ΤΗΜΜΥ
Αναφορά εργασίας στην Εξόρυξη Δεδομένων
Εαρινό Εξάμηνο 2020

"Μια ανάλυση των αμερικάνικων εκλογών από την σκοπιά του data mining"

Νίκος Πλέσσας - ΑΕΜ: 615

1. Περιγραφή Εργασίας

Σε αυτήν την εργασία προσπάθησα να κάνω μια ανάλυση 2 του λόγου υποψηφίων για τις Αμερικάνικες Εκλογές του 2020, του σημερινού Προέδρου Donald Trump και του Bernie Sanders, γερουσιαστή και υποψήφιου για το χρίσμα των Δημοκρατικών. Βασική πηγή πληροφορίας για αυτήν την ανάλυση είναι τα tweets των 2 υποψηφίων.
Χρησιμοποιώντας τον αλγόριθμο k-means για συσταδοποίηση προσπάθησα να εντοπίσω θεματικές στα tweets τους και να εξάγω συμπεράσματα για πιθανές θεματικές που σχηματίζονται, να διαπιστώσω αν μπορούμε να εξάγουμε συμπεράσματα για την πολιτική γραμμή του καθενός. Επίσης για το πως αναφέρονται στους πολιτικούς τους αντιπάλους, ποια #hashtags χρησιμοποιούν περισσότερο, αν τα tweets τους ειναι περισσότερο αρνητικά ή θετικά φορτισμένα (sentiment analysis).
Στο 2ο επίπεδο της ανάλυσης, χρησιμοποίησα το εργαλείο του sentiment analysis για να διαπιστώσω τι γνώμη έχουν τα media για τους 2 υποψηφίους καθώς και άλλοι χρήστες του twitter.
Στο τέλος δημιούργησα τις εξής 2 εφαρμογές:
Έναν ταξινομητή (classifier) τον οποίο "εκπαίδευσα" με τα tweets των 2 υποψηφίων ωστε να κάνει μια εκτίμηση του ιδιοκτήτη του tweet.
Ένα μικρό πρόγραμμα το οποίο κάνει σε πραγματικό χρόνο sentiment analysis tweets χρηστών που αναφέρονται σε κάθε υποψήφιο.

2. Περιγραφή της Ανάλυσης

Η ανάλυση των tweets και γενικά κειμένου είναι ένα πρόβλημα επεξεργασίας φυσικής γλώσσας (Natural Language Proccesing - NLP). Η μεθοδολογία που ακολούθησα είναι η εξής:
Προεπεξεργασία δεδομένων:

- Data scraping για την ανάκτηση των δεδομένων μέσω crawlers σε python
- Φορτωση των .csv αρχείων των datasets σε pandas dataframe
- Μετατροπή σε πεζά , αφαίρεση μικρών λέξεων (<2 χαρακτήρες), αφαίρεση stopwords (λέξεων που δεν προσφέρουν κάτι στο νοήμα του κειμένου πχ there, and, me κτλ), αφαίρεση #hashtags & @usernames, αφαίρεση σημειων στίξης.
- Tokenization: Χωρισμός κάθε tweet σε λέξεις (tokens)
- Lemmatization: απάλοιφή καταλήξεων πχ studies,studying -> study
- Dekonization: Δημιουργία "καθαρών" strings για την ανάλυση

Μετά την προεπεξεργασία έγιναν τα ακόλουθα βήματα για κάθε υποψήφιο:

- Clustering των tweets με τον αλγόριθμο K-means για ομαδοποίηση συναφών tweets (επέλεξα αυθαίρετα k=40)
- Sentiment analysis κάθε tweet
- Εύρεση των #hashtags και @usernames που χρησιμοποιεί περισσότερο κάθε υποψήφιος
- Network analysis: Συνδεδεμένος γράφος που αναπαριστά τις συνδέσεις των λέξεων που χρησιμοποιούνται συχνά (κατώφλι 80 λέξεις)
- Δημιουργία wordclouds
- Εύρεση των tweets που αναφέρονται στον άλλον υποψήφιο, sentiment analysis + clustering
- Τα αποτελέσματα των παραπάνω οπτικοποιήθηκαν με γραφήματα και υπάρχουν στην παρουσίαση μου.

Για την ανάλυση δημοφιλίας κάθε υποψήφιου έκανα sentiment analysis στα ακόλουθα datasets:

- 1500 tweets από το CNN politics
- 25.000 tweets χρηστών που αναφέρουν έναν από τους 2 υποψηφίους
- 2.000 τίτλοι και περιλήψεις άρθρων από διάφορα μέσα μαζικής ενημέρωσης

Όλοι οι παραπάνω κώδικες γράφτηκαν στο Google Colab, μια online εκδοχή jupyter notebooks που προσφέρει η Google στην οποία οι περισσότερες βιβλιοθήκες για data science/machine learning είναι προεγκατεστημένες. Επίσης για τους υπολογισμούς δεν χρησιμοποιείται η υπολογιστική δύναμη του μηχανήματός σου αλλά του virtual machine που σου προσφέρει η Google.

3. Εφαρμογές

Ο **ταξινομητής** δημιουργήθηκε ως εξής:

Αρχικά ένωσα τα 2 ξεχωριστά datasets σε ένα και έκανα την γνωστή προπεξεργασία. Μετά επέλεξα ως μοντέλο τον Multinomail Naive Bayes Classifier, τον οποία "εκπαίδευσα" με είσοδο τον πίνακα που προκύπτει από το TfidVectorizer (μετατροπή του κειμένου σε πίνακα συχνότητας εμφάνισης κάθε λέξης).
Ως feature variable X επιλέχθηκαν τα (μπερδεμένα) tweets και ως target variable Y η ταυτότητα του συγγραφέα κάθε tweet (κωδικοποημένη σε 0,1).

Το accuracy score προέκυψε 0.954.

Ως εφαρμογή για τον χρήστη δημιούργησα ένα υποτυπώδες παραθυρικό front end με την

βιβλιοθήκη tkinter της python στο οποίο ζητάει από τον χρήστη να εισάγει ένα tweet από κάποιον από τους 2 υποψήφιους και του απαντάει με το ποιος υποψήφιος το έγραψε. Ο ταξινομητής δουλεύει άριστα με μεγάλη ακρίβεια και σε μεταγενέστερα tweets που δεν υπάρχουν στο training dataset.

Live sentiment analysis:

Με χρήση της βιβλιοθήκης tweepy δημιουργώ μια κλάση listener η οποία ξεκινάει ενα stream από την στιγμή που θα αρχίσει να τρέχει και προς τα πίσω στον χρόνο, φιλτράρει το stream για ξεχωρίσει τα tweets που μας ενδιαφέρουν (στην προκειμένη tweets χρηστών που αναφέρονται στους 2 υποψήφιους), τα "καθαρίζει" και κάνει sentiment analysis στο καθένα. Μετά τα αποθηκεύει σε πραγματικό χρόνο σε ενα .csv.
Ένα δεύτερο πρόγραμμα διαβάζει επίσης σε πραγματικό χρόνο το .csv και σχεδιάζει το αθροιστικό sentiment για κάθε υποψήφιο.

4. Συμπεράσματα

Θεωρώ πως το κομματι της ανάλυσης μας βοηθά αρκετά να εξάγουμε συμπεράσματα για τον κάθε υποψήφιο, δίνοντας μια καλή σύνοψη πχ των πολιτικών του θέσεων, του "υφους" γραφής. Σε ένα project πρόβλεψης του νικητή το sentiment analysis θα μπορούσε να βοηθήσει στην πρόβλεψη των αποτελεσμάτων με μια πιο συγκεκριμένη στόχευση πχ κρίσιμες πολιτείες.

Ο ταξινομητής έχει τόσο μεγάλη επιτυχία γιατί οι 2 υποψήφιοι χρησιμοποιούν πολύ διαφορετικό ύφος γραφής. Πειραματικά δοκίμασα να τρέξω το ίδιο πρόγραμμα αλλά με dataset tweets toy Sanders και το Joe Biden (αντίπαλοι για το χρίσμα των Δημοκρατικών). Σε αυτή την περίπτωση το accuracy έπεσε στο 0.69 και σε πολλές περιπτώσεις δεν έβρισκε σωστά το ποιος το έγραψε.