-based Opinion Mining of Hotel Reviews». Στην υλοποίηση αυτή χρησιμοποιείται LDA ώστε να μοντελοποιηθούν γνώμες πάνω στα ξεχωριστά θέματα και Επεξεργασία Φυσικής Γλώσσας σε επίπεδο προτάσεων ώστε να βρεθούν οι εξαρτήσεις μεταξύ των λέξεων. Με τον τρόπο αυτό μπορεί να αναγνωριστεί το συναίσθημα της πρότασης. Τέλος, εκπαιδεύεται ένας ταξινομητής Naïve Bayes με τα δεδομένα αυτά ώστε να ταξινομήσει τις προτάσεις στις αντίστοιχες κατηγορίες. Το μοντέλο πέτυχε 72% επιτυχία σε όλες τις περιπτώσεις [23].

Όσον αφορά την συναισθηματική ανάλυση από δεδομένα κοινωνικών δικτύων αναφέρουμε το «A Classifier Ensemble Approach to Detect Emotions Polarity in Social Media . Παρουσιάζεται ένα σύνολο λειτουργιών που δημιουργούν μία προσέγγιση συναισθηματικής ανάλυσης στο περιεχόμενο κοινωνικών δικτύων., του οποίου την απόδοση εξετάζουμε κάτω από διαφορετικές μεθόδους. Συγκεκριμένα, χρησιμοποιώντας bagging, μία τεχνική εκπαίδευσης διαφορετικών ταξινομητών σε διαφορετικά υποσέτ δεδομένων και τεχνικών εξόρυξης συναισθήματος σε επίπεδο προτάσεων με λεξικά, καταλήγει να πετυχαινει σκορ 82% με Naïve Bayes εκπαιδευμένα πάνω στο ISEAR Dataset [24]

Μία διαφορετική προσέγγιση για συναισθηματική ανάλυση σε δεδομένα κοινωνικών δικτύων παρουσιάζεται στο « Aspect based sentiment analysis in social media with classifier ensembles». Εδώ αξιοποιείται το LDA ώστε να προσδιοριστούν τα θέματα

που θίγει ο χρήστης. Κάθε σχόλιο αναλύεται αντλώντας τις γραμματικές εξαρτήσεις του. Ένα σύνολο ταξινομητών από Naïve Bayes, Maximum Entropy και Support Vector Machines αξιοποιείται για να αναγνωριστεί η πολικότητα του συναισθήματος του χρήστη για κάθε θέμα. Τα αποτελέσματα δείξαν σημαντικές βελτιώσεις σε σχέση με αυτοτελείς ταξινομητές και υποδεικνύουν πως το σύστημα μας είναι επεκτάσιμο και ακριβές [25].

1.4.2 Αναγνώριση συναισθήματος σε Στίχους

Οι Ashley M. Oudenne και Sarah E. Chasins χρησιμοποίησαν επεξεργασία φυσικής γλώσσας για να κατηγοριοποιήσουν ένα τραγούδι σαν χαρούμενο ή στενάχωρο, με βάση κάποια σετ συχνοτήτων λέξεων και αλγορίθμων μηχανικής μάθησης, πάνω σε 420 τραγούδια.

Συγκεκριμένα, χρησιμοποιώντας διαφορετικούς αλγορίθμους, προσπάθησαν να επιτύχουν την συναισθηματική κατηγοριοποίηση. Αρχικά με λίστες λέξεων, οι οποίες περιέχουν λίστες με τις πιο συχνές λέξεις που υπάρχουν *αποκλειστικά* σε χαρούμενα τραγούδια και τις πιο συχνές λέξεις που υπάρχουν *αποκλειστικά* σε στενάχωρα τραγούδια. Στην συνέχεια, για κάθε τραγούδι που έχουμε προς κατηγοριοποίηση, ελέγχουμε ποιες λέξεις των στίχων του υπάρχουν σε ποιες λίστες. Όποια λίστα έχει περισσότερες αναφορές, κατοχυρώνει το συναίσθημα της στο τραγούδι. Στην συνέχεια, δοκίμασαν με λεξικό, αξιοποιώντας τον αλγόριθμο Present in One. Δημιούργησαν ένα λεξικό που περιλαμβάνουν τις λέξεις των τραγουδιών με θετικό συναίσθημα, και ένα λεξικό που περιλαμβάνει τις λέξεις των τραγουδιών με αρνητικό συναίσθημα. Για κάθε λέξη που υπάρχει μόνο στο ένα λεξικό, το συναισθηματικό σκορ αυτό έπαιρνε ένα πόντο. Τέλος αξιοποίησαν και τον ταξινομητή Naïve-Bayes. Το καλύτερο σκορ που πέτυχαν ήταν 68.6% [26].

Οι Dang Trung Thanh, Kiyoaki Shirai παρουσιάζουν ένα τρόπο συναισθηματικής ανάλυσης σε στίχος, χρησιμοποιώντας ταξινομητές SVM, Naïve Bayes και με γραφικές μεθόδους. Δημιουργώντας 5 συστάδες διάθεσης και αντλώντας 6000 τραγούδια, καταφέρνουν μια επιτυχία γύρω στο 55%, καταλήγοντας στο ότι οι απλές μέθοδοι ταξινόμησης δεν αρκούν για ένα επαρκές αποτέλεσμα. Αυτό οφείλεται στο ότι η μουσική περιέχει υποκειμενικά στοιχεία και στο ότι οι στίχοι περιέχουν μέρη του λόγου όπως μεταφορές, τις οποίες μόνο άνθρωπος μπορεί να καταλάβει. Παρόλα αυτά, θεωρούν ότι αν δοθεί παραπάνω ενδιαφέρον στη φύση των στίχων, δηλαδή στην επαναληπτικότητα του ρεφρέν και στο στίχο που περιλαμβάνει τον τίτλο του τραγουδιού, μπορούν να γίνουν βελτιώσεις [27].

Στο “MusicMood: Predicting the mood of music from song lyrics using machine learning” χτίζεται ένα σύστημα πρότασης μουσικής βασιζόμενο σε ένα ταξινομητή Naïve Bayes, ο οποίος εκπαιδεύτηκε ώστε να προβλέπει τα συναισθήματα τραγουδιών με βάση μόνο τους στίχους. Τα πειραματικά αποτελέσματα δείξαν πως η μουσική που αντιστοιχεί σε θετικά συναισθήματα, μπορεί να αναγνωριστεί με μεγάλη ακρίβεια με βάση γλωσσολογικά στοιχεία τα οποία πάρθηκαν μέσα από στίχους. Από το σετ δεδομένων Million Song Dataset επιλέχτηκε ένα τυχαίο δείγμα 10.000 τραγουδιών. Οι στίχοι αυτοί περιείχαν θόρυβο, και επιλέχθηκαν συνειδητά έναντι άλλων επιλογών, ώστε να συγκριθούν διαφορετικές τεχνικές εξαγωγής χαρακτηριστικών και καθαρισμού. Καθώς οι συναισθηματικές κλάσεις αποκτήθηκαν από περιεχόμενο

χρηστών ήταν λίγες, ήταν επόμενο πως για να καλυφθεί η ανάγκη αυτή, έπρεπε να ανατεθούν στο «χέρι». Χρησιμοποιώντας διανύσματα με διαφορετικές παραμέτρους, όπως διαφορετικά n-grams , διαγραφή συχνών λέξεων, αλλά και διαφορετικών αναπαραστάσεων των διανυσμάτων δημιουργήθηκε ένα μοντέλο το οποίο πέτυχε σκορ 88.89 σε ένα σετ δεδομένων 200 τραγουδιών [28].

Οι Çano, Erion. στο «MoodyLyrics: A Sentiment Annotated Lyrics Dataset» ασχολήθηκαν με την αναγνώριση συναισθήματος και προτάσεων ως προς τον τρόπο με τον οποίο ο κόσμος βρίσκει και ακούει τη μουσική που θέλει. Χρησιμοποιώντας μόνο λεξικά και τη σχέση τους με το valance - arousal και τις μουσικές κλάσεις από το μοντέλο του Russel, πέτυχαν επιτυχία 74.25%. Κατέληξαν επίσης πως, σαν παράγοντας αποκλειστικότητας, το valance είναι καλύτερο διαχωριστικό στοιχείο για την ταξινόμηση από ότι η διέγερση, αμφισβητώντας πως η ακριβής συναισθηματική αναγνώριση μουσικής απαιτεί πάντα ανθρώπινο παράγοντα [29].

Αξίζει να αναφερθούμε και στο «When lyrics outperform Audio for Music Mood Classification: A Feature Analysis» των Xiao Hu J. Stephen Downie. Στη δημοσίευση αυτή, αντλήθηκαν 5,296 τραγούδια, με το στιχουργικό και ακουστικό τους περιεχόμενο, μαζί με τις συναισθηματικές ταμπέλες από το last.fm. Με συλλογή στιχουργικών χαρακτηριστικών με το Bag-of-Words και συνδυασμό n-grams και SVM ταξινομητές έδειξαν ότι από τις 18 διαφορετικές κατηγορίες συναισθημάτων, οι στίχοι είχαν μεγαλύτερο ποσοστό επιτυχίας, και μόνο σε μία κατηγορία η μουσική είχε μεγαλύτερο ποσοστό επιτυχίας [30].

1.4.2 Αναγνώριση διάθεσης σε Μουσική

Μία από τις πρώτες δημοσιεύσεις για το ζήτημα αποκλειστικά της μουσικής, από τον Li και Ogihara στο «Detecting Emotion in Music» χρησιμοποίησε ακουστικά χαρακτηριστικά με βάση τις αρμονικές σειρές, το ρυθμό και τις συχνότητες για να εκπαιδεύσει support vector machines (SVMs) για να ταξινομήσει μουσική σε 13 διαφορετικές διαθέσεις. Χρησιμοποιώντας μια βιβλιοθήκη από 499 αποσπάσματα τραγουδιών των 30 δευτερολέπτων, που αξιολογήθηκαν από άνθρωπο, μεταξύ διαφορετικών μουσικών ειδών, από jazz μέχρι ambient, πέτυχαν μία επιτυχία 45% [31].

Στο «Automatic mood detection and tracking of music audio signals» οι L. Lu, D. Liu, and H. J. Zhang, χρησιμοποιώντας αντίστοιχα ακουστικά σήματα, να αξιολογήσουν την διάθεση. Χρησιμοποίησαν τον ταξινομητή Gaussian Mixture Models (GMMs) για τις τέσσερις κυρίαρχες κλάσεις στην V-A αναπαράσταση που είδαμε προηγουμένως. Το σύστημα εκπαιδεύτηκε χρησιμοποιώντας 800 αποσπάσματα μουσικής από 250 τραγούδια, διάρκειας 20 δευτερολέπτων και πέτυχαν απόδοση 85% [32].

Οι J. Skowronek, M. McKinney, and S. van de Par στο “A demonstrator for automatic music mood estimation” ανέπτυξαν δυαδικούς ταξινομητές για 12 μουσικές κατηγορίες, μη αμοιβαία αποκλειόμενες χρησιμοποιώντας 1059 αποσπάσματα τραγουδιών και αντλώντας στοιχεία όπως tempo, ρυθμός, κλειδί και περιπτώσεις κρουστών ήχων. Δημιουργώντας διαφορετικές συναρτήσεις για κάθε διάθεση με την βιβλιοθήκη MIRtoolbox της Matlab: πέτυχαν από 77% για την διάθεση «carefree-playful», έως 91% για την calming-soothing [33].

1.4.3 Υβριδικά μοντέλα αναγνώρισης συναισθήματος

Ένα από τα ζητήματα που απασχόλησε την ερευνητική κοινότητα, ήταν το ότι οι κοινωνικοί παράγοντες επηρεάζουν τα μουσικά δεδομένα. Για παράδειγμα, τα Χριστουγεννιάτικα τραγούδια έχουν σαφή χαρούμενη διάσταση, ακόμα και αν αυτό δεν είναι ξεκάθαρο από τα μουσικά τους δεδομένα, αλλά από τα στιχουργικά. Συνεπώς, η εξαγωγή αποκλειστικά μουσικής πληροφορίας, η ανάκτηση μουσικής πληροφορίας (Music-Information Retrieval) έχει φραγμένη απόδοση ως προς την αξιολόγηση συναισθηματικής κατάστασης. Για το λόγο αυτό, στράφηκε το ενδιαφέρον στον συνδυασμό στιχουργικών και μουσικών δεδομένων.

Υπάρχουν δύο προσεγγίσεις για τη δημιουργία υβριδικών μοντέλων, που συνδυάζουν γλωσσολογικά και ακουστικά στοιχεία. Η πιο εύληπτη είναι αυτή όπου τα διαφορετικού τύπου δεδομένα ενώνονται, αποθηκεύονται σε ένα κοινό διάνυσμα και ο αλγόριθμος ταξινόμησης τρέχει πάνω στο διάνυσμα αυτό [34]. Η άλλη μέθοδος είναι αυτή που ονομάζεται «αργοπορημένη ένωση» και συνδυάζει τα αποτελέσματα των ξεχωριστών ταξινομητών για διαφορετικά δεδομένα , είτε χρησιμοποιώντας μέσο όρο με βάρη [35] είτε με πολλαπλασιασμό [36].

Το πρώτο σύστημα που συνπεριείχε ανάλυση μουσικής και στίχων για την κατηγοριοποίηση μουσικής, ήταν το «Disambiguating Music Emotion Using Software Agents» από Dan Yang WonSook Lee. Χρησιμοποιώντας στίχους και πολλά ακουστικά χαρακτηριστικά, όπως tempo, και 12-χαμηλού επιπέδου χαρακτηριστικά ήχου MPEG-7 σε σετ δεδομένων 145 τραγουδιών (των οποίων πάρθηκε απόσπασμα 30 δευτερολέπτων) αξιολόγησαν άνθρωποι την συναισθηματική κλάση με βάση 11 συναισθηματικές κατηγορίες και κατέληξαν σε ένα μοντέλο με επιτυχία 82.8% [37].

Στο «Lyric text mining in music mood classification» ο Hu συνδύασε ήχο και στίχους για μία μεγάλη βάση 3000 τραγουδιών. Για τους στίχους, χρησιμοποίησε προσέγγιση Bag-of-Words με TF-IDF βάρη, stemming. Το διάνυσμα που δημιουργήθηκε, ενώθηκε με τα ακουστικά στοιχεία και εκπαιδεύτηκε ένας SVM ταξινομητής. Είναι ενδιαφέρον να σημειωθεί, ότι χρησιμοποιώντας τους στίχους μόνο, το αποτέλεσμα είναι καλύτερο από το να χρησιμοποιούμε τα ηχητικά χαρακτηριστικά για 12 από τις 18 κατηγορίες. Τα ηχητικά μαζί με τα κειμενικά δεδομένα έχουν καλύτερα αποτελέσματα για 13 από τις 18 κατηγορίες. Η ηχητική προσέγγιση έχει τα καλύτερα αποτελέσματα για κάποιες από τις «χαρούμενες κλάσεις» (happy,upbeat,desire), ενώ μόνο ο στίχος έχει καλύτερα αποτελέσματα για τις κλάσεις «grief,exciting» [38].

Μία σημαντική έρευνα ήταν αυτή για την αναγνώριση του είδους της μουσικής, στο “Exploring Customer Reviews for Music Genre Classification and Evolutionary Studies”. Στη δημοσίευση αυτή, χρησιμοποιήθηκε ένα σετ δεδομένων από 65.000 άλμπουμ, κατασκευασμένο από κριτικές στο amazon, μεταδεδομένα από το MusicBrain και ακουστικά χαρακτηριστικά από το AcousticBrains. Με τον τρόπο αυτό, δημιουργούμε ένα τεράστιο σετ δεδομένων στο οποίο μπορούμε να κάνουμε πειράματα στη κατηγοριοποίηση είδους μουσικής, χρησιμοποιώντας διαφορετικά χαρακτηριστικά, όπως γλωσσολογικά, ακουστικά και συναισθηματικά .Τα πειράματα αυτά δείξαν ότι η μοντελοποίηση της σημασιολογικής πληροφορίας συνεισφέρει καθοριστικά στο να πετύχουμε επιτυχή ταξινόμηση. Παράλληλα, παρέχει μια διαχρονική έρευνα για την κριτική των μουσικών ειδών μέσα από ποσοτική ανάλυση της πολικότητας που αποδίδεται στα μουσικά χαρακτηριστικά. Η ανάλυση αυτή

υποδεικνύει μία πιθανή συσχέτιση μεταξύ σημαντικών γεωπολιτικών γεγονότων και κουλτούρας με τη γλώσσα και τα συναισθήματα που βρίσκονται σε κριτικές μουσικής.

Στη μελέτη αυτή χρησιμοποιήθηκε Επεξεργασία Φυσικής Γλώσσας, Opinion Mining και συναισθηματική ανάλυση για να εξάγουμε την επιθυμητή πληροφορία. Για κάθε άλμπουμ, χρησιμοποιήθηκαν συγκεκριμένου τύπου bi-grams και unigrams. Συγκεκριμένα, αναζητήθηκαν bigrams όπου περιλαμβάνουν ουσιαστικό και ουσιαστικό (πχ chorus arrangement) ή επίθετο που ακολουθείται από ουσιαστικό (πχ original sound), εξαιρώντας τα bigrams στα οποία το επίθετο φέρει συναισθηματική πληροφορία (πχ excellent,terrible).Για το μουσικό μέρος , ο στόχος ήταν να δημιουργηθεί ένα υποσέτ κατάλληλο για κατηγοριοποίηση είδους μουσικής, που θα περιλάμβανε 100 άλμπουμ ανά είδος. Αναζητήθηκαν άλμπουμ από διαφορετικούς καλλιτέχνες, των οποίων κείμενα κριτικών και τα ακουστικά χαρακτηριστικά μας ήταν διαθέσιμα. Οι ταξινομητές που επιλέχθηκαν ήταν ο LinearSVM,Ridge Classifier,RandomForest και ο Naïve Bayes και για τα κειμενικά δεδομένα επιλέχθηκε η αναπαράσταση TF-IDF. Δημιουργώντας διαφορετικούς συνδυασμούς για τα διαφορετικού τύπου δεδομένα που έχουμε, πέτυχε καλύτερο σκορ με 69% επιτυχία χρησιμοποιώντας στίχους και μεταδεδομένα [39] .

1.5 Στόχος Διπλωματικής Εργασίας

Αν και το πρόβλημα της συναισθηματικής ανάλυσης σε κριτικές ήδη χρησιμοποιείται ευρέως από την αγορά, δεν έχει γίνει η ανάλυση αυτή σε αντίστοιχο εύρος για μουσικά τραγούδια. Περιγράφουμε θεωρητικά τις τεχνικές Επεξεργασίας Φυσικής Γλώσσας και Μηχανικής Μάθησης που θα χρησιμοποιήσουμε, και μετέπειτα αναλύουμε την υλοποίηση μας.

Στόχος της εργασίας αυτής είναι να μελετήσουμε με διαφορετικούς τρόπους την εξεύρεση συναισθηματικής ποιότητας των στίχων και της μουσικής, χρησιμοποιώντας Μηχανική Μάθηση. Μελετώντας διαφορετικά σετ δεδομένων, στίχων, κριτικών και μουσικών θα δημιουργήσουμε διαφορετικά μοντέλα, τα οποία στο τέλος θα συγκρίνουμε πάνω στα δεδομένα μας, ώστε να βρούμε το καλύτερο.

Κίνητρο για την εκπόνηση της διπλωματικής εργασίας είναι να θέσουμε την βάση συναισθηματικής ανάλυσης σε μουσικά τραγούδια, έτσι ώστε στο μέλλον να βρούμε τον κατάλληλο συνδυασμό γλωσσολογικών και μουσικών στοιχείων που θα αρκούν για να κατηγοριοποιηθεί ένα τραγούδι σε συγκεκριμένες μουσικές κλάσεις.

ΚΕΦΑΛΑΙΟ 2. ΕΠΕΞΕΡΓΑΣΙΑ ΦΥΣΙΚΗΣ ΓΛΩΣΣΑΣ

Η Επεξεργασία Φυσικής Γλώσσας, (Natural Language Processing, NLP) είναι πεδίο της Τεχνητής Νοημοσύνης, το οποίο αφορά το πως πολλά στοιχεία φυσικής γλώσσας επεξεργάζονται από τον υπολογιστή. Συγκεκριμένα, γίνεται σχεδιασμός και υλοποίηση μοντέλων της φυσικής γλώσσας, ώστε να αναγνωρίζεται ή να κατανοείται η φυσική γλώσσα, να παράγεται φυσική γλώσσα ή συνδυασμός τους. Όντας ευρύ πεδίο έχει πολλές εφαρμογές, όπως στην μετάφραση από γλώσσα σε γλώσσα, συναισθηματική ανάλυση, πρόβλεψη εισαγωγής κειμένου [40] [41]. Εδώ και πολλά χρόνια τεράστιος ερευνητικός χρόνος έχει αφιερωθεί και αφιερώνεται ώστε να επιτευχθεί η καλύτερη δυνατή αποτύπωση της Φυσικής Γλώσσας σε Υπολογιστή [42].

Για τη διπλωματική εργασία αυτή χρησιμοποιήσαμε τις βιβλιοθήκες NLTK, Sci-kit Learn για να επεξεργαστούμε τη φυσική γλώσσα του σετ δεδομένων μας.

2.1 Εισαγωγή

Η Επεξεργασία Φυσικής Γλώσσας (ΕΦΓ) ασχολείται με την «χρήση ανθρώπινης γλώσσας από Υπολογιστή» [43]. Συνεπώς το πεδίο της ΕΦΓ ενσωματώνει όλες τις αλληλεπιδράσεις μεταξύ ανθρώπου και υπολογιστή, με χρήση γραπτής ή ομιλούμενης γλώσσας. Όλα τα χαρακτηριστικά της γλώσσας λοιπών, τα φωνήματα, η φωνολογία, η μορφολογία, η γραμματική και το συντακτικό πρέπει να λαμβάνονται υπόψιν ώστε να υπάρξει πλήρης κατανόηση του μηνύματος. Διαφορετικά πεδία της ΕΦΓ χρησιμοποιούνται ανάλογα με τη φύση του προβλήματος . Για παράδειγμα, στην αναγνώριση φωνής, χρησιμοποιούνται τα φωνήματα, τις ακουστικές λεπτομέρειες του ήχου δηλαδή που παράγεται όταν μιλά ένας άνθρωπος. Η μορφολογία αφορά το νόημα και την αρχιτεκτονική των λέξεων. Η λημματοποίηση που αναλύουμε παρακάτω βασίζονται σε αυτό.

2.2 Τεχνικές

Καθώς σπάνια τα κείμενα που αναλύουμε έχουν συγκεκριμένη και σαφή δομή, είναι πολύ σημαντικό να καταφέρουμε να αντλήσουμε το περιεχόμενο που θέλουμε στον υπολογιστή, που ζητά δομή και αριθμητικά δεδομένα. Η ενότητα αυτή περιλαμβάνει τεχνικές με τις οποίες μπορούμε να επεξεργαστούμε την φυσική γλώσσα αποφεύγοντας τον θόρυβο στα δεδομένα μας.

2.2.1 Διαμερισμός

Ο στόχος του σταδίου αυτού είναι να διαιρέσουμε το κείμενο σε μικρά σχετικά στοιχεία, συνήθως σαν ξεχωριστές λέξεις (**tokens)** ή σαν σειρά προκαθορισμένου αριθμού λέξεων (**n-grams).** Αποτελεί πρωταρχικό στάδιο της φάσης Επεξεργασίας Φυσικής Γλώσσας και χρησιμεύει στον πλήρη έλεγχο των λέξεων, για να επιλέγουμε ποια κειμενική πληροφορία θεωρούμε χρήσιμη, και να την αποθηκεύουμε, να την διώχνουμε, ή να την επεξεργαζόμαστε ανάλογα με τις ανάγκες του προβλήματος μας [44].

Η μέθοδος **word_tokenizer()** της NLTK βιβλιοθήκης χρησιμοποιείται για το tokenization των documents μας.

2.2.2 Λημματοποίηση

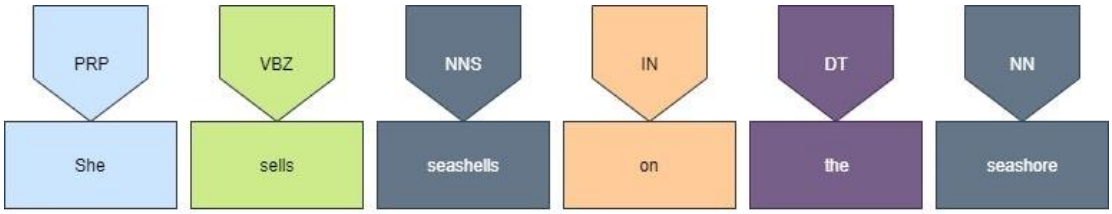
Η λημματοποίηση χρησιμοποιείται για την μείωση παραγόμενων λέξεων στο λήμμα τους. (Πχ “reports” → “report ή “τα good, better, best” λημματοποιούνται σε “good, good, good”). Η λημματοποίηση αναφέρεται σε ορθή ανάλυση της λέξης, λεξιλογικά

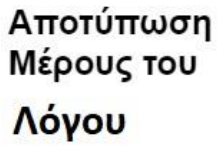
και μορφολογικά. Καθώς δεν ενδιαφερόμαστε τόσο για το περιεχόμενο του κειμένου, όσο για την αναγνώριση του συναισθήματος που εξάγει το περιεχόμενο, η λημματοποίηση βοηθά για να αντλήσουμε κοινό συναισθηματικό περιεχόμενο από παράγωγες λέξεις. Η λημματοποίηση αποτελεί βασική φάση της Επεξεργσίας Φυσικής Γλώσσας. Ένα σημαντικό πλεονέκτημα που προσφέρει, είναι η ομαδοποίηση των παραγόμεων λέξεων από την ίδια ρίζα. Με τον τρόπο αυτό, μπορούμε να διαχειριστούμε τα δεδομένα μας με μεγαλύτερη ευελιξια ως προς το περιεχόμενο μας [45]. Η μέθοδος WordNetLemmatizer() χρησιμοποιείται για να αντλήσουμε τα λήμματα από τις λέξεις.

2.2.3 Αποτύπωση Μέρος του Λόγου

Στο συντακτικό της αγγλικής γλώσσας, διδάσκονται 9 διαφορετικά μέρη λόγου: ουσιαστικά, ρήματα, άρθρα, επίθετα, προθέσεις, επιρρήματα, αντωνυμίες και συζεύξεις. Προφανώς υπάρχουν περισσότερες κατηγορίες και υπο-κατηγορίες.

Στη γλωσσολογική ανάλυση το Part-of-Speech tagging είναι η διαδικασία σημείωσης πάνω σε μία λέξη ενός κειμένου του μέρους του λόγου της, πάνω τόσο στον ορισμό της όσο και στο περιεχόμενο της – για παράδειγμα της σχέσης της με συναφείς και γειτονικές λέξεις της σε μία φράση, πρόταση ή παράγραφο.

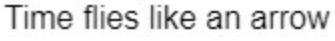




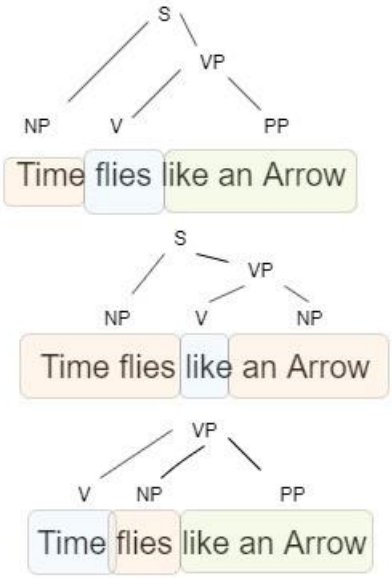
***Εικόνα 4**: Αποτύπωση του Μέρους του Λόγου*

Η αναγνώριση των μερών του λόγου σε ένα κείμενο είναι μια πολύπλοκη διαδικασία, που δεν εξαντλείται απλά στην αντιστοίχιση λέξεων στο αντίστοιχο μέρος του λόγου τους. Ίδιες λέξεις χρησιμοποιούνται διαφορετικά σε διαφορετικές προτάσεις, τόσο σαν μέρος του λόγου όσο και στο περιεχόμενο. Συνεπώς είναι αδύνατο να έχουμε έναν τεράστιο χάρτη για κάθε διαφορετικό ενδεχόμενο μέρους του λόγου.

Για παράδειγμα , η φράση *"Time flies like an arrow"* έχει διαφορετικές ερμηνείες. Το νόημα και κατ' επέκταση το POS Tagging θα διαφέρει για κάθε σημασία.



***Εικόνα 5**: Φράση προς POS-Tagging*



***Εικόνα 6**: Διαφορετικές Περιπτώσης αποτύπωσης μέρους του λόγου λέξεων*

Όπως βλέπουμε δεν μπορούμε να βρίσκουμε σαν άνθρωποι το μέρος του λόγου κάθε πρότασης σε ένα κείμενο. Καθώς η γλώσσα είναι δυναμική και όχι στατική, θα δημιουργούνται νέες λέξεις διαρκώς, άλλες λέξεις θα αλλάζουν περιεχόμενο με τα χρόνια, ακόμα και ίδιες λέξεις θα έχουν άλλη σημασία ανα διαφορετική διάλεκτο. Αυτό οποίο σημαίνει πως το manual POSTagging δεν είναι επεκτάσιμο.

Η αποτύπωση μέρος του λόγου αποτελεί σημαντική φάση της Επεξεργασίας Φυσικής Γλώσσας. Με αυτήν έχουμε μεγαλύτερο έλεγχο πάνω στο περιεχόμενο της Φυσικής Γλώσσας και μπορούμε να χρησιμοποιήσουμε ουσιαστικές γλωσσολογικές αναλύσεις στην Επεξεργασία Φυσικής Γλώσσας [46].

2.2.4 Εξαγωγή Χαρακτηριστικών από το κείμενο

Καθώς η αναπαράσταση λέξεων δεν εξυπηρετεί τις τεχνολογίες, για την δημιουργία μοντέλων Μηχανικής Μάθησης, ακολουθήθηκαν διαφορετικοί τρόποι.

Διανυσματοποίηση (Vectorization) ονομάζουμε την διαδικασία όπου η φυσική γλώσσα αναπαρίσταται στον υπολογιστή μέσω αριθμητικών τιμών. Κάθε αριθμητική τιμή υποδηλώνει την συχνότητα χρήσης της λέξης σε κάθε κείμενο. Η βιβλιοθήκη Sci-kit Learn υλοποιεί το vectorization μέσω 3 βημάτων:

- **Διαμερισμός (tokenizing)**: διαχωρισμός strings, όπου τους ανατίθεται ένας ακέραιος αριθμός για κάθε διαφορετικό token, για παράδειγμα χρησιμοποιώντας σαν delimeter το κενό ή το ‘,’.
- **Μέτρηση (Term Frequency)** των εμφανίσεων του κάθε token.
- **Κανονικοποίηση** χρησιμοποιώντας βάρος για να βρεθούν τα σημαντικότερα tokens και να αφαιρεθούν τα λιγότερο σημαντικά.

Αξίζει να σημειωθεί ότι αν και οι Vectorizers χρησιμοποιούν δικό τους tokenizer, ο οποίος κάνει προεπεξεργασία (μετατροπή κεφαλαίων σε μικρών, διαγραφή διπλών στοιχείων κτλ) εμείς θα αξιοποιήσουμε αυτόν που φτιάξαμε, που ταιριάζει στο σετ δεδομένων που έχουμε.

Οι διαφορετικές μέθοδοι με τις οποίες γίνεται η εύρεση των στοιχείων που θα αξιοποιήσουμε, δηλαδή μαθαίνουμε το **vocabulary** των στίχων, το λεγόμενο **feature extraction** μέσα από την καταμέτρηση-κανονικοποιήση που χρησιμοποιήσαμε είναι:

Για κάθε σετ δεδομένων, θα εφαρμόσουμε τα 2 μοντέλα που αντλούν το vocabulary και τα βάρη τους και θα υλοποιήσουμε ένα απλό μοντέλο μηχανικής μάθησης, ώστε να δούμε ποιο αποκρίνεται καλύτερα. Στην συνέχεια, θα παραμετροποιήσουμε κατάλληλα τον vectorizer αυτό ώστε να έχουμε καλύτερα αποτελέσματα.

2.2.4.1 Μοντέλο Bag-of-Words

Το μοντέλο bag-of-words είναι ένα μοντέλο απλοϊκής αναπαράστασης στο οποίο το κείμενο αναπαρίσταται σαν ένα σύνολο λέξεων, στο οποίο αγνοείται η γραμματική και η σειρά των λέξεων, αλλά διατηρείται η πολλαπλότητα τους .Επειδή στους αλγορίθμους μηχανικής μάθησης δεν μπορούμε να αξιοποιήσουμε φυσική γλώσσα, μετατρέπουμε το κείμενο στην ποσότητα συχνότητας εμφάνισης της εκάστοτε λέξης [47] και βασίζεται πάνω στην σκέψη ότι «η βαρύτητα κάθε όρου που εμφανίζεται σε ένα κείμενο είναι απλά αναλογικό στην συχνότητα του όρου αυτού» [48].

Το σύνολο των λέξεων που κρατήσαμε ονομάζεται **vocabulary.** Αυτό γίνεται αναθέτοντας σε κάθε λέξη έναν μοναδικό αριθμό. Έτσι, δημιουργούμε ένα τεράστιο πίνακα σταθερού μήκους, με γραμμές το κάθε κείμενο και στήλες το σύνολο των λέξεων που επιλέξαμε. Η τιμή κάθε κελιού αναπαριστά το πόσες φορές η κάθε λέξη φάνηκε σε κάθε κείμενο.

Για παράδειγμα, στα εξής documents:

1. I love dogs.

2. I hate dogs and knitting.

3. Knitting is my hobby and my passion.

Έχουμε το εξής bag of words:

	I	love	dogs	hate	and	knitting	is	my	hobby	passion
Doc 1	1	1	1							
Doc 2	1		1	1	1	1				
Doc 3					1	1	1	2	1	1

Πίνακας 1: *Διάνυσμα Bag-of-Words*

Υλοποιούμε το Bag-of-Words μέσω της βιβλιοθήκης Scikit-Learn με τον CountVectorizer().

2.2.4.2 TF-IDF Vectorizing

Η καταμέτρηση λέξεων είναι καλός αρχικός τρόπος, αν και είναι πολύ απλός.

Ένα σημαντικό πρόβλημα με απλές καταμετρήσεις, είναι πως κάποιες λέξεις, όπως το "the" θα εμφανιστούν πολλές φορές με μεγάλη συχνότητα, χωρίς να έχει ουσιαστικό βάρος στο μοντέλο που θα φτιάξουμε.

Ένας άλλος τρόπος υπολογισμός συχνοτήτων είναι ο TF-IDF. Ο TF-IDF είναι ακρωνύμιο για "*Term Frequency – Inverse Document*" Frequency.

- **Term Frequency**: Περιγράφει πόσες φορές υπάρχει ο όρος στο κάθε document. Για να γίνει κανονικοποίηση του όρου μεταξύ documents με διαφορετικό αριθμό λέξεων, διαιρούμε τη συχνότητα αυτή με τον συνολικό αριθμό των λέξεων του κάθε document.
  - TF (t) = (Συχνότητα εμφάνισης λέξης *t* σε ένα document) / (Συνολικός αριθμός λέξεων του document).
- **Inverse Document Frequency**: Περιγράφει σε πόσα διαφορετικά κείμενα υπάρχει η λέξη αυτή, σε αντεστραμμένη κλασματική μορφή.
  - IDF (t) = $\log_e$ (Συνολικός αριθμός documents / Σύνολο documents που περιλαμβάνουν τον όρο *t*.).

Με τον TF-IDF αναζητούμε λέξεις συχνές σε ένα κείμενο, οι οποίες να μην είναι συχνές σε άλλα κείμενα. Δηλαδή να έχουμε μεγάλο Term Frequency και χαμηλό Document Frequency, συνεπώς μεγάλο Inverse Document Frequency [49].
Υλοποιούμε την TF-IDF ζύγιση μέσω της βιβλιοθήκης Scikit Learn και της μεθόδου TFIDFVectorizer().

2.2.5 Παραμετροποίηση Διανυσμάτων

Η παραμετροποίηση στους vectorizers μας βοηθά να πετύχουμε το καλύτερο δυνατό αποτέλεσμα, με βάση το σετ δεδομένων μας.

Θα παρουσιάσουμε παραμέτρους, με βάση τους οποίους μπορούμε να επιλέξουμε συγκεκριμένους όρους από το κείμενο μας.:

2.2.5.1 N-grams

Το πρόβλημα με το Bag of Words είναι ότι χάνεται η σειρά των λέξεων που θέλουμε να αναλύσουμε. Αυτό δημιουργεί προβλήματα, καθώς ο τρόπος γραφής σπάνια εξαντλεί το συναισθηματισμό αλλά και το περιεχόμενο που θέλει να προσδώσει σε μία μόνο λέξη. Αυτός είναι ο βασικός στόχος των n-grams. Μπορούμε να συλλάβουμε τη δομή της γλώσσας από μία στατιστική άποψη, όπως ποια λέξη είναι πιθανό να ακολουθήσει κάποια άλλη.
Όσο μεγαλύτερο είναι το n-gram, τόσο περισσότερο περιεχόμενο μπορούμε να αξιοποιήσουμε σαν ενιαίο σύνολο, σε βάρος της απόδοσης και υπολογιστικής ισχύος ώστε να εκπαιδεύσουμε το μοντέλο μας [50].
Για παράδειγμα:

1. San Francisco (2-gram)
2. The Three Musketeers (3-gram)
3. She stood up slowly (4-gram)

Όπως καταλαβαίνουμε, οι προτάσεις (1) και (2) είναι πιο συχνές από ότι η (3).

Τώρα, αν αναθέσουμε μία πιθανότητα για την εμφάνιση ενός N-gram ή την πιθανότητα εμφάνισης μίας λέξης σαν μέρος ακολουθίας λέξεων (πχ “Francisco” μετά το “San”), έχουμε πολλά πιθανά κέρδη.

Πρώτα από όλα , μπορούμε να κωδικοποιήσουμε το n-gram σαν μία ολοκληρωμένη οντότητα, πχ “high school”.

Μπορούμε να κάνουμε πρόβλεψη λέξης. Για παράδειγμα στη πρόταση “Please hand over your” μπορούμε με μηχανική μάθηση να προβλέψουμε ότι είναι πολύ πιο πιθανό να έχουμε τη λέξη ”test” ή ” assignment ”.

Μετέπειτα, μπορούμε να το αξιοποιήσουμε για να κάνουμε ορθογραφικό έλεγχο. Για παράδειγμα, η φράση “drink coffee” μπορεί να διορθωθεί σε ”drink coffee”, εάν γνωρίζουμε πως η λέξη “coffee” έχει μεγάλη πιθανότητα να εμφανιστεί μετά τη λέξη “drink”, καθώς και γιατί οι λέξεις ”cofe” “coffee” έχουν πολλά κοινά γράμματα μεταξύ τους.

2.2.5.2 Max DF

Το Max Document Frequency είναι το όριο για τη μέγιστη συχνότητα κειμένων ενός token. Βοηθά ώστε να ελεγχθούν οι λέξεις που εμφανίζονται πολύ συχνά σε ένα συγκεκριμένο κείμενο, δηλαδή λέξεις που αφορούν κυρίως ένα κείμενο, παρά μια συνολική συναισθηματική πληροφορία.

2.2.5.3 Max Features

Η δημιουργία διανυσμάτων μετατρέπει κάθε λέξη των κειμένων μας σε στοιχείο που θα χρησιμοποιηθεί για μηχανική μάθηση. Καταλαβαίνουμε πως ο όγκος των δεδομένων αυτών είναι τεράστιος, χιλιάδες ίσως και εκατομμύρια στοιχεία. Η παράμετρος Max Features βάζει ένα άνω φράγμα στο πόσα στοιχεία θα χρησιμοποιηθούν στο διάνυσμα μας.

2.3 Συναισθηματική Ανάλυση στην Επεξεργασία Φυσικής Γλώσσας

Για την συγγραφή της διπλωματικής χρησιμοποιήσαμε εργαλεία της βιβλιοθήκης NLTK που παρουσιάζουν συναισθηματικό σκορ για τα δεδομένα που τους παρέχουμε.

2.3.1 Προσέγγιση μέσω SentiWordNet

Η προσέγγιση μέσω λεξικού είναι η πιο βασική προσέγγιση που χρησιμοποιείται για την συναισθηματική ανάλυση για ένα κείμενο. Στην προσέγγιση αυτή, έχουμε ένα λεξικό με λέξεις μαζί με ένα προκαθορισμένο σκορ, το οποίο αποκαλούμε **polarity score**. Χρησιμοποιούμε το άθροισμα των σκορ σε επίπεδο προτάσεων ή κειμένων για να αντλήσουμε το συναίσθημα.

Στην διπλωματική αυτή χρησιμοποιήσαμε ένα λεξικό δημόσιας πρόσβασης, το **SentiWordNet** [51]**.** Το SentiWordNet χρησιμοποιείται για opinion mining και ενσωματώνεται και αυτό μέσω της βιβλιοθήκης NLTK.

```
In [3]: print "pos score: ", swn.senti_synsests("happy", 'a')[0].pos_score()
        print "neg score: ", swn.senti_synsets("happy", 'a')[0].neg_score()

        pos score:  0.875
        neg score:  0.0
```

***Εικόνα 7**: Εμφάνιση συναισθηματικών σκορ για τη λέξη Happy:*

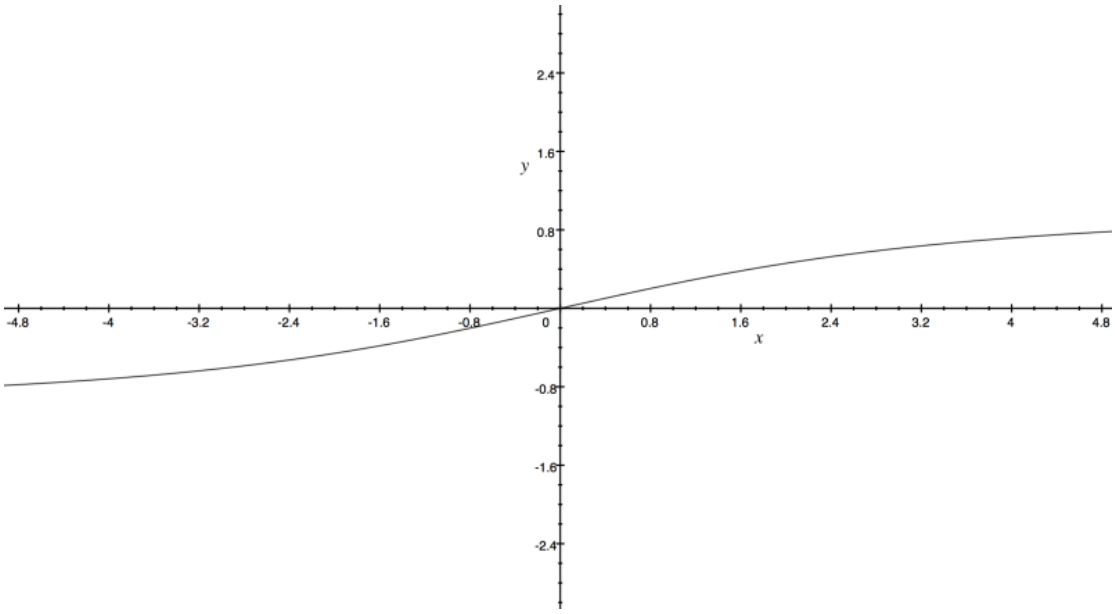
2.3.2 Προσέγγιση μέσω Vader

Το VADER αποτελεί ένα εργαλείο συναισθηματικής ανάλυσης βασισμένο σε απλούς κανόνες. Αποδείχτηκε εξαιρετικά ισχυρό εργαλείο το οποίο συμπεριλαμβάνει πολλούς διαφορετικούς κανόνες για την Επεξεργασία Φυσικής Γλώσσας, ώστε να αποτυπώνει το συναισθηματικό περιεχόμενο όσο το δυνατόν πιο κοντά γίνεται στην ανθρώπινη κατανόηση Φυσικής Γλώσσας [52]. Η συναισθηματική ανάλυση με **Vader** δέχεται είσοδο ένα κείμενο (ιδανικά μικρής έκτασης) και επιστρέφει σαν έξοδο συναισθηματικό σκορ στο εύρος -1 με 1, από το πιο αρνητικό στο πιο θετικό.

Το συναισθηματικό σκορ υπολογίζεται αθροίζοντας τα συναισθηματικά σκορ της κάθε λέξης σε κάθε πρόταση, η οποία βρίσκεται μέσα σε ένα υπάρχον dictionary. Η κάθε λέξη έχει ένα προαποθηκευμένο σκορ από -4 έως 4, το οποίο μετέπειτα κανονικοποιείται σε -1 με 1 με την εξής συνάρτηση:

$$\frac{x}{\sqrt{x^2+\alpha}}$$

Όπου x είναι το άθροισμα των συναισθηματικών σκορ της κάθε μίας λέξης μιας πρότασης και a είναι μια παράμετρος κανονικοποίησης που έχει τεθεί στο 15. Η κανονικοποίηση φαίνεται παρακάτω:



***Εικόνα 8**: Αποτέλεσμα Κανονικοποίησης*

Βλέπουμε πως όσο μεγαλώνει το x προς οποιαδήποτε κατεύθυνση, τόσο προσεγγίζει το -1 ή το 1. Αν έχουμε πολλές λέξεις στο document μας και εφαρμόσουμε συναισθηματική ανάλυση Vader, τότε το αποτέλεσμα θα βγει κοντά στο 1 ή το -1. Για το λόγο αυτό η συναισθηματική ανάλυση Vader είναι αποδοτικότερη σε μικρά κείμενα, tweets, προτάσεις, παρά μεγάλα κείμενα.

Φυσικά, τα λεξικολογικά στοιχεία δεν είναι τα μόνα που επηρεάζουν το συναίσθημα σε μία πρόταση. Σημεία στίξης, κεφαλαία γραφή, εναντιωματικοί σύνδεσμοι καθώς και επιθετικοί προσδιορισμοί μπορούν να ενισχύσουν ή να αποσβέσουν την ένταση ενός συναισθήματος. Η ανάλυση Vader συμπεριλαμβάνει **5 ευριστικές συναρτήσεις** για να ενσωματώσει το ζήτημα αυτό:

Η *πρώτη* ευριστική είναι τα **σημεία στίξης**. Αρχικά, ο αλγόριθμος υπολογίζει το συναισθηματικό σκορ της πρότασης. Αν το σκορ είναι θετικό, τότε για κάθε θαυμαστικό προστίθεται μια τιμή 0.292. Για κάθε ερωτηματικό, το 0.18. Αντίστοιχα, αν το σκορ είναι αρνητικό, ο Vader αφαιρεί αυτές τις τιμές. Οι τιμές αυτές καθιερώθηκαν εμπειρικά μέσω δοκιμής και λάθους ανά τα χρόνια.

```
#Baseline sentence
sentiment_analyzer_scores('The food here is good')
```

```
The food here is good------------------- {'neg': 0.0, 'neu': 0.58, 'pos': 0.42,
'compound': 0.4404}
```

```
#Punctuation
print(sentiment_analyzer_scores('The food here is good!'))
print(sentiment_analyzer_scores('The food here is good!!'))
print(sentiment_analyzer_scores('The food here is good!!!'))
```

```
The food here is good!------------------- {'neg': 0.0, 'neu': 0.556, 'pos': 0.44
4, 'compound': 0.4926}
None
The food here is good!!------------------ {'neg': 0.0, 'neu': 0.534, 'pos': 0.46
6, 'compound': 0.5399}
None
The food here is good!!!------------------ {'neg': 0.0, 'neu': 0.514, 'pos': 0.48
6, 'compound': 0.5826}
None
```

***Εικόνα 9**: Vader και σημεία στίξης*

Η *δεύτερη* ευριστική είναι τα **κεφαλαία γράμματα**. Μία πρόταση τύπου “AMAZING performance” είναι σίγουρα πιο έντονη από το “amazing performance”. Συνεπώς ο Vader λαμβάνει υπόψιν του το στοιχείο αυτό, αυξάνοντας ή μειώνοντας το σκορ κατά 0.733, ανάλογα αν το αθροιστικό σκορ είναι θετικό ή αρνητικό, αντίστοιχα.

```
#Baseline sentence
sentiment_analyzer_scores('The food here is great!')
```

```
The food here is great!---------------- {'neg': 0.0, 'neu': 0.477, 'pos': 0.52
3, 'compound': 0.6588}
```

```
#Capitalisation
sentiment_analyzer_scores('The food here is GREAT!')
```

```
The food here is GREAT!---------------- {'neg': 0.0, 'neu': 0.438, 'pos': 0.56
2, 'compound': 0.729}
```

***Εικόνα 10**: Vader και κεφαλαία*

Η *Τρίτη* ευριστική αφορά λέξεις τροποποίησης έντασης, όπως «πολύ». Συνήθως προηγείται από επιθετικούς προσδιορισμούς. Προτάσεις τύπου «sort of cute" έχουν άλλο συναισθηματικό βάρος από την πρόταση "cute". Η ένταση της τροποποίησης εξαρτάται από την απόσταση της λέξης τροποποίησης από την λέξη που τροποποιείται. Μεγαλύτερη απόσταση σημαίνει λιγότερη ένταση, και αντίστροφα. Μια λέξη τροποποίησης που είναι δίπλα στην λέξη που τροποποιείται, προσθέτει ή αφαιρεί βάρος 0.293. Για κάθε παραπάνω λέξη ενδιάμεσα τους, το βάρος που θα προστεθεί ή θα αφαιρεθεί μειώνεται κατά 5%. Δηλαδή, για μία απόσταση 3 λέξεων θα έχουμε το 85% του 0.293.

```
#Baseline sentence
sentiment_analyzer_scores('The service here is good')
```

```
The service here is good---------------- {'neg': 0.0, 'neu': 0.58, 'pos': 0.42,
'compound': 0.4404}
```

```
#Degree Modifiers
print(sentiment_analyzer_scores('The service here is extremely good'))
print(sentiment_analyzer_scores('The service here is marginally good'))
```

```
The service here is extremely good------ {'neg': 0.0, 'neu': 0.61, 'pos': 0.39,
'compound': 0.4927}
None
The service here is marginally good----- {'neg': 0.0, 'neu': 0.657, 'pos': 0.34
3, 'compound': 0.3832}
None
```

***Εικόνα 11**: Vader και λέξεις ποσοτικής τροποποίησης*

Η *τέταρτη* ευριστική είναι η αλλαγή στη πόλωση με αντιθετικούς όρους όπως "but". Η λέξη αλλά συνδέει 2 προτάσεις με αντίθετα συναισθήματα, από τα οποία όμως το κύριο είναι το δεύτερο. Για παράδειγμα, στο "I love you, but I don't want to be with you anymore." Η πρώτη πρόταση έχει θετικό βάρος, η δεύτερη έχει αρνητικό βάρος αλλά προφανώς η δεύτερη έχει μεγαλύτερη συναισθηματική βαρύτητα. Ο Vader περιλαμβάνει έλεγχο για τέτοιους όρους, το οποίο όταν βρεθεί δίνει βάρος 50% στην αξία της πρώτης πρότασης, και 150% στο βάρος της δεύτερης πρότασης.

```
#Conjunctions
sentiment_analyzer_scores('The food here is great, but the service is horrible')
```

```
The food here is great, but the service is horrible {'neg': 0.31, 'neu': 0.523,
'pos': 0.167, 'compound': -0.4939}
```

***Εικόνα 12:** Vader και αντιθετικοί όροι*

Η *Πέμπτη* ευριστική αναλύει το tri-gram της κάθε λέξης με σκοπό να βρει κάποιον συναισθηματικά αντιθετικό όρο. Αν βρεθεί, η άρνηση πολλαπλασιάζει το σκορ της λέξης που βρήκαμε με την εμπειρικά καθορισμένη τιμή του 0.74.

Πλεονεκτήματα του Vader

- Είναι πολύ αποτελεσματικό σε κείμενο από κοινωνικά δίκτυα.
- Δεν χρειάζεται σετ δεδομένων, μηχανική μάθηση και εκπαίδευση. Δημιουργήθηκε έχοντας βάση ένα λεξικό.
- Είναι γρήγορο, και ικανό να χρησιμοποιηθεί όσο κάνουμε stream δεδομένα.

ΚΕΦΑΛΑΙΟ 3. ΜΗΧΑΝΙΚΗ ΜΑΘΗΣΗ

3.1 Εισαγωγή

Η μηχανική μάθηση (Machine Learning) αποτελεί κλάδο της Τεχνητής Νοημοσύνης, και ασχολείται με την μελέτη και κατασκευή αλγορίθμων οι οποίοι μπορούν να μαθαίνουν από δεδομένα και να κάνουν προβλέψεις σε σχέση με αυτά.

Η κύρια ιδέα πίσω από τη μηχανική μάθηση, είναι η δημιουργία ενός μοντέλου βασιζόμενο σε συγκεκριμένους αλγορίθμους το οποίο μπορεί να κάνει προβλέψεις και να λάβει αποφάσεις για συγκεκριμένα προβλήματα, χωρίς να έχει εκπαιδευτεί από κάποιο χρήστη για τον συγκεκριμένο σκοπό. Ουσιαστικά, εκπαιδεύουμε ένα σύστημα, πάνω σε ένα σετ δεδομένων, ώστε με βάση αυτά, να δεχτεί γνώση αντιλαμβανόμενο την συσχέτιση μεταξύ τους και τελικά να είναι ικανό, να προβλέψει νέα δεδομένα

Το ενδιαφέρον για μεθόδους Μηχανικής Μάθησης έχει αυξηθεί μαζί με την αύξηση των δεδομένων στα οποία έχουμε πρόσβαση. Κοινωνικά Δίκτυα, Blogs, Forums,.είναι μόλις μερικά από τα δομικά στοιχεία στα οποία ο κόσμος έχει δείξει ενδιαφέρον παράγοντας τεράστιο όγκο (κειμενικών) πληροφοριών.

Οι αλγόριθμοι που χρησιμοποιούνται στην Μηχανική Μάθηση προέρχονται από τα πεδία της στατιστικής και των μαθηματικών. Ένα μεγάλο πλεονέκτημα είναι πως η διαδικασία άντλησης γνώσης είναι αυτοματοποιημένη. Παρόλα αυτά, το πρόβλημα της αυτοματοποιημένης κατηγοριοποίησης παραμένει δύσκολο.

Ο τομέας της Μηχανικής Μάθησης περιλαμβάνει τρεις τρόπους μάθησης, ανάλογα με τη φύση του εκάστοτε προβλήματος: επιβλεπόμενη μάθηση, μη επιβλεπόμενη μάθηση και ενισχυτική μάθηση.

Πιο αναλυτικά:

- **Επιβλεπόμενη Μάθηση** (Supervised Learning) είναι η διαδικασία όπου ο αλγόριθμος κατασκευάζει μια συνάρτηση που απεικονίζει δεδομένες εισόδους (σύνολο εκπαίδευσης) σε γνωστές επιθυμητές εξόδους, με απώτερο στόχο τη γενίκευση της συνάρτησης αυτής και για εισόδους με άγνωστη έξοδο. Χρησιμοποιείται σε προβλήματα: ο Ταξινόμησης (Classification) ο Πρόγνωσης (Prediction) ο Διερμηνείας (Interpretation)
- **Μη Επιβλεπόμενη Μάθηση** (Unsupervised Learning), όπου ο αλγόριθμος κατασκευάζει ένα μοντέλο για κάποιο σύνολο εισόδων υπό μορφή παρατηρήσεων χωρίς να γνωρίζει τις επιθυμητές εξόδους. Χρησιμοποιείται σε προβλήματα: ο Ανάλυσης Συσχετισμών (Association Analysis) ο Ομαδοποίησης (Clustering)
- **Ενισχυτική Μάθηση** (Reinforcement Learning), όπου ο αλγόριθμος μαθαίνει μια στρατηγική ενεργειών μέσα από άμεση αλληλεπίδραση με το περιβάλλον. Χρησιμοποιείται κυρίως σε προβλήματα Σχεδιασμού (Planning), όπως για παράδειγμα ο έλεγχος κίνησης ρομπότ και η βελτιστοποίηση εργασιών σε εργοστασιακούς χώρους.

Αφού το πρόβλημα που αντιμετωπίζουμε είναι πρόβλημα κατηγοριοποίησης, θα αναλύσουμε την επιβλεπόμενη .

3.2 Ανάλυση Επιβλεπόμενης Μάθησης

Παρουσιάζουμε στον υπολογιστή ένα σετ δεδομένων εισόδου X μαζί με το επιθυμητό αποτέλεσμα εξόδου Y. Στόχος είναι ο υπολογιστής να φτιάξει ένα κανόνα συσχέτισης εισόδου – εξόδου, μία συνάρτηση Y = f(X). Μετέπειτα, παρουσιάζοντας νέα δεδομένα εισόδου X στον υπολογιστή αυτός μπορεί να προβλέψει την έξοδο Y. Αυτό γίνεται χρησιμοποιώντας διαφορετικούς ταξινομητές που θα αναλύσουμε μετέπειτα.

Η επιβλεπόμενη μάθηση προϋποθέτει πως το σετ δεδομένων εισόδου μας έχει κατηγοριοποιηθεί με την σωστή έξοδο. Για παράδειγμα, ένας αλγόριθμος κατηγοριοποίησης που μαθαίνει να αναγνωρίζει είδη ζώων πάνω σε εικόνες, προϋποθέτει πως στα δεδομένα εκπαίδευσης που δώσαμε, σωστά ονομάσαμε κάθε ζώο με βάση το είδος του.

Η *κατηγοριοποίηση* που χρησιμοποιούμε στην διπλωματική εργασία αυτή, θεωρείται η δημοφιλέστερη τεχνική Εξόρυξης Γνώσης με εφαρμογές σε πολλά επιστημονικά πεδία. Γενικά, αφορά στη διαδικασία ταξινόμησης αντικειμένων σε προκαθορισμένες κατηγορίες, οι οποίες συχνά αναφέρονται και ως κλάσεις. Ας υποθέσουμε ότι τα δεδομένα που αφορούν σε ένα πρόβλημα παρουσιάζονται με τη μορφή εγγραφών (x,y), όπου το διάνυσμα x αντιστοιχεί στο σύνολο των μεταβλητών του δείγματος, ενώ το y αντιστοιχεί στη μεταβλητή απόφασης (κλάση ή ετικέτα) [53].

Τότε ορίζουμε ως κατηγοριοποίηση: «τη διαδικασία μάθησης μιας συνάρτησης στόχου f η οποία αντιστοιχεί κάθε διάνυσμα μεταβλητών x σε ένα προκαθορισμένο πλήθος κλάσεων y»

Η μεταβλητή y παίρνει συνήθως διακριτές ή κατηγορικές τιμές. Στην περίπτωση κατά την οποία η μεταβλητή y παίρνει οποιαδήποτε τιμή σε ένα διάστημα πραγματικών αριθμών [α, β] τότε έχουμε το πρόβλημα της παλινδρόμησης. Η συνάρτηση f αναφέρεται συχνά και ως μοντέλο ταξινόμησης ή κατηγοριοποίησης και χρησιμοποιείται συνήθως για περιγραφή ή πρόβλεψη. Για τη δημιουργία ενός μοντέλου ταξινόμησης απαιτείται ένα σύνολο δεδομένων (x,y) με γνωστές κλάσεις, το οποίο ονομάζεται σύνολο δεδομένων εκπαίδευσης.

3.3 Αλγόριθμοι ταξινόμησης

3.3.1 Naive Bayes

Ο Naive Bayes [54] είναι μια οικογένεια πιθανοτικών ταξινομητών, οι οποίοι στηρίζονται στο θεώρημα του Bayes, θεωρώντας, κατόπιν απλούστευσης, ότι έχουμε ανεξαρτησία μεταξύ των δεδομένων εισόδου. Ανάλογα με τον τύπο δεδομένων εισόδου και εξόδου, χρησιμοποιούμε διαφορετική παραλλαγή του, η οποία θεωρεί διαφορετική κατανομή: Πολυνυμιακή, Γκαουσιανή κτλ. Αποτελεί μια απλή, γρήγορη και αρκετά αποτελεσματική μέθοδο ταξινόμησης.

Επειδή τα δεδομένα εισόδου μας ήταν πίνακες συχνότητες λέξεων στο κείμενο, όπως συνήθως γίνεται στην αναπαράσταση κειμένου, και τα δεδομένα εξόδου ήταν μία δυαδική κλάση, χρησιμοποιήσαμε τον Multinomial Naive Bayes που θεωρεί πολυνωμιακή κατανομή, όπου οι εκθέτες του πολυωνύμου είναι οι συχνότητες κάθε λέξης.

Σύμφωνα με το θεώρημα Bayes, ισχύει ότι:

P (A | B) = P(A) P(B | A) P(B)

Όπου :

Το P(A|B) η δεσμευμένη πιθανότητα του ενδεχομένου A, δεδομένου του ενδεχομένου B.

Το P (A) είναι η πιθανότητα πραγματοποίησης του ενδεχομένου A και είναι γνωστή ως «εκ των προτέρων πιθανότητα του A».

Το P(B|A) είναι η δεσμευμένη πιθανότητα του ενδεχομένου B, δεδομένου του A. Η πιθανότητα αυτή είναι δυνατόν να υπολογιστεί από τη γνώση που διαθέτουμε για το συγκεκριμένο πρόβλημα.

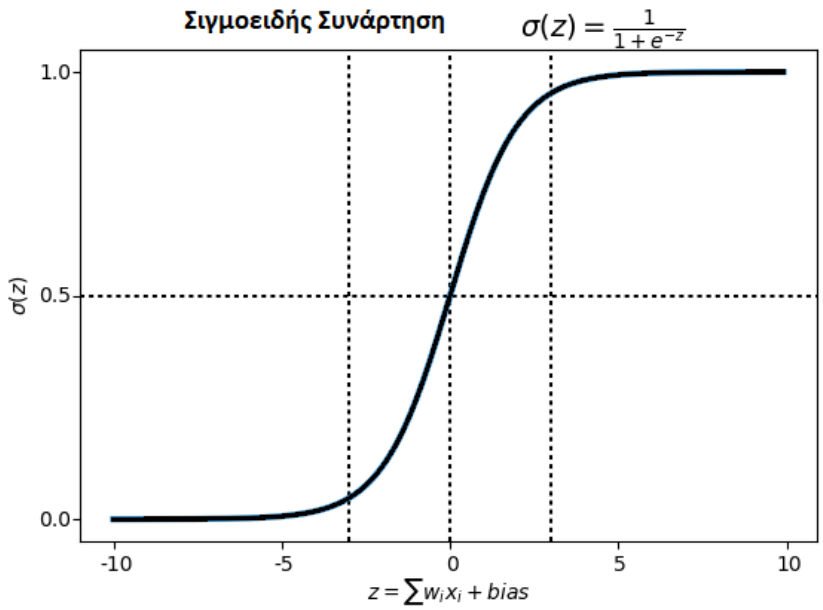
Το P(B) είναι η πιθανότητα πραγματοποίησης του ενδεχομένου B.

Ο ταξινομητής Bayes χρησιμοποιείται για την εκτίμηση της πιθανότητας ενός στιγμιότυπου να ανήκει σε μια από τις προκαθορισμένες κλάσεις υπό την υπόθεση ότι τα χαρακτηριστικά είναι μεταξύ τους ανεξάρτητα. Η υπόθεση της ανεξαρτησίας των χαρακτηριστικών δεν ισχύει πάντοτε, όμως απλοποιεί κατά πολύ τους υπολογισμούς οδηγώντας σε καλή εκτίμηση της πιθανότητας χωρίς να απαιτεί μεγάλο σύνολο εκπαίδευσης.

Εάν κάποιο χαρακτηριστικό ή κάποια κλάση δεν εμφανιστεί ποτέ στα δεδομένα εκπαίδευσης, τότε η στατιστική ανάλυση συχνότητας θα υπολογιστεί 0. Κάτι που είναι προβληματικό, καθώς θα μηδενίσει όλες τις υπόλοιπες πιθανότητες, σαν μέρος των πολλαπλασιαστέων. Συνεπώς είναι επιθυμητό, να έχουμε ένα μικρό sample correction που λέγεται ψευδομέτρηση σε κάθε πιθανοτική εκτίμηση, με τρόπο τέτοιο ώστε καμία πιθανότητα να είναι ακριβώς 0. Ο τρόπος κανονικοποίησης αυτός του Naïve Bayes λέγεται Laplace Smoothing.

3.3.2 Λογιστική Παλινδρόμηση

Η λογιστική παλινδρόμηση αποτελεί έναν ταξινομητή που χρησιμοποιείται κυρίως για προβλήματα κατηγοριοποίησης και βασίζεται στις πιθανότητες. Η συνάρτηση κόστους της βασίζεται στην σιγμοειδή συνάρτηση [55].



***Εικόνα 13**: Σιγμοειδής Συνάρτηση*

Οι μεταβλητές εισόδου συνδυάζονται γραμμικά χρησιμοποιώντας βάρη (B) ώστε να προβλεφθούν τα δεδομένα εξόδου. Χρησιμοποιούμε την λογιστική παλινδρόμηση όταν θέλουμε να μοντελοποιήσουμε τα δεδομένα εξόδου σαν δυαδική τιμή. Η λογιστική παλινδρόμηση χρησιμοποιείται όταν κάποιες από τις ανεξάρτητες μεταβλητές είναι αριθμητικές ή βαθμωτές και κάποιες άλλες ονομαστικές ενώ η εξαρτημένη μεταβλητή είναι δυαδική ή διχοτομική με μόνο δύο δυνατές τιμές (0-1 ή Επιτυχία-Αποτυχία). Το αποτέλεσμα είναι η πιθανότητα να συμβεί το ένα (το θετικό) αποτέλεσμα.

Η μαθηματική εξίσωση για την πιθανότητα αυτή είναι

$$E(y) = \frac{1}{1 + \exp\left[-\left(b_0 + b_1 x_1 + b_2 x_2 + \ldots + b_k x_k\right)\right]}$$

Με E(y): Η πιθανότητα του να είναι y = το επιθυμητό αποτέλεσμα, x1, x2, …, xk είναι οι ανεξάρτητες μεταβλητές από τις οποίες εξαρτάται η y, bo είναι η σταθερά της εξίσωσης και b1, b2, … bk είναι οι συντελεστές παλινδρόμησης. Για τον υπολογισμό της E(y) πρέπει να προσδιοριστούν πριν οι συντελεστές παλινδρόμησης.

Είναι σημαντικό να σημειώσουμε το πλενέκτημα της λογισιτκής παλινδρόμησης, καθώς δεν προυποθέτει κάποια κατανομή για τα δεδομένα μας.

3.3.3 Μάθηση Βασισμένη σε Στιγμιότυπα

Οι μέθοδοι μάθησης βασισμένες σε στιγμιότυπα (Instance Based methods) στηρίζονται στην ομοιότητα μεταξύ των δεδομένων, η οποία υπολογίζεται με τη χρήση κατάλληλης μετρικής. Από τους γνωστότερους αλγόριθμους κατηγοριοποίησης βάσει στιγμιότυπων είναι αυτός των **πλησιέστερων γειτόνων**, ο οποίος χρησιμοποιείται και για προβλήματα παλινδρόμησης. Ο αλγόριθμος των k πλησιέστερων γειτόνων (kNN) βρίσκει k στιγμιότυπα στο σύνολο εκπαίδευσης τα οποία βρίσκονται πλησιέστερα σε ένα συγκεκριμένο στιγμιότυπο. Στη συνέχεια αποδίδει το στιγμιότυπο σε εκείνη την κλάση η οποία υπερέχει μεταξύ των k στιγμιότυπων [56] [57].

Σημαντικές παράμετροι οι οποίες καθορίζουν την αποτελεσματικότητα της μεθόδου αποτελούν:

• Το σύνολο των δεδομένων εκπαίδευσης.

• Η τιμή του k, η οποία αντιστοιχεί στο πλήθος των πλησιέστερων γειτόνων (στιγμιότυπων).Συνήθως επιλέγουμε περιττό k, ώστε στην δυαδική ψηφοφορία να υπάρχει ξεκάθαρος νικητής.

•Η μετρική που θα χρησιμοποιηθεί για τον υπολογισμό της απόστασης μεταξύ των στιγμιότυπων. Αξίζει να σημειωθεί ότι η μέθοδος των k πλησιέστερων γειτόνων δύναται να χρησιμοποιηθεί για δύσκολα προβλήματα κατηγοριοποίησης με κατάλληλη τροποποίηση.

3.3.4 Decision Trees

Τα «Δένδρα Απόφασης/ταξινόμησης» (Decision/Classification trees) αποτελούν έναν αλγόριθμο μάθησης, τα οποία δημιουργούν ένα μοντέλο απλών κανόνων απόφασης με βάση τα δεδομένα εισόδου[58]. Κάθε κόμβος που δεν είναι φύλλο συσχετίζεται με μία διαίρεση, μία απόφαση. Τα δεδομένα που εισέρχονται στον κόμβο θα χωριστούν σε διαφορετικές κατευθύνσεις του δέντρου, ανάλογα με τις διαφορετικές τιμές που έχουν στα δεδομένα εισόδου. Κάθε κόμβος-φύλλο συνδέεται με μία κατηγορία, η οποία και ανατίθεται στα νέα δεδομένα όταν αυτά εισέρχονται στον κόμβο αυτόν. Όταν επιχειρείται μια πρόβλεψη, ο αλγόριθμος απλά ακολουθεί τους κόμβους του δέντρου, ανάλογα με τα δεδομένα εισόδου μας, μέχρι να βρει έναν κόμβο φύλλο. Τα μοντέλα δέντρων δεν περιορίζονται σε ζητήματα ταξινόμησης, καθώς χρησιμοποιούνται με επιτυχία για διαφορετικά προβλήματα μηχανικής μάθησης

3.3.5 Τυχαία Δάση

Τα τυχαία δάση είναι ένας ταξινομητής που αποτελεί εξέλιξη των Δενδρών Απόφασης. Για την ακρίβεια αποτελείται από πολλά δέντρα απόφασης [59]. Υλοποιείται σε 4 βήματα και η λειτουργία του είναι η παρακάτω:

Αρχικά, αναπτύσσονται πολλά δέντρα απόφασης. Κάθε δέντρο δίνει μία ταξινόμηση «Το δέντρο ψηφίζει αυτήν την κλάση». Έτσι, κάθε κλάση έχει έναν αριθμό «ψήφων» (votes). Η τελική και οριστική ταξινόμηση γίνεται με το «δάσος» να διαλέγει την κλάση με τις περισσότερες votes Εύκολα προκύπτει το συμπέρασμα, ότι στα παραπάνω βήματα κύριο ρόλο παίζει το πώς ακριβώς αναπτύσσονται τα δέντρα απόφασης. Εφαρμόζοντας την τεχνική αυτή, σε μεθόδους ταξινόμησης, δίνεται η δυνατότητα να δημιουργηθούν από ένα σετ δεδομένων περισσότερα από ένα σετ εκπαίδευσης. Αυτός είναι και ο λόγος που χρησιμοποιείται στα τυχαία δάση, καθώς θέλουμε να

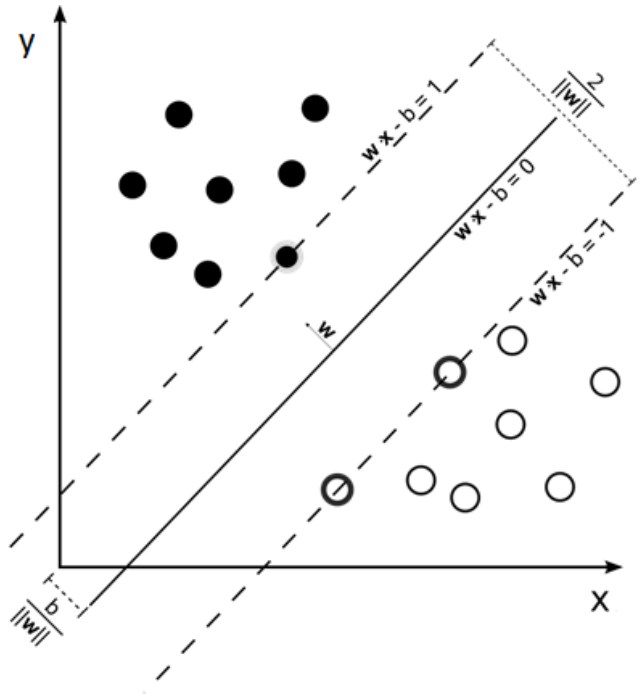
δημιουργήσουμε πολλά διαφορετικά δένδρα και επομένως πολλά σετ εκπαίδευσης (εφόσον κάθε δένδρο πρέπει να έχει το δικό του σετ εκπαίδευσης).

3.3.6 Μηχανές Υποστήριξης Διανυσμάτων

Οι Μηχανές Διανυσμάτων Υποστήριξης [60] [61] [62] είναι μια μέθοδος μηχανικής μάθησης που χρησιμοποιείται για δυαδικά προβλήματα ταξινόμησης και εφαρμόζεται με πολύ μεγάλη επιτυχία στην κατηγοριοποίηση των αρχείων κειμένου. Συγκαταλέγεται ανάμεσα στους πιο αποδοτικούς ταξινομητές, καθώς μπορεί να χειριστεί μεγάλα σύνολα χαρακτήρων όπως για παράδειγμα μεγάλα σε όγκο είδη κειμένου. Η λειτουργία του SVM περιγράφεται παρακάτω:

Αρχικά, χαρτογραφείται το σετ δεδομένων εκπαίδευσης, σε ένα πιθανό πολυδιάστατο χώρο διανυσμάτων. Μετέπειτα, προσπαθεί να εντοπίσει στον χώρο αυτό ένα πεδίο, ένα διάστημα, το οποίο να διαχωρίζει τα θετικά από τα αρνητικά παραδείγματα. Όταν βρεθεί ο χώρος αυτός, ο αλγόριθμος μπορεί στην συνέχεια να προβλέψει την κατηγορία ενός νέου δεδομένου ελέγχου, καθώς μπορεί να το τοποθετήσει στον χώρο που έφτιαξε, αναζητώντας σε ποια πλευρά του διαχωριστικού πεδίου θα βρίσκεται το νέο δεδομένο.

Άρα, είναι σημαντικό να διαλέξουμε ένα διαχωριστικό πεδίο, από τα πολλά υποψήφια που να ικανοποιεί την εξής συνθήκη: Ο SVM αλγόριθμος επιλέγει το πεδίο που διατηρεί το μεγαλύτερο διάστημα μεταξύ οποιουδήποτε σημείου στο εκπαιδευτικό σύνολο.



***Εικόνα 14**: SVM*

Αναλυτικότερα, όλα τα διανύσματα εισόδου μπορούν να χωριστούν από τα πεδία που φαίνονται παραπάνω. Κάποια διανύσματα της περιοχής του χώρου της μίας κατηγορίας είναι πιο κοντά στην περιοχή του χώρου μιας άλλης κατηγορίας. Βρίσκουμε τα σημεία λοιπόν, των διαφορετικών κλάσεων που έχουν την μικρότερη απόσταση μεταξύ τους. Η απόσταση αυτή, ορίζει τον χώρο, στον οποίο θα τρέξει η αναζήτηση του αλγορίθμου.

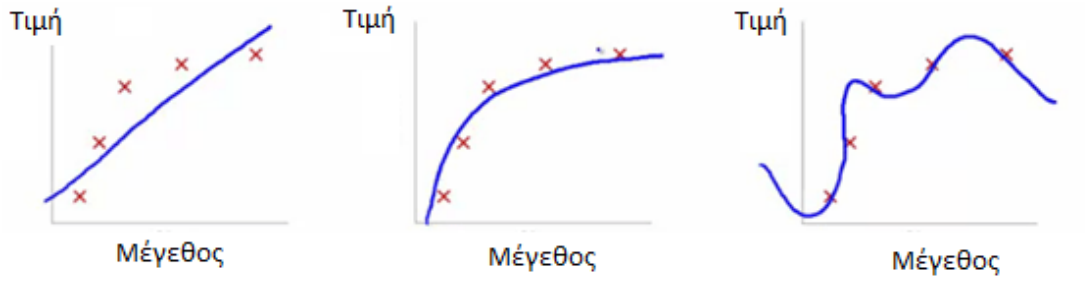
Τα διανύσματα αυτά, τα ονομάζονται support vectors (διανύσματα υποστήριξης) και φαίνονται κυκλωμένα στο παραπάνω σχήμα.

Ο στόχος του αλγόριθμου είναι να επιλέξει ένα διαχωριστικό πεδίο $(w \cdot x_i - b) = 0$ το οποίο μεγιστοποιεί το διάστημα μεταξύ του H1 $(w \cdot x_i - b) = -1$ και του H 2 $(w \cdot x_i - b) = 1$.

Μόλις αυτό βρεθεί, μπορούμε να εφαρμόσουμε την ταξινόμηση νέων στοιχείων.

3.4 Υπερπροσαρμογή / Υποπροσαρμογή

Το πρόβλημα της προσαρμογής στο σετ δεδομένων μας περιγράφει την ευελιξία των μοντέλων που φτιάχνουμε, καθώς και την δυνατότητα τους να εκφράσουν συναρτησιακά όσο περισσότερα δεδομένα εισόδου γίνεται. Κατά πόσο δηλαδή, η εκπαίδευση του μοντέλου μας προσαρμόζεται στα δεδομένα εκπαίδευσης [63]. Παρουσιάζουμε ένα παράδειγμα στο οποίο προσπαθούμε να βρούμε τη σχέση μεταξύ μεγέθους και τιμής προιόντων.



Εικόνα 15: *Παραδείγματα Υπερπροσαρμογής / Ορθής Προσαρμογής / Υποπροσαρμογής*

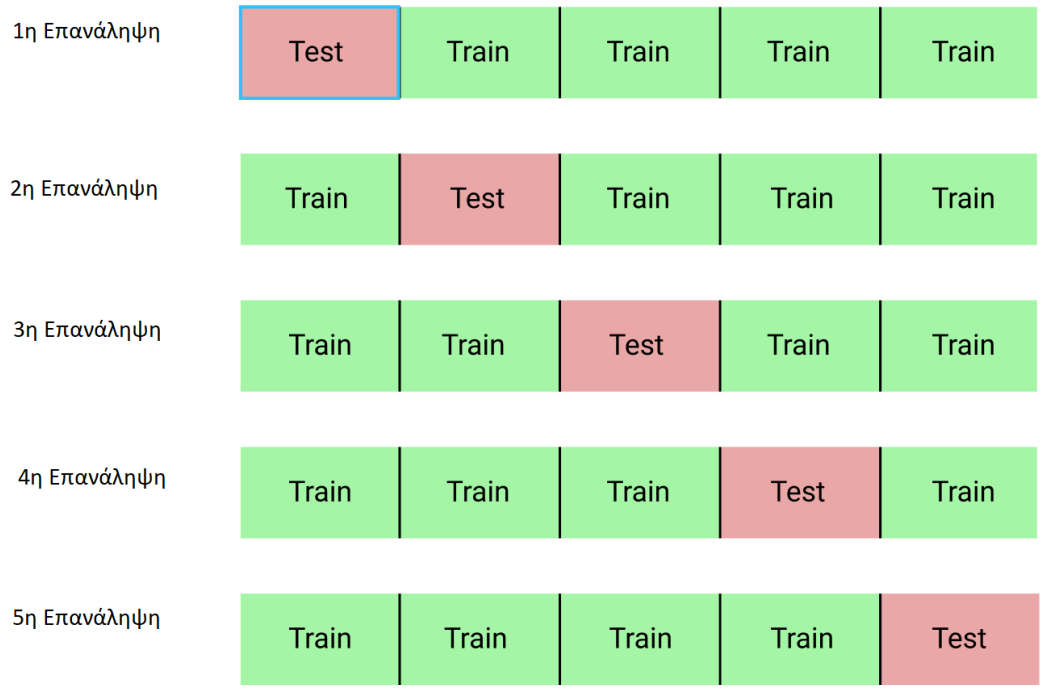
1. Παρουσιάζουμε μία γραμμική σχέση μεταξύ μεγέθους και τιμής, όπως φαίνεται στο σχήμα 1. Όπως βλέπουμε, η συνάρτηση μας έχει λάθη, καθώς έχει μεγάλες αποστάσεις από τα δεδομένα εισόδου που δώσαμε. Το μοντέλο αυτό λοιπόν, δεν θα ανταπεξέλθει επαρκώς σε νέες τιμές. Το μοντέλο αυτό είναι **underfitted, καθώς αποτυγχάνει να βρει ορθή σχέση μεταξύ των δεδομένων μας**.
2. Στην δεύτερη περίπτωση, έχουμε βρει ορθή σχέση μεταξύ των δεδομένων μας. Το μοντέλο είναι έτοιμο και ικανό να προβλέψει νέες τιμές για μεγέθη που θα του δώσουμε.
3. Στην Τρίτη γραφική παράσταση, βρίσκουμε και παρουσιάζουμε μία σχέση μεταξύ των δεδομένων μας η οποία έχει σχεδόν μηδενικό σφάλμα ως προς τα training data. Αυτό γίνεται γιατί η σχέση αναπτύσσεται ενσωματώνοντας κάθε δεδομένο με την ίδια βαρύτητα, συμπεριλαμβανοντας και θόρυβο. Συνεπώς το μοντέλο είναι πολύ ευαίσθητο και αντί να βρίσκει τη γενική ιδέα της σχέσης, την υπερειδικοποιεί, με αποτέλεσμα να εκφράζει αποκλειστικά και μόνο το σετ εκπαίδευσης μας. Αυτό ονομάζουμε **overfitting.**

Συνεπώς, στόχος είναι να δημιουργήσουμε ένα μοντέλο το οποίο δεν θα «απομνημονεύει» τα δεδομένα προς εκπαίδευση, αλλά θα βρίσκει την πραγματική

σχέση μεταξύ των εξαρτημένων και ανεξάρτητων μεταβλητών. Πως θα διαλέξουμε το καλύτερο μοντέλο, λοιπόν, όταν ελέγχουμε μόνο την ακρίβεια σαν μετρική αποτελεσματικότητας? Είναι πιθανό, ένα μοντέλο το οποίο είναι overfitted μπορεί να έχει το καλύτερο σκορ επιτυχίας πλασματικά. Για την λύση του προβλήματος αυτού, στην επιστήμη δεδομένων υπάρχει η τεχνική του Validation.

3.5 Επικύρωση

Πριν καταλήξουμε στα τελικά αποτελέσματα επιτυχίας του ταξινομητή, είναι σημαντικό να επικυρώσουμε τα δεδομένα μας. Όπως είπαμε, είναι πιθανό τα διανύσματα εκπαίδευσης μας να συμπεριλαμβάνουν τα δεδένα ελέγχου, με αποτέλεσμα να έχουμε πλασματικά σκορ. Με την επικύρωση (Cross Validation) το σύνολο των δεδομένων εκπαίδευσης μας χωρίζεται σε N υποσύνολα, και μετά κάθε ένα από αυτά χρησιμοποιείται διαδοχικά σαν σύνολο ελέγχου, ενώ τα υπόλοιπα N-1 υποσύνολα ενώνονται και χρησιμοποιούνται σαν σύνολο εκπαίδευσης. Στο τέλος των N εκπαιδεύσεων, χρησιμοποιούνται τα αποτελέσματα για να βγει ένας μέσος όρος ακρίβειας για το μοντέλο [64]. Η τυπική τιμή για το N είναι 10. Με τον τρόπο αυτό, φροντίουμε να μην υπάρχουν τυχαία καλά αποτελέσματα, αλλά τα σκορ επιτυχίας που θα προκύψουν μέσα από το μέσο όρο των 10 προσπαθειών να εκφράζουν όντως την ποιότητα του ταξινομητή μας.



***Εικόνα 16**:Επικύρωση (Validation) με cv=5*

3.6 Μετρικές Αξιολόγησης

Μετά το πέρας της εκπαίδευσης, το μηχανικό σύστημα μπορεί να ξεκινήσει την ταξινόμηση με την δοθείσα είσοδο . Για την αξιολόγηση της απόδοσης ενός ταξινομητή υπάρχουν συγκεκριμένες μετρικές για τον υπολογισμό της επίδοσης του εκάστοτε αλγορίθμου μηχανικής μάθησης.

Τα στοιχεία που θα προσμετρηθούν ως προς τις μετρικές είναι:

TP = Σωστή ταξινόμηση / το πλήθος των στιγμιοτύπων που ανήκουν στην θετική κλάση και ταξινομήθηκαν στην θετική κλάση.

TN = Σωστή ταξινόμηση / το πλήθος των στιγμιοτύπων που ανήκουν στην αρνητική κλάση και ταξινομήθηκαν στην θετική κλάση

FP = Λανθασμένη ταξινόμηση/ το πλήθος των στιγμιοτύπων που ανήκουν στην αρνητική κλάση και ταξινομήθηκαν στην αρνητική κλάση.

FN = Λανθασμένη ταξινόμηση/ το πλήθος των στιγμιοτύπων που ανήκουν στην θετική κλάση και ταξινομήθηκαν στην αρνητική κλάση

Οι μετρικές είναι:

- Η πιο βασική μετρική είναι η ορθότητα (**accuracy**), η οποία υπολογίζεται από τον τύπο:

$$\text{Accuracy} = \frac{\text{Number of correct predictions}}{\text{Total number of predictions}}$$

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

- Η ευαισθησία ή ανάκληση(**recall**) που δίνεται από τον τύπο:

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

- Η ακρίβεια (**precision**) που δίνεται από τον τύπο:

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

- Η μετρική F1-, η οποία συνδυάζει δύο από τις παραπάνω μετρικές:

$$2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}.$$

Παράλληλα είναι σημαντικό να αναφερθούμε στο Confusion Matrix, το οποίο αποτελείται από ένα πίνακα που δείχνει τις πραγματικές τιμές σε σχέση με τις τιμές που προβλέψαμε.

Πραγματικές τιμές	Τιμές που Προβλέψαμε: Θετική τιμή (1)	Τιμές που Προβλέψαμε: Αρνητική τιμή (0)
Θετική τιμή (1)	TP	FN
Αρνητική τιμή (0)	FP	TN

ΚΕΦΑΛΑΙΟ 4. ΥΛΟΠΟΙΗΣΗ

4.1 Συναρτήσεις

Για κάθε διαφορετικό σετ δεδομένων, ακολουθήσαμε την ίδια λογική με διαφορετικές μεθόδους. Δημιουργήσαμε το αρχείο ImportsDefinitions.py, το οποίο περιέχει όλες τις βιβλιοθήκες και συναρτήσεις που θα χρησιμοποιήσουμε, το οποίο και φορτώνουμε στην αρχή. Επιπλέον, ορίσαμε τις εξής συναρτήσεις:

- SentimentAnalysis_Sentiwordnet() και SentimentAnalysis_Vader(), οι οποίες υλοποιούν συναισθηματική ανάλυση με διαφορετικό τρόπο στα στιχουργικά δεδομένα μας, έτσι ώστε να έχουμε κλάσεις συναισθημάτων στα δεδομένα που έχουμε. Τα αποτελέσματα των δύο συναρτήσεων αυτών θα δημιουργήσουν 2 σετ δεδομένων με βάση τα οποία θα εφαρμόσουμε αλγορίθμους μηχανικής μάθησης
- Tokenizer_preprocessor() και tokenizer_preprocessor_imdb() για να χωρίσουμε σε λέξεις τα σετ κειμενικών δεδομένων, και να κάνουμε προεπεξεργασία των δεδομένων αυτών, καθαρίζοντας τα από λέξεις που δεν περιλαμβάνουν συναισθηματικό χαρακτήρα.
- countvect_test_simple() και tfidf_test_simple(). Οι συναρτήσεις αυτές εξάγουν τα χαρακτηριστικά των σετ δεδομένων μας που θέλουμε να κρατήσουμε. Συμπεριλαμβάνουν σαν παράμετρο το token_izer, το οποίο αν είναι 1, υλοποιεί την συνάρτηση tokenizer_preprocessor() σαν όρισμα του tokenizer του vectorizer, ενώ αν είναι 2 υλοποιεί την συνάρτηση tokenizer_preprocessor_imdb() σαν όρισμα του tokenizer στο vectoizer. Αν είναι κάτι άλλο, υλοποιούν το vectorizing χωρίς κάποιον προεπιλεγμένο tokenizer.
- countvect_test_maxdf() και tfidf_test_maxdf (). Οι συναρτήσεις αυτές τρέχουν τους 2 διαφορετικούς vectorizers, Countvectorizer() και TfidfVectorizer() για το το max_df που έχουμε δώσει.
- countvect_test_ngrams() και tfidf_test_ngrams(). Οι συναρτήσεις αυτές τρέχουν τους 2 διαφορετικούς vectorizers, countvectorizer() και TfidfVectorizer() για τα n-grams που έχουμε επιλέξει σαν όρισμα.
- countvect_test_maxfeat() και tfidf_test_maxfeat(). Οι συναρτήσεις αυτές τρέχουν τους 2 διαφορετικούς vectorizers, countvectorizer() και TfidfVectorizer() για το όρισμα max_features που έχουμε δώσει στο όρισμα.
- classifier_finder() και classifier_finder_music(). Οι συναρτήσεις αυτές τρέχουν τους διαφορετικούς ταξινομητές, ώστε να βρούμε με cross_validation τον καλύτερο.
- saveList() και loadList(). Οι συναρτήσεις αυτές χρησιμοποιούνται στην συναισθηματική ανάλυση κειμένου, ώστε να αποθηκεύσουμε στο δίσκο μας ή να φορτώσουμε στο πρόγραμμα μας επιθυμητές λίστες.

4.2 Μεθοδολογία

Χωρίζουμε την διπλωματική εργασία σε δύο μέρη. Στην ανάλυση μουσικών δεδομένων και στην ανάλυση κειμενικών δεδομένων.

Αρχικά, αντλούμε τα σετ δεδομένων.

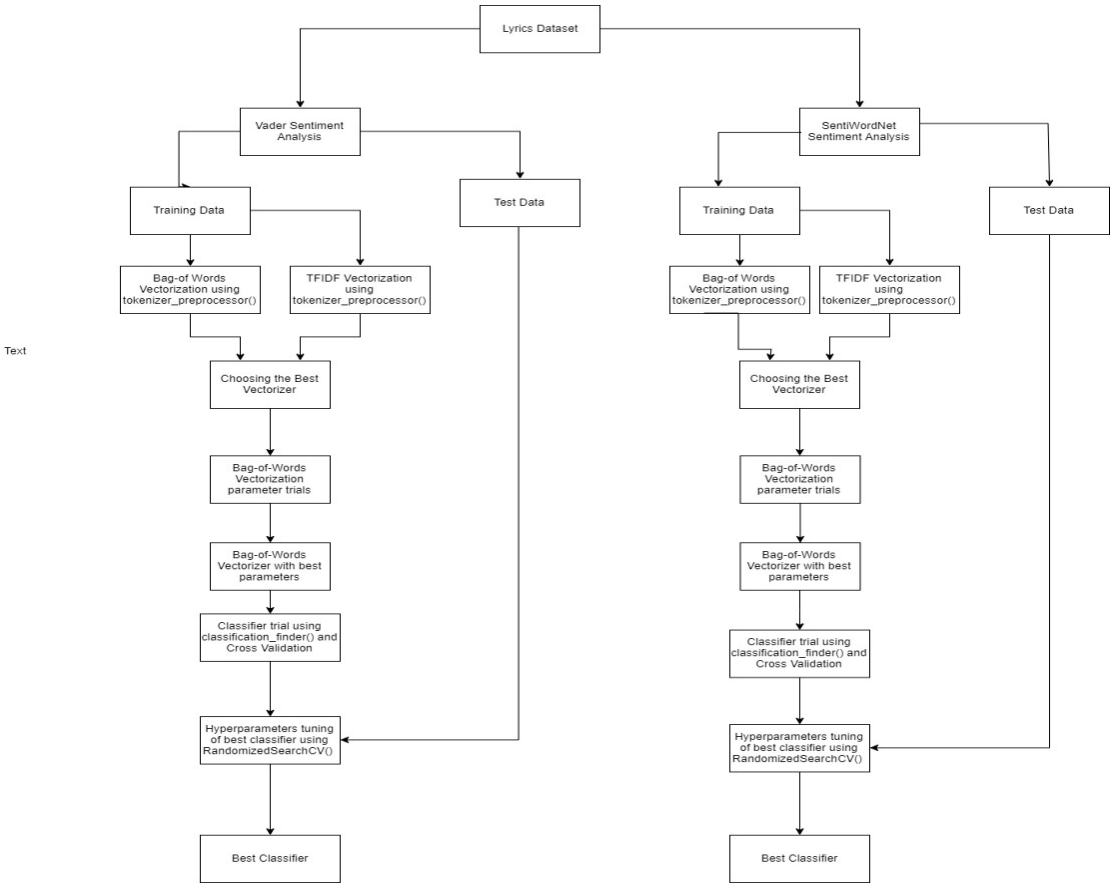
Για τα κειμενικά δεδομένα, αντλούμε ένα σετ στίχων το οποίο δεν περιέχει συναισθηματική κλάση και ένα σετ με κριτικές ταινιών που περιέχει συναισθηματική κλάση. Για τα στιχουργικά δεδομένα που δεν περιλαμβάνουν συναισθηματική κλάση, για να μπορέσουμε να εκπαιδεύσουμε τους ταξινομητές μας κατάλληλα, θα εφαρμόσουμε συναισθηματική ανάλυση με δύο διαφορετικούς τρόπους: με Vader και με SentiWordNet. Καθώς η αναπαράσταση κειμενικής πληροφορίας πρέπει να γίνει με διανυσματοποίηση, εφαρμόζουμε διαφορετικά διανύσματα με διαφορετικές παραμέτρους, για τις τρεις διαφορετικές περιπτώσεις. Στην συνέχεια, μόλις αποθηκεύσουμε στα διανύσματα το λεξιλόγιο με τις επιθυμητές παραμέτρους, είναι σειρά να δοκιμάσουμε διαφορετικούς ταξινομητές. Αφού δούμε πως ανταποκρίνονται οι ταξινομητές αυτοί, θα διαλέξουμε αυτόν με το καλύτερο σκορ για το κάθε σετ δεδομένων μας και θα προσπαθήσουμε να βρούμε τις καλύτερες παραμέτρους. Τέλος, θα καταλήξουμε με 3 διαφορετικούς ταξινομητές, φτιαγμένους από 3 σετ δεδομένων. Θα προσπαθήσουμε να προβλέψουμε τα αποτελέσματα με βάση κάποιο διάνυσμα ελέγχου, για κάθε σετ δεδομένων. Ο ταξινομητής που θα ανταποκριθεί συνολικά καλύτερα, θα είναι και ο καλύτερος.

Αντίστοιχα, για τα μουσικά δεδομένα. Αφού αντλήσουμε τα δεδομένα μας, θα δημιουργήσουμε 3 περιπτώσεις. Κάθε περίπτωση θα μελετηθεί ξεχωριστά. Καθώς το σετ μουσικών δεδομένων απεικονίζεται με τρόπο με τον οποίο μπορεί να γίνει επεξεργασία και εκπαίδευση μοντέλων, θα κάνουμε μόνο κανονικοποίηση των διανυσμάτων μας. Πάλι, θα δοκιμάσουμε διαφορετικούς ταξινομητές, και θα παραμετροποιήσουμε τον καλύτερο. Τέλος, θα καταλήξουμε με 3 ταξινομητές, που θα τους εφαρμόσουμε και στα 3 υποσετ δεδομένων, ώστε να βρούμε τον καλύτερο ταξινομητή.

4.3 Δομή Υλοποίησης

Στην συνέχεια, παραθέτουμε τη δομή της διπλωματικής που θα ακολουθήσουμε, για κάθε διαφορετικό σετ δεδομένων.

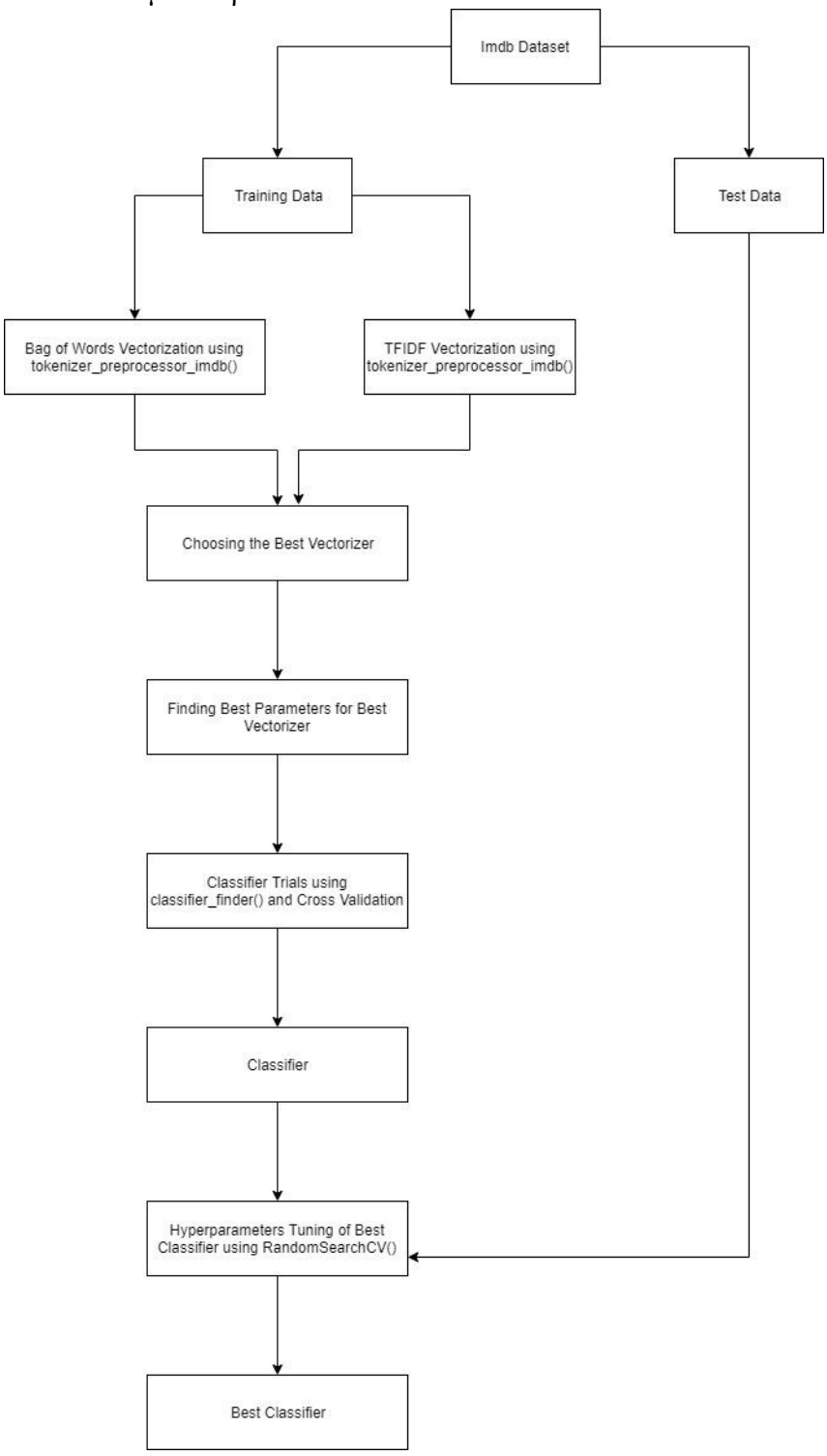
Αρχικά για τα στιχουργικά δεδομένα:



***Εικόνα 17**: Δομή Υλοποίησης Στιχουργικών Δεδομένων*

Παρατηρούμε ότι το στιχουργικό σετ δεδομένων το χωρίζουμε σε δυο διαφορετικά υποσετ, που θα προκύψουν όταν κάνουμε συναισθηματική ανάλυση με δύο διαφορετικούς τρόπους. Στην συνέχεια, θα ακολουθήσουμε ομόλογη διαδικασία για να αναπαραστήσουμε τα δεδομένα αυτά σε διάνυσμα, το οποίο θα παραμετροποιήσουμε χρησιμοποιώντας επικύρωση (validation). Μετά, θα δοκιμάσουμε διαφορετικούς αλγορίθμους ταξινόμησης, και μόλις επιλέξουμε τον καλύτερο, θα τον παραμετροποιήσουμε, ώστε να καταλήξουμε στον καλύτερο δυνατό ταξινομητή για το σετ δεδομένων μας.

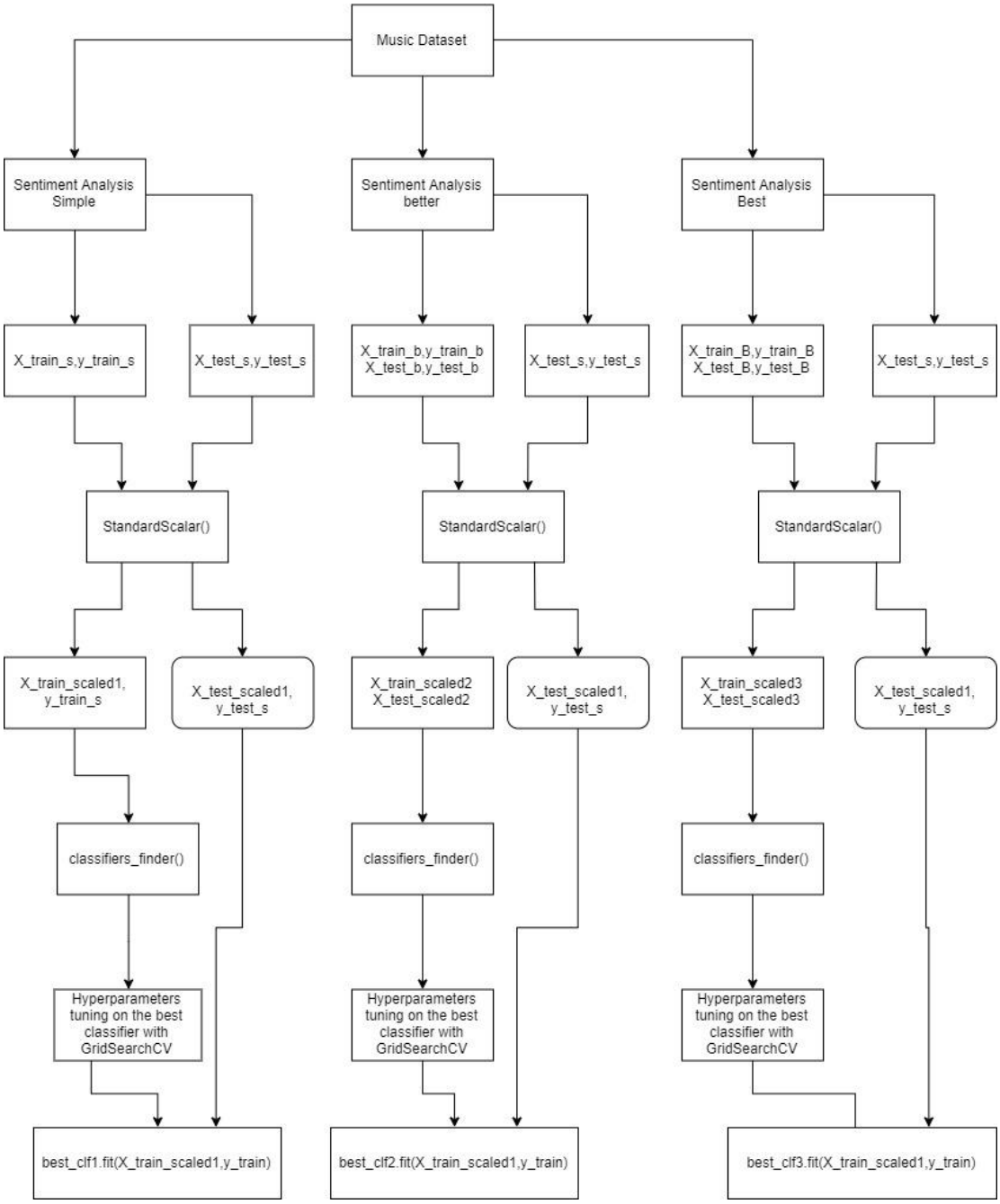
Για τα δεδομένα κριτικών από imdb:



***Εικόνα 18**: Δομή Υλοποίησης Κριτικών IMDB*

Αντίστοιχα, η δομή για το σετ Imdb υλοποιείται με παρόμοιο τρόπο όπως τα κειμενικά δεδομένα. Σημειώνουμε την διαφορά, ως προς την συνάρτηση προεπεξεργασίας που θα χρησιμοποιήσουμε, καθώς έχουμε άλλη φύση δεδομένων η οποία θα χρειαστεί συγκεκριμένη προεπεξεργασία.

Τέλος για τα μουσικά δεδομένα:



***Εικόνα 19**: Δομή Υλοποίησης Μουσικού Σετ Δεδομένων*

Στο μουσικό σετ δεδομένων παρατηρούμε ότι κανονικοποιούμε τα δεδομένα μας για να έχουμε καλύτερα αποτελέσματα. Παράλληλα, καθώς τα δεδομένα μας αναπαρίστανται με αριθμητική μορφή δεν χρειάζεται να χρησιμοποιήσουμε εξαγωγή στοιχείων.

4.4 Εξέταση Δεδομένων

4.4.1 Κειμενικά Δεδομένα

i) Σετ δεδομένων Στίχων

Χρησιμοποιήσαμε διαφορετικά σετ δεδομένων, για να φτιάξουμε διαφορετικά μοντέλα και να τα αξιολογήσουμε. Ένα σετ δεδομένων με *στίχους χωρίς συναισθηματική κλάση*, στο οποίο θα εφαρμόσουμε εμείς συναισθηματική ανάλυση και ένα σετ *κειμενικών δεδομένων με κριτικές από το imdb, το οποίο περιέχει συναισθηματικές κλάσεις*.

Το σετ στιχουργικών δεδομένων πάρθηκε από το Lyricsfreak, τα οποία δεν περιέχουν συναισθηματική κλάση και περιλαμβάνει 57650 διαφορετικά τραγούδια. Αποθηκεύουμε τα δεδομένα μας σε ένα dataframe στο οποίο θα γίνει μετέπειτα συναισθηματική ανάλυση, και με βάση τα στοιχεία αυτά θα ξεκινήσει η μηχανική μάθηση. Ελέγχουμε αν το δείγμα που κρατήσαμε περιέχει ελλιπή δεδομένα, τα οποία διαγράφουμε από το dataframe μας. Ελέγχουμε ένα μέρος του δείγματος αυτού, με τη μέθοδο head. Από όσο βλέπουμε στο σετ δεδομένων αυτό δεν υπάρχει μουσικοδομικές ενδείξεις, όπως ρεφραίν κουπλέ κτλ. Παρόλα αυτά το δείγμα μας θα χρειαστεί preprocessing για να αφαιρεθούν μικρές λέξεις, αλλαγές σειράς κτλ.

```
Τα NULL δεδομένα στο dataset μας είναι: 0
Έχουμε 57650 διαφορετικά τραγούδια
0    Look at her face, it's a wonderful face  \nAnd...
1    Take it easy with me, please  \nTouch me gentl...
2    I'll never know why I had to go  \nWhy I had t...
3    Making somebody happy is a question of give an...
4    Making somebody happy is a question of give an...
5    Well, you hoot and you holler and you make me ...
6    Down in the street they're all singing and sho...
7    Chiquitita, tell me what's wrong  \nYou're enc...
8    I was out with the morning sun  \nCouldn't sle...
9    I'm waitin' for you baby  \nI'm sitting all al...
Name: text, dtype: object
```

Εικόνα 20: *Εξέταση Δεδομένων Στίχων*

Στο σετ δεδομένων αυτό, εφαρμόσαμε συναισθηματική ανάλυση χρησιμοποιώντας το εργαλείο ανάλυσης Vader, καθώς και συναισθηματική ανάλυση με λεξικό.

ii) Σετ Δεδομένων Imdb

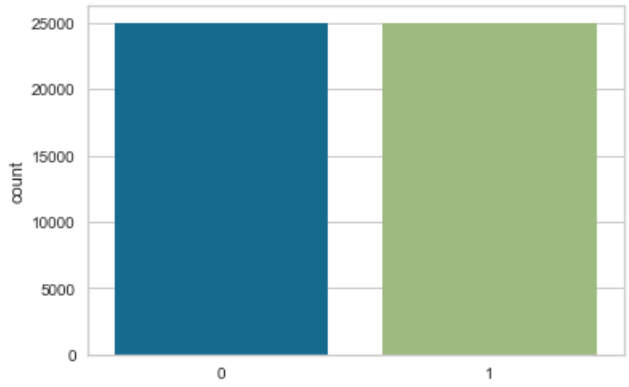
Επιπλέον, χρησιμοποιήσαμε ένα σετ δεδομένων με κριτικές για το imdb, ώστε να αντλήσουμε γλωσσικά στοιχεία από διαφορετικό κλάδο, ώστε να αναλύσουμε και να αξιολογήσουμε τα διαφορετικά μοντέλα .

Το σετ που αντλήσαμε περιλαμβάνει συνολικά 50000 κριτικές ταινιών με δυαδικές κλάσεις συναισθήματος 0 και 1, δηλαδή με αρνητικό και με θετικό συναίσθημα αντίστοιχα και κατασκευάστηκε ώστε να μην επιτρέπει παραπάνω από 30 κριτικές ανά ταινία. Οι συγγραφείς αποφάσισαν να συμπεριλάβουν μόνο κριτικές οι οποίες κατηγοριοποιούνται σαν θετικές ή αρνητικές. Έτσι, κριτικές που παρέχουν βαθμολογία

7 και πάνω κρίνονται θετικές, και κριτικές με βαθμολογία από 4 και κάτω κρίνονται αρνητικές. Οι κριτικές που παρείχαν ενδιάμεσες βαθμολογίες αγνοήθηκαν. [65]

```
                                                review sentiment
0  One of the other reviewers has mentioned that ...  positive
1  A wonderful little production. <br /><br />The...  positive
2  I thought this was a wonderful way to spend ti...  positive
3  Basically there's a family where a little boy ...  negative
4  Petter Mattei's "Love in the Time of Money" is...  positive
Τα NULL δεδομένα στο dataset μας είναι: review       0
sentiment    0
dtype: int64
Έχουμε 50000 κριτικές στο dataset μας
```

***Εικόνα 21**: Εξέταση Δεδομένων Κριτικών IMDB*




```
Έχουμε 25000 χαρούμενα τραγούδια
Έχουμε 25000 στενάχωρα τραγούδια
```

***Εικόνα 22**: Ιστόγραμμα για τα διαφορετικού συναισθήματος τραγούδια*

4.4.2 Μουσικά Δεδομένα

Το Σετ δεδομένων που θα χρησιμοποιήσουμε αποτελείται από 200 τραγούδια περιλαμβάνει μουσικά στοιχεία που παρθήκαν από αρχεία wav στα οποία ανατέθηκε συναισθηματικό βάρος και αποτέλεσε προιόν επιστημονικής έρευνας. Σκοπός ήταν η διαχείριση 7 χαρακτηριστιών του ήχου ώστε να αποφανθεί πόσο αυτά συμμετέχουν στην αντίληψη συναισθημάτων. Για κάθε χαρακτηριστικό, προσπάθησαν να έχουν παραπάνω από 2 τιμές (Για παράδειγμα, παίξαν την ίδια μουσική σε 6 διαφορετικά κλειδιά), έτσι ώστε καταλάβουμε αν οι παράγοντες αυτοί συνεισφέρουν στα συναισθήματα που βιώνουμε με γραμμικό τρόπο. Τα δεδομένα που περιλαμβάνει έχουν κανονικοποιηθεί σε τιμές που ξεκινάνε από το 1 και αναπαριστούν συγκεκριμένες κλάσεις. [66]

Συγκεκριμένα, τα στοιχεία που περιλαμβάνει είναι:

Register, 6 τιμές, από 1-6 που αντιπροσωπεύουν 6 διαφορετικά επίπεδα τόνου (53, 59, 65, 71, 77, και 83 σε MIDI pitch).

Mode, 2 τιμές, 1-2, που αντιπροσωπεύουν το μουσικό δρόμο Μινόρε ή Ματζόρε

Tempo, 5 τιμές, από 1-5 που αντιπροσωπεύουν τον μέσο όρο νοτών ανά δευτερόλεπτο (1.2, 2, 2.8, 4.4, και 6 NPS)

Sound level, 5 τιμές που αντιπροσωπεύουν την ένταση του κομματιού στα διαφορετικά επίπεδα (-10, -5, 0, +5, +10 dB)

Articulation, 4 τιμές (1, 0.75, 0.5, 0.25) που αντιπροσωπεύουν τα διαφορετικά articulations, από λεγκάτο σε στακάτο

Timbre, 3 τιμές, μία για κάθε όργανο (1= trumpet, 2 = flute, 3 = horn)

Melody, 4 κατηγορίες συναισθημάτων (1 = Sad, 2 = Happy, 3 = Scary, 4 = Peaceful].

	Nro	Register	Mode	Tempo	Soundlevel	Articulation	Timbre	Melody
0	1	4	1	4	4	2	2	4
1	2	5	1	4	1	1	2	2
2	3	2	2	5	1	1	2	1
3	4	1	1	5	4	4	1	2
4	5	3	2	1	3	2	2	1

Εικόνα 23: *Μουσικά Δεδομένα Εισόδου*

4.5 Καθαρισμός Σετ Δεδομένων

4.5.1 Μουσικά Δεδομένα

Όπως βλέπουμε, στα δεδομένα εισόδου μας έχουμε μία στήλη αρίθμησης, την 'Nro', η οποία πρέπει να αφαιρεθεί για να μην επηρεάσει τον ταξινομητή μας.

Ταυτόχρονα, καθώς η στήλη Mode περιλαμβάνει δυαδική πληροφορία, αν δηλαδή έχουμε κλίμακα Ματζόρε ή Μινόρε, θα μετατρέψουμε αντίστοιχα τα δεδομένα αυτά σε δυο άλλες στήλες, για να προετοιμάσουμε καλύτερα την δημιουργία του ταξινομητή μας.

Μετά από προεπεξεργασία στα δεδομένα μας, έχουμε:

	Register	Tempo	Soundlevel	Articulation	Timbre	Melody	Major	Minor
0	4	4	4	2	2	4	0	1
1	5	4	1	1	2	2	0	1
2	2	5	1	1	2	1	1	0
3	1	5	4	4	1	2	0	1
4	3	1	3	2	2	1	1	0

Εικόνα 24*: Τροποποίηση Μουσικών Δεδομένων Εισόδου*

4.5.2 Κειμενικά Δεδομένα

Για να υπολογιστεί η σύνδεση των συναισθηματικών λέξεων με την τελική τιμή, πρέπει πρώτα να προετοιμάσουμε τα δεδομένα μας για επεξεργασία. Δεν είναι όλα τα δεδομένα εισόδου μας χρήσιμα, οπότε πρέπει να τα τροποποιήσουμε κατάλληλα για να

μην εισάγουμε θόρυβο, που είτε καθυστερεί είτε αλλοιώνει την αποδοτικότητα του μοντέλου που θέλουμε να φτιάξουμε.

Ο θόρυβος θα μπορούσε να είναι πολύ μικρές λέξεις, λέξεις όπως *"chorus"* ή *"solo"*, για τα στιχουργικά κείμενα οι οποίες απλά μας δείχνουν τη δομή του τραγουδιού. Επειδή κάνοντας examine το σετ δεδομένων μας δεν βρήκαμε τέτοιου είδους δεδομένα, προσπερνάμε το βήμα αυτό.

Θα φτιάξουμε δύο συναρτήσεις που θα επιτελεί το ρόλο αυτό, την **tokenizer_preproccessor()**, για τα στιχουργικά δεδομένα και την **tokenizer_preprocessor_imdb()** για το σετ δεδομένων από το imdb. Οι δύο συναρτήσεις θα παίρνουν σαν όρισμα κείμενο και θα το καθαρίζουν από περιττές πληροφορίες.

Ξεκινάμε κάνοντας **tokenize** τις λέξεις κάθε τραγουδιού, δηλαδή κάνοντας κάθε λέξη και ένα token, ώστε να μπορούμε να επεξεργαστούμε κάθε λέξη ξεχωριστά. Στη συνέχεια, βρίσκουμε το συντακτικό μέρος κάθε λέξης χάρη στο **POS_TAGGER** και τέλος κάνουμε **lemmatize** τη λέξη αυτή. Τέλος, απομακρύνουμε λέξεις που έχουν πολύ μικρό μήκος, καθώς και λέξεις που περιλαμβάνονται στο λεγόμενο σετ **stopwords**, λέξεις δηλαδή που έχουν κυρίως συντακτικό/γραμματικό περιεχόμενο, όπως άρθρα, αντωνυμίες κτλ. Με τον τρόπο αυτό κρατάμε μόνο τη χρήσιμη πληροφορία προς συναισθηματική ανάλυση.

Τέλος, επιστρέφουμε τα λήμματα που βρήκαμε για τις λέξεις, τα οποία θα είναι και τα features που θα χρησιμοποιήσουμε, για να βρούμε την σχέση μεταξύ τους και του συναισθηματικού προσήμου.

Αντίστοιχα, κάνουμε το ίδιο βήμα για τα κειμενικά δεδομένα κριτικών του imdb. Επειδή έχουμε διαφορά στον τύπο δεδομένων, θα εισάγουμε στα stopwords τις λέξεις *'movie', 'film', 'character' ώστε να αγνοηθούν από την συναισθηματική ανάλυση.*

4.6 Συναισθηματική Ανάλυση

4.6.1 Κειμενικά Δεδομένα

Θα υλοποιήσουμε δύο διαφορετικές συναισθηματικές αναλύσεις στα κειμενικά δεδομένα μας, μία χρησιμοποιώντας το εργαλείο **Vader** της NLTK βιβλιοθήκης, και συναισθηματική ανάλυση χρησιμοποιώντας λεξικό, και συγκεκριμένα το **SentiWordNet**. Με τον τρόπο αυτό, θα δημιουργήσουμε διαφορετικά μοντέλα, τα οποία θα αξιολογήσουμε μετέπειτα.

4.6.1.1 Υλοποίηση συναισθηματικής ανάλυσης με Λεξικό

Για την συναισθηματική ανάλυση με Λεξικό, θα αξιοποιήσουμε τα εργαλεία επεξεργασίας φυσικής γλώσσας της NLTK tokenize(), lemmatize(), pos_tag(), καθώς επίσης θα διαγράψουμε πιθανά stopwords, λέξεις με μικρό μήκος καθώς και σημεία στίξης. Θα εκτελέσουμε κάθε βήμα σταδιακά, καθώς κάθε περίπτωση ανάλογα με τις παραμέτρους που θα βάλουμε, θα δώσει διαφορετικά αποτελέσματα για κάθε σετ δεδομένων. Αν και δεν θα μπορέσουμε ποτέ να φτιάξουμε μια τέλεια συναισθηματική ανάλυση, που να εξάγει συναισθηματική πληροφορικά σε τέτοιο βάθος όπως ο άνθρωπος, μπορούμε να υλοποιήσουμε μία πολύ καλή προσέγγιση, καθώς σκοπός είναι εν τέλει να κατηγοριοποιούμε τα στιχουργικά κείμενα σε χαρούμενα και στενάχωρα.

Παρακάτω περιγράφουμε σταδιακά την διαδικασία που ακολουθήσαμε, χρησιμοποιώντας την συνάρτηση **SentimentAnalysis_Sentiwordnet()**.

Αρχικά, για κάθε κείμενο, χωρίζουμε τις λέξεις με το **tokenize()**, μετατρέποντας παράλληλα όλα τα γράμματα σε πεζά.

```
#PREPROCESSING KAI YLOPOIISI DIKHS MAS SYNAISTHIMATIKHS ANALYSIS
#kanoume tokenize tis lekseis

wnl = WordNetLemmatizer()
my_sentiments=[] #h lista pou tha periexei tosynaisthitiko score kathe keimenou
my_sentiments_class=[]
#print(stopwords)
stop = set(stopwords.words('english'))
#print(len(text_samples))
for text in text_samples:
    #kanoume preprocessing ta dedomena mas
    word_tokens=word_tokenize(text.lower()) #kanoume tokenize
    print(word_tokens)
```

```
['oh', ',', 'come', 'little', 'children', 'oh', ',', 'come', 'one', 'and', 'all', 'christmas', 'with', 'its', 'trees', 'and',
'windows', 'all', 'of', 'love', 'christmas', 'with', 'its', 'snow', 'and', 'ice', 'and', 'mistletoe', 'christmas', 'ca',
"n't", 'you', 'hear', 'the', 'church', 'bells', 'ringing', '?', 'from', 'within', 'ca', "n't", 'you', 'hear', 'the', 'choir',
'boy', 'singing', '?', 'oh', ',', 'silent', 'night', ',', 'floats', 'up', 'upon', 'the', 'air', 'holy', 'night', ',', 'bring
s', 'peace', 'and', 'joy', 'everywhere', 'from', 'your', 'heart', 'let', 'this', 'joyous', 'message', 'come', 'peace', 'on',
'earth', ',', 'merry', 'christmas', 'all', 'christmas', 'with', 'its', 'trees', 'and', 'windows', 'all', 'of', 'love', 'chris
tmas', 'with', 'its', 'snow', 'and', 'ice', 'and', 'mistletoe', 'from', 'your', 'heart', 'let', 'this', 'joyous', 'message',
'come', 'peace', 'on', 'earth', ',', 'merry', 'christmas', 'all', 'merry', 'christmas']
['dee-goo-pee-oo-poo', 'ta-dan', '!', 'bad', 'conscience', ':', 'does', 'this', 'kind', 'of', 'life', 'look', 'interesting',
'to', 'you', '?', 'night', 'after', 'night', ',', 'dinners', 'with', 'herb', 'cohen', '.', 'thrill-packed', ',', 'fun-fille
d', 'evenings', 'on', 'the', 'french', 'riviera', 'at', 'the', 'midem', 'convention', '.', 'a', 'big', 'tie', ',', 'the', 'wh
ole', 'bit', '.', 'watch', 'mutt', 'eat', ',', 'and', 'leon', 'feed', 'the', 'geese', '.', 'one', 'thousand', 'green', 'busin
ess', 'cards', ',', 'with', 'your', 'name', 'and', 'the', 'wrong', 'address', '.', 'plus', 'six', 'royalty', 'statements',
',', 'inspected', 'and', 'customized', 'by', 'ran', 'toon', 'tan', 'han', 'toon', 'frammet', 'and', 'dee', '.', 'followed',
'by', 'twelve', 'potential', 'suicides', 'as', 'the', 'members', 'of', 'your', 'group', ',', 'past', 'and', 'present', ',',
'find', 'out', 'they', 'ca', "n't", 'collect', 'unemployment', '.', 'a', 'dog', ',', 'a', 'car', ',', 'an', 'epidemic', 'of',
'body', 'lice', 'with', 'your', 'own', 'record', 'company', ',', 'your', 'name', 'on', 'the', 'door', ',', 'electric', 'buzze
r', 'to', 'the', 'inner', 'office', ',', 'ona', "'s", 'tits', ',', 'and', 'a', 'three', 'month', 'supply', 'of', 'german', 'b
```

***Εικόνα 25**: Αποτέλεσμα tokenization*

Παρατηρούμε ότι το tokenize έκανε καλή δουλειά για να χωρίσει τα κειμενικά μας δεδομένα σε tokens, παρόλα τα μικρά προβλήματα, όπως πχ το *"can't"* χωρίστηκε σε *"ca"* και *"n't"*.Ανάλογα με το διαφορετικό tokenizer() που θα χρησιμοποιήσουμε ο διαχωρισμός αυτός είναι διαφορετικός. Για παράδειγμα, ο Whitespace_tokenizer() το κρατά σαν μία λέξη (*"didn't"*), ενώ ο word_tokenizer() το χωρίζει σε δύο λέξεις (*'did', "n't")*. Καθώς τα εργαλεία συναισθηματικής ανάλυσης που θα χρησιμοποιήσουμε αναγνωρίζουν το *"n't"* , ακόμα και μετά την αφαίρεση σημείων στίξεων ('nt') σαν λέξη με αρνητικό συναίσθημα (με το sentisynset δίνεται αρνητικό βάρος 0.25) θα χρησιμοποιήσουμε αυτόν.

Από τα δεδομένα που εκτυπώνουμε, βλέπουμε ότι έχουμε πολλά *stopwords* που δεν παρέχουν συναισθηματική πληροφορία.

Εδώ βλέπουμε τα καταγεγραμμένα stopwords:

```
In [16]: stop = set(stopwords.words('english'))
         print(stop)

{'to', 'you', 'down', 'its', 'any', 'against', 'between', 'more', 'ma', 'we', 'up', 'again', "she's", 'for', 'him', 'own', 'l
l', "you'll", "you've", 'it', 'didn', 'wasn', 'while', 'their', 'hasn', 'yourself', 'm', "didn't", 'after', 'being', 'itself',
'nor', 'once', 'off', 'theirs', 'about', 'from', 'who', 'won', 'has', 'can', 'out', 'her', 'ours', 'over', 't', 'is', 'all', 'a
bove', 'hers', 'of', 'd', "mightn't", "aren't", 'does', 'just', 'and', 'should', "that'll", 'under', 'what', 'ourselves', 'is
n', 'did', 'there', 'such', 'which', 'haven', 'in', 'she', 's', 've', 'few', "mustn't", 'mustn', 'weren', "wasn't", 'be', "wo
n't", 'each', 'where', "you'd", 'most', 'that', 're', "weren't", 'so', 'during', 'was', 'very', 'no', 'yourselves', "needn't",
'o', 'wouldn', 'or', 'our', 'ain', 'now', 'how', 'hadn', 'my', 'shan', 'shouldn', "shouldn't", 'other', "should've", 'had', 'co
uldn', 'having', 'these', "shan't", 'through', "wouldn't", 'he', 'into', 'same', 'yours', "don't", 'then', 'some', "hasn't", 't
hem', 'those', 'why', 'herself', 'further', 'because', 'at', 'doesn', 'here', 'the', 'as', 'do', 'until', 'your', 'on', 'but',
'his', 'aren', "couldn't", 'they', "doesn't", "isn't", "haven't", 'whom', 'am', 'too', 'y', 'don', 'have', 'not', 'below', 'onl
y', 'when', 'a', 'were', 'been', 'before', 'are', 'with', "hadn't", "it's", 'than', 'will', 'doing', 'an', 'i', 'mightn', 'bot
h', 'me', 'if', 'by', 'needn', "you're", 'themselves', 'himself', 'this', 'myself'}
```

***Εικόνα 26**: Stopwords*

Βλέπουμε λοιπόν, ότι πέρα από προσωπικές αντωνυμίες, άρθρα, συνήθη επιρρήματα, έχουμε λέξεις ενωμένες για να πιάσουμε όλες τις διαφορετικές περιπτώσεις, όπως *"that'll"*. Θα αφαιρέσουμε τα stopwords αυτά, και θα δούμε τι αποτελέσματα έχουμε:

```
In [26]: #PREPROCESSING KAI YLOPOIISI DIKHS MAS SYNAISTHIMATIKHS ANALYSIS
         #kanoume tokenize tis lekseis

         wnl = WordNetLemmatizer()
         my_sentiments=[] #h lista pou tha periexei tosynaisthitiko score kathe keimenou
         my_sentiments_class=[]
         #print(stopwords)
         stop = set(stopwords.words('english'))
         #print(len(text_samples))
         for text in text_samples:
             #kanoume preprocessing ta dedomena mas
             word_tokens=word_tokenize(text.lower()) #kanoume tokenize
             print("Εχουμε " + str(len(word_tokens))+ " tokens")
             filtered_word_tokens = [word for word in word_tokens if word not in stop] #svinoume ta stopwords
             print("Αφού αφαιρέσαμε τα stopwords, έχουμε τελικά " + str(len(filtered_word_tokens)) + " tokens")
             print(filtered_word_tokens)

Εχουμε 225 tokens
Αφού αφαιρέσαμε τα stopwords, έχουμε τελικά 123 tokens
['tell', "'re", 'leaving', "'ve", 'got', 'alone', 'sometimes', 'seem', 'like', "'re", 'far', 'away', 'different', 'danger',
'zone', 'could', "n't", 'stop', 'wanted', 'found', 'somebody', 'else', 'sure', "n't", 'know', "'ve", 'found', 'love', 'like',
'revolving', 'door', 'yeah', 'temptation', "'s", 'gon', 'na', 'come', 'lives', 'gon', 'na', 'break', 'happy', 'home', 'tempta
tion', 'going', 'know', 'better', 'leave', 'thing', 'alone', 'always', 'seem', 'distracted', 'like', 'mind', 'somewhere', 'el
se', "n't", 'want', 'alone', 'strange', 'guy', "'re", 'gon', 'na', 'wind', 'shelf', 'yeah', 'temptation', "'s", 'gon', 'na',
'come', 'lives', 'gon', 'na', 'break', 'happy', 'home', 'temptation', 'going', 'know', 'better', 'yeah', 'yeah', 'leave', 'th
ing', 'alone', 'temptation', "'s", 'gon', 'na', 'go', 'lives', 'gon', 'na', 'break', 'happy', 'home', 'temptation', 'going',
'know', 'better', 'yeah', 'yeah', 'temptation', "'s", 'gon', 'na', 'go', 'lives', 'gon', 'na', 'break', 'happy', 'home', 'tem
ptation', 'going', 'know', 'better', 'yeah', 'yeah', 'oh']
Εχουμε 94 tokens
Αφού αφαιρέσαμε τα stopwords, έχουμε τελικά 49 tokens
```

***Εικόνα 27:** Αφαίρεση stopwords και καταμέτρηση tokens*

Βλέπουμε πως κάποιες λέξεις έχουν παραμείνει, όπως τα *"'ve"*, επίσης το *"couldn't"* έγινε *"could"* και *"n't"*.

Επιλέγουμε να κρατήσουμε τα σημεία στίξης *μέσα* στις λέξεις, και να αφαιρέσουμε τα σημεία στίξης που έχουν αποθηκευτεί σαν tokens, όπως κόμματα, αποσιωπητικά, ερωτηματικά και θαυμαστικά. Αν και όλα τα σημεία στίξης φέρουν συναισθηματική πληροφορία, κυρίως το ερωτηματικό και το θαυμαστικό, επειδή έχουμε στιχουργικό περιεχόμενο, θα έχουμε κυρίως ερωτηματικά. Ταυτόχρονα διαγράφουμε τις λέξεις που έχουν μήκος 1 χαρακτήρα.

```
In [38]: #PREPROCESSING KAI YLOPOIISI DIKHS MAS SYNAISTHIMATIKHS ANALYSIS
         #kanoume tokenize tis lekseis

         wnl = WordNetLemmatizer()
         my_sentiments=[] #h lista pou tha periexei tosynaisthitiko score kathe keimenou
         my_sentiments_class=[]
         #print(stopwords)
         stop = set(stopwords.words('english'))
         import re
         for text in text_samples:
             #kanoume preprocessing ta dedomena mas
             word_tokens=word_tokenize(text.lower()) #kanoume tokenize
             print("Εχουμε " + str(len(word_tokens))+ " tokens")
             filtered_word_tokens = [word for word in word_tokens if word not in stop] #svinoume ta stopwords
             print("Αφού αφαιρέσαμε τα stopwords, έχουμε τελικά " + str(len(filtered_word_tokens)) + " tokens")
             #print(filtered_word_tokens)
             filtered_word_tokens = [re.sub(r'[^A-Za-z0-9]+', '', x) for x in filtered_word_tokens] #svinoume ta punctuations
             filtered_word_tokens = [word for word in filtered_word_tokens if len(word)>1] #svinoume tis mikres lekseis
             print("Αφού αφαιρέσαμε τα σημεία στίξης και τις μικρές λέξεις, έχουμε τελικά " + str(len(filtered_word_tokens)) + " tokens")
             print(filtered_word_tokens)

         Εχουμε 225 tokens
         Αφού αφαιρέσαμε τα stopwords, έχουμε τελικά 123 tokens
         Αφού αφαιρέσαμε τα σημεία στίξης και τις μικρές λέξεις, έχουμε τελικά 119 tokens
         ['tell', 're', 'leaving', 've', 'got', 'alone', 'sometimes', 'seem', 'like', 're', 'far', 'away', 'different', 'danger', 'zon
         e', 'could', 'nt', 'stop', 'wanted', 'found', 'somebody', 'else', 'sure', 'nt', 'know', 've', 'found', 'love', 'like', 'revol
         ving', 'door', 'yeah', 'temptation', 'gon', 'na', 'come', 'lives', 'gon', 'na', 'break', 'happy', 'home', 'temptation', 'goin
         g', 'know', 'better', 'leave', 'thing', 'alone', 'always', 'seem', 'distracted', 'like', 'mind', 'somewhere', 'else', 'nt',
         'want', 'alone', 'strange', 'guy', 're', 'gon', 'na', 'wind', 'shelf', 'yeah', 'temptation', 'gon', 'na', 'come', 'lives', 'g
         on', 'na', 'break', 'happy', 'home', 'temptation', 'going', 'know', 'better', 'yeah', 'yeah', 'leave', 'thing', 'alone', 'tem
         ptation', 'gon', 'na', 'go', 'lives', 'gon', 'na', 'break', 'happy', 'home', 'temptation', 'going', 'know', 'better', 'yeah',
         'yeah', 'temptation', 'gon', 'na', 'go', 'lives', 'gon', 'na', 'break', 'happy', 'home', 'temptation', 'going', 'know', 'bett
         er', 'yeah', 'yeah', 'oh']
         Εχουμε 94 tokens
         Αφού αφαιρέσαμε τα stopwords, έχουμε τελικά 49 tokens
         Αφού αφαιρέσαμε τα σημεία στίξης και τις μικρές λέξεις, έχουμε τελικά 42 tokens
```

Εικόνα 28: *Αφαίρεση stopwords, μικρών λέξεων και σημείων στίξης*

Τώρα, με αυτά τα tokens, θα χρησιμοποιήσουμε το **pos_tag(),** ώστε να βρούμε το μέρος του λόγου κάθε λέξη, έτσι ώστε να λημματοποιηθεί, και να εξεταστεί η συναισθηματική του αξία με βάση το μέρος του λόγου που έχει αυτή η λέξη.

```
for text in text_samples:
    #kanoume preprocessing ta dedomena mas
    word_tokens=word_tokenize(text.lower()) #kanoume tokenize
    print("Εχουμε " + str(len(word_tokens))+ " tokens")
    filtered_word_tokens = [word for word in word_tokens if word not in stop] #svinoume ta stopwords
    print("Αφού αφαιρέσαμε τα stopwords, έχουμε τελικά " + str(len(filtered_word_tokens)) + " tokens")
    #print(filtered_word_tokens)
    filtered_word_tokens = [re.sub(r'[^A-Za-z0-9]+', '', x) for x in filtered_word_tokens]  #svinoume ta punctuations
    filtered_word_tokens = [word for word in filtered_word_tokens if len(word)>1] #svinoume tis mikres lekseis
    print("Αφού αφαιρέσαμε τα σημεία στίξης και τις μικρές λέξεις, έχουμε τελικά " + str(len(filtered_word_tokens)) + " tokens")
    print(filtered_word_tokens)
    tagged_filtered_tokens=nltk.pos_tag(filtered_word_tokens) #kanoume pos tag tis lekseis gia syntaktiki analysi
    print(tagged_filtered_tokens)

[('tell', 'VB'), ('re', 'NN'), ('leaving', 'VBG'), ('ve', 'JJ'), ('got', 'VBD'), ('alone', 'RB'), ('sometimes', 'RB'), ('see
m', 'VBP'), ('like', 'IN'), ('re', 'NN'), ('far', 'RB'), ('away', 'RB'), ('different', 'JJ'), ('danger', 'NN'), ('zone', 'N
N'), ('could', 'MD'), ('nt', 'VB'), ('stop', 'VB'), ('wanted', 'VBN'), ('found', 'IN'), ('somebody', 'NN'), ('else', 'JJ'),
('sure', 'JJ'), ('nt', 'NN'), ('know', 'VBP'), ('ve', 'NN'), ('found', 'VBN'), ('love', 'IN'), ('like', 'IN'), ('revolving',
'VBG'), ('door', 'NN'), ('yeah', 'NN'), ('temptation', 'NN'), ('gon', 'NN'), ('na', 'TO'), ('come', 'VB'), ('lives', 'NNS'),
('gon', 'VBG'), ('na', 'TO'), ('break', 'VB'), ('happy', 'JJ'), ('home', 'NN'), ('temptation', 'NN'), ('going', 'VBG'), ('kno
w', 'VBP'), ('better', 'JJR'), ('leave', 'JJ'), ('thing', 'NN'), ('alone', 'RB'), ('always', 'RB'), ('seem', 'VBP'), ('distra
cted', 'VBN'), ('like', 'IN'), ('mind', 'NN'), ('somewhere', 'RB'), ('else', 'RB'), ('nt', 'JJ'), ('want', 'VBP'), ('alone',
'JJ'), ('strange', 'JJ'), ('guy', 'NN'), ('re', 'NN'), ('gon', 'NN'), ('na', 'TO'), ('wind', 'VB'), ('shelf', 'NN'), ('yeah',
'JJ'), ('temptation', 'NN'), ('gon', 'NN'), ('na', 'TO'), ('come', 'VB'), ('lives', 'NNS'), ('gon', 'VBG'), ('na', 'TO'), ('b
reak', 'VB'), ('happy', 'JJ'), ('home', 'NN'), ('temptation', 'NN'), ('going', 'VBG'), ('know', 'VBP'), ('better', 'JJR'),
('yeah', 'NN'), ('yeah', 'NNS'), ('leave', 'VBP'), ('thing', 'NN'), ('alone', 'JJ'), ('temptation', 'NN'), ('gon', 'NN'), ('n
a', 'TO'), ('go', 'VB'), ('lives', 'NNS'), ('gon', 'RB'), ('na', 'TO'), ('break', 'VB'), ('happy', 'JJ'), ('home', 'NN'), ('t
emptation', 'NN'), ('going', 'VBG'), ('know', 'VBP'), ('better', 'JJR'), ('yeah', 'NN'), ('yeah', 'NN'), ('temptation', 'N
N'), ('gon', 'NN'), ('na', 'TO'), ('go', 'VB'), ('lives', 'NNS'), ('gon', 'RB'), ('na', 'TO'), ('break', 'VB'), ('happy', 'J
J'), ('home', 'NN'), ('temptation', 'NN'), ('going', 'VBG'), ('know', 'VBP'), ('better', 'JJR'), ('yeah', 'NN'), ('yeah', 'N
```

Εικόνα 29*: Εφαρμογή POS_TAG*

Βλέπουμε λοιπόν πως σε κάθε token έχει αποδοθεί ένα μέρος του λόγου. Χρησιμοποιώντας αυτά τα στοιχεία μαζί με το **lemmatizer()** μπορούμε να περάσουμε κάθε λέξη από το εργαλείο **senti_synsets**, ώστε να βρούμε το συναισθηματικό βάρος κάθε λέξης. Για να το κάνουμε αυτό, όμως, πρέπει να μετατρέψουμε το output του pos_tag ώστε να είναι συμβατό από το senti_synset. Επειδή δεν χρειαζόμαστε όλες τις περιπτώσεις στο βάθος που τα αναλύει το pos_tag (πχ ο πληθυντικός ουσιαστικών, NNP, θα γίνει απλά ουσιαστικό, N), θα τα προσαρμόσουμε κατάλληλα. Παρακάτω εκτυπώνουμε τα απλοποιημένα μέρη του λόγου που θα μπουν σαν είσοδο στο senti_synset, μαζί με τις λημματοποιημένες λέξεις.

```
    tagged_filtered_tokens=nltk.pos_tag(filtered_word_tokens) #kanoume pos tag tis lekseis gia syntaktiki analysi
    #print(tagged_filtered_tokens)
            #ftiaxnoume ta pos tags wste na mpoun san input sto sentisynset
    #epeidh ta dedomena tou postag ginontai tuples, ta metatrepoume se lista
    newtags=[] #lista me ta nea tags kai idio counter me ta tagged words
    for tag in tagged_filtered_tokens:
        if tag[1] in set(['VB', 'VBD', 'VBG', 'VBN', 'VBP', 'VBZ']):
            newtags.append('v')
        elif tag[1] in set(['JJ', 'JJR', 'JJS']):
             newtags.append('a')
        elif tag[1] in set(['RB', 'RBR', 'RBS']):
             newtags.append('r')
        elif tag[1] in set(['NNS', 'NN', 'NNP', 'NNPS']):
             newtags.append('n')
        else:
             newtags.append('a')

    lem_words=[] #edw tha mpoun oi lematized lekseis pou exoume kratisei apo to preprocessing
    counter=0 #vazoume ton counter gia na kanoume iterate ta stoixeia tis listas twn tags
    for word in tagged_filtered_tokens:
        lem_words.append(wnl.lemmatize(word[0],newtags[counter]))
        counter+=1
       # print (newtags)
   # new_words_tags_dict = {'word':'synscore'}
    print(newtags,lem_words)
```

```
Αφού αφαιρέσαμε τα σημεία στίξης και τις μικρές λέξεις, έχουμε τελικά 119 tokens
['v', 'n', 'v', 'a', 'v', 'r', 'r', 'v', 'a', 'n', 'r', 'r', 'a', 'n', 'n', 'a', 'v', 'v', 'v', 'a', 'n', 'a', 'a', 'n', 'v',
'n', 'v', 'a', 'a', 'v', 'n', 'n', 'n', 'n', 'a', 'v', 'n', 'v', 'a', 'v', 'a', 'n', 'n', 'v', 'v', 'a', 'a', 'n', 'r', 'r',
'v', 'v', 'a', 'n', 'r', 'r', 'a', 'v', 'a', 'a', 'n', 'n', 'n', 'a', 'v', 'n', 'a', 'n', 'n', 'a', 'v', 'n', 'v', 'a', 'v',
'a', 'n', 'n', 'v', 'v', 'a', 'n', 'n', 'v', 'n', 'a', 'n', 'n', 'a', 'v', 'n', 'r', 'a', 'v', 'a', 'n', 'n', 'v', 'v', 'a',
'n', 'n', 'n', 'n', 'a', 'v', 'n', 'r', 'a', 'v', 'a', 'n', 'n', 'v', 'v', 'a', 'n', 'n', 'n'] ['tell', 're', 'leave', 've',
'get', 'alone', 'sometimes', 'seem', 'like', 're', 'far', 'away', 'different', 'danger', 'zone', 'could', 'nt', 'stop', 'wan
```

***Εικόνα 30**: Δημιουργία λίστας POS_TAG κατάλληλη για να περαστεί στο SentWordNet και λημματοποιημένων tokens*

Στην συνέχεια, παίρνουμε τα 2 ορίσματα αυτά και τα περνάμε μέσα από το **senti_synset**. Το senti_synset ελέγχει ένα προαποθηκευμένο λεξικό, με τη λέξη και το μέρος του λόγου, και επιστρέφει, θετικό και αρνητικό σκορ. Για κάθε κείμενο,

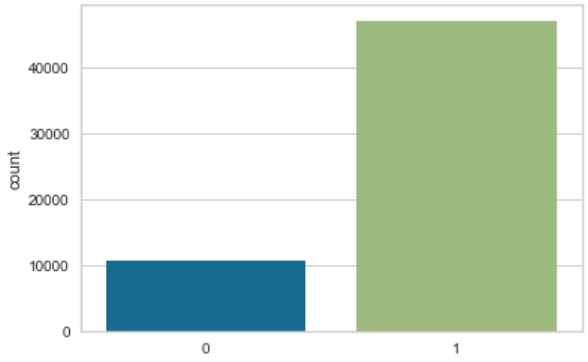
```
    #ypologizoume to synaisthima kathe leksis , mazi me to POSTAG tis
    posscore=0
    negscore=0
    for i in range(len(lem_words)):
        synsets = swn.senti_synsets(lem_words[i],newtags[i])
        for synst in synsets: #athroizoume ta thetika kai ta arnhtika score kathe leksis
            posscore=posscore+synst.pos_score()
            negscore=negscore+synst.neg_score()
    my_sentiments.append(posscore-negscore)

    if (posscore-negscore)>=0:
        my_sentiments_class.append(1)
    else:
        my_sentiments_class.append(0)

print(my_sentiments_class)
print(my_sentiments)
```

```
[1, 1, 1, 0, 1, 1, 1, 1, 1, 1]
[74.388, 4.5, 6.5, -12.875, 53.918000000000006, 16.418000000000006, 25.689999999999998, 8.834000000000001, 9.680999999999997,
4.75]
```

***Εικόνα 31**: Αποτελέσματα Senti_Synset και δημιουργία συναισθηματικής κλάσης*

υπολογίζουμε και τα 2 αθροίσματα, και αναθέτουμε στη κλάση συναισθημάτων μας το ανάλογο συναίσθημα: 1 αν το θετικό συναισθηματικό σκορ του αθροίσματος των λέξεων κάθε κειμένου είναι μεγαλύτερο από το άθροισμα των αρνητικών συναισθηματικών σκορ, διαφορετικά 0.

Τελικά, αποθηκεύουμε την κλάση που δημιουργήσαμε στο **my_sentiments_class**, η οποία έδωσε τα εξής αποτελέσματα:




```
Έχουμε 47095 χαρούμενα τραγούδια
Έχουμε 10555 στενάχωρα τραγούδια
```

***Εικόνα 32:** Ιστόγραμμα με Συναισθηματικές κλάσεις που προέκυψαν από την ανάλυση με SentiWordNet*

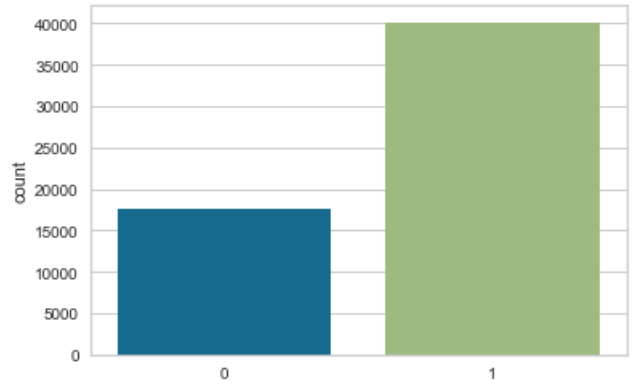
4.6.1.2 Συναισθηματική Ανάλυση με Vader

Καθώς ο Vader έχει μεγαλύτερη αποτελεσματικότητα σε μικρού εύρους κείμενα, θα χωρίσουμε το κάθε κείμενο μας σε προτάσεις, τις οποίες θα αξιολογήσουμε συναισθηματικά. Στο τέλος κάθε κειμένου θα αθροίζουμε τα θετικά και τα αρνητικά σκορ που βρήκαμε. Αν το θετικό σκορ είναι μεγαλύτερο από το αρνητικό, το κείμενο

θα θεωρείται χαρούμενο, και θα σημειώνουμε σε μία νέα λίστα *Vader_class_sentiments* 1 για θετικά τραγούδια και 0 για αρνητικά.

Παραθέτουμε τα αποτελέσματα συναισθηματικής ανάλυσης με Vader:

```
Έχουμε 40128 χαρούμενα τραγούδια
Έχουμε 17522 στενάχωρα τραγούδια
```



***Εικόνα 33**: : Ιστόγραμμα με Συναισθηματικές κλάσεις που προέκυψαν από την ανάλυση με Vader*

4.6.2 Μουσικά Δεδομένα

Παραθέτουμε τα δεδομένα που έχουμε με τις συναισθηματικές κλάσεις:

	Nro	Scary	Happy	Sad	Peaceful
0	1	1.2889	4.4667	1.7111	3.1333
1	2	1.0667	5.4444	1.4889	4.4889
2	3	2.0222	1.4889	3.7778	2.7111
3	4	2.2889	4.1111	1.2667	1.4889
4	5	1.4000	1.4667	5.0444	3.8444

***Εικόνα 34**: Συναισθηματικές Κλάσεις για μουσικό σετ δεδομένων*

Το σετ δεδομένων έχει χωρίσει σε 4 συναισθηματικές κλάσεις. Αν και εξετάζοντας το σετ δεδομένων, βλέπουμε ότι κατά κύριο λόγο τραγούδια που έχουν μεγάλη βαθμολογία στη κλάση *Happy* έχουν επίσης μεγάλη βαθμολογία στο *Peaceful*. Επίσης, τραγούδια που έχουν βαθμολογία, μεγάλη στο *Sad* έχουν μεγάλη βαθμολογία στο *Scary*. Δεν είναι όμως όλα, έτσι. Μετά από έλεγχο που κάναμε, προέκυψε το εξής:

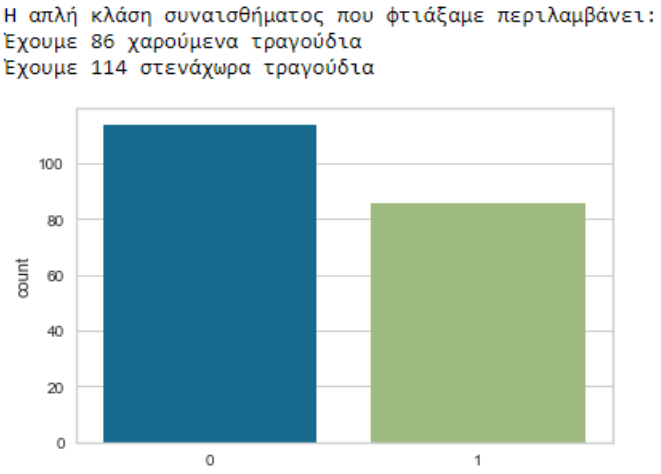
```
Έχουμε 22 περιπτώσεις όπου happy>sad και peaceful < scary
Συνεπώς υπάρχει αμφισημία μεταξύ των κλάσεων από το dataset μας.
Θα δημιουργήσουμε 3 διαφορετικές περιπτώσεις με βάση τις υπάρχουσες κλάσεις, ώστε να καλύψουμε τις διαφορετικές εκδοχές
```

***Εικόνα 35**: Εξέταση Μουσικού Σετ Δεδομένων*

Καθώς τα συναισθήματα “Happy” και “Peaceful” είναι ομόλογα, όπως και τα “Sad” “Scary” θα εξετάσουμε διαφορετικές περιπτώσεις για να μπορούμε να εκφράσουμε τις 4 αυτές συναισθηματικές κλάσεις σε δύο, καθώς έχουμε δυαδική συναισθηματική ανάλυση. Συνεπώς, πρέπει να εξετάσουμε πως θα χειριστούμε τις 22 περιπτώσεις αυτές. Καταλήγουμε στις εξής περιπτώσεις:

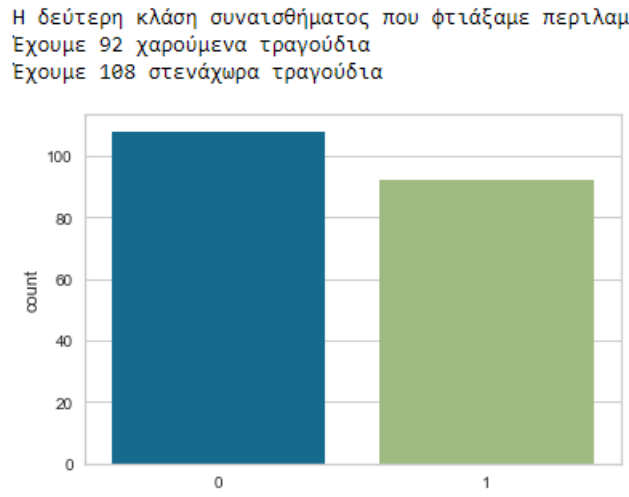
Για να καλύψουμε όλες τις περιπτώσεις, και να συσχετίσουμε το σετ δεδομένων αυτό με τη διπλωματική εργασία, θα χωρίσουμε 3 περιπτώσεις, δημιουργώντας 3 διαφορετικά μοντέλα.

a) Στην πρώτη και απλή περίπτωση, θα εξετάσουμε σαν συναισθηματική κλάση μόνο τις ταμπέλες *Happy και Sad: Αν Happy>Sad το τραγούδι είναι χαρούμενο.*



***Εικόνα 36**: Ιστόγραμμα πρώτης περίπτωσης για συναισθηματικές κλάσεις μουσικού σετ*

b) Στην δεύτερη περίπτωση, το τραγούδι θεωρείται *Happy* αν *Happy + Peaceful > Sad + Scary*, αλλιώς είναι *Sad.*

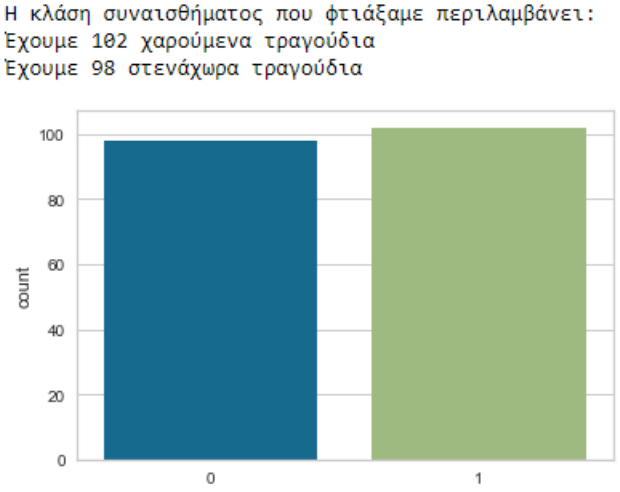


***Εικόνα 37**: Ιστόγραμμα δεύτερης περίπτωσης για συναισθηματικές κλάσεις μουσικού σετ*

c) Στην Τρίτη περίπτωση, εφαρμόζουμε το εξής: Καθώς το Happy είναι ομόλογο συναίσθημα με το Peaceful,όπως και το Sad με το Scary, θεωρούμε το Happy σαν συναίσθημα πιο σημαντικό για την δυαδική μας κατηγοριοποίηση. Συνεπώς:

i) *Αν το Happy >= Sad & Peaceful >= Scary, τότε το τραγούδι θα θεωρηθεί χαρούμενο, και αντίστροφα*

ii) *Αν το Happy >= Sad & Peaceful < Scary, τότε έχουμε ανισορροπία μεταξύ των συναισθηματικών μας κλάσεων. Τότε εξετάζουμε ποια διαφορά είναι μεγαλύτερη: Αν 1.5 *(Happy-Sad) > 0.5* (Scary – Peaceful), τότε το τραγούδι είναι χαρούμενο, και αντίστροφα.*



***Εικόνα 38**: Ιστόγραμμα τρίτης περίπτωσης για συναισθηματικές κλάσεις μουσικού σετ*

Όπως είπαμε, τα σετ δεδομένων κριτικών των imdb και στίχων δεν χρειάζονται τροποποίηση, καθώς ο τρόπος με τον οποίο είναι δομημένος η πληροφορία, είναι ταιριαστός με την ανάλυση που θέλουμε να κάνουμε.

4.7 Εκπαίδευση Μοντέλων Μηχανικής Μάθησης

Έχοντας κρατήσει το λεξιλόγιο των documents μας με τα βάρη που προαποφασίσαμε θα εφαρμόσουμε διαφορετικά μοντέλα μηχανικής μάθησης, τα οποία θα μπορούν να προβλέψουν μελλοντικά συναισθήματα με βάση το στίχο.

4.7.1 Διαχωρισμός Σετ Εκπαίδευσης / Σετ Ελέγχου

Αρχικά, θα χωρίσουμε το σετ δεδομένων μας σε σετ εκπαίδευσης και σετ ελέγχου. Έτσι, θα εκπαιδεύσουμε το μοντέλο μας με το σετ εκπαίδευσης και θα το αξιολογήσουμε με το σετ ελέγχου. Η συνήθης αναλογία μεταξύ τους είναι 80/20, την οποία και θα εφαρμόσουμε.

Έτσι, καλούμε την train_test_split() με τα κατάλληλα ορίσματα για τα διαφορετικά σετ δεδομένωνs μας, με τις κατάλληλες συναισθηματικές κλάσεις. Θα καλέσουμε την συνάρτηση αυτή για κάθε διαφορετικό σετ δεδομένων ή διαφορετικό output.

Για τα **στιχουργικά** δεδομένα:

Αρχικά, στο **X_train_s** αποθηκεύουμε το 80% των *στιχουργικών* δεδομένων μας, και στο **y_train_s** αποθηκεύουμε τις κλάσεις των συναισθημάτων αυτών που προέκυψαν από *συναισθηματική ανάλυση με SentiWordNet*. Αυτά θα αποτελέσουν τα **διανύσματα εκπαίδευσης** των ταξινομητών μας. Αντίστοιχα, δημιουργούμε τα **X_test_s** και **y_test_s** σαν **διανύσματα ελέγχου** των ταξινομητών.

Μετέπειτα, στο **X_train_v** αποθηκεύουμε το 80% των *στιχουργικών* δεδομένων μας, και στο **y_train_v** αποθηκεύουμε τις κλάσεις των συναισθημάτων αυτών που προέκυψαν από την συναισθηματική ανάλυση μέσω του *Vader*. Αυτά θα αποτελέσουν τα **διανύσματα εκπαίδευσης** των ταξινομητών μας. Αντίστοιχα, δημιουργούμε τα **X_test_v** και **y_test_v** σαν **διανύσματα ελέγχου** των ταξινομητών..

Στην συνέχεια, στο **X_train** αποθηκεύουμε το 80% των *κειμενικών* δεδομένων μας από την ανάλυση κριτικών στο imdb, και στο **y_train** αποθηκεύουμε τις κλάσεις των συναισθημάτων αυτών που προέκυψαν *από το σετ δεδομένων αυτό*. Αυτά θα αποτελέσουν τα **διανύσματα εκπαίδευσης** των ταξινομητών μας. Αντίστοιχα, δημιουργούμε τα **X_test** και **y_test** σαν **διανύσματα ελέγχου** των ταξινομητών.

Για τα **μουσικά** δεδομένα, δημιουργήσαμε αντίστοιχα διανύσματα εκπαίδευσης και διανύσματα ελέγχου, για κάθε υποσετ δεδομένων. Αξίζει να σημειώσουμε, πως κατόπιν δοκιμών αποφασίσαμε να κανονικοποιήσουμε τα μουσικά δεδομένα μας χρησιμοποιώντας την συνάρτηση StandardScalar(), δημιουργώντας τα διανύσματα X_train_scaled1, X_train_scaled2,X_train_scaled3, , X_test_scaled2, X_test_scaled3.

Αρχικά, στο **X_train_s** αποθηκεύουμε το 80% των *μουσικών* δεδομένων μας, και στο **y_train_s** αποθηκεύουμε τις κλάσεις των συναισθημάτων αυτών που προέκυψαν από την *απλή ανάλυση, που εξετάζει μόνο το 'Happy' και το 'Sad'*. Αντίστοιχα, δημιουργούμε τα **X_test_s** και **y_test_s**. Μετά την κανονικοποιήση, δημιουργούμε το **X_train_scaled1** το οποίο μαζί με το y_train_s θα αποτελέσει το **διανύσματα εκπαίδευσης** των ταξινομητών μας. Τέλος . σαν **διανύσματα ελέγχου** μαζί με το y_test_s, δημιουργούμε μετά από κανονικοποίηση το **X_test_scaled1**.

Μετέπειτα, στο **X_train_b** αποθηκεύουμε το 80% των *μουσικών* δεδομένων μας, και στο **y_train_b** αποθηκεύουμε τις κλάσεις των συναισθημάτων αυτών που προέκυψαν από την *ανάλυση που εξετάζει τα αθροίσματα 'Happy'+'Peaceful' και 'Scary'+'Sad'*. Αντίστοιχα, δημιουργούμε τα **X_test_b** και **y_test_b**. Μετά την κανονικοποιήση, δημιουργούμε το **X_train_scaled2** το οποίο μαζί με το y_train_s θα αποτελέσει το **διανύσματα εκπαίδευσης** των ταξινομητών μας. Τέλος . σαν **διανύσματα ελέγχου** μαζί με το y_test_b, δημιουργούμε μετά από κανονικοποίηση το **X_test_scaled2**.

Στην συνέχεια, στο **X_train_B** αποθηκεύουμε το 80% των *μουσικών* δεδομένων μας, και στο **y_train_B** αποθηκεύουμε τις κλάσεις των συναισθημάτων αυτών που προέκυψαν *την ανάλυση που εξετάζει τις διαφορετικές περιπτώσεις μεταξύ των διαφορών 'Happy'-'Sad' – 'Peaceful – 'Scary'*. Αντίστοιχα, δημιουργούμε τα **X_test_B** και **y_test_B**. Μετά την κανονικοποιήση, δημιουργούμε το **X_train_scaled3** το οποίο μαζί με το y_train_B θα αποτελέσει το **διανύσματα εκπαίδευσης** των ταξινομητών μας. Τέλος . σαν **διανύσματα ελέγχου** μαζί με το y_test_B, δημιουργούμε μετά από κανονικοποίηση το **X_test_scaled3**.

4.7.2 Εξαγωγή Στοιχείων Κειμένου

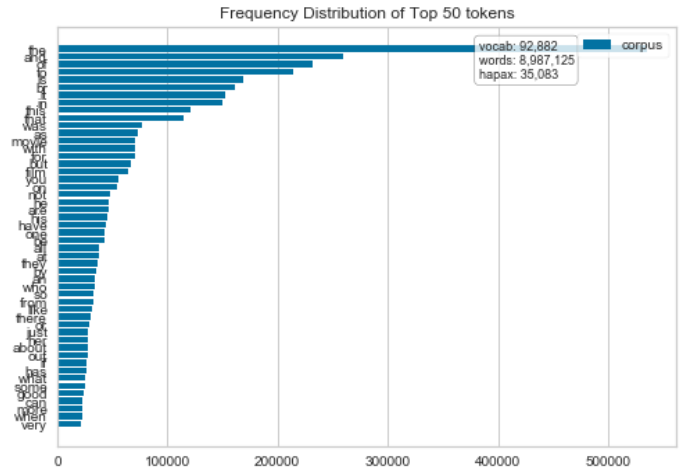
Καθώς για τα κειμενικά μας δεδομένα έχουμε τεράστιο αριθμό στοιχείων, σε αντίθεση με τα μουσικά, θα βρούμε τα καλύτερα δυνατά διανύσματα για τα σετ δεδομένων μας, τα οποία στη συνέχεια θα παραμετροποιήσουμε. Συγκεκριμένα , θα εφαρμόσουμε την συνάρτηση fit() και transform(), ώστε ο CountVectorizer() και ο TFidfVectorizer() να μάθουν το vocabulary του κάθε σετ κειμενικών δεδομένων.

Η παραμετροποίηση διανυσμάτων θα γίνει σταδιακά. Αρχικά, για κάθε σετ δεδομένων, θα παρουσιάσουμε μία βάση διανυσμάτων, χωρίς να χρησιμοποιήσουμε τις συναρτήσεις καθαρισμού και τμηματοποίησης tokenizer_preprocessor() και tokenizer_preprocessor_imdb()που φτιάξαμε, και στη συνέχεια τα διανύσματα με τις συναρτήσεις αυτές.

Σετ δεδομένων Imdb:

Με CountVectorizer() χωρίς tokenizer:

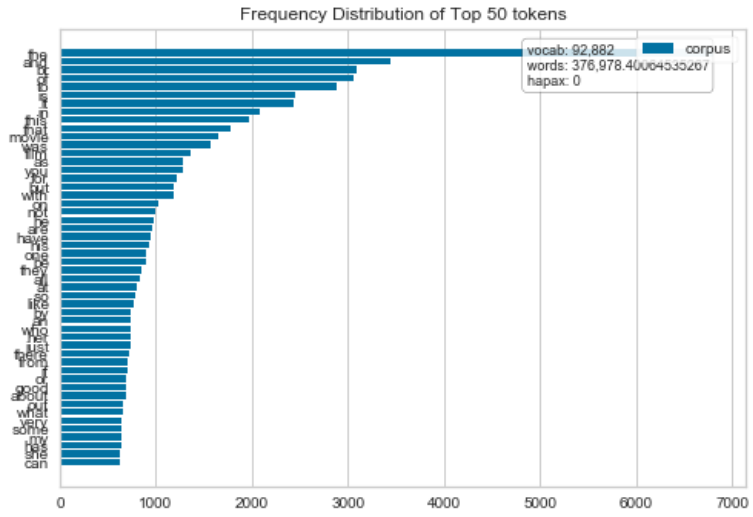
```
Παρουσιάζουμε την επιτυχία των vectorizers για τη συναισθηματική κλάση από SentiWordNet ΧΩΡΙΣ TOKENIZER
Number of features: 92882
to accuracy tou CountVectorizer me NB einai: 0.8466500942490011
```



Εικόνα 39: *Παρουσίαση πιο συχνών features για Σετ Δεδομένων IMDB με CountVectorizer() χωρίς την συνάρτηση tokenizer_preprocessor()*

Με TFidfVectorizer() χωρίς tokenizer:

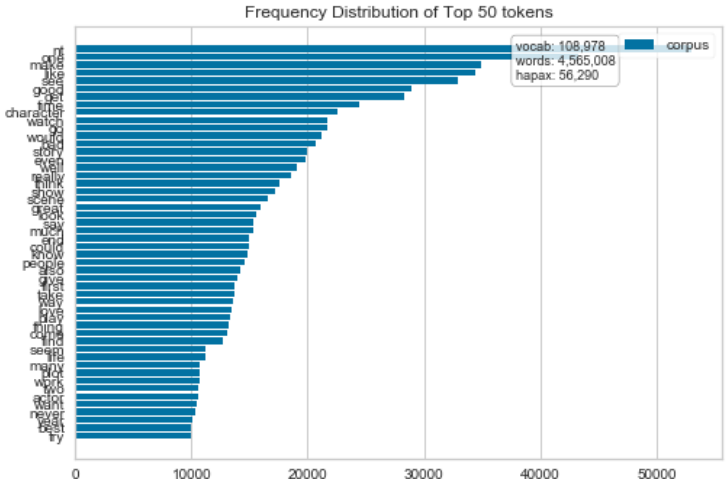
```
Number of features: 92882
to accuracy tou TFIDF me NB einai 0.8610001468192514
```



***Εικόνα 40**: : Παρουσίαση πιο συχνών features για Σετ Δεδομένων IMDB με TFIdfVectorizer() χωρίς την συνάρτηση tokenizer_preprocessor()*

Με CountVectorizer() με tokenizer:

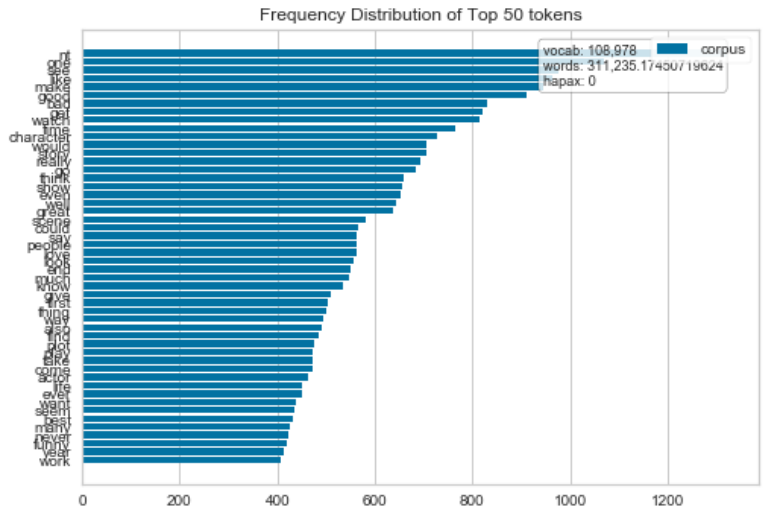
```
Παρουσιάζουμε την επιτυχία των vectorizers για τη συναισθηματική κλάση από SentiWordNet ΜΕ TOKENIZER
Number of features: 108978
to accuracy tou CountVectorizer me NB einai: 0.8562751167917511
```



***Εικόνα 41**:: Παρουσίαση πιο συχνών features για Σετ Δεδομένων IMDB με CountVectorizer() με την συνάρτηση tokenizer_preprocessor()*

Με TFidfVectorizer() με tokenizer

```
Number of features: 108978
to accuracy tou TFIDF me NB einai 0.8618500543180005
```



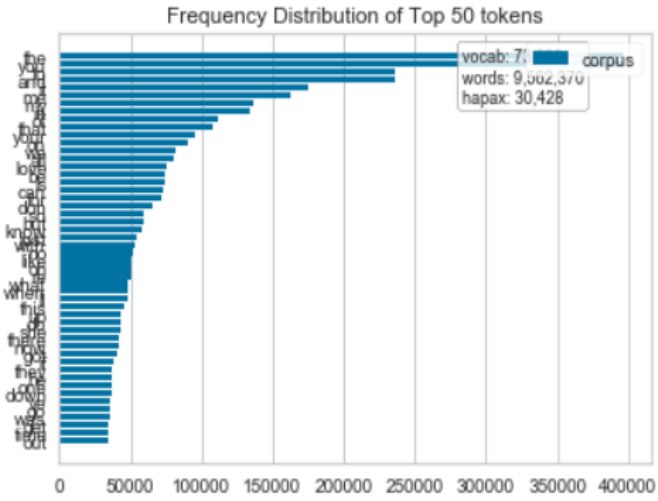
***Εικόνα 42**: : Παρουσίαση πιο συχνών features για Σετ Δεδομένων IMDB με TfidftVectorizer() με την συνάρτηση tokenizer_preprocessor()*

Παρατηρούμε ότι καλώς αφαιρέσαμε τις λέξεις *'film','movie'* καθώς εμφανίζουν μεγάλη συχνότητα, όπως και τα stopwords όπως *'the','and'*,'this'. Οι λέξεις αυτές καταλαμβάνουν θέσεις στο feature vector, στο οποίο ιδανικά θέλουμε να κρατήσουμε τις λέξεις ή τα n-grams τα οποία φέρουν συναισθηματική πληροφορία, τα οποία θα συσχετιστούν μετ΄έπειτα με την συναισθηματική κλάση.Η αφαίρεση τους βελτίωσε την απόδοση μας.

Σετ στιχουργικών δεδομένων με ανάλυση Vader

Με CountVectorizer() χωρίς tokenizer:

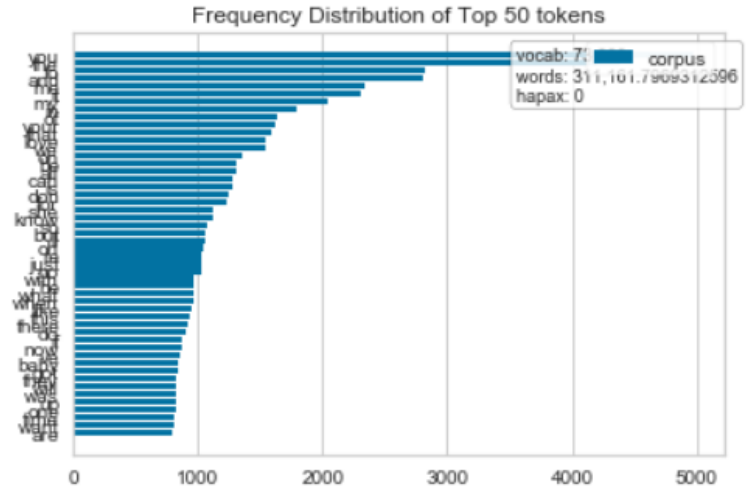
```
Παρουσιάζουμε την επιτυχία των vectorizers για τη συναισθηματική κλάση από vader ΧΩΡΙΣ tokenizer:
Number of features: 73880
to accuracy tou CountVectorizer me NB einai: 0.7728100607111882
```



Εικόνα 43: *Παρουσίαση πιο συχνών features για Σετ Δεδομένων από συναισθηματική Ανάλυση με Vader με CountVectorizer() χωρίς την συνάρτηση tokenizer_preprocessor()*

Με TfidfVectorizer() χωρίς tokenizer:

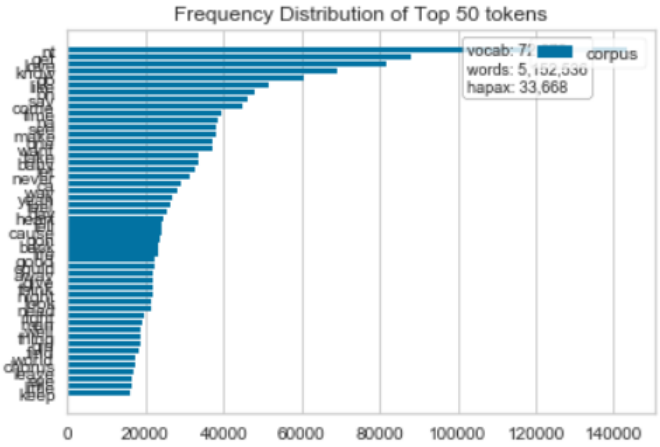
```
Number of features: 73880
to accuracy tou TFIDF me NB einai 0.7040546400693841
```



Εικόνα 44*: Παρουσίαση πιο συχνών features για Σετ Δεδομένων από συναισθηματική Ανάλυση με Vader με TfidfVectorizer() χωρίς την συνάρτηση tokenizer_preprocessor()*

Με CountVectorizer() με tokenizer:

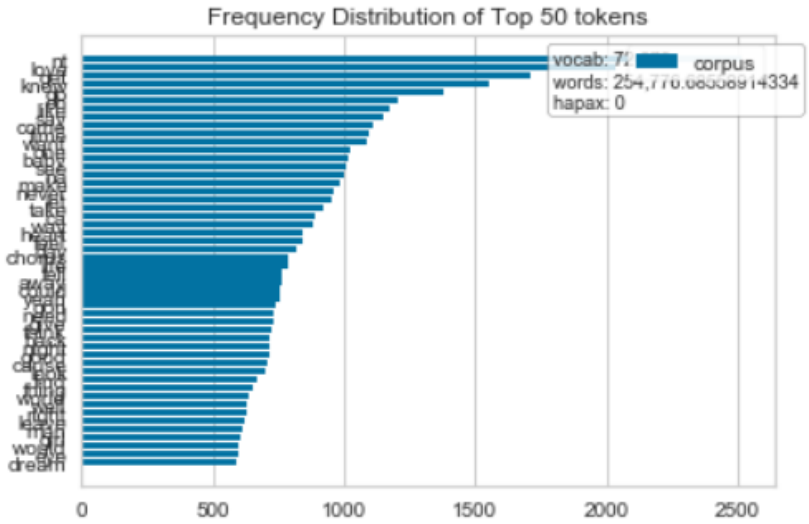
```
Παρουσιάζουμε την επιτυχία των vectorizers για τη συναισθηματική κλάση από vader ΜΕ tokenizer
Number of features: 72672
to accuracy tou CountVectorizer me NB einai: 0.7801170858629661
```



Εικόνα 45: *Παρουσίαση πιο συχνών features για Σετ Δεδομένων από συναισθηματική Ανάλυση με Vader με CountVectorizer() με την συνάρτηση tokenizer_preprocessor()*

Με TfidftVectorizer() με tokenizer:

```
Number of features: 72672
to accuracy tou TFIDF me NB einai 0.7082827406764961
```



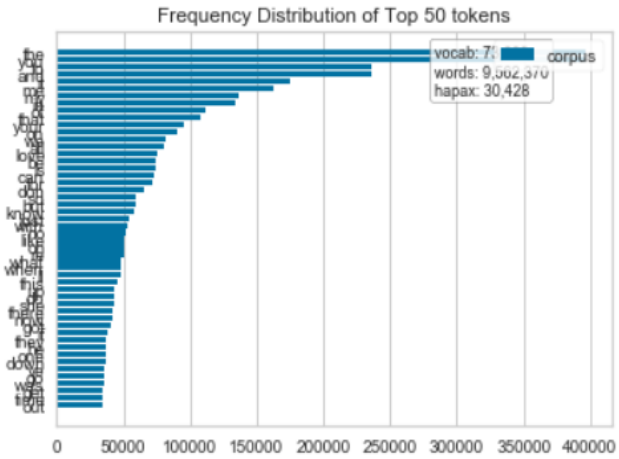
Εικόνα 46*: Παρουσίαση πιο συχνών features για Σετ Δεδομένων από συναισθηματική Ανάλυση με Vader με TfidfVectorizer() με την συνάρτηση tokenizer_preprocessor()*

Παρατηρούμε και εδώ, ότι η αφαίρεση των stopwords πέρα από την βελτίωση της ταχύτητας, βελτιώνει και την απόδοση μας.

Σετ δεδομένων με Ανάλυση με SentiWordNet

Με CountVectorizer() χωρίς tokenizer:

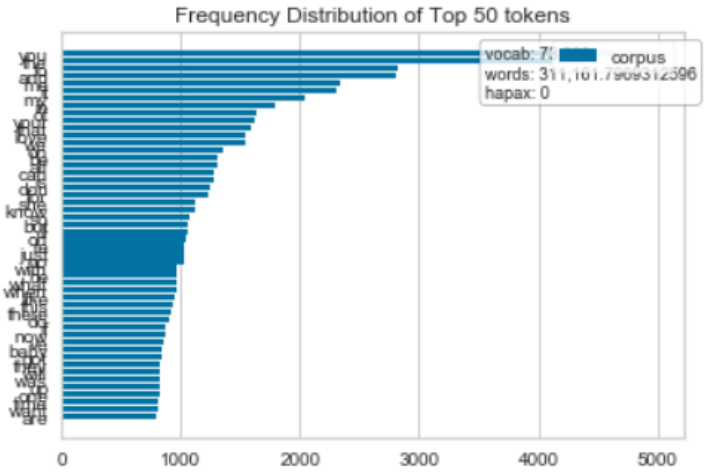
```
Παρουσιάζουμε την επιτυχία των vectorizers για τη συναισθηματική κλάση από SentiWordNet ΧΩΡΙΣ TOKENIZER
Number of features: 73880
to accuracy tou CountVectorizer me NB einai: 0.8073934933150149
```



***Εικόνα 47**: Παρουσίαση πιο συχνών features για Σετ Δεδομένων από συναισθηματική Ανάλυση με SentiWordNet με CountVectorizer() χωρίς την συνάρτηση tokenizer_preprocessor()*

Με TfidftVectorizer() χωρίς tokenizer:

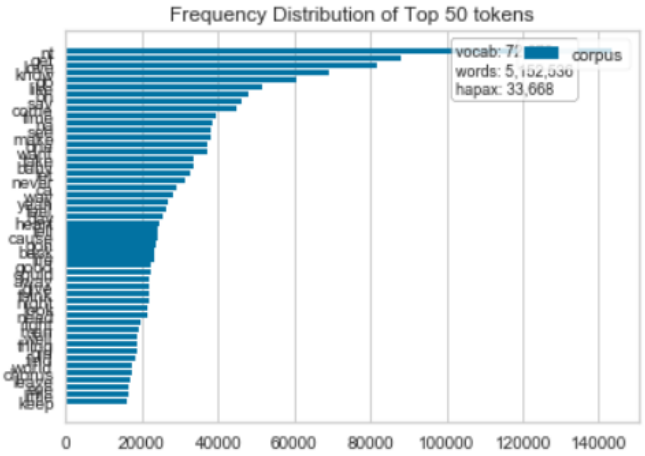
```
Number of features: 73880
to accuracy tou TFIDF me NB einai 0.8149826513687691
```



***Εικόνα 48:** Παρουσίαση πιο συχνών features για Σετ Δεδομένων από συναισθηματική Ανάλυση με SentiWordNet με TfidftVectorizer() χωρίς την συνάρτηση tokenizer_preprocessor()*

Με CountVectorizer() με tokenizer:

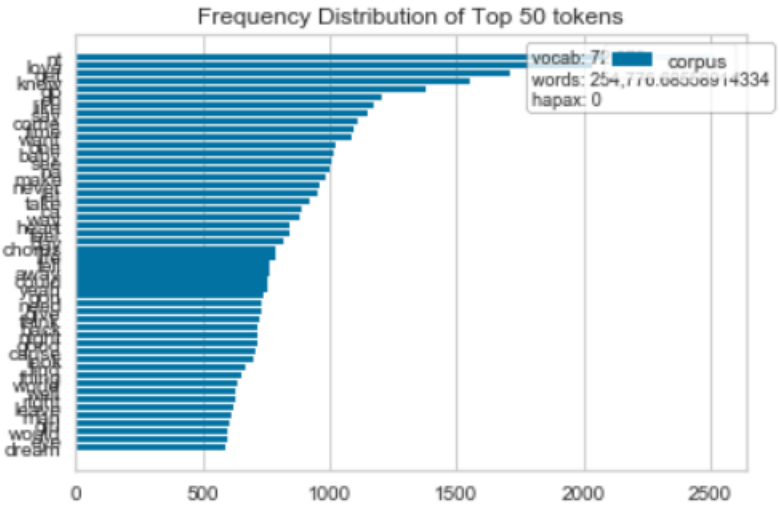
```
Παρουσιάζουμε την επιτυχία των vectorizers για τη συναισθηματική κλάση από SentiWordNet ΜΕ TOKENIZER
Number of features: 72672
to accuracy tou CountVectorizer me NB einai: 0.8197527259160483
```



Εικόνα 49: *Παρουσίαση πιο συχνών features για Σετ Δεδομένων από συναισθηματική Ανάλυση με SentiWordNet με CountVectorizer() με την συνάρτηση tokenizer_preprocessor()*

Με TfidfVectorizer() με tokenizer:

```
Number of features: 72672
to accuracy tou TFIDF me NB einai 0.8150043376967286
```



Εικόνα 50*: Παρουσίαση πιο συχνών features για Σετ Δεδομένων συναισθηματικής Ανάλυσης με SentiWordNet με TfidfVectorizer() με την συνάρτηση tokenizer_preprocessor()*

4.7.3 Αξιολόγηση Διανυσμάτων

Για κάθε ένα από τα κειμενικά σετ δεδομένων μας, θα αξιολογήσουμε τα διαφορετικά Vectorizers που χρησιμοποιήσαμε (CountVectorizer και TF-IDF) χρησιμοποιώντας ένα απλό Naïve Bayes μοντέλο χωρίς παραμετροποίηση, ώστε να δούμε ποιο είναι καλύτερο με το για το κάθε σετ δεδομένων μας. Στην συνέχεια, θα παραμετροποιήσουμε τον vectorizer αυτόν, και τελικά θα εφαρμόσουμε ταξινομητές. Η αξιολόγηση θα γίνει απλοϊκά, χρησιμοποιώντας το accuracy_score() της βιβλιοθήκης NLTK.

4.7.3.1 Σετ δεδομένων Imdb

Παραθέτουμε τα αποτελέσματα για το vectorizing που κάναμε στο σετ δεδομένων, χρησιμοποιώντας CountVectorizer() και TFIDFVectorizer().

	params	scores
0	cvec without tokenizer	0.846650
1	tvec without tokenizer	0.861000
2	cvec with tokenizer	0.856275
3	tvect with tokenizer	0.861850

Πίνακας 2: *Αποτελέσματα για τα διαφορετικά διανύσματα αναπαράστασης κειμενικών στοιχείων για σετ δεδομένων IMDB*

Θα χρησιμοποιήσουμε τον TFidfVectorizer() για εκπαίδευση του μοντέλου μας.

4.7.3.2 Σετ Στιχουργικών δεδομένων με Ανάλυση Vader

Παραθέτουμε τα αποτελέσματα εξαγωγής στοιχείων στη συναισθηματική ανάλυση στίχων με χρήση Vader, με διαφορετικά διανύσματα.Αρχικά δημιουργήσαμε διανύσματα χωρίς την συνάρτηση tokenizer_preprocessor() που έχουμε φτιάξει, για να έχουμε μία βάση ανάλυσης.

	params	scores
0	cvec without tokenizer	0.772810
1	tvec without tokenizer	0.704055
2	cvec with tokenizer	0.780117
3	tvect with tokenizer	0.708283

Πίνακας 3*: Αποτελέσματα για τα διαφορετικά διανύσματα αναπαράστασης κειμενικών στοιχείων για σετ δεδομένων από συναισθηματική ανάλυση με Vader*

Όπως βλέπουμε, μετά την επαλήθευση που κάναμε το καλύτερο σκορ για τους 2 διαφορετικούς τρόπους να συλλέξουμε τα δεδομένα μας είναι με τον CountVectorizer().

4.7.3.3 Σετ δεδομένων με συναισθηματική Ανάλυση με SentiWordNet

	params	scores
0	cvec without tokenizer	0.807393
1	tvec without tokenizer	0.814983
2	cvec with tokenizer	0.819753
3	tvect with tokenizer	0.815004

Πίνακας 4: *Αποτελέσματα για τα διαφορετικά διανύσματα αναπαράστασης κειμενικών στοιχείων για σετ δεδομένων από συναισθηματική ανάλυση με SentiWordNet*

Όπως βλέπουμε και εδώ ο καλύτερος τρόπος συλλογής στοιχείων είναι ο CountVectorizer().

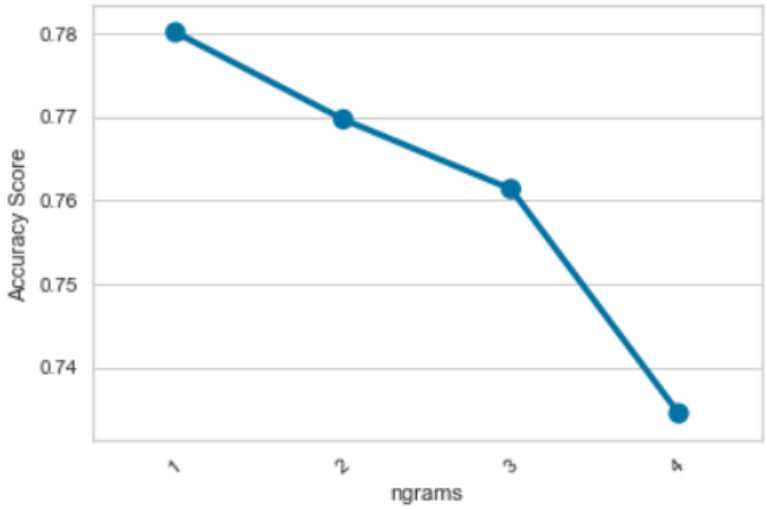
4.7.4 Παραμετροποίηση Διανυσμάτων

Στη συνέχεια, θα δοκιμάσουμε διαφορετικές παραμέτρους στα διανύσματα μας, χρησιμοποιώντας τις συναρτήσεις που έχουμε φτιάξει. Συγκεκριμένα, θα δοκιμάσουμε διαφορετικές παραμέτρους για τα max_df, max_features και n_grams. Θα κρατήσουμε το καλύτερο σκορ από αυτά και θα το εφαρμόσουμε στον ταξινομητή μας. Όλοι οι υπολογισμοί γίνονται με επαλήθευση (validation).

4.7.4.1 Σετ δεδομένων με ανάλυση Vader

Παραθέτουμε τα αποτελέσματα μας για τα διαφορετικά n-grams:

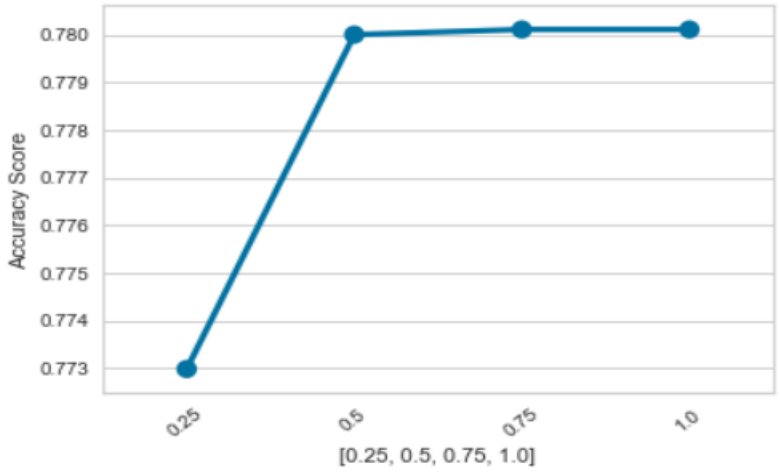
```
Τα αποτελέσματα για τα διαφορετικά ngrams είναι
   params    scores
0       1  0.780117
1       2  0.769688
2       3  0.761340
3       4  0.734454
```



Εικόνα 51*: Γραφική παράσταση και αποτελέσματα για διαφορετικά n-grams του CountVectorizer για: Σετ δεδομένων Vader*

Παραθέτουμε τα αποτελέσματα για τα διαφορετικά max df

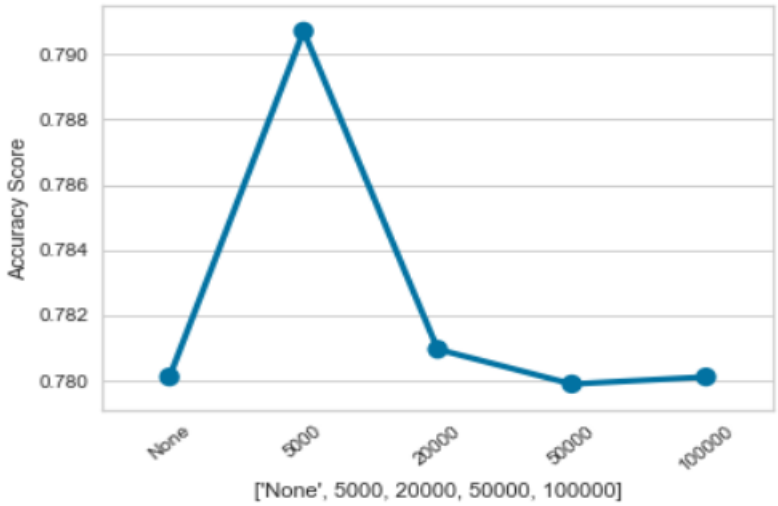
```
Τα αποτελέσματα για τα διαφορετικά max df είναι
   params    scores
0    0.25  0.772984
1    0.50  0.780009
2    0.75  0.780117
3    1.00  0.780117
```



***Εικόνα 52**: Γραφική παράσταση και αποτελέσματα για διαφορετικά max_df του CountVectorizer για: Σετ δεδομένων Vader*

Παραθέτουμε τα αποτελέσματα για τα διαφορετικά max features

```
Τα αποτελέσματα για τα διαφορετικά max features είναι
   params    scores
0    None  0.780117
1    5000  0.790698
2   20000  0.780963
3   50000  0.779900
4  100000  0.780117
```



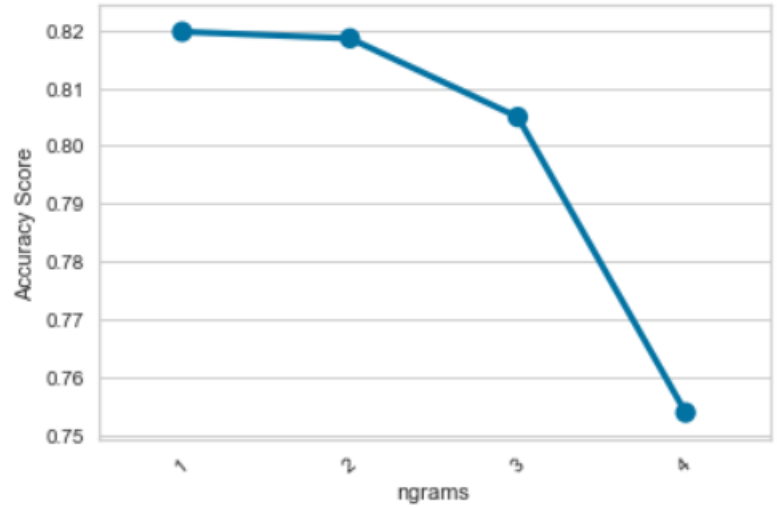
***Εικόνα 53**: Γραφική παράσταση και αποτελέσματα για διαφορετικά max_features του CountVectorizer για: Σετ δεδομένων Vader*

Το καλύτερο σκορ που πετύχαμε είναι για max features=5000. Θα κρατήσουμε την παράμετρο αυτή και θα την εφαρμόσουμε για να βρούμε τον καλύτερο ταξινομητή.

4.7.4.2 Σετ δεδομένων με ανάλυση SentiWordNet

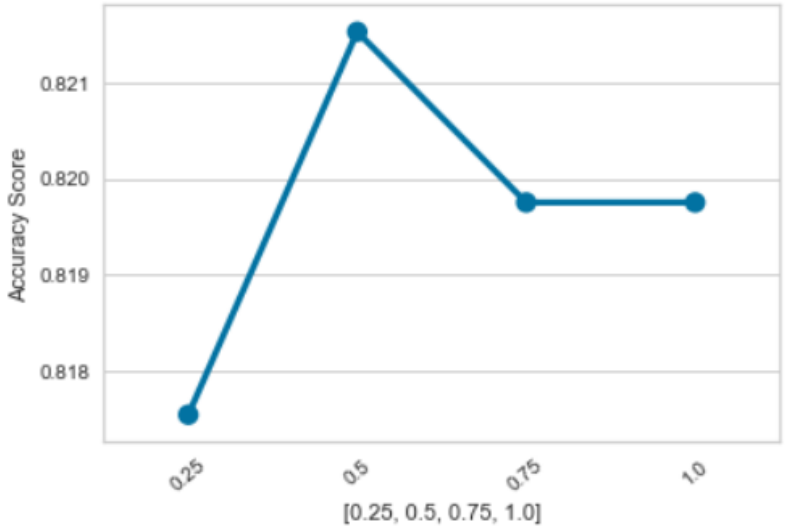
Τα αποτελέσματα για τα διαφορετικά n-grams είναι:

	params	scores
0	1	0.819753
1	2	0.818604
2	3	0.804987
3	4	0.753838



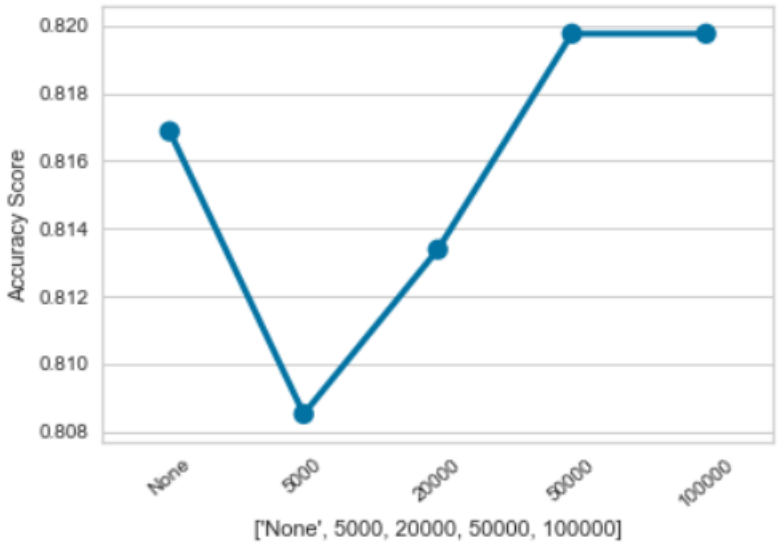
Εικόνα 54: *Γραφική παράσταση και αποτελέσματα για διαφορετικά n-grams του CountVectorizer για: Σετ δεδομένων SentiWordNet*

Τα αποτελέσματα για τα διαφορετικά max df είναι:



Εικόνα 55*: Γραφική παράσταση και αποτελέσματα για διαφορετικά max_df του CountVectorizer για: Σετ δεδομένων SentiWordNet*

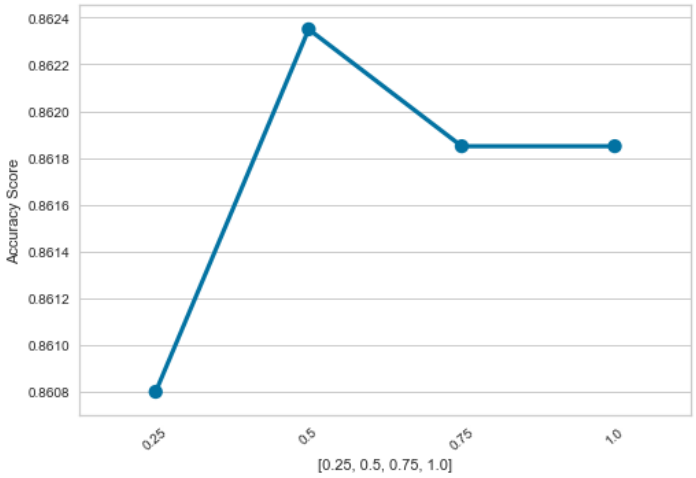
Τα αποτελέσματα για τα διαφορετικά max_features είναι:



***Εικόνα 56**: Γραφική παράσταση και αποτελέσματα για διαφορετικά max_features του CountVectorizer για: Σετ δεδομένων SentiWordNet*

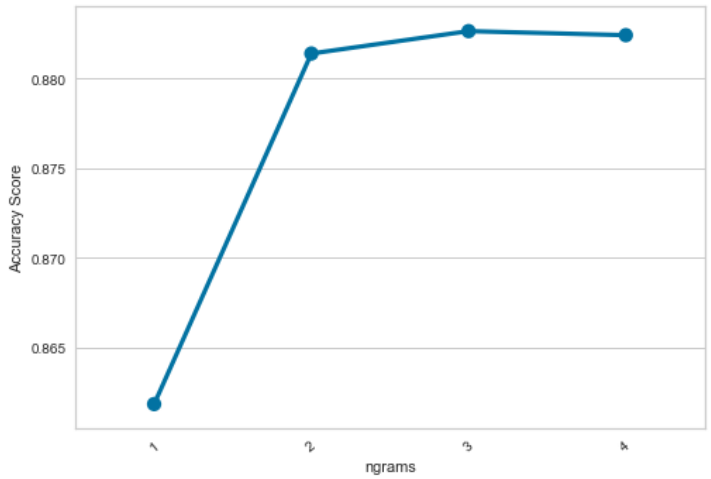
4.7.4.3 Σετ δεδομένων από Imdb

```
Τα αποτελέσματα για τα διαφορετικά max df είναι
   params   scores
0    0.25  0.86080
1    0.50  0.86235
2    0.75  0.86185
3    1.00  0.86185
```



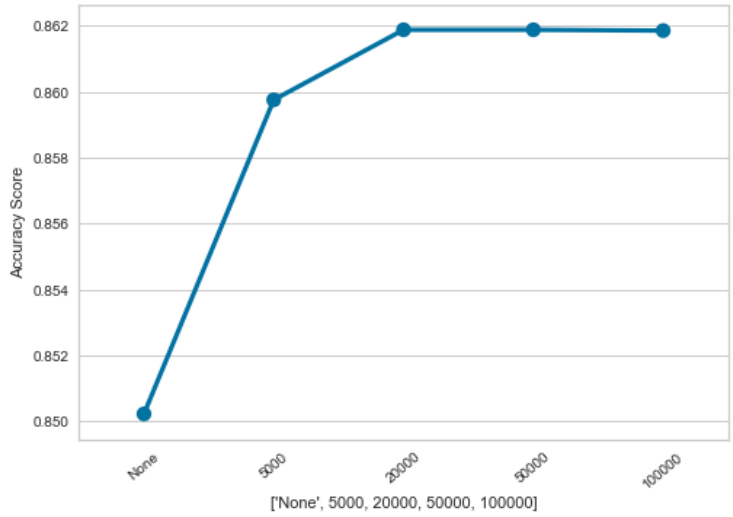
***Εικόνα 57**: Γραφική παράσταση και αποτελέσματα για διαφορετικά max_df του TfidftVectorizer για: Σετ δεδομένων IMDB*

```
Τα αποτελέσματα για τα διαφορετικά ngrams είναι
   params    scores
0       1  0.861850
1       2  0.881400
2       3  0.882650
3       4  0.882425
```



***Εικόνα 58**: Γραφική παράσταση και αποτελέσματα για διαφορετικά n-grams του TfidftVectorizer για: Σετ δεδομένων IMDB*

```
Τα αποτελέσματα για τα διαφορετικά max features είναι
   params    scores
0    None  0.850225
1    5000  0.859750
2   20000  0.861875
3   50000  0.861875
4  100000  0.861850
```



***Εικόνα 59**: Γραφική παράσταση και αποτελέσματα για διαφορετικά max_features του TfidftVectorizer για: Σετ δεδομένων IMDB*

4.7.5 Εφαρμογή Ταξινομητών

Κειμενικά Σετ δεδομένων

Θα εφαρμόσουμε διαφορετικούς ταξινομητές, ώστε να βρούμε τον καλύτερο για τα σετ δεδομένων μας. Παραθέτουμε τα σκορ για τους ταξινομητές που χρησιμοποιήσαμε.

Σετ δεδομένων με ανάλυση Vader

	params	scores
0	Logistisc Regression	0.844276
1	Multinomial Naive Bayes	0.792867
2	Decision Tree	0.717043
3	Random Forest	0.765785
4	Linear SVC	0.834193

***Πίνακας 5**: Σκορ ταξινομητών για σετ δεδομένων Vader*

Συνεπώς για το σετ δεδομένων που προέκυψε από συναισθηματική ανάλυση με Vader θα χρησιμοποιήσουμε **LogisticRegression().**

Σετ δεδομένων με ανάλυση SentiWordNet

	params	scores
0	Logistisc Regression	0.916869
1	Multinomial Naive Bayes	0.818928
2	Decision Tree	0.823894
3	Random Forest	0.835212
4	Linear SVC	0.897637

***Πίνακας 6:** Σκορ ταξινομητών για σετ δεδομένων SentiWordNet*

Συνεπώς για το σετ δεδομένων που προέκυψε από συναισθηματική ανάλυση με Vader θα χρησιμοποιήσουμε **LogisticRegression().**

Σετ δεδομένων από Imdb

	params	scores
0	Logistisc Regression	0.881025
1	Multinomial Naive Bayes	0.884775
2	Decision Tree	0.729150
3	Random Forest	0.759375
4	Linear SVC	0.900250

***Πίνακας 7**: Σκορ ταξινομητών για σετ δεδομένων IMDB*

Συνεπώς για το σετ δεδομένων που προέκυψε από το IMDB θα χρησιμοποιήσουμε **LinearSVC().**

Μουσικό σετ δεδομένων

Για την πρώτη περίπτωση συναισθηματικών κλάσεων

	params	scores
0	Logistisc Regression	0.893554
1	Decision Tree	0.819338
2	Random Forest	0.868186
3	Linear SVC	0.899804
4	K-nn	0.887304

Πίνακας 8: *Σκορ ταξινομητών για μουσικό σετ δεδομένων της πρώτης περίπτωσης συναισθηματικών κλάσεων*

Συνεπώς για το σετ δεδομένων που προέκυψε από την πρώτη περίπτωση συναισθηματικών κλάσεων θα χρησιμοποιήσουμε **LinearSVC()**.

Για την δεύτερη περίπτωση συναισθηματικών κλάσεων

	params	scores
0	Logistisc Regression	0.839608
1	Decision Tree	0.790294
2	Random Forest	0.839191
3	Linear SVC	0.839608
4	K-nn	0.807475

Πίνακας 9*: Σκορ ταξινομητών για μουσικό σετ δεδομένων της δεύτερης περίπτωσης συναισθηματικών κλάσεων*

Συνεπώς για το σετ δεδομένων που προέκυψε από την δεύτερη περίπτωση συναισθηματικών κλάσεων θα χρησιμοποιήσουμε **LinearSVC()**.

Για την Τρίτη περίπτωση συναισθηματικών κλάσεων

	params	scores
0	Logistisc Regression	0.832549
1	Decision Tree	0.820784
2	Random Forest	0.840000
3	Linear SVC	0.832549
4	K-nn	0.838431

Πίνακας 10*: Σκορ ταξινομητών για μουσικό σετ δεδομένων της τρίτης περίπτωσης συναισθηματικών κλάσεων*

Συνεπώς για το σετ δεδομένων που προέκυψε από την δεύτερη περίπτωση συναισθηματικών κλάσεων θα χρησιμοποιήσουμε **RandomForestClassifier()**.

4.7.6 Παραμετροποίηση Ταξινομητών

Τέλος, θα βρούμε τις καλύτερους παραμέτρους για τους ταξινομητές που επιλέξαμε προηγουμένως, χρησιμοποιώντας την GridSearchCV(). Οι διαφορετικές παράμετροι

που χρησιμοποιήσαμε ήταν με βάση την παράμετρο τιμωρίας για κάθε λάθος, τα οποία έγιναν με cross validation με cv=10. Με τον τρόπο αυτό θα δημιουργήσουμε τα καλύτερα δυνατά μοντέλα και στο τέλος θα τα συγκρίνουμε μεταξύ τους, καταλήγοντας σε ένα μοντέλο για τα κειμενικά δεδομένα και ένα μοντέλο για τα μουσικά δεδομένα.

4.7.6.1 Κειμενικά Δεδομένα

Σετ δεδομένων με ανάλυση Vader

```
tuned hpyerparameters :(best parameters)  {'C': 0.1, 'penalty': 'l1'}
accuracy : 0.8605810928013877
```

***Εικόνα 60**: Καλύτεροι Παράμετροι και Σκορ για τον ταξινομητή που προέκυψε από το σετ δεδομένων συναισθηματικής ανάλυσης Vader μετά από GridSearchCV*

Συνεπώς, βρήκαμε τον καλύτερο ταξινομητή που προέκυψε από το σετ δεδομένων από συναισθηματική ανάλυση Vader, τον best_clf_vader.

Σετ δεδομένων με ανάλυση SentiWordNet

```
tuned hpyerparameters :(best parameters)  {'C': 0.1, 'penalty': 'l1'}
accuracy : 0.9290763226366002
```

***Εικόνα 61:** Καλύτεροι Παράμετροι και Σκορ για τον ταξινομητή που προέκυψε από το σετ δεδομένων συναισθηματικής ανάλυσης SentiWordNet μετά από GridSearchCV*

Συνεπώς, βρήκαμε τον καλύτερο ταξινομητή που προέκυψε από το σετ δεδομένων από συναισθηματική ανάλυση SentiWordNet, τον best_clf_sentiwordnet.

Σετ δεδομένων από Imdb

```
Best: 0.902525 using {'C': 10, 'max_iter': 110}
Execution time: 726.641857624054 ms
```

***Εικόνα 62**: Καλύτεροι Παράμετροι και Σκορ για τον ταξινομητή που προέκυψε από το σετ δεδομένων IMDB μετά από GridSearchCV*

Συνεπώς, βρήκαμε τον καλύτερο ταξινομητή που προέκυψε από το σετ imdb, τον best_clf_imdb.

Μουσικό σετ δεδομένων

Για την απλή κλάση

Εφαρμόζοντας GridSearchCV με Cross Validation στα δεδομένα εκπαίδευσης στον ταξινομητή LinearSVC() βρήκαμε ότι οι καλύτεροι παράμετροι είναι:

```
Best: 0.900000 using {'C': 0.5, 'dual': True, 'max_iter': 110}
Execution time: 0.8750011920928955 ms
```

***Εικόνα 63**:Καλύτεροι Παράμετροι και Σκορ για τον ταξινομητή που προέκυψε από το μουσικό σετ δεδομένων της πρώτης περίπτωσης συναισθηματικών κλάσεων μετά από GridSearchCV*

Για την δεύτερη κλάση συναισθημάτων

Εφαρμόζοντας GridSearchCV με Cross Validation στα δεδομένα εκπαίδευσης στον ταξινομητή LinearSVC() βρήκαμε ότι οι καλύτεροι παράμετροι είναι:

```
Best: 0.843750 using {'C': 2.5, 'dual': True, 'max_iter': 110}
Execution time: 1.1395602226257324 ms
```

***Εικόνα 64**: Καλύτεροι Παράμετροι και Σκορ για τον ταξινομητή που προέκυψε από το μουσικό σετ δεδομένων της δεύτερης περίπτωσης συναισθηματικών κλάσεων μετά από GridSearchCV*

Για την τρίτη κλάση συναισθημάτων

Εφαρμόζοντας GridSearchCV με Cross Validation στα δεδομένα εκπαίδευσης στον ταξινομητή RandomForestClassifier() βρήκαμε ότι οι καλύτεροι παράμετροι είναι:

```
Best: 0.862500 using {'criterion': 'gini', 'max_depth': 4, 'max_features': 'auto', 'n_estimators': 500}
Execution time: 224.93200087547302 ms
```

***Εικόνα 65:** Καλύτεροι Παράμετροι και Σκορ για τον ταξινομητή που προέκυψε από το μουσικό σετ δεδομένων της τρίτης περίπτωσης συναισθηματικών κλάσεων μετά από GridSearchCV*

Συνεπώς, δημιουργήσαμε ένα ταξινομητή, με τα καλύτερες παραμέτρους για κάθε υποσετ δεδομένων. Στη συνέχεια θα τα μετρήσουμε την επιτυχία των ταξινομητών αυτών, στα διανύσματα ελέγχου του κάθε υποσέτ.

ΚΕΦΑΛΑΙΟ 5. ΑΠΟΤΕΛΕΣΜΑΤΑ ΚΑΙ ΣΥΜΠΕΡΑΣΜΑΤΑ

Για να συγκρίνουμε τους διαφορετικούς ταξινομητές, θα εφαρμόσουμε κάθε ταξινομητή σε όλα τα σετ δεδομένων. Για κάθε ταξινομητή θα πολλαπλασιάσουμε τα σκορ τους σε κάθε σετ δεδομένων, από τα οποία θα προκύψει ένα σκορ. Ο ταξινομητής με το μεγαλύτερο σκορ, είναι ο καλύτερος.

5.1 Μουσικό σετ δεδομένων

Καθώς για το μουσικό σετ δεδομένων έχουμε κοινά δεδομένα εισόδου με διαφορετικές κλάσεις, θα εφαρμόσουμε σε ένα τυχαίο δείγμα 25% των 3 διαφορετικών σετ μας τους 3 αλγορίθμους. Θα βρούμε τον μέσο όρο των αποτελεσμάτων για το κάθε σετ δεδομένων.

```
Επιτυχία πρώτου ταξινομητη, LinearSVC :
Η επιτυχία του ταξινομητή για το πρώτο σετ δεδομένων είναι 0.8
Η επιτυχία του ταξινομητή για το δεύτερο σετ δεδομένων είναι 0.85
Η επιτυχία του ταξινομητή για το τρίτο σετ δεδομένων είναι 0.825

Βρίσκουμε τον μέσο όρο του ταξινομητή για τα διαφορετικά σετ δεδομένων
Μέσος Όρος πρώτου ταξινομητή: 0.8249999999999998
```

***Εικόνα 66**:Ο πρώτος ταξινομητής που φτιάξαμε για τα διαφορετικά σετ Μουσικών Δεδομένων*

```
Αποτελέσματα 1ου ταξινομητή
                Σετ  scores
0  1ο σετ δεδομένων   0.800
1  2ο σετ δεδομένων   0.850
2  3ο σετ δεδομένων   0.825
```

***Πίνακας 11**: Αποτελέσματα πρώτου ταξινομητή*

```
Επιτυχία δεύτερου ταξινομητη, LinearSVC :
Η επιτυχία του ταξινομητή για το πρώτο σετ δεδομένων είναι 0.825
Η επιτυχία του ταξινομητή για το δεύτερο σετ δεδομένων είναι 0.825
Η επιτυχία του ταξινομητή για το τρίτο σετ δεδομένων είναι 0.875

Βρίσκουμε τον μέσο όρο του ταξινομητή για τα διαφορετικά σετ δεδομένων
Μέσος Όρος δεύτερου ταξινομητή: 0.8416666666666667
```

***Εικόνα 67**:Ο δεύτερος ταξινομητής που φτιάξαμε για τα διαφορετικά σετ Μουσικών Δεδομένων*

```
Αποτελέσματα 2ου ταξινομητή
                Σετ  scores
0  1ο σετ δεδομένων   0.825
1  2ο σετ δεδομένων   0.825
2  3ο σετ δεδομένων   0.875
```

Πίνακας 12:Αποτελέσματα δεύτερου ταξινομητή

```
Σκορ 3ου ταξινομητη, RandomForest :
Η επιτυχία του ταξινομητή στο πρώτο σετ δεδομένων είναι 0.875
Η επιτυχία του ταξινομητή στο δεύτερο σετ δεδομένων είναι 0.95
Η επιτυχία του ταξινομητή στο τρίτο σετ δεδομένων είναι 0.875

Βρίσκουμε τον μέσο όρο του ταξινομητή για τα διαφορετικά σετ δεδομένων
Μέσος όρος τρίτου ταξινομητή: 0.9
```

***Εικόνα 68**:Ο τρίτος ταξινομητής που φτιάξαμε για τα διαφορετικά σετ Μουσικών Δεδομένων*

```
Αποτελέσματα 3ου ταξινομητή
                Σετ  scores
0  1ο σετ δεδομένων   0.875
1  2ο σετ δεδομένων   0.950
2  3ο σετ δεδομένων   0.875
```

***Πίνακας 13:** Αποτελέσματα τρίτου ταξινομητή*

Βλέπουμε πως ο ταξινομητής με βάση τον RandomForestClassifier(), που προέκυψε από την για την τρίτη περίπτωση έχει την μεγαλύτερη επιτυχία σε όλα τα διαφορετικά υποσετ δεδομένων.

5.2 Κειμενικά Σετ δεδομένων

5.2.1 Ταξινομητής από ανάλυση Vader

```
Η επιτυχία του ταξινομητή του Vader για το σετ δεδομένων Vader είναι 0.8638334778837814
Η επιτυχία του ταξινομητή του Vader για το σετ δεδομένων SentiWordNet είναι 0.7676496097137901
Η επιτυχία του ταξινομητή του Vader για το σετ δεδομένων Imdb είναι 0.6371
Πολλαπλασιάζουμε τα σκορ που βρήκαμε
Μέσος Όρος:  0.7561943625325238
Μέσος Όρος μόνο για στίχους:  0.8157415437987858
```

***Εικόνα 69**: Σκορ ταξινομητή που προέκυψε από το σετ δεδομένων συναισθηματικής Ανάλυσης με Vader σε όλα τα σετ κειμενικών δεδομένων*

5.2.2 Ταξινομητής από ανάλυση SentiWordNet

```
Η επιτυχία του ταξινομητή του SentiWordNet για το σετ δεδομένων Vader είναι 0.7393755420641804
Η επιτυχία του ταξινομητή του SentiWordNet για το σετ δεδομένων SentiWordNet είναι 0.9300086730268864
Η επιτυχία του ταξινομητή του SentiWordNet για το σετ δεδομένων Imdb είναι 0.5945
Πολλαπλασιάζουμε τα σκορ που βρήκαμε
Μέσος Όρος:  0.7546280716970223
Μέσος Όρος μόνο για στίχους:  0.8346921075455334
```

***Εικόνα 70**: Σκορ ταξινομητή που προέκυψε από το σετ δεδομένων συναισθηματικής Ανάλυσης με SentiWordNet σε όλα τα σετ κειμενικών δεδομένων*

5.2.3 Ταξινομητής από imdb

```
Η επιτυχία του ταξινομητή του Imdb για το σετ δεδομένων Vader είναι 0.6427580225498699
Η επιτυχία του ταξινομητή του Imdb για το σετ δεδομένων SentiWordNet είναι 0.6480485689505637
Η επιτυχία του ταξινομητή του Imdb για το σετ δεδομένων Imdb είναι 0.908
Πολλαπλασιάζουμε τα σκορ που βρήκαμε
Μέσος Όρος:  0.7329355305001446
Μέσος Όρος μόνο για στίχους:  0.6454032957502168
```

***Εικόνα 71**: Σκορ ταξινομητή που προέκυψε από το σετ δεδομένων IMDB σε όλα τα σετ κειμενικών δεδομένων*

Αξίζει να σημειωθεί, πως τα μοντέλα που προέκυψαν από τις δύο διαφορετικές συναισθηματικές αναλύσεις στο ίδιο σετ, δεν τα πήγαν ιδιαίτερα καλά όταν συνάντησαν τις κριτικές του imdb. Αυτό δείχνει ότι η συναισθηματική πληροφορία που μαζέψαμε από τους στίχους, ενώ είναι επαρκής για νέα στιχουργικά δεδομένα, δεν επαρκεί για κριτικές. Αντίθετα, το μοντέλο που προέκυψε από την ανάλυση κριτικών του imdb τα πήγε επαρκώς καλά στην ανάλυση στίχων.

Όπως βλέπουμε, το μοντέλο που προέκυψε από την συναισθηματική ανάλυση με SentiWordNet χρησιμοποιώντας Logistic Regression, έχει σκορ από 75% σε δεδομένα διαφορετικού τύπου, έως 93% το καλύτερο. Είναι σημαντικό να πούμε πως ο ταξινομητής αυτός έχει και τα καλύτερα αποτελέσματα μόνο για στιχουργικά δεδομένα, με 83%

ΚΕΦΑΛΑΙΟ 6. ΑΡΧΕΙΑ ΥΛΟΠΟΙΗΣΗΣ

Ο κώδικας μας περιλαμβάνει τα αρχεία ImportsDefinitions.py, Vader Lyrics Sentiment Analysis.py, SentiWordNet Lyrics Sentiment Analysis.py, Text Data Clf Comparison.py, Music & Sentiment Analysis.py.

6.1 Φόρτωση Βιβλιοθηκών/Συναρτήσεων

Το αρχείο αυτό περιλαμβάνει τις βιβλιοθήκες και τις συναρτήσεις που θα χρησιμοποιήσουμε.

Όνομα αρχείου: ImportsDefinitions.py

6.2 Μοντέλο Μηχανικής Μάθησης από ανάλυση με Vader

Εδώ περιλαμβάνεται η δημιουργία του μοντέλου ταξινόμησης με βάση στιχουργικά δεδομένα, για τα οποία δημιουργήσαμε συναισθηματικές κλάσεις με το εργαλείο Vader. Στο τέλος του αρχείου αποθηκεύουμε τα σετ εκπαίδευσης και ελέγχου, τον ταξινομητή και το dtm, ώστε να τα φορτώσουμε σε άλλο αρχείο για να γίνει η σύγκριση.

Όνομα αρχείου: Vader Lyrics Sentiment Analysis.py

6.3 Μοντέλο Μηχανικής Μάθησης με δεδομένα από ανάλυση με SentiWordNet

Εδώ περιλαμβάνεται η δημιουργία του μοντέλου ταξινόμησης με βάση στιχουργικά δεδομένα, για τα οποία δημιουργήσαμε συναισθηματικές κλάσεις με το εργαλείο SentiWordNet. Στο τέλος του αρχείου αποθηκεύουμε τα σετ εκπαίδευσης και ελέγχου, τον ταξινομητή και το dtm, ώστε να τα φορτώσουμε σε άλλο αρχείο για να γίνει η σύγκριση.

Όνομα αρχείου SentiWordNet Lyrics Sentiment Analysis.py

6.4 Μοντέλο Μηχανικής Μάθησης με δεδομένα από Imdb

Εδώ περιλαμβάνεται η δημιουργία του μοντέλου ταξινόμησης με βάση κριτικές που πήραμε από το Imdb. Στο τέλος του αρχείου αποθηκεύουμε τα σετ εκπαίδευσης και ελέγχου, τον ταξινομητή και το dtm, ώστε να τα φορτώσουμε σε άλλο αρχείο για να γίνει η σύγκριση.

Όνομα αρχείου: Imdb-Movie-Reviews-Sentiment-Analysis.py

6.5 Σύγκριση Ταξινομητών Κειμενικών Δεδομένων

Το αρχείο αυτό φορτώνει τα διανύσματα εκπαίδευσης που δημιουργήσαμε στα προηγούμενα αρχεία, καθώς και τα μοντέλα της κάθε ανάλυσης. Στην συνέχεια, τρέχουμε κάθε ταξινομητή για κάθε σετ ελέγχου.

Όνομα αρχείου: Text Data Clf Comparison.py

6.6 Μοντέλο Μηχανικής Μάθησης από ανάλυση μουσικών δεδομένων

Το αρχείο αυτό περιλαμβάνει τη δημιουργία 3 υποσετ δεδομένων, για το σετ με μουσικά χαρακτηριστικά, καθώς και τη δημιουργία μοντέλων ταξινόμησης και σύγκρισης τους.

Όνομα αρχείου: Music & Sentiment Analysis.py

REFERENCES

[1]. Perlovsky, L. (2012). Cognitive Function of Music. Part I. Interdisciplinary Science Reviews. 37. 131-144.

[2]. Corrigall, K., A. (2013). Music: The language of emotion. The Library of Congress

[3]. Juslin, P, N & Vastfj, D. (2008). Emotional responses to music: the need to consider underlying mechanisms. The Behavioral and brain sciences, 31(5):559–621

[4]. Besson, M., Faïta, F., Peretz, I., Bonnel, A., & Requin, J. (1998). Singing in the Brain: Independence of Lyrics and Tunes.Psychological Science, 9(6), 494-498.

[5]. Picard, R.W.(2003). Affective Computing. M.I.T Media Laboratory Perceptual Computing Section Technical Report No. 321

[6]. Calvo R., A., & D'Mello S. (2010). Affect Detection: An Interdisciplinary Review of Models, Methods, and Their Applications. *IEEE Transactions on Affective Computing*, vol. 1, no. 1, pp. 18-37.

[7]. Ekman, P. (1999). Basic emotions. In: Handbook of cognition and emotion, pp. 45–60.

[8]. Ortony, A., Clore, G., Collins, A. (1988). The Cognitive Structure of Emotions. Cambridge University Press.

[9]. Pak, A., & Paroubek, P. (2010, May). Twitter as a corpus for sentiment analysis and opinion mining. In LREc (Vol. 10, No. 2010, pp. 1320-1326).

[10]. Agarwal, A., Xie, B., Vovsha, I., Rambow, O., & Passonneau, R. (2011, June). Sentiment analysis of twitter data. In Proceedings of the workshop on languages in social media (pp. 30-38). Association for Computational Linguistics.

[11]. Mohammad, S. M. (2012, June). # Emotional tweets. In Proceedings of the First Joint Conference on Lexical and Computational Semantics-Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation (pp. 246-255). Association for Computational Linguistics.

[12]. Chikersal, P., Poria, S., & Cambria, E. (2015). SeNTU: sentiment analysis of tweets by combining a rule-based classifier with supervised learning. In Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015) (pp. 647-651).

[13]. Hasan, M., Agu, E., & Rundensteiner, E. (2014). Using hashtags as labels for supervised learning of emotions in twitter messages. In ACM SIGKDD Workshop on Health Informatics, New York, USA.

[14]. Spatiotis, N., Mporas, I. & Paraskevas, & Perikos, I. (2016). Sentiment Analysis for the Greek Language.

[15]. Russell, J. (1980). A Circumplex Model of Affect. Journal of Personality and Social Psychology.

[16]. Bibi, M. (2017). Sentiment Analysis at Document Level.

[17]. Khan, J., & Alam, A. (2016). Sentiment Analysis at Sentence Level for Heterogeneous Datasets.

[18]. Sharma, P., & Mishra, N. (2016). Feature level sentiment analysis on movie reviews.

[19]. Perikos, I., & Hatzilygeroudis, I. (2016). Recognizing emotions in text using ensemble of classifiers. Engineering Applications of Artificial Intelligence, 51, 191-201.

[20]. Perikos, I., & Hatzilygeroudis, I. (2013, September). Recognizing emotion presence in natural language sentences. In International conference on engineering applications of neural networks (pp. 30-39). Springer, Berlin, Heidelberg.

[21]. Meyer, L. B. (1956). Emotion and meaning in music. Chicago: University of Chicago Press.

[22]. Sloboda, J. A., & Juslin, P. N. (2001). Psychological perspectives on music and emotion. In P. N. Juslin and J. A. Sloboda (Eds.), Music and Emotion: Theory and Research. New York: Oxford University Press, pp. 71-104.

[23]. Perikos, I., Kovas, K., Grivokostopoulou, F., & Hatzilygeroudis, I. (2017, April). A System for Aspect-based Opinion Mining of Hotel Reviews. In WEBIST (pp. 388-394).

[24]. Perikos, I., & Hatzilygeroudis, I. (2016). A Classifier Ensemble Approach to Detect Emotions Polarity in Social Media. In WEBIST (1) (pp. 363-370).

[25]. Perikos, I., & Hatzilygeroudis, I. (2017, May). Aspect based sentiment analysis in social media with classifier ensembles. In 2017 IEEE/ACIS 16th International Conference on Computer and Information Science (ICIS) (pp. 273-278). IEEE.

[26]. Oudenne, A., M., & Chasins, S., E. (2010). Identifying the Emotional Polarity of Song Lyrics through Natural Language Processing.

[27]. Dang, T., T., & Shirai K., (2009) Machine Learning Approaches for Mood Classification of Songs toward Music Search Engine, *2009 International Conference on Knowledge and Systems Engineering*, Hanoi, pp. 144-149.

[28]. Raschka, S. (2016). MusicMood: Predicting the mood of music from song lyrics using machine learning.

[29]. Cano, E., & Morisio, M. (2017). MoodyLyrics: A Sentiment Annotated Lyrics Dataset. Proceedings of the 2017 International Conference on Intelligent Systems, Metaheuristics & Swarm Intelligence - ISMSI 17.

[30]. Hu, X., & Downie, J.S. (2010). When Lyrics Outperform Audio for Music Mood Classification: A Feature Analysis.*ISMIR*.

[31]. Li, T., & Ogihara, M. (2003). Detecting emotion in music.*ISMIR*.

[32]. Lu, L., Liu, D., & Zhang, H.-J. (2006). Automatic mood detection and tracking of music audio signals. IEEE Transactions on Audio, Speech and Language Processing.

[33]. Skowronek, J., McKinney, M.F., & Par, S.V. (2007). A Demonstrator for Automatic Music Mood Estimation.*ISMIR*.

[34]. Mayer, R., Neumayer, R., & Rauber, A. (2008). Rhyme and Style Features for Musical Genre Classification by Song Lyrics. *ISMIR*.

[35]. Bischoff, K., Firan, C., Paiu, R., Nejdl, W., Laurier, C., & Sordo, M. (2009b). Music mood and theme classification - a hybrid approach. In Proceedings of the 10th International Conference on Music Information Retrieval (ISMIR'09), Kobe, Japan, pp. 657-662.

[36]. Li, T., & Ogihara, M. (2004). Semi-supervised learning from different information sources. Knowledge and Info rmation Systems, 7(3): 289-309.

[37]. Yang, D., & Lee, W. (2004). Disambiguating Music Emotion Using Software Agents. *ISMIR*.

[38]. Hu, X., Downie, J.S., & Ehmann, A.F. (2009). Lyric Text Mining in Music Mood Classification.*ISMIR*.

[39]. Oramas, S., Espinosa-Anke L., Lawlor A.,Serra X., & Saggion H. (2016). Exploring Customer Reviews for Music Genre Classification and Evolutionary Studies. 17th International Society for Music Information Retrieval Conference (ISMIR'16).

[40]. Johnson, M. (2009). How the statistical revolution changes (computational) linguistics. Proceedings of the EACL 2009 Workshop on the Interaction between Linguistics and Computational Linguistics.

[41]. Winograd, T. (1971). Procedures as a Representation for Data in a Computer Program for Understanding Natural Language.

[42]. Schank, R. C. & Abelson, R. P. (1977). Scripts, plans, goals, and understanding: An inquiry into human knowledge structures.

[43]. Goodfellow I., Bengio, Y., & Courville, A. () Deep Learning.

[44]. Webster, J.J., & Kit, C. (1992). Tokenization As The Initial Phase In NLP.

[45]. Halácsy, P. (2006). Benefits of deep NLP-based Lemmatization for Information Retrieval.

[46]. Tian, Y. & Lo, D. (2015). A comparative study on the effectiveness of part-of-speech tagging techniques on bug reports. 2015 IEEE 22nd International Conference on Software Analysis, Evolution, and Reengineering, SANER 2015 - Proceedings. 570-574. .

[47]. Zhang, Y., Jin, R. & Zhou, Z.H. (2010). Understanding bag-of-words model: A statistical framework. International Journal of Machine Learning and Cybernetics. 1. 43-52.

[48]. Luhn, H. P. (1957). A Statistical Approach to Mechanized Encoding and Searching of Literary Information. IBM Journal of Research and Development.

[49]. Ramos, J.E. (2003). Using TF-IDF to Determine Word Relevance in Document Queries.

[50]. Majumder, P., & Mitra, M. (2002). N-gram: a language independent approach to IR and NLP.N-gram: a language independent approach to IR and NLP

[51]. Baccianella, S., Esuli, A.& Sebastiani, F. (2010). SentiWordNet 3.0: An Enhanced Lexical Resource for Sentiment Analysis and Opinion Mining. Proceedings of LREC. 10.

[52]. Hutto, C.J. & Gilbert, Eric. (2015). VADER: A Parsimonious Rule-based Model for Sentiment Analysis of Social Media Text. Proceedings of the 8th International Conference on Weblogs and Social Media, ICWSM 2014

[53]. Kotsiantis, S. (2007). Supervised Machine Learning: A Review of Classification Techniques. Informatica (Ljubljana).

[54]. McCallum, A., & Nigam, K. (1998, July). A comparison of event models for naive bayes text classification. In AAAI-98 workshop on learning for text categorization (Vol. 752, No. 1, pp. 41-48).

[55]. Feng, J., Xu, H., Mannor, S., & Yan, S. (2014). Robust Logistic Regression and Classification.*NIPS*.

[56]. Altman, N.S. (1992). An introduction to kernel and nearest-neighbor nonparametric regression. The American Statistician.46(3): 175–185.

[57]. Coomans D. & Massart, D.L. (1982). Alternative k-nearest neighbour rules in supervised pattern recognition: Part 1. k-Nearest neighbour classification by using alternative voting rules. Analytica Chimica Acta.136: 15–27.

[58]. Quinlan, J. R. (1986). Induction of decision trees. Machine learning, 1(1), 81-106.

[59]. Breiman, L. (2001). "Random Forests".Machine Learning.45(1): 5–32.

[60]. Cortes, C., & Vapnik, V. (1995). Support-vector networks. Machine learning, 20(3), 273-297.

[61]. Guyon, I., Boser, B. & Vapnik, V. (1993). Automatic capacity tuning of very large VC-dimension classifiers. In Advances in neural information processing systems (pp. 147-155).

[62]. Cristianini, N., & Shawe-Taylor, J. (2000). An introduction to support vector machines and other kernel-based learning methods. Cambridge university press.

[63]. Tetko, I. V.; Livingstone, D. J. & Luik, A. I. (1995). Neural network studies. 1. Comparison of Overfitting and Overtraining. Journal of Chemical Information and Modeling.35(5): 826–833.

[64]. Kohavi, R. (1995). A study of cross-validation and bootstrap for accuracy estimation and model selection.Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence.

[65]. Maas, A. L., Daly, R. E., Pham, P. T., Huang, D., Ng, A. Y., & Potts, C. (2011, June). Learning word vectors for sentiment analysis. In Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies-volume 1 (pp. 142-150). Association for Computational Linguistics.

[66]. Eerola, T. (2016). Music and emotion dataset (Primary Musical Cues). Harvard Dataverse, V1