Χαροκόπειο Πανεπιστήμιο Αθηνών

Τμήμα Πληροφορικής και Τηλεματικής

Πτυχιακή Εργασία

Τιτλος

Εξαγωγή Γνώσης Από Δεδομένα Αναπαραγωγής Ειδήσεων

Φοιτητής

ΔΗΜΗΤΡΗΣ ΦΑΣΑΡΑΚΗΣ-HILLIARD

Επιβλέπων

ΗΡΑΚΛΗΣ ΒΑΡΛΑΜΗΣ

Μέλη Επιτροπής

ΔΗΜΗΤΡΙΟΣ ΜΙΧΑΗΛ

ΔΗΜΟΣΘΕΝΗΣ ΑΝΑΓΝΩΣΤΟΠΟΥΛΟΣ

30 Οκτωβρίου 2015

ΣΕΛΙΔΑ ΣΚΟΠΙΜΑ ΚΕΝΗ

ΠΕΡΙΛΗΨΗ

Σκοπός της παρούσας πτυχιακής εργασίας είναι να αναλυθούν δεδομένα που αφορούν την αναπαραγωγή της ίδιας είδησης από πολλαπλές ειδησεογραφικές πηγές με στόχο την εξαγωγή χρήσιμης γνώσης που σχετίζεται α) με τον εντοπισμό πηγών μεγάλης επιρροής, β) τον εντοπισμό ομάδων πηγών (κλικών) που παρουσιάζουν συστηματικά παρόμοιο περιεχόμενο, γ) τον έγκαιρο εντοπισμό ειδήσεων που πρόκειται να συγκεντρώσουν μεγάλο όγκο άρθρων.

Για την ανάλυση κλικών και την εύρεση πηγών με μεγάλη επιρροή χρησιμοποιούμε γραφο-θεωρητικές τεχνικές και αλγορίθμους τις οποίες εφαρμόζουμε σε κατευθυνόμενους ζυγισμένους γράφους, ενώ για την εύρεση ειδήσεων που αναμένεται να έχουν μεγάλη αρθρογραφία χρησιμοποιούνται μαθηματικά μοντέλα εκτίμησης του ρυθμού αύξησης των σχετικών άρθρων.

Πιο συγκεκριμένα, η ανάλυση αρχίζει με τον εντοπισμό πηγών που δημοσιεύουν πρώτες μια είδηση και των πηγών που στην συνέχεια τις αναπαράγουν. Στην συνέχεια μοντελοποιείται η επιρροή των πηγών ανά κατηγορία ειδήσεων με σκοπό την ανακάλυψη της αυθεντίας, από την σκοπιά της επιρροής, κάθε ειδησεογραφικής πηγής ανά θεματική κατηγορία. Τελικά, αφού αναπαρασταθεί γραφικά το σύνολο των δεδομένων γίνεται μια ανάλυση του γράφου (graph) που επιτρέπει τον εντοπισμό ομάδων πηγών που συχνά αναπαράγουν περιεχόμενο μεταξύ τους.

Τέλος, για τον εντοπισμό ειδήσεων με μεγάλο όγκο άρθρων και κατά συνέπεια με μεγάλη σημαντικότητα έγινε μια επιπρόσθετη ανάλυση η οποία βασίστηκε στην μοντελοποίηση του ρυθμού αύξησης μιας είδησης (buzz) και στον εντοπισμό ειδήσεων με τη μεγαλύτερη αύξηση. Η τεχνική αυτή μας επέτρεψε να εντοπίσουμε ειδήσεις οι οποίες τελικά είχαν μεγάλο όγκο άρθρων από τα πρώτα κιόλας λεπτά της εμφάνισής τους και παράλληλα μας επέτρεψε να εκπαιδεύσουν διαφορετικά μοντέλα ανά κατηγορία ειδήσεων.

Λέξεις Κλειδιά: *Επιρροή, Εξαγωγή Γνώσης, Ανάλυση Δεδομένων, Γράφοι, Μηχανική Μάθηση*

ΕΥΧΑΡΙΣΤΙΕΣ

Σύντομες αλλά σημαντικές.

Θα ήθελα να ευχαριστήσω την οικογένεια μου (και την Νούλα!), τους φίλους μου και τους καθηγητές μου για όλη την βοήθεια και υποστήριξη που μου έχουν δείξει τα ακαδημαικά αυτά μου χρόνια.

Επίσης, ευχαριστώ και τον καφέ που πίνω καθώς γράφω αυτό το κείμενο, **ευχαριστώ θερμά**.

ΠΕΡΙΕΧΟΜΕΝΑ

ΛΙΣΤΑ ΕΙΚΟΝΩΝ

ΛΙΣΤΑ ΠΙΝΑΚΩΝ

ΣΕΛΙΔΑ ΣΚΟΠΙΜΑ ΚΕΝΗ

ΚΕΦΑΛΑΙΟ 1

ΕΙΣΑΓΩΓΙΚΕΣ ΕΝΝΟΙΕΣ

Στο κεφάλαιο αυτό παρουσιάζονται οι εισαγωγικές έννοιες που απαιτούνται για την κατανόηση των εννοιών, της διαδικασίας και των αλγορίθμων που χρησιμοποιήθηκαν. Στην ενότητα [1.2] γίνεται μια γρήγορη περιγραφή της έννοιας ανάλυσης δεδομένων (Data Analysis). Στην ενότητα [1.3] γίνεται μια αναφορά στις σχετικές ιδέες της θεωρίας των γράφων (Graph Theory) ενώ στην ενότητα [1.4] υπάρχει μια μικρή περιγραφή του κλάδου της Μηχανικής Μάθησης (Machine Learning). Τελικά, στην ενότητα [1.5] υπάρχουν χρήσιμοι σύνδεσμοι για πιθανούς αναγνώστες που επιθυμούν να διαβάσουν περισσότερα πάνω στο θέμα. Όπως προαναφέρθηκε, οι ενότητες αυτές είναι εισαγωγικές συνεπώς αναγνώστες με γνώση πάνω στο θέμα μπορούν να αγνοήσουν αυτό το κεφάλαιο και να συνεχίσουν στα υπόλοιπα.

1.1 Πρόλογος

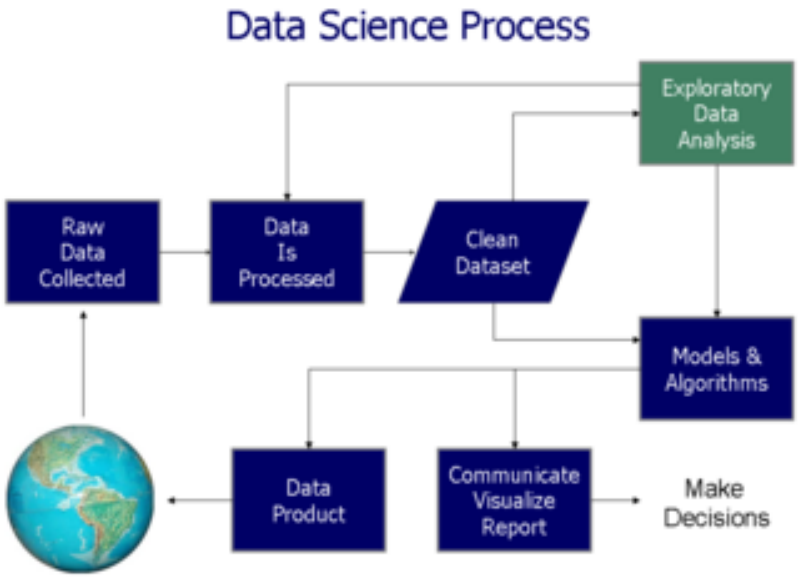
Το κύριο χαρακτηριστικό που ορίζει την εποχή μας είναι η εμφάνιση και κυριαρχία της πληροφορίας (information). Η εποχή της Πληροφορίας είναι ο πλέον de facto όρος που χρησιμοποιείται για να περιγράψει τις εξελίξεις των τελευταίων δεκατιών. Η Πληροφορία, σύμφωνα με τον ορισμό της, είναι αυτό το οποίο πληροφορεί ή αυτό από το οποίο μπορούμε να εξάγουμε γνώση ή δεδομένα (data). Με την επέκταση των ηλεκτρονικών συστημάτων πληροφορίας σε κάθε σκέλος της ζωής μας, ο όγκος της πληροφορίας ο οποίος παράγεται και μεταφέρεται, όχι μόνο είναι πολύ μεγάλος αλλά αυξάνεται και με σταθερά πιο γρήγορους ρυθμούς.

Τα δεδομένα (data) σαν έννοια αφορούν ουσιαστικά την μοντελοποίηση ή κωδικοποίηση

της πληροφορίας/γνώσης με τρόπο που μπορεί να επεξεργαστεί περαιτέρω από συστήματα υπολογιστών. Η κωδικοποίηση της πληροφορίας είναι η γνωστή binary στην οποία οι αριθμοί, οι λέξεις, οι εικόνες κ.α όλες αναπαρίστανται ως ακολουθίες από μηδενικά και άσσους (0 & 1). Μέσω της κωδικοποίησης αυτής η επεξεργασία των δεδομένων, που σκοπό έχει να παράξει καινούργια μή γνωστή πληροφορία/γνώση, μπορεί να γίνει με πολύ μεγαλύτερη ευκολία. Πλέον, η συλλογή των πληροφοριών μπορεί να γίνει με πολύ μεγαλύτερη ευκολία και από διάφορες πηγές. Ενδεικτικά παραδείγματα μπορεί να είναι τα κείμενα που αναρτώνται στην Wikipedia, τα Tweets των χρηστών του Twitter, τα ειδησεογραφικά κείμενα, η τοπολογία του WWW, τα δεδομένα φωνής, εικόνες και πολλά άλλα.

1.2 Ανάλυση Δεδομένων

Τα δεδομένα, στην μορφή που έχουν συλλεχθεί, συνήθως δεν είναι σε θέση να μας δώσουν καινούργια πληροφορία. Η μορφή αυτή, που ονομάζεται **raw data**, απαιτεί επιπλέον βήματα ώστε να μεταλλαχθεί σε μια μορφή ικανή για εξαγωγή συμπερασμάτων. Η Ανάλυση Δεδομένων, μια διαδικασία που ακολουθεί την συλλογή των δεδομένων, αποτελείται από πολλά βήματα, το καθένα με συγκεκριμένο σκοπό, όπως αναπαρίσταται στην εικόνα [1.1].



Εικόνα 1.1: Διάγραμμα Ροής της Επεξεργασίας Δεδομένων. Κάθε στάδιο έχει τις δικές του ξεχωριστές διαδικασίες και μεθοδολογίες.
Πηγή: [Wikipedia]

Τα βήματα και μια μικρή περιγραφή τους ακολουθούν στην επόμενη λίστα:

- **Επεξεργασία Δεδομένων (Data Processing)**

Η μορφή στην οποία βρίσκονται τα δεδομένα μας όταν τα πρωτοσυλλέξουμε συνήθως δεν ενδείκνυται για την εφαρμογή μοντέλων ή αλγορίθμων που θα παράξουν ή θα αποκαλύψουν την κρυμμένη γνώση. Για να έρθουν στην κατάλληλη μορφή συνήθως χρησιμοποιούμε μια ή και περισσότερες διαδικασίες όπως ταξινόμηση (sorting), άθροιση (Summarization), πρόσμιξη (aggregation), με σκοπό την μετατροπή των δεδομένων στην μορφή την οποία χρειαζόμαστε.

- **Καθαρισμός Δεδομένων (Data Cleaning)**

  Μετά το βήμα της επεξεργασίας (ή και πριν) μπορεί να παρατηρηθεί ένα σύνολο ανωμαλιών/σφαλμάτων στα επεξεργασμένα δεδομένα. Επειδή τα συμπεράσματα που εξάγουμε εξαρτώνται πλήρως από την ποιότητα των δεδομένων, για να διασφαλίζουμε πως τα συμπεράσματα θα είναι ορθά, πρέπει να αντιμετωπίσουμε όσο περισσότερο μπορούμε τις ανωμαλίες που παρατηρούνται. Ελλειπή δεδομένα, χαρακτηριστικά που παίρνουν υπερβολικά ακραίες τιμές καθώς και διπλότυπες καταχωρήσεις συνίσταται να αντιμετωπίζονται.

- **Μοντέλα & Αλγόριθμοι (Models & Algorithms)**

  Αφού έχουμε φέρει τα δεδομένα στην μορφή την οποία θέλουμε, αναπτύσσουμε μαθηματικά μοντέλα (i.e αλγόριθμους) που ενεργούν επάνω τους. Τα μοντέλα που αναπτύσσουμε, που εξαρτώνται από τις εκάστοτε απαιτήσεις που έχουμε, μπορούν να είναι από πολύ απλοϊκά (π.χ απλή άθροιση τιμών), έως πιο πολύπλοκα (αλγόριθμους για ανάλυση γράφων, αλγορίθμους για μηχανική μάθηση). Όποια η μορφή τους, τα μοντέλα αυτά χρησιμοποιούνται (σχεδόν) πάντοτε με σκοπό την ανακάλυψη συσχετίσεων μεταξύ μεταβλητών στα δεδομένα μας.

- **Οπτικοποίηση Ευρημάτων (Visualization of Findings)**

  Σαν τελικό στάδιο της επεξεργασίας δεδομένων έχουμε την οπτικοποίηση των αποτελεσμάτων μας. Ουσιαστικά, η οπτικοποίηση είναι μια αφαιρετική αναπαράσταση των συμπερασμάτων σε μια μορφή πιο εύκολα αντιληπτή από τον χρήστη, απλούστερα, είναι μια απεικόνιση που κρύβει την υποκείμενη πολυπλοκότητα των προηγούμενων βημάτων και παρουσιάζει τα αποτελέσματα με συνοπτικό και κατανοητό τρόπο που δεν απαιτεί από τους χρήστες να έχουν τεχνική γνώση για να το καταλάβουν. Συνήθως, η οπτικοποίηση γίνεται με γραφικές μεθόδους όπως pie charts, bar charts αλλά και με γράφους όπως περιγράφονται στην επόμενη ενότητα.

Σε κάθε περίπτωση η ακολουθία αυτή δεν ακολουθείται αυστηρά, όπως επίσης, δεν πραγματοποιείται μόνο μια φορά. Είναι μια επαναληπτική διαδικασία με βήματα τα οποία καθορίζονται

ανάλογα με τις απαιτήσεις που προκύπτουν.

Στην παρούσα εργασία θα επιχειρήσουμε να εξάγουμε γνώση από δεδομένα που συλλέγονται από ειδησεογραφικές πηγές. Οι τεχνικές που θα χρησιμοποιήσουμε εντάσσονται στα ευρύτερα πεδία της ανάλυσης γράφων και της μηχανικής μάθησης. Στις επόμενες δύο υποενότητες αναφέρουμε τις βασικές έννοιες των δύο αυτών πεδίων.

1.3 Θεωρία Γράφων

Πέρα από την ικανότητα τους να οπτικοποιήσουν πληροφορία, οι γράφοι, σαν μαθηματική θεώρηση, είναι ένα πολύ χρήσιμο εργαλείο για την ανάλυση των σχέσεων μεταξύ οντοτήτων. Συνοπτικά, η Θεωρία Γράφων είναι ο μαθηματικός τομέας που ασχολείται με την μοντελοποίηση, ανάλυση και, σε γενικές γραμμές, την μελέτη των γράφων.

Ο γράφος είναι μια μαθηματική δομή που αποτελείται από δύο σύνολα (sets). Το σύνολο των οντοτήτων, οι ***κόμβοι*** του γράφου, συνήθως αντιπροσωπεύουν αντικείμενα του πραγματικού κόσμου. Το σύνολο των συνδέσεων, οι ***ακμές***, μεταξύ των οντοτήτων του γράφου, αντιπροσωπεύουν σχέσεις/συσχετίσεις μεταξύ των αντικειμένων αυτών. Στη πιο συνήθη περίπτωση, κάθε ακμή συνδέει δύο διαφορετικούς κόμβους. Ενδεικτικά, ένα σύνολο κόμβων θα μπορούσε να ήταν οι φοιτητές ενός πανεπιστημιακού ιδρύματος και το σύνολο των ακμών να συνέδεε τους φοιτητές που έχουν δουλέψει μαζί σε μία ομαδική εργασία. Γενικότερα, αυτό που μας προσφέρει αυτή η γραφική απεικόνιση, είναι μια μοντελοποίηση των συσχετίσεων μεταξύ αντικειμένων και την δυνατότητα μαθηματικής ανάλυσης τους για εξαγωγή γνώσης.

Οι κατηγορίες στις οποίες μπορούν να ταξινομηθούν οι γράφοι είναι ποικίλες και πολυπληθείς και κατ'επέκταση, για λόγους χωρητικότητας, μόνο οι σχετικές κατηγορίες θα αναφερθούν εδώ. Συγκεκριμένα, η μορφή των γράφων με την οποία θα ασχοληθεί αυτή η πτυχιακή είναι ***κατευθυνόμενοι*** γράφοι με ***βάρος*** στις ακμές. Οι δύο αυτοί όροι, που αφορούν τις ακμές που υπάρχουν σε έναν γράφο, ορίζονται ως εξής:

- **Κατευθυνόμενοι (Directed)**

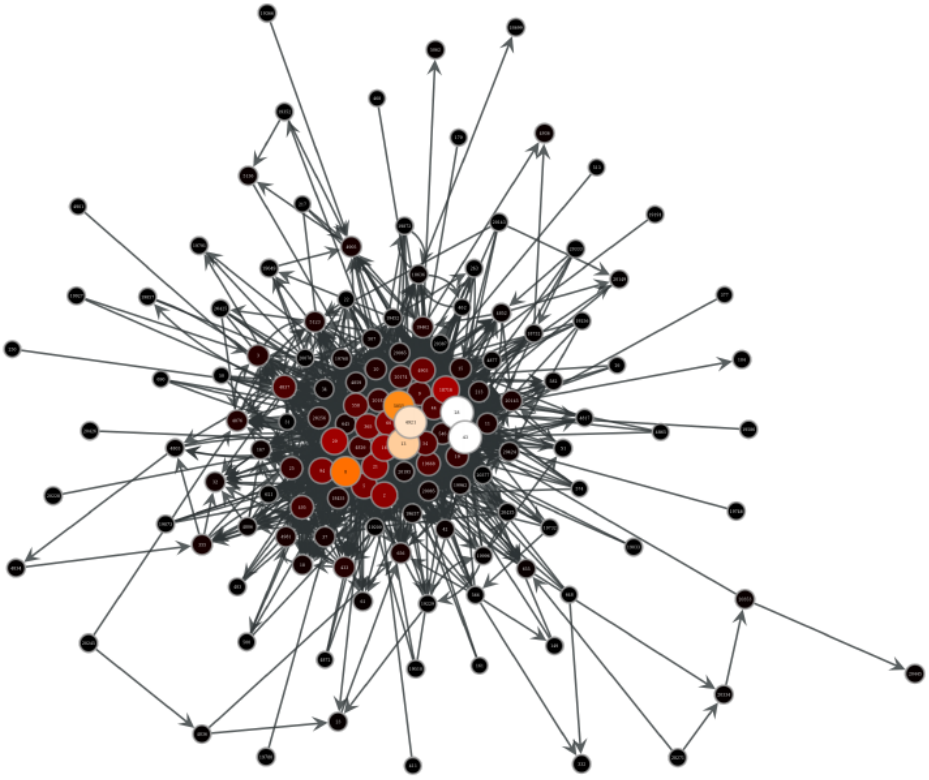
  Όταν λέμε πως ένας γράφος είναι κατευθυνόμενος αυτό που εννοούμε είναι πως κάθε ακμή που υπάρχει μεταξύ δύο κόμβων, πέρα απο την σημασιολογική έννοια που της έχουμε αποδώσει, προσδίδει και κατεύθυνση. Το απλούστερο παράδειγμα ώστε να γίνει κατανοητό είναι αυτό δύο ιστοσελίδων όπου η μία έχει έναν υπερσύνδεσμο που σε κατευθείνει στην άλλη.

Στην περίπτωση αυτή οι σελίδες μοντελοποιούνται ως κόμβοι και μεταξύ τους υπάρχει μια κατευθυνόμενη ακμή που αρχίζει από την σελίδα που περιέχει τον υπερσύνδεσμο και τελειώνει στην σελίδα στην οποία δείχνει ο σύνδεσμος. Η σύμβαση που χρησιμοποιείται στην μοντελοποίηση της κατεύθυνσης είναι μια ακμή βέλος που συνδέει τους δύο κόμβους.

- **Με Βάρος (Weighted)**

  Η παράμετρος αυτή του γράφου, όπως και πριν, αναφέρεται στις ακμές που συνδέουν δύο κόμβους. Το βάρος, αναλόγως πως ορίζεται σε κάθε περίπτωση, προσδίδει βαθμό στην συσχέτιση. Στις περισσότερες περιπτώσεις, μια ακμή με μικρό βάρος μπορεί να εννοεί μικρότερο βαθμό συσχέτισης μεταξύ των οντοτήτων που ενώνει ενώ, αντίστοιχα, μια με μεγάλο βάρος, μεγάλη. Μια άλλη περίπτωση όμως, για παράδειγμα, θα μπορούσε να ήταν το Google Maps όπου ο αρχικός κόμβος είναι η αρχική μας θέση, ο τελικός κόμβος ο προορισμός μας και το βάρος της ακμής η απόσταση μεταξύ τους. Στο παράδειγμα αυτό φαίνεται και πως το βάρος ορίζεται από το θεματικό πλαίσιο, αφού εδώ το μικρότερο βάρος ορίζει μεγαλύτερη σχέση μεταξύ των δύο κόμβων (i.e βρίσκονται πιο κοντά).

Ένα παράδειγμα ενός κατευθυνόμενου γράφου με βάρη (τα βάρη για λόγους παρουσίασης δεν έχουν μοντελοποιηθεί) παρουσιάζεται στην εικόνα [1.2].



Εικόνα 1.2: Παράδειγμα ενός κατευθυνόμενου γράφου με βάρη.

Ο γράφος που θα χρησιμοποιηθεί στην πτυχιακή αυτή έχει ως κόμβους ειδησεογραφικές

πηγές, ενώ οι ακμές που συνδέουν τις πηγές αυτές έχουν κατεύθυνση και βάρος και υποδηλώνουν την κατεύθυνση και το βαθμό επιρροής μεταξύ των πηγών.

1.4 Μηχανική Μάθηση

Η μηχανική μάθηση είναι ένας υποτομέας της Επιστήμης Υπολογιστών που ασχολείται με την μελέτη και κατασκευή αλγορίθμων που χρησιμοποιούνται για την δημιουργία και εκπαίδευση ενός μαθηματικού μοντέλου. Οι αλγόριθμοι αυτοί χτίζουν το μοντέλο χρησιμοποιώντας ως είσοδο τα δεδομένα που έχουμε (αφού πρώτα τα φέρουμε στην σωστή μορφή) κατά την φάση η οποία ονόμαζεται **εκπαίδευση** του μοντέλου. Στην συνέχεια, μέσω ένος παρόμοιου σύνολου δεδομένων με αυτό που εκπαιδεύσαμε το μοντέλο, το εκπαιδευμένο πλέον μοντέλο χρησιμοποιείται για να κάνει προβλέψεις. Οι δύο γενικές κατηγοριοποιήσεις των αλγορίθμων μηχανικής μάθησης αφορούν το τρόπο με τον οποίο μπορεί να εκπαιδευτεί το μοντέλο και το είδος της πρόβλεψης το οποίο θέλουμε να κάνουν.

Ανάλογα με τον τρόπο που μπορεί να εκπαιδευτεί ένα μοντέλο, οι κατηγορίες είναι:

- **Επιτηρούμενη Μάθηση (Supervised Learning)**

  Όπου ο αλγόριθμος κατασκευάζει μια συνάρτηση που απεικονίζει δεδομένες εισόδους σε γνωστές, επιθυμητές εξόδους (σύνολο εκπαίδευσης), με απώτερο στόχο τη γενίκευση της συνάρτησης αυτής και για εισόδους με άγνωστη έξοδο (σύνολο ελέγχου).

- **Μη Επιτηρούμενη Μάθηση (Unsupervised Learning)**

  Όπου ο αλγόριθμος κατασκευάζει ένα μοντέλο για κάποιο σύνολο εισόδων χωρίς να γνωρίζει επιθυμητές εξόδους για το σύνολο εκπαίδευσης.

- **Ενισχυτική Μάθηση (Reinforcement Learning)**

  Όπου ο αλγόριθμος μαθαίνει μια στρατηγική ενεργειών για μια δεδομένη παρατήρηση.

Ανάλογα με τον είδος της πρόβλεψης που θέλουμε να κάνει ένα μοντέλο, οι (κύριες) κατηγορίες είναι:

- **Ταξινόμηση (Classification)**

  Στην ταξινόμηση τα δεδομένα εισόδου στον αλγόριθμο ταξινομούνται σε ένα διακριτό σύνολο επιτρεπόμενων τιμών που ονομάζονται κλάσεις. Το μοντέλο προσπαθεί, δεχόμενο ένα σύνολο δεδομένων που δεν έχει ταξινομηθεί, να ταξινομήσει τα δεδομένα στις κλάσεις που

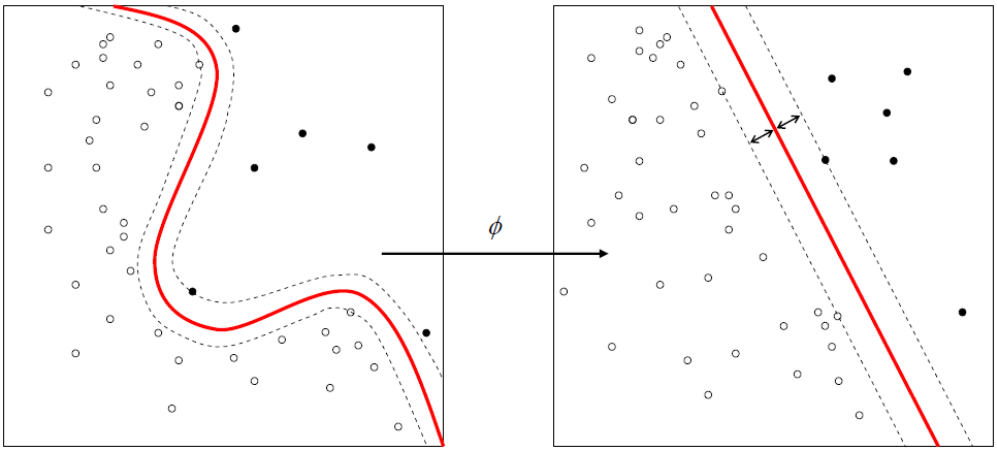
ανήκουν. Η μέθοδος εκπαίδευσης που χρησιμοποιείται με την ταξινόμηση είναι ή **Επιτηρούμενη Μάθηση**.

- **Παλινδρόμηση (Regression)**

  Όπως και στην ταξινόμηση έτσι και στην Παλινδρόμηση, το μοντέλο εκπαιδεύεται χρησιμοποιώντας όμως τώρα δεδομένα τα οποία "ταξινομούνται" σε ένα συνεχές σύνολο επιτρεπόμενων τιμών. Το μοντέλο όπως και πριν, ταξινομεί καινούργια δεδομένα σε τιμές που ανήκουν σε αυτό το σύνολο. Η μέθοδος εκπαίδευσης που χρησιμοποιείται με την Παλινδρόμηση είναι ή **Επιτηρούμενη Μάθηση**.

- **Ομαδοποίηση (Clustering)**

  Στην ομαδοποίηση ένα σύνολο δεδομένων πρέπει να ομαδοποιηθεί σε ομάδες που μοιράζονται κοινά χαρακτηριστικά. Η διαφοροποίηση από τις προηγούμενες κατηγορίες είναι πως στην περίπτωση αυτή οι ομάδες δεν είναι γνωστές εκ των προτέρων και ο αλγόριθμος καλείται να τις συμπεράνει από μόνος του.
  Η μέθοδος εκπαίδευσης που χρησιμοποιείται με την Ομαδοποίηση είναι ή **Μη Επιτηρούμενη Μάθηση**.



Εικόνα 1.3: Ταξινόμηση με SVM (Support Vector Machine). Ένα παράδειγμα αλγόριθμου επιτηρούμενης μάθησης που ταξινομεί άγνωστα δεδομένα αφού πρώτα τα προεπεξεργαστεί με μεθόδους που σκοπό έχουν να αυξήσουν την ακρίβεια πρόβλεψης του μοντέλου.

Πηγή: [Wikipedia]

Στην εικόνα [1.3] φαίνεται ένα γραφικό παράδειγμα του SVM αλγορίθμου που χρησιμοποιείται στην ταξινόμηση. Καθώς η εργασία μας ασχολείται με τις ειδησεογραφικές πηγές και τις ειδήσεις που αυτές δημοσιεύουν, θα εστιάσουμε στην τεχνικής της Ταξινόμησης/Κατηγοριοποίησης. Συγκεκριμένα, θα επιχειρήσουμε την ταξινόμηση άρθρων που βρίσκονται στο διαδίκτυο σε δύο κατηγορίες: 'Breaking', που υποδηλώνει άρθρα μεγάλης σημαντικότητας/έκτακτες ειδήσεις οι οποίες

δημιουργούν ξαφνικά μεγάλη αρθρογραφία, και, Non-Breaking άρθα τα οποία έχουν μικρή σημαντικότητα και κατ' επέκταση "περνούν στα ψιλά" χωρίς να αναπαράγονται και να συζητώνται.

1.5 Σύνδεσμοι

Στην ενότητα αυτή υπάρχουν χρήσιμοι σύνδεσμοι για άτομα που ενδιαφέρονται για περισσότερες εισαγωγικές λεπτομέριες επάνω στα θέματα που παρουσιάστικαν.

1. *Google, your first option for way too many things.*
   Δημοφιλής μηχανή αναζήτησης. Λόγια περιττά.

2. *Introduction to Machine Learning from Andrew Ng [Coursera]*
   Εισαγωγικά μαθήματα μηχανικής μάθησης από τον Andrew Ng του Stanford University. Προσφέρονται μέσω της ηλεκτρονικής πλατφόρμας μάθησης Coursera.

3. *Graph Theory Introductory playlist by Sarada Herke [YouTube]*
   Youtube Video Playlist όπου παρουσιάζονται εισαγωγικές έννοιες γράφων από την Sarada Herke του Πανεπιστημίου Monash.

4. *Data Analysis [Wiki]*
   Σελίδα της Wikipedia που παρουσιάζει τον βασικό κορμό της Ανάλυσης Δεδομένων που παράλληλα περιέχει πολλούς χρήσιμους συνδέσμους σε σχετικές έννοιες

ΣΕΛΙΔΑ ΣΚΟΠΙΜΑ ΚΕΝΗ

ΚΕΦΑΛΑΙΟ 2

ΓΝΩΣΗ, ΔΕΔΟΜΕΝΑ, ΕΡΓΑΛΕΙΑ

Στα πλαίσια της παρούσας εργασίας ο σκοπός μας είναι η "**Εξαγωγή Γνώσης** από **Δεδομένα Αναπαραγωγής Ειδήσεων**". Μιας και οι έννοιες αυτές είναι λίγο γενικές από μόνες τους, μια πιο αναλυτική περιγραφή τους ακολουθεί στις επόμενες ενότητες. Στην ενότητα [2.1] παρουσιάζεται ποία ακριβώς είναι η γνώση που θέλουμε να εξάγουμε από τα δεδομένα μας. Στην συνέχεια, τα δεδομένα που χρησιμοποιήθηκαν περιγράφονται με περισσότερη λεπτομέρεια στην ενότητα [2.2]. Τελικά, στην ενότητα [2.3] υπάρχει μια μικρή περιγραφή των εργαλείων που χρησιμοποιήθηκαν ενώ στην ενότητα [2.4] παρουσιάζονται μερικές συμβάσεις καθώς και η μαθηματική σημειογραφία των μαθηματικών εννοιών που χρησιμοποιούνται.

2.1 Εξαγωγή Γνώσης

Η εξαγωγή γνώσης, όπως προαναφέρθηκε στην εισαγωγή, έχει να κάνει με την επεξεργασία δεδομένων με απώτερο σκοπό την ανακάλυψη καινούργιας, μη γνωστής πληροφορίας. Στα πλαίσια αυτής της πτυχιακής, η γνώση που προσπαθούμε να εξάγουμε από τα δεδομένα που έχουμε συλλέξει είναι:

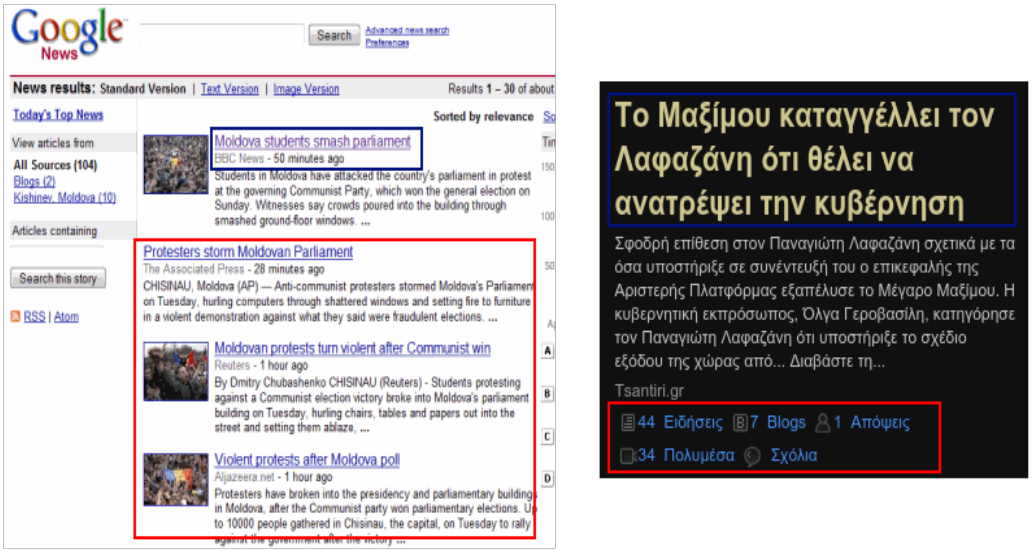
- Ποιες πηγές στο σύνολο δεδομένων έχουν την μεγαλύτερη επιρροή (influence), όπως αυτή ορίζεται, όπως επίσης, ποιες είναι οι πηγές που επηρεάζονται περισσότερο (influenced). Αυτή η πληροφορία πρέπει να εμφανίζεται και στο σύνολο της αλλά και ανά κατηγορία ειδησεογραφικών θεμάτων [2.1.1].

- Τις ομάδες των ειδησεογραφικών πηγών που επηρεάζουν συχνά η μία την άλλη στα πλαίσια της ομάδας τους. [2.1.2].
- Τα χαρακτηριστικά αυτά που όταν, στα πλαίσια της μηχανικής μάθησης, χαρακτηρίζουν μια ομάδα άρθρων, μπορούν να χρησιμοποιηθούν ως παράμετροι για την πρόβλεψη της "σημαντικότητας" μιας είδησης [2.1.3].

2.1.1 Επιρροή Πηγών

Σαν έννοια, η επιρροή υποδηλώνει την ικανότητα ατόμων/πραγμάτων να επηρεάζουν την συμπεριφορά, τις απόψεις ή τις πράξεις άλλων ατόμων/πραγμάτων. Στο ειδησεογραφικό πλαίσιο που μελετήσαμε, η επιρροή εκφράζεται ως η ικανότητα μιας πηγής ειδήσεων να επηρεάσει μια άλλη πηγή με το να την κάνει να αναπαράγει μια είδηση που δημοσίευσε. Ουσιαστικά, λέμε πως μια πηγή έχει επηρεάσει μια άλλη όταν παρατηρούμε την δεύτερη να έχει αναπαράγει το ίδιο άρθρο που δημοσίευσε η πρώτη *αλλά* σε μεθύστερη χρονική στιγμή.

Στην εικόνα [2.1] φαίνονται δύο παραδείγματα τέτοιων ειδήσεων. Όπως φαίνεται, υπάρχει ο τίτλος μιας είδησης [μπλε χρώμα] που έχει προέλθει από μια ειδησεογραφική πηγή, και στη συνέχεια, ένα σύνολο ειδησεογραφικών πηγών που αναμετέδωσαν την ίδια είδηση σε αργότερες χρονικές στιγμές [κόκκινο χρώμα].



Εικόνα 2.1: Εικόνες από Google News (αριστερά) και Palo.gr (δεξιά) που δείχνουν την ομαδοποίηση άρθρων στο διαδίκτυο με βάση το θέμα τους.

Στην αριστερή εικόνα φαίνεται ένα παράδειγμα από το Google News [3] ενώ στην δεξιά από τον

ιστότοπο Palo.gr [6]. Στην περίπτωση του Palo, όπως βλέπουμε έχουμε ένα ομαδοποιημένο σύνολο **44 ειδήσεων** που όλες *μιλάνε για το ίδιο θέμα*. Στο σύνολο αυτό αναγνωρίζουμε την πηγή που *επηρεάζει* από το γεγονός ότι χρονικά δημοσίευσε πρώτη ένα άρθρο που μιλάει γι'αυτό το θέμα. Οι υπόλοιπες 43 πηγές που απομένουν έχουν δημοσιεύσει άρθρα σε αργότερες χρονικές στιγμές και κατ'επέκταση λέμε πως επηρεάστηκαν από την αρχική πηγή. Μέσω αυτής της ομαδοποίησης των ειδησεογραφικών πηγών, μπορούμε να δημιουργήσουμε ένα σύνολο δεδομένων που θα περιέχει, έμμεσα, ακριβώς αυτές τις πληροφορίες.

2.1.2 Ομάδες Πηγών

Στην περίπτωση αυτή, σκοπός είναι να βρεθούν ομάδες ειδησεογραφικών πηγών οι οποίες επηρεάζουν πολύ η μια την άλλη. Ουσιαστικά, αυτό σημαίνει πως σε κάθε ομάδα που εντοπίζεται ανήκουν πηγές, οι οποίες, σαν κύριο χαρακτηριστικό, έχουν μια ισχυρή (από άποψη βάρους) ακμή με κατεύθυνση από και προς άλλες πηγές που βρίσκονται στην ίδια ομάδα. Η ακμή αυτή μπορεί να υποδηλώνει οτιδήποτε στην γενικότερη περίπτωση. Στην δική μας, έχουν αποτυπωθεί πάνω τις αριθμοί που υποδηλώνουν ποσοτικά το μέγεθος της επιρροής. Οι ομάδες θα αποκαλύπτουν τις πηγές αυτές που συχνά αναπαράγουν περιεχόμενο μεταξύ τους.

Όπως υποδηλώνει και η χρήση της λέξης ακμής, η ανακάλυψη των ομάδων από πηγές ανήκει στον τομέα της Θεωρίας Γράφων και ανάγεται στον εντοπισμό "κλικών", δηλαδή ισχυρά συνδεδεμένων υπο-γράφων του συνολικού γράφου. Άρα, αρχικά, πρέπει να δημιουργηθεί ένας γράφος όπου οι κόμβοι του θα μοντελοποιούν τις ειδησεογραφικές πηγές και οι ακμές του την επιρροή μεταξύ των πηγών αυτών. Στην συνέχεια, αλγόριθμοι ομαδοποίησης/τοπολογίας γράφων μπορούν να εφαρμοστούν επάνω στον γράφο για να εντοπίσουν τις ομάδες που υπάρχουν.

2.1.3 Σημαντικότητα Είδησης

Η τελευταία περίπτωση που θέλουμε να εξετάσουμε έχει να κάνει με την "Σημαντικότητα" μιας είδησης/άρθρου. Η σημαντικότητα αναφέρεται στο κατά πόσο μπορούμε να χαρακτηρίσουμε ένα άρθρο που εμφανίζεται στο διαδίκτυο ως έκτακτο ή όχι. Ο χαρακτηρισμός αυτός, γενικά, μπορεί να εξαρτάται από το κατά πόσο αναπαράγουν ένα άρθρο άλλες ειδησεογραφικές πηγές, από το ποίες πηγές το αναπαράγουν ή και από το πόσο γρήγορα αναπαράγεται το άρθρο. Συγκεκριμένα, η γνώση που εξάγεται στην περίπτωση αυτή, είναι το σύνολο των χαρακτηριστικών ενός άρθρου που, όταν χρησιμοποιηθούν ως είσοδο για την εκπαίδευση ενός αλγορίθμου μηχανικής μάθησης, δημιουργούν ένα αξιόπιστο μοντέλο το οποίο μπορεί στην συνέχεια να προβλέψει, με σχετική

ακρίβεια, αν μελλοντικές ειδήσεις έχουν την ιδιότητα του να είναι έκτακτες και μεγάλου ενδιαφέροντος ή όχι. Προφανώς, ο τομέας με τον οποίο ασχολείται το συγκεκριμένο κομμάτι είναι αυτός της Μηχανικής Μάθησης, και συγκεκριμένα, η κατηγορία της ταξινόμησης.

Όπως θα φανεί και στην ενότητα [2.2], λόγω έλλειψής του κειμένου κάθε άρθρου, η διαδικασία αυτή θα γίνει χωρίς να λάβουμε υπόψιν χαρακτηριστικά που πηγάζουν από το κείμενο ή/και τίτλο του άρθρου. Αντιθέτως, τα χαρακτηριστικά εξαρτώνται κυρίως από το σε ποία ομάδα ανήκει ένα άρθρο καθώς και από τους χρόνους δημιουργίας/δημοσίευσης του άρθρου από την αρχική πηγή και αναπαραγωγής του από άλλες ειδησεογραφικές πηγές.

2.2 Περιγραφή Δεδομένων

Τα δεδομένα που χρησιμοποιήθηκαν για την εκπλήρωση των στόχων της εξαγωγής γνώσης που περιγράφηκαν στην προηγούμενη ενότητα προσφέρθηκαν από τον ιστότοπο Palo.gr [6]. Παρείχαν ένα στιγμιότυπο του περιεχόμενου και της κίνησης (από άποψη αναδημοσιεύσεων) των ηλεκτρονικών ειδήσεων για την περίοδο Φεβρουάριος - Μάρτιος 2015. Στο σύνολο τους αποτελούνταν από 239 ειδησεογραφικές πηγές που λειτουργούν κατά βάση στην Ελλάδα και περίπου 179.000 [1] περιπτώσεις αναδημοσίευσης μιας είδησης. Η μορφή τους ήταν σε comma seperated value (.csv) αρχείο, κάνοντας τα βολικά για επεξεργασία.

	A	B	C	D	E	F	G
1	id	url	date	categoryId	siteId	clusterId	count(ac.clusterId)
2	299068103	http://www.mediasoup.gr/node/1017	2015-02-25 00:00:02	80	4877	12425221	1
3	299068167	http://www.newsit.gr/default.php?pna	2015-02-25 00:00:12	5	21	12425216	1
4	299068471	http://www.imerisia.gr/article.asp?cat	2015-02-25 00:01:05	53	43	12425176	1
5	299068527	http://www.sofokleousin.gr/archives/	2015-02-25 00:01:18	1	621	12425238	1
6	299068533	http://www.onsports.gr/Auto-moto/Fc	2015-02-25 00:01:19	61	405	12425239	1
7	299068685	http://www.euro2day.gr/news/highligl	2015-02-25 00:01:50	1	13	12425074	1
8	299068819	http://www.outnow.gr/news/world/ena	2015-02-25 00:02:14	2	20256	12425232	1
9	299068873	http://www.zougla.gr/politiki/article/kr	2015-02-25 00:02:23	30	29	12425074	1
10	299068991	http://www.iefimerida.gr/news/19315	2015-02-25 00:02:37	2	4826	12422353	1
11	299068994	http://www.iefimerida.gr/news/19319	2015-02-25 00:02:38	2	4826	12424837	1
12	299069012	http://news247.gr/eidiseis/athlitika/Sc	2015-02-25 00:02:39	64	2	12421551	1
13	299069057	http://www.zougla.gr/sports/article/pi	2015-02-25 00:02:48	4	29	12421551	1
14	299069098	http://www.newsbomb.gr/sports/podc	2015-02-25 00:02:56	4	365	12423796	1
15	299069245	http://www.music.net.cy/easyconsole	2015-02-25 00:03:23	84	4901	12425228	1
16	299069290	http://www.pontos-news.gr/article/13	2015-02-25 00:03:34	3	18455	12421599	1
17	299069291	http://www.onsports.gr/Spor/Kolymbh	2015-02-25 00:03:35	68	405	12425290	1
18	299069374	http://www.hellasforce.com/blog/%CE	2015-02-25 00:03:43	1	20065	12425288	1
19	299069386	http://www.onsports.gr/Spor/Stibos/it	2015-02-25 00:03:46	66	405	12425287	1
20	299069413	http://www.pontos-news.gr/article/13	2015-02-25 00:03:50	24	18455	12422555	1

Εικόνα 2.2: Ένα snapshot των είκοσι πρώτων καταχωρήσεων άρθρων για τις 25-2-2015. Πηγή δεδομένων: Palo.gr

Ως έχει, το σύνολο δεδομένων περιέχει *ομαδοποιημένες* καταχωρήσεις άρθρων μαζί με τα βασικά χαρακτηριστικά τους. Η ομαδοποίηση των άρθρων με βάση το περιεχόμενο έχει γίνει

[1] Ο ακριβής αριθμός δεν είναι γνωστός εκ των προτέρων μιας και οι καταχωρήσεις υποδηλώνουν είτε δημοσίευση ή αναδημοσίευση.

σε προηγούμενο στάδιο από έναν αλγόριθμο, ο οποίος, χρησιμοποιώντας τεχνικές υπολογισμού ομοιότητας, εξετάζει την ομοιότητα μεταξύ του κύριου κειμένου δύο άρθρων και, στη συνέχεια, υπολογίζει κατά πόσο τα δύο άρθρα υπό εξέταση είναι όμοια, δηλαδή, μιλάνε για την ίδια είδηση. Αν το αποτέλεσμα αυτού του υπολογισμού για τα δύο άρθρα υπερβαίνει ένα προκαθορισμένο threshold, τότε μπαίνουν στην ίδια ομάδα. Το γεγονός αυτό αντιπροσωπεύεται στο αρχείο μας από έναν αριθμό που λειτουργεί ως αναγνωριστικό ομάδας. Τελικά, καταλήγουμε με πολλές ομάδες άρθρων που κάθε μια περιέχει άρθρα τα οποία έχουν μεγάλο βαθμό ομοιότητας μεταξύ τους. Από αυτά, *ένα είναι το αρχικό άρθρο που δημοσίευσε πρώτη μια πηγή ενώ τα υπόλοιπα είναι τα σημασιολογικά όμοια άρθρα που αναδημοσίευσαν άλλες πηγές*.

Τα χαρακτηριστικά/στοιχεία που περιείχε το .csv αρχείο φαίνονται στην εικόνα [2.2]. Παρόλο που δεν χρησιμοποιήθηκαν όλα από αυτά, ακολουθεί μια μικρή περιγραφή του συνόλου τους για λόγους πληρότητας:

- **Identifier:**

  Το identifier λειτουργεί ως το αναγνωριστικό κάθε γραμμής στο αρχείο δεδομένων. Δεν έχει κάποια άλλη σημασιολογική αξία πέρα από το να να ξεχωρίσει κάθε μοναδική καταχώρηση που υπάρχει.

- **URL:**

  Το URL είναι η ηλεκτρονική διεύθυνση κάθε άρθρου στο διαδίκτυο.

- **Date:**

  Το date είναι ένα timestamp που υποδηλώνει την ώρα κατά την οποία μια ειδησεογραφική πηγή δημοσίευσε ή αναδημοσίευσε ένα άρθρο. Το συγκεκριμένο χαρακτηριστικό είναι αρκετά σημαντικό για τις περισσότερες λειτουργίες που θα εκτελεστούν αφού όχι μόνο υποδηλώνει ποίος δημοσίευσε πρώτος αλλά υποδηλώνει και πόσο γρήγορα αναδημοσίευσαν άλλες πηγές το ίδιο άρθρο.

- **Category Identifier:**

  Το Category ID είναι το αναγνωριστικό της κάθε κατηγορίας στην οποία ανήκει ένα άρθρο. Το χαρακτηριστικό αυτό είναι επίσης σημαντικό στα πλαίσια της ανάλυσης μας και συγκεκριμένα στο πλαίσιο της επιρροής. Οι κατηγορίες που υπάρχουν είναι τυπικές και πολυπληθείς, ενδεικτικά παραδείγματα είναι οι κατηγορίες Ελλάδα, Πολιτική, Αθλητικά, Οικονομικά κ.τ.λ.π.

- **Site Identifier:**

  Το αναγνωριστικό κάθε ειδησεογραφικής πηγής. Όπως προαναφέρθηκε, στα δεδομένα μας τυχαίνει να υπάρχουν συνολικά 239 πηγές. Αυτές προφανώς και δεν αντιπροσωπεύουν το σύνολο των Ελληνικών ειδησεογραφικών πηγών που υπάρχουν στο διαδίκτυο αλλά, το σύνολο που υπάρχει στα δεδομένα μας. Το χαρακτηριστικό αυτό είναι επίσης μεγάλης σημασίας.

- **Cluster Identifier:**

  Το αναγνωριστικό κάθε ομάδας (cluster) άρθρων. Άρθρα τα οποία είναι όμοια από άποψη περιεχομένου έχουν παρόμοιο αναγνωριστικό. Όπως και με τα τρία προηγούμενα χαρακτηριστικά, το χαρακτηριστικό αυτό είναι ζωτικής σημασίας.

- **Count (ac.ClusterID):**

  Το άθροισμα των ομάδων στο οποίο ανήκει μια είδηση.

2.3 Βιβλιοθήκες

Στην ενότητα αυτή αναφέρονται όλες οι βιβλιοθήκες που χρησιμοποιήθηκαν για την υλοποίηση των λειτουργιών που απαιτούνταν. Όλος ο κώδικας που γράφηκε υλοποιήθηκε με χρήση της γλώσσας προγραμματισμού Python version 2.7.6 [8]. Η Python ενδείκνυται για γρήγορη ανάπτυξη λόγω της user-friendly σύνταξης που προσφέρει παράλληλα με τις πολλές βιβλιοθήκες που είναι κατάλληλες για ανάλυση δεδομένων, μηχανική μάθηση, επεξεργασία γράφων και μοντελοποίησης αποτελεσμάτων.

2.3.1 Python Libraries

Στην υποενότητα αυτή παρουσιάζονται οι κύριες Python βιβλιοθήκες που χρησιμοποιήθηκαν (βοήθησαν) στην υλοποίηση των απαιτήσεων της πτυχιακής καθώς και παραπομπές στον επίσημο ιστότοπο τους. Built-in libraries της Python δεν θα αναφερθούν για προφανής λόγους συντομίας. Ως την ημερομηνία γραφής αυτού του κειμένου, όλες οι βιβλιοθήκες της επόμενης λίστας είναι open source.

1. **Pandas:** [7]

   Το Pandas είναι μια βιβλιοθήκη που προσφέρει δομές δεδομένων και εργαλεία ανάλυσης δεδομένων. Είναι γραμμένη με τέτοιο τρόπο ώστε να είναι και υψηλής απόδοσης αλλά και εύκολη στην χρήση.

2. **NumPy:** [5]

   Το NumPy είναι η βασική βιβλιοθήκη για επιστημονικούς υπολογισμούς στην Python. Κύρια χαρακτηριστικά του είναι οι πολύ ευέλικτοι και αποδοτικοί πίνακες καθώς και πολλές μέθοδοι γραμμικής άλγεβρας. Πάνω σε αυτήν στηρίζονται αρκετές άλλες βιβλιοθήκες επιστημονικού περιεχομένου όπως Pandas, Scikit-Learn κ.α.

3. **matplotLib:** [4]

   Το matplotlib είναι μια βιβλιοθήκη κατασκευής 2D διαγραμμάτων. Παράγει διαγράμματα υψηλής ποιότητας σε μια ποικιλία από format και δένει όμορφα με βιβλιοθήκες όπως NumPy και Pandas.

4. **Scikit-Learn:** [10]

   Το Scikit-Learn είναι μια από τις βασικές βιβλιοθήκες μηχανικής μάθησης στην Python. Προσφέρει απλά αλλά αποδοτικά εργαλεία για εξόρυξη και ανάλυση δεδομένων. Εξαρτάται από τις βιβλιοθήκες Numpy, SciPy και matplotlib.

5. **Graph-Tool:** [2]

   Το Graph-tool είναι μια αποδοτική Python βιβλιοθήκη για επεξεργασία και στατιστική ανάλυση γράφων. Σε αντίθεση με άλλες βιβλιοθήκες, οι κύριες δομές δεδομένων είναι υλοποιημένες σε C++ βασισμένες στην Boost Graph βιβλιοθήκη. Αυτό δίνει ένα επίπεδο απόδοσης συγκρίσιμο (και από άποψη μνήμης και από άποψη ταχύτητας) με αυτό μιας C/C++ βιβλιοθήκης.

6. **ReportLab:** [9]

   Το Report Lab προσφέρει την δυνατότητα παραγωγής πλούσιων και όμορφων PDF αρχείων σε γρήγορες ταχύτητες μέσω ενός απλού API.

2.4 Σημειογραφία - Συμβάσεις

Στην ενότητα αυτή όσες συμβάσεις χρησιμοποιήθηκαν στην πτυχιακή για τον κώδικα και για τους μαθηματικούς συμβολισμούς παρουσιάζονται.

2.4.1 Κώδικας

Μιας και στα επόμενα κεφάλαια θα γίνονται συχνά αρκετές αναφορές σε κώδικα που χρησιμοποιήθηκε, εδώ παρουσιάζονται μερικές συμβάσεις που υιοθετήθηκαν. Αρχικά, όλα τα snippets κώ-

δικα θα γράφονται inline με την γραμματοσειρά `typewriter` για να ξεχωρίζουν από το υπόλοιπο κείμενο. Δεύτερον, για λόγους συντομίας, θα χρησιμοποιούνται συντομογραφίες για τα ονόματα των βιβλιοθηκών που ορίστηκαν πιο πάνω. Συγκεκριμένα για την Numpy η `np`, για την Pandas η `pnd`, για την matplotlib η `mpl`, για την Scikit-Learn η `skl` και τελικά, για την Graph-Tool η `gt`. Οι συντομογραφίες αυτές θα παρουσιάζονται και στην αρχή των κεφαλαίων στα οποία χρησιμοποιούνται. Τελικά, όσοι ενδιαφέρονται μπορούν να βρουν όλο τον κώδικα που χρησιμοποιήθηκε στην πτυχιακή online στο GitHub [1]

2.4.2 Μαθηματικών

Στην υποενότητα αυτή παρουσιάζεται η σημειογραφία για έννοιες της Ανάλυση Δεδομένων, της Θεωρία Γράφων και της Μηχανικής Μάθησης.

Ανάλυση Δεδομένων

Το σύνολο των κατηγοριών που μπορούν να ανήκουν τα άρθρα συμβολίζεται ως C άρα $C = \{c_1, c_2, \ldots, c_r\}$. Αντίστοιχα, το σύνολο των πηγών στο οποίο ανήκουν οι πηγές συμβολίζεται ως $S = \{S_1, S_2, \ldots, S_l\}$

Θεωρία Γράφων

Ακολουθώντας τυπική σημειογραφία γράφων, κάθε γράφος G αναπαρίσταται ως ο συνδυασμός συνόλων V και E, δηλαδή $G = (V, E)$. Στην δικιά μας περίπτωση έχουμε και μια συνάρτηση βάρους που αντιστοιχεί τιμές στις ακμές του γράφου. Για να το υποδηλώσουμε αυτό ο γράφος γράφεται ως το σύνολο $G = (V, E, w)$

- Το σύνολο $V = \{v_1, v_2, \ldots, v_n\}$ όπου $|V| = n$ ο αριθμός των κόμβων, αντιπροσωπεύει το σύνολο των κόμβων του γράφου G.
- Το σύνολο E αντιπροσωπεύει τις ακμές του γράφου, δηλαδή $E = \{e_1, e_2, \ldots, e_m\}$ όπου $|E| = m$ ο αριθμός των ακμών. Κάθε ακμή e αντιστοιχεί δύο κόμβους ενώ παράλληλα δείχνει και την κατεύθυνση που έχει, δηλαδή $e = (v_t, v_h)$, ακμή με κατεύθυνση από $v_t \longrightarrow v_h$.
- Η συνάρτηση w ενεργεί επάνω στο σύνολο των ακμών και αντιστοιχεί πραγματικούς αριθμούς σε κάθε ακμή, δηλαδή, $w : E \mapsto \mathbb{R}$

Μηχανική Μάθηση

Ακολουθώντας πάλι τυπική σημειογραφία, δύο μαθηματικές έννοιες που θα χρησιμοποιηθούν στα πλαίσια της Μηχανικής Μάθησης είναι η είσοδος που προσφέρουμε στο μοντέλο εκπαίδευσης και η έξοδος που αυτό παράγει.

- *Έξοδος*: Η έξοδος του μοντέλου μας παίρνει τιμές από ένα προκαθορισμένο σύνολο Y το οποίο γράφεται ως $Y = \{Y_1, Y_2, \ldots, Y_i\}$. Οι τιμές του Y_i στην περίπτωση της ταξινόμησης ανήκουν στον σύνολο $\mathbb{R}$.

- *Είσοδος*: Εφόσον η περίπτωση που μελετάμε είναι η επιτηρούμενη μάθηση, ως είσοδο στο μοντέλο εκπαίδευσης χρησιμοποιούμε δύο μεγέθη. Αρχικά, ένα διάνυσμα $\overrightarrow{X} = [X_1, X_2, \ldots, X_j]$ που περιέχει τα χαρακτηριστικά που έχουμε ορίσει και, δεύτερον, μια γνωστή τιμή του συνόλου Y που υποδεικνύει την γνωστή επιθυμητή έξοδο/τιμή που αντιστοιχεί στο διάνυσμα X.

  Άρα, τελικά, η είσοδος είναι της μορφής $\texttt{Input} = [\overrightarrow{X}, Y_i]$

ΣΕΛΙΔΑ ΣΚΟΠΙΜΑ ΚΕΝΗ

ΚΕΦΑΛΑΙΟ 3

ΕΠΙΡΡΟΗ ΚΑΙ ΟΜΑΔΕΣ ΠΗΓΩΝ

Όπως αναφέρθηκε και στο προηγούμενο κεφάλαιο (ενότητες [2.1.1] και [2.1.2]) οι κύριες ερωτήσεις που πρέπει να απαντηθούν είναι:

1. Ποίες πηγές στο σύνολο δεδομένων αλλά και για κάθε (αξιόλογη) κατηγορία έχουν την μεγαλύτερη επιρροή (influence), όπως επίσης, ποίες είναι οι πηγές που επηρεάζονται περισσότερο (influenced);

2. Ποίες είναι οι ομάδες των ειδησεογραφικών πηγών που, στα πλαίσια μιας ομάδας, επηρεάζουν συχνά η μία την άλλη;

Στο κεφάλαιο αυτό, οι απαντήσεις στις ερωτήσεις αυτές καθώς και η μοντελοποίηση των αποτελεσμάτων θα παρουσιαστούν. Αρχίζουμε με την προετοιμασία των δεδομένων που σκοπεύουμε να χρησιμοποιήσουμε στην ενότητα [3.1], συνεχίζουμε, στην ενότητα [3.2], με το να αναφέρουμε τον τρόπο με τον οποίο αυτόματα παράγουμε αναφορά για την επιρροή καθώς και να παρουσιάσουμε διάφορα γραφήματα που σχεδιάστηκαν για την επιρροή. Τελικά, στην ενότητα [3.3], δείχνουμε πως δημιουργήσαμε τον γράφο και πως, μέσω συγκεκριμένων αλγορίθμων, βρήκαμε τις ομάδες αλληλοεπηρεαζόμενων πηγών.

3.1 Προετοιμασμός Δεδομένων

Πριν αρχίσουμε να ψάχνουμε για οτιδήποτε στα δεδομένα μας, πρέπει πρώτα να τα προ-επεξεργαστούμε ώστε να σιγουρευτούμε πως δεν υπάρχουν τυχόν ανωμαλίες σε αυτά, και, στην συνέχεια, να τα επεξεργαστούμε ώστε να τα φέρουμε στην μορφή την οποία χρειαζόμαστε για τα επόμενα στάδια. Οι

κύριες βιβλιοθήκες που χρησιμοποιούμε σε αυτή την ενότητα είναι η `pandas` και η `numpy`. Στις επόμενες παραγράφους, κώδικας που τις χρησιμοποιεί θα αναφέρεται σε αυτές με την συντομογραφία `pnd` και `np`.

3.1.1 Πρώτο Στάδιο Επεξεργασίας Δεδομένων

Οι λειτουργίες που περιγράφονται στην υποενότητα αυτή αντιστοιχούν στην `preprocess_data()` μέθοδο του αρχείου **dataMethods.py**.

Αρχική Εικόνα

Στην μορφή που έχουμε τα δεδομένα [2.2] υπάρχουν στήλες οι οποίες δεν χρειάζονται, καταχωρήσεις οι οποίες τυχαίνει να μην έχουν ομαδοποιηθεί σε κάποιο cluster καθώς και τιμές οι οποίες πρέπει να μετατραπούν σε άλλο τύπο ώστε να επεξεργαστούν πιο αποδοτικά. Η εικόνα [3.1] δείχνει ένα σχετικό παράδειγμα καταχωρήσεων οι οποίες δεν μπήκαν σε κάποια ομάδα (έλλειψη τιμής για *clusterId* και *count* = 0) κατά το αρχικό στάδιο της ομαδοποίησης και ως εκ τούτου δεν μπορούν να χρησιμοποιηθούν.

	A	B	C	D	E	F	G
1	id	url	date	categoryId	siteId	clusterId	count(ac.clusterId)
2	299069002	http://www.novasports.gr/	2015-02-25 00:02:39	58	433		0
3	299069454	http://www.lifo.gr/lifoland/s	2015-02-25 00:03:58	5	507		0
4	299069475	http://www.sentragoal.gr/r	2015-02-25 00:04:01	4	43		0
5	299069476	http://www.sentragoal.gr/r	2015-02-25 00:04:01	4	43		0

Εικόνα 3.1: Ένα snapshot πέντε καταχωρήσεων άρθρων που δεν έχουν ομαδοποιηθεί σε κάποιο cluster και κατ'επέκταση δεν πρέπει να συμπεριληφθούν με τα υπόλοιπα δεδομένα τα οποία είναι ομαδοποιημένα. Πρόκειται είτε για άρθρα χωρίς περιεχόμενο, είτε για άρθρα που αφορούν ειδήσεις που δεν αναπαράχθηκαν από άλλες πηγές.

Επιλογή Στηλών

Σαν αρχικό στάδιο επεξεργασίας, πρέπει, λαμβάνοντας υπόψιν τις απαιτήσεις μας, να διαλέξουμε τις στήλες τις οποίες σκοπεύουμε να χρησιμοποιήσουμε. Για τους δικούς μας σκοπούς οι στήλες που χρησιμοποιήθηκαν ήταν οι:

- **date:** Η ημερομηνία δημοσίευσης/αναδημοσίευσης.
- **categoryId:** Το αναγνωριστικό της κατηγορίας που ανήκουν.
- **siteId:** Το αναγνωριστικό της πηγής.
- **clusterId:** Το αναγνωριστικό του cluster που ανήκουν.

Οι στήλες *id*, *url* και *count(ac.clusterId)* δεν είχαν καμία απολύτως χρήση στα δικά μας πλαίσια και κατ'επέκταση δεν χρησιμοποιήθηκαν.

Dataframe Objects and Filtering

Αφού έχουμε διαλέξει τις στήλες τις οποίες θα χρησιμοποιήσουμε μπορούμε να αρχίσουμε να επεξεργαζόμαστε τα δεδομένα μας. Το πρώτο πράγμα που κάνουμε είναι να φορτώσουμε το .csv αρχείο που έχουμε στη pandas βιβλιοθήκη μέσω της μεθόδου `pnd.read_csv(kwargs**)`. Η μέθοδος αυτή μετατρέπει το αρχείο που έχουμε από την .csv μορφή του σε ένα `pandas dataframe object` που επιτρέπει sql-like ερωτήσεις επάνω στις στήλες του. Παράλληλα, μέσω κατάλληλων παραμέτρων που προσφέρουν επιπρόσθετες λειτουργίες κατά την φόρτωση του αρχείου, μπορέσαμε αρχικά, να επιλέξουμε τις στήλες που προαναφέρθηκαν όπως επίσης και να ταξινομήσουμε τις γραμμές του dataframe σε αύξουσα σειρά ημερομηνίας. Τελικά, στο σύστημα μας είχαμε ένα dataframe αντικείμενο που σαν περιεχόμενα είχε τα περιεχόμενα των τεσσάρων στηλών ταξινομημένα ανάλογα με την ημερομηνία τους.

Επόμενο βήμα ήταν η αντιμετώπιση του αρχικού προβλήματος των καταχωρήσεων που δεν είχαν ομαδοποιηθεί σωστά σε clusters. Σαν λύση, για κάθε καταχώρηση (κάθε γραμμή) που είχαμε στο dataframe μας, ελέγξαμε κατά πόσο η τιμή της στήλης *clusterid* ήταν έγκυρος (not NaN), δηλαδή, η τιμή της ήταν ένας finite αριθμός. Σαν διαδικασία, αυτό ήταν εύκολο να πραγματοποιηθεί με το να περάσουμε την μέθοδο `np.isfinite()` σαν column based function filter στο dataframe μας. Σαν column filter, ενεργώντας πάνω σε κάθε τιμή της στήλης *clusterId*, η μέθοδος αυτή διέγραφε κάθε καταχώρηση της οποίας η τιμή δεν ήταν έγκυρος αριθμός.

Adding Additional Columns

Στην συνέχεια, έγινε αντιληπτό πως οι τιμές για το timestamp κάθε καταχώρησης δεν ήταν σε μορφή κατάλληλη για επεξεργασία [1]. Συγκεκριμένα, εμείς χρειαζόμασταν να μετατρέψουμε τις ημερομηνίες σε δευτερόλεπτα ώστε να είναι πιο εύκολα στην επακόλουθη επεξεργασία. Για να το κάνουμε αυτό γράψαμε μια συνάρτηση που σαν παράμετρο παίρνει την στήλη *date* σαν πίνακα και μετέτρεπε τις τιμές του από την αρχική τους μορφή σε Unix Time (seconds from 1970). Στην συνέχεια, επέστρεφε τις καινούργιες τιμές του πίνακα και εμείς απλά τις αναθέταμε σε μια καινούργια στήλη που ονομάσαμε *date_in_seconds*.

Από την στιγμή που ένα από τα θέματα με τα οποία διαπραγματεύεται η πτυχιακή είναι η επιρροή, χρειάστηκε με κάποιον τρόπο να κρατάμε και μια τιμή γι'αυτήν στο dataframe με το οποίο δουλεύαμε. Η λύση η οποία βρέθηκε ήταν η δημιουργία μιας ακόμη στήλης που θα κρατούσε

[1] Τα timestamps και γενικότερα η επεξεργασία αντικειμένων που αναπαριστούν ημερομηνίες είναι θέμα μεγάλο και ως εκ τούτου δεν δύναται να αναλυθεί περισσότερο εδώ.

την τιμή για την επιρροή. Η στήλη αυτή, με όνομα *influence*, είχε value = 1 για κάθε καταχώρηση στα δεδομένα μας μιας και κάθε καταχώρηση, στην πλειοψηφία των περιπτώσεων τουλάχιστον, υποδείκνυε μια αναδημοσίευση ενός άρθρου και άρα 1 περίπτωση επιρροής μέσω αναδημοσίευσης. Βεβαίως, επειδή η προσθήκη στήλης είναι column wise operation, δηλαδή μια λειτουργία που εκτελείται για όλες τις τιμές της επιλεγμένης στήλης, η τιμή αυτή ανατέθηκε **και** για καταχωρήσεις που υποδείκνυαν την αρχική δημοσίευση ενός άρθρου. Για να αντιμετωπίσουμε το γεγονός αυτό, στην συνέχεια όταν εντοπίζαμε τις αρχικές πηγές για κάθε cluster, διαγράφαμε την τιμή αυτή.

GroupBy - Cluster Processing

Σαν επόμενο βήμα, έπρεπε να εκτελέσουμε μερικές λειτουργίες ανάλογα με την ομάδα στην οποία άνηκαν οι καταχωρήσεις. Για κάθε ομάδα, δηλαδή για κάθε ξεχωριστό *clusterId*, έπρεπε να βρούμε ποία ήταν η πηγή η οποία έκανε την αρχική δημοσίευση του άρθρου, ποίες οι πηγές που την αναδημοσίευσαν, πόσο χρόνο κάνανε να την αναδημοσιεύσουν όπως επίσης και σε ποία κατηγορία άνηκαν τα άρθρα. Άρα, έπρεπε να ομαδοποιήσουμε τις καταχωρήσεις στο dataframe ανάλογα με το *clusterId* που είχαν και να εκτελέσουμε λειτουργίες πάνω σε αυτές τις ομάδες. Ευτυχώς για εμάς, το dataframe object το οποίο χειριζόμαστε έχει ορισμένη μια μέθοδο που κάνει ακριβώς αυτό το πράγμα. Η `dataframe.groupby(kwargs**)` μέθοδος ενεργεί επάνω στο dataframe και ομαδοποιεί δεδομένα σύμφωνα με την στήλη που της δίνουμε ως παράμετρο. Καλώντας την μέθοδο αυτή και χρησιμοποιώντας σαν παράμετρο την στήλη *clusterId* μας επιστρέφεται ένα groupby αντικείμενο. Λειτουργεί σαν hash function που σαν κλειδί έχει το *clusterId* και σαν value τις καταχωρήσεις που ανήκουν σε αυτό το cluster (σε μορφή dataframe).

Με το groupBy αντικείμενο στα χέρια μας, μπορούμε να αρχίσουμε να επεξεργαζόμαστε κάθε ομάδα με το να κάνουμε ένα iteration των κλειδιών του. Στα πλαίσια κάθε ομάδας, εφόσον έχουμε ήδη ταξινομήσει τις καταχωρήσεις ανάλογα με την ημερομηνία τους, το πρώτο στοιχείο, δηλαδή αυτό με την πιο παλιά ημερομηνία, θα αντιπροσωπεύει την **αρχική πηγή** που δημοσίευσε το άρθρο πρώτη. Γι'αυτό το στοιχείο κρατάμε τις τιμές των στηλών *date_in_seconds, date, categoryId, siteId* σε μεταβλητές για μετέπειτα χρήση. Αφού κρατήσουμε τα χαρακτηριστικά αυτά, **διαγράφουμε** την καταχώρηση αυτή από τα δεδομένα μας, λύνοντας και το πρόβλημα που αναφέραμε νωρίτερα.

Στη συνέχεια, εφόσον βρήκαμε την αρχική πηγή της ομάδας και εφόσον ξέρουμε πως **υπάρχει μόνο μια αρχική πηγή**, εύκολα συμπεραίνουμε πως κάθε επακόλουθη καταχώρηση θα υποδηλώνει μια αναδημοσίευση του άρθρου της αρχικής πηγής. Έτσι, για κάθε καταχώρηση που έχει απομείνει, η διαδικασία που πραγματοποιείται είναι η εξής:

- Για λόγους συνέπειας, αναθέτουμε την τιμή της κατηγορίας της αρχικής πηγής σαν τιμή για την κατηγορία των υπόλοιπων καταχωρήσεων. Αυτό για να αποφύγουμε τις περιπτώσεις όπου μερικά ειδησεογραφικά site αλλάζουν την κατηγορία ενός άρθρου για δικούς τους εσωτερικούς λόγους.

- Φτιάχνουμε μια καινούργια στήλη με όνομα *original_source_id* για κάθε καταχώρηση που σαν τιμή κρατάει το id της αρχικής πηγής.

- Για σκοπούς που θα γίνουν ξεκάθαροι στο επόμενο κεφάλαιο, δημιουργούμε μια ακόμη στήλη με όνομα *article_num* που ουσιαστικά είναι η απαρίθμηση κάθε άρθρου στα πλαίσια του cluster. Άρα, για παράδειγμα, το 5ο άρθρο θα έχει article_num = 5, το 6ο ίσο με 6 κ.ο.κ.

- Σαν τελευταία πράξη δημιουργούμε μια καινούργια στήλη με όνομα *delay* την οποία υπολογίζουμε με το να αφαιρέσουμε την τιμή του *date_in_seconds* για κάθε καταχώρηση από την ίδια τιμή, που είχαμε κρατήσει, για την αρχική πηγή.

Αφού τελειώσει αυτό το στάδιο, αρχικά, ενώνουμε όλα τα groupBy αντικείμενα σε ένα data frame χρησιμοποιώντας την μέθοδο `pnd.concat()`. Τελικά, επειδή υπήρξε η ανάγκη ομαλοποίησης (unity based normalization) των τιμών της στήλης *delay*, γράψαμε μια συνάρτηση 3.1 που έπαιρνε ως είσοδο την τιμή της στήλης *delay* για κάθε καταχώρηση και ομαλοποιούσε τις τιμές της στο διάστημα $[0, 1]$.

$$w_i = 1 - \frac{D_i - D_{min}}{D_{max} - D_{min}} \tag{3.1}$$

Στην συνάρτηση αυτή το i υποδηλώνει ποία καταχώρηση χρησιμοποιούμε. Η μεταβλητή w_i υποδηλώνει την ομαλοποιημένη τιμή για κάθε καταχώρηση , η μεταβλητή D_i το delay της, ενώ οι D_{max}, D_{min} είναι σταθερές, υποδηλώνοντας το μεγαλύτερο και μικρότερο delay που παρατηρήθεικε συνολικά στα δεδομένα μας αντίστοιχα. Την καινούργια στήλη με τις ομαλοποιημένες τιμές την ονομάσαμε *quickness* αφού ουσιαστικά υποδηλώνει πόσο γρήγορα, μετά από την δημοσίευση του άρθρου, την αναδημοσίευσε η επηρεασμένη πηγή.

Τελικά, κλείνουμε το πρώτο στάδιο της επεξεργασίας με το να διαγράψουμε τις στήλες *date, date_in_seconds* οι οποίες πλέον δεν έχουν λόγο ύπαρξης (εκτός ίσως από το να πιάνουν χώρο). Για να μην χρειαστεί να ξανα-επεξεργαστούμε το ίδιο αρχείο μελλοντικά, κυρίως γιατί για μεγαλύτερα αρχεία ο χρόνος επεξεργασίας είναι αισθητός, αποθηκεύουμε το data frame ως .csv πάλι χρησιμοποιώντας την μέθοδο `dataframe.to_csv()`.

Ενδιάμεση μορφή Δεδομένων

Στην εικόνα [3.2] φαίνεται η (ενδιάμεση) μορφή των δεδομένων μας όπως είναι μετά το πρώτο στάδιο επεξεργασίας. Οι στήλες περιέχουν τις τιμές όπως περιγράφηκαν στις προηγούμενες παραγράφους.

	categoryId	siteId	clusterId	influence	original_sourc	delay	article_num	quickness
176909	64	32	12570615	1	37	12	1	0.9999910346
176910	64	355	12570615	1	37	37	2	0.9999723567
176908	3	4903	12570530	1	18716	580	1	0.9995666726
176907	24	4903	12570525	1	44	563	1	0.9995793736
176906	4	4859	12570516	1	16	530	1	0.9996040285
176904	3	16	12570440	1	18716	1161	1	0.9991325982
176905	3	4859	12570440	1	18716	1245	2	0.9990698404
176903	24	19668	12570432	1	18716	582	1	0.9995651784
176902	24	29	12570429	1	5055	1204	1	0.9991004722
176901	64	2	12570428	1	25	5	1	0.9999962644
176899	3	13	12570404	1	42	1685	1	0.9987411093
176900	3	4859	12570404	1	42	1711	2	0.9987216843
176898	24	28	12570400	1	20085	170	1	0.9998729903
176897	30	18636	12570383	1	20174	151	1	0.9998871855
176896	30	19668	12570371	1	4921	1955	1	0.998539388
176893	3	558	12570343	1	19	156	1	0.9998834499
176894	3	9	12570343	1	19	984	2	0.9992648377
176895	3	13	12570343	1	19	999	3	0.999253631
176891	3	60	12570338	1	13	2575	1	0.998076176

Εικόνα 3.2: Ένα snapshot είκοσι καταχωρήσεων άρθρων όπως είναι η μορφή τους μετά το πρώτο στάδιο επεξεργασίας δεδομένων.

3.1.2 Δεύτερο Στάδιο Επεξεργασίας Δεδομένων

Οι λειτουργίες που περιγράφονται στην υποενότητα αυτή αντιστοιχούν στην `group_stats()` μέθοδο του αρχείου **dataMethods.py**.

Επίπεδα Δεδομένων

Σαν δεύτερο στάδιο επεξεργασίας ορίζουμε το στάδιο εκείνο στο οποίο βρίσκουμε την επιρροή μιας ειδησεογραφικής πηγής για κάθε περίπτωση. Η επιρροή αυτή θα είναι αθροιστική για κάθε πηγή, με βάση την επιμέρους επιρροή που έχουν τα άρθρα που δημοσιεύει η πηγή. Πιο συγκεκριμένα όταν έχουμε μια σειρά από άρθρα, προερχόμενα από διαφορετικές πηγές, που όλα αφορούν την ίδια είδηση και συνεπώς έχουν ανατεθεί στην ίδια συστάδα, μόνο η πρώτη πηγή που δημοσίευσε το άρθρο θα συγκεντρώσει όλη την επιρροή εντός της συστάδας. Φυσικά σε μια άλλη είδηση, μια άλλη πηγή θα δημοσιεύσει πρώτη και θα την ακολουθήσουν οι υπόλοιπες ενισχύοντας την επιρροή της κ.ο.κ. Σε γενικές γραμμές, τα δεδομένα που έχουμε μπορούν να επεξεργαστούν σε τρία επίπεδα:

- **Cluster Level:** Το επίπεδο αυτό εστιάζει στην εύρεση επιρροής μιας πηγής εντός ενός cluster. Μέσω αυτού μπορούμε να βρούμε π.χ ποίο άρθρο (στην ουσία ποια πηγή) είχε την μεγαλύ-

τερη επιρροή συνολικά για όλα τα clusters.

- **Category Level:** Αντίστοιχα, το επίπεδο αυτό εστιάζει στις κατηγορίες. Μέσω αυτού μπορούμε να βρούμε ένα από τα ζητούμενα που είναι η επιρροή μιας πηγής αθροιστικά για όλα τα clusters μιας κατηγορίας. Φυσικά το επίπεδο αυτό βασίζεται στα αποτελέσματα του προηγούμενου επιπέδου.
- **Overall Level:** Τελικά, το επίπεδο αυτό μας δίνει την εικόνα της επιρροής για το σύνολο των δεδομένων χωρίς ουσιαστικούς περιορισμούς. Μέσω αυτού μπορούμε να βρούμε ποια πηγή είχε τη μεγαλύτερη ή μικρότερη επιρροή συνολικά στην ειδησεογραφία.

Τα επίπεδα αυτά είναι σημαντικό κομμάτι και της τωρινής αλλά και της επακόλουθης ανάλυσης. Το επόμενο στάδιο της επεξεργασία που θα ακολουθήσει έχει την ίδια λογική για κάθε επίπεδο που ορίστηκε, με την μόνη διαφορά να βρίσκεται στις στήλες τις οποίες κρατάμε για κάθε περίπτωση. Όπως και πριν, θα συνεχίσουμε και εδώ να χειριζόμαστε ένα dataframe αντικείμενο. Στην περίπτωση αυτή βέβαια το αντικείμενο αυτό έχει την μορφή που παρουσιάστηκε στην τελευταία υποενότητα του [3.1.1].

Columns ανά Επίπεδο

Στην επόμενη ενότητα η μεταβλητή $cols$ αλλάζει ανάλογα με το επίπεδο στο οποίο βρισκόμαστε. Για τα επίπεδα που έχουμε ορίσει η τιμή της είναι:

- **Cluster Level:** $cols = [$*categoryId, original_source_id, siteId, clusterId* $]$
- **Category Level:** $cols = [$*categoryId, original_source_id, siteId* $]$
- **Overall Level:** $cols = [$*original_source_id, siteId* $]$

GroupBy (again) and Aggregate

Αξιοποιώντας περαιτέρω την δύναμη των GroupBy, σε αυτή την υποενότητα την χρησιμοποιούμε σε συνδυασμό με την εξίσου δυνατή Aggregate [2] συνάρτηση για να επιτύχουμε τον σκοπό μας. **Ασχέτως επιπέδου, η λογική είναι παρόμοια**: κάνουμε GroupBy με βάση τα $cols$ και στην συνέχεια aggregate τις τιμές των στηλών που δεν ανήκουν στο $cols$ χρησιμοποιώντας μια συνάρτηση σαν παράμετρο (συνήθως αθροιστική). Αυτή η διαδικασία γίνεται και για το *influence* αλλά και για

[2] Με λίγα λόγια, το Aggregate απλά κάνει collapse τις καταχωρήσεις κάθε group σε μια καταχώρηση. Όσες στήλες ανήκουν στο GroupBy είναι εξ'ορισμού όμοιες άρα δεν επηρεάζονται. Για όσες δεν βρίσκονται, ορίζουμε εμείς μια συνάρτηση που ορίζει τι γίνεται με τις τιμές τους στη διαδικασία του collapse.

το *quickness* κάθε cluster. Άρα, ουσιαστικά, στην δικιά μας περίπτωση, αυτό που κάνει η groupBy είναι να δημιουργήσει μια συσχέτιση μεταξύ των πηγών που ανήκουν στο S, σε διαφορετικά επίπεδα κάθε φορά. Σε κάθε περίπτωση, μετά από το groupBy εμείς έχουμε ένα influenced set $S_i = S_k$ για κάθε πηγή στο S, όπου το S_k αναφέρεται σε **μία** πηγή η οποία *έχει επηρεαστεί περισσότερες των μια φορών στα πλαίσια του ίδιου cluster, category ή γενικά* (αναλόγως επιπέδου). Δηλαδή το S_k είναι και αυτό ένα σύνολο της μορφής $S_k = \{S_{k1}, S_{k2}, \ldots, S_{kn}\}$. Παράλληλα, λόγω της στήλης *influence* και *quickness* έχουμε και τις τιμές $I(S_k),\ Q(S_k)$ της επιρροής και του quickness για την πηγή αυτή.

Επειδή το Aggregate δεν μπορεί να εφαρμόσει δύο διαφορετικές συναρτήσεις καθώς κάνει collapse τις καταχωρήσεις και διότι εμείς χρειαζόμαστε να εφαρμόσουμε μια αθροιστική συνάρτηση στην επιρροή και μια συνάρτηση που υπολογίζει τον μέσο για το *quickness*, απαιτείται να σπάσουμε το dataframe μας σε δύο. Έτσι, δημιουργούμε το dataframe_influence που κρατάει την επιρροή και το dataframe_quickness που κρατάει το *quickness*. Αυτά τα δύο dataframes δημιουργούνται με το να κάνουμε drop τις στήλες *quickness* και *influence* αντίστοιχα για κάθε περίπτωση, μέσω της μεθόδου `dataframe.drop([column_names])`.

Αφού έχουμε τα dataframes για κάθε περίπτωση, χρησιμοποιούμε και πάλι την μέθοδο groupBy για να ομαδοποιήσουμε τα δεδομένα που έχουν όμοιες τιμές στις στήλες του $cols$. Τώρα μπορεί να γίνει προφανής η επιλογή των στηλών ανά επίπεδο. Εμείς θέλουμε κυρίως την τιμή του *influence* για καταχωρήσεις που έχουν όμοιο *original_source_id* και *siteId*. Με το να συμπεριλαμβάνουμε περισσότερες στήλες στην μεταβλητή $cols$ κάνουμε το groupBy πιο συγκεκριμένο αφού περισσότερες στήλες πρέπει πλέον να έχουν όμοιες τιμές ώστε να κάνουν match. Βάζοντας τις κατάλληλες στήλες για κάθε περίπτωση, μπορούμε να δημιουργήσουμε και τα ανάλογα groups για κάθε επίπεδο.

Έχοντας τα groups για κάθε περίπτωση, μπορούμε τώρα να εφαρμόσουμε το aggregate για να κάνουμε collapse όλες τις καταχωρήσεις σε μία. Το aggregate λειτουργεί πάνω σε groupBy αντικείμενα δεχόμενο ως παράμετρο μια συνάρτηση που καθορίζει το τι θα κάνει με τις τιμές των στηλών που θα γίνουν collapsed. Για την περίπτωση του dataframe_influence, αφού προφανώς έχουμε κάνει το groupBy, καλούμε την aggregate με παράμετρο την αθροιστική συνάρτηση `np.sum` με κλήση της `influence_groupBy.agg(np.sum)`. Αυτή θα πάρει το σύνολο όλων των τιμών των καταχωρήσεων για την στήλη *influence* και θα τις αθροίσει για το τελικό αποτέλεσμα.

$$I(S_i) = \sum_{i=1}^{n} I(S_{ki}) \ , \quad \forall\ S_i \in S \tag{3.2}$$

Για την περίπτωση του dataframe_quickness από την άλλη, καλούμε πάλι την aggregate αλλά αυτή τη φορά χρησιμοποιώντας ώς συνάρτηση την `np.mean` που υπολογίζει τον μέσο όρο των τιμών της στήλης quickness.

$$Q(S_i) = \frac{\sum_{i=1}^{n} Q(S_{ki})}{n}, \quad \forall\ S_i \in S \tag{3.3}$$

Τώρα απομένουν δυο τελευταίες λειτουργίες πριν τελειώσει και αυτό το στάδιο. Πρώτον, πρέπει να ενώσουμε τα δυο dataframes σε ένα μοναδικό dataframe και, δεύτερον, να κρατήσουμε τα δεδομένα αυτά για μετέπειτα χρήση. Για την πρώτη λειτουργία, η λύση είναι απλή, δημιουργούμε μια καινούργια στήλη σε ένα από τα dataframes (δεν παίζει ρόλο σε ποιο διαλέξουμε) και αναθέτουμε την στήλη του άλλου στην καινούργια αυτή στήλη. Για την δεύτερη, εφόσον το κομμάτι αυτό της επεξεργασίας εκτελείται σχεδόν άμεσα, δεν κρίθηκε απαραίτητο να αποθηκευτεί σε κάποιο αρχείο. Αντιθέτως απλά φτιάξαμε ένα `dict` object στην Python και για κάθε επίπεδο αποθηκεύαμε το dataframe αυτού του επιπέδου.

Τελική Μορφή Δεδομένων

Επιτέλους το στάδιο της επεξεργασίας τελείωσε. Στην εικόνα [3.3] φαίνονται τα δεδομένα στην τελική τους μορφή, για κάθε επίπεδο, όπως αυτό ορίστηκε.

	categoryId	original_sour	siteId	clusterId	influence	quickness
2	40	15	20602	12540656	205	0.8742021893
3	80	15	20602	12518722	173	0.9198153595
4	1	20	5055	12513458	137	0.8240869914
5	1	20	645	12513458	115	0.8364794848
6	1	20	21	12513458	90	0.8402521268
7	2	19668	20602	12512829	87	0.9561843268
8	1	20	4837	12513458	80	0.8352272914
9	1	20	365	12513458	75	0.8508849939
10	40	15	20602	12501287	74	0.9780886891

	categoryId	original_sour	siteId	influence	quickness
2	64	25	2	955	0.9994840214
3	1	13	9	899	0.9994974008
4	2	4826	20195	693	0.9970212209
5	64	2	25	635	0.999571241
6	1	13	558	526	0.996637254
7	64	8	405	478	0.9934828469
8	64	8	4859	473	0.9935743803
9	3	13	4859	425	0.9924902802
10	40	15	20602	388	0.9919313316

	original_source_id	siteId	influence	quickness
2	25	2	1249	0.9996850037
3	13	9	1041	0.993065351
4	4826	20195	953	0.9861444285
5	13	558	857	0.9938425289
6	2	25	849	0.9990338337
7	16	4859	774	0.9950999497
8	9	558	681	0.9949776482
9	15	20602	580	0.9913719072
10	43	13	529	0.9870408807

Εικόνα 3.3: Ένα snapshot δέκα aggregated καταχωρήσεων για κάθε ένα από τα επίπεδα που ορίστικαν, όπως παράγονται μετά το δεύτερο στάδιο επεξεργασίας. Στην πάνω εικόνα φαίνονται τα αποτελέσματα για το επίπεδο των clusters, στην κάτω αριστερή για το επίπεδο των κατηγοριών ενώ στην κάτω δεξιά εικόνα φαίνεται το αποτέλεσμα συνολικά.

Εύκολα παρατηρείται πως μερικά από τα ερωτήματα που είχαμε μπορούν να απαντηθούν άμεσα, για άλλα, απαιτούνται μερικά "queries" πάνω στο dataframe. Ανεξαρτήτως περιπτώσεως, τα απο-

τελέσματα απαιτείται να παρουσιαστούν σε μια πιο user friendly μορφή με την χρήση γραφικών μεθόδων οπτικοποίησης.

3.2 Αυτόματη Παραγωγή Αναφοράς

Στην ενότητα αυτή, παρουσιάζουμε μερικά από τα αποτελέσματα της ανάλυσης για την επιρροή, αλλά, μέσω μιας αυτοματοποιημένης διαδικασίας που δημιουργήθηκε. Η αυτοματοποιημένη παραγωγή αναφοράς, που γράφηκε με την βοήθεια του **reportLab**, απλά δημιουργεί ένα pdf document και στην συνέχεια, μέσω άλλων διαδικασιών, παράγει bar charts για τις πηγές που υπάρχουν στα δεδομένα μας. Παράλληλα, έχει γραφεί με τέτοιο τρόπο ώστε να μας δίνεται η δυνατότητα να προσδιορίσουμε διάφορες παραμέτρους. Στις επόμενες υποενότητες θα δείξουμε πως λειτουργεί αυτό, τα plots που χρησιμοποιεί και τις δυνατότητες που προσφέρουν οι παράμετροι του. Ένα sample pdf βρίσκεται στο GitHub στο repository του κώδικα. Όλες οι λειτουργίες είναι υλοποιημένες στο python package **reporter** του φακέλου **src/breakingNews/**.

3.2.1 Δημιουργία Plots

Σκοπός της αναφοράς αυτής, είναι να παρουσιάσει τα αποτελέσματα της επιρροής μεταξύ διαφορετικών πηγών με έναν αυτοματοποιημένο τρόπο. Άρα ένα από τα αρχικά θέματα που πρέπει να εξετάσουμε είναι τι είδους πληροφορία θα παρουσιάζει η αναφορά και πως θα την παρουσιάζει. Η εύκολη επιλογή, για τον τρόπο παρουσίασης των αποτελεσμάτων, έγινε με το να διαλέξουμε την μορφή των bar charts για τα διαγράμματα που θα δημιουργήσουμε. Αυτό, κυρίως γιατί είναι τα πιο εύκολα και κατανοητά διαγράμματα που υπάρχουν. Η επιλογή για το τι είδους πληροφορία θα παρουσιάζεται για κάθε πηγή στα δεδομένα μας ήταν λίγο πιο δύσκολη αλλά, τελικά, αποφασίσαμε να δείχνουμε την εξής πληροφορία:

- Για κάθε πηγή, θα δείχνουμε τις *NUM_OF_CATEGORIES_TO_PLOT* [3] κατηγορίες στις οποίες επηρέασε τις περισσότερες πηγές.
- Για κάθε πηγή και για κάθε κατηγορία, όπως αυτές ορίζονται από το προηγούμενο bullet, θα δείχνουμε τις *NUM_OF_SOURCES_TO_PLOT* πηγές που επηρεάστηκαν περισσότερο από την πηγή αυτή στις κατηγορίες αυτές.

Στις υποενότητες που ακολουθούν, οι κύριες μέθοδοι που χρησιμοποιήθηκαν ήταν οι `find_leaders()` του πακέτου `bN` και η `charts()` του πακέτου `reporter`. Όπως και πριν, η λογική και για τις δυο

[3] Όλες οι παράμετροι για την αυτόματη παραγωγή αναφοράς βρίσκονται στο αρχείο reportConf.py στο reporter πακέτο.

περιπτώσεις είναι αρκετά όμοια και η διαφορά έγκειται σε παραμέτρους που περνάμε. Η μορφή των δεδομένων που χρησιμοποιείται είναι όπως φαίνεται στην εικόνα [3.3].

Συνολική Επιρροή ανά Κυρίαρχη Κατηγορία

Μιας και το να βρούμε την επιρροή για κάθε κατηγορία που υπάρχει είναι όχι μόνο παράλογο από άποψη πόρων που καταναλώνουμε αλλά και επειδή πολλές πηγές επηρεάζουν μόνο σε συγκεκριμένες κατηγορίες, εμείς, προσπαθήσαμε να βρούμε τις κύριες κατηγορίες στις οποίες μια πηγή ασκεί τη μεγαλύτερη επιρροή. Ο αριθμός των κατηγοριών αυτών, που ονομάστηκαν **κυρίαρχες κατηγορίες**, ορίστηκε να είναι έξι (6), αριθμός αρκετά κατατοπιστικός.

Από τη στιγμή που έχουμε τα δεδομένα επεξεργασμένα με τον τρόπο που ορίσαμε στην προηγούμενη ενότητα, το να βρούμε τις κατηγορίες είναι μια διαδικασία πολύ πιο απλουστευμένη. Εφόσον ψάχνουμε κατηγορίες το επίπεδο των δεδομένων, όπως αυτό ορίστηκε, που θα χρησιμοποιήσουμε είναι το Category Level. Στο αρχείο αυτό, όπως φαίνεται και από την εικόνα [3.3], έχουμε συσχετίσεις πηγών ανά κατηγορία με την συνολική ποσότητα επιρροής που παρατηρήθηκε μεταξύ τους.

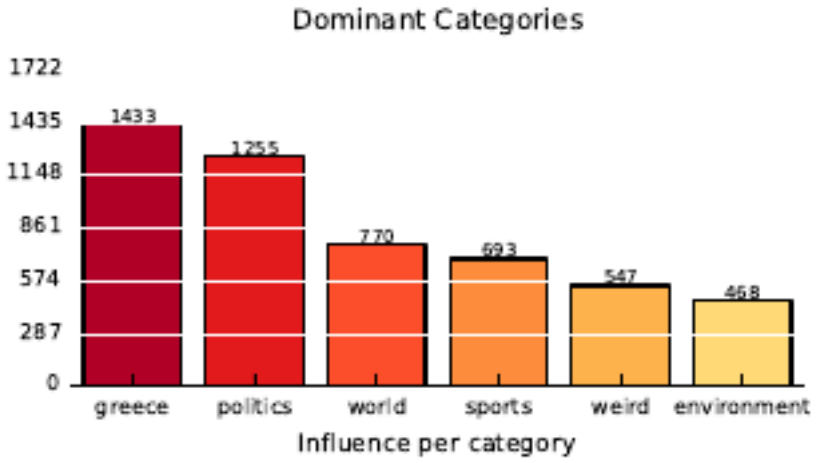
Σε κάθε ξεχωριστή κατηγορία c στο σύνολο των κατηγοριών C , μια πηγή S_i μπορεί να έχει επηρεάσει ένα distinct σύνολο πηγών $S_1, S_2, \ldots, S_n$ με τιμές $I(S_1), I(S_2), \ldots, I(S_n)$ αντίστοιχα. Άρα η λειτουργία που πρέπει να εκτελέσουμε εμείς είναι η εξής:

$$I_c(S_i) = \sum_{\substack{j=1 \\ j \neq i}}^{n} I(S_n) \quad \forall c \in C \tag{3.4}$$

Δηλαδή, όπως και νωρίτερα, ένα groupBy και ένα sum aggregate. Ενώ όμως το sum aggregate εδώ είναι παρόμοιο με πριν, το groupBy πλέον πρέπει να αλλάξει λίγο ώστε να αντιστοιχεί κάθε πηγή S_i με ένα σύνολο *διαφορετικών* πηγών $S_1, S_2, \ldots, S_n$ που έχει επηρεάσει σε κάθε κατηγορία. Αυτό, γνωρίζοντας πως δουλεύει το groupBy, δεν είναι τόσο δύσκολο πλέον. Για να αντιστοιχίσουμε τις επηρεασμένες πηγές για κάθε πηγή σε κάθε κατηγορία πρέπει απλά να κάνουμε groupBy ανάλογα με την αρχική πηγή και την κατηγορία. Δηλαδή, εδώ το $cols = [$*original_source_id, categoryId* $]$. Στην συνέχεια, εκτελούμε το aggregate όπως πριν για να πάρουμε το άθροισμα της επιρροής για κάθε κατηγορία. Τελικά, εκτελούμε και ένα sorting της επιρροής σε αύξουσα σειρά ώστε οι κατηγορίες στις οποίες επηρέασε περισσότερο μια πηγή να είναι στην αρχή.

Τώρα, εφόσον έχουμε την επιρροή για κάθε κατηγορία, μπορούμε να δημιουργήσουμε το bar chart μέσω της βιβλιοθήκης matplotlib. Η διαδικασία με την οποία κάνουμε το plotting των

δεδομένων ονομάζεται `charts()` και η βασική παράμετρος που χρειάζεται είναι τα δεδομένα που θέλουμε να σχεδιάσουμε. Έτσι, αφού διαλέξουμε τις *NUM_OF_CATEGORIES_TO_PLOT* κατηγορίες από τα δεδομένα μας, το μεταφέρουμε στην διαδικασία, αυτή, εσωτερικά, επεξεργάζεται τα δεδομένα και παράγει την εικόνα όπως φαίνεται στην [3.4].



Εικόνα 3.4: Οι κυρίαρχες κατηγορίες για κάθε πηγή είναι αυτές στις οποίες έχει επηρρεάσει τις τις περισσότερες άλλες πηγές (οι οποίες αναπαρήγαγαν την είδηση). Στην προκειμένη περίπτωση φαίνονται οι έξι πιο κυρίαρχες κατηγορίες για την πηγή Newsit.gr

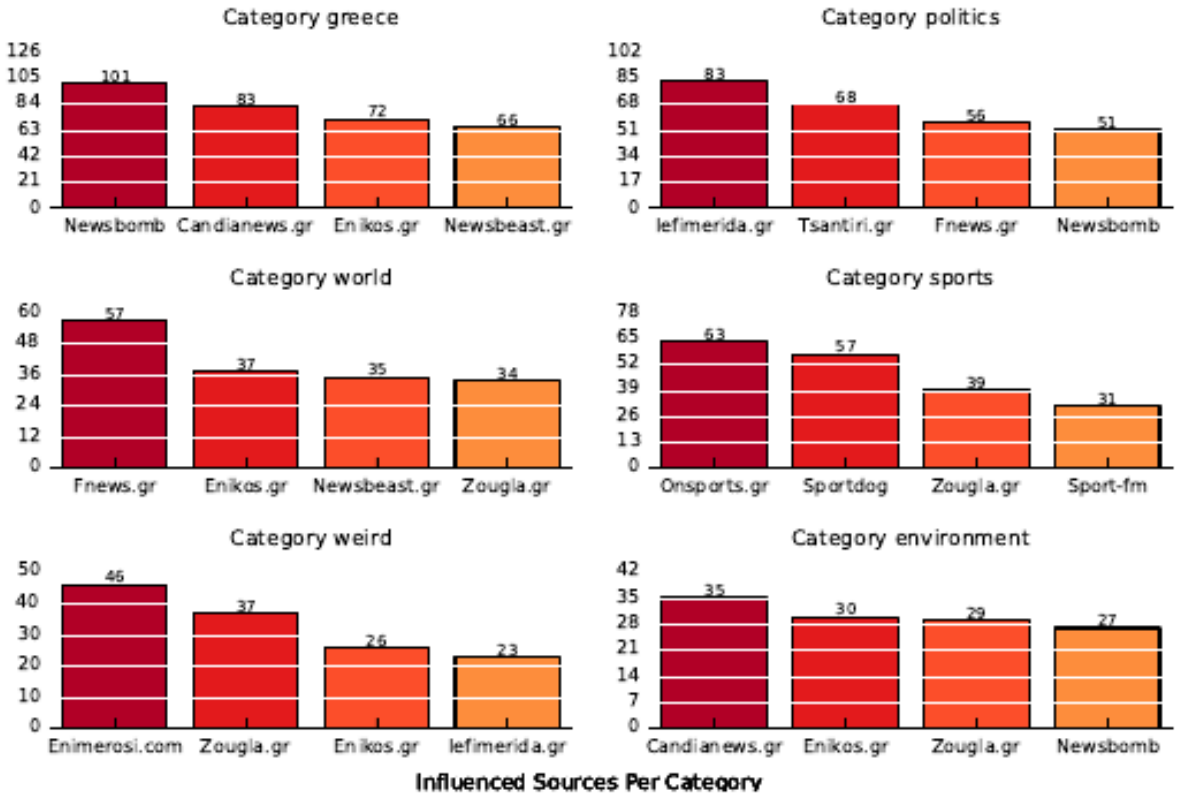
Επιρροή πηγών ανά Κατηγορία

Όπως και στην περίπτωση των κατηγοριών, δεν έχει νόημα να δημιουργήσουμε charts για κάθε μια από τις πηγές που επηρεάζονται. Γι'αυτό και εδώ θα αρκεστούμε στο να παρουσιάσουμε τις *NUM_OF_SOURCES_TO_PLOT* πηγές ανά κατηγορία. Τα δεδομένα και πάλι ανήκουν στο Category Level.

Η διαφορά στην περίπτωση αυτή είναι πως τώρα θέλουμε για κάθε κυρίαρχη κατηγορία να έχουμε **και** τις πηγές που επηρεάστηκαν. Αυτό που σημαίνει αυτό είναι πως το groupBy δεν πρέπει πλέον να αγνοεί το *siteId* για κάθε κατηγορία αλλά αντιθέτως να το χρησιμοποιεί στην ομαδοποίηση. Το $cols$ δηλαδή πρέπει να ισούται με το $cols$ της προηγούμενης περίπτωσης μαζί με την στήλη *siteId*. Στην συνέχεια, το groupBy και aggregate είναι παρόμοιο με αυτό που πραγματοποιήθηκε στο [3.1.2]. Πάλι, πραγματοποιούμε ένα sort της επιρροής για να έχουμε τους πιο επηρεασμένους στην αρχή.

Εφόσον πάλι έχουμε την επιρροή για κάθε πηγή σε κάθε κατηγορία, μπορούμε να δημιουργήσουμε το bar chart μέσω της διαδικασίας `charts()`. Η διαφορά βρίσκεται στα δεδομένα στα οποία περνάμε και μιας άλλης παραμέτρου που υποδηλώνει πως η περίπτωση εδώ είναι οι πηγές ανά κατηγορία. Αφού διαλέξουμε τις *NUM_OF_SOURCES_TO_PLOT* πηγές ανά κατηγορία από τα δεδομένα μας, το μεταφέρουμε στην διαδικασία η έξοδος της οποίας φαίνεται στην εικόνα

[3.5].



Εικόνα 3.5: Για κάθε κατηγορία που ανήκει στις κυρίαρχες κατηγορίες μιας πηγής, κάνουμε plot τις τέσσερις πιο επιρρεασμένες πηγές. Στην εικόνα φαίνονται οι τέσσερις πιο επιρρεασμένες πηγές για τις έξι κυρίαρχες κατηγορίες της ειδησεογραφικής πηγής Euro2day.gr

3.2.2 Δημιουργία PDF

Στην υποενότητα αυτή μια γρήγορη περιγραφή της δημιουργίας του PDF παρουσιάζεται. Υπενθυμίζουμε πως η δημιουργία του έγινε με την βοήθεια του reportLab και πως ο κώδικας όπως και ένα sample PDF βρίσκονται στο GitHub. Η λειτουργία που δημιουργεί και γεμίζει το PDF είναι η `create_report_stats()` που βρίσκεται στο πακέτο **reportUtils**. Η λειτουργία αυτή παίρνει δυο σημαντικές παραμέτρους που επηρεάζουν το παραγόμενο PDF. Η `selected_source_ids` δίνει την δυνατότητα προσδιορισμού των πηγών για τις οποίες θέλουμε να δημιουργήσουμε αναφορά. Η παράμετρος `range_filter`, που δέχεται ως τιμές weekly, monthly, yearly, δίνει την δυνατότητα προσδιορισμού της χρονικής διάστασης της αναφοράς. Εδώ πρέπει να αναφερθεί πως εάν δεν προσδιορίσουμε τουλάχιστον ένα source_id για την 1η παράμετρο, **σαν default, διαλέγουμε τις 20 πηγές με την μεγαλύτερη επιρροή**.

Το PDF παράγεται μέσω της δημιουργίας και του "χτίσιμου" ενός story στο reportLab. Έτσι, αρχικά, δημιουργούμε ένα empty story object που περιέχει τον τίτλο, την ημερομηνία και μια εισαγωγή της παραμετροποίησης που προσφέρεται. Αφού γίνει αυτό, δημιουργούμε τα δεδο-

μένα όπως περιγράφηκαν στην προηγούμενη ενότητα για τις κυρίαρχες κατηγορίες και τις πηγές ανά κατηγορία. Στη συνέχεια, αν δώσουμε τιμή/ες στην παράμετρο `selected_source_ids` κάνουμε ένα iteration για κάθε στοιχείο της, αλλιώς, βρίσκουμε τις 20 πηγές που έχουν επηρεάσει το περισσότερο (συνολικά) άλλες πηγές και κάνουμε ένα iteration για τις πηγές αυτές. Μέσα σε κάθε iteration, φτιάχνουμε τα γραφήματα για κάθε περίπτωση και τα συμπεριλαμβάνουμε στο story. Παράλληλα τυπώνουμε και ένα μικρό εισαγωγικό κείμενο για κάθε περίπτωση και το όνομα της πηγής για την οποία τα δημιουργήθηκαν τα γραφήματα.

3.2.3 Επιπλέον Γραφήματα

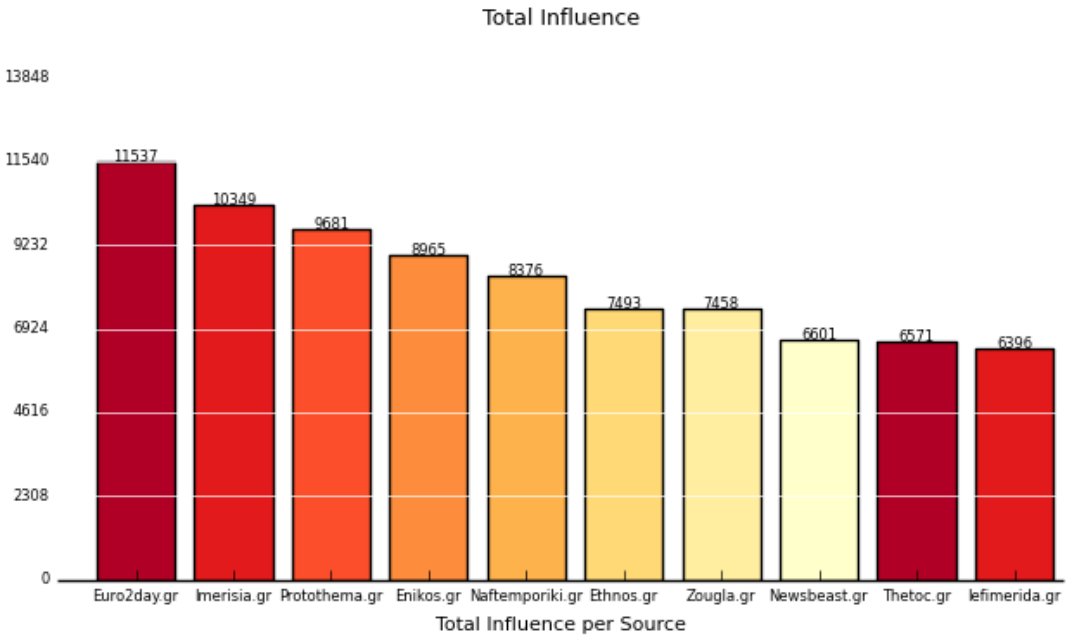
Στην υποενότητα αυτή επιπλέον γραφήματα επιρροής θα παρουσιαστούν για περιπτώσεις που δεν καλύφθηκαν στις προηγούμενες ενότητες. Συγκεκριμένα, θέλουμε να δούμε:

- Συνολική επιρροή. Ποίες πηγές επηρεάζουν το περισσότερο για όλες τις κατηγορίες αθροιστικά. [Εικόνα 3.6]
- Συνολικά επηρεαζόμενοι: Ποίες είναι οι πηγές που επηρεάστηκαν το περισσότερο για όλες τις κατηγορίες αθροιστικά. [Εικόνα 3.7]
- Τις δυο προαναφερόμενες περιπτώσεις αλλά και για τα υπόλοιπα επίπεδα που έχουμε ορίσει (Category, Cluster).

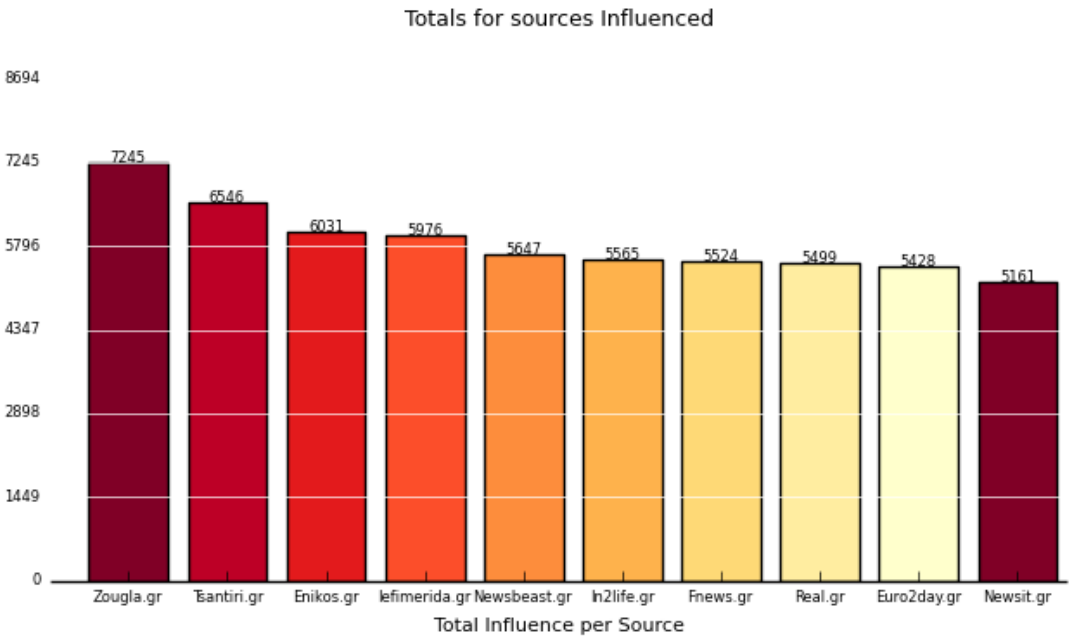
Ενώ οι περιπτώσεις δεν ήταν ήδη υλοποιημένες, δεν είναι δύσκολο να δημιουργηθούν για όποιο επίπεδο και αν εξετάσουμε από την στιγμή που έχουμε επεξεργασμένα δεδομένα και την μέθοδο `find_leaders(kwargs**)`. Η δεύτερη, δίνοντας της ως όρισμα τα δεδομένα για την X κατηγορία, θα μας επιστρέψει τις "αυθεντίες" που επηρεάζουν τις υπόλοιπες πηγές για τις ειδήσεις μιας κατηγορίας. Σαν ενδεικτικό παράδειγμά, θα εξηγήσουμε πάλι την διαδικασία για την εύρεση αυτών που έχουν επηρεαστεί το περισσότερο. Έχοντας τα δεδομένα απλά καλούμε την `find_leaders(data, influenced=true)`, αυτή, μας επιστρέφει για κάθε πηγή τις πηγές που επηρέασε περισσότερο.[4] Μέσω αυτής της συσχέτισης έχουμε και την ποσότητα που επηρεάστηκε κάποια πηγή, στο σύνολο της, από άλλες πηγές. Γνωρίζοντας πως δουλεύει το GroupBy, η διαδικασία για να βρούμε συνολικά πόσο έχει επηρεαστεί μια πηγή είναι απλή. Απλά κάνουμε groupBy ανάλογα με το Id της και μετά κάνουμε ένα αθροιστικό aggregate για τις τιμές της στήλης του influence. Στην συνέχεια, εφόσον έχουμε τα aggregated δεδομένα (στήλες δηλαδή), μπορούμε να τα κάνουμε plot χρησιμοποιώντας όποια βιβλιοθήκη θέλουμε.Εμείς απλά ξανα-χρησιμοποιήσαμε την matplotlib

[4] Επειδή επιστρέφει ένα dictionary object η συγκεκριμένη μέθοδος, μπορούμε να προσδιορίσουμε το επίπεδο που θέλουμε μέσω της `find_leaders(kwargs**)['level']`

για να κάνουμε plot τα δεδομένα μέσω μιας βοηθητικής συνάρτησης `test_charts(kwargs**)` που γράψαμε στα γρήγορα.



Εικόνα 3.6: Στην είκονα αυτή φαίνονται οι δέκα πηγές που επηρέασαν το περισσότερο στο σύνολο των κατηγοριών. Ο αριθμός που μπορεί να γίνει plot, δέκα στην εικόνα, είναι μεταβλητός αλλά θέλει προσοχή ώστε το layout να βγαίνει ομοιόμορφο.



Εικόνα 3.7: Στην είκονα αυτή φαίνονται οι δέκα πηγές που επηρεάστηκαν το περισσότερο στο σύνολο των κατηγοριών.

Ένα βασικό σχόλιο που βγαίνει από τα δύο γραφήματα είναι ότι υπάρχουν πηγές που απασχολούν δημοσιογράφους για να γράφουν νέο περιεχόμενο και άλλες που μόνο αναπαράγουν ειδήσεις. Υπάρχουν και άλλες (π.χ. zougla) που κάνουν και τα δύο.

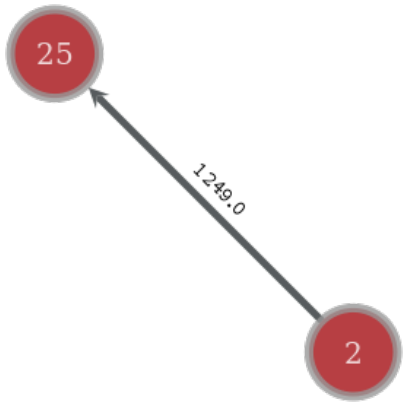
3.3 Τα Δεδομένα ως Γράφος

Στην ενότητα αυτή θα δημιουργήσουμε έναν γράφο από τα δεδομένα μας με σκοπό την εξαγωγή γνώσης από αυτόν μέσω λειτουργιών που ενεργούν επάνω του. Ο κύριος λόγος που εξετάζουμε τα δεδομένα μας με αυτή τη μορφή είναι, αρχικά, επειδή ο γράφος αποτελεί μια ακριβή απεικόνιση αντικειμένων του πραγματικού κόσμου και κατ'επέκταση είναι πιο απτό στην κατανόηση και, δεύτερον, υπάρχουν ήδη υλοποιημένοι πολλοί αλγόριθμοι που ενεργούν σε επίπεδο γράφου και, κατ'επέκταση, διευκολύνουν στην ανάλυση.

Στην υποενότητα [3.3.1] δείχνουμε πως δημιουργήσαμε τον γράφο από τα δεδομένα μας καθώς και μερικά views του γράφου. Στην υποενότητα [3.3.2] δείχνουμε τις λειτουργίες που επιτελέσαμε πάνω του καθώς και τα αποτελέσματα τους. Η κύρια βιβλιοθήκη που χρησιμοποιούμε είναι η **graph-tool** που μας προσφέρει πολλές βοηθητικές μεθόδους για την δημιουργία γράφων και την γρήγορη εφαρμογή αλγορίθμων επάνω τους. Ο κώδικας της ενότητας αυτή βρίσκεται στο πακέτο **dwgraph** του φακέλου **src/**.

3.3.1 Δημιουργία Γράφου

Όπως προαναφέρθηκε στην εισαγωγή [1.3], ο γράφος που θα δημιουργήσουμε έχει ως κύρια χαρακτηριστικά την κατεύθυνση και το βάρος στις ακμές του. Οι κόμβοι συμβολίζουν τις ειδησεογραφικές πηγές και οι ακμές την επιρροή μεταξύ τους[5].



Εικόνα 3.8: Δύο ειδησεογραφικές πηγές με αναγνωριστικά 25 και 2. Η ακμή μεταξύ τους με κατεύθυνση από την 2 στην 25 δείχνει απλά πως η πηγή 2 έχει επηρεαστεί από την πηγή 25 συνολικά 1249 φορές.

Στην εικόνα [3.8] φαίνονται τα χαρακτηριστικά αυτά για ένα μικρό παράδειγμα με δύο κόμβους, στην γενική περίπτωση όμως, δεν θα απεικονίζονται τα βάρη των ακμών για λόγους παρουσίασης.

[5]Οι ακμές μπορούν να συμβολίζουν και το quickness μεταξύ δύο ακμών με σκοπό να δείχνει πόσο γρήγορα επηρεάζεται μια πηγή από μια άλλη. Αλλά δεν θα εξεταστεί η περίπτωση αυτή εδώ.

Υπενθυμίζεται πως συνολικά στο σύνολο δεδομένων υπάρχουν 239 κόμβοι και περίπου 170.000 ακμές.

Δημιουργία

Η λογική της δημιουργίας του γράφου έγκειται σε μια μεγάλη επανάληψη που διαβάζει γραμμές, φτιάχνει τους κόμβους και αναθέτει τα βάρη μεταξύ τους. Αρχικό (αλλά προαιρετικό!) βήμα είναι η δημιουργία ενός edge list αρχείου, δηλαδή ενός αρχείου που περιέχει τρεις στήλες, το id της αρχικής πηγής, το id της επηρεαζόμενης πηγής και την επιρροή μεταξύ τους. Χρησιμοποιώντας dataframes αυτό είναι εύκολο με το να κάνουμε απλά drop τις στήλες που δεν χρησιμοποιούμαι από το επεξεργασμένο αρχείο. Αυτό ακριβώς κάνει η μέθοδος `to_edge_list()` δεχόμενη τα δεδομένα ως είσοδο και επιστρέφοντας το edge list.

Έχοντας το edge list μπορούμε πλέον να δημιουργήσουμε έναν καινούργιο γράφο γι'αυτά τα δεδομένα. Όπως είναι υλοποιημένο αυτό γίνεται με την δημιουργία ενός καινούργιου dwgGraph αντικειμένου με κύρια παράμετρο το edge list. Κατά την δημιουργία του αντικειμένου, αυτό, εσωτερικά, καλεί την μέθοδο του `build_graph(edge_list)` για να κατασκευάσει τον γράφο.

Η μέθοδος build_graph()

Η μέθοδος αυτή είναι που εκτελεί την επανάληψη για κάθε γραμμή στο edge list, που έχουμε δώσει ως παράμετρο, και δημιουργεί τους κόμβους και τις ακμές. Η λογική με την οποία λειτουργεί είναι η εξής. Πριν την εκτέλεση της επανάληψης δημιουργούμε δύο αρχικά άδεια σύνολα seen_source_ids $= \{\}$ και seen_site_ids $= \{\}$, με την πρώτη να κρατάει κόμβους που επηρεάζουν και την δεύτερη κόμβους που επηρεάζονται. Η ανάγκη δύο διαφορετικών συνόλων πηγάζει από το γεγονός ότι είναι σύνηθες να υπάρχει πηγή που και επηρεάζει και επηρεάζεται και, εφόσον δεν θέλουμε διπλότυπους κόμβους, χρησιμοποιούμε τα σύνολα αυτά για να μπορούμε να εξετάσουμε κάθε περίπτωση. Στον αλγόριθμο [1] φαίνονται τα βήματα που ακολουθήσαμε.

Με λίγα λόγια ο ο αλγόριθμος δημιουργεί τα σύνολα V, E που περιέχουν τους κόμβους και της ακμές αντίστοιχα, και, αναθέτει τα βάρη w για κάθε $e \in E$. Όταν τελειώσει η επανάληψη αυτή έχουμε δημιουργήσει έναν κατευθυνόμενο γράφο με βάρη $G = (V, E, w)$ που αντιπροσωπεύει πλέον τα στοιχεία στα δεδομένα μας. Η γραφική παρουσίαση του γράφου G θα παρουσιαστεί στην επόμενη υποενότητα.

Algorithm 1: Δημιουργία Γράφου

Data: `edge_list (DataFrame)`

Result: `graph-tool.Graph (Object)`

seen_source_ids $\leftarrow \emptyset$

seen_site_ids $\leftarrow \emptyset$

$V, E \leftarrow \emptyset$

while *edge list has line* **do**

  if *original_source_id* $\notin$ *seen_source_ids* **then**

    if *original_source_id* $\notin$ *seen_site_ids* **then**

      seen_source_ids $\leftarrow$ *seen_source_ids* $\cup$ *original_source_id*

      $V \leftarrow V \cup$ *original_source_id*

    end

  end

  if *site_id* $\notin$ *seen_site_ids* **then**

    if *site_id* $\notin$ *seen_source_ids* **then**

      seen_site_ids $\leftarrow$ *seen_site_ids* $\cup$ *site_id*

      $V \leftarrow V \cup$ *original_site_id*

    end

  end

  if $e =$ (*original_source_id, site_id*) $\in E$ **then**

    $w(e)$ += $w(e)$

  else

    $w(e) = w(e)$

    $E \leftarrow e$(*original_source_id, site_id*)

  end

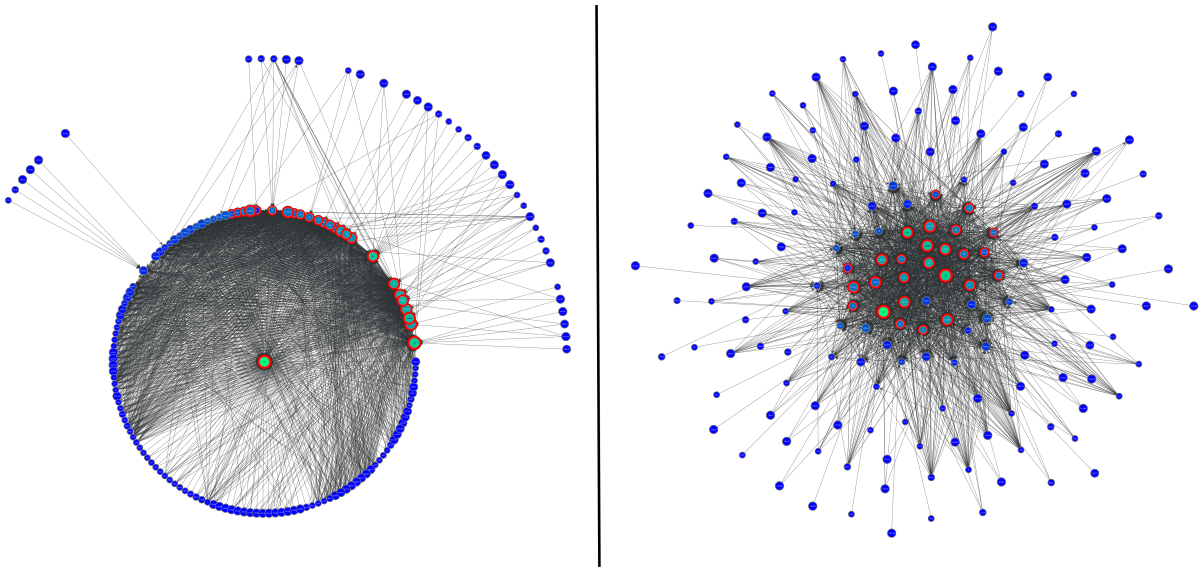
end

Graph View

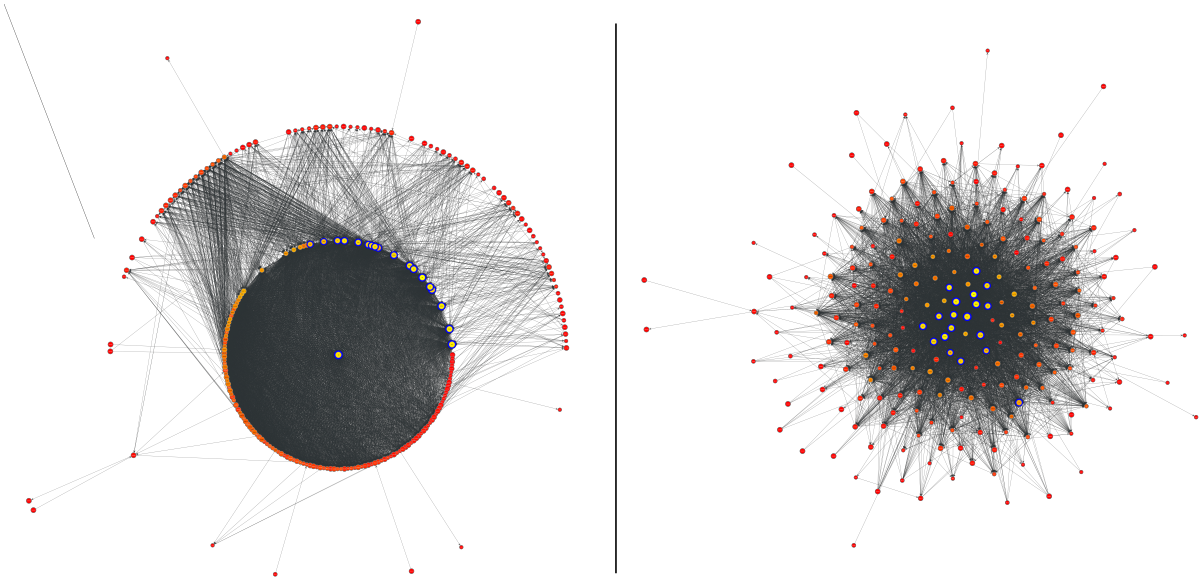
Πέρα από την δημιουργία γράφων η βιβλιοθήκη **graph-tool** μας δίνει και την δυνατότητα να κάνουμε visualize τον γράφο μέσω άλλων βιβλιοθηκών. Στις επόμενες εικόνες φαίνονται σχέδια των γράφων που δημιουργήθηκαν με την βιβλιοθήκη γραφικών **cairo** για δύο διαφορετικές περιπτώσεις. Πρώτον, στην εικόνα [3.9] φαίνεται ο γράφος για την κατηγορία "Ελλάδα" σε δύο διαφορετικά layouts *radial tree, sfdp* [6]. Δεύτερον στην εικόνα [3.10] φαίνεται ο γράφος για το σύνολο των

[6]Τα layouts αυτά είναι ουσιαστικά δύο διαφορετικοί αλγόριθμοι για το πως θα "ζωγραφιστούν" οι κόμβοι και οι ακμές ενός γράφου

δεδομένων που είχαμε και πάλι χρησιμοποιώντας τα ίδια layouts όπως πριν.



Εικόνα 3.9: Δύο διαφορετικές αναπαραστάσεις του γράφου για την κατηγορία '2' : Ελλάδα. Στην αριστερή εικόνα ο κόμβος με την μεγαλύτερη επιρροή φαίνεται στην μέση ενώ στην δεξιά οι κόμβοι με την μεγαλύτερη επιρροή στο κέντρο.



Εικόνα 3.10: Δύο διαφορετικές αναπαραστάσεις του γράφου για το σύνολο των δεδομένων μας. Στην αριστερή εικόνα ο κόμβος με την μεγαλύτερη επιρροή φαίνεται στην μέση ενώ στην δεξιά οι κόμβοι με την μεγαλύτερη επιρροή στο κέντρο.

Σε κάθε περίπτωση οι κόμβοι είναι χρωματισμένοι ώστε οι πιο ανοιχτόχρωμοι να υποδηλώνουν μεγαλύτερη επιρροή. Επίσης, για τους κόμβους με αρκετά μεγάλη επιρροή, ζωγραφίζουμε και ένα halo γύρω τους για να το δηλώσουμε.[7]

[7]Παρόλο που επειδή το μέγεθος τους σε pixel είναι σχετικά μεγάλο, με δυσκολία φαίνεται..

3.3.2 Λειτουργίες Πάνω στον Γράφο

Ανεξαρτήτως επιπέδου (ανά κατηγορία ειδήσεων ή συνολικά) πλέον, εφόσον έχουμε δημιουργήσει τον γράφο σαν αντικείμενο, μπορούμε να εκτελέσουμε διαφορετικούς αλγορίθμους πάνω του ανάλογα με τις περιπτώσεις που θέλουμε να εξετάσουμε. Στις επόμενες υποενότητες εξετάζουμε δύο περιπτώσεις, το PageRank των κόμβων του γράφου καθώς και το κατά πόσο μπορούμε να χωρίσουμε τον γράφο σε ομάδες πηγών που επηρεάζουν πολύ η μία την άλλη.

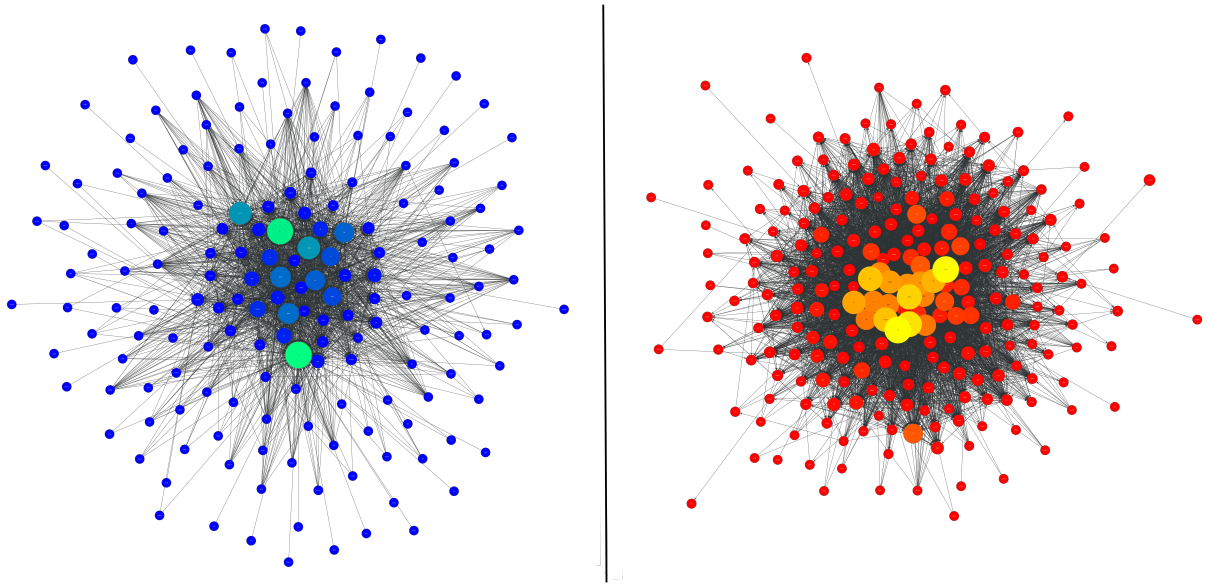
PageRank

Ο PageRank, ένας αλγόριθμος που αναπτύχθηκε από την Google στην οποία χρησιμοποιείται και σήμερα, δέχεται ως είσοδο έναν κατευθυνόμενο γράφο και παράγει μια κατάταξη των κόμβων ανάλογα με την σημαντικότητα τους. Σύμφωνα με την Google ο Pagerank δουλεύει με το να μετράει τον αριθμό και την ποιότητα των ακμών σε έναν κόμβο για να συμπεράνει πόσο σημαντικός είναι. Η υπόθεση είναι πως σημαντικοί κόμβοι είναι πιο πιθανό να δέχονται περισσότερες ακμές από άλλους κόμβους.

Στην εξίσωση [3.5] φαίνεται ο υπολογισμός του PageRank PR για κάθε κόμβο $v \in V$. Η μεταβλητή d δηλώνει την πιθανότητα, για κάθε βήμα, ενός imaginary surfer που τυχαία ακολουθεί ακμές, να συνεχίσει να ακολουθεί ακμές έναντι του να σταματήσει. Το ονομάζουμε damping factor και γενικά η τιμή που του ορίζεται είναι 0.85. Το $N = n$ εδώ, δηλαδή το N συμβολίζει το σύνολο των κόμβων στο γράφο μας. Το $\Gamma^-(v)$ και $d^+(u)$ για τους κόμβους v και u συμβολίζουν τους γείτονες του κόμβου και το out-degree του αντίστοιχα.

$$PR(v) = \frac{1-d}{N} + d \sum_{u \in \Gamma^-(v)} \frac{PR(u)}{d^+(u)} \tag{3.5}$$

Το άθροισμα της εξίσωσης υπολογίζει, για κάθε γείτονα u του κόμβου v, το PageRank του $PR(u)$ δια τον αριθμό των ακμών που αρχίζουν από τον u και τελειώνουν σε κάποιον άλλον κόμβο. Η διαίρεση πραγματοποιείται ώστε η τιμή του PageRank για κάθε κόμβο να μοιράζεται στου γείτονες του με ομοιόμορφο τρόπο. Στην εικόνα [3.11] το PageRank και πάλι για την κατηγορία 2 και για την περίπτωση του συνόλου φαίνεται. Οι πιο σημαντικές πηγές είναι πιο ανοιχτόχρωμες και με μεγαλύτερο μέγεθος ενώ, στον πίνακα [3.1] φαίνονται οι δέκα πηγές που έχουν το μεγαλύτερο pagerank value για το σύνολο των κατηγοριών.



Εικόνα 3.11: Το PageRank για της δύο διαφορετικές περιπτώσεις. Στα αριστερά για την κατηγορία 2: 'Ελλάδα' ενώ, στα δεξιά, για το σύνολο των δεδομένων.

PAGERANK VALUES

News Source	Value
Euro2day.gr	0.0478238661685
Imerisia.gr	0.0470638430403
Protothema.gr	0.0426127786659
Enikos.gr	0.0395576157629
Zougla.gr	0.036728189514
Naftemporiki.gr	0.0362524003234
Ethnos.gr	0.0336622515437
Sport-fm	0.0308966813659
Thetoc.gr	0.029691758954
Newsbeast.gr	0.0285247409145

Πίνακας 3.1: Top ten news sources based on their PageRank Value

Ομάδες Πηγών

Όπως αναφέραμε και στην αρχή αυτής της ενότητας, ένας από τους σκοπούς μας ήταν η ανακάλυψη ομάδων στις οποίες οι κόμβοι αλληλοεπηρεάζονται σε μεγάλο βαθμό. Οι ομάδες αυτές ουσιαστικά αποτελούν ένα υποσύνολο του συνόλου V στο οποίο οι κόμβοι έχουν πολλές ακμές μεταξύ τους. Δεν απαιτείται να υπάρχουν ακμές από κάθε κόμβο προς κάθε κόμβο αλλά να υπάρχουν αρκετοί κόμβοι οι οποίοι επηρεάζουν τις ίδιες πηγές ή/και επηρεάζονται από κοινές πηγές. Επίσης, ένας κόμβος δύναται να ανήκει σε περισσότερες από μια ομάδες. Για την επίτευξη του σκοπού αυτού, αναπτύξαμε έναν αλγόριθμο που προσπαθεί να βρει τα σύνολα αυτά για κάθε διαφορετική πηγή. Ο αλγόριθμος έτσι όπως υλοποιήθηκε στις μεθόδους `mutual_sets()`, `node_set_groups()`,

παρουσιάζεται στην συνέχεια.

- `mutual_sets()`: Η μέθοδος αυτή υπολογίζει για κάθε κόμβο το σύνολο των κοινών κόμβων που μοιράζεται με τους γειτονές [8] τους.
- `node_set_groups()`: Η μέθοδος αυτή ελέγχει για κάθε σύνολο που επιστρέφει η `mutual_sets()` κατα πόσο η επιρροή μεταξύ των πηγών υπερβαίνει ένα προκαθορισμένο κατώφλι.

Algorithm 2: Find Node Groups

Data: `graph-tool.Graph (Object)`

Result: `Groups`

mutual_sets $\leftarrow \emptyset$

for *vertex* $v \in V$ **do**

  common $\leftarrow N_{\Gamma}^{+}(v) \cap N_{\Gamma}^{-}(v)$

  for *neighbour* $\in$ *common* **do**

    neighbour_common $\leftarrow N_{\Gamma}^{+}(neighbour) \cap N_{\Gamma}^{-}(neighbour)$

    size $\leftarrow$ *common* $\cap$ *neighbour_common*

    if $\|size\| > threshold$ **then**

      retain neighbour common set in a list

    end

  end

  mutual_set $\leftarrow$ *commons*

  for *set in list of neighbour commons* **do**

    mutual_set $\leftarrow$ *mutual_set* $\cap$ *set*

  end

end

for *every edge* e *between nodes* $\in$ *mutual_sets* **do**

  if $w(e) <$ *influence_threshold* **then**

    remove node from set

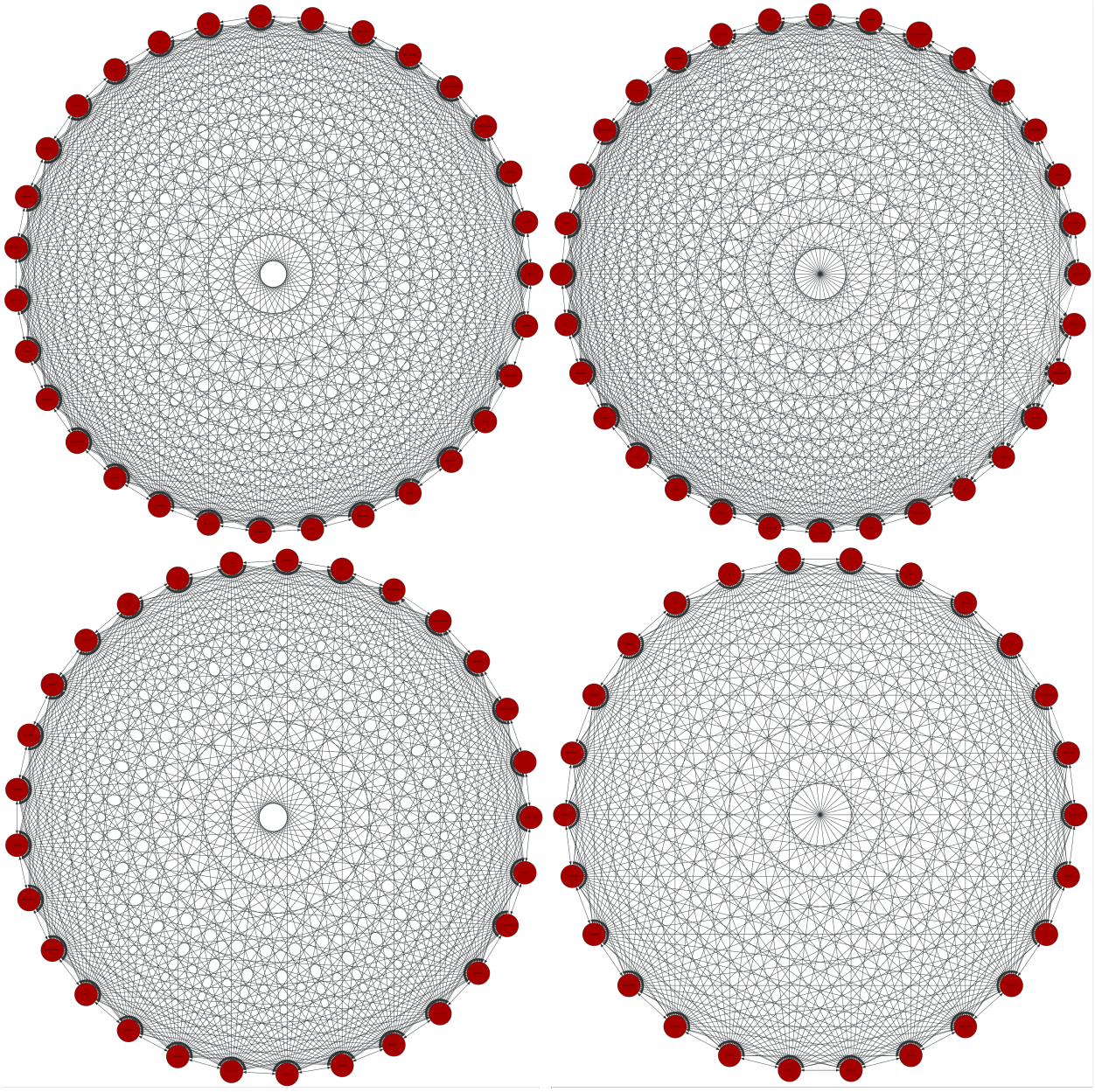
  end

end

Αφού ολοκληρωθεί η εκτέλεση έχουμε ως αποτέλεσμα για κάθε κόμβο την πιθανή του ομάδα. Ένας κόμβος μπορεί να ανήκει σε περισσότερες από μια ομάδες αν βεβαίως πληρεί τα κριτήρια. Στην συνέχεια, κάνοντας μια μικρή επεξεργασία, "ζωγραφίζουμε"[9] μερικές ομάδες όπως φαίνεται στην εικόνα [3.12]. Στην εικόνα αυτή φαίνονται οι τρεις μεγαλύτερες ομάδες που βρέθηκαν για χαλαρά κριτήρια του influence threshold.

[8] Δηλαδή αυτούς με τους οποίους συνδέεται με μια ακμή

[9] Κυριολεκτικά ζωγραφίζουμε!



Εικόνα 3.12: Σύνολα διαφορετικών ομάδων με χαλαρό κριτήριο ομαδοποίησης. Για ομάδες με μέγεθος 31 (πάνω αριστερά), 32 (πάνω δεξιά), 29 (κάτω αριστερά) και 26 (κάτω δεξιά).

ΣΕΛΙΔΑ ΣΚΟΠΙΜΑ ΚΕΝΗ

ΚΕΦΑΛΑΙΟ 4

BREAKING NEWS

Στο κεφάλαιο αυτό θα εξετάσουμε κατά πόσο μπορούμε να φτιάξουμε ένα μοντέλο μηχανικής μάθησης που δεχόμενο ένα cluster από άρθρα σε κάποια χρονική στιγμή, θα μπορεί να προβλέψει κατά πόσο η είδηση για την οποία μιλάνε τα άρθρα αυτά (αφού κάθε άρθρο του ίδιου cluster μιλάει για το ίδιο θέμα) είναι έκτακτη. Συγκεκριμένα, η ακολουθία των βημάτων που πρέπει να ακολουθήσουμε είναι η εξής:

- Επιλογή χαρακτηριστικών που θεωρούμε πως έχουν το μεγαλύτερο **predictive value** για τα ειδησεογραφικά μας cluster. Αφού διαλέξουμε τα χαρακτηριστικά αυτά, πρέπει να επεξεργαστούμε τα δεδομένα ώστε να τα παράξουμε για κάθε cluster. [4.1]
- Δημιουργία των δεδομένων εισόδου και εξόδου από τα χαρακτηριστικά που δημιουργήσαμε στο προηγούμενο βήμα. Τα δεδομένα αυτά θα λειτουργήσουν ως σύνολο εκπαίδευσης του μοντέλου μας. [4.2]
- Επιλογή μοντέλου μηχανικής μάθησης, εκπαίδευση του και αξιολόγηση της ικανότητας που έχει να προβλέπει ορθά "άγνωστα" δεδομένα. Παράλληλα, εξέταση του συσχετισμού της ακρίβειας του μοντέλου με διάφορες παραμέτρους που έχουν οριστεί. [4.3]

Οι λειτουργίες που υλοποιούν τα βήματα αυτά βρίσκονται στο πακέτο **breakingNews/** του φακέλου **src/**.

4.1 Χαρακτηριστικά Ειδήσεων

Στην ενότητα αυτή, προσπαθούμε να βρούμε τα χαρακτηριστικά εκείνα που, όταν χαρακτηρίζουν τα clusters που έχουμε, συμβάλλουν στην έγκυρη μελλοντική κατηγοριοποίηση τους. Στην υποενότητα [4.1.1] εξηγούμε την λογική με την οποία διαλέξαμε τα χαρακτηριστικά που διαλέξαμε με βάση το θεματικό πλαίσιο πρόβλεψης που έχουμε.

4.1.1 Επιλογή Χαρακτηριστικών

Για να επιλέξουμε τα χαρακτηριστικά που ορίζουν κάθε cluster πρέπει αρχικά να αναλύσουμε το τι ακριβώς θέλουμε να προβλέψουμε. Στα πλαίσια της πτυχιακής, οι προβλέψεις που θέλουμε το μοντέλο μας να μπορεί να κάνει είναι κατά πόσο ένα άρθρο, στα πλαίσια ενός cluster, είναι άρθρο που μιλάει για μια έκτακτη είδηση ή ένα άρθρο που μιλάει για μια είδηση μέτριου ή χαμηλού ενδιαφέροντος. Το επίθετο έκτακτη χρησιμοποιείται loosely στην περίπτωση αυτή και στις επακόλουθες υποενότητες θα το χρησιμοποιούμε μαζί με την λέξη "σημαντικό/σημαντικότητα" χωρίς να εννοούμε κάτι διαφορετικό.

Γνωρίζοντας το τι θέλουμε να προβλέψουμε, μπορούμε πλέον να προσπαθήσουμε να σκεφτούμε ποία θα ήταν τα χαρακτηριστικά εκείνα που θα βοηθούσαν στην πρόβλεψη. Στα πλαίσια του δικτύου αναπαραγωγής ειδήσεων που έχουμε, τα κύρια χαρακτηριστικά που ορίζουν κατά πόσο μια είδηση είναι είδηση με μεγάλη σημαντικότητα είναι δύο:

1. Το γεγονός ότι πολλοί κόμβοι στο δίκτυο έχουν αναπαράγει αυτή την είδηση. Άρα το μέγεθος του cluster για μια είδηση είναι μεγάλο.

2. Το γεγονός ότι αυτή η είδηση μεταδίδεται/αναπαράγεται πολύ γρήγορα στα πλαίσια του δικτύου που εξετάζουμε. Άρα πως οι χρόνοι αναπαραγωγής των ειδήσεων στο cluster έχουν πολύ μικρή διαφορά μεταξύ τους.

Αν το σκεφτούμε περισσότερο συμπεραίνουμε πως δεν αρκεί μόνο ένα από τα δύο χαρακτηριστικά για να είναι έγκυρος ο ορισμός. Συγκεκριμένα, επειδή ένα cluster έχει πολύ μεγάλο μέγεθος, αυτό δεν συνεπάγεται πως είναι μια είδηση μεγάλης σημαντικότητας. Μπορεί απλούστατα να έκανε πολύ χρόνο ώστε να φτάσει στο μέγεθος που είναι (π.χ 24 ώρες) υποδηλώνοντας ουσιαστικά ένα άρθρο που έχει ενδιαφέρον για πολλές πηγές. Αντίστοιχα, ούτε ένα cluster στο οποίο οι χρόνοι αναπαραγωγής είναι πολύ γρήγοροι υποδηλώνει μια είδηση μεγάλης σημαντικότητας. Διότι το cluster μπορεί να έχει μέγεθος 2 με την μια πηγή να αναπαρήγαγε την είδηση από την άλλη σχεδόν άμεσα. Στην περίπτωση αυτή, μπορεί να μεγάλωσε όσο πιο γρήγορα γινόταν αλλά επειδή το

μέγεθος του είναι πολύ μικρό δεν μας επιτρέπει να το χαρακτηρίσουμε ως σημαντικό. Φαίνεται δηλαδή, πως για να μπορέσουμε να χαρακτηρίσουμε ορθά κάθε cluster για την περίπτωση μας, έπρεπε να συνδυάσουμε τα δυο αυτά χαρακτηριστικά.

Αρχικά, πρέπει να δημιουργήσουμε δυο σύνολα δεδομένων για το μοντέλο μας: Το πρώτο με τιμές που αναπαριστούν clusters των οποίων οι ειδήσεις έχουν μεγάλη σημαντικότητα, δηλαδή το σύνολο των clusters που είναι **έγκυρα** στο δικό μας πλαίσιο. Το δεύτερο με τιμές που αναπαριστούν clusters των οποίων οι ειδήσεις δεν έχουν μεγάλη σημαντικότητα, δηλαδή, τα **μη-έγκυρα** clusters. Οι τιμές γι'αυτά τα σύνολα είναι, για το πρώτο, οι ρυθμοί αύξησης του cluster κάθε t χρονική στιγμή για clusters με μέγεθος που ξεπερνάει ένα προκαθορισμένο κατώφλι και, για το δεύτερο, οι ρυθμοί αύξησης και πάλι, αλλά τώρα, για τα clusters τα οποία είναι μικρά σε μέγεθος. Συνδυάζοντας αυτά τα δύο σύνολα χρησιμοποιώντας τα χαρακτηριστικά που ορίσαμε θα μπορούμε να εκπαιδεύσουμε και ελέγξουμε σωστά το μοντέλο μηχανικής μάθησης που θα φτιάξουμε στην πορεία.

4.2 Δημιουργία Χαρακτηριστικών

Για την δημιουργία των χαρακτηριστικών όπως ορίστηκαν νωρίτερα γίνεται έντονη χρήση της NumPy βιβλιοθήκης. Όλες οι μέθοδοι που χρησιμοποιήσαμε για τον σκοπό αυτό βρίσκονται στο αρχείο **bNDataMethods** του πακέτου **breakingNews**.

4.2.1 Χαρακτηριστικά Για Κάθε Σύνολο

Τα δεδομένα που χρησιμοποιούμε στην περίπτωση αυτή ανήκουν αναγκαστικά στην κατηγορία **Cluster Level** για εμφανέστατους λόγους. Για το σύνολο των έγκυρων δεδομένων καθώς και για το σύνολο των μη-έγκυρων δεδομένων η μεθοδολογία είναι παρόμοια. Όπως θα φανεί, η μόνη αλλαγή που απαιτείται έγκειται στις τιμές των σχετικών παραμέτρων που περνάμε στην κλήση των μεθόδων και γι'αυτό, δεν θα γραφούν σχετικές υποενότητες για το καθένα. Αντιθέτως, όπου χρειάζεται, θα παρουσιάζεται η αλλαγή που απαιτείται για την κάθε περίπτωση. Το σύνολο των έγκυρων δεδομένων αναγνωρίζεται με την λέξη-πρόθεμα "breaking" ενώ το σύνολο των μη-έγκυρων με την αντίστοιχη "not-breaking". Στις επόμενες παραγράφους, η μεθοδολογία που ακολουθήθηκε, παρουσιάζεται.

Get Cluster Leaders

Ασχέτως περιπτώσεως, πρέπει να βρούμε τα clusters τα οποία πληρούν τις προ-υποθέσεις που έχουμε θέσει. Για την "breaking" περίπτωση, ο πρώτος περιορισμός που έχουμε θέσει για να τα δεχθούμε ως έγκυρα είναι πως τα clusters πρέπει να έχουν μέγεθος που υπερβαίνει μια συγκεκριμένη τιμή. Για την not-breaking περίπτωση, ο περιορισμός αυτός είναι ανάστροφος, δηλαδή, ότι δεν ανήκει στο πρώτο σύνολο θεωρούμε πως ανήκει στο δεύτερο. Για τα δεδομένα μας το μέγεθος κάθε cluster είναι ουσιαστικά ο αριθμός της στήλης *influence* αφού η τιμή αυτή δείχνει πόσες αναδημοσιεύσεις υπήρξαν, άρα, πόσα άρθρα γράφηκαν για την συγκεκριμένη είδηση.

Γνωρίζοντας το μέγεθος κάθε cluster μπορούμε τώρα να δημιουργήσουμε ένα "φίλτρο" (threshold) που ως σκοπό θα έχει να φιλτράρει όσα clusters είναι μικρότερα ή μεγαλύτερα της τιμής του. Η τιμή που μπορεί να επιλεχθεί για το φίλτρο είναι γενικά αυθαίρετη και, σαν αρχική τιμή πειραματισμού, επιλέχθηκε να είναι ίση με 15.[1] Το φιλτράρισμα είναι ενσωματωμένο στην μέθοδο **find leaders** μέσω της παραμέτρου `find_leaders(filtered=True)` την οποία χρησιμοποιούμε με παρόμοια τιμή και για τις δύο περιπτώσεις μας. Αντιθέτως, για να μπορούμε να ξεχωρίσουμε τις περιπτώσεις, έχουμε ορίσει μια λογική παράμετρο με όνομα **operator** και επιτρεπόμενες τιμές '>', '<'. Η τιμή '>' φιλτράρει clusters μεγαλύτερα του φίλτρου (δηλαδή, τα breaking) ενώ η '<' clusters τα οποία είναι μικρότερα (not-breaking). Άρα, σε κάθε περίπτωση, η κλήση της μεθόδου είναι της μορφής `find_leaders(data, filtered=True, operator=operator)` και μας επιστρέφει τα clusters ανάλογα με το threshold που ορίσαμε.

Get Cluster Data

Η μορφή που μας επιστρέφεται από το προηγούμενο βήμα, ενώ έχει τα ids των clusters που σκοπεύουμε να χρησιμοποιήσουμε, δεν έχει τα δεδομένα στην μορφή την οποία θέλουμε. Συγκεκριμένα, περιέχει τις στήλες (όπως αυτές είχαν οριστεί) σε μια aggregated μορφή. Εφόσον όμως εμείς θέλουμε να ελέγξουμε τον ρυθμό αύξησης του μεγέθους κάθε cluster, χρειαζόμαστε τους χρόνους που αναπαράγονται, δηλαδή, είτε το delay είτε το quickness κάθε αναπαραγόμενου άρθρου του cluster. Άρα, σαν επόμενο βήμα, πρέπει να συλλέξουμε τα δεδομένα που αντιστοιχούν σε αυτά τα clusters.

Η αντιστοίχηση των clusters που έχουν βρεθεί, και για τις δύο περιπτώσεις, με τα δεδομένα τους σε not-aggregated μορφή, υλοποιείται μέσω της μεθόδου `get_leader_data(kwargs**)`. Αυτή, δεχόμενη ως είσοδο τα σύνολα των clusters μαζί με τα δεδομένα τους, πραγματοποιεί την

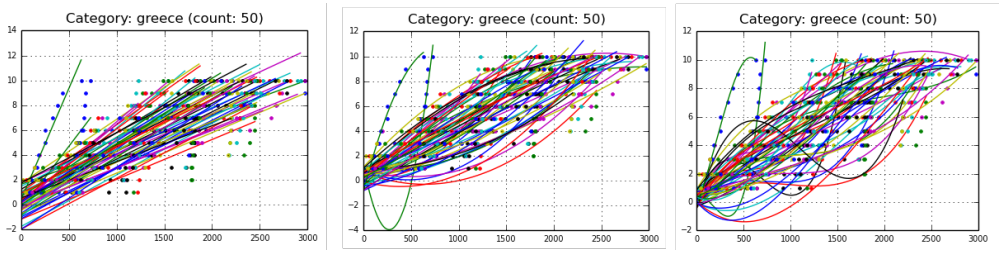
[1] Το φίλτρο αυτό με όνομα INFLUENCE_THRESHOLD βρίσκεται στο αρχείο **bnConf** σε μορφή μεταβλητής.

αντιστοίχηση που επιθυμούμε. Τα δεδομένα για κάθε cluster είναι όπως φαίνονται στην εικόνα [3.2] μετά το πρώτο στάδιο επεξεργασίας.[2] Μετά την κλήση αυτής της μεθόδους έχουμε τα δεδομένα των clusters για την breaking και not-breaking περίπτωση.

Finding the Curve Fits

Για να μπορέσουμε να βρούμε τον ρυθμό αύξησης του μεγέθους ενός cluster πρέπει να δημιουργήσουμε την καμπύλη που αναπαριστά την αύξηση του μεγέθους του σε συνάρτηση με τον χρόνο. Ως ένδειξη του χρόνου διαλέξαμε την στήλη *delay* για κάθε cluster. Στην μορφή που έχουμε τα δεδομένα μας, δεν μπορούμε κατευθείαν να βρούμε την καμπύλη αυτή μιας και έχουμε διακριτά σημεία που αναπαριστούν τον χρόνο αναδημοσίευσης για κάθε άρθρο. Άρα, αρχικά πρέπει να κάνουμε fit τα delay values για κάθε cluster σε μια κατάλληλη καμπύλη. Η λειτουργία αυτή εκτελείται με την μέθοδο `fit_curve_to_data()` και σαν μόνη παράμετρο παίρνει τα δεδομένα (δηλαδή τα delay points). Εσωτερικά ορίζονται δύο ακόμη παράμετροι σημασίας που πρέπει να αναφερθούν.

Ο πρώτος, με όνομα *DELAY_POINTS*, ορίζει πόσα από τα σημεία που έχουμε για κάθε cluster θέλουμε να χρησιμοποιήσουμε για να κάνουμε fit την καμπύλη. Η παράμετρος αυτή έχει μεγάλη σημασία για την μετέπειτα πρόβλεψη αφού ορίζει τον ελάχιστον αριθμό αναδημοσιεύσεων (άρα κατ'επέκταση delay points) που απαιτείται να έχουν μελλοντικά clusters που θέλουμε να προβλέψουμε. Υπάρχει ένα trade-off εδώ, όσα περισσότερα delay points ορίσουμε, τόσο πιο καλό γίνεται το μοντέλο αλλά, πρέπει να περιμένουμε περισσότερο χρόνο να "γεμίσει" ένα πιθανό cluster ώστε να μπορέσουμε να αποφανθούμε για την κατάσταση του. Αυτό που διαλέξαμε εμείς τελικά ήταν *DELAY_POINTS*$= 5$, μια τιμή που δεν είναι ούτε πολύ μικρή αλλά ούτε και πολύ μεγάλη.



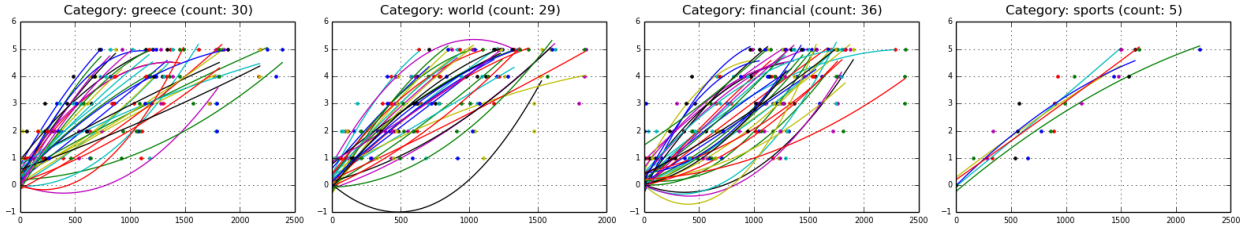
Εικόνα 4.1: Διαφορετικό degree για κάθε καμπύλη σημαίνει διαφορετικό fit της καμπύλης στα σημεία που έχουμε. Από τα αριστερά προς τα δεξιά, με DELAY_POINTS=10, φαίνονται τα fits της καμπύλης για deg $= [1, 2, 3]$.

Η δεύτερη παράμετρος με όνομα *CURVE_DEGREE* ορίζει το πολυωνομικό degree που θέλουμε να έχει η καμπύλη που σκοπεύουμε να κάνουμε fit στα σημεία που αντιπροσωπεύουν το delay. Όπως φαίνεται και από την εικόνα [4.1] η επιλογή που θα διαλέξουμε έχει σημαντική

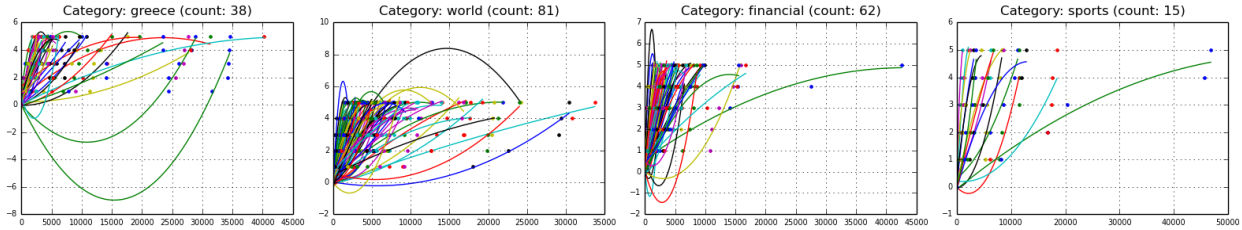
[2] και έχουμε πρόσβαση σε αυτά μέσω της `import_processed_data()`.

επίπτωση στον ρυθμό αύξησης που θα βρούμε. Για την πρώτη (deg $=$ 1) περίπτωση ο ρυθμός αύξησης θα είναι σταθερός γιατί μιλάμε για ευθεία. Για την δεύτερη (deg $=$ 2) θα έχουμε μια διακύμανση του αλλά μαζί με λίγες περιπτώσεις όπου η καμπύλη έχει μεγαλύτερη διακύμανση απ'ότι θα θέλαμε. Στην τρίτη περίπτωση, για degree $=$ 3, έχουμε καλύτερο fit αλλά με περισσότερες περιπτώσεις διακύμανσης της καμπύλης. Για το δικό μας μοντέλο αποφασίσαμε τελικά να διαλέξουμε την δεύτερη περίπτωση όπου $deg = 2$.[3]

Ορίζοντας τις τιμές αυτές για τις παραμέτρους *CURVE_DEGREE* και *DELAY_POINTS* καλούμε την μέθοδο περνώντας ως όρισμα τα σύνολα breaking και not-breaking. Αυτή για κάθε περίπτωση υπολογίζει τις καμπύλες που κάνουν fit στα δεδομένα μας. Μια εικόνα των καμπύλων αυτών φαίνεται για την περίπτωση των breaking στην εικόνα [4.2] και για την περίπτωση των not-breaking στην εικόνα [4.3] για τις διαφορετικές κατηγορίες που έχουμε.



Εικόνα 4.2: Τα fits των συναρτήσεων στα delay points του breaking συνόλου. Εδώ έχουμε φιλτράρει και όσες καμπύλες εμφανίζουν μεγάλη διακύμανση.



Εικόνα 4.3: Τα fits των συναρτήσεων στα delay points του not-breaking συνόλου. Στην περίπτωση αυτή δεν έχουμε φιλτράρει όσες καμπύλες έχουν μεγάλη διακύμανση.

Η διαφορά των δύο αυτών διαγραμμάτων φαίνεται κυρίως στον άξονα x που δείχνει την εξέλιξη του μεγέθους με βάση τον χρόνο. Όπως φαίνεται, δεν υπάρχουν πολλές καμπύλες για κάθε περίπτωση. Το γεγονός αυτό συνεπάγεται αργότερα σε λιγότερα δεδομένα για το μοντέλο μηχανικής μάθησης αν θελήσουμε να φτιάξουμε ένα μοντέλο ανά ξεχωριστή κατηγορία. Τελικά αποφασίσαμε να φτιάξουμε ένα γενικό μοντέλο που δεν θα διαχωρίζει κατηγορίες.

[3] Αυτό γιατί τα αποτελέσματα του μοντέλου αργότερα δείχνουν σχετικά καλύτερη πρόβλεψη για την τιμή αυτή.

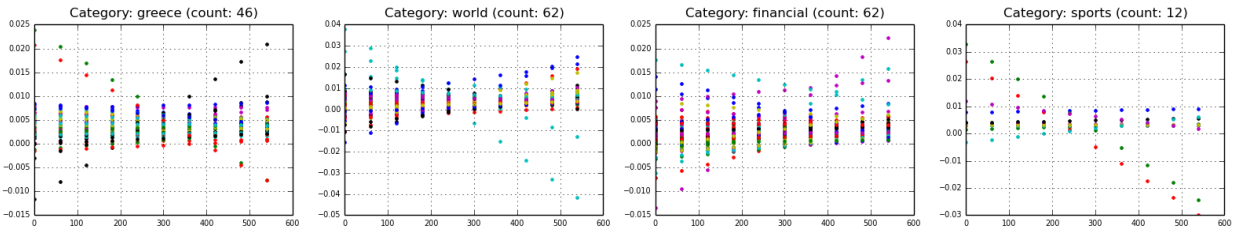
Finding the Slopes for each Curve

Για κάθε ένα από τα σύνολα έχουμε πλέον τις καμπύλες που ορίζουν την σχέση μεταξύ χρόνου και μεγέθους, δηλαδή, έχουμε τις $f(t) = M$ όπου το M είναι το μέγεθος cluster την χρονική αυτή στιγμή. Τώρα πρέπει απλά να ορίσουμε τα σημεία αυτά στα οποία θα ελέγχουμε την κλίση της καμπύλης για κάθε καμπύλη στα σύνολα μας. Όπως και με τις προηγούμενες παραμέτρους που έχουμε ορίσει, έτσι και εδώ, δεν υπάρχει εξαρχής κάποια ξεκάθαρη απάντηση για το πόσα και ποία πρέπει να είναι τα σημεία αυτά.

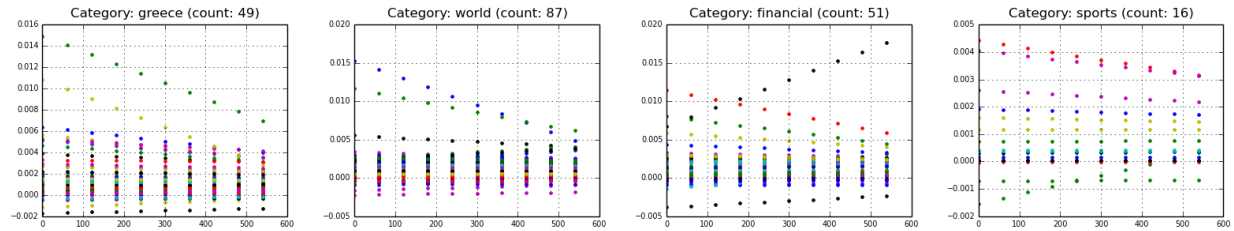
Αυτό που διαλέξαμε να κάνουμε εμείς ήταν να ορίσουμε ένα διάστημα *TIME_PERIOD*, σε λεπτά, στο οποίο κάθε *SLOPE_WINDOW* θα υπολογίζουμε την κλίση της καμπύλης. Η κλίση, που τυπικά συμβολίζεται με m, υπολογίζεται μέσω του κλασσικού της τύπου όπως φαίνεται στην εξίσωση [4.1]. Για το διάστημα $[0,$ TIME_PERIOD $*$ $60]$ υπολογίζαμε το m για το διάστημα *SLOPE_WINDOW* $* i$, SLOPE_WINDOW $* (i + 1)$ για $i = [1, \ldots,$ TIME_PERIOD$]$.

$$m = \frac{\Delta y}{\Delta x} = \frac{y_2 - y_1}{x_2 - x_1} \tag{4.1}$$

Οι τιμές που διαλέξαμε για τις παραμέτρους αυτές ήταν 10 λεπτά για το TIME_PERIOD με διάστημα ενός λεπτού σαν τιμή στο SLOPE_WINDOW. Με τις τιμές αυτές ορισμένες μπορούμε να καλέσουμε την `find_slopes()` μέθοδο που τις χρησιμοποιεί για να υπολογίσει τους ρυθμούς αύξησης. Άρα, τελικά, για κάθε καμπύλη, η μέθοδος επέστρεφε σε δέκα σημεία την κλίση της καμπύλης. Για την περίπτωση του breaking συνόλου τα σημεία αυτά φαίνονται στην εικόνα [4.4] ενώ για το σύνολο των not-breaking στην εικόνα [4.5].



Εικόνα 4.4: Οι ρυθμοί αύξησης για το σύνολο των breaking clusters. Στην εικόνα φαίνονται τα δέκα σημεία που υπολογίστηκαν.



Εικόνα 4.5: Οι ρυθμοί αύξησης για το σύνολο των not-breaking clusters. Στην εικόνα φαίνονται, και πάλι, τα δέκα σημεία που υπολογίστηκαν.

4.2.2 Δημιουργία Εισόδου-Εξόδου

Αφού έχουμε φτιάξει τα χαρακτηριστικά για τα breaking clusters και για τα not-breaking clusters, μπορούμε πλέον να δημιουργήσουμε και το σύνολο που θα αντιπροσωπεύει την είσοδο στο μοντέλο μηχανικής μάθησης. Ο λόγος που η υποενότητα ονομάζεται δημιουργία εισόδου **και** εξόδου είναι παρόμοιος με τον λόγο για τον οποίο έχουμε και τα δύο σύνολα. Κοινώς, επειδή ασχολούμαστε με κατηγοριοποίηση στα πλαίσια της επιβλεπόμενης μάθησης, πρέπει ως είσοδο να προσφέρουμε ένα σύνολο που έχει τα χαρακτηριστικά που ορίσαμε στην προηγούμενη ενότητα μαζί με την κατηγορία στην οποία ανήκουν [breaking, not-breaking]. Δηλαδή, όπως είχαμε αναφέρει και στο κεφάλαιο 2, πρέπει `Input` $= [\overrightarrow{X}, Y_i]$ όπου $\overrightarrow{X} = [X_1, X_2, \ldots, X_j]$ είναι το σύνολο των χαρακτηριστικών και $Y = \{Y_1, Y_2, \ldots, Y_i\}$ το σύνολο των κατηγοριών.

Το διάνυσμα $\overrightarrow{X}$ στην δικιά μας περίπτωση περιέχει τις δέκα τιμές των ρυθμών αύξησης, άρα, $\overrightarrow{X} = [\text{slope}_1, \text{slope}_2, \ldots, \text{slope}_j]$ με το $j = 1, \ldots, 10$. Το Y από την άλλη έχει δύο ορισμένες κατηγορίες, τις breaking και not-breaking. Επειδή οι classifiers δουλεύουν με αριθμητικά δεδομένα, δημιουργήσαμε μια αντιστοίχηση στην οποία:

- Breaking $\rightarrow 1$, δηλαδή, $Y = 1$ `if case breaking`.
- Not-Breaking $\rightarrow 0$, δηλαδή, $Y = 0$ `if case not-breaking`.

Τελικά η έξοδος που παράγει το μοντέλο καθώς και η τιμή που έπρεπε να συμπεριλάβουμε στα δεδομένα εκπαίδευσης έπαιρνε τιμές $Y = [0, 1]$.

Για να φτιάξουμε τα τελικά δεδομένα αυτό που πρέπει να κάνουμε είναι να περάσουμε τα δέκα σημεία που επέστρεφε η μέθοδος `find_slopes()`, και για τις δύο περιπτώσεις μας, στην μέθοδο `create_data_set()`. Εσωτερικά, αυτή αντιστοιχεί τα χαρακτηριστικά των συνόλων με την κατάλληλη τιμή τους ανάλογα με το σύνολο στο οποίο ανήκουν. Επιστρέφει το σύνολο $\overrightarrow{X}$ που περιέχει τα χαρακτηριστικά για κάθε cluster και το σύνολο Y που περιέχει τις αντίστοιχες κατηγορίες. Παράλληλα, επειδή το σύνολο των not-breaking είναι συνήθως πολύ μεγαλύτερο, φροντίζει να μειώσει το σύνολο σε τιμή περίπου ίση με αυτό του συνόλου breaking. Αυτό γίνεται για να αποφευχθούν μετέπειτα class impalances που μπορούν να επηρεάσουν την επίδοση του μοντέλου μας.[4].

[4]Ουσιαστικά το class imbalance σημαίνει πως η μία (η μία από τις κλάσεις) έχει πολλά περισσότερα στοιχεία από την άλλη (ή άλλες).

4.3 Predicting Breaking News

Στην ενότητα αυτή, φτιάχνουμε το μοντέλο μηχανικής μάθησης που θα χρησιμοποιήσουμε, το εκπαιδεύουμε και ελέγχουμε την ικανότητα του να προβλέψει σωστά τα clusters που θέλουμε. Κύρια βιβλιοθήκη που χρησιμοποιήθηκε εδώ ήταν η **scikit-learn** που προσφέρει πολλούς αλγορίθμους μηχανικής μάθησης και συνδέεται στενά με την **NumPy**. Ο κώδικας του classifier βρίσκεται στο αρχείο **bnClassifier** του πακέτου **breakingNews**.

Το Μοντέλο

Το μοντέλο/αλγόριθμος που χρησιμοποιήθηκε στην περίπτωση αυτή ήταν ο **Random Forest**. Ο αλγόριθμος αυτός, που ανήκει στην ειδική κατηγορία των ensemble αλγορίθμων, λειτουργεί με το να δημιουργεί ένα user defined σύνολο από decision trees. Κατά την στιγμή της αξιολόγησης ενός καινούργιου αντικειμένου ο αλγόριθμος χρησιμοποιεί τα decision trees που έχουν οριστεί για να προβλέψουν την κατηγορία του. Δηλαδή, αφού όλα τα decision trees έχουν κάνει μια πρόβλεψη της κατηγορίας στην οποία πιστεύουν ότι ανήκει το αντικείμενο, ο αλγόριθμος, για την περίπτωση της κατηγοριοποίησης, διαλέγει την κατηγορία η οποία ήταν πιο συχνή στο σύνολο των προβλέψεων που έκαναν. Πέρα από τα καλά επίπεδα ακρίβειας που παρουσιάζει, ο αλγόριθμος αυτός ενδείκνυται γιατί προσφέρει ένα ακόμα επίπεδο προστασίας ενάντιας το overfitting των δεδομένων.

Το μοντέλο αυτό εύκολα χρησιμοποιείται μέσω του πακέτου **scikit-learn**. Έχουμε πρόσβαση σε αυτόν μέσω του ensemble πακέτου με το όνομα *RandomForestClassifier*. Η μόνη παράμετρος που απαιτεί είναι ο αριθμός των δέντρων που θέλουμε να χρησιμοποιεί τον οποίο εμείς θέσαμε σε δέκα `ensemble.RandomForestClassifier(num_estimators=10)`.

Training and Testing

Πριν μπορέσουμε να χρησιμοποιήσουμε το μοντέλο μας για να προβλέψουμε, πρέπει να το εκπαιδεύσουμε χρησιμοποιώντας τα δεδομένα που δημιουργήσαμε. Το στάδιο της εκπαίδευσης (training phase) χρησιμοποιείται ώστε ο αλγόριθμος να μπορέσει να προσαρμοστεί σε μοτίβα των δεδομένων τα οποία για εμάς μπορεί να μην είναι εμφανή. Βλέποντας περιπτώσεις των μοτίβων μαζί με τις αντίστοιχες κλάσεις τους, οι αλγόριθμοι, μαθαίνουν τις αντιστοιχίσεις και μπορούν μελλοντικά να κατηγοριοποιήσουν σωστά καινούργια παρόμοια μοτίβα. Η εκπαίδευση του μοντέλου ήταν εύκολη μέσω της βιβλιοθήκης που χρησιμοποιούμε και απλά απαιτούσε κλήση της μεθόδου `fit(X,Y)` του RandomForestClassifier που είχαμε.

Η φάση του ελέγχου (test/validation phase) χρησιμοποιείται για να ελέγξουμε την απόδοση του μοντέλου μας. Η φάση αυτή μας δείχνει το πόσο ακριβές είναι το μοντέλο, τα συνολικά λάθη και τις επιτυχίες του (false/true positive/negative) δίνοντας μας την δυνατότητα να αναθεωρήσουμε για τις παραμέτρους που ορίζουμε ή/και για το μοντέλο που χρησιμοποιούμε. Για να εκτελέσουμε την φάση αυτή πρέπει βέβαια να δημιουργήσουμε το σύνολο δεδομένων χωρίς την κλάση τους για να μπορέσει το μοντέλο να επιχειρήσει να τις προβλέψει.

Στην πλειοψηφία των περιπτώσεων δεν χρησιμοποιούμε ένα test set από μόνο του για να ελέγξουμε την απόδοση αλλά, αντιθέτως, χρησιμοποιούμε μια τεχνική που ονομάζεται k-fold validation. Με την τεχνική αυτή μπορούμε, δίνοντας το σύνολο των δεδομένων, να εκπαιδεύσουμε και να ελέγξουμε την απόδοση του μοντέλου k φορές. Λειτουργεί με το να χωρίζει το σύνολο των δεδομένων σε k μέρη, να χρησιμοποιεί τα $k-1$ σύνολα για την εκπαίδευση του μοντέλου και το σύνολο που απομένει για έλεγχο. Την διαδικασία αυτή την πραγματοποιεί k φορές διαλέγοντας διαφορετικά σύνολα κάθε φορά. Μετα το πέρας της διαδικασίας υπολογίζονται οι μέσες αποδόσεις για όλες τις περιπτώσεις και παρουσιάζονται. Η τεχνική αυτή χρησιμοποιήθηκε στην δική μας περίπτωση με τιμή $k = 10$ όπως συνηθίζεται. Τα αποτελέσματα της θα παρουσιαστούν στην επόμενη υποενότητα.

Prediction Statistics

Τα αποτελέσματα του cross-validation παρουσιάζονται στην υποενότητα αυτή για διαφορετικές τιμές των παραμέτρων που έχουμε ορίσει. Σαν πρώτη περίπτωση, εξετάζουμε την επίπτωση της παραμέτρου *CURVE_DEGREE* που παρουσιάσαμε στην ενότητα [4.2]. Συγκεκριμένα, παρουσιάζουμε τις τιμές βασικών prediction statistics για τιμές του degree $= [1, 2, 3]$. Στους τρείς πίνακες που ακολουθούν, φαίνονται τα **recall, precision** και **f-score** για τις τιμές που μπορεί να πάρει το degree. Τα στατιστικά αυτά, ενώ συνδέονται, έχουν το καθένα διαφορετική αξία:

- **Recall**: Το recall, εν συντομία, είναι η ικανότητα του μοντέλου μας να μπορεί να βρει όλες τις σωστές κλάσεις.
- **Precision**: Το precision, η ικανότητα του μοντέλου να μην κατηγοριοποιεί ως σωστή μια κλάση η οποία είναι λάθος
- **F-Score**: Το f-score είναι ο αρμονικός μέσος των δύο προηγούμενων στατιστικών (recall, precision).

Τα αποτελέσματα αυτά φαίνονται ξεχωριστά για κάθε σύνολο (breaking, not-breaking) αλλά και σύνολικά για τα δύο σύνολα. Όπως φαίνεται, δεν έχουν τεράστια απόκλιση για τις διαφορες περι-

πτώσεις που εξετάζουμε.

Parameters: $deg = 1$, Influence Threshold $= 15$, delay points $= 5$				
Classifier Scores	Precision	Recall	f1-score	Support
Breaking Set	0.83	0.81	0.82	423
Not-Breaking Set	0.81	0.83	0.82	412
Averages/Total	0.82	0.82	0.82	835

Πίνακας 4.1: Αποτελέσματα για degree ίσο με 1

Parameters: $deg = 2$, Influence Threshold $= 15$, delay points $= 5$				
Classifier Scores	Precision	Recall	f1-score	Support
Breaking Set	0.86	0.83	0.85	423
Not-Breaking Set	0.82	0.85	0.84	412
Averages/Total	0.84	0.84	0.84	835

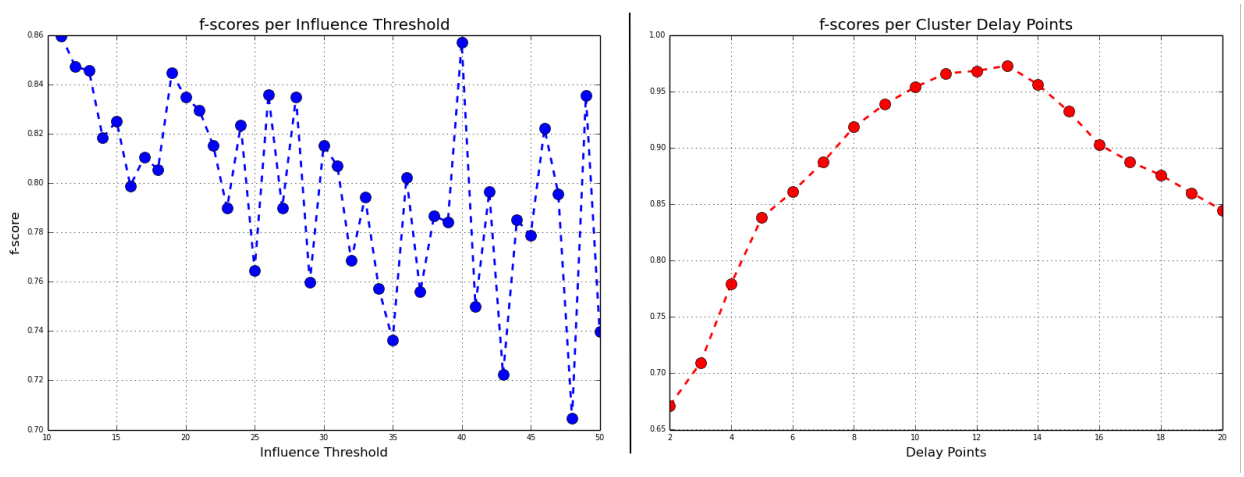
Πίνακας 4.2: Αποτελέσματα για degree ίσο με 2

Parameters: $deg = 3$, Influence Threshold $= 15$, delay points $= 5$				
Classifier Scores	Precision	Recall	f1-score	Support
Breaking Set	0.83	0.82	0.83	423
Not-Breaking Set	0.82	0.83	0.83	412
Averages/Total	0.83	0.83	0.83	835

Πίνακας 4.3: Αποτελέσματα για degree ίσο με 3

Parameter Tests

Όπως βλέπουμε, η περίπτωση για degree $= 2$ είναι, με μικρή διαφορά, η καλύτερη περίπτωση. Πλέον, διαλέγοντας το degree $= 2$ συνεχίζουμε εξετάζοντας δύο ακόμη παραμέτρους. Αρχικά, θέλουμε να δούμε πως εξελίζεται η τιμή του **f-score** συναρτήσει της παραμέτρου *INFLUENCE THRESHOLD*. Στο δεξιό plot της εικόνας [4.6] η σχέση αυτή φαίνεται για τιμές του threshold στο διάστημα $[11, 50]$. Από το αποτέλεσμα, βλέπουμε πως η σχέση μεταξύ των δύο είναι λίγο χαοτική χωρίς να υπάρχει κάποιο άμεσο correlation. Η μέγιστη τιμή παρατηρήθηκε για το threshold $= [11, 40]$ ενώ η μικρότερη στην τιμή 46.



Εικόνα 4.6: Το f-score συναρτήσει των παραμέτρων INFLUENCE_THRESHOLD (αριστερά) και DELAY_POINTS (δεξιά).

Η δεύτερη περίπτωση εξετάζει την σχέση της **f-score**, και πάλι, αλλά τώρα συναρτήσει με τις τιμές της παραμέτρου *DELAY_POINTS*. Το σύνολο τιμών που διαλέξαμε να εξετάσουμε για την Delay Points ήταν το $[2, 20]$ (αυθαίρετα). Τα αποτελέσματα του πειράματος αυτού φαίνονται και πάλι στην εικόνα [4.6] αλλά τώρα στο δεξιό της plot. Απ'ότι βλέπουμε, στην περίπτωση αυτή, υπάρχει ένας δυνατός συσχετισμός μεταξύ των τιμών. Μέχρι το τιμή 13, στην οποία παρατηρείται και η μέγιστη τιμή για το f-score, όσο αυξάνεται η τιμή των delay points τόσο αυξάνονται και οι τιμές για το f-score, δηλαδή, έχουμε έναν θετικό συσχετισμό (positive correlation). Από το σημείο αυτό, ο συσχετισμός αλλάζει και γίνεται αρνητικός. Επίσης μπορούμε να επαληθεύσουμε αυτό που είπαμε νωρίτερα, όσα περισσότερα delay points χρησιμοποιούμε, τόσο πιο έγκυρα θα μπορούμε να προβλέψουμε.

ΣΕΛΙΔΑ ΣΚΟΠΙΜΑ ΚΕΝΗ

ΚΕΦΑΛΑΙΟ 5

ΣΥΜΠΕΡΑΣΜΑΤΑ

5.1 Συμπεράσματα

Στην πτυχιακή αυτή εξετάσαμε αρκετές πτυχές της επιρροής και της σημαντικότητας στο πλαίσιο των ειδησεογραφικών δεδομένων. Αρχίσαμε με την επεξήγηση της ανάλυσης δεδομένων που κάναμε, παρουσιάσαμε σχέδια των πηγών μαζί με την επιρροή τους, μοντελοποιήσαμε τα δεδομένα ως γράφο με σκοπό την εκτέλεση λειτουργιών πάνω του και τελικά εξετάσαμε κατά πόσο μπορούμε να προβλέψουμε αν μια είδηση είναι breaking. Τα συμπεράσματα που μπορούμε να βγάλουμε είναι αρκετά, συγκεκριμένα, για κάθε περίπτωση μπορούμε να πούμε τα εξής.

Για την περίπτωση της επιρροής των πηγών, βρήκαμε τις πηγές οι οποίες επηρεάζουν τις υπόλοιπες σε μεγάλο βαθμό και λειτουργούν ως sources ειδήσεων, ενώ παράλληλα είδαμε τις πηγές οι οποίες επηρεάζονται το περισσότερο. Το ενδιαφέρον συμπέρασμα εδώ είναι πως δεν υπάρχουν πηγές που μόνο επηρεάζουν ή που μόνο επηρεάζονται αλλά, αντιθέτως, κάθε πηγή που ανήκει στο ένα από τα δύο σύνολα συνήθως ανήκει και στο άλλο. Παράλληλα είδαμε πως για κάθε πηγή υπάρχουν συγκεκριμένες κατηγορίες στις οποίες επηρεάζει το περισσότερο, δηλαδή, υπάρχουν κατηγορίες, που τις ονομάσαμε κυρίαρχες, που μπορούν να χαρακτηρίσουν θεματικά μια πηγή. Για κάθε από τις κατηγορίες είδαμε και τις πηγές που επηρεάζει, διαφορετικές για κάθε κατηγορία.

Για την περίπτωση του γράφου αρχίσαμε με το να αναπαραστήσουμε τα δεδομένα μας ως έναν. Στην συνέχεια, μέσω του PageRank βγάλαμε τις πηγές που λειτουργούν ως Hubs στο δικό μας δίκτυο και το ενδιαφέρον αποτέλεσμα εδώ, σχετικό με αυτό της προηγούμενης, είναι πως τα μεγαλύτερα hubs ήταν και αυτά τα οποία επηρέασαν τους περισσότερους. Στη συνέχεια προσπα-

θήσαμε να ομαδοποιήσουμε τα δεδομένα μας και συμπεράναμε πως υπάρχουν ομάδες πηγών που επηρεάζονται πολύ στα πλαίσια της. Το μέγεθος των ομάδων αυτό έχει άμεση συσχέτιση με το πόσο αυστηρό θέλουμε να είναι το επίπεδο επιρροής μεταξύ τους.

Με χρήση τεχνικών μηχανικής μάθησης, δημιουργήσαμε ένα random forest classifier που χρησιμοποιήσαμε για την πρόβλεψη του κατά πόσο ένα άρθρο είναι έκτακτο ή όχι. Είδαμε πως, ορίζοντας την κλίση της καμπύλης που περιγράφει το μέγεθος των clusters και εξάγοντας χαρακτηριστικά από την καμπύλη αυτή, μπορούμε αξιόπιστα να προβλέψουμε μελλοντικά στιγμιότυπα cluster. Παράλληλα κάναμε και μια μικρή ανάλυση της συσχέτισης μερικών παραμέτρων με την απόδοση του μοντέλου. Είδαμε πως για την περίπτωση του degree αυτή ήταν μικρή, για την περίπτωση του influence threshold ήταν χαοτική ενώ για τα delay points ήταν αρχικά θετική μέχρι ένα maximum και στη συνέχεια αρνητική.

5.2 Μελλοντική Δουλειά

Μελλοντικές κατευθύνσεις που μπορούν υπάρξουν είναι αρκετές. Για τους γράφους που φτιάξαμε μια ιδέα θα ήταν η εισαγωγή και του χρόνου στις ακμές του για την εξέταση των κόμβων εκείνων που επηρεάζονται από άλλους με γρήγορους ρυθμούς καθώς και για να δούμε πως και πόσο γρήγορα μεταδίδεται ένα άρθρο στο δίκτυο. Στα πλαίσια της μηχανικής μάθησης θα μπορούσε κάποιος να εξετάσει την επίπτωση και άλλων χαρακτηριστικών επάνω στο μοντέλο, πιθανότατα χρησιμοποιώντας χαρακτηριστικά εξαγόμενα από το κείμενο. Στην συνέχεια μπορεί να κάνει και ένα attribute evaluation για να δει την αξία των χαρακτηριστικών στα πλαίσια της πρόβλεψης.

Μια άλλη κατεύθυνση θα μπορούσε να ήταν η χρήση δεδομένων διαφορετικού τομέα. Δεδομένα από το twitter με retweets, δεδομένα από το facebook με shares και γενικότερα όποιο δίκτυο χρησιμοποιεί έντονα την λογική του sharing, θα μπορούσε να χρησιμοποιηθεί με την ίδια λογική για την εύρεση παρόμοιας γνώσης.

ΣΕΛΙΔΑ ΣΚΟΠΙΜΑ ΚΕΝΗ

ΒΙΒΛΙΟΓΡΑΦΙΑ

[1] Dimitris Fasarakis-Hilliard. Knowledge discover in news sphere [github code]. https://github.com/DimitrisJim/Influence.

[2] Graph-Tool. Fast and efficient graph library for python. https://graph-tool.skewed.de/.

[3] Google Inc. Google search engine. https://www.google.com.

[4] matplotlib. Efficient plotting library. http://matplotlib.org/.

[5] NumPy. Scientific computing library. http://www.numpy.org/.

[6] Palo. The news reading experience. http://www.palo.gr/.

[7] Pandas. Data analysis library. http://pandas.pydata.org/.

[8] Python. Python programming language. https://www.python.org/.

[9] ReportLab. Content to pdf solutions. http://www.reportlab.com/.

[10] Scikit-Learn. Simple and efficient tools for data mining and data analysis. http://scikit-learn.org/stable/.