Τεχνητή Νοημοσύνη
Εργασία 2η - Machine Learning
Naive Bayes - Multinomial/Bernoulli

Γεώργιος Χείρμπος 3130230

Χειμερινό εξάμηνο 2018-2019

1 Εισαγωγή

Στα πλαίσια την 2η εργασίας επιλέχθηκε η υλοποίηση του αλγορίθμου Naive Bayes με δύο διαφορετικές υλοποιήσεις, Multinomial και Bernoulli. Οι υλοποίησεις βασίστηκαν στον ψευδοκώδικα που αναφέρονται στην βιβλιογραφία [1], [2]. Επιπλέον υλοποιήθηκαν οι αντίστιχοι classfiers από το πακέτο scikit-learn [3] για σύγκριση αποτελεσμάτων. Οι προβλέψεις έγιναν με χρήση των δεδομένων Ling-Spam, PU. (Το πακέτο Enron παρήγαγε σφάλμα "illegal multibyte sequence", και λόγω έλλειψης χρόνου παραλείφθηκε.)

2 Multinomial Naive Bayes

Η ιδέα στον Multinomial Naive Bayes είναι η εξής. Αρχικά κατασκευάζουμε το λεξικό που περιέχει όλες τις λέξεις που βρίσκονται στα κείμενα που μας δίνονται. Έπειτα για τα κείμενα που γνωρίζουμε σε ποια κλάση ανοίκουν, να υπολογίσουμε τις apriori πιθανότητες για κάθε κλάση βάση τις εμφανίσεις κειμένων κάθε κλάση προς τον αριθμό των κειμένων.
Μετά αναλύουμε κάθε κείμενο για κατασκευάζουμε ένα λεξικό που περιέχει όλες τις λέξεις του λεξικού και των αριθμό που εμφανίζονται αυτές ανα κείμενο. Επιπλέον έχουμε τόσα λεξικά όσες και οι κλάσεις που περιέχουν πάλι τις λέξεις του λεξιλογίου και τον αριθμό των εμφανίσεων των λέξεων αυτών βάση των κειμένων που ανήκουν στις κλάσεις αυτές.
Η πιθανότητα για κάθε λέξη προκύπτει ξεχωριστά για κάθε κλάση και είναι η:

$$\frac{Term_{ct} + 1}{\sum(Term_{ct} + 1)}$$

, στον αριθμητή, Term είναι ο αριθμός εμφανίσεων συγκεκριμένου όρου για συγκεκριμένη τάξη, ενώ στον παρονομαστή είναι το άθροισμα των αριθμών εμφανίσεων όλων των όρων της τάξης. Ο άσσος προστίθεται για να αποφύγουμε την περίπτωση του 0 (στον υπολογισμο λογαρίθμου).
Για την πρόβλεψη, αναλύουμε κάθε αρχείο ξεχωριστά, μετράμε τις εμφανίσεις των όρων σε αυτό και αθροίζουμε τόσες πιθανότητες όσες και ο αριθμός των τάξεων. Το μεγαλύτερο άθροισμα από αυτές δείχνει και σε ποια κλάση πιστεύουμε ότι ανήκει το αρχείο.

3 Bernoulli Naive Bayes

Η εναλλακτική μέθοδος αυτή, δεν υπολογίζει αριθμό εμφανίσεων λέξης μέσα στα κείμενα. Αρκεί να υπάρχει το πολύ μια φορά σε αυτό. Όπως και πριν, μέχρι το σημείο του υπολογισμού apriori πιθανοτήτων είναι το ίδιο. Μετά αναλύουμε το λεξικό λέξη προς λέξη και υπολογίζουμε σε πόσα κείμενα εμφανίζεται ανα κλάση. Για κάθε κλάση ο συγκεκριμένος όρος έχει πιθανότητα

$$\frac{N_{ct} + 1}{N_c + 2}$$

, όπου στον αριθμητή έχουμε τον αριθμό κειμένων της συγκεκριμένης κλάσης που εμφανίζεται ο όρος συν 1 προς το σύνολο κειμένων της τάξης συν 2. Ο άσσος είναι πάλι για να αποφύγουμε τα μηδενικά, ενώ το 2 συμβολίζει τις δυο καταστάσεις που έχει ένας όρος, δηλαδή υπάρχει ή όχι.

Για την πρόβλεψη πάλι η ίδια ανάλυση, μόνο που τώρα αναλύουμε κατα όρο και υπολογίζουμε την λογαριθμική πιθανότητα του ανα κλάση στην περίπτωση που υπάρχει ο όρος στο κείμενο, ενώ αν δεν υπάρχει ο όρος υπολογίζουμε το 1-την λογαριθμική πιθανότητα ανα κλάση. Πάλι το μεγαλύτερο άθροισμα δείχνει σε ποια ταξη πιστεύουμε ότι ανήκει το αρχείο.

4 Εκτέλεση

Για την σωστή εκτέλεση του προγράμματος πρέπει τα δεδομένα να βρίσκονται σε έναν φάκελο με τον όνομα datasets στον ίδιο directory που είναι και τα εκτελέσιμα αρχείο. Επιπλέον σε πρώτο επίπεδο να υπάρχουν οι φάκελοι lingspam_public, pu_corpora_public και σε δεύτερο επίπεδο οι υποφάκελοι για κάθε dataset (πχ. lemm, stop, pu1, pu2).

5 Αποτελέσματα

Τα αποτελέσματα περιγράφονται από τις εξής 4 μετρικές accuracy, precision, recall, F1.
True positive (TP) = Σωστή θετική πρόβλεψη.
True negative (TN) = Σωστή αρνητική πρόβλεψη
False positive (FP) = Λανθάνων θετική πρόβλεψη
False negative (FN) = Λανθάνων αρνητική πρόβλεψη

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN}, Precision = \frac{TP}{TP+FP}, Recall = \frac{TP}{TP+FN}, F1 = 2 * \frac{Precision*Recall}{Precision+Recall}$$

Dataset : lingspam_public: bare

Algorithm	Split	Accuracy	Precision	Recall	F1
3130230 MNB	0.4	0.98839	0.96234	0.97457	0.96842
3130230 BNB	0.4	0.87703	0.89690	0.36864	0.52252
SKL MNB	0.4	0.98607	0.95798	0.96610	0.96202
SKL BNB	0.4	0.87703	0.89690	0.36864	0.52252

Dataset : lingspam_public: lemm

Algorithm	Split	Accuracy	Precision	Recall	F1
3130230 MNB	0.4	0.98878	0.94936	0.98684	0.96774
3130230 BNB	0.4	0.88863	0.90721	0.38596	0.54153
SKL MNB	0.4	0.98579	0.93723	0.98245	0.95931
SKL BNB	0.4	0.88863	0.90721	0.38596	0.54153

Dataset : lingspam_public: lemm_stop

Algorithm	Split	Accuracy	Precision	Recall	F1
3130230 MNB	0.4	0.99385	0.96682	0.99512	0.98076
3130230 BNB	0.4	0.90930	0.95789	0.44390	0.60666
SKL MNB	0.4	0.98539	0.91891	0.99512	0.95550
SKL BNB	0.4	0.90930	0.95789	0.44390	0.60666

Dataset : lingspam_public: stop

Algorithm	Split	Accuracy	Precision	Recall	F1
3130230 MNB	0.4	0.99242	0.96774	0.98591	0.97674
3130230 BNB	0.4	0.90303	0.92079	0.43661	0.59235
SKL MNB	0.4	0.98257	0.90948	0.99061	0.94831
SKL BNB	0.4	0.90303	0.92079	0.34883	0.59235

Dataset : pu_corpora_public: pu1

Algorithm	Split	Accuracy	Precision	Recall	F1
3130230 MNB	0.4	0.96210	0.92187	0.98333	0.95161
3130230 BNB	0.4	0.95258	0.98765	0.88888	0.93567
SKL MNB	0.4	0.93965	0.88000	0.97777	0.92631
SKL BNB	0.4	0.95258	0.98765	0.88888	0.93567

Dataset : pu_corpora_public: pu2

Algorithm	Split	Accuracy	Precision	Recall	F1
3130230 MNB	0.4	0.90390	0.65591	1.0	0.79220
3130230 BNB	0.4	0.85285	0.8000	0.26229	0.39506
SKL MNB	0.4	0.94594	0.79452	0.95081	0.86567
SKL BNB	0.4	0.85285	0.8000	0.26299	0.39506

Dataset : pu_corpora_public: pu3

Algorithm	Split	Accuracy	Precision	Recall	F1
3130230 MNB	0.4	0.95553	0.92488	0.97524	0.94939
3130230 BNB	0.4	0.90841	0.97888	0.80321	0.88239
SKL MNB	0.4	0.95023	0.91608	0.9277	0.94357
SKL BNB	0.4	0.90841	0.97888	0.80321	0.88239

Dataset : pu_corpora_public: pua

Algorithm	Split	Accuracy	Precision	Recall	F1
3130230 MNB	0.4	0.95694	0.95815	0.95020	0.95416
3130230 BNB	0.4	0.94520	0.90804	0.98340	0.94422
SKL MNB	0.4	0.95696	0.96997	0.93775	0.95358
SKL BNB	0.4	0.94520	0.90804	0.98340	0.94422

6 Files

File	Description
naive.py	Main executable, initializes classifiers
common.py	Init dataset, Metrics method, scikit helper
multinomialNB.py	Multinomial train, predict
bernoulliNB.py	Bernoulli train, predict
scikitNB.py	Scikit Multinomial/Bernoulli train, predict

Αναφορές

[1] Multinomial Naive Bayes https://nlp.stanford.edu/IR-book/html/htmledition/naive-bayes-text-classification-1.html.

[2] Bernoulli Naive Bayes https://nlp.stanford.edu/IR-book/html/htmledition/the-bernoulli-model-1.html.

[3] Scikit Learn. https://scikit-learn.org/stable/index.html.