ΑΠΑΛΛΑΚΤΙΚΗ ΕΡΓΑΣΙΑ ΣΤΟ ΜΑΘΗΜΑ ΤΕΧΝΙΚΕΣ ΕΞΟΡΥΞΗΣ ΔΕΔΟΜΕΝΩΝ

Η εργασία είναι **ατομική**

Προθεσμία υποβολής **11-06-2020, 23:55**

Σκοπός είναι να δημιουργήσετε ένα σύστημα μηχανικής μάθησης που λαμβάνει σαν είσοδο ένα σχόλιο (comment) και ταξινομεί αυτό το σχόλιο ως προσβλητικό ή όχι προσβλητικό (ουδέτερο). Αυτό το σύστημα θα εκπαιδευτεί σε ένα σύνολο δεδομένων με σχόλια στα οποία οι ετικέτες (προσβολή ή μη προσβολή) είναι γνωστά. Προκειμένου να μετρηθεί η αποτελεσματικότητά του θα χρησιμοποιήσετε τους παρακάτω αλγόριθμους ταξινόμησης : Naive Bayes, SVM, Random Forest. Στο φάκελο αυτή της εργασίας θα βρείτε τα δεδομένα εκπαίδευσης αλλά και δοκιμής (train, test) σε μορφή CSV.

Data Set

Τα δεδομένα που σας δίνονται αποτελούνται από 6,182 σχόλια (comments) που έχουν συλλεχθεί από διαδικτυακά φόρουμ. Είναι προσημασμένα (pre-labeled) με τις τιμές 1 (προσβλητικά σχόλια) ή 0 (ουδέτερα σχόλια). Το σύνολο των δεδομένων έχει χωριστεί σε δεδομένα εκπαίδευσης (training set) τα οποία αποτελούνται από 2,898 ουδέτερα σχόλια και 1,050 προσβλητικά σχόλια καθώς και δεδομένα δοκιμής (test set) αποτελούμενο από 1954 ουδέτερα σχόλια και 694 προσβλητικά. Στα παρακάτω παραδείγματα μπορείτε να δείτε τους δύο τύπους σχολίων:

Insult:”Oh, you are such an idiot.....you just confirmed thatyou can’t read ... dumb@rse!”

Neutral:”You get the gold star!The best post I’ve seen on here in months!!Hilarious...Great job,Canadian!PS I think we need you to come down here.I’ll sponsor you!!

Προεπεξεργασία και καθάρισμα των δεδομένων

Σαν πρώτο βήμα πρέπει να μετατρέψετε όλους τους χαρακτήρες σε μικρούς και αφαιρούμε τα links αν υπάρχουν. *(1 μονάδα)*

Μέθοδος

Αρχικά θα χρησιμοποιήσετε για το classification τον κλασικό NaiveBayes αλγόριθμο (σε αυτό το βήμα θα μετατρέψετε τα σχόλια σε word vectors χρησιμοποιώντας τον CountVectorizer της βιβλιοθήκης sklearn sklearn.feature_extraction.text.CountVectorizer — scikit-learn 0.23.1 documentation) . *(1 μονάδα)*

Θα βελτιστοποιήσετε τον Naive bayes: (1) κάνοντας lemmatization, (2) αφαιρώντας τα stop words, (3) χρησιμοποιώντας bigrams (φράσεις δύο λέξεων) αντί για μοναδικές λέξεις, (4) χρησιμοποιώντας Laplace Smoothing . *(2 μονάδες)*

Στη συνέχεια θα δημιουργήσετε ένα πιο σύνθετο πίνακα χαρακτηριστικών που θα περιέχει part-of-speech χαρακτηριστικά και TF-IDF-based χαρακτηριστικά. *(2 μονάδες)*

Θα δοκιμάσετε τα σύνολο των χαρακτηριστικων που έχετε εξάγει σε επιπλέον δύο διαφορετικά μοντέλα: ένα SVM και ένα random decision forest. *(2 μονάδες)*

Εκτός από τα παραπάνω μπορείτε να πειραματιστείτε με ότι μέθοδο/feature θέλετε με σκοπό να πετύχετε όσο το δυνατόν καλύτερα αποτελέσματα στο test set. Το καλύτερο δυνατό μοντέλο ως τώρα έχει F1-score: 0.952. *(1 μονάδα)*

Σύνοψη των χαρακτηριστικών που πρέπει να εξάγετε

Part-of-Speech Based Features: Χρησιμοποιώντας τον part-of-speech tagger (επισημειωτής ονοματικών οντοτήτων) που παρέχει το NTLK, επισημειώστε (tag) κάθε λέξη σε κάθε σχόλιο με το μέρος του λόγου της (noun, verb, adverb ή adjective). Στη συνέχεια υπολογίστε το ποσοστό των συγκεκριμένων tags για κάθε κείμενο. Δηλαδή το frequency του εκάστοτε POS tag σε κάθε δείγμα, υπολογιζόμενο ως προς το συνολικό αριθμό των λεξεων του δειγματος. Με αυτό τον τρόπο θα έχετε 4 νέα χαρακτηριστικά σε κάθε κείμενο, fractionAdverbs, fractionVerbs, fractionAdjectives, fractionNouns. Η ίδια διαδικασία πρέπει να γίνει και στο test set.

TF-IDF Based Features: όπως έχετε κάνει και στις προηγούμενες εργασίες

Παραδοτέο

Θα παραδώσετε ένα αρχείο τύπου **python notebook** στο οποίο θα έχετε τον κώδικα σας και τα αποτελέσματα/συμπεράσματα *(1 μονάδα)* των αλγορίθμων ταξινόμησης. Μετρικές απόδοσης (i) classification accuracy, (ii) F1 score.

Τέλος να σημειώσουμε πως ο κώδικας πρέπει να περιέχει σχόλια που να περιγράφουν τα βήματα σας αναλυτικά, μπορείτε να χρησιμοποιήσετε τις υλοποιήσεις από το sklearn και εφόσον χρησιμοποιήσετε οποιαδήποτε άλλο κώδικα που έχετε βρει στο διαδίκτυο πρέπει να υπάρχει αναφορά της ιστοσελίδας από όπου προήλθε ο κώδικας.