ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΤΟΜΕΑΣ ΗΛΕΚΤΡΙΚΩΝ ΒΙΟΜΗΧΑΝΙΚΩΝ ΔΙΑΤΑΞΕΩΝ ΚΑΙ ΣΥΣΤΗΜΑΤΩΝ ΑΠΟΦΑΣΕΩΝ

Χρήση Crawlers για την εξαγωγή δεδομένων από το διαδίκτυο

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

της

Ελένης Καρανικόλα

Επιβλέπων : Δημήτριος Ασκούνης
Καθηγητής Ε.Μ.Π.

Αθήνα, Μάρτιος 2021

ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΈΑΣ ΤΕΧΝΟΛΟΓΙΑΣ ΠΛΗΡΟΦΟΡΙΚΗΣ
ΚΑΙ ΥΠΟΛΟΓΙΣΤΩΝ

Χρήση Crawlers για την εξαγωγή δεδομένων από το διαδίκτυο

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

της

Ελένης Καρανικόλα

Επιβλέπων : Δημήτριος Ασκούνης
Καθηγητής Ε.Μ.Π.

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 18 Μαρτιου του 2021.

.....	.....	.....
Δημήτριος Ασκούνης	Ιωάννης Ψαρράς	Χρυσόστομος Δούκας
Καθηγητής Ε.Μ.Π.	Καθηγητής Ε.Μ.Π.	Καθηγητής Ε.Μ.Π.

Αθήνα, Μάρτιος 2021

.....

ΕΛΕΝΗ ΚΑΡΑΝΙΚΟΛΑ

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

Περίληψη

Σκοπός της παρούσας διπλωματικής εργασίας είναι η κατασκευή μιας έξυπνης "πλατφόρμας" ειδησεογραφικού περιεχομένου και η διευκόλυνση της πρόσβασης του ανθρώπου σε ένα ευρύ φάσμα ειδήσεων και χρήσιμων περαιτέρω αναλύσεων.

Πρώτο στάδιο για την δημιουργία της πλατφόρμας αποτέλεσε η συλλογή των δεδομένων. Για τον λόγο αυτό επιλέχθηκαν 20 διαφορετικές ελληνικές ιστοσελίδες με ποικιλία ως προς το περιεχόμενο, την ιδεολογία και τα ενδιαφέροντα. Στην συνέχεια κατασκευάστηκαν 10 "αράχνες" και χρησιμοποιήθηκαν για την αυτοματοποιημένη συλλογή δεδομένων. Συνολικά συγκεντρώθηκαν 35.000 άρθρα από 10 κατηγορίες.Τα άρθρα αυτά στην συνέχεια χρησιμοποιήθηκαν για την ανάλυση δεδομένων.

Οι τρεις βασικοί άξονες της ανάλυσης των άρθρων ήταν ως προς το περιεχόμενο, την ποιότητα και τους συγγραφείς. Σημαντικό κομμάτι της ανάλυσης αποτελεί και η κατασκευή του κατηγοριοποιητή άρθρων. Για την κατασκευή του κατηγοριοποιητή χρησιμοποιήθηκε ο αλγόριθμος Naive Bayes, με ποσοστό επιτυχίας 68%. Επιπλέον κατασκευάστηκε ένα σύστημα προτάσεων όμοιων άρθρων βάση της ομοιότητας των λέξεων. Για κάθε άρθρο προτείνεται στην ιστοσελίδα τα 3 ομοιότερα ως προς αυτό άρθρα για ανάγνωση.

Στόχος της παραπάνω ανάλυσης είναι ο αναγνώστης να αποκτήσει μία ευρύτερη και οξύτερη κατανόηση πάνω σε όσα διαβάζει. Παράλληλα δίνεται η δυνατότητα πλοήγησης σε έναν ιστότοπο με πολύπλευρα ενδιαφέροντα και επιρροές, αποκλειστικά ειδησεογραφικού περιεχομένου, μετατρέποντας την εμπειρία ενημέρωση, στοχευμένη και γρήγορη.

Λέξεις Κλειδιά: Web Crawling, Big Data, Web Development, Naive Bayes, Smart Websites, classification, flask

Abstract

In our modern society thanks to the power of the internet, we come across too much information that we simply can't process. This thesis comes in to solve this exact problem by creating a smart news website.

First step in creating our platform was the collection of our data. For this cause we carefully chose 20 different greek websites with diverse content. Next step was the creation of a web crawler that scans through our chosen websites and collects data from their articles. The data were divided in 10 categories and we came up with the sum of total 35.000 articles. Furthermore we analyzed our data based on different parameters and came up with several metrics, we also created a classifier and a suggestion system for further reading with similar context.

For our data analysis we came up with three basic ideas: analyze content, quality and authors. Another powerful tool of the website is the article's classifier. Naive Bayes algorithm was used for its creation and has up to 68% for each new article.

Finally, our suggestion system is based on word similarity. Based on the words in the article our user chose to read we calculate and suggest three articles with the most similar words to continue his reading.

The goal of this project is making the readers understanding of the running news wider and deeper. We are providing a way to stay informed in multiple fields and interests quicker and more sufficient.

Keywords: Web Crawling, Big Data, Web Development, Naive Bayes, Smart Websites, classification, flask

Ευχαριστίες

Η παρούσα διπλωματική εργασία εκπονήθηκε στον τομέα Ηλεκτρικών Βιομηχανικών Διατάξεων και Συστημάτων Αποφάσεων της Σχολής Ηλεκτρολόγων Μηχανικων και Μηχανικών Υπολογιστών ΕΜΠ. Πρώτα θα ήθελα να ευχαριστήσω τον επιβλέπων καθηγητή μου κ. Δημήτριο Ασκούνη που μου έδωσε την ευκαιρία να διευρύνω τις γνώσεις μου πάνω στο θέμα εξαγωγή και ανάλυση δεδομένων από το διαδίκτυο. Επίσης θα ήθελα να ευχαριστήσω τον κ. Παναγιώτη Καψάλη, Υποψήφιο Διδάκτωρ ΕΜΠ, για την άψογη συνεργασία του και την πολύτιμη καθοδήγηση του, χωρίς την βοήθεια και την επιμέλεια του οποίου δεν θα ήταν δυνατή η ολοκλήρωση της διπλωματικής. Κλείνοντας δεν θα μπορούσα να παραλείψω τις ευχαριστίες στην οικογένεια μου, που με στήριξαν όλα αυτά τα χρόνια των σπουδών καθώς και των φίλων μου που υπήρξαν δίπλα μου σε κάθε βήμα.

Πίνακας περιεχομένων

1. Εισαγωγή

1.1. Αντικείμενο διπλωματικής

Στόχος της διπλωματικής εργασίας είναι η χρήση αυτοματοποιημένων μέσων για την συλλογή, της ανάλυση και την παρουσίαση μεγάλου όγκου άρθρων με απλό και εύχρηστο τρόπο. Απευθύνεται σε απλούς χρήστες χωρίς απαραίτητα εξειδικευμένες γνώσεις.

Πρώτο ζήτημα για την επίτευξη του στόχου ήταν η συλλογή του υλικού. Φυσικά ελεύθερα στο διαδίκτυο υπάρχει τεράστια ποικιλία ιστοτόπων, τόσο ξένων όσο και ελληνικών, τα οποία δημιουργούν καθημερινά νέο ειδησεογραφικό υλικό. Στα πλαίσια της εργασίας χρησιμοποιούμε αποκλειστικά ελληνικές ιστοσελίδες καθώς απευθυνόμαστε στον μέσο κάτοικο της Ελλάδας. Για την συλλογική συλλογή άρθρων προγραμματίσαμε έναν crawler με την βοήθεια του εργαλείου Scrapy και της γλώσσας προγραμματισμού python.

Δεύτερο στάδιο για την εκπόνηση της διπλωματικής ήταν η αποθήκευση των δεδομένων. Για τις ανάγκες της αποθήκευσης έγινε η χρήση της mySQL και του mySQL Workbench. Αφού σχεδιάσαμε τους πίνακες αποθήκευσης δεδομένων στην συνέχεια ενώσαμε την “αράχνη” απευθείας στην βάση ώστε όσα δεδομένα συγκεντρώνει να τα αποθηκεύει επιτόπου σε έναν τοπικό στον υπολογιστή μας χώρο.

Συνολικά συγκεντρώθηκαν πάνω από 35.000 άρθρα γεγονός που μας επέστρεψε στο αρχικό πρόβλημα του τεράστιο και κατ επέκταση μη επεξεργάσιμου όγκου πληροφοριών. Για την ανάγκη αυτή χρησιμοποιήθηκαν διαφορετικές μαθηματικές μετρικές ώστε να ποσοτικοποιήσουμε τα δεδομένα μας. Σκοπός της ποσοτικοποίησης και της μέτρησης των δεδομένων είναι η αντικειμενική και ξεκάθαρη ανάλυση των δεδομένων. Διαθέτουμε ανοιχτά τα αποτελέσματα των μετρήσεων και ο κάθε χρήστης είναι ελεύθερος να ερμηνεύσει και να

αξιοποιήσει τα δεδομένα στην διάθεση του, με όποιον τρόπο κρίνει εκείνος σοφότερο. Οι μαθηματικές αναλύσεις των άρθρων έγιναν πάνω στα jupyter notebooks.

Τέλος για την ολοκλήρωση της εργασίας κατασκευάστηκε η ιστοσελίδα παρουσίασης των άρθρων και όλων των παραγόμενων μετρήσεων. Για την κατασκευή της ιστοσελίδας χρησιμοποιήθηκε flask, python, javascript, html και css. Στην ιστοσελίδα παρουσιάζονται όλα τα άρθρα που συγκεντρώθηκαν από την αραχνη καθώς και όλες οι μετρήσεις που έγιναν.

1.2. Δομή Κειμένου

Στο πρώτο κεφάλαιο παρουσιάζεται περιληπτικά ο στόχος της διπλωματικής , τα βήματα που ακολουθήθηκαν και την εκπόνηση της καθώς και τα εργαλεία που χρησιμοποιήθηκαν σε κάθε βήμα.

Στο τρίτο κεφάλαιο γίνεται μία εμβάθυνση στα εργαλεία που αναφέρθηκαν στο κεφάλαιο ένα. Πρώτα παρουσιάζεται το εργαλείο Scrapy, οι βιβλιοθήκες που χρησιμοποιήθηκαν, η δομή των αραχνών με αντίστοιχες παραπομπές στο κώδικα. Πέρα από τα πρακτικά στοιχεία παρουσιάζονται και τα λογικά ζητήματα όπως ποιές ιστοσελίδες επιλέχθηκαν και γιατί, οι θεματικές κατηγορίες που ακολουθήθηκαν και η θεωρητική δομή των “αραχνών”. Στην συνέχεια παρουσιάζεται το MySQL Workbench και ο ρόλος του στην εργασία. Έπειτα γίνεται η παρουσίαση του θεωρητικού υποβαθρου που χρησιμοποιείται για την ανάλυση των άρθρων. Παρουσιάζεται το Jupyter Notebooks, ο αλγόριθμος Naive Bayes και η λογική των Similar Articles. Τέλος γίνεται η παρουσίαση του εργαλείου flask που χρησιμοποιήθηκε για την κατασκευή της ιστοσελίδας.

Στο τέταρτο κεφάλαιο δίνονται τα σχήματα αρχιτεκτονικής που παρήχθησαν κατά την εκπόνηση της διπλωματικής. Παρουσιάζεται η σχηματική απεικόνιση της αρχιτεκτονικής του Scrapy, το διάγραμμα ER της βάσης δεδομένων για την

αποθήκευση των άρθρων και τέλος έχουμε το σχήμα αρχιτεκτονικής της λειτουργίας των APIs.

Στο πέμπτο κεφάλαιο γίνεται μία εικονική πλοήγηση στον ιστότοπο που κατασκευάστηκε με την βοήθεια screenshots.

1.3. Πίνακας Εικόνων

1.4. *Πίνακας Πινάκων*

2. Βιβλιογραφία

2.1. Web Crawling with scrapy

Τα τελευταία χρόνια έχει αναπτυχθεί σημαντικά η τεχνολογία του web crawling. Web crawling ορίζουμε την κατασκευή ενός web crawler ή spider δηλαδή ενός internet bot που διατρέχει συστηματικά το διαδίκτυο και συγκεντρώνει δεδομένα. Χρησιμοποιείται συχνά σε εφαρμογές που εξαρτώνται από δεδομένα άλλων εφαρμογών για την λειτουργία τους όπως για παράδειγμα οι μηχανές αναζήτησης ή οποίες θέλουν να επιστρέφουν δεδομένα επίκαιρα. Η δύναμη της τεχνολογίας αυτής είναι πολύ μεγάλη καθώς πλέον τα δεδομένα είναι άφθονα στο διαδίκτυο. Έχει επίσης ανοίξει το θέμα συζήτησης για την "ευγενική" συμπεριφορά στο διαδίκτυο και έχει δημιουργηθεί και η ανάγκη για προστασία όσων ιστοσελίδων το επιθυμούν από τέτοια bot, προστασία τόσο για την χρήση των πόρων που κάνουν όσο και για τα δεδομένα που φέρνουν. Στα πλαίσια της διπλωματικής σεβαστήκαμε απόλυτα τις απαιτήσεις κάθε σελίδας και φέραμε δεδομένων μόνο από μέρη που το επέτρεπαν.

Πρώτο λοιπόν εργαλείο που χρησιμοποιήσαμε ήταν για την δημιουργία του Crawler, το scrapy framework. Το εργαλείο αυτό χρησιμοποιείται σε project που θέλουν να συγκεντρώσουν πολλά δεδομένα που βρίσκονται διαθέσιμα στο διαδίκτυο με έναν γρήγορο αυτοματοποιημένο τρόπο. Η λογική ενός crawler είναι ότι ορίζεις στο σώμα του μηχανισμού τις ιστοσελίδες από τις οποίες ενδιαφέρεσαι να πάρεις δεδομένα και συγκεντρώνεις την μορφή των δεδομένα που σε ενδιαφέρουν να βρεις. Είναι ένα σύγχρονο, ελεύθερου λογισμικού εργαλείο που γράφεται σε python και έχει χρησιμοποιηθεί σε πολλές αντίστοιχες εφαρμογές.

Στο παρελθόν έχει χρησιμοποιηθεί ερευνητικά σε πολλούς και διαφορετικούς τομείς . Όπως για παράδειγμα έχει χρησιμοποιηθεί για την εύρεση επίκαιρων θεμάτων που απασχολούν την Κίνα.[1] Για την ανακάλυψη παράνομων χρηστών που πουλούσαν ψεύτικα φάρμακα στο διαδίκτυο [2]. Για την προσπάθεια σύνδεσης επιστημονικών δεδομένων που υπάρχουν σε επιστημονικά άρθρα στο διαδίκτυο [3]. Σε κάθε περίπτωση χαρακτηριστικό της εφαρμογής είναι ότι τα δεδομένα υπάρχουν στο διαδίκτυο και προφανώς εντάσσονται στα πλαίσια των μεγάλων δεδομένων.

Συμπερασματικά η τεχνολογία του web crawling έχει συστηματοποιηθεί και βελτιστοποιηθεί σε πολύ μεγάλο βαθμό γεγονός που καθιστά την γνώση της σπουδαίο εργαλείο, ειδικά έχει στην σημερινή εποχή της άφθονης πληροφορίας και της ελεύθερης πρόσβασης σε αυτή.[4] Έχει αυτοματοποιήσει την διαδικασία συγκέντρωσης των δεδομένων βοηθώντας με τον τρόπο αυτό στην διαδικασία ανάλυσης τους και στην εξαγωγή πολύτιμων συμπερασμάτων από αυτές. Έχει ήδη χρησιμοποιηθεί σε σπουδαία έργα και έχει ακόμα πολλά να προσφέρει στην σύγχρονη επιστήμη

2.2. Classification with naive bayes algorithm

Classification είναι η λειτουργία της κατηγοριοποίησης, δηλαδή δεδομένης μιας οποιαδήποτε εισόδου, κατηγοριοποίηση είναι η διαδικασία πρόβλεψης της κατηγορίας, ο ορισμός ενός label στα δεδομένα. Ένα παράδειγμα κατηγοριοποίησης είναι η κατηγοριοποίηση ενός παρόχου ηλεκτρονικού ταχυδρομείου των email σε spam ή όχι. Αυτό αποτελεί μία μορφή κατηγοριοποίησης σε ένα δυϊκό σύστημα ταξινόμησης. Όλα τα δεδομένα επιτελούνται κατηγοριοποίησης σε δύο ή και περισσότερες κατηγορίες. Η διαδικασία της κατηγοριοποίησης χρησιμοποιεί διαφορετικούς αλγορίθμους για την πρόβλεψη της και σε γενικές γραμμές δεν υπάρχει κάποια συγκεκριμένη απάντηση ως προς το ποιά μέθοδος είναι καλύτερη. Η επιλογής και η απόδοσης της κατάλληλης μεθόδου εξαρτάται από την μορφή, των όγκο των δεδομένων καθώς και τις ανάγκες κάθε εφαρμογές και τις υπολογιστικές δυνάμεις που διατίθενται.

Στα πλαίσια της διπλωματικής για την κατηγοριοποίηση χρησιμοποιήθηκε ο αλγόριθμος naive Bayes. Ο αλγόριθμος Naive Bayes βασίζεται πάνω στο θεώρημα του Bayes και στην εποχή της μηχανικής μάθησης έχει γνωρίσει μεγάλη δημοτικότητα ως μία αξιόπιστη και γρήγορη μέθοδος για την πρόβλεψη αποτελεσμάτων με βάση μία μεγάλη βάση πληροφοριών. Είναι μία πιθανολογική μέθοδος και στηρίζεται στην παραδοχή ότι τα δεδομένα μας είναι ανεξάρτητα. Το πρόβλημα αυτού του αλγορίθμου είναι ότι δεν μπορεί να παράξει πρόβλεψη όταν η πιθανότητα είναι μηδέν για κάποια συγκεκριμένη τιμή, το πρόβλημα αυτό λύνεται μόνο με Laplace estimator. Η περίπτωση αυτή δεν υπάρχει στα δεδομένα μας. Στην εφαρμογή χρησιμοποιείται για την πρόβλεψη της κατηγορίας ενός άρθρου όταν αυτή για κάποιο λόγο δεν είναι γνωστή. Ο προβλέπτης κατατάσει το άρθρο σε μία από τις 10 κατηγορίες που έχουμε ορίσει.

Η κατηγοριοποίηση των δεδομένων βάση του naive Bayes έχεις χρησιμοποιηθεί σε πολλές εφαρμογές. Έχει χρησιμοποιηθεί για την κατηγοριοποίηση δεδομένων από το twitter ως προς το χροιά των περιεχομένων του, αν είναι δηλαδή θετικά ή αρνητικά [48]. Στην κατηγοριοποίηση ιατρικού περιεχομένου δεδομένα σε κατηγορίες. [49] Στην κατηγοριοποίηση

μεγάλης κλίμακας κινεζικών κειμένων από το διαδίκτυο [50]. Τον εντοπισμό λεκτικά επιθετικού κειμένου από το διαδίκτυο [51] . Μερικές από τις εφαρμογές που έχουν χρησιμοποιήσει τον αλγόριθμο για κατηγοριοποίηση κειμένων είναι Syskill & Webert, NewsDude, Daily Learner, LIBRA και ITR.[52]

Το θεώρημα του Bayes λέει ότι η πιθανότητα του A ενδεχόμενου δεδομένου του B, ισούται με την πιθανότητα του B ενδεχόμενου δεδομένου του A, επί την πιθανότητα του A δία την πιθανότητα B: [5]

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

Ο naive Bayes classifier θεωρείται αποδοτικός ειδικά σε περιπτώσεις με ελλιπή δεδομένα καθώς επίσης και η απόδοση του αυξάνεται σημαντικά καθώς αυξάνονται τα δεδομένα. Από την μία το φαινόμενο έλλειψης δεδομένων είναι αρκετά συνηθισμένο σε πρακτικές εφαρμογές, εκεί έγκειται και η δύναμη της μεθόδου του.[6] Από την άλλη πάλι το γεγονός ότι η απόδοση του αυξάνεται με την αύξηση της πληροφορίας των καθιστά ένα πολύ δυνατό εργαλεία για μεγάλες βάσεις δεδομένων.[7] Στην περίπτωση της διπλωματικής δεν εφαρμόστηκε πάνω σε ελλιπή δεδομένα παρόλα αυτά θα μπορεί ο κατηγοριοποιητής να επιστρέψει καλές προβλέψεις και σε αυτή την περίπτωση αν χρειαστεί. Εφαρμόστηκε όμως σε μεγάλα δεδομένα για τον λόγο αυτό και έχουμε αρκετά καλή πρόβλεψη.

Ο αλγόριθμος του Naive Bayes έχει χρησιμοποιηθεί ευρέως και σε άλλες εφαρμογές πρόβλεψης κατηγοριών όπως στην πρόβλεψη κατηγοριών για δεδομένα από το twitter. [8] και η πρόταση της προβλεπόμενης κατηγορίας στους χρήστες. Επιπρόσθετα έχει αποδειχθεί χρήσιμος σε μοντέλα πρόβλεψης κινδύνου επιτυχίας προγραμμάτων λογισμικού. [9] Συνεπώς ο κατηγοριοποιητής βασισμένος πάνω στο θεώρημα του Naive Bayes είναι ένα απλό αλλά πολύ δυνατό εργαλείο και έχει γνωρίσει χρησιμότητα σε πολλούς τεχνολογικούς τομείς.

2.3. Similar Articles

Η ιδέα για τα όμοια άρθρα στηρίζεται στην λογική μοντελοποίησης λεκτικών δεδομένων. Στην ουσία είναι ένας μαθηματικός τρόπος για να μετρήσουμε και συγκρίνουμε δεδομένα κειμένου.[10] Η πληροφορία από ψηφιακά κείμενα είναι τεράστια και είναι σε μία μορφή που δεν μπορεί να κατανοήσει ο υπολογιστής, διανύουμε επομένως την εποχή που μαθαίνουμε στα μηχανήματα να επεξεργάζονται και τέτοιου τύπου δεδομένα καθώς υπάρχει όλο και μεγαλύτερη ανάγκη για την εκμετάλλευση τους.

Ένα σύνηθες πρόβλημα που χρήζει αντιμετώπισης είναι η κατασκευή του συστήματος προτάσεων με βάση τις επιλογές του χρήστη.[11] Ή αλλιώς ο ορισμός της έννοιας ομοιότητας κειμένων. Δεδομένου ότι στην περίπτωση της παρούσας εργασίας τα δεδομένα μας είναι άρθρα, η ομοιότητα τους έγκειται στην ομοιότητα των λέξεων τους επομένως θεωρούμε ως όμοια άρθρα τα άρθρα που εχουν κοινές λέξεις. Προφανώς, στην σύγχρονη γλώσσα υπάρχει μία τεράστια γκάμα λέξεων που εμφανίζονται πολύ συχνά όπως τα άρθρα, επομένως τέτοιες λέξεις πρέπει να αγνοηθούν καθώς αποτελούν λέξεις χωρίς πληροφορία.[12]

Πρώτο βήμα επομένως για κάθε τύπου ανάλυσης λεκτικών δεδομένων είναι ο καθαρισμός των περιττών λέξεων.[13] Ως περιττές λέξεις θεωρούμε όπως προαναφέραμε τα άρθρα οπως επίσης και τα ρήματα, τις συνδετικές λέξεις και τις αντωνυμίες. Στα πλαίσια της καλύτερης ανάλυσης αφεραίθηκαν μικρού μήκους λέξεις, αριθμοί, ειδικοί χαρακτήρες, συνηθέστερες λέξεις καθώς και αυτά δεν αποτελούν καλό κριτήριο ανάλυσης. Με τον καθαρισμό αυτό επιτυγχάνεται η ανάλυση δεδομένων που πλέον έχουν πολύτιμη πληροφορία για την ανάλυση μας.

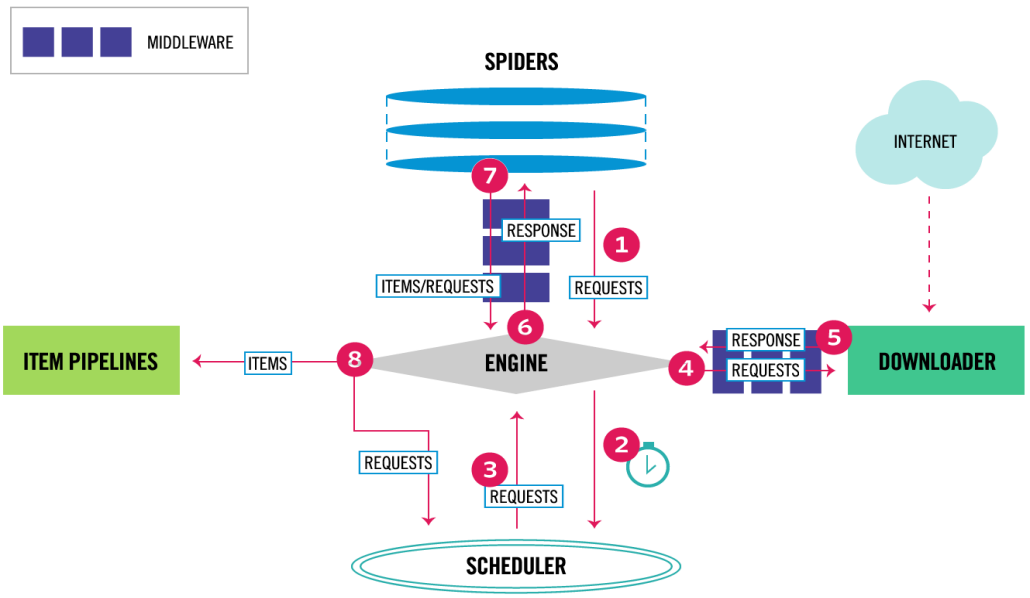
Δεύτερο βήμα μετά τον καθαρισμό των περιττών λέξεων από τα άρθρα είναι η αναζήτηση ομοιότητας λέξεων στα νέα δεδομένα. Για την επίτευξη αυτού του στόχου αναζητήσαμε σε όλα τα άρθρα ξεχωριστά να βρούμε ποιά άλλα άρθρα μοιράζονται τις ίδιες λέξεις και από αυτά τα τρία κοντινότερα τα προτείνουμε στον χρήστη ως καλές επιλογές για την συνέχεια της ανάγνωσης του.

Παρόμοια συστήματα προτάσεων έχει αποδειχθεί ότι μπορεί να προσφέρουν μεγάλα κέρδη όταν αξιοποιούνται σωστά από επιχειρήσεις. Παραδείγματα πλατφορμών που χρησιμοποιούν τέτοια συστήματα προτάσεων είναι το LinkedIn, Workable, Netflix, Google, Twitter και πάρα πολλές άλλες[53] οι οποίες έχουν καταφέρει με εξελιγμένους αλγορίθμους προτάσεων όμοιων προϊόντων ή υπηρεσιών να κερδίζουν τον πελάτη και να αυξάνουν τα κέρδη τους.

3. Εργαλεία

3.1. Scrapy

Για την συλλογή των άρθρων από το διαδίκτυο χρησιμοποιήθηκε το εργαλείο Scrapy. Το Scrapy είναι ένα open source framework γραμμένο σε python. Συγκεκριμένα χρησιμοποιούμε την κλάση CrawlSpider και την βιβλιοθήκη linkextractors. Η κλάση αυτή μας δίνει την δυνατότητα να ορίσουμε κανόνες (Rules). Με τους κανόνες αυτούς θέτουμε τα αναγνωριστικά στα url που θα επισκεφτεί η αράχνη μας. Η σχηματική αναπαράσταση της δομής του Scrapy Framework:



Εικόνα 3.1

Από το σχήμα παραπάνω παρατηρούμε ότι πρώτα δέχεται τα requests για crawl το Spider. Αφού δεχτεί τα αιτήματα τα προωθεί στον scheduler ο οποίος είναι υπεύθυνος για την διαχείριση αυτών των αιτημάτων και από αυτόν παίρνει τα next requests που θα κάνει crawl. Στην συνέχεια ο Scheduler στέλνει τα next request στο Engine. Ο Engine με την σειρά του στέλνει τα request στον Downloader αφού περάσει από τον Downloader Middleware. Μόλις κατέβει η σελίδα ο Downloader γεννά ένα Response με την σελίδα αυτή και την επιστρέφει στο Engine περνώντας πρώτα από το Downloader Middleware. Ο Engine παίρνει το response και το στέλνει στον Spider για επεξεργασία αφού περάσει πρώτα από το spider middleware. Έπειτα ο Spider παίρνει το response το επεξεργάζεται και επιστρέφει στον Engine νέα requests να ακολουθήσουν καθώς και αντικείμενα που πήρε από την σελίδα. Τέλος ο Engine παίρνει τα αντικείμενα που επέστρεψε ο Spider και τα στέλνει στο Item Pipelines και τα νέα requests στον Scheduler. Η διαδικασία επαναλαμβάνεται από το βήμα 1 μέχρι να μην υπάρχουν άλλα requests στον Scheduler.

Η λογική που ακολουθήσαμε για τις ανάγκες της εργασίας είναι ότι κάθε ιστότοπος έχει συγκεκριμένη δομή, δεδομένου λοιπόν ότι η δομή είναι σταθερή ορίσαμε από πιο σημείο του html αρχείου να τραβήξει δεδομένα η αράχνη μας.[14]

Από κάθε ιστοσελίδα συλλέγουμε:

1. Ημερομηνία έκδοσης.
2. Κατηγορία του άρθρου.
3. Υποκατηγορία του άρθρου.
4. Τίτλο.
5. Κείμενο του άρθρου.
6. Συγγραφέα.
7. Το όνομα της ιστοσελίδας.
8. url της ιστοσελίδας.

Από προγραμματιστική σκοπιά η δομη κάθε αράχνης αποτελείται:

- allowed domains: Με τον τρόπο αυτό ορίζουμε τους ιστότοπους που επιτρέπεται να επισκεφτεί η αραχνη.
- start_url: Ορίζονται τα url από όπου θα ξεκινήσει η αναζήτηση δεδομένων και νέων url για να επισκεφτεί.
- rules: Είναι οι κανόνες που πρέπει να ισχύουν σε ένα url για να το επεξεργαστεί η αράχνη μας.

Ο κώδικας για την εγκατάσταση και εκτέλεση του Crawler βρίσκονται διαθέσιμα στο GitHub[1] με αναλυτικές οδηγίες εγκατάστασης.

Για τις ανάγκες της εργασίας κατασκευάστηκαν 10 αράχνες. Κάθε αράχνη ασχολήθηκε με διαφορετική θεματολογία. Ένα από τα προβλήματα που προέκυψαν είναι ότι ανάλογα τον server κάθε ιστοσελίδας κάποιες ιστοσελίδες μονοπωλούσαν την αράχνη και επέστρεφαν περισσότερα δεδομένα από άλλες. Συνδυαστικά με αυτό το πρόβλημα ήταν και το πρόβλημα του πλήθους των άρθρων, δηλαδή κάποιες ιστοσελίδες είχαν άρθρα απο το 2009 γεγονός που έκανε το τελικό μας σύνολο ανομοιογενές το οποίο στη συνέχεια θα αλλοίωνε την ανάλυση μας.

Για το πρώτο πρόβλημα ορίσαμε στα settings το DOMAIN_DEPTHS θέτοντας για κάποιες ιστοσελίδες ξεχωριστά μεγέθη και στην συνέχεια χρησιμοποιούσαμε την μεταβλητή αυτή στο middleware ώστε να ελέγχουμε το βάθος που έμπαινε η αράχνη σε κάθε ιστοσελίδα. Κάποιες ιστοσελίδες είχαν ανάμεικτα τα άρθρα τους ανά κατηγορία γεγονός που δημιουργούσε την ανάγκη η αράχνη να κρατάει βάθος βαθμού 1 προκειμένου να βρίσκει πιο αποτελεσματικά την κατηγορία άρθρων που οριζόταν στο start_urls.

Για το πρόβλημα απενεργοποίησης ενός Rule μόλις έβρισκε κάποιο συγκεκριμένο πλήθος άρθρων, συγκεκριμένα μόλις έβρισκε 300 άρθρα ανά κατηγορία και ιστοσελίδα, επεκτείναμε τον Rule ορίζοντας μία συνάρτηση process_request. Η συνάρτηση που καλούμε εκεί έτρεχε πριν από την συνάρτηση αναζήτησης στην ιστοσελίδα και αν ο μετρητής μας ήταν

[1] https://github.com/elenisproject/NewsCrawler

κάτω από 300 επέστρεφε το IgnoreRequest αλλιώς έκανε raise ένα error οπότε σταματούσε και το crawling.[15]

Τελευταίο στάδιο για την δημιουργία και την εκτέλεση της αράχνης ήταν η σύνδεση μας με την βάση δεδομένων που θα τα αποθηκεύσαμε. Προφανώς πρώτα σχεδιάστηκε και κατασκευάστηκε η βάση δεδομένων μας και έπειτα εκτελέσαμε την σάρωση με τους crawlers. Η σύνδεση με την βάση δεδομένων έγινε μέσα από pipelines.py[2] και με την βοήθεια της βιβλιοθήκης mysql.connector. Ορίσαμε τα στοιχεία της βάσης δεδομένων που θέλουμε να συνδεθούμε και το query που θέλουμε να εκτελέσουμε, το οποίο εισήγαγε τα αντικείμενα που επέστρεφε η αράχνη στις σωστές θέσεις του πίνακα μας.[16]

3.2. MySQL Workbench

Για την αποθήκευσή των άρθρων μας χρησιμοποίηθηκε η γλώσσα MySQL και το εργαλείο MySQL Workbench[17]. Το εργαλείο MySQL Workbench ενεργοποιεί μια τοπική βάση δεδομένων στον υπολογιστή και σε βοηθάει στην ανάπτυξη και την σχεδίαση της. Επίσης έχει την δυνατότητα να δημιουργήσει και να οπτικοποιήσει ER models.[18] Βοηθάει στην εξοικονόμηση χρόνου και στην καλύτερη εποπτεία της τελικής βάσης που έχει υλοποιηθεί. Συγχρόνως δίνει την δυνατότητα εκτέλεσης queries, ελέγχου την απόδοση της και του περιεχομένου της. [19]. Στα πλαίσια της εργασίας κατασκευάσαμε την βάση δεδομένων[3] μας πριν την εκτέλεση του Crawler.[20]

[2] https://github.com/elenisproject/NewsCrawler/blob/master/NewsCrawler/pipelines.py
[3]https://github.com/elenisproject/NewsBackend/blob/master/Database_Configuration/create_table_articles.sql

Screenshot από την βάση δεδομένων για τον πίνακα articles:

id	topic	subtopic	website	title	article_date	author	article_body	url
36588	World	World	newsit.gr	Βρε…	2020-05-25	newsit.gr	Τη λύπη του…	https://www.n…
36589	World	World	newsit.gr	Κορ…	2020-05-25	newsit.gr	Σε δεύτερη…	https://www.n…
36590	World	World	newsit.gr	Νέα…	2020-05-26	newsit.gr	Αυξήθηκαν τ…	https://www.n…
36591	World	World	newsit.gr	Ξεπέ…	2020-05-26	newsit.gr	Δραματικός…	https://www.n…
36592	World	World	newsit.gr	Κορ…	2020-05-26	newsit.gr	Επτά νέα κρ…	https://www.n…
36593	World	World	newsit.gr	Κρα…	2020-05-26	newsit.gr	Το πιο φημισ…	https://www.n…
36594	World	World	newsit.gr	Βίντ…	2020-05-26	newsit.gr	Εντύπωση πρ…	https://www.n…
36595	World	World	newsit.gr	Καν…	2020-05-26	newsit.gr	Το Κινηματο…	https://www.n…
36596	World	World	newsit.gr	Μει…	2020-05-26	newsit.gr	Ο απόλυτος…	https://www.n…
36597	World	World	newsit.gr	Ραγ…	2020-05-26	newsit.gr	Άλλους 19 θ…	https://www.n…

Εικόνα 3.2.1

Screenshot από την βάση δεδομένων για τον πίνακα similar_articles:

id	first_article	second_article	third_article	fourth_arti…	fifth_article
36590	63673	71353	63514	45518	63697
36591	55708	63673	63514	63697	45608
36592	63673	36843	63697	55708	36780
36593	58852	58877	58907	58971	59033
36594	46343	48454	51509	59153	59155
36595	60042	63874	77737	46688	45608
36596	71353	63673	63514	36676	55708
36597	55708	63673	63514	63657	36696

Εικόνα 3.2.2

3.3. Data Analysis

Τα δεδομένα που αποθηκεύονται στην βάση δεδομένων είναι όπως αυτά συγκεντρώθηκαν από τους crawlers. Σημαντικό όμως κομμάτι της εργασίας είναι η ανάλυση των λεκτικών δεδομένων και η εξαγωγή των ουσιωδών πληροφοριών. Για την διαδικασία αυτή χρειάζεται ο ‘‘καθαρισμός’’ των λεκτικών μας πληροφοριών.[26] Ένας πρώτος καθαρισμός ως προς την όψη τους έγινε από τον ίδιο τον crawler με την βοήθεια των regex. Δεύτερο στάδιο ήταν ο καθαρισμός με python script και με την βοήθεια των βιβλιοθηκών της python.

Αρχικό φίλτρο ήταν η κανονικοποίηση των λέξεων. Με κανονικοποίηση των λέξεων αναφερόμαστε στην αφαίρεση, των αριθμών, των ειδικών χαρακτήρων, των λατινικών γραμμάτων, των τόνων, και των κεφαλαίων. Συγκεντρώθηκαν επομένως όλα τα δεδομένα σε ελληνικούς, απλούς, μικρούς χαρακτήρες[4]. Δεύτερο φιλτράρισμα αποτέλεσε η αφαίρεση των περιττών λέξεων. Για να βρούμε τις περιττές λέξεις θέσαμε κάποια κριτήρια. Θεωρήθηκαν ως περιττές οι λέξεις με την συχνότερη εμφάνιση. Επομένως κατασκευάσαμε ένα script το οποίο μετράει τον αριθμό εμφάνισης όλων των λέξεων των δεδομένων μας στο σώμα των δεδομένων μας. Από εκεί κρατήσαμε τις 500 συνηθέστερες ως φίλτρο. Ένα άλλο κριτήριο θεωρήθηκε το μήκος της λέξεις. Κατασκευάστηκε ένα δεύτερο script με τις λέξεις που δεν ξεπερνούν τις 3 συλλαβές και προστέθηκαν και αυτές στο φίλτρο των περιττών λέξεων[5]. Στο ίδιο σύνολο προσθέσαμε λέξεις που βρήκαμε στο διαδίκτυο ως stop_words. Δηλαδή ως λέξεις που θεωρούνται περιττές και τέλος προστέθηκαν κάποια ρήματα όπου εμπειρικά κρίθηκαν ότι δεν προσφέρουν κάποια επιπλέον πληροφορία στο περιεχόμενο. Συνδυαστικά όλα τα παραπάνω χρησιμοποιήθηκαν για τον καθαρισμό των δεδομένων.[6]

---

[4] https://github.com/elenisproject/NewsBackend/blob/ac69ceff437ecc234026ded00d60d3d0f0e83a49/utilities.py#L33

[5] https://github.com/elenisproject/NewsBackend/blob/ac69ceff437ecc234026ded00d60d3d0f0e83a49/utilities.py#L53

[6] https://github.com/elenisproject/NewsBackend/blob/ac69ceff437ecc234026ded00d60d3d0f0e83a49/utilities.py#L38

Μετά την διαδικασία καθαρισμού έγινε η εξαγωγή και η αποθήκευση των δεδομένων τοπικά σε ένα αρχείο csv.[25] Από το αρχείο αυτό έγινε περαιτέρω ανάλυση πάνω στα δεδομένα για την εξαγωγή συμπερασμάτων και μετρήσεων. Επίσης η ίδια μέθοδος καθαρισμού γίνεται στην είσοδο των άρθρων στον classifier για την πρόβλεψη της κατηγορίας τους. Επομένως είναι ένα πολύ δυνατό και χρήσιμο εργαλείο σε όλο το project.

3.4. Jupyter Notebooks

Για την ανάλυση των δεδομένων[21] χρησιμοποιούμε το jupyter notebook. Το jupyter notebooks είναι ένα open source ιστότοπος που σου δίνει την δυνατότητα να γράψεις και να μοιραστείς ζωντανά κώδικα. Στα πλαίσια της εργασίας το εργαλείο αυτό χρησιμοποιήθηκε για την ανάλυση των δεδομένων μας και για την οπτικοποίηση των αποτελεσμάτων.[22] Επίσης ο κώδικας αυτούσιος είναι διαθέσιμος να τον τρέξει ο κάθε χρήστης στην ιστοσελίδα μας.

Η εργασία υλοποιήθηκε σε περιβάλλον MacOS. Για το installation του jupyter notebooks χρησιμοποιήθηκε το homebrew. Επομένως πρώτο βήμα είναι να ανοίξουμε το terminal και να τρέξουμε την εντολή:

$ /bin/bash -c "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/HEAD/install.sh)"

Αφού ολοκληρωθεί η εγκατάσταση του homebrew πρέπει να εγκατασταθεί η python στο περιβάλλον του υπολογιστεί μας επομένως στο ίδιο terminal εκτελούμε

$ brew install python

Εφόσον όλα είναι έτοιμα κάνουμε install και το jupyter notebooks εκτελώντας στο terminal

$ brew install jupyter

Για να ξεκινήσει το jupyter notebook τρέχουμε

$ jupyter notebook

Εφόσον τρέχουν τα jupyter notebooks, επόμενο βήμα είναι η ανάλυση των δεδομένων. Ως προς τα δεδομένα μελετήθηκαν:[23][24]

1. Μέσο μέγεθος άρθρου ανά κατηγορία.
2. Μέτρηση και ποσοτικοποίηση άρθρων σε καλά και δυσάρεστα νέα.
3. Αναζήτηση των χωρών που συζητούνται.
4. Συνηθέστερες λέξεις ανά κατηγορία.
5. Εύρεση δημοφιλέστερων πηγών πληροφοριών.
6. Δημοφιλέστεροι αρθρογράφοι.
7. Ποσοστό της ειδησεογραφίας για τους δημοφιλέστερους συγγραφής.
8. Παρακολούθηση δημοτικότητας λέξεων στον χρόνο.
9. Ποιότητα αρθρογραφίας.
10. Δημοφιλέστερες λέξεις ανά χρόνο.

Η ανάλυση και η οπτικοποίηση των δεδομένων στα notebooks έγινε πάνω στα "καθαρά" δεδομένα όπως περιγράφονται στην παράγραφο 3.3 από το τοπικά αποθηκευμένο csv αρχείο. Επίσης σε κάθε βήμα που χρησιμοποιούμε νέα δεδομένα όπως για παράδειγμα στην εύρεση των πιο συζητημένων χωρών, η ίδια κανονικοποίηση έγινε και στα δεδομένα εκείνα που εισήχθησαν μετέπειτα στα πλαίσια την ανάλυσης.

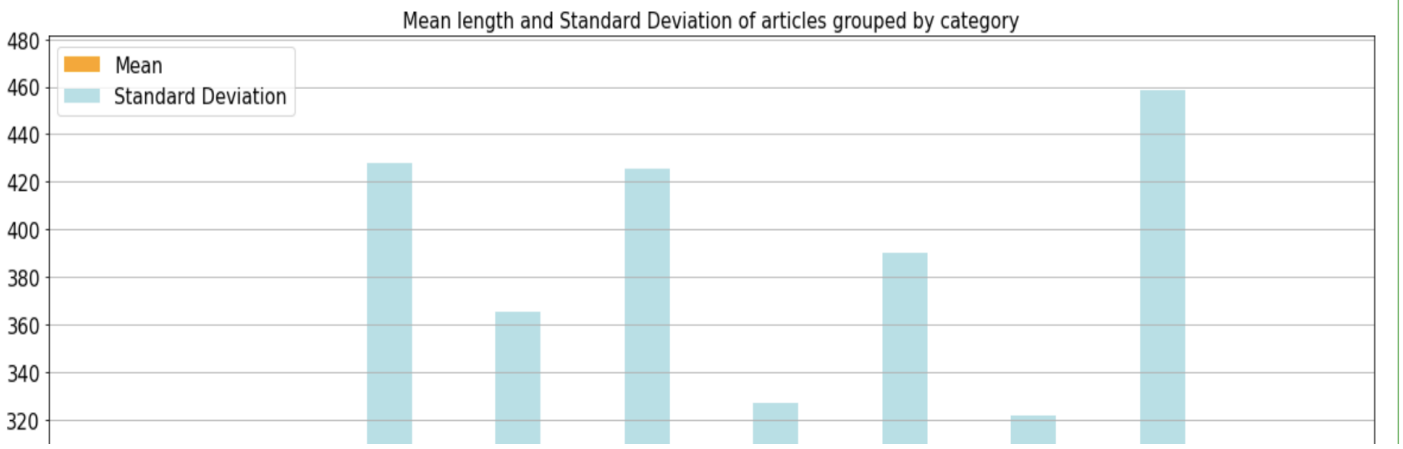
Για κάθε ένα από τα δέκα ερωτήματα δημιουργήσαμε ένα ξεχωριστό Notebook εισάγωντας εκεί τα δεδομένα και τις βιβλιοθήκες που θα χρειαζόμασταν. Μετά την ανάλυση των δεδομένων τα αποτελέσματα τα χρησιμοποιούμε για την κατασκευή γραφημάτων. Η υλοποίηση των γραφημάτων έγινε με χρήση του εργαλείου **matplotlib** και **wordcloud.**[27] Με την βοήθεια αυτών κατασκευάσαμε pies, barcharts, stacked barcharts για να παρουσιάσουμε τις μετρικές που προέκυψαν.[28][29]

Για να τρέξουν τα notebooks μέσα από το website έπρεπε πρώτα να ορίσουμε στον φάκελο jupyter_notebook_config.py τα στοιχεία της ιστοσελίδας μας ώστε να έχουμε πρόσβαση στα notebooks μέσω του tag iframe. Με τον τρόπο αυτό μπορεί ο κάθε χρήστης να τρέξει κάθε κομμάτι του notebook ξεχωριστα μέσα από την ιστοσελίδα.

Ενδεικτικά όταν ανοίγει η ιστοσελίδα:

The main code:

In [4]:
```
#List unique values in the df['topic'] column
categories = list(df.topic.unique())
#list with the average length
average = []
#list with the standar deviation of the articles
len_articles = []

#print the charts
for category in categories:
    # call length_category to get the mean of each category and a list of their articles lengths
    average_std = length_category(category)
    #save the mean to the average list
    average.append(average_std[0])
    #calculate the standard deviation of each category and save it to len_articles list
    len_articles.append(stdev(average_std[1]))

plot_barchart(average,len_articles,categories)
```



Εικόνα 3.3

Προκειμένου να τρέξει ο χρήστης μόνος του τον κώδικα πρέπει να πατήσει το κουμπί run σε κάθε code block με την σειρά που εμφανίζονται στο notebook.

3.5. Classification Tool using Naive Bayes

Σαν συνέχεια στην ανάλυση δεδομένων μας κατασκευάσαμε έναν κατηγοριοποιητή[7]. Ο κατηγοριοποιητης δέχεται ως είσοδο ένα άρθρο και με βάση τις κατηγορίες που ορίσαμε στα άρθρα της βάσης μας, το εντάσσει σε μία κατηγορία. Για την πρόβλεψη του χρησιμοποιείται ο αλγόριθμος Naive Bayes και τεστάροντας τον βρήκαμε ότι έχει ποσοστό επιτυχίας 63%. [30]

Για την υλοποίηση του αλγορίθμου ακολουθήσαμε την παρακάτω διαδικασία. Βρήκαμε τις 100 δημοφιλέστερες λέξεις από τα δεδομένα μας. Εννοείται όταν λέμε δεδομένα αναφερόμαστε στα "καθαρισμένα" δεδομένα όπως περιγράψαμε στην παράγραφο 2.2. Από τα δεδομένα αυτά κατασκευάσαμε ένα λεξικό με κλειδία τις 100 αυτές λέξεις και τιμή το πλήθος εμφάνισής τους σε όλα τα δεδομένα. Έπειτα κατασκευάσαμε ένα δεύτερο λεξικό με κλειδία τις δέκα κατηγορίες των άρθρων μας και τιμή ένα νέο εμφωλευμένο λεξικό. Κάθε εμφωλευμένο λεξικό είχε ξανά ως κλειδία τις 2000 δημοφιλέστερες λέξεις και ως τιμή το ποσοστό εμφάνισης τους στην κατηγορία x, όπου x είναι η κατηγορία του κλειδιού που ανήκει το συγκεκριμένο λεξικό.

Για τον υπολογισμό του ποσοστού εμφάνισης της κάθε λέξης σε κάθε κατηγορία έγιναν τα εξής. Συγκεντρώσαμε τα δεδομένα μας ανά κατηγορία, δηλαδή ενώσαμε ανά κατηγορία όλες τις λέξεις σε μία λίστα. Διασχίσαμε τα κλειδιά του λεξικού με τις 100 συνηθέστερες λέξεις και μετρήσαμε πόσες φορές εμφανίζεται κάθε λέξη σε κάθε κατηγορία. Το ποσοστό εμφάνισης μιας λέξης προκύπτει διαιρώντας το πλήθος εμφάνισης της ανά κατηγορία με το πλήθος εμφάνισης της σε όλα τα δεδομένα.

Δηλαδή:

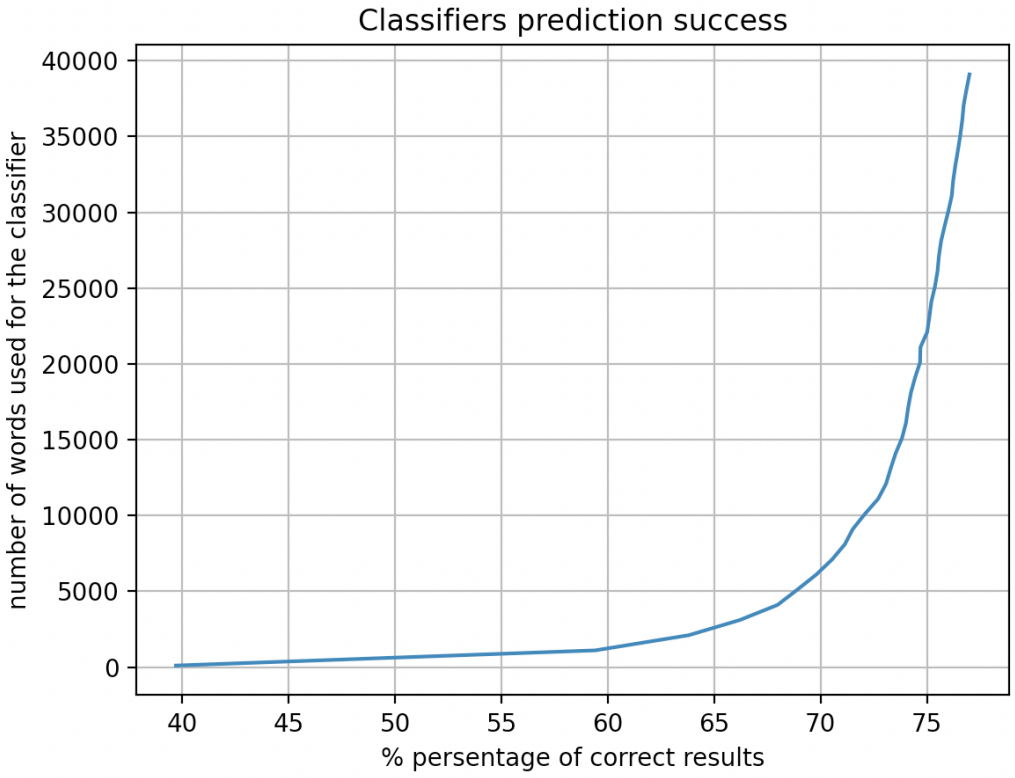
ποσοστό εμφάνισης λέξης ανά κατηγορία = πλήθος εμφάνισης της λέξης ανά κατηγορία/ πλήθος εμφάνισης της λέξεις σε όλα τα δεδομένα [31][32]

[7] https://github.com/elenisproject/NewsBackend/blob/master/API/classifier.py

Προκειμένου να κατατάξουμε ένα άρθρο σε μία κατηγορία πρέπει φυσικά να γίνει πρώτα ανάλογη επεξεργασία καθαρισμού στην είσοδο. Επομένως αφού δώσει ο χρήστης ένα άρθρο ως είσοδο γίνεται η ίδια διαδικασία καθαρισμού που περιγράφεται στην παράγραφο 2.2. αφαιρώντας αγγλικά, νούμερα, ειδικούς χαρακτήρες, μικρές λέξεις και τις συνηθέστερες λέξεις. Στην συνέχεια χωρίζουμε το άρθρο σε λέξεις και για κάθε κατηγορία μετράμε το νούμερο που συγκεντρώνει κάθε κατηγορία όταν αθροίζουμε το ποσοστό εμφάνισης των λέξεων του κειμένου. Η κατηγορία που συγκεντρώνει το μεγαλύτερο άθροισμα θεωρούμε ότι είναι και η κατηγορία του άρθρου μας που αναζητούμε. [33]

Ο ελεγχος του ποσοστού επιτυχίας έγινε ανακατεύοντας και χωρίζοντας τα δεδομένα. Το 70% των δεδομένων χρησιμοποιήθηκε για να υπολογιστούν τα ποσοστά εμφάνισης των λέξεων ενώ τα υπόλοιπα 30% χρησιμοποιήθηκαν για έλεγχο[8]. Δεδομένου ότι κάθε άρθρο έχει την δική του κατηγορία, δώσαμε ως είσοδο το κείμενο και με βάση την πρόβλεψη που μας επέστρεψε ο κατηγοριοποιητής έγινε σύγκριση με την πραγματικη τους κατηγορία για να δούμε το ποσοστό σφάλματος. [34][35] Τέλος μελετήθηκε η συνάρτηση απόδοσης του classifier συναρτήσει του πλήθους δημοφιλέστερων λέξεων που κρατάει για την πρόβλεψη του.

[8] https://github.com/elenisproject/NewsBackend/blob/master/tester.py



Εικόνα 3.4.1

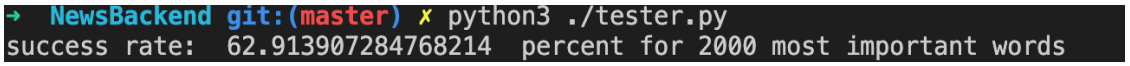
Το διάγραμμα που προκύπτει είναι λογικό γιατί ο αλγόριθμος του Naive Bayes όπως γνωρίζουμε, αποδίδει καλύτερα σε μεγάλο πλήθος δεδομένων επομένως σε όσες περισσότερες λέξεις κρατάμε το ποσοστό επιτυχίας, αναμενόμενο είναι να κάνει και καλύτερη πρόβλεψη. Συγκεκριμένα βλέπουμε ότι μπορεί να ξεπεράσει και το 75% για 35000 λέξεις στο λεξικό. Για λόγους απόδοσης του API κρατάμε λιγότερες λέξεις στα πλαίσια της διπλωματικής. Στο πλήθος των 2000 λέξεων η πρόβλεψη του Naive Bayes έχει ποσοστό επιτυχίας 63%. Η έξοδος για διαφορετικό πλήθος λέξεων φαίνεται παρακάτω όπως αυτές παρήχθησαν από τον tester.

Πλήθος λέξεων	Ποσοστό Επιτυχίας %
100	40.18
1000	58.93
2000	63.73
3000	66.0
4000	67.82
5000	68.69
10000	71.96

20000	74.78
30000	76.63
40000	77.7

Πίνακας 3.4

Για 2000 λέξεις που χρησιμοποιούμε εμείς έχουμε ποσοστό επιτυχίας 62.91 όπως φαίνεται παρακάτω



Εικόνα 3.4.2

3.6. *Similar Articles Tool*

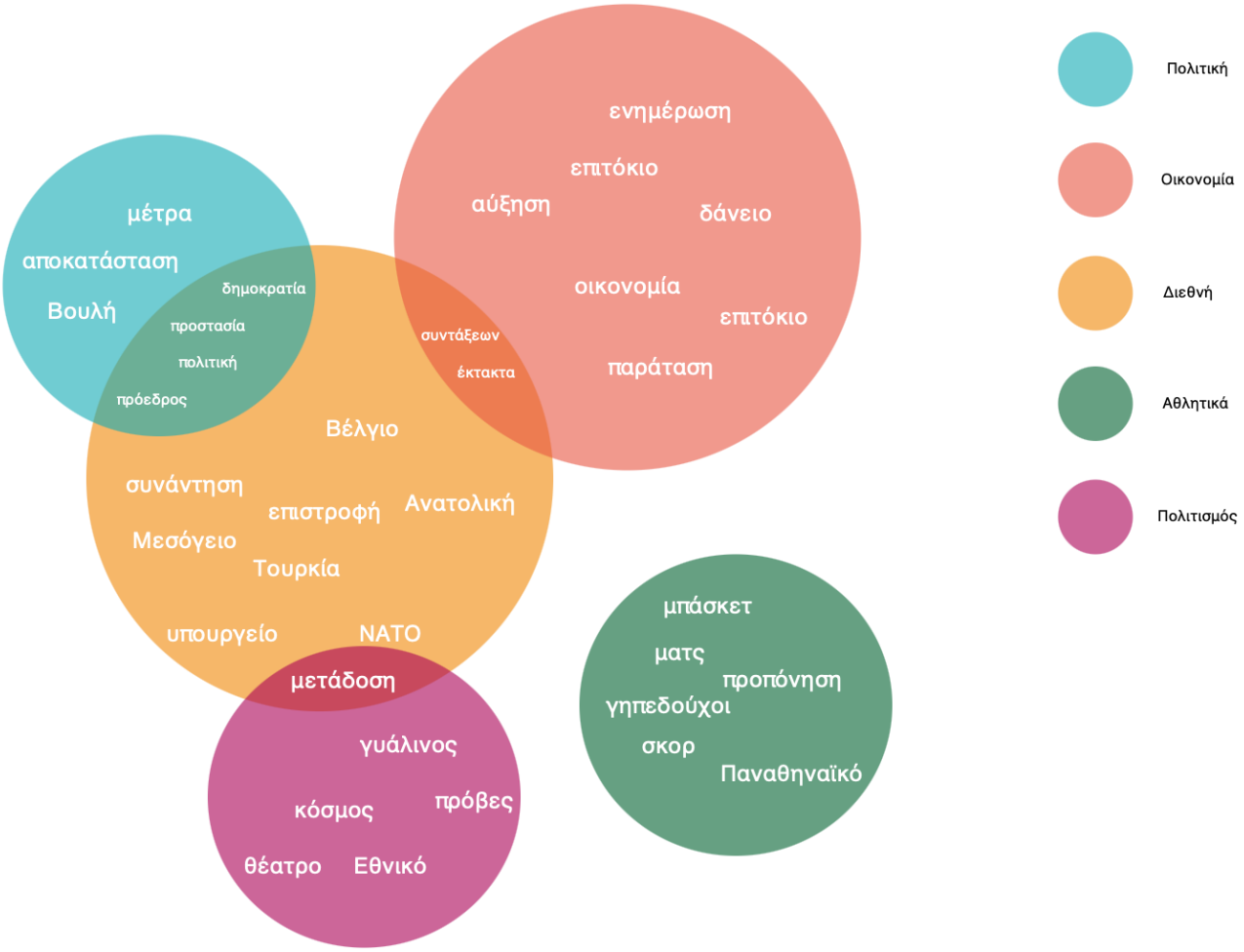
Μία ακόμα δυνατότητα της ιστοσελίδας είναι η πρόταση όμοιων άρθρων από την βάση δεδομένων με κριτήριο το άρθρο που επέλεξε ο χρήστης να διαβάσει. Κριτήριο για την εύρεση όμοιων άρθρων ήταν η ομοιότητα των λέξεων. Αναλυτικότερα, όμοια άρθρα θεωρούμε εκείνα τα οποία αποτελούνται από τις ίδιες λέξεις. Η ανάλυση αυτή έγινε πάνω στα ήδη "καθαρισμένα" δεδομένα μας.[36] Η διαδικασία που ακολουθήθηκε χωρίζεται σε 2 στάδια.

Αρχικά παρατηρήσαμε ότι ο υπολογισμός των άρθρων είχε πολυπλοκότητα $O(n^2)$ εφόσον για κάθε άρθρο έπρεπε να διασχίσουμε και να αναλύσουμε από την αρχή όλα τα δεδομένα μας. Συνεπώς μέσω του API αργούσε να υπολογιστεί το αποτέλεσμα και να φορτώσει η σελίδα γεγονός που δυσκολεύει την εμπειρία του χρήστη. Για τον λόγο αυτό κατασκευάστηκε ένας νέος πίνακας στην βάση δεδομένων μας ο οποίος το συσχετίστηκε με τον αρχικό μας πίνακα με μία σχέση 1:1. Δηλαδή για κάθε άρθρο του πίνακα υπολογίσαμε τα 5 ομοιότερα άρθρα και έπειτα αποθηκεύτηκαν στον νέο πίνακα με foreign key το id του άρθρου υπό συζήτηση. Συνεπώς λοιπόν όταν η ιστοσελίδα χρειάζεται να φορτώσει τα προτεινόμενα άρθρα κάνει μία απλή κλήση στην βάση στον νέο πίνακα και τα αποτελέσματα επιστρέφουν στιγμιαία.[37]

Δεύτερο στάδιο είναι ο υπολογισμός των όμοιων άρθρων.[38][39] Προσπελάστηκαν τα δεδομένα του πίνακα μας κατά γραμμή δηλαδή κατά άρθρο και για κάθε άρθρο δημιουργήσαμε ένα σύνολο setA με τις μοναδικές του λέξεις, σε κάθε γραμμή κάναμε μία δεύτερη προσπέλαση και δημιουργείται ένα setB για τις μοναδικές λέξεις κάθε άρθρου. [40]Το πλήθος όμοιων λέξεων το υπολογίζαμε με βάση το μήκος της ένωσης των δύο συνόλων. Το πλήθος αυτό το αποθηκεύτηκε σε ένα λεξικό με κλειδί το id του άρθρου και τιμή το πλήθος στοιχείων της ένος. Τέλος κάναμε μία ταξινόμηση στο λεξικό με βάση τις τιμές και αποθηκεύτηκαν τα ids των 5 μεγαλύτερων στον νέο πίνακα. Αναλυτικά ο κώδικα για αυτή την λειτουργία βρίσκεται στο αρχείο common_articles_finder.py[9]. Τα όμοια άρθρα εμφανίζονται στο τέλος της σελίδας όταν ανοίξει ο χρήστης ένα άρθρο να διαβάσει. Εμφανίζονται μετά το τέλος του κειμένου του άρθρου τα 3 ομοιότερα σε αυτό άρθρα.[41][42]

Η σχηματική απεικόνιση του εργαλείου:

[9] https://github.com/elenisproject/NewsBackend/blob/master/TextPreprocessor/common_articles_finder.py



Εικόνα 3.5.1

Στην ουσία, όπως βλέπουμε παραπάνω, τα ομοιότερα άρθρα είναι εκείνα που η τομή του συνόλου των λέξεων τους είναι μεγαλύτερη.

Πρακτικά παίρνοντας ένα παράδειγμα από την εφαρμογή. Με είσοδο για παράδειγμα το παρακάτω άρθρο:

Εικόνα 3.5.2

Δηλαδή όταν το κείμενο ανάγνωσης είναι:

Αυξήθηκαν τις τελευταίες 24 ώρες οι θάνατοι και τα κρούσματα στη Γερμανία , καθώς η χώρα παλεύει με τον κορονοϊό ενώ προσπαθεί να επιστρέψει στην κανονικότητά της. Μέσα σε μια μέρα, στη Γερμανία, έχασαν τη ζωή τους από την COVID-19, την ασθένεια που προκαλεί ο κορονοϊός, 45 άνθρωποι, σύμφωνα με τα στοιχεία του Ινστιτούτου Ρόμπερτ Κοχ. Πλέον, ο αριθμός των νεκρών από την πανδημία στη Γερμανία έχει φτάσει στους 8.302. Εκτός όμως από τους θανάτους, μέσα στις τελευταίες 24 ώρες, αυξήθηκαν και τα κρούσματα μόλυνσης από τον SARS-CoV-2 (κορονοϊός) κατά 432 και έφτασαν στα 179.002. Χθες, οι θάνατοι από τον κορονοϊό στη Γερμανία ήταν 10 και τα κρούσματα 289.

Το δεύτερο κοντινότερο άρθρο σε αυτό το κείμενο εισόδου είναι:

Η πανδημία του νέου κορονοϊού έχει στοιχίσει τη ζωή σε τουλάχιστον 266.919 ανθρώπους σε όλον τον κόσμο από τον περασμένο Δεκέμβριο, όταν πρωτοεμφανίστηκε ο κορονοϊός SARS-CoV-2 στην Κίνα. Έχουν καταγραφεί περισσότερα από 3.806.440 επιβεβαιωμένα κρούσματα σε 195 χώρες και εδάφη. Ο αριθμός αυτός των διαγνωσμένων μολύνσεων, ωστόσο, αντανακλά μόνο ένα κλάσμα του πραγματικού αριθμού λοιμώξεων, με μεγάλο αριθμό χωρών να εξετάζουν μόνο τα κρούσματα που απαιτούν νοσοκομειακή περίθαλψη. Τουλάχιστον 1.197.100 από αυτούς τους ασθενείς έχουν πλέον αναρρώσει. ΗΠΑ Στις ΗΠΑ καταγράφηκαν χθες Πέμπτη πάνω από 2.400 επιπλέον θάνατοι εξαιτίας της πανδημίας του κορονοϊού μέσα σε ένα εικοσιτετράωρο, αριθμός ο οποίος αύξησε τον συνολικό απολογισμό σε πάνω από 75.500 νεκρούς, σύμφωνα με την καταμέτρηση του πανεπιστημίου Τζονς Χόπκινς Γερμανια Άλλοι 147 ασθενείς υπέκυψαν στην ασθένεια COVID-19 στη Γερμανία το προηγούμενο 24ωρο, όπου το σύνολο των θυμάτων της πανδημίας του κορονοϊού έφθασε τα 7.266, δείχνουν τα δεδομένα που συγκεντρώνει και μεταφορτώνει καθημερινά στον ειδικό ιστότοπο που έχει δημιουργήσει το Robert Koch-Institut, το οποίο ειδικεύεται στις μολυσματικές ασθένειες. Ο αριθμός των νέων θανάτων αυξήθηκε σε σύγκριση με αυτόν που καταγραφόταν μία ημέρα νωρίτερα (123). Το ίδιο διάστημα, τα επιβεβαιωμένα κρούσματα μόλυνσης από τον SARS-CoV-2 αυξήθηκαν κατά 1.209 στα 167.300, κατά τα στοιχεία του Ινστιτούτου Ρόμπερτ Κοχ. Ο αριθμός των κρουσμάτων μειώθηκε σε σύγκριση με το προηγούμενο 24ωρο, παρέμεινε όμως πάνω από το επίπεδο των 1.200, παρότι για ένα διήμερο αυτή την εβδομάδα (τη Δευτέρα και την Τρίτη) τα κρούσματα δεν ξεπέρασαν τα 700. Γαλλία Οι νεκροί από τον νέο κορονοϊό στη Γαλλία ανέρχονται πλέον στους 25.987, καθώς άλλοι 178 άνθρωποι προστέθηκαν στη μακριά λίστα των θυμάτων το προηγούμενο 24ωρο. Ο αριθμός αυτός πάντως είναι ο μικρότερος που έχει καταγραφεί τις τέσσερις τελευταίες ημέρες. Νωρίτερα ο πρωθυπουργός Εντουάρ Φιλίπ ανακοίνωσε ότι η Γαλλία θα αρχίσει να αίρει την καραντίνα από την Δευτέρα, 11 Μαΐου. Στην ανακοίνωσή του το υπουργείο Υγείας ανέφερε ότι ο αριθμός των ασθενών που νοσηλεύονται σε μονάδες εντατικής θεραπείας έπεσε στους 2.961 (-186). Είναι η πρώτη φορά από τις 25 Μαρτίου ότι οι νοσηλευόμενοι σε ΜΕΘ

μειώνονται κάτω από τους 3.000. Έχει μειωθεί επίσης στους 23.208 (από 23.983) ο συνολικός αριθμός των ασθενών σε νοσοκομεία. Στην κορύφωση της επιδημίας , στις 8 Απριλίου, οι νοσηλευόμενοι ασθενείς είχαν φτάσει τους 32.292 και εκείνοι που είχαν εισαχθεί σε ΜΕΘ τους 7.148. Λατινική Αμερική Το υπουργείο Υγείας της Βραζιλίας ανακοίνωσε χθες Πέμπτη ότι τις προηγούμενες 24 ώρες κατέγραψε 610 θανάτους ασθενών που υπέκυψαν στην COVID-19 και 9.888 επιβεβαιωμένα κρούσματα μόλυνσης από τον SARS-CoV-2. Με τα νεότερα δεδομένα, ο απολογισμός της πανδημίας του κορονοϊού αυξήθηκε επισήμως σε 9.146 νεκρούς επί συνόλου 135.106 επιβεβαιωμένων κρουσμάτων μόλυνσης. Πρόκειται για τον πιο βαρύ απολογισμό μεταξύ των αναπτυσσόμενων χωρών. Ειδικοί ωστόσο τονίζουν πως ο επίσημος απολογισμός είναι πολύ υποτιμημένος. Μολαταύτα, η κυβέρνηση υπό τον ακροδεξιό πρόεδρο Ζαΐχ Μπολσονάρου συνεχίζει να εναντιώνεται στα μέτρα περιορισμού που έχουν επιβάλει οι κυβερνήτες πρακτικά όλων των πολιτειών της χώρας για να αποτραπεί η εξάπλωση της πανδημίας. Το υπουργείο Υγείας του Μεξικού ανακοίνωσε την Πέμπτη ότι το προηγούμενο 24ωρο καταγράφηκαν 257 θάνατοι εξαιτίας της COVID-19 και 1.982 κρούσματα μόλυνσης από τον SARS-CoV-2, με τον απολογισμό της πανδημίας του κορονοϊού να φθάνει έτσι τους 2.961 νεκρούς επί συνόλου 29.616 επιβεβαιωμένων κρουσμάτων μόλυνσης. Στελέχη της μεξικανικής ομοσπονδιακής κυβέρνησης έχουν ωστόσο επισημάνει επανειλημμένα ότι ο πραγματικός αριθμός των κρουσμάτων είναι πιθανότατα πολλαπλάσιος των επιβεβαιωμένων.

3.7. Flask Framework

Τελευταίο εργαλείο για την ολοκλήρωση της διπλωματικής εργασίας ήταν η δημιουργία της ιστοσελίδας. Η ιστοσελίδα χτίστηκε πάνω στο flask web framework.[43] Η διαδικασία που ακολουθήσαμε ήταν αρχικά ο σχεδιασμός των σελίδων, η οργάνωση των περιεχομένων, η συλλογή οπτικού υλικού για την παρουσίαση και τέλος η υλοποίηση του API.[44] [45] Η εφαρμογή μας χωρίζεται σε models[10], forms[11], και routes[12]. [46][47]

Με τα models.py ορίζουμε την δομή των δεδομένων που διαχειρίζεται και δέχεται η ιστοσελίδα μας. Δεδομένου ότι τα δεδομένα βρίσκονται σε δύο πίνακες ορίσαμε και δύο διαφορετικά μοντέλα με βάση τα στοιχεία κάθε πίνακα. Μέσω των μοντέλων ορίζουμε τον τύπο δεδομένων που θα δεχτούμε και μέσω αυτού του αντικειμένου θα αποκτούμε πρόσβαση σε όλα τα υπόλοιπα αρχεία της εφαρμογής.

Οι κλήσεις του API από το frontend στο backend υλοποιούνται κατά κύριο λόγο με την βοήθεια της flask_wtf βιβλιοθήκης. Οι φόρμες αυτές αναλαμβάνουν να πάρουν σαν είδοσο την είσοδο που πληκτρολογεί ο χρήστης και την μεταφέρει στην εφαρμογή μας, για επεξεργασία, ανάλυση ή αναζήτηση.

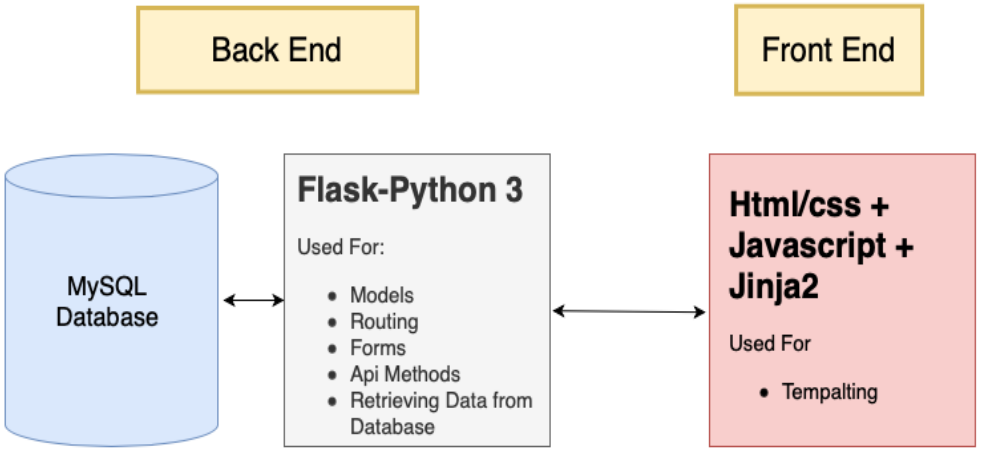
Στο routes ορίζονται οι διευθύνσεις που διαχειρίζεται η εφαρμογή μας. Όταν πληκτρολογηθεί ένα url που υπάρχει στην εφαρμογή καλείται η κατάλληλη συνάρτηση που επεξεργάζεται το request μας και επιστρέφει ένα response ή κάνει redirect την ιστοσελίδα σε κάποια άλλη διεύθυνση. Στο σημείο αυτό ορίζουμε και τι μεθόδους μπορεί να εκτελέσει το συγκεκριμένο endpoint, post ή get. Ορίζεται post μέθοδος όταν θέλουμε να πάρουμε κάποια δεδομένο από την σελίδα, ενώ get είναι η μέθοδος κατά την οποία κάνουμε μια κλήση στην βάση δεδομένων για να φέρει αποτελέσματα στην σελίδα.

Τα αρχεία που παρουσιάστηκαν αποτελούν την βασική δομή της εφαρμογής, κατ' επέκταση έχουμε και τα στατικά στοιχεία που είναι οι html σελίδες[13]. Στα αρχεία αυτά

[10] https://github.com/elenisproject/NewsBackend/blob/master/API/app/models.py
[11] https://github.com/elenisproject/NewsBackend/blob/master/API/app/forms.py
[12] https://github.com/elenisproject/NewsBackend/blob/master/API/app/routes.py
[13] https://github.com/elenisproject/NewsBackend/tree/master/API/app/templates

συνδυαστικά με την βοήθεια του jinja παρουσιάζονται τα δεδομένα στον χρήστη. Στο σημείο αυτό έχουμε και τα notebooks[14] καθώς και τον javascript κώδικα[15] για κάποια πιο εξελιγμένα charts. Όσων αφορά τα notebooks, στόχος ήταν να παρουσιάζονται αυτούσια μέσα από το jupyter notebook ώστε να μπορούν να εκτελεστούν από τον ίδιο τον χρήστη και αν είναι πιο διαδραστικός ο ιστότοπος Η υλοποίηση αυτή έγινε με την βοήθεια του iframe.. Αντίστοιχα τα notebooks χρησιμοποιούν script[16] σε python ώστε να παραχθούν τα δεδομένα που παρουσιάζονται.

Η δομή της λειτουργίας του flask φαίνεται στο παρακάτω σχήμα:



Εικόνα 3.6.1

Στο σχήμα φαίνεται ο δίαυλος επικοινωνίας του flask. Το flask "σερβιρει" σελίδες στο front end τις οποίες βλέπει ο χρήστης. Χρησιμοποιεί javascript για ποιο διαδραστικές εικόνες ενώ χρησιμοποιεί στις σελίδες αυτές jinja για να επικοινωνεί με τον χρήστη. Όταν λέμε επικοινωνία με τον χρήστη αναφερόμαστε στα δεδομένα που φέρνει από την βάση δεδομένων για να τα παρουσιάσει στον χρήστη όπως επίσης και στα δεδομένα που πληκτρολογεί ο χρήστης για να επικοινωνήσει με την βάση. Στην προκειμένη περίπτωση να κάνει κάποια αναζήτηση ή μία κατάταξη άρθρου. Πέρα από το κομμάτι του front end έχουμε και το κομμάτι του backend

[14] https://github.com/elenisproject/NewsBackend/tree/master/API/app/templates/notebooks
[15] https://github.com/elenisproject/NewsBackend/tree/master/API/app/templates/charts
[16] https://github.com/elenisproject/NewsBackend/tree/master/ChartFunctions

στο οποίο το flask αναλαμβάνει να επικοινωνήσει με την βάση δεδομένων και να πάρει τα απαραίτητα δεδομένα.

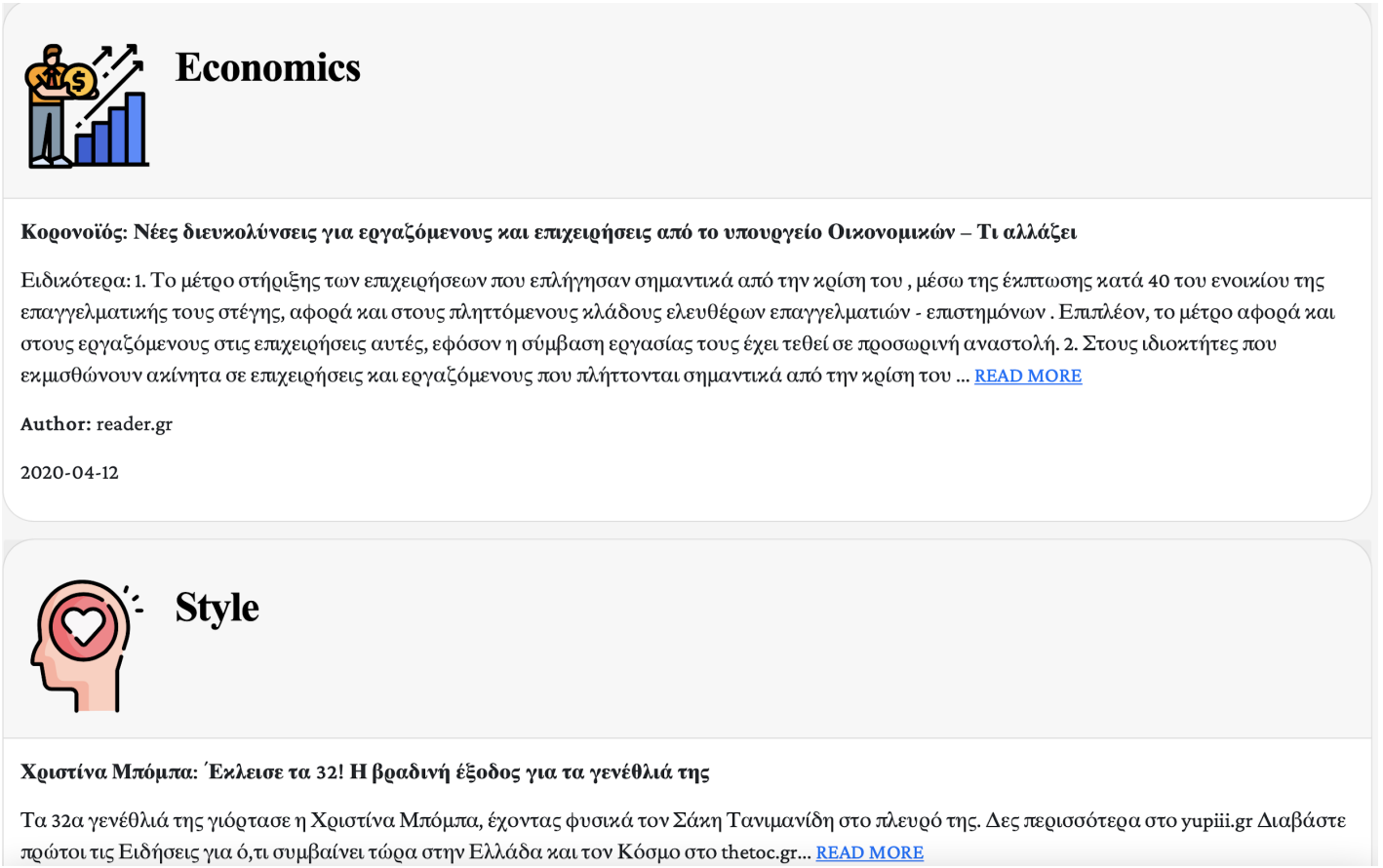
Για παράδειγμα έχουμε το API της αρχικής σελίδας, το οποίο κάνει κλήση στην βάση δεδομένων για να φέρει τυχαία 10 άρθρα με κάποια από τα δεδομένα τους.

Η μορφή του API είναι όπως φαίνεται στην συνέχεια.

```
@app.route('/',methods=["POST","GET"])

@app.route('/home')
def home():
    page = request.args.get('page', 1, type=int)
    articles = Articles.query.order_by(func.rand()).paginate(
        page, app.config['POSTS_PER_PAGE'], False)

    next_url = url_for('home', page=articles.next_num) \
        if articles.has_next else None
    prev_url = url_for('home', page=articles.prev_num) \
        if articles.has_prev else None

    return render_template('home.html', title='All Articles', articles=articles.items, next_url=next_url, prev_url=prev_url)
```

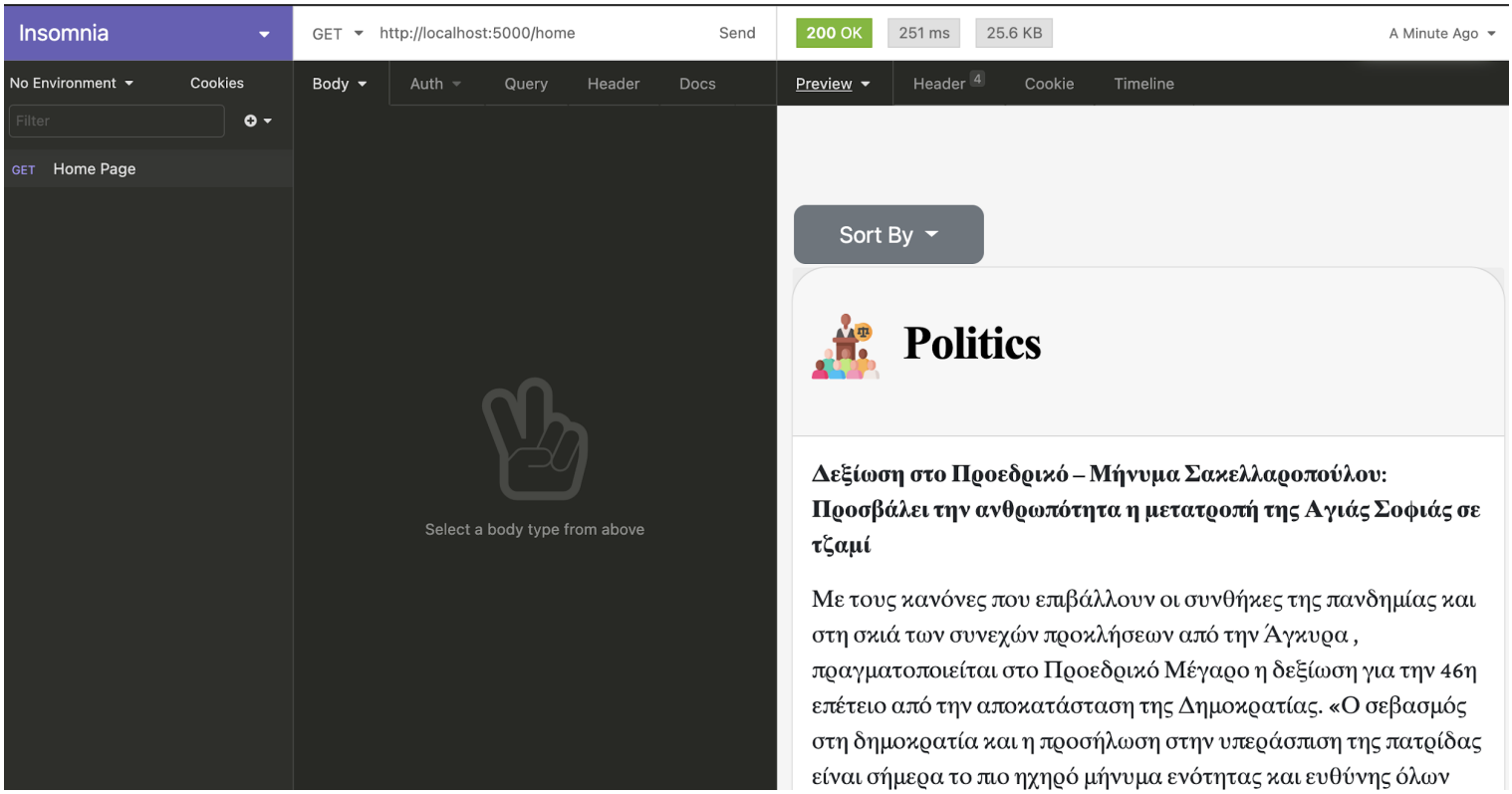
Εικόνα 3.6.2

και το front end που φτάνει στον χρήστη:



Εικόνα 3.6.3

Για καλύτερη κατανόηση της αντιστοίχισης κλήσεων API με το front end δίνεται στην η εικόνα μέσα από το insomnia.


Insomnia
GET
http://localhost:5000/home
Send
200 OK
251 ms
25.6 KB
A Minute Ago
No Environment
Cookies
Body
Auth
Query
Header
Docs
Preview
Header 4
Cookie
Timeline
Filter
GET Home Page
Select a body type from above
Sort By
Politics
Δεξίωση στο Προεδρικό – Μήνυμα Σακελλαροπούλου: Προσβάλει την ανθρωπότητα η μετατροπή της Αγιάς Σοφιάς σε τζαμί
Με τους κανόνες που επιβάλλουν οι συνθήκες της πανδημίας και στη σκιά των συνεχών προκλήσεων από την Άγκυρα , πραγματοποιείται στο Προεδρικό Μέγαρο η δεξίωση για την 46η επέτειο από την αποκατάσταση της Δημοκρατίας. «Ο σεβασμός στη δημοκρατία και η προσήλωση στην υπεράσπιση της πατρίδας είναι σήμερα το πιο ηχηρό μήνυμα ενότητας και ευθύνης όλων

Εικόνα 3.6.4

4. Σχήμα Αρχιτεκτονικής

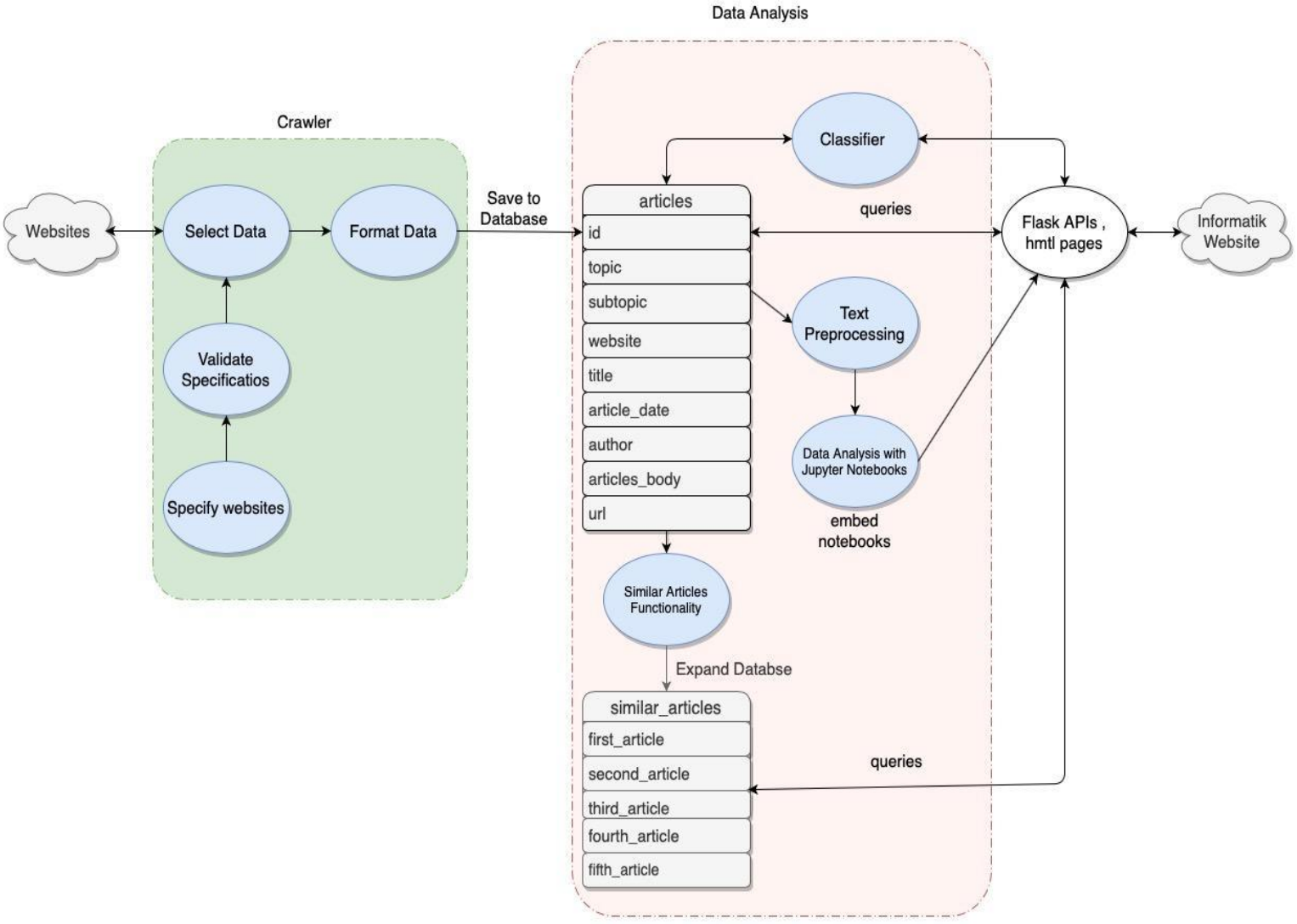
4.1. Project Structure

Στην συνέχεια παρουσιάζεται η πορεία των δεδομένων από τις αρχικές αρθρογραφικές ιστοσελίδες που δημοσιεύθηκαν ως τον τελικό ιστότοπο έξυπνης πλοήγησης της παρούσας διπλωματικής. Επομένως λοιπόν τα δεδομένα είναι διάσκορπα σε διαφορετικούς ειδησεογραφικούς ιστότοπους και πρέπει να καθοριστούν οι ιστότοποι που μας ενδιαφέρουν. Επίσης πρέπει να προσδιοριστούν οι θεματικές κατηγορίες που θα χρησιμοποιηθούν ως αρχικά tags.

Πρώτο βήμα στην πορεία των δεδομένων λοιπόν αποτελεί ο Crawler, ο οποίος είναι υπεύθυνος για την συλλογή των απαιτούμενων πληροφοριών από κάθε άρθρο και από συγκεκριμένους ιστότοπους. Στο επίπεδο του web crawling γίνεται και η πρώτη επεξεργασία για την μορφοποιηση των δεδομένων και την εισαγωγή στην βάση δεδομένων. Αφού λοιπόν δομήσει ο crawler κατάλληλα τα δεδομένα, κάνει κλήσεις στην βάση για την είσοδό των πληροφοριών που συλλέχθηκαν στον πίνακα articles.

Δεύτερο βήμα της διαδικασίας είναι η ανάλυση των δεδομένων και η διαμόρφωση των διαγραμμάτων από την ανάλυση αυτή. Η Ανάλυση γίνεται ως προς το περιεχόμενο των άρθρων, την ημερομηνία συγγραφής, τον συγγραφέα και την ποιότητα της αρθρογραφίας. Για να γίνει η ανάλυση αυτή πρέπει πρώτα να καθαριστούν τα δεδομένα από κάθε περιττή πληροφορία, να αποθηκευτεί σε ένα csv αρχεία και από εκεί να χρησιμοποιηθεί από τα jupyter Notebooks για περαιτέρω ανάλυση.

Τελευταίο βρίσκεται το κομμάτι του API για την παρουσίαση των δεδομένων και των notebooks στον χρήστη. Χρησιμοποιείται το flask για τις κλήσεις στην βάση δεδομένων και την παρουσίαση στον ιστότοπο. Επίσης μέσα από το flask διατίθενται και τα notebooks για την άμεση συμμετοχή και παρατήρηση του χρήστη στην διαδικασία επεξεργασίας των δεδομένων.



Εικόνα 4.1

4.2. Data Model

Η βάση δεδομένων αποτελείται στην ουσία από δύο πίνακες που συνδέονται με συνάρτηση 1 προς 1, δηλαδή σε κάθε id του πίνακα articles αντιστοιχεί ακριβώς ένα id του πίνακα similar_articles. Τα ids των δύο πινάκων έχουν άμεση σχέση καθώς το id των similar_articles είναι foreign key απο το id του πίνακα articles, συνεπώς είναι ίδια τιμή.

Ο πίνακας articles, περιέχει όλη την πληροφορία των άρθρων. Εκεί έχει αποθηκεύσει ο crawler τα raw δεδομένα όπως τα βρήκε στο διαδίκτυο. Ο πίνακας έχει ως πεδία τα id, topic, subtopic, website, title, article_date, author, article_body, url. Το id είναι τύπου int και αποτελεί έναν μοναδικό τυχαία αριθμό για κάθε άρθρο. Το topic ορίζεται ως varchar τύπος και είναι μία από τις δέκα κατηγορίες που ορίστηκαν κατά τον σχεδιασμό του project και αντιστοιχήθηκαν με κάποια από τις κατηγορίες που είχε κάθε ιστοσελίδα. Subtopic τύπου πάλι varchar, είναι η υποκατηγορία που ανήκει το άρθρο, αυτές ήρθαν από την ίδια την ιστοσελίδα. Για παράδειγμα στον αθλητισμό ένα subtopic είναι το basket. Αν ο ιστότοπος δεν είχε κάποια δική του υποκατηγορία, το πεδίο στην περίπτωση αυτή έπαιρνε την τιμής της κατηγορίας. Έπειτα αποθηκεύεται το πεδία website, είναι τύπου varchar και προφανώς εκεί αποθηκεύεται η ιστοσελίδα προέλευσης του άρθρου. Title, είναι τύπου text καθώς σε αρκετές περιπτώσεις ο τίτλος δεν χωράει σε μία varchar μεταβλητή. Όπως προμηνύει και το όνομα αυτός είναι ο τίτλος κάθε άρθρου. Article_date, είναι τύπου date και εκεί βρίσκεται η ημερομηνία δημοσίευσης του άρθρου στην ιστοσελίδα προέλευσης του. Author, είναι επίσης τύπου varchar και φυλάσσεται το όνομα του αρθρογράφου. Στην περίπτωση που ο αρθρογράφος δεν κατονομάζεται, τοποθετείται στο πεδίο αυτό η ιστοσελίδα που το δημοσίευσε. Σημαντικό πλέον πεδία είναι το article_body το οποίο αποτελεί το σώμα του κειμένου και είναι τύπου longtext. Τέλος αποθηκεύεται και url κάθε άρθρο στην στήλη με το αντίστοιχο όνομα, ο τύπος αυτός ορίζεται ως τύπος varchar. Το πεδίο αυτό είναι εξίσου σημαντικό καθώς από εκεί θα μπορέσει να ανακατευθυνθεί ο χρήστης, αν θελήσει να δει το άρθρο στην αρχική του μορφή εκεί που τραβήχτηκε από τον crawler.

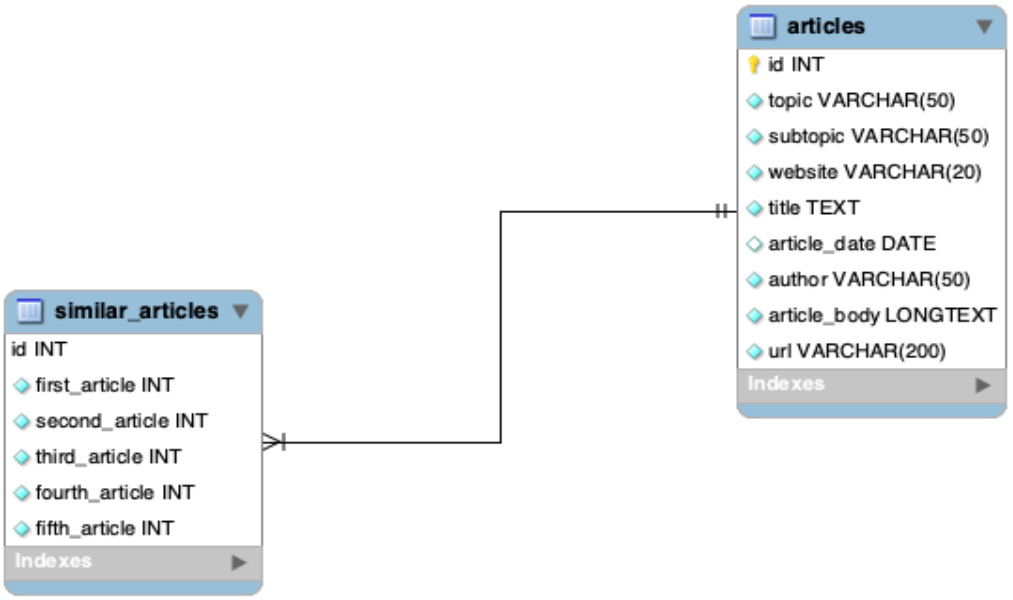
Ο πίνακας όπως περιγράφηκε παραπάνω:

COLUMN NAME	REPRESENTING	TYPE
id	random unique article id	int
topic	one of our ten categories	varchar
subtopic	article subtopic if no subtopic is given, this values is set the same as topic	varchar
website	article website domain	varchar
title	article title	text
article_date	the date the article was published	date
authors	author of the article, in case no author is named, we give as author the website	varchar
article_body	the text of the article	text
url	the url we crawled to get the article	varchar

Πίνακας 4.2

Ο δεύτερος πίνακας σχεδιάστηκε και υλοποιήθηκε στο στάδιο επεξεργασίας των δεδομένων για την γρηγορότερη εξυπηρέτηση των κλήσεων του API και άρα την καλύτερη εμπειρία του χρήστη. Στον δεύτερο πίνακα υπάρχουν έξι πεδία όλα τύπου int. Το πρώτο πεδίο είναι το id κάθε στήλης το οποίο είναι foreign key στον πίνακα articles ωστε να συνδέονται ένα προς ένα, οι δύο πίνακες. Στα υπόλοιπα πέντε πεδία βρίσκονται τα ids των πέντε ομοιότερων άρθρων σε φθίνουσα σειρά, όπως αυτά βρέθηκαν στο στάδιο ανάλυσης των δεδομένων. Αν και έχουμε υπολογίσει τα πέντε κοντινότερα στο website προτείνονται και παρουσιάζονται τα τρία ομοιότερα άρθρα σε εκείνο που επέλεξε να διαβάσει ο χρήστης.

Παρακάτω παρουσιάζεται το ER-Diagram της βάσης.



Εικόνα 4.2

4.3. System Platform

Για την υλοποίηση των APIs χρησιμοποιείται το flask framework. Με την βοήθεια αυτού υλοποιείται η σύνδεση και η παρουσίαση των δεδομένων από την βάση στην ιστοσελίδα. Για τις ανάγκες του ιστότοπου έχουν υλοποιηθεί οι εξής υπηρεσίες:

Παρουσίαση[17] όλων των άρθρων ως αρχική σελίδα. Στην αρχική σελίδα είναι διατεταγμένα τα άρθρα σε τυχαία σειρά. Για την υλοποίηση αυτή γίνεται ένα query στην βάση δεδομένων που ζητάει να φέρει κάποια πρώτα στοιχεία για την περιληπτική παρουσίαση του άρθρου. Ζητάμε επομένως από την βάση να μας γυρίσει τον τίτλο, τον συγγραφέα, την ημερομηνία συγγραφής, την κατηγορία και τους 500 πρώτους χαρακτήρες του άρθρου για μία γρήγορη εποπτεία. Τα δεδομένα που έρχονται από την βάση αναλαμβάνεται από τις html σελίδες να παρουσιαστούν στον χρήστη. Η σελίδα αυτή βρίσκεται στο url http://localhost:5000/home.

Υπάρχει επίσης η δυνατότητα ταξινόμηση των άρθρων στην αρχική σελίδα με βάση την ημερομηνία συγγραφής κατά φθίνουσα[18] ή αύξουσα[19] σειρά. Με την επιλογή αυτή από τον χρήστη μέσω του dropdown list στην αρχική σελίδα στέλνει ένα νέο αίτημα το API στην βάση δεδομένων να εκτελεστεί ένα query το οποίο αντί να επιστρέφει τυχαία δεδομένα θα τα επιστρέφει ταξινομημένα. Επομένως τα δεδομένα έρχονται ταξινομημένα από την βάση και παρουσιάζονται στις σελίδες που σερβίρει η εφαρμογή. Η σελίδες ταξινόμηση βρίσκονται στην τοποθεσία http://localhost:5000/newest και http://localhost:5000/oldest. Στην ίδια dropdown λίστα υπάρχει και η δυνατότητα αντίστοιχης ταξινόμησης των άρθρων στην αρχική

[17] https://github.com/elenisproject/NewsBackend/blob/ac69ceff437ecc234026ded00d60d3d0f0e83a49/API/app/routes.py#L22

[18] https://github.com/elenisproject/NewsBackend/blob/ac69ceff437ecc234026ded00d60d3d0f0e83a49/API/app/routes.py#L73

[19] https://github.com/elenisproject/NewsBackend/blob/ac69ceff437ecc234026ded00d60d3d0f0e83a49/API/app/routes.py#L86

σελίδα αλφαβητικά ξανά σε αύξουσα[20] ή φθίνουσα[21] ταξινόμηση. Η λογική που ακολουθείται είναι η ίδια με τροποποιημένα αυτή την φορά queries για την συγκεκριμένη ταξινόμηση. Οι σελίδες για αύξουσα και φθίνουσα σε αλφαβητική σειρά βρίσκονται στο http://localhost:5000/alphabetical, http://localhost:5000/alphabeticalDesc αντίστοιχα. Μία τελευταία δυνατότητα ίδια λογικής είναι η ταξινόμηση όλων των άρθρων στην αρχική σελίδα ανά κατηγορία[22] σε φθίνουσα κατάταξη στην σελίδα http://localhost:5000/alphabeticalCat .

Παράλληλα δίνεται η δυνατότητα παρουσίασης των άρθρων ανά κατηγορία[23]. Η επιλογή της κατηγορίας γίνεται σε dropdown list στο navbar του site. Η λογική είναι ίδια όπως και στα υπόλοιπα services παρουσίασης των άρθρων. Η μόνη διαφορά είναι ότι το query αντί να ταξινομεί τα άρθρα τα συλλέγει ανά ένα ορισμένο κριτήριο στην συγκεκριμένη ανά την επιλεγμένη κατηγορία. Για κάθε κατηγορία το API ανοίγει διαφορετική σελίδα Υπάρχουν επομένως 10 διαφορετικές σελίδες. Κάθε κατηγορία στο τέλος του url αντιπροσωπεύει την κατηγορία άρθρων που παρουσιάζονται στην σελίδα.

1. http://localhost:5000/World
2. http://localhost:5000/Politics
3. http://localhost:5000/Society
4. http://localhost:5000/Economics
5. http://localhost:5000/Tech
6. http://localhost:5000/Culture
7. http://localhost:5000/Sport

[20] https://github.com/elenisproject/NewsBackend/blob/ac69ceff437ecc234026ded00d60d3d0f0e83a49/API/app/routes.py#L47

[21] https://github.com/elenisproject/NewsBackend/blob/ac69ceff437ecc234026ded00d60d3d0f0e83a49/API/app/routes.py#L60

[22] https://github.com/elenisproject/NewsBackend/blob/ac69ceff437ecc234026ded00d60d3d0f0e83a49/API/app/routes.py#L99

[23] https://github.com/elenisproject/NewsBackend/blob/ac69ceff437ecc234026ded00d60d3d0f0e83a49/API/app/routes.py#L205

8. http://localhost:5000/Environment

9. http://localhost:5000/Food

10. http://localhost:5000/Style

Στην συνέχεια αφού επιλέξει ο χρήστης τον τρόπο που θέλει να διασχίσει τα άρθρα μπορεί να επιλέξει κάποιο άρθρο προς ανάγνωση. Στην περίπτωση αυτή γίνεται ξανά ένα query στην βάση, αυτή την φορά όμως ζητάμε όλα τα δεδομένα που υπάρχουν στην βάση[24] όπως επίσης και τα δεδομένα των ομοίων άρθρων[25] στον πίνακα similar_articles για να προτείνει μετά την ανάγνωση του άρθρου την συνέχιση της ανάγνωσης του σε παρόμοια άρθρα. Η σελίδα κάθε άρθρου είναι διαφορετική όμως η μορφή της είναι ανάλογη με το http://localhost:5000/article/49298 . Το νούμερο που ακολουθεί στο τέλος του url είναι το id του άρθρου που διαβάζει την στιγμή αυτή ο χρήστης. Ανάλογα το άρθρο που επιλέξει υπάρχει και διαφορετικό id.

Επόμενο service είναι η συνολική παρουσίαση όλων των notebooks σε μία σελίδα[26], στην συνέχεια μπορεί να επιλεξει ο χρήστης ποιο notebook επιθυμεί να μελετήσει. Η κεντρική σελίδα παρουσίασης των άρθρων στην ουσία περιέχει μία εικόνα του γραφήματος που υπάρχει και το τίτλο του notebook που μπορείς να δεις. Η κεντρική σελίδα των notebooks βρίσκεται στο url: http://localhost:5000/analytics . Έπειτα μπορεί ο χρήστης να επιλέξει κάποιο notebook[27] για μία πιο διαδραστική ενασχόληση με τις δυνατότητες του notebook όπως και την εκτέλεση live time του κώδικα λεκτικής ανάλυσης. Κάθε notebook ανοίγει σε μία νέα σελίδα, κάθε τέτοια σελίδα έχει εμφωλευμένο ένα notebook το οποίο εκτελείται τοπικά στον υπολογιστή μου. Η διαδικασία αυτή αναφέρεται αναλυτικά στο κεφάλαιο 3.3 “Jupyter

[24] https://github.com/elenisproject/NewsBackend/blob/ac69ceff437ecc234026ded00d60d3d0f0e83a49/API/app/routes.py#L348

[25] https://github.com/elenisproject/NewsBackend/blob/ac69ceff437ecc234026ded00d60d3d0f0e83a49/API/app/routes.py#L359

[26] https://github.com/elenisproject/NewsBackend/blob/ac69ceff437ecc234026ded00d60d3d0f0e83a49/API/app/routes.py#L140

[27] https://github.com/elenisproject/NewsBackend/blob/ac69ceff437ecc234026ded00d60d3d0f0e83a49/API/app/routes.py#L144

Notebooks". Επομένως υπάρχουν 10 σελίδες για την παρουσία κάθε notebook χωριστά. Οι σελίδες είναι:

1. http://localhost:5000/article_size_per_category
2. http://localhost:5000/goodnews_badnews
3. http://localhost:5000/most_common_countries
4. http://localhost:5000/most_common_per_category
5. http://localhost:5000/most_news_source
6. http://localhost:5000/most_popular_authors
7. http://localhost:5000/percentage_top_authors
8. http://localhost:5000/popular_word_per_year
9. http://localhost:5000/quality_of_articles
10. http://localhost:5000/word_popularity_in_time

Όμοια αρχιτεκτονική έχει το κομμάτι των javascript charts. Υπάρχει δηλαδή μια κεντρική σελίδα παρουσίασης όλων των επιπλέον γραφημάτων[28] στη διεύθυνση http://localhost:5000/charts , ξανά ο χρήστης στο σημείο αυτό μπορεί να επιλέξει ποιο επιθυμεί να δει.. Οι επιμέρους σελίδες[29] των γραφημάτων για περεταίρω ανάλυση των δεδομένων βρίσκονται σε διαφορετική σελίδα με βάση το όνομα τους. Έχουμε επομένως τις παρακάτω έξι διευθύνσεις.

1. http://localhost:5000/articles_available_from_each_year
2. http://localhost:5000/beginner_reading
3. http://localhost:5000/categorize_sites
4. http://localhost:5000/articles_size
5. http://localhost:5000/countries
6. http://localhost:5000/sites_analysis

[28] https://github.com/elenisproject/NewsBackend/blob/ac69ceff437ecc234026ded00d60d3d0f0e83a49/API/app/routes.py#L112

[29] https://github.com/elenisproject/NewsBackend/blob/ac69ceff437ecc234026ded00d60d3d0f0e83a49/API/app/routes.py#L124

Επίσης σημαντική λειτουργία της ιστοσελίδας είναι ο κατηγοριοποιητής. Ο κατηγοριοποιητής[30] ο οποίος βρίσκεται στην διεύθυνση http://localhost:5000/classifier . Για να μεταβεί ο χρήστης στην σελίδα επιλέγει τον classifier από το navbar. Στην σελίδα αυτή βρίσκεται η φόρμα αναζήτησης η οποία δέχεται το κείμενο προς ταξινόμηση. Για να γίνει η ταξινόμηση πρέπει να πατηθεί το κουμπί αναζήτησης και τρέχει ένα script που υπολογίζει την κατηγορία όπως περιγράφεται αναλυτικά στο κεφάλαιο 3.4. Αφού υπολογιστεί η κατηγορία γίνεται ένα νέο query στην βάση και επιστρέφονται τα πρώτα δέκα άρθρα από την βάση από την ίδια κατηγορία. Τα άρθρα αυτά μαζί με την κατηγορία της πρόβλεψης επιστρέφονται στην σελίδα του classifier.

Τέλος υπάρχει το service του search. Το search bar βρίσκεται στο navbar. Στην φόρμα εκεί ο χρήστης μπορεί να πληκτρολογήσει την λέξη που τον ενδιαφέρει, το API θα πάρει την λέξη από την σελίδα και θα την στείλει στην βάση να αναζητήσει άρθρα που την περιέχουν[31]. Τα άρθρα επιστρέφονται και παρουσιάζονται στην σελίδα http://localhost:5000/search?q=διπλωματική . Η παρουσίαση τους γίνεται με αντίστοιχο τρόπο όπως στην αρχική σελίδα του site και ο χρήστης μπορεί να επιλέξει κάποιο αποτέλεσμα για ανάγνωση.

Συγκεντρωτικά τα url που περιγράφηκαν παραπάνω:

URL	PAGE
http://localhost:5000/home	home page containing all the articles in random
http://localhost:5000/newest	page with articles ordered by their date descending
http://localhost:5000/oldest	page with articles ordered by their date ascending

30 https://github.com/elenisproject/NewsBackend/blob/ac69ceff437ecc234026ded00d60d3d0f0e83a49/API/app/routes.py#L35

31 https://github.com/elenisproject/NewsBackend/blob/ac69ceff437ecc234026ded00d60d3d0f0e83a49/API/app/routes.py#L189

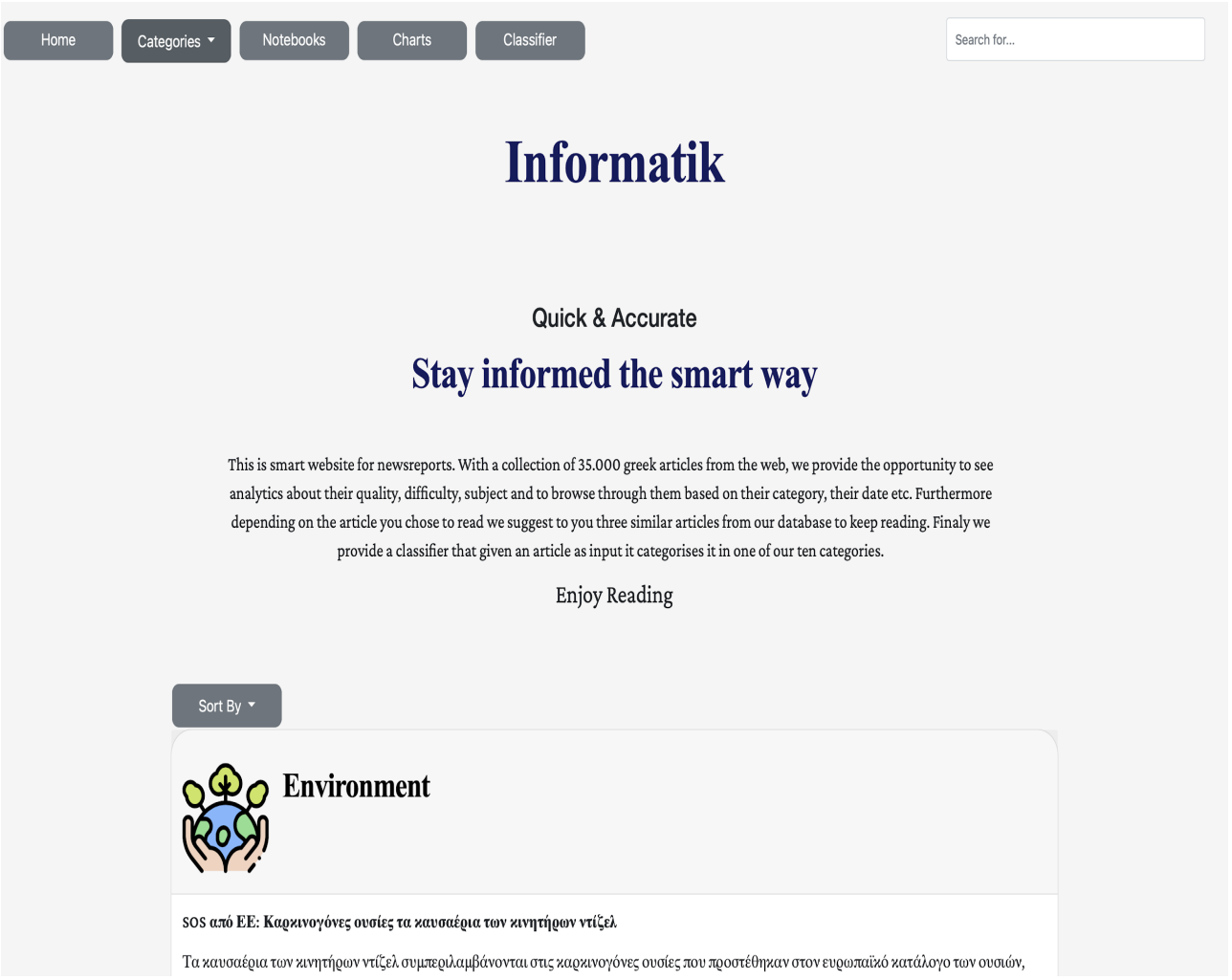
http://localhost:5000/alphabetical	page with articles ordered by their title ascending
http://localhost:5000/alphabeticalDesc	page with articles ordered by their title descending
http://localhost:5000/alphabeticalCat	page with articles ordered by their category ascending
http://localhost:5000/<Topic>	where topic is one of our ten categories
http://localhost:5000/article/<article_id>	the page where the whole article is presented
http://localhost:5000/analytics	main page with the notebooks layout
http://localhost:5000/<analyics_name>	page containing a specific notebook
http://localhost:5000/charts	main page with the chart layout
http://localhost:5000/<charts_name>	page containing a specific chart
http://localhost:5000/classifier	page with the classifier
http://localhost:5000/search?q=διπλωματικη	the url for the search service

Πίνακας 4.3

5. Περιήγηση

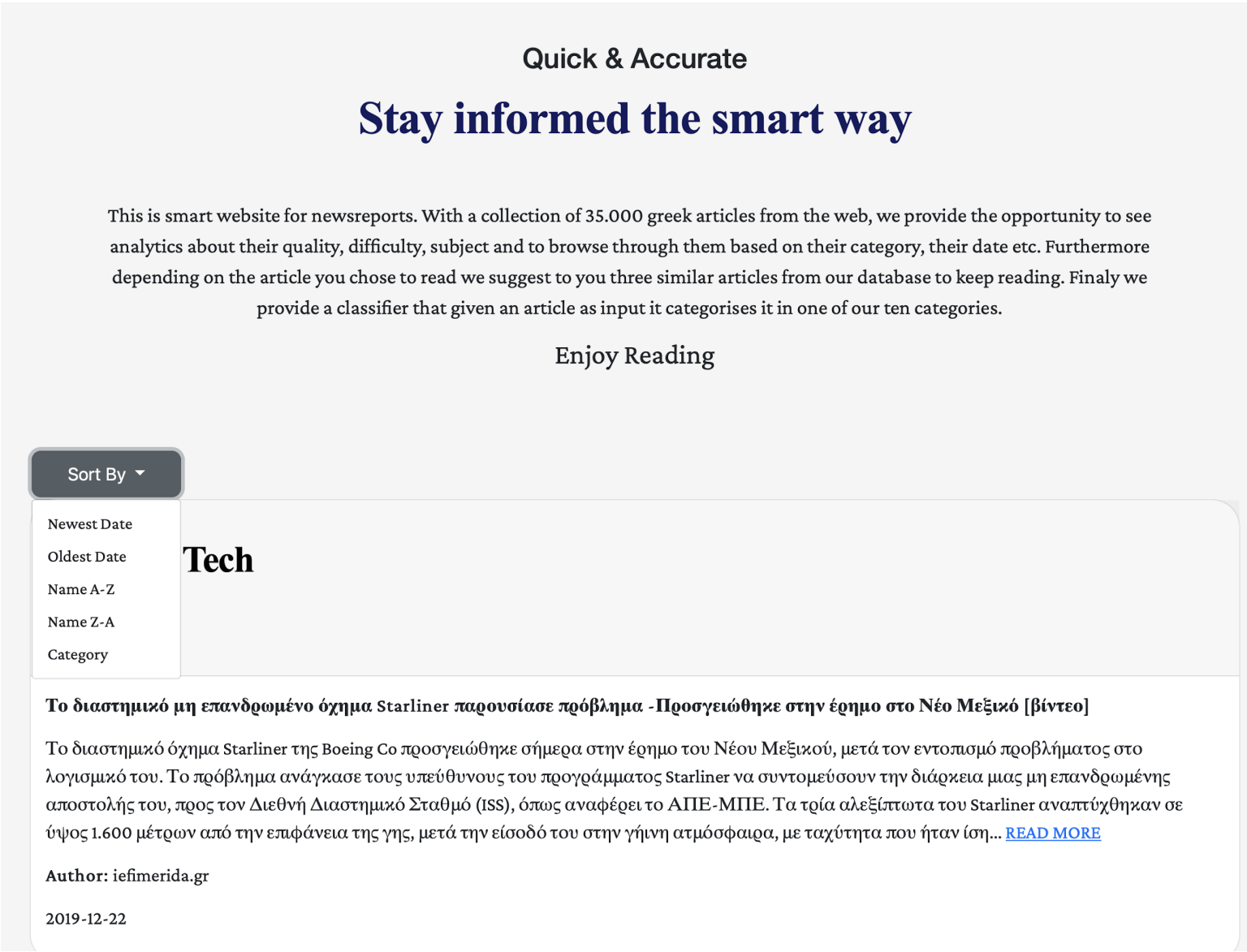
5.1. Home Page

Η ολική παρουσίαση της πλατφόρμας υπάρχει σε μορφή βίντεο στον παρακάτω σύνδεσμο του youtube: https://www.youtube.com/watch?v=L_pdCzFcpjc



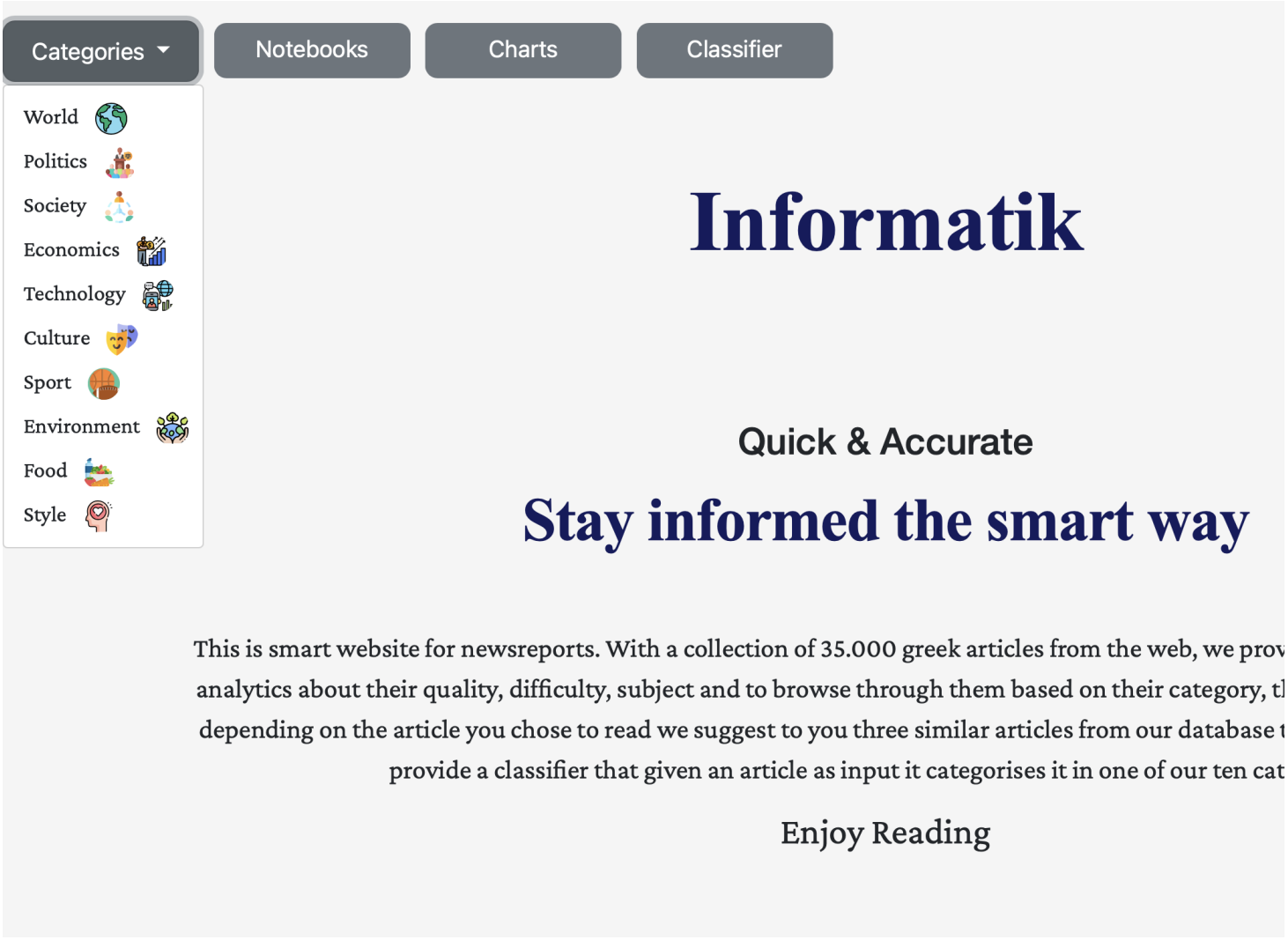
Εικόνα 5.1.1

Η αρχική σελίδα είναι η πρώτη σελίδα που βλέπει κάποιος χρήστης όταν συνδέεται στην ιστοσελίδα. Στο σημείο αυτό διατίθεται η συνοπτική παρουσίαση τους σκοπού της ιστοσελίδας. Στην συνέχεια ο χρήστης μπορεί να πλοηγηθεί στα διαθέσιμα άρθρα της βάσης. Τα άρθρα πρώτα παρουσιάζονται με τυχαία εντελώς σειρά. Υπάρχει όπως η δυνατότητα πλοήγησης με βάση κάποια ταξινόμηση. Η ταξινόμηση μπορεί να γίνει από το dropdown list πάνω αριστερά και μπορεί ο χρήστης να επιλέξει τα άρθρα να εμφανίζονται με βάση φθίνουσα ή αύξουσα ημερομηνία, φθίνουσα ή αύξουσα αλφαβητική σειρά του τίτλου και τέλος με βάση την φθίνουσα αλφαβητική σειρά των κατηγοριών.



Εικόνα 5.1.2

5.2. *Navigation Bar*



Εικόνα 5.2

Σε κάθε σελίδα του ιστότοπου εμφανίζεται το navigation bar. Από το σημείο αυτό ο χρήστης αποκτά πρόσβαση στις κύριες λειτουργία του ιστότοπου. Πρώτο κουμπί είναι το κουμπί επιστροφής στην αρχική σελίδα με την τυχαία διάταξη των άρθρων. Δεύτερο κουμπί από τα αριστερά είναι η λίστα με τις κατηγορίες των άρθρων. Επιλέγοντας μία από αυτές, εμφανίζονται σε τυχαία σειρά όλα τα άρθρα της επιλεγμένης κατηγορίας. Τρίτο κουμπί οδηγεί στην σελίδα με όλα τα διαθέσιμα notebooks. Τέταρτο κουμπί ανοίγει την σελίδα με τα

υπάρχοντα data charts. Το τελευταίο κουμπί της μπάρας, στέλνει τον χρήστη στην σελίδα του κατηγοριοποιητή. Φυσικά, όπως συνηθίζεται στα περισσότερα ui συστήματα, πάνω δεξιά υπάρχει η μπάρα αναζήτησης. Στην μπάρα αναζήτησης ο χρήστης έχει την δυνατότητα να αναζητήσει οποιαδήποτε λέξει. Μετά την αναζήτηση, επιστρέφονται όσα άρθρα περιέχουν την λέξης που πληκτρολόγησε ο χρήστης.

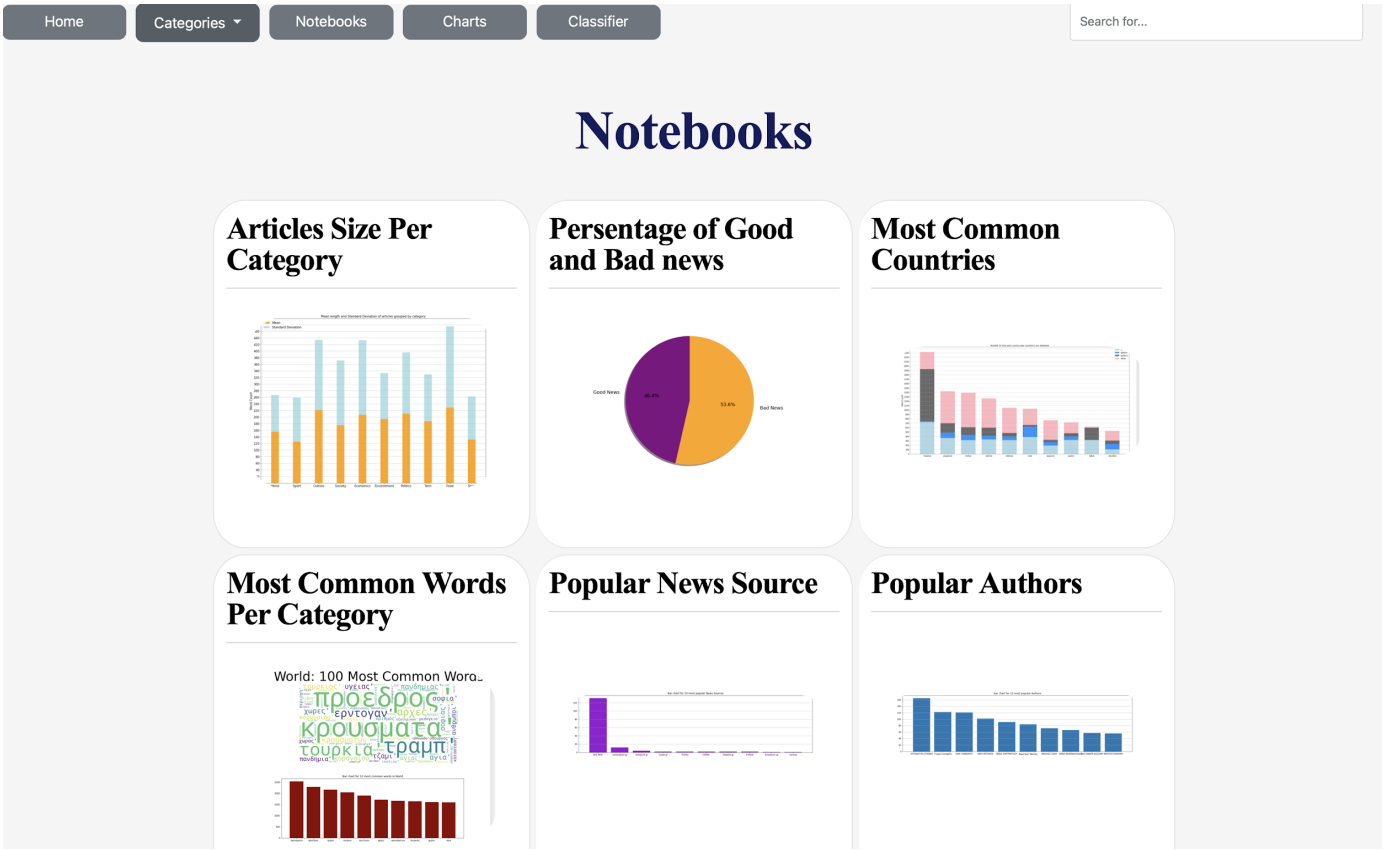
5.3. *Article Page*

Εικόνα 5.3

Εφόσον επιλεχθεί κάποιο άρθρο προς ανάγνωση ανοίγει νέα σελίδα. Στην νέα σελίδα παρουσιάζονται όλα τα δεδομένα που έχουμε συγκεντρώσει στην βάση μας για το άρθρο αυτό. Αρχικά υπάρχει ο τίτλος του άρθρου. Πάνω δεξιά από το τίτλο βρίσκεται η κατηγορία του άρθρου και η ημερομηνία συγγραφής. Το σημείο αυτό είναι και κουμπί το οποίο σε μεταφέρει στην σελίδα με όλα τα άρθρα της ίδιας κατηγορίας. Πάνω αριστερά υπάρχει ένα άλλο κουμπί

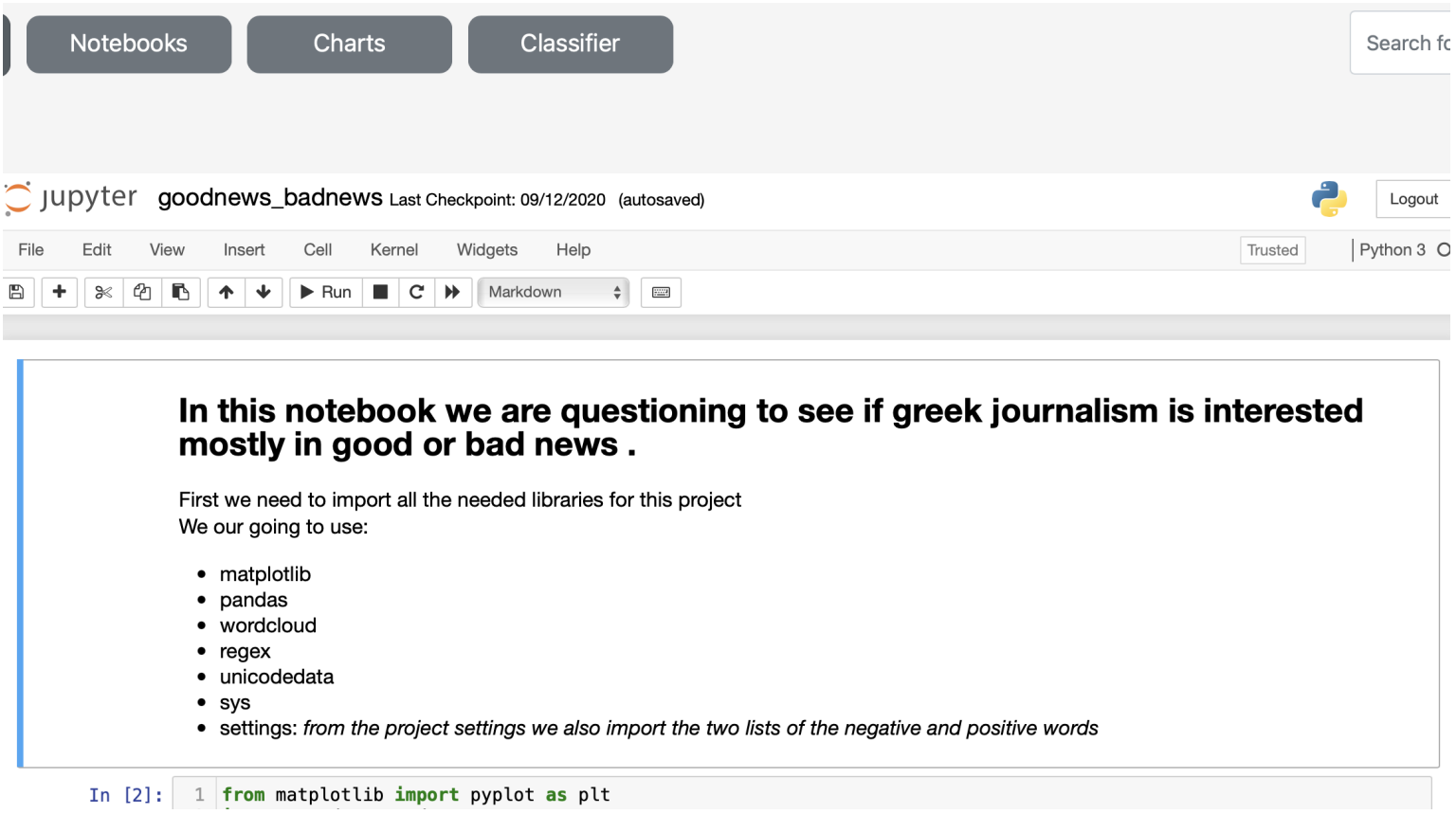
για την επιστροφή στην αρχική σελίδα. Μετά τον τίτλο υπάρχει το κύριο σώμα του άρθρου η πηγή και συγγραφέας. Με το τέλος του άρθρου πάνω από το footnote υπάρχουν τρεις προτάσεις όμοιων άρθρων για συνέχιση της ανάγνωσης. Στο σημείο αυτό όλο το πλαίσιο είναι clickable και πατώντας το, ανοίγει το συγκεκριμένα άρθρο.

5.4. *Jupyter Notebooks*



Εικόνα 5.4.1

Διατίθεται προς τον χρήστη που πλοηγείται στον ιστότοπο, σελίδα που παρουσιάζονται συνοπτικά όλα τα κατασκευασμένα notebooks. Στην σελίδα αυτή εμφανίζεται ο τίτλος κάθε notebook και το τελικό γράφημα που προκύπτει. Ο τίτλος και η φωτογραφία σε κάθε τετράγωνο πλαίσιο είναι clickable και ανοίγει το αντίστοιχο notebook.



Εικόνα 5.4.2

Αφού πατήσει ο χρήστης το notebook που επιθυμεί να δει, ανοίγει νέα σελίδα με το εμφωλευμένο notebook. Ο κώδικας στην σελίδα είναι εκτελέσιμος και εφόσον το επιθυμεί κάποιος μπορεί να τρέξει κάθε κομμάτι κώδικα χωριστά και να παράξει ζωντάνα εκ νέου το τελικό γράφημα της ανάλυσης. Τέλος πάνω αριστερά υπάρχει και το κουμπί επιστροφής στην σελίδα με τα συγκεντρωμένα notebooks και να επιλέξει από την αρχή.

5.5. *Javascript Charts*

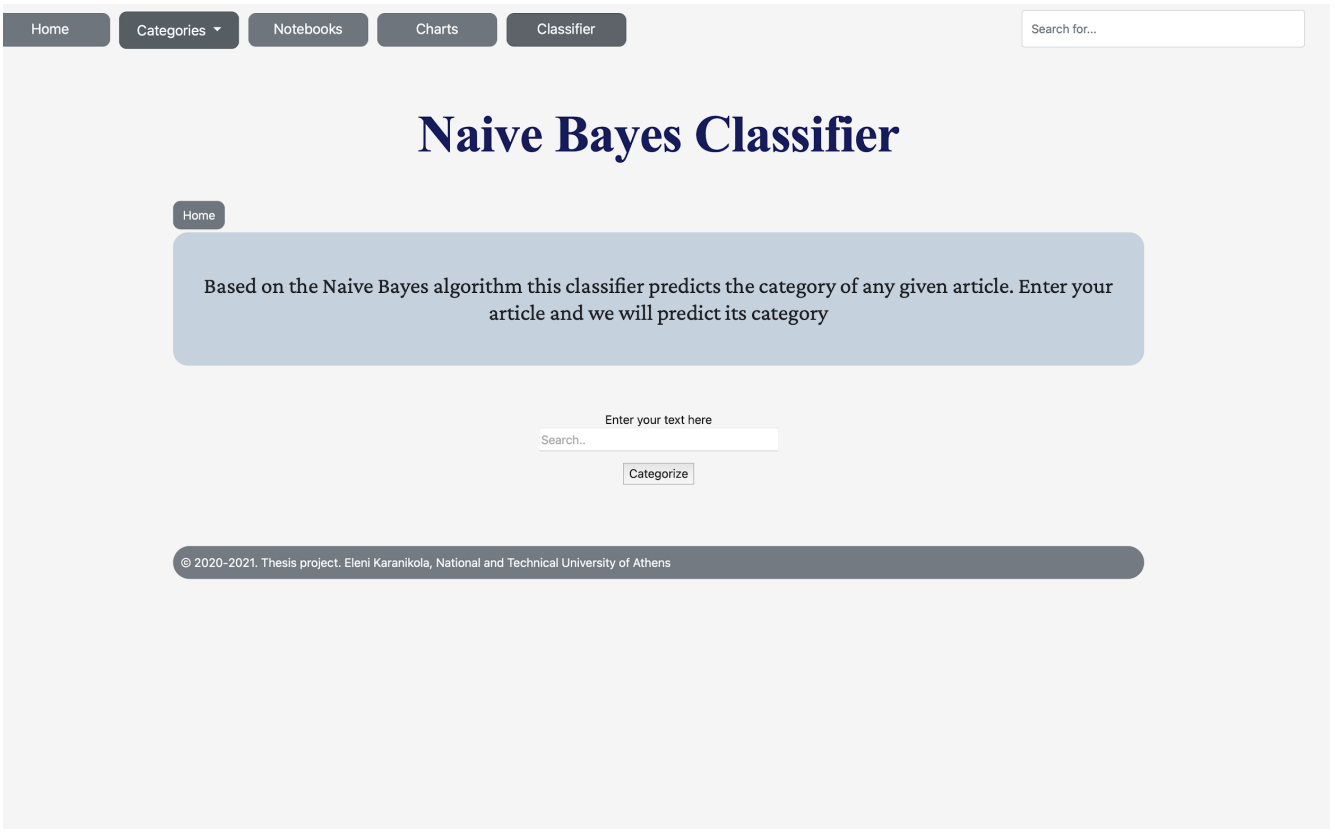
Εικόνα 5.5.1

Αντίστοιχα όπως τα Jupyter Notebooks υπάρχουν και τα Javascript Charts. Πρόκειται για περαιτέρω ανάλυση στα δεδομένα μας με τις αντίστοιχες οπτικοποιήσεις των αποτελεσμάτων. Από το navbar επιλέγοντας το κουμπί chart μεταφέρεσαι στην σελίδα με όλες τις διαθέσιμες επιλογές. Η παρουσίαση των επιλογών γίνεται με την χρήση του τίτλου και του τελικού διαγράμματος, και τα δύο είναι clickable ενεργοποιώντας την μεταφορά στην σελίδα με το αντίστοιχο chart που επιλέχθηκε.

Εικόνα 5.5.2

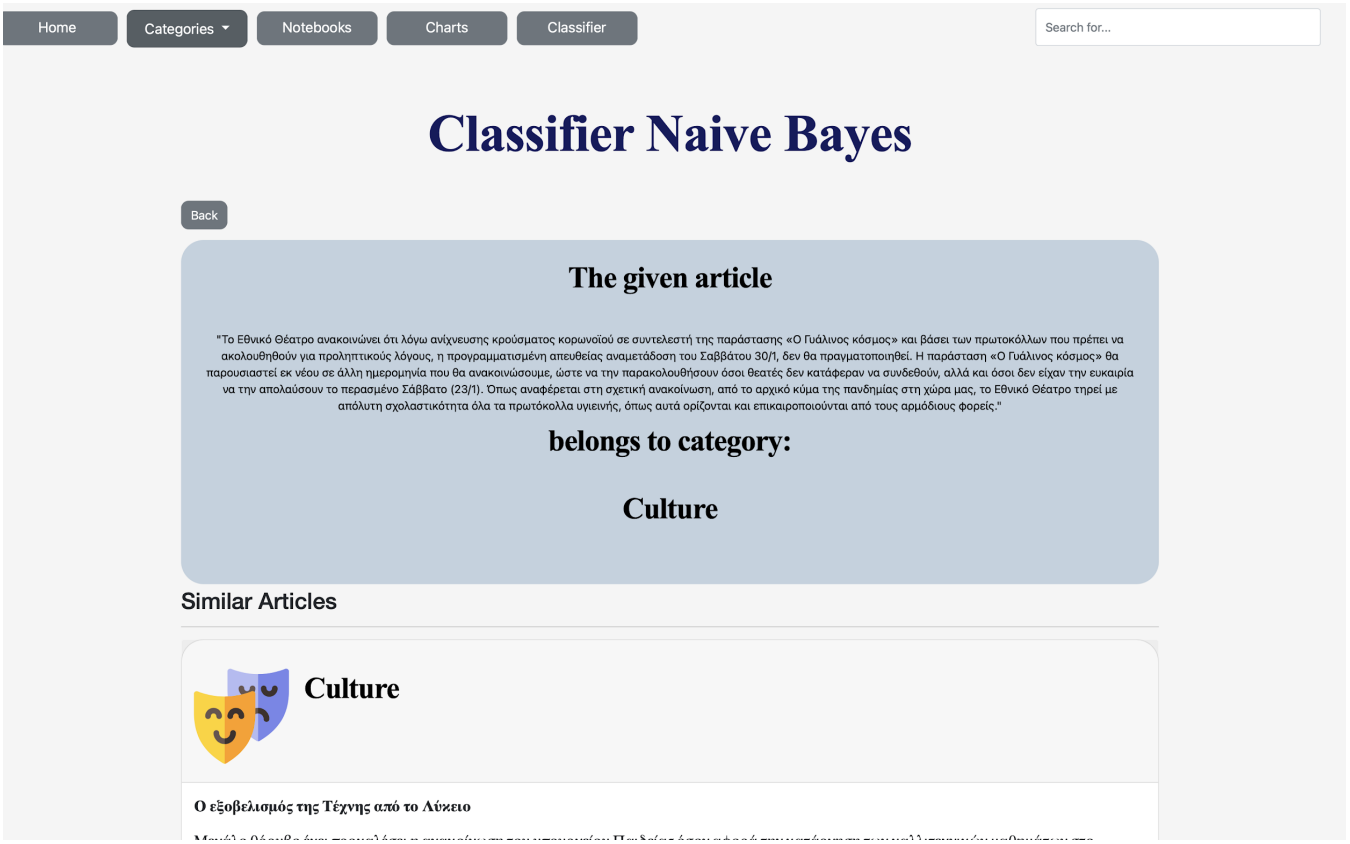
Επιλέγοντας κάποιο από τα πλαίσια ανοίγει το αντίστοιχο γράφημα όπως φαίνεται παραπάνω. Στην εικόνα εδώ βλέπουμε τον παγκόσμιο χάρτη και διατρέχοντας τον κέρσορα πάνω από κάθε χώρα εμφανίζεται το πλήθος των αναφορών που βρέθηκε στα άρθρα μας. Τα χρώματα σε κάθε χώρα συμβολίζουν την κατηγορία που ανήκει κάθε χώρα ανάλογα το πλήθος των αναφορών της. Χωρίσαμε τις χώρες σε χώρες με: καμία αναφορά, λιγότερο από 100 αναφορές, αναφορές μεταξύ 100 και 500, αναφορές μεταξύ 500 και 1000 και σε χώρες με παραπάνω από 1000 αναφορές. Αντίστοιχα με τα notebooks πάνω αριστερά υπάρχει το κουμπί επιστροφής στην σελίδα με όλα τα chart μαζεμένα.

5.6. *Classifier*



Εικόνα 5.6.1

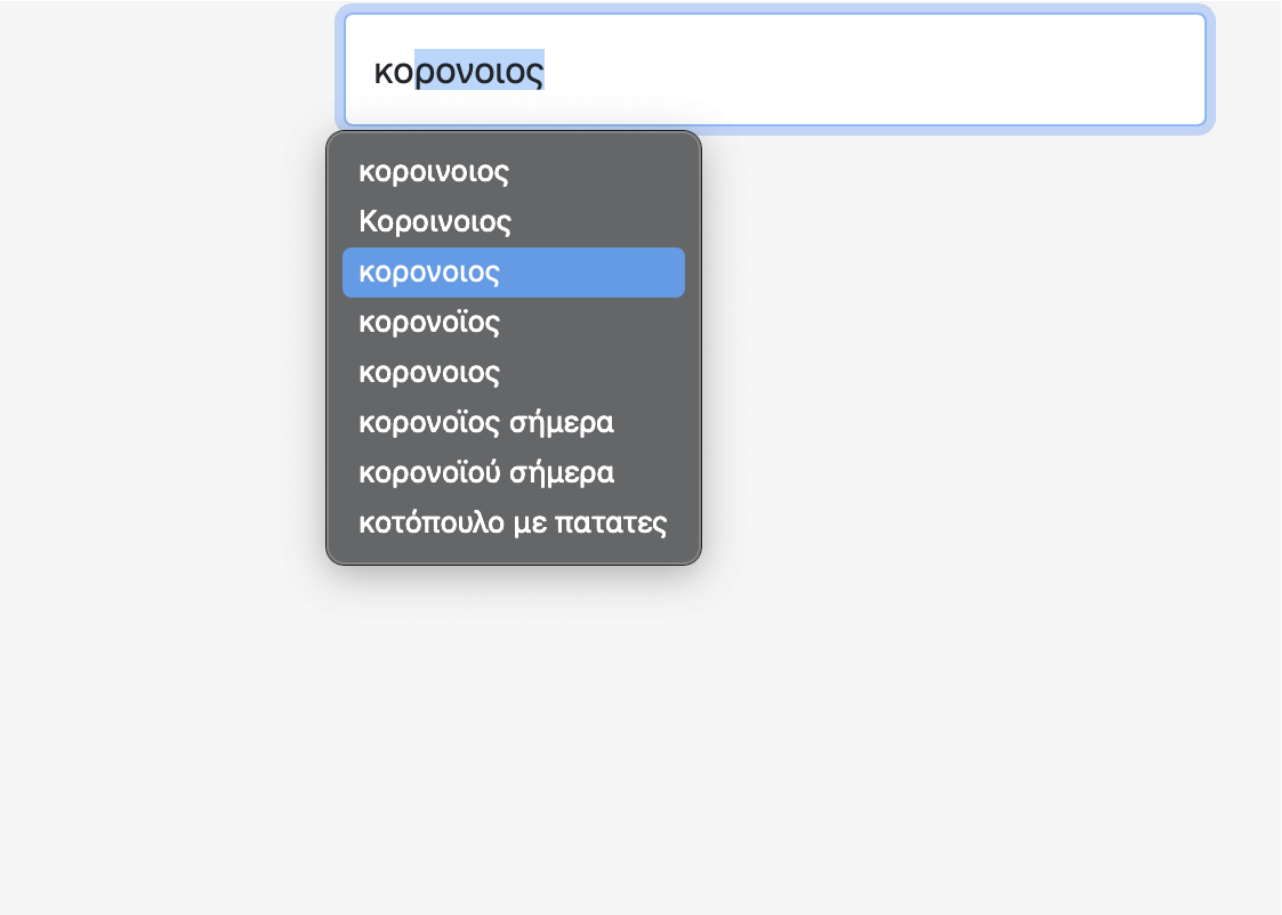
Πολύ σημαντική λειτουργία στην ιστοσελίδα είναι και η σελίδα του κατηγοριοποιητή. Επιλέγοντας την σελίδα αυτή από το Navbar ο χρήστης μπορεί να δει μία μικρή παρουσίαση του εργαλείου. Επίσης μπορεί να βάλει ένα δικό του άρθρο και να αφήσει τον κατηγοριοποιητή να το βάλει σε μία από τις δέκα κατηγορίες. Όπως σε όλες τις σελίδες μέχρι στιγμής πάνω αριστερά υπάρχει το κουμπί επιστροφής στην αρχική σελίδα Ένα μικρό παράδειγμα της λειτουργίας τους φαίνεται παρακάτω.



Εικόνα 5.6.2

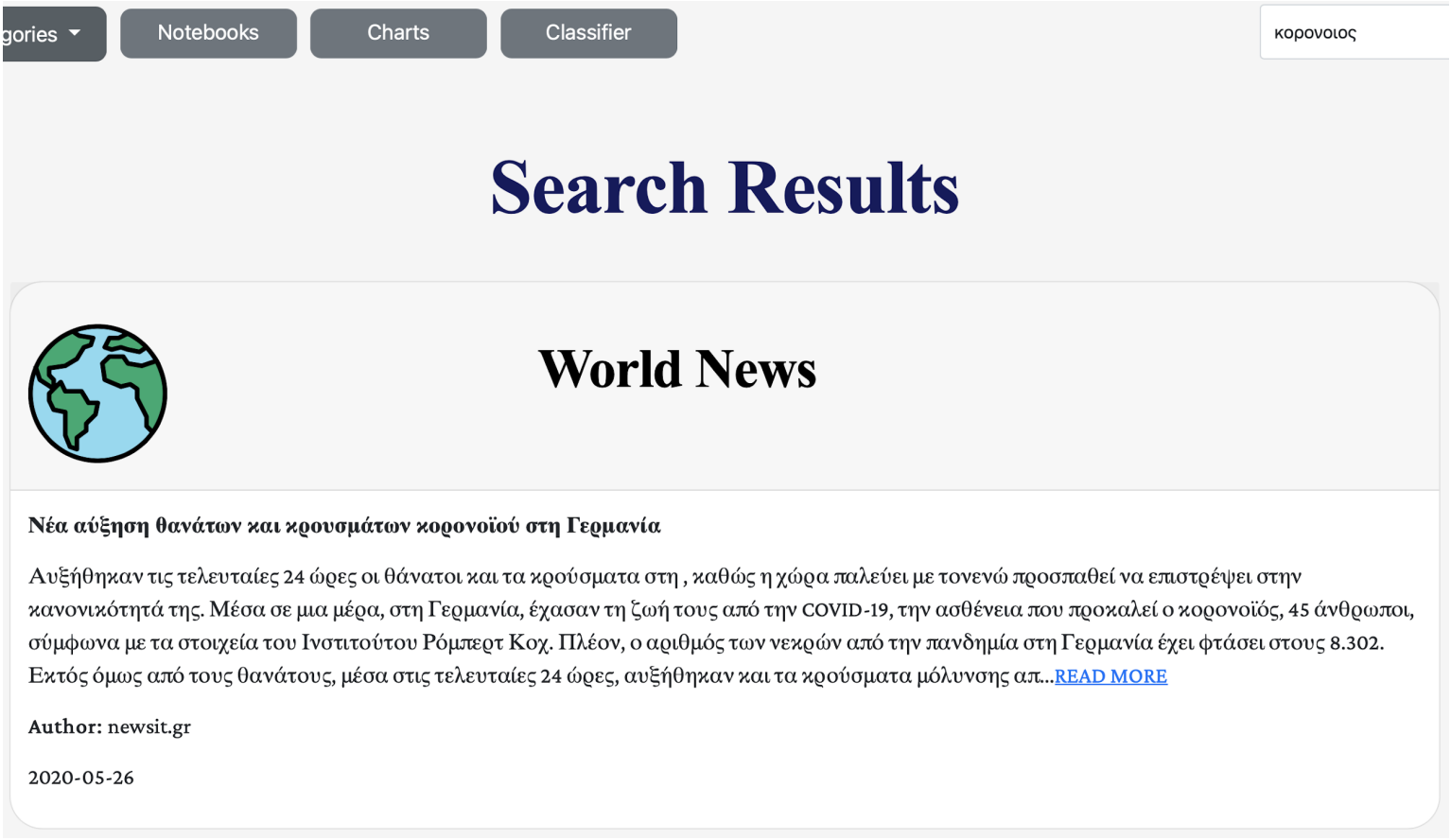
Στο παράδειγμα αυτό βάλαμε ένα άρθρο από την καθημερινή. Το άρθρο άνηκε στην κατηγορία πολιτισμός. Αντιγράψαμε το σώμα του άρθρου από την ιστοσελίδα στον κατηγοριοποιητή. Βλέπουμε επομένως ότι ο κατηγοριοποιητές κατέταξε σωστά το άρθρο στην κατηγορία Πολιτισμός. Κάτω από την πρόβλεψη εμφανίζονται 10 άρθρα από την βάση μας που ανήκουν στην ίδια κατηγορία.

5.7. *Search Bar*



Εικόνα 5.7.1

Τελευταία λειτουργία της ιστοσελίδα είναι η μπάρα αναζήτησης. Η μπάρα αναζήτησης βρίσκεται στο Navbar, αυτό σημαίνει ότι είναι εμφανείς από όλες τις σελίδες του ιστότοπου. Στο πλαίσιο αυτό ο χρήστης μπορεί να πληκτρολογήσει όποια λέξει επιθυμεί και θα επιστραφούν τα άρθρα της βάσης μας, που έχουν αυτή την λέξη στο σώμα τους. Στο παράδειγμα εδώ φαίνονται τα αποτελέσματα για την λέξη πρόεδρος.



Εικόνα 5.7.2

Η ολική παρουσίαση της πλατφόρμας υπάρχει σε μορφή βίντεο στον παρακάτω σύνδεσμο του youtube: https://www.youtube.com/watch?v=L_pdCzFcpjc

6. *Συμπεράσματα & Προτάσεις*

Στα πλαίσια της παρούσας διπλωματικής εργασίας αναπτύχθηκε ένα ισχυρό εργαλείο για την καλύτερη και πιο στοχευμένη ενημέρωση στην σύγχρονη εποχή. Το εργαλείο που αναπτύχθηκε παρέχει την δυνατότητα άμεσης πρόσβασης σε μεγάλη γκάμα άρθρων από διαφορετικούς συγγραφείς και διαφορετικά οπτικά πεδία, άρα δίνεται η πρόσβαση σε μία σφαιρικής και πολύπλευρη ενημέρωση. Επιπρόσθετα υπάρχει η παροχή της ανάλυσης των άρθρων ως προς το περιεχόμενο τους, την δυσκολία τους και το αντικείμενο ενασχόλησης τους. Επιπλέον λειτουργίες της εφαρμογής είναι η λειτουργία αναζήτησης και η λειτουργία πρότασης όμοιων άρθρων με βάση το άρθρο που επιλέγεται προς ανάγνωση. Τέλος διατίθεται προς χρήση ένας κατηγοριοποιητής κειμένου με ποσοστό επιτυχίας 63% . Η ανάγκη για την ύπαρξη ενός αντίστοιχου εργαλείου σε επίπεδο παραγωγής μπορεί να φανεί πολύ χρήσιμο για το μέσο άνθρωπο που δεν έχει χρόνο ή ενέργεια να αφιερώσει στην αντικειμενική του ενημέρωση. Όμως η δύναμη της πληροφορίας είναι πολύ ισχυρή για να είναι κατευθυνόμενη, γεγονός που καθιστά την ανάγκη αυτή ακόμα πιο επείγουσα.

Στα πλαίσια της ανάγκης για αντικειμενική και αναλυτική ενημέρωση η παρούσα εργασία θέτει τα θεμέλια, υπάρχουν όμως σημεία που επιδέχεται βελτιστοποίηση και περαιτέρω ανάπτυξη. Μερικά από τα σημεία που θα μπορούσαν να υλοποιηθούν είναι αρχικά το κομμάτι συνεχής ανανέωσης των άρθρων στη βάση. Προς το παρόν ο crawler εκτελέστηκε μία φορά και πάνω σε αυτά τα δεδομένα χτίστηκε η εφαρμογή. Για να βγει η εφαρμογή σε επίπεδο παραγωγής χρήσιμο θα ήταν η ανάπτυξη μιας λειτουργίας που θα ανανεώνει ανά χρονικά διαστήματα την βάση δεδομένων ώστε το υλικό να είναι επίκαιρο.

Δεύτερο σημείο που επιδέχεται βελτιστοποίηση είναι το κομμάτι της αναζήτησης λέξεων. Η αναζήτησης στην παρούσα διπλωματική υλοποιήθηκε με χρήση queries

σε επίπεδο API. Χρήσιμο θα ήταν να υλοποιηθεί μία πιο έξυπνη και κομψή υλοποίηση δεδομένου ότι υπάρχουν πολλά σύγχρονα εργαλεία που μπορούν να χρησιμοποιηθούν για την ανάπτυξη της λειτουργίας αυτής.

Όσων αφορά τον κατηγοριοποιητή. Το ποσοστό επιτυχίας του είναι 63% με την μέθοδο του Naive Bayes. Επόμενο βήμα σε αυτό είναι η αύξηση του ποσοστού επιτυχίας, όπως μελετήθηκε, με την αύξηση των λέξεων ως κριτήριο για την κατάταξη ενός κειμένου, η μέθοδος αύξανε το ποσοστό επιτυχίας ως και πάνω από 75%. Μία ακόμη δυνατότητα είναι να να γίνει διπλό layering. δηλαδή αφού βρούμε τις δύο πιθανότερες κατηγορίες στις οποίες ανήκει το άρθρο με τον πρώτο κατηγοριοποιητή, να κατασκευαστεί έναν δεύτερο κατηγοριοποιητή όπου θα κρατάει τα ποσοστά εμφάνισης στις δημοφιλεστερες λέξεις, αυτών των δύο κατηγοριών και έπειτα να αποφασίζει την κατηγορία στην οποία ανήκει το άρθρο. Υπάρχουν επομένως αρκετά που μπορούν να δοκιμαστούν για να αυξηθεί το ποσοστό επιτυχίας του κατηγοριοποιητή.

Τέλος στο σύστημα πρότασης άρθρων παρατηρήθηκε ότι η μέθοδος που χρησιμοποιήθηκε προτείνει συνήθως ως όμοια άρθρα, άρθρα μεγάλης έκτασης. Αυτό μπορεί να ερμηνευθεί εύκολα αφού ένα μεγαλύτερο άρθρο σε έκταση είναι πιθανό να περιέχει και περισσότερες από της λέξεις του κειμένου αναζήτησης. Επομένως έχουν καλύτερα ποσοστά επιτυχίας τα μεγάλα άρθρα. Μία επιπλέον παράμετρος που μπορεί να προσμετρηθεί λοιπόν είναι η έκταση του κειμένου, μπορεί δηλαδή γίνεται πρώτα μία διαίρεση του πλήθους των κοινών λέξεων με το πλήθος των λέξεων του κειμένου που εξετάζεται. Και με βάση το νούμερο αυτό να βρίσκονται τα ομοιότερα άρθρα.

Αυτές είναι κάποιες προτάσεις για την ανάπτυξη της παρούσας εργασίας. Η ανάγκη για την ύπαρξη ενός αντίστοιχου εργαλείου παραμένει και οι δυνατότητες του μπορούν να φέρουν σημαντική αλλαγή στην καθημερινότητα μας καθώς η ανάγκη για ενημέρωση είναι διαχρονική και άκρως απαραίτητη.

7. References

[1] C. Hu, Y. Li, Y. Wang and L. Wu, "Analysis of Hot News Based on Big Data," in *2018 IEEE/ACIS 17th International Conference on Computer and Information Science (ICIS)*, Singapore, Singapore, 2018 pp. 678-681.

doi: 10.1109/ICIS.2018.8466427

keywords: {natural language processing;industries;data mining;media;tag clouds;web pages;cultural differences}

url: https://doi.ieeecomputersociety.org/10.1109/ICIS.2018.8466427

[2] Y. Liang, Y. Zhao, D. Que, X. Zhang and C. Xu, "Online Fake Drug Detection System in Heterogeneous Platforms Using Big Data Analysis," in *2016 7th International Conference on Cloud Computing and Big Data (CCBD)*, Macau, China, 2016 pp. 308-311.

doi: 10.1109/CCBD.2016.067

keywords: {drugs;crawlers;biomedical imaging;character recognition;optical character recognition software;uniform resource locators;databases}

url: https://doi.ieeecomputersociety.org/10.1109/CCBD.2016.067

[3] S. Purohit, *et al.*, "Effective Tooling for Linked Data Publishing in Scientific Research," in *2016 IEEE Tenth International Conference on Semantic Computing (ICSC)*, Laguna Hills, CA, USA, 2016 pp. 24-31.

doi: 10.1109/ICSC.2016.87

keywords: {publishing;semantics;data mining;resource description framework;data visualization;triples (data structure);distributed databases}

url: https://doi.ieeecomputersociety.org/10.1109/ICSC.2016.87

[4] A. Sampaio and J. Barbosa, "A Study on Cloud Cost Efficiency by Exploiting Idle Billing Period Fractions," in *2017 IEEE International Conference on Smart Cloud (SmartCloud)*, New York City, NY, USA, 2016 pp. 138-143.

doi: 10.1109/IC2EW.2016.41

keywords: {scheduling;cloud computing;computational modeling;quality of service;scheduling algorithms;program processors}

url: https://doi.ieeecomputersociety.org/10.1109/IC2EW.2016.41

[5] Joyce, James. "Bayes' Theorem." *Stanford Encyclopedia of Philosophy*, 28 June 2003,https://plato.stanford.edu/entries/bayes-theorem/.

[6] H. Wang, S. Wang and C. Leng, "Learning Naive Bayes Classifiers with Incomplete Data," in *Artificial Intelligence and Computational Intelligence, International Conference on*, Shanghai, China, 2009 pp. 350-353.

doi: 10.1109/AICI.2009.402

keywords: {incomplete data;naive bayes classifier;iterative learning;gibbs sampling}

url: https://doi.ieeecomputersociety.org/10.1109/AICI.2009.402

[7] B. Liu, E. Blasch, Y. Chen, D. Shen and G. Chen, "Scalable sentiment classification for Big Data analysis using Naïve Bayes Classifier," in *2013 IEEE International Conference on Big Data*, Silicon Valley, CA, USA, 2013 pp. 99-104.

doi: 10.1109/BigData.2013.6691740

keywords: {}

url: https://doi.ieeecomputersociety.org/10.1109/BigData.2013.6691740

[8]C. Tseng, N. Patel, H. Paranjape, T. Lin and S. Teoh, "Classifying twitter data with Naïve Bayes Classifier," in *2013 IEEE International Conference on Granular Computing (GrC)*, Hangzhou, China China, 2012 pp. 294-299.

doi: 10.1109/GrC.2012.6468706

keywords: {blogs;computers}

url: https://doi.ieeecomputersociety.org/10.1109/GrC.2012.6468706

[9] T. Mori, S. Tamura and S. Kakui, "Incremental Estimation of Project Failure Risk with Naive Bayes Classifier," in *2013 ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM)*, Baltimore, Maryland, 2013 pp. 283-286.

doi: 10.1109/ESEM.2013.40

keywords: {software;estimation;predictive models;data models;bayes methods;capability maturity model}

url: https://doi.ieeecomputersociety.org/10.1109/ESEM.2013.40

[10] Roberts, Carl W., editor. *Text Analysis for the Social Sciences: Methods for Drawing Statistical Inferences From Texts and Transcripts*. Routledge, 2020.

[11] Ashlee Humphreys, Rebecca Jen-Hui Wang, Automated Text Analysis for Consumer Research, *Journal of Consumer Research*, Volume 44, Issue 6, April 2018, Pages 1274–1306, https://doi.org/10.1093/jcr/ucx104

[12] John, K. *Statistical Techniques for Data Analysis*. CRC Press, 2004.

[13] Banks, G.C., Woznyj, H.M., Wesslen, R.S. *et al.* A Review of Best Practice Recommendations for Text Analysis in R (and a User-Friendly App). *J Bus Psychol* 33, 445–459 (2018). https://doi.org/10.1007/s10869-017-9528-3

[14] Mitchell, R. *Web Scraping with Python*. 2 ed., O'Reilly Media, 2018.

[15] Bookstack. "Scrapy 2.0 Documentation." *bookstack.cn*, 05 03 2020, https://www.bookstack.cn/read/scrapy-2.0-en/733d56c872cb62ff.md.

[16] PYnative. "Python MySQL Database Connection Explained with Examples." *PYnative Python Programming*, 2020, https://pynative.com/python-mysql-database-connection/.

[17] Bush, Jos ephine. *Packt Publishing*. Learn SQL Database Programming, 2020.

[18] M. M. Patil, A. Hanni, C. H. Tejeshwar and P. Patil, "A qualitative analysis of the performance of MongoDB vs MySQL database based on insertion and retriewal operations using a web/android application to explore load balancing — Sharding in MongoDB and its advantages," *2017 International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC)*, Palladam, 2017, pp. 325-330, doi: 10.1109/I-SMAC.2017.8058365.

[19] G. Ongo and G. P. Kusuma, "Hybrid Database System of MySQL and MongoDB in Web Application Development," *2018 International Conference on Information Management and Technology (ICIMTech)*, Jakarta, 2018, pp. 256-260, doi: 10.1109/ICIMTech.2018.8528120.

[20] C. Curino, H. Moon, M. Ham and C. Zaniolo, "The PRISM Workwench: Database Schema Evolution without Tears," in *2013 IEEE 29th International Conference on Data Engineering (ICDE)*, null, 2009 pp. 1523-1526.
doi: 10.1109/ICDE.2009.46 url: https://doi.ieeecomputersociety.org/10.1109/ICDE.2009.46

[21] F. Mili, "Data analysis in scientific databases," in *Proceedings of 1993 IEEE Conference on Tools with Al (TAI-93)*, Boston, MA, USA, 1993 pp. 442,443,444.
doi: 10.1109/TAI.1993.633993
url: https://doi.ieeecomputersociety.org/10.1109/TAI.1993.633993

[22] X. Wu, "Data Mining: Artificial Intelligence in Data Analysis," in *Web Intelligence, IEEE / WIC / ACM International Conference on*, Beijing, China, 2004 pp.7-7.
doi: 10.1109/WI.2004.10000
url: https://doi.ieeecomputersociety.org/10.1109/WI.2004.10000

[23] K. Almgren, M. Kim and J. Lee, "Mining Social Media Data Using Topological Data Analysis," in *2017 IEEE International Conference on Information Reuse and Integration (IRI)*, San Diego, CA, 2017 pp. 144-153.
doi: 10.1109/IRI.2017.41
url: https://doi.ieeecomputersociety.org/10.1109/IRI.2017.41

[24] X. Wu, "Data Mining: Artificial Intelligence in Data Analysis," in *Web Intelligence, IEEE / WIC / ACM International Conference on*, Beijing, China, 2004pp. 7-7.
doi: 10.1109/WI.2004.10000
url: https://doi.ieeecomputersociety.org/10.1109/WI.2004.10000

[25]Sharma, Abhishek. “A Beginner’s Guide to Exploratory Data Analysis (EDA) on Text Data (Amazon Case Study).” *Analytics Vidhya*, 27 April 2020, https://www.analyticsvidhya.com/blog/2020/04/beginners-guide-exploratory-data-analysis-text-data/.

[26] Li, Susan. “A Complete Exploratory Data Analysis and Visualization for Text Data.” *towards data science*, 19 March 2019, https://towardsdatascience.com/a-complete-exploratory-data-analysis-and-visualization-for-text-data-29fb1b96fb6a.

[27] Es, Shahul. “Exploratory Data Analysis for Natural Language Processing: A Complete Guide to Python Tools.” *neptune.ai*, 14 January 2020, https://neptune.ai/blog/exploratory-data-analysis-natural-language-processing-tools.

[28] J. James, T. Moh and C. Edwards, "Web-Based Visualization of Marine Environmental Data: Performance Analysis of a MatPlotLib Implementation," in *2016 International Conference on Collaboration Technologies and Systems (CTS)*, Orlando, FL, 2016 pp. 288-293.
doi: 10.1109/CTS.2016.0061
url: https://doi.ieeecomputersociety.org/10.1109/CTS.2016.0061

[29] H. Wu, F. Liu, L. Zhao and Y. Shao, "Data Analysis and Crawler Application Implementation Based on Python," in *2020 International Conference on Computer Network, Electronic and Automation (ICCNEA)*, Xi'an, China, 2020 pp. 389-393.
doi: 10.1109/ICCNEA50255.2020.00086
url: https://doi.ieeecomputersociety.org/10.1109/ICCNEA50255.2020.00086

[30] Bevans, Rebecca. “Statistical tests: which one should you use?” *Scribbr*, 28 January 2020, https://www.scribbr.com/statistics/statistical-tests/.

[31] Gandhi, Rohith. “Naive Bayes Classifier.” *towards data science*, 5 May 2018, https://towardsdatascience.com/naive-bayes-classifier-81d512f50a7c.

[32] Chauhan, Nagesh Singh. “Naïve Bayes Algorithm: Everything you need to know.” *KDnuggets*,2020, https://www.kdnuggets.com/2020/06/naive-bayes-algorithm-everything.html.

[33] Department of Industrial and Systems Engineering, The Hong Kong Polytechnic University,. “Is Naïve Bayes a Good Classifier for Document Classification?” *International Journal of Software Engineering and Its Applications*, vol. 5, no. 3, 2011, http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.643.6611&rep=rep1&type=pdf.

[34] Mishra, Aditya. “Metrics to Evaluate your Machine Learning Algorithm.” *towards data science*,2018, https://towardsdatascience.com/metrics-to-evaluate-your-machine-learning-algorithm-f10ba6e38234.

[35] P. Sajda, A. Gerson, K. -. Muller, B. Blankertz and L. Parra, "A data analysis competition to evaluate machine learning algorithms for use in brain-computer interfaces," in *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 11, no. 2, pp. 184-185, June 2003, doi: 10.1109/TNSRE.2003.814453

[36] Arora, Sanjeev. *Computational Complexity: A Modern Approach*. Cambridge University Press., https://books.google.gr/books?hl=en&lr=&id=nGvI7cOuOOQC&oi=fnd&pg=PA9&dq=Computational+Complexity:+A+Modern+Approach.+Cambridge+University+Press.,&ots=Dab3zLaDov&sig=IJmjmUnshaAqn20Uq86HJKsMVHM&redir_esc=y#v=onepage&q=Computational%20Complexity%3A%20A%20Modern%20Approach.%20Cambridge%20University%20Press.%2C&f=false

[37] Belin, A., Lewkowycz, A. & Sárosi, G. Complexity and the bulk volume, a new York time story. *J. High Energ. Phys.* 2019, 44 (2019). https://doi.org/10.1007/JHEP03(2019)044

[38] *Tiziano Piccardi, Michele Catasta, Leila Zia, and Robert West. 2018. Structuring Wikipedia Articles with Section Recommendations. In The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval(SIGIR '18). Association for Computing Machinery, New York, NY, USA, 665–674. DOI:https://doi.org/10.1145/3209978.3209984*

[39] P. Vogel, T. Klooster, V. Andrikopoulos and M. Lungu, "A Low-Effort Analytics Platform for Visualizing Evolving Flask-Based Python Web Services," *2017 IEEE Working Conference on Software Visualization (VISSOFT)*, Shanghai, 2017, pp. 109-113, doi: 10.1109/VISSOFT.2017.13

[40] *R.-L. Liu International Journal of Knowledge Content Development & Technology Vol.7, No.3, 5-27 (September, 2017)*

[41] Wang, S., Koopman, R. Clustering articles based on semantic similarity. *Scientometrics*111, 1017–1031 (2017). https://doi.org/10.1007/s11192-017-2298-x

[42] Bernard N., Weber J., Forestier G., Hassenforder M., Latard B. (2020) Knowledge-Based Categorization of Scientific Articles for Similarity Predictions. In: Hall M., Merčun T., Risse T., Duchateau F. (eds) Digital Libraries for Open Knowledge. TPDL 2020. Lecture Notes in Computer Science, vol 12246. Springer, Cham. https://doi.org/10.1007/978-3-030-54956-5_11

[43] P. Vogel, T. Klooster, V. Andrikopoulos and M. Lungu, "A Low-Effort Analytics Platform for Visualizing Evolving Flask-Based Python Web Services," in *2017 IEEE Working Conference on Software Visualization (VISSOFT)*, Shanghai, China, 2017 pp. 109-113.
doi: 10.1109/VISSOFT.2017.13

url: https://doi.ieeecomputersociety.org/10.1109/VISSOFT.2017.13

[44] Ashley D. (2020) Using Flask and Jinja. In: Foundation Dynamic Web Pages with Python. Apress, Berkeley, CA. https://doi.org/10.1007/978-1-4842-6339-6_5

[45] Ashley D. (2020) Comparing CGI, SSI, Flask, and Django. In: Foundation Dynamic Web Pages with Python. Apress, Berkeley, CA. https://doi.org/10.1007/978-1-4842-6339-6_7

[46] Pallets. “Flask Documentation (1.1.x).” *Flask*, https://flask.palletsprojects.com/en/1.1.x/.

[47] Mitchell, R. *Web Scraping with Python*. 2 ed., O'Reilly Media, 2018.

[48] Suppala, Kavya, and Narasinga Rao. “Sentiment Analysis Using Naïve Bayes Classifier.” *International Journal of Innovative Technology and Exploring Engineering (IJITEE)*, vol. 8, no. 8, 2019, http://www.zeynepaltan.info/3-SentimentAnalysiswithNAiveBAyes.pdf.

[49] Chen X., Zeng G., Zhang Q., Chen L., Wang Z. (2018) Classification of Medical Consultation Text Using Mobile Agent System Based on Naïve Bayes Classifier. In: Long K., Leung V., Zhang H., Feng Z., Li Y., Zhang Z. (eds) 5G for Future Wireless Networks. 5GWN 2017. Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, vol 211. Springer, Cham. https://doi.org/10.1007/978-3-319-72823-0_35

[50] Liu, P., Zhao, Hh., Teng, Jy. *et al.* Parallel naive Bayes algorithm for large-scale Chinese text classification based on spark. *J. Cent. South Univ.* 26, 1–12 (2019). https://doi.org/10.1007/s11771-019-3978-x

[51] B. Sri Nandhini and J. I. Sheeba. 2015. Cyberbullying Detection and Classification Using Information Retrieval Algorithm. In Proceedings of the 2015 International Conference on Advanced Research in Computer Science Engineering & Technology (ICARCSET 2015) (ICARCSET '15). Association for Computing Machinery, New York, NY, USA, Article 20, 1–5. DOI:https://doi.org/10.1145/2743065.2743085

[52] Lops, Pasquale & de Gemmis, Marco & Semeraro, Giovanni. (2011). Content-based Recommender Systems: State of the Art and Trends. 10.1007/978-0-387-85820-3_3.

[53] Verma, Mayuri. (2017). Company Recommender System using Text Mining and Machine Learning in R.