Εργαστηριακή Άσκηση για το μάθημα της Θεωρίας Αποφάσεων

Χειμερινό Εξάμηνο 2023-24

Θέμα:"Σχεδιασμός Ταξινομητή για Κατηγοριοποίηση και Πρόβλεψη Καρκίνου του Μαστού"

Σε αυτή την εργαστηριακή άσκηση καλείστε να χρησιμοποιήσετε ένα σύνολο δεδομένων για να εκπαιδεύσετε ένα σύνολο ταξινομητών για την κατηγοριοποίηση και πρόβλεψη του καρκίνου του μαστού.

Το σύνολο δεδομένων που σας δίνετε στο αρχείο BreastTissue.xls παρέχει ένα σύνολο βιοιατρικών μετρήσεων για κάθε ασθενή καθώς και την πληροφορία για τον αν ο ιστός του μαστού είναι Καρκίνωμα (Carcinoma), Ινο-αδένωμα (Fibro-adenoma), Μαστοπάθεια (Mastopathy), Αδενικός (Glandular), Συνδετικός (Connective), Λιπώδης (Adipose). Κάθε γραμμή στο αρχείο αυτό περιέχει πληροφορία για διαφορετικό ασθενή. Η πρώτη στήλη αφορά τον κωδικό της κάθε καταγραφής ενώ η δεύτερη (status) δείχνει τη διάγνωση, δηλ. αν η καταγραφή αφορά ασθενή με: Καρκίνωμα (Carcinoma), Ινο-αδένωμα (Fibro-adenoma), Μαστοπάθεια (Mastopathy), Αδενικός (Glandular), Συνδετικός (Connective), Λιπώδης (Adipose).

Car	Carcinoma
Fad	Fibro-adenoma
Mas	Mastopathy
Gla	Glandular
Con	Connective
Adi	Adipose

Όλες οι άλλες στήλες αντιστοιχούν σε βιοιατρικές μετρήσεις που θα πρέπει να χρησιμοποιήσετε σαν εισόδους στους ταξινομητές που θα δημιουργήσετε. Για περισσότερες πληροφορίες για τα δεδομένα αυτά μπορείτε να δείτε την αναλυτική περιγραφή τους στην βάση δεδομένων μηχανικής μάθησης UCI (https://archive.ics.uci.edu/dataset/192/breast+tissue) από την οποία προέρχονται. Σε αυτό το link, θα βρείτε μια σύντομη περιγραφή του αρχείου (χαρακτηριστικά, πλήθος, κλπ.)

Ερώτημα 1. Προεπεξεργασία δεδομένων (10 μόρια)

Το εύρος τιμών των δεδομένων που σας έχουν δοθεί διαφέρει σημαντικά ανά χαρακτηριστικό. Για αυτό τον λόγο, για να μην υπερεκτιμηθεί η συνεισφορά κάποιου χαρακτηριστικού έναντι άλλων, θα πρέπει πριν την επεξεργασία των χαρακτηριστικών εισόδου να κανονικοποιηθούν στο εύρος [-1,1]. Χρησιμοποιήστε το matlab (ή όποιο

περιβάλλον προγραμματισμού επιθυμείτε) τόσο για το διάβασμα του αρχείου που σας δίνεται όσο και για την κανονικοποίηση των δεδομένων εισόδου στο εύρος τιμών [-1,1]. Ακόμη θα πρέπει να αντιστοιχίσετε την κλάση του ασθενούς σε μία τιμή. Για αυτό το λόγο να αντιστοιχίσετε κάθε ασθενή, στην κατηγορία που ανήκει, σύμφωνα με τον παρακάτω πίνακα.

Car	Carcinoma	1
Fad	Fibro-adenoma	2
Mas	Mastopathy	3
Gla	Glandular	4
Con	Connective	5
Adi	Adipose	6

Η στήλη αυτή δεν χρειάζεται κανονικοποίηση.

Απάντηση:

Για την υλοποίηση του ερωτήματος 1 χρησιμοποίησα την γλώσσα **python** και τις εξής βιβλιοθήκες: **Pandas** για το διάβασμα του αρχείου των δεδομένων, την αποθήκευση τους σε πίνακες και την αντιστοίχιση των κλάσεων των ασθενών σε τιμές. Ακόμη χρησιμοποίησα την **scikit-learn(sklearn)** για την κανονικοποίηση των αριθμητικών δεδομένων στο διάστημα [-1,+1].

Αρχικά φορτώνω τα δεδομένα του αρχείου **BreastTissue.xlsx** και τα φορτώνω στον πίνακα δεδομένων(dataframe για την python) μέσω της **read_excel()** της pandas. Στη συνέχεια με τη χρήση της βιβλιοθήκης **sklearn.preprocessing** και μέσω των συναρτήσεων της : **MinMaxScaler()** και **fit_transform()** κανονικοποιώ τις αριθμητικές στήλες στο εύρος [-1,1].

Τέλος με τη χρήση της βιβλιοθήκης pandas και της συνάρτησης της **map()** αντιστοιχίζω τις κλάσεις των ασθενών στις τιμές 1,2,3,4,5,6.

Στιγμιότυπα εκτέλεσης:

Αρχικά πριν την κανονικοποίηση και αντιστοίχιση των κλάσεων:

	Case #	Class	I0	...	Max IP	DR	P
0	1	car	524.794072	...	60.204880	220.737212	556.828334
1	2	car	330.000000	...	69.717361	99.084964	400.225776
2	3	car	551.879287	...	77.793297	253.785300	656.769449
3	4	car	380.000000	...	88.758446	105.198568	493.701814
4	5	car	362.831266	...	69.389389	103.866552	424.796503
..	...	...	...	...	...	...	...
101	102	adi	2000.000000	...	204.090347	478.517223	2088.648870
102	103	adi	2600.000000	...	418.687286	977.552367	2664.583623
103	104	adi	1600.000000	...	103.732704	432.129749	1475.371534
104	105	adi	2300.000000	...	178.691742	49.593290	2480.592151
105	106	adi	2600.000000	...	154.122604	729.368395	2545.419744

Έπειτα από την αφαίρεση της στήλης: Case # η οποία δεν προσφέρει κάποια πληροφορία πέρα από απλή απαρίθμηση:

	Class	I0	PA500	...	Max IP	DR	P
0	car	524.794072	0.187448	...	60.204880	220.737212	556.828334
1	car	330.000000	0.226893	...	69.717361	99.084964	400.225776
2	car	551.879287	0.232478	...	77.793297	253.785300	656.769449
3	car	380.000000	0.240855	...	88.758446	105.198568	493.701814
4	car	362.831266	0.200713	...	69.389389	103.866552	424.796503
..	...	...	...	...	...	...	...
101	adi	2000.000000	0.106989	...	204.090347	478.517223	2088.648870
102	adi	2600.000000	0.200538	...	418.687286	977.552367	2664.583623
103	adi	1600.000000	0.071908	...	103.732704	432.129749	1475.371534
104	adi	2300.000000	0.045029	...	178.691742	49.593290	2480.592151
105	adi	2600.000000	0.069988	...	154.122604	729.368395	2545.419744

Και τέλος έπειτα από την κανονικοποίηση των αριθμητικών και την κωδικοποιήση των κατηγορικών δεδομένων:

----------After encoding categorical attributes :

	Class	I0	PA500	...	Max IP	DR	P
0	1	-0.687212	0.012109	...	-0.755981	-0.533862	-0.688376
1	1	-0.831665	0.240161	...	-0.711543	-0.780418	-0.801381
2	1	-0.667127	0.272452	...	-0.673817	-0.466882	-0.616258
3	1	-0.794587	0.320888	...	-0.622594	-0.768028	-0.733928
4	1	-0.807318	0.088799	...	-0.713076	-0.770727	-0.783650
..	...	...	...	...	...	...	...
101	6	0.406748	-0.453078	...	-0.083824	-0.011411	0.416992
102	6	0.851687	0.087790	...	0.918659	1.000000	0.832589
103	6	0.110122	-0.655903	...	-0.552642	-0.105426	-0.025551
104	6	0.629218	-0.811302	...	-0.202473	-0.880725	0.699820
105	6	0.851687	-0.667003	...	-0.317247	0.496997	0.746600

Ερώτημα 2. (20 μόρια) Να κάνετε μια σύντομη παρουσίαση του ταξινομητή Support Vector Machine και του Naive Bayes ταξινομητή. Να αντλήσετε υλικό από το διαδίκτυο και να αναφέρετε τις πηγές σας. Να κάνετε μια σύντομη σύγκριση των παραπάνω ταξινομητών με τον KNN ταξινομητή.

Απάντηση:

Support Vector Machine (Ταξινομητής)

Ο ταξινομητής με τη χρήση μηχανών διανυσμάτων υποστήριξης (support vector machine classifier) είναι ένας αλγόριθμος επιβλεπόμενης μάθησης(αφού μιλάμε για ταξινόμηση). Πρόκειται για αλγόριθμο ο οποίος δουλεύει καλά επάνω σε πολυδιάστατα δεδομένα , μπορεί να χειριστεί μη γραμμικές συσχετίσεις μεταξύ των δεδομένων. Με τη χρήση του 'κόλπου του πυρήνα' ο αλγόριθμος από **γραμμικός** ταξινομητής γίνεται μη γραμμικός . Πρόκειται για έναν ευέλικτο αλγόριθμο ταξινόμησης ως προς την συνάρτηση πυρήνα που θα χρησιμοποιήσει(ενδειχκτικά υπάρχουν γραμμικές, πολυωνυμικές, η rbf, η σιγμοειδής και άλλες πολλές συναρτήσεις πυρήνα). Ο στόχος του αλγορίθμου είναι να βρεί το βέλτιστο υπερ-επίπεδο σε έναν N-διάστατο χώρο όπου N ο αριθμός των χαρακτηριστικών του συνόλου δεδομένων(των ανεξάρτητων δηλαδή μεταβλητών), το οποίο ταξινομεί/χωρίζει τα σημεία των δεδομένων(των χαρακτηριστικών) στις τιμές των κλάσεων που υπάρχουν. Το υπερ-επίπεδο αυτό ορίζει ότι το περιθώριο μεταξύ των κοντινότερων σημείων που ανήκουν σε διαφορετική κλάση είναι όσο μεγαλύτερο γίνεται. Το περιθώριο είναι το εύρος κατά το οποίο το όριο θα μπορούσε να αυξηθεί πριν "χτυπήσει" ένα σημείο δεδομένων στον N-διάστατο χώρο. Αποτελεί έναν χρησιμότατο σε πραγματικές εφαρμογές αλγόριθμο ταξινόμησης όπως την ταξινόμηση εικόνων, κειμένων και την αναγνώριση χειρόγραφων χαρακτήρων.

Βιβλιογραφία:

1) SVMs 1
2) SVMs 2
3) Support Vector Machine Wiki

Naïve Bayes(Ταξινομητής)

Ο αφελής ταξινομητής Bayes πρόκειται για έναν αλγόριθμο **επιβλεπόμενης** μάθησης(αφού μιλάμε για ταξινόμηση) που βασίζεται στο θεώρημα του Bayes(πιθανοτικός ταξινομητής). Πρόκειται για έναν απλό, γρήγορο αλγόριθμο μηχανικής μάθησης. Το βασικό μειονέκτημα του είναι ότι υποθέτει ότι όλα τα χαρακτηριστικά είναι ανεξάρτητα και ασυσχέτιστα επομένως δεν μπορεί να μάθει από τη συσχέτιση μεταξύ των χαρακτηριστικών.. Το θεώρημα του Bayes υπολογίζει την πιθανότητα μίας υπόθεσης με εκ των προτέρων γνώση. Στην περίπτωση που χειριζόμαστε ένα σύνολο δεδομένων η βασική υπόθεση του αφελή μπευζιανού ταξινομητή ορίζει ότι:

Τα χαρακτηριστικά είναι ανεξάρτητα μεταξύ τους και συνεισφέρουν το ίδιο στον καθορισμό της κλάσης εξόδου(της απόφασης ή της τιμής της εξαρτώμενης μεταβλητής διαφορετικά).

Η πιθανότητα δίνεται από τον τύπο:

$P(A|B)=(P(B|A)*P(A))/P(B)$.

Όπου P(A|B) η εκ των υστέρων πιθανότητα του B(αφού έχει συμβεί δηλαδή το B) και P(A) η εκ των προτέρων πιθανότητα του A(πριν δηλαδή συμβεί). Σε ένα πρακτικό σενάριο/σύνολο δεδομένων θα είχαμε ένα διάνυσμα X=(x1,x2,…,xn) των ανεξάρτητων χαρακτηριστικών και μία κλάση εξόδου y η εξαρτημένη μεταβλητή που θέλουμε να προβλέψουμε. Στην ουσία θέλουμε να βρούμε την ακόλουθη πιθανότητα:

P(y|X). Αυτή από τον κανόνα του Bayes θα ισχύει: P(y|X) = (P(X|y)*P(y))/P(X)=(P(x1,x2,…,xn|y)*P(y))/P(x1,x2,…,xn). Η **αφελής υπόθεση**(ισχυρή ανεξαρτησία των χαρακτηριστικών μεταξύ τους) ορίζει ότι: P(x1,x2,…,xn)=P(x1)*P(x2)*…*P(xn) και = P(x1,x2,…,xn|y)=P(x1|y)*P(x2|y)*…*P(xn|y). Αφού ο παρονομαστής αποτελείται μόνο από την ανεξάρτητη μεταβλητή-διάνυσμα X μας ενδιαφέρουν οι τιμές του y οι οποίες μεγιστοποιούν τον αριθμητή και επομένως την πιθανότητα P(y|X). Αναλυτικό παράδειγμα παρατίθεται στην **βιβλιογραφία** παρακάτω:

1) Naive Bayes 1
2) Naive Bayes 2
3) Naive Bayes Wiki

Σύγκριση ταξινομητών με τον k-NN ταξινομητή

Naïve Bayes και k-NN

Ο αφελής Bayes ταξινομητής είναι ένας γραμμικός ταξινομητής ενώ ο k-NN δεν είναι. Είναι αποδοτικότερος σε μεγάλο όγκο δεδομένων (πολλές πλειάδες δεδομένων δηλαδή) και ταχύτερος σε αυτήν την εφαρμογή. Επομένως είναι γρηγορότερος του k-NN. Ο αφελής Bayes διαθέτει δύο υπερ-παραμέτρους τις α και β ενώ ο k-NN έχει ως υπερ-παράμετρο μόνο το k δηλαδή των αριθμό των πλησιέστερων γειτόνων. Ο αφελής Bayes δεν επηρεάζεται από την ‘κατάρα της διαστατικότητας’ δηλαδή από δεδομένα αρκετών διαστάσεων(χαρακτηριστικών) ενώ η απόδοση του k-NN επηρεάζεται αρκετά αρνητικά όταν το σύνολο δεδομένων έχει μεγάλο αριθμό χαρακτηριστικών(συνήθως πάνω από 10). Σε αυτή τη περίπτωση εφαρμόζονται αρχικά επάνω στα δεδομένα τεχνικές μείωσης της διαστατικότητας(π.χ PCA κ.ο.κ). Ο k-NN δεν χρειάζεται να εκπαιδευτεί επάνω στα δεδομένα ενώ το μοντέλο του αφελή Bayes χρειάζεται εκπαίδευση. Γενικά ο k-NN είναι προτιμότερος σε μικρά σύνολα δεδομένων λίγων χαρακτηριστικών και σε περιπτώσεις που δεν πρέπει να κάνουμε κάποια υπόθεση για τη σχέση των δεδομένων μεταξύ τους(ανεξαρτησία, κατανομή κλπ). Ο αφελής Bayes είναι προτιμότερος σε μεγάλα σύνολα δεδομένων με μεγάλο αριθμό χαρακτηριστικών και σε δεδομένα για των οποίων τα χαρακτηριστικά υποθέτουμε ανεξαρτησία μεταξύ τους ή για τα οποία γνωρίζουμε την κατανομή(λ.χ κανονική κατανομή).

Βιβλιογραφία:

k-NN vs Naive Bayes

SVMs και k-NN

Ο ταξινομητής με τη χρήση διανυσμάτων υποστήριξης είναι πιο εύρωστος στη διαχείριση των ακραίων τιμών(outliers) σε σύγκριση με τον ταξινομητή k-NN. Από την άλλη ο ταξινομητής k-NN είναι καλύτερος από τον SVM όταν ο αριθμός των δεδομένων(των πλειάδων δεδομένων) είναι πολύ μεγαλύτερος από τον αριθμό των χαρακτηριστικών. Έτσι ο ταξινομητής SVM είναι καλύτερος σε λίγα δεδομένα(λίγες πλειάδες δεδομένων) μεγάλου αριθμού χαρακτηριστικών.

Βιβλιογραφία:

k-NN vs SVM

Ερώτημα 3. (30 μόρια) Με χρήση της μεθόδου 5-fold cross validation και του matlab (ή όποιο περιβάλλον προγραμματισμού επιθυμείτε), εκπαιδεύστε τους παρακάτω τρεις ταξινομητές και παρουσιάστε την απόδοσή τους:

- Support Vector Machine (με Radial Basis Function σαν kernel function):
    - Ρυθμίστε την παράμετρο C με διαδοχική αναζήτηση του βέλτιστου C στο διάστημα 1-200 με βήμα 5 και χρήση γραμμικών SVM. Στη συνέχεια, ρυθμίστε την παράμετρο γ με χρήση του βέλτιστου C που βρέθηκε από πριν, και διαδοχική αναζήτηση του βέλτιστου γ στο διάστημα 0-10 με βήμα 0.5 και χρήση RBFSVM.
- Ταξινομητής KNN-K Κοντινότερου Γείτονα
    - Ρυθμίστε την παράμετρο K με διαδοχική αναζήτηση της βέλτιστης τιμής στο διάστημα 3-15.
- Αφελής Μπεϋζιανός (Naive Bayes) Ταξινομητής

Παρουσιάστε για κάθε ταξινομητή την μέση απόδοσή του με χρήση 5 fold cross validation σε σχέση με την μετρική του γεωμετρικού μέσου (Geometric Mean) της ευαισθησίας (Sensitivity) και της ειδίκευσής (Specificity) του:

Geometric Mean = sqrt (Sensitivity * Specificity)

Η μετρική αυτή χρησιμοποιείται για προβλήματα ταξινόμησης όπου παραδείγματα εκπαίδευσης της μίας κλάσης είναι περισσότερα από τα παραδείγματα εκπαίδευσης της άλλης κλάσης.

Στη συνέχεια, παρουσιάστε τα ενδιάμεσα αποτελέσματα που πήρατε από τα πειράματα για την ρύθμιση των παραμέτρων των αλγορίθμων. Γιατί η κάθε μέθοδος ταξινόμησης δίνει διαφορετικά αποτέλεσματα; Ποιά μέθοδο προτείνετε εσείς και γιατί;

Απάντηση:

Για την υλοποίηση του ερωτήματος 3 χρησιμοποίηθηκε η γλώσσα **python** (έκδοση 3.10) με τσι εξής βιβλιοθήκες: **pandas, sklearn.preprocessing** για το διάβασμα των δεδομένων από το αρχείο και την προ-επεξεργασία τους στο ερώτημα 1. Από την **sklearn** χρησιμοποιώ τις υπο-βιβλιοθήκες **svm**, **naïve_bayes**, **neighbors** για να υλοποιήσω τους ζητούμενους ταξινομητές. Από την **sklearn** χρησιμοποιώ ακόμη τις υπο-βιβλιοθήκες **metrics** και **model_selection** για την κατασκευή της μετρικής του γεωμετρικού μέσου και την ρύθμιση των υπερ-παραμέτρων των ταξινομητών αντίστοιχα(και κατά συνέπεια την εύρεση των βέλτιστων τιμών τους). Αρχικά έχοντας προ-επεξεργαστεί τα δεδομένα όπως ζητείται στο ερώτημα 1 περνάω ως παραμέτρους στην συνάρτηση **GridSearchCV**(Grid search cross validation) το είδος του ταξινομητή, μετά ένα dictionary με τις υπερ-παραμέτρους που θέλω να ελέγξω, έπειτα το είδος της μετρικής πάνω στην οποία θα γίνει η αξιολόγηση του μοντέλου και τέλος τον αριθμό των folds = 5. Η **GridSearchCV()** έχει το πλεονέκτημα ότι επιστρέφει αμέσως **τη μέση απόδοση του μοντέλου για τα 5 folds** χωρίς να υπάρχει ανάγκη για περισσότερες πράξεις. Συγκεκριμένα για την κατασκευή της μετρικής του γεωμετρικού μέσου, έχω χρησιμοποιήσει τη συνάρτηση **make_scorer()** και **geometric_mean_score()**. Στη συνέχεια 'ενσωματώνω' στο μοντέλο του **GridSearchCV** τα δεδομένα όπως τα είχα διασπάσει προηγουμένως σε χαρακτηριστικά και μεταβλητή κλάσης. Έπειτα επιλέγω τα πεδία**:** **best_score_** και **best_params_** , δηλαδή τις τιμές των υπερ-παραμέτρων για τις οποίες μεγιστοποιείται ο γεωμετρικός μέσος καθώς και την τιμή αυτού.

Η διαδικασία αυτή επαναλαμβάνεται αντιστοίχως για κάθε ταξινομητή ενώ για τον ταξινομητή SVM γίνεται δύο φορές, λόγω των περιορισμών που θέτει η άσκηση.

Στην παρακάτω εικόνα φαίνεται ένα στιγμιότυπο του κώδικα για την εύρεση των βέλτιστων υπερπαραμέτρων για τον ταξινομητή k-NN:

```
margin = range(3,16)
scores =[]
gm_scorer = make_scorer(geometric_mean_score, greater_is_better=True)
for k in margin:
    knn = KNeighborsClassifier(n_neighbors=k)
    score = cross_val_score(knn,X,y,cv=5,scoring=gm_scorer)#to score einai ena dianysma diastasis 5 diladi 5 times.
    scores.append(np.mean(score))#o mesos aytou tou dianysmatos(athrisma stixion/5) = apodosi. #vazo to skor tou ekastote k sth lista

plt.plot( *args: margin,scores,'x')#kai tin kano visualize
plt.xlabel('values of k')
plt.ylabel('Accuracy score')
plt.show()
print('5-Fold cross validation knn optimal k gmean:',100*max(scores))

classifier = KNeighborsClassifier()
parameters = [{'n_neighbors':margin}]
grid_search = GridSearchCV(estimator = classifier,param_grid = parameters,scoring = gm_scorer, cv = 5)
grid_search = grid_search.fit(X, y)
best_gmean = grid_search.best_score_
best_parameters = grid_search.best_params_
print('Best gmean:',100*best_gmean,' best parameters(k) : ',best_parameters)
```

Και οι δύο υλοποιήσεις , η μία μέσω της **GridSearchCV()** και η άλλη μέσω του βρόχου επανάληψης και τη χρήση της συνάρτησης **cross_val_score()** παράγουν τα ίδια αποτελέσματα.

Στην παρακάτω εικόνα φαίνεται η εκτέλεση των ταξινομητών με τις βέλτιστες τιμές των υπερ-παραμέτρων που μεγιστοποιούν την τιμή του γεωμετρικού μέσου:

```
[8 rows x 10 columns]
5-Fold cross validation knn optimal k gmean: 37.80982940578783
Best KNN gmean: 37.80982940578783  best parameters(k) :  {'n_neighbors': 4}
Best naive bayes gmean: 37.73809884179288  best parameters() :  {}
5-Fold cross validation naive bayes optimal gmean:  37.73809884179288
Best linear SVM gmean: 48.34597206646429  best parameters :  {'C': 66}
Best rbf SVM gmean: 38.80183048981409  best parameters :  {'gamma': 2.0}

Process finished with exit code 0
```

Παρατηρώ ότι η απόδοση του SVM και στα δύο σενάρια είναι καλύτερη από αυτή του k-NN η οποία με τη σειρά της είναι καλύτερη από τον naïve bayes.(SVM>K-NN>Naïve Bayes).

Ερώτημα 4 (10 μόρια). Να κάνετε μια σύντομη παρουσίαση της μεθόδου Student t-test. Να αντλήσετε υλικό από το διαδίκτυο και να αναφέρετε τις πηγές σας.

Απάντηση:

Η μέθοδος **Student t-test** αποτελεί μία μέθοδο φιλτραρίσματος με στόχο την επιλογή χαρακτηριστικών από ένα σύνολο δεδομένων έτσι ώστε να αυξηθεί η απόδοση κάποιου μοντέλου αλγορίθμου μηχανικής μάθησης. Είναι μία univariate μέθοδος δηλαδή μέθοδος που εξετάζει κάθε χαρακτηριστικό ξεχωριστά από τα υπόλοιπα. Με το **t-test** συγκρίνουμε δύο κατανομές μόνο , κοινώς τις τιμές που παίρνει το χαρακτηριστικό όταν ανήκει σε μία κλάση, με αυτές που παίρνει το χαρακτηριστικό όταν ανήκει στην άλλη κλάση. Οπότε αφορά προβλήματα δυαδικής κατηγοριοποίησης , δηλαδή προβλήματα στα οποία τα δεδομένα ταξινομούνται σε δύο μόνο κλάσεις. Ξεκινώντας με την 'κενή'(null-H_0) υπόθεση ότι δεν υπάρχει ιδιαίτερη σχέση μεταξύ του χαρακτηριστικού και της μεταβλητής -στόχου(οι κατανομές πιθανότητας των χαρακτηριστικών ταυτίζονται), προσπαθούμε να βρούμε χαρακτηριστικά που αντιτίθονται στην κενή υπόθεση και ακολουθούν την εναλλακτική υπόθεση(H_1) ότι δηλαδή υπάρχει σημαντική συσχέτιση μεταξύ του χαρακτηριστικού και της μεταβλητής-στόχου(εξόδου), ουσιαστικά οι παραπάνω κατανομές διαφέρουν. Η διαδικασία επιλογής των χαρακτηστικών είναι η εξής:

Για κάθε χαρακτηριστικό του συνόλου δεδομένων υπολογίζεται μία τιμή **p** και μία τιμή **t** όπου **t** = $(x_1-x_2)/\sqrt{(\sigma_1^2/n + \sigma_2^2/n)}$,με x_1,x_2 να είναι οι μέσες τιμές των τιμών του χαρακτηριστικού που ανήκουν στη πρώτη και την δεύτερη κλάση αντίστοιχα, σ_1,σ_2 οι αντίστοιχες τυπικές αποκλίσεις και n ο αριθμός των δειγμάτων . Αυτή η τιμή του **t** αντιστοιχεί σε μία τιμή της **p**(t-table) , δηλαδή την πιθανότητα να ισχύει η H_0. Μία μικρή τιμή της **p**

(συνήθως κάτω από κάποιο κατώφλι της τάξης του 1-5%), πάει ενάντια στην αρχική (κενή) υπόθεση. Δηλαδή το χαρακτηριστικό συσχετίζεται αρκετά με την μεταβλητή κλάσης οπότε αξίζει να διατηρηθεί(να μην φιλτραριστεί). Αντίθετα μία τιμή **p** μεγαλύτερη από το προαναφερθέν κατώφλι υποστηρίζει την κενή υπόθεση και επομένως δεν υπάρχει ισχυρή συσχέτιση μεταξύ του χαρακτηριστικού και της μεταβλητής κλάσης επομένως αξίζει να φιλτραριστεί. Μία παραλλαγή του **t-test** είναι το **ANOVA test** με το οποίο μπορούμε να συγκρίνουμε περισσότερες κατανομές, κοινώς συγκρίνουμε τις τιμές που παίρνει το χαρακτηριστικό όταν ανήκει σε κάθε μία από τις πολλές κλάσεις του συνόλου δεδομένων(συνήθως πάνω από 3) με τις τιμές που παίρνει όταν ανήκει στις υπόλοιπες.

Τέλος η μέθοδος **t-test** όπως και οι άλλες μέθοδοι επιλογής χαρακτηριστικών λειτουργούν ανεξάρτητα κάποιου ταξινομητή.

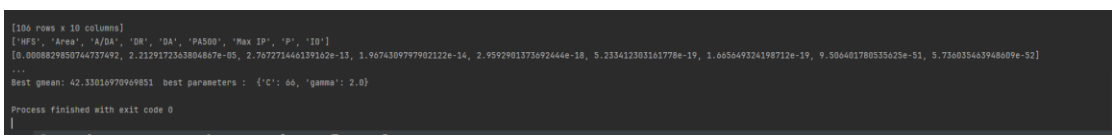
Βιβλιογραφία:

[Student t-test]

Ερώτημα 5. (30 μόρια) Για τα δεδομένα που σας δίνονται διατάξτε τα χαρακτηριστικά εισόδου με χρήση της μεθόδου student t-test με βάση την σημαντικότητά τους στην πρόβλεψη του καρκίνου του μαστού. Στη συνέχεια, κρατείστε τα 4 πιο σημαντικά χαρακτηριστικά, και επαναλάβετε την εκπαίδευση του βέλτιστου ταξινομητή που βρήκατε από προηγούμενο ερώτημα. Σχολιάστε την τελική απόδοση.

Απάντηση:

Για το ερώτημα 5 πέρα από τις βιβλιοθήκες που χρησιμοποίησα στα προηγούμενα ερωτήματα, χρησιμοποιώ την βιβλιοθήκη **scipy** και συγκεκριμένα την υπο-βιβλιοθήκη της **stats**. Από αυτή χρησιμοποιώ την συνάρτηση **f_oneway()** για να υλοποίησω το **ANOVA test** μία παραλλαγή του **t-test** μιας και τα δεδομένα που πραγματεύομαι δεν ανήκουν σε δύο κλάσεις αλλά 6. Οπότε δε μπορώ να εφαρμόσω άμεσα το **t-test**. Αφού πάλι προ-επεξεργαστώ όπως ζητείται τα δεδομένα για κάθε χαρακτηριστικό υπολογίζω μέσω της **f_oneway()** το **f** και **p** value του. Στη συνέχεια μέσω των συναρτήσεων **sorted()** και **zip()** διατάσσω τα χαρακτηριστικά βάσει του **p** value τους από το μεγαλύτερο στο μικρότερο. Όπως ζητείται κρατάω τα κορυφαία 4 **p** values και φιλτράρω τα υπόλοιπα. Στη συνέχεια μέσω του 5-fold cross validation και των τιμών των υπερ-παραμέτρων που βρήκα από το ερώτημα 3 (του βέλτιστου ταξινομητή) εφαρμόζω την ίδια διαδικασία μέσω της **GridSearchCV()** με μόνο τα 4 πιο σημαντικά χαρακτηριστικά αυτή τη φορά. Στην παρακάτω εικόνα φαίνεται η τελική απόδοση έπειτα από την επιλογή των 4 κορυφαίων χαρακτηριστικών:



Συγκριτικά με το ερώτημα 3 πριν την επιλογή των χαρακτηριστικών η απόδοση για τις ίδιες τιμές των υπερ-παραμέτρων C=66 και gamma=2.0 ήταν 38.8 ενώ τώρα έπειτα από το φιλτράρισμα 42.33. Αυτό συμβαίνει διότι απομακρύνοντας χαρακτηριστικά που δεν καθορίζουν την τιμή της μεταβλητής κλάσης σε μεγάλο βαθμό δίνεται περισσότερο βάρος σε αυτά που την καθορίζουν οπότε αναπόφευκτα θα αυξηθεί και η απόδοση οποιουδήποτε μοντέλου ταξινομητή χρησιμοποιήσουμε.

Παρακάτω φαίνεται και ένα στιγμιότυπο του κώδικα που υλοποίει το **ANOVA test**:

```
############
p_values = []
#ANOVA test because we have multiple classes
for feature in X.columns:
    #t_stat,p_value = stats.ttest_ind(X[feature][y == 1],X[feature][y == 2],X[feature][y == 3],X[feature][y == 4],X[feature][y == 5],X[feature][y == 6])
    f_stat,p_value = f_oneway( *samples: X[feature][y == 1],X[feature][y == 2],X[feature][y == 3],X[feature][y == 4],X[feature][y == 5],X[feature][y == 6])
    p_values.append(p_value)


alpha = 0.05#katofli an kai de xreiazetai epeidh apla diataso ta p values
sorted_features = [x for y, x in sorted(zip(p_values, X.columns),reverse=True)]
sorted_features2 = [y for y, x in sorted(zip(p_values, X.columns),reverse=True)]
print(sorted_features)
print(sorted_features2)

input('...')
gm_scorer = make_scorer(geometric_mean_score, greater_is_better=True)
X_new = df[['HFS','Area','A/DA','DR']]
y_new = df['Class']
classifier = SVC(kernel='rbf')
#parameters  = [{'C': range(1,201,5),'gamma':np.arange(0.0,11.0,0.5)}]
parameters = [{'C':[66],'gamma':[2.0]}]
grid_search = GridSearchCV(estimator=classifier,param_grid=parameters,scoring=gm_scorer, cv=5)
grid_search = grid_search.fit(X_new, y_new)
best_gmean = grid_search.best_score_
best_parameters = grid_search.best_params_
print('Best gmean:',100*best_gmean,' best parameters : ',best_parameters)
```

ΠΑΡΑΤΗΡΗΣΕΙΣ:

- Η εργασία αποτελείται από 2 μέρη. Στο 1^{ο} μέρος θα πρέπει να απαντήσετε τα ερωτήματα 1 και 2. Η αναφορά του 1^{ου} μέρους της εργασίας πρέπει να αναρτηθεί στη σελίδα του μαθήματος στο e-class, στις 17/12/2023 μέχρι τις 23.55.
- Η αναφορά του 2^{ου} μέρους της εργασίας πρέπει να αναρτηθεί στη σελίδα του μαθήματος στο e-class την παραμονή της εξέτασης του μαθήματος, μέχρι τις 23.55 (μετά από αυτή την ώρα το σύστημα θα κλείσει).
- Για την αναφορά της εργασίας σας, θα χρησιμοποιήσετε το αρχείο της εκφώνησης. Μετά από κάθε υποερώτημα θα έχετε την απάντησή σας, στην οποία θα περιγράφετε τη μεθοδολογία, τα εργαλεία που χρησιμοποιήσατε κλπ. Στα ερωτήματα που αναπτύσσετε κώδικα, θα πρέπει να έχετε κάποια screen shots από τα τρεξίματα και σε χωριστό αρχείο τον κώδικα. Το όνομα του αρχείου με τον κώδικα, θα πρέπει να περιέχει τον αριθμό του αντίστοιχου υποερωτηματος.
- Θα αναρτήσετε ένα .zip αρχείο που θα περιέχει την αναφορά και τον κώδικα. Στο όνομα του αρχείου πρέπει να υπάρχει το επώνυμό σας και το αρχικό γράμμα του ονόματός σας και ο ΑΜ.