ΕΡΓΑΣΙΑ – ΜΗΧΑΝΙΚΗ ΜΑΘΗΣΗ

Οικονομόπουλος Αχιλλέας 3150132

Καραστατήρης Παναγιώτης 3150064

Το κύριο dataset πάνω στο οποίο λειτούργησε ο αλγόριθμος ήταν το PU3 με την προϋπόθεση ότι τα παραδείγματα χωρίζονται ως εξής: 80% παραδείγματα εκπαίδευσης βρίσκονται μέσα σε ένα φάκελο με τίτλο "education", 10% παραδείγματα επικύρωσης σε ένα φάκελο με τίτλο "validation", και 10% παραδείγματα αξιολόγησης σε ένα φάκελο με τίτλο "test". Οι 3 φάκελοι μπορούν είτε να περιέχουν απευθείας τα αρχεία(πιο γρήγορη εκτέλεση), είτε τους φακέλους "part". Το dataset είναι διαθέσιμο στο σύνδεσμο: http://nlp.cs.aueb.gr/software_and_datasets/PU123ACorpora.tar.gz

1.Οι κλάσεις – Αρχιτεκτονική

-fileAttributeAnalyzer

Η κλάση που παρέχει τις μεθόδους για την επιλογή των ιδιοτήτων που θα χρησιμοποιήσει ο αλγόριθμος. Η κύρια μέθοδος είναι η getBestAttributes(File folder), όπου folder θα είναι ο φάκελος με τα παραδείγματα εκπαίδευσης.

Η μέθοδος καλεί πρώτα την findAttributes(folder), η οποία διαβάζει το όνομα του αρχείου για να προσδιορίσει την κατηγορία (spam=1, ham=0) που ανήκει, και στη συνέχεια διαβάζει το αρχείο. Η ανάγνωση του αρχείου γίνεται γραμμή - γραμμή με έναν BufferedReader και έναν StringTokenizer. Κάθε φορά που ο StringTokenizer προχωρά στο επόμενο token, καθορίζει εάν το έχει συναντήσει σε οποιοδήποτε άλλο προηγούμενο αρχείο ελέγχοντας αν η λέξη έχει index στον πίνακα των tokens. Αν όχι, το προσθέτει στον πίνακα και δημιουργεί καταχωρήσεις - μετρητές στους πίνακες legitTks, spamTks που αντιστοιχούν στο ίδιο index του πίνακα tokens και αντιπροσωπεύουν τον αριθμό των legit και spam μηνυμάτων στα οποία έχει εντοπιστεί τουλάχιστον μία φορά η συγκεκριμένη ιδιότητα- token. Όταν τελειώσει με τον άνω έλεγχο εξετάζει εάν έχει ξανασυναντήσει το token στο ίδιο αρχείο και αν όχι, ενημερώνει τον αντίστοιχο counter.

Αφού δημιουργηθεί ο πίνακας με κάθε token που συναντήθηκε στα αρχεία εκπαίδευσης, καλείται η calculateIG() για να υπολογιστεί το Information Gain κάθε ιδιότητας. Κάθε παράγοντας που θα χρησιμοποιηθεί στις σχέσεις αρχικοποιείται στο 0 και η τιμή του αλλάζει μόνο εάν γίνει βέβαιο πως δε θα τείνει στο $\pm$ άπειρο. Οι τύποι που χρησιμοποιούνται είναι αυτοί των διαφανειών και ο λόγος για τον οποίο επιλέχθηκαν τα σύνολα που συμμετέχουν στον υπολογισμό των πιθανοτήτων εξηγείται στα σχόλια του κώδικα.

Τέλος αφού ενημερωθούν όλοι οι πίνακες της κλάσης από τις private μεθόδους που κλήθηκαν πιο πριν, η getBestAttributes ελέγχει ποιες από τις ιδιότητες ξεπερνούν ένα ελάχιστο Information Gain που τίθεται (αιτιολόγηση επιλογής της υπερπαραμέτρου-minIG στο κομμάτι για τις υπερπαραμέτρους) και τις απορρίπτει ή τις τοποθετεί στον πίνακα bestTokens, που αποτελεί τον πίνακα ιδιοτήτων που θα χρησιμοποιήσει τελικά ο αλγόριθμος, και τον οποίο επιστρέφει στη main.

Υποσημείωση: Η κλάση αυτή υπολογίζει και το πλήθος του education set βάσει του αριθμού των αρχείων που προσπελαύνει και το επιστρέφει με την getEdSetSize().

ΕΡΓΑΣΙΑ – ΜΗΧΑΝΙΚΗ ΜΑΘΗΣΗ

-LR (Logistic Regression)

Η κλάση που παρέχει όλες τις μεθόδους που έχουν να κάνουν με Logistic Regression, τόσο για την εκπαίδευση του αλγορίθμου, όσο και για την εκτίμηση της κατηγορίας ενός παραδείγματος επικύρωσης/αξιολόγησης. Η κατασκευή αντικειμένου LR, δέχεται μια υπερπαράμετρο λ και μια ArrayList<String> με τις ιδιότητες που θα αναζητήσουμε στα αρχεία εκπαίδευσης (εδώ αυτές που επέστρεψε η findBestAttributes - βλ.main). Οι δύο public μέθοδοι που προσφέρει, είναι η LRegression(File folder,int size) και η evaluate(File folder, ArrayList<Double> Win, int size).

-educate

Προκειμένου να παραχθούν οι καμπύλες μάθησης το πρόγραμμα εκπαιδεύεται σταδιακά σε ένα ποσοστό των παραδειγμάτων εκπαίδευσης. Για να διατηρηθεί λοιπόν η δομή των φακέλων με τα αρχεία εισόδου η educate (και κάθε μέθοδος που ασχολείται με αυτά, όπως φαίνεται και παρακάτω) παίρνει ως όρισμα τον φάκελο παραδειγμάτων εκπαίδευσης, αλλά πρέπει να ξέρει και πόσα κομμάτια του θα εξετάσει (κάθε φορά από 1 μέχρι (size/10 * τον αριθμό του βήματος)).

Όσον αφορά τη λειτουργία της, παράγει τα βάρη πάνω στα οποία θα βασιστεί η ταξινόμηση και είναι επαναληπτική. Οι επαναλήψεις συμβαίνουν μέχρι να συμπληρωθεί ο μέγιστος αριθμός εποχών, ή μέχρι να παρατηρηθεί σύγκλιση σφάλματος με τον εξής τρόπο: αν 5 διαδοχικές επαναλήψεις του αλγορίθμου ανά δύο διαφέρουν κατ' απόλυτο λιγότερο από 0.5, θεωρούμε ότι το σφάλμα έχει ξεκινήσει να συγκλίνει.

Αφού λοιπόν η μέθοδος μπει στο loop, καλεί την epochfunction(folder,size). Η μέθοδος αυτή διαβάζει πάλι ένα-ένα τα αρχεία εκπαίδευσης. Η ανάγνωση γίνεται όπως και παραπάνω, με τη διαφορά ότι το indexing αφορά τον πίνακα των bestAttributes. Αν όντως βρεθεί index για το token που εξετάζεται, σημειώνουμε στον πίνακα X (διάνυσμα χαρακτηριστικών του τρέχοντος παραδείγματος) το 1 στο ίδιο ακριβώς index. Γενικά, ο πίνακας X είναι ένας πίνακας μεγέθους ίδιου με αυτό του bestAttributes + 1 (X0) που υποδεικνύει για κάθε μία από τις ιδιότητες αν υπάρχει(1) ή όχι(0) στο αρχείο που εξετάζεται εκείνη τη στιγμή. Στη συγκεκριμένη υλοποίηση, αντί ο X να αρχικοποιείται με μηδενικά σε κάθε επανάληψη, έχει τεθεί ως global και απλώς μηδενίζεται πριν ξεκινήσει η ανάγνωση ενός νέου αρχείου. Έχοντας δημιουργήσει το διάνυσμα X για ένα παράδειγμα, η epochfunction καλεί την updateWeights(category).

Η updateWeights(double y) αφορά τον υπολογισμό της συνάρτησης λάθους (αποθήκευση σε global μεταβλητή για να τη βλέπει η educate) και της λογιστικής στοχαστικής ανάβασης, χρησιμοποιώντας τους τύπους των διαφανειών και της άσκησης 18.2. Το όρισμα y είναι η κατηγορία του τρέχοντος παραδείγματος που έχει ανιχνεύσει η epochfunction στον τίτλο του αρχείου. Όλες οι ανανεώσεις βαρών από κάθε παράδειγμα γίνονται πάνω στον εσωτερικό πίνακα W της κλάσης.

Αφού εκτελεσθούν τα παραπάνω βήματα ο έλεγχος επιστρέφει στην educate που αυξάνει το μετρητή πιθανής σύγκλισης αν το |lastErr – s| είναι μικρότερο του 0.5, ενώ σε διαφορετική περίπτωση τον μηδενίζει. Τέλος, αφού παραβιαστεί μία από τις συνθήκες της while, επιστρέφει τον πίνακα W όπως έχει τροποποιηθεί μέχρι εκείνη τη στιγμή

-evaluate

Η evaluate είναι ουσιαστικά η μέθοδος ταξινόμησης ενός παραδείγματος/πολλαπλών παραδειγμάτων (παραδείγματα του folder από 1 μέχρι (size/10 * τον αριθμό του βήματος)). Όσον αφορά τον υπολογισμό

του διανύσματος ιδιοτήτων X ενός παραδείγματος, λειτουργεί με τον ίδιο ακριβώς τρόπο που λειτουργεί και η epochfunction, ενώ διαφέρει στο 2ο όρισμα, στην μέθοδο/συνάρτηση που καλεί στο τέλος της, και στο γεγονός ότι καταγράφει πόσα παραδείγματα από αυτά που ελέγχει είναι πραγματικά spam (εδώ ο counter relevant).

Το 2ο όρισμα είναι ένας πίνακας βαρών (εδώ αυτός που έχει υπολογίσει η educate - βλ. main) βάσει του οποίου θα γίνει η ταξινόμηση. Μετά τον καθορισμό του X, η evaluate καλεί την sigmoidClassification(ArrayList<Double> Win), όπου Win ο πίνακας βαρών, και της επιστρέφεται η κατηγορία ταξινόμησης με τη χρήση του τύπου $(1/(1+e^{-W^T X}))$. Αν η συνάρτηση δηλαδή δώσει τιμή >0.5 επιστρέφει 1(spam), ενώ αν δώσει <=0.5 επιστρέφει 0(ham). Κατόπιν, η sigmoidClassification θα ενημερώσει τον predSpam counter εάν ταξινομήσει το παράδειγμα ως spam, κάτι που θα χρησιμεύσει στον υπολογισμό των precision και recall.

Όταν η μέθοδος ταξινόμησης επιστρέψει την πρόβλεψή της, η evaluate ελέγχει αν ήταν όντως σωστή και ενημερώνει τον αντίστοιχο counter correct / incorrect. Τέλος, αν το παράδειγμα κατατάχθηκε λανθασμένα στα spam μηνύματα, αυξάνεται κατά ένα ο counter falseSpam που εξυπηρετεί τον ίδιο σκοπό με αυτό του predSpam.

-resetStats
Μηδενίζει όλους τους global counters του αντικειμένου ώστε σε περίπτωση που θέλουμε να ξανακαλέσουμε την evaluate για διαφορετικό σύνολο παραδειγμάτων, αυτοί να ξεκινήσουν να μετρούν απ' την αρχή

-printStats
Υπολογίζει και εμφανίζει Accuracy, Precision, Recall, F1 από τα δεδομένα αξιολόγησης που έχει το αντικείμενο αποθηκευμένα εκείνη τη στιγμή (δηλαδή τους άνω counters). Σκοπός είναι να κληθεί μετά από μια evaluate.

-NaiveBayes

Η κλάση που περιέχει όλες τις μεθόδους που υλοποιούν εκπαίδευση και ταξινόμηση με Naive Bayes. Κατά την ανάγνωση αρχείων χρησιμοποιεί τεχνικές που έχουν αναλυθεί προηγουμένως, τόσο για την ανίχνευση ιδιοτήτων, όσο και για την τήρηση μετρητών που χρησιμοποιούνται στις στατιστικές, γι' αυτό και θα παραληφθεί η επεξήγησή τους.

-**educate**
Μιας και η Naive Bayes μέθοδος χρησιμοποιεί πιθανότητες, ο κύριος στόχος της εκπαίδευσης είναι να καταγράψει τα στοιχεία των συνόλων που συμμετέχουν στον υπολογισμό των πιθανοτήτων αυτών. Για το λόγο αυτό χρησιμοποιούνται 2 λίστες (existsLegit, existsSpam) μεγέθους ίδιου με το πλήθος των ιδιοτήτων στις οποίες θα βασιστεί η εκπαίδευση. Η μία λίστα στη θέση i περιέχει τον αριθμό των legit αρχείων στα οποία έχει ανιχνευθεί τουλάχιστον μία φορά η λέξη attSet(i), ενώ η άλλη τον αντίστοιχο αριθμό spam αρχείων. Αφού λοιπόν η μέθοδος διαβάσει όλα τα αρχεία του train set, έχει ενημερώσει πλήρως τις δύο λίστες αυτές και έχει καταγράψει το πλήθος των legit και spam μηνυμάτων του set. Σε αυτό το σημείο καλεί την calculateProbabilities που υπολογίζει τις καθολικές πιθανότητες:

P(C=1) -> πιθανότητα ενός μηνύματος να είναι spam (spam messages/total messages)
P(C=0) -> πιθανότητα ενός μηνύματος να είναι legit (legit messages/total messages)

και για κάθε ιδιότητα $X_{(i)}$ τις εξής πιθανότητες (τις οποίες αποθηκεύει σε 4 διαφορετικές ArrayList):
P($X_{(i)}$=1|C=1) -> πιθανότητα η $X_{(i)}$ να υπάρχει σε ένα spam μήνυμα (spam messages that contain $X_{(i)}$/total spam messages)
P($X_{(i)}$=0|C=1) -> πιθανότητα η $X_{(i)}$ να μην υπάρχει σε ένα spam μήνυμα ((total spam messages – spam messages that contain $X_{(i)}$)/total spam messages)
P($X_{(i)}$=1|C=0), P($X_{(i)}$=0|C=0) -> αντίστοιχες με τις άνω για τα legit μηνύματα

Στον υπολογισμό των δεσμευμένων πιθανοτήτων χρησιμοποιείται εκτιμήτρια Laplace.

-evaluate
Η evaluate εδώ λειτουργεί ακριβώς όπως και στην LR, με τη διαφορά ότι στο τέλος της ανάγνωσης του αρχείου καλεί για ταξινόμηση την NBayesClassification περνώντας ως όρισμα το διάνυσμα ιδιοτήτων X που έχει καθορίσει. Τότε εκτελεί τα εξής βήματα για να υπολογίσει τις πιθανότητες P(C=1|X) και P(C=0|X)
1) Αρχικοποίηση των δύο πιθανοτήτων ως P(C=1) και P(C=0) αντίστοιχα
2) Έλεγχος για την τιμή x κάθε ιδιότητας $X_{(i)}$ στο αρχείο ελέγχοντας τη θέση i στο διάνυσμα X
3) Πολλαπλασιασμός του P($X_{(i)}$=x|C=1) στο P(C=1|X) και του P($X_{(i)}$=x|C=0) στο P(C=0|X)
4) Σύγκριση P(C=1|X), P(C=0|X) και επιστροφή εκτίμησης

-resetStats
Ίδια με της LR

-resetClass
Εδώ χρειάζεται και μια επιπλέον μέθοδος επανεκκίνησης σε περίπτωση που θελήσουμε να κάνουμε εκ νέου εκπαίδευση του αλγορίθμου, προκειμένου να μηδενιστούν μετρητές και αποθηκευμένες πιθανότητες της προηγούμενης εκπαίδευσης.

-printStats
Ίδια με της LR

2.Η main – Χρήση

Η main σε αυτήν την περίπτωση είναι πολύ απλή, καθώς το μόνο που κάνει είναι να καλεί τις μεθόδους των προηγούμενων κλάσεων, και να φροντίζει να περνά τα αντικείμενα-επιστροφές της μίας ως είσοδο της άλλης. Έχουμε λοιπόν τα εξής βήματα:

-Δημιουργία των αρχικών αντικειμένων που θα χρησιμοποιηθούν

-Κλήση της getBestAttributes με όρισμα τον φάκελο που περιέχει τα παραδείγματα εκπαίδευσης και αποθήκευση του πίνακα ιδιοτήτων που θα επιστραφεί

-Κλήση της getEdSetSize για να πάρουμε το πλήθος των παραδειγμάτων εκπαίδευσης και αποθήκευσή του στη μεταβλητή size

α) Δημιουργία αντικειμένου LR με όρισμα την υπερπαράμετρο λ και το ArrayList ιδιοτήτων της getBestAttributes

ΕΡΓΑΣΙΑ – ΜΗΧΑΝΙΚΗ ΜΑΘΗΣΗ

-Loop μηχανικής μάθησης με λογιστική παλινδρόμηση για (i*10%) των παραδειγμάτων εκπαίδευσης

1. Υπολογισμός του πλήθους των παραδειγμάτων εκπαίδευσης που θα χρησιμοποιηθούν ως (size/10)*10 (+ το υπόλοιπο του size/10 αν βρισκόμαστε στην τελευταία επανάληψη)
2. Επανεκκίνηση των μετρητών του αντικειμένου για να σιγουρευτούμε πως θα ξεκινήσουν από το 0
3. Εκπαίδευση του αλγορίθμου στα x/8 των παραδειγμάτων εκπαίδευσης περνώντας ως 2ο όρισμα του educate το επιθυμητό x και ανάκτηση του πίνακα βαρών
4. Αξιολόγηση πάνω στα ίδια ακριβώς δεδομένα εκπαίδευσης και εμφάνιση των στατιστικών
5. Επανεκκίνηση των μετρητών για να λάβουμε παρακάτω τις σωστές στατιστικές
6. Αξιολόγηση στα παραδείγματα επικύρωσης και εμφάνιση των στατιστικών
7. Επανεκκίνηση των μετρητών για να λάβουμε παρακάτω τις σωστές στατιστικές
8. Αξιολόγηση στα παραδείγματα αξιολόγησης και εμφάνιση των στατιστικών

β) Δημιουργία αντικειμένου NaiveBayes με όρισμα το ArrayList ιδιοτήτων της getBestAttributes

-Loop μηχανικής μάθησης με Naive Bayes για (i*10%) των παραδειγμάτων εκπαίδευσης

Αντίστοιχο με το πάνω loop (δεν αποθηκεύουμε κάποιο πίνακα βαρών) + μια επανεκκίνηση των μετρητών εκπαίδευσης με την resetClass() μετά το βήμα 1.

Στον κώδικα της main που παραδίδεται περιλαμβάνονται και τα δύο loops με αυτό της λογιστικής παλινδρόμησης να βρίσκεται εντός σχολίων (πολύ πιο γρήγορος ο Bayes).

3.Μέθοδοι τεχνητής νοημοσύνης

-Λογιστική Παλινδρόμηση με ανάβαση κλίσης και όρο κανονικοποίησης

Όπως και οι τύποι, έτσι και ο αλγόριθμος ακολουθεί βήματα που έχουν δοθεί στις διαφάνειες. Καθώς όμως οι διαφάνειες αναφέρονται σε στοχαστική κατάβαση κλήσης γραμμικού διαχωρισμού, εφαρμόστηκαν οι εξής αλλαγές:

-Συνάρτηση λάθους: $E_i(\vec{w}) = \frac{1}{2}\left(\frac{1}{1+e^{-W^T x^{(i)}}} - y^{(i)}\right)^2$

-Κανόνας ενημέρωσης βαρών: $w_l = (1 - 2\lambda\eta) * w_l + \eta\left[y^{(i)} - \frac{1}{1+e^{-W^T x^{(i)}}}\right] * x_l$, όπου το l παίρνει τιμές από 0 μέχρι και το μέγεθος του πίνακα βαρών (που προκύπτει από τον πίνακα ιδιοτήτων που επιστρέφει η getBestAttributes). Δηλαδή μία επανάληψη / η εξέταση ενός μόνο παραδείγματος, επηρεάζει όλα τα βάρη του πίνακα W.

Ο κανόνας προκύπτει από ανάλυση της παραγώγου (όπως μας καθοδηγεί η ανάβαση κλήσης) της πιθανοφάνειας $\nabla_{\vec{w}} l(\vec{w})$ (πόσο σίγουροι είμαστε ότι το αποτέλεσμα του αλγορίθμου είναι σωστό) που καλούμαστε να μεγιστοποιήσουμε, ενώ η παράμετρος λ και γενικότερα η κανονικοποίηση στοχεύει στην ανάθεση πολύ μικρών βαρών σε όσες περισσότερες ιδιότητες γίνεται, με στόχο την καλύτερη γενίκευση του αλγορίθμου και την αποφυγή overfitting.

ΕΡΓΑΣΙΑ – ΜΗΧΑΝΙΚΗ ΜΑΘΗΣΗ

-Naive Bayes με ανεξάρτητες πιθανότητες για κάθε ιδιότητα του διανύσματος X και εκτιμήτρια Laplace

Χρησιμοποιούνται ακριβώς οι τύποι που περιλαμβάνονται στη διαφάνεια 6 της διάλεξης 16, ενώ ο τρόπος με τον οποίο υπολογίζεται κάθε παράγοντας των τύπων αναφέρεται στην educate της NaiveBayes.

-Υπολογισμός Accuracy, Precision, Recall

Accuracy = correct / (correct + incorrect), δηλαδή οι σωστές αξιολογήσεις προς το σύνολο όλων των αξιολογήσεων

Precision = (predSpam - falseSpam) / predSpam, δηλαδή τα μηνύματα που **σωστά** κατέταξε ο αλγόριθμος ως spam (true positives) προς το σύνολο όλων των μηνυμάτων που κατέταξε ως spam (true positives + false positives).

Recall = (predSpam - falseSpam) / relevant, δηλαδή τα true positives προς το σύνολο όλων των spam μηνυμάτων -> spam που κατατάχθηκαν στα spam (true positives) + spam που κατατάχθηκαν στα legit (false negatives)

-Επιλογή υπερπαραμέτρων

Κατά την εκτέλεση του προγράμματος, χρησιμοποιούνται οι εξής υπερπαράμετροι:

-(LR,NB)minIG: Χρησιμοποιείται για την επιλογή των k καλύτερων ιδιοτήτων βάσει IG, χωρίς να χρειάζεται συνεχές sort των IG που υπολογίστηκαν προηγουμένως για να καθοριστεί ποια ιδιότητα πρέπει να αντικατασταθεί και ποια όχι. Αν σκοπεύουμε λοιπόν να πάρουμε ~500 ιδιότητες, ρυθμίζουμε το minIG μέχρι ο αλγόριθμος κατά την εκτέλεσή του να μας ενημερώσει πως επέλεξε τον επιθυμητό αριθμό ιδιοτήτων.

-Πλήθος ιδιοτήτων: Ο αριθμός δηλαδή πάνω στον οποίο θα βασίζεται η ρύθμιση των προηγούμενων IG. Εδώ για το dataset του PU3 αποφασίστηκε να είναι το 217 για την LR και το 95 για την NB βάσει των αποτελεσμάτων στο validation set.

Για το πλήθος στην LR ρόλο έπαιξε και η συνολική ταχύτητα των educate και evaluate καθώς περισσότερες ιδιότητες = περισσότερες εκχωρήσεις στη λίστα attSet = περισσότερες αναζητήσεις index στη δημιουργία του διανύσματος ιδιοτήτων X για κάθε αρχείο, καθυστέρηση που ενισχύεται ακόμη παραπάνω από τη χρήση πολλαπλών εποχών κατά την εκπαίδευση. Γενικά από ένα σημείο και μετά θεωρήθηκε πως η βελτίωση στα αποτελέσματα, αν και μπορεί να επιτευχθεί, δεν είναι ανάλογη της χρονικής επιβάρυνσης.

-η, λ: Όπως και το πλήθος ιδιοτήτων, έτσι και αυτές βασίζονται σε αποτελέσματα του αλγορίθμου πάνω στο validation set. Εδώ παίρνουν τις τιμές 0.007 και 0.01 αντίστοιχα και γενικά τιμές πολύ κοντά σε αυτές δείχνουν να έχουν το ίδιο καλά αποτελέσματα πάνω στο set του PU3.

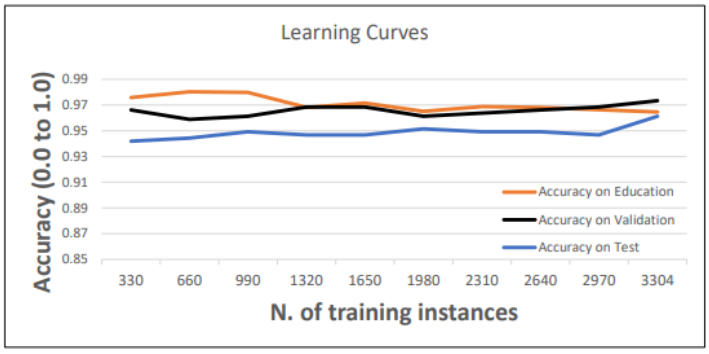
-maxEpochs: Φαίνεται να επηρεάζει σε πολύ μικρότερο βαθμό τα αποτελέσματα της Logistic Regression με είσοδο το validation set μιας και α) συνήθως η ενημέρωση βαρών λήγει λόγω σύγκλισης λάθους, β) η ανανέωση της $2^{ης}$ εποχής πάνω στα βάρη της $1^{ης}$ φαίνεται να είναι ο μεγαλύτερος παράγοντας της αυξημένης ακρίβειας του αλγορίθμου (μεγάλη μείωση ποσοστού λαθών).

Ως αποτελέσματα του αλγορίθμου πάνω στο validation set εννοούνται τα αποτελέσματα της ταξινόμησης του set όταν ο αλγόριθμος έχει εκπαιδευτεί στο 100% των παραδειγμάτων εκπαίδευσης.

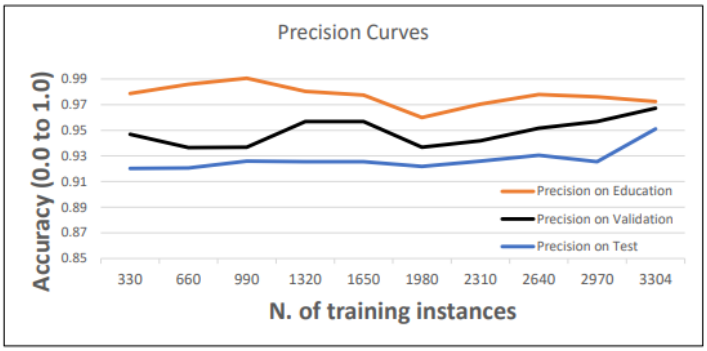
ΕΡΓΑΣΙΑ – ΜΗΧΑΝΙΚΗ ΜΑΘΗΣΗ

4.Δυνατότητες

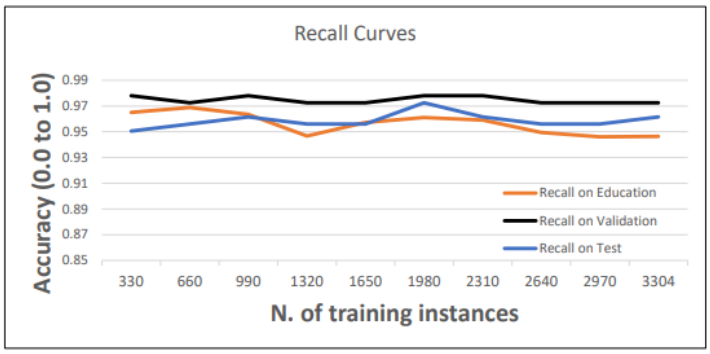
Logistic Regression



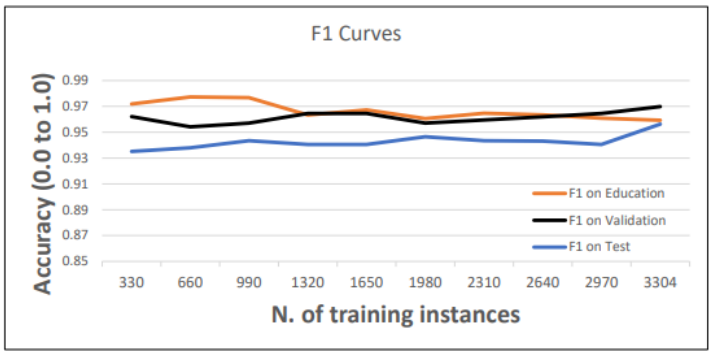
N. of instances	Accuracy on Education	Accuracy on Validation	Accuracy on Test
330	0.975757576	0.966101695	0.94188862
660	0.98030303	0.958837772	0.944309927
990	0.97979798	0.96125908	0.949152542
1320	0.968181818	0.968523002	0.946731235
1650	0.971515152	0.968523002	0.946731235
1980	0.965151515	0.96125908	0.95157385
2310	0.968831169	0.963680387	0.949152542
2640	0.968181818	0.966101695	0.949152542
2970	0.966329966	0.968523002	0.946731235
3304	0.964588378	0.973365617	0.96125908



N. of instances	Precision on Education	Precision on Validation	Precision on Test
330	0.978723404	0.946808511	0.920212766
660	0.985915493	0.936507937	0.920634921
990	0.990610329	0.936842105	0.925925926
1320	0.980427046	0.956756757	0.925531915
1650	0.977496484	0.956756757	0.925531915
1980	0.959954233	0.936842105	0.921875
2310	0.970384995	0.941798942	0.925925926
2640	0.977934687	0.951612903	0.930481283
2970	0.976152623	0.956756757	0.925531915
3304	0.972477064	0.967213115	0.951086957



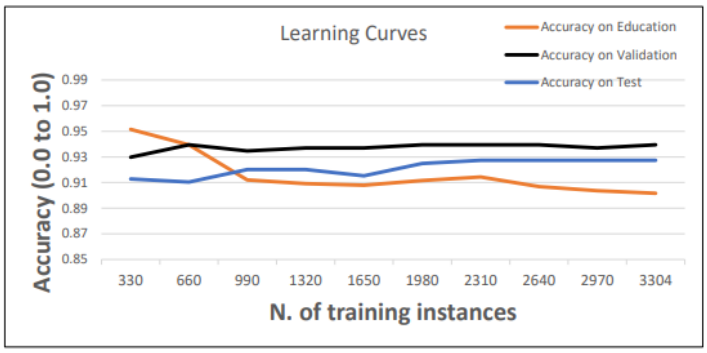
N. of instances	Recall on Education	Recall on Validation	Recall on Test
330	0.965034965	0.978021978	0.950549451
660	0.968858131	0.972527473	0.956043956
990	0.96347032	0.978021978	0.961538462
1320	0.946735395	0.972527473	0.956043956
1650	0.957300275	0.972527473	0.956043956
1980	0.961053837	0.978021978	0.972527473
2310	0.95902439	0.978021978	0.961538462
2640	0.949443016	0.972527473	0.956043956
2970	0.946070878	0.972527473	0.956043956
3304	0.946428571	0.972527473	0.961538462



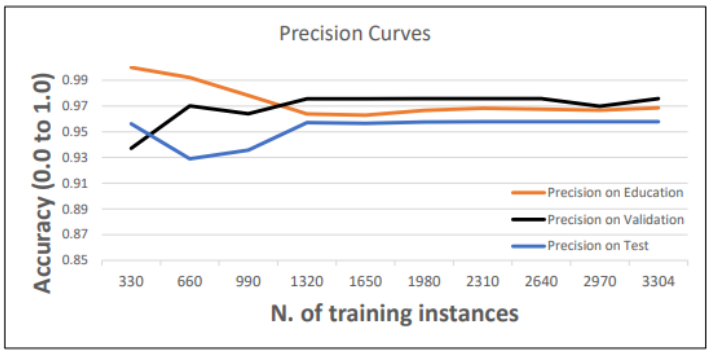
N. of instances	F1 on Education	F1 on Validation	F1 on Test
330	0.971830986	0.962162162	0.935135135
660	0.977312391	0.954177898	0.938005391
990	0.976851852	0.956989247	0.943396226
1320	0.963286713	0.964577657	0.940540541
1650	0.967292971	0.964577657	0.940540541
1980	0.960503721	0.956989247	0.946524064
2310	0.964671246	0.959568733	0.943396226
2640	0.963478261	0.961956522	0.943089431
2970	0.960876369	0.964577657	0.940540541
3304	0.959276018	0.969863014	0.956284153

ΕΡΓΑΣΙΑ – ΜΗΧΑΝΙΚΗ ΜΑΘΗΣΗ

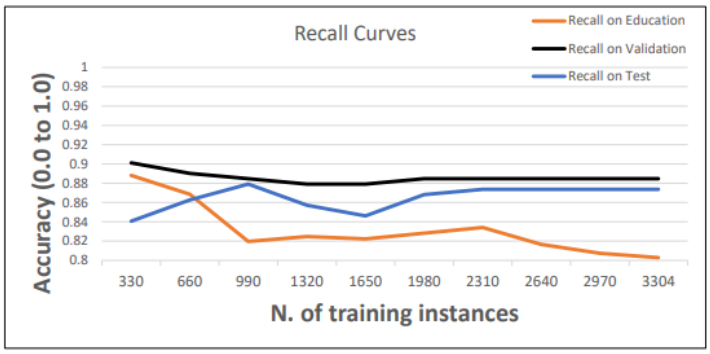
Naive Bayes



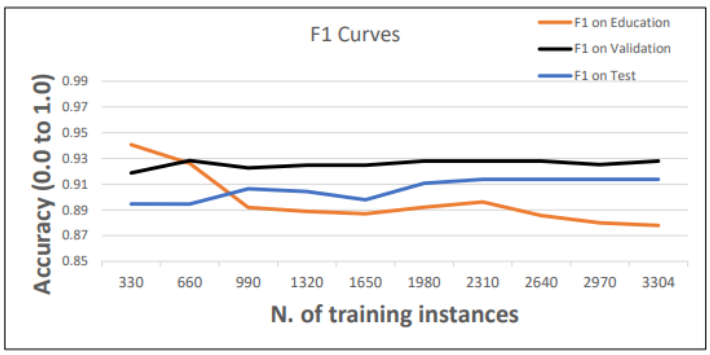
N. of instances	Accuracy on Education	Accuracy on Validation	Accuracy on Test
330	0.951515152	0.929782082	0.91283293
660	0.939393939	0.939467312	0.910411622
990	0.912121212	0.934624697	0.920096852
1320	0.909090909	0.937046005	0.920096852
1650	0.907878788	0.937046005	0.915254237
1980	0.911616162	0.939467312	0.924939467
2310	0.914285714	0.939467312	0.927360775
2640	0.906818182	0.939467312	0.927360775
2970	0.903703704	0.937046005	0.927360775
3304	0.901634383	0.939467312	0.927360775



N. of instances	Precision on Education	Precision on Validation	Precision on Test
330	1	0.937142857	0.95625
660	0.992094862	0.97005988	0.928994083
990	0.978201635	0.964071856	0.935672515
1320	0.963855422	0.975609756	0.957055215
1650	0.962903226	0.975609756	0.956521739
1980	0.96657754	0.975757576	0.957575758
2310	0.968289921	0.975757576	0.957831325
2640	0.96751269	0.975757576	0.957831325
2970	0.966789668	0.969879518	0.957831325
3304	0.968516984	0.975757576	0.957831325



N. of instances	Recall on Education	Recall on Validation	Recall on Test
330	0.888111888	0.901098901	0.840659341
660	0.868512111	0.89010989	0.862637363
990	0.819634703	0.884615385	0.879120879
1320	0.824742268	0.879120879	0.857142857
1650	0.82231405	0.879120879	0.846153846
1980	0.828178694	0.884615385	0.868131868
2310	0.834146341	0.884615385	0.873626374
2640	0.816623822	0.884615385	0.873626374
2970	0.807395994	0.884615385	0.873626374
3304	0.802884615	0.884615385	0.873626374



N. of instances	F1 on Education	F1 on Validation	F1 on Test
330	0.940740741	0.918767507	0.894736842
660	0.926199262	0.928366762	0.894586895
990	0.891925466	0.922636103	0.906515581
1320	0.888888889	0.924855491	0.904347826
1650	0.887072808	0.924855491	0.897959184
1980	0.892041949	0.92795389	0.910662824
2310	0.896226415	0.92795389	0.913793103
2640	0.885687732	0.92795389	0.913793103
2970	0.87993283	0.925287356	0.913793103
3304	0.877957191	0.92795389	0.913793103