Εργασία στην Προηγμένη Μηχανική Μάθηση και Ανακάλυψη Γνώσης

Γραβάνης Γεώργιος
ΑΕΜ: 339

29 Απριλίου 2018

Περιεχόμενα

Κατάλογος σχημάτων

Κατάλογος πινάκων

Πίνακας 1: Περιγραφή των Συνόλων Δεδομένων

Dataset Name	Abbr.	Repos	Data type	Classes	Instances	Balanced
Deceptive opinion spam corpus	1	kaggle	text	truthfull / deceptive	1600	yes
Ironic corpus	2	kaggle	text	yes / no	1950	yes
Sentiment Labelled Sentences - Amazon	3	UCI	text	positive / negative	1000	no
Sentiment Labelled Sentences - IMDB	4	UCI	text	positive / negative	1000	yes
Sentiment Labelled Sentences - Yelp	5	UCI	text	positive / negative	1000	yes
SMS Spam collection	6	UCI	text	spam / ham	5090	no
Twitter airline sentiment	7	Kaggle	text	positive / negative	11540	no
Youtube Spam Collection - KatyPerry	8	UCI	text	spam / ham	350	yes
Youtube Spam Collection - LMFAO	9	UCI	text	spam / ham	438	yes
Youtube Spam Collection - Shakira	10	UCI	text	spam / ham	370	yes

1 Μέρος Α

1.1 Περιγραφή

Στο πρώτο μέρος της εργασίας ζητήθηκε η μελέτη των αποτελεσμάτων τεσσάρων ensemble αλγορίθμων για δέκα σύνολα δεδομένων, με την εφαρμογή μεθόδων σύγκρισης που εξασφαλίζουν την ορθότητα των αποτελεσμάτων και την στατιστική σημασία τους.

Συγκεκριμένα, για την υλοποίηση των παραπάνω επιλέχθηκαν τα εξής:

- **Σύνολα Δεδομένων**
  Για την εκπόνηση της συγκεκριμένης εργασίας επιλέχθηκαν σύνολα δεδομένων κειμένου με στόχο την δυαδική ταξινόμηση τους και παρεμφερές περιεχόμενο της μορφής spam/ham, deceptive/truthfull κλπ. Τα γνωρισμάτα που χρησιμοποιήθηκαν ήταν βασικές μετρικές όπως ο αριθμός των λέξεων, ο αριθμός των συλλαβών, δείκτες αναγνωσιμότητας (π.χ. flesch - kincaid, smog index), καθώς επίσης και απαριθμητές λέξεων που ανήκουν σε συγκεκριμένες κατηγορίες π.χ. [senses, ...]

- **Pipeline - Αλγόριθμοι** Για την εκπόνηση των πειραμάτων ακολουθήθηκε η εξής διαδικασία: Το κάθε σύνολο δεδομένων χωρίστηκε αρχικά σε train και test set (70% - 30% αντίστοιχα). Στη συνέχεια, αφού τα αρχικά σύνολα κανονικοποιήθηκαν, για κάθε αλ-

γόριθμο εκτελέστηκε ένα εκτεταμένο gridsearch με 10-fold cross validation στο train set. Το μοντέλο που προέκυψε από την παραπάνω διαδικασία αξιολογήθηκε στο test set του αρχικού συνόλου δεδομένων. Δεδομένου ότι στα σύνολα που χρησιμοποιήσαμε οι δύο κλάσεις είναι κατα κύριο λόγο ισοκατανεμημένες και καθώς δεν υπάρχει κάποιο κόστος λανθασμένης αξιολογήσης, η μετρική που χρησιμοποιήσαμε ήταν η ακρίβεια (accuracy).
Οι αλγόριθμοι που επιλέχθηκαν προς σύγκριση ήταν οι εξής: adaBoost[1], Bagging[1], GradientBoost & RandomForest.

1.2 Αποτελέσματα

Σε αυτή την ενότητα θα παρατεθούν τα αποτελέσματα της πειραματικής διαδικασίας και θα γίνει ο σχολιασμός τους.

1.2.1 Boxplots

Για την παρακολούθηση της επιλογής παραμέτρων για τον κάθε αλγόριθμο και για κάθε σύνολο δεδομένων, αποφασίσθηκε η εκτύπωση των αποτελεσμάτων του nested cross validation με την μορφή Boxplots. Επιπλέον, το συγκεκριμένο γράφημα συνοδεύται από την εκτύπωση σε barchart των αποτελεσμάτων του βέλτιστου μοντέλου στο test set που αποκλείστηκε αρχικά από την διαδικασία μάθησης. Τα αποτελέσματα φαίνονται στον πίνακα 2.

1.2.2 Καμπύλες μάθησης

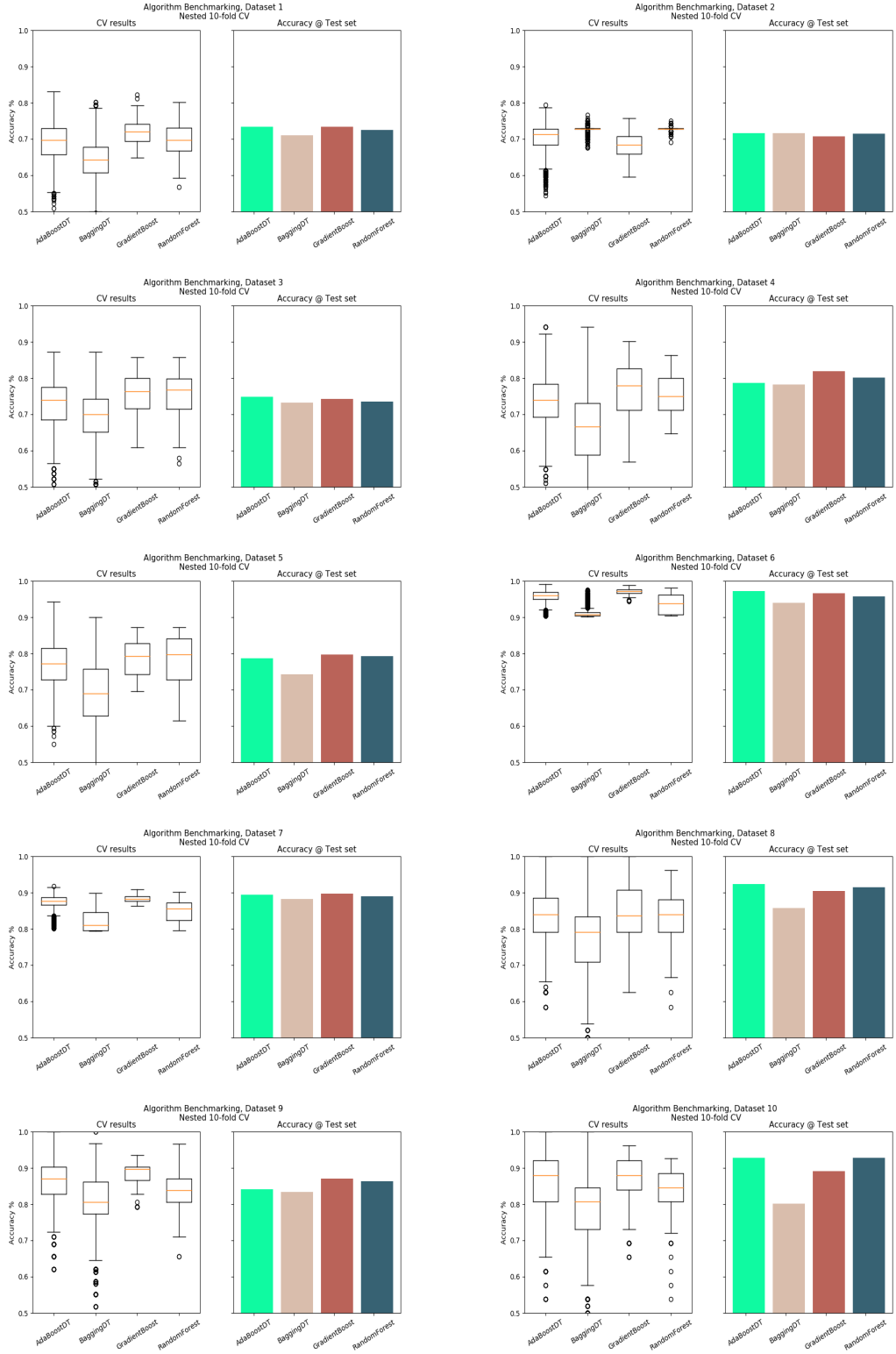
Για την καλύτερη εποπτεία και αξιολόγηση των αποτελεσμάτων αποφασίσθηκε η εκτύπωση των καμπυλών μάθησης. Όπως φαίνεται και στο Σχήμα 1, μετά το εκτεταμένο grid search είναι εμφανής η τάση για σφαλματώδη υπεραρμογή (overfitting) στις περισσότερες των περιπτώσεων. Αξίζει να σημειώσουμε αυτήν την παρατήρηση αλλά δεν θα ασχοληθούμε με την επίλυση του συγκεκριμένου προβλήματος στην παρούσα εργασία καθώς δεν είναι εντός των πλαισίων της.

1.2.3 Συγκριτική αξιολόγηση

Για την αξιολόγηση των αλγορίθμων αρχικά επιλέχθηκε η μετρική της ακρίβειας (accuracy). Με το πέρας των πειραμάτων και για την τελική αξιολόγηση της κατάταξης των αλγορίθμων, επιλέχθηκε η μέθοδος Friedman για την εύρεση ή μη στατιστικά σημαντικών διαφορών. Όπως αναφέρει ο [**demvsar**] για για την εκτέλεση του εξελιγμένου Friedman test

[1]with Decision Tree as an estimator

Πίνακας 2: Αποτελέσματα από την εκπαίδευση των αλγορίθμων και από την αξιολόγησή τους στο test set του κάθε συνόλου δεδομένων.



Σχήμα 1: Κάθε γράφημα απεικονίζει την καμπύλη μάθησης για κάθε σύνολο δεδομένων και για κάθε αλγόριθμο. Στο άξονα των y έχουμε το σφάλμα % ενώ στον άξονα των x έχουμε το ποσοστό του συνόλου δεδομένων που έχει χρησιμποιηθεί για την εκπαίδευση και την αξιολόγηση του κάθε μοντέλου. Το βήμα που έχει χρησιμοποιηθεί για την κατανομή σε μέρη είναι το 3% του κάθε συνόλου.

όπως το πρότειναν οι [**\color {red}Iman and Davenport**],αρχικά θα πρέπει να δημιουργηθεί ο Πίνακας κατάταξης (βλ. Πίνακα 3) και στη συνέχεια να υπολογιστούν οι εξής τιμές:

$$x_F^2 = \frac{12N}{k(k+1)}\Big[\sum_j R_j^2 - \frac{k(k+1)^2}{4}\Big] \tag{1}$$

$$F_F = \frac{(N-1)x_F^2}{N(k-1)-x_F^2} \tag{2}$$

Σε περίπτωση που το αποτέλεσμα της εξίσωσης 2 είναι μεγαλύτερο από την τιμή της κατανομής όπως ορίζεται από τον πίνακα, τότε θα πρέπει να συνεχίσουμε με το post hoc test για το οποίο θα πρέπει να υπολογίσουμε την τιμή ***CD***.

$$CD = q_a\sqrt{\frac{k(k+1)}{6N}} \tag{3}$$

Πράγματι, αν εφαρμόσουμε τις τιμές του Πίνακα 3 στις εξισώσεις 1 & 2 για το συγκεκριμένο πρόβλημα της εργασίας θα έχουμε τα παρακάτω αποτελέσματα

$$x_F^2 = \frac{12\cdot 10}{4\cdot(4+1)}\Big[(1.85^2+3.75^2+1.95^2+2.55^2) - \frac{4\cdot(4+1)^2}{4}\Big] = 16.74 \tag{4}$$

$$F_F = \frac{9\cdot 16.74}{10\dot{3}-16.74} = 11.36 \tag{5}$$

Επειδή η τιμή που προκύπτει από την 5 είναι μεγαλύτερο από το χχχ, απορίπτουμε την παραδοχή ότι δεν υπάρχει στατιστικά σημαντική διαφορά στην κατάταξη των αλγορίθμων και συνεχίζουμε στον υπολογισμό της τιμής CD όπως φαίνεται στην εξίσωση 6.

$$CD = 2.569\sqrt{\frac{4\cdot 5}{6\cdot 10}} = 1.4832 \tag{6}$$

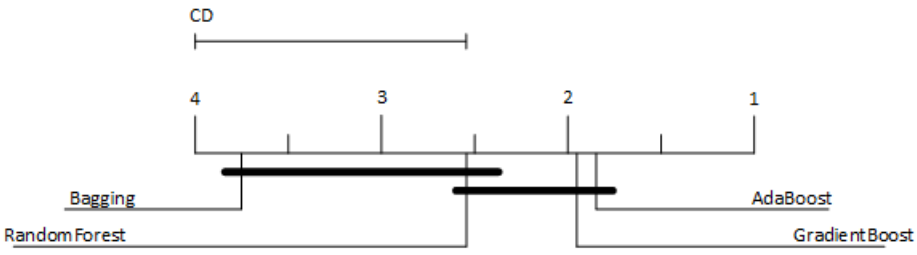
Όπως φαίνεται και στο Σχήμα 2 από τα αποτελέσματα της post hoc διαδικασίας βλέπουμε ότι η διαφορά μεταξύ του τελευταίου στην κατάταξη αλγορίθμου Bagging και των δύο πρώτων (AdaBoost και GradientBoost) είναι στατιστικά σημαντική.

2 Μέρος Β

Στο δεύτερο μέρος της εργασίας καλούμαστε να μελετήσουμε ένα πρόβλημα ταξινόμησης δοθέντος ενός πίνακα κόστους προκειμένου να επιτευχθεί βελτιστοποίηση βάση του κόστους μίας λανθασμένης ταξινόμησης και όχι κάποιας άλλης μετρικής. Θα πρέπει να σημειώσουμε εδώ ότι ενώ για το πρώτο και τρίτο μέρος της εργασίας η υλοποίηση έγινε σε Python 2.7 με την χρήση της βιβλιοθήκης scikit - learn, δυστυχώς δεν υπάρχει κάποια ώριμη υλοποίηση για την επίλυση προβλημάτων ταξινόμησης με κριτήριο το κόστος. Γι αυτόν τον λόγο, για

Πίνακας 3: Αποτελέσματα και κατάταξη αλγορίθμων

Dataset	AdaBoost	Bagginng	GradBoost	RandForest
1	0.733 (1.5)	0.710 (4.0)	0.733 (1.5)	0.725 (3.0)
2	0.716 (1.5)	0.716 (1.5)	0.708 (4.0)	0.715 (3.0)
3	0.749 (1.0)	0.732 (4.0)	0.742 (2.0)	0.736 (3.0)
4	0.787 (3.0)	0.783 (4.0)	0.819 (1.0)	0.801 (2.0)
5	0.787 (3.0)	0.743 (4.0)	0.797 (1.0)	0.793 (2.0)
6	0.972 (1.0)	0.940 (4.0)	0.966 (2.0)	0.958 (3.0)
7	0.893 (2.0)	0.883 (4.0)	0.897 (1.0)	0.889 (3.0)
8	0.924 (1.0)	0.857 (4.0)	0.905 (3.0)	0.914 (3.0)
9	0.841 (3.0)	0.833 (4.0)	0.871 (1.0)	0.864 (2.0)
10	0.928 (1.5)	0.802 (4.0)	0.892 (3.0)	0.928 (1.5)
avg rank	1.85	3.75	1.95	2.55



Σχήμα 2: Σύγκριση όλων των ταξινομητων μεταξύ τους με την χρήση του Nemenyi test. Οι ομάδες των ταξινομητών μεταξύ των οποίων δεν υπάρχει στατιστικά σημαντική διαφορά για p = 0.05 ενώνονται μεταξύ τους.

Πίνακας 4: My caption

Τεχνική	Naive Bayes		Linear SVM		Random Forest	
Plain	125	25	120	30	119	31
	31	89	31	89	50	70
	180	79.3%	185	77.4%	281	70%
MetaCost	110	40	105	45	10	140
	15	105	18	102	3	117
	115	79.6%	135	76.7%	155	47%
CostSensitive	114	36	95	55	14	136
	19	101	16	104	2	118
	131	79.6%	135	74%	146	48.9%

την υλοποίηση του Β Μέρους επιλέχθηκε το Weka. Ως συνέπεια αυτού, η απεικόνιση των αποτελεσμάτων θα είναι πιο φτωχή σε σχέση με τα υπόλοιπα μέρη της εργασίας.

2.1 Γενικά

Όπως περιγράφεται στην εκφώνηση της εργασίας, στο Β Μέρος ζητείται η μελέτη τριών αλγορίθμων ταξινόμησης (Naive Bayes, SVM με γραμμικό πυρήνα και Random Forest) σε συνδυασμό με τεχνικές που βελτιώνουν την μάθηση σε σχέση με έναν δωθέντα πίνακα κόστους. Η σύγκριση που θα γίνει περιλαμβάνει αρχικά την ταξινόμηση χωρίς εφαρμογή αυτών των τεχνικών για την αποτύπωση μίας αρχικής κατάστασης και στη συνέχεια την εφαρμογή του MetaCost και του CostSensitive.

2.2 Αποτέλεσματα

Παρατηρώντας τον πίνακα αποτελεσμάτων (Πίνακας 4) μπορούμε εύκολα να συμπεράνουμε ότι για το συγκεκριμένο πρόβλημα, η επιλογή της μεθόδου MetaCost σε συνδυασμό με τον αλγόριθμο ταξινόμησης Naive Bayes οδηγεί στην ταξινόμηση των αποτελεσμάτων με το μικρότερο δυνατό κόστος. Στην προκειμένη περίπτωση αν και ουσιασικά η παρατήρηση είναι χωρίς αξία, θα πρέπει να αναφέρουμε ότι ο συγκεκριμένος συνδυασμός επιτυγχάνει και την μεγαλύτερη ακρίβεια ταξινόμησης (accuracy). Παρατηρούμε επίσης ότι οι αλγόριθμοι Naive Bayes και Linear SVM αντιμετωπίζουν το πρόβλημα με σεβασμό και στις δύο κλάσεις, ενώ ο Random Forest προκειμένου να αποφύγει την αύξηση του κόστους μειώνει στο ελάχιστο τα αποτελέσματα που κατηγοριοποιεί στην κλάση 0 χωρίς αυτό να αποτελεί καλή επιλογή.

3 Μέρος Γ

3.1 Προϋποθέσεις

Τα `*.tex` αρχεία θα πρέπει να χρησιμοποιούν κωδικοποίηση UTF-8 και ο text editor που χρησιμοποιείται να υποστηρίζει και να είναι ρυθμισμένος ώστε να χρησιμοποιεί αυτή τη κωδικοποίηση χαρακτήρων.

Το παράδειγμα αυτό έχει δοκιμαστεί μόνο σε σύστημα Linux, αλλά πέρα από τη χρήση του Makefile για τη δημιουργία του τελικού κειμένου σε μορφή PDF, δεν υπάρχει κάποιος λόγος ώστε να μην μπορεί να χρησιμοποιηθεί και σε άλλο λειτουργικό σύστημα. Το default font είναι το Linux Libertine. Σε συστήματα Linux είναι συνήθως προεγκατεστημένο, αλλιώς φροντίστε να το εγκαταστήσετε. Ή αλλάξτε το στο αρχείο ***main.tex*** τέλος πάντων.

3.2 Δημιουργία PDF κλπ

Τρέχοντας απλά την εντολή `make` δημιουργείται το PDF, με πλήρη υποστήριξη βιβλιογραφίας. Παράλληλα παρέχονται και οι επιλογές `make docx` και `make odt` οι οποίες μπορούν να δημιουργήσουν ένα αρχείο MS Word ή OpenDocument αντίστοιχα. Μην περιμένετε τα τελευταία να δείχνουν τέλεια όπως το PDF, αλλά αν είστε αναγκασμένοι να δημιουργήσετε κάτι τέτοιο, είναι μια λύση. Το τελευταίο απαιτεί εγκατεστημένο το ***pandoc***. Σε οποιαδήποτε περίπτωση δημιουργείται ένα αρχείο με όνομα ***output*** και την αντίστοιχη επέκταση.

4 Αναφορές

Το κομμάτι της βιβλιογραφίας έχει χωριστεί σε δύο μέρη, το πρώτο με τίτλο «Βιβλιογραφικές αναφορές» και το δεύτερο με τίτλο «Διαδικτυακές αναφορές». Στο δεύτερο μπαίνουν αυτόματα όσες αναφορές είναι του τύπου `@MISC`, ενώ στο πρώτο όλες οι υπόλοιπες. Στις «Διαδικτυακές αναφορές» υποστηρίζεται και η προσθήκη της ημερομηνίας προσπέλασης. Δείτε τα παραδείγματα στο αρχείο της βιβλιογραφίας ***main.bib***, στο οποίο περιλαμβάνονται δυο βιβλία [**goossens93**][**Syropoulos**] και μια ιστοσελίδα [**JABREF**] (Συμβουλή: χρησιμοποιήστε ένα εξειδικευμένο editor για τη βιβλιογραφία, όπως το JabRef [**JABREF**]). Ταυτόχρονη χρήση ελληνικών και λατινικών χαρακτήρων υποστηρίζεται φυσικά και στη βιβλιογραφία.