ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΤΟΜΕΑΣ ΣΥΣΤΗΜΑΤΩΝ ΜΕΤΑΔΟΣΗΣ ΠΛΗΡΟΦΟΡΙΑΣ ΚΑΙ ΤΕΧΝΟΛΟΓΙΑΣ ΥΛΙΚΩΝ

«Ανάπτυξη αλγορίθμου για την αυτοματοποιημένη διάκριση όγκων μαστού σε καλοήθεις και κακοήθεις μέσω ανάλυσης των υπερηχογραφημάτων τους»

Εξαμηνιαία Εργασία

στο μάθημα **«Εργαστήριο Βιοϊατρικής Τεχνολογίας»**

των φοιτητών

Πανάγου Ζαχαρίας , Α.Μ.: 03119114
Προύντζος Παναγίωτης, Α.Μ.: 03119077
Μπούλιαρης Μαλατέστας Στέλιος, Α.Μ.: 03119214
Σταματόπουλος Νικόλαος, Α.Μ.: 03119020

Διδάσκοντες: Κ. Νικήτα, Δ. Κουτσούρης, Ο. Πετροπούλου

Αθήνα, Ιούνιος 2022

Η σελίδα αυτή είναι σκόπιμα λευκή.

Περίληψη

Ο καρκίνος του μαστού είναι ένας από τους πιο συνήθεις τύπους καρκίνου για τις γυναίκες. Το 2020 διαγνώστηκαν 2.300.000 γυναίκες με καρκίνο του μαστού, ενώ παράλληλα σημειώθηκαν 685.000 θάνατοι που οφείλονται σε αυτόν, παγκοσμίως. Όπως και με άλλους τύπους καρκίνου, η διάγνωση συνήθως γίνεται αρκετά αργά και ώς εκ τούτου η διαδικασία της θεραπείας γίνεται πολύ πιο δύσκολη. Αυτή η μορφή καρκίνου αφορά την αφύσικη ανάπτυξη των κυττάρων στους διαφόρους ιστούς του μαστού, με αποτέλεσμα την δημιουργία ενός όγκου ο οποίος επεκτείνεται πολύ γρήγορα. Παρ' όλα αυτά δεν είναι όλοι οι όγκοι επικίνδυνοι. Υπάρχουν 2 κατηγορίες όγκων, οι καλοήθεις και οι κακοήθεις. Οι καλοήθεις έχουν ένα χαρακτηριστικά στρογγυλό σχήμα το οποίο έχει ομαλή περιφέρεια, ενώ ταυτόχρονα αποτελείται από μη-καρκινικά κύτταρα. Αντίθετα, οι κακοήθεις όγκοι αποτελούνται από καρκινικά κύτταρα, με αποτέλεσμα να έχουν ένα πιο ακανόνιστο σχήμα και να επεκτείνονται πολύ πιο γρήγορα από τους καλοήθεις. Οι πιο τυπικές θεραπευτικές αγωγές περιλαμβάνουν την χημειοθεραπεία ή την χειρουργική επέμβαση, όμως όλες έχουν τις δικές τους παρενέργειες. Πριν από αυτές τις αγωγές όμως, προηγείται μία ακριβής διάγνωση, συνήθως με την χρήση ακτινογραφιών. Σκοπός της εργασίας αυτής, λοιπόν είναι η υλοποίηση ενός αλγορίθμου στην Python το οποίο, με τις μεθόδους της μηχανικής μάθησης, θα μπορεί να πραγματοποιεί μια ακριβή πρόβλεψη για τον τύπο του όγκου που απεικονίζεται σε μία ακτινογραφία.

Λέξεις Κλειδιά

Ταξινόμηση, Μηχανική Μάθηση, Καρκίνος του μαστού, Υπερηχογράφημα, Radiomics

Η σελίδα αυτή είναι σκόπιμα λευκή.

Πίνακας περιεχομένων

Η σελίδα αυτή είναι σκόπιμα λευκή.

1

Εισαγωγή

1.1 State of the art στα μοντέλα διάγνωσης καρκίνου του μαστού 1.1

Η μηχανική μάθηση χρησιμοποιείται αρκετά συχνά στην διάγνωση του καρκίνου. Ο λόγος για αυτό είναι ότι ένα μοντέλο μηχανικής μάθησης μπορεί να χειριστεί έναν πολύ μεγάλο όγκο δεδομένων σε ένα μικρό χρονικό διάστημα. Επομένως, έχει υπάρξει σημαντική πρόοδος σε αυτόν τον κλάδο της βιοϊατρικής μηχανικής στα τελευταία χρόνια.

1.1.1 Random Forest Classifier 1.1.1

Ένας συνηθισμένος classifier για το machine learning είναι ο random forest classifier. Στην εφαρμογή της κατηγοριοποίησης όγκων ο συγκεκριμένος classifier, έχει αποδειχθεί να δίνει ποσοστό AUC(area under the ROC curve) έως και 98.9%, σε συνδυασμό με μεθόδους super-resolution για την εξαγωγή των χαρακτηριστικών από μία ακτινογραφία [1].

1.1.2 Support Vector Machine Classifier 1.1.2

Ο support vector machine classifier είναι ένας ακόμα classifier ο οποίος χρησιμοποιείται ευρέως στον κλάδο της μηχανικής μάθησης. Σε ένα μοντέλο 5-fold

cross-validation αυτή η μέθοδος έχει αποδειχθεί να φτάνει accuracy μέχρι 95.24% με AUC 96.14%[2]. Επιπλέον, με την χρήση του μοντέλου SVM, το VGG16 deep learning feature extractor, και τα characteristic features έχει επιτευχθεί accuracy 92.5% [3].

1.1.3 Logistic Regression 1.1.3

Το Logistic Regression είναι μία διαδεδομένη μέθοδος για Classification όταν επιθυμούμε να κάνουμε κατηγοριοποίηση μεταξύ 2 κατηγοριών. Συγκεκριμένα, βασίζεται στην κατασκευή μίας μετρικής από ένα feature vector και στον μετασχηματισμό της μέσω της λογιστικης συνάρτησης. Ανάλογα με τον αριθμό που προκύπτει γίνεται και η κατηγοριοποίηση αυτού του Vector. Η συγκεκριμένη μέθοδος έχει Accuracy έως και 98.25% [4].

1.1.4 Feature Extraction 1.1.4

Για την σωστή εκπαίδευση του μοντέλου, είναι απαραίτητο να επιλεγούν τα πιο σημαντικά χαρακτηριστικά από ένα υπερηχογράφημα. Με την χρήση ενός Linear Support Vector Machine Classifier, απεδείχθη το accuracy στην κατηγοριοποίηση των όγκων είναι:

1. 82.69% Με την χρήση των Boundary features ενός όγκου
2. 73.08% Με την χρήση των χαρακτηριστικών σχήματος
3. 63.46% Με την χρήση των χαρακτηριστικών υφής
4. 86.54% Με την χρήση όλων των παραπάνω [5]

2

Υλικό και Μέθοδοι

2.1 Breast Ultrasound Images Dataset 2.1

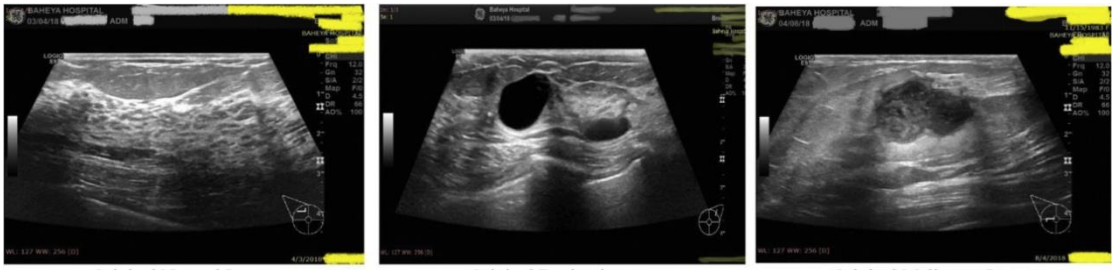
Το dataset συνίσταται από 1.581 εικόνες υπερηχογραφημάτων μαστού σε format PNG[6]. Τα δεδομένα συλλέχτηκαν από 600 γυναίκες ασθενείς μεταξύ των ηλικιών 25 με 75. Οι εικόνες κατηγοριοποιούνται σε φυσιολογικές (normal), οι οποίες αφορούν σε υγιείς μαστούς, και σε καλοήθεις (benign) και κακοήθεις (malignant), οι οποίες αναπαριστούν μαστούς με όγκους που έχουν ήδη χαρακτηριστεί. Για τις ανάγκες της εργασίας χρησιμοποιήσαμε τις δύο τελευταίες κατηγορίες. Κάθε δείγμα περιλαμβάνει μία πρωτότυπη grayscale εικόνα (780 στο σύνολο) με μέσο όρο μεγέθους 500*500 pixels (Εικόνα 2) και 1-2 μάσκες (ground truth images) για τον κάθε όγκο που έχει εντοπισθεί (Εικόνα 3).

Τα δεδομένα συλλέχτηκαν το 2018 στο νοσοκομείο Baheya της Γκίζας. Οι απεικονιστικές εξετάσεις έγιναν με τη χρήση των LOGIQ E9 και LOGIQ E9 Agile συστημάτων υπερήχων και παρήγαγαν αρχικά 1100 grayscale εικόνες ανάλυσης 1280*1024 (Εικόνα 1). Προκειμένου να γίνει εφικτή η εξόρυξη πληροφορίας από τις εικόνες, ακολουθήθηκε μία διαδικασία προεπεξεργασίας που περιλάμβανε την αφαίρεση διπλών εικόνων, την επανεξέταση της κατηγοριοποίησης από

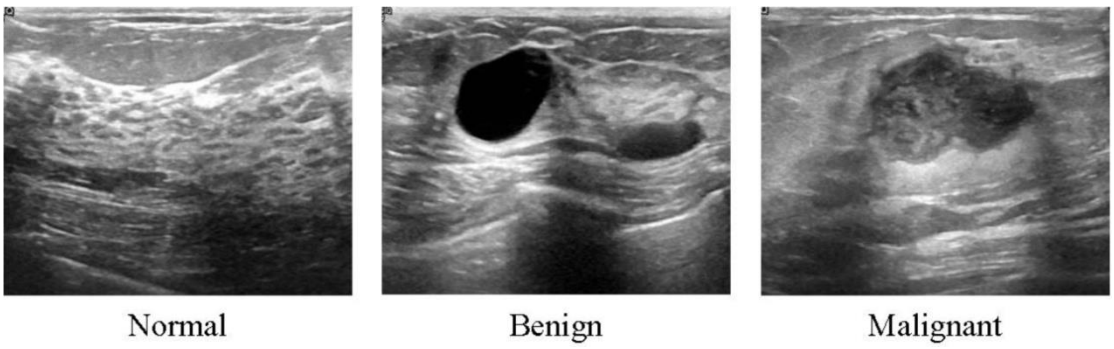
επαγγελματίες ραδιολόγους του νοσοκομείου και την μετατροπή των αρχείων από πρότυπο DICOM σε PNG. Στη συνέχεια μειώθηκαν τα όρια που παράγουν τα συστήματα υπερήχων ώστε να αφαιρεθούν άχρηστες πληροφορίες που θα επηρέαζαν αρνητικά διαδικασίες μηχανικής μάθησης, παράγοντας έτσι το τελικό σύνολο των 780 εικόνων.

Η τμηματοποίηση των εικόνων για την δημιουργία των μασκών έγινε “freehand”, δηλαδή χωρίς τη χρήση κάποιου αλγορίθμου αλλά με ελεύθερη σχεδίαση επάνω στις grayscale εικόνες, μέσω του Matlab. Η κατασκευή των μασκών ήταν απαραίτητη, καθώς επιτρέπει την εξαγωγή των μορφολογικών χαρακτηριστικών των όγκων για διαδικασίες όπως η εκτέλεση προγραμμάτων εποπτευόμενης ταξινόμησης.

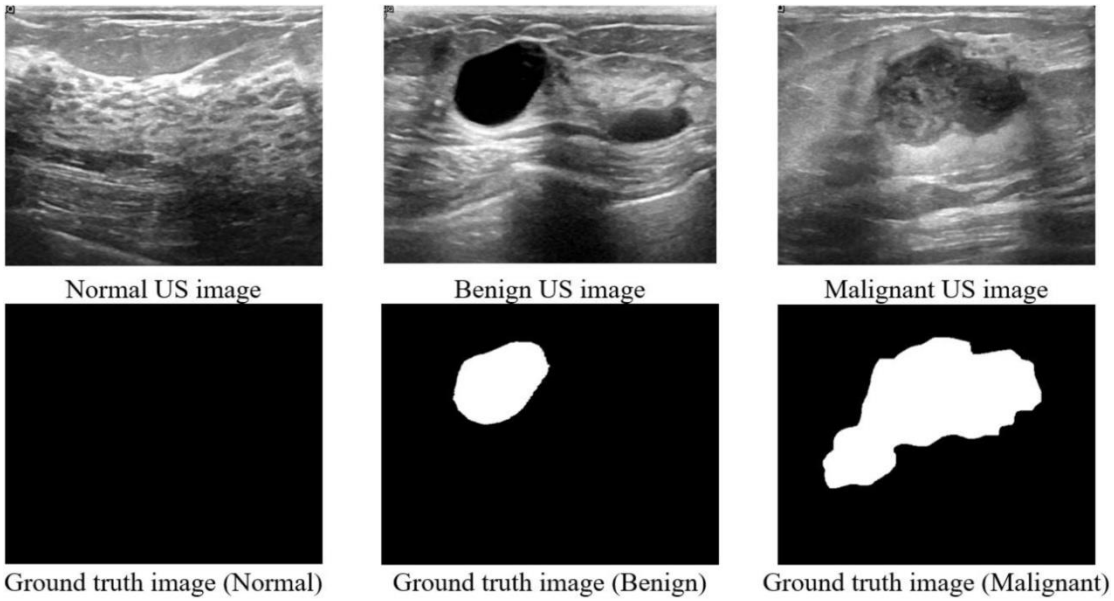
Η συλλογή των δεδομένων έγειρε προβληματισμούς ηθικής φύσεως αναφορικά με τη διαφύλαξη των προσωπικών ιατρικών δεδομένων των ασθενών. Προκειμένου να εξασφαλιστεί το ιατρικό απόρρητο, οι ερευνητές ενημέρωσαν τους ασθενείς για το πλαίσιο της έρευνας και έλαβαν τη συγκατάθεσή τους για τη χρήση των ιατρικών τους δεδομένων, ενώ επίσης παρέμειναν ανώνυμες οι εικόνες που προέκυψαν.



Εικόνα 1: Δείγματα πρωτότυπων εικόνων του LOGIQ E9 συστήματος υπερήχων



Εικόνα 2: Δείγματα grayscale υπερηχογραφημάτων από το dataset



Εικόνα 3: Δείγματα grayscale υπερηχογραφημάτων και των αντίστοιχων ground truth εικόνων

2.2 PyRadiomics 2.2

Για την εξαγωγή των χαρακτηριστικών από τις εικόνες του dataset χρησιμοποιήσαμε τη βιβλιοθήκη PyRadiomics της Python. Η open source βιβλιοθήκη φιλοξενείται στο GitHub[1] και κατασκευάστηκε από το πρόγραμμα Artificial Intelligence in Medicine του Πανεπιστημίου του Harvard με στόχο την εξαγωγή χαρακτηριστικών υφής και μορφολογίας από 2D και 3D ιατρικές εικόνες και δυαδικές μάσκες.

Το πακέτο των Radiomics περιλαμβάνει 8 κλάσεις χαρακτηριστικών, εκ των οποίων εμείς χρησιμοποιήσαμε τις First Order Statistics (19 χαρακτηριστικά) και Shape-based (2D) (10 χαρακτηρστικά).

Τα First Order Features χρησιμοποιούν συνήθεις βασικές παραμέτρους για να περιγράψουν την κατανομή της φωτεινότητας στη gray-scale εικόνα στην περιοχή ενδιαφέροντος που ορίζει η αντίστοιχη μάσκα. Για 2D εικόνες ορίζουμε:

- **X** το σύνολο από N_p pixels στην περιοχή ενδιαφέροντος
- **P(i)** το πρώτης τάξεως ιστόγραμμα με N_g διακριτά επίπεδα φωτεινότητας, το οποίο αναπαριστά το πλήθος των pixels για κάθε διάστημα gray-level τιμών
- **p(i)** το κανονικοποιημένο ιστόγραμμα πρώτης τάξεως που προκύπτει από τη μαθηματική σχέση $\frac{P(i)}{Np}$

Ακολουθεί πίνακας που περιγράφει τα διαφορετικά χαρακτηριστικά της κλάσης:

[1] https://github.com/AIM-Harvard/pyradiomics

Χαρακτηριστικό	Μαθηματικός Τύπος	Συνάρτηση Python	Περιγραφή
Energy	$\sum_{i=1}^{N_P}(\mathrm{X}(i)+\mathrm{c})^2$	getEnergyFeatureValue()	Περιγράφει το ύψος της έντασης των pixels στην εικόνα.
Total Energy	$\mathrm{A_{pixel}}\sum_{i=1}^{N_P}(\mathrm{X}(i)+1)^2$	getTotalEnergyFeatureValue()	Παριστάνει την ενέργεια αναλογικά με την επιφάνεια του pixel σε mm^2.
Entropy	$-\sum_{i=1}^{N_g} p(i)\log_2(p(i)+\varepsilon)$ $\varepsilon \sim 2.2 \times 10^{-16}$	getEntropyFeatureValue()	Αναπαριστά την τυχαιότητα των τιμών της εικόνας. Υπολογίζει τη μέση τιμή πληροφορίας που χρειάζεται για να κωδικοποιηθούν.
Minimum	min(X)	getMinimumFeatureValue()	Η ελάχιστη τιμή gray level έντασης στην περιοχή ενδιαφέροντος.
Maximum	max(X)	getMaximumFeatureValue()	Η μέγιστη τιμή gray level έντασης στην περιοχή ενδιαφέροντος.
10th percentile		get10PercentileFeatureValue()	Η τιμή του 10ου εκατοστημορίου.
90th percentile		get90PercentileFeatureValue()	Η τιμή του 90ου εκατοστημορίου.
Mean	$\frac{1}{N_P}\sum_{i=1}^{N_P}\mathrm{X(i)}$	getMeanFeatureValue()	Η μέση τιμή της gray level έντασης εντός της περιοχής ενδιαφέροντος.
Median		getMedianFeatureValue()	Η ενδιάμεση τιμή της gray level έντασης εντός της περιοχής ενδιαφέροντος.
Interquartile Range	$\mathrm{P_{75}} - \mathrm{P_{25}}$	getInterquartileRangeValue()	Το διατεταρτημοριακό διάστημα, δηλαδή το διάστημα από το 25ο εκατοστημόριο και το 75ο εκατοστημόριο της διανομής.
Range	max(X) – min(X)	getRangeFeatureValue()	Το διάστημα τιμών στην περιοχή ενδιαφέροντος.
Mean Absolute Deviation (MAD)	$\frac{1}{N_P}\sum_{i=1}^{N_P}\lvert\mathrm{X(i)}-\overline{\mathrm{X}}\rvert$	getMeanAbsoluteDeviationFeatureValue()	Η μέση απόσταση όλων των τιμών φωτεινότητας από τη μέση τιμή φωτεινότητας του πίνακα τιμών.
Root Mean Squared (RMS)	$\sqrt{\frac{1}{N_P}\sum_{i=1}^{N_P}(\mathrm{X}(i)+c)^2}$	getRootMeanSquaredFeatureValue()	Η τετραγωνική ρίζα της μέσης τιμής όλων των τετραγωνισμένων τιμών έντασης. Αποτελεί ένα ακόμη μέτρο για το ύψος των τιμών.
Variance	$\frac{1}{N_P}\sum_{i=1}^{N_P}(\mathrm{X}(i)-\overline{\mathrm{X}})^2$	getVarianceFeatureValue()	Η συνολική απόκλιση από τη μέση τιμή.
Standard Deviation	$\sqrt{\frac{1}{N_P}\sum_{i=1}^{N_P}(\mathrm{X}(i)-\overline{\mathrm{X}})^2}$	getStandardDeviationFeatureValue()	Η τετραγωνική ρίζα της απόκλισης.Αποτελεί επίσης μέτρο για την απόκλιση από την μέση τιμή.

Skewness	$\frac{\frac{1}{N_P}\sum_{i=1}^{N_P}(X(i)-\bar{X})^3}{\left(\sqrt{\frac{1}{N_P}\sum_{i=1}^{N_P}(X(i)-\bar{X})^2}\right)^3}$	getSkewnessFeatureValue()	Υπολογίζει την ασυμμετρία της κατανομής των τιμών έντασης γύρω από την μέση τιμή.
Kurtosis	$\frac{\frac{1}{N_P}\sum_{i=1}^{N_P}(X(i)-\bar{X})^4}{\left(\sqrt{\frac{1}{N_P}\sum_{i=1}^{N_P}(X(i)-\bar{X})^2}\right)^2}$	getKurtosisFeatureValue()	Αναπαριστά την περιοχή συγκέντρωσης των τιμών έντασης. Χαμηλή κυρτότητα σημαίνει ότι η πλειοψηφία των τιμών συγεντρώνεται γύρω από τη μέση τιμή, ενώ υψηλή κύρτωση σημαίνει ότι συγκεντρώνονται στις ουρές του ιστογράμματος.
1Uniformity	$\sum_{i=1}^{N_g} p(i)^2$	getUniformityFeatureValue()	Μετρά την ομογένεια της εικόνας. Μεγαλύτερη ομοιομορφία σημαίνει μικρότερο διάστημα διακριτών τιμών έντασης.

Πίνακας 1: Χαρακτηριστικά της κλάσης First Order Features της βιβλιοθήκης PyRadiomics

Τα 2D Shape Features περιλαμβάνουν περιγραφείς για το δισδιάστατο μέγεθος και σχήμα της περιοχής ενδιαφέροντος και εφαρμόζονται στις εικόνες-μάσκες. Για την εξαγωγή αυτών των χαρακτηριστικών κατασκευάζεται ένα πλέγμα από ευθείες που ενώνουν τα σημεία που βρίσκονται ανάμεσα σε pixels εντός και εκτός της περιοχής ενδιαφέροντος, δηλαδή τα άκρα της περιοχής. Ορίζουμε:

- N_P τον αριθμό των pixels εντός της περιοχής ενδιαφέροντος
- N_f τον αριθμό των γραμμών που ορίζουν το περιμετρικό πλέγμα
- A το εμφαδόν του πλέγματος σε mm^2
- p την περίμετρο του πλέγματος σε mm

Ακολουθεί πίνακας που περιγράφει τα διαφορετικά χαρακτηριστικά της κλάσης:

Χαρακτηριστικό	**Μαθηματικός Τύπος**	**Συνάρτηση Python**	**Περιγραφή**
Mesh Surface	$A_i = \frac{1}{2} Oa_i \times Ob_i$ (1) $A = \sum_{i=1}^{N_f} A_i$ (2) όπου Oa_i και Ob_i τα άκρα του i-οστού τριγώνου στο πλέγμα	getMeshSurfaceFeatureValue()	Υπολογίζει το συνολικό εμβαδό του πλέγματος, αθροίζοντας το εμβαδό όλων των τριγώνων.
Pixel Surface	$\sum_{k=1}^{N_U} A_k$	getPixelSurfaceFeatureValue()	Υπολογίζει το εμβαδό της περιοχής ενδιαφέροντας χωρίς να κάνει χρήση του πλέγματος, πολλαπλασιάζοντας τον αριθμό των pixels με το εμβαδό ενός pixel.
Perimeter	$Pi = \sqrt{(a_i - b_i)^2}$ (1) $P = \sum_{i=1}^{N_f} P_i$ (2)	getPerimeterFeatureValue()	Υπολογίζει την περίμετρο αθροίζοντας τις περιμέτρους κάθε υποπεριοχής.
Perimeter to Surface ratio	$\frac{P}{A}$	getPerimeterSurfaceRatioFeatureValue()	Η περίμετρος διά το εμβαδόν. Μικρότερη τιμή σημαίνει μεγαλύτερη κυκλικότητα.

Sphericity	$\frac{2\sqrt{\pi A}}{P}$	getSphericityFeatureValue()	Είναι το πηλίκο της περιμέτρου της περιοχής ενδιαφέροντος διά την περίμετρο ενός κύκλου με το ίδιο εμβαδόν. Όσο πιο κοντά στο 1 είναι η τιμή της, τόσο μεγαλύτερη κυκλικότητα έχουμε.
Spherical Disproportion	$\frac{P}{2\sqrt{\pi A}}$	getSphericalDisproportionFeatureValue()	Η αντίστροφη συνάρτηση της σφαιρικότητας.
Maximum 2D diameter		getMaximumDiameterFeatureValue()	Η μέγιστη Ευκλίδεια απόσταση ανάμεσα σε δύο σημεία του πλέγματος.
Major Axis Length	$4\sqrt{\lambda_{major}}$	getMajorAxisLengthFeatureValue()	Η μεγαλύτερη ευθεία που μπορεί να τραβηχτεί από ένα άκρο της περιοχής ενδιαφέροντος σε ένα αλλο.
Minor Axis Length	$4\sqrt{\lambda_{minor}}$	getMinorAxisLengthFeatureValue()	Η μικρότερη ευθεία που μπορεί να τραβηχτεί από ένα άκρο της περιοχής ενδιαφέροντος σε ένα αλλο.
Elongation	$\sqrt{\frac{\lambda_{minor}}{\lambda_{major}}}$	getElongationFeatureValue()	Δείχνει την σχέση ανάμεσα στους δύο ακραίους άξονες.

Πίνακας 2: Χαρακτηριστικά της κλάσης Shape-based (2D) Features της βιβλιοθήκης PyRadiomics

2.3 SciKit-Learn 2.3

Για την διαδικασία της μηχανικής μάθησης χρησιμοποιήσαμε τη βιβλιοθήκη Scikit-Learn της Python. Το project ξεκίνησε το 2007 από τον David Cournapeau στο πλαίσιο του Google Summer of Code και το 2010 επεκτάθηκε από ερευνητές του Εθνικού Ινστιτιούτου Έρευνας στην Πληροφορική και τον Αυτοματισμό της Γαλλίας (abr. INRIA). Περιλαμβάνει υλοποιήσεις για μία πληθώρα από αλγορίθμους επιβλεπόμενης και μη επιβλεπόμενης μηχανικής μάθησης και σκοπεύει στην ευρεία διάδοση εύχρηστων μεθόδων μηχανικής μάθησης[7].

Για τις ανάγκες της εργασίας τρέξαμε 6 διαφορετικούς αλγόριθμους ταξινόμησης προκειμένου να συγκρίνουμε τα αποτελέσματά τους. Το θεωρητικό υπόβαθρο του κάθε αλγορίθμου παρουσιάζεται εν συντομία παρακάτω.

2.3.1 Gaussian Naïve-Bayes 2.3.1

Έστω ότι έχουμε δύο κλάσεις ClassA και ClassB με χαρακτηριστικά {char1, char2, …} και επιθυμούμε να ταξινομήσουμε ένα Instance {x, y, …}. Η μέθοδος Gaussian Naïve-Bayes ακολουθεί την εξής διαδικασία:

1. υπολογίζει τη μέση τιμή και την τυπική απόκλιση για το κάθε χαρακτηριστικό της κάθε κλάσης και κατασκευάζει την κανονική κατανομή που αντιστοιχεί στις συγκεκριμένες τιμές,
2. εκτιμά, με βάση το dataset, μία αρχική πιθανότητα p(PriorClassA) και p(PriorClassB) να ανήκει το Instance σε καθεμία από τις κλάσεις,
3. υπολογίζει, σύμφωνα με τις κανονικές κατανομές, την εξαρτημένη πιθανότητα p(char1 = x | Instance ∈ ClassA) και p(char1 = x | Instance ∈ ClassB) για κάθε χαρακτηριστικό του Instance,
4. παράγει τις συνολικές πιθανότητες να ανήκει το Entity στην κάθε κλάση σύμφωνα με τους τύπους (1) και (2),
5. συγκρίνει τις p(ClassA) και p(ClassB) και ταξινομεί το Instance στην κλάση με την υψηλότερη τιμή.

Ακολουθούν μαθηματικοί τύποι:

$$p(ClassA) = log[p(PriorClassA] + \sum_{\substack{char \in \{char1, char2, \dots\} \\ n \in \{x, y, \dots\}}} \log(p(char = n \mid Instance \in ClassA)) \quad (1)$$

$$p(ClassB) = log[p(PriorClassB] + \sum_{\substack{char \in \{char1, char2, \dots\} \\ n \in \{x, y, \dots\}}} \log(p(char = n \mid Instance \in ClassB)) \quad (2)$$

2.3.2 Logistic Regression 2.3.2

Έστω ότι έχουμε δύο κλάσεις ClassA και ClassB και θέλουμε να ταξινομήσουμε ένα Instance με χαρακτηριστικά {x1, x2, …}. Η μέθοδος Logistic Regression ακολουθεί την εξής διαδικασία:

1. αντιστοιχίζει την ClassA στην τιμή 1 και την ClassB στην τιμή 0,
2. υπολογίζει την εξαρτημένη μεταβλητή $y = \beta 0 + \beta 1x1 + \beta 2x2 + \dots$ όπου
3. εφαρμόζει τη σιγμοειδή συνάρτηση $p = \frac{1}{1+e^{-y}}$ και
4. εάν p > 0,5 ταξινομεί το Instance στην ClassA, αλλιώς το ταξινομεί στην ClassB.

2.3.3 Decision Tree Classification 2.3.3

Η μέθοδος ταξινόμησης με δέντρο αποφάσεων κατασκευάζει ένα δυαδικό δέντρο το οποίο διαιρεί αναδρομικά το dataset. Η κατασκευή του δέντρου ακολουθεί την εξής διαδικασία:

1. εισάγεται στη ρίζα του δέντρου το σύνολο των ταξινομημένων δεδομένων,

2. ξεκινώντας από τη ρίζα, εισάγουμε conditions για τα χαρακτηριστικά των instances, παράγοντας δύο κόμβους-παιδιά που χωρίζουν τα instances σε αυτά που ικανοποιούν τα conditions και σε αυτά που δεν τα ικανοποιούν,
3. επαναλαμβάνεται η παραπάνω διαδικασία για κάθε κόμβο που περιέχει δεδομένα που ανήκουν και στις δύο κλάσεις, μέχρι να δημιουργηθούν κόμβοι-φύλλα που περιέχουν δεδομένα από μία μόνο κλάση.

Για την επιλογή των conditions γίνεται σύγκριση διαφορετικών διαιρέσεων των δεδομένων. Ο αλγόριθμος επιλέγει το condition που μεγιστοποιεί το κέρδος σε πληροφορία (IG), σύμφωνα με τη συνάρτηση Εντροπίας για κάθε κόμβο.

Ακολουθούν μαθηματικοί τύποι:

$$Entropy = \sum -p_i \log p_i\,, \qquad p_i = probability\ of\ class\ i \quad (3)$$

$$IG = Entropy(parent) - \sum w_i\, Entropy(child_i)\,, \qquad w_i = \frac{size\ of\ data\ in\ child}{size\ of\ data\ in\ parent} \quad (4)$$

Για την ταξινόμηση νέων Instances γίνεται εισαγωγή στη ρίζα του κατασκευασμένου δέντρου αποφάσεων, οπότε ακολουθείται ένα μονοπάτι από κόμβους απόφασης και ανάλογα με το φύλλο στο οποίο θα καταλήξει ταξινομείται στην αντίστοιχη κλάση.

2.3.4 Random Forest Classification 2.3.4

Ο αλγόριθμος Random Forest Classification αξιοποιεί την τεχνική bagging, δηλαδή το συνδυασμό πολλών μοντέλων εκμάθησης προκειμένου να επιτευχθεί ακριβέστερη ταξινόμηση.

Συγκεκριμένα, διαιρεί τα δεδομένα εκμάθησης σε υποσύνολα για να παράξει πολλαπλά, ασυσχέτιστα δέντρα αποφάσεων, εισάγει το νέο Instance σε κάθε ένα από τα δέντρα και το ταξινομεί στην κλάση που προέκυψε με υψηλότερη συχνότητα από τα δέντρα.

Η συγκεκριμένη μέθοδος έχει μεγαλύτερη ακρίβεια από την ταξινόμηση με μοναδικό δέντρο αποφάσεων, ωστόσο απαιτεί υψηλότερη υπολογιστική ισχύ και χρόνο.

2.3.5 k-Nearest Neighbours 2.3.5

Η ταξινόμηση κ-Κοντινότερου Γείτονα ακολουθεί την εξής διαδικασία:

1. υπολογίζει την απόσταση ενός Instance από όλα τα υπόλοιπα,

2. βρίσκει τα περιττά κ Instances με τη μικρότερη απόσταση και
3. ταξινομεί στην κλάση που ανήκει η πλειοψηφία από αυτά.

2.3.6 XGBClassifier 2.3.6

Το XGBClassifier ανήκει στην open-source βιβλιοθήκη XGBoost η οποία δημιουργήθηκε το 2014 και είναι γραμμένη σε C++ [2]. Οι αλγόριθμοι υλοποιούνται από Random Forest και Decision Trees μαζί με τεχνικές Boosting (boosted decision trees algorithm)[3]. Επιπλέον έχουν ενσωματομένους αλγόριθμους για Feature Importance και κανονικοποίηση των δεδομένων με στόχο να μειωθεί το Overfit των μοντέλων και να μεγιστοποιηθεί την απόδοση τους.

[2] Άρθο Wikipedia για το XGBoost : https://en.wikipedia.org/wiki/XGBoost

[3]Υλοποίηση του XGBoost : https://www.geeksforgeeks.org/xgboost/

3

Υλοποίηση με Python

3.1 Εξαγωγή χαρακτηριστικών με την PyRadiomics 3.1

Η διαδικασία της εκπαίδευσης ενός μοντέλου μηχανικής μάθησης με χρήση εικόνων απαιτεί την μετατροπή της κάθε εικόνας σε έναν κατάλληλο για επεξεργασία τύπο δεδομένων. Σκοπός, λοιπόν, αυτού του τμήματος κώδικα είναι να μετατρέψουμε κάθε ζεύγος υπερηχογραφήματος-μάσκας του Dataset σε ένα διάνυσμα το οποίο θα περιέχει όλες τις απαραίτητες πληροφορίες τις οποίες μπορούμε να εξάγουμε. Μας ενδιαφέρουν τα χαρακτηριστικά της κλάσης “Shape-2D”, δηλαδή αυτά που περιγράφουν την γεωμετρία του όγκου και τα χαρακτηριστικά της κλάσης “First Order Statistics” τα οποία περιγράφουν τα στατιστικά της κατανομής των Voxels στην εικόνα.

3.1.1 Η προσπέλαση του DataSet 3.1.1

Ξεκινώντας την μετατροπή του DataSet σε Vectors, φορτώσαμε τις εικόνες στο script μας. Οι εικόνες διαβάστηκαν με την μέθοδο imread της βιβλιοθήκης cv2, η οποία τις αποθηκεύει σε arrays.

```
radcategories=['firstorder','shape2D','glcm','glszm','glrlm','ngtdm','gldm']
path=r'C:\Users\swagp\Desktop\archive\Dataset_BUSI_with_GT'
categories=["benign","malignant"]
benignmask=[]
malignantmask=[]
benigngray=[]
malignantgray=[]
def read_data():
    for i in categories :
        address = os.path.join(path,i)
        tag=categories.index(i)
        listy=[]
        for j in os.listdir(address):
            img = cv2.imread(os.path.join(address,j),cv2.IMREAD_GRAYSCALE)
            img = sitk.GetImageFromArray(img)
            if 'mask' in j:
                if(not(tag)):
                    listy.append(img)
                else:
                    listy.append(img)

            else :
                if(not(tag) and len(listy)!=0):
                    benignmask.append(listy)
                    listy=[]
                elif len(listy)!=0:
                    malignantmask.append(listy)
                    listy=[]
                if(not(tag)):
                    benigngray.append(img)
                else:
                    malignantgray.append(img)
        if(not(tag)):
            benignmask.append(listy)
            listy=[]
        else:
            malignantmask.append(listy)
            listy=[]
```

Έπειτα μετατρέψαμε τις εικόνες του Dataset από arrays σε .nrrd. Αυτό έγινε με την μέθοδο SimpleITK.GetImageFromArray της βιβλιοθήκης SimpleITK, η οποία μπορεί να μετατρέψει ένα Array 2 διαστάσεων σε ένα αρχείο .nrrd. Αφού μετατράπηκαν οι εικόνες σε αυτό τον τύπο δεδομένων, αποθηκεύτηκαν σε τέσσερεις λίστες, ανάλογα με το εάν είναι καλοήθεις ή κακοήθεις οι όγκοι που αναπαριστούν και εάν είναι μάσκες ή υπερηχογραφήματα. Όπως φαίνεται στον κώδικα, οι λίστες benignmask, malignantmask, benigngray, malignantgray αρχικοποιούνται κενές. Στις λίστες που περιέχουν υπερηχογραφήματα (benigngray, malignantgray) αποθηκεύονται απευθείας τα αρχεία σε μορφή .nrrd, ενώ στις λίστες που περιέχουν τις μάσκες (benignmask, malignantmask) αποθηκεύονται νεές λίστες οι οποίες περιέχουν όλες τις μάσκες που αντιστοιχούν σε ένα υπερηχογράφημα. Αυτό γίνεται επειδή στο Dataset περιέχονται ορισμένα υπερηχογραφήματα τα οποία απεικονίζουν παραπάνω από έναν όγκο. Επομένως, το στοιχείο της λίστας benigngray με index i αντιστοιχεί στο i-οστό υπερηχογράφημα και το στοιχείο της λίστας benignmask με index i αντιστοιχεί σε μία λίστα που περιέχει όλες τις μάσκες που προκύπτουν από το i-οστό υπερηχογράφημα. Η παραπάνω διαδικασία εκτελείται καλώντας την μέθοδο read_data(), με δεδομένο το path του Dataset.

3.1.2 Εξαγωγή των χαρακτηριστικών από εικόνες 3.1.2

Έχοντας πλέον αποθηκεύσει αυτά τα αρχεία σε 4 λίστες, ακολουθεί η εξαγωγή των χαρακτηριστικών, η οποία πραγματοποιείται με την PyRadiomics, όπως φαίνεται στον παρακάτω κώδικα.

```
def extract_chars(im3,im4):
    imagedata = {}
    s2=[]
    f0=[]
    extractor=rad.featureextractor.RadiomicsFeatureExtractor()
    extractor.disableAllFeatures()
    extractor.enableFeatureClassByName("shape2D")
    extractor.enableFeatureClassByName("firstorder")
    result=extractor.execute(im3,im4,label=255)
    for featureName in result.keys():
            if(("shape2D" in featureName)):
                s2.append((featureName,float(result[featureName])))
            elif(("firstorder" in featureName)):
                f0.append((featureName,float(result[featureName])))
    imagedata["shape2D"] = s2
    imagedata["firstorder"] = f0
    return imagedata
```

Για αυτή την διαδικασία ορίσαμε την μέθοδο extract_chars() η οποία θα δέχεται ώς ορίσματα δύο μεταβλητές .nrrd (SimpleITK.Image), μία για το υπερηχογράφημα και μία για την μάσκα και επιστρέφει ένα dictionary με τα επιθυμητά χαρακτηριστικά. Η βιβλιοθήκη PyRadiomics περιέχει το "RadiomicsFeatureExtractor" object. Απο αυτό το object καθορίζονται τα χαρακτηριστικά τα οποία θα εξαχθούν. Επομένως, για αυτή την εφαρμογή χρειάστηκε αρχικά να απενεργοποιήσουμε όλες τις κλάσεις χαρακτηριστικών που είχαν ενεργοποιηθεί από τον ορισμό του extractor (με την μέθοδο extractor.disableAllFeatures()) και να ενεργοποιήσουμε όλα όσα μας ενδιαφέρουν, δηλαδή τα First Order Statistics και τα 2D Shape Characteristics (με την μέθοδο extractor.enableFeatureClassByName()). Ο Extractor έχει την μέθοδο execute(image,mask) η οποία επιστρέφει ένα dictionary με τα χαρακτηριστικά τα οποία έχουμε επιλέξει για δεδομένο ζεύγος υπερηχογραφήματος και μάσκας και με δεδομένα που αφορούν την εκτέλεση της μεθόδου, τα οποία δεν είναι χρήσιμα για αυτή την εφαρμογή. Όπως είχαμε αναφέρει προηγουμένως, ορισμένα υπερηχογραφήματα του Dataset απεικονίζουν παραπάνω από έναν όγκο. Η επεξεργασία σε αυτή την περίπτωση γίνεται ξεχωριστά για κάθε όγκο, δηλαδή εξάγονται τα χαρακτηριστικά για κάθε πιθανό ζεύγος υπερηχογραφήματος-μάσκας που προκύπτει από μία ακτινογραφία. Για όλα αυτά τα ζεύγη λοιπόν καλέσαμε αυτή την μέθοδο και συνεπώς, μετατρέψαμε το Dataset σε ένα πλήθος από dictionaries τα οποία έχουν όλες τις επιθυμητές πληροφορίες.

3.1.3 Μετατροπή δεδομένων από dictionaries σε lists και αποθήκευση 3.1.3

Τέλος, μετατρέψαμε το κάθε dictionary σε μια λίστα για ευκολία στην επεξεργασία και η διαδικασία εξαγωγής χαρακτηριστικών ολοκληρώθηκε.

```
def vectorize_dict(a):
    FeatureVector=[]
    for i in a.values():
        for j in i:
            FeatureVector.append(j[1])
    return FeatureVector
```

Αρκεί απλά για κάθε στοιχείο του dictionary να αποθηκεύσουμε σε ένα numpy array το Value, δηλαδή την τιμή του χαρακτηριστικού. Για περεταίρω διευκόλυνση

αποθηκεύσαμε αυτό το array στον υπολογιστή μας με την εντολή np.save() για να μην χρειάζεται να εκτελείται το συγκεκριμένο τμήμα κώδικα κάθε φορά που επιθυμούμε να εκπαιδεύσουμε κάποιο μοντέλο.

```
a=[]
for i in range(len(benignmask)):
    for j in range(len(benignmask[i])):
        a.append(vectorize_dict(extract_chars(benignngray[i], benignmask[i][j])))
```

```
b=[]
for i in range(len(malignantmask)):
    for j in range(len(malignantmask[i])):
        b.append(vectorize_dict(extract_chars(malignantgray[i], malignantmask[i][j])))
```

```
benign=np.array(a)
np.save(r'C:\Users\swagp\Desktop\archive\Dataset_BUSI_with_GT\vectors\benignvectors',benign)
```

```
malignant=np.array(b)
np.save(r'C:\Users\swagp\Desktop\archive\Dataset_BUSI_with_GT\vectors\malignantvectors',malignant)
```

Σε αυτό το σημείο του κώδικα γίνεται η μετατροπή του Dataset στα Vectors με τα χαρακτηριστικά, χρησιμοποιώντας τις προαναφερθείσες μεθόδους. Αξίζει να σημειωθεί ότι η βιβλιοθήκη PyRadiomics χρησιμοποιείται μόνο στις εκδόσεις 3.5, 3.6 και 3.7 της Python. Για αυτόν τον λόγο, ο κώδικας έχει χωριστεί σε 2 scripts, ένα για την εξαγωγή των δεδομένων και ένα για την εκπαίδευση των μοντέλων μηχανικής μάθησης χρησιμοποιώντας αυτά τα δεδομένα.

3.2 Εκπαίδευση Μοντέλων με skLearn 3.2

Στο κομμάτι ανάλυσης και επεξεργασίας δεδομένων αποφασίσαμε να χρησιμοποιήσουμε πολλά διαφορετικά classifiers για να δούμε ποιό είναι το καταλληλότερο για την συγκεκριμένη εφαρμογή. Συγκεκριμένα χρησιμοποιήσαμε τα παρακάτω Classifiers :

- Logistic Regression
- Gaussian Naïve Bayes
- Random Forest Classifier
- Decision Tree Classifier
- K nearest neighbours Classifier (KNN)
- XGB Classifier

Περιμένουμε ότι η κατηγοριοποίηση με Logistic Regression θα φέρει τα καλύτερα αποτελέσματα, καθώς το πρόβλημα που καλούμαστε να αντιμετωπίσουμε είναι δυαδικής κατηγοριοποίησης (ο ασθενής νοσεί ή δεν νοσεί από καρκίνο του μαστού.)

Αντιθέτως, αλγόριθμοι κατηγοριοποίησης όπως ο KNN δεν κρίνεται κατάλληλος για προβλήματα τέτοιου τύπου, καθώς δεν αποδίδει σε επιθυμητό βαθμό σε προβλήματα κατηγοριοποίησης υψηλών διαστάσεων.

Τέλος διαλέξαμε μερικούς από τους πιό διαδεδομένους αλγόριθμους κατηγοριοποιήσης, προκειμένου να συγκρίνουμε τα αποτελέσματα τους, καθώς και

τον XGBoost ο οποίος αποτελεί έναν αλγόριθμο τελευταίας τεχνολογίας με ενσωματομένο feature selection.

3.2.1 Επεξεργασία Δεδομένων& Αρχική Εκπαίδευση 3.2.1

Αρχικά, αφού έχουμε επιλέξει τους αλγορίθμους που πρόκειται να χρησιμοποιήσουμε, ξεκινάμε διαβάζοντας τα δεδομένα και μετατρέποντας τα σε ένα **Pandas Dataframe**. Κάθε στοιχείο του dataframe αντιστοιχεί σε ένα υπερηχογράφημα και περιέχει τα χαρακτηριστηκά που υπολογίστηκαν μέσω της Pyradiomics, καθώς και τον τύπο του όγκου (Η τιμή 1 αντιστοιχεί σε κακοήθη και η τιμή 0 σε καλοήθη).

Έπειτα, για να είμαστε σε θέση να ελέγξουμε την απόδοση του κάθε αλγορίθμου κρατάμε το 10% των δεδομένων μας ως ένα Holdout set, και με το υπόλοιπο 90% εκπαιδεύουμε τα μοντέλα.

Η εκπαίδευση έγινε με ένα 10 fold Stratified Cross Validation και τα αποτελέσματα αξιολογήθηκαν βάση της μέσης ακρίβειας (mean accuracy), της ευαισθησίας (sensitivity), και της τιμής AUC (Area under ROC).

Τα αποτελέσματα ήταν τα εξής:

```
GaussianNB:
   Test accuracy: 71.898% (±3.919%)
   Area under ROC: 77.689%
   Sensitivity: 36.553%
LogisticRegression:
   Test accuracy: 55.873% (±6.929%)
   Area under ROC: 55.63%
   Sensitivity: 51.658%
RandomForestClassifier:
   Test accuracy: 91.46% (±3.817%)
   Area under ROC: 96.26%
   Sensitivity: 78.921%
KNeighborsClassifier:
   Test accuracy: 71.585% (±4.775%)
   Area under ROC: 71.976%
   Sensitivity: 47.921%
DecisionTreeClassifier:
   Test accuracy: 90.802% (±3.897%)
   Area under ROC: 90.271%
   Sensitivity: 88.737%
XGBClassifier:
   Test accuracy: 96.494% (±2.405%)
   Area under ROC: 98.45%
   Sensitivity: 91.816%
```

Παρατηρούμε, ότι παρότι είχαμε εκτιμήσει πως το μοντέλο Logistic Regression θα είχε υψηλή απόδοση εδώ φαίνεται ότι στην πραγματικότητα έχει την χειρότερη καθώς κατάφερε να ανιχνεύσει μόνο το 51,66% των κακοήθων όγκων. Αντιθέτως, οι Random Forest, Decision Tree και XBG Classifiers είχαν την υψηλότερη απόδοση. Αυτό συμβαίνει διότι και οι τρείς αυτοί αλγόριθμοι είναι βασισμένοι σε Decision Trees όπου λόγω της λειτουργίας τους, αποδίδουν καλύτερα όταν εκπαιδεύονται σε δεδομένα μεγαλύτερων διαστάσεων.

Τα επιπλέον δεδομένα, παρότι οδηγούν σε καλύτερη απόδοση για τους τρείς αυτούς αλγορίθμους, λειτουργούν ως θόρυβος για τους υπόλοιπους classifiers. επομένως, με σκοπό να δοκιμάσουμε να βελτιώσουμε την απόδοση τους, υλοποιήσαμε έναν αλγόριθμο για feature selection.

3.2.2 Επιλογή Χαρακτηριστικών 3.2.2

Ο στόχος της διαδικασίας feature selection είναι να επιλέξουμε τα χαρακτηριστικά που περιέχουν πληροφορία ωφέλιμη στην κατηγοριοποίηση των όγκων. Η διαδικασία feature selection χωρίστηκε σε δύο στάδια:

Σύγκριση χαρακτηριστικών της ίδιας ομάδας όγκων:

Στο πρώτο στάδιο, για κάθε ένα από τα δύο είδη όγκων, υπολογίσαμε τη διασπορά (variance) κάθε χαρακτηριστικού. Έτσι μπορούμε να γνωρίζουμε ποια χαρακτηριστικά διατηρούν σταθερότερη την τιμή τους ανάλογα με το είδος του όγκου. Είναι προφανές ότι τα χαρακτηριστικά που παρουσιάζουν μεγάλη διασπορά δεν είναι ωφέλιμα για την κατηγοριοποίηση των όγκων, καθώς, οι τιμές τους ενδέχεται να διαφέρουν σημαντικά ακόμη και εντός της ίδιας κατηγορίας.

Σύγκριση χαρακτηριστικών μεταξύ των δύο ομάδων όγκων:

Γίνεται εύκολα αντιληπτό ότι τα χαρακτηριστικά με τα οποία επιλέγουμε να εκπαιδεύσουμε το μοντέλο μας πρέπει να καθιστούν εύκολη τη διάκριση μεταξύ των δυο κατηγοριών. Στο δεύτερο στάδιο λοιπόν, έπρεπε να ελέγξουμε αν οι τιμές των παραπάνω χαρακτηριστικών διαφέρουν ανά κατηγορία. Για κάθε ένα από τα δύο είδη όγκων υπολογίσαμε τη μέση τιμή κάθε χαρακτηριστικού και κρατήσαμε αυτά που παρουσίαζαν μεγάλη διαφορά μεταξύ τους.

Τα πέντε χαρακτηριστικά τα οποία επιλέξαμε να αποκλείσουμε από την εκπαίδευση των μοντέλων, μέσω της παραπάνω διαδικασίας, είναι τα εξής:

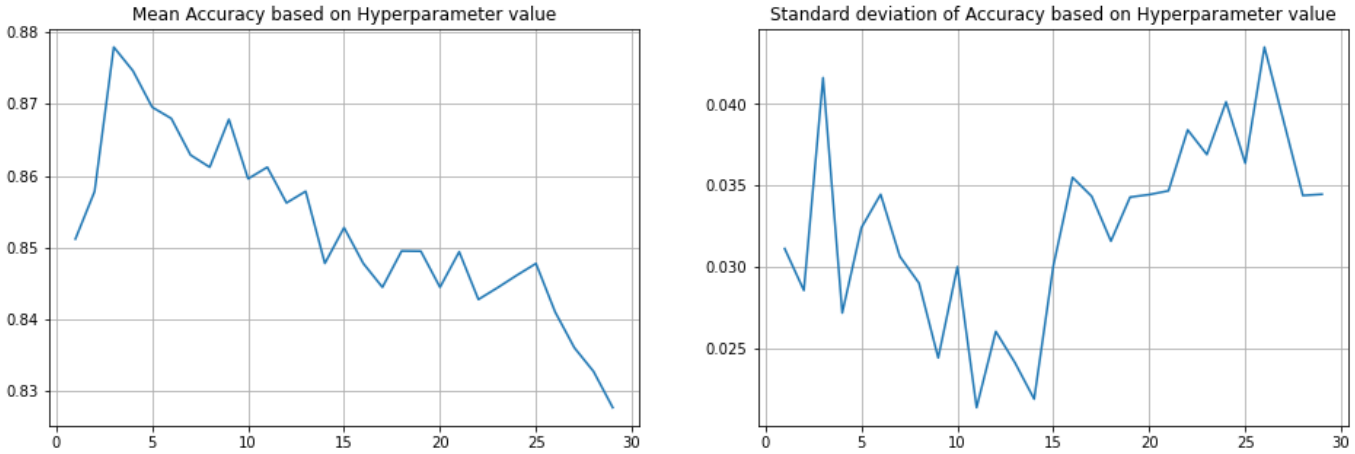
- *firstorder_Energy*
- *firstorder_TotalEnergy*
- *shape2D_ PerimeterSurfaceRatio*
- *firstorder_Uniformity*
- *firstorder_Variance*

Στη συνέχεια, προκειμένου να βεβαιωθούμε ότι η μέθοδος που ακολουθήσαμε είχε το επιθυμητό αποτέλεσμα, χρησιμοποιήσαμε τον αλγόριθμο Recursive Feature Elimination της SKlearn για τα τρία καλύτερα μοντέλα (RandomForest, DecisionTree, XGB), καθώς και για το Logistic Regression, το οποίο αναμένουμε να έχει την καλύτερη απόδοση μετά το feature selection. Κρατήσαμε τα χαρακτηριστικά τα οποία βελτιστοποιούσαν την διάκριση των όγκων για την πλειοψηφία των Classifiers. Με αυτή τη μέθοδο καταφέραμε να μειώσουμε τα χαρακτηριστικά κάθε υπερηχογραφήματος στα 15.

3.2.3 Βελτιστοποίηση Μοντέλων με τη χρήση HyperParameters 3.2.3

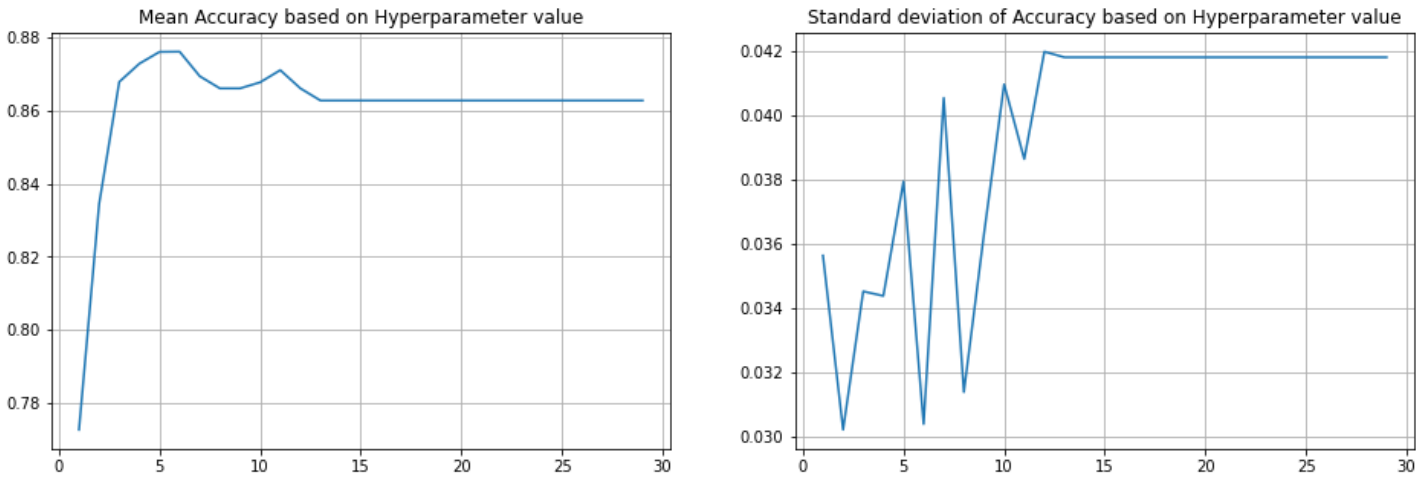
Τέλος, προκειμένου να μεγιστοποιήσουμε την απόδοση του κάθε μοντέλου, υλοποιήσαμε μια επαναληπτική διαδικασία, για να βελτιστοποιήσουμε τα hyper parameters για κάθε μοντέλο. Η επιλογή των παραμέτρων έγινε βάση της μέσης ακρίβειας και της τυπικής απόκλισης της ακρίβειας κάθε μοντέλου.

1. KNN – Βέλτιστο πλήθος γειτόνων: 9



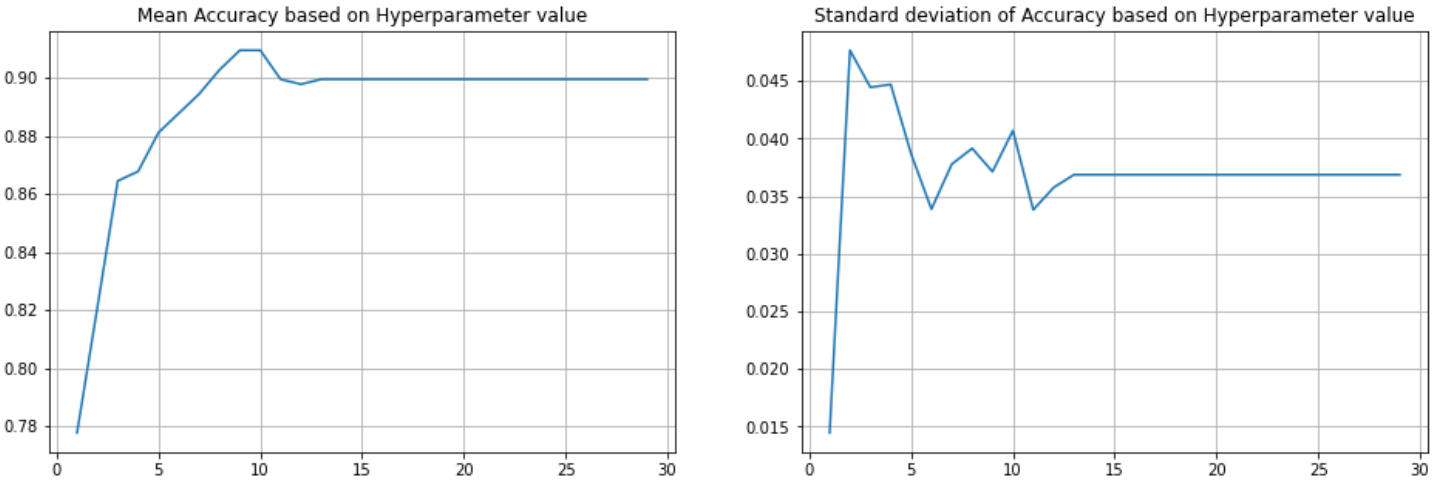
Εικόνα 4: Μεταβολή της απόδοσης του μοντέλου σύμφωνα με το Πλήθος γειτόνων

2. Decision Tree – Βέλτιστο βάθος: 5



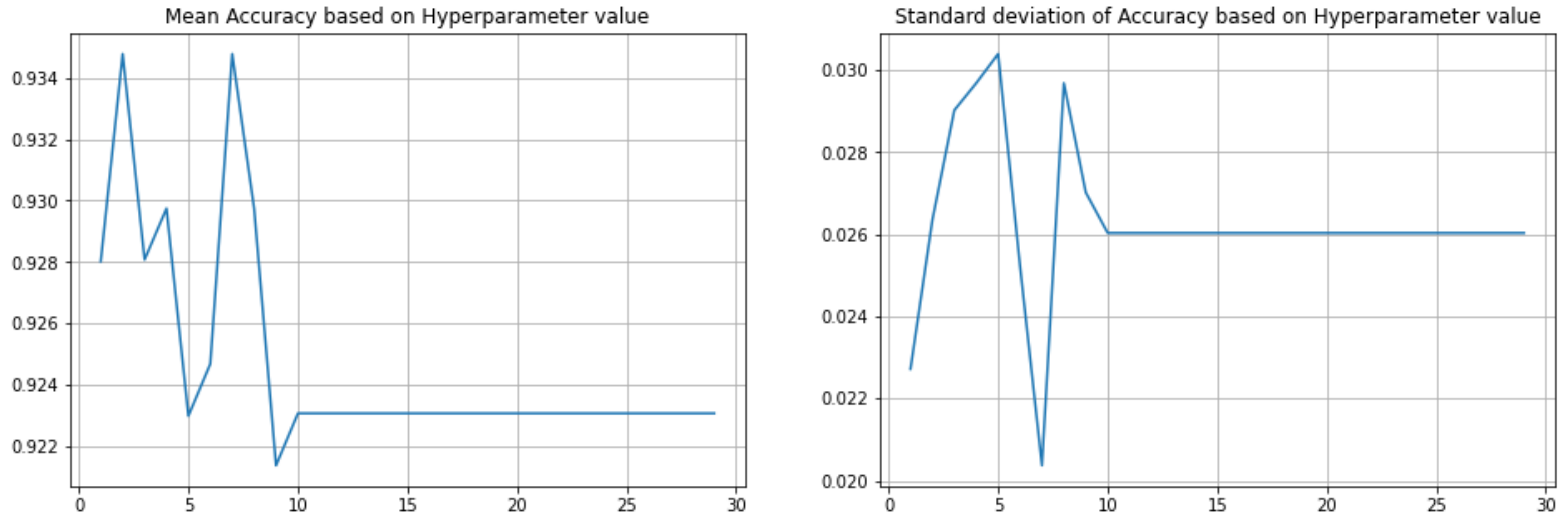
Εικόνα 5: Μεταβολή της απόδοσης του μοντέλου σύμφωνα με το μέγιστο βάθος δέντρου

3. Random Forest - Βέλτιστο βάθος: 9



Εικόνα 6: Μεταβολή της απόδοσης του μοντέλου σύμφωνα με το μέγιστο βάθος δέντρου

4. XGB – Βέλτιστο βάθος: 7



Εικόνα 7: Μεταβολή της απόδοσης του μοντέλου σύμφωνα με το μέγιστο βάθος δέντρου

Σε κάθε μια από τις παραπάνω περιπτώσεις επιλέξαμε τον καλύτερο δυνατό συμβιβασμό μεταξύ της μέγιστης μέσης ακρίβειας και της ελάχιστης τυπικής απόκλισης της ακρίβειας.

Τα αποτελέσματα μετά το Feature selection ήταν τα εξής:

```
GaussianNB:
   Test accuracy: 78.757% (±4.687%)
   Area under ROC: 86.106%
   Sensitivity: 77.5%
LogisticRegression:
   Test accuracy: 97.328% (±1.997%)
   Area under ROC: 98.872%
   Sensitivity: 94.868%
RandomForestClassifier:
   Test accuracy: 90.958% (±3.714%)
   Area under ROC: 94.796%
   Sensitivity: 80.447%
KNeighborsClassifier:
   Test accuracy: 86.785% (±2.44%)
   Area under ROC: 90.697%
   Sensitivity: 80.0%
DecisionTreeClassifier:
   Test accuracy: 87.613% (±3.794%)
   Area under ROC: 88.187%
   Sensitivity: 77.921%
XGBClassifier:
   Test accuracy: 93.477% (±2.037%)
   Area under ROC: 96.767%
   Sensitivity: 85.605%
```

Σε αυτό το σημείο είμαστε σε θέση να δημιουργήσουμε τα τελικά μοντέλα μας. Έτσι τα αρχικοποιούμε όλα με τις βέλτιστες παραμέτρους που υπολογίσαμε στο προηγούμενο βήμα και τα εκπαιδεύουμε πάνω σε όλα τα διαθέσημα δεδομένα μας (90% του dataset). Τέλος για να εκτημίσουμε λίγο καλύτερα την ακρίβεια τους τα ελέγχουμε με το holdout set (67 δείγματα) που είχαμε εξαρχής κρατήσει. Αυτή η διαδικασία έχει ως σκοπό να ελέγξει την συμπεριφορά του μοντέλου όταν έρχεται σε επαφή με καινούργια δεδομένα στα οποία δεν εκπαιδεύτηκε και αποτελλεί την καλύτερη εκτίμηση που μπορούμε να έχουμε για την απόδοση του. Τα αποτελέσματα, σε Confusion Matrix, φαίνονται παρακάτω.

```
Confusion Matrix for GaussianNB:
 [[40 12]
 [ 4 11]]
Sensitivity: 73.333%
Specificity: 76.923%
Negative Predictive Value: 90.909%
Accuracy: 76.119%


Confusion Matrix for LogisticRegression:
 [[51  1]
 [ 0 15]]
Sensitivity: 100.0%
Specificity: 98.077%
Negative Predictive Value: 100.0%
Accuracy: 98.507%


Confusion Matrix for RandomForestClassifier:
 [[50  2]
 [ 4 11]]
Sensitivity: 73.333%
Specificity: 96.154%
Negative Predictive Value: 92.593%
Accuracy: 91.045%


Confusion Matrix for KNeighborsClassifier:
 [[46  6]
 [ 4 11]]
Sensitivity: 73.333%
Specificity: 88.462%
Negative Predictive Value: 92.0%
Accuracy: 85.075%
```

```
Confusion Matrix for DecisionTreeClassifier:
 [[47  5]
 [ 3 12]]
Sensitivity: 80.0%
Specificity: 90.385%
Negative Predictive Value: 94.0%
Accuracy: 88.06%


Confusion Matrix for XGBClassifier:
 [[52  0]
 [ 4 11]]
Sensitivity: 73.333%
Specificity: 100.0%
Negative Predictive Value: 92.857%
Accuracy: 94.03%
```

Και στο 10-fold cross validation αλλά και στο holdout set φαίνεται ότι την καλύτερη απόδοση είχε το Logistic Regression μοντέλο (με ακρίβεια ίση με 97% και 98.5%) αντιστοίχως και ακολουθεί το XGBClassifier (93.5% και 94%) και μετά το Random Forest Classifier (91% και 91%). Άρα όπως αναμέναμε το Logistic Regression ανταποκρίθηκε καλύτερα στην επιλογή χαρακτηριστικών αφού αυξήθηκε η μέση ακρίβεια (test accuracy) από 56% σε 97%.

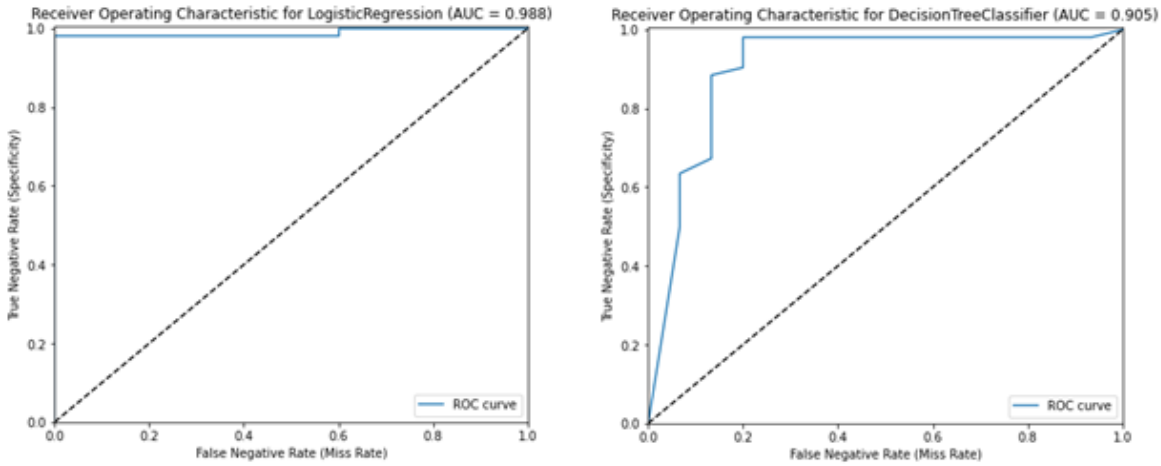
4

Συμπεράσματα

4.1 Τελική αξιολόγηση των μοντέλων 4.1

Σε αυτό το σημείο καλούμαστε να αποφανθούμε ποιο είναι το μοντέλο το οποίο αποδίδει καλύτερα συγκρινόμενο με τα υπόλοιπα. Για να ληφθεί μια τέτοια απόφαση σίγουρα πρέπει να λάβουμε υπ' όψιν το περιβάλλον μέσα στο οποίο θα λειτουργεί καθώς και τον βαθμό αυθεντίας που θα κατέχει στο εν' λόγω περιβάλλον. Έτσι, για την βέλτιστη αξιοποίηση του μοντέλου, κρίθηκε σκόπιμη η χρήση του σε συνεργασία πάντα με τους γιατρούς του ακτινιολογικού τομέα του νοσοκομείου. Ο ασθενής θα είναι σε θέση να λάβει τα αποτελέσματα των εξετάσεων του απευθείας από τον υπερηχογράφο μέσω του ενσωματομένου σε αυτόν αλγόριθμο κατηγοριοποίησης όγκων. Ιδανικά σε περίπτωση αρνητικού αποτελέσματος θα μπορούμε να είμαστε βέβαιοι ότι ο όγκος του ασθενούς δεν είναι κακοήθης και κατασυνέπεια ο ίδιος δε διατρέχει κίνδυνο. Σε περίπτωση θετικού αποτελέσματος θα καλείται ο γιατρός ο οποίος θα είναι σε θέση να εκτιμήσει περαιτέρω τον ασθενή. Γίνεται εμφανές ότι το βέλτιστο μοντέλο για αυτόν το σκοπό θα είναι αυτό που θα μπορεί με μεγάλη ακρίβεια να αποφανθεί πότε ένας όγκος δεν είναι κακοήθης ενώ συγχρόνως να μην «αφήνει» σε καμία περίπτωση έναν κακοήθει όγκο να αναγνωριστεί ως καλοήθεις (false negatives ή type II error). Έστι για την τελική αξιολόγηση των μοντέλων λαμβάνεται αποκλειστικά υπ' όψιν ο Confuson Matrix με τις τιμές των δεικτών της Ευαισθησίας (το ποσοστό των κακοήθων όγκων που ανίχνευσε ορθώς το μοντέλο μας) και της Αρνητικής Προγνωστικής Αξίας (το ποσοστό των ορθών ανιχνεύσεων των καλοήθων όγκων). Όπως είδαμε και πρωτύτερα, πάνω στα 67 περηχογραφήματα

του holdout set, καλύτερα αποτελέσματα είχε το Logistic Regression το οποίο ανίχνευσε ορθά όλους τους κακοήθεις όγκους και το 98% των καλοήθων όγκων. Ακολουθεί το Decision Tree Classifier το οποίο κατάφερε να ανιχνεύσει το 80% των κακοήθων όγκων (12/15 υπερυχογραφήματα) και είχε και τιμή δείκτη NPV ίση με 94%. Έπειτα ακολουθούν το XGB Classifier, το Random Forest Classifier, ο αλγόριθμος κοντινότερου γείτονα KNN και τέλος το Gaussian Naïve Bayes. Τέλος, εξετάζουμε την ROC καμπύλη των δύο καλύτερων μοντέλων (με άξονες το False Negative Rate και True Negative Rate) καθώς και την τιμή AUC τους (Εικόνα 4.4).



Εικόνα 8: ROC curve και AUC για Logistic Regression και Decision Tree Classifier

Έτσι γίνεται πλέον εμφανές ότι το μοντέλο Logistic Regression είναι κατά πολύ πιο αποτελεσματικό από το Decision Tree Classifier γεγονός το οποίο γίνεται εύκολα κατανοητό από την τιμή AUC (0.988 και 0.905 αντιστοίχως). Εδώ αξίζει να σημειώθει και αξία του ROC με την βοήθεια του οποίου βλέπουμε ότι υπάρχει κατάλληλο (και στην περίπτωσή μας βέλτιστο) decision threshold στο οποίο το μοντέλο παρουσιάζει περίπου 98% True Negative Rate ενώ συγχρόνως διατηρεί το False Negative Rate στο 0%. Αυτό σημαίνει ότι το μοντέλο ανιχνεύει το 98% των καλοήθων όγκων ενώ ταυτόχρονα δεν ανιχνεύει κανέναν κακοήθη όγκο ως καλοήθη (type II error).

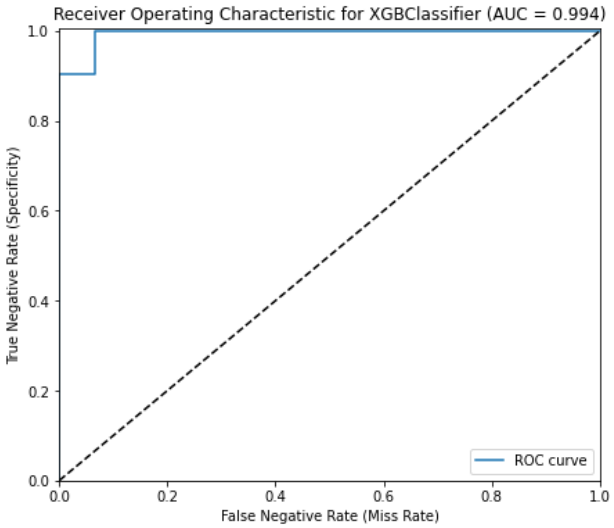
4.2 Σύγκριση με το State of the Art 4.2

Όσον αφορά τα αποτελέσματα των επιστημονικών άρθρων στα οποία έγινε αναφορά προηγουμένως, άμεση είναι η σύγκριση των αποτελεσμάτων των αλγορίθμων Random Forest Classifier και Logistic Regression που υλοποιήθηκαν. Συγκεκριμένα, σε προηγούμενη έρευνα ο Random Forest είχε αποδειχθεί ότι μπορεί να εμφανίσει τιμή AUC εώς και 98.9%[1] σε σύγκριση με την υλοποίηση μας που εμφάνισε 94.8% και 95.1% (σε 10-fold validation και στο holdout set αντιστοίχως). Επιπλέον, ο Logistic Regression σε προηγούμενες έρευνες υλοποιήθηκε με ακρίβεια κατηγοριοποίησης έως και 98.25%[4] σε σχέση με το δικό μας μοντέλο με Accuracy 97.3% και 98.1%. Τέλος, οι υλοποιήσεις με Support Vector Machine Classifier έδειξαν ακρίβεια από 86.5% έως 96%[2,3,5] και AUC ίση με 96.14%[2] σε σύγκριση με τις τιμές που πετύχαμε με το μοντέλο Logistic Regression (AUC 98.9% και 98.8%). Είναι εμφανές ότι η υλοποίηση μας έχει καλύτερη απόδοση.

4.3 Σύγκριση με το State of the Art αλγόριθμο XGB Classifier 4.3

Το μοντέλο XGB Classifier αφού εκπαιδεύτικε πάνω στα δεδομένα όπου εμείς έχουμε επιλέξει τα χαρακτηριστικά των υπερηχογραφημάτων αποδείχθηκε πως είναι λιγότερο αποδοτικό από το Logistic Regression σύμφωνα πάντα με τις μεθόδους αξιολόγησης που διαλέξαμε (Sensitivity: 73.333% και Negative Predictive Value: 92.857%). Παρ' όλ' αυτά ενδιαφέρον έχει η περίπτωση που το XGB Classifier εκπαιδευτεί πάνω στα ίδια 67 υπερηχογραφήματα και με τα 27 χαρακτηριστικά που είχαμε αρχικά διαλέξει. Σε αυτή την περίπτωση η τιμή του δείκτη Ευαισθησίας αυξάνεται στο 86% και η τιμή της Αρνητικής Προγνωστικής Αξίας στο 96.3%. Η απρόσμενη αυτή αύξηση της αποδοτικότητας οφείλεται στον αλγόριθμο Feature Importance που έχει ενσωματωμένο ο οποίος επιλέγει αυτόματα τα χαρακτηριστικά του υπερηχογραφήματος τα οποία είναι τα ποιο χρήσιμα. Στην εικόνα 4.2 φαίνεται η ROC καμπύλη και η τιμή AUC.

Η καμπύλη ROC είναι αρκετά όμοια με αυτή του μοντέλου Logistic Regression και μάλιστα η τιμή της AUC αίναι λίγο μεγαλύτερη. Συνεπώς, κανείς θα μπορούσε να υποστηρίξει ότι αυτό το μοντέλο υπερτερεί του Logistic Regression. Ισχύει ότι η διαφορά στην απόδοση τους είναι ελάχιστη, παρ' όλ' αυτά εξετάζοντας καλύτερα την ROC curve μπορεί κανείς να παρατηρήσει ότι το βέλτιστο decision threshold είναι αυτό στο οποίο το μοντέλο παρουσιάζει περίπου 91% True Negative Rate και ταυτόχρονα παρουσιάζει μηδενικό False Negative Rate. Συνεπώς με 98% True Negative Rate και μηδενικό False Negative Rate το μοντέλο Logistic Regression είναι πιο αποδοτικό.



Εικόνα 9: Roc curve και AUC για XGB Classifier

5

Βιβλιογραφία

[1] M. Abdel-Nasser, J. Melendez, A. Moreno, O. A. Omer, and D. Puig, "Breast tumor classification in ultrasound images using texture analysis and super-resolution methods," *Engineering Applications of Artificial Intelligence*, vol. 59, pp. 84–92, Mar. 2017, doi: 10.1016/j.engappai.2016.12.019.

[2] W. J. Wu, S. W. Lin, and W. K. Moon, "Combining support vector machine with genetic algorithm to classify ultrasound breast tumor images," *Computerized Medical Imaging and Graphics*, vol. 36, no. 8, pp. 627–633, 2012, doi: 10.1016/j.compmedimag.2012.07.004.

[3] Z. Zhuang, Z. Yang, S. Zhuang, A. N. Joseph Raj, Y. Yuan, and R. Nersisson, "Multi-Features-Based Automated Breast Tumor Diagnosis Using Ultrasound Image and Support Vector Machine," *Computational Intelligence and Neuroscience*, vol. 2021, pp. 1–12, May 2021, doi: 10.1155/2021/9980326.

[4] A. S. Assiri, S. Nazir, and S. A. Velastin, "Breast Tumor Classification Using an Ensemble Machine Learning Method," *Journal of Imaging*, vol. 6, no. 6, p. 39, Jun. 2020, doi: 10.3390/jimaging6060039.

[5] L. Ren, Y. Liu, Y. Tong, X. Cao, and Y. Wu, "[Multi-feature Extraction and Classification of Breast Tumor in Ultrasound Image]," *Zhongguo Yi Liao Qi Xie Za Zhi = Chinese Journal of Medical Instrumentation*, vol. 44, no. 4, pp. 294–301, Apr. 2020, doi: 10.3969/j.issn.1671-7104.2020.04.003.

[6] W. Al-Dhabyani, M. Gomaa, H. Khaled, and A. Fahmy, "Dataset of breast ultrasound images," *Data in Brief*, vol. 28, p. 104863, Feb. 2020, doi: 10.1016/j.dib.2019.104863.

[7] F. Pedregosa *et al.*, “Scikit-learn: Machine Learning in Python,” *arXiv:1201.0490 [cs]*, Jun. 2018, [Online]. Available: https://arxiv.org/abs/1201.0490.