ΑΡΙΣΤΟΤΕΛΕΙΟ ΠΑΝΕΠΙΣΤΗΜΙΟ
ΘΕΣΣΑΛΟΝΙΚΗΣ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

Μεταπτυχιακό Πρόγραμμα Σπουδών

2η Εργασία στην Προηγμένη Μηχανική Μάθηση και Ανακάλυψη Γνώσης

Ονοματεπώνυμο: Γιώργος Γραβάνης

ΑΕΜ: 339

14 Ιουνίου 2018

Περιεχόμενα

1 Εισαγωγή

Στα πλαίσια της δεύτερης εργασίας του μαθήματος Advanced Machine Learning καλούμαστε να εφαρμόσουμε διάφορες τεχνικές σχετικές κυρίως με προβλήματα μάθησης από πολλαπλές ετικέτες (multi-label) και πολλαπλών παραδειγμάτων (multi-instance), καθώς επίσης και να εφαρμόσουμε την μέθοδο ενεργούς μάθησης (active learning) σε ένα δεδομένο πρόβλημα ταξινόμησης. Για την εφαρμογή όλων των παραπάνω, θα χρησιμοποιηθεί το σύνολο δεδομένων ***DeliciousMIL*** όπως έχει δοθεί σχετική οδηγία στην εκφώνηση της εργασίας.

1.1 Σύνολο Δεδομένων

Το ***DeliciousMIL*** είναι ένα σύνολο δεδομένων το οποίο δημιουργήθηκε από τους Soleimani and Miller, 2016 με στόχο να εξυπηρετήσει την έρευνα γύρω από προβλήματα ταξινόμησης πολλαπλών ετικετών και πολλαπλών παραδειγμάτων με κύριο αντικείμενο την ταξινόμηση κειμένου σε επίπεδο εγγράφων και προτάσεων.

Τα έγγραφα που περιέχονται στην εν λόγω συλλογή προέρχονται από το site delicious.com και συλλέχθηκαν την περίοδο του Ιουνίου 2008. Οι ετικέτες που αναγνωρίστηκαν είναι συνολικά 20 και περιγράφονται στον Πίνακα 1.

Πίνακας 1: Περιγραφή των Ετικετών

index	class	index	class	index	class	index	class
1	reference	6	web	11	style	16	politics
2	design	7	java	12	language	17	religion
3	programming	8	writing	13	books	18	science
4	internet	9	English	14	education	19	history
5	computer	10	grammar	15	philosophy	20	culture

Το μέγεθος του συνόλου των δεδομένων είναι 12234 έγγραφα τα οποία έχουν χωριστεί σε δύο υποσύνολα:

- Εκπαίδευσης με 8251 έγγραφα
- Επαλήθευσης με 3983 έγγραφα

Η συχνότητα εμφάνισης της κάθε κλάσης φαίνεται στην Εικόνα 1

Το DeliciousMIL είναι ελεύθερα διαθέσιμο στο αποθετήριο του UC Irvine Machine Learning Repository [1]

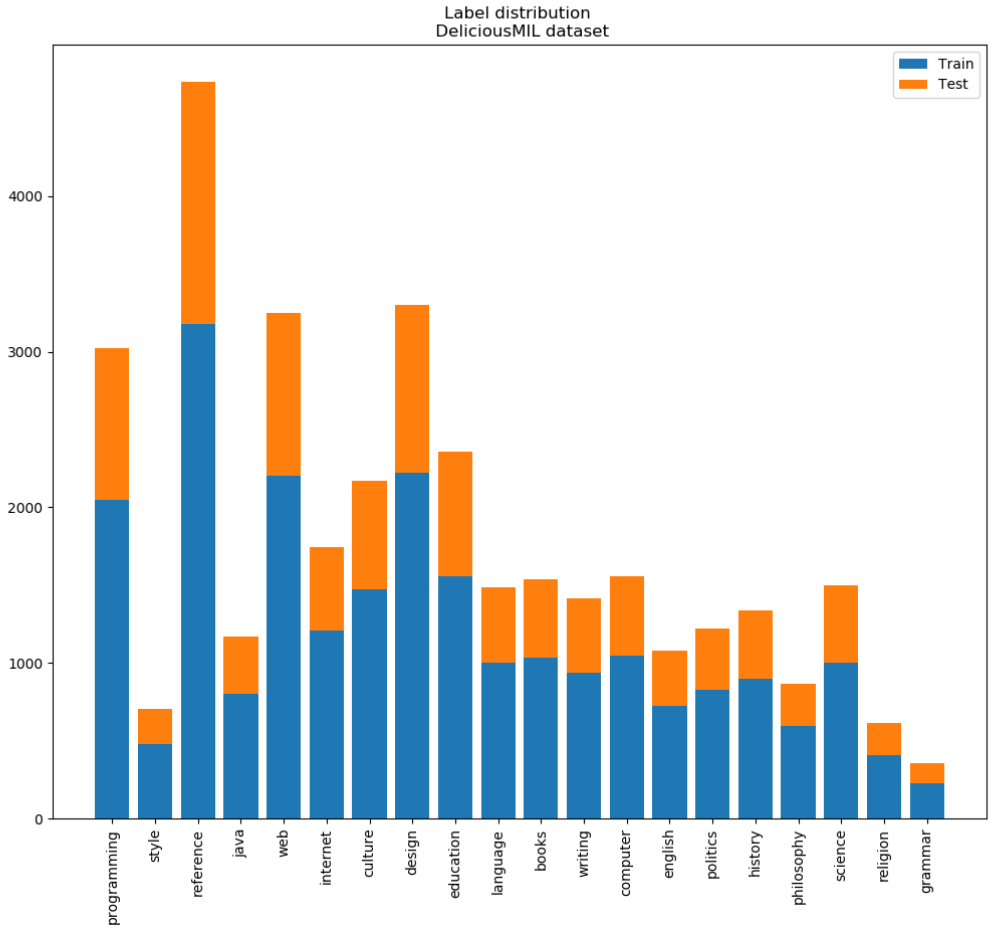
2 Μέρος Α

Για το πρώτο μέρος της εργασίας ζητήθηκε η εφαρμογή μία μεθόδου μάθησης από δεδομένα πολλαπλών ετικετών. Για την υλοποίηση του παραπάνω χρησιμοποιήθηκε η μέθοδος Classifier Chain και συγκεκριμένα εφαρμόσθηκε μία ensemble εκδοχή με vote averaging. Για την αξιολόγηση των αποτελεσμάτων επιλέχθηκε η μετρική ***macro-averaged F-measure***. Στην θέση του βασικού αλγορίθμου για την εφαρμογή της μεθόδου χρησιμοποιήθηκαν οι Naive Bayes, Decision Tree & SVM with gaussian kernel.

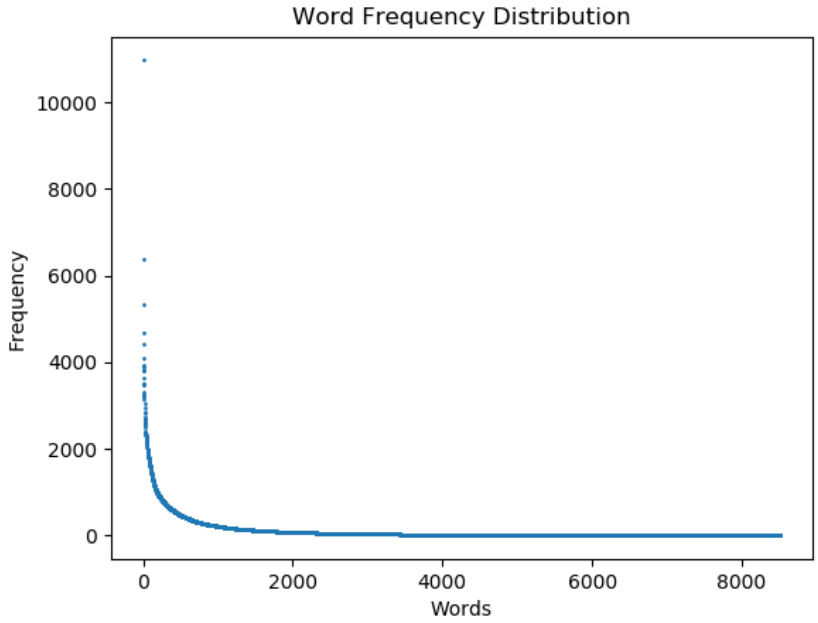
2.1 Προεπεξεργασία δεδομένων

Για την υλοποίηση των παραπάνω ήταν απαραίτητη η προ-επεξεργασία των δεδομένων προκειμένου να είναι έτοιμα προς χρήση. Αρχικά, από κάθε έγγραφο αφαιρέθηκε η πληροφορία για το πόσες λέξεις υπάρχουν ανά έγγραφο και ανά πρόταση. Σε δεύτερο στάδιο από την απλή καταγραφή των λέξεων που δινόταν, δημιουργήθηκε ένα διάνυσμα με το σύνολο των λέξεων και την καταγραφή των εμφανίσεων της κάθε λέξης ανά έγγραφο. Για την συνέχεια της εργασίας επιλέχθηκε η χρήση της συγκεκριμένης μορφοποίησης των δεδομένων. Η κατανομή της συχνότητας εμφάνισης της κάθε λέξης φαίνεται στο Σχήμα 2

[1] https://archive.ics.uci.edu/ml/machine-learning-databases/00418/



Σχήμα 1: Πλήθος εμφανίσεων της κάθε ετικέτας στο Train και στο Test σετ των δεδομένων.



Σχήμα 2: Αριθμός εμφανίσεων των λέξεων που βρίσκονται στο σύνολο της συλλογής. Η εκτύπωση είναι σε αύξοντα αριθμό εμφανίσεων.

2.2 Πειραματική Διαδικασία

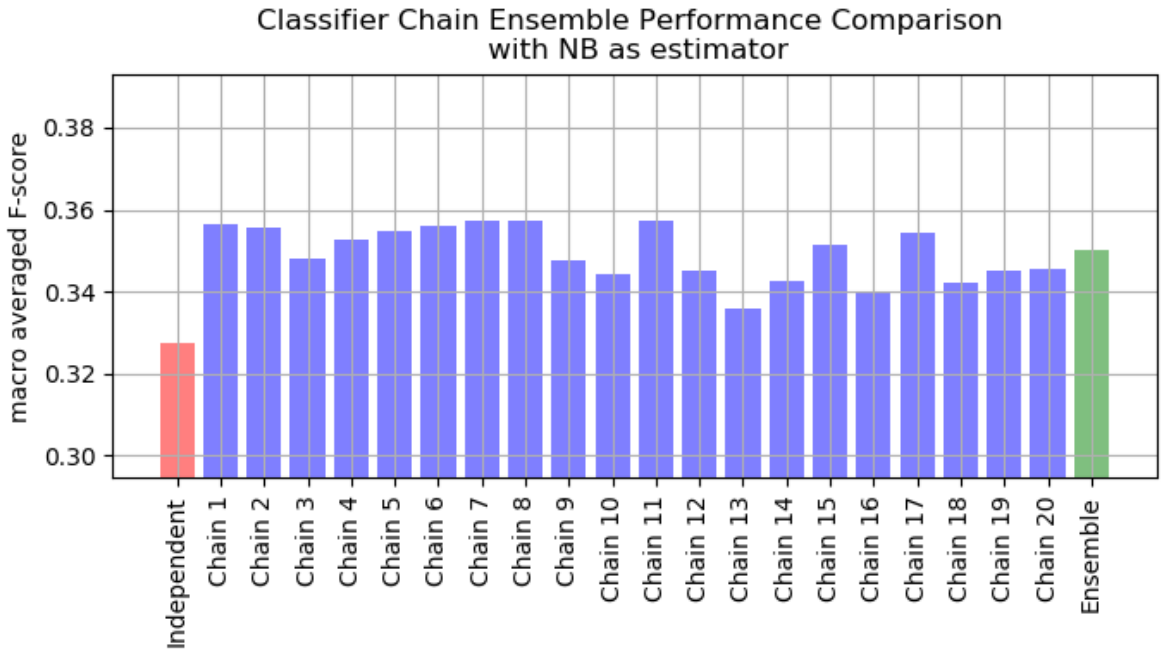
Αφού ολοκληρώθηκε η προ-επεξεργασία των δεδομένων, και για την εφαρμογή της μεθόδου classifier chain ήταν απαραίτητη η αντιμετώπιση του προβλήματος σε δυαδικό επίπεδο. Για την ικανοποίηση της συγκεκριμένης απαίτησης επιλέχθηκε η μέθοδος OneVsRest. Αφού προέκυψε το μοντέλο με τα συγκεκριμένα χαρακτηριστικά, εφαρμόσθηκε η μέθοδος Classifier Chain όπου ορίστηκαν 20 chains και η F-measure macro ως μετρική αξιολόγησης. Για το σύνολο της παραπάνω διαδικασίας επιλέχθηκε η εφαρμογή διαφόρων αλγορίθμων ταξινόμησης. Συγκεκριμένα εφαρμόσθηκαν οι SVM with gaussian kernel, Decision Tree & Naive Bayes.

Για την υλοποίηση του πρώτου μέρους χρησιμοποιήθηκε η python 2.7 και η βιβλιοθήκη scikit-learn. Επιπλέον, χρησιμοποιήθηκαν τροποποιημένα τα παραδείγματα που υπάρχουν στην βιβλιοθήκη για την εφαρμογή των μεθόδων και την εκτύπωση των αποτελεσμάτων. Πλέον της μεθόδου Classifier Chain, υπολογίστηκαν αποτελέσματα και με την OneVSRest μέθοδο.

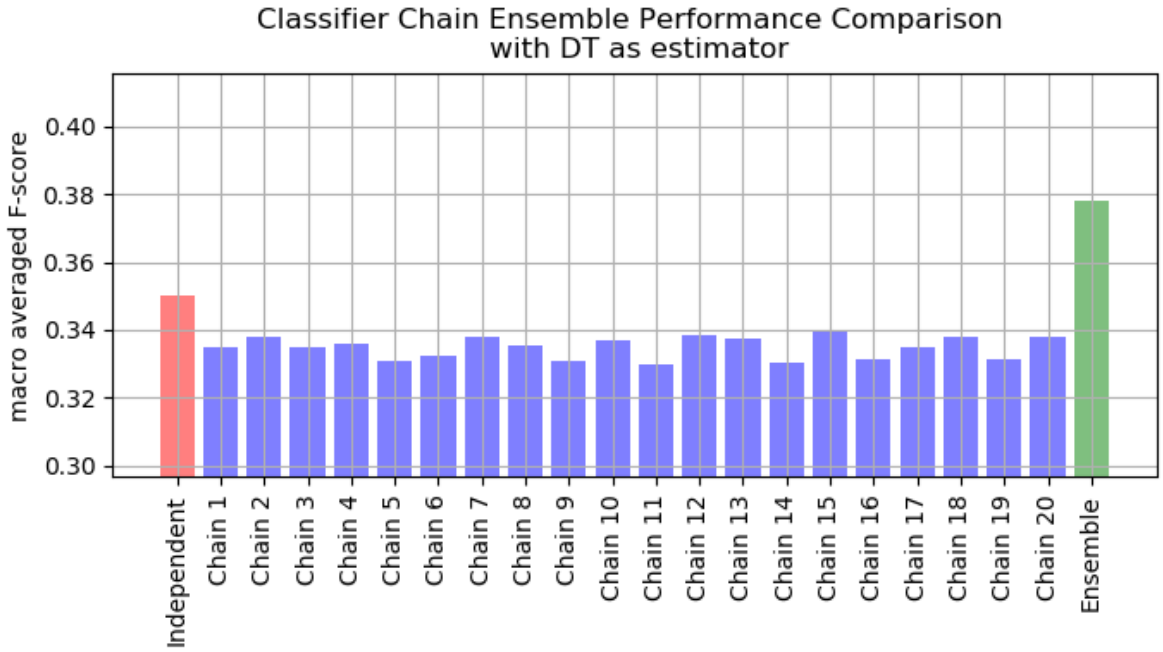
2.3 Αποτελέσματα

Στα Σχήματα 3, 4 & 5 φαίνονται οι αποδόσεις της εφαρμογής του classifier chain με διαφορετικό αλγόριθμο ταξινόμησης κάθε φορά.

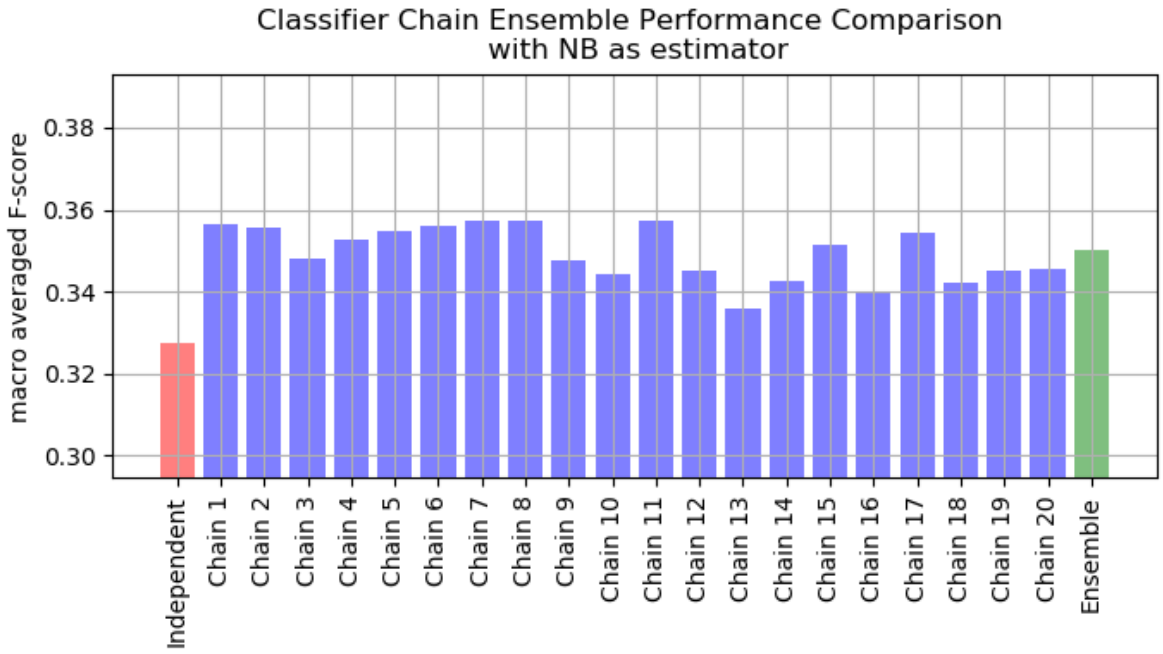
Στα Σχήματα 6, 7 & 8 φαίνονται οι αποδόσεις μετά την εφαρμογή της μεθόδου OneVsRest για κάθε ετικέτα με διαφορετικό αλγόριθμο ταξινόμησης κάθε φορά.



Σχήμα 3: F score Macro Average για όλα τα chains στον Classifier Chain. Η πρώτη μπάρα (Independent) μας δείχνει την απόδοση ταξινομητή πριν την εφαρμογή του Classifier Chain ενώ η τελευταία μας δείχνει το αποτέλεσμα της ensemble διαδικασίας. Ο αλγόριθμος ταξινόμησης είναι ο Naive Bayes



Σχήμα 4: F-score Macro Average για όλα τα chains στον Classifier Chain. Η πρώτη μπάρα (Independent) μας δείχνει την απόδοση ταξινομητή πριν την εφαρμογή του Classifier Chain ενώ η τελευταία μας δείχνει το αποτέλεσμα της ensemble διαδικασίας. Ο αλγόριθμος ταξινόμησης είναι ένα Δέντρο Απόφασης.



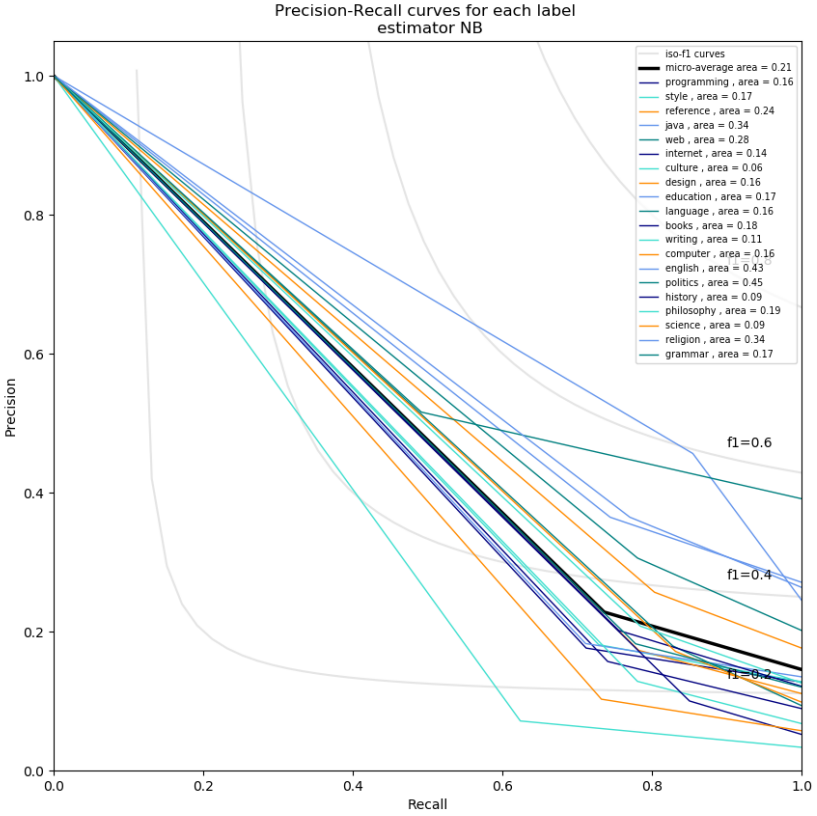
Σχήμα 5: F-score Macro Average για όλα τα chains στον Classifier Chain. Η πρώτη μπάρα (Independent) μας δείχνει την απόδοση ταξινομητή πριν την εφαρμογή του Classifier Chain ενώ η τελευταία μας δείχνει το αποτέλεσμα της ensemble διαδικασίας. Ο αλγόριθμος ταξινόμησης είναι ένα SVM with rbf kernel.

3 Μέρος Β

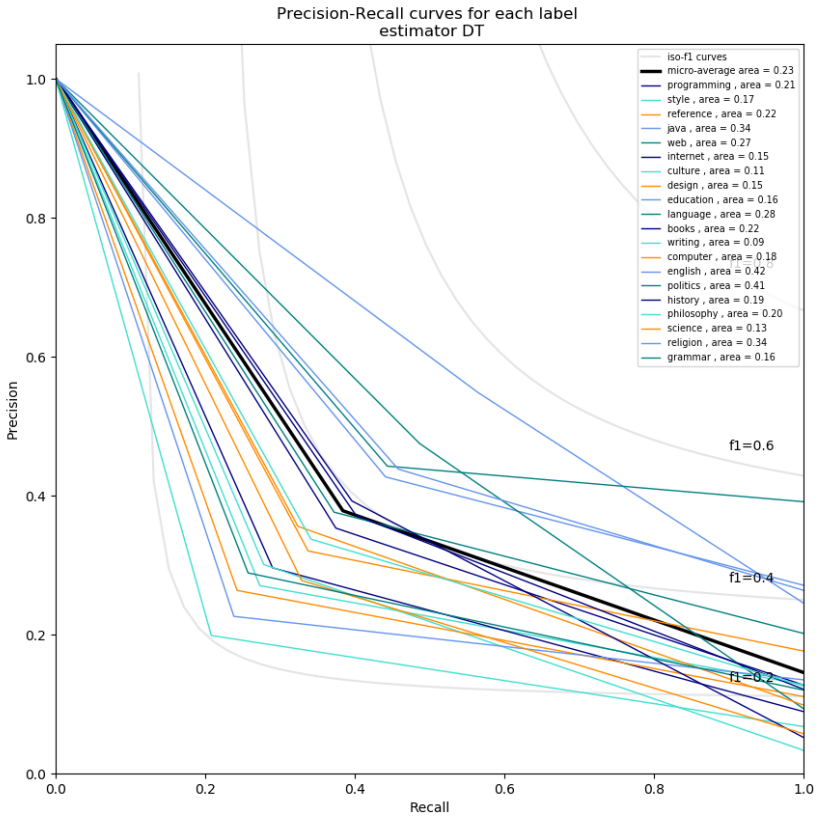
Στο δεύτερο μέρος της εργασίας ζητούμενο η επίλυση ενός προβλήματος μάθησης από σάκους περιπτώσεων. Για την απλοποίηση του προβλήματος κληθήκαμε να δουλέψουμε με την συχνότερη κλάση έτσι ώστε να μετατραπεί σε πρόβλημα δυαδικής ταξινόμησης. Σύμφωνα με το Σχήμα 1 η κλάση με τις περισσότερες εμφανίσεις στο σύνολο δεδομένων είναι η ***reference***. Για την συγκεκριμένη υλοποίηση αποφασίσθηκε ο μετασχηματισμός του προβλήματος σε ένα κλασσικό πρόβλημα μηχανικής μάθησης.

3.1 Μετασχηματισμός Δεδομένων

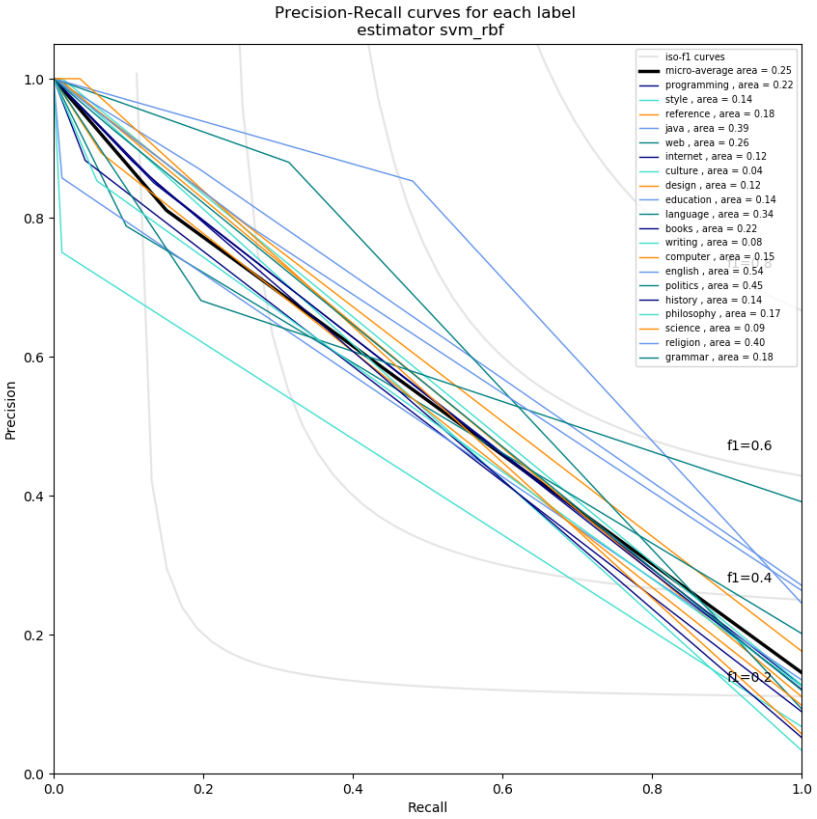
Για τον μετασχηματισμό των δεδομένων, χρησιμοποιήθηκε η "κλασσική υπόθεση" (classical assumption) ενώ "σάκος" θεωρήθηκε το κάθε έγγραφο και "περίπτωση" η κάθε πρόταση. Στη συνέχεια, για λόγους επάρκειας υπολογιστικών πόρων, περιορίστικε το μέγεθος του train set σε 3000 έγγραφα και το test set σε 1000. Σε αυτά τα σύνολα δεδομένων εφαρμόσθηκε ο παραπάνω μετασχηματισμός και κατόπιν ο αλγόριθμος 20-means. Μετά την εφαρμογή του 20-means και τον εκ νέου μετασχηματισμό της αναπαράστασης των δεδομένων, το πρόβλημα ήταν δυνατό να επιλυθεί με την εφαρμογή ενός αλγορίθμου δυαδικής ταξινόμησης. Για αυτό το μέρος χρησιμοποιήθηκε ο SVM linear, ο Naive Bayes και ο Decision Tree.



Σχήμα 6: micro F-score για όλες τις ετικέτες μετά την εφαρμογή της OneVSRest μεθόδου με αλγόριθμο ταξινόμησης τον Naive Bayes.



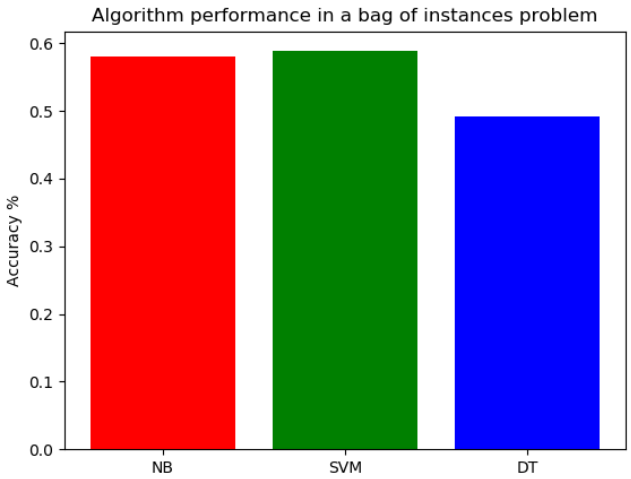
Σχήμα 7: micro F-score για όλες τις ετικέτες μετά την εφαρμογή της OneVSRest μεθόδου με αλγόριθμο ταξινόμησης ένα Δέντρο Απόφασης.



Σχήμα 8: micro F-score για όλες τις ετικέτες μετά την εφαρμογή της OneVSRest μεθόδου με αλγόριθμο ταξινόμησης ένα SVM.

Πίνακας 2: Αποτελέσματα για την ταξινόμηση με σάκους περιπτώσεων για την κλάση ”reference”

		true					
		NB		SVM		DT	
		0	1	0	1	0	1
predict	0	538	45	571	12	299	284
	1	375	43	400	18	224	194



Σχήμα 9: Αποτελέσματα ταξινόμησης για την περίπτωση της ταξινόμησης με σάκους περιπτώσεων.

3.2 Αποτέλεσματα

Τα αποτελέσματα της παραπάνω διαδικασίας περιγράφονται στον Πίνακα 2. Στην πρώτη περίπτωση ο ταξινομητής πέτυχε Acc= 0.58, στην δεύτερη πέτυχε Acc= 0.59 ενώ στην τρίτη Acc=0.49. Παρατηρώντας τον πίνακα σύγχυσης (Πίνακας 2) μπορεί κανείς να συμπεράνει εύκολα ότι τα αποτελέσματά μας δεν είναι ιδιαίτερα καλά. Είναι πολύ πιθανό να έχουμε χάσει πληροφορία στην αρχική προσέγγιση που κάναμε σε σχέση με την αναπαράσταση των λέξεων με bag of words και αυτό να μας έχει οδηγήσει εδώ.

4 Μέρος Γ

Στο τρίτο και τελευταίο μέρος της εργασίας, κληθήκαμε να δουλέψουμε στο πρόβλημα της ενεργούς μάθησης. Για την υλοποίηση του συγκεκριμένου μέρους, χρησιμοποιήσαμε μόνο το test set του DeliciousMIL dataset. Το πρόβλημα από πολλαπλών ετικετών μετασχηματίστηκε σε δυαδικής ταξινόμησης καθώς ασχοληθήκαμε μόνο με την σπανιότερη κλάση (grammar) όπως φαίνεται από την κατανομή των κλάσεων στο Σχήμα 1.

4.1 Περιγραφή Πειράματος

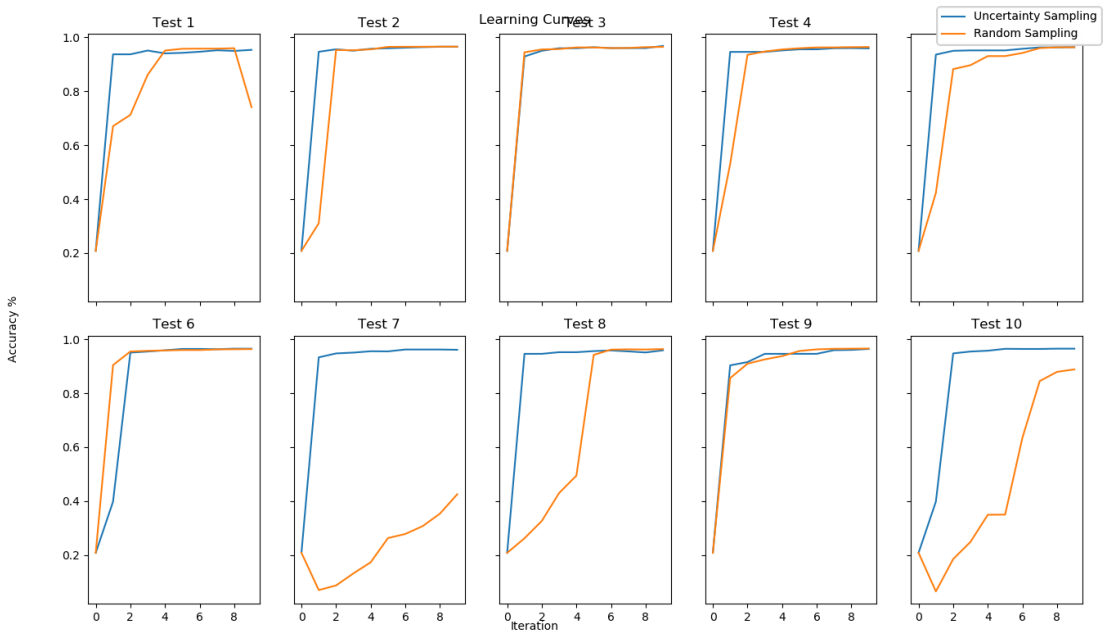
Όπως αναφέρθηκε και παραπάνω, για την υλοποίηση του συγκεκριμένου μέρους της εργασίας χρησιμοποιήθηκε μόνο το test set του αρχικού συνόλου δεδομένων, το οποίο χωρίστηκε σε δύο επι μέρους σύνολα, το "unlabeled pool set" και το test set. Για την αρχικοποίηση του πειράματος δημιουργήθηκε ένα πρώτο train set από το "unlabeled pool set" το οποίο αποτελείται από 8 παραδείγματα, 4 της κλάσης 0 και 4 της κλάσης 1.

Για την εφαρμογή του αλγορίθμου uncertainty sampling ακολουθήθηκε η παρακάτω διαδικασία: Με δεδομένο το αρχικό σύνολο εκπαίδευσης που περιγράφηκε παραπάνω, εκπαιδεύτηκε ο αλγόριθμος SVM. Από το μοντέλο που προέκυψε υπολογίστηκαν οι πιθανότητες για κάθε παράδειγμα του υπολειπόμενου unlabeled pool set. Σε αυτές τις τιμές εφαρμόσθηκε ο τύπος $x* = argmin_x|P(y_1|x) - 0.5|$ και επιλέχθηκε το επόμενο παράδειγμα το οποίο πρόκειται να δοθεί για annotation. Αφού έγινε ο σχολιασμός του συγκεκριμένου παραδείγματος, αυτό τοποθετήθηκε στο train set και αφαιρέθηκε από το unlabeled pool set. Με το train set εκπαιδεύτηκε εκ νέου ο αλγόριθμος SVM και συνεχίστηκε η διαδικασία για συνολικά δέκα επαναλήψεις.

4.2 Αποτελέσματα

Για την αξιολόγηση της διαδικασίας του active learning with uncertainty sampling επιλέχθηκε η εκτύπωση καμπυλών μάθησης σε σύγκριση με την τυχαία επιλογή δειγμάτων. Τα αποτελέσματα φαίνονται στο Σχήμα 10

Παρατηρώντας το Σχήμα 10 μπορούμε να συμπεράνουμε ότι τις περισσότερες φορές η μέθοδος active learning with uncertainty sampling οδηγεί γρηγορότερα και με μεγαλύτερη ασφάλεια στην αύξηση της ακρίβειας του ταξινομητή μας. Επιπλέον, είναι φανερό ότι τις φορές που υπερτερεί η random sampling είναι καθαρά θέμα τύχης. Μία ακόμα παρατήρηση που παρουσιάζει ενδιαφέρον είναι η σταθερότητα απόδοσης του μοντέλου με την μέθοδο uncertainty sampling σε αντίθεση με την random sampling κάτι που φαίνεται από την συμπεριφορά σε όλα τα πειράματα καθώς επίσης και ότι δεν επιρρεάζεται από τον αριθμό των επαναλήψεων σε αντίθεση με την random sampling, που για παράδειγμα στο πείραμα 1 στην δέκατη επανάληψη η ακρίβεια πρόβλεψης μειώνεται σημαντικά.



Σχήμα 10: Καμπύλες μάθησης i. για την διαδικασία Active Learning with Uncertainty sampling, ii. για Active Learning with Random Sampling. Για κάθε περίπτωση υπήρξαν 10 επαναλήψεις. Για το ίδιο dataset έγιναν 10 επαναλήψεις στο σύνολο.

5 Τεχνικά

Στην ενότητα αυτή γίνεται μία σύνοψη των εργαλείων που για την εκπόνηση της εργασίας. Όλα τα μέρη αυτής της εργασίας υλοποιήθηκαν σε Python 2.7 με την χρήση της βιβλιοθήκης scikit-learn (Pedregosa et al., 2011). Για ορισμένες διαδικασίες χρησιμοποιήθηκε κώδικας που βρίσκεται στα παραδείγματα της εν λόγω βιβλιοθήκης. Το σύνολο της εργασίας (κώδικας, σύνολα δεδομένων καθώς και τα αποτελέσματα μαζί με τα .tex αρχεία βρίσκονται στο https://github.com/ggravanis/AdvancedML .

Τα αρχεία κώδικα που αφορούν την δεύτερη εργασία είναι τα part[21-22-23].py. Στους φακέλους results/part[21-22-23] υπάρχουν επιπλέον αποτελέσματα από αυτά που βρίσκονται στο παρόν, ενώ στον φάκελο assignment2 βρίσκονται όλα τα αρχεία του τεύχους.

References

Pedregosa, F. et al. (2011). “Scikit-learn: Machine Learning in Python”. In: ***Journal of Machine Learning Research*** 12, pp. 2825–2830.

Soleimani, Hossein and David J. Miller (2016). “Semi-supervised Multi-Label Topic Models for Document Classification and Sentence Labeling”. In: ***Proceedings of the 25th ACM International on Conference on Information and Knowledge Management***. CIKM ’16. Indianapolis, Indiana, USA: ACM, pp. 105–114. ISBN: 978-1-4503-4073-1. DOI: 10.1145/2983323.2983752. URL: http://doi.acm.org/10.1145/2983323.2983752.