ΜΗΧΑΝΙΚΗ ΜΑΘΗΣΗ

ΓΡΑΠΤΗ ΕΡΓΑΣΙΑ 8

Final Project

Σκοπός της συγκεκριμένης εργασίας είναι η πρακτική εφαρμογή των γνώσεων που αποκτήσαμε στη διάρκεια του εξαμήνου, πάνω σε ένα πραγματικό πρόβλημα. Συγκεκριμένα ζητείται η κατηγοριοποίηση κειμένων σε συγκεκριμένες κατηγορίες με βάση το περιεχόμενό τους. Τα κείμενα αφορούν επιλεγμένα tweets χρηστών τα οποία κατηγοριοποιήθηκαν ότι περιέχουν ρητορική μίσους, προσβλητικό λόγο ή τίποτε από τα δύο.

Πρόκειται για μία κλασική Supervised Learning περίπτωση. Η εκφώνηση ζητά την υλοποίηση ενός αλγορίθμου, αλλά εγώ είχα την ευχέρεια χρόνου να υλοποιήσω κσι να συγκρίνω την απόδοση 3 συνολικά αλγορίθμων και συγκεκριμένα των: Multinomial Naive Bayes, Logistic Regression και Support Vector Machines, μέσω των αντίστοιχων μεθόδων που παρέχει η βιβλιοθήκη sci-kit learn της Python. Η αξιολόγηση και η σύγκριση της απόδοσης των αλγορίθμων έγινε, σύμφωνα με τις οδηγίες της εκφώνησης, με χρήση των ακόλουθων μετρικών (υπολογίζονται αυτόματα μέσω του πακέτου Classification Report της sci-kit learn):

- **precision**: Ορίζεται ως TP / (TP+FP)
- **recall**: Ορίζεται ως TP / (TP+FN)
- **f1**: Ορίζεται ως 2*precision*recall / (precision+recall)

Α. Προεπεξεργασία των δεδομένων (κειμένων)

Λόγω της φύσης της προέλευσης των κειμένων κατέστη αναγκαία η προεπεξεργασία τους και ο καθαρισμός τους (αυτή είναι η προβλεπόμενη διαδικασία, αλλά μου έκανε έντονη εντύπωση ότι όταν χρησιμοποίησα ακατέργαστα τα κείμενα πήρα τα ίδια ή και ελαφρώς καλύτερα αποτελέσματα και με τους 3 αλγορίθμους!). Για το σκοπό αυτό χρησιμοποιήθηκαν μια σειρά τεχνικών βασισμένων σε Κανονικές Εκφράσεις (Regular Expressions) καθώς και σε έτοιμα εργαλεία preprocessing που παρέχει η βιβλιοθήκη sci-kit learn. Λεπτομέρειες για τις τεχνικές και τον κώδικα python που χρησιμοποιήθηκε, εμφανίζονται στο συνοδευτικό αρχείο *Iliapikos_ge8_teliki.py*. Περιελήφθησαν οι ακόλουθες κατεργασίες:

- Tokenization των κειμένων σε λέξεις (tokens)
- Μετατροπή χαρακτήρων σε πεζούς
- Διαγραφή όλων των tokens που ξεκινάνε με @
- Διαγραφή όλων των χαρακτήρων που δεν είναι γράμμα ή αριθμός
- Διαγραφή όλων των tokens που δεν αποτελούνται αποκλειστικά από γράμματα
- Διαγραφή όλων των tokens με μήκος μικρότερο του 3
- Ανασύσταση των tweets ως strings από τα tokens

Δοκιμάστηκε και η δυνατότητα stemming ή lemmatization των tokens (μέσω των μεθόδων που προσφέρει η βιβλιοθήκη NLTK), αλλά η βελτίωση στην απόδοση ήταν απειροελάχιστη σε σχέση με το επιπλέον υπολογιστικό κόστος, οπότε δεν υιοθετήθηκε.

Ως αποτέλεσμα της προεπεξεργασίας, ένα tweet που περιείχε αρχικά το κείμενο:

> "!!!!!!!!!!!!! RT @ShenikaRoberts: The shit you hear about me might be true or it might be faker than the bitch who told it to ya "

να μετατραπεί τελικά στο:

> "the shit you hear about might true might faker than the bitch who told"

Β. Υλοποίηση των Αλγορίθμων

Απαραίτητη προϋπόθεση για την υλοποίηση των αλγορίθμων ήταν η μετατροπή των κειμένων των tweets σε διανύσματα. Χρησιμοποιήθηκε η τεχνική TF-IDF, μέσω την υλοποίησης που παρέχει η βιβλιοθήκη sci-kit learn. Με τον τρόπο αυτό, παράλληλα με τη διανυσματοποίηση, υποβαθμίστηκαν σε αξία και οι πλέον κοινές χωρίς υψηλή διακριτικότητα λέξεις.

Η έξοδος του vectorizer οδηγείται αυτόματα στην είσοδο του εκάστοτε classifier, μέσω της χρήσης ενός Pipeline. Παράλληλα πραγματοποιείται εκτενής διερεύνηση για τη βέλτιστη παραμετροποίηση τόσο του vectorizer όσο και του classifier, μέσω της χρήσης της μεθόδου GridSearch, εκμεταλλευόμενος παράλληλα τη δυνατότητα CrossValidation που παρέχει. Η βέλτιστη παραμετροποίηση του κάθε αλγορίθμου που χρησιμοποιήθηκε φαίνεται στα ScreenShots της επόμενης ενότητας καθώς και στον αντίστοιχο κώδικα στο συνοδευτικό αρχείο *Iliapikos_ge8_teliki.py*.

Γ. Σύγκριση απόδοσης Αλγορίθμων

Στα ακόλουθα ScreenShots φαίνεται η έξοδος που παράγει ο κώδικας για κάθε αλγόριθμο:

```
********** Multinomial Naive Bayes Algorithm **********

Classification Report:
              precision    recall  f1-score   support

           0       0.64      0.12      0.20       357
           1       0.86      0.98      0.92      4798
           2       0.85      0.57      0.68      1041

   micro avg       0.86      0.86      0.86      6196
   macro avg       0.79      0.56      0.60      6196
weighted avg       0.85      0.86      0.84      6196


Training Accuracy: 0.916 (+/- 0.00)
Validation Accuracy: 0.858 (+/- 0.01)

Βέλτιστες παράμετροι:  {'TfIdf__max_df': 0.35, 'TfIdf__max_features': 10000, 'TfIdf__min_df':
5, 'TfIdf__ngram_range': (1, 3), 'TfIdf__norm': 'l2', 'TfIdf__use_idf': True, 'clf__alpha':
0.01, 'clf__fit_prior': True}
```

```
********** Logistic Regression Algorithm **********

Classification Report:
              precision    recall  f1-score   support

           0       0.50      0.26      0.34       357
           1       0.91      0.95      0.93      4798
           2       0.82      0.80      0.81      1041

   micro avg       0.89      0.89      0.89      6196
   macro avg       0.75      0.67      0.70      6196
weighted avg       0.88      0.89      0.88      6196


Training Accuracy: 0.973 (+/- 0.00)
Validation Accuracy: 0.882 (+/- 0.01)

Βέλτιστες παράμετροι:  {'TfIdf__max_df': 0.45, 'TfIdf__max_features': 10000, 'TfIdf__min_df':
5, 'TfIdf__ngram_range': (1, 2), 'TfIdf__norm': 'l2', 'TfIdf__use_idf': True, 'clf__C': 10,
'clf__multi_class': 'multinomial', 'clf__solver': 'saga'}
```

```
********** Support Vector Machines Algorithm **********

Classification Report:
              precision    recall  f1-score   support

           0       0.58      0.17      0.26       357
           1       0.92      0.96      0.94      4798
           2       0.82      0.84      0.83      1041

   micro avg       0.89      0.89      0.89      6196
   macro avg       0.77      0.66      0.67      6196
weighted avg       0.88      0.89      0.88      6196


Training Accuracy: 0.932 (+/- 0.00)
Validation Accuracy: 0.894 (+/- 0.01)

Βέλτιστες παράμετροι:  {'TfIdf__max_df': 0.35, 'TfIdf__max_features': 5000, 'TfIdf__min_df': 3,
'TfIdf__ngram_range': (1, 2), 'TfIdf__norm': 'l2', 'TfIdf__use_idf': True, 'clf__C': 1,
'clf__degree': 1, 'clf__kernel': 'linear'}
```

Παρατηρείται σταδιακή και αναμενόμενη αύξηση στην απόδοση των αλγορίθμων, ανάλογα με τη φύση τους. Πρόβλημα παρατηρείται γενικά με την κατηγοριοποίηση των παραδειγμάτων της κατηγορίας 0, τα οποία είναι πολύ λιγότερα σε σχέση με τα παραδείγματα των υπόλοιπων κατηγοριών.

Η γενική απόδοση των αλγορίθμων κρίνεται ικανοποιητική, αν και στη βιβλιογραφία έχουν αναφερθεί ανάλογες περιπτώσεις που η απόδοση (στις συγκεκριμένες μετρικές) ξεπερνάει το 95%. Αυτό πιθανότατα έχει να κάνει με το συγκεκριμένο dataset που χρησιμοποιήθηκε, που παρουσιάζει μεγάλη ανομοιογένεια στις κατανομές των κατηγοριών.

ΒΙΒΛΙΟΓΡΑΦΙΑ

1. Υλικό μαθήματος.

2. Müller, A.C & Guido, S. - Introduction to Machine Learning with Python. A Guide for Data Scientists, O' Reilly, 2017.

3. Τεκμηρίωση από τον ιστότοπο της βιβλιοθήκης Sklearn.