

# Εργασία Υπολογιστικής Γλωσσολογίας

## Ομαδική

### Μέλος Α

Ονοματεπώνυμο	Αλέξανδρος Φώτιος Ντογραματζής
ΑΜ	CS2180010
Σχολή	Τμήμα Πληροφορικής & Τηλεπικοινωνιών
Όνομα Μεταπτυχιακού	ΠΜΣ Πληροφορικής
Ειδίκευση	Διαχείριση Δεδομένων, Πληροφορίας και Γνώσης
Ακαδημαϊκό Έτος	2018-2019
Μάθημα	M170 Υπολογιστική Γλωσσολογία

### Μέλος Β

Ονοματεπώνυμο	Μανώλης Ρεπρές
ΑΜ	IC1180012
Σχολή	Τμήμα Πληροφορικής & Τηλεπικοινωνιών
Όνομα Μεταπτυχιακού	ΠΜΣ Τεχνολογίες Πληροφορικής και Επικοινωνιών
Ακαδημαϊκό Έτος	2018-2019
Μάθημα	M170 Υπολογιστική Γλωσσολογία

## Τίτλος: Fake News Detection

### Περιγραφή Προβλήματος

Τα τελευταία χρόνια , έχει αυξηθεί σε ποσοστό καθολικά σε όλα τα μέσα ενημέρωσης η παραπληροφόρηση και η μετάδοση ψεύτικων ειδήσεων. Μάλιστα με τη χρήση των νέων τεχνολογιών τέτοιες ψεύτικες πληροφορίες παρουσιάζονται πολύ αληθοφανείς που ακόμα και διάσημα μέσα ενημέρωσης πέφτουν στην παγίδα και τις μεταδίδουν, ενώ επιπλέον με την ανάπτυξη των τεχνολογιών του διαδικτύου τέτοιες ειδήσεις μεταδίδονται πάρα πολύ γρήγορα. Σε κοινωνικά δίκτυα όπως το facebook και το twitter παρατηρείται τα τελευταία χρόνια να γίνεται ολοένα και μεγαλύτερη παραπληροφόρηση. Το ζητούμενο είναι ότι τέτοιες πληροφορίες διαμορφώνουν μια στρεβλή εικόνα της πραγματικότητας. Συχνά μέσα από τέτοιες ειδήσεις μεταδίδονται επικίνδυνες ιδέες και παρόλο που τις περισσότερες φορές αποδεικνύεται γρήγορα ότι είναι μη έγκυρες και ψευδείς, έχουν ήδη επηρεάσει πολλούς ανθρώπους. Το γεγονός είχε σαν αποτέλεσμα τη μείωση της αξιοπιστίας των μέσων μαζικής ενημέρωσης διεθνώς. Τα fake news παρουσιάζονται με διάφορες μορφές:

- 🚩 Σάτιρα ή διακωμώδηση κάποιου γεγονότος ή κάποιου προσώπου: το περιεχόμενο τους είναι τέτοιο που προφανώς δύσκολα θα μπορούσε να θεωρηθεί αληθινό και μεταδίδονται από

troll sites που μεταδίδουν συνεχώς fake news. Ωστόσο υπάρχουν περιπτώσεις ανθρώπων που ξεγελιούνται. Γενικά τέτοιου είδους fake news δεν έχουν στόχο να κάνουν προπαγάνδα ή να βλάψουν κάποιον. Αν και ορισμένες φορές είναι δύσκολο να τεθούν όρια ανάμεσα στη σάτιρα και την προπαγάνδα.

- ✚ Λανθασμένη σύνδεση: Ο τίτλος ενός άρθρου είναι παραπλανητικός και δεν σχετίζεται άμεσα με το περιεχόμενο του(εδώ αν και είχαμε στις περισσότερες περιπτώσεις διαθέσιμους τους τίτλους των άρθρων, δεν τους χρησιμοποιήσαμε και δεν εντοπίσαμε ψευδείς ειδήσεις αυτής της μορφής).
- ✚ Παραπλανητικό περιεχόμενο: παραπλανητική χρήση πληροφοριών για ένα θέμα ή ένα άτομο.
- ✚ Ψευδές περιεχόμενο: Ανάμεσα στις πραγματικές ειδήσεις υπάρχουν διάσπαρτα ψεύτικα ή παραπλανητικά στοιχεία
- ✚ Πολλοί κακόβουλοι χρήστες ή ομάδες κακόβουλων χρηστών υποκλέπτουν στοιχεία τέτοια που τους δίνουν τη δυνατότητα να παρουσιάζονται ως αξιόπιστες πηγές και μεταδίδουν πολύ εύκολα ψεύτικες ειδήσεις. Με άλλα λόγια οι χρήστες ξεγελιούνται και νομίζουν ότι οι σελίδες τέτοιων χρηστών είναι όντως αξιόπιστες και θεωρούν έγκυρες τις ειδήσεις που διαβάζουν σε αυτές.
- ✚ Χρησιμοποίηση εικόνων ή άλλων οπτικοακουστικών μέσων και πληροφοριών που μπορεί να εμπεριέχουν και κάποια αληθινά στοιχεία που παρουσιάζονται με υποκειμενικό τρόπο και έχουν σαν στόχο την χειραγώγηση των μαζών
- ✚ Κατασκευασμένο περιεχόμενο: δημιουργείται ένα άρθρο με φανταστικές πληροφορίες και εικόνες με μόνο στόχο την παραπλάνηση

Εδώ απλώς αναφέρθηκαν διάφορες μορφές και είδη fake news , ωστόσο στα πλαίσια αυτής της εργασίας ασχοληθήκαμε αποκλειστικά με τον εντοπισμό fake news σε ειδησεογραφικά κείμενα χωρίς να ενδιαφερόμαστε για τη μορφή ή το είδος του fake news.

## Fake news και Πολιτική Προπαγάνδα

Το διαδίκτυο μετατρέπεται σταδιακά στο νέο μέσο μαζικής ενημέρωσης. Τα διαδραστικά μέσα κοινωνικής δικτύωσης υποκαθιστούν με ταχύτατους ρυθμούς τη χρήση των παλαιών μέσων μαζικής επικοινωνίας στον τομέα της ενημέρωσης. Οι εφημερίδες και τα περιοδικά βρίσκονται σε καθοδική πορεία εδώ και πολύ καιρό ενώ πρόσφατα άρχισε και η αποκαθήλωση της τηλεόρασης που κατείχε για δεκαετίες την πρωτοκαθεδρία στο χώρο των ΜΜΕ.

Το πιο πολυσυζητημένο μέσο κοινωνικής δικτύωσης στο χώρο της πολιτικής επικοινωνίας υπήρξε αναμφίβολα το Twitter. Σήμερα το Twitter αποτελεί βασικό εργαλείο διάδοσης πολιτικών μηνυμάτων και κατ'επέκταση μέσο πολιτικής προπαγάνδας.

Ένα άλλο σημερινό φαινόμενο που εντοπίζεται σε όλα τα μέσα ενημέρωσης και πολύ περισσότερο στα δίκτυα κοινωνικής δικτύωσης είναι η δημιουργία και διαμόρφωση ειδήσεων και νέων με βάση φήμες που πολλές φορές αποδεικνύονται ανυπόστατες. Χαρακτηριστικό παράδειγμα τέτοιας περίπτωσης είναι η ψευδής φήμη για απαγωγή ενός κοριτσιού από συμμορία ανδρών, στη πολιτεία του Jhorkhand το Μάιο του 2017 είχε ως αποτέλεσμα να λιντζαριστούν μέχρι θανάτου τουλάχιστον επτά αθώοι. Άλλη μια παρόμοια περίπτωση περιγράφεται παρακάτω. Μετά από μια καμπάνια του συστήματος υγείας στη πολιτεία της Κεράλα, για τη προώθηση του εμβολιασμού των βρεφών, το διαδίκτυο κατακλύστηκε από μηνύματα που ισχυρίζονταν πως τα εμβόλια εναντίον της ιλαράς και της ερυθράς, στοχεύουν στη μείωση της γονιμότητας των νεαρών Ινδών, ώστε να ελεγχθεί ο ρυθμός αύξησης του πληθυσμού στην εν δυνάμει πιο πολυπληθή χώρα του κόσμου

Η χρήση των ΜΜΕ για προπαγανδιστικούς σκοπούς είναι στενά συνδεδεμένη με την ίδια την ύπαρξη των ΜΜΕ. Δεν αποτελεί δηλαδή κάποιου είδους καινοτόμα προσέγγιση στο πεδίο των πολιτικών

ανταγωνισμών. Η διαφορά στη διαδικτυακή εκδοχή των ΜΜΕ έγκειται στο γεγονός ότι σήμερα υπάρχουν εργαλεία που αυτοματοποιούν τη διαδικασία παραγωγής πολιτικών μηνυμάτων και τη δημιουργία μηχανισμών διασποράς τους στο ευρύ κοινό. Δηλαδή στα νέα μέσα κοινωνικής δικτύωσης δεν υπάρχουν μόνο πομποί και δέκτες πολιτικών μηνυμάτων που σε κάθε περίπτωση είναι φυσικά πρόσωπα αλλά ένα συνολικό ανθρωπίνου και ρομπότ που υπηρετούν στρατευμένα μια βιομηχανική εκδοχή της πολιτικής προπαγάνδας.

Πολιτικοί, υποψήφιοι, εταιρείες, κόμματα, δημοσιογραφικοί οργανισμοί και think tanks εκμεταλλεύονται αυτοματοποιημένα συστήματα (bots) για να προβάλλουν τις απόψεις τους ή να δυσφημίσουν τους πολιτικούς τους αντιπάλους. Δεν θα ήταν λάθος να πούμε ότι πολλές πολιτικές εκστρατείες και καμπάνιες προσπαθούν να κάνουν προπαγάνδα και να μεταδώσουν πληθώρα ψεύτικων μηνυμάτων στο διαδίκτυο. Στόχος τους είναι ο έλεγχος μεγάλων ομάδων στα κοινωνικά δίκτυα και μέσω αυτού να κερδίσουν κάποιες ψήφους. Δεν είναι τυχαία άλλωστε η αύξηση μετάδοσης πολιτικών ειδήσεων-ανακοινώσεων στα κοινωνικά δίκτυα και η δημιουργία πολλών νέων λογαριασμών πολιτικών προσώπων(ψεύτικων και αληθινών). Τα στοιχεία αυτά πιστοποιούνται από πρόσφατη μελέτη της MIIR (Μεσογειακό Ινστιτούτο Ερευνητικής Δημοσιογραφίας) για τους ψεύτικους και αυτοματοποιημένους λογαριασμούς.

## Χρησιμότητα εφαρμογής

Αναφέρθηκε παραπάνω ότι ψεύτικες ειδήσεις σατυρικού περιεχομένου γίνονται συνήθως εύκολα αντιληπτές από αρκετούς ανθρώπους. Για τις υπόλοιπες κατηγορίες fake news υπάρχουν κάποια στοιχεία που μαρτυρούν με κάποια ακρίβεια που μαρτυρούν την αναξιπιστία μιας είδησης. Τέτοια στοιχεία είναι τα παρακάτω:

- ✚ Απουσία παράθεσης πηγής (ενημερωτικού μέσου δημοσιογράφου ή άλλου φορέα) από την οποία μεταδόθηκε για πρώτη φορά είδηση
- ✚ Η είδηση αναφέρεται σε ένα και μόνο μέσο ενημέρωσης και σε κανένα άλλο.

Γενικά, όμως οι άνθρωποι στην πλειονότητα τους συνήθως δεν κάνουν τον κόπο να ελέγξουν τα παραπάνω. Βέβαια, σήμερα οι πολίτες της ΕΕ είναι ιδιαίτερα καχύποπτοι με όλα τα ΜΜΕ και ειδικά με το διαδίκτυο. Μάλιστα το 37% του συνόλου των πολιτών της Ένωσης δηλώνουν ότι είναι απολύτως βέβαιοι πως έρχονται καθημερινά σε επαφή με ψευδείς ειδήσεις ή παραπλανητική πληροφόρηση μέσα από το διαδίκτυο ενώ άλλο ένα 32% θεωρούν πως πέφτουν θύματα παραπληροφόρησης τουλάχιστον μια φορά την εβδομάδα. Εν ολίγοις τα ⅔ των Ευρωπαίων είναι απολύτως βέβαιοι πως πέφτουν θύματα παραπληροφόρησης μέσω του διαδικτύου τουλάχιστον μια φορά την εβδομάδα. Όλα αυτά καταδεικνύουν την αναγκαιότητα ανάπτυξης ενός ανιχνευτών ψευδών ειδήσεων θα αποδειχθεί ένα πολύ χρήσιμο εργαλείο τόσο για τους απλούς πολίτες όσο και για τους δημοσιογράφους και τα ειδησεογραφικά πρακτορεία και θα καταφέρει να διασφαλίσει σε κάποιο βαθμό την αξιόπιστη και αδιάβλητη μετάδοση ειδήσεων και πληροφοριών. Για αυτό το λόγο άλλωστε έχουν επενδυθεί τόσα χρήματα στην έρευνα αυτού του πεδίου. Τα κύρια προβλήματα σε αυτή τη προσπάθεια είναι δύο ειδών:

- ✚ Το πρώτο βασικό είναι ότι δεν μπορεί να υπάρξει ένα σαφής διαχωρισμός μεταξύ ψευδών και αληθινών ειδήσεων, ειδικά όταν στο πολιτικό μήνυμα κυριαρχεί η έκφραση γνώμης. Για παράδειγμα θα μπορούσε να θεωρηθεί ως ψευδής μια είδηση που αναφέρει ότι το προηγούμενο Ιούνιο εισήλθαν στη κοινότητα 37.000 παράτυποι μετανάστες από τα θαλάσσια περάσματα της Μεσογείου. Αν οι αρμόδιες αρχές βεβαιώνουν ότι οι παράτυποι μετανάστες που εισήλθαν δεν υπερέβησαν τις 17.000, η παραπληροφόρηση είναι αυταπόδεικτη. Εάν όμως το πολιτικό μήνυμα που εκπέμπει και αναμεταδίδεται αναφέρει πως ένας τεράστιος αριθμός ανεπιθύμητων λαθρομεταναστών πέρασε για άλλη μια φορά τα σύνορα της ΕΕ, δεν είναι εξίσου εύκολο να αποδείξει κάποιος ότι το συγκεκριμένο μήνυμα που εκπέμπεται και αναμεταδίδεται αναφέρει πως ένα τεράστιος αριθμός ανεπιθύμητων λαθρομεταναστών πέρασε για άλλη μια φορά τα σύνορα της ΕΕ, δεν είναι εξίσου εύκολο να

αποδείξει κάποιος ότι το συγκεκριμένο μήνυμα είναι ψευδές προϊόν παραπληροφόρησης και ξενοφοβικής πολιτικής προπαγάνδας.

✚ Το δεύτερο πρόβλημα που δυστυχώς επιτείνει το πρώτο, είναι πως οι πολίτες δε γνωρίζουν με ποιό τρόπο χρησιμοποιούνται αυτοματοποιημένες αναρτήσεις για να προωθηθούν πολιτικά μηνύματα. Πρόκειται για μια αθέατη διαδικασία, από τους συνήθεις προπαγανδιστικούς μηχανισμούς. Είναι σίγουρα αντιδεολογικό, αλλά τα όρια της νομιμότητας και κατά πόσο αυτά παραβιάζονται μπορεί να γίνει ξεκάθαρο μόνο σε πολύ οριακές περιπτώσεις, Στο θέμα των bots, δηλαδή των λογαριασμών που προγραμματίζονται έτσι ώστε να αντιδρούν αυτόματα με τρόπο που θα αντιδρούσε και ένα φυσικό πρόσωπο, τείνει να καταστεί μείζον πολιτικό πρόβλημα. Σύμφωνα με μελέτη που πραγματοποίησε ομάδα Βρετανών καθηγητών που εργάζεται για το Oxford Internet Institute και ανακάλυψε δεκάδες εκατομμύρια αναρτήσεων από 7 μέσα κοινωνικής δικτύωσης σε 9 χώρες, η προγραμματιστική προπαγάνδα τείνει να μετατραπεί σε ένα από τα ισχυρότερα εργαλεία καταστράτηγησης της δημοκρατίας.

Για να γίνει ευκρινέστερο το πρόβλημα που δημιουργείται, αρκεί να συνειδητοποιήσουμε πως με τη χρήση των κατάλληλων προγραμματιστικών τεχνικών, μπορεί πέντε άνθρωποι με γνώσεις πληροφορικής να δημιουργήσουν την αίσθηση πως 5 άνθρωποι κανονικοί συζητούν για ένα κοινωνικό ζήτημα ή διαμαρτύρονται για κάποια πολιτική απόφαση.

Με άλλα λόγια, να καθίσταται δύσκολο να μετράμε τις κοινωνικές πλειοψηφίες που δημιουργούνται γύρω από συγκεκριμένα κοινωνικά ζητήματα με τη χρήση των γνωστών στατιστικών εργαλείων. Γιατί τώρα δε μετράμε μόνο τις γνώμες των φυσικών προσώπων. Συχνά καταφέρνουν να αθροίζονται μέσα σε αυτές και γνώμες που έχουν δημιουργηθεί από αλγόριθμους διάφορων bots. Αλλά αν πραγματικές καταμετρήσεις μπορούν να πραγματοποιούνται πλέον μόνο κάθε 4 ή 5 χρόνια στις φυσικές κάλπες, η πραγματική δημοκρατία είναι βέβαιο πως πάσχει.

## Σκοπός Εργασίας

Στα πλαίσια αυτής της εργασίας χρησιμοποιείται ως dataset το train.csv αρχείο που βρίσκεται στο παρακάτω link [train dataset](#). Το αρχείο αυτό έχει 20800 άρθρα ειδήσεων στα αγγλικά και για το καθένα υπάρχουν τα παρακάτω πεδία:

- ✚ id: μοναδικό id
- ✚ title: τίτλος
- ✚ author: συγγραφέας
- ✚ text: σώμα κειμένου
- ✚ label : εκφράζει αν το άρθρο είναι αξιόπιστο (τιμή 0) ή αναξιόπιστο (τιμή 1)

Το πρόβλημα πρόβλεψης της αξιοπιστίας ενημερωτικών άρθρων ανάγεται σε ένα πρόβλημα κατηγοριοποίησης κειμένων.

Χρησιμοποιούνται κάποιοι κλασικοί αλγόριθμοι μηχανικής μάθησης(παρουσιάζονται παρακάτω συνοπτικά) και με είσοδο το train set εκπαιδεύουν ένα μοντέλο και στη συνέχεια γίνεται για το test set εκτίμηση για το πόσο καλά δουλεύει το εκπαιδευμένο μοντέλο. Στη συνέχεια χτίζεται και σχεδιάζεται ένα μοντέλο με deep neural network και γίνεται εκπαίδευση του με το ίδιο σώμα κειμένων(άρθρων) ως είσοδο.

Επισημαίνεται ότι γενικά τα βασικά βήματα και στις δύο περιπτώσεις είναι τα εξής:

1. Εισαγωγή modules και μεθόδων που θα χρησιμοποιηθούν
2. Χρησιμοποιείται η βιβλιοθήκη panda για την ανάγνωση του αρχείου train.csv
3. Προ-επεξεργασία dataset και εξαγωγή διανυσμάτων με τα στοιχεία για εκπαίδευση

4. Χωρισμός dataset σε train και test set(στους αλγόριθμους μηχανικής μάθησης γίνεται 10-fold cross validation με 90% training set και 10% test set ενώ για το deep neural network γίνεται 90% train set 5% valid set- ρυθμίζει κάποιες επιπλέον παραμέτρους του νευρωνικού – 5% test set)
5. Επιλογή παραμέτρων αλγορίθμων Μηχανικής Μάθησης/ Σχεδιασμός deep neural network
6. Εκπαίδευση μοντέλου με training set
7. Αξιολόγηση μοντέλου με test set

Τέλος χρησιμοποιείται αυτό το εκπαιδευμένο deep neural network για να κάνουμε προβλέψεις σε άρθρα real time.

Η εργασία δομείται στις εξής ενότητες:

- ✚ Υλοποίηση Fake News Detector με χρήση αλγορίθμων μηχανικής μάθησης
- ✚ Υλοποίηση Fake News Detector με χρήση deep neural networks
- ✚ Αποτελέσματα εκπαίδευσης παραπάνω μοντέλων
- ✚ Συνοπτική περιγραφή εφαρμογής που προβλέπει σε real time αν ένα άρθρο είναι αξιόπιστο ή όχι
- ✚ Αποτελέσματα και σχόλια για προβλέψεις σε άρθρα σε πραγματικό χρόνο
- ✚ Μελλοντικές Επεκτάσεις

## Υλοποίηση Fake News Detector με χρήση αλγορίθμων μηχανικής μάθησης

### Προεπεξεργασία Δεδομένων

Αρχικά αφαιρέσαμε τα stop words (καθημερινές λέξεις που υπάρχουν σε όλα τα κείμενα σε υψηλό βαθμό και δεν επηρεάζουν το νόημα) και μετατρέψαμε όλους τους χαρακτήρες των λέξεων σε πεζούς(μικρά γράμματα). Στη συνέχεια ακολουθεί μια διαδικασία tokenization των κειμένων του dataset. Για το tokenization επιλέχθηκε μοντέλο διγραμμάτων δηλαδή δεν πήραμε και κωδικοποιήσαμε κάθε λέξη με ένα word index αλλά δυάδες λέξεων. Στη συνέχεια υπολογίστηκε κάθε τέτοια δυάδα πόσες φορές εμφανίζεται σε κάθε κείμενο και έτσι προκύπτει για κάθε κείμενο ένα διάνυσμα που έχει σαν στοιχεία το πλήθος φορές που εμφανίζεται σε αυτό κάθε δυάδα λέξεων από το σώμα κειμένων(count vectorizer). Ωστόσο αυτή η προσέγγιση παρουσιάζει το μειονέκτημα ότι δίνει μεγαλύτερη βαρύτητα σε όρους που εμφανίζονται πιο συχνά στα άρθρα και μικρότερη βαρύτητα σε όρους που εμφανίζονται πιο σπάνια. Γενικά όμως τέτοιοι όροι που εμφανίζονται συχνά δεν αποτελούν καλή πληροφορία για να εκτιμηθεί αν είναι άρθρο είναι fake ή true. Για να λυθεί αυτό το πρόβλημα υπάρχουν tf-idf weights(tfidf vectorizer),τους οποίους και χρησιμοποιήσαμε στην ενότητα αυτή για να πάρουμε τα στοιχεία κειμένου που θα δοθούν ως είσοδοι στους αλγορίθμους μηχανικής μάθησης. Ουσιαστικά αυτό που διαφοροποιεί την συγκεκριμένη προσέγγιση από την προηγούμενη είναι ότι δεν υπολογίζει τη συχνότητα εμφάνισης ενός όρου αλλά και πόσο σημαντικός είναι αυτός όρος για αυτό το κείμενο δηλαδή δεν βλέπει τον όρο ανεξάρτητα από τους υπόλοιπους όρους των κειμένων αλλά σε συσχέτιση με αυτούς. Πρακτικά αυτό έχει σαν αποτέλεσμα να υποβαθμίζει τη σημασία όρων με μεγάλη συχνότητα και να αναβαθμίζει τη σημασία των όρων με μικρή συχνότητα. Αυτό γίνεται βρίσκοντας όχι το πλήθος εμφάνισης ενός όρου αλλά τη συχνότητα εμφάνισης του. Έτσι υπολογίζεται

για κάθε κείμενο όπως και πριν το διάνυσμα με το πλήθος εμφάνισης κάθε όρου και στη συνέχεια το διάνυσμα αυτό κανονικοποιείται διαιρώντας με το μέτρο του. Αυτό είναι το term frequency. Το inverse document frequency είναι ένα μέγεθος που δείχνει για κάθε όρο πόση πληροφορία παρέχει. Οι όροι αυτοί αποτελούν τα στοιχεία του διαγώνιου πίνακα inverse document frequency. Υπάρχει ακόμα ο πίνακας όπου κάθε γραμμή του είναι το διάνυσμα με τιμές συχνότητας των όρων για κάθε έγγραφο. Τα στοιχεία του πίνακα που είναι το εξωτερικό γινόμενο του πίνακα των διανυσμάτων συχνότητας όρων με τον διαγώνιο πίνακα είναι οι τιμές των tf-idf για κάθε όρο.

## Παρουσίαση αλγορίθμων μηχανικής μάθησης που χρησιμοποιήθηκαν για την εκπαίδευση του μοντέλου ενός ανιχνευτή ψευδών ειδήσεων

### Support Vector Machines (SVM)

Αποτελεί μια τεχνική κατηγοριοποίησης η οποία έχει τύχει αξιοσημείωτης προσοχής. Έχει τις ρίζες της στη θεωρία της στατιστικής εκπαίδευσης και έχει επιδείξει ελπιδοφόρα εμπειρικά αποτελέσματα σε πολλές πρακτικές εφαρμογές(π.χ αναγνώριση χειρόγραφων ψηφίων, κατηγοριοποίηση κειμένων και μπορεί να δουλεύει καλά για δεδομένα πολλών διαστάσεων. Μια άλλη μοναδική πτυχή αυτής της προσέγγισης είναι ότι αντιπροσωπεύει το όριο της απόφασης, χρησιμοποιώντας υποσύνολα δειγμάτων εκπαίδευσης γνωστά και ως διανύσματα υποστήριξης. Η μέθοδος αυτή βρίσκει υπερεπίπεδα που να τοποθετούν όλα τα δείγματα της ίδιας κατηγορίας να βρίσκονται στο ίδιο επίπεδο. Σε ένα training set μπορεί να υπάρχουν άπειρα υπερεπίπεδα που διαχωρίζουν σε κατηγορίες τα δείγματα με 100% επιτυχία αλλά κανείς δεν εγγυάται ότι όλα αυτά τα υπερεπίπεδα θα δουλεύουν καλά και για άγνωστα δείγματα. Ο κατηγοριοποιητής πρέπει να επιλέξει τα υπερεπίπεδα εκείνα που αντιπροσωπεύουν τα όρια απόφασης του με βάση το πόσο καλά αναμένεται να λειτουργήσουν πάνω στα δείγματα ελέγχου. Ο αλγόριθμος προτιμά τα όρια απόφασης κατηγοριών με μεγάλα περιθώρια γιατί τείνουν να έχουν καλύτερα σφάλμα γενίκευσής(δηλαδή καλύτερα αποτελέσματα σε άγνωστα δείγματα) σε σχέση με εκείνα που έχουν μικρά περιθώρια(επιρρεπής σε υπερπροσαρμογή των δεδομένων στα training data). Δίνει αρκετά καλά αποτελέσματα για το test set μας ενώ δουλεύει και αρκετά γρήγορα για τον γραμμικό ταξινομητή SVM.[1]

### Logistic Regression(Λογιστική Παλινδρόμηση)

Αποτελεί μια ακόμη τεχνική που δανείζεται η μηχανική μάθηση από το πεδίο της στατιστικής και χρησιμοποιείται σε προβλήματα δυαδικής ταξινόμησης. [2]Συνιστά μια μέθοδο παλινδρόμησης και όπως όλες οι μέθοδοι παλινδρόμησης ακολουθεί μια προβλεπτική ανάλυση. Χρησιμοποιείται όταν η εξαρτημένη μεταβλητή είναι δυαδική και περιγράφει τα δεδομένα και εξηγεί τις μεταξύ τους σχέσεις(σχέσεις ανάμεσα στην εξαρτημένη μεταβλητή και σε μία ή περισσότερα ανεξάρτητες μεταβλητές.[3] Έχει στόχο να βρει το καλύτερο fitting model για να περιγράψει τη σχέση μεταξύ μια χαρακτηριστικής συνάρτησης διαχωρισμού και ενός συνόλου ανεξάρτητων μεταβλητών. [4]. Χρησιμοποιείται για την αναπαράσταση της μια συνάρτηση που μοιάζει με αυτή της γραμμικής παλινδρόμησης και για αυτό το λόγο αποτελεί μια γραμμική μέθοδο και η οποία μοντελοποιεί την πιθανότητα της προεπιλεγμένης κλάσης. Ωστόσο οι προβλέψεις(πιθανότητες ένα δείγμα να ανήκει σε μια κατηγορία) δεν συνιστούν ένα γραμμικό συνδυασμό των εισόδων όπως στη γραμμική παλινδρόμηση αλλά πριν το τελικό στάδιο της πρόβλεψης προηγείται ένας λογαριθμικός μετασχηματισμός. Ουσιαστικά το μοντέλο είναι ακόμα ένας γραμμικός μετασχηματισμός των εισόδων που συνδέεται όμως με τα η λογαριθμική πιθανότητα ένα δείγμα να ανήκει σε μια κλάση. Οι παράμετροι της συνάρτησης της λογιστικής παλινδρόμησης υπολογίζονται κατά την εκπαίδευση του μοντέλου με χρήση των training data. [2]



## Multinomial Naïve Bayes

Αποτελεί μια ακόμη δημοφιλή ταξινομητή μηχανικής μάθησης για categorical data και ειδικά για κείμενα. Σε αυτή τα δείγματα(διανύσματα στοιχείων ) αναπαριστούν τις συχνότητες με τις οποίες συγκεκριμένες γεγονότα(διγράμματα λέξεων εδώ) συμβαίνουν. Το διάνυσμα στοιχείων είναι επομένως ένα ιστόγραμμα που εμπεριέχει πόσες φορές συνέβη ένα γεγονός σε κάποιο στιγμιότυπο(άρθρο). Ο ταξινομητής ουσιαστικά υπολογίζει δεσμευμένες πιθανότητες της μορφής  $p(x|C_k)$  με  $x$ :το διάνυσμα στοιχείων και  $C_k$  να αντιστοιχεί στην κατηγορία του αντικειμένου. Η προσέγγιση αυτή είναι γραμμική στον λογαριθμικό χώρο.[5]

## Random Forest

Το τυχαίο δάσος(Random Forest) είναι μια κατηγορία μεθόδων ομάδας, η οποία έχει σχεδιασθεί ειδικά για κατηγοριοποιητές δέντρων απόφασης. Συνδυάζει τις προβλέψεις που γίνονται από πολλά δέντρα απόφασης(εδώ πήραμε 10), όπου κάθε δένδρο παράγεται με βάση από τις τιμές ενός ανεξάρτητου συνόλου τυχαίων διανυσμάτων. Τα τυχαία διανύσματα παράγονται από μια σταθερή κατανομή πιθανότητας.. Η εμφωλίωση με χρήση δένδρων απόφασης αποτελεί μια ειδική περίπτωση τυχαίων δασών, όπου η τυχειότητα εισάγεται στη διαδικασία δημιουργίας του μοντέλου μέσα από την επιλογή τυχαία  $N$  δειγμάτων, με αντικατάσταση από το αρχικό σύνολο εκπαίδευσης. Η εμφωλίωση χρησιμοποιεί την ίδια ομοιόμορφη κατανομή πιθανότητας για να παράγει τα αυτοδύναμα δείγματα της , κατά τη διάρκεια της συνολικής διαδικασίας δημιουργίας του μοντέλου. Υπάρχουν διάφοροι τρόποι παραγωγής των δέντρων του αλγορίθμου. Κάθε δέντρο απόφασης χρησιμοποιεί ένα τυχαίο διάνυσμα, το οποίο παράγεται από μια αμετάβλητη κατανομή πιθανότητας.[6]

## Τρόπος Αξιολόγησης αλγορίθμων

### k-Fold Cross Validation

Υπάρχουν διάφοροι τρόποι μέτρησης της απόδοσης του μοντέλου στο σύνολο ελέγχου(test set) γιατί ένα τέτοιο μέτρο παρέχει μια αμερόληπτη εκτίμηση του σφάλματος γενίκευσης του μοντέλου. Η ακρίβεια ή ο βαθμός σφάλματος που υπολογίζεται από το σύνολο ελέγχου, μπορεί επίσης να χρησιμοποιηθεί για τη σύγκριση της σχετικής απόδοσης διαφορετικών classifiers στο ίδιο πεδίο. Ωστόσο για να γίνει αυτό πρέπει να είναι γνωστές τα labels των εγγράφων του test set . Δεδομένου αυτών των labels εδώ χρησιμοποιείται για την εκτίμηση της απόδοσης των classifier η μέθοδος k-fold Cross Validation(  $k=10$  εδώ). Η προσέγγιση αυτή χωρίζει τα δεδομένα σε  $k$  ίσου μεγέθους τμήματα και κάθε φορά ένα τμήμα επιλέγεται για test και τα υπόλοιπα για train . Αυτή η διαδικασία επαναλαμβάνεται  $k$  φορές, ώστε κάθε τμήμα να χρησιμοποιηθεί για έλεγχο ακριβώς μία φορά.

## Μετρικές Σχέσεις

Για την αξιολόγηση των μοντέλων μηχανικής μάθησης(αλλά και του deep neural network) που χρησιμοποιήθηκαν για fake news detection χρησιμοποιούνται οι παρακάτω μετρικές σχέσεις

- 🚩 **Accuracy:** αριθμός σωστών προβλέψεων προς συνολικό αριθμό προβλέψεων
- 🚩 **Prediction:** αριθμός αυτών που αξιολογήθηκαν σωστά ως fake προς το συνολικό αριθμό που αξιολογήθηκαν ως fake (αντίστοιχα και για real).

- ✚ **Recall:** αριθμός αυτών που αξιολογήθηκαν σωστά ως fake προς το συνολικό αριθμό που είναι πραγματικά fake(σωστά αξιολογημένα fake και λανθασμένα αξιολογημένα fake)- αντίστοιχα και για real
- ✚ **F-Measure:** αρμονικός μέσος precision και recall

## Σχεδιασμός και Εκπαίδευση deep neural network που χρησιμοποιήθηκε για την εκπαίδευση του μοντέλου ενός ανιχνευτή ψευδών ειδήσεων

### Προεπεξεργασία Δεδομένων

Με τον ίδιο τρόπο όπως έγινε και στην προηγούμενη ενότητα, αφαιρούνται stop words και σημεία στίξης από το σώμα κειμένου των άρθρων. Στη συνέχεια μετατρέπονται όλοι χαρακτήρες σε πεζά γράμματα και γίνεται tokenization με μονογράμματα εδώ(δηλαδή απλά λέξεις). Ακολουθεί η διάσπαση των δεδομένων(πλέον είναι οι ακολουθίες m word\_ids των αρχικών δεδομένων) σε τρία κομμάτια 90% train set, 5% validation set, 5% test set. Το ίδιο γίνεται και για τα labels.

### Σχεδιασμός και Ανάλυση Νευρωνικού Δικτύου

#### Embedding Layer of Neural Network

Εδώ το μοντέλο του fake news detector σε αντίθεση με την κλασική προσέγγιση που παρουσιάστηκε παραπάνω δεν χρησιμοποιείται κάποιος vectorizer ο οποίος θα μετράει τη συχνότητα εμφάνισης λέξεων ή εκφράσεων σε κείμενα και θα προκύπτει ένας πίνακας με τα διανύσματα στοιχείων των κειμένων μαζί με τα labels που θα δίνονται σε κάποιο γνωστό αλγόριθμο classification όπως αυτοί που αναφέρθηκαν παραπάνω. Εδώ η ουσία είναι να μη στηριχθεί ο αλγόριθμος που θα χρησιμοποιηθεί απλά στη συχνότητα εμφάνισης μιας λέξης στα κείμενα αλλά να γίνονται λογικοί συνδυασμοί-συσχετίσεις μεταξύ των λέξεων αυτών. Έτσι κάθε μια λέξη(ids) από τα δεδομένα πρέπει να αντιστοιχεί σε ένα k-διαστατό διάνυσμα που περιγράφει τη θέση και τον προσανατολισμό της λέξης στον k-διάστατο χώρο και κωδικοποιεί την πληροφορία για την σχέση των λέξεων σε χωρικό επίπεδο μέσω των σχέσεων των διανυσμάτων. Έτσι γίνεται αντιληπτό ότι λέξεις με κάποια συνάφεια θα μοιράζονται την ίδια ή περίπου την ίδια τιμή σε κάποια ή κάποιες διαστάσεις του χώρου. Για παράδειγμα οι λέξεις σκύλος και γάτα μοιράζονται μια διάσταση του χώρου ως κατοικίδια ζώα και η τίγρη και ο λύκος μοιράζονται επίσης μια άλλη διάσταση του χώρου ως άγρια ζώα, αλλά υπάρχει και ανάμεσα τους μια σχέση(γάτα->τίγρης, σκύλος->λύκος). Σε πρώτο επίπεδο είναι όλα ζώα αλλά και βιολογικά σκύλος και λύκος μοιράζονται κάποια κοινά στοιχεία και το ίδιο η γάτα και η τίγρης. Ουσιαστικά υπάρχει ένα νευρωνικό δίκτυο που δέχεται σαν είσοδο ένα διάνυσμα με κάποια στοιχεία για μια λέξη και μας δίνει ένα embedding vector σαν έξοδο. Μέσω των embedding vectors των λέξεων που προκύπτουν ως έξοδοι του νευρωνικού δικτύου, επιχειρείται να χτιστεί ένα δομημένο δίκτυο με τις σχέσεις των σχέσεων μεταξύ των λέξεων θα μπορούσαμε να το χαρακτηρίσουμε και σαν μια οντολογία. Ένας τρόπος αναπαράστασης της σχέσης αυτών των λέξεων είναι μέσω τις διαφορές των διανυσμάτων τους. Σημειώνεται ότι σημασιολογικά οι λέξεις είναι μοναδικές και ακόμα και λέξεις με πολύ παρόμοια σημασία π.χ άνθρωπος-άτομο θα παρουσιάζουν διαφορές στις τιμές των διανυσμάτων τους.



Για τον υπολογισμό του embedded matrix, πρέπει να χρησιμοποιηθεί ένα λεξικό το οποίο θα περνάει από ένα νευρωνικό δίκτυο όπως είπαμε και θα κάνει την εκπαίδευση και θα δίνει το embedded matrix. Ένας τρόπος να γίνει αυτό είναι να σχεδιασθεί ένα μοντέλο που θα μπορούσε να μαντεύει το περιεχόμενο της λέξης από την ίδια την λέξη. Η στρατηγική αυτή είναι χρήσιμη και μας επιτρέπει να επιβεβαιώνουμε ότι το μοντέλο μαντεύει πάντα σωστά αλλά είναι απίθανο να επιβεβαιωθεί αν είναι όντως σωστό. Το embedding matrix είναι τυχαία αρχικοποιημένο και οι τιμές του προσδιορίζονται από το μοντέλο που έχει προκύψει. Το μοντέλο εκπαιδεύεται χρησιμοποιώντας gradient descent(continuous bag of words : Efficient Estimation of Word Representations in Vector Space [7]).

Αρχικά σκεφτήκαμε να χρησιμοποιηθεί για την εκτίμηση του embedding matrix οι λέξεις που είναι στο dataset με τα άρθρα που χρησιμοποιήσαμε, όμως σύντομα μια τέτοια λύση αποκλείστηκε για τους παρακάτω λόγους:

- ✚ Το πλήθος των κειμένων δεν είναι τόσο μεγάλο και συνεπώς το πλήθος των λέξεων(tokens) δεν είναι μεγάλο ούτε συγκρίσιμο σε μέγεθος με αυτό μιας πραγματικά μεγάλης συλλογής κειμένων ή ενός λεξικού.
- ✚ Με τη χρήση λέξεων μόνο από το συγκεκριμένο dataset θα φτιαχτεί ένας embedding πίνακας ο οποίος θα επικεντρώνεται περισσότερο στην εύρεση συσχετίσεων μεταξύ λέξεων σχετικών με τα θέματα που πραγματεύονται τα άρθρα του dataset.

Συμπεράναμε ότι με χρήση του δικού μας dataset θα γινόταν εκτίμηση συσχετίσεων μεταξύ λέξεων από ένα σύνολο λέξεων που δεν είναι αντιπροσωπευτικό δείγμα του συνόλου των λέξεων της αγγλικής γλώσσας και συνεπώς και το embedding matrix που θα προέκυπτε δε θα ήταν αντιπροσωπευτικό. Για αυτό χρησιμοποιήθηκε κάποιο μεγαλύτερο dataset άρθρων που περιλαμβάνει πολύ περισσότερες λέξεις.

Ωστόσο η υλοποίηση και εκπαίδευση του νευρωνικού δικτύου για την εξαγωγή του embedding matrix φαίνεται να είναι μια δύσκολη και ιδιαίτερα χρονοβόρα διαδικασία και

για αυτό το λόγο χρησιμοποιήθηκε το έτοιμο embedding matrix που υλοποιήθηκε στα πλαίσια του Stanford GloVe project[8] και είναι διαθέσιμο στην παρακάτω διεύθυνση embedding matrix [9] όπου τα δεδομένα λέξεων έχουν παρθεί Wikipedia 2014 [10]& Gigaword 5 [11] και περιλαμβάνει 400.000 λέξεις. Στο παραπάνω link χρησιμοποιούνται embedding matrix διαφορετικών μεγεθών, εμείς προς το παρόν χρησιμοποιούμε το μεγαλύτερο. Ουσιαστικά αυτό που γίνεται είναι να γίνει ανάγνωση του αρχείου του embedding matrix. Κάθε γραμμή αυτού του αρχείου έχει την εξής μορφή:

`<word> <embedding vector>`

Τα στοιχεία του embedding\_vector χωρίζονται με space μεταξύ τους. Διαβάζουμε το αρχείο γραμμή-γραμμή και φτιάχνεται ένα πίνακας-διάνυσμα για το κάθε του embedding\_vector το οποίο αποθηκεύεται ως value σε ένα dictionary στο record με key την αντίστοιχη λέξη. Στη συνέχεια ελέγχεται ποια από αυτές τις λέξεις σε αυτό το dictionary υπάρχουν στις ακολουθίες m word\_ids των άρθρων του αρχικού dataset. Αν κάποια λέξη του dataset δεν βρεθεί σαν key στο dictionary που φτιάχτηκε παραπάνω, τότε embedding vector της θεωρείται μηδενικό σε διαφορετική περίπτωση παίρνουμε το value της αντίστοιχης λέξης-key από το dictionary. Πλέον όταν τελειώσει αυτή η διαδικασία έχουμε διαθέσιμο τα βάρη του embedding layer που υλοποιείται χρησιμοποιώντας την αντίστοιχη μέθοδο του tensorflow.keras.

### Glove

Ο GloVe(Global Vectors for Word Representation) είναι ένας αλγόριθμος μη επιβλεπόμενης μάθησης για να αποκτηθούν διανύσματα για την αναπαράσταση των λέξεων. Το training πραγματοποιείται σε συγκεντρωτικά στατιστικά στοιχεία από ένα πολύ μεγάλο σώμα κειμένων όπου εμφανίζονται πάρα πολλές λέξεις. Έχουν χρησιμοποιηθεί pre-trained word vectors. Για την εύρεση της συσχέτισης δύο διανυσμάτων λέξεων χρησιμοποιείται το cosine similarity που παρέχει μια αποτελεσματική μέθοδο μέτρησης λεξιλογικών ή σημασιολογικών ομοιοτήτων μεταξύ των λέξεων αυτών και με αυτό τον τρόπο βρίσκονται κοντινοί γείτονες των λέξεων αυτών είναι σχετικές λέξεις αλλά όχι τόσο συνηθισμένες. Έτσι υπολογίζεται ένα μέγεθος σχετικότητας δύο λέξεων όμως η λογική αυτή αποδεικνύεται

προβληματική γιατί οι συσχετίσεις μεταξύ των λέξεων πολλές φορές είναι αρκετά σύνθετες (π.χ άντρας –γυναίκα θεωρούνται ίδια γιατί είναι άνθρωποι). Με το GloVe επιχειρούνται να λυθούν τέτοια προβλήματα. Χρησιμοποιεί τιμές δεσμευμένων πιθανοτήτων  $p(\text{word}_i | \text{word}_j)$  με στόχο να εντοπίσει επιπλέον σημασιολογικές πληροφορίες. Το training του GloVe γίνεται με στόχο να μάθει διανύσματα λέξεων τέτοια ώστε το εξωτερικό γινόμενο τους να ισούται με το  $\log(p(\text{word}_i | \text{word}_j))$ . Έτσι λογάριθμος του λόγου 2 τέτοιων δεσμευμένων πιθανοτήτων συσχετίζεται με την διαφορά των διανυσμάτων στο χώρο των διανυσμάτων των λέξεων. Περισσότερες λεπτομέρειες: GloVe: Global Vectors for Word Representation[12]

Χρησιμοποιήθηκαν οι παρακάτω τιμές  $n=50000$ ,  $m=1500$ ,  $k=300$

### Next Layers of Neural Network

Αφού καθοριστεί η είσοδος στο embedding layer γίνονται οι κατάλληλοι υπολογισμοί και προκύπτει ένα διάνυσμα εξόδου που δίνεται στα επόμενα επίπεδα τα οποία έχουν καθοριστεί ως εξής:

- ✚ Χρησιμοποιείται ένα μονοδιάστατο συνελκτικό επίπεδο με `window_size` 5 δηλαδή δημιουργείται ένα `convolution_kernel` (το default είναι το `glorot_uniform`-εξήγηση παρακάτω) μεγέθους 5 και στη συνέχεια εφαρμόζεται στα δεδομένα εισόδου συνέλιξη με αυτό το `kernel` και δίνουν έξοδο ένα διάνυσμα μεγέθους 64 (συνολικός αριθμός `kernels-filters` που θα χρησιμοποιηθούν). Για συνάρτηση ενεργοποίησης του νευρώνων ,χρησιμοποιείται η `relu` ( με τις default τιμές στο `keras`, ενεργοποιείται ο νευρώνας για θετικές τιμές και είναι ανενεργός για μη θετικές τιμές)
- ✚ Για να επιτευχθεί αυτή η μείωση διαστάσεων που απαιτείται σε αυτό το συνελκτικό επίπεδο, μετά τη συνέλιξη εφαρμόζεται ένα `pooling` φίλτρο μεγίστου που παίρνει την μέγιστη έξοδο από κάθε ομάδα 5 νευρώνων του προηγούμενου επιπέδου (έξοδοι μετά τη συνέλιξη) `MaxPooling`
- ✚ Ακολουθεί ένα δεύτερο μονοδιάστατο συνελκτικό επίπεδο με `window_size` 3 και δημιουργείται ένα `convolution_kernel` μεγέθους 3 και στη συνέχεια εφαρμόζεται στα δεδομένα εισόδου συνέλιξη με αυτό το `kernel` και δίνουν έξοδο ένα διάνυσμα μεγέθους 128. Για συνάρτηση ενεργοποίησης του νευρώνων ,χρησιμοποιείται η `relu`
- ✚ Πάλι εφαρμόζεται η συνάρτηση `MaxPooling` με τον ίδιο τρόπο με προηγούμενως αλλά τώρα παίρνει την μέγιστη έξοδο από κάθε ομάδα 3 νευρώνων του προηγούμενου επιπέδου (έξοδοι μετά τη συνέλιξη)
- ✚ Ακολουθεί ένα τρίτο μονοδιάστατο συνελκτικό επίπεδο με `window_size` 2 και δημιουργείται ένα `convolution_kernel` μεγέθους 2 και στη συνέχεια εφαρμόζεται στα δεδομένα εισόδου συνέλιξη με αυτό το `kernel` και δίνουν έξοδο ένα διάνυσμα μεγέθους 256.
- ✚ Εφαρμόζεται στα δεδομένα εξόδου της συνέλιξης φίλτρο `global average pooling` χρησιμοποιείται όπως και το `max pooling` φίλτρο για να μειώσει τον αριθμό των χωρικών διαστάσεων της εξόδου που προέκυψε μετά τη συνέλιξη αλλά σε μεγαλύτερο βαθμό από ότι ένα `max pooling` φίλτρο. Ουσιαστικά παίρνει τον καθολικό μέσο όρο των εξόδων
- ✚ Φτιάχνεται ένα πλήρως διασυνδεδεμένο επίπεδο στο νευρωνικό επίπεδο με διάνυσμα εξόδου με μήκος 2048, παίρνει δηλαδή το εξωτερικό γινόμενο της εισόδου με ένα `kernel` που φτιάχνει το επίπεδο (το default είναι το `glorot_uniform`) που τραβάει δείγματα από μια ομοιόμορφη κατανομή με τιμές από  $[-1,1]$  με

$$l = \sqrt{\frac{6}{(fan_{in} + fan_{out})}}$$

fan\_in: πλήθος διανυσμάτων εισόδου στο επίπεδο

fan\_out: πλήθος διανυσμάτων εξόδου στο επίπεδο)

Συνθήκη ενεργοποίησης νευρώνων relu.

- ✚ Γίνεται dropout σε ποσοστό x1 δηλαδή επιλέγονται τόσοι κόμβοι όσοι αντιστοιχούν στο ποσοστό x1 και πετιούνται για να αποφεύγεται το overfitting
- ✚ Φτιάχνεται με ίδιο τρόπο ένα δεύτερο πλήρως διασυνδεδεμένο επίπεδο στο νευρωνικό επίπεδο με διάνυσμα εξόδου με μήκος 512.
- ✚ Γίνεται πάλι dropout σε ποσοστό x2 δηλαδή επιλέγονται τόσοι κόμβοι όσοι αντιστοιχούν στο ποσοστό x2 και πετιούνται για να αποφεύγεται το overfitting.
- ✚ Φτιάχνεται ένα πλήρως διασυνδεδεμένο δίκτυο με 2 εξόδους και συνάρτηση ενεργοποίησης νευρώνα εξόδου softmax ώστε το συνολικό άθροισμα των εξόδων να είναι 1 δηλαδή οι εξοδοι παίρνουν θετικές τιμές από 0-1 τέτοιες ώστε το άθροισμά τους να κάνει 1.

## Υλοποίηση Μοντέλου

Ορίζονται για το μοντέλο η διάσταση και ο τύπος δεδομένων εισόδου και εξόδου. Ορίζεται ο τύπος της αντικειμενική συνάρτηση που επιχειρεί να βελτιστοποιήσει το μοντέλο(ως μοντέλο χαρακτηρίζεται το νευρωνικό δίκτυο που περιγράφηκε παραπάνω είναι η

```
model = Model(input_data, out)
model.compile(loss='sparse_categorical_crossentropy', optimizer='adamax', metrics=['acc'])
```

keras.losses.sparse\_categorical\_crossentropy. Προσδιορίζεται η τεχνική βελτιστοποίησης του μοντέλου Adamax [13] . Τέλος προσδιορίζεται το accuracy ως η μετρική ποσότητα(ποσοστό σωστά ταξινομημένων άρθρων) της οποίας η τιμή πρέπει να μεγιστοποιηθεί

## Εκπαίδευση Μοντέλου

Γίνεται εκπαίδευση του μοντέλου με 90% των m word\_ids ακολουθιών των άρθρων

του train.csv ως train\_set και τα άλλα 2 5% ως valid & test set. Έγιναν διάφορες δοκιμές με τις παραμέτρους του μοντέλου και καταλήξαμε:

Οι τελικοί παράμετροι που επιλέχθηκαν μετά από εξαντλητικές δοκιμές ήταν:

α)Glove: glove.6B.300d.txt

Dropout για το πρώτο layer που δέχεται την είσοδο 0.8

Dropout για το επόμενο layer 0.3

Epochs: 60

batch\_size: 4500

β)Για ταχύτερη εκμάθηση με λίγο μικρότερη ακρίβεια στα τελικά αποτελέσματα:

Glove: glove.6B.300d.txt

*Dropout για το πρώτο layer που δέχεται την είσοδο 0.8*

*Dropout για το επόμενο layer 0.5*

*Epochs: 25*

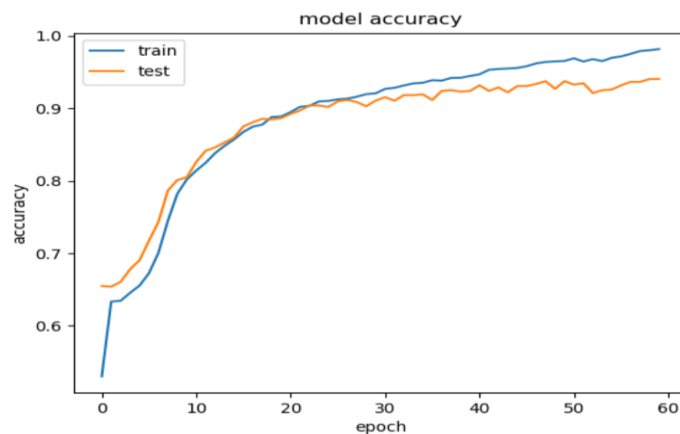
*batch\_size: 2048*

## Αξιολόγηση του deep neural network και των υπόλοιπων αλγορίθμων μηχανικής μάθησης

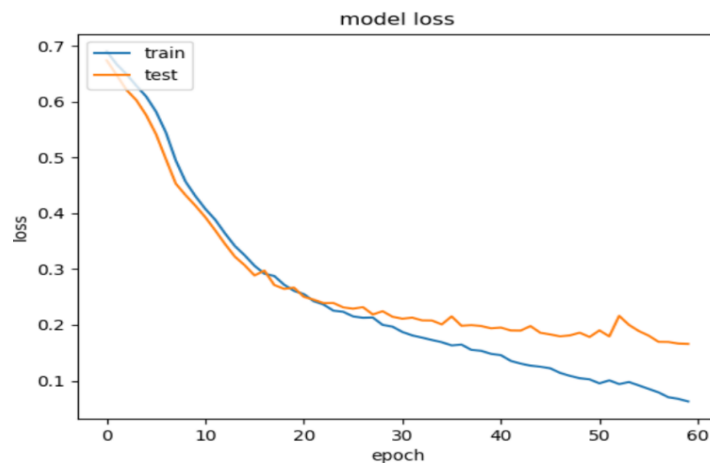
Γενικά από τα αποτελέσματα φαίνεται το νευρωνικό δίκτυο που σχεδιάσαμε και υλοποιήσαμε, δίνει καλύτερα αποτελέσματα από τους αλγόριθμους μηχανικής οι οποίοι δίνουν όμως και αυτοί καλά αποτελέσματα. Εδώ πρέπει να επισημανθεί ότι το μοντέλο που υλοποιήσαμε, δύναται να προβλέπει επιτυχώς τα fake άρθρα σε πληθώρα περιπτώσεων που δοκιμάστηκε. Ωστόσο το γεγονός ότι το μοντέλο σχεδιάστηκε έτσι ώστε να μην κατατάσσει κάποιο άρθρο που είναι fake ως real και λιγότερο στην ελαχιστοποίηση των σφαλμάτων ταξινόμησης, έχει σαν αποτέλεσμα να δίνει έμφαση σε μία από τις δύο κατηγορίες και αυτό έχει σαν άμεση συνέπεια πολύ σύντομα άρθρα (1-2 προτάσεων) να χαρακτηριστούν εσφαλμένα ως fake από το μοντέλο. Παρουσιάζονται τα διαγράμματα για το accuracy και το loss σε συνάρτηση του αριθμού το epochs(αριθμών επαναλήψεων εκπαίδευσης). Για τις περιπτώσεις α & β της υποενότητας εκπαίδευση μοντέλου της προηγούμενης ενότητας

α) Για το τελικό μοντέλο (60 epochs)

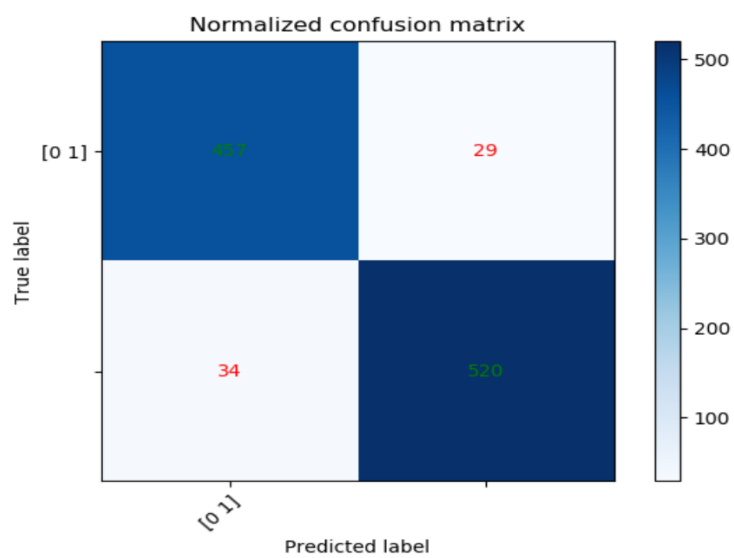
*Accuracy: Train: 98%, Test: 94%*



Loss: Train: 6%, Test: 16%

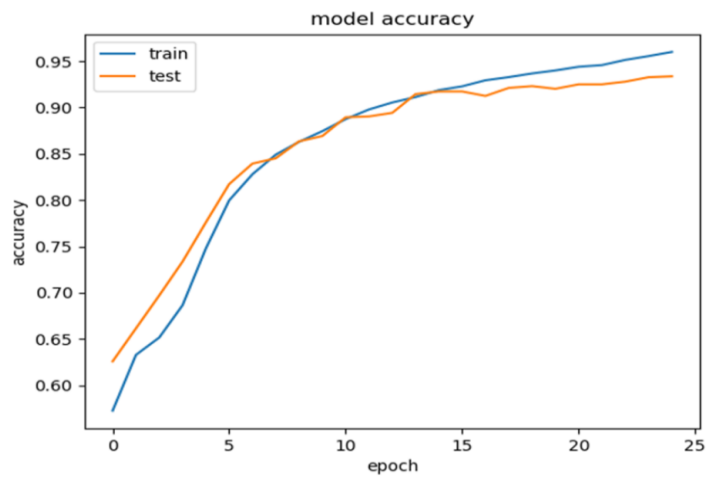


Παρατίθεται και το confusion matrix που δείχνει πως ταξινομήθηκαν τα άρθρα του test set:

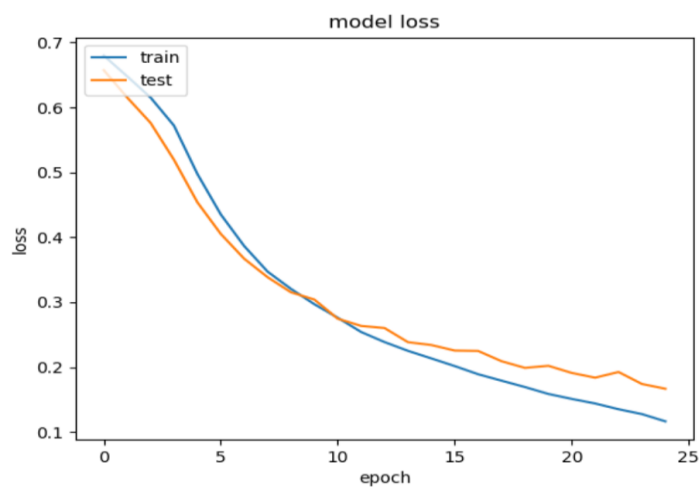


β) Για το μοντέλο που εκπαιδεύεται ταχύτερα (25 epochs)

Accuracy: Train: 96%, Test: 93%

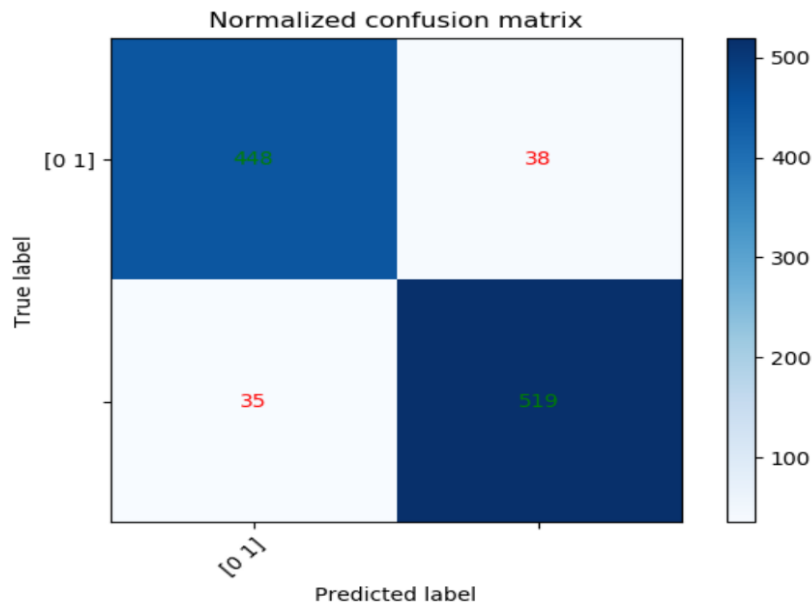


Loss: Train: 11%, Test: 16%



Παρατίθεται και το confusion matrix που δείχνει πως ταξινομήθηκαν τα άρθρα του test set





Ακολουθεί ένας πίνακας με τα αποτελέσματα όλων των μοντέλων που χρησιμοποιήθηκαν για τις μετρικές accuracy, precision, recall και f-measure

Statistic Measure	SVM	Logistic Regression	Multinomial Naive Bayes	Random Forest	Deep Neural Network
Accuracy	0.9526	0.9518	0.9408	0.9047	0.9700
Precision	0.9536	0.9521	0.9450	0.9048	0.9647
Recall	0.9526	0.9518	0.9410	0.9048	0.9756
FMeasure	0.9526	0.9518	0.9407	0.9047	0.9701

## Περιγραφή Εφαρμογής

### Βασικές Έννοιες

Πριν περιγράψουμε τη λειτουργία της εφαρμογής οφείλουμε να περιγράψουμε κάποιες έννοιες που σχετίζονται με αυτή για λόγους πληρότητας

#### JSON FORMAT

Αποτελεί ένα format για την για την αναπαράσταση και την δικτυακή μετάδοση αντικειμένων (JavaScript Object Notation). Βασικοί τύποι δεδομένων που υποστηρίζει:

- 🚦 δεκαδικοί αριθμοί(με προαιρετικό δεκαδικό μέρος ή εκφρασμένοι εκθετικά)
- 🚦 συμβολοσειρές
- 🚦 boolean (true και false)
- 🚦 πίνακες, αντικείμενα (συλλογές ζευγών κλειδιών και τιμών),
- 🚦 null( αναπαράσταση του κενού)

Οι πληροφορίες δίνονται σε μορφή json και θα δούμε τον τρόπο σε επόμενη ενότητα

## RabbitMQ

Παίζει το ρόλο του message broker δηλαδή του ενδιάμεσου για διάφορες υπηρεσίες services. Μειώνουν το φορτίο και το χρόνο εξυπηρέτησης από web εφαρμογές. Τέτοια συστήματα που χρησιμοποιούν Rabbit MQ ακολουθούν τη βασική αρχιτεκτονική της ουράς μηνυμάτων οι εφαρμογές των clients(producers) δημιουργούν μηνύματα, στην περίπτωση μας αυτά έχουν μορφή json και είναι το σώμα κειμένου ενός ή περισσότερων άρθρων, τα οποία παραδίδονται στην ουρά μηνυμάτων του broker στη συνέχεια άλλες εφαρμογές(consumers) συνδέονται στην ουρά και εγγράφονται στα μηνύματα της και τα επεξεργάζονται. Εδώ ουσιαστικά οι consumers παίρνουν τα άρθρα και με βάση το φορτωμένο εκπαιδευμένο μοντέλο κάνουν πρόβλεψη αν τα άρθρα είναι fake ή real. Σημειώνεται ότι κάποιο λογισμικό μπορεί να παίζει το ρόλο του producer ή του consumer ή και τους δύο ρόλους. Τα μηνύματα αποθηκεύονται στην ουρά και παραμένουν αποθηκευμένα εκεί μέχρι να καταναλωθούν.

Οι ουρές μηνυμάτων επιτρέπουν στους web servers να απαντούν στα αιτήματα πολύ γρήγορα αντί να εκτελούν βαριές εργασίες για τους πόρους. Επίσης ένα ακόμη πλεονέκτημα της το οποίο δεν αξιοποιήθηκε εδώ είναι ότι αποδίδει καλά σε περιπτώσεις που ένα μήνυμα πρέπει να διανεμηθεί σε πολλούς παραλήπτες για κατανάλωση ή εξισορρόπηση φορτίων μεταξύ workers. Στην ουρά επιτρέπονται να γίνονται ταυτόχρονα πολλές λειτουργίες δηλαδή είναι εφικτό ένας consumer να παίρνει ένα μήνυμα από την ουρά και να αρχίσει να το επεξεργάζεται ενώ ταυτόχρονα ο producer τοποθετεί νέα μηνύματα στην ουρά. Σημειώνεται ότι producers και consumers μπορεί να βρίσκονται σε εντελώς διαφορετικούς servers. Η δημιουργία και η επεξεργασία μηνυμάτων είναι εφικτό να είναι σε 2 εντελώς διαφορετικές γλώσσες προγραμματισμού . Με τις 2 εφαρμογές να επικοινωνούν μεταξύ του μέσα από τα μηνύματα που ανταλλάσσουν. Εδώ βέβαια και οι δύο εφαρμογές(αυτή που αντλεί τα άρθρα που το web και αυτή που τα επεξεργάζεται και αποφασίζει αν είναι fake είναι και οι δύο υλοποιημένες σε python)

Γενικά τα μηνύματα δεν δημοσιεύονται απευθείας σε μια ουρά αλλά στέλνονται από τον producer σε ένα exchange το οποίο είναι υπεύθυνο για τη δρομολόγηση των μηνυμάτων σε διαφορετικές ουρές (εδώ χρησιμοποιείται για απλότητα μόνο μία). Ένα exchange δέχεται μηνύματα από μια εφαρμογή producer και τα δρομολογεί σε ουρές μηνυμάτων με τη βοήθεια συνδέσμων μεταξύ των ουρών και των exchanges(bindings) και routing keys. Διαφορετικοί τύποι exchanges δέχονται διαφορετικά attributes των μηνυμάτων(π.χ routing keys). Η exchange δρομολογεί τα μηνύματα στις ουρές ανάλογα με τα attributes των μηνυμάτων.

Για την υλοποίηση του request/reply pattern χρησιμοποιήσαμε τον wrapper “Pika” ώστε να διοχετεύουμε μηνύματα στην ουρά στα επιθυμητά κανάλια και από εκεί να προωθούνται στο αντίστοιχο service που θα τα εξυπηρετήσει. Η διαφορά του request/reply pattern από το γνωστότερο publish/subscribe είναι πως στο τελευταίο ο publisher P εκτελεί μια ενέργεια (μέθοδο) στον subscriber S, ενώ στην περίπτωσή μας ο publisher P αφότου εκτελέσει μια ενέργεια στον subscriber S αναμένει ένα response. Είναι σαν να παίζουν αμφότεροι και τους δύο ρόλους περιοδικά.[14]

Συνολικά δημιουργήθηκαν 4 κανάλια επικοινωνίας ώστε να προωθούν μηνύματα στα services που εξυπηρετούν:

- ✚ **Daemon process:** ελέγχει ένα website για νέα άρθρα και μόλις βρει κάποιο νέο το στέλνει στο message broker και στη συνέχεια οι consumers γράφονται στην ουρά για να το επεξεργαστούν και να το καταναλώσουν.
- ✚ **Service** στο οποίο δίνεται ένα άρθρο και το εκπαιδευμένο μοντέλο προβλέπει αν είναι fake ή real.
- ✚ **Batch επεξεργασία άρθρων** .Παρόμοια με την προηγούμενη αλλά για περισσότερα άρθρα
- ✚ Κατέβασμα άρθρων από ένα συγκεκριμένο URL και πρόβλεψη με βάση το μοντέλο αν είναι fake

Προφανώς θα μπορούσαν να χρησιμοποιηθούν λιγότερα κανάλια, αλλά έτσι είναι ευκολότερο να παρατηρήσουμε το workload μας και αν κάποιο κανάλι λαμβάνει μεγάλη κίνηση απλώς δημιουργούμε

περισσότερα services που «ακούνε» σε αυτό. Τότε ο broker θα στέλνει εναλλάξ σε κάθε service άρα το scale είναι αρκετά εύκολο.

Σημείωση: για το κατέβασμα άρθρων από ιστοσελίδες σε real time χρησιμοποιήθηκε το module newspaper της rython που παρέχει τα εργαλεία που με αυτόματο τρόπο κατεβάζει άρθρα από ισότοπους. Πρόκειται ουσιαστικά για ένα web scraper που κατεβάζει άρθρα από σελίδες.

### Βάση Δεδομένων

Όλα τα services που λαμβάνουν άρθρα από εξωτερικές πηγές τα αποθηκεύουν σε μια βάση δεδομένων σε POSTGRE SQL. Κάθε εγγραφή διαθέτει:

- ✚ ένα id που η τιμή του εκχωρείται αυτόματα
- ✚ ο timestamp(ημερομηνία-ώρα) της στιγμής που αποθηκεύτηκε
- ✚ το σώμα του άρθρου
- ✚ Πρόβλεψη δηλαδή αν το άρθρο έχει χαρακτηριστεί fake (1) ή real(0)
- ✚ Truth ratio: ποσοστό αλήθειας περιεχομένου του άρθρου
- ✚ Fake ratio: δείχνει σε τι ποσοστό το άρθρο περιέχει ψευδείς ή παραπλανητικές πληροφορίες

Σε περίπτωση που οι εγγραφές πληθαίνουν και η βάση καταλαμβάνει αρκετό χώρο υπάρχει service που διαγράφει όλα τα δεδομένα της βάσης περιοδικά (μέσω του task scheduler του λειτουργικού). Ο λόγος που αποθηκεύονται οι εγγραφές είναι για πιθανή εκτίμηση της ποιότητας των άρθρων για τα οποία ζητούν πρόβλεψη και για τη δημιουργία μελλοντικού dataset για περαιτέρω εκπαίδευση. Δοκιμαστικά η διαγραφή των άρθρων γίνεται αν οι εγγραφές ξεπεράσουν τις 10000.

### REST API

Αποτελεί ένα interface που ο χρήστης μέσω του browser ή άλλων εφαρμογών εδώ(χρησιμοποιήθηκε το postman) στέλνει HTTP REQUEST{ μεθόδους GET,POST,PUT,DELETE} σε κάποια ιστοσελίδα στην οποία υπάρχει διαθέσιμο ένα web service (εδώ είναι ο μηχανισμός κατεβάσματος και αξιολόγησης περιεχομένου των άρθρων) και επιστρέφει ένα response σε json που λέει αν το άρθρο ή τα άρθρα που αξιολογήθηκαν είναι fake. Ουσιαστικά το Rest Api αποτελεί το μέσο με το οποίο χρήστες και άλλες εφαρμογές μπορούν με την εφαρμογή και να λαμβάνουν προβλέψεις για τα άρθρα που τους ενδιαφέρουν.

### Οδηγίες Λειτουργίας:

Εγκατάσταση όσων αναφέρονται στο τέλος στην ενότητα εγκατάσταση λογισμικού

Από ένα τερματικό εκτελούμε τα 4 αρχεία server\*.py και το αρχείο RestApi.py. απαιτείται η δημιουργία μιας βάσης με όνομα fakenews ώστε το db\_creation.py να δημιουργήσει τον πίνακα που χρειαζόμαστε. Τα αρχεία cleaner.py, client-daemon.py πρέπει να εκτελούνται ως daemon services με τη βοήθεια του λειτουργικού. Σχετικά με τα υπόλοιπα αρχεία client\_rest\* είναι βοηθητικά για τη λειτουργία του Rest API, ενώ τα αρχεία plot\_model.py, store\_data\_plot.py δημιουργούν διαγράμματα για το μοντέλο και τα αποθηκευμένα άρθρα αντίστοιχα. Το test\_politics.py χρησιμοποιείται για να αναλύσει αρχεία από παλιό dataset σε csv. Τέλος τα αρχεία news\_\*, preload\_\* εκπαιδεύουν – αποθηκεύουν το μοντέλο από τα αρχικά datasets και διαβάζουν τα τοπικά αποθηκευμένα βάρη ώστε να προβούν σε προβλέψεις αντίστοιχα. Χρησιμοποιήθηκαν στο στάδιο του tuning και έκτοτε η λειτουργία της εφαρμογής στηρίζεται στο daemon process και στο Rest API.

Το Rest API λαμβάνει τα ακόλουθα ερωτήματα και επιστρέφει τα εξής:

Λειτουργία πρώτη:

Consumes:

```
{  
  "article": "replace this with your article"  
}
```

Response:

```
{  
  "article": n,  
  "real": real_ratio,  
  "fake": fake_ratio,  
  "prediction": p  
}
```

Λειτουργία δεύτερη:

Consumes:

```
{  
  "articles": ["article1", "article2"]  
}
```

Response:

```
{  
  "results": [  
    {  
      "article": "article1",  
      "real": real_ratio,  
      "fake": fake_ratio,  
      "prediction": p  
    },  
    {  
      "article": "article2",  
      "real": real_ratio,  
      "fake": fake_ratio,  
      "prediction": p  
    }  
  ]  
}
```

Λειτουργία Τρίτη:

Consumes:

```
{  
  "url": "replace this with a url",  
  "num": "number of articles (must be integer)"  
}
```

Response (same as above):

```
{  
  "results": [  
    {  
      "article": "article1",  
      "real": real_ratio,  
      "fake": fake_ratio,  
      "prediction": p  
    }  
  ]  
}
```

```






},
{
  "article": "article2",
  "real": real_ratio,
  "fake": fake_ratio,
  "prediction": p
}
]
}

```

Για περισσότερο άρθρα το json response έχει στον πίνακα result περισσότερα στοιχεία.

Καθώς υπήρχε ήδη το πλήρως λειτουργικό περιβάλλον του message broker που αναφέρθηκε παραπάνω, εκμεταλλευτήκαμε αυτό. Κατά συνέπεια το API προωθεί το JSON στον broker για να το επεξεργαστεί το αντίστοιχο service. Όταν το service ολοκληρώσει την πρόβλεψη των άρθρων επιστρέφει το αποτέλεσμα στον broker ώστε να κατευθυνθεί τελικά στο API και να επιστραφεί στο format των παραπάνω responses.

Τέλος υπάρχει μια επιπλέον δυνατότητα η οποία απεικονίζει για εκείνη τη χρονική στιγμή δύο ιστογράμματα για τα άρθρα που υπάρχουν στην βάση δεδομένων. Το ένα ιστόγραμμα απεικονίζει σε ποσοστά πόσα άρθρα είναι fake και πόσα δεν είναι και στο άλλο τα άρθρα χωρίζονται σε 5 κατηγορίες ανάλογα με τα fake ratios:

-  [0,0.2)
-  [0.2,0.4)
-  [0.4,0.6)
-  [0.6,0.8)
-  [0.8,1]

και μετριέται κάθε κατηγορία τι ποσοστό άρθρων έχει σε σχέση με το συνολικό πλήθος άρθρων

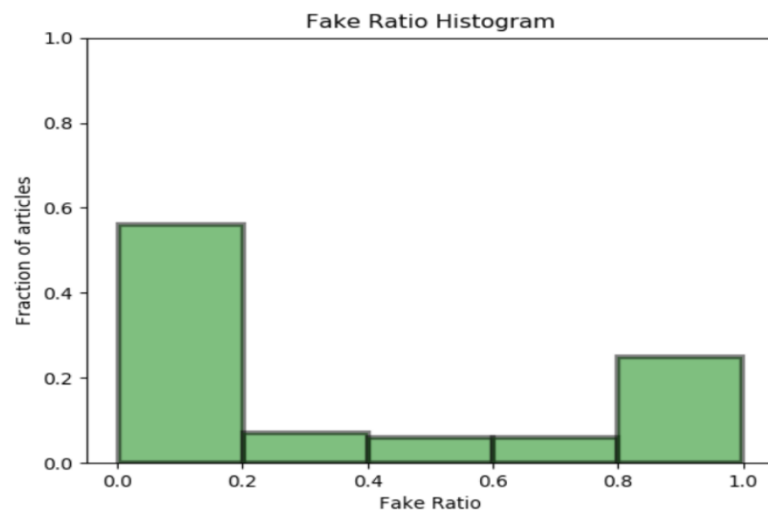
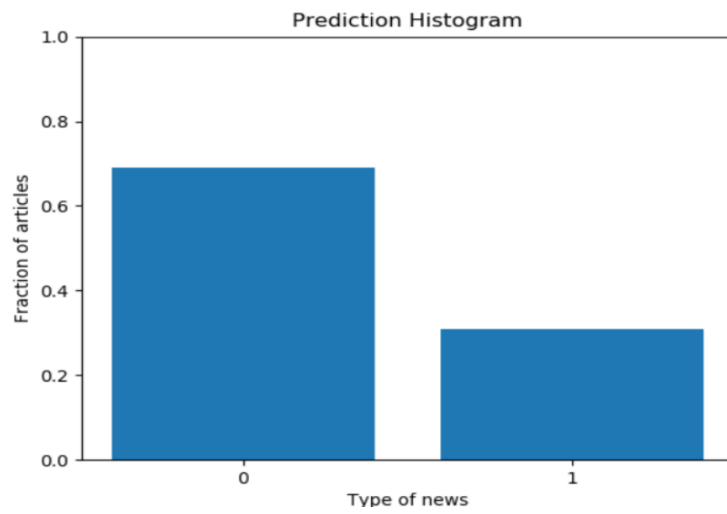
## Προβλέψεις σε real-time

Χρησιμοποιήθηκαν τα services του παραπάνω rest-api για να γίνουν προβλέψεις σε άρθρα διάφορων ιστοσελίδων. Τα αποτελέσματα δεν είναι τα αναμενόμενα γιατί ο web scraper άρθρων κατεβάζει

άρθρα ποικίλης θεματολογίας από τις παραπάνω σελίδες και όχι αποκλειστικά πολιτικά άρθρα όπως αυτά που έχουμε στο dataset. Επίσης το γεγονός ότι ο ταξινομητής δίνει έμφαση στο να μην καταταχτεί κάποιο άρθρο που είναι fake ως real και λιγότερο στην ελαχιστοποίηση των σφαλμάτων ταξινόμησης, επηρεάζει και αυτό τα αποτελέσματα.

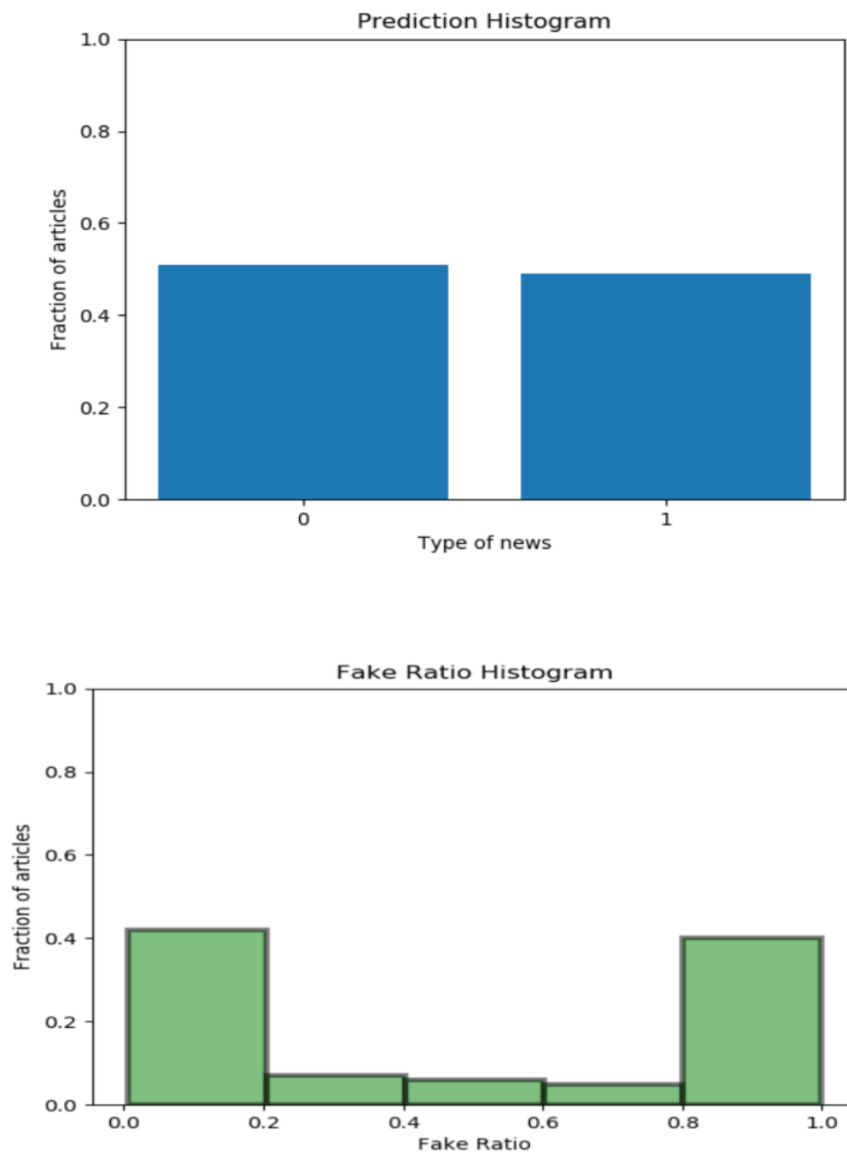
Αποτελέσματα για πραγματικά δεδομένα:

REUTERS (100 άρθρα)



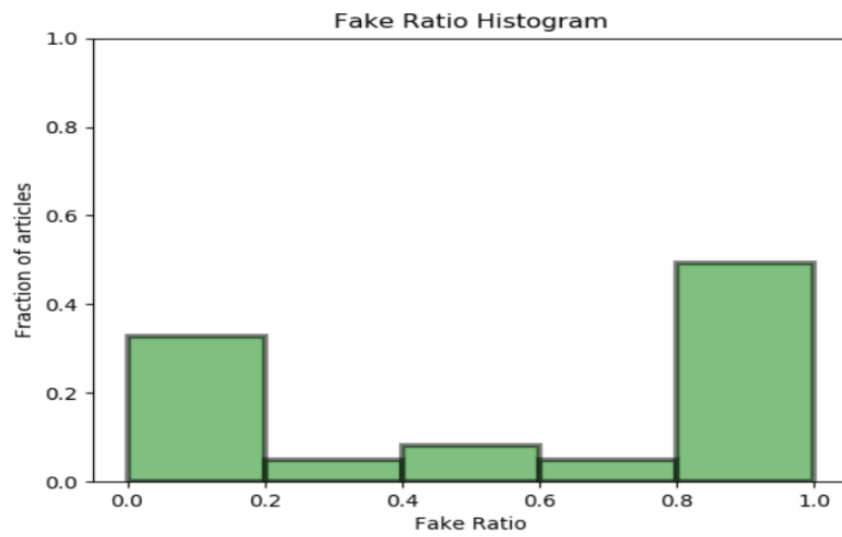
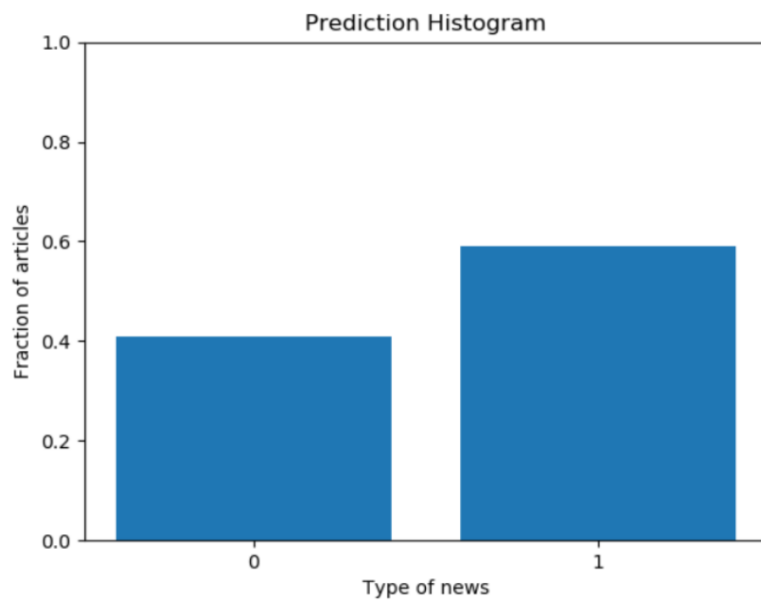


*Fox News (100 άρθρα)*

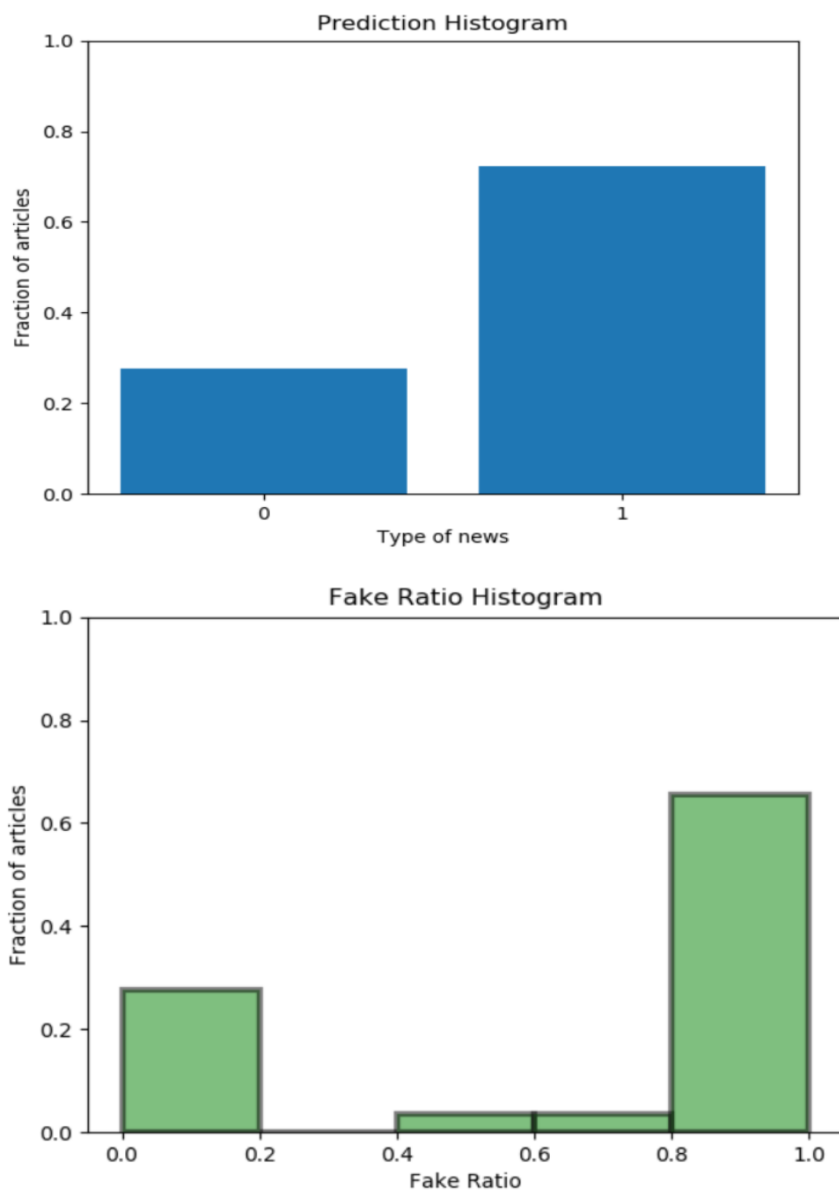


Για μη αξιόπιστες σελίδες με πολλά fake news υπήρχε το πρόβλημα ότι σε πολλές ο scraper άρθρων της βιβλιοθήκης newspaper έβρισκε λίγα άρθρα.

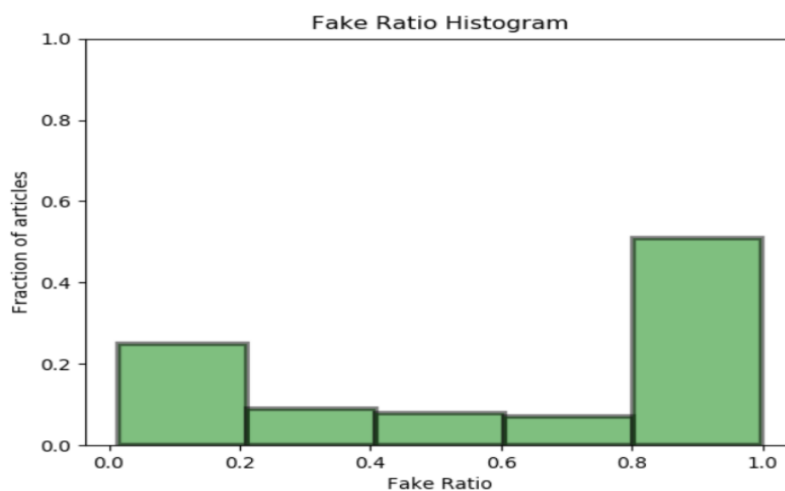
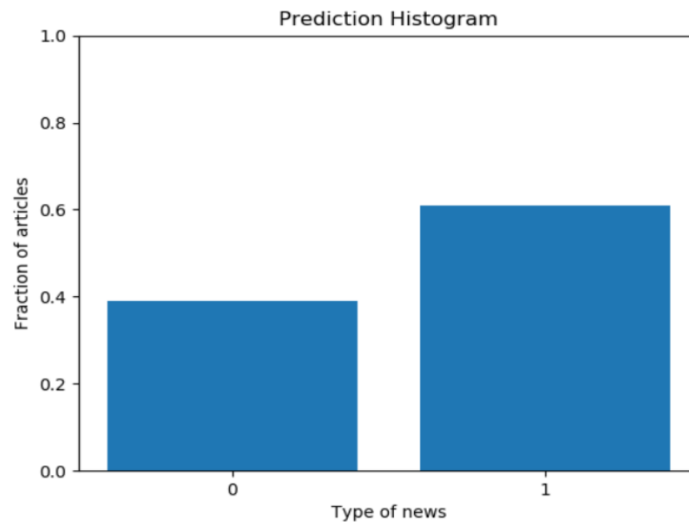
<https://babylonbee.com/> (61 άρθρα)



<https://www.theonion.com/> (29 άρθρα)



<https://www.thedailymash.co.uk/news-100-άρθρα>



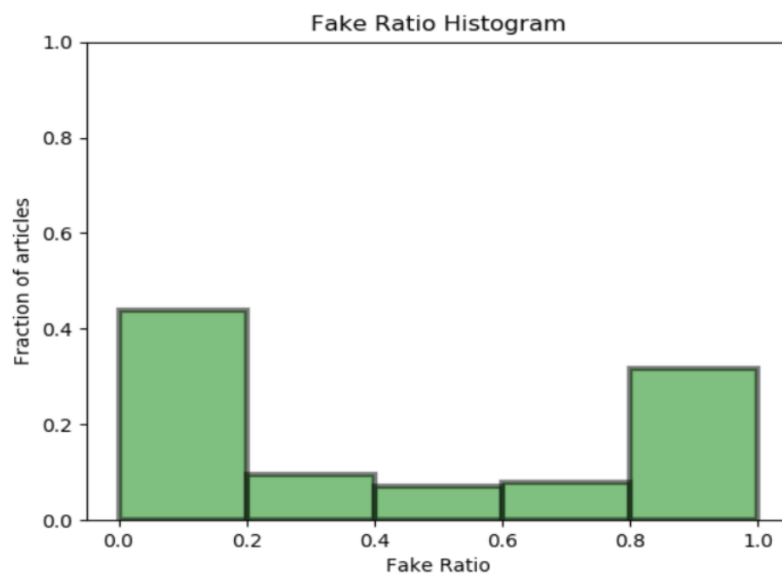
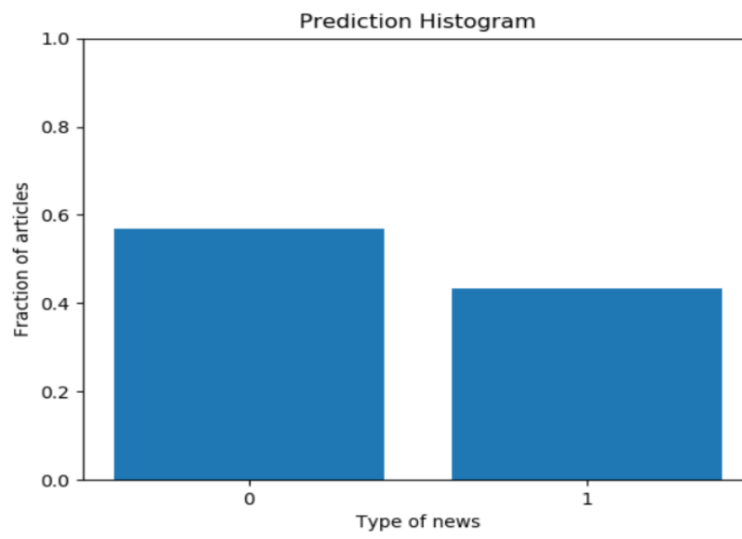
Ενδιαφέρουσες παρατηρήσεις:

- ✚ Καθώς το daemon service κατέβαζε αρκετά άρθρα από έμπιστα δημοσιογραφικά sites παρατηρήθηκε πως αρκετά κατατάχθηκαν ως fake. Παρατηρώντας τα συγκεκριμένα άρθρα, καθώς δεν θα ήταν λογικό να υπάρχουν τόσα ψευδή σε αξιόπιστα sites, αποδείχθηκε ότι αποτελούσαν διαφημίσεις. Ελέγχοντας περαιτέρω αυτήν την υπόθεση το μοντέλο κατατάσσει τη συντριπτική πλειοψηφία των διαφημίσεων ως fake-news.
- ✚ Το training set περιέχει κατά κύριο λόγο πολιτικά άρθρα και επομένως το μοντέλο είναι biased και προβλέπει καλύτερα αυτά τα άρθρα αυτής της κατηγορίας κάτι που φαίνεται και στα ιστογράμματα που υπάρχουν παρακάτω

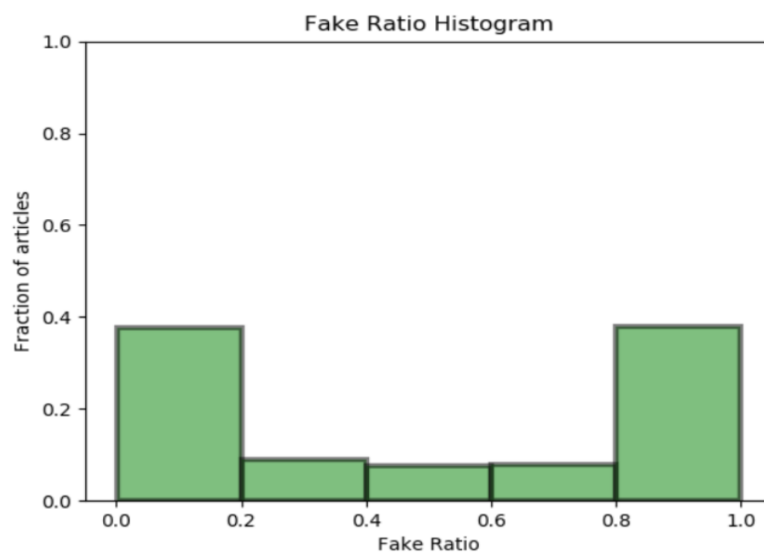
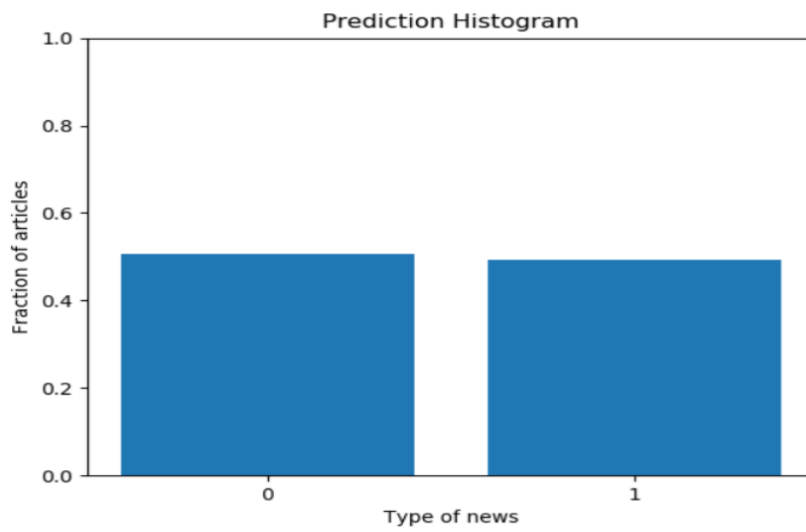
Παρακάτω δείχνουμε πώς δουλεύει το μοντέλο σε διάφορες κατηγορίες άρθρων που είναι σε μεγάλο ποσοστό έγκυρα. Το dataset για αυτά βρίσκεται στο παρακάτω link:

<https://drive.google.com/open?id=108CsyzsuIQvALUFeQZJNMhh8FSxYys6>

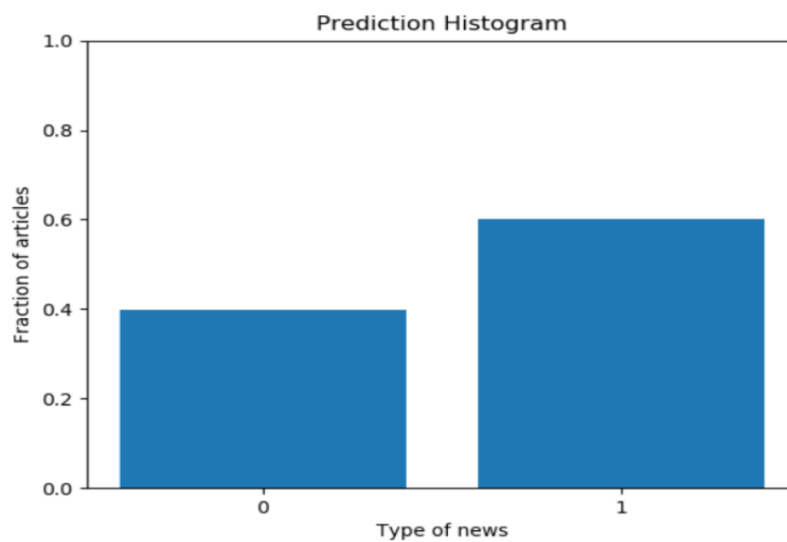
*Politics (2682 άρθρα)*



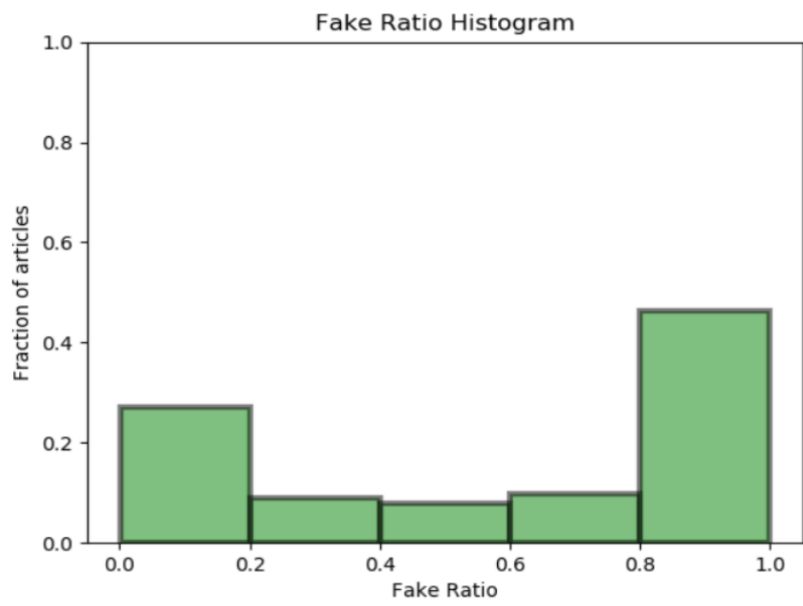
*Business (2735 άρθρα)*



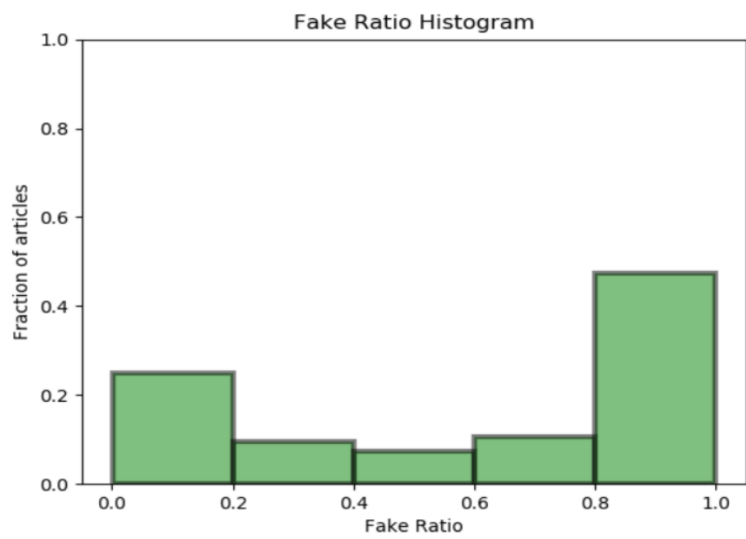
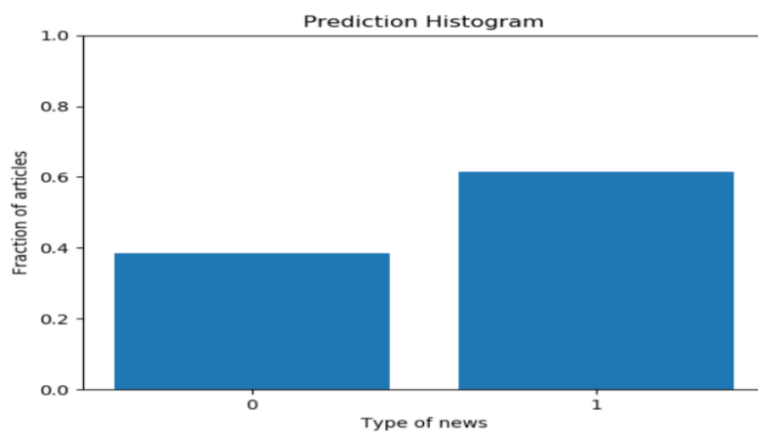
*Football (3120 άρθρα)*



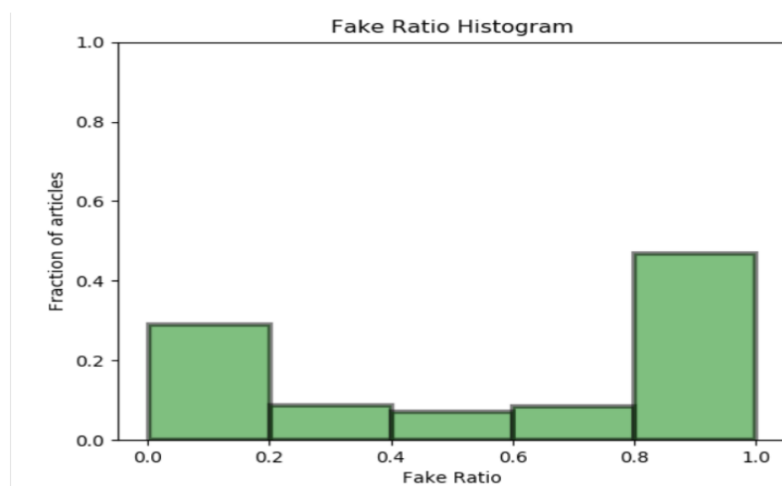
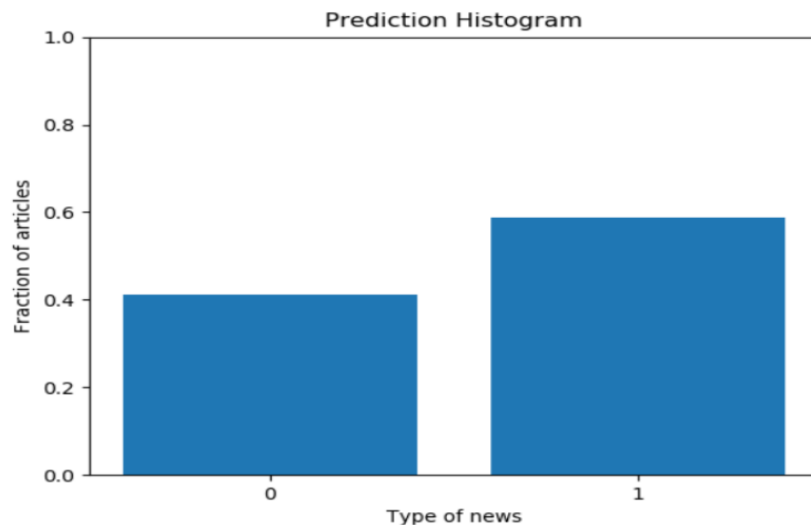




Film (2239 άρθρα)



Technology (1486 άρθρα)



## Εγκατάσταση Λογισμικού

Εγκατάσταση python 3.6 από το site της python:

<https://www.python.org/>

Εγκατάσταση RabbitMQ

Η εγκατάσταση του παραπάνω message broker απαιτεί απλώς τη εγκατάσταση της συναρτησιακής γλώσσας Erlang, έπειτα ο installer του RabbitMQ ολοκληρώνει τη διαδικασία.

<https://www.rabbitmq.com/>

<https://www.erlang.org/downloads>

### Πρόσθετα modules

Η προσπάθεια υλοποίησης του ανιχνευτή ψευδών ειδήσεων γίνεται σε γλώσσα python. Στα πλαίσια της εργασίας χρησιμοποιήθηκαν τα παρακάτω modules . Για να εγκατασταθούν ανοίγουμε command line των windows μεταβαίνουμε στο φάκελο της python και στον υπόφακελο Scripts: cd <path\_python>/Scripts και εκτελούμε τις παρακάτω εντολές:

```
pip install pandas
```

```
pip install -U scikit-learn
```

```
pip install matplotlib
```

```
pip install tensorflow
```

```
pip install keras
```




```
pip install numpy
```

```
pip install psycpg2
```

```
pip install pika
```








```
pip install newspaper3k
```

Σημειώνεται ότι τα παραπάνω δούλεψαν με :

-  Python 3.6
-  tensorflow 2.0.0-alpha0
-  keras 2.2.4

### Προτάσεις για μελέτη στο μέλλον

Στην παρούσα ενότητα γίνονται κάποιες ερευνητικές προτάσεις εργασίας για το μέλλον:

-  Βελτίωση αποτελεσμάτων και ταχύτητας της εκπαίδευση του deep neural network
-  Χρησιμοποίηση περισσότερων στοιχείων(τίτλος,όνομα συγγραφέα,πηγή) στη διαδικασία εκπαίδευσης εκπαίδευση του deep neural network για εντοπισμό των fake news
-  Γενίκευση αλγορίθμου σε άρθρα διαφορετικής κατηγορίας
-  Γενίκευση αλγορίθμου για άρθρα διαφορετικών γλωσσών από την αγγλική
-  Εντοπισμό και κατηγοριοποίηση fake news σε επιμέρους κατηγορίες
-  Βελτίωση εφαρμογή με σχεδιασμό και υλοποίηση ενός user interface που είναι πιο φιλικό στο χρήστη
-  Χρησιμοποίηση ενός πιο καλύτερου και πιο γρήγορου web scraper στην εφαρμογή

## Βιβλιογραφία

- [1] Εισαγωγή στην Εξόρυξη Δεδομένων, Pang Ning Tan-Michal Steinbach-Vipin Kumar, Εκδόσεις Τζιόλας, 5.5-5.5.1, 282-284
- [2] <https://machinelearningmastery.com/logistic-regression-for-machine-learning/>
- [3] <https://www.statisticssolutions.com/what-is-logistic-regression/>
- [4] [https://www.medcalc.org/manual/logistic\\_regression.php](https://www.medcalc.org/manual/logistic_regression.php)
- [5] [https://en.wikipedia.org/wiki/Naive\\_Bayes\\_classifier#Multinomial\\_naive\\_Bayes](https://en.wikipedia.org/wiki/Naive_Bayes_classifier#Multinomial_naive_Bayes)
- [6] ]Εισαγωγή στην Εξόρυξη Δεδομένων, Pang Ning Tan-Michal Steinbach-Vipin Kumar, Εκδόσεις Τζιόλας 5.6.6, 1, 321-323
- [7] <https://arxiv.org/pdf/1301.3781.pdf>
- [8] <https://nlp.stanford.edu/projects/glove/>
- [9] [nlp.stanford.edu/data/glove.6B.zip](https://nlp.stanford.edu/data/glove.6B.zip)
- [10] <https://dumps.wikimedia.org/enwiki/20140102/>
- [11] <https://catalog.ldc.upenn.edu/LDC2011T07>
- [12] <https://nlp.stanford.edu/pubs/glove.pdf>
- [13] <https://arxiv.org/pdf/1412.6980.pdf>
- [14] <https://www.cloudamqp.com/blog/2015-05-18-part1-rabbitmq-for-beginners-what-is-rabbitmq.html>

link για κώδικα εφαρμογής:

[project code](#)