



ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ Π.Ε.Σ.

Διπλωματική εργασία

επιβλέπων καθηγητής: Ευστάθιος Σταματάτος

Καθηγητής

.....
Ανίχνευση Κρίσης στα Κοινωνικά Δίκτυα

Crisis Detection in Social Media
.....

του προπτυχιακού φοιτητή

<13059> Βασίλη Ηλίογλου



Περίληψη

Διάφορες πρόσφατες μελέτες έχουν αναδείξει την σημασία των κοινωνικών μέσων δικτύωσης σε καταστάσεις κρίσεις και σε ανάγκες προσφοράς ανθρωπιστικής βοήθειας. Μεταξύ άλλων, είναι υπαρκτό το πρόβλημα της ροής πληροφοριών για τους αγνοούμενων & διασωθέντες κατά την διάρκεια πανικού από τους εθελοντές & τους διασώστες. Αυτή η διπλωματική μελετά τα tweet που δημοσιεύονται στο Twitter κατά την διάρκεια κρίσεων που μπορούν να χρησιμοποιηθούν από συντονιστικά κέντρα, εθελοντές, διασώστες και άλλους, εφόσον επεξεργαστούν σωστά. Από 8 καταστροφές στην αλλοδαπή γίνεται επεξεργασία περίπου 12.000 κατηγοριοποιημένων tweet και στην συνέχεια εκπαίδευση μοντέλων μηχανικής μάθησης με αυτά. Έπειτα από τις φονικές πυρκαγιές στην Αττική το 2018 γίνεται συλλογή περίπου 15.000 tweet, κατηγοριοποίηση αυτών και εκπαίδευση τριών μοντέλων μηχανικής μάθησης. Παράλληλα γίνεται εκτενείς σχολιασμός των περιπτώσεων και των αποτελεσμάτων σε κάθε στάδιο. Τέλος προσφέρεται ολόκληρος ο κώδικας.

Λέξεις κλειδιά: επεξεργασία φυσικής γλώσσας, μηχανική μάθηση, τεχνητή νοημοσύνη, εξόρυξη δεδομένων, ανάκτηση πληροφορίας, επιστήμη δεδομένων

Abstract

Several recent studies have shown the paramount role importance of social media in crisis situations and offer of humanitarian aid. Among others, the existential issue of the flow of information for the missing and found people amidst the disaster's panic challenges the volunteer & rescue teams. This work studies the tweets posted at Twitter during the onset of a crisis that can be useful for mission control, the volunteers, the rescue teams, and others, as long as they get processed correctly. From 8 foreign disasters we process approximately 12.000 labeled tweets that are then used to train machine learning models. Moving on, approximately 15.000 tweets are collected from the lethal Attica 2018 Wildfires, which are subsequently labeled and ultimately three machine learning models get trained upon them. Ample commentary and reasoning explanation is given in each state and the results as well. Ultimately the complete code is available for use.

Keywords: natural language processing, machine learning, artificial intelligence, data mining, information retrieval, data science

Ευχαριστίες

Θέλω να ευχαριστώ πρώτα απ' όλα τους γονείς μου την υποστήριξη όλα αυτά τα χρόνια στην ολοκλήρωση των σπουδών μου αλλά και για την υπομονή & κατανόηση που επέδειξαν όταν τα πράγματα δυσκόλευαν. Θέλω να ευχαριστώ πολύ τον αδερφό μου που έκανε την πολύ ανιαρή διαδικασία της κατηγοριοποίησης των ελληνικών tweet και τα κατάφερε περίφημα με υπομονή και επιμονή. Τελευταία, όμως όχι λιγότερο σημαντικά από τους προαναφερθέντες, θέλω να ευχαριστώ τον καθηγητή μου κι επιβλέπων καθηγητή της παρούσας διπλωματικής κ. Σταματάτο, ο οποίος μέσω της διδασκαλίας του αποτέλεσε το έναυσμα που μου προξένησε το ενδιαφέρον να ασχοληθώ με την παρούσα διπλωματική και μου έδωσε την ευκαιρία κι για αυτό τον ευχαριστώ ειλικρινά.

Δήλωση γνησιότητας

Δηλώνω υπεύθυνα ότι αυτή η διατριβή είναι το πρωτότυπο έργο μου, που συγγράφηκε ειδικά για την εκπλήρωση των σκοπών και στόχων αυτής της μελέτης και δεν έχει προηγουμένως υποβληθεί σε οποιοδήποτε άλλο Πανεπιστήμιο στην Ελλάδα ή στο εξωτερικό. Δηλώνω επίσης ότι οι πηγές που χρησιμοποιήθηκαν για την εκπόνηση της συγκεκριμένης διατριβής αναφέρονται στο σύνολό τους, δίνοντας πλήρεις αναφορές στους συγγραφείς, συμπεριλαμβανομένων και των πηγών που ενδεχομένως χρησιμοποιήθηκαν από το Διαδίκτυο.

Σάμος, Οκτώβριος 2020

Βασίλης Ηλίουγλου

Υπογραφή

Πίνακας περιεχομένων

Περίληψη.....	2
Abstract	3
Ευχαριστίες.....	4
Λίστα εικόνων.....	8
Λίστα πινάκων	9
Εισαγωγή	10
Δεδομένα.....	10
Εξόρυξη δεδομένων.....	11
Μεγάλα δεδομένα	12
Διαλογή χαρακτηριστικών	13
Ανάκτηση Πληροφορίας	14
Βαρύτητα όρων	15
Τεχνητή νοημοσύνη.....	15
Μηχανική μάθηση	17
Μηχανές διανυσμάτων υποστήριξης	24
Απλοποιημένο θεώρημα Bayes	25
Τυχαία δάση	28
Επεξεργασία φυσικής γλώσσας.....	29
Μελέτη περίπτωσης	31
Συναφή έργα.....	33
Twitter as a Lifeline: Human-annotated Twitter Corpora for NLP of Crisis-related Messages	35
Ανέλεγκτη φωνική δασική πυρκαγιά στο Μάτι & στην Κινέτα Αττικής.....	38
Συμβάν.....	39
Επακόλουθα	41
Υλοποίηση	43
Προδιαγραφές, Framework & βιβλιοθήκες.....	43
Δημιουργία περιβάλλοντος.....	43
Twitter as a Lifeline: Human-annotated Twitter Corpora for NLP of Crisis-related Messages	45
Συλλογή δεδομένων	45
Προεπεξεργασία δεδομένων	46
Εκπαίδευση & σύγκριση μοντέλων ML.....	52
Twitter as a Lifeline for the deadly Attica 2018 Wildfires	57
Συλλογή δεδομένων	58
Προεπεξεργασία δεδομένων	60
Εκπαίδευση & σύγκριση μοντέλων ML.....	63
Συμπεράσματα & Προοπτικές	64

Το preprocessing είναι πάντα καλό	64
Πρέπει να επιλέγεται το σωστό πλήθος των επιλεγμένων feature	64
Πρέπει να επιλέγεται το σωστό πλήθος των fold	65
Έργα που αναφέρονται	67
Παραρτήματα	71
Παράρτημα Α – Terms of use.txt	71
Παράρτημα Β – Ειδησεογραφική βιβλιογραφία για τις πυρκαγιές τις Αττικής 2018	72
Παράρτημα Γ1 – requirements.txt.....	73
Παράρτημα Γ2 – install-packages.py	75
Παράρτημα Δ – twitterCredentials.py	76
Παράρτημα Ε – collectTweets.py.....	77
Παράρτημα ΣΤ – preprocess.py	79
Παράρτημα Ζ – classifiers.py	85
Παράρτημα Η – SVMparameters.py	91
Παράρτημα Θ – NBparameters.py.....	94
Παράρτημα Ι – RFparameters.py	97
Παράρτημα Κ – Φονικός σεισμός της Σάμου 2020.....	100
Συλλογή δεδομένων	100
Προεπεξεργασία δεδομένων	101
Εκπαίδευση & σύγκριση μοντέλων ML.....	101
Συμπεράσματα	102

Λίστα εικόνων

Εικόνα 1: Απεικόνιση δομημένων & αδόμητων δεδομένων	11
Εικόνα 2: Σχέση AI με DS	12
Εικόνα 3: Είδη N-gram αναλόγως με τον αριθμό των επιλεγμένων χαρακτηριστικών	13
Εικόνα 4: Χρονοδιάγραμμα ετήσιας ποσοστιαίας εξέλιξης AI.....	16
Εικόνα 5: Απεικόνιση των σημαντικότερων κατηγοριών ML και μερικών παραδειγμάτων	19
Εικόνα 6: Παραδείγματα συστημάτων μάθησης υπό επιτήρηση.....	20
Εικόνα 7: Εξίσωση F1	21
Εικόνα 8: Απεικόνιση precision & recall	21
Εικόνα 9: Ένα μέσο διάγραμμα ROC-AUC.....	22
Εικόνα 10: Ιδανικό διάγραμμα ROC-AUC = 1.....	23
Εικόνα 11: Χειρότερη περίπτωση, ανίκανο να προβλέψει κατηγορία ROC-AUC = 0.5.....	23
Εικόνα 12: Ιδανική ανάποδη πρόβλεψη ROC-AUC = 0	23
Εικόνα 13: Πιθανοί διαχωριστές σε δείγμα δύο κλάσεων.....	24
Εικόνα 14: Ελεύθερα & περιθωριακά διανύσματα υποστήριξης	25
Εικόνα 15: Το θεώρημα Bayes με L ως κατηγορία(label) και features ως δειγματοστοιχεία	25
Εικόνα 16: Posterior πιθανότητες Bayes	26
Εικόνα 17: Το θεώρημα NB	26
Εικόνα 18: Εξίσωση BNB.....	26
Εικόνα 19: Εξίσωση MNB.....	27
Εικόνα 20: Εξίσωση GNB	27
Εικόνα 21: Παράδειγμα Gaussian NB	27
Εικόνα 22: Εξέλιξη απόδοσης αλγορίθμου κατηγοριοποίησης με επέκτασης διάστασης(dimensionality)	28
Εικόνα 23: Απεικόνιση ενός RF	29
Εικόνα 24: Παραδείγματα part-of-speech tagging	30
Εικόνα 25: Συχνότητα εμφανίσεων των λέξεων-κλειδιών missing & found ανάλογα το ύψος του tweet.....	32
Εικόνα 26: Όγκος retweet των tweets που περιέχουν hashtag #missing και #found.....	33
Εικόνα 27: Όγκος retweet των tweets που περιέχουν hashtag παραπλήσια με #missing και #found.	33
Εικόνα 28: 18 dataset κρίσεων με λεπτομέρειες του τύπου της κρίσης, του ονόματος, της γλώσσας των μηνυμάτων, της χώρας αναφοράς, του αριθμού των tweet.	36
Εικόνα 29: Μάτι Αττικής πριν & μετά την πυρκαγιά(2018).	39
Εικόνα 30: Χάρτης πρόβλεψης κινδύνου πυρκαγιάς για 23/07/2018.	40
Εικόνα 31: Πορεία της πυρκαγιάς στο Μάτι.	42
Εικόνα 32: Διαδικασία λήψης όλων των απαραίτητων module του NLTK.....	45
Εικόνα 33: Στιγμιότυπο εξόδου κατά την εκτέλεση κώδικα preprocess.py.....	50
Εικόνα 34: Απεικόνιση διαδικασίας Cross Validation με 5-fold.....	54
Εικόνα 35: Σχεδιάγραμμα ROC-AUC του 1ου fold στο dataset 2015_Nepal.....	57
Εικόνα 36: Στιγμιότυπο του Project App από το Developer Portal του Twitter	58
Εικόνα 37: Όρια συλλογής tweet ανά μήνα.	60

Λίστα πινάκων

Πίνακας 1: Αντιπαραβάλλοντας τις παραμέτρους εφαρμογών Μηχανικής Γνώσης	21
Πίνακας 2: Δημοφιλή hashtag σε tweet σχετικά με τους διασωθέντες.	31
Πίνακας 3: Δημοφιλή hashtag σε tweet σχετικά με τους αγνοούμενους.....	32
Πίνακας 4: Πιθανές κατηγορίες tweet και η περιγραφή τους.....	36
Πίνακας 5: Στιγμιότυπο ΟΟV dictionary - όροι(αριστερά) μαζί με την διόρθωση(δεξιά) που αντιστοιχούν.....	37
Πίνακας 6: Αποτελέσματα κατηγοριοποίησης με δείκτη την ROC-AUC σε 8 επιλεγμένα dataset με SVM, NB και RF classifiers.	38
Πίνακας 7: Στιγμιότυπο αρχείου 2014_California_Earthquake_CF_labeled_data.csv.....	46
Πίνακας 8: Αριθμός document ανά class ανά dataset.....	49
Πίνακας 9: Παράδειγμα αντικατάστασης συνωνύμου.	50
Πίνακας 10: Παράδειγμα τυχαίου ανακατέματος.....	51
Πίνακας 11: Βηματική εξέλιξη preprocessing ενός document στα αγγλικά.	52
Πίνακας 12: Συγκριτικά σκορ macro-average ROC-AUC για την αγγλική υλοποίηση	53
Πίνακας 13: Τα 15 πιο σημαντικά feature για το dataset των πυρκαγιών 2014 της Καλιφόρνιας(φθίνουσα σειρά).	56
Πίνακας 14: Πολιτική συλλογής ελληνικών tweet για τις πυρκαγιές Αττικής 2018.....	59
Πίνακας 15: Σχολιασμός ευρημάτων ελληνικών tweet κατά την διάρκεια της κατηγοριοποίησης.....	61
Πίνακας 16: Αριθμός document ανά class για το Attica 2018.	61
Πίνακας 17: Βηματική εξέλιξη preprocessing ενός document στα ελληνικά.....	63
Πίνακας 18: Τα 15 πιο σημαντικά feature για τις πυρκαγιές Αττικής 2018 (φθίνουσα σειρά).....	63
Πίνακας 19: Συγκριτικά σκορ macro-average ROC-AUC για την ελληνική υλοποίηση.	63
Πίνακας 20: Απεικόνιση διαφοράς απόδοσης μοντέλων με preprocessing και χωρίς preprocessing.....	64
Πίνακας 21: Διάγραμμα εξέλιξης σκορ μοντέλων όσο επιλέγονται περισσότερα feature	65
Πίνακας 22: Βελτίωση ROC-AUC σκορ καθώς έχουμε περισσότερα split(με κόστος στον χρόνο εκτέλεσης)	66
Πίνακας 23: Μικρή βελτίωση ROC-AUC σκορ σε μεγάλα dataset καθώς έχουμε περισσότερα split(με μεγάλο κόστος στον χρόνο εκτέλεσης)	66
Πίνακας 24: Στιγμιότυπο κατηγοριοποιημένων tweet με την σωστή και την προβλεπόμενη κατηγορία του dataset της Σμύρνης.....	103
Πίνακας 25: Στιγμιότυπο κατηγοριοποιημένων tweet με την σωστή και την προβλεπόμενη κατηγορία του dataset της Σάμου.	104
Πίνακας 26: Αναφορά κατηγοριοποίησης των μοντέλων για το dataset της Σμύρνης.....	105
Πίνακας 27: Αναφορά κατηγοριοποίησης των μοντέλων για το dataset της Σάμου.	106

Εισαγωγή

Η μάθηση είναι μια εγγενής ικανότητα που κατέχουν όλοι οι ζωντανοί οργανισμοί στον πλανήτη μας. Πολλοί επιστήμονες από διαφορετικούς τομείς έχουν προσπαθήσει να ορίσουν το τι εστί η διαδικασία της μάθησης με απλό και κατατοπιστικό τρόπο. Κατά τον H. Roth μάθηση είναι: η βελτίωση ή απόκτηση νέων μορφών συμπεριφοράς και ικανοτήτων. Κατά τον R. Gagne μάθηση είναι: η διαδικασία που υποβοηθάει τους οργανισμούς να τροποποιήσουν τη συμπεριφορά τους σε ένα σχετικό σύντομο χρονικό διάστημα και με ένα μόνιμο τρόπο, ώστε η ίδια τροποποίηση ή αλλαγή να μην επαναλαμβάνεται σε κάθε νέα περίπτωση. Παρατηρούμε αποκλίσεις αναμεταξύ τους γιατί ο ορισμός της μάθησης είναι ζήτημα οπτικής. Ο ψυχολόγος U. Neisser θα την όριζε ως «η επεξεργασία των πληροφοριών» ενώ ο νευρολόγος I. Pavlov ως «δημιουργία υποκατάστατων ανακλαστικών νευρώνων». [1]

Αφαιρετικά μπορεί να οριστεί ως:

Η διαδικασία εμπέδωσης καινούργιων δεδομένων σε υποκείμενα ικανά προς αυτό, η μετατροπή τους σε δεδομένα συμβατά με τις λειτουργίες του υποκειμένου και η ανάκλησή τους. [2]

Ένα παράδειγμα μπορεί να είναι η μάθηση ενός μικρού παιδιού στο πώς να κάνει ποδήλατο. Ή το προς τα πού υπάρχει ήλιος για να μεγαλώσει τα φύλλα του ένα φυτό. Αυτό είναι που μας κάνει να εξελισσόμαστε με την πάροδο του χρόνου.

Έχοντας αναλύσει το θεμέλιο της μάθησης ακολουθούν περισσότερες επιδερμικές αποσαφηνίσεις για βασικούς όρους που θεωρούνται εκ των ων ουκ άνευ για την πορεία αυτής της διπλωματικής.

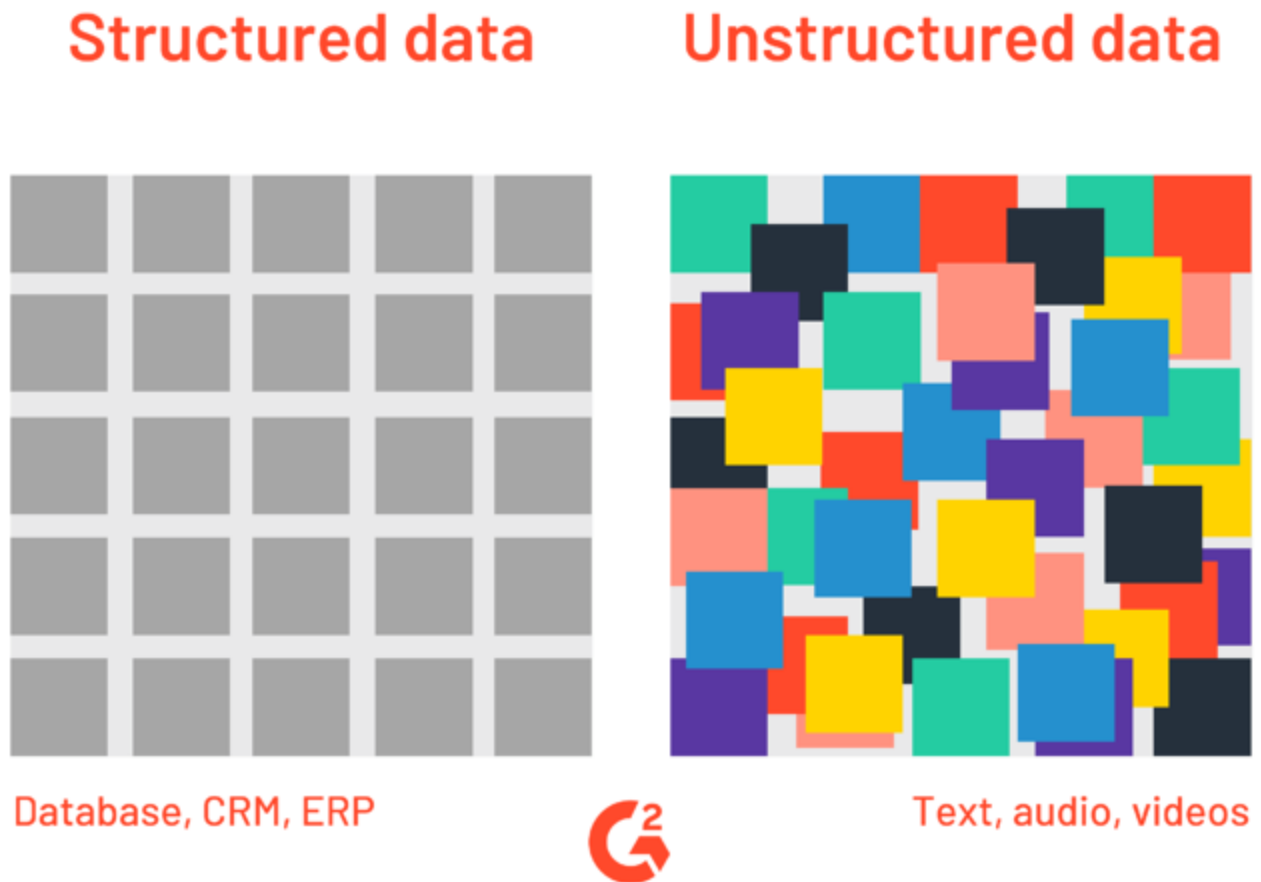
Δεδομένα

Σε γενικές γραμμές, δεδομένα(Data – εφεξής data) μπορούν θεωρηθούν οποιαδήποτε συλλογή από απλά μαθηματικά γεγονότα, όπως αριθμοί, λέξεις, μετρήσεις, παρατηρήσεις, φωτογραφίες & βίντεο, δημοσιεύσεις σε κοινωνικά δίκτυα, κ.ο.κ., τα οποία συλλέγονται & αποθηκεύονται με πολλούς και διαφορετικούς τρόπους. Τα δεδομένα μπορεί να είναι ποιοτικά, δηλαδή να περιγράφουν μια πληροφορία(π.χ. “σήμερα θα έχει καλό καιρό”), είτε να είναι ποσοτικά, δηλαδή να είναι αριθμητικά στοιχεία(π.χ. 3,1415926...).

Όλα τα δεδομένα όμως δεν είναι ίσα. Ο τρόπος συλλογής, επεξεργασίας & ανάλυσης των δεδομένων εξαρτάται από την μορφή που έχουν. Για παράδειγμα τα δεδομένα που έχουν παραχθεί από τα social media διαφέρουν σημαντικά από ένα φύλλο εργασίας excel. Στην επιστήμη των δεδομένων μερικά είναι δομημένα, αλλά τα περισσότερα είναι αδόμητα. [3]

- **Δομημένα(structured):** αναφερόμαστε σε data τα οποία έχουν ξεκάθαρη & σαφώς ορισμένη μορφή, και είναι τα πιο συνήθης να επεξεργαζόμαστε. Ονομάζονται δομημένα επειδή είναι οργανωμένα και κυρίως ποσοτικά σε πρότυπη μορφή κι έτσι μπορούν να αποθηκευτούν και να επεξεργαστούν σχετικά εύκολα από μηχανές, όπως σε βάσεις δεδομένων & υπολογιστικά φύλλα, και να υπολογιστούν τάσεις, μοτίβα & νέα δεδομένα. Μερικά παραδείγματα είναι το τηλέφωνο μιας επαφής στο κινητό μας, η πληκτρολόγηση της φυσικής μας διεύθυνσης σε μια φόρμα παραγγελίας, το SMS που στέλνεται από ένα κινητό, το κείμενο του Tweet που γίνεται στο Twitter, οι γεωγραφικές συντεταγμένες κατά την πλοήγηση κ.ο.κ..
- **Αδόμητα(unstructured):** αναφερόμαστε σε data τα οποία δεν έχουν ξεκάθαρη, σημασιολογικά σταθερή & εύκολη στην κατανόηση για έναν υπολογιστή δομή. Είναι συνήθως ποιοτικά δεδομένα και για αυτό δεν μπορούν να επεξεργαστούν από έναν υπολογιστή με συμβατικά μέσα, λόγου χάρη αθροίσεις. Εκτιμάται ότι περισσότερα από το 80% των δεδομένων που παράγονται σήμερα θεωρούνται αδόμητα, αν και στην πραγματικότητα κανένα από τα αδόμητα δεδομένα δεν είναι στην ουσία αδόμητο, γιατί μπορεί να

διασπαστεί σε τμήματα data τα οποία είναι δομημένα. Αυτό είναι κυρίως που μελετά η επιστήμη των δεδομένων. Μερικά παραδείγματα αυτών είναι κείμενα, βίντεο, εικόνες, ήχος, IP πακέτα, δραστηριότητα κοινωνικών μέσων(π.χ. ποιους ακολουθεί ένας λογαριασμός στο Twitter), κινήσεις από έναν αισθητήρια γυροσκοπίου, κ.ο.κ.. Επειδή τα δεδομένα αυτά δεν έχουν αρραγής δομή δεν μπορούν να αποθηκευτούν σε συμβατικές βάσεις δεδομένων και χρειάζονται επομένως διαφορετικές προσεγγίσεις από ευέλικτες βάσεις όπως η NoSQL και για να επεξεργασθούν σε υψηλό επίπεδο(π.χ. εξόρυξη δεδομένων) απαιτείται συγκερασμός πολλών διαφορετικών επιστημών όπως η στατιστική & η πληροφορική.



Εικόνα 1: Απεικόνιση δομημένων & αδόμητων δεδομένων

(Πηγή: [3])

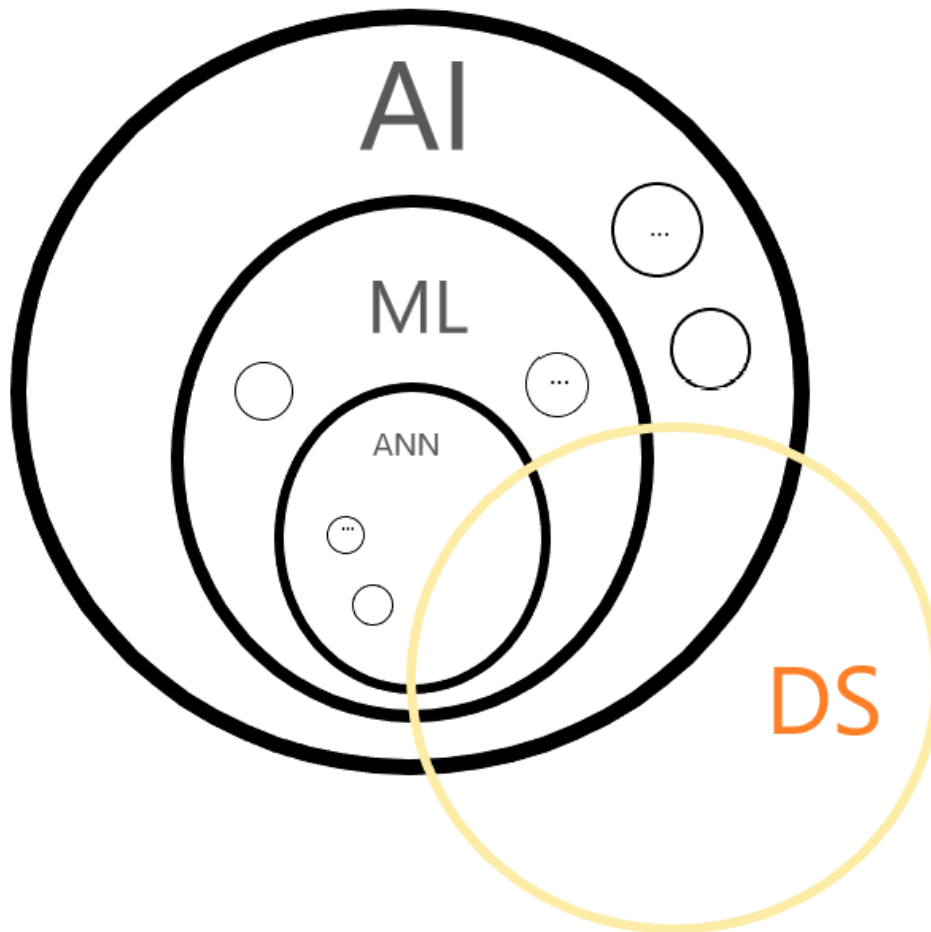
Καταληκτικά, τα δομημένα data είναι καλά οργανωμένα και προτυποποιημένα, ιδανικά για άμεση επεξεργασία από μια μηχανή. Στον αντίποδα, τα αδόμητα data δεν έχουν κάποιο πρότυπο ή οργάνωση, καθιστώντας τα δύσκολα στην συλλογή, επεξεργασία & ανάλυση από μια μηχανή.

Εξόρυξη δεδομένων

Μια από τις πολλές πλευρές που ασχολείται η επιστήμη των δεδομένων είναι η εξόρυξη αυτών. Ως εξόρυξη δεδομένων(Data Mining) ορίζουμε την διαδικασία που ψάχνει για μοτίβα & τάσεις σε μεγάλα σετ δεδομένων. Αυτή τη διαδικασία επιτυγχάνεται με διεπιστημονική προσπάθεια της επιστήμης της Τεχνητής Νοημοσύνης, των Στατιστικών, της Ανάκτησης Πληροφορίας και της Πληροφορικής ή έστω και με απλές ερωτήσεις(queries) σε μια βάση δεδομένων. Ανάλογα του περί τίνος πρόκειται στα δεδομένα, υπάρχουν αντίστοιχες πτυχές εξόρυξης δεδομένων όπως για παράδειγμα αναφορικά με κείμενο τότε η κατηγορία αυτή εξόρυξης

δεδομένων ονομάζεται εξόρυξη κειμένου(Text Mining). Στις δραστηριότητες της εξόρυξης δεδομένων συμπεριλαμβάνονται και η αποτελεσματική διαχείριση των δεδομένων αυτών, η δημιουργία & εκπαίδευση μοντέλων και η κατά δύναμιν μείωση της πολυπλοκότητας αυτών, η πρόβλεψη & διόρθωση τελμάτων, οι μετρήσεις & τα σκορ επιτυχίας των αποτελεσμάτων, η μετέπειτα ανατροφοδότηση αυτών, κ.α.. [4]

Διάφορες τεχνικές έχουν αναπτυχθεί για την εξόρυξη δεδομένων, ιδίως σε αδόμητα δεδομένα, οι οποίες βοηθούν τους αναλυτές & τις επιχειρήσεις να ανακαλύψουν χρήσιμα στοιχεία, όπως τάσεις αγορών, τάσεις αναζητήσεων σε μηχανές αναζήτησης, την πολιτική κλίση, τα ανθρώπινα συναισθήματα, τις εκφράσεις προσώπου(σε συνδυασμό με την Υπολογιστική Όραση), την προγνωστική ανάλυση(π.χ. από διάφορους αισθητήρες με στόχο την έγκαιρη προειδοποίηση) και πολλά ακόμη. [3]



Εικόνα 2: Σχέση AI με DS

Μεγάλα δεδομένα

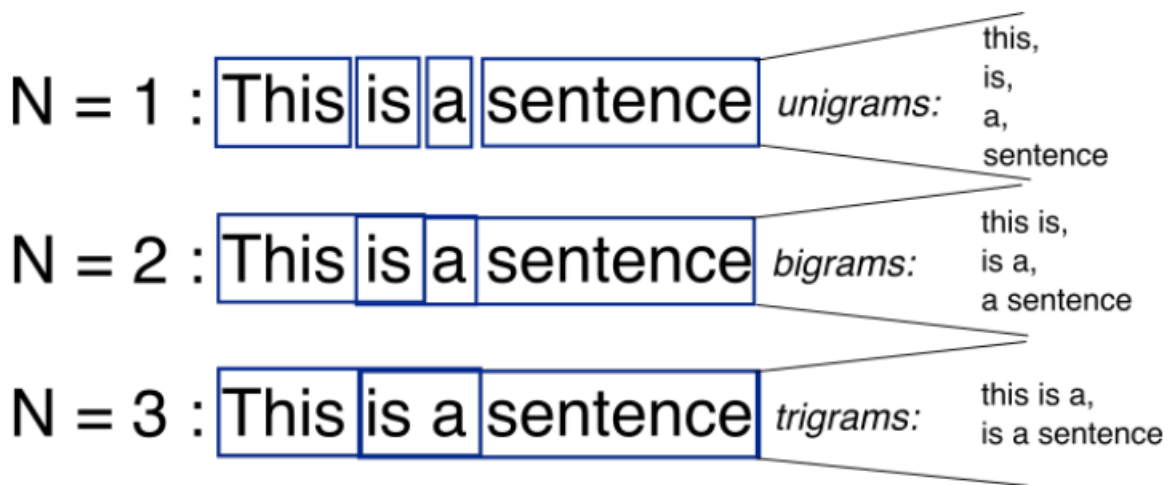
Μεγάλα δεδομένα(Big Data – εφεξής big data) ονομάζονται τα δεδομένα αυτά τα οποία είναι μεγάλα σε όγκο και ετερογενή, συμπεριλαμβάνοντας ταυτόχρονα δομημένα & αδόμητα δεδομένα, τα οποία δεν μπορούν να αποθηκευτούν και να επεξεργαστούν με συμβατικές μεθόδους όπως με συμβατικές βάσεις δεδομένων. Θεωρούνται πάρα πολλά για έναν άνθρωπο να κατανοήσει, δημιουργούνται πολύ γρήγορα για έναν άνθρωπο να προλάβει να τα μελετήσει και μεταβάλλονται πολύ γρήγορα για έναν άνθρωπο να παρακολουθήσει. [5] Οι τρεις ιδιότητές τους είναι:

1. **Όγκος(volume)**: χάρις το IoT & τις προσιτές τιμές μνήμης όλο και περισσότερες συσκευές παράγουν & αποθηκεύουν περισσότερα δεδομένα. Επιπλέον οι άνθρωποι παράγουν περισσότερα δεδομένα από ποτέ όπως φωτογραφίες, βίντεο, δημοσιεύσεις στα μέσα κοινωνικής δικτύωσης, κ.α..
2. **Ταχύτητα(velocity)**: ολοένα και περισσότερες συσκευές IoT παράγουν διαρκώς ολοένα περισσότερα κάθε στιγμή, όπως αισθητήρες, smart συσκευές, κτλ.. Επιπλέον οι άνθρωποι δημιουργούν με μεγαλύτερη ταχύτητα δεδομένα, όπως π.χ. ημερησίως δημιουργούνται περίπου 500 εκατομμύρια tweet, δηλαδή περίπου 200 δισεκατομμύρια tweet τον χρόνο, για την φετινή χρονιά του 2020. [6]
3. **Ετερογένεια(variety)**: όπως είναι λογικό από τα προαναφερθέντα παραδείγματα, τα δεδομένα αυτά έρχονται σε διαφορετικές μορφές, δομημένα & αδόμητα, είτε πρόκειται για κείμενο, είτε για φωτογραφία, κ.ο.κ..

Διαλογή χαρακτηριστικών

Όμως όλοι οι όροι μέσα σε ένα dataset δεν είναι το ίδιο χρήσιμοι. Για αυτό κάνουμε διαλογή χαρακτηριστικών (Feature Selection – εφεξής feature selection), δηλαδή επιλέγουμε τα σημαντικότερα από ολόκληρο το dataset.

Πάντα έχει σημασία η φύση των δεδομένων που θέλουμε να επεξεργαστούμε, το τι θέλουμε να κάνουμε με αυτά και τέλος, το τι αποτελέσματα προσδοκούμε να δημιουργήσουμε ώστε να επιλέξουμε την κατάλληλη μέθοδο. Μια τέτοια μέθοδος είναι γνωστή ως N-grams, δηλαδή να επιλέγονται χαρακτηριστικά σε πλειάδες, ως υποθέσουμε ανά δύο (τότε αποκαλούνται bigrams), αντί για μονάδες (τότε αποκαλούνται unigrams). Μια άλλη διαδικασία είναι να παίρνουμε μόνο τα χαρακτηριστικά που είναι τα πιο σημαντικά. Αυτό μπορεί να γίνει υπολογίζοντας τη βαρύτητα των όρων (που μπορεί να είναι unigrams, bigrams, trigrams, κ.ο.κ.), η οποία αποσαφηνίζεται σε βάθος παρακάτω.



Εικόνα 3: Είδη N-gram αναλόγως με τον αριθμό των επιλεγμένων χαρακτηριστικών

(Πηγή: <http://recognize-speech.com/language-model/n-gram-model/comparison>)

Οποιαδήποτε μέθοδο κι εάν επιλέξουμε θα περιμένουμε βελτιώσεις στην επίδοση, επειδή μειώνεται η διάσταση (dimensionality) των δεδομένων, δηλαδή τα δεδομένα απλοποιούνται αποσιωπώντας τα χαρακτηριστικά που είναι πιο σπάνια ή δεν είναι απαραίτητα/χρήσιμα για το υπάρχον πρόβλημα που επιχειρούμε να επιλύσουμε, κι έτσι υπάρχουν λιγότερες «εκπλήξεις», δηλαδή καταλήγουμε σε μικρότερη εντροπία και μεγαλύτερη ομοιομορφία στα δεδομένα. Τα τρία κυριότερα πλεονεκτήματα της feature selection είναι: [7]

1. **Μειώνει την υπερεκπαίδευση(overfitting):** λιγότερα πλεονάζοντα δεδομένα σημαίνει λιγότερες πιθανότητες να γίνουν αποφάσεις βασισμένες σε άχρηστα δεδομένα(γνωστά ως θόρυβος)
2. **Βελτιώνει την ακρίβεια(accuracy):** λιγότερα παραπλανητικά δεδομένα σημαίνει καλύτερη ακρίβεια αλγορίθμου
3. **Μειώνει τον χρόνο εκπαίδευσης μοντέλων μηχανικής μάθησης(training time):** λιγότερα δεδομένα σημαίνει πως οι αλγόριθμοι εκπαιδεύονται γρηγορότερα. Περισσότερα θα αναπτυχθούν παρακάτω.

Μια από τις σημαντικότερες μεθόδους είναι η *αμοιβαία πληροφορία(Mutual Information ή παραπλήσια αλλιώς Information Gain – εφεξής IG)*¹ η οποία ουσιαστικά μετράει την εντροπία(ή αλλιώς την «έκπληξη») ενός dataset, δηλαδή το πόσο το κάθε χαρακτηριστικό εξαρτάται από ένα άλλο, και κατ' επέκταση το πόσο συνεισφέρει στην επίλυση του προβλήματος, όπως παραδείγματος χάρη σε ένα email το πόσο η κάθε λέξη βοηθά στο να χαρακτηριστεί το μήνυμα σαν spam. Στην προκειμένη περίπτωση μερικά χαρακτηριστικά συνεισφέρουν περισσότερο, όπως για παράδειγμα οι λέξεις "cheap deal", ενώ άλλα χαρακτηριστικά όχι τόσο, όπως για παράδειγμα οι λέξεις "order sent". Για δύο μεταβλητές X, Y υπολογίζεται απλά από τον παρακάτω τύπο:

$$I(X, Y) = H(X) - H(X|Y) \text{ [8]}$$

με I ως αμοιβαία πληροφορία και H ως εντροπία. Η σχέση αυτή αντιπροσωπεύεται από έναν αριθμό ο οποίος είναι ίσος με 0 εάν δεν υπάρχει καμία συσχέτιση, δηλαδή τα δύο χαρακτηριστικά είναι ανεξάρτητα μεταξύ τους, και μεγαλύτερες τιμές εάν τα χαρακτηριστικά εξαρτώνται περισσότερο. Ο στόχος είναι να επιλεγούν τα χαρακτηριστικά με το υψηλότερο IG, αυτά δηλαδή που είναι πιο σημαντικά & συνεισφέρουν περισσότερο. [9]

Ανάκτηση Πληροφορίας

Η έννοια του όρου *ανάκτηση πληροφορίας* μπορεί να είναι πολύ γενική. Απλώς ανοίγοντας ένα λεξικό ώστε να βρούμε τον ορισμό μιας λέξης κάνουμε ανάκτηση πληροφορίας ή βγάζοντας έξω την κάρτα από το πορτοφόλι μας ώστε να μπορέσουμε να πληκτρολογήσουμε τον αριθμό της κάρτας ή αναζητώντας κάτι στον ιστό μέσω μιας μηχανής αναζήτησης. Σε ακαδημαϊκό όμως επίπεδο ορίζουμε *Ανάκτηση Πληροφορίας(Information Retrieval – εφεξής IR)* ως:

Την εύρεση υλικού(συνήθως documents) από μια αδόμητη μορφή(συνήθως κείμενο) που ικανοποιεί την οποιαδήποτε πληροφοριακή ανάγκη μέσα από ένα ευρύτερο μεγάλο σύνολο(συνήθως αποθηκευμένο σε υπολογιστές). [10]

Βασική ορολογία IR στην περίπτωση ενός dataset κειμένου είναι:

- **Corpus:** το σώμα του κειμένου. Για παράδειγμα, σε ένα λεξικό, corpus είναι ολόκληρο το λεξικό.
- **Document:** η οποιαδήποτε μονάδα μέτρησης έχουμε σε ένα σύστημα. Για παράδειγμα, στο κοινωνικό δίκτυο Twitter, document είναι μια εγγραφή, δηλαδή ένα tweet.
- **Term:** η οποιαδήποτε μονάδα μέτρησης ενός document, δηλαδή ο καθένας όρος του. Για παράδειγμα, σε ένα tweet, term είναι συνήθως η καθεμία λέξη ή σημείο στίξης. Καλύτερα να αναφέρονται όμως σαν term γιατί όροι όπως ένα emoji (😄) δεν θεωρούνται λέξεις ή σημεία στίξης.

¹ Ενώ η αμοιβαία πληροφορία(mutual information) και η ωφέλεια πληροφορίας(information gain) υπολογίζουν το ίδιο πράγμα οπότε και θεωρούνται συνώνυμα, σημασιολογικά έχουν μια μικρή διαφορά: το πρώτο απεικονίζει την επίδραση που έχει η επιλογή συσχετισμένων χαρακτηριστικών ενώ το τελευταίο απεικονίζει την αναμεταξύ τους συσχέτιση των μεταβλητών. Πρακτικά όσο περισσότερη συνεισφορά υπάρχει σε δεδομένα(mutual information) τόσο μεγαλύτερη η ωφέλεια της πληροφορίας(information gain). [8]

Στην σύγχρονη εποχή IR συναντάμε παντού, από την αναζήτηση μιας πληροφορίας στο διαδίκτυο μέχρι την εύρεση της διεύθυνσης μιας οδού, η οποία έχει ξεπεράσει την τυπική αναζήτηση σε μια βάση δεδομένων με την χρήση ενός κλειδιού. Ένα καλό παράδειγμα σύγχρονης εφαρμογής IR αποτελεί η αναζήτηση ηλεκτρονικών ταχυδρομικών μηνυμάτων(εφεξής email) μέσα σε φακέλους ή ακόμα και η αναζήτηση μέσα στις λέξεις που περιέχουν ή τα αρχεία που εσωκλείουν και η επιστροφή της βέλτιστης απάντησης. Αυτή είναι μια περίπτωση δυαδικής ανάκτησης όπου είτε έχουμε επιτυχές αποτέλεσμα είτε όχι. Σε έναν λογαριασμό ηλεκτρονικού ταχυδρομείου τα emails θα έχουν ήδη κατηγοριοποιηθεί σε φακέλους, λόγω χάρη σε ανεπιθύμητα και μη, οπότε η IR θα πρέπει να αναζητήσει σε καθέναν από αυτούς. Αμέσως προκύπτουν ζητήματα όπως η διαχείριση μεγάλου όγκου μηνυμάτων και η γρήγορη ανάκτηση της πληροφορίας μέσα σε αυτά με την μικρότερη δυνατή χρήση πόρων, ή την διαχείριση της ανομοιογένειας των συνημμένων αρχείων στην διαδικασία της αναζήτησης & ανάκτησης της πληροφορίας, αλλά όλα αυτά αφορούν τα μοντέλα fat client στα οποία όλη η δουλειά γίνεται στο κομμάτι του client.

Αντιπαραβάλλοντας το προηγούμενο παράδειγμα, μια IR σε αναρτήσεις χρηστών στο Twitter αποτελεί μια εφαρμογή thin client. Σε αυτή την περίπτωση, υπάρχουν πιο ουσιαστικά ζητήματα IR που πρέπει να εξετάσουμε όπως, η σωστή δόμηση του όρου αναζήτησης, η χρήση των κατάλληλων όρων που θα μας αποφέρουν τα συναφή αποτελέσματα και η αναζήτηση με παραμέτρους.

Βαρύτητα όρων

Δοθίσας μιας εντολής ερωτήματος(query – εφεξής query) και ενός document ορίζουμε ως *βαρύτητα όρου*(term weight) ή αλλιώς *συχνότητα όρου*(term frequency) το αριθμητικό δεκαδικό σκορ που αντιπροσωπεύει την πιθανότητα να προκύψει το document d από το query q εντός του διαστήματος $[0, 1]$. [11] Στην περίπτωση που είναι βέβαιο πως σε ένα σύνολο θα προκύψει το d τότε παίρνει τιμές κοντά στο 1. Στον αντίποδα, όταν το d είναι απίθανο να βρεθεί τότε παίρνει τιμές κοντά στο 0. Έτσι οι όροι με το μεγαλύτερο σκορ θεωρούνται πιο σημαντικοί για τον αλγόριθμο και πιο σχετικοί, οπότε και κατατάσσονται υψηλότερα στην λίστα από αυτούς με χαμηλότερο σκορ και λιγότερη σχετικότητα.

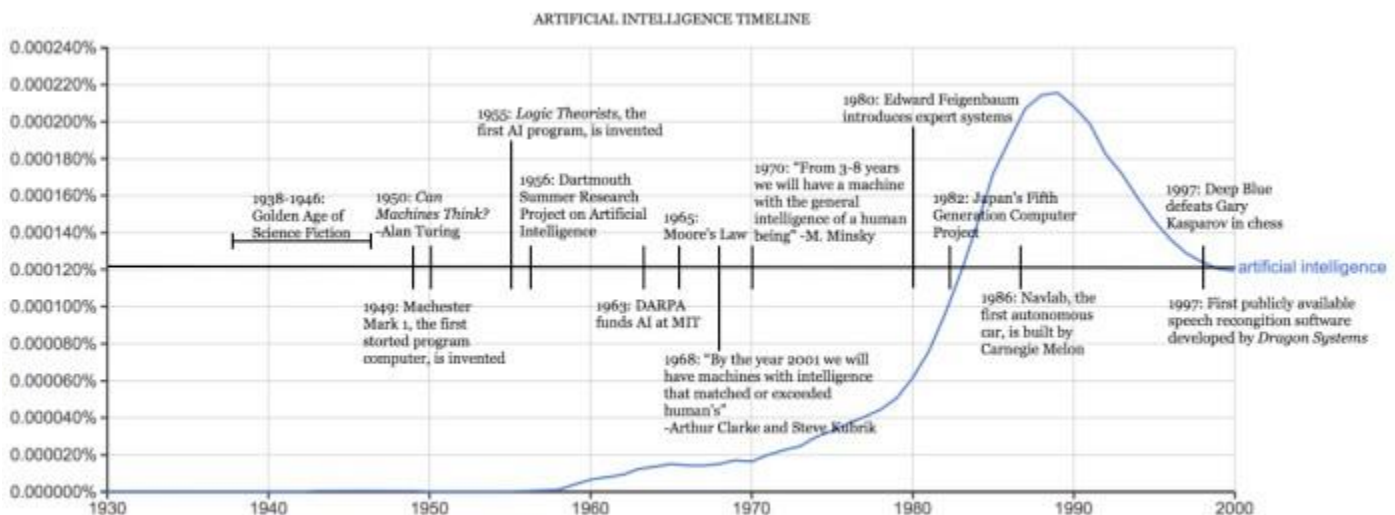
Πολλές μέθοδοι έχουν αναπτυχθεί για τον προσδιορισμό του βάρους των όρων. Μια διαισθητική μέθοδος είναι να προσδιορίζουμε το βάρος με βάσει την συχνότητα εμφάνισης του όρου, δηλαδή όσο πιο συχνά εμφανίζεται ένας όρος μέσα σε ένα corpus τότε τόσο πιο σχετικός είναι από κάποιον άλλον όπου κάνει την εμφάνισή του λιγότερες φορές, γι' αυτό και πρέπει να του δοθεί μεγαλύτερο σκορ από τον τελευταίο. Αυτό μπορεί να επιτευχθεί με την απλή διαδικασία του να προσμετράμε τις φορές που εμφανίζεται ένας όρος μέσα στο corpus, χωρίς να μας ενδιαφέρει η σειρά με την οποία εμφανίζονται οι όροι. Η διαδικασία αυτή είναι γνωστή στον τομέα της επεξεργασίας φυσικής γλώσσας και της ανάκτησης πληροφορίας ως bag of words.

Τεχνητή νοημοσύνη

Ο όρος της *τεχνητής νοημοσύνης*(Artificial Intelligence – εφεξής AI) είναι γνωστός στους περισσότερους από τις χολιγουντιανές ταινίες και τα μυθιστορήματα επιστημονικής φαντασίας όμως ο ορισμός δεν είναι αυτός που απεικονίζεται και σίγουρα δεν είναι τόσο θαυματουργός, τόσο μακιαβελικός, ούτε επίσης τόσο παντοδύναμος. Ο γνωσιακός επιστήμονας John McCarthy όρισε αυτόν τον κλάδο ως την επιστήμη και μεθοδολογία της δημιουργίας νοημόνων μηχανών ικανών στη μάθηση, την κατανόηση από τα συμφραζόμενα, την επίλυση ως προς την ερώτηση και γενικά τον συμπερασμό από την δοθείσα κατάσταση x σε μια y επιδεικνύοντας κάποια ευφυΐα που να προσιδιάζει την ανθρώπινη. [12]

Στις αρχές του περασμένου αιώνα ο κόσμος άρχισε να φαντασιώνεται έναν κόσμο όπου ρομπότ θα μπορούσαν να συμπεριφέρονται όπως οι άνθρωποι, μέσα από την λογοτεχνία και τις πρώτες κινούμενες εικόνες (προάγγελος της 7ης τέχνης). Τότε ήταν που είχε αρχίσει να συλλαμβάνεται σαν ιδέα και είχε ενστερνιστεί από διάφορους επιστημονικούς κλάδους όπως των μαθηματικών, της μηχανολογίας, της φιλοσοφίας, κ.ο.κ.. Αρχικά ήταν ο Alan Turing που προσπάθησε να συλλάβει την έννοια της AI προσεγγίζοντάς την με μαθηματικά μοντέλα, καταλήγοντας σε τέλμα καθώς οι υπολογιστές τότε ήταν μονολιθικοί με πολύ περιορισμένες δυνατότητες και μικρή μνήμη καταστάσεων και φυσικά είδος πολυτελείας.

Μόλις οι υπολογιστικές επιδόσεις βελτιώθηκαν και έγιναν πιο προσιτές υπήρξε διεπιστημονική προσπάθεια και κρατική υποστήριξη για να επιτευχθεί ο ορισμός της AI και να γίνουν άλματα προόδου (αναμεταξύ μερικών απογοητεύσεων) κρατώντας την αισιοδοξία ψηλά. Τα big data έχουν καταστήσει την AI απαραίτητο εργαλείο σε διάφορους κλάδους όπως τον τραπεζικό και των πωλήσεων. Ακόμη και εάν οι αλγόριθμοι σταματήσουν κάποια μέρα να βελτιώνονται, τα όλο και περισσότερα δεδομένα είναι σίγουρο πως θα συνεχίσουν να βελτιώνουν την ποιότητα (δοθέντος ότι είναι αρραγή και χωρίς διενέξεις). [13]



Εικόνα 4: Χρονοδιάγραμμα ετήσιας ποσοστιαίας εξέλιξης AI

(Πηγή: <https://i1.wp.com/sitn.hms.harvard.edu/wp-content/uploads/2017/08/Anyoha-SITN-Figure-2-AI-timeline-2.jpg>)

Στην πραγματικότητα η AI είναι δύο διαφορετικές ιδέες. Σχεδόν όλα τα γνώριμα επιτεύγματα AI είναι τεχνητή περιορισμένη νοημοσύνη (Artificial Narrow Intelligence – εφεξής ANI), δηλαδή είναι AI που είναι κατασκευασμένα να κάνουν ένα συγκεκριμένο πράγμα, όπως είναι ένα έξυπνο ηχείο ή ένας ρομποτικός βραχίονας. Στον αντίποδα είναι η τεχνητή γενική νοημοσύνη (Artificial General Intelligence – εφεξής AGI), είναι δηλαδή AI που μπορούν (θεωρητικά) να κάνουν οτιδήποτε μπορεί κι ένας άνθρωπος, όπως για παράδειγμα η προσπάθεια που γίνεται με το ανδροειδές Atlas της ερευνητικής εταιρείας Boston Dynamics ή ακόμα και να ξεπερνούν σε δυνατότητες τις εγγενείς ανθρώπινες όπως είναι για παράδειγμα το Skynet στην ταινία Terminator. Φυσικά είμαστε πολύ μακριά ακόμα από αυτό αν και αυτός είναι ο μακροπρόθεσμος στόχος του AI (προφανώς στην καλή του μορφή). Δυστυχώς, επειδή τα σύγχρονα επιτεύγματα στο AI είναι συνήθως ANI, έχουν προκαλέσει στην κοινωνία την ψευδαίσθηση πως το AI είναι αχαλίνωτο σε μορφή AGI προξενώντας παράλογους συμπερασμούς και φόβο για κακά ρομπότ που σύντομα θα ξεπεράσουν και εξουσιάζουν την ανθρωπότητα. [14]

Στην καθημερινότητα ήδη συναντούμε ολοένα και περισσότερες εφαρμογές AI(ANI) όπως τα αυτοκινούμενα αυτοκίνητα, την άμεση μετάφραση μεταξύ δύο διαφορετικών γλωσσών, κ.α. και είμαστε βέβαιοι πως το μέλλον θα συναρπάξει. Καθώς γινόμαστε μάρτυρες αυτής της ραγδαίας ανάπτυξης συναντούμε και ηθικούς προβληματισμούς που αποτελούν αντίσταση στην εξέλιξη και την επιβραδύνουν, όπως την αντικατάσταση χειρωνακτικών εργασιών όπως το χτίσιμο κτηρίων από μηχανές ή η χρήση της AI στην στρατιωτική βιομηχανία το 2018 όπου είδε την διαμαρτυρία και την έξοδο υπαλλήλων της Microsoft [15]. Σε κάθε κρίσιμο στάδιο πρέπει να υπάρχει σοβαρή συζήτηση για την κατεύθυνση των ερευνών και τον αντίκτυπό τους όπως επίσης και τους ηθικούς ενδοιασμούς.

Το AI όμως έχει περιορισμούς & προβλήματα που ακόμη και σήμερα είναι δύσκολοι να ξεπεραστούν. [16] Μερικά παραδείγματα αυτών είναι:

- Περιορισμοί επίδοσης: ακόμα με τα σύγχρονα άλματα στον τομέα της επιστήμης των υπολογιστών & της πληροφορίας, αρκετοί αλγόριθμοι παρουσιάζουν προβλήματα επίδοσης, ειδικά με μεγάλα σετ δεδομένων ή πολύπλοκους αλγορίθμους. Οι υπολογιστικοί πόροι άλλωστε είναι πάντα περιορισμένοι.
- Δύσκολη επεξήγηση: το να εξηγήσεις σε έναν άνθρωπο τη λογική σου πίσω από μια απόφαση είναι εύκολο. Δεν ισχύει όμως το ίδιο για πολλούς αλγορίθμους AI καθώς είναι δύσκολο να αποφανθεί κανείς από πού ξεκίνησε η λογική και πώς κατέληξε σε ένα συγκεκριμένο συμπέρασμα, ειδικά λόγου χάρη σε πολύπλοκα νευρωνικά δίκτυα με εκατομμύρια νευρώνες. [17]
- Επιμονή(ή αλλιώς προκατάληψη): τα προκατειλημμένα δεδομένα, δηλαδή δεδομένα που επιμένουν σε έναν συγκεκριμένο συμπέρασμα χωρίς λογική επεξήγηση, δημιουργούν προκατειλημμένους αλγορίθμους AI οι οποίοι με την σειρά τους παράγουν περισσότερα προκατειλημμένα δεδομένα. Για παράδειγμα ένα AI που έχει μάθει από dataset πως όλοι οι γιατροί είναι άντρες τότε θα παράγει και αντίστοιχα αποτελέσματα όταν ερωτηθεί εάν το ιατρικό επάγγελμα είναι ανδρικό.
- Επιρρεπής σε επιθέσεις: σημαντικά συστήματα είναι ανοικτά σε κακόβουλες επιθέσεις τα οποία προσπαθούν να ξεγελάσουν και κατά την κατασκευή αυτών πρέπει να υπάρχει μέριμνα για αυτές τις επιθέσεις ώστε να υπάρχουν αναχώματα ασφαλείας. Ένα παράδειγμα είναι τα αυτόματα σχόλια στα μέσα κοινωνικής δικτύωσης τα οποία αποστέλλονται από ρομπότ και περνούν από τα συστήματα δημιουργώντας την ψευδαίσθηση ότι υπάρχει πραγματικός διάλογος.
- Κακόβουλες χρήσεις AI: μπορεί η AI να χρησιμοποιείται για καλό σκοπό όμως υπάρχουν και λωποδύτες οι οποίοι την χρησιμοποιούν για μοχθηρούς σκοπούς. Μερικά παραδείγματα αυτών είναι τα Deep Fakes δηλαδή παρασκευασμένα βίντεο & ήχους από ανθρώπους που λένε ή κάνουν πράγματα τα οποία δεν συνέβησαν ποτέ. Τα καταπιεστικά καθεστώτα, όπως της Λαϊκής Δημοκρατίας της Κίνας που χρησιμοποιεί την υπολογιστική όραση για την παρακολούθηση & κατεύθυνση των πολιτών ανάλογα με τις κυβερνητικές επιθυμίες. Την παραβίαση ιδιωτικότητας, όπως το σκάνδαλο της Cambridge Analytica με τις αμερικάνικες εκλογές του 2016, όπου επιστρατεύτηκε η AI για πολυεπίπεδη παρακολούθηση των χρηστών με σκοπό την ανακάλυψη των αδυναμιών τους και μετέπειτα την στοχευμένη διαφήμιση παραπλανητικών διαφημίσεων & την ανάλογη έκθεσή τους σε σχόλια ρομπότ. Τα συστήματα anti-spam, τα οποία μάχονται τα συστήματα spam προσπαθώντας να αλλοιώσουν την γνωσιακή τους βάση και να παραβιάσουν τα φίλτρα τους. Και πολλά ακόμη...

Μηχανική μάθηση

Ως Μηχανική Μάθηση(Machine Learning – εφεξής ML) ορίζουμε τον κλάδο της AI που ασχολείται αποκλειστικά και μόνο στη δημιουργία αλγορίθμων & προσαρμοστικών μοντέλων που μαθαίνουν από την εμπειρία, δηλαδή από τα ιστορικά δεδομένα. Βασίζεται πάνω στην γραμμική άλγεβρα, την στατιστική, τις πιθανότητες

και τον ευρύτερο κλάδο των μαθηματικών συνδυάζοντάς τα με την εφαρμοσμένη πληροφορική και την ανάλυση δεδομένων. [18] Η διεπιστημονική κατασκευή αυτών των αλγορίθμων, δηλαδή αλγορίθμων ικανών να αποφανθούν από μια κατάσταση $x \xrightarrow{\text{διαδικασία } h:} y$, είναι που επιτρέπει την κατασκευή εφαρμογών AI. Σύμφωνα με τον καθηγητή Tom Mitchell:

Ένα πρόγραμμα θεωρείται πως μαθαίνει από την εμπειρία E σχετικά με διαδικασία T και με επίδοση P εάν η επίδοσή του στην διαδικασία T με τον τρόπο που έχει μετρηθεί ως P βελτιώνεται από την εμπειρία E. [19]

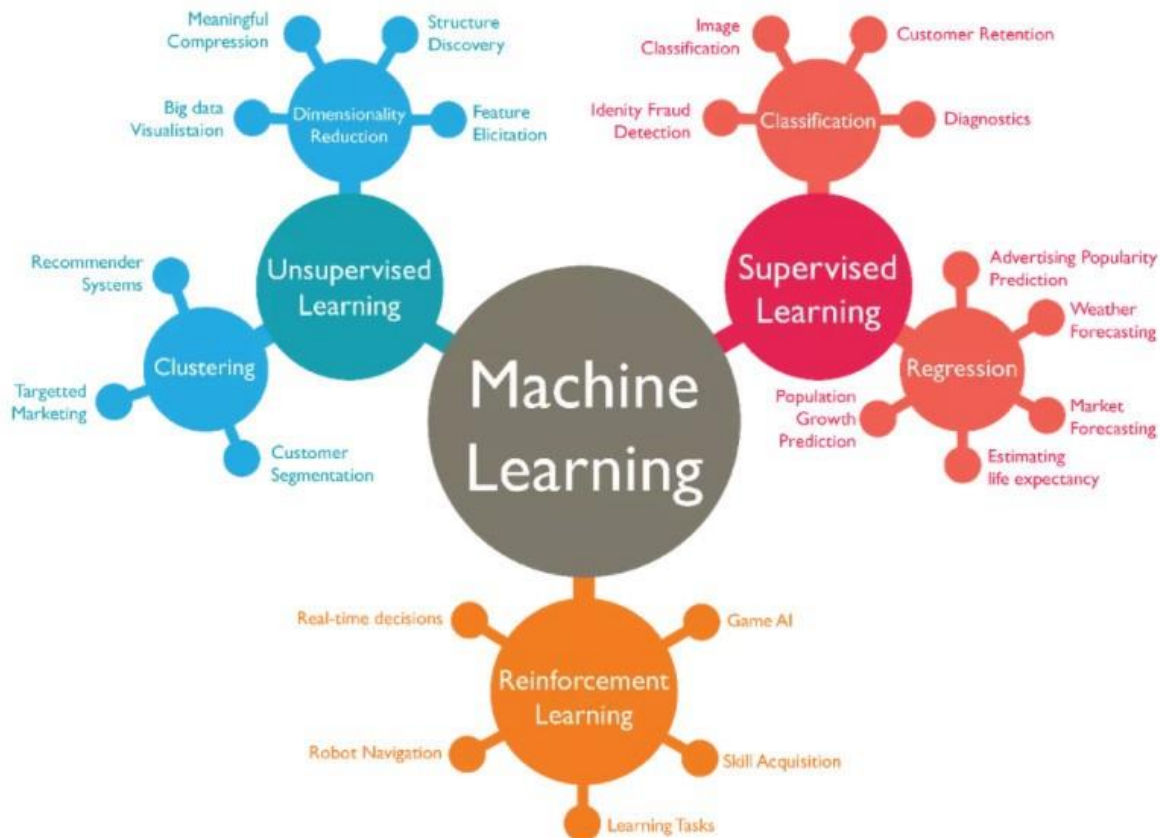
Στην ML ισχύει η ρήση *garbage in-garbage out(GIGO)*. Όσα περισσότερα «καλά» δεδομένα, τόσο περισσότερα βελτιώνονται οι επιδόσεις, οι αλγόριθμοι γίνονται εξυπνότεροι και μειώνεται η ασάφεια & η λεγόμενη «έκπληξη» από νέα δεδομένα. Όταν αναφερόμαστε σε «καλά» δεδομένα εννοούμε δεδομένα που συνεισφέρουν θετικά, δηλαδή είναι συγκεκριμένα και αληθή. Στον αντίποδα, «κακά» δεδομένα είναι όσα δεν συνεισφέρουν θετικά στο μοντέλο, όπως για παράδειγμα υπερβολές, ή δεδομένα που είναι ασαφή, ή δεδομένα που είναι άσχετα με τον στόχο, ή υπάρχουν ακούσιες ανωμαλίες στα δεδομένα που συνήθως προκύπτουν από αστοχία. Υπάρχουν περιπτώσεις όπου γίνονται κακόβουλες επιθέσεις σε αλγορίθμους ML εισάγοντας δεδομένα-σκουπίδια τα οποία αποτελούν θόρυβο στις αποφάσεις και αποσυντονίζουν τους αλγορίθμους από τις σωστές προβλέψεις.

Τα περισσότερα δεδομένα είναι ισχυρότερα από τα καλύτερα μοντέλα. Τα καλύτερα δεδομένα είναι ισχυρότερα από τα περισσότερα δεδομένα. — Riley Newman

Υπάρχουν διαφορετικές κατηγορίες ML, οι οποίες συνήθως επεκτείνουν η μια την άλλη. [20] Οι κυριότερες είναι:

1. **Μάθηση υπό επίβλεψη(supervised learning)**: περιγράφει προβλήματα τα οποία χρησιμοποιούν μοντέλα που μαθαίνουν από ήδη απαντημένα παραδείγματα, δηλαδή είναι γνωστά τα αποτελέσματα y των ερωτηθεισών καταστάσεων x με βάσει τον αλγόριθμο/σύστημα h . Τα μοντέλα αυτά μαθαίνουν από το σύνολο εκπαίδευσης(training set) την είσοδο x στον αλγόριθμο/σύστημα h και την σωστή έξοδο y , διορθώνοντας διαρκώς τον αλγόριθμο στις προβλέψεις όταν πέφτουν έξω(σ.σ. εξ ου και αποκαλούνται *υπό επίβλεψη*), και στη συνέχεια εφαρμόζονται σε προβλήματα όπου ζητούνται να προβλέψουν τις εξόδους σε νέες εισόδους. Τα προβλήματα αυτά χωρίζονται κυρίως σε δύο κατηγορίες. Σε αυτά της στατιστικής κατηγοριοποίησης(classification), δηλαδή της πρόβλεψης μιας κατηγορίας που ανήκει μια είσοδος(π.χ. πρόβλεψη εάν email είναι spam), και σε αυτά της παλινδρόμησης(regression), δηλαδή της πρόβλεψης μιας αριθμητικής τιμής(π.χ. πρόβλεψη τιμών οικοπέδου). Δημοφιλής αλγόριθμοι μάθησης υπό επίβλεψη είναι μεταξύ άλλων οι μηχανές διανυσμάτων υποστήριξης(SVM), τα τυχαία δάση(RF) και το απλοποιημένο θεώρημα Bayes(NB) τα οποία θα αναφερθούν παρακάτω.
2. **Μάθηση άνευ επίβλεψης(unsupervised learning)**: περιγράφει προβλήματα τα οποία χρησιμοποιούν μοντέλα που περιγράφουν μοτίβα ή προσδιορίζουν τις σχέσεις μεταξύ δεδομένων από τις ιδιότητές τους. Σε αντίθεση με τη μάθηση υπό επίβλεψη, δεν υπάρχει κάποιος «διδασκτής» για να κατευθύνει & να διορθώνει τον αλγόριθμο(εξ ου και αποκαλούνται *άνευ επίβλεψης*) και οι σωστές σχέσεις πρέπει να ανακαλυφθούν εξ ολοκλήρου από τον αλγόριθμο. Τα προβλήματα αυτά χωρίζονται κυρίως σε δύο κατηγορίες. Σε αυτά της ομαδοποίησης(clustering), δηλαδή της κατάταξης των δεδομένων σε πλειάδες(π.χ. πρόβλεψη ποια προϊόντα αγοράζονται μαζί), και σε αυτά της εκτίμησης της πυκνότητας(density estimation), δηλαδή της εύρεσης της κατανομής των δεδομένων σε κάποιον χώρο αποσκοπώντας να προβλέψει μελλοντικές τιμές(π.χ. ποια θα είναι η θερμοκρασία σε 12 μήνες). Δημοφιλής αλγόριθμοι μάθησης άνευ επίβλεψης είναι μεταξύ άλλων οι k-Μέσο(k-Means) και Kernel Density Estimation.

3. **Ενισχυτική μάθηση(reinforcement learning):** περιγράφει προβλήματα τα οποία χρησιμοποιούν μοντέλα όπου μαθαίνουν με βάση την ανατροφοδότηση που τους γίνεται σε κάθε απόφαση που λαμβάνουν και στόχος τους είναι να μεγιστοποιήσουν το σκορ. Τα μοντέλα πρέπει να ανακαλύψουν μόνα τους τις σωστές αποφάσεις που πρέπει να πάρουν καθώς σε κάθε θετική ανατροφοδότηση επιβραβεύονται αυξάνοντας το σκορ τους ενώ σε κάθε αρνητική το σκορ μειώνεται(π.χ. η εκμάθηση των αυτοκινούμενων αυτοκινήτων στο να παραμένουν στην λωρίδα του δρόμου). Δημοφιλείς αλγόριθμοι ενισχυτικής μάθησης είναι μεταξύ άλλων οι Q-Learning και Temporal-Difference Learning.



Εικόνα 5: Απεικόνιση των σημαντικότερων κατηγοριών ML και μερικών παραδειγμάτων

(Πηγή: https://miro.medium.com/max/875/0*BanMS95RscTacoTA)

Υπάρχουν πολλές διαφορετικές εφαρμογές & προσεγγίσεις μηχανικής μάθησης ανάλογα με το πρόβλημα που καλούμαστε να επιλύσουμε. [19] Μεταξύ των πιο δημοφιλών εφαρμογών είναι:

- **Βιοπληροφορική(Bioinformatics):** χρησιμοποιούνται αλγόριθμοι ML οι οποίοι ενισχύουν τις ιατρικές προγνώσεις αναζητώντας μοτίβα για να ταυτοποιήσουν τις υπάρχουσες ασθένειες(π.χ. εντοπισμός καρκινικών κυττάρων πάνω σε γονίδια μέσω μιας ακτινογραφίας) ή ακόμα και στην βοήθεια ανακάλυψης νέων φαρμάκων όπως λόγω χάρη γίνεται ήδη για την παρασκευή του εμβολίου κατά του νέου κορονοϊού. [21]
- **Υπολογιστική όραση(Computer Vision):** χρησιμοποιούνται αλγόριθμοι ML & νευρωνικών δικτύων για τον εντοπισμό χαρακτηριστικών προσώπου, την περιγραφή εικόνων, τον εντοπισμό προσώπων, την ταυτοποίηση & αυθεντικοποίηση ατόμων, την οδήγηση αυτοκινούμενων αυτοκινήτων, κ.α..
- **Επεξεργασία φυσικής γλώσσας(Natural Language Processing):** χρησιμοποιούνται αλγόριθμοι ML με κύριο στόχο την αναγνώριση της φυσικής γλώσσας. Σε μεγαλύτερο βαθμό αναπτύσσεται παρακάτω.

- **Μηχανών αναζήτησης ιστού(Web search engines):** χρησιμοποιούνται αλγόριθμοι ML και IR για να βελτιωθούν τα αποτελέσματα αναζητήσεων, να προβλεφθεί καλύτερα το τι ψάχνει ο χρήστης, να διορθωθούν τυχόν σφάλματα στον όρο αναζήτησης και να κατανοηθούν καλύτερα τα προαναφερθέντα queries αναλύοντάς τη συμπεριφορά των χρηστών κατά την αναζήτηση αυτών.
- **Χρηματοπιστωτική ανάλυση(Fintech analysis):** χρησιμοποιούνται αλγόριθμοι ML και νευρωνικά δίκτυα για την πρόβλεψη χρηματιστηριακών τάσεων, μελλοντικών χρηματιστηριακών τιμών των μετοχών και αυτοματοποίηση αγοραπωλησιών και μοχλεύσεων κεφαλαίων, τα οποία εργαλεία δεν λείπουν πλέον από την φαρέτρα των χρηματιστών.

Supervised Learning Systems

As two pioneers in the field, Tom Mitchell and Michael I. Jordan, have noted, most of the recent progress in machine learning involves mapping from a set of inputs to a set of outputs. Some examples:

INPUT X	OUTPUT Y	APPLICATION
Voice recording	Transcript	Speech recognition
Historical market data	Future market data	Trading bots
Photograph	Caption	Image tagging
Drug chemical properties	Treatment efficacy	Pharma R&D
Store transaction details	Is the transaction fraudulent?	Fraud detection
Recipe ingredients	Customer reviews	Food recommendations
Purchase histories	Future purchase behavior	Customer retention
Car locations and speed	Traffic flow	Traffic lights
Faces	Names	Face recognition

Εικόνα 6: Παραδείγματα συστημάτων μάθησης υπό επιτήρηση

(Πηγή: https://miro.medium.com/max/855/0*4Dr8NtbZ6XwQ7N07)

Όμως η ML δεν είναι πανάκεια σε όλα τα προβλήματα. Υπάρχουν περιορισμοί στο τι μπορεί να επιτύχει. Ο διακεκριμένος επιστήμονας & επίτιμος καθηγητής AI Andrew Ng παρουσίασε έναν ατελή απλό τρόπο, αναφορικά με την μάθηση υπό επίβλεψη, ο οποίος είναι να σκεφτεί κανείς τι μπορεί να επιτύχει ένας άνθρωπος μέσα σε ένα περίπου δευτερόλεπτο. Για παράδειγμα, μέσα σε ένα δευτερόλεπτο ένας άνθρωπος μπορεί να εντοπίσει την θέση άλλων αυτοκινήτων στον αυτοκινητόδρομο, ή να αποφανθεί εάν μια κούπα είναι σπασμένη. [22] Λίγα δευτερόλεπτα χρειάζεται ένας άνθρωπος επίσης για να καταλήξει εάν το tweet που μόλις διάβασε ανήκει σε μια συγκεκριμένη κατηγορία. Στον αντίποδα, κάτι που η ML δεν μπορεί να κάνει είναι να αναλύσει τις μακροοικονομικές συνθήκες ενός κλάδου και να συντάξει μια 200 σελίδων αναφορά – ένας άνθρωπος σίγουρα δεν μπορεί να το κάνει αυτό μέσα σε λίγα δευτερόλεπτα. Ή όταν δεν υπάρχουν αρκετά δεδομένα επί του θέματος. Οι μηχανές μπορούν να προβλέψουν το μέλλον, όσο το μέλλον δεν μοιάζει πολύ διαφορετικό από το παρόν. [23] Οπότε λοιπόν μπορούμε να συνοψίσουμε την παραπάνω λογική σε δύο καταστάσεις, σε αυτές όπου μπορούν να λυθούν με ML και σε αυτές που δεν μπορούν να λυθούν με ML. Τουλάχιστον με τα σημερινά δεδομένα.

συνήθως δουλεύουν καλά όταν

Το σενάριο είναι απλό**Υπάρχουν πολλά δεδομένα**

συνήθως αποτυγχάνουν όταν

Πολύπλοκο σενάριο με λίγα δεδομένα**Πρέπει να ανταπεξέλθει σε πολύ διαφορετικό
από ό,τι το εκπαιδευμένο σύνολο**

Πίνακας 1: Αντιπαραβάλλοντας τις παραμέτρους εφαρμογών Μηχανικής Γνώσης

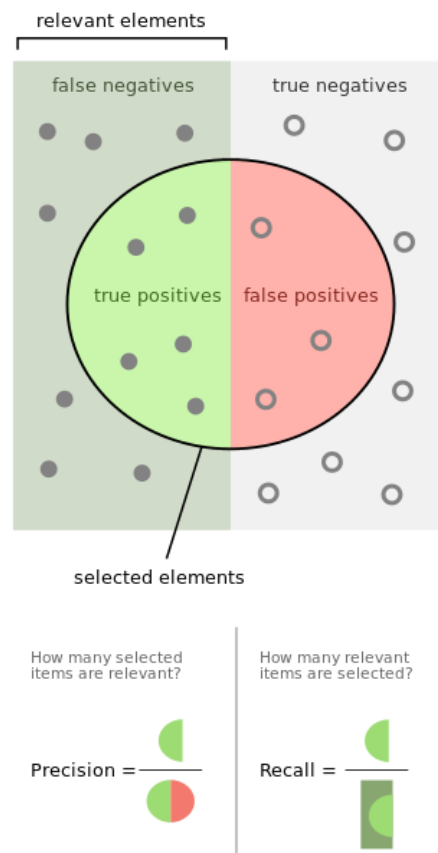
(Πηγή: <https://www.coursera.org/lecture/ai-for-everyone/what-machine-learning-can-and-cannot-do-rv1fw>)

Στην μάθηση υπό επίβλεψη η αποδοτικότητα των αλγορίθμων αυτών μετράται υπολογίζοντας διάφορα σκορ και μετρικά αποτελέσματα, ανάλογα την προσέγγιση. [24] Μερικά από τα πιο δημοφιλή είναι:

- Ακρίβεια(accuracy): το ποσοστό σε % του πόσα από τα δείγματα προέβλεψε σωστά την κατηγορία τους
- F-score(F1): υπολογίζεται χρησιμοποιώντας δύο δείκτες. Του αγγλικού όρου precision(P), δηλαδή του πόσα από τα δείγματα που έχουν επιλεχθεί είναι σχετικά με την κατηγορία, και του όρου recall(R), δηλαδή πόσα από τα δείγματα της κατηγορίας έχουν επιλεχθεί και υπολογίζονται εύκολα όπως φαίνεται στην παρακάτω εικόνα.

$$F_1 = \frac{2}{\text{recall}^{-1} + \text{precision}^{-1}} = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}} = \frac{\text{tp}}{\text{tp} + \frac{1}{2}(\text{fp} + \text{fn})}$$

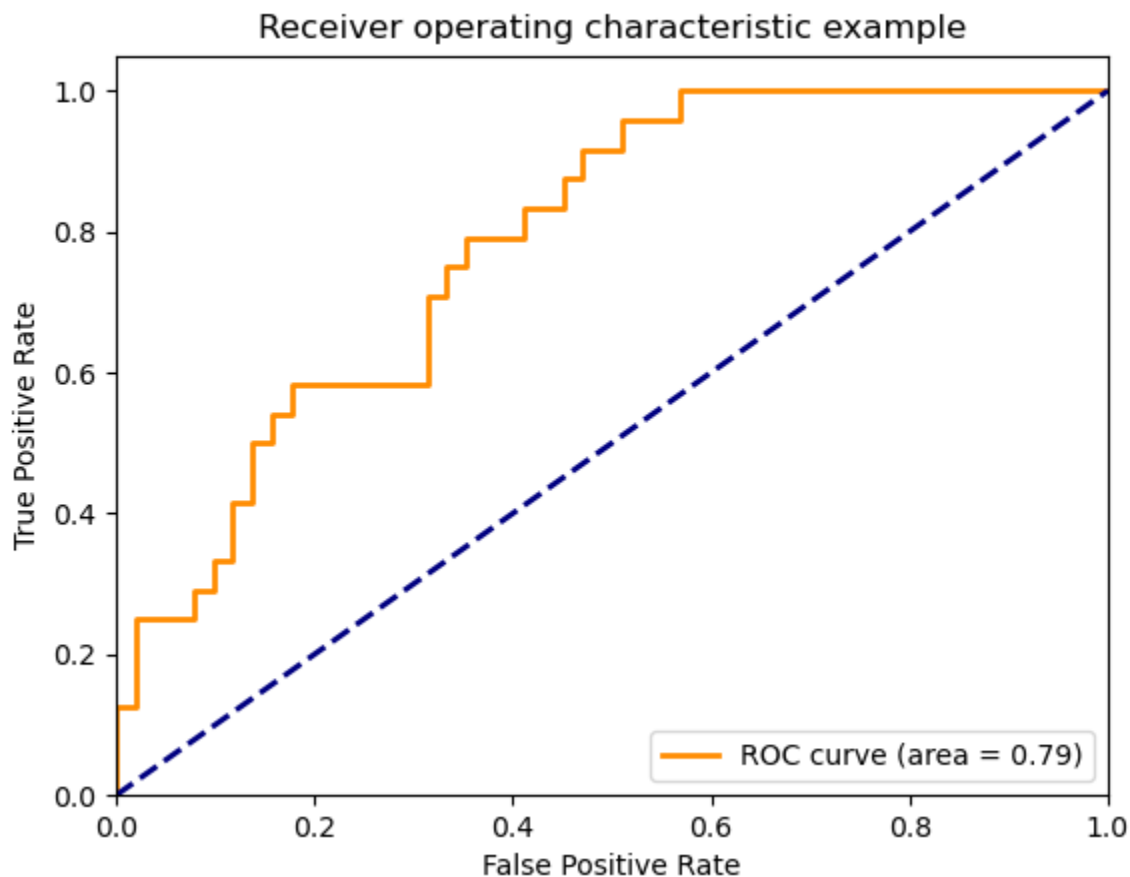
Εικόνα 7: Εξίσωση F1

(Πηγή: <https://en.wikipedia.org/wiki/F-score>)

Εικόνα 8: Απεικόνιση precision & recall

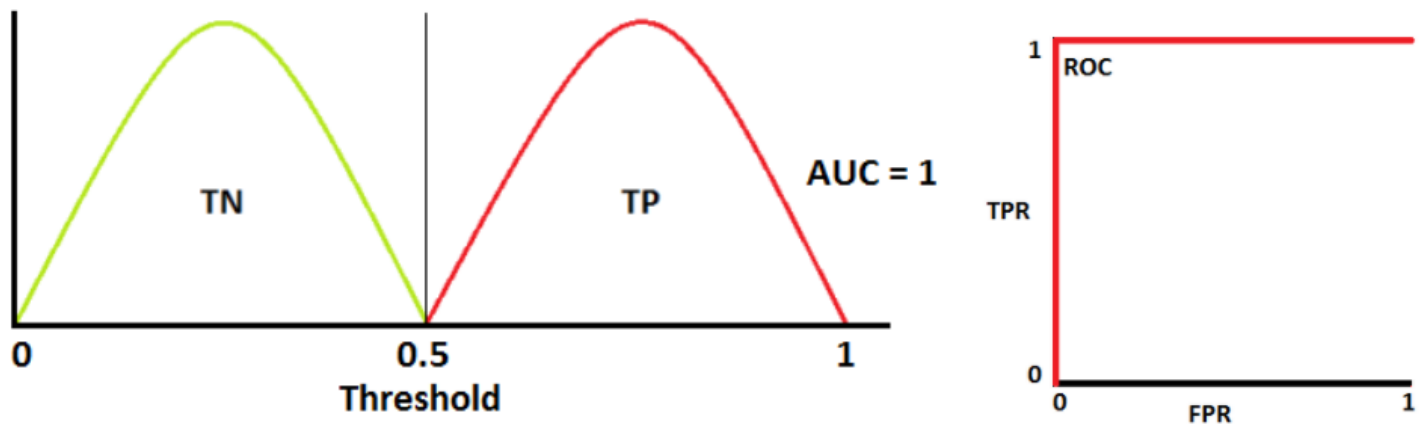
(Πηγή: <https://en.wikipedia.org/wiki/F-score>)

- Receiver Operating Characteristics - Area Under the Curve (ROC-AUC): δείχνει πόσο ικανό είναι το μοντέλο να ξεχωρίσει ένα δείγμα ανάμεσα σε δύο κλάσεις. Το ROC είναι η καμπύλη πιθανότητας και το AUC είναι το εμβαδόν κάτω από αυτή την καμπύλη. Όσο μεγαλύτερο το AUC, δηλαδή η περιοχή κάτω από την καμπύλη, ή αλλιώς όσο πιο ψηλά είναι η καμπύλη τόσο το καλύτερο μπορεί το μοντέλο να διακρίνει τις κλάσεις. Εάν η καμπύλη είναι μια ευθεία που διχοτομεί στην μέση το εμβαδόν του χώρου τότε το σκορ είναι στο 50%, δηλαδή το μοντέλο έχει την χειρότερη επίδοση και δεν έχει καμία ικανότητα να ξεχωρίσει τις κλάσεις. Εάν η καμπύλη βρίσκεται πάνω από το 50%, δηλαδή πάνω από τη μέση τείνοντας προς το 1, τότε το μοντέλο έχει την ικανότητα να διακρίνει σωστά την κλάση υπό εξέταση αντιπαραβάλλοντάς την με μια άλλη. Εάν η καμπύλη βρίσκεται κάτω από το 50%, δηλαδή κάτω από την μέση τείνοντας προς το 0, τότε το μοντέλο προβλέπει την ανάποδη κλάση από ό,τι θα έπρεπε, το οποίο είναι λάθος αλλά είναι τουλάχιστον καλύτερο από το ROC-AUC με σκορ 50%. [25] Το πόσο απότομα ανεβαίνει η καμπύλη είναι επίσης σημαντικό καθώς το ιδανικό είναι να μεγιστοποιείται στην επάνω αριστερή γωνία το γρηγορότερο δυνατό αυξάνοντας το true positive rate(TPR) ενώ ταυτόχρονα ελαχιστοποιεί το false positive rate(FPR) και έτσι μεγαλώνει το συνολικό εμβαδό ή αλλιώς την περιοχή που βρίσκεται κάτω από την καμπύλη. [26] Το πιο σημαντικό χαρακτηριστικό απ' όλα για το ROC-AUC είναι πως δεν εξαρτάται ούτε επηρεάζεται από την ασυμμετρία των κλάσεων(class imbalance).



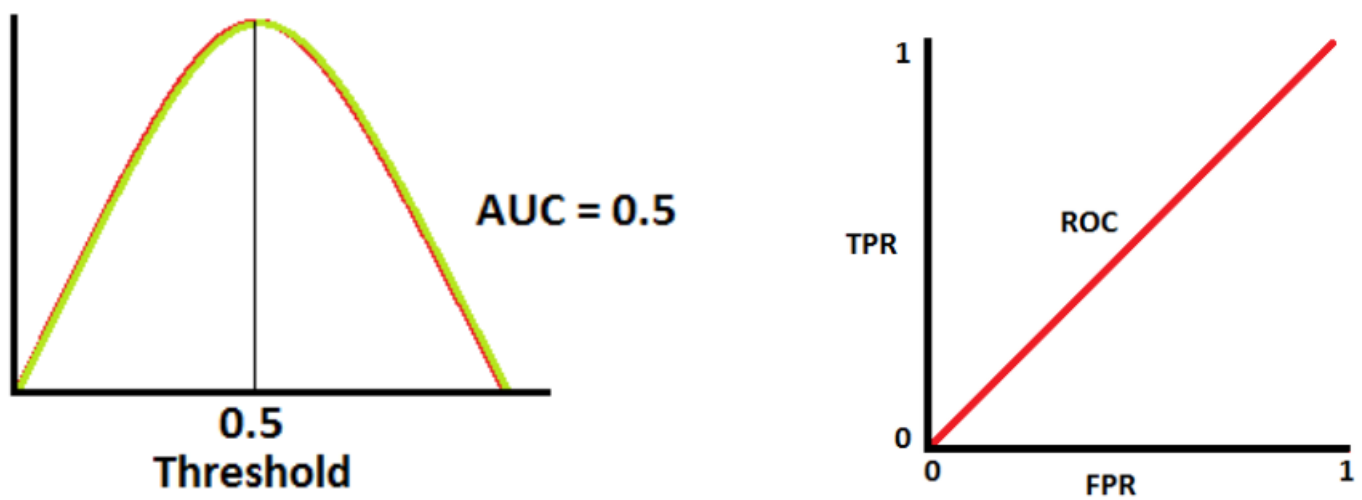
Εικόνα 9: Ένα μέσο διάγραμμα ROC-AUC

(Πηγή: [26])



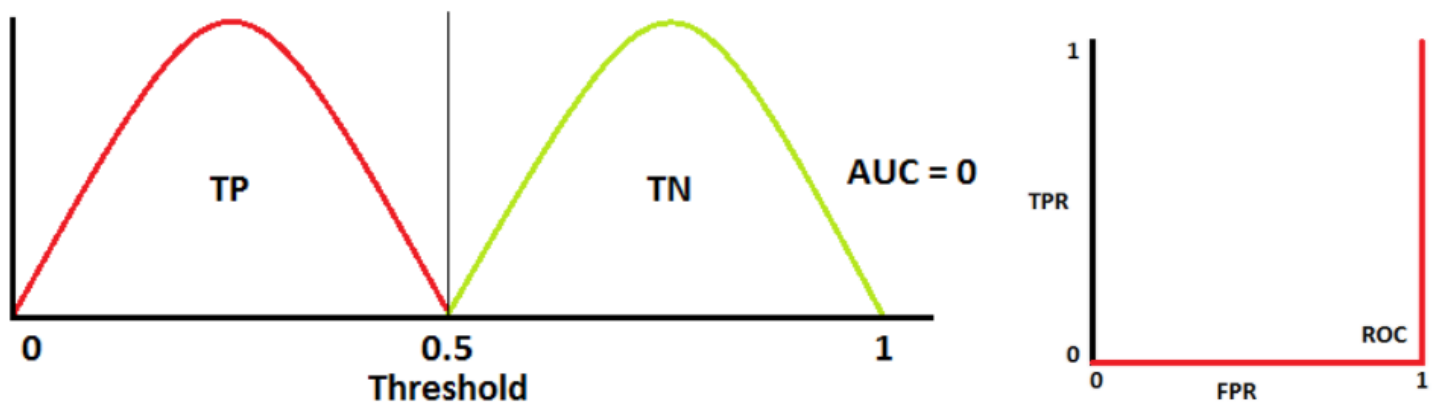
Εικόνα 10: Ιδανικό διάγραμμα ROC-AUC = 1

(Πηγή: [25])



Εικόνα 11: Χειρότερη περίπτωση, ανίκανο να προβλέψει κατηγορία ROC-AUC = 0.5

(Πηγή: [25])



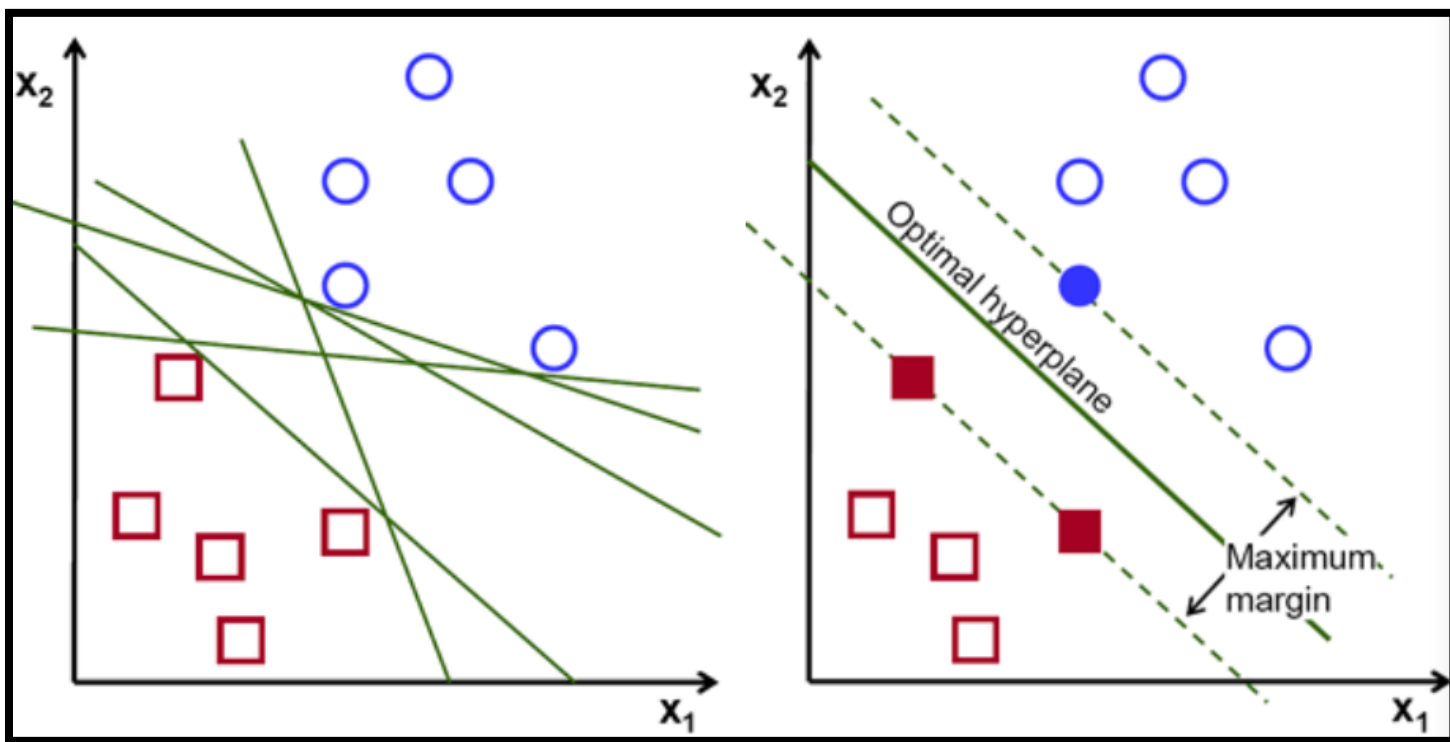
Εικόνα 12: Ιδανική ανάποδη πρόβλεψη ROC-AUC = 0

(Πηγή: [25])

Μηχανές διανυσμάτων υποστήριξης

Από τους πιο δημοφιλείς και γρήγορους αλγορίθμους στην κατηγοριοποίηση είναι οι μηχανές διανυσμάτων υποστήριξης (Support Vector Machines – εφεξής SVM) βρίσκοντας και διαχωρίζοντας με ιδανικό περιθώριο (ή αλλιώς όριο) τα δείγματα όπου ανήκουν στις εκάστοτε κλάσεις από τα δεδομένα. Καθώς έρχονται νέα δείγματα, η δουλειά των μηχανών είναι να υπολογίσει σε ποια πλευρά θα «πέσει» το σημείο, βρίσκοντας έτσι την κλάση του.

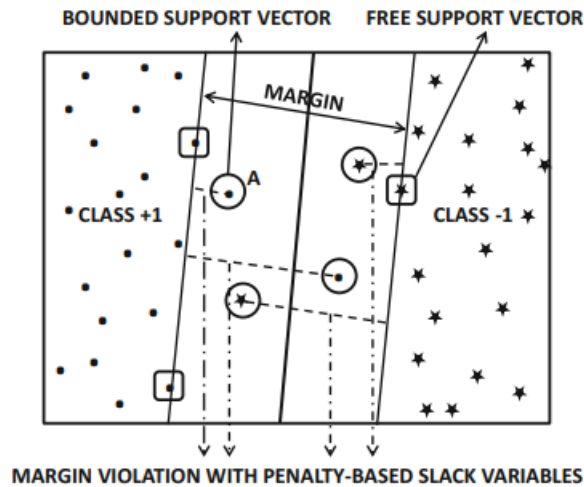
Βασίζεται πάνω στην γραμμική παλινδρόμηση, δηλαδή απλοϊκά στην εύρεση της εκείνης διαχωριστικής γραμμής όπου διαχωρίζει σωστά τα σημεία ανάμεσα σε δυο πλευρές. Η βασική ιδέα είναι να υπολογιστούν όλα τα πιθανά διαχωριστικά σημεία τα οποία αποτελούν όρια αναμεταξύ των κλάσεων και στο τέλος να επιλεγεί εκείνο το οποίο έχει το μεγαλύτερο περιθώριο απόκλισης ανάμεσα από τις κλάσεις, δηλαδή την μεγαλύτερη απόσταση από τα πλησιέστερα σημεία της καθεμιάς κλάσης.



Εικόνα 13: Πιθανοί διαχωριστές σε δείγμα δύο κλάσεων

(Πηγή: <https://towardsdatascience.com/support-vector-machine-introduction-to-machine-learning-algorithms-934a444fca47>)

Τα επονομαζόμενα *διανύσματα υποστήριξης (support vectors)* είναι τα σημεία δεδομένων και με βάσει αυτά υπολογίζεται το που θα γίνει ο διαχωρισμός ανάμεσα στις κλάσεις. Καθώς προσθέτονται ή αφαιρούνται αλλάζει ανάλογα και η θέση του διαχωρισμού. [27] Είναι σημαντικό να βελτιστοποιηθεί το διαχωριστικό πλαίσιο το οποίο μπορεί να αγγίζει ένα ή περισσότερα σημεία το οποίο ονομάζεται *βέλτιστο*. Αυτά τα σημεία που αγγίζει το πλαίσιο ονομάζονται ως *ελεύθερα διανύσματα υποστήριξης (free support vectors)*. Τα σημεία τα οποία ξεπερνούν τα όρια του πλαισίου ονομάζονται *περιθωριακά διανύσματα υποστήριξης (bounded support vectors)* και σε αυτά επιβάλλεται «ποινή», δηλαδή μείωση στην τιμή τους. [28]



Εικόνα 14: Ελεύθερα & περιθωριακά διανύσματα υποστήριξης

(Πηγή: [28])

Αυτά τα σημεία είναι που βοηθούν να χτιστεί εν τέλη η μηχανή η οποία υπολογίζει το «κόστος» των σημείων που μπορεί να κυμαίνεται μεταξύ $[-1, 1]$. Τα σημεία που σκοράρουν πιο κοντά στο 1 ανήκουν στην μια κλάση ενώ αυτά που σκοράρουν -1 ανήκουν στην άλλη, με την ιδανική γραμμή(ιδανικό πλαίσιο) να σκοράρει 0. Η εξίσωση που υπολογίζει τις τιμές των διανυσμάτων, βασίζεται πάνω στην εξίσωση της γραμμικής παλινδρόμησης και για χάρη της απλότητας δεν θα αναφερθεί καθώς διαφέρει ανάλογα με τους παράγοντες, όπως την εκάστοτε εφαρμογή, την χρήση προκατάληψης(bias) ή όχι, τη σκληρότητα της ποινής και άλλων ιδιοτήτων.

Απλοποιημένο θεώρημα Bayes

Ένας απλός κατηγοριοποιητής, γρήγορος και συνηθέστερα χρησιμοποιούμενος είναι μάλλον ο αλγόριθμος απλοποιημένου θεωρήματος Bayes(Naive Bayes Theorem – εφεξής NB), που είναι πολύ ικανός να ανταπεξέλθει σε μεγάλα σετ δεδομένων, όχι όμως και στα πιο πολύπλοκα. Μοντελοποιεί την κατανομή των δειγμάτων σε κάθε κλάση βασιζόμενο στο γνωστό θεώρημα Bayes.

$$P(L | \text{features}) = \frac{P(\text{features} | L)P(L)}{P(\text{features})}$$

Εικόνα 15: Το θεώρημα Bayes με L ως κατηγορία(label) και features ως δειγματοστοιχεία

(Πηγή: <https://jakevdp.github.io/PythonDataScienceHandbook/05.05-naive-bayes.html>)

Ο NB είναι πολύ διαισθητικός στην κατανόηση. Εάν προσπαθούμε να αποφασίσουμε μεταξύ ποιών δύο κατηγοριών ανήκει ένα feature (ας υποθέσουμε L_1 και L_2) τότε για να υπολογίσουμε τις μεταγενέστερες(posterior) πιθανότητες να ανήκει στην κλάση 1 θα παίρναμε όλες το ποσοστό των προηγούμενων feature που ανήκουν στην κλάση 1 και το ποσοστό αυτών που ανήκουν στην κλάση 2 και θα το πολλαπλασιάζαμε με το ποσοστό των document που περιέχουν το feature και ανήκουν στην κλάση 1 προς το ποσοστό των document που περιέχουν το feature και ανήκουν στην κλάση 2. [29] Η μαθηματική απεικόνισή του είναι:

$$\frac{P(L_1 | \text{features})}{P(L_2 | \text{features})} = \frac{P(\text{features} | L_1) P(L_1)}{P(\text{features} | L_2) P(L_2)}$$

Εικόνα 16: Posterior πιθανότητες Bayes

(Πηγή: <https://jakevdp.github.io/PythonDataScienceHandbook/05.05-naive-bayes.html>)

Αποκαλείται απλοποιημένο(ή σε απευθείας μετάφραση «αφελές») επειδή υποθέτει ότι όλα τα feature είναι ανεξάρτητα μεταξύ τους, πράγμα που δεν ισχύει σχεδόν ποτέ, και έτσι ο αλγόριθμος απλά ψάχνει πού υπάρχουν περισσότερες πιθανότητες να ανήκει ένα feature για να αποφανθεί την κατηγορία του με την Μπαεσιανή κατηγοριοποίηση, χωρίς ακόμη να λαμβάνει υπόψιν την βαρύτητα(δηλαδή χρησιμότητα) αυτού του feature. Αυτό βέβαια δεν είναι πάντα επιθυμητό... Εάν για παράδειγμα κατασκευάζαμε ένα φίλτρο ανεπιθύμητης αλληλογραφίας(spam) για email και στο dataset όλα τα μηνύματα που ήταν κάτω από 50 χαρακτήρες θεωρούνταν σαν spam τότε μηνύματα που ίσως προέρχονταν από αξιόπιστους αποστολείς αλλά είχαν λιγότερους από 50 χαρακτήρες, επειδή περιείχαν εικόνες ή συνημμένα αρχεία, τότε και αυτά θα θεωρούνταν ανεπιθύμητα. Φυσικά προϋποθέτει τα dataset των κλάσεων να μην ίδια σε έκταση(ώστε να μην καταλήξουμε σε πιθανότητες $50/50$) το οποίο είναι πολύ σπάνιο, αν όχι απίθανο. [30] Οπότε λοιπόν ο προηγούμενος τύπος καταλήγει σε:

$$\frac{P(L_1 | \text{features})}{P(L_2 | \text{features})} = \frac{\cancel{P(\text{features} | L_1)} P(L_1)}{\cancel{P(\text{features} | L_2)} P(L_2)}$$

Εικόνα 17: Το θεώρημα NB

Οι αλγόριθμοι NB είναι πολλών χρήσεων. Οι τρεις πιο δημοφιλείς παραλλαγές του NB [31] είναι:

1. **Μπερνούλι NB(Bernoulli NB)**: ένας δυαδικός αλγόριθμος (0 ή 1) οπού αποφασίζεται εάν ένα feature συμβαίνει(δηλαδή υπάρχει) ή όχι σε μια κλάση.

$$P(X) = \begin{cases} p & \text{if } X = 1 \\ q & \text{if } X = 0 \end{cases}$$

$$\text{where } q = 1 - p \text{ and } 0 < p < 1$$

Εικόνα 18: Εξίσωση BNB

(Πηγή: [31])

2. **Πολυωνυμικός NB(Multinomial NB)**: ένας αλγόριθμος όπου κάθε διάνυσμα ενός feature αντιπροσωπεύεται από έναν αριθμό που δηλώνει πόσες φορές εμφανίζεται σε μια κατηγορία ή αλλιώς αντιπροσωπεύει την συχνότητά του, δηλαδή υποθέτει την πολυωνυμική κατανομή των features. Επειδή ο MNB περιγράφει την πιθανότητα αριθμητικών εμφανίσεων ανά κατηγορία, είναι πολύ αποτελεσματικός σε περιπτώσεις επεξεργασίας φυσικής γλώσσας ή όταν τα feature είναι οργανωμένα σε μορφή λεξιλογίου. [30]

$$P(X_1 = x_1 \cap X_2 = x_2 \cap \dots \cap X_k = x_k) = \frac{n!}{\prod_i x_i!} \prod_i p_i^{x_i}$$

Εικόνα 19: Εξίσωση MNB

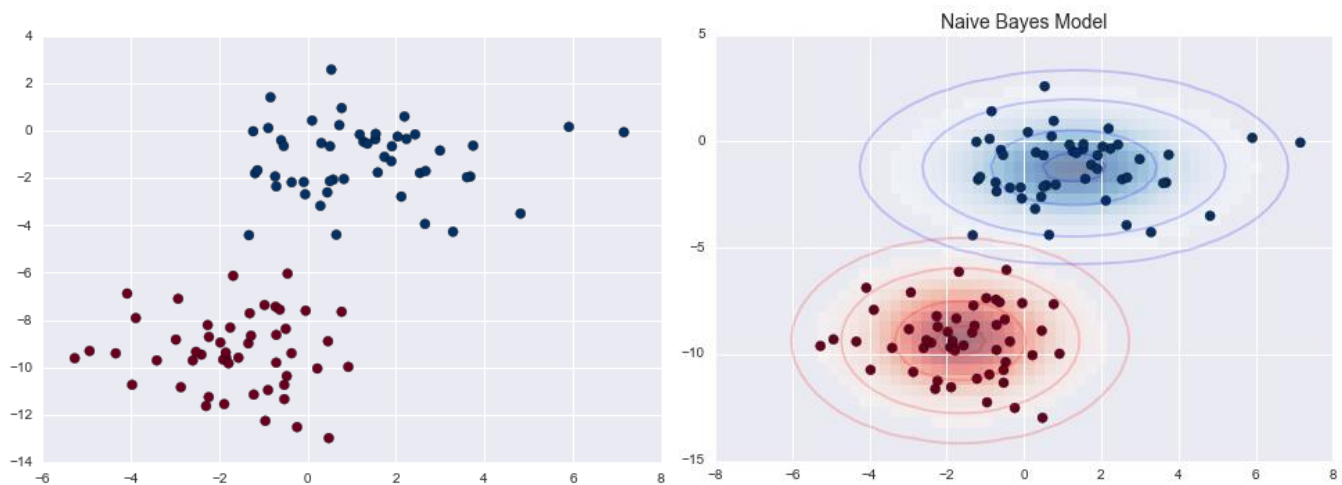
(Πηγή: [31])

3. **Γκαουσιανός NB(Gaussian NB)**: είναι ο πιο εύκολος αλγόριθμος από τους τρεις γιατί υποθέτει πως τα δεδομένα σε κάθε κατηγορία έχουν επιλεχθεί με βάσει την γκαουσιανή(δηλαδή τυχαία) κατανομή, χωρίς κάποια συνδιακύμανση μεταξύ τους, οπότε αρκεί να βρεθεί η μαθηματική τους ελπίδα και η τυπική απόκλιση αυτών ώστε να αποφανθεί η κατηγορία στην οποία ανήκει ένα σημείο.

$$L(\mu; \sigma^2; x_i | y) = \log \prod_k P(x_i^{(k)} | y) = \sum_k \log P(x_i^{(k)} | y)$$

Εικόνα 20: Εξίσωση GNB

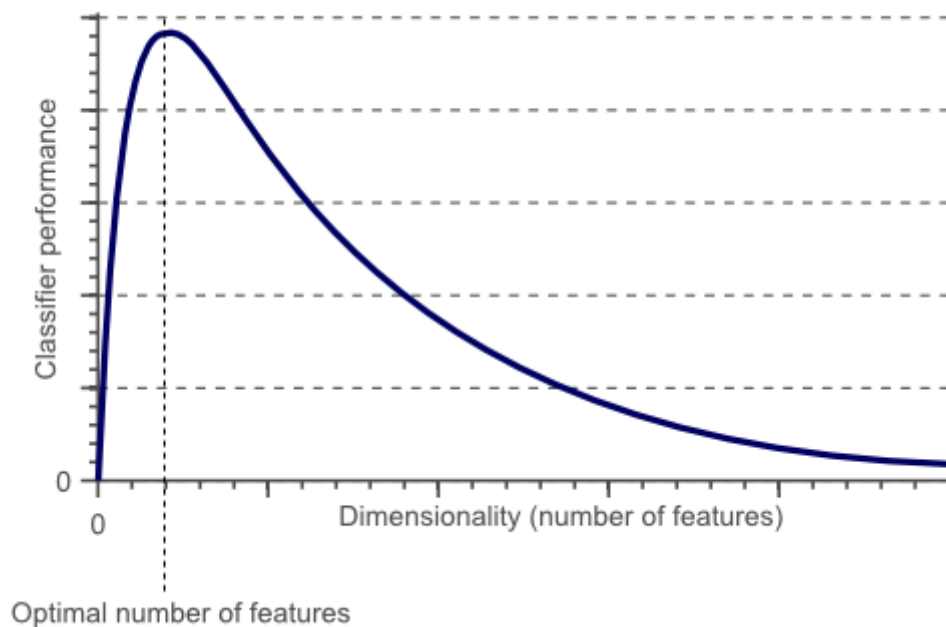
(Πηγή: [31])



Εικόνα 21: Παράδειγμα Gaussian NB

(Πηγή: [30])

Οι αλγόριθμοι προτιμώνται γενικά επειδή, είναι πολύ γρήγοροι κατά την εκπαίδευση και κατά την πρόβλεψη, είναι απλοί, είναι εύκολοι στην κατανόηση και έχουν πολύ λίγες υπερ-παραμέτρους(hyper parameters). Εάν αποδίδουν καλά σε ένα dataset τότε μάλλον είναι απλό ή καλά ορισμένο, με τις κατηγορίες να είναι ξεκάθαρες και να διαφέρουν αρκετά(dimensionality). Εάν δεν αποδίδει τόσο καλά τότε μάλλον κάτι από τα προηγούμενα δεν ισχύει και θα πρέπει να καταφύγει κανείς σε πιο εκλεπτυσμένες μεθόδους ή να μειώσει την dimensionality. [30]



Εικόνα 22: Εξέλιξη απόδοσης αλγορίθμου κατηγοριοποίησης με επέκτασης διάστασης (dimensionality)

(Πηγή: https://miro.medium.com/max/638/1*vah8lolNqlxNHq9ysVzYkw.png)

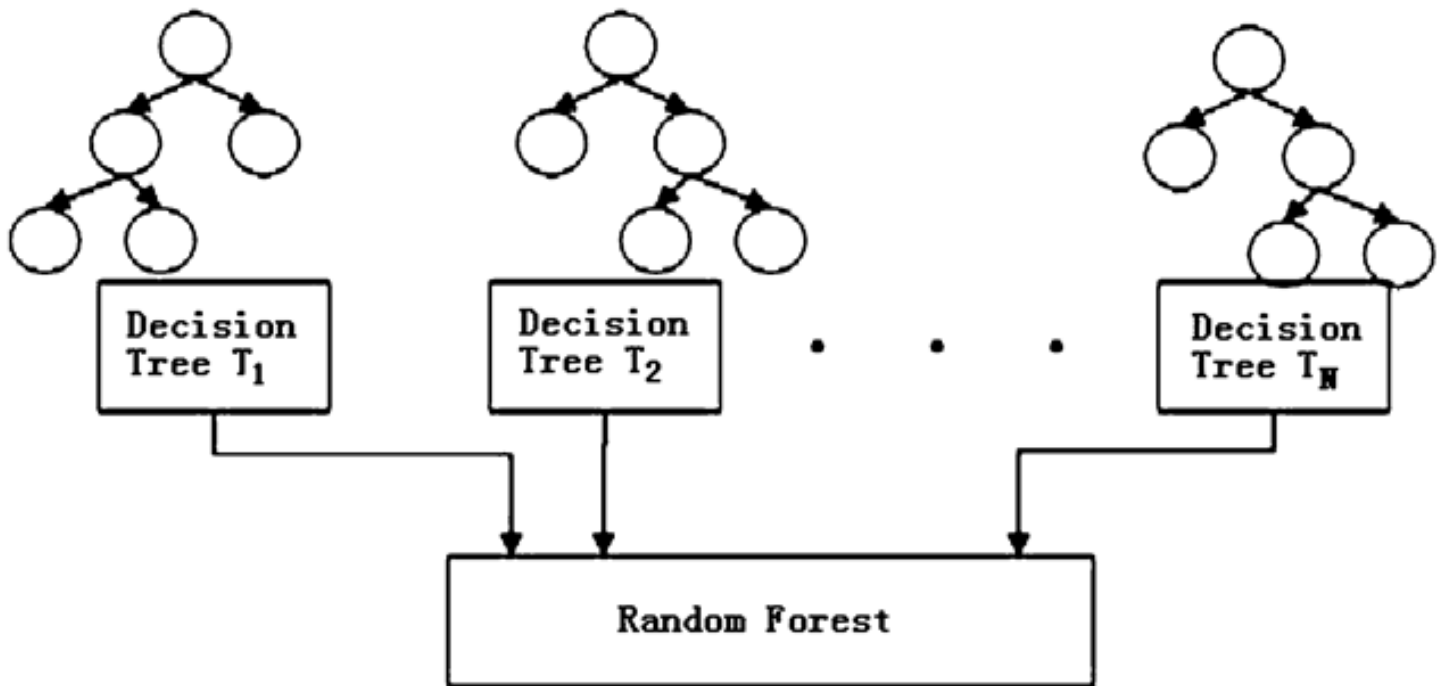
Τυχαία δάση

Ένα δένδρο απόφασης είναι μια ιεραρχική δομή η οποία (όπως προκύπτει και από το όνομα) προσιδιάζει αυτή των δένδρων, διαιρώντας τα δεδομένα σε κόμβους ανάλογα τις ιδιότητές τους. Η λογική είναι να διαιρέσει τα δεδομένα σε περιοχές με συγκεκριμένα χαρακτηριστικά τα οποία εμφανίζονται σε κάποια συγκεκριμένη κλάση. Επομένως οι εκάστοτε περιοχές των δένδρων μαζί με τα υποδένδρα αυτών ανήκουν σε μια κλάση το καθένα. Τα υποδένδρα χωρίζονται σε κόμβους (ή αλλιώς σε φύλλα) ανάλογα με τις συνθήκες που έχουν οριστεί από τον γονέα-φύλλο τους, από επάνω προς τα κάτω. Η ρίζα του δένδρου (δηλαδή ο αρχικός κόμβος) περιέχει ολόκληρο το dataset.

Όσο πιο αυστηρές οι συνθήκες τόσο πιο μεγάλα τα δένδρα, τόσο περισσότεροι οι διαχωρισμοί αλλά και πιο αργά να υπολογιστούν όμως προσφέρουν υψηλή ακρίβεια και θεωρούνται πολύ ισχυρά απέναντι σε μεγάλα dataset. Τα κριτήρια με τα οποία «σπάει» κάθε φύλλο σε κόμβους μοιάζουν λίγο-πολύ με την διαδικασία του feature selection και είναι συνήθως κάποιοι περιορισμοί σε εμφανίσεις feature. Λόγου χάρη σε ένα dataset αυτοκινήτων, σε ένα κλαδί του δέντρου, ο διαχωρισμός σε έναν κόμβο θα αποτελείται από αυτά που είναι ηλεκτρικά και από όλα τα υπόλοιπα (βενζινοκίνητα, πετρελαιοκίνητα, κτλ.) που δεν είναι, και κατά αυτόν τον τρόπο θα συνεχίσει ο διαχωρισμός προς τα κάτω μέχρις ότου κάθε παιδί-φύλλο περιέχει μόνο μια κλάση. Είναι σύνηθες ένας κόμβος να έχει μόνο δύο φύλλα, όπως απορρέει λογικά από το προαναφερθέν παράδειγμα που δόθηκε, μεγαλώνοντας έτσι τα κλαδιά σε μια μορφή που θυμίζει δένδρο. Υπάρχουν πολλοί τρόποι για να οριστεί ο κανόνας διαχωρισμού ενός κόμβου αλλά για χάρη της απλότητας δεν θα αναφερθούν καθώς διαφέρουν ανάλογα με τους παράγοντες, όπως την εκάστοτε εφαρμογή, το μέγεθος του dataset, την dimensionality, κ.α..

Όπως απορρέει από το παράδειγμα που δόθηκε παραπάνω, τα δέντρα απόφασης χωρίζουν τμηματικά τους κόμβους τους με όλο και μικρότερα dataset, το οποίο αυτό είναι ιδιαίτερα προβληματικό σε μικρά dataset και επιφέρει λανθασμένα αποτελέσματα & προκατάληψη (bias) σχετικά με την συνέπεια της κατηγοριοποίησης. Εδώ είναι που τα τυχαία δάση (Random Forest – εφεξής RF) έρχονται να προσφέρουν την λύση προσφέροντας

ένα μικρότερο τυχαίο σετ από πιθανά κριτήρια με τα οποία σπάει κάθε φύλλο, οπότε κάθε δένδρο είναι διαφορετικό και επιλέγει το καλύτερο κριτήριο για σπάσει ένα κλαδί από αυτά που του προσφέρονται, εξ ου και η απόδοση του “τυχαίο”, και στην συνέχεια κάθε νέος κόμβος από κάτω χρησιμοποιεί και αυτός διαφορετικά κριτήρια από το κάθε σετ που του προσφέρεται. Όπως είναι λογικό, όσο λιγότερα διαφορετικά κριτήρια προφέρονται σε κάθε κόμβο για να επιλέξει, τόσα περισσότερα διαφορετικά δένδρα κατασκευάζονται. Τέλος, η κατηγοριοποίηση γίνεται βάσει ψηφοφορίας, δηλαδή ποια είναι κλάση την οποία συμφώνησαν τα περισσότερα τυχαία δάση ότι ανήκει ένα feature υπό εξέταση. Επειδή η τελική απόδοση της κατηγορίας γίνεται μέσω ψηφοφορίας(δηλαδή αθροίσματος ψήφων) αποφεύγεται έτσι η προκατάληψη & η ασυνέπεια, ιδιαίτερα στα μικρότερα dataset, για αυτό και προτιμάται ιδιαιτέρως σε αυτά. [32]



Εικόνα 23: Απεικόνιση ενός RF

(Πηγή: [33])

Επεξεργασία φυσικής γλώσσας

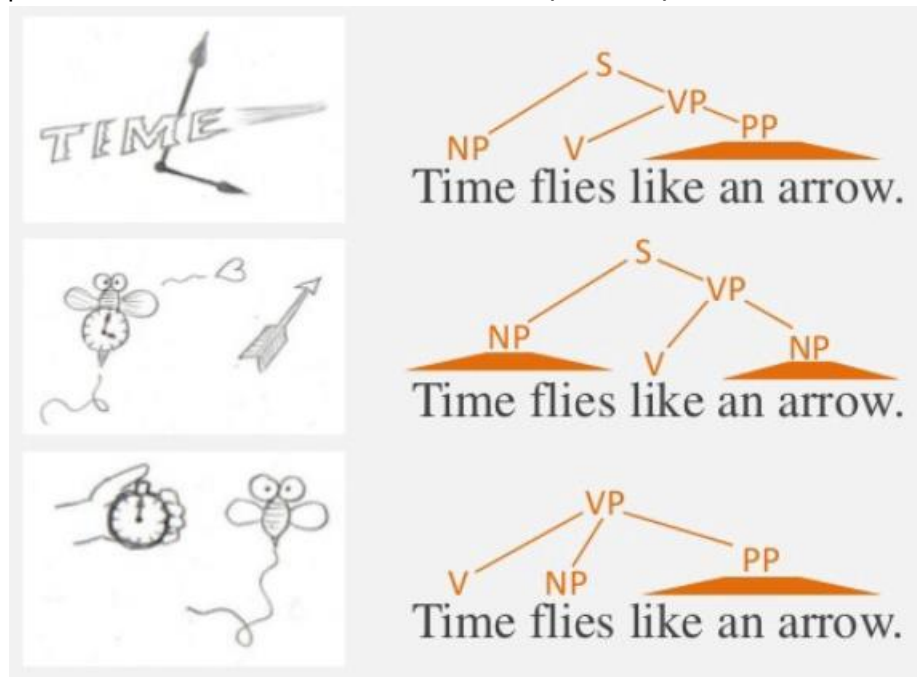
Στην πιο απλή μορφή, ένα κείμενο είναι μια σειρά από δείγματα, επονομαζόμενα ως token, τα οποία έχουν κάποια σχέση μεταξύ τους. Αυτή την συσχέτιση στοχεύει να ανακαλύψει η επεξεργασία φυσικής γλώσσας(Natural Language Processing – εφεξής NLP) η οποία βασίζεται στις τακτικές που ακολουθούνται στην IR, ML, την στατιστική και την γλωσσολογία. Με άλλα λόγια κύριος στόχος της NLP είναι να κατανοήσει την ανθρώπινη γλώσσα, ιδανικά όπως θα την αντιλαμβανόταν κι ένας άνθρωπος.

Εφαρμόζεται στην κατηγοριοποίηση κειμένου, όπως επίσης και στην ανάλυση των ανθρώπινων συναισθημάτων που προβάλλονται από το κείμενο. Εφαρμόζεται στην IR, όπως για παράδειγμα σε μια μηχανή αναζήτησης. Ακόμα, χρησιμοποιείται στην μηχανική μετάφραση.

Το μεγαλύτερο πρόβλημα είναι φυσικά η πολυπλοκότητα της ανθρώπινης γλώσσας, τα διαφορετικά νοήματα, η αργκό & τα ορθογραφικά λάθη, το διαφορετικό ύφος του κειμένου(π.χ. σαρκασμός), κ.α. τα οποία οι μηχανές ακόμη δυσκολεύονται να αντιληφθούν.

Διάφορες μέθοδοι προσέγγισης έχουν αναπτυχθεί. [34] Οι δύο σημαντικότεροι είναι:

- **Τμηματοποίηση & προ επεξεργασία(tokenization & preprocessing):** συνήθως αποτελούν τα πρώτα βήματα. Τα κείμενα διασπώνται σε tokens τα οποία μπορεί να είναι σκέτα γράμματα, σκέτες λέξεις, ολόκληρες προτάσεις ή και ολόκληρες παραγράφους ανάλογα την περίπτωση. Οι λέξεις μπορεί να φέρονται στην μορφή ρίζας(stemming) ή στη μορφή λεξιλογίου(lemmatizing) ή μπορεί να γίνεται αντικατάσταση λέξεων σύμφωνα με ένα λεξικό(out of vector dictionary) ή σύμφωνα με κάποιο προπαρασκευασμένο διάνυσμα λέξεων(word vector), ή κατάργηση άχρηστων λέξεων(stop words), κ.α..
- **Επισήμανση μερών του λόγου & αναγνώριση οντότητας(parts-of-speech tagging & entity recognition):** στο καθένα token δίνεται μια ετικέτα για το τι μέρους του λόγου είναι(ουσιαστικό, ρήμα, υποκείμενο, επίρρημα, αντικείμενο, κ.α.). Επιπλέον δίνονται ετικέτες όπως εάν είναι οντότητα, ή οργανισμός, ή χώρα, κ.α., συνολικά πάνω από 180 δυνατές ετικέτες.



Εικόνα 24: Παραδείγματα part-of-speech tagging

(Πηγή: <https://www.freecodecamp.org/news/an-introduction-to-part-of-speech-tagging-and-the-hidden-markov-model-953d45338f24/>)

Μελέτη περίπτωσης

Οι πλατφόρμες κοινωνικής δικτύωσης, όπως είναι το Twitter προσφέρουν ανοικτά κανάλια & ζωντανή επικοινωνία κατά την διάρκεια μεγάλων παραστάσεων, ιστορικών γεγονότων ή ακόμα και εκτάκτων αναγκών όπως τυφώνων, σεισμών, ιατρικού επείγοντος και δασικές πυρκαγιές. Κατά την αρχή αλλά και κατά την διάρκεια των κρίσεων, οι άνθρωποι που επηρεάζονται αναρτούν χρήσιμες πληροφορίες στο Twitter που μπορεί να είναι περιγραφές της έκτακτης ανάγκης, νέες εξελίξεις, η αναζήτηση ανθρωπιστικής βοήθειας & πληροφοριών για την κρίση, ψυχολογική υποστήριξη, αναζήτηση αγνοουμένων, κ.α..

Ο κ. Waqas και οι συνεργάτες του πραγματοποίησαν μια ανάλυση του τι δεδομένα δημιουργούνται και το πόσο χρήσιμα είναι αυτά κατά διάρκεια του πανικού των ανεξέλεγκτων φονικών δασικών πυρκαγιών της Καλιφόρνιας τον Νοέμβριο του 2018, δημιουργώντας κατά την διάρκεια αυτών μέχρι και 1.200 αγνοουμένους, καταλήγοντας με 86 ανθρώπους νεκρούς και κατακαίγοντας πάνω από 18.000 κτήρια. Η έρευνα έδειξε πως το Twitter περιέχει σημαντική πληροφορία η οποία είναι δυνητικά χρήσιμη με κατάλληλη εξεργασία (όπως text mining & ML) για χρήση από ομάδες πολιτικής προστασίας & εθελοντές για την έγκαιρη ανταπόκριση. Πολλά tweet βρέθηκαν να περιέχουν όρους-λέξεις όπως ονόματα, επώνυμα, ή πλήρες ονοματεπώνυμο, τοποθεσίες ανθρώπων σε ανάγκη, προσφορές Σαμαρειτών για νερό τροφή & κατάλυση, και άλλα απαραίτητα στοιχεία που συνεισφέρουν στην εύρεση & διάσωση των αγνοουμένων ανθρώπων. [35]

Η συλλογή των tweet τους ήταν με την χρήση λέξεων-κλειδιά. Με το κατάλληλο φιλτράρισμα κατέληξαν στις εξής πιο σημαντικές λέξεις σχετικές με αγνοουμένους & διασωθέντες που είναι πιθανόν να προκύψουν κατά την διάρκεια της κρίσης μιας ανεξέλεγκτης πυρκαγιάς:

Δημοφιλή hashtag σχετικά με τους διασωθέντες

#FOUND	#NORTHVALLEYCOMMUNITYFOUNDATIO
#FOUND!	#FOUNDHORSE
#CAMPFIREFOUNDPETS	#ISRAELI-FOUNDED
#FOUNDDOG	#FOUNDHORSES
#FOUNDDOGS	#LOSTANDFOUND
#CAMPFIREMISSINGFOUNDPETS	#FOUNDCAT
#ANIMALSFOUNDINPARA	#CAFIREFOUND
#FOUNDPETS	#CAMPFIREPETSFOUND
#PUGFOUND	#CAMPFIREFOUNDDOGS
#FOUNDCATS	#FOUNDPET

Πίνακας 2: Δημοφιλή hashtag σε tweet σχετικά με τους διασωθέντες.

(Πηγή: [35])

Δημοφιλή hashtag σχετικά με τους αγνοουμένους

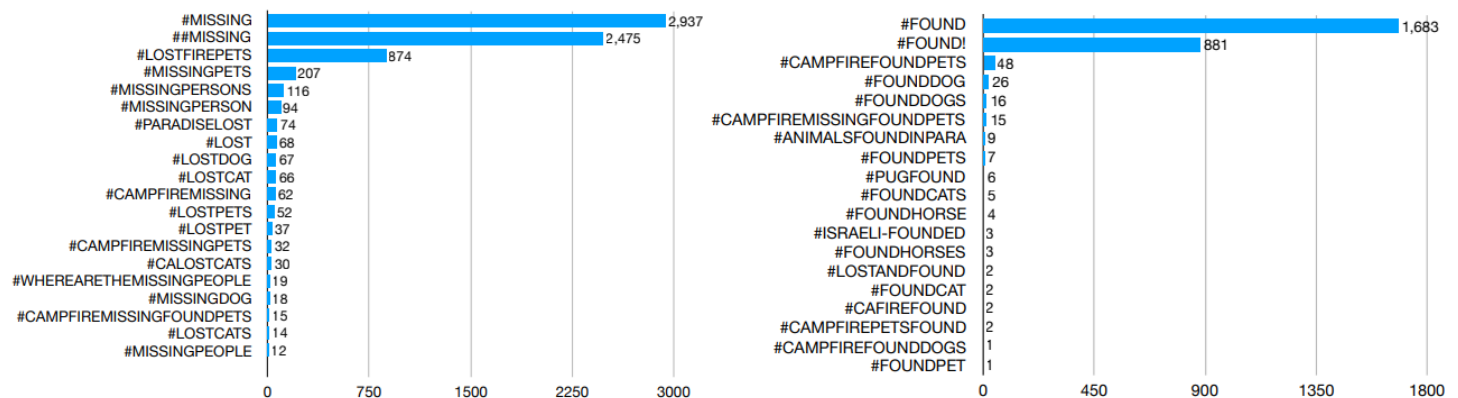
#MISSING	#1000MISSING
##MISSING	#CALIFORNIA71DEATH1000MISSING
#MISSINGPETS	#631MISSING
#MISSINGPERSONS	#WILDFIREMISSING
#MISSINGPERSON	#LISTOFMISSING
#CAMPFIREMISSING	#631PEOPLEMISSING
#CAMPFIREMISSINGPETS	#228PEOPLEMISSING
#WHEREARETHEMISSINGPEOPLE	#LOSTFIREPETS
#MISSINGDOG	#PARADISELOST
#CAMPFIREMISSINGFOUNDPETS	#LOST
#MISSINGPEOPLE	#LOSTDOG

#MISSINGPERSO	#LOSTCAT
#CAMPFIREMISSINGCAT	#LOSTPETS
#CAMPFIREMISSINGLIST	#LOSTPET
#CALIFORNIAMISSING	#CALOSTCATS
#MISSING?	#LOSTCATS
#MISSINGCAT	#CAMPFIRELOST
#CAMPFIREMISSINGLOVE	#CAMPFIRELOSTPETS
#MISSINGPET	#LOST!
#MISSINGCAMPFIREPETS	#CAMPFIRELOSTPET
#PEOPLEMISSING	#CAMPFIRELOSTCATS
#PARADISEMISSING	#LOSTANDFOUND
#LOCATETHEMISSING	#LOSTANIMALS
#FINDMISSING	#LOSTINFIRE
#MISSING!	#LOSTHABITATS
#PEOPLEDIEDANDAREMISSING	#MANYLIVESLOST
#MISSINGLIST	#MISSINGANIMAL
#200PEOPLEMISSING	#MISSINGHOME
#LISTOFMISSINGPERSONS	#THEMISSING
#MISSINGCATS	#MISSINGPERS
#MISSINGINFIRE	#MISSINGPE
#1276PEOPLEMISSING	#CALIFORNIACAMPFIREMISSINGPERSONS

Πίνακας 3: Δημοφιλή hashtag σε tweet σχετικά με τους αγνοούμενους.

(Πηγή: [35])

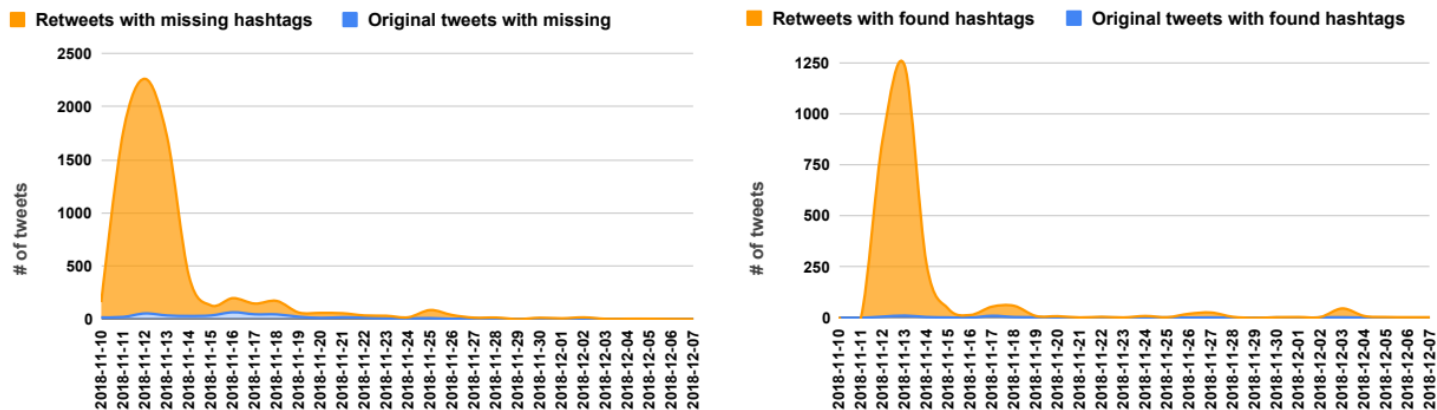
Από όλα, τα πιο δημοφιλή θα έλεγε κανείς πως είναι τα hashtags που περιέχουν την λέξη «missing» και την λέξη «found». Πράγματι αυτό δείχνει η έρευνά τους, ότι δηλαδή οι άνθρωποι που ήθελαν να περάσουν το μήνυμα απόγνωσης τους για τους αγαπημένους τους που αγνοούνται χρησιμοποιούσαν κατά κόρον αυτή την λέξη-κλειδί. Ομοίως, για αυτούς θέλουν να ανακαλέσουν τις αναζητήσεις και να καθησυχάσουν, χρησιμοποιούσαν κατά κόρον την λέξη-κλειδί found.



Εικόνα 25: Συχνότητα εμφανίσεων των λέξεων-κλειδιών missing & found ανάλογα το ύψος του tweet.

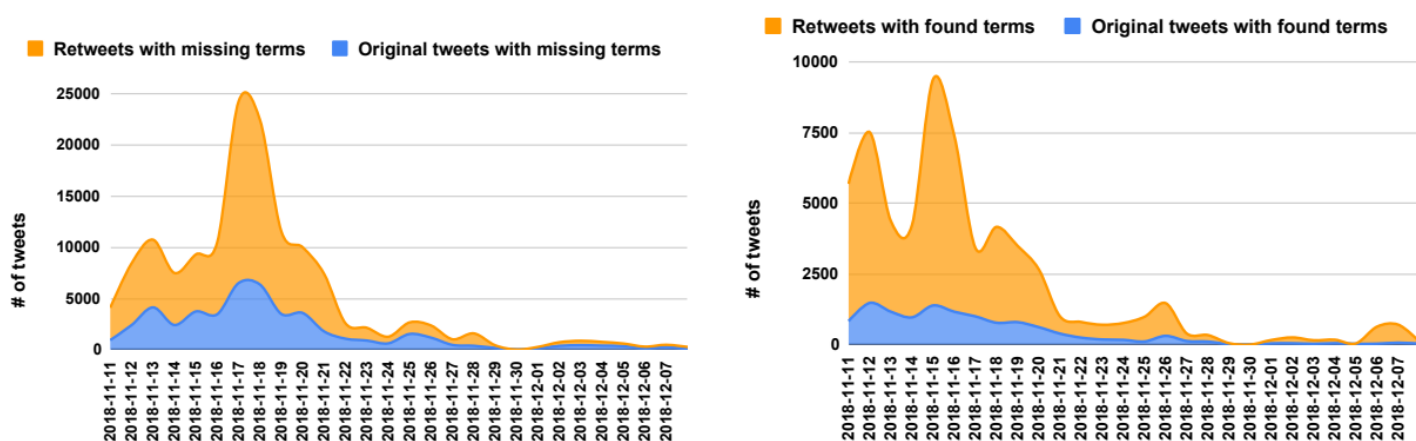
(Πηγή: [35])

Η πιο σημαντική παρατήρηση είναι πως τα προαναφερθέντα tweet, δηλαδή τα tweet που περιέχουν τις λέξεις-κλειδιά «missing» και «found», αντήχησαν περισσότερο στους χρήστες του Twitter οι οποίοι τα έκαναν ανά-δημοσίευση(στο Twitter αυτό αποκαλείται retweet) και έτσι απέκτησαν την μεγαλύτερη κυκλοφορία καταφέροντας να μπουν στις δημοφιλείς τάσεις(trends) του Twitter και να φτάσουν στα περισσότερα μάτια.



Εικόνα 26: Όγκος retweet των tweets που περιέχουν hashtag #missing και #found.

(Πηγή: [35])



Εικόνα 27: Όγκος retweet των tweets που περιέχουν hashtag παραπλήσια με #missing και #found.

(Πηγή: [35])

Όπως φαίνεται από τις παραπάνω εικόνες, η αξιοποίηση αυτών των δεδομένων είναι δύσκολη από έναν άνθρωπο λόγω του τεράστιου όγκου και της ίδιας της φύσης του προβλήματος καθώς, σε έκτακτες ανάγκες, δεν υπάρχει τίποτα πιο πολύτιμο από ένα λεπτό που πάει χαμένο. Η παρούσα διπλωματική αποσκοπεί να μελετήσει μια προϋπάρχουσα περίπτωση του εξωτερικού και στην συνέχεια να εφαρμόσει την ίδια διαδικασία σε ελληνικά δεδομένα. Η περίπτωση του εξωτερικού που επιλέχθηκε είναι η «[Twitter as a Lifeline: Human-annotated Twitter Corpora for NLP of Crisis-related Messages](#)» και για την ελληνική περίπτωση θα γίνει απόπειρα να αναπτυχθεί παρόμοιο σύστημα για την «[Ανέλεγκτη φωνική δασική πυρκαγιά στο Μάτι & στην Κινέτα Αττικής](#)» του 2018. Οι περιπτώσεις περιγράφονται παρακάτω και στην επόμενη ενότητα επεξηγείται η διαδικασία υλοποίησής τους.

Συναφή έργα

Μια ανασκόπηση της πρόσφατης βιβλιογραφίας επιβεβαιώνει το ευρέως διαδεδομένο ενδιαφέρον για την ανίχνευση και την IR από τις αναρτήσεις χρηστών σε κοινωνικά μέσα που περιγράφουν τα τρέχοντα γεγονότα. Τελευταία, η IR από γεγονότα σχετιζόμενα με κρίσεις από τα κοινωνικά μέσα έχει λάβει σημαντική προσοχή.

Ο κ. Alharbi και οι συνεργάτες του χρησιμοποίησαν αραβικά tweets τα οποία συλλέξαν για να αξιολογήσουν κλασσικά μοντέλα τεχνητών νευρωνικών δικτύων (ANN) με κλασσικούς αλγορίθμους μηχανικής μάθησης όσον αφορά τις αποδόσεις τους στην αναγνώριση δημοσιεύσεων σχετιζόμενες με κρίσεις. Τα αποτελέσματα έδειξαν πως τα τεχνητά νευρωνικά δίκτυα απέδωσαν πολύ καλά παρά το μικρό μέγεθος του δείγματος. [36]

Ο κ. Kireyen και οι συνεργάτες του πειραματίστηκαν στην ομαδοποίηση tweets στην αγγλική γλώσσα με βάση το θέμα της ανάρτησης και εισήγαγε μια νέα τεχνική αποκαλούμενη «δυναμική βελτίωση σώματος» (dynamic corpus refinement method) κατά την οποία το γενικό σύνολο δεδομένων ειδικεύεται σε μια διαφορετική κατάσταση κρίσης με τρόπο αυτόματο και χωρίς την ανάγκη επίβλεψης ώστε να εμπεριέχει περισσότερο συμφυή προς την υπό εξέταση κρίση χαρακτηριστικά. [37]

Ο κ. Kshirsagar και οι συνεργάτες του παρουσιάζουν μοντέλα επεξήγησης κρίσεων χωρίς επίβλεψη. Τα μοντέλα αυτά στη συνέχεια τα αξιολογούν με βάση την εγκυρότητα και την ποιότητα της παραγωγής των αυτόματων εξηγήσεων που προσφέρουν αναφορικά με το πώς κατέληξαν στο συμπέρασμα κρίσης. Χρησιμοποίησαν αναδρομικά νευρωνικά δίκτυα (RNN) για να αποδείξουν τη αποτελεσματικότητά τους συγκριτικά με άλλα γραμμικά μοντέλα με την χρήση τριών τεχνικών τροφοδοσίας λέξεων εισόδου στις οποίες βασίζεται ένας καινοτόμος αλγόριθμος αυτόματης δημιουργίας εξηγήσεων που δημιούργησαν. Ως μοντέλο εκμάθησης χρησιμοποίησαν μηνύματα χρηστών που τους παρείχε το δίκτυο ψυχολογικής υποστήριξης χρηστών Koko. [38]

Ο κ. Nguyen και οι συνεργάτες του χρησιμοποίησαν αναδρομικά τεχνητά νευρωνικά δίκτυα (RNN) για ταχεία εκμάθηση και κατηγοριοποίηση δεδομένων σχετικά με κρίσης από το Twitter. Με τον αλγόριθμό τους απέδειξαν πως είναι εφικτό να χρησιμοποιηθεί το μοντέλο τους για τον προσδιορισμό μιας διαφορετικής κρίσης σε νέα δεδομένα χωρίς την ανάγκη συγκεκριμένων πληροφοριών για την υπάρχουσα κρίση αλλά με τη χρήση δεδομένων από παρελθοντικές κρίσεις. [39]

Ο κ. Yu και οι συνεργάτες του χρησιμοποίησαν βασικά συνελκτικά τεχνητά νευρωνικά δίκτυα (CNN) για την κατηγοριοποίηση tweets χρησιμοποιώντας ως δεδομένα αναρτήσεις χρηστών από τρεις τυφώνες στις Η.Π.Α συγκρίνοντάς τα με κλασσικές μεθόδους κατηγοριοποίησης όπως η μηχανική διανυσμάτων υποστήριξης (SVM) και λογιστικής παλινδρόμησης (LR). Τα αποτελέσματα των αλγορίθμων τους έδειξαν πως τα συνελκτικά τεχνητά νευρωνικά δίκτυα είχαν σταθερά μεγαλύτερη ακρίβεια και ήταν ικανά να εκπαιδευτούν από tweets προηγούμενων κρίσεων ώστε στη συνέχεια να μπορούν να αναγνωρίσουν επερχόμενες κρίσεις. Παρόλα αυτά απαιτούσαν και περισσότερο χρόνο. [40]

Από μια διαφορετική οπτική γωνία, ο κ. Abel και οι συνεργάτες του κατασκεύασαν ένα εργαλείο εμπλουτισμού γνώσης και δημιουργίας μεταδεδομένων χρησιμοποιώντας κλασσικά εργαλεία μηχανικής μάθησης και μοντέλα επεξεργασίας φυσικής γλώσσας (NPL) από δημοσιεύσεις στο Twitter για καταστάσεις έκτακτης ανάγκης στην Δανία. Το πιο σημαντικό τους κομμάτι είναι η διασύνδεση των μεταδεδομένων και η κατηγοριοποίηση αυτών όπου εν συνεχεία διευκολύνει ραγδαία την αναζήτηση περαιτέρω πληροφοριών σχετικά με την υπό εξέταση κρίση. [41]

Ο κ. Cameron και οι συνεργάτες του, σε συνεργασία με την κυβέρνηση της Αυστραλίας και το κυβερνητικό κέντρο συντονισμού κρίσεων, κατασκεύασαν και δοκίμασαν ένα αυτοματοποιημένο εργαλείο συλλογής tweets πολλών επιπέδων με σκοπό την υποβοήθηση των αξιωματούχων στην εκτίμηση των επιπτώσεων μιας κρίσης και μεταγενέστερα της βέλτιστης πορείας ανάκαμψης με την μέθοδο των προβλέψεων μεταγενέστερων καταστάσεων από το εργαλείο. [42]

Πιο κοντά στη γειτονική πραγματικότητα, ο κ. Öztürk και ο συνεργάτης του χρησιμοποίησαν αλγορίθμους εξόρυξης κειμένου (TM) και μοντέλα επεξεργασίας φυσικής γλώσσας (NPL) από μηνύματα στο Twitter στα

τούρκικα και στα αγγλικά για αναλύσουν τα συναισθήματα των πολιτών της Τουρκίας σχετικά με την προσφυγική κρίση που μαστίζει τη χώρα από τη Συρία. Το όλο έργο το πραγματοποίησαν στην προγραμματιστική γλώσσα R και χρησιμοποιώντας βιβλιοθήκες της. [43]

Ο κ. Stieglitz και οι συνεργάτες του συλλέξαν μηνύματα από το Twitter σχετιζόμενα με το σκάνδαλο Dieselgate της VW για να αναλύσουν τα συναισθήματα και το ύφος χρησιμοποιώντας το εργαλείο SentiStrength καθώς επίσης και τις αντιδράσεις των πολιτών του κόσμου σχετικά με τις αποκαλύψεις και την έκταση του σκανδάλου ανά την υφήλιο. Στη συνέχεια ανέλυσαν στατιστικώς τα αποτελέσματα και τις περιπτώσεις κατευνασμού ή οργής έπειτα από ανακοινώσεις τις εταιρείας VW και των πορισμάτων των ερευνών. [44]

Ο κ. Imran και οι συνεργάτες του συλλέξαν δεδομένα από 19 διαφορετικές κρίσεις μέσω τις πλατφόρμας AIDR(και άλλων) από το αναρτήσεις χρηστών στο Twitter. Στη συνέχεια τα κατηγοριοποίησαν μέσω των αλγορίθμων μηχανικής εκμάθησης Naive Bayes (NB), μηχανικής διανυσμάτων υποστήριξης (SVM) και τυχαία δάση (RF) και δημιούργησαν τις διανυσματικές παραστάσεις λέξεων με τη χρήση της δημοφιλούς εφαρμογής word2vec και της αρχιτεκτονικής Continuous Bag of Words (CBOW). [45]

Σε μια πιο πολυδιάστατη μελέτη ο κ. Coppersmith και οι συνεργάτες του μελετούν τη χρήση εργαλείων μηχανική εκμάθησης, συνελκτικών τεχνητών νευρωνικών δικτύων (CNN) και επεξεργασίας φυσικής γλώσσας δημοσιεύσεων στο Facebook για την έγκαιρη αποτροπή αυτοκτονιών ειδοποιώντας τις αρχές. Χρησιμοποίησαν την εφαρμογή Samaritans' Radar App η οποία αναγνωρίζει χρήστες που σχεδιάζουν να αυτοκτονήσουν και έπειτα ενημερώνει τους φίλους του χρήστη για την επικείμενη πράξη. Επιπλέον μελετούν και την εξίσου σημαντική ηθική πλευρά της επεξεργασίας δεδομένων και της αυτοματοποιημένης επέμβασης στις ζωές των υπό εξέταση ανθρώπων. [46]

Σε παρόμοιο μήκος κύματος ο κ. Walsh και οι συνεργάτες του εφάρμοσαν αλγορίθμους μηχανικής εκμάθησης και κατηγοριοποίησης σε ιατρικά δεδομένα κλινικά ψυχολογικών ασθενών με παρελθόν στον επιθυμητό αυτοτραυματισμό με στόχο την μελλοντική πρόβλεψη επιχειρήσεων αυτοκτονιών. Χρησιμοποίησαν αρκετές βιβλιοθήκες της Python και της R καθώς επίσης και τυχαία δάση (RF) και λογιστική παλινδρόμησης(LR) σε αυτά τα δεδομένα. Εξέτασαν τους αλγορίθμους αυτούς αναφορικά με την επιτυχία τους με την στατιστική καμπύλη αστερισμού (AUROC). Τα ευρήματα έδειξαν σημαντικά μεγάλη βελτίωση στην πρόωρη αποτροπή αυτοκτονιών και βελτιώνονται ακόμα περισσότερο με το πέρασμα του χρόνου. [47]

Twitter as a Lifeline: Human-annotated Twitter Corpora for NLP of Crisis-related Messages

Όπως προαναφέρθηκε, ο κ. Imran και οι συνεργάτες του συλλέξαν δεδομένα από 19 διαφορετικές κρίσεις που συνέβησαν μεταξύ του 2013 και του 2015, με το τελικό ποσό των συλλεγμένων tweet να ανέρχεται στα 52 εκατομμύρια που είναι σχετικά με κρίσεις. Με όλα τα tweet κατασκεύασαν διανυσματικές παραστάσεις λέξεων με την εφαρμογή word2vec. Οι κρίσεις με τα αντίστοιχα tweet που συλλέχθηκαν φαίνονται στον πίνακα παρακάτω.

Crisis type	Crisis name	Country	Language	# of Tweets	Start-date	End-date
Earthquake	Nepal Earthquake	Nepal	English	4,223,937	2015-04-25	2015-05-19
Earthquake	Terremoto Chile	Chile	Spanish	842,209	2014-04-02	2014-04-10
Earthquake	Chile Earthquake	Chile	English	368,630	2014-04-02	2014-04-17
Earthquake	California Earthquake	USA	English	254,525	2014-08-24	2014-08-30
Earthquake	Pakistan Earthquake	Pakistan	English	156,905	2013-09-25	2013-10-10
Typhoon	Cyclone PAM	Vanuatu	English	490,402	2015-03-11	2015-03-29
Typhoon	Typhoon Hagupit	Phillippines	English	625,976	2014-12-03	2014-12-16
Typhoon	Hurricane Odile	Mexico	English	62,058	2014-09-15	2014-09-28
Volcano	Iceland Volcano	Iceland	English	83,470	2014-08-25	2014-09-01
Landslide	Landslides worldwide	Worldwide	English	382,626	2014-03-12	2015-05-28
Landslide	Landslides worldwide	Worldwide	French	17,329	2015-03-12	2015-06-23
Landslide	Landslides worldwide	Worldwide	Spanish	75,244	2015-03-12	2015-06-23
Floods	Pakistan Floods	Pakistan	English	1,236,610	2014-09-07	2014-09-22
Floods	India Floods	India	English	5,259,681	2014-08-10	2014-09-03
War & conflict	Palestine Conflict	Palestine	English	27,770,276	2014-07-12	2014-10-02
War & conflict	Peshawar Attack Pakistan	Pakistan	English	1,135,655	2014-12-16	2014-12-28
Biological	Middle East Respiratory Syndrome	Worldwide	English	215,370	2014-04-27	2014-07-14
Infectious disease	Ebola virus outbreak	Worldwide	English	5,107,139	2014-08-02	2014-10-27
Airline accident	Malaysia Airlines flight MH370	Malaysia	English	4,507,157	2014-03-11	2014-07-12

Εικόνα 28: 18 dataset κρίσεων με λεπτομέρειες του τύπου της κρίσης, του ονόματος, της γλώσσας των μηνυμάτων, της χώρας αναφοράς, του αριθμού των tweet.

(Πηγή: [45])

Από αυτά περίπου 50.000 tweet κατηγοριοποιήθηκαν από εθελοντές & πληρωμένους εργάτες από το CrowdFlower ανάλογα το περιεχόμενό τους που αποφάνθηκαν ότι έχουν. Τουλάχιστον 3ς διαφορετικοί εργάτες έπρεπε να συμφωνούν σε μια κατηγορία ώστε επιλεγθεί εκείνη. Οι κάθε εργάτες ολοκλήρωναν έως 200 tweet και στην συνέχεια παύονταν από το project και ένας καινούργιος εργάτης μάθαινε λεπτομερώς την διαδικασία και το περιεχόμενο που πρέπει να έχει ένα tweet για να ανήκει σε μια κατηγορία και ύστερα ξεκινούσε να αποδίδει κατηγορίες μέχρι κι αυτός να κατηγοριοποιήσει 200 tweet. Οι κατηγορίες μαζί με το συμπέρασμα που πρέπει να αποδίδουν φαίνονται στον παρακάτω πίνακα και είναι με βάσει τις οδηγίες του γραφείου Συντονισμού Ανθρωπιστικών Υποθέσεων των Ηνωμένων Εθνών(UN-OCHA):

Πιθανές κατηγορίες tweet	
<i>Injured or dead people</i>	Reports of casualties and/or injured people due to the crisis
<i>Missing, trapped, or found people</i>	Reports and/or questions about missing or found people
<i>Displaced people and evacuations</i>	People who have relocated due to the crisis, even for a short time (includes evacuations)
<i>Infrastructure and utilities damage</i>	Reports of damaged buildings, roads, bridges, or utilities/services interrupted or restored
<i>Donation needs or offers or volunteering services</i>	Reports of urgent needs or donations of shelter and/or supplies such as food, water, clothing, money, medical supplies or blood; and volunteering services
<i>Caution and advice</i>	Reports of warnings issued or lifted, guidance and tips
<i>Sympathy and emotional support</i>	Prayers, thoughts, and emotional support
<i>Other useful information</i>	Other useful information that helps understand the situation
<i>Not related or irrelevant</i>	Unrelated to the situation or irrelevant

Πίνακας 4: Πιθανές κατηγορίες tweet και η περιγραφή τους.

(Πηγή: [45])

Στη συνέχεια βρήκαν όρους που είναι εκτός λεξιλογίου(out-of-vocabulary – OOV), όπως αργκό(π.χ. brb=be right back, pls=please), αναφορές χρηστών στο twitter(@), συντομογραφίες & φωνητικές συντομογραφίες(π.χ. y=yes, 4ever=forever), ορθογραφικά λάθη(π.χ. earthquak=earthquake, missin=missing), λέξεις χωρίς κενά(π.χ. prayfornepal=pray for Nepal, weneedshelter=we need shelter) κτλ. και κατασκεύασαν ένα OOV λεξικό το οποίο περιείχε τον όρο και δίπλα την διόρθωση αυτού. Στη συνέχεια διόρθωσαν όλα τα tweet με βάσει το λεξικό αυτό. Ένα στιγμιότυπο του λεξικού φαίνεται στον παρακάτω πίνακα:

Όροι μαζί με την διόρθωση που αντιστοιχούν	
2dae	Today
ACPT	Accept
grappler	rapper
fench	french
scen	scene
ledgend	legend
harshh	harsh
dswd	department of social welfare and development
WTF	What the *freak*
QIK	Quick

Πίνακας 5: Στιγμιότυπο OOV dictionary - όροι(αριστερά) μαζί με την διόρθωση(δεξιά) που αντιστοιχούν

(Πηγή: [45])

Από τις 18 κρίσεις, επέλεξαν 8 που προσφέρονται στα αγγλικά και έχουν κατηγοριοποιημένα tweet από πληρωμένους εργάτες.

Στη συνέχεια έκαναν preprocessing. Αφαίρεσαν stop-words, URL, αναφορές χρηστών και πραγματοποίησαν stemming στα αγγλικά. Έπειτα πραγματοποίησαν feature selection με IG στα κορυφαία 1.000 χαρακτηριστικά.

Τέλος με τα επιλεγμένα feature εκπαιδύσαν 3ς classifiers, SVM, NB και RF. Η εκπαίδευση & η δοκιμή έγιναν με 10-fold cross validation(σ.σ. περισσότερα αναφέρονται στην υλοποίηση) και ως μέτρηση είχαν το ROC-AUC. Οι κρίσεις και τα αποτελέσματα αυτών φαίνονται στον παρακάτω πίνακα:

Datasets	Classifier	Caution and advice	Displaced people and evacuations	Donation needs or offers	Infrastructure and utilities damage	Injured or dead people	Missing trapped or found people	Sympathy emotional support	Other useful information	Not related or irrelevant
2014 Chile earthquake	Size(%)	15%	2.80%	0.76%	1.70%	5.60%	0.54%	25%	30%	19%
	SVM	0.87	0.89	0.57	0.90	0.97	0.23	0.93	0.86	0.93
	NB	0.86	0.93	0.78	0.88	0.97	0.64	0.93	0.87	0.95
	RF	0.83	0.86	0.67	0.74	0.96	0.46	0.94	0.86	0.92
2015 Nepal earthquake	Size(%)	2.10%	3.10%	28%	4.50%	11%	5.80%	17%	22%	6.50%
	SVM	0.47	0.80	0.89	0.85	0.95	0.86	0.88	0.76	0.75
	NB	0.68	0.82	0.91	0.90	0.95	0.89	0.91	0.79	0.84
	RF	0.56	0.73	0.89	0.74	0.94	0.87	0.89	0.76	0.75
2013 Pakistan earthquake	Size(%)	6.30%	0.82%	15%	2%	17%	0.49%	5.60%	35%	18%
	SVM	0.77	0.80	0.92	0.76	0.95	0.63	0.82	0.84	0.84
	NB	0.82	0.87	0.94	0.91	0.93	0.74	0.83	0.84	0.84
	RF	0.68	0.70	0.92	0.77	0.95	0.69	0.78	0.88	0.83
2015 Cyclone Pam	Size(%)	7%	3.10%	17%	11%	7.20%	1.30%	5%	25%	24%
	SVM	0.76	0.80	0.92	0.85	0.95	0.39	0.66	0.77	0.90
	NB	0.79	0.82	0.92	0.86	0.97	0.56	0.79	0.80	0.94
	RF	0.68	0.80	0.90	0.80	0.95	0.47	0.71	0.79	0.92
2014 Typhoon Hagupit	Size(%)	20%	6.60%	5.50%	5.10%	3%	0.58%	13%	33%	13%
	SVM	0.74	0.95	0.88	0.76	0.94	0.44	0.92	0.74	0.81
	NB	0.75	0.96	0.89	0.82	0.96	0.57	0.92	0.78	0.81
	RF	0.71	0.97	0.84	0.73	0.94	0.58	0.91	0.75	0.80
2014 India floods	Size(%)	3.60%	1.40%	2.60%	4.30%	47%	0.87%	1.30%	14%	25%
	SVM	0.82	0.80	0.92	0.92	0.97	0.66	0.63	0.87	0.97
	NB	0.89	0.92	0.93	0.90	0.93	0.79	0.83	0.89	0.98
	RF	0.83	0.79	0.86	0.87	0.97	0.66	0.65	0.91	0.96
2014 Pakistan floods	Size(%)	3.90%	6.20%	25%	5.40%	13%	6.40%	6%	32%	2.30%
	SVM	0.71	0.84	0.82	0.77	0.94	0.85	0.88	0.74	0.47
	NB	0.83	0.80	0.85	0.79	0.94	0.85	0.89	0.77	0.65
	RF	0.72	0.80	0.87	0.78	0.95	0.84	0.86	0.79	0.59
2014 California earthquake	Size(%)	6.30%	0.48%	4.30%	18%	10%	0.51%	4.10%	47%	9.40%
	SVM	0.84	0.54	0.93	0.88	0.97	0.62	0.84	0.77	0.72
	NB	0.88	0.57	0.94	0.86	0.97	0.79	0.90	0.78	0.77
	RF	0.81	0.49	0.87	0.89	0.98	0.57	0.88	0.81	0.77

Πίνακας 6: Αποτελέσματα κατηγοριοποίησης με δείκτη την ROC-AUC σε 8 επιλεγμένα dataset με SVM, NB και RF classifiers.

(Πηγή: [45])

Για την παρούσα εργασία δηλώνεται ξεκάθαρα πως, η χρήση αυτής της έρευνας γίνεται με την συμμόρφωση των όρων χρήσης που παρέχονται με το dataset, όπως προσδιορίζονται & απαιτούνται στο Παράρτημα Α' από τους ίδιους τους συντάκτες.

Ανέλεγκτη φονική δασική πυρκαγιά στο Μάτι & στην Κινέτα Αττικής

Οι πυρκαγιές στο Μάτι & στην Κινέτα Αττικής έμειναν ανεξίτηλες στη μνήμη των Ελλήνων εξαιτίας του ατυχήματος της πρόκλησης των δύο από ανθρώπινο παράγοντα αλλά και της ιδιαίτερα φονικής πυρκαγιάς στο Μάτι αφήνοντας πίσω 102 νεκρούς και 187 τραυματίες. Όλα τα ελληνικά κανάλια και μέσα ενημέρωσης κάλυψαν το γεγονός τότε. Μάλιστα το ΣΚΑΙ αφιέρωσε το 2019 ντοκιμαντέρ για την ολέθρια αυτή πυρκαγιά. Επιπλέον, ξένα κανάλια, όπως το BBC News, ABC News, HuffPost UK, Euronews, TRT World, Daily Mail, κ.α. αναφέρθηκαν στο γεγονός αυτό με εκτενή ρεπορτάζ.

Το Μάτι αποτελεί οικισμό στο δήμο Μαραθώνος της Ανατολικής Αττικής στη Νέα Μάρκη. Την ονομασία του έλαβε, πιθανώς, λόγω της επιβλητικής θέας προς τη θάλασσα. Κατοικήθηκε την δεκαετία του 1960 στην αρχή με εξοχικές κατοικίες και από την δεκαετία του 2000 ο οικισμός σημείωσε μεγάλη ανάπτυξη, έγινε πολύ περισσότερο πυκνοκατοικημένος, χωρίς να χάσει ποτέ τον παραθεριστικό του χαρακτήρα. Σύμφωνα με την απογραφή πληθυσμού του 2011, η Νέα Μάρκη έχει πληθυσμό 15.554.



Εικόνα 29: Μάτι Αττικής πριν & μετά την πυρκαγιά(2018).

(Πηγή: <https://slpress.gr/wp-content/uploads/2018/07/maxresdefault-31.jpg>)

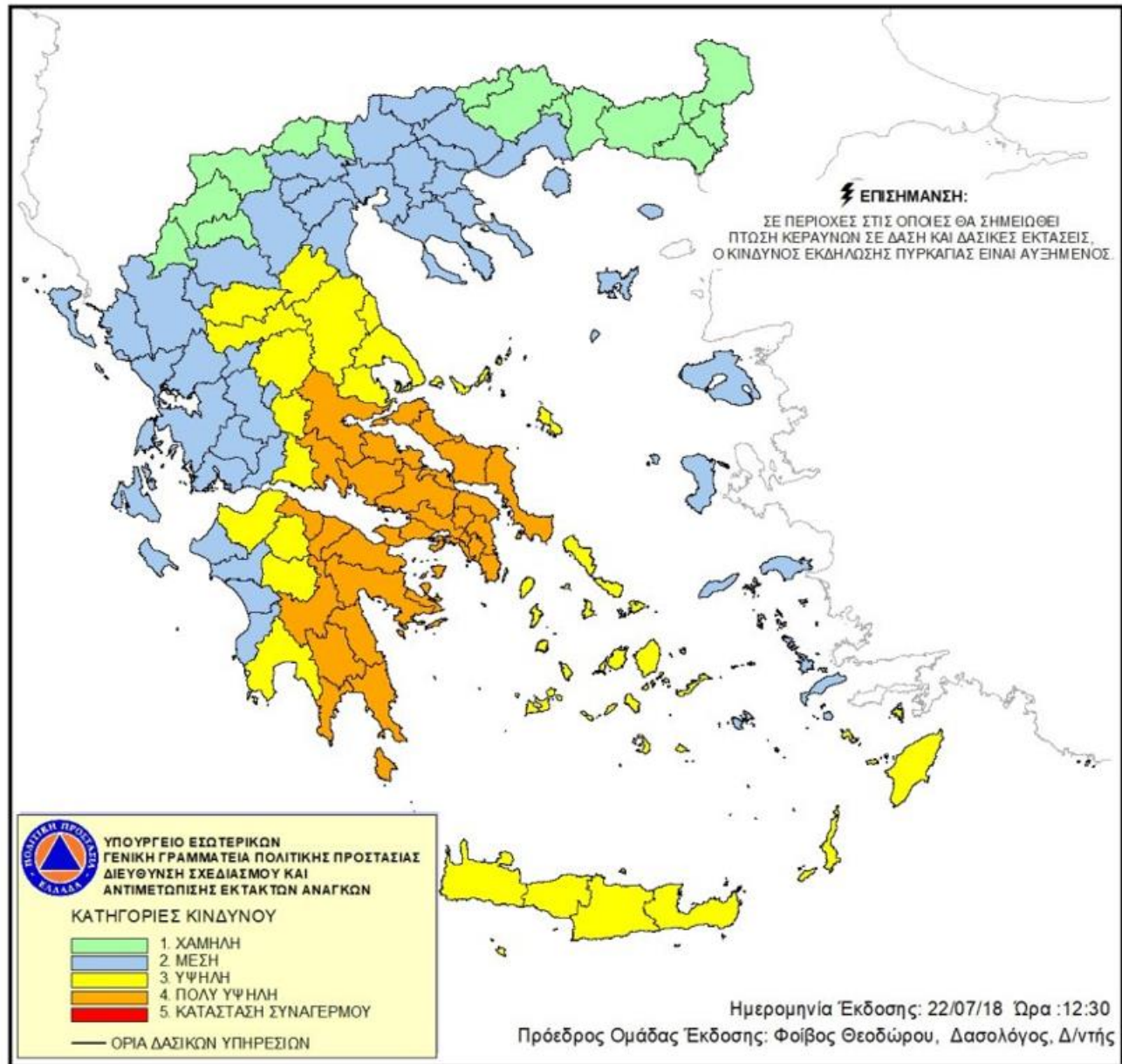
Συμβάν

Την Δευτέρα στις 23 Ιουλίου του 2018, ώρα 12:03 ξέσπασε πυρκαγιά στην Κινέτα από βλάβη στα καλώδια της ΔΕΗ. Την ίδια στιγμή λίγο αργότερα 23 Ιουλίου του 2018, ώρα 16:41, ξέσπασε πυρκαγιά, κοντά στο Νταού Πεντέλης, βόρεια της κεντρικής πλατείας του χωριού. Η πυρκαγιά αυτή ξεκίνησε από 65χρονο ιδιώτη ο οποίος αγνοώντας τις προειδοποιήσεις για την επικινδυνότητα ξεσπάσματος πυρκαγιάς, επιχείρησε να κάψει ξερά κλαδιά και ξύλα. Θεωρώντας πως είχε σβήσει την φωτιά, έφυγε από την τοποθεσία, αφήνοντας την πυρκαγιά να εξελιχθεί.

Στην αρχή η φωτιά απλωνόταν με φυσιολογικούς ρυθμούς, ωστόσο, σύντομα ο άνεμος σημείωσε απότομη αύξηση, κατευθυνόμενος σε οικισμούς στο Μάτι, με αποτέλεσμα η πυρκαγιά να βγει γρήγορα εκτός ελέγχου. Το πυροσβεστικό φυλάκιο του δήμου Ραφήνας παρατηρεί τον καπνό, καλεί το κέντρο της πυροσβεστικής και κινητοποιούνται οχήματα. Οι πυροσβέστες δίνουν μάχη με την φωτιά προσπαθώντας να μην την αφήσουν να περάσει την λεωφόρο Μαραθώνος, καθώς θα χανόταν ο έλεγχος. Εναέρια μέσα δυσκολεύονται να απογειωθούν λόγω των ισχυρών ανέμων και επικρατεί σύγχυση στην επικοινωνία των πυροσβεστών τόσο μεταξύ τους όσο και με τις άλλες υπηρεσίες. Οι κάτοικοι στο Μάτι δεν γνωρίζουν τίποτα για την πυρκαγιά. Έπειτα, η φωτιά περνάει την λεωφόρο Μαραθώνος, γίνεται μη ελέγξιμη και είναι πλέον εμφανές ότι πρόκειται να βρεθούν σε κίνδυνο, όχι μόνο κατοικίες, αλλά και ανθρώπινες ζωές. Έτσι, ξεκινούν οι προσπάθειες εκκένωσης ιδρυμάτων και κατασκηνώσεων. Απεγνωσμένοι κάτοικοι που βρίσκονται εγκλωβισμένοι στα σπίτια τους καλούν το κέντρο. Πυροσβεστικά οχήματα και αμάξια με ανθρώπους εγκλωβίζονται μέσα στην φωτιά. Γίνεται προσπάθεια να κινητοποιηθούν αστυνομικοί, προκειμένου να απεγκλωβίσουν ανθρώπους από σπίτια. Όταν φτάνουν ασθενοφόρα και αστυνομικοί, αρνούνται να μπουν σε περιοχές που βρισκόταν εγκλωβισμένοι άνθρωποι, λόγω του μεγάλου βαθμού επικινδυνότητας. Αρκετοί άνθρωποι κατευθύνονται

προς την παραλία, όπου η ατμόσφαιρα σταδιακά γίνεται αποπνικτική. Ο σύλλογος καταδρομέων επικοινωνεί με την πυροσβεστική, από την οποία λαμβάνει εντολή να απεγκλωβίσει όσο το δυνατόν περισσότερα άτομα μπορεί από το Μάτι και το Κόκκινο Λιμανάκι. Στις 23:00, ο πρωθυπουργός φτάνει στο συντονιστικό κέντρο της πυροσβεστικής. Στις 01:00 εντοπίζονται τα πρώτα πτώματα από πυροσβέστες στο Κόκκινο Λιμανάκι.

ΧΑΡΤΗΣ ΠΡΟΒΛΕΨΗΣ ΚΙΝΔΥΝΟΥ ΠΥΡΚΑΓΙΑΣ ΠΟΥ ΙΣΧΥΕΙ ΓΙΑ Δευτέρα 23/07/18



Εικόνα 30: Χάρτης πρόβλεψης κινδύνου πυρκαγιάς για 23/07/2018.

(Πηγή: Υπηρεσία Πολιτικής Προστασίας Ελλάδος)

Ο αριθμός των νεκρών σημειώθηκε πως είναι 102. Πιο συγκεκριμένα, τα θύματα της ολέθριας αυτής πυρκαγιάς είναι τα εξής:

Δημητρίου (αβάπτιστο βρέφος), Αβραμίδου Αικατερίνη, Αθανασοπούλου Ζωή, Αλεξόπουλος Δημήτριος, Ανδρουλιδάκη Αιμιλία, Βονικόπουλος Ευθύμιος, Γαζεπίδου-Γαζέπη Τάνια, Γαλάνη Μαρία, Γιαννακοδήμος Βασίλειος, Γιαννακοπούλου Σοφία, Γιαννοπούλου Αγγελική, Δεσύπρη Θέκλα, Διονυσιώτη Μαργαρίτα, Δόξας

Παναγιώτης, Δόξας Αργύριος, Δρούβης Χαράλαμπος, Ζαφειρίου Ευστράτιος, Θάνος Ιωάννης, Κάκκαρης Σπυρίδων, Καλαμπανίδα Μαρία, Καλαμπανίδα Μαριάνθη, Καλεγιαννάκος Ιωάννης, Καλεγιαννάκου Παρασκευή, Καλεγιαννάκος Νικόλαος, Κανδαλέπα Σμαράγδα, Καραγκιαούρη Γαρυφαλλιά, Καραλή Αθηνά, Κατσαργύρης Βασίλειος, Κατσουλάκης Μάρκος, Κατσουλάκης Πάρης, Κέρτσος Κωνσταντίνος, Κλάδιος Ιάκωβος, Κλάδιου Μαρία, Κλάδιος Ιωάννης, Κοκκινίδη Πιπίνα, Κοκκινίδης Παναγιώτης, Κοκκινίδης Γεώργιος, Κοκκινίδης Δημήτριος, Κοκκόνης Επαμεινώνδας, Κόσσορας Νικόλαος, Κούκλα – Πολυχρονίδου Μαρία, Κούκλας Χρηστος, Κωνσταντάκη Μαριάνθη, Κωστοπούλου Βασιλεία, Λαμπρίδης Μιχαήλ, Μάσχα Αγγελική, Μάσχας Άγγελος, Μαυρίδου Βασιλεία, Μίχας Βίκτωρας, Μόσχου Μαρία, Μωραΐτη Ασπασία, Νιτσοτόλη Κατίνα, Ντούτσα (Γαβάκη) Νικολέτα, Ορφανού Κωστούλα, Παγωμένου Μαρία, Παλιούρα Βασιλική, Παντελιδάκης Στυλιανός, Παπαιωάννου Ιωάννης, Παπαιωάννου Παναγιώτης, Παπαιωάννου Ζόγκα Φλώρα, Παπαποστόλου Σπυρίδων, Πατελάρου Μοίρα, Πεταλά Ευαγγελία, Πεταλάς Διονύσιος, Πλυμάκης Λεωνίδας, Πολίτου Ευγενία, Πολυπόρτης Μιχαήλ, Σαριέβα Ελένη, Σιαπέρα Ευαγγελία, Σιαπκάρα Καλλιόπη, Σιδέρη Ελένη, Σκαραμαγκά Ζαμπέτα, Σκαραμαγκάς Δημήτριος, Σπανού Κασσιανή, Σπηλιώτη Χρύσα, Σπυρίδης Σπύρος, Σπυροπούλου Θάλεια, Τζούλιας Δημήτριος, Τουρναβίτης Δημήτριος, Τσαούση Θεοδώρα, Τσαρμπός Παύλος, Τσέκος Βίκτωρας, Φιλιππόπουλος Φίλιππος, Φιλιπποπούλου Σοφία, Φιλιπποπούλου Σοφία, Φιλιπποπούλου Βασιλική, Φύτρος Ανδρέας, Φύτρος Γρηγόριος, Φύτρου Εβίτα, Χαμηλοθώρης Παναγιωτης, Χαρδαλούπα Ελισαβετ, Χατζόπουλος Νικόλαος, Χερουβείμ Εβελίνα, Χερουβείμ Ευπραξία, Χερουβείμ Μαρίλια, Χερουβείμ Βασιλική, Χριστοφίδου Στυλιανή, Bonte Dirk, Korzeniowska-Beata Terasa, Korzeniowski Kacper Nikolaj, O'Callaghan-Westropp Brian.

Επακόλουθα

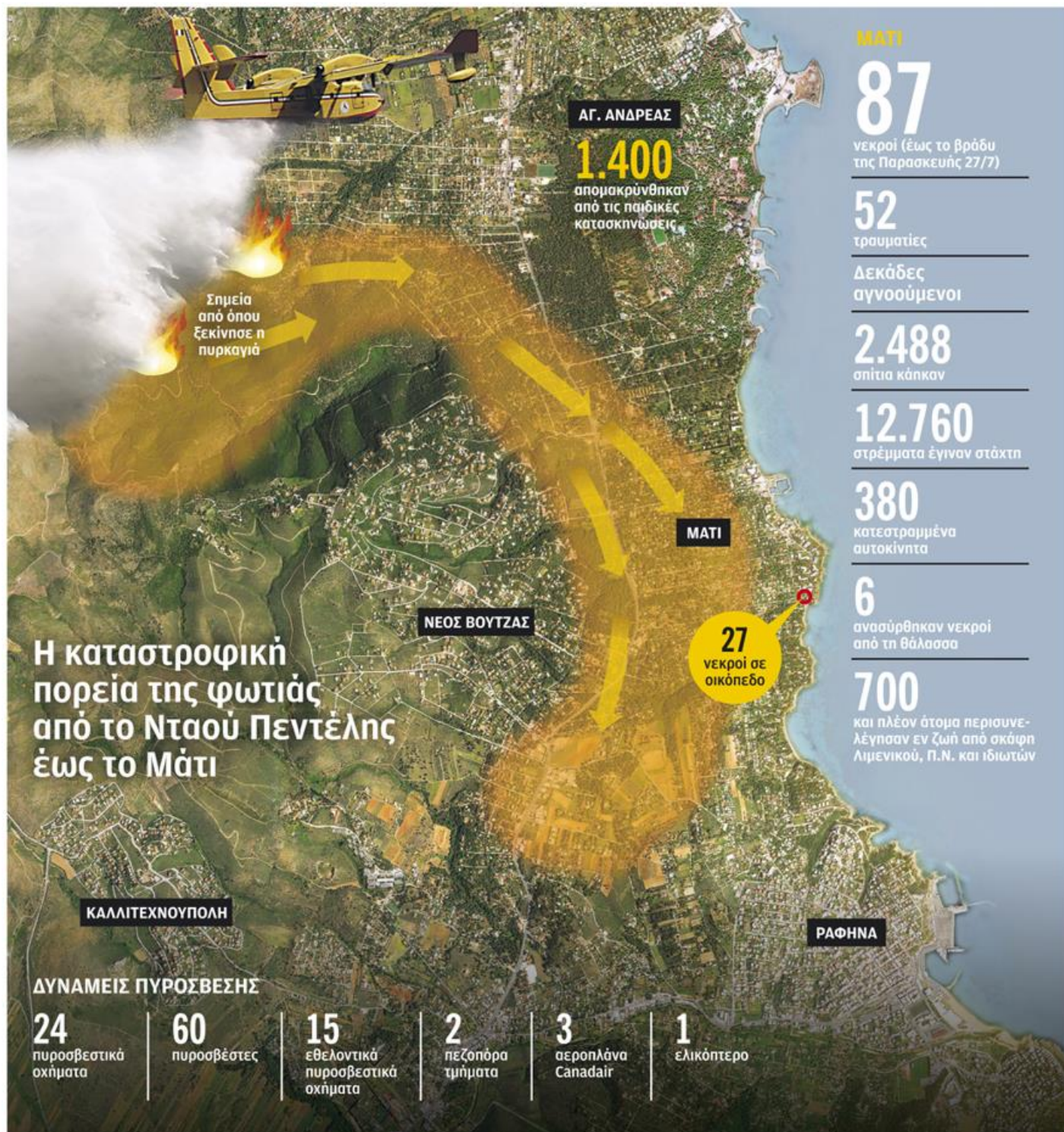
Η καταστροφική αυτή πυρκαγιά είχε ως αποτέλεσμα να χάσουν την ζωή τους 102 άνθρωποι, να καταστραφούν χιλιάδες κατοικίες και οχήματα, να αποτεφρωθούν τεράστιες εκτάσεις πευκοδασών και να σημειωθούν ζημιές στο δίκτυο ηλεκτροδότησης, τηλεπικοινωνιών και ύδρευσης της περιοχής.

Την βοήθεια τους πρόσφεραν οι εξής χώρες: Αλβανία, Αυστραλία, Αρμενία, Βουλγαρία, Γαλλία, Γερμανία, ΗΠΑ, Ισπανία, Ιταλία, Ισραήλ, Κροατία, Κύπρος, Λιθουανία, Βόρεια Μακεδονία, Πολωνία, Πορτογαλία, Ρουμανία, Ρωσία, Σερβία και Τουρκία.

Τον Μάρτιο του 2019 η Εισαγγελία Αθήνας άσκησε ποινική δίωξη για εμπρησμό από αμέλεια, ανθρωποκτονία από αμέλεια κατά συρροή και σωματική βλάβη κατά συρροή κατά του εξηνταπεντάχρονου που ξεκίνησε την φωτιά και ποινική δίωξη για ανθρωποκτονία από αμέλεια κατά συρροή και σωματική βλάβη κατά συρροή κατά 19 ατόμων (μερικοί από τους οποίους ήταν ο τότε υπαρχηγός του Πυροσβεστικού Σώματος και ο δήμαρχος Μαραθώνα).

Μέχρι και σήμερα, το 2020, γίνονται έρευνες για την πιθανόν προσπάθεια πολιτικών να αποκρύψουν τις διαστάσεις του γεγονότος και τις ευθύνες των ιδίων σε μια προσπάθεια απόδοσης δικαιοσύνης.

Η ειδησεογραφική βιβλιογραφία των γεγονότων βρίσκεται στο παράτημα Β'.



Η ΚΑΘΗΜΕΡΙΝΗ

Εικόνα 31: Πορεία της πυρκαγιάς στο Μάτι.

(Πηγή: Η Καθημερινή)

Υλοποίηση

Από την περιγραφή των δύο περιπτώσεων και με βάσει την θεωρία που έχει αναλυθεί στην πρώτη ενότητα καταλήγουμε στα εξής συμπεράσματα:

1. Είναι πρόβλημα ποιοτικών αδόμητων δεδομένων big data όπου με dimensionality θα τα μετατρέψουμε σε δομημένα, κάνοντας text mining και ANI supervised ML.
2. Σε αυτά τα data κειμένου θα πρέπει να γίνει πολυεπίπεδο preprocessing με NLP το οποίο ένα κομμάτι θα είναι language dependent και το άλλο θα είναι language independent.
3. Θα πρέπει να γίνει feature selection με IG με $k=1.000$.
4. Θα πρέπει να γίνει 10-fold CV με classifiers με default παραμέτρους τους SVM, MNB και RF.

Επιπλέον, για το ελληνικό κομμάτι και μόνο, θα πρέπει να γίνει IF όσα περισσότερα tweet γίνεται σχετικά με το συμβάν.

Τα βήματα προσέγγισης προβλημάτων ML είναι συνήθως:

1. Συλλογή δεδομένων(data collection)
2. Ανάλυση δεδομένων(data preprocessing)
3. Δημιουργία αλγορίθμων(algorithm development)
4. Έλεγχος αλγορίθμων(algorithm testing)
5. Σχολιασμός αποτελεσμάτων(results conclusion)

Ολόκληρος ο φάκελος του έργου με τα dataset και τον κώδικα βρίσκεται στο OneDrive [εδώ](#).

Προδιαγραφές, Framework & βιβλιοθήκες

Ο προγραμματισμός θα γίνει σε γλώσσα Python 3.8.5, σε απομονωμένο περιβάλλον Conda, καθώς είναι μια μινιμαλιστική και εύκολη στην κατανόηση γλώσσα με πλούσιο οπλοστάσιο ML και NLP, και πληθώρα υποστηρικτικών βιβλιοθηκών & framework. Αναφορικά τα πιο σημαντικά που θα χρησιμοποιηθούν είναι τα:

- **Tweepy**: την χρήση του API του Twitter στην python
- **Numpy**: για προηγμένες μαθηματικές πράξεις
- **Pandas**: για φόρτωση των δεδομένων σε DataFrames
- **Matplotlib**: για την απεικόνιση των μετρήσεων σε σχήματα
- **NLTK**: για το preprocessing
- **ScikitLearn**: για το feature selection & ML

Οι κώδικες για κάθε βήμα θα βρίσκονται στην οριστική τους μορφή στα παραρτήματα, με αρκετά σχόλια όπου χρειάζεται και παράλληλα θα γίνεται αναλυτική επεξήγησή τους σε κάθε βήμα της υλοποίησης.

Δημιουργία περιβάλλοντος

Έχοντας ορίσει πλέον μια ξεκάθαρη στρατηγική προσέγγισης, το πρώτο βήμα με το οποίο ξεκινάμε είναι να δημιουργήσουμε το περιβάλλον εργασίας. Μια καλή στρατηγική είναι να χρησιμοποιήσουμε την γλώσσα Python, μια γλώσσα υψηλού επιπέδου και γενικού σκοπού με ισχυρό οπλοστάσιο data science και AI.

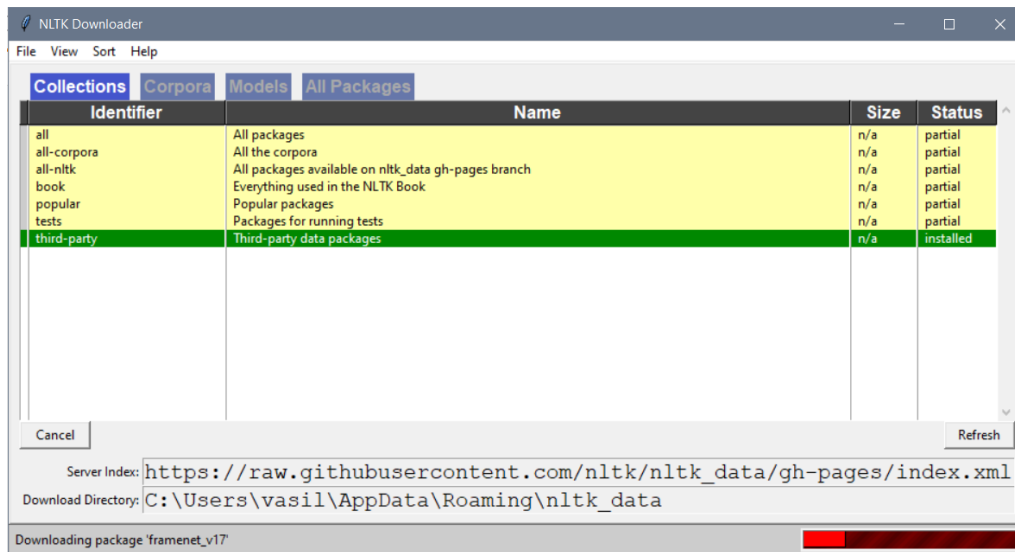
Δημιουργήθηκε το 1991 και δυστυχώς, όπως είναι η περίπτωση με τις περισσότερες γλώσσες που έχουν δημιουργηθεί πριν την νέα χιλιετία(πλην της την C και Java), χρησιμοποιεί μόνο έναν πυρήνα για να εκτελέσει κώδικα, από την εποχή όπου οι υπολογιστές ήταν μονοπύρηνιοι, και εμείς σκοπεύουμε να την

χρησιμοποιήσουμε για έργο AI τα οποία ως γνωστών χρειάζονται αρκετό χρόνο να εκπαιδευτούν. Επειδή ο χρόνος είναι πολύτιμος και ο άνθρωπος δεν μπορεί να δημιουργήσει χρόνο, καλό θα ήταν να χρησιμοποιηθούν τεχνικές οι οποίες επιταχύνουν τον χρόνο εκτέλεσης, και μια εύκολη τακτική είναι να προσθέσουμε εντολές παράλληλης εκτέλεσης σε νήματα, όπως φαίνεται σε παραδείγματα [εδώ](#). Αλλά αυτό μάλλον στην περίπτωσή μας δεν θα ήταν συνετό γιατί θα χαλούσε τη ροή αποτελεσμάτων χωρίς να πρόσφερε ιδιαίτερη επιτάχυνση ώστε να αξίζει ο κόπος(σ.σ. θα εξηγήσω γιατί παρακάτω) οπότε το βήμα θα παραληφθεί.

Μια γρήγορη λύση θα ήταν να εγκαταστήσουμε την Python απευθείας στον υπολογιστή και να ξεκινήσουμε να εγκαθιστούμε πακέτα. Όμως αυτό δεν θα ήταν συνετό για τις μακροπρόθεσμες προοπτικές καθώς στην περίπτωση όπου θέλουμε να αναβαθμίσουμε πακέτα θα ήταν δύσκολο, δεδομένου ότι αλληλεξαρτώνται, αλλά και επίσης μπορεί να υπήρχε ασυμφωνία μεταξύ project(λόγου χάρη του αγγλικού κομματιού αυτής της διπλωματικής με του ελληνικού κομματιού).

Για τον λόγο αυτό και χωρίς άλλη καθυστέρηση θα χρησιμοποιήσουμε το Conda, το οποίο είναι ένα ΕΛ/ΛΑΚ σύστημα διαχείρισης πακέτων, κάλυψης εξαρτήσεων και διαχείρισης περιβαλλόντων για οποιαδήποτε γλώσσα προγραμματισμού, συμπεριλαμβανομένης και της Python που εμείς θα χρησιμοποιήσουμε. Το Conda μπορεί γρήγορα να εγκαθιστά, να εκτελεί και να ενημερώνει πακέτα. Επιπλέον μπορεί εύκολα και γρήγορα να δημιουργεί και να αλλάζει περιβάλλοντα εργασίας τα οποία είναι απομονωμένοι χώροι με συγκεκριμένη έκδοση python. Για αρχή λοιπόν, αφού έχουμε πρώτα εγκαταστήσει την τελευταία έκδοση Conda, δημιουργούμε ένα νέο περιβάλλον εκτελώντας την εντολή `conda create --name thesis python=3.8.5` στο τερματικό η οποία δημιουργεί ένα νέο περιβάλλον καθαρό περιβάλλον με την python 3.8.5. Στη συνέχεια ενεργοποιούμε αυτό το περιβάλλον εκτελώντας την εντολή `conda activate thesis` στο τερματικό και πλέον είμαστε σε θέση να ξεκινήσουμε να εγκαθιστούμε τα πακέτα μας. Για να μην μακρυγορούμε σε κάθε βήμα που κάνουμε, όλα τα πακέτα που θα χρειαστούμε είναι απαραίτητα και για το αγγλικό και για το ελληνικό κομμάτι και βρίσκονται στο παράρτημα Γ'. Μπορούμε να εγκαταστήσουμε όλα τα απαραίτητα πακέτα εκτελώντας τις δύο εντολές που φαίνονται στο παράρτημα Γ2' ή καλύτερα ενδείκνυται να εγκαταστήσουμε τις συγκεκριμένες εκδόσεις των πακέτων μέσω του Conda εκτελώντας την εντολή `conda install --yes --file requirements.txt` στο τερματικό, με το αρχείο `requirements.txt` να είναι το αρχείο που φαίνεται στο παράρτημα Γ1' και στην συνέχεια εκτελώντας και την επόμενη εντολή που φαίνεται στο τέλος του παρατήματος Γ1' στο τερματικό.

Προτού να είμαστε έτοιμοι πρέπει να εκτελέσουμε σε ένα άδειο αρχείο python την εντολή `nlTK.download()` για να κατεβάσουμε όλα τα απαραίτητα module του NLTK για να επεξεργαστούμε την φυσική γλώσσα των tweet όπως φαίνεται στην παρακάτω εικόνα.



Εικόνα 32: Διαδικασία λήψης όλων των απαραίτητων module του NLTK.

Twitter as a Lifeline: Human-annotated Twitter Corpora for NLP of Crisis-related Messages

Έχοντας ήδη αναφέρει την λογική προσέγγιση των ερευνητών θα επιχειρήσω να επαναλάβω την διαδικασία που αναφέρουν και αισιοδοξώ να βγάλω παρόμοια αποτελέσματα. Θα προσπαθήσω να τους μιμηθώ προσθέτοντας μερικές μικρές βελτιώσεις όπου μπορούν να εφαρμοστούν χωρίς να αποκλίνω από την διαδικασία που ακολούθησαν.² Εφόσον έχουμε ορίσει το περιβάλλον τότε χωρίς άλλη καθυστέρηση ξεκινάει η υλοποίηση με το πρώτο βήμα.

Συλλογή δεδομένων

Όπως προαναφέρθηκε, το πείραμά που κάνουν ο Imran και οι συνεργάτες του βασίζεται σε κατηγοριοποιημένα tweet από πληρωμένους εργάτες από 8 κρίσεις, όπως φαίνονται στον πίνακα 6.

Αυτό που πρέπει να γίνει είναι να κατεβάσουμε το dataset των κατηγοριοποιημένων δεδομένων από πληρωμένους εργάτες από [εδώ](#) και στην συνέχεια να συλλέξουμε αυτά τα 8 αρχεία που αντιπροσωπεύουν τις 8 κρίσεις που φαίνονται στον Πίνακα 6. Τα εν λόγω αρχεία φαίνεται να είναι:

1. 2013_Pakistan_eq_CF_labeled_data.csv
2. 2014_California_Earthquake_CF_labeled_data.csv
3. 2014_Chile_Earthquake_en_CF_labeled_data.csv
4. 2014_India_floods_CF_labeled_data.csv
5. 2014_Pakistan_floods_CF_labeled_data.csv
6. 2014_Philippines_Typhoon_Hagupit_en_CF_labeled_data.csv
7. 2015_Cyclone_Pam_en_CF_labeled_data.csv
8. 2015_Nepal_Earthquake_en_CF_labeled_data.csv

² Σε αυτό το σημείο πρέπει να αναφερθεί πως οι διαδικασίες που ακολούθησαν δεν περιγράφονται με ακρίβεια και υπάρχει γενικευμένη ασάφεια ως προς την σειρά των βημάτων που ακολούθησαν & ως προς τους συγκεκριμένους αλγορίθμους που χρησιμοποίησαν με τις συγκεκριμένες υπερπαραμέτρους (hyperparameters). Για τον λόγο αυτό, στην παρούσα υλοποίηση, θεωρείται πως η σειρά που ακολουθήθηκε είναι η αναγραφόμενη (δηλαδή με την σειρά που τα περιγράφουν) και πως σε όλους τους αλγορίθμους χρησιμοποιήθηκαν οι προεπιλεγμένες μεταβλητές χωρίς καμία παραμετροποίηση, καμία βελτιστοποίηση, κανένα hyperparameter GridSearch, κ.ο.κ..

Ένα στιγμιότυπο ενός αρχείου φαίνεται να περιέχει τις εξής ιδιότητες:

- **tweet_id**: είναι το αναγνωριστικό του tweet με βάσει την βάση δεδομένων του twitter. Εάν για παράδειγμα δοκιμάσει κανείς την διεύθυνση <https://twitter.com/user/status/504075795435094016> τότε θα πρέπει να το tweet με κείμενο «Bridges, overpasses earn approval after Northern California quake - SFGate <http://t.co/hjblw7aHiJ> #earthquakesf #NAPA #earthquake».
- **tweet_text**: είναι το κείμενο που αντιστοιχεί το στο tweet με το αναγραφόμενο tweet_id. Αυτή είναι μια σημαντική πληροφορία που, με βάσει αυτά που αναφέρθηκαν στην εισαγωγή, αντιπροσωπεύει το X στους classifiers.
- **label**: η κατηγορία στην οποία αποφάνθηκε ότι ανήκει, με βάσει τις οδηγίες του πίνακα 4. Αυτή είναι μια σημαντική πληροφορία που, με βάσει αυτά που αναφέρθηκαν στην εισαγωγή, αντιπροσωπεύει το Y στους classifiers.

tweet_id	tweet_text	label
'504075795435094016'	Bridges, overpasses earn approval after Northern California quake - SFGate http://t.co/hjblw7aHiJ #earthquakesf #NAPA #earthquake	infrastructure and utilities damage
'503765744333901824'	Say what you want, but the earthquake that hit California was all San Andrea's fault.	other useful information
'503741064067313665'	Cool data analysis of today's Napa earthquake & effect on sleep https://t.co/NkCM0SU0Yd by @Jawbone data engineer @eugmandel via @jwfbean	other useful information
'503910039715725312'	California assessing damage after large quake http://t.co/RTc6hTDSsO	infrastructure and utilities damage
'503892224589848576'	RT @heyyouapp: California USA Rancho Cucamonga http://t.co/TPSViMWZMs BREAKING NEWS 400 Earthquake Powerful Earthquake Slams San Francisco	caution and advice
'503653968279793664'	RT @GoodBelly: California Winemakers Cleaning up After Quake http://t.co/798bYeBWTI via @Reverb	other useful information
'504668324777967616'	RT @friend_beast: Oh, California RT @vickydnnguyen: Skaters finding upside to quake damage. Photo from #nbcbayarea photog Jeremy Carroll http://t.co/798bYeBWTI	missing trapped or found people
'503712067942350849'	USA California Hit With Earthquake Of 6.0 Magnitude (AP) An earthquake with a preliminary magnitude of ... http://t.co/cBMLXD30I7	other useful information
'503662312923086848'	RT @Zamaanie: Zionist US supplies Israel with WMD for Terrorism #Israel #Gaza #Palestine #USA #Ferguson #UK #US #napaquake #p2 http://t.co/798bYeBWTI	not related or irrelevant

Πίνακας 7: Στιγμιότυπο αρχείου 2014_California_Earthquake_CF_labeled_data.csv

Προεπεξεργασία δεδομένων

Όπως διαφαίνεται από τον πίνακα 7 τα δεδομένα έχουν αρκετό θόρυβο και πλεονάζοντα χαρακτηριστικά τα οποία και πρέπει να αφαιρεθούν με για να επιτύχουμε καλύτερο feature selection και να καρπώσουμε τα οφέλη αυτού, όπως περιγράφηκαν στην εισαγωγή.

Στα «στραβά» των dataset που χρίζουν διόρθωσης διακρίνουμε:

- συνδέσμους, όπως το <http://t.co/cBMLXD30I7> οι οποίοι δεν προσφέρουν κάποια χρήσιμη πληροφορία άρα αποτελούν θόρυβο και πρέπει να σβηστούν από το κείμενο
- δεσμευμένες λέξεις, όπως RT(που σημαίνει retweet) και FAV(που σημαίνει favorite) και δεν προσφέρουν κάποια χρήσιμη πληροφορία άρα αποτελούν θόρυβο και πρέπει να σβηστούν από το κείμενο
- επισημάνσεις χρηστών, όπως το @Zamaanie που δεν προσφέρουν κάποια χρήσιμη πληροφορία άρα αποτελούν θόρυβο και πρέπει να σβηστούν από το κείμενο
- hashtag, όπως το #earthquake τα οποία δυνητικά ΔΕΝ είναι άχρηστη πληροφορία οπότε μάλλον πρέπει να σβήσουμε την δέση (#) μπροστά από αυτές τις λέξεις και να κρατήσουμε την λέξη, στην προκύπτουσα περίπτωση το earthquake καθώς βοηθά στην κατηγοριοποίηση του tweet
- χαρακτήρες εκτός ASCII, όπως είναι το ÆÆ+Æ' USA ÆÆ+Æ' που δεν προσφέρουν κάποια χρήσιμη πληροφορία άρα αποτελούν θόρυβο και πρέπει να σβηστούν από το κείμενο
- διπλότυπα tweet, δηλαδή tweet που έχουν ακριβώς το ίδιο κείμενο, το οποίο έχει συνήθως προκύψει από ρομπότ ή από retweet και καλό θα ήταν αφαιρεθούν τα διπλότυπα τα οποία αποτελούν θόρυβο, γιατί κάνουν τους αλγορίθμους feature selection να τα θεωρούν ως σημαντικά αφού εμφανίζονται πολλές φορές αλλά επίσης μεγαλώνουν το dataset ανάξια και ως εκ τούτου καθυστερούν τους αλγορίθμους
- διπλότυπες εγγραφές, δηλαδή εγγραφές που είναι ίδιες, με ίδιους κωδικούς ID και για αυτό ευθύνεται η κακή δομή του dataset. Γενικά παρατηρώντας τα dataset που κατεβάσαμε στο προηγούμενο βήμα βλέπουμε πως δεν είναι πολύ προσεγμένο με χαρακτηριστικό παράδειγμα ένα dataset το οποίο περιέχει 50.000 εγγραφές με το ακριβώς ολόιδιο tweet. Για αυτό στην διαδικασία του preprocessing θα πρέπει πάντα να υπάρχει μέριμνα για «κακά» dataset, όπως αναφέρθηκαν στην εισαγωγή
- άδειες εγγραφές, δηλαδή twitter με κενό tweet_text που μπορεί να έχει προκύψει με κενή ανάρτηση από σφάλμα ή αναρτήσεις που περιέχουν μόνο emoji ή μόνο ένα URL και θα ήταν καλό να αφαιρεθούν επειδή μεγαλώνουν το dataset ανάξια και ως εκ τούτου καθυστερούν τους αλγορίθμους
- άνισες κλάσεις, όπως φαίνεται από στον πίνακα 6, όπου είναι ευδιάκριτο πως ορισμένες κλάσεις αντιπροσωπεύουν 35% του δείγματος ενώ άλλες μόνο δεκαδικές μονάδες αυτού. Για αυτό στην διαδικασία του preprocessing θα πρέπει μάλλον να κάνουμε κάτι με αυτό όπως να πάρουμε λιγότερα δείγματα από μια κλάση που έχει πολλά(under sampling) ή να πάρουμε περισσότερα από μια κλάση που έχει λίγα(over sampling) ή να παράγουμε μηχανικά νέα documents με τεχνικές data augmentation
- όροι OOV: όπως φαίνονται παραδείγματα στον πίνακα 5, και καλό θα ήταν αυτοί οι όροι να αντικατασταθούν από λέξεις με νόημα ώστε να διευκολύνουν το feature selection και τα σκορ
- όροι στίξης, όπως τελείες, κόμματα, κ.α. τα οποία δεν συνεισφέρουν θετικά στο dataset και πρέπει να αφαιρεθούν
- πολλαπλοί διαδοχικά κενοί χαρακτήρες, οι οποίοι εμφανίστηκαν είτε επειδή ο χρήστης τους πληκτρολόγησε ή μπορεί να εμφανιστούν κατά την αφαίρεση όρων σε ένα από τα παραπάνω βήματα και θα πρέπει να αφαιρεθούν για να μικρύνει το dataset αλλά και για όρους αισθητικής ώστε να προσιδιάζει το tweet σε κείμενο
- άχρηστες λέξεις(stopwords) όπως για παράδειγμα όροι and, the, is, κ.α., οι οποίοι δεν βοηθούν με κάποιον τρόπο το feature selection καθώς είναι παρόντες σχεδόν σε κάθε tweet και άρα κατά συνέπεια δεν βοηθούν με κάποιον τρόπο τον αλγόριθμο ML να διακρίνει την κλάση όπου ανήκει ένα tweet και θα πρέπει να αφαιρεθούν.
- εναλλαγή κεφαλαίων και μικρών χαρακτήρων, όπως για παράδειγμα οι όροι Northern και northern, οι οποίοι για τον άνθρωπο είναι ίδιοι όχι όμως και για τον αλγόριθμο ML και για τον λόγο αυτό πρέπει να κανονικοποιηθούν, δηλαδή να έρθουν στην ίδια μορφή, είτε μετατρέποντας όλα τα γράμματα σε

κεφαλαία είτε σε μικρά. Για την δική μας περίπτωση και για λόγους ευκολίας στο μάτι, όλοι οι όροι θα πρέπει να μετατραπούν σε μικρά γράμματα

- διαφορετικές μορφές συνώνυμων λέξεων, όπως για παράδειγμα όροι supply και supplied και supplying, κ.ο.κ., οι οποίοι για τον ίδιο λόγο με το προηγούμενο σημείο εκλαμβάνονται ως διαφορετικοί και θα πρέπει κανονικοποιηθούν φέρνοντάς τους σε μορφή ρίζας(δηλαδή suppli) μέσω λήμματος.

Εμβαθύνοντας περισσότερο στα dataset διακρίνουμε άλλο ένα πρόβλημα που θα προκύψει παρακάτω. Όπως ήδη γνωρίζουμε υπάρχει ακραία ανισότητα(extreme imbalance) στις κλάσεις, όμως βλέποντας τον αριθμό δειγμάτων(document) ανά κλάση στον παρακάτω πίνακα(πίνακας 8) παρατηρούμε πως σε 4 από τα 8 dataset υπάρχουν κλάσεις που έχουν λιγότερα από 10 document.

Αυτό είναι προβληματικό διότι θέλουμε να κάνουμε 10fold-CV(σ.σ. περισσότερα για αυτό θα αναλυθούν παρακάτω) και για να γίνει αυτό χρειαζόμαστε τουλάχιστον 10 δείγματα ανά κλάση. Άρα απευθείας μπορούμε να αποκλείσουμε το under sampling κλάσεων καθώς θα μειώναμε έτσι την ανισότητα των κλάσεων αλλά θα εξακολουθούσαμε να έχουμε κλάσεις με λιγότερα από 10 document. Θα μπορούσαμε να κάνουμε over sampling, δηλαδή να παίρναμε περισσότερες από 1 φορές document από την κάθε κλάση, όμως αυτό στην περίπτωση μας δεν θα ήταν συνετό καθώς θα καταλήγαμε να έχουμε διπλότυπα tweet, το οποίο όπως προαναφέρθηκε δημιουργεί πρόβλημα στην περίπτωση text processing, οπότε θα το αποφύγουμε και αυτό θα καταλήξουμε στην τελευταία επιλογή που είναι η επαύξηση δεδομένων(data augmentation) με ελάχιστο over sampling.

Datasets	Caution and advice	Displaced people and evacuations	Donation needs or offers	Infrastructure and utilities damage	Injured or dead people	Missing trapped or found people	Sympathy emotional support	Other useful information	Not related irrelevant	# Σύνολο
2014 Chile earthquake	246	63	10	26	116	6	504	619	342	1.932
2015 Nepal earthquake	35	85	887	129	344	187	522	638	176	3.003
2013 Pakistan earthquake	69	12	299	28	326	5	106	724	312	1.881
2015 Cyclone Pam	92	48	363	212	142	18	93	540	496	2.004
2014 Typhoon Hagupit	348	130	113	92	66	8	290	731	232	2.010
2014 India floods	41	27	45	66	893	14	30	206	498	1.820
2014 Pakistan floods	51	85	463	86	220	103	118	618	25	1.769

2014 California earthquake	72	4	77	305	183	6	80	868	106	1.701
----------------------------------	----	---	----	-----	-----	---	----	-----	-----	-------

Πίνακας 8: Αριθμός document ανά class ανά dataset.

Ως επαύξηση δεδομένων ορίζουμε την διαδικασία με την οποία προσπαθούμε να παράγουμε νέα δεδομένα χρησιμοποιώντας τα παλιά, δηλαδή μετασχηματίζοντας τα παλιά δεδομένα ώστε να φαίνονται διαφορετικά(τουλάχιστον στην θεωρία) ενώ ουσιαστικά είναι το ίδιο. Για παράδειγμα στην υπολογιστική όραση θα κάναμε data augmentation περιστρέφοντας μια εικόνα 90°, οπότε στην θεωρία έχουμε δημιουργήσει μια νέα εικόνα ενώ στην ουσία ένας άνθρωπος θα έλεγε πως είναι η ίδια εικόνα. Στην περίπτωση του κειμένου, προφανώς δεν μπορούμε να περιστρέψουμε ένα κείμενο 90° οπότε θα πρέπει να καταφύγουμε σε άλλους έξυπνους κι εύκολους τρόπους. Από αυτούς, οι 4 πιο απλοί και αποτελεσματικοί είναι: [48] [49]

1. **Αντικατάσταση συνωνύμου:** τυχαία επιλογή μερικών λέξεων από το text που δεν είναι stop words και αντικατάσταση αυτών με ένα συνώνυμό τους.
2. **Τυχαία εισαγωγή λέξης:** τυχαία επιλογή μερικών λέξεων από το text που δεν είναι stop words και πρόσθεση ενός συνωνύμου τους σε μια τυχαία θέση του text.
3. **Τυχαίο ανακάτεμα:** ανακάτεμα μερικών ή ακόμα και όλων των λέξεων του text
4. **Τυχαία αφαίρεση:** τυχαία επιλογή μερικών λέξεων από το text που δεν είναι stop words και αφαίρεση αυτών από το text

Αυτές οι τεχνικές είναι γνωστές ως EDA(Easy Data Augmentation). Από τα παραπάνω, δοθείσας της φύσης των document, μάλλον είναι ιδανικό να γίνουν τα 1 και 3. Αυτό είναι γιατί πρόκειται για tweet, δηλαδή document με περιορισμένου αριθμού όρους(μέγιστο όριο χαρακτήρων 140 την εποχή που συλλέχθηκαν) οπότε δεν θέλουμε το 2 για να μην χαλάσουμε την ομοιομορφία των document(δηλαδή document που καταλήγουν να έχουν μέγεθος παραγράφου), αλλά ούτε και το 4 μας βοηθά για τον ίδιο τον λόγο της φύσης του document, δηλαδή 140 χαρακτήρες είναι λίγοι και παράγουν λίγους όρους άρα το τελευταίο που θα θέλαμε είναι να τους λιγοστέψουμε κι άλλο.

Επεξήγηση κώδικα

Έχοντας πλέον συλλέξει τα 8 αρχεία των dataset, τα εναποθέτουμε σε έναν φάκελο με ονομασία *preprocess* και δημιουργούμε έναν υποφάκελο με όνομα *processed*. Στην συνέχεια εκτελούμε τον κώδικα που βρίσκεται στο παράρτημα ΣΤ' και παρακάτω εξηγείται απλοϊκά η ροή εκτέλεσης με βήματα:

1. **Γραμμές 1-18:** εδώ φορτώνονται τα απαραίτητα πακέτα για την εκτέλεση του προγράμματος
2. **Γραμμές 170-173:** εδώ ξεκινάει η ουσιαστική εκτέλεση του προγράμματος με μερικές οπτικές αλλαγές-βελτιώσεις στην εμφάνιση των αποτελεσμάτων
3. **Γραμμές 175-179:** εδώ προ φορτώνεται το OOV dictionary με το οποίο θα κάνουμε τις αντικαταστήσεις όρων
4. **Γραμμές 184-187:** εδώ δηλώνονται τι θέλουμε το πακέτο *pp* να αφαιρέσει από το text και φορτώνεται ένα λεξικό σε μορφή regular expression(regex) με όλα τα γνωστά διαθέσιμα emoji μέχρι σήμερα το οποίο regex στην συνέχεια θα χρησιμοποιηθεί για την αφαίρεση emoji από το κείμενο
5. **Γραμμές 189-193:** εδώ σαρώνει τον φάκελο για αρχεία dataset και για το καθένα που βρίσκει το στέλνει στην συνάρτηση *preprocessTweets()* για να ξεκινήσει η διαδικασία προ επεξεργασίας. Δηλαδή θα γίνουν 8 διαφορετικές προ επεξεργασίες για καθένα από τα 8 dataset
6. **Γραμμές 21-22:** εδώ φορτώνεται το CVS αρχείο σε μορφή DataFrame από το Pandas

7. **Γραμμές 25-29:** εδώ αφαιρούνται οι εγγραφές που έχουν ίδιο tweet_id από το Pandas, είναι δηλαδή μια ανωμαλία στο dataset με διπλότυπες εγγραφές, κι έπειτα αφαιρούνται εγγραφές που έχουν κενό tweet_text, δεν περιέχουν δηλαδή κάποιο κείμενο
8. **Γραμμές 31-42:** εδώ υπολογίζονται πόσα document ανά class υπάρχουν όπως φαίνεται και στον πίνακα 8, και στην περίπτωση που υπάρχουν λιγότερα από 11 ανά κλάση εκτυπώνει μια ειδοποίηση στον χρήστη με κίτρινα γράμματα πως θα γίνει data augmentation για την αναφερόμενη κλάση που έχει X αριθμό δειγμάτων, όπως φαίνεται στο παρακάτω στιγμιότυπο:

```
Vocabulary loading complete.
Greek model loading complete.

Reading file: ./preprocess/2013_Pakistan_eq_CF_labeled_data.csv
Warning: insufficient documents detected (5) - data augmentation triggered for
[missing trapped or found people].
Output file: ./preprocess/processed/pp_2013_Pakistan_eq_CF_labeled_data.csv
Processing complete.
Reading file: ./preprocess/2014_California_Earthquake_CF_labeled_data.csv
Warning: insufficient documents detected (6) - data augmentation triggered for
[missing trapped or found people].
Warning: insufficient documents detected (4) - data augmentation triggered for
[displaced people and evacuations].
Output file: ./preprocess/processed/pp_2014_California_Earthquake_CF_labeled_data.csv
Processing complete.
Reading file: ./preprocess/2014_Chile_Earthquake_en_CF_labeled_data.csv
```

Εικόνα 33: Στιγμιότυπο εξόδου κατά την εκτέλεση κώδικα preprocess.py

Στην περίπτωση όπου υπάρχουν κλάσεις με λιγότερα από 10 δείγματα τότε πρέπει να γίνει data augmentation και εκτελούνται τα παρακάτω βήματα 8.x.

- 8.1. **Γραμμές 44-49:** εδώ υπολογίζεται ποιες είναι οι κλάσεις που χρειάζονται data augmentation και πόσα επιπλέον document πρέπει να παραχθούν. Για παράδειγμα στο dataset του σεισμού της Καλιφόρνιας το 2014 είναι δύο οι κλάσεις(όπως φαίνονται και στον πίνακα 8 και στην εικόνα 33), οι *missing trapped or found people* και *displaced people and evacuations* με 6 δείγματα και 4 αντίστοιχα η καθεμία οπότε χρειαζόμαστε επιπλέον 4 και 6 δείγμα αντίστοιχα για την κάθε μία.
- 8.2. **Γραμμές 54-85:** εδώ δημιουργούνται τα νέα documents με την τεχνική του συνωνύμου που προαναφέρθηκε. Αρχικά γίνεται word tokenize του text, δηλαδή σπάει η πρόταση σε όρους, και ανιχνεύεται τι μέρος του λόγου είναι ο κάθε όρος(term) μέσω του NLTK. Στην συνέχεια για τον καθένα όρο που είναι ρήμα, επίρρημα ή επίθετο επιλέγεται τυχαία ένα συνώνυμο από όλα τα πιθανά και στην συνέχεια το συνώνυμο αυτό παίρνει την θέση της αρχικής λέξης. Εάν μια λέξη δεν έχει συνώνυμο τότε ο όρος απλά παραμένει. Για παράδειγμα ένα tweet κατηγορίας missing trapped of found people από το dataset του σεισμού της Καλιφόρνιας το 2014 θα μετασχηματιστεί όπως φαίνεται στον παρακάτω πίνακα.

tweet_id	tweet_text	label
'504668324777967616'	RT @friend_beast: Oh, California RT @vickynguyen: Skaters finding upside to quake damage. Photo from #nbcbayarea photog Jeremy Carroll http://t.co/...	missing trapped or found people
'354572'	RT @ friend_beast : Oh , California RT @ vickynguyen : Skaters finding upper_side to tremor damage . Photo from # nbcbayarea photog Jeremy Carroll http://t.co/...	missing trapped or found people

Πίνακας 9: Παράδειγμα αντικατάστασης συνωνύμου.

Τέλος όταν έχει ολοκληρωθεί η αντικατάσταση των συνωνύμων, το νέο document που έχει παραχθεί παίρνει το label που είχε και έναν νέο τυχαίο αριθμό tweet_id(για λόγους ομοιομορφίας – στην πραγματικότητα δεν υπάρχει αυτό το tweet στο Twitter). Εάν ένα document(tweet) δεν έχει κανέναν

όρο που να είναι ρήμα, επίρρημα ή επίθετο τότε το document απλά περνάει όπως είναι, δηλαδή πρακτικά διπλασιάζεται και αυτή είναι μια περίπτωση over sampling. Η διαδικασία σταματάει όταν ο αριθμός των document φτάσει το 10.

- 8.3. **Γραμμές 87-95:** εάν με το προηγούμενο βήμα, δηλαδή την αντικατάσταση συνωνύμων, τα document που υπάρχουν σε μια κλάση δεν έχουν φτάσει σε πλήθος το 10 τότε εδώ δημιουργούνται νέα document με την τεχνική του τυχαίου ανακατέματος όρων που προαναφέρθηκε. Για παράδειγμα στο dataset του σεισμού της Καλιφόρνιας το 2014 η κλάση *displaced people and evacuations* είχε 4 document. Με την μέθοδο της αντικατάστασης συνωνύμων, τα document από 4 γίναν 8(4 αρχικά + 4 παρασκευασμένα) τα οποία όμως δεν επαρκούν γιατί ακόμα είναι λιγότερα από 10. Άρα σε αυτή την περίπτωση θα γίνουν δύο shuffles των λέξεων όπως φαίνεται στον παρακάτω πίνακα.

tweet_id	tweet_text	label
'503651936785403904'	In the aftermath of the South Napa Earthquake that hit northern California this morning, one couple is trying to... http://t.co/gNxVsxs1Vt	displaced people and evacuations
'833542'	northern couple South //t.co/gNxVsxs1Vt morning to is Earthquake In Napa California this aftermath http the : of that the ... hit one trying ,	displaced people and evacuations

Πίνακας 10: Παράδειγμα τυχαίου ανακατέματος.

Το νέο document που έχει παραχθεί παίρνει το label που είχε και έναν νέο τυχαίο αριθμό tweet_id(για λόγους ομοιομορφίας – στην πραγματικότητα δεν υπάρχει αυτό το tweet στο Twitter). Η διαδικασία σταματάει όταν ο αριθμός των document φτάσει το 10 και εδώ το data augmentation ολοκληρώνεται έχοντας προστεθεί όλα τα παρασκευασμένα documents στο corpus.

9. **Γραμμές 96-110:** εδώ ξεκινάει να γίνεται το πραγματικό preprocessing όπου διορθώνονται όλα τα «στραβά» του dataset που προαναφέρθηκαν παραπάνω. Σε αυτές τις γραμμές, για καθένα document, εντοπίζεται η γλώσσα που είναι γραμμένη το document. Εάν το document για κάποιον λόγο έχει έρθει ως αυτό το στάδιο και είναι κενό, δηλαδή δεν περιέχει καμία λέξη ή χαρακτήρα ή οποιονδήποτε άλλον όρο τότε μαρκάρεται για μετέπειτα διαγραφή.
- 9.1. **Γραμμές 111-112:** αφαιρούνται URL(π.χ. twitter.com), αναφορές(@), δεσμευμένες λέξεις(RT, FAV) και γραμματικές φατσούλες(π.χ. _(ツ)_/)
- 9.2. **Γραμμές 113-114:** αφαιρούνται τα emoji με βάση το λεξικό που δημιουργήθηκε στο βήμα 4
- 9.3. **Γραμμές 115-123:** εάν η γλώσσα είναι η αγγλική τότε γίνεται αντικατάσταση ΟΟV λέξεων μέσω του ΟΟV Dictionary που προ φορτώθηκε στο βήμα 3. Επίσης διαγράφονται όλοι οι χαρακτήρες που δεν ανήκουν στον πίνακα ASCII
- 9.4. **Γραμμές 124-125:** αφαιρείται η δέση στο hashtag
- 9.5. **Γραμμές 126-129:** αφαιρούνται τα σημεία στίξης
- 9.6. **Γραμμές 130-131:** αφαιρούνται όλοι οι μονοί χαρακτήρες(π.χ. a, l, 6, κτλ.)
- 9.7. **Γραμμές 132-133:** αφαιρούνται περεταίρω οι URLs που έχουν διαφύγει στο βήμα 9.1 λόγω διαφορετικής δομής(π.χ. <https://www.twitter.com/>)
- 9.8. **Γραμμές 134-135:** αντικαθίστανται οι πολλαπλοί διαδοχικοί κενοί χαρακτήρες από όλες τις αφαιρέσεις που έχουν γίνει έως τώρα με έναν κενό χαρακτήρα
- 9.9. **Γραμμές 136-137:** μετατρέπονται όλοι οι όροι σε μικρά γράμματα(lowercase)
- 9.10. **Γραμμές 139-152:** γίνεται word tokenize, αφαιρούνται τα stop words και πραγματοποιείται η αντικατάσταση ρίζας(stemming).
- 9.11. **Γραμμές 153-160:** Τέλος εάν μετά τα όλα παραπάνω βήματα καταλήξαμε σε ένα κενό document τότε το document αυτό αφαιρείται από το corpus και εδώ ολοκληρώνεται το preprocessing.

Αναλυτικά η εξέλιξη ενός document παραδείγματος από το dataset του σεισμού της Καλιφόρνιας το 2014 φαίνεται συγκεντρωτικά στον παρακάτω πίνακα(πίνακας 11).

10. **Γραμμές 162-167:** αποθηκεύεται το επεξεργασμένο αρχείο με μορφή *pp_XXXX.csv* στον υποφάκελο *processed* και η διαδικασία το συγκεκριμένο dataset ολοκληρώνεται.

Βήμα	Document
Αρχικό	RT @Zamaanie: Zionist US supplies Israel with WMD for Terrorism #Israel #Gaza #Palestine #USA #Ferguson #UK #US #napaquake #p2 http://t.co/ÃtÂ€Â!
9.1	Zionist US supplies Israel with WMD for Terrorism #Israel #Gaza #Palestine #USA #Ferguson #UK #US #napaquake #p2
9.2	Zionist US supplies Israel with WMD for Terrorism #Israel #Gaza #Palestine #USA #Ferguson #UK #US #napaquake #p2
9.3	Zionist US supplies Israel with WMD for Terrorism # Israel # Gaza # Palestine # USA # Ferguson # UK # US # napaquake # p2
9.4	Zionist US supplies Israel with WMD for Terrorism # Israel # Gaza # Palestine # USA # Ferguson # UK # US # napaquake # p2
9.5	Zionist US supplies Israel with WMD for Terrorism Israel Gaza Palestine USA Ferguson UK US napaquake p2
9.6	Zionist US supplies Israel with WMD for Terrorism Israel Gaza Palestine USA Ferguson UK US napaquake p2
9.7	Zionist US supplies Israel with WMD for Terrorism Israel Gaza Palestine USA Ferguson UK US napaquake p2
9.8	Zionist US supplies Israel with WMD for Terrorism Israel Gaza Palestine USA Ferguson UK US napaquake p2
9.9	zionist us supplies israel with wmd for terrorism israel gaza palestine usa ferguson uk us napaquake p2
9.10	zionist us suppli israel wmd terror israel gaza palestin usa ferguson uk us napaquak p2
9.11	zionist us suppli israel wmd terror israel gaza palestin usa ferguson uk us napaquak p2

Πίνακας 11: Βηματική εξέλιξη *preprocessing* ενός document στα αγγλικά.

Έτσι καταλήγουμε σε με 8 νέα αρχεία στον υποφάκελο *processed* και το *preprocessing* έχει ολοκληρωθεί.

Εκπαίδευση & σύγκριση μοντέλων ML

Έχοντας πλέον ολοκληρώσει το *preprocessing* τα δεδομένα είναι σχεδόν έτοιμα. Αυτό που μας προβληματίζει είναι η ακραία ανισότητα στις κλάσεις, η οποία μπορεί να οδηγήσει σε προκατάληψη(*bias*) των αλγορίθμων, δηλαδή οι αλγόριθμοι να επιλέγουν με μεγαλύτερη συχνότητα κατηγορίες για τις οποίες υπάρχουν περισσότερα δείγματα και εξαιτίας αυτού αισθάνονται περισσότερο σίγουροι. Για να αποφευχθεί αυτό ως μετρικό σκορ θα χρησιμοποιήσουμε το *macro-average ROC-AUC* που προαναφέρθηκε στην εισαγωγή επειδή το σκορ του δεν επηρεάζεται από *class imbalance*.

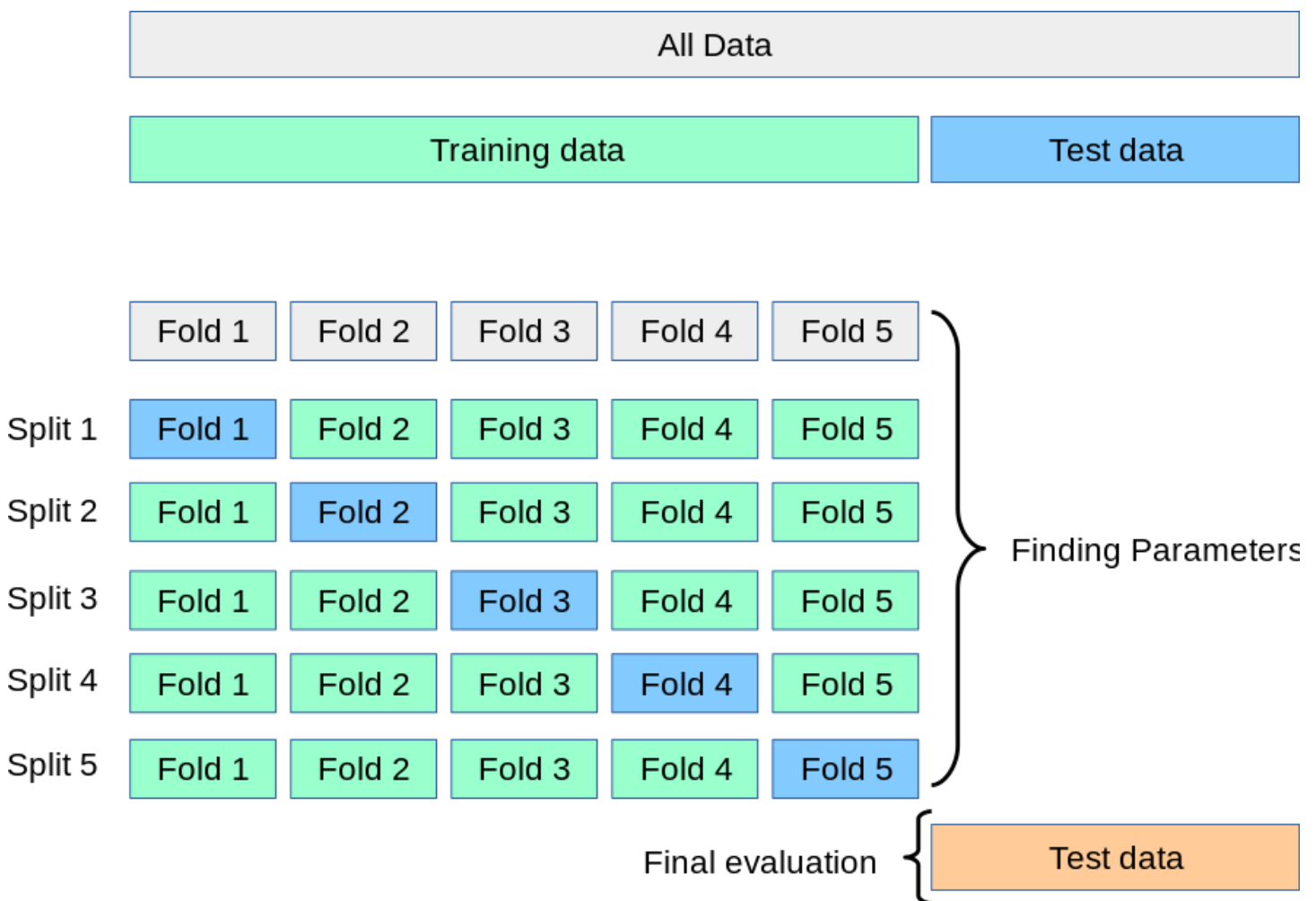
Επειδή τα dataset είναι μικρά(έχουν δηλαδή λίγα documents) καλό θα ήταν να χρησιμοποιήσουμε αλγορίθμους ML οι οποίοι διαχειρίζονται αποτελεσματικά τα μικρά dataset. Από όλους τους αλγορίθμους, επιλέχθηκαν τρεις, οι SVM, MNB, και RF, όπως περιγράφονται στην εισαγωγή. Οι classifiers φορτώθηκαν με τις προεπιλεγμένες τους προεπιλεγμένες τους τιμές(από το Scikit) χωρίς καμία περεταίρω παραμετροποίηση για τον λόγο που περιγράφεται στην υποσημείωση 2. Τα δικά μου αποτελέσματα(στα δεξιά της στήλης) σε σύγκριση τα αποτελέσματα της έρευνας που διεξήγαν ο Imran και οι συνεργάτες του(στα αριστερά της στήλης) είναι καλύτερα ή τουλάχιστον ίσα σε κάθε μέτρηση, και φαίνονται αναλυτικά στον επόμενο πίνακα.

Datasets	Classifier	Caution and advice		Displaced people and evacuations		Donation needs or offers		Infrastructure and utilities damage		Injured or dead people		Missing trapped or found people		Sympathy emotional support		Other useful information		Not related irrelevant		Mean macro-average ROC-AUC	
2014 Chile earthquake	Size(%)	15.00%		2.80%		0.76%		1.70%		5.60%		0.54%		25.00%		30.00%		19.00%		100.40%	
	SVM	0.87	0.88	0.89	0.96	0.57	0.70	0.90	0.90	0.97	0.98	0.23	0.91	0.93	0.96	0.86	0.91	0.93	0.95	0.79	0.91
	NB	0.86	0.86	0.93	0.88	0.78	0.72	0.88	0.73	0.97	0.97	0.64	0.89	0.93	0.96	0.87	0.88	0.95	0.94	0.87	0.87
	RF	0.83	0.88	0.86	0.96	0.67	0.64	0.74	0.89	0.96	0.98	0.46	0.9	0.94	0.96	0.86	0.90	0.92	0.95	0.80	0.90
2015 Nepal earthquake	Size(%)	2.10%		3.10%		28.00%		4.50%		11.00%		5.80%		17.00%		22.00%		6.50%		100.00%	
	SVM	0.47	0.67	0.80	0.87	0.89	0.93	0.85	0.90	0.95	0.97	0.86	0.93	0.88	0.93	0.76	0.81	0.75	0.85	0.80	0.87
	NB	0.68	0.64	0.82	0.84	0.91	0.93	0.90	0.92	0.95	0.97	0.89	0.9	0.91	0.93	0.79	0.80	0.84	0.84	0.85	0.87
	RF	0.56	0.55	0.73	0.86	0.89	0.92	0.74	0.88	0.94	0.97	0.87	0.9	0.89	0.92	0.76	0.78	0.75	0.81	0.8	0.85
2013 Pakistan earthquake	Size(%)	6.30%		0.82%		15.00%		2.00%		17.00%		0.49%		5.60%		35.00%		18.00%		100.21%	
	SVM	0.77	0.58	0.80	0.82	0.92	0.97	0.76	0.86	0.95	0.96	0.63	1.0	0.82	0.89	0.84	0.90	0.84	0.86	0.81	0.87
	NB	0.82	0.78	0.87	0.71	0.94	0.96	0.91	0.87	0.93	0.95	0.74	1.0	0.83	0.88	0.84	0.90	0.84	0.85	0.86	0.88
	RF	0.68	0.71	0.70	0.75	0.92	0.97	0.77	0.89	0.95	0.97	0.69	1.0	0.78	0.90	0.88	0.89	0.83	0.87	0.80	0.88
2015 Cyclone Pam	Size(%)	7.00%		3.10%		17.00%		11.00%		7.20%		1.30%		5.00%		25.00%		24.00%		100.60%	
	SVM	0.76	0.79	0.80	0.85	0.92	0.94	0.85	0.90	0.95	0.98	0.39	0.77	0.66	0.81	0.77	0.84	0.90	0.94	0.78	0.87
	NB	0.79	0.76	0.82	0.76	0.92	0.94	0.86	0.89	0.97	0.96	0.56	0.60	0.79	0.81	0.80	0.81	0.94	0.93	0.83	0.83
	RF	0.68	0.78	0.80	0.85	0.90	0.94	0.80	0.88	0.95	0.98	0.47	0.74	0.71	0.79	0.79	0.81	0.92	0.94	0.78	0.86
2014 Typhoon Hagupit	Size(%)	20.00%		6.60%		5.50%		5.10%		3.00%		0.58%		13.00%		33.00%		13.00%		99.78%	
	SVM	0.74	0.80	0.95	0.98	0.88	0.93	0.76	0.87	0.94	0.97	0.44	0.69	0.92	0.94	0.74	0.81	0.81	0.84	0.80	0.87
	NB	0.75	0.79	0.96	0.96	0.89	0.94	0.82	0.83	0.96	0.97	0.57	0.82	0.92	0.95	0.78	0.81	0.81	0.83	0.83	0.88
	RF	0.71	0.78	0.97	0.98	0.84	0.94	0.73	0.87	0.94	0.96	0.58	0.74	0.91	0.95	0.75	0.80	0.80	0.80	0.80	0.87
2014 India floods	Size(%)	3.60%		1.40%		2.60%		4.30%		47.00%		0.87%		1.30%		14.00%		25.00%		100.07%	
	SVM	0.82	0.92	0.80	0.95	0.92	0.92	0.92	0.91	0.97	0.98	0.66	0.86	0.63	0.73	0.87	0.92	0.97	0.98	0.84	0.91
	NB	0.89	0.89	0.92	0.84	0.93	0.90	0.90	0.90	0.93	0.97	0.79	0.78	0.83	0.70	0.89	0.88	0.98	0.97	0.90	0.87
	RF	0.83	0.90	0.79	0.82	0.86	0.89	0.87	0.94	0.97	0.98	0.66	0.72	0.65	0.65	0.91	0.92	0.96	0.98	0.83	0.87
2014 Pakistan floods	Size(%)	3.90%		6.20%		25.00%		5.40%		13.00%		6.40%		6.00%		32.00%		2.30%		100.20%	
	SVM	0.71	0.77	0.84	0.82	0.82	0.91	0.77	0.78	0.94	0.96	0.85	0.90	0.88	0.87	0.74	0.82	0.47	0.69	0.78	0.84
	NB	0.83	0.80	0.80	0.79	0.85	0.90	0.79	0.82	0.94	0.95	0.85	0.87	0.89	0.90	0.77	0.80	0.65	0.69	0.82	0.84
	RF	0.72	0.76	0.80	0.80	0.87	0.90	0.78	0.83	0.95	0.97	0.84	0.88	0.86	0.86	0.79	0.82	0.59	0.63	0.80	0.83
2014 California earthquake	Size(%)	6.30%		0.48%		4.30%		18.00%		10.00%		0.51%		4.10%		47.00%		9.40%		100.09%	
	SVM	0.84	0.78	0.54	1.0	0.93	0.97	0.88	0.90	0.97	0.99	0.62	0.99	0.84	0.93	0.77	0.84	0.72	0.71	0.79	0.90
	NB	0.88	0.85	0.57	0.98	0.94	0.97	0.86	0.90	0.97	0.98	0.79	0.99	0.90	0.94	0.78	0.85	0.77	0.75	0.83	0.91
	RF	0.81	0.84	0.49	1.0	0.87	0.98	0.89	0.91	0.98	0.99	0.57	1.0	0.88	0.94	0.81	0.85	0.77	0.72	0.79	0.91

Πίνακας 12: Συγκριτικά σκορ macro-average ROC-AUC για την αγγλική υλοποίηση

Το να εκπαιδεύεις και να τεστάρεις τους classifiers πάνω στα ίδια δεδομένα είναι πηγαία λάθος, και αυτό γιατί το να ζητάς από ένα μοντέλο να σου κατηγοριοποιήσει κάτι που έχει ήδη μάθει να κατηγοριοποιεί θα σου επιστρέψει ένα υπεραισιόδοξο σκορ, αν όχι τέλειο. Αυτό το φαινόμενο αποκαλείται *overfitting*. Για να το αποφύγουμε θα πρέπει να σπάσουμε το dataset σε train data και σε test data (συνήθως 70%-30% αντίστοιχα) για να εκπαιδεύσουμε τα μοντέλα (classifiers) στο train και να τα τεστάρουμε στο test. Όμως κάτι τέτοιο στην δική μας περίπτωση δεν θα ήταν πολύ σοφό καθώς τα dataset θα συρρικνώνονταν δραστικά και περιέχουν ήδη λίγα documents το οποίο αναπόφευκτα θα οδηγήσει σε πολύ χειρότερα σκορ και πολύ μεγαλύτερη προκατάληψη. [50]

Σε αυτές τις περιπτώσεις η λύση είναι να κάνουμε *k-fold Cross Validation (CV)*, δηλαδή να σπάσουμε ολόκληρο το dataset σε k κομμάτια τα οποία περιέχουν n/k τυχαία documents το καθένα από αυτά και στην συνέχεια να κάνουμε train set τα μοντέλα με τα $k-1$ κομμάτια αυτών, κρατώντας ένα κομμάτι απέξω ως test set ώστε στην συνέχεια να ελέγξουμε το σκορ τους στο $k-(k-1)$ τμήμα. Αυτή η διαδικασία επαναλαμβάνεται k φορές και το συνολικό σκορ είναι η μαθηματική ελπίδα (macro-average) όλων των σκορ στα test sets. Απεικονίζεται διαισθητικά στην παρακάτω εικόνα:



Εικόνα 34: Απεικόνιση διαδικασίας Cross Validation με 5-fold.

(Πηγή: [50])

Στην προκειμένη περίπτωση θα πρέπει να γίνει πρώτα feature selection επιλέγοντας τα 1.000 καλύτερα features με IG, όπως δηλαδή επιτάσσει η έρευνα. Η εκπαίδευση και το ROC-AUC τεστ των classifiers θα γίνουν με την τεχνική του 10-fold Stratified Cross Validation, δηλαδή οι κλάσεις θα είναι στρωματοποιημένες που αυτό σημαίνει πως κάθε fold θα αναπαρίστανται όλες οι κλάσεις, το οποίο είναι απαραίτητο για να έχουν

νόημα τα σκορ. Τα fold που χρειαζόμαστε είναι συνήθως 5, εδώ θα είναι 10 γιατί σε μικρά dataset όπως το δικό μας θέλουμε τα folds να είναι πολλά και να είναι το κατά δύναμιν πιο ετερογενή ώστε να έχουν καλύτερες πιθανότητες να ανταπεξέλθουν καλά σε νέα ξένα δεδομένα.

Στο συγκεκριμένο multiclass πρόβλημα οι classifiers δουλεύουν με τακτική One-VS-Rest classification (OvR ή γνωστό αλλιώς ως One-VS-All – OvA), δηλαδή για παράδειγμα δημιουργούνται τόσοι classifiers όσοι και οι κλάσεις (εξετάζουν εάν ανήκει ένα document στην κατηγορία *caution and advise* και εάν όχι μετά ένας άλλος classifier εξετάζει εάν ανήκει στην κατηγορία *displaced people and evacuations*, κ.ο.κ. μέχρι να βρει ένας classifier αν το document ανήκει στην κλάση που έχει αναλάβει. [51] Παρότι λοιπόν ως αλγόριθμος επιλέχθηκε ο MNB, χάρις την επιλογή να εκπαιδευτούν οι classifiers με OvR τακτική, εύκολα παρατηρούμε πως δουλεύει όπως ο BNB όπως γνωρίζουμε από την θεωρία που έχει προαναφερθεί στην εισαγωγή. Η OvR επιλογή είναι αναγκαστική γιατί και μόνο έτσι υπολογίζεται το σκορ ROC-AUC, δηλαδή όπως είναι γνωστό από την θεωρία, είναι ανάμεσα σε δύο και μόνο κλάσεις όπως φαίνεται και στην εικόνα 9.

Από τον παραπάνω πίνακα (πίνακας 12) παρατηρούμε ότι όλα τα δικά μου αποτελέσματα είναι καλύτερα, με την εξαίρεση του NB στις πλημμύρες του Πακιστάν το 2014. Αυτό δεν μπορώ να συμπεράνω γιατί συμβαίνει καθώς, όπως δήλωσα και στην υποσημείωση 2, είναι ασαφής ο τρόπος που υλοποίησαν τον Naive Bayes και δεν αναφέρουν ποιον συγκεκριμένα χρησιμοποίησαν. Με βάσει την θεωρία και την βιβλιογραφία, ο καλύτερος αλγόριθμος Naive Bayes σε text classification είναι κατά γενική ομολογία ο πολυωνυμικός (δηλαδή ο MNB). Αποπειράθηκα να δοκιμάσω και τον BNB ο οποίος μου επέστρεψε το ίδιο αποτέλεσμα με τον OvR MNB, το οποίο είναι λογικό όπως εξηγήθηκε από την προηγούμενη παράγραφο. Ο GNB είναι για προβλήματα ποσοτικά οπότε εγγενώς δεν δουλεύει για την δική μας περίπτωση.

Χωρίς άλλη καθυστέρηση είμαστε έτοιμοι να ξεκινήσουμε την κατηγοριοποίηση χάρις την προ επεξεργασία που κάναμε στο προηγούμενο βήμα.

Επεξήγηση κώδικα

Έχοντας πλέον προ επεξεργαστεί τα 8 αρχεία των dataset και βρίσκονται στον υποφάκελο *processed* του φακέλου *preprocess*, τα εναποθέτουμε σε έναν νέο εξωτερικό φάκελο με ονομασία *classification*. Στην συνέχεια εκτελούμε τον κώδικα που βρίσκεται στο παράρτημα Ζ' και παρακάτω εξηγείται απλοϊκά η ροή εκτέλεσης με βήματα:

1. **Γραμμές 1-19:** εδώ φορτώνονται τα απαραίτητα πακέτα για την εκτέλεση του προγράμματος
2. **Γραμμές 162-164:** εδώ ξεκινάει η ουσιαστική εκτέλεση του προγράμματος με μερικές οπτικές αλλαγές-βελτιώσεις στην εμφάνιση των αποτελεσμάτων
3. **Γραμμές 166-170:** εδώ σαρώνει τον φάκελο για αρχεία dataset και για το καθένα που βρίσκει το στέλνει στην συνάρτηση `classify()` για να ξεκινήσει η διαδικασία προ επεξεργασίας. Δηλαδή θα γίνουν 8 διαφορετικές προ κατηγοριοποιήσεις για καθένα από τα 8 dataset. Στις μεταβλητές φορτώνουμε τον αριθμό των fold που επιθυμούμε (δηλαδή 10), εάν επιθυμούμε για το καθένα να εκτυπωθούν τα πιο σημαντικά feature που επιλέχθηκαν και εάν θέλουμε το σχεδιάγραμμα του ROC-AUC για κάθε fold για κάθε dataset.
4. **Γραμμές 103-104:** εδώ φορτώνεται το CVS αρχείο σε μορφή DataFrame από το Pandas
5. **Γραμμές 106-114:** εδώ φορτώνονται τα documents στο X, τα labels στο Y, προσδιορίζεται το όνομα του αρχείου και αρχικοποιούνται οι classifiers SVM, MNB και RF με τις προεπιλεγμένες τους τιμές
6. **Γραμμές 118-126:** εδώ υπολογίζονται τα ROC-AUC σκορ για τον κατηγοριοποιητή SVM με χάρις την εξωτερική κλήση της συνάρτησης `calculateROC_AUC()` και μόλις είναι έτοιμα εκτυπώνεται το macro-average σκορ στο τερματικό

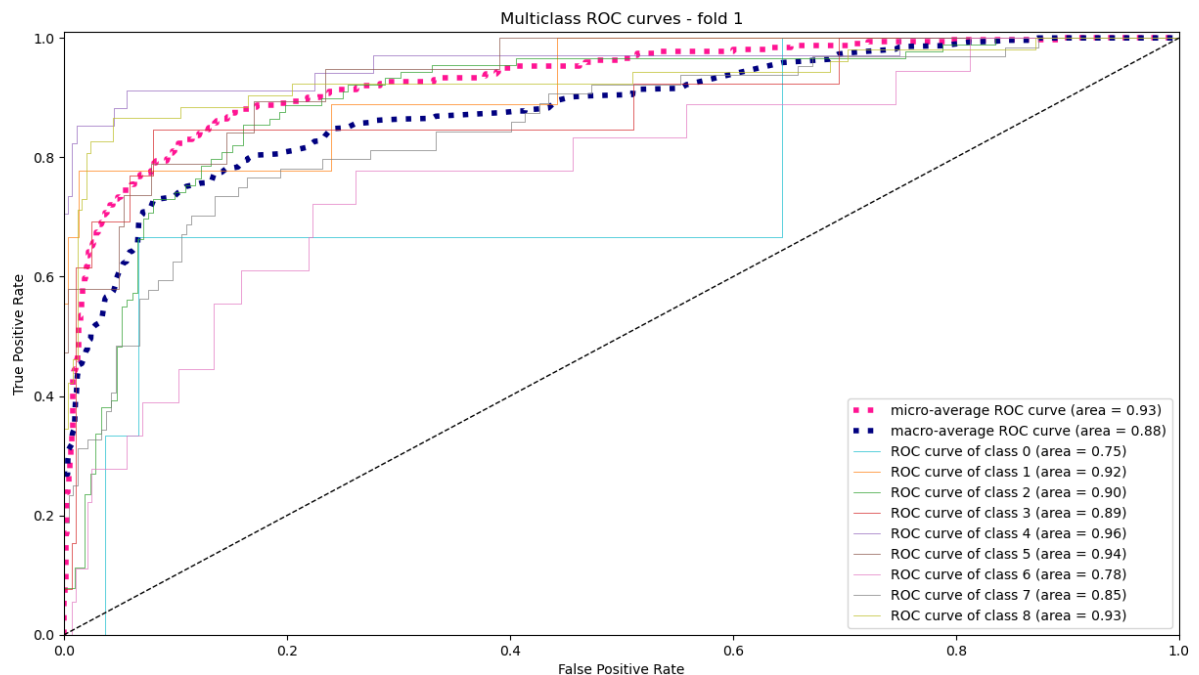
Τα σκορ από κάθε fold αναλυτικά αποθηκεύονται στον φάκελο ROC_AUC σε αρχείο της μορφής XXXX_SVM.csv.

- 6.1. Γραμμές 23-27: εδώ αρχικοποιούνται τα αντικείμενα που θα χρησιμοποιηθούν για να βρούμε τα top 1000 feature και δηλώνουμε ότι θα πάρουμε όρους σαν unigrams και bigrams.
- 6.2. Γραμμές 29-31: εδώ δηλώνουμε ότι επιθυμούμε 10-fold Stratified CV με ανακάτεμα document σε κάθε fold και δημιουργούμε τα train set και test set με βάσει τον τρόπο που αναφέρθηκε στην θεωρία του CV
- 6.3. Γραμμές 33-36: εδώ μετατρέπουμε τους όρους(terms) των train set και test set σε αριθμητικά διανύσματα(γιατί οι classifiers δουλεύουν με μαθηματικές φόρμουλες και προφανώς δεν μπορούν να διαχειριστούν κείμενο) και στην συνέχεια βρίσκουμε τα 1.000 καλύτερα feature με IG. Στον παρακάτω πίνακα φαίνεται ένα στιγμιότυπο με τα 15 καλύτερα feature για το dataset του σεισμού της Καλιφόρνιας το 2014:

# εμφανίσεων	Όρος(term)	Αριθμητικό διάνυσμα(σκορ)
20633	toll	0.082521
5135	death	0.082298
5157	death toll	0.076864
15994	prayer	0.076777
5741	donat	0.075034
8878	help	0.059555
15941	pray	0.052753
16967	relief	0.050741
5053	dead	0.048174
10758	kill	0.037374
20369	thought	0.031792
17657	rubbl	0.031473
12401	miss	0.027647
5984	earthquak	0.026325
6652	evacu	0.024846

Πίνακας 13: Τα 15 πιο σημαντικά feature για το dataset των πυρκαγιών 2014 της Καλιφόρνιας(φθίνουσα σειρά).

- 6.4. Γραμμές 39-42: εδώ εκτυπώνουμε(κατ' απαίτηση) τα καλύτερα feature που επιλέχθηκαν
- 6.5. Γραμμές 44-47: εδώ μετατρέπουμε τις κλάσεις(που είναι κείμενο) σε αριθμούς, από το 0 έως το 8
- 6.6. Γραμμές 49-50: εδώ εκπαιδεύεται το μοντέλο στο train set και τεστάρεται στο test set υπολογίζοντας τις πιθανότητες του καθενός document να ανήκει σε μια κλάση.
- 6.7. Γραμμές 52-79: εδώ γίνεται ο υπολογισμός του ROC-AUC όπως είναι γνωστός από την Scikit Learn
- 6.8. Γραμμές 81-100: εδώ γίνεται το σχεδιάγραμμα(κατ' απαίτηση) του ROC-AUC και επιστρέφεται η κλήση στην αρχική συνάρτηση. Ένα παράδειγμα για το πρώτο fold του dataset του φονικού σεισμού στο Νεπάλ το 2015 είναι το παρακάτω και το οποίο διακρίνουμε ότι είναι πολύ καλό, δηλαδή το μοντέλο έχει συνολική πιθανότητα να προβλέψει την σωστή κλάση 88%, και πως η μπλε διακεκομμένη γραμμή έχει μεγάλη περιοχή(AUC) από κάτω της:



Εικόνα 35: Σχεδιάγραμμα ROC-AUC του 1ου fold στο dataset 2015_Nepal

7. Γραμμές 128-136: εδώ υπολογίζονται τα ROC-AUC σκορ για τον κατηγοριοποιητή MNB με χάρις την εξωτερική κλήση της συνάρτησης `calculateROC_AUC()` και μόλις είναι έτοιμα εκτυπώνεται το macro-average σκορ στο τερματικό. Τα σκορ από κάθε fold αναλυτικά αποθηκεύονται στον φάκελο ROC_AUC σε αρχείο της μορφής XXXX_MNB.csv.

7.1. Ομοίως με τα 6.x

8. Γραμμές 139-146: εδώ υπολογίζονται τα ROC-AUC σκορ για τον κατηγοριοποιητή RF με χάρις την εξωτερική κλήση της συνάρτησης `calculateROC_AUC()` και μόλις είναι έτοιμα εκτυπώνεται το macro-average σκορ στο τερματικό. Τα σκορ από κάθε fold αναλυτικά αποθηκεύονται στον φάκελο ROC_AUC σε αρχείο της μορφής XXXX_RF.csv.

8.1. Ομοίως με τα 6.x και 7.x

Έτσι καταλήγουμε με 24 νέα αρχεία στον φάκελο ROC_AUC και τα συγκεντρωτικά τους αποτελέσματα είναι τα αναγραφόμενα του πίνακα 12. Εδώ ολοκληρώνεται το αγγλικό μέρος της διπλωματικής

Twitter as a Lifeline for the deadly Attica 2018 Wildfires

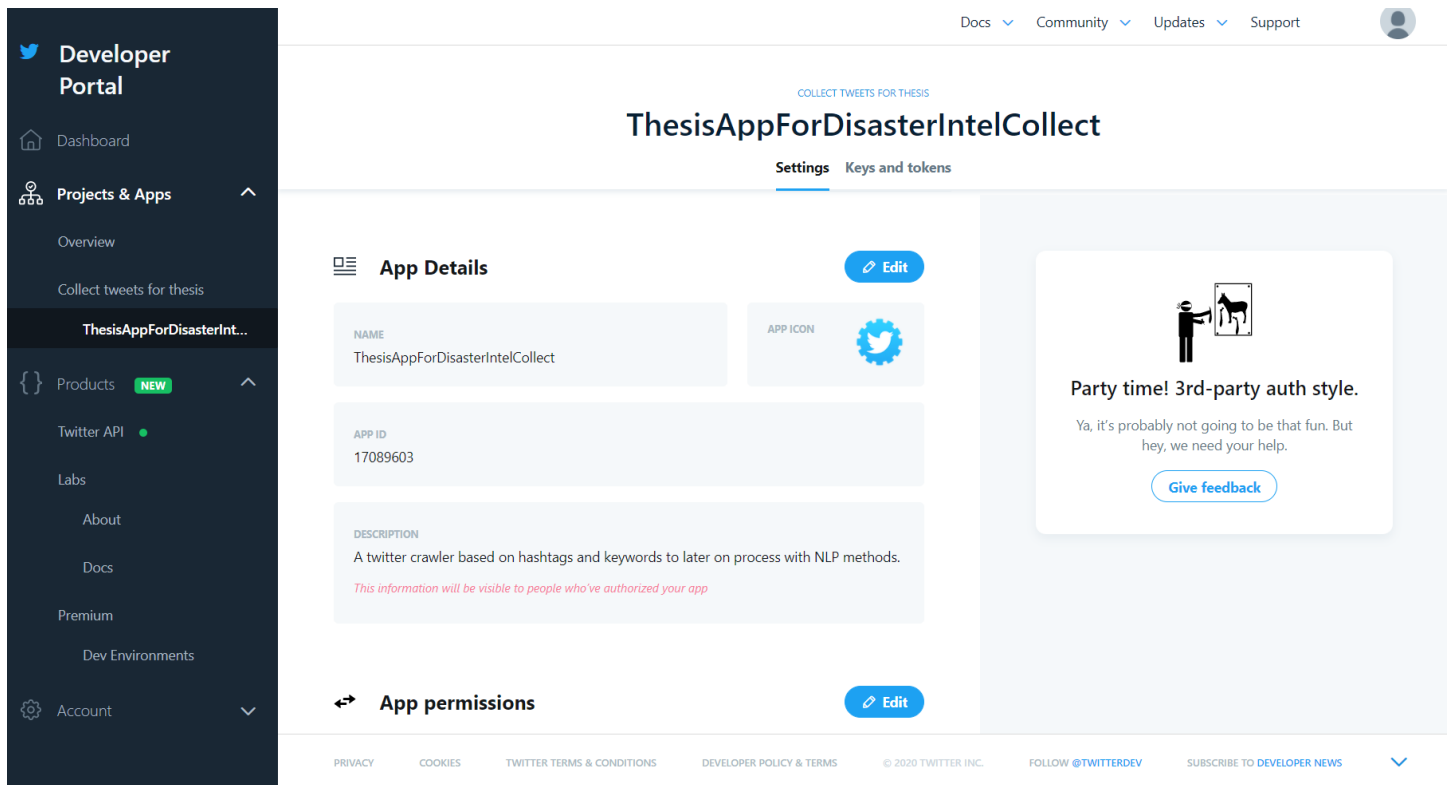
Έχοντας ήδη αναφέρει τα γεγονότα του τραγικού συμβάντος και έχουν πραγματοποιήσει την υλοποίηση κατά την λογική προσέγγιση των ερευνητών για το αγγλικό κομμάτι θα επιχειρήσω να επαναλάβω την διαδικασία που αναφέρουν και αισιοδοξώ να βγάλω παρόμοια αποτελέσματα. Η διαδικασία θα είναι η ίδια με μικρές παραμετροποιήσεις σε λίγα σημεία, απόρροια της αλλαγής της γλώσσας του corpus (δηλαδή των tweet) από τα αγγλικά στα ελληνικά. Δεν θα αναφερθώ ξανά σε διαδικασίες & λογισμούς που έχω αναφερθεί ήδη στο αγγλικό κομμάτι οπότε προφανώς η περιγραφή του κομματιού που ακολουθεί θα είναι μικρότερη σε έκταση χωρίς αυτό να σημαίνει πως μικρότερη είναι η υλοποίησή του.

Συλλογή δεδομένων

Σε αντίθεση με την προηγούμενη περίπτωση, τα tweet που είναι σχετικά με το μάτι δεν έχουν συλλεχθεί και θα πρέπει να αναπτυχθεί εδώ ο τρόπος και η λογική συλλογής τους, ακριβώς όπως έκαναν και ο Imran και οι συνεργάτες του όταν τα συλλέξαν από μόνοι τους. Άλλη μια διακριτή διαφορά είναι πως τα tweet που θα συλλέξουμε δεν θα είναι κατηγοριοποιημένα και θα πρέπει αυτό να γίνει από εμάς.

Χωρίς άλλη καθυστέρηση θα ξεκινήσουμε να συλλέγουμε τα tweet με βάσει την μορφή που περιεγράφηκε στον πίνακα 7. Η συλλογή τους θα γίνει με κλήσεις στο API του Twitter μέσω της βιβλιοθήκης Tweepy που υπάρχει για την python. Για να έχουμε πρόσβαση στο API του Twitter πρέπει να έχουμε τα κλειδιά πρόσβασης του API τα οποία επιδίδονται κατά την χειραψία. Για να αποκτήσουμε τα κλειδιά API πρέπει έναν λογαριασμό developer σε αυτό οπότε θα πάμε στην ιστοσελίδα developer.twitter.com και θα δημιουργήσουμε έναν. Εγώ δημιούργησα τον λογαριασμό με αναφορά το @CrtdT κι όνομα CrtdForThes που είναι αρκτικόλεξο για Created For Thesis.

Στην συνέχεια στο Developer Portal του Twitter πρέπει να δημιουργήσουμε μια νέα Project App συμπληρώνοντας την φόρμα ώστε να αποκτήσουμε τα κλειδιά API. Όταν ελεγχθεί & εγκριθεί η δημιουργία της εφαρμογής από το Twitter τότε μετά μπορούμε να πάρουμε τα 4 κλειδιά API που είναι απαραίτητα για την συλλογή των tweet. Εγώ δημιούργησα την εφαρμογή ThesisAppForDisasterIntelCollect, όπως φαίνεται στην παρακάτω εικόνα.



Εικόνα 36: Στιγμιότυπο του Project App από το Developer Portal του Twitter

Έχοντας δημιουργήσει τα 4 νέα κλειδιά API τότε το μόνο που μένει είναι να δημιουργήσουμε ένα νέο αρχείο με όνομα twitterCredentials.py και να τα δηλώσουμε εκεί μέσα όπως φαίνεται και στο παράρτημα Δ'.

Πλέον μπορούμε να αρχίσουμε να συλλέγουμε τα tweet με βάσει ένα query. Το καλύτερο query που θα μας επιστρέψει τα πιο σχετικά δεδομένα αποφάνθηκε να είναι(έπειτα από δοκιμές) το:

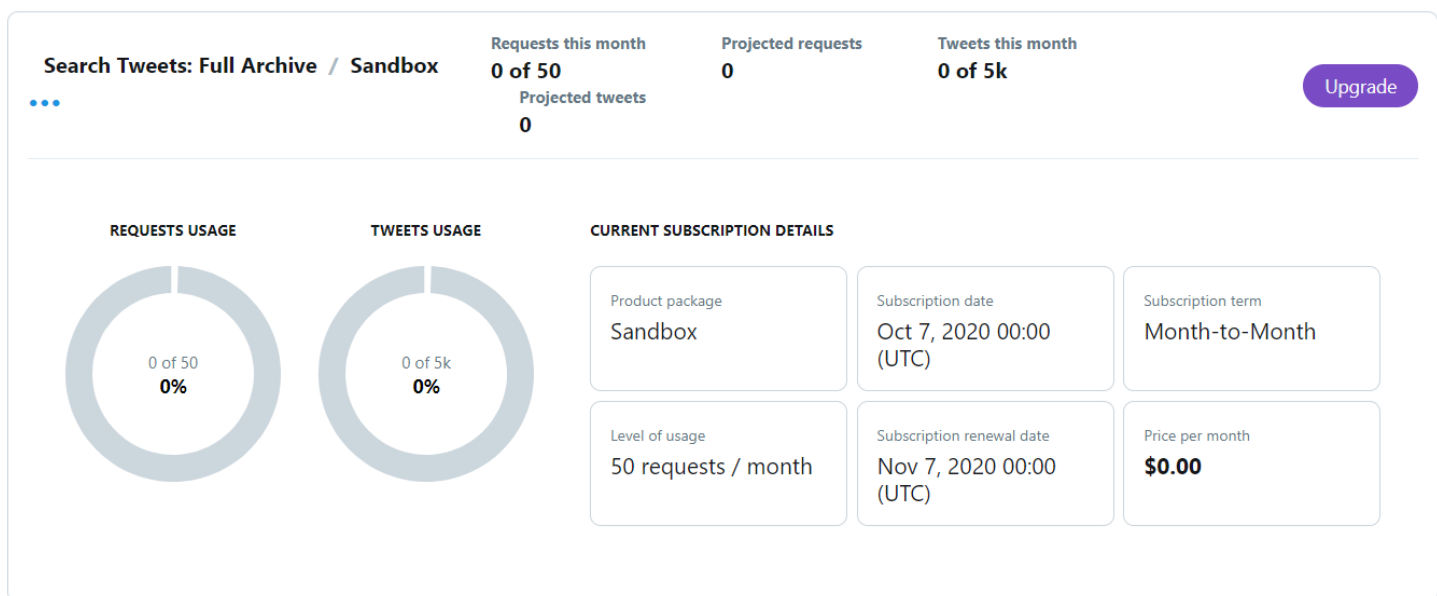
"Μάτι OR Ματι OR Φωτιά OR Φωτια OR Πυρκαγιά OR Πυρκαγια"

καθώς αυτό είναι που μας επιστρέφει τα πιο σχετικά αποτελέσματα. Το χρονικό διάστημα που θα συλλεχθούν τα tweet θα είναι από τις 23 Ιουλίου 2018 τις 12:00 το μεσημέρι(12:03 ξέσπασε η πρώτη πυρκαγιά) έως και τις 27 Ιουλίου 2018 τα μεσάνυχτα. Ο στόχος είναι να δειγματοληπτούμε ωριαία 100 tweet. Στον παρακάτω πίνακα φαίνονται το πλήθος των tweet που συλλέχθηκαν ανά ώρα:

Από	Έως	# συλλεγμένων tweet	Παρατηρήσεις
23 Ιουλίου 2018 – 12:00	23 Ιουλίου 2018 – 23:59	1.200	Κατά την αρχή της κρίσης τα περισσότερα tweet ήταν άσχετα με το αντικείμενο καθώς δεν ήταν ανεξέλεγκτες οι πυρκαγιές ούτε υπήρχαν θύματα.
24 Ιουλίου 2018 – 00:00	24 Ιουλίου 2018 – 23: 59	6.300	Συλλέχθηκαν 100 ωριαία πλην της 23:00 το βράδυ, άρα 2.300 μέσα στην μέρα, κι επιπλέον 4.000 στο τέλος της μέρας τότε ήταν που είχαν τελειώσει τα γεγονότα, είχε περάσει ο πανικός και ο κόσμος έψαχνε τους αγνοουμένους ή βοήθεια
25 Ιουλίου 2018 – 00:00	25 Ιουλίου 2018 – 23:59	2.400	Πολλά tweet που ψάχνουν αγνοουμένους, ανακαλύπτονται σχεδόν όλοι οι νεκροί
26 Ιουλίου 2018 – 00:00	26 Ιουλίου 2018 – 23:59	2.400	Δίνονται στην δημοσιότητα όλα τα ονόματα των νεκρών, ξεχειλίζει η οργή
27 Ιουλίου 2018 – 00:00	27 Ιουλίου 2018 – 22:59	2.300	Το ενδιαφέρον αρχίζει να φθίνει ενώ οι υπάρχουν οι περισσότερες κραυγές αγανάκτησης

Πίνακας 14: Πολιτική συλλογής ελληνικών tweet για τις πυρκαγιές Αττικής 2018.

Προσοχή πρέπει να δοθεί στα όρια που δικαιούμαστε να συλλέξουμε ανά μήνα τα οποία είναι τα 5.000 tweet όπως φαίνεται στην εικόνα παρακάτω. Ως εκ τούτου τα tweet αυτά συλλέχθηκαν σε διάστημα 3 μηνών.



Εικόνα 37: Όρια συλλογής tweet ανά μήνα.

Επεξήγηση κώδικα

Πλέον ήρθε η στιγμή να συλλέξουμε τα tweet με τον τρόπο που περιγράψαμε και ο κώδικας όπου γίνεται αυτό φαίνεται στο παράρτημα Ε'. Έχοντας έτοιμο το αρχείο με τα διαπιστευτήρια που φαίνεται στο παράρτημα Δ', τώρα πρέπει να δημιουργήσουμε έναν νέο κενό φάκελο με όνομα *fetchTweets* και στην συνέχεια εκτελούμε τον κώδικα. Παρακάτω εξηγείται απλοϊκά η ροή εκτέλεσης με βήματα:

1. **Γραμμές 1-3:** εδώ φορτώνονται τα απαραίτητα πακέτα για την εκτέλεση του προγράμματος
2. **Γραμμές 5-9:** εδώ φορτώνονται τα διαπιστευτήρια για το API του Twitter
3. **Γραμμές 10-12:** εδώ αρχικοποιούνται οι μεταβλητές που θα χρησιμοποιήσουμε για την αναζήτηση
4. **Γραμμές 14-23:** εδώ γίνεται η συλλογή των tweet όπως προαναφέρθηκε, δηλαδή 100 ωριαία, ανά ώρα για τις 24 ώρες της ημέρας. Από όλα τα αδόμητα δεδομένα που έχει ένα tweet εμείς διατηρούμε μόνο 3, δηλαδή το *tweet_id*, την ημερομηνία δημιουργίας του και το *tweet_text*

Στην συνέχεια επαναλαμβάνουμε το βήμα 4 για όσες μέρες θέλουμε να συλλέξουμε και στην τελευταία γραμμή τα tweet αποθηκεύονται εξαγοντας το αρχείο *greek.csv* που περιέχει όλα τα unlabeled tweets. Εδώ ολοκληρώνεται η διαδικασία της συλλογής.

Προεπεξεργασία δεδομένων

Όπως αναμενόταν τα δεδομένα αυτά έχουν αρκετό θόρυβο(παρόμοιο με αυτόν του πίνακα 7) και θα πρέπει να διορθωθούν και εδώ όλα τα «στραβά» που αναφέρθηκαν στο αγγλικό κομμάτι. Πρώτα όμως θα πρέπει να κατηγοριοποιηθούν και η κατηγοριοποίηση θα γίνει με βάσει τις κατηγορίες και την περιγραφή τους που αναγράφονται στον πίνακα 4. Η συνολική διαδικασία κατηγοριοποίησης διήρκεσε περίπου 10 ημέρες. Στην κάθε κατηγορία πήγαν tweet όπως φαίνονται μερικά δείγματα & σχόλια στον πίνακα παρακάτω:

Πιθανό περιεχόμενο tweet ανά κατηγορία & σχόλια	
<i>Injured or dead people</i>	Πχ. «26 άνθρωποι που κάηκαν ... », δεν βρέθηκαν πολλοί injured και τα περισσότερα ήταν από νεκρούς, τα δίδυμα παιδιά που ήταν αρχικά στην κατηγορία Missing και μετά μαθεύτηκε σκοτώθηκαν και ήρθαν κατηγορία Dead, κ.α.

<i>Missing, trapped, or found people</i>	Πχ. Έψαχναν συγγενείς, πολλά tweet με αγνοούμενα & διασωθέντα ζώα που τους έλεγαν να επικοινωνήσουν με ένα τηλέφωνο, διάδοση της πλατφόρμας που δημιούργησε ένας για αγνοούμενους, μια άλλη πλατφόρμα της Google για τον ίδιο σκοπό, κ.α.
<i>Displaced people and evacuations</i>	Πάρα πολύ λίγα βρέθηκαν, τα οποία ήταν για έναν κτήριο φιλοξενούσε ορφανά και κήκε και δεν είχαν που να μείνουν, και για μια άλλη κατασκήνωση που κήκε.
<i>Infrastructure and utilities damage</i>	Βρέθηκαν λίγα, κυρίως tweet για καμένα οικόπεδα και αυτοκίνητα
<i>Donation needs or offers or volunteering services</i>	Πάρα πολλά βρέθηκαν, όπως για συγκέντρωση τροφίμων, συντονισμός βοήθειας από άλλες πόλεις όπως την Θεσσαλονίκη, δωρεά Κύπρου 10εκ., δωρεά ΠΑΟΚ, δωρεά ΟΛΥΜΠΙΑΚΟΥ, λογαριασμοί IBAN για τους πληγέντες, προσφορές εστιατορίων, προσφορές ΣΚΛΑΒΕΝΙΤΗ, προσφορές από Λονδίνο, φιλοξενία για τους πληγέντες, καταφύγιο για ζώα, κ.α.
<i>Caution and advice</i>	Ποια είναι η καλύτερη αλοιφή για εγκαύματα, προσοχή για ψεύτικους λογαριασμούς IBAN, αναφορές για υπερφόρτωση των τηλεφωνικών γραμμών, συμβουλές για όταν στέλνουν βοήθεια να θυμούνται στα πακέτα να συμπεριλαμβάνουν και παιχνίδια για τα παιδιά θύματα, κ.α.
<i>Sympathy and emotional support</i>	Προσευχές & συμπόνια(και από ξένες χώρες), υπήρχαν επιζήσαντες που μοιράζονταν την εμπειρία τους, ευχές για τα δίδυμα παιδάκια που κήκαν ζωντανά στον φράχτη, κ.α.
<i>Other useful information</i>	Αναφορές για το πώς εξελίχθηκε η φωτιά, τι την προξένησε, εικόνες, βίντεο, εικόνα με έναν σκύλο να διασώζει ένα γατάκι κρατώντας το στο στόμα του, κ.α.
<i>Not related or irrelevant</i>	Κυρίως έβριζαν πολιτικούς και το κράτος, tweet με άγνωστη γλώσσα(όπως πχ. Τούρκικα), φωτιές σε άλλες περιοχές που συνέβαιναν εκείνες τις μέρες(όπως πχ. στη Ρόδο, φωτιά στο Λονδίνο, κ.α.), άσχετα tweet για ξεματιάσμα, κ.α.

Πίνακας 15: Σχολιασμός ευρημάτων ελληνικών tweet κατά την διάρκεια της κατηγοριοποίησης.

Ο αριθμός των document ανά κλάση φαίνεται παρακάτω και όπως παρατηρούμε, καμία κλάση δεν περιέχει λιγότερα από 10 document, οπότε το data augmentation σε αυτή την περίπτωση θα αποφευχθεί:

Datasets	Caution and advice	Displaced people and evacuations	Donation needs or offers	Infrastructure and utilities damage	Injured or dead people	Missing trapped or found people	Sympathy emotional support	Other useful information	Not related or irrelevant	# Σύνολο
2018 Attica	308	19	2.897	131	555	2.087	1.749	3.410	3.444	14.600

Πίνακας 16: Αριθμός document ανά class για το Attica 2018.

Πέραν από τα «στραβά» που διακρίναμε στο αγγλικό κομμάτι και ισχύουν και εδώ, παρατηρούμε ακόμα δύο νέες δυσκολίες που κληθούν να ανταπεξέλθουν τα μοντέλα ML:

1. Τα δίδυμα που βρίσκονται σε τρεις κατηγορίες(αρχικά στα Missing, μετά στα Dead και τέλος στο Sympathy). Τα μοντέλα θα δυσκολεύονταν να αποφανθούν σε ποια κατηγορία ανήκουν εάν λαμβάναμε feature μόνο ως unigrams. Αυτό έρχονται να επιδιορθώσουν τα bigrams.
2. Διαφορετικές σημασίες λέξεων, όπως «το Μάτι» που είναι η φυσική περιοχή, η έκφραση «το μάτι» που μπορεί να σημαίνει το μέρος το σώματος ή «το μάτι» με την έννοια του ξεματιάσματος.

Επεξήγηση κώδικα

Η διαδικασία & ο κώδικας δεν διαφέρει σε σύγκριση με το αγγλικό κομμάτι, για αυτό κι δεν θα αναφερθεί ξανά, παραμόνο σε δύο σημεία που θα εκτελεστούν διαφορετικά:

1. **Γραμμές 180-182:** εδώ φορτώνεται το μοντέλο ελληνικών διανυσματικών λέξεων με το Spacy από τα οποία στην συνέχεια θα απλοποιηθούν οι λέξεις σε μορφή λεξικού.
2. **Γραμμές 145-150:** εδώ, επειδή το NLTK δεν υποστηρίζει την ελληνική γλώσσα(όπως φαίνεται σε ελληνική έρευνα [52]), γίνεται lemmatizing των όρων με το Spacy

Βήμα	Document
Αρχικό	RT @mairiboofanpage: 🆘 SOS ⚠️ (κάντε ένα #Retweet για να το δουν όσοι πιο πολλοί γίνεται. ΣΗΜΑΝΤΙΚΟ) - Για όσους δεν έχουν να διανυκτέρευση προσφέρω διαμέρισμα. Φειδιππίδου 60, Άγιος Παντελεήμονας - 2294 057272
9.1	🆘 SOS ⚠️ (κάντε ένα #Retweet για να το δουν όσοι πιο πολλοί γίνεται. ΣΗΜΑΝΤΙΚΟ) - Για όσους δεν έχουν να διανυκτέρευση προσφέρω διαμέρισμα. Φειδιππίδου 60, Άγιος Παντελεήμονας - 2294 057272
9.2	🆘 SOS ⚠️ (κάντε ένα #Retweet για να το δουν όσοι πιο πολλοί γίνεται. ΣΗΜΑΝΤΙΚΟ) - Για όσους δεν έχουν να διανυκτέρευση προσφέρω διαμέρισμα. Φειδιππίδου 60, Άγιος Παντελεήμονας - 2294 057272
9.3	(παραλείπεται)
9.4	🆘 SOS ⚠️ (κάντε ένα Retweet για να το δουν όσοι πιο πολλοί γίνεται. ΣΗΜΑΝΤΙΚΟ) - Για όσους δεν έχουν να διανυκτέρευση προσφέρω διαμέρισμα. Φειδιππίδου 60, Άγιος Παντελεήμονας - 2294 057272
9.5	🆘 SOS ⚠️ (κάντε ένα Retweet για να το δουν όσοι πιο πολλοί γίνεται. ΣΗΜΑΝΤΙΚΟ) - Για όσους δεν έχουν να διανυκτέρευση προσφέρω διαμέρισμα. Φειδιππίδου 60, Άγιος Παντελεήμονας - 2294 057272
9.6	SOS κάντε ένα Retweet για να το δουν όσοι πιο πολλοί γίνεται ΣΗΜΑΝΤΙΚΟ Για όσους δεν έχουν να διανυκτέρευση προσφέρω διαμέρισμα Φειδιππίδου 60 Άγιος Παντελεήμονας 2294 057272
9.7	SOS κάντε ένα Retweet για να το δουν όσοι πιο πολλοί γίνεται ΣΗΜΑΝΤΙΚΟ Για όσους δεν έχουν να διανυκτέρευση προσφέρω διαμέρισμα Φειδιππίδου 60 Άγιος Παντελεήμονας 2294 057272
9.8	SOS κάντε ένα Retweet για να το δουν όσοι πιο πολλοί γίνεται ΣΗΜΑΝΤΙΚΟ Για όσους δεν έχουν να διανυκτέρευση προσφέρω διαμέρισμα Φειδιππίδου 60 Άγιος Παντελεήμονας 2294 057272
9.9	sos κάντε ένα retweet για να το δουν όσοι πιο πολλοί γίνεται σημαντικό για όσους δεν έχουν να διανυκτέρευση προσφέρω διαμέρισμα φειδιππίδου 60 άγιος παντελεήμονας 2294 057272

9.10	sos κάντε ένα retweet δω όσοι πιο πολλός γίνομαι σημαντικός όσους έχω διανυκτέρευση προσφέρω διαμέρισμα φειδιππίδου 60 άγιο παντελεήμονας 2294 057272
9.11	sos κάντε ένα retweet δω όσοι πιο πολλός γίνομαι σημαντικός όσους έχω διανυκτέρευση προσφέρω διαμέρισμα φειδιππίδου 60 άγιο παντελεήμονας 2294 057272

Πίνακας 17: Βηματική εξέλιξη preprocessing ενός document στα ελληνικά.

Εκπαίδευση & σύγκριση μοντέλων ML

Η όλη διαδικασία & ο κώδικας είναι ολόιδιος με την αγγλική υλοποίηση οπότε δεν θα αναφερθώ σε λεπτομέρειες.

Τα 15 πιο σημαντικά feature είναι:

# εμφανίσεων	Όρος(term)	Αριθμητικό διάνυσμα(σκορ)
58067	ψάχνω	0.040686
41386	πατέρας	0.038539
10985	ανάγκη	0.038525
30671	κοριτσάκι	0.035634
47926	σκάφος	0.031141
18322	δίδυμος	0.027942
37523	νεκρός	0.027098
26791	κάηκε	0.026819
3626	hayatını	0.024611
16680	βρεθώ	0.024525
33885	μήπως	0.023853
23783	επικοινωνήσω	0.023531
16468	βρέθηκε	0.023101
18325	δίδυμος κοριτσάκι	0.022845
31674	κόρα	0.022590

Πίνακας 18: Τα 15 πιο σημαντικά feature για τις πυρκαγιές Αττικής 2018 (φθίνουσα σειρά).

Τα αποτελέσματα των ROC-AUC είναι ικανοποιητικά και φαίνονται στον παρακάτω πίνακα:

Datasets	Classifier	Caution and advice	Displaced people and evacuations	Donation needs or offers	Infrastructure and utilities damage	Injured or dead people	Missing trapped or found people	Sympathy emotional support	Other useful information	Not related irrelevant	Mean macro-average ROC-AUC
2018 Attica Wildfires	Size(%)	2.10%	0.13%	19.84%	0.89%	3.80%	14.29%	11.97%	23.35%	23.58%	99.95%
	SVM	0.91	0.79	0.97	0.94	0.93	0.98	0.91	0.89	0.91	0.91
	NB	0.88	0.61	0.95	0.92	0.93	0.96	0.88	0.83	0.88	0.87
	RF	0.92	0.68	0.98	0.95	0.94	0.99	0.93	0.91	0.93	0.91

Πίνακας 19: Συγκριτικά σκορ macro-average ROC-AUC για την ελληνική υλοποίηση.

Σε αυτό το σημείο έχει ολοκληρωθεί η ελληνική υλοποίηση.

Συμπεράσματα & Προοπτικές

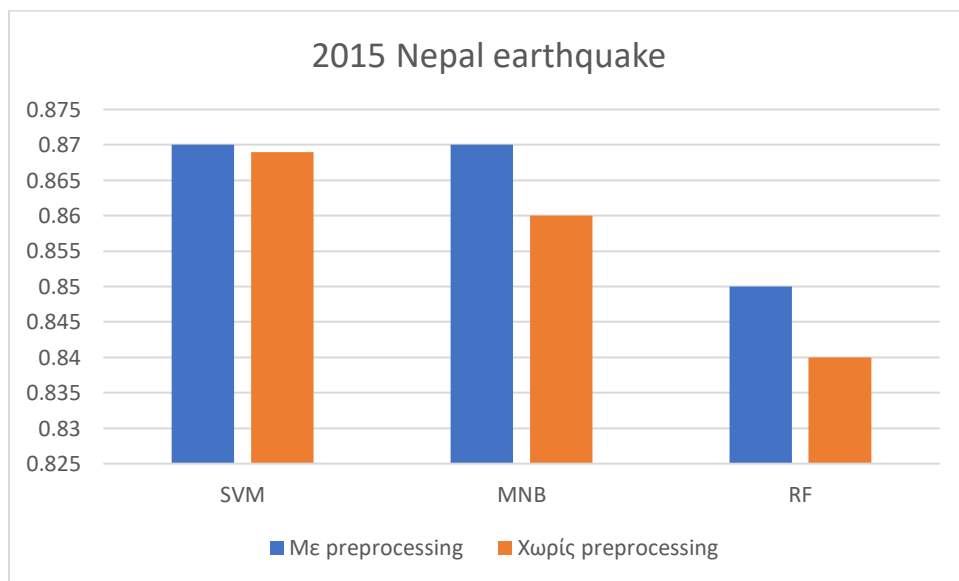
Από την εμπειρική εφαρμογή των δύο περιπτώσεων βλέπουμε πως καλώς υπάρχει ενδιαφέρον για [Data Science for Social Good](#). Οι άνθρωποι πράγματι καταφεύγουν στα κοινωνικά μέσα όταν υπάρχει μια κρίση, είτε για να ψάξουν πληροφορίες, είτε για να ζητήσουν βοήθεια, είτε για να εκφράσουν την υποστήριξή τους.

Παρότι υπάρχουν τα «στραβά» που αναφέρθηκαν κατά την υλοποίηση αυτών, τα μοντέλα καταφέρνουν σε πολύ ικανοποιητικό βαθμό να κατηγοριοποιήσουν πληροφορία χρήσιμη για έναν εθελοντή ή για τους διασώστες και θα πρέπει να βλέπουμε περισσότερες εφαρμογές και πρωτοβουλίες που δυνητικά μπορούν να σώσουν ζωές. Σε μελλοντικά βήματα θα μπορούσε να χρησιμοποιηθεί AI που όλα αυτά τα tweet που έχουν κατηγοριοποιηθεί να παράγει μηνύματα με τα οποία περιγράφει με φυσική γλώσσα την φύση της κρίσης ή να κατασκευάζει αυτόματα την λίστα με τους αγνοουμένους.

Οι περισσότερες παρατηρήσεις έχουν ήδη γίνει, αλλά σε αυτό το σημείο θα γίνουν 3ς τεχνικές παρατηρήσεις πατώντας στα dataset που χρησιμοποιήθηκαν.

Το preprocessing είναι πάντα καλό

Παρακάτω απεικονίζεται ένα δείγμα από τον σεισμό του 2015 στο Νεπάλ. Όπως διαφαίνεται, τα όλα τα μοντέλα αποδίδουν καλύτερα ή τουλάχιστον το ίδιο. Στο διάγραμμα φαίνεται πως οι διαφορές τους είναι της τάξης του 1% όμως ακόμα και το 1% σε πολλά προβλήματα είναι πολύ σημαντικό. Σε αυτό το πρόβλημα που παρουσίασε η παρούσα διπλωματική το 1% μπορεί να είναι ζήτημα ζωής ή θανάτου, άρα λοιπόν σε τέτοιες περιπτώσεις πρέπει σίγουρα να προ επεξεργασία για να αποκομίσουμε τα οφέλη που συζητήθηκαν στις ενότητές τους.

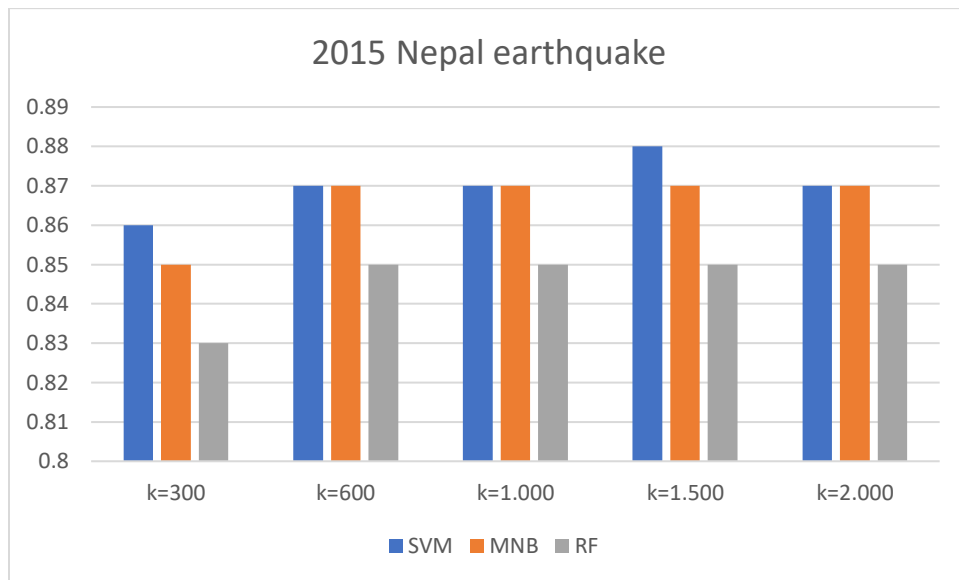


Πίνακας 20: Απεικόνιση διαφοράς απόδοσης μοντέλων με preprocessing και χωρίς preprocessing.

Πρέπει να επιλέγεται το σωστό πλήθος των επιλεγμένων feature

Κατά την διαδικασία feature selection υπάρχουν πολλοί προβληματισμοί, όπως αναφέρθηκαν στην εισαγωγή & στην υλοποίηση. Όπως προαναφέρθηκε, ένας από αυτούς είναι το πόσα feature να επιλέξω για το dataset μου. Η απάντηση είναι πως δεν υπάρχει τέλειος αριθμός και πως η χρυσή τομή θα πρέπει να βρεθεί με την διαδικασία της προσπάθειας & της αποτυχίας (trial & error). Πρέπει πάντα να υπάρχει και η σκέψη της χρησιμοποίησης των υπολογιστικών πόρων, και να βρούμε την ισορροπία ανάμεσα σε μοντέλα που θέλουν

μέρες να εκπαιδευτούν και στην ακρίβεια αυτών. Όπως είναι λογικό, δεν υπάρχει άπλετος χρόνος ούτε ο άνθρωπος μπορεί να δημιουργήσει χρόνο.



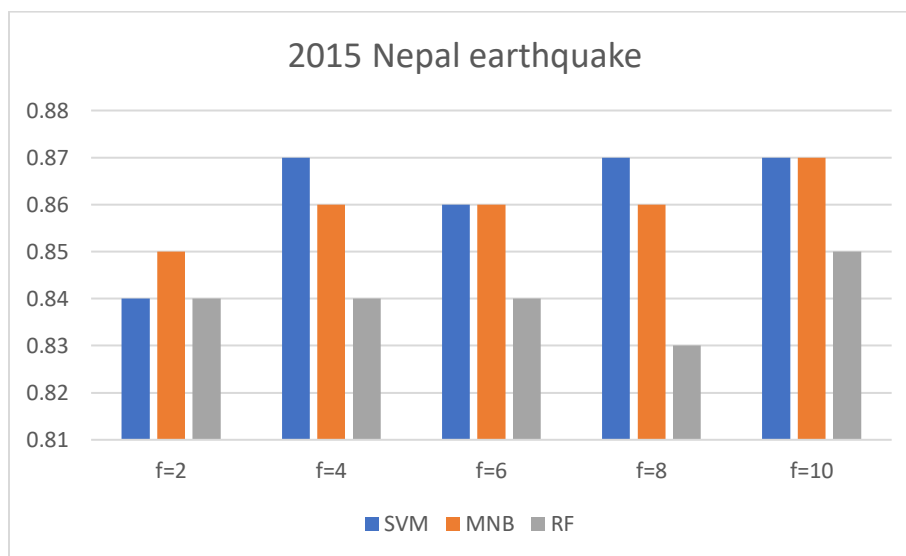
Πίνακας 21: Διάγραμμα εξέλιξης σκορ μοντέλων όσο επιλέγονται περισσότερα feature

Στον παραπάνω πίνακα διακρίνουμε πως είτε επιλέγαμε 500-600 feature είτε 1000 τότε θα είχαμε την ίδια επίδοση. Άρα μάλλον θα έπρεπε να επιλέξουμε 500 και να γλυτώσουμε πολύτιμο υπολογιστικό χρόνο. Μια αυτόματη μέθοδος να βρεθεί ένας τέτοιος αριθμός είναι με Grid Search που δεν υλοποιήθηκε σε αυτή την διπλωματική ώστε να μην υπάρξει καμία παραμετροποίηση αλλά υπάρχει σαν κώδικας στον φάκελο με όλα τα αρχεία της διπλωματικής που αναφέρεται στην αρχή της ενότητας της υλοποίησης και προστέθηκαν ενδεικτικά στα παραρτήματα Η', Θ' και Ι'. Οι καλύτερες hyperparameters παρέμειναν στον κώδικα του παραρτήματος Ζ' μέσα σε σχόλια ως αναφορά.

Προσοχή όμως! Η πολύ παραμετροποίηση & tuning οδηγεί σε overfitting, κι αυτό επειδή κανένα dataset δεν είναι τέλειο και πάντα θα υπάρχει μέσα θόρυβος και έτσι οι αλγόριθμοι γίνονται ανεκτικοί σε συγκεκριμένους θορύβους και ύστερα τους ψάχνουν σε ξένα dataset.

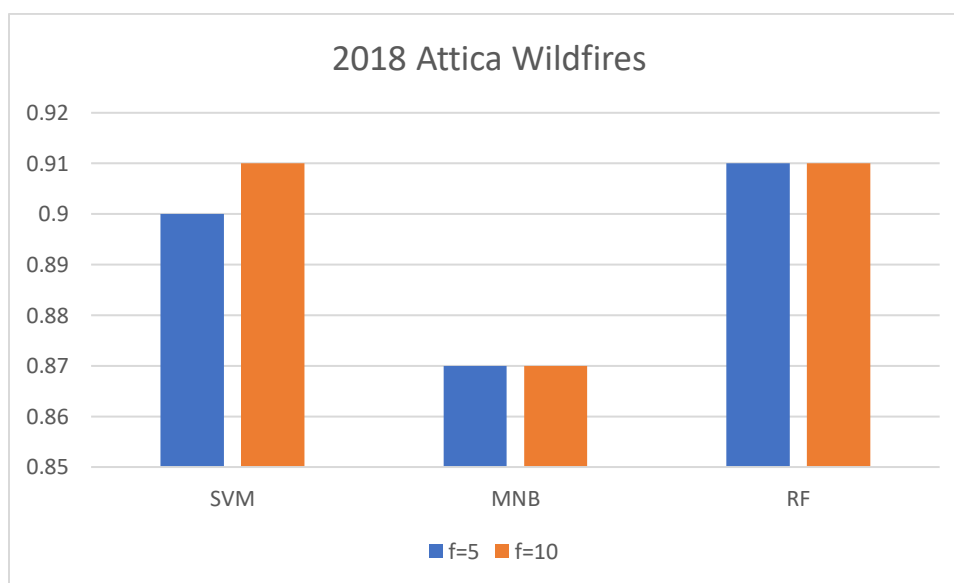
Πρέπει να επιλέγεται το σωστό πλήθος των fold

Για όλους τους λόγους που αναφέρθηκαν παραπάνω, τα ίδια ισχύουν και για τα fold κατά το CV. Όπως φαίνεται στο παρακάτω διάγραμμα, καλώς ορίζει η θεωρία για μικρά dataset να επιλέγονται 10 ή παραπάνω folds.



Πίνακας 22: Βελτίωση ROC-AUC σκορ καθώς έχουμε περισσότερα split(με κόστος στον χρόνο εκτέλεσης)

Όπως φαίνεται και στο επόμενο διάγραμμα για μεγάλα dataset, όπως είναι το ελληνικό που πραγματοποιήθηκε σε αυτή την διπλωματική και είναι ελεύθερο για λήψη στον φάκελο της διπλωματικής, ακόμα 5fold-CV είναι αρκετό να παράγει καλά αποτελέσματα και σε χρόνο πολύ πιο γρήγορο από ό,τι θα ήταν για 10 folds.



Πίνακας 23: Μικρή βελτίωση ROC-AUC σκορ σε μεγάλα dataset καθώς έχουμε περισσότερα split(με μεγάλο κόστος στον χρόνο εκτέλεσης)

Ενδεικτικά, στον ίδιο υπολογιστή, το 10fold-CV διήρκεσε περίπου 8 -10 ώρες ενώ το 5fold-CV μόλις 50'.

Την ημέρα της παρουσίασης συνέβη ο [φονικός σεισμός στο Αιγαίο](#). Ένεκα αυτού στο παράρτημα Κ' γίνεται μια δοκιμή της επιτυχίας των μοντέλων που αναπτύχθηκαν, εκτός του στόχου της παρούσας διπλωματικής.

- Τέλος -

- [1] Κ. Νικόλαος, «ΑΝΑΠΤΥΞΗ ΗΛΕΚΤΡΟΝΙΚΟΥ ΜΑΘΗΜΑΤΟΣ ΓΙΑ ΤΗΝ ΔΙΔΑΣΚΑΛΙΑ ΑΡΧΩΝ ΤΗΣ ΣΤΑΤΙΣΤΙΚΗΣ ΣΤΙΣ ΚΟΙΝΩΝΙΚΕΣ ΕΠΙΣΤΗΜΕΣ,» 2009. [Ηλεκτρονικό]. Available: <http://ikee.lib.auth.gr/record/112873/files/%CE%9A%CE%9F%CE%9C%CE%9C%CE%91%CE%A4%CE%91%CE%A3.pdf>. [Πρόσβαση 2020].
- [2] Anonymous, «Μάθηση,» [Ηλεκτρονικό]. Available: <https://el.wikipedia.org/wiki/%CE%9C%CE%AC%CE%B8%CE%B7%CF%83%CE%B7>. [Πρόσβαση 2020].
- [3] D. Pickell, «Structured vs Unstructured Data – What's the Difference?,» G2, 16 November 2018. [Ηλεκτρονικό]. Available: <https://learn.g2.com/structured-vs-unstructured-data>. [Πρόσβαση 2020].
- [4] Anonymous, «Data mining,» Wikipedia, [Ηλεκτρονικό]. Available: https://en.wikipedia.org/wiki/Data_mining. [Πρόσβαση 2020].
- [5] Anonymous, «Big Data - What it is and why it matters,» SAS, [Ηλεκτρονικό]. Available: https://www.sas.com/en_us/insights/big-data/what-is-big-data.html. [Πρόσβαση 2020].
- [6] D. Sayce, «The Number of tweets per day in 2020,» David Sayce, [Ηλεκτρονικό]. Available: <https://www.dsayce.com/social-media/tweets-day/#:~:text=Every%20second%2C%20on%20average%2C%20around%206%2C000%20tweets%20are%20tweeted%20on,200%20billion%20tweets%20per%20year..> [Πρόσβαση 2020].
- [7] J. Brownlee, «Feature Selection For Machine Learning in Python,» Machine Learning Mastery, 20 May 2016. [Ηλεκτρονικό]. Available: <https://machinelearningmastery.com/feature-selection-machine-learning-python/>. [Πρόσβαση 2020].
- [8] J. Brownlee, «Information Gain and Mutual Information for Machine Learning,» Machine Learning Mastery, 16 October 2019. [Ηλεκτρονικό]. Available: <https://machinelearningmastery.com/information-gain-and-mutual-information/>. [Πρόσβαση 2020].
- [9] Anonymous, «Mutual Information Classifier,» Scikit Learn, [Ηλεκτρονικό]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.mutual_info_classif.html. [Πρόσβαση 2020].
- [10] C. D. Manning, P. Raghavan και H. Schütze, «An Introduction to Information Retrieval,» Cambridge, England, Cambridge University Press, 2009, pp. 1-5.
- [11] C. D. Manning, P. Raghavan και H. Schütze, σε *An Introduction to Information Retrieval*, Cambridge, England, Cambridge University Press, 2009, pp. 146-152.
- [12] «Artificial Intelligence - What is it and why it matters,» SAS, [Ηλεκτρονικό]. Available: https://www.sas.com/en_us/insights/analytics/what-is-artificial-intelligence.html. [Πρόσβαση 2020].
- [13] R. Anyoha, «The History of Artificial Intelligence,» Harvard Science in the News, 28 August 2017. [Ηλεκτρονικό]. Available: <http://sitn.hms.harvard.edu/flash/2017/history-artificial-intelligence/>. [Πρόσβαση 2020].
- [14] A. Ng, «Introduction to AI,» Coursera, [Ηλεκτρονικό]. Available: <https://www.coursera.org/learn/ai-for-everyone/lecture/SRWLN/week-1-introduction>. [Πρόσβαση 2020].

- [15] J. C. Wong, «'We won't be war profiteers': Microsoft workers protest \$480m army contract,» The Guardian UK, 22 Febuary 2019. [Ηλεκτρονικό]. Available: <https://www.theguardian.com/technology/2019/feb/22/microsoft-protest-us-army-augmented-reality-headsets>. [Πρόσβαση 2020].
- [16] A. Ng, «A realistic view of AI,» Coursera, [Ηλεκτρονικό]. Available: <https://www.coursera.org/lecture/ai-for-everyone/a-realistic-view-of-ai-BYHnY>. [Πρόσβαση 2020].
- [17] I. Wladawsky-Berger, «What Machine Learning Can and Cannot Do,» Wall Street Journal, 18 July 2018. [Ηλεκτρονικό]. Available: <https://blogs.wsj.com/cio/2018/07/27/what-machine-learning-can-and-cannot-do/>. [Πρόσβαση 2020].
- [18] S. Pasupuleti, «A Practical Beginner's Guide to Machine Learning,» Towards Data Science, 27 Febuary 2019. [Ηλεκτρονικό]. Available: <https://towardsdatascience.com/a-practical-beginners-guide-to-machine-learning-dcc9cf5c692b>. [Πρόσβαση 2020].
- [19] Anonymous, «Machine Learning Definition,» DeepAI, [Ηλεκτρονικό]. Available: <https://deepai.org/machine-learning-glossary-and-terms/machine-learning>. [Πρόσβαση 2020].
- [20] J. Brownlee, «14 Different Types of Learning in Machine Learning,» Machine Learning Mastery, 19 November 2019. [Ηλεκτρονικό]. Available: <https://machinelearningmastery.com/types-of-learning-in-machine-learning/>. [Πρόσβαση 2020].
- [21] Anonymous, «AI Against the Coronavirus,» Deeplearning.ai, 1 April 2020. [Ηλεκτρονικό]. Available: <https://blog.deeplearning.ai/blog/the-batch-ai-against-coronavirus-datasets-voice-cloning-for-the-masses-finding-unexploded-bombs-seeing-see-through-objects-optimizing-training-parameters>. [Πρόσβαση 2020].
- [22] A. Ng, «What machine learning can and cannot do,» Coursera, [Ηλεκτρονικό]. Available: <https://www.coursera.org/lecture/ai-for-everyone/what-machine-learning-can-and-cannot-do-rv1fW>. [Πρόσβαση 2020].
- [23] R. Lao, «A Beginner's Guide to Machine Learning,» Towards Data Science, 3 July 2018. [Ηλεκτρονικό]. Available: <https://towardsdatascience.com/a-beginners-guide-to-machine-learning-5d87d1b06111>. [Πρόσβαση 2020].
- [24] Anonymous, «Metrics and scoring: quantifying the quality of predictions,» Scikit Learn, [Ηλεκτρονικό]. Available: https://scikit-learn.org/stable/modules/model_evaluation.html. [Πρόσβαση 2020].
- [25] S. Narkhede, «Understanding AUC - ROC Curve,» Towards Data Science, 26 June 2018. [Ηλεκτρονικό]. Available: <https://towardsdatascience.com/understanding-auc-roc-curve-68b2303cc9c5>. [Πρόσβαση 2020].
- [26] Anonymous, «Receiver Operating Characteristic (ROC),» Scikit Learn, [Ηλεκτρονικό]. Available: https://scikit-learn.org/stable/auto_examples/model_selection/plot_roc.html. [Πρόσβαση 2020].
- [27] R. Gandhi, «Support Vector Machine — Introduction to Machine Learning Algorithms,» Towards Data Science, 7 July 2018. [Ηλεκτρονικό]. Available: <https://towardsdatascience.com/support-vector-machine-introduction-to-machine-learning-algorithms-934a444fca47>. [Πρόσβαση 2020].
- [28] C. C. Aggarwal, «IBM T. J. Watson Research Center,» σε *Machine Learning for Text*, Yorktown Heights, NY, USA, Springer, 2018, pp. 193-195.
- [29] C. Dinant, «What's so naive about naive Bayes'?,» Towards Data Science, 9 May 2018. [Ηλεκτρονικό]. Available: <https://towardsdatascience.com/whats-so-naive-about-naive-bayes-58166a6a9eba>. [Πρόσβαση 2020].

- [30] J. VanderPlas, «In Depth: Naive Bayes Classification,» Python Data Science Handbook, [Ηλεκτρονικό]. Available: <https://jakevdp.github.io/PythonDataScienceHandbook/05.05-naive-bayes.html>. [Πρόσβαση 2020].
- [31] P. E. Staff, «Implementing 3 Naive Bayes classifiers in scikit-learn,» Packt Publishing, 18 May 2018. [Ηλεκτρονικό]. Available: <https://hub.packtpub.com/implementing-3-naive-bayes-classifiers-in-scikit-learn/>. [Πρόσβαση 2020].
- [32] C. C. Aggarwal, «IBM T. J. Watson Research Center,» σε *Machine Learning for Text*, Yorktown Heights, NY, USA, Springer, 2018, pp. 158-163.
- [33] M. B. Fraj, «In Depth: Parameter tuning for Random Forest,» Medium, 21 December 2017. [Ηλεκτρονικό]. Available: <https://medium.com/all-things-ai/in-depth-parameter-tuning-for-random-forest-d67bb7e920d>. [Πρόσβαση 2020].
- [34] C. C. Aggarwal, «IBM T. J. Watson Research Center,» σε *Machine Learning for Text*, Yorktown Heights, NY, USA, Springer, 2018, pp. 395-399.
- [35] H. Waqas και M. Imran, «CampFireMissing: An Analysis of Tweets About Missing and Found People From California Wildfires,» SCRAM, Valencia, Spain, 2019.
- [36] A. Alharbi και M. Lee, «Crisis Detection from Arabic Tweets,» *Proceedings of the 3rd Workshop on Arabic Corpus Linguistics*, 2019.
- [37] K. Kireyev, L. Palen και K. M. Anderson, «Applications of Topics Models to Analysis of Disaster-Related Twitter Data,» *NIPS workshop on applications for topic models: text and beyond*, τόμ. 1, 2009.
- [38] R. Kshirsagar, R. Morris και S. R. Bowman, «Detecting and Explaining Crisis,» *arXiv preprint arXiv:1705.09585*, 2017.
- [39] D. T. Nguyen, K. A. A. Mannai, S. Joty, H. Sajjad, M. Imran και P. Mitra, «Robust Classification of Crisis-Related Data on Social Networks Using Convolutional Neural Networks,» *Eleventh International AAAI Conference on Web and Social Media*, 2017.
- [40] M. Yu, Q. Huang, H. Qin, C. Scheele και C. Yang, «Deep learning for real-time social media text classification for situation awareness – using Hurricanes Sandy, Harvey, and Irma as case studies,» *International Journal of Digital Earth*, pp. 1-18, 2019.
- [41] F. Abel, C. Hauff, G.-J. Houben, K. Tao και R. Stronkman, «Twitcident: Fighting Fire with Information from Social Web Streams,» *Proceedings of the 21st International Conference on World Wide Web*, pp. 305-308, 2012.
- [42] M. A. Cameron, R. Power, B. Robinson και J. Yin, «Emergency Situation Awareness from Twitter for Crisis Management,» *Proceedings of the 21st International Conference on World Wide Web*, pp. 695-698, 2012.
- [43] N. Öztürk και S. Ayvaz, «Sentiment analysis on Twitter: A text mining approach to the Syrian refugee crisis,» *Telematics and Informatics*, τόμ. 1, αρ. 35, pp. 136-147, 2017.
- [44] S. Stieglitz, M. Mirbabaie και T. Potthof, «Crisis Communication on Twitter during a Global Crisis of Volkswagen - The Case of "Dieselgate",» *Proceedings of the 51st Hawaii International Conference on System Sciences*, 2018.
- [45] M. Imran, P. Mitra και C. Castillo, «Twitter as a Lifeline: Human-annotated Twitter Corpora for NLP of Crisis-related Messages,» *arXiv preprint arXiv:1605.05894*, 2016.
- [46] G. Coppersmith, R. Leary, P. Crutchley και A. Fine, «Natural Language Processing of Social Media as Screening for Suicide Risk,» *Biomedical informatics insights*, τόμ. 10, αρ. 1178222618792860, 2018.

- [47] C. G. Walsh, J. D. Ribeiro και J. C. Franklin, «Predicting Risk of Suicide Attempts Over Time Through Machine Learning,» *Clinical Psychological Science*, τόμ. 5, αρ. 3, pp. 457-469, 2017.
- [48] J. Wei και K. Zou, «Eda: Easy data augmentation techniques for boosting performance on text classification tasks,» arXiv preprint arXiv:1901.11196, Protago Labs Research, Tysons Corner, Virginia, USA, 2019.
- [49] J. Wei, «These are the Easiest Data Augmentation Techniques in Natural Language Processing you can think of — and they work,» Towards Data Science, 3 Febuary 2019. [Ηλεκτρονικό]. Available: <https://towardsdatascience.com/these-are-the-easiest-data-augmentation-techniques-in-natural-language-processing-you-can-think-of-88e393fd610>. [Πρόσβαση 2020].
- [50] Anonymous, «Cross-validation: evaluating estimator performance,» Scikit Learn, [Ηλεκτρονικό]. Available: https://scikit-learn.org/stable/modules/cross_validation.html. [Πρόσβαση 2020].
- [51] Anonymous, «OneVsRest Classifier,» Scikit Learn, [Ηλεκτρονικό]. Available: <https://scikit-learn.org/stable/modules/generated/sklearn.multiclass.OneVsRestClassifier.html>. [Πρόσβαση 2020].
- [52] Σ. Β. Λεωνίδας, «Μελέτη και έλεγχος του Python Natural Language Toolkit στην Ελληνική γλώσσα,» Εργαστήριο Ενσύρματης Τηλεπικοινωνίας - Τμήμα Ηλεκτρολόγων Μηχανικών & Τεχνολογίας Υπολογιστών - Πανεπιστήμιο Πατρών, Πάτρα, 2012.
- [53] Anonymous, «Beginner's Guide to Machine Learning with Python,» Codecus, 10 April 2019. [Ηλεκτρονικό]. Available: <https://codequs.com/p/HJan4TcF4/beginner-s-guide-to-machine-learning-with-python>. [Πρόσβαση 2020].

Παράρτημα Α – Terms of use.txt

=====

Terms of Use

=====

The Crisis Computing team at Qatar Computing Research Institute (QCRI) collects data for research on humanitarian computing. This is publicly-available data collected using publicly-available application programming interfaces (APIs).

IN NO EVENT SHALL QCRI OR QCRI's AFFILIATES BE LIABLE TO THE REQUESTER OR TO THE USER OF THE DATA SUPPLIED FOR ANY SPECIAL, INCIDENTAL, PUNITIVE, INDIRECT, EXEMPLARY, OR CONSEQUENTIAL DAMAGES. THIS DATA IS PROVIDED ON AN "AS IS" BASIS AND YOUR USE OF IT IS AT YOUR SOLE RISK.

To the extent permitted by applicable law, you agree to defend, indemnify and hold harmless QCRI, and its parent QF, officers, directors, employees and agents, from and against any and all claims, damages, obligations, losses, liabilities, costs or debt, and expenses (including but not limited to

reasonable attorney's fees) arising from: (i) your use of and access to this data; (ii) your violation of any term of these terms in this Agreement; (iii) your violation of any third party right, including without limitation any copyright, property, or privacy right; or (iv) any claim that your Content caused

damage to a third party. This defense and indemnification obligation will survive this Agreement and your use of the data.

By downloading or using any of the material available in this dataset, you automatically acknowledge to the following conditions:

- You will not defame, abuse, harass, stalk, threaten or otherwise violate the legal rights (such as rights of privacy and publicity) of others;
- You will not publish, post, distribute or disseminate any defamatory, infringing, obscene, indecent or unlawful material or information;
- You will use the data only for research on humanitarian computing;
- You will keep the contents of these dataset(s) confidential (you can always share the tweet-ids);
- The dataset(s) will be deleted by you upon our request at any time (you can always keep the tweet-ids);
- The dataset(s) will be deleted by you upon completion of the research (you can always keep the tweet-ids);
- You will cite the corresponding paper in derived publications;

Παράρτημα Β – Ειδησεογραφική βιβλιογραφία για τις πυρκαγιές τις Αττικής 2018

- https://el.wikipedia.org/wiki/%CE%9D%CE%AD%CE%B1_%CE%9C%CE%AC%CE%BA%CF%81%CE%B7 (πληθυσμός Νέας Μάρκης)
- https://el.wikipedia.org/wiki/%CE%9C%CE%AC%CF%84%CE%B9_%CE%91%CF%84%CF%84%CE%B9%CE%BA%CE%AE%CF%82 (περιγραφή περιοχής Μάτι)
- <https://www.iefimerida.gr/news/434243/fotia-sto-mati-stoiheia-oti-xekinise-apo-ameleia-idioti-ekaige-xyla-kai-horta-eikones> (αιτία φωτιάς)
- https://el.wikipedia.org/wiki/%CE%A0%CF%85%CF%81%CE%BA%CE%B1%CE%B3%CE%B9%CE%AD%CF%82_%CF%83%CF%84%CE%B7%CE%BD_%CE%91%CF%84%CF%84%CE%B9%CE%BA%CE%AE_%CF%84%CE%BF_2018 (εξέλιξη της πυρκαγιάς και επακόλουθα)
- <https://www.youtube.com/watch?v=GNLUNFerCOI&list=WL&index=7&t=0s> (δοκιμαντέρ ΣΚΑΙ)
- <https://www.tovima.gr/2020/07/23/society/fotia-sto-mati-ta-102-thymata-pou-den-prepei-na-ksexasoume-pote-ola-ta-onomata/> (ονόματα θυμάτων)
- <https://www.kathimerini.gr/977325/interactive/epikairothta/ereynes/to-xroniko-ths-e8nikhs-tragwdias#secondPage>

Παράρτημα Γ1 – requirements.txt

```
astroid==2.4.2
backcall==0.2.0
blis==0.4.1
catalogue==1.0.0
certifi==2020.6.20
chardet==3.0.4
click==7.1.2
colorama==0.4.3
cycller==0.10.0
cymem==2.0.3
decorator==4.4.2
idna==2.10
ipython==7.18.1
ipython-genutils==0.2.0
isort==5.5.1
jedi==0.17.2
joblib==0.16.0
kiwisolver==1.2.0
langdetect==1.0.8
lazy-object-proxy==1.5.1
matplotlib==3.3.1
mccabe==0.6.1
murmurhash==1.0.2
nltk==3.5
numpy==1.19.1
oauthlib==3.1.0
pandas==1.1.1
parso==0.8.0
pickleshare==0.7.5
Pillow==7.2.0
plac==1.1.3
preshed==3.0.2
prompt-toolkit==3.0.7
Pygments==2.6.1
```

```
pylint==2.6.0
pyparsing==2.4.7
PySocks==1.7.1
python-dateutil==2.8.1
pytz==2020.1
regex==2020.7.14
requests==2.24.0
requests-oauthlib==1.3.0
rope==0.17.0
scikit-learn==0.23.2
scipy==1.5.2
seaborn==0.10.1
six==1.15.0
spacy==2.3.2
srsly==1.0.2
thinc==7.4.1
threadpoolctl==2.1.0
toml==0.10.1
tqdm==4.48.2
traitlets==5.0.3
tweepy==3.9.0
tweet-preprocessor==0.6.0
twitter-text-python==1.1.1
urllib3==1.25.10
wasabi==0.8.0
wcwidth==0.2.5
wincertstore==0.2
wrapt==1.12.1
```

Στο τερματικό:

```
python -m spacy download el_core_news_sm
```

Παράρτημα Γ2 – install-packages.py

```
pip install astroid backcall blis catalogue certifi chardet click colorama cycler decorator el-core-news-  
lg idna ipython ipython-genutils isort jedi joblib kiwisolver langdetect lazy-object-  
proxy matplotlib mccabe murmurhash nltk numpy oauthlib pandas parso pickleshare Pillow pip plac preshed prompt-  
toolkit Pygments pylint pyparsing PySocks python-dateutil pytz regex requests requests-oauthlib rope scikit-  
learn scipy seaborn setuptools six threadpoolctl toml tqdm traitlets tweepy tweet-preprocessor twitter-text-  
python urllib3 wasabi wcwidth wrapt
```

Στο τερματικό:

```
python -m spacy download el_core_news_sm
```

Παράρτημα Δ – twitterCredentials.py

```
# Twitter API credentials
ACCESS_TOKEN = "1202188394027257857-6KJoXXXXXXXXXXXXXXXXXXXXX"
ACCESS_TOKEN_SECRET = "c2GFCfwV6h7LtzMS42YMXXXXXXXXXXXXXXXXXXXXX"
CONSUMER_KEY = "lywHAURmf3RdrXXXXXXXXXXXXX"
CONSUMER_SECRET = "oXMGGoDojjCdvQ5RMuBqLSZKjKgPAVhXXXXXXXXXXXXXXXXXXXXX"
```

3

³ Τα κλειδιά είναι λογοκριμένα για λόγους ασφαλείας.

Παράρτημα Ε – collectTweets.py

```
import twitterCredentials
import tweepy
import pandas as pd

# Authenticate app
auth = tweepy.OAuthHandler(twitterCredentials.CONSUMER_KEY, twitterCredentials.CONSUMER_SECRET)
auth.set_access_token(twitterCredentials.ACCESS_TOKEN, twitterCredentials.ACCESS_TOKEN_SECRET)
api = tweepy.API(auth, wait_on_rate_limit=True, wait_on_rate_limit_notify=True)

df = pd.DataFrame(columns=['tweet_id', 'tweet_text'])
searchFor = "Μάτι OR Ματι OR Φωτιά OR Φωτια OR Πυρκαγιά OR Πυρκαγια"
applicationName = "ThesisDev"

# Start collecting tweets every hour from day 1
startDate = 201807232100
endDate = 201807232159
for i in range(0,12):
    for status in tweepy.Cursor(api.search_full_archive, applicationName, searchFor, fromDate=str(startDate), toDate=str(endDate)).items(100):
        print(status.id_str, "\t", status.text)
        df = df.append({'tweet_id': status.id_str, 'created_at': status.created_at, 'tweet_text': status.text}, ignore_index=True)
    # Go to next hour
    startDate += 100
    endDate += 100

# Start collecting tweets every hour from day 2
startDate = 201807240000
endDate = 201807240059
for i in range(0,24):
    for status in tweepy.Cursor(api.search_full_archive, applicationName, searchFor, fromDate=str(startDate), toDate=str(endDate)).items(100):
        print(status.id_str, "\t", status.text)
```

```

        df = df.append({'tweet_id': status.id_str, 'created_at': status.created_at, 'tweet_text': status.text}, ignore_index=True)
        # Go to next hour
        startDate += 100
        endDate += 100

# Start collecting tweets every hour from day 3
startDate = 201807250000
endDate = 201807250059
for i in range(0,14):
    for status in tweepy.Cursor(api.search_full_archive, applicationName, searchFor, fromDate=str(startDate), toDate=str(endDate)).items(100):
        print(status.id_str, "\t", status.text)
        df = df.append({'tweet_id': "\"" + status.id_str + "\"", 'created_at': status.created_at, 'tweet_text': status.text}, ignore_index=True)

    # Go to next hour
    startDate += 100
    endDate += 100

# Save tweets to file
df.to_csv("./fetchedTweets/tweets.csv", encoding="utf_8_sig", index=False)

```

```
import pandas as pd
import numpy as np
import csv
import random
import string
import spacy
import re
import nltk
from nltk.corpus import stopwords, wordnet
from nltk.tokenize import word_tokenize
from nltk.stem import PorterStemmer
import preprocessor as p
from collections import Counter
import langdetect as ld
from langdetect import DetectorFactory
from os import listdir
from os.path import isfile, join
from colorama import Fore, Back, Style

def preprocessTweets(corpusFile, targetResultFile, tweetIdColumnName, tweetTextColumnName, labelColumnName, OOV_Dictionary):
    print("Reading file:", corpusFile)
    df = pd.read_csv(corpusFile, encoding="utf-8", skipinitialspace=True, usecols=[tweetIdColumnName, tweetTextColumnName, labelColumnName])

    # Remove duplicate tweets
    df.drop_duplicates(subset=[tweetIdColumnName], keep="first", inplace=True)
    df.reset_index(drop=True, inplace=True)
    # Remove empty tweet records
    df.dropna(subset=[tweetTextColumnName], inplace=True)

    # Calculate number of documents per category
    documentsPerCategory = df[labelColumnName].value_counts().reset_index().to_numpy()
    # Check if there is a category with less than 10 documents and save them
```

```

insufficientCategories = list()
for docs in documentsPerCategory:
    if(docs[1] < 10):
        print(Fore.YELLOW + "Warning: insufficient documents detected ({0}) - oversampling triggered for [{1}].".format(
docs[1], docs[0]), Style.RESET_ALL)
        insufficientCategories.append(docs)
    else:
        continue

insufficientCategories = np.asarray(insufficientCategories)

# If any then populate them with Data Augmentation techniques like Synonyms and shuffling the words
for case in insufficientCategories:
    # Calculate number of documents we need to add to reach an ensemble of 10
    documentsNeeded = 10 - case[1]

    categoryDf = df.loc[df[labelColumnName] == case[0]]

    # Augment data using each and every document that already exists, up to 10
    documentsCreated = 0
    documentsShuffled = 0
    for _, text in categoryDf.iterrows():
        # Stop adding documents if we reached an ensemble of 10
        if documentsCreated == documentsNeeded:
            break

        # Tokenize sentence & tag the part of speech
        tokens = word_tokenize(text[1])
        tagged = np.asarray(nltk.pos_tag(tokens))
        sentence = list()
        for words in tagged:
            # If a word is any form of verb(VB), adjective(JJ) or adverb(RB) then replace it with a random synonym, else
            just append
            if re.match(r"VB.?.?|JJ.?.?|RB.?.?", words[1]):
                try:

```

```

        synonyms = list()
        for syn in wordnet.synsets(words[0]):
            for l in syn.lemmas():
                synonyms.append(l.name())

        sentence.append(synonyms[random.randint(0, len(synonyms))])

        # If no synonym found, just append
        except:
            sentence.append(words[0])
            continue
    else:
        sentence.append(words[0])

    # Append the created document to df
    wholeSentence = ' '.join(sentence)
    df = df.append(pd.DataFrame({tweetIdColumnName:[""] + str(random.randint(0, 1000000)) + ""}, tweetTextColumnName:[wholeSentence], labelColumnName:[text[2]]}), ignore_index=True)
    documentsCreated += 1

    # If by augmenting data we can not reach 10 samples, then shuffle original documents until reach 10
    if case[1] * 2 < 10 and documentsShuffled < 10 - case[1] * 2:
        tokens = word_tokenize(text[1])
        random.shuffle(tokens)

        # Append the shuffled document to df
        shuffledSentence = ' '.join(tokens)
        df = df.append(pd.DataFrame({tweetIdColumnName:[""] + str(random.randint(0, 1000000)) + ""}, tweetTextColumnName:[shuffledSentence], labelColumnName:[text[2]]}), ignore_index=True)
        documentsShuffled += 1

foundLanguages = list()
tweetTexts = df[tweetTextColumnName]
for idx, val in enumerate(tweetTexts):
    # detect language

```

```

try:
    lang = ld.detect(val)
    foundLanguages.append(lang)
    """
    # if language detected is other than greek or english then erase tweet text
    if not lang in "el" and "en":
        val = ""
    """
except:
    # if no text(i.e. only a URL, only an Emoji, etc.) then set language = None
    lang = None
# remove URLs, Mentions, Reserved words (RT, FAV), Smileys
val = p.clean(val)
# remove Emojis
val = emoji_pattern.sub(r'', val)
if lang == "en":
    # tokenize sentence
    text_tokens = word_tokenize(val)
    # replace out of vocabulary words
    val = [word if word not in OOV_Dictionary else OOV_Dictionary[word] for word in text_tokens]
    # join the words again to form the sentence
    val = ' '.join(val)
    # replace consecutive non-ASCII characters with a space
    val = re.sub(r'^\x00-\x7F+', ' ', val)
# remove the # in hashtag
val = re.sub(r'#([^\s]+)', r'\1', val)
# remove punctuation
val = [char for char in val if char not in string.punctuation]
# join the characters again to form the string.
val = ''.join(val)
# remove all single characters
val = ' '.join([w for w in val.split() if len(w) > 1])
# further remove elusive URLs
val = re.sub('http', ' ', val)
# substituting multiple spaces with single space

```

```

val = re.sub(r'\s+', ' ', val, flags=re.I)
# convert text to lower-case
val = val.lower()
# tokenize sentence, remove stopwords
if lang == "en":
    text_tokens = word_tokenize(val)
    val = [word for word in text_tokens if not word in stopwords.words("english")]
    # and perform stemming (available in english language only)
    val = [PorterStemmer().stem(word) for word in val]
    val = ' '.join(val)
elif lang == "el":
    text_tokens = word_tokenize(val)
    val = [word for word in text_tokens if not word in stopwords.words("greek")]
    # and perform lemmatizing
    val = [word.lemma_ for word in greekNLP(' '.join(val))]
    val = ' '.join(val)
else:
    pass
# return as sentence
tweetTexts[idx] = val
# return NaN if string is empty
if tweetTexts[idx] == "":
    tweetTexts[idx] = None

# Remove empty tweet records after processing(if any)
df.dropna(subset=[tweetTextColumnName], inplace=True)

# Export processed file
if Counter(foundLanguages).most_common(1)[0][0] == "el":
    df.to_csv(targetResultFile, encoding="utf_8_sig", index=False)
else:
    df.to_csv(targetResultFile, index=False)
print("Output file:", targetResultFile)

```

```

#### Start program ####
# Optional dataframe view options
pd.set_option('display.max_colwidth', None)
pd.set_option('mode.chained_assignment', None)

# Load out of vocabulary dictionary
with open('./OOV/OOV_Dictionary_V1.0.tsv', 'r', encoding="ISO-8859-1") as f:
    reader = csv.reader(f, delimiter='\t')
    OOV_Dictionary = {r[0].lower(): r[1].lower() for r in reader}
print(Fore.GREEN + "Vocabulary loading complete.", Style.RESET_ALL)
# Load language model
greekNLP = spacy.load("el_core_news_lg")
print(Fore.GREEN + "Greek model loading complete.\n", Style.RESET_ALL)
# Preprocessor options
p.set_options(p.OPT.URL, p.OPT.MENTION, p.OPT.RESERVED, p.OPT.SMILEY)
# Known emojis
emoji_pattern = re.compile("[\"u\"\\U0001F600-\\U0001F64F" u"\\U0001F300-\\U0001F5FF" u"\\U0001F680-\\U0001F6FF" u"\\U0001F1E0-\\U0001F1FF"]+", flags=re.UNICODE)

# Scan for files in directory and preprocess them
onlyFiles = [f for f in listdir("./preprocess/") if.isfile(join("./preprocess/", f))]
for files in onlyFiles:
    preprocessTweets("./preprocess/" + files, "./preprocess/processed/pp_" + files, 'tweet_id', 'tweet_text', 'label', OOV_Dictionary)
    print(Fore.GREEN + "Processing complete.", Style.RESET_ALL)

print(Fore.GREEN + "\nOperations complete.", Style.RESET_ALL)

```

Παράρτημα Z – classifiers.py

```
import warnings
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from colorama import Fore, Back, Style
from os import listdir
from os.path import isfile, join
from pprint import pprint
from sklearn.svm import SVC
from sklearn.naive_bayes import MultinomialNB
from sklearn.ensemble import RandomForestClassifier
from sklearn.multiclass import OneVsRestClassifier
from sklearn.model_selection import StratifiedKFold
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.feature_selection import mutual_info_classif
from sklearn.feature_selection import SelectKBest
from sklearn.metrics import roc_curve, auc
from sklearn.preprocessing import label_binarize
from itertools import cycle

def calculateROC_AUC(clf, X, y, numberOfFolds, printBestFeatures, plotPerFold):
    # Initialize
    ROCscores = list()
    classNames = np.array([])
    vectorizer = CountVectorizer(ngram_range=(1, 2))
    bestFeatures = SelectKBest(mutual_info_classif, k=1000)

    kfold = StratifiedKFold(n_splits=numberOfFolds, shuffle=True, random_state=42)
    for outerFold, (tr_idx, te_idx) in enumerate(kfold.split(X, y)):
        X_train, X_test, y_train, y_test = X[tr_idx], X[te_idx], y[tr_idx], y[te_idx]

        X_train = vectorizer.fit_transform(X_train)
        X_train = bestFeatures.fit_transform(X_train, y_train)
```

```

X_test = vectorizer.transform(X_test)
X_test = bestFeatures.transform(X_test)

# Calculate & print best features
if printBestFeatures:
    featureScores = pd.concat([pd.DataFrame(vectorizer.get_feature_names()), pd.DataFrame(bestFeatures.scores_)], axis=1)

    featureScores.columns = ['Feature', 'Score']
    print(featureScores.nlargest(10, 'Score'))

# Binarize labels for ROC AUC OVR and create a class-names dictionary
classNames = np.unique(y_test)
y_test = label_binarize(y_test, classes=classNames)
n_classes = y_test.shape[1]

# Train the classifier and predict OVR probabilities
classifierScore = OneVsRestClassifier(clf, n_jobs=-1).fit(X_train, y_train).predict_proba(X_test)

# Compute ROC curve and ROC area for each class
fpr = dict()
tpr = dict()
roc_auc = dict()
for i in range(n_classes):
    fpr[i], tpr[i], _ = roc_curve(y_test[:, i], classifierScore[:, i])
    roc_auc[i] = auc(fpr[i], tpr[i])

# Compute micro-average ROC curve and ROC area
fpr["micro"], tpr["micro"], _ = roc_curve(y_test.ravel(), classifierScore.ravel())
roc_auc["micro"] = auc(fpr["micro"], tpr["micro"])

# For macro-average, first aggregate all false positive rates
all_fpr = np.unique(np.concatenate([fpr[i] for i in range(n_classes)]))

# Then interpolate all ROC curves at this points
mean_tpr = np.zeros_like(all_fpr)

```

```

for i in range(n_classes):
    mean_tpr += np.interp(all_fpr, fpr[i], tpr[i])

# Finally average it and compute AUC
mean_tpr /= n_classes

fpr["macro"] = all_fpr
tpr["macro"] = mean_tpr
roc_auc["macro"] = auc(fpr["macro"], tpr["macro"])

ROCscores.append(list(roc_auc.values()))

# Plot all ROC curves
if plotPerFold:
    plt.figure()
    plt.plot(fpr["micro"], tpr["micro"], label='micro-
average ROC curve (area = {0:0.2f})'.format(roc_auc["micro"]), color='deeppink', linestyle=':', linewidth=4)
    plt.plot(fpr["macro"], tpr["macro"], label='macro-
average ROC curve (area = {0:0.2f})'.format(roc_auc["macro"]), color='navy', linestyle=':', linewidth=4)

    colors = cycle(['tab:cyan', 'tab:orange', 'tab:green', 'tab:red', 'tab:purple', 'tab:brown', 'tab:pink', 'tab:gr
ey', 'tab:olive'])
    for i, color in zip(range(n_classes), colors):
        plt.plot(fpr[i], tpr[i], color=color, linewidth=0.5, label='ROC curve of class {0} (area = {1:0.2f})'.format
(i, roc_auc[i]))

    plt.plot([0, 1], [0, 1], 'k--', linewidth=1)
    plt.xlim([0.0, 1.0])
    plt.ylim([0.0, 1.01])
    plt.xlabel('False Positive Rate')
    plt.ylabel('True Positive Rate')
    plt.title('Multiclass ROC curves - fold {0}'.format(outerFold + 1))
    plt.legend(loc="lower right")
    plt.show()

```

```

return classNames, ROCscores

def classify(corpusFile, targetResultFile, tweetTextColumnName, labelColumnName, numberOfFolds, printBestFeatures, plotPerFold):
    df = pd.read_csv(corpusFile, encoding="utf-8", skipinitialspace=True, usecols=[tweetTextColumnName, labelColumnName])

    X = df[tweetTextColumnName].to_numpy()
    y = df[labelColumnName].to_numpy()
    fileName = corpusFile[+17:-4]

    # Initialize
    ROCscores = list()
    SVM_classifier = SVC(probability=True)
    MNB_classifier = MultinomialNB()
    RF_classifier = RandomForestClassifier(n_jobs=-1)

    print(Fore.GREEN + "\nTraining & predicting:", Style.RESET_ALL, fileName)

    # Calculate SVM ROC AUC scores
    classNames, ROCscores = calculateROC_AUC(SVM_classifier, X, y, numberOfFolds, printBestFeatures, plotPerFold)

    # Summarize & save scores
    ROCscores = np.array(ROCscores).reshape((numberOfFolds, -1))
    ROCscores = np.vstack((ROCscores, np.mean(ROCscores, axis=0)))
    df = pd.DataFrame(ROCscores, index=[np.append(range(numberOfFolds), ['Mean'])], columns=[np.append(classNames, ['Micro-average', 'Macro-average'])]).round(2)
    print("SVM macro-average ROC-AUC score:", Style.BRIGHT, df.iloc[numberOfFolds][len(classNames) + 1], Style.RESET_ALL)
    df.to_csv(targetResultFile + "_SVM.csv", index=True)

    # Calculate MNB ROC AUC scores
    classNames, ROCscores = calculateROC_AUC(MNB_classifier, X, y, numberOfFolds, printBestFeatures, plotPerFold)

    # Summarize & save scores
    ROCscores = np.array(ROCscores).reshape((numberOfFolds, -1))

```

```

ROCscores = np.vstack((ROCscores, np.mean(ROCscores, axis=0)))
df = pd.DataFrame(ROCscores, index=[np.append(range(numberOfFolds), ['Mean'])], columns=[np.append(classNames, ['Micro-
average', 'Macro-average'])]).round(2)
print("MNB macro-average ROC-AUC score:", Style.BRIGHT, df.iloc[numberOfFolds][len(classNames) + 1], Style.RESET_ALL)
df.to_csv(targetResultFile + "_MNB.csv", index=True)

# Calculate RF ROC AUC scores
classNames, ROCscores = calculateROC_AUC(RF_classifier, X, y, numberOfFolds, printBestFeatures, plotPerFold)

# Summarize & save scores
ROCscores = np.array(ROCscores).reshape(numberOfFolds, -1)
ROCscores = np.vstack((ROCscores, np.mean(ROCscores, axis=0)))
df = pd.DataFrame(ROCscores, index=[np.append(range(numberOfFolds), ['Mean'])], columns=[np.append(classNames, ['Micro-
average', 'Macro-average'])]).round(2)
print("RF macro-average ROC-AUC score:", Style.BRIGHT, df.iloc[numberOfFolds][len(classNames) + 1], Style.RESET_ALL)
df.to_csv(targetResultFile + "_RF.csv", index=True)

"""
# Train the SVM classifier and predict classes
SVM_classifier = SVC(kernel='poly', degree=1, decision_function_shape='ovo', cache_size=1000, coef0=0.1, shrinking=False
, random_state=0)
SVM_pred = SVM_classifier.fit(X_train, y_train).predict(X_test)

# Train the NB classifier and predict classes
MNB_classifier = MultinomialNB(alpha=0.4)
MNB_pred = MNB_classifier.fit(X_train, y_train).predict(X_test)

# Train the RF classifier and predict classes
RF_classifier = RandomForestClassifier(bootstrap=False, max_features='sqrt', min_samples_leaf=3, min_samples_split=10, n
_estimators=330, n_jobs=-1)
RF_pred = RF_classifier.fit(X_train, y_train).predict(X_test)
"""

#### Start program ####
# Optional mute warning messages

```

```
warnings.filterwarnings('ignore', category=UserWarning, append=True)

# Scan for files in directory and classify them
onlyFiles = [f for f in listdir("./classification/") if isfile(join("./classification/", f))]
for files in onlyFiles:
    classify("./classification/" + files, "./classification/ROC_AUC/r_" + files, 'tweet_text', 'label', 10, False, False)
    print(Fore.GREEN + "Classification complete.", Style.RESET_ALL)

print(Fore.GREEN + "\nOperations complete.", Style.RESET_ALL)
```

Παράρτημα Η – SVMparameters.py

```
import pandas as pd
import numpy as np
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.model_selection import train_test_split
from sklearn.svm import SVC
from sklearn.metrics import classification_report, confusion_matrix, accuracy_score
from pprint import pprint

df = pd.read_csv("./classification/pp_n_2014_Philippines_Typhoon_Hagupit_en_CF_labeled_data.csv", encoding="ISO-8859-1", skipinitialspace=True, usecols=['tweet_text', 'label'])

X = df['tweet_text']
y = df['label']

# Convert text documents X into TFIDF feature values
tfidfconverter = TfidfVectorizer(max_features=1000, min_df=3, max_df=0.8, ngram_range=(1, 2), use_idf=False, sublinear_tf=True)
X = tfidfconverter.fit_transform(X)

# Divide data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.33, shuffle=True, random_state=0)

# Create a default classifier
svm = SVC(random_state=0)
print('\nParameters currently in use:')
pprint(svm.get_params())

# Create parameter grid
grid = {'C' : [float(x) for x in np.linspace(start = 0.9, stop = 1.1, num = 3)],
        'kernel' : ['poly'],
        'degree' : [1, 2],
        'decision_function_shape' : ['ovo'],
        'cache_size' : [1000],
        'coef0' : [float(x) for x in np.linspace(start = 0.1, stop = 0.5, num = 5)],
```

```

        'shrinking' : [False]]
print("\nParameter grid:")
pprint(grid)

"""
# Random search
from sklearn.model_selection import RandomizedSearchCV
svm_search = RandomizedSearchCV(estimator = svm, param_distributions = grid, n_iter = 50, cv =5, verbose=1, random_state=42,
    n_jobs =-1)

# Random search to the data
svm_search.fit(X_train, y_train)
# {'kernel': 'linear', 'gamma': 'scale', 'degree': 0, 'C': 1} 0.700
"""

# Grid search
from sklearn.model_selection import GridSearchCV
svm_search = GridSearchCV(estimator = svm, param_grid = grid, cv = 5, n_jobs = -1, verbose = 1)

# Grid search to the data
svm_search.fit(X_train, y_train)

print("\nBest parameters:")
print(svm_search.best_params_)
# {'C': 1.1, 'cache_size': 1000, 'coef0': 0.3, 'decision_function_shape': 'ovo', 'degree': 2, 'kernel': 'poly'} 0.62
# {'C': 1.1, 'cache_size': 1000, 'coef0': 0.3, 'decision_function_shape': 'ovo', 'degree': 2, 'kernel': 'poly'} 0.62
# {'C': 1.0, 'cache_size': 1000, 'coef0': 0.4, 'decision_function_shape': 'ovo', 'degree': 2, 'gamma': 'scale', 'kernel': 'poly', 'probability': False, 'shrinking': False}
# {'C': 0.9, 'cache_size': 1000, 'coef0': 0.5, 'decision_function_shape': 'ovo', 'degree': 2, 'kernel': 'poly', 'shrinking': False} 0.637
# {'C': 0.9, 'cache_size': 1000, 'coef0': 0.5, 'decision_function_shape': 'ovo', 'degree': 2, 'kernel': 'poly', 'shrinking': False} 0.637
# {'C': 0.9, 'cache_size': 1000, 'coef0': 0.5, 'decision_function_shape': 'ovo', 'degree': 2, 'kernel': 'poly', 'shrinking': False} 0.637

```

```

# Train with best parameters
best = svm_search.best_estimator_
y_pred = best.predict(X_test)

# Print scores
print(confusion_matrix(y_test,y_pred))
print(classification_report(y_test,y_pred))
print(accuracy_score(y_test, y_pred))

# Train the SVM classifier and predict classes
SVMClassifier = SVC(kernel='poly', degree=1, decision_function_shape='ovo', cache_size=1000, coef0=0.1, shrinking=False, random_state=0)
y_pred = SVMClassifier.fit(X_train, y_train).predict(X_test)
# Print scores
print(confusion_matrix(y_test,y_pred))
print(classification_report(y_test,y_pred))
print(accuracy_score(y_test, y_pred))

```

Παράρτημα Θ – NBparameters.py

```
import pandas as pd
import numpy as np
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.model_selection import train_test_split
from sklearn.naive_bayes import MultinomialNB
from sklearn.metrics import classification_report, confusion_matrix, accuracy_score
from pprint import pprint

df = pd.read_csv("./classification/pp_n_2014_Philippines_Typhoon_Hagupit_en_CF_labeled_data.csv", encoding="ISO-8859-1", skipinitialspace=True, usecols=['tweet_text', 'label'])

X = df['tweet_text']
y = df['label']

# Convert text documents X into TFIDF feature values
tfidfconverter = TfidfVectorizer(max_features=1000, min_df=3, max_df=0.8, ngram_range=(1, 2), use_idf=False, sublinear_tf=True)
# tfidfconverter = TfidfVectorizer(max_features=3000, min_df=5, max_df=0.8, ngram_range=(1, 2), use_idf=False, sublinear_tf=True)
X = tfidfconverter.fit_transform(X)

# Divide data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.33, shuffle=True, random_state=0)

# Create a default classifier
mnb = MultinomialNB()
print('\nParameters currently in use:')
pprint(mnb.get_params())

# Create parameter grid
grid = {'alpha' : [float(x) for x in np.linspace(start = 0.1, stop = 0.8, num = 8)],
        'fit_prior' : [False, True]}
print("\nParameter grid:")
pprint(grid)
```

```

# MNB {'clf__alpha': 0.1,
# 'tfidf__binary': False,
# 'tfidf__max_df': 0.7777777777777778,
# 'tfidf__max_features': 1000,
# 'tfidf__min_df': 3,
# 'tfidf__ngram_range': (1, 2),
# 'tfidf__smooth_idf': False,
# 'tfidf__sublinear_tf': False,
# 'tfidf__use_idf': False}

# Grid search
from sklearn.model_selection import GridSearchCV
mnb_search = GridSearchCV(estimator = mnb, param_grid = grid, cv = 5, n_jobs = -1, verbose = 1)

# Grid search to the data
mnb_search.fit(X_train, y_train)

print("\nBest parameters:")
print(mnb_search.best_params_)

# Train with best parameters
best = mnb_search.best_estimator_
y_pred = best.predict(X_test)
# {'alpha': 0.4, 'fit_prior': True} 0.703

# Print scores
print(confusion_matrix(y_test, y_pred))
print(classification_report(y_test, y_pred))
print(accuracy_score(y_test, y_pred))

# Train the NB classifier and predict classes
MNBClassifier = MultinomialNB(alpha=0.4)
y_pred = MNBClassifier.fit(X_train, y_train).predict(X_test)

# Print scores

```

```
print(confusion_matrix(y_test, y_pred))  
print(classification_report(y_test, y_pred))  
print("Accuracy:", accuracy_score(y_test, y_pred))
```

Παράρτημα Ι – RFparameters.py

```
import pandas as pd
import numpy as np
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import classification_report, confusion_matrix, accuracy_score
from pprint import pprint

df = pd.read_csv("./classification/pp_n_2014_Philippines_Typhoon_Hagupit_en_CF_labeled_data.csv", encoding="ISO-8859-1", skipinitialspace=True, usecols=['tweet_text', 'label'])

X = df['tweet_text']
y = df['label']

# Convert text documents X into TFIDF feature values
tfidfconverter = TfidfVectorizer(max_features=1000, min_df=3, max_df=0.8, ngram_range=(1, 2), use_idf=False, sublinear_tf=True)
X = tfidfconverter.fit_transform(X)

# Divide data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.33, shuffle=True, random_state=0)

# Create a default classifier
rf = RandomForestClassifier(n_jobs=-1, random_state = 42)
print('\nParameters currently in use:')
pprint(rf.get_params())

# Create parameter grid
grid = {'n_estimators' : [int(x) for x in np.linspace(start = 250, stop = 450, num = 5)],
        'max_features' : ['sqrt'],
        'min_samples_split' : [12, 13, 14, 15],
        'min_samples_leaf' : [2, 3],
        'bootstrap' : [False]}
print("\nRandom grid:")
```

```

pprint(grid)

"""
# Random search
from sklearn.model_selection import RandomizedSearchCV
rf_search = RandomizedSearchCV(estimator = rf, param_distributions = grid, n_iter = 50, cv =5, verbose=1, random_state=0, n_
jobs =-1)

# Random search to the data
rf_search.fit(X_train, y_train)
# {'n_estimators': 270, 'min_samples_split': 3, 'min_samples_leaf': 3, 'max_features': 'auto', 'max_depth': None, 'bootstrap
': False} 0.718
"""

# Grid search
from sklearn.model_selection import GridSearchCV
rf_search = GridSearchCV(estimator = rf, param_grid = grid, cv = 5, n_jobs = -1, verbose = 1)

# Grid search to the data
rf_search.fit(X_train, y_train)

print("\nBest parameters:")
print(rf_search.best_params_)
# {'bootstrap': False, 'max_features': 'sqrt', 'min_samples_leaf': 2, 'min_samples_split': 12, 'n_estimators': 366} 0.713
# {'bootstrap': False, 'max_features': 'sqrt', 'min_samples_leaf': 2, 'min_samples_split': 11, 'n_estimators': 400} 0.708
# {'bootstrap': False, 'max_features': 'sqrt', 'min_samples_leaf': 2, 'min_samples_split': 13, 'n_estimators': 430} 0.713
# {'bootstrap': False, 'max_features': 'sqrt', 'min_samples_leaf': 2, 'min_samples_split': 12, 'n_estimators': 383} 0.711

# Train with best parameters
best = rf_search.best_estimator_
y_pred = best.predict(X_test)

# Print scores
print(confusion_matrix(y_test,y_pred))
print(classification_report(y_test,y_pred))

```

```
print(accuracy_score(y_test, y_pred))

# Train the RF classifier and predict classes
RFclassifier = RandomForestClassifier(bootstrap=False, max_features='sqrt', min_samples_leaf=3, min_samples_split=10, n_estimators=330, n_jobs=-1)
y_pred = RFclassifier.fit(X_train, y_train).predict(X_test)
# Print scores
print(confusion_matrix(y_test, y_pred))
print(classification_report(y_test, y_pred))
print("Accuracy:", accuracy_score(y_test, y_pred))
```

Παράρτημα Κ – Φονικός σεισμός της Σμύρνης και Σάμου 2020

Ως μια λυπητερή ένδειξη του τέλειου χρονισμού, δύο ώρες πριν την επίσημη παρουσίαση αυτού του έργου, γίνεται ένας δυνατός σεισμός 7.0 της κλίμακας ρίχτερ στην πόλη του Πανεπιστημίου Αιγαίου την Σάμο, αφήνοντας πίσω 2 νεκρούς εφήβους από κατολίσθηση κτηρίου στην Σάμο αλλά και 114 θανάτους στην γειτονική πόλη της Τουρκίας την Σμύρνη. Αυτό το τραγικό γεγονός παρότι κλόνισε το κλίμα της παρουσίας και την τοπική κοινότητα, ανέδειξε παράλληλα την σημασία όλου του έργου που πραγματεύεται και κίνησε την περιέργεια του πόσο καλά αποδίδουν πράγματι αυτά τα μοντέλα απέναντι σε αναρτήσεις χρηστών εκείνης της τρομακτικής ημέρας. Παρακάτω περιγράφεται η διαδικασία και σχολιάζονται τα αποτελέσματα που προέκυψαν. Όλα τα αρχεία μπορούν να βρεθούν στον φάκελο του έργου που προσφέρεται στην αρχή του κεφαλαίου της υλοποίησης.

Συλλογή δεδομένων

Την ημέρα εκείνη, Παρασκευή, 30 Οκτωβρίου 2020 και ώρα 13:51, ο δυνατός σεισμός έγινε αισθητός σε έκταση 200 χιλιομέτρων γύρω από την εστία του λίγο έξω από την Σάμο. Αυτό σημαίνει πως καταστροφές και ανθρώπινες ζωές χάθηκαν όχι μόνο σε μια χώρα όπως την Σάμο της Ελλάδας αλλά και στην γειτονική Σμύρνη της Τουρκίας.

Οι δύο αυτές χώρες χρησιμοποιούν διαφορετική γλώσσα πράγμα που αμέσως καθιστά φανερό την ανάγκη διαχωρισμού του dataset των tweet σε δύο κομμάτια ανάλογα την γλώσσα. Το ένα κομμάτι θα είναι σίγουρα στα ελληνικά και θα είναι πολύ βολικό να εξεταστεί πάνω στο μοντέλο που εκπαιδεύτηκε και αυτό σε ελληνικά tweet, όπως για παράδειγμα την φονική πυρκαγιά της Αττικής του 2018 που εξετάστηκε στην παρούσα διπλωματική. Το άλλο κομμάτι θα πρέπει να είναι σε γλώσσα της Τουρκίας, όμως αυτό δεν είναι εφικτό με την παρούσα υλοποίηση καθώς έχουμε dataset μόνο με αγγλικά tweet. Για τον λόγο αυτό, για το κομμάτι της Σμύρνης, θα ληφθούν υπόψιν μόνο tweet τα οποία αναφέρονται για τον σεισμό εκείνον και είναι γραμμένα στα αγγλικά, τα οποία μετά μπορούν εύκολα να εξεταστούν σε ένα από τα μοντέλα που εκπαιδεύτηκαν πάνω σε αγγλικά tweet, όπως για παράδειγμα ο φονικός σεισμός στο Νεπάλ το 2015 που εξετάστηκε στην παρούσα διπλωματική.

Για να ελέγξουμε με σωστό τρόπο την ακρίβεια των μοντέλων, χρειαζόμαστε ένα αντιπροσωπευτικό δείγμα του παλμού που υπήρχε στο twitter εκείνες τις πρώιμες ώρες. Για τον λόγο αυτό συλλέχθηκαν 100 tweet στα ελληνικά και 100 tweet στα αγγλικά για τις επιδράσεις του σεισμού στην Σάμο και την Σμύρνη αντίστοιχα. Η πολιτική της συλλογής ήταν να λαμβάνεται δείγμα 10 tweet ανά ώρα ξεκινώντας από την ημέρα του συμβάντος στις 14:00 και ολοκληρώνοντας ενδοημερήσια στις 23:59(δηλαδή 10 διαδικασίες δειγματοληψίας), με τους τρόπους που έχουν ήδη περιγραφεί στον κυρίως κορμό της διπλωματικής στην παράγραφο της υλοποίησης. Ο κώδικας λοιπόν είναι παρόμοιος με αυτόν που φαίνεται στο παράρτημα Ε' με την διαφορά του ό,τι οι όροι αναζήτησης πλέον(query) είναι 'σεισμός OR σεισμος OR Σάμος OR Σαμος' και 'earthquake OR Greece OR Turkey OR Samos' για την Σάμο και την Σμύρνη αντίστοιχα. Τα νέα δημιουργημένα dataset είναι αποθηκευμένα στον φάκελο *fetchedTweets* με όνομα *samosTweets.csv* και *izmirTweets.csv* αντίστοιχα.

Προεπεξεργασία δεδομένων

Το κομμάτι της προ επεξεργασίας των δεδομένων είναι όμοιο με αυτό που αναφέρεται στον κυρίως κορμό της διπλωματικής στην παράγραφο της υλοποίησης, δηλαδή αρχικά η κατηγοριοποίηση αυτών των 200 tweet με βάσει τον πίνακα 4 και στην συνέχεια όλη η διαδικασία της προ επεξεργασίας εκτός από το κομμάτι της επαύξης των δεδομένων(data augmentation) και αυτό διότι δεν αποσκοπείται να δημιουργηθούν νέα μοντέλα από τα δύο αυτά μικρά dataset αλλά αντιθέτως να εξεταστούν τα υπάρχοντα μοντέλα με βάσει αυτά τα dataset. Έτσι λοιπόν ο κώδικας είναι παρόμοιος με αυτόν που παρουσιάζεται στο παράρτημα ΣΤ' εκτός από το κομμάτι του data augmentation δηλαδή εκτός του βήματος 9(γραμμές 44-95) ο οποίος πρέπει να παραληφθεί. Τα επεξεργασμένα dataset είναι αποθηκευμένα στον φάκελο *processed* με όνομα *pp_samosTweets.csv* και *pp_izmirTweets.csv* αντίστοιχα.

Εκπαίδευση & σύγκριση μοντέλων ML

Αυτό το κομμάτι διαφέρει κάπως από την εφαρμογή που υλοποιήθηκε στον κυρίως κορμό της διπλωματικής στην παράγραφο της υλοποίησης στο κομμάτι της εκπαίδευσης των μοντέλων μηχανικής μάθησης και του υπολογισμού του σκορ αυτών.

Εδώ ως train set λαμβάνεται ολόκληρο το dataset αναφοράς, δηλαδή για παράδειγμα λαμβάνεται ολόκληρο το dataset των ελληνικών tweet από τις πυρκαγιές της Αττικής, και ως test set είναι ολόκληρο το dataset που συλλέξαμε, δηλαδή τα 100 tweet για τον σεισμό της Σάμου. Αντίστοιχα, ως training set θα ληφθεί ολόκληρο το dataset από τον σεισμό στο Νεπάλ και ως testing set θα χρησιμοποιηθούν τα 100 tweet για τον σεισμό στην Σμύρνη.

Επειδή το training set και το testing set λαμβάνονται ολόκληρα και είναι ξεχωριστά δεν υπάρχει ο κίνδυνος διαρροής δεδομένων⁴(data leak), δηλαδή «διαρροής» γνώσης από το training set στο testing set, για τον λόγο αυτό δεν είναι απαραίτητο να πραγματοποιηθεί το CV και για αυτό παραλείπεται.

Επιπλέον ως μέθοδο σκορ(scoring method) που χρειαζόμαστε είναι η επιτυχία της σωστής πρόβλεψης της κατηγορίας από το μοντέλο σε σύγκριση με την ανθρώπινη κατηγορία που δόθηκε κατά την κατηγοριοποίηση. Έτσι λοιπόν η μέθοδος σκορ δεν θα είναι η ROC-AUC αλλά η ακρίβεια των προβλέψεων(accuracy score). Επιπλέον δεν παράγονται συγκεντρωτικά αρχεία με ROC-AUC σκορ στο τέλος αυτής της διαδικασίας αλλά νέα αρχεία που περιέχουν

Ο τροποποιημένος κώδικας *classifiers.py* της εκπαίδευσης των μοντέλων φαίνεται στο τέλος αυτού του παραρτήματος.

⁴ Ως διαρροή δεδομένων ορίζουμε την αθέμιτη μεταφορά γνώσης από το training set στο testing set που τα μοντέλα μηχανικής μάθησης δεν θα έπρεπε να έχουν στην κατοχή τους για να βελτιώσουν τους αλγόριθμους τους. Η διαρροή γνώσης οδηγεί σε καλύτερα αποτελέσματα όμως τα αποτελέσματα αυτά είναι ψευδή και αισιόδοξα και οδηγούν σε overfitting.

Συμπεράσματα

Στους παρακάτω πίνακες φαίνονται τα 10 πιο αντιπροσωπευτικά tweet που επέλεξα μαζί με την κατηγορία που ανήκουν με βάσει την δική μου κατηγοριοποίηση και με βάσει την κατηγορία που πρόβλεψαν τα μοντέλα ότι ανήκουν σύμφωνα με το καλύτερο μοντέλο, δηλαδή τα Τυχαία Δάση.

Αποτελέσματα RF πρόβλεψης κατηγοριών στο dataset της Σμύρνης			
#	tweet_text	label	label_predicted
1	RT @JoeBiden: Jill and I send our prayers to the people of Greece and Turkey following today's earthquake. We're grateful for the brave res	Sympathy and emotional support	Sympathy and emotional support
2	RT @bahathmag: The Nuruosmaniye Mosque Istanbul, Turkey Constructed in the 18th century the mosque was designed in the Ottoman-Baroque	Not related or irrelevant	Not related or irrelevant
3	Bayraktar drones are used to bomb homes, schools, and churches of Armenians and #Garmin continuously sells its products https://t.co/h8nJwa6hJf	Not related or irrelevant	Other useful information
4	RT @metesohtaoglu: #Turkey- The colored points are the shaking felt by the earthquake eyewitnesses The intensity of the shaking decreases	Other useful information	Other useful information
5	RT @ArdyKassakhian: An earthquake in Smyrna (Izmir, Turkey). Let's see how #Erdogan handles this. Wonder if he will hire Jihadist mercenaries	Not related or irrelevant	Other useful information
6	RT @ragipsoylu: Horrible. A building collapses after a 7.0 earthquake near Izmir. Hopefully people were able to leave No information on	Infrastructure and utilities damage	Infrastructure and utilities damage
7	RT @MiddleEastMnt: 17 killed, 709 injured in quake in Izmir, Aegean Turkey https://t.co/YdPN7g6Dbv	Injured or dead people	Injured or dead people
8	@davidcoverdale Someone tried to fry a frozen turkey???	Not related or irrelevant	Not related or irrelevant
9	A strong earthquake in the Turkish city of Izmir, damage to buildings, loss of life and property Intensity recorded https://t.co/zroOPRTPsV	Injured or dead people	Infrastructure and utilities damage
10	"Delirious woman is being pulled out of the rubble in Izmir's Buse district #earthquake #Turkey #Izmir https://t.co/7mLqPJ5ARy "	Missing trapped or found people	Missing trapped or found people

Πίνακας 24: Στιγμιότυπο κατηγοριοποιημένων tweet με την σωστή και την προβλεπόμενη κατηγορία του dataset της Σμύρνης.

Όπως φαίνεται στην περίπτωση της Σμύρνης οι κατηγορίες είναι οι περισσότερες σωστά αποδοσμένες. Για παράδειγμα το tweet 1 ανήκει καταφανώς στην κατηγορία *sympathy and emotional support*. Το tweet 3 δεν θα έπρεπε να έχει προβλεφθεί ως *other useful information* καθώς δεν συσχετίζεται με τον σεισμό. Το tweet 5 χρησιμοποιεί το συμβάν για να λασπολογήσει την πολιτική του Προέδρου Ρετζέπ Ταγίπ Ερντογάν και κακώς αναγνωρίζεται ως *other useful information*. Το tweet 9 αναφέρεται σε νεκρούς και υλικές ζημιές όμως η σωστή κατηγορία με βάση την πολιτική της κατηγοριοποίησης που έχει προαναφερθεί είναι η *injured or dead people* και το μοντέλο το αναγνωρίζει ως *infrastructure and utilities damage* που δεν είναι λάθος καθώς εμπεριέχεται και αυτή η πληροφορία. Tweet όπου ανήκουν σε περισσότερες από μια κατηγορίες συναντούμε αρκετά όμως όταν δεν αναγνωρίζεται η «πιο σημαντική» κατηγορία τότε τα εκλαμβάνουμε ως λάθος ενδείξεις.

Αποτελέσματα RF πρόβλεψης κατηγοριών στο dataset της Σάμου			
#	tweet_text	label	label_predicted
1	RT @PrimeministerGR: Η Σάμος άντεξε. Κυβερνητικό κλιμάκιο μεταβαίνει στο νησί για την αντιμετώπιση των προβλημάτων. Ενημερώνομαι διαρκώς γι...	Other useful information	Not related or irrelevant
2	"RT @amna_news: Μια μεγάλη ρωγμή κατά μήκος του κρηπιδώματος της Προκυμαίας στη Χίο εμφανίστηκε μετά το σημερινό μεγάλο σεισμό Πολύ σημαντι..."	Caution and advice	Other useful information
3	"RT @MariaInIstanbul: İzmir and Samos. No words, only sympathy to everyone there and to their loved ones. İzmir'e ve Samos'a: Hiç sözler ak..."	Sympathy and emotional support	Not related or irrelevant
4	Σάμος: C-130 θα μεταφέρει στην Αθήνα 14χρονο παιδί με σοβαρά κατάγματα από τον σεισμό https://t.co/YEx5EuUhly μέσω @GrTimed	Injured or dead people	Not related or irrelevant
5	"RT @npanagioto: Έχει διαταχθεί τοπική επιφυλακή. Έχουν διατεθεί μηχανήματα στην Περιφέρεια. Οι Ένοπλες Δυνάμεις δίπλα όπως πάντα στους πολ..."	Donation needs or offers or volunteering services	Not related or irrelevant
6	"RT @Marka149133376: Η φωτογραφία της ελπίδας !! Απίστευτη,συγκλονιστική !!! 😞😞😞 #TurkeyEarthquake #izmir #σεισμος #Σαμος https://t.co/Nzl... "	Sympathy and emotional support	Sympathy and emotional support

7	Βαριά τραυματισμένη από τον σεισμό των 6,7 ρίχτερ η Σάμος - Δύο νεκροί, τραυματίες και ετοιμόρροπα κτίρια https://t.co/kyPX5UQan2	Injured or dead people	Injured or dead people
8	RT @SheikYerbutti2: Έπρεπε να υπάρξουν νεκροί για να ψελλίσει το «Σάμος» που μέχρι τότε ήταν «αλλού», άνευ ονόματος. Αρχηγός ελληνικού κόμμ...	Not related or irrelevant	Injured or dead people
9	"🤯🤯🤯🤯 #σεισμος #Σαμος #συμυρη #earthquake https://t.co/kwWGw4xUcm "	Other useful information	Not related or irrelevant
10	"RT @Marka149133376: Το πάρκινγκ αυτοκινήτων στο Βαθύ Σάμου. Στο νησί υπάρχουν τεράστιες καταστροφές #Σαμος #σεισμος https://t.co/EDiifimK... "	Infrastructure and utilities damage	Sympathy and emotional support

Πίνακας 25: Στιγμιότυπο κατηγοριοποιημένων tweet με την σωστή και την προβλεπόμενη κατηγορία του dataset της Σάμου.

Όπως φαίνεται στην περίπτωση της Σάμου τα μοντέλα δεν τα πήγαν καλά. Για παράδειγμα το tweet 1 ενώ αναφέρεται σε κυβερνητική δράση για τον σεισμό λανθασμένα αναγνωρίζεται ως άσχετο. Το tweet 2 ενώ εφιστά την προσοχή για ρήγματα που έχουν ανοίξει από τον σεισμό και θα έπρεπε να θεωρηθεί ως *caution and advise* αναγνωρίζεται ως *other useful information* που δεν είναι εντελώς λάθος αλλά ούτε το σωστό. Το 3^ο tweet ενώ είναι στα αγγλικά και αναφέρεται σε συμπαράσταση λανθασμένα αναγνωρίζεται ως *not related or irrelevant*, πιθανών επειδή δεν υπάρχουν αρκετά αγγλικά tweet στο training set. Το 4^ο ενώ αναφέρεται σε μεταφορά τραυματία λανθασμένα αναγνωρίζεται ως άσχετο. Το 5^ο ενώ μιλά για προσφορά βοήθειας δεν αναγνωρίζεται ως τέτοιο αλλά ως άσχετο. Το 8^ο tweet καταφέρεται εναντίων του βουλευτή Γιάννη Βαρουφάκη που συλλυπήθηκε μόνο την Τουρκία για τον σεισμό και είναι άσχετο με την κατάσταση έκτακτης ανάγκης αλλά αναγνωρίζεται λανθασμένα ως *injured or dead people* που δεν είναι τελείως αναπάντεχο καθώς υπάρχει η λέξη «νεκρός» αλλά σίγουρα δεν αποτελεί σχόλιο για την κρίση του σεισμού. Το tweet 10 ενώ αναφέρεται φανερά σε υλικές ζημιές αναγνωρίζεται ως *sympathy and emotional support* που είναι εντελώς λάθος καθώς δεν υπάρχει καμία τέτοια προαίρεση στο κείμενο.

Στα προβλήματα κατηγοριοποίησης, ανάλογα την φύση του προβλήματος, τίθεται η υποκειμενική έννοια της αξίας της κατηγορίας. Στη δική μας περίπτωση μπορεί να θεωρούμε πως η κατηγορία *injured or dead people* έχει μεγαλύτερη αξία να αναγνωρίζεται περισσότερες φορές σωστά απ' ό,τι η κατηγορία *sympathy and emotional support* ή *not related or irrelevant*.

Στους παρακάτω πίνακες φαίνονται τα συγκεντρωτικά αποτελέσματα της κατηγοριοποίησης. Τα μετρικά σκορ *precision*, *recall*, *accuracy* και *F1* έχουν ήδη εξηγηθεί στις εικόνες 7 & 8. *Support*(υποστήριξη) είναι ο αριθμός των document που υπάρχουν στην κάθε κατηγορία. *Macro average*, όπως έχει ήδη εξηγηθεί, είναι ο γνωστή μαθηματική ελπίδα, χωρίς βάρη. Στον αντίποδα είναι το *Micro average*(*weighted average*) υπολογίζει την μαθηματική ελπίδα πολλαπλασιάζοντας το κάθε σκορ με το ποσοστό συμμετοχής(*Support percentage*) που έχει στο dataset. Σε γενικές γραμμές το *micro average F1-score* είναι το πιο σημαντικό σκορ που πρέπει να χρησιμοποιείται για συγκρίνει μοντέλα και όχι η γενική ακρίβεια(*accuracy*).

99 ⁵ tweet για την Σμύρνη με βάσει τους σεισμούς του Νεπάλ το 2015	Precision			Recall			F1-score			Support
	SVM	MNB	RF	SVM	MNB	RF	SVM	MNB	RF	SVM / MNB / RF
Donation needs or offers or volunteering services	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1
Infrastructure and utilities damage	0.83	0.83	0.83	0.83	0.83	0.83	0.83	0.83	0.83	6
Injured or dead people	1.00	0.80	1.00	0.60	0.80	0.80	0.75	0.80	0.89	5
Missing trapped or found people	1.00	0.33	1.00	0.50	0.50	0.50	0.67	0.40	0.67	2
Not related or irrelevant	0.82	1.00	0.79	0.69	0.08	0.69	0.75	0.14	0.74	39
Other useful information	0.33	0.33	0.36	0.71	0.57	0.64	0.45	0.42	0.46	14
Sympathy and emotional support	0.92	0.84	0.85	0.72	0.84	0.72	0.81	0.84	0.78	32
Accuracy							0.70	0.48	0.70	99
Macro average	0.70	0.59	0.69	0.58	0.52	0.60	0.61	0.49	0.62	99
Micro(weighted) average	0.79	0.81	0.76	0.70	0.48	0.70	0.72	0.49	0.72	99

Πίνακας 26: Αναφορά κατηγοριοποίησης των μοντέλων για το dataset της Σμύρνης.

Όπως φαίνεται στον παραπάνω πίνακα τα μοντέλα RF & SVM απέδωσαν σε γενικές γραμμές ικανοποιητικά με γενική ακρίβεια F1 στο 70% και micro average F1 στο 72%, όχι όμως και το MNB που τα σκορ του ήταν απογοητευτικά επιδεικνύοντας γενική ακρίβεια F1 48% και micro average F1 49%, δηλαδή πιθανότητα επιτυχημένης εύρεσης 50-50, καθιστώντας το μοντέλο MNB ακατάλληλο για χρήση. Το καλύτερο από τα τρία θα λέγαμε πως είναι τα Τυχαία Δάση και αυτό γιατί όπως φαίνεται έχει καλύτερο F1 σκορ σε κατηγορίες που μας ενδιαφέρουν υποκειμενικά περισσότερο, πετυχαίνοντας 89% σε *injured or dead people* και 83% σε *infrastructure and utilities damage* και ένα μέτριο 67% σε *missing trapped or found people* που είναι χρήσιμες πληροφορίες την στιγμή της έκτακτης ανάγκης. Βλέπουμε πως κανένα από τα μοντέλα δεν κατάφερε να αναγνωρίσει το ένα και μοναδικό tweet για *donation needs or offers or volunteering services* και για αυτό η συγκεκριμένη κατηγορία έχει σκορ 0% όμως αυτό το σκορ δεν είναι αντιπροσωπευτικό καθώς υπάρχει υποστήριξη(support) μόνο 1 από ολόκληρο το δείγμα. Για πιο αντιπροσωπευτική αξιολόγηση της ικανότητας του μοντέλου να αναγνωρίζει επιτυχώς την συγκεκριμένη κατηγορία θα έπρεπε να υπάρχουν περισσότερα δείγματα. Ομοίως, χρειάζονται περισσότερα δείγματα και για το *missing trapped or found people* τα οποία ίσως να βελτιώσουν ακόμα περισσότερο το σκορ.

⁵ 99 και όχι 100 διότι ένα tweet διαγράφηκε ως κενό(σ.σ. βλέπε διαδικασία προ επεξεργασίας στο κεφάλαιο της υλοποίησης)

100 tweet για την Σάμο με βάσει τις πυρκαγιές της Αττικής το 2018	Precision			Recall			F1-score			Support
	SVM	MNB	RF	SVM	MNB	RF	SVM	MNB	RF	SVM / MNB / RF
Caution and advice	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1
Displaced people and evacuations	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1
Donation needs or offers or volunteering services	0.00	0.20	0.00	0.00	0.50	0.00	0.00	0.29	0.00	2
Infrastructure and utilities damage	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	6
Injured or dead people	0.60	0.43	0.60	0.23	0.23	0.23	0.33	0.30	0.33	13
Missing trapped or found people	1.00	0.20	0.50	1.00	1.00	1.00	1.00	0.33	0.67	1
Not related or irrelevant	0.25	0.33	0.26	0.81	0.71	0.86	0.38	0.45	0.40	21
Other useful information	0.00	0.14	0.33	0.00	0.07	0.07	0.00	0.09	0.11	29
Sympathy and emotional support	0.76	0.70	0.93	0.50	0.62	0.50	0.60	0.65	0.65	26
Accuracy							0.34	0.38	0.37	100
Macro average	0.29	0.22	0.29	0.28	0.35	0.30	0.26	0.23	0.24	100
Micro(weighted) average	0.34	0.35	0.48	0.34	0.38	0.37	0.29	0.34	0.34	100

Πίνακας 27: Αναφορά κατηγοριοποίησης των μοντέλων για το dataset της Σάμου.

Όπως φαίνεται στον παραπάνω πίνακα όλα τα μοντέλα απέδωσαν απογοητευτικά με το καλύτερο απ' όλα να είναι τα Τυχαία Δάση. Σε σημαντικές κατηγορίες όπως το injured or dead people με αρκετά δείγματα(support = 13) σκόραραν ένα ανίσχυρο 30-33% F1-score έχοντας ένα 23% recall, δηλαδή έχοντας ανακαλύψει σωστά μόνο το 23% των tweet που αναφέρονται σε τραυματίες & νεκρούς, που όταν αναφερόμαστε σε ανθρώπινες ζωές αυτό το σκορ είναι καταστροφικό. Στους αγνοούμενους ανθρώπους υπάρχει υποστήριξη(support) μόνο 1 από ολόκληρο το δείγμα και το 67% F1-score που πετυχαίνει δεν είναι αντιπροσωπευτικό. Η μόνο κατηγορία που εντοπίζεται κάπως ικανοποιητικά είναι η sympathy and emotional support με F1 65% που σε κατάσταση έκτακτης ανάγκης είναι μάλλον από τα τελευταία που θα ενδιέφεραν τον κόσμο & τα διασωστικά συνεργεία να αναγνωρίσουν έγκαιρα, το οποίο όμως έχει recall μόνο 50%.

Βλέπουμε ότι στην Σμύρνη τα μοντέλα έχουν πέσει αποδίδουν κάπως ικανοποιητικά με F1-score στο 72% και αυτό το σκορ μειώθηκε εκεί επειδή στο training set υπήρχε ακραία ασυμμετρία κλάσεων(extreme class imbalance) η οποία όπως προαναφέρθηκε δεν επηρεάζει το ROC-AUC σκορ, επηρεάζει όμως το F1-score και εδώ φαίνεται πραγματικά η δυσκολία του μοντέλου να αποφανθεί με σιγουριά ορισμένες κατηγορίες.

Γιατί όμως στην Σάμο τα μοντέλα τα κατάφεραν τόσο αποκαρδιωτικά; Είναι σίγουρα επειδή και στο ελληνικό training set υπήρχε extreme class imbalance το οποίο οδηγεί σε προκατάληψη(bias). Όπως φαίνεται και στον πίνακα 16 η μεγαλύτερη κλάση είναι η *not related or irrelevant* με τα περισσότερα documents που κάνει τα μοντέλα να κατατάσσουν εκεί όσα tweet δεν νιώθουν σίγουρα για αυτά (όπως φαίνεται και στον πίνακα-δείγμα 25 με τα μισά tweet αναγνωρισμένα ως άσχετα) και έτσι παρατηρούμε στον πίνακα 27 ένα precision της κλάσης αυτής στο 26%.

Ο υπ' αριθμόν ένα όμως παράγοντας που κατακρήμνισε το F1-score και την γενική ακρίβεια της Σάμου και ομολογουμένως τα μοντέλα απέτυχαν είναι αυτός που αναφέρεται και στον πίνακα 1, δηλαδή η μεγάλη ανομοιοότητα των training set και testing set συνόλων. Τα μοντέλα της Σμύρνης εκπαιδεύτηκαν σε tweet σχετικά με σεισμό και κλήθηκαν να δοκιμαστούν σε δεδομένα σχετικά με σεισμό ενώ τα μοντέλα της Σάμου εκπαιδεύτηκαν σε tweet σχετικά με δασική πυρκαγιά και κλήθηκαν να δοκιμαστούν σε δεδομένα σχετικά με αστικό σεισμό. Στις πυρκαγιές της Αττικής υπήρχε μεγαλύτερο εγχώριο & διεθνές ενδιαφέρον καθώς συνέβη έξω από μεγάλη αστική πόλη(που οδηγεί σε μεγαλύτερη ποικιλία-dimensionality δεδομένων), ενώ στην Σάμο όχι. Στο Μάτι υπήρχαν πολλά ζώα που χάθηκαν ή κάηκαν ζωντανά καθώς ήταν δασική πυρκαγιά στην εξοχή, ενώ εδώ όχι γιατί οι καταστροφές ήταν εντός των πόλεων. Στο Μάτι οι άνθρωποι καίγονταν ζωντανοί και απανθρακώνονταν, εδώ όχι – εδώ καταπλακώνονταν και περιγράφονταν ως «απλώς» σκοτωμένοι. Στο Μάτι ο μεγαλύτερος «καημός» ήταν ο χαμός εκατοντάδων ζώων, ενώ εδώ η μεγαλύτερη κίνηση σε tweet ήταν η πρόληψη και η προστασία των ανθρώπων, το τσουνάμι, οι υλικές ζημιές στα κτίσματα των ανθρώπων, η παρηγοριά σε κάτι που έχουν βιώσει και που μπορούν να ταυτιστούν οι περισσότεροι άνθρωποι, οι ερωτήσεις για το τι γίνεται μετά κ.α.. Στο Μάτι τα παιδιά που σκοτώθηκαν ήταν μικρά ανήλικα δίδυμα, ενώ εδώ όχι. Όπως φαίνεται στον πίνακα 18, το No.1 bigram feature ήταν το «δίδυμο κοριτσάκι» και από τα πιο σημαντικά unigram features ήταν ο πατέρας που ψάχνει τα δίδυμα κοριτσάκια ενώ εδώ δεν υπήρχαν αυτά. Μπορούν λοιπόν να καταλάβουμε την σύγχυση των μοντέλων όταν κλήθηκαν να κατηγοριοποιήσουν αυτά τα 100 tweet που τους δόθηκαν και έτσι εξηγείται η αποτυχία των μοντέλων με micro average F1-score των RF στο 34%.

Εδώ ολοκληρώνονται οι παρατηρήσεις για την μελέτη της περίπτωσης εφαρμογής των μοντέλων για τον φονικό σεισμό της Σμύρνης και Σάμου το 2020.

Τροποποιημένος κώδικας – classifiers.py

```
import pandas as pd
import numpy as np
from colorama import Fore, Back, Style
from pprint import pprint
from sklearn.svm import SVC
from sklearn.naive_bayes import MultinomialNB
from sklearn.ensemble import RandomForestClassifier
from sklearn.feature_extraction.text import CountVectorizer
```

```

from sklearn.feature_selection import mutual_info_classif
from sklearn.feature_selection import SelectKBest
from sklearn.metrics import accuracy_score, classification_report

def classify(corpusFile, testSet, targetResultFile, tweetTextColumnName, labelColumnName):
    df = pd.read_csv(corpusFile, encoding="utf-8", skipinitialspace=True)
    df_test = pd.read_csv(testSet, encoding="utf-8", skipinitialspace=True)

    # Initialize
    X = df[tweetTextColumnName].to_numpy()
    y = df[labelColumnName].to_numpy()
    X_test = df_test[tweetTextColumnName].to_numpy()
    y_test = df_test[labelColumnName].to_numpy()
    fileName = corpusFile[+17:-4]
    SVM_classifier = SVC(probability=True)
    MNB_classifier = MultinomialNB()
    RF_classifier = RandomForestClassifier(n_jobs=-1)

    print(Fore.GREEN + "\nTraining & predicting:", Style.RESET_ALL, fileName)

    # Vectorize sets & calculate bbest features
    vectorizer = CountVectorizer(ngram_range=(1, 2))
    bestFeatures = SelectKBest(mutual_info_classif, k=1000)
    X = vectorizer.fit_transform(X)
    X = bestFeatures.fit_transform(X, y)
    X_test = vectorizer.transform(X_test)
    X_test = bestFeatures.transform(X_test)

    # Train the SVM classifier and predict classes
    SVM_pred = SVM_classifier.fit(X, y).predict(X_test)
    print("SVM:", accuracy_score(y_test, SVM_pred))
    print(classification_report(y_test, SVM_pred))
    df_testSVM = df_test
    df_testSVM['label_predicted'] = SVM_pred

```

```

dftestSVM.to_csv(targetResultFile + "_SVM.csv", index=False)

# Train the NB classifier and predict classes
MNB_pred = MNB_classifier.fit(X, y).predict(X_test)
print("MNB:", accuracy_score(y_test, MNB_pred))
print(classification_report(y_test, MNB_pred))
dftestMNB = dftest
dftestMNB['label_predicted'] = MNB_pred
dftestMNB.to_csv(targetResultFile + "_MNB.csv", index=False)

# Train the RF classifier and predict classes
RF_pred = RF_classifier.fit(X, y).predict(X_test)
print("RF:", accuracy_score(y_test, RF_pred))
print(classification_report(y_test, RF_pred))
dftestRF = dftest
dftestRF['label_predicted'] = RF_pred
dftestRF.to_csv(targetResultFile + "_RF.csv", index=False)

#### Start program ####

# Scan for files in directory and classify them
classify("./classification/pp_n_2015_Nepal_Earthquake_en_CF_labeled_data.csv", "./classification/pp_izmirTweets.csv", "./classification/results/r_pp_izmirTweets.csv", 'tweet_text', 'label')
print(Fore.GREEN + "Classification complete.", Style.RESET_ALL)
classify("./classification/pp_greek.csv", "./classification/pp_samosTweets.csv", "./classification/results/r_pp_samosTweets.csv", 'tweet_text', 'label')
print(Fore.GREEN + "Classification complete.", Style.RESET_ALL)

print(Fore.GREEN + "\nOperations complete.", Style.RESET_ALL)

```

