ΕΛΛΗΝΙΚΗ ΔΗΜΟΚΡΑΤΙΑ
Εθνικόν και Καποδιστριακόν
Πανεπιστήμιον Αθηνών
ΙΔΡΥΘΕΝ ΤΟ 1837

Πρόγραμμα Μεταπτυχιακών Σπουδών (ΠΜΣ)
Μηχανική Μάθηση
Διδάσκων: Κ Γιάννης Παναγάκης
Εαρινό εξάμηνο 2024-2025

Ομάδα:
Κατωπόδης Οδυσσέας, Α.Μ.: 7115112400019

Γκούμας Βασίλειος, Α.Μ.: 7115132400008

Χελβατζή Δήμητρα, Α.Μ.: 7115132400007

1. Περιεχόμενα

2. Εισαγωγή και περιγραφή του προβλήματος

2.1 Επιλογή συνόλου δεδομένων

Ομάδα	**A3**	
Μέλη	**Datasets**	**Αριθμός Μητρώου**
Κατωπόδης Οδυσσέας	Medical MNIST	7115112400019
Γκούμας Βασίλειος	BreastMNIST 224x224	7115132400008
Χελβατζή Δήμητρα	CT Medical Images / Medical MNIST [1]	7115132400007

2.2 BreastMNIST 224x224[2]

Ο καρκίνος του μαστού είναι ο πιο συχνός καρκίνος στις γυναίκες παγκοσμίως και βασική αιτία θανάτων. Η έγκαιρη και ακριβής διάγνωση βελτιώνει σημαντικά την επιβίωση. Μέθοδοι όπως η μαστογραφία είναι κομβικές για την αντιμετώπιση του προβλήματος, αλλά η ερμηνεία τους απαιτεί χρόνο και μπορεί να διαφέρει μεταξύ ιατρών. [1] Η μηχανική μάθηση συμβάλλει σημαντικά στην ανάλυση ιατρικών εικόνων, επιτρέποντας την ακριβή διάκριση καλοηθών και κακοηθών αλλοιώσεων.
Το συγκεκριμένο σύνολο του BreastMNIST είναι ένα προεπεξεργασμένο υποσύνολο του MedMNIST+. Οι δημιουργοί του πήραν το αρχικό BreastMNIST+ dataset, το οποίο παρέχει τις εικόνες του σε ανάλυση μέχρι και 224x224, σε αντίθεση με το 28x28 του αρχικού MedMNIST, του αφαίρεσαν τις επαναλαμβανόμενες εικόνες, διόρθωσαν λάθη στο annotation, cropped ώστε να αφαιρέσουν άχρηστα ή ασήμαντα όρια από τις εικόνες και ελέχθηκαν από ειδικούς ακτινολόγους για τυχόν άλλα λάθη. [2]

2.3 Medical MNIST

Το συγκεκριμένο σύνολο δεδομένων αποτελείται μόνο από εικόνες χωρισμένο σε 6 κλάσεις και προφανώς προορίζεται για κατηγοριοποίηση (classification).
Από το σύνολο των εικόνων και μετά από συζήτηση στην αίθουσα του μαθήματος καταλήξαμε να χωρίσουμε μεταξύ μας (Οδυσσέας και Δήμητρα) το σετ “MedicalMNIST”.
Συγκεκριμένα αποφασίσαμε
Οδυσσέας κλάσεις :AbdomenCT, BreastMRI, Hand, and HeadCT
Δήμητρα κλάσεις: BreastMRI, ChestCT, CXR, and Hand
Οι επιλογές έγιναν με την ιδέα να προκύψουν ενδιαφέρουσες συγκρίσεις αργότερα.

2.4 CT Medical Images

Πραγματοποιήθηκε εξερευνητική ανάλυση δεδομένων (EDA) για το dataset CT Medical Images, εστιάζοντας σε 100 εικόνες που περιλάμβαναν πλήρη μεταδεδομένα και labels, τα οποία προέρχονταν από τα αρχεία dicom_dir, tiff_images και overview, το οποίο, ωστόσο, δεν αξιοποιήθηκε συνολικά, καθώς το αρχείο *full_archive.npz* (μορφή NumPy arrays) **δεν περιλάμβανε labels (π.χ 0 και 1), τα οποία είναι απαραίτητα για την εκπαίδευση ή την αξιολόγηση μοντέλων μηχανικής μάθησης.**

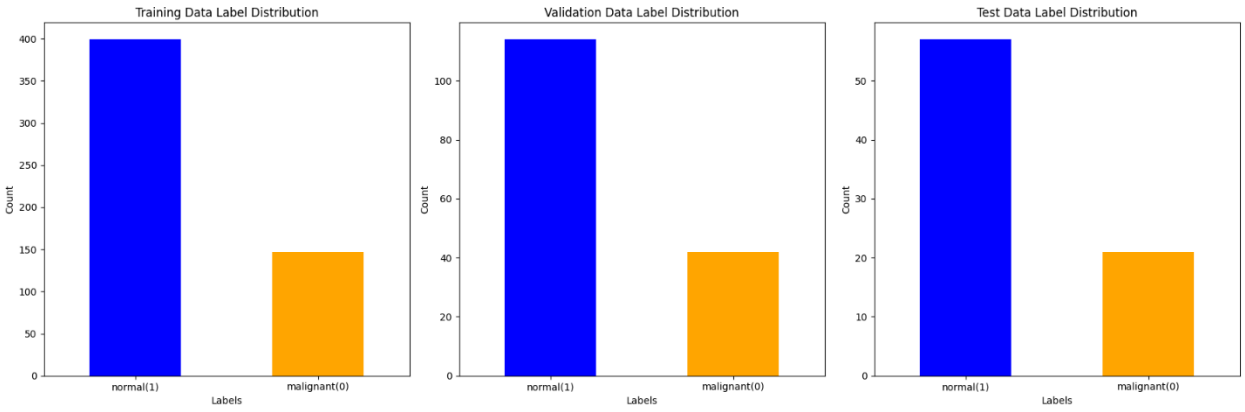
3. Προ επεξεργασία και Εξερεύνηση Δεδομένων EDA

3.1 BreastMNIST 224x224

3.1.a Εξερευνητική Ανάλυση Δεδομένων

Το dataset αποτελείται από 780 εικόνες οι οποίες χωρίζονται σε 3 σετ (train, validation, test) των 546 (70.00%), 78 (10.00)% και 156 (20.00%) αντιστοίχως. Μεταξύ τους ήταν labeled με σταθερό αλλά μη ισορροπημένο τρόπο, 73.8% του κάθε σετ να αντιστοιχεί σε υγιείς (normal/benign) εικόνες και το άλλο 26.92% σε εικόνες καρκινοπαθών (malignant) σε κάθε υποσύνολο.

[1] Η Δήμητρα μετά απο συζήτηση στο μάθημα αναγκάστηκε να λάβει / ασχοληθεί με κομμάτι από το Medical MNIST Dataset, καθώς από το δικό της έλλειπαν τα labels.
[2] Αρχικά ανατέθηκε στον Βασίλη το εξής σύνολο δεδομένων: https://huggingface.co/datasets/sanaa13/breastmnist2/blob/main/README.md αλλά επειδή δεν φόρτωνε αξιοποιήθηκε το εξής: https://zenodo.org/records/10519652, το οποίο βρίσκεται στην πρωτότυπη σελίδα του MNIST: https://medmnist.com/)



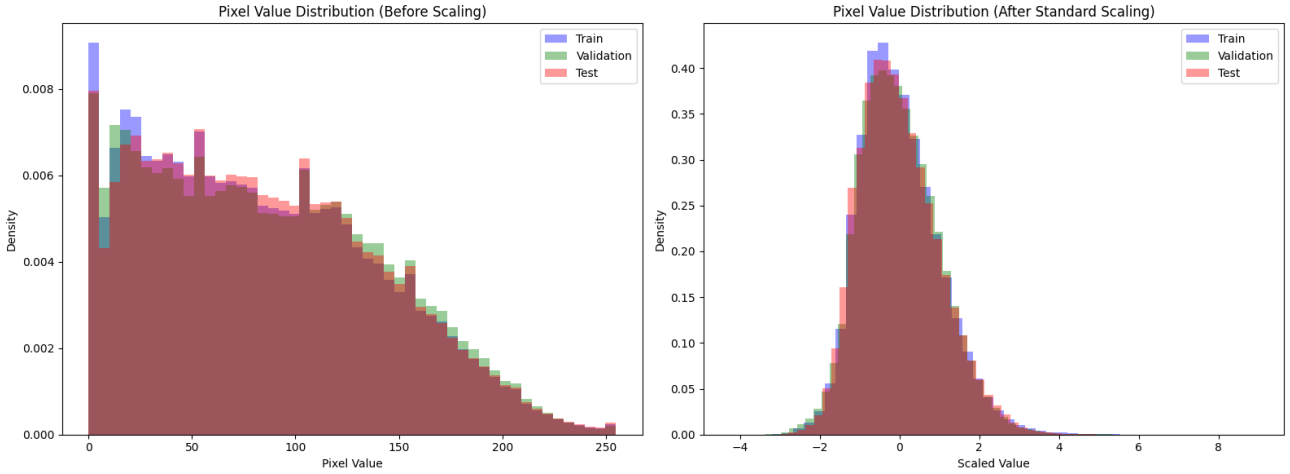
Ο

3.1.b Προ-επεξεργασία δεδομένων

Δεν έλλειπαν τιμές οπότε δεν χρειάστηκε να εφαρμωστεί κάποια μέθοδος χειρισμού τους. Τα dataset ερευνήθηκαν για corrupted εικόνες με σύντομο τρόπο εξερευνόντας εικόνες με μέγιστες και ελάχιστες τιμές φωτεινότητας στα pixel και μετέπειτα τις εικόνες με μέγιστο και ελάχιστο μέσο όρο φωτεινότητας στα pixel, ανα σετ. Οι εικόνες που προέκυψαν δεν εμφάνισαν διαφθορά οπότε διατηρήθηκαν. Επίσης έγινε αναζήτηση για outlier εικόνες, αλλά βρεθηκε μόνο μια εικόνα, η οποία λογικά θεωρήθηκε outlier επειδή ήταν πολύ φωτεινή. Δεδομένου πως είναι μόνο μια εικόνα που δεν παρουσιάζει φθορά και το dataset δεν είναι τόσο μεγάλο κρίθηκε πως δεν είναι αναγκαίο να αφαιρεθεί.

3.1.c Κανονικοποίηση

Για την κανονικοποίηση/τυποποίηση των χαρακτηριστικών, εφαρμόστηκε η μέθοδος Standard Scaling (Z-score normalization), ώστε κάθε χαρακτηριστικό να έχει μέση τιμή 0 και τυπική απόκλιση 1. Η επιλογή αυτή κρίθηκε πιο κατάλληλη επειδή οι αλγόριθμοι PCA, LDA και Logistic Regression είναι ευαίσθητοι στην κλίμακα των χαρακτηριστικών.



3.1.d Επιλογή και μετασχηματισμός χαρακτηριστικών

Εφαρμώστηκε αφαίρεση χαρακτηριστικών με χαμηλή διακύμανση με threshold 0.1 γιατί φάνηκε να αρμώζει καλύτερα για τα δεδομένα. Ερευνήθηκε και το ενδεχόμενο πολυωνυμικών χαρακτηρστικών αλλά απορίφθηκε πολύ εύκολα από νωρίς καθώς είχε κακά αποτελέσματα και δεν συνιθίζεται σε τασκ εικόνων ούτως ή άλλως. Εφαρμόστηκαν τεχνικές εξαγωγής και μηχανικής χαρακτηριστικών (feature engineering), στατιστικά ανά εικόνα (μέση τιμή, τυπική απόκλιση, skewness, κ.λπ.), υφής (LBP) και σχήματος (π.χ. area, perimeter, compactness), για την ενίσχυση της διακριτικής ικανότητας των μοντέλων. Όλα τα παραπάνω συνοδεύονται από οπτικοποιήσεις (boxplots, pairplots, heatmaps) για καλύτερη κατανόηση της κατανομής και των συσχετίσεων των χαρακτηριστικών. Τα κυριότερα που εξαχθηκαν από αυτά είναι πως ο μέσος όρος είναι ελαφρώς μεγαλύτερος από τη διάμεσο για τα περισσότερα χαρακτηριστικά, επομένως η κατανομή είναι αρνητικά λοξή και διαπιστώθηκε ένας εύκολος τρόπος για διαχωρισμό των labels δίχως την χρήση κάποιου μοντέλου, κοιτώντας απλά το mean των pixel της εκάστοτε εικόνας, το οποίο είναι συνολικά υψηλότερο για τις κανονικές εικόνες. Επιπροσθέτως επιχειρήθηκε χρήση autoencoder αλλά απορρίφθηκε λόγο δυσκολίας στην χρήση.

3.2 Medical MNIST

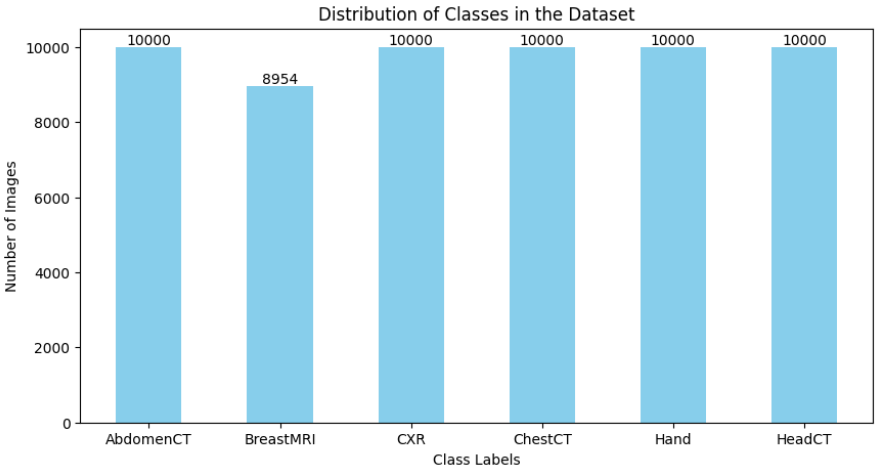
Κατωπόδης Οδυσσέας & Δήμητρα Χελβατζή

3.2.a Εξερευνητική Ανάλυση Δεδομένων

Στατιστική περιγραφή των δεδομένων:

Το dataset αποτελείται μόνο από συνολικά 58954 εικόνες κατανεμημένες σε φακέλους (κλάσεις) όπως παρουσιάζεται στην Εικόνα 1.

Κατανομή δεδομένων :

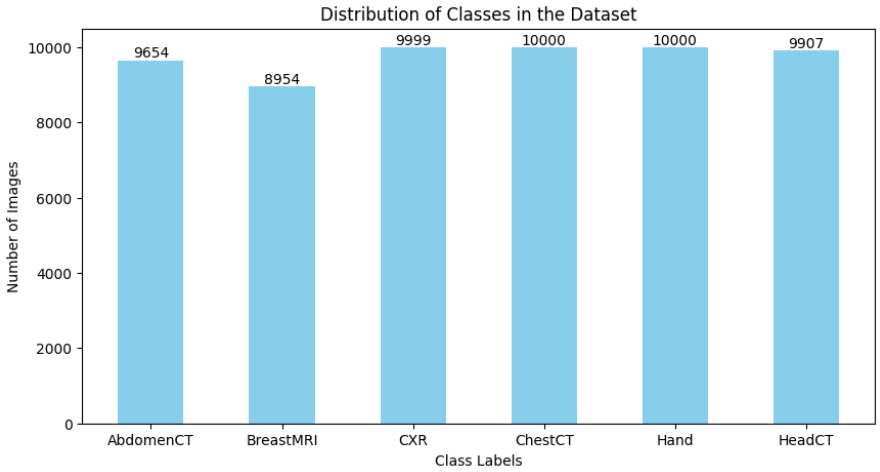


Εικόνα 1 Κατανομή αρχικών ανεπεξέργαστων δεδομένων

3.2.b Προ-επεξεργασία δεδομένων

Καθώς τα δεδομένα δεν περιέχουν αριθμητικές τιμές, συνήθεις στατιστικοί υπολογισμοί (όπως μέση τιμή, διάμεσος, τυπική απόκλιση, min, max, εκατοστημόρια) καθώς και έλεγχος για ακραίες τιμές και διαχείριση αυτών δεν έχουν ιδιαίτερο νόημα.
Θα γίνει όμως έλεγχος και διαχείριση για τα παρακάτω:

1. **Έλεγχος αν οι αναλύσεις όλων των εικόνων είναι ίδιες και αν το μέγεθος αυτών βοηθούν / επιβαρύνουν την επεξεργασία / ταξινόμηση.**
   Όλες οι εικόνες είναι 64x64 pixels και δεν χρειάστηκε να γίνει resize σε καμία. Το μέγεθος κρίνεται επαρκές για τις υπολογιστικές δυνατότητες του Google colab και δεν θα γίνει resize.
2. **Έλεγχος αν όλες οι εικόνες είναι ασπρόμαυρες και αν όχι μετατροπή αυτών.**
   Όλες οι εικόνες είναι ασπρόμαυρες και δεν χρειάστηκε μετατροπή καμίας.
   Απόσπασμα από το output του Jupiter notebook
3. **Έλεγχος για διπλότυπα και διαγραφή αυτών αν βρεθούν.**
   Αυτό είναι ένα πολύ σημαντικό βήμα καθώς θα μπορούσε να επηρεάσει ίσως και σημαντικά το μέγεθος και την ποιότητα του δείγματος συμπαρασείροντας και την ποιότητα της μεταγενέστερης κατηγοριοποιήσεις σε κλάσεις.
   Αν το δείγμα έχει πολλά διπλότυπα σε κάποια ή και πολλές κλάσεις τότε η εκπαίδευση του μοντέλου δεν θα είναι «αντικειμενική» και αν μειωθεί πολύ ο αριθμός των εικόνων από διαγραφές σε κάποια ή όλες από τις κλάσεις πάλι θα δυσκολέψει και την εκπαίδευση του μοντέλου καθώς δεν θα υπάρχουν επαρκή «δείγματα».
   Βρέθηκαν συνολικά 440 διπλότυπα, 346 στην κλάση *'AbdomenCT', 1 στην κλάση 'CXR', και 93 στην κλάση 'HeadCT'*. Ευτυχώς το μέγεθος των μοναδικών εικόνων ανά κλάση δεν επηρεάστηκε πολύ και κρίνεται επαρκές για να προχωρήσουμε δίχως να χρειάζεται άλλο χειρισμό από εμάς.

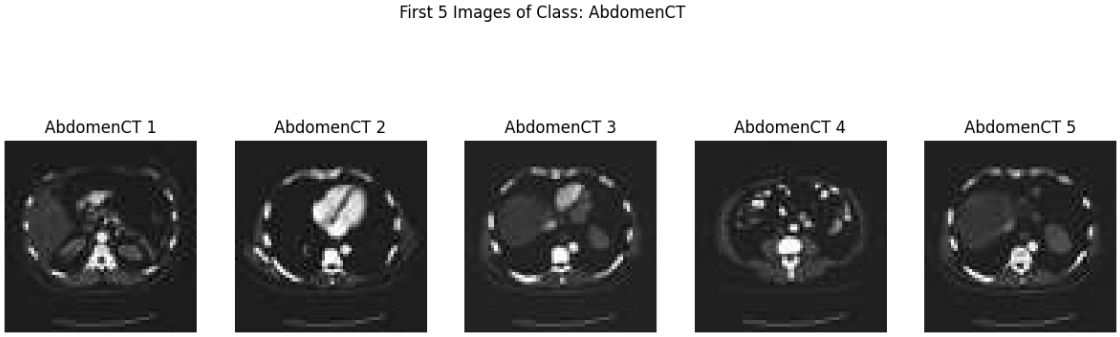


Εικόνα 2 Κατανομή επεξεργασμένων δεδομένων

4. **Εισαγωγή των δεδομένων, μετατροπή σε numpy array και παρουσίαση μερικών βασικών χαρακτηριστικών.**
   Αυτά θα μας βοηθήσουν να δούμε αν έχουν εισαχθεί σωστά ως arrays έχουν το σωστό τύπο μέγεθος και σχήμα έτσι ώστε να συνεχίσουμε προς την κατηγοριοποίηση αυτό το βήμα βοηθά ιδιαίτερα ως προς την «φυσική» κατανόηση του όλου ζητήματος.
   Απόσπασμα από το output του Jupiter notebook

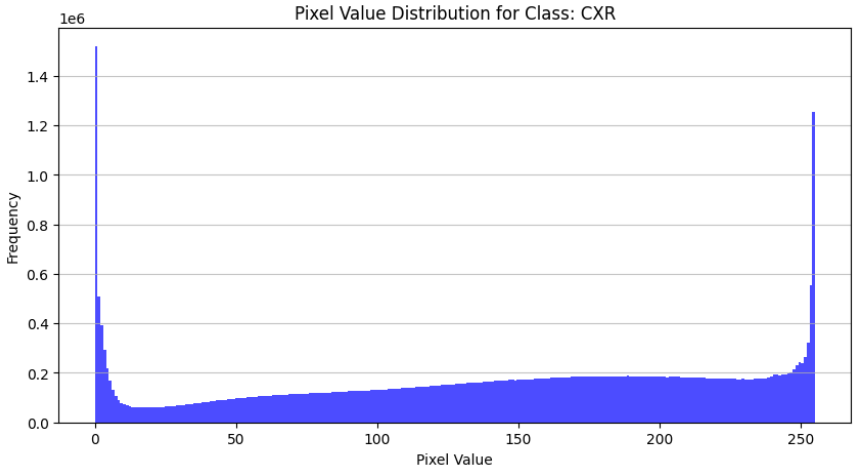
Loaded 58514 images.
Image shape: (64, 64)
Labels: ['AbdomenCT' 'BreastMRI' 'CXR' 'ChestCT' 'Hand' 'HeadCT']
Array shape: (58514, 64, 64)
Array data type: uint8

5. **Εκτύπωση των 5 πρώτων εικόνων ανά κλάση.**
Μας βοηθάει στη φυσική κατανόηση του προβλήματος.



Εικόνα 3 Παράδειγμα εκτύπωσης πρώτων 5 εικόνων από την κλάση 'AbdomenCT'

6. **Δημιουργία Histogram των pixels που μας δείχνει ποιες τιμές εμφανίζονται περισσότερο ανά κλάση.**



Εικόνα 4 Ιστόγραμμα τιμών pixel

3.2.c Άλλα ζητούμενα όπως στατιστικά μεγέθη, ακραίες τιμές, ελλείπουσες τιμές

Λόγω της φύσης των δεδομένων πού είναι οι εικόνες τα διάφορα στατιστικά μεγέθη όπως η μέση τιμή, διάμεσος, τυπική απόκλιση, ελάχιστο ,μέγιστο και εκατοστημόρια δεν έχουν κάποιο νόημα καθώς όλες οι τιμές μας θα είναι στο εύρος 0 (σκοτάδι)- 255 (πλήρης φωτεινότητα) και ανάλογα την εικόνα θα εμφανίζεται περισσότερα μαύρα είτε λευκά pixel σε διαφορετικό σημείο.
Ταυτόχρονα δεν τίθεται θέμα ελλειπουσών τιμών από τις εικόνες ούτε διαχείριση αυτών φυσικά.

3.2.d Κανονικοποίηση

Αρχικά, βασιζόμενοι στις ελάχιστες απαιτήσεις της της εργασίας (>= 1000 παραδείγματα δεδομένων), επιλέξαμε να κρατήσουμε **1000 εικόνες από κάθε κλάση** (θυμίζω Οδυσσέας κλάσεις :AbdomenCT, BreastMRI, Hand, and HeadCT και Δήμητρα: BreastMRI, ChestCT, CXR, Hand), μέγεθος επαρκές και εύκολα διαχειρίσιμο και από το google colab.
Πριν για να συνεχίσουμε με τη μείωση διαστάσεων, έγινε «flatten» στα δεδομένα των εικόνων από (1000, 64,64)→(1000, 4096) και κανονικοποίηση των τιμών των pixels με χρήση του standard scaler scikit learn. Τα δεδομένα καταλήγουν με αυτόν τον τρόπο να έχουν μέση τιμή = 0 και τυπική απόκλιση=1. [3]
Εντοπίστηκαν 7.750 ακραιες τιμές (outliers), οι οποίες θεωρήθηκαν αναμενόμενες λόγω της φύσης των εικόνων και διατηρήθηκαν. Η τυποποίηση έδειξε ικανοποιητικά αποτελέσματα, χωρίς ελλείπουσες τιμές, διευκολύνοντας την περαιτέρω ανάλυση.

3.2.e Επιλογή και μετασχηματισμός χαρακτηριστικών

Εφαρμόσαμε (Variance Threshold) για να αφαιρέσουμε χαρακτηριστικά με μικρή διακύμανση (κάτω από το 0.01), αλλά μετά την εφαρμογή του, παρατηρήσαμε ότι παρέμειναν 4096 χαρακτηριστικά, υποδεικνύοντας ότι δεν υπήρχαν χαρακτηριστικά με χαμηλή διακύμανση που να μπορούν να αφαιρεθούν. Επομένως, μπορούμε να παραλείψουμε αυτό το βήμα του feature engineering, καθώς τα δεδομένα μας έχουν ήδη αρκετή ποικιλία, και να προχωρήσουμε σε μείωση διαστάσεων για βελτίωση της απόδοσης του μοντέλου.

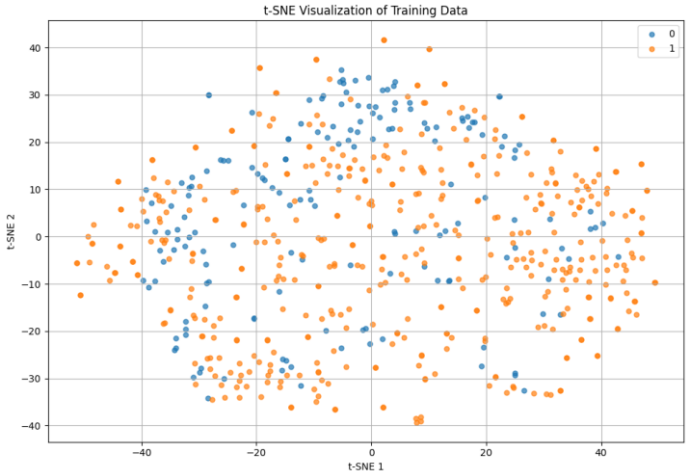
3.3 CT Medical Images

Η μέση ηλικία των ασθενών ήταν 67,48 έτη, με εύρος από 39 έως 83. Οι τιμές των pixels παρουσίαζαν σημαντική διακύμανση (από -1024 έως 3071), υποδεικνύοντας την ανάγκη για κανονικοποίηση πριν την εφαρμογή σε μοντέλα μηχανικής μάθησης. Εντοπίστηκαν επίσης ακραίες τιμές (outliers), οι οποίες υποδεικνύουν ετερογένεια στο σύνολο των δεδομένων. Καμία από τις εικόνες δεν εμφάνισε ελλείπουσες τιμές στα pixels. Στην προεπεξεργασία, εντοπίστηκαν δύο ελλείπουσες τιμές στη στήλη ContrastTag, οι οποίες αντικαταστάθηκαν με την επικρατούσα τιμή "NONE", χωρίς να επηρεάσουν την κατανομή των δεδομένων. Η τελική, καθαρή εκδοχή του συνόλου αποθηκεύτηκε στο αρχείο eda_cleaned_updated.csv. Όπως αναφέρθηκε και παραπάνω, το dataset δεν αξιοποιήθηκε καθώς το αρχείο *full_archive.npz* (μορφή NumPy arrays) δεν περιλάμβανε labels (π.χ 0 και 1).

4. Dimensionality reduction

4.1 BreastMNIST 224x224

Εφαρμόστηκε PCA με 50 components και LDA στα δεδομένα για να γίνει καλύτερη εκπαίδευση μοντέλων. Ωστόσο ακόμα και έτσι, τα δεδομένα δεν ήταν διαχωρίσημα, όπως είναι εμφανές από το tsne plot παρακάτω.



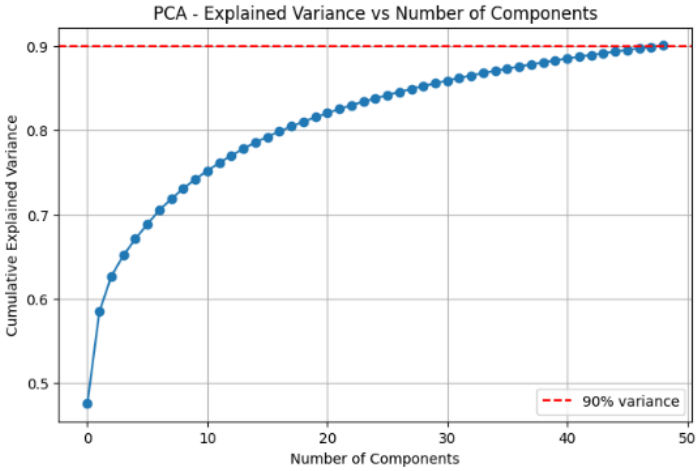
Δοκιμάστηκαν διάφορες τιμές και δεν μπορούσε να βρεθεί αποτελεσματικός τρόπος να εμφανιστούν τα δεδομένα, εμφανιζόντοσυαν όλα σε μορφή συγκεχημένου νέφους.

4.2 Medical MNIST

Γενικά η όποια μείωση διαστάσεων και επεξεργασία εκτελέσθηκε μόνο στο κομμάτι των δεδομένων train έτσι ώστε να μην εισχωρήσουν πληροφορίες από τα άλλα κόμματα όπως το test στον αλγόριθμο και στην ουσία με αυτό τον τρόπο «κλέψουμε λίγο». Εφαρμόστηκαν 2 τεχνικές η Principal component analysis και η Linear Discriminant Analysis.

4.2.a Principal component analysis

Η λογική πίσω από την επιλογή των ποσών κύριων συνιστωσών θα κρατήσουμε βασίζεται στη διατήρηση του 90% της συνολικής διακύμανσης των δεδομένων. [4]
Στην βιβλιογραφία βρήκα και λίγο πιο αυστηρές προδιαγραφές που πρότειναν να επιλέγεται τέτοιος αριθμός από κύριες συνιστώσες ώστε να διατηρείται το 95% της συνολικής διακύμανσης. [5]
Στα όρια της συγκεκριμένης εργασίας και επειδή θα ήθελα να είμαι «ευκίνητος» καθώς θα χρειαστεί να γίνει η εκτέλεση κώδικα πολλές φορές μέχρι να καταλήξω στον τελικό προτίμησα να κρατήσω το 90% της συνολικής διακύμανσης.
Σε όλη την εργασία παρατήρησα ότι αυτή η μέθοδος παράγει τα καλύτερα αποτελέσματα και ιδιαίτερα σημαντικό ότι μειώνει το κόστος του υπολογισμού άρα και το χρόνο εκτέλεσης.



Εικόνα 5 Συνολική διακύμανση σε σχέση με τον αριθμό των κύριων συνιστωσών

4.2.b Linear Discriminant Analysis

Με βάση την βιβλιογραφία, ο μέγιστος αριθμός components που μπορεί η LDA να βρει είναι ίσος με τον αριθμό των κλάσεων-1 άρα για εμένα ίσος με 3. [6]
Αυτό συμβαίνει λόγω του μαθηματικού μοντέλου της LDA.

5. Classification

5.1 BreastMNIST 224x224

5.1.a PCA & LDA με κλασσικούς αλγορίθμους

Τα δεδομένα που προέκυψαν από PCA και LDA χρησιμοποιούνται ως είσοδος σε διάφορους αλγορίθμους μηχανικής μάθησης, όπως Logistic Regression, Support Vector Machines (SVM), k-Κοντινότεροι Γείτονες (k-NN) και Naive Bayes. Για κάθε αλγόριθμο πραγματοποιήθηκε αναζήτηση υπερπαραμέτρων (Grid Search) και αξιολόγηση της απόδοσης με μετρικές όπως η ακρίβεια, η precision, η recall, το F1-score και η καμπύλη ROC, αλλά δόθηκε παραπάνω σημασία στο recall των τιμών 0 καθώς είναι ιατρικό task. Η συστηματική αυτή διαδικασία επέτρεψε τη σύγκριση των μοντέλων και την επιλογή της βέλτιστης προσέγγισης για το συγκεκριμένο πρόβλημα.
Κατα την εξερεύνση των datasets που προέκυψαν από την προεπεξεργασία η επίλογη κατάλληλου συνόλου αποκαλύφθηκε να ειναι περίπλοκη. Πραγματοποιήθηκε εκτενής πειραματική αξιολόγηση διαφορετικών συνόλων χαρακτηριστικών, όπως στατιστικά ανά εικόνα, υφής (LBP), σχήματος και οι συνδυασμοί τους, με στόχο τη βελτιστοποίηση της απόδοσης των μοντέλων μηχανικής μάθησης. Για κάθε σύνολο χαρακτηριστικών εκπαιδεύτηκαν και αξιολογήθηκαν μοντέλα Logistic Regression τόσο στο validation όσο και στο test set, με και χωρίς τη χρήση PCA, το οποίο χρησιμοποιήθηκε για μείωση της διαστασιμότητας. Η χρήση Logistic Regression μοντέλων έγινε για την ταχύτητα τους και ο έλεγχος γινόταν και στο test και στο validation set για τυχόν διαφοροποιήσεις που μπορεί να προκύψουν. Η υπόθεση ήταν πως τα μοντέλα με δεδομένα που είναι με περισσότερες πληροφορίες μπορει να ευδοκιμούν καλύτερα αποτελέσματα, ωστόσο δεν ήταν προφανές στην αρχή ποιά features και ποιοί συνδιασμοί θα ήταν αυτοί με τα βέλτιστα αποτελέσματα.
Δυστυχως, κατά τη διάρκεια των δοκιμών παρατηρήθηκαν σημαντικές διακυμάνσεις στην απόδοση των μοντέλων ανάλογα με το σύνολο χαρακτηριστικών και το split των δεδομένων, γεγονός που δυσκόλεψε την επιλογή των «καλύτερων» χαρακτηριστικών. Αφενώς, οι προβλέψεις στα test και validation sets παρουσίαζαν σοβαρές διακυμάνσεις και μεταξύ τους και αφετέρου, χαρακτηριστικά που εμφάνιζαν υψηλή ακρίβεια σε ένα split, παρουσίαζαν χαμηλότερη απόδοση σε άλλο, υποδεικνύοντας ότι η σταθερότητα και η γενίκευση των αποτελεσμάτων δεν ήταν δεδομένες. Έτσι, η τελική επιλογή των χαρακτηριστικών θα χρειαζόταν να γίνει με βάση τη συνολική συμπεριφορά τους και όχι μόνο μεμονωμένες επιδόσεις, λαμβάνοντας υπόψη τη συνέπεια και τη γενικευσιμότητα των αποτελεσμάτων. Στην τελική επέλεξα να χρησιμοποιήσω το dataset που συνδίαζε κανονικοποίηση και variance thresholding είχε επίσης τις στατιστικές από το mean και std, είχε τα καλύτερα αποτελέσματα και βρίσκει μια καλή ισορροπία μεταξύ λίγης και πολύς προεπεξεργασίας.
Επίσης άλλες δοκιμές που έκανα ήταν μεταξύ των σετ έτσι όπως τα είχα από το npz αρχείο και με έναν νέο διαχωρισμό των δεδομένω. Πρόσεξα να γίνει ισορροπημένα ώστε να υπάρχουν αντίστοιχα ποσοστά μεταξύ των υποσυνόλων και να ανακατευθούν ώστε να εγγυηθεί διαφοροποίηση από τον αρχικό χωρισμό. Το αποτέλεσμα αυτού ήταν τραγικά χειρότερα αποτελέσματα από ότι πριν. Ύστερα από διερεύνση των δεδομεών δεν μπροούσα να καταλήξω σε κάποιο συμπέρασμα για το γιατί συνέβαινε αυτό. Ενδεχομένως να ε΄χιε χωριστεί με ένα συγκεκριμένο σκεπτικό με αυτόν τον τρόπο τον οποίο δεν διευκρίνισαν πουθενά ή να ήταν εντελώς τυχαίο, σε κάθε περίπτωση έκρινα πως θα είναι καλύτερο να διατηρήσω τα δεδομένα χωρισμένα όπως πρότειναν. Επίσης επειδή θα εφαρμόσουμε k-folding στους αλγόριθμους που ακολουθούν επέλεξα να

συνδιάσω το validation set (10% του συνόλου) στο training set (70% του συνόλου) ώστε να έχω το Test set (20% του συνόλου) για δοκιμές που είναι μεγαλύτερο και άρα θα μειώνει το variance, δίνοντας έτσι πιο έγκυρες προβλέψεις.
Θέλω να ελαχιστοποιήσω τις περιπτώσεις ψευδώς αρνητικών, δηλαδή τις περιπτώσεις όπου υπάρχει κακοήθης όγκος (κλάση 0) αλλά το μοντέλο προβλέπει "φυσιολογικό" (κλάση 1). Συνεπώς, πρέπει να μεγιστοποιήσω την ανάκληση (recall) για την τιμή 0. Το βασικό μοντέλο είχε πολύ χαμηλή ανάκληση για την κλάση 0 (περίπου 34%), πιθανότατα επειδή το σύνολο δεδομένων δεν είναι ισορροπημένο.
Για να το αντιμετωπίσω, ρύθμισα την παράμετρο class_weight του logistic regression σε 'balanced' και αφαίρεσα την επιλογή multi_class='multinomial', καθώς το πρόβλημά μου αφορά μόνο δύο κατηγορίες, σε αντίθεση με των υπολοίπων στην ομάδα μου. Εστίασα την GridSearchCV στη χρήση του αλγορίθμου lbfgs, καθώς το σύνολο δεδομένων μου δεν είναι αρκετά μεγάλο ώστε να αξιοποιήσω τον saga, και έψαξα για υπερπαραμέτρους που θα μεγιστοποιούσαν το συνολικό recall.
Ωστόσο, αυτό οδήγησε σε recall 0.96 για την κλάση 1 αλλά μόλις 0.40 για την κλάση 0. Χρησιμοποιώντας έναν προσαρμοσμένο μετρητή (custom scorer) που επικεντρώνεται στην ανάκληση της κλάσης 0, το μοντέλο έφτασε σε ανάκληση 0.67 για την κλάση 0, που παρότι αποτελεί βελτίωση, παραμένει μη ικανοποιητική. Τουλάχιστον, η συνολική ακρίβεια στο test set δεν μειώθηκε σημαντικά, πηγαίνοντας από 0.81 σε 0.79. Τέλος, εξερεύνησα διαφορετικές επιλογές για PCA components, αλλά χωρίς επιτυχία.

Με παρόμοιο τρόπο όπως στο Logistic Regression, εφάρμοσα grid search για την εύρεση των βέλτιστων υπερπαραμέτρων και στους υπόλοιπους αλγορίθμους. Ακολουθούν τα αποτελέσματα:

	PCA Components	**Hyperparameters**	**Results**
Logistic Regression	50	'C': np.float64(0.0001), 'class_weight': 'balanced', 'max_iter': 100, 'penalty': 'l2', 'solver': 'lbfgs'	Recall_0: 0.67 Test Accuracy: 0.7948717948717948
SVM	50	'C': 1, 'class_weight': 'balanced', 'gamma': 'scale', 'kernel': 'rbf'	Recall_0: 0.67 Test Accuracy: 0.7948717948717948
K-NN	50	'metric': 'manhattan', 'n_neighbors': np.int64(2), 'weights': 'uniform'	Recall_0: 0.76 Test Accuracy: 0.7051282051282052
Naive Bayes	50	‘var_smoothing’: np.float64(2.782559402207126e-06	Recall_0: 0.76 Test Accuracy: 0.6602564102564102

Η ανάκληση των KNN και Naïve Bayes έφτασε στο υψηλότερο επίπεδο των 0.76, αλλά η ακρίβεια στο test set ήταν 0.705, αρκετά χαμηλότερη από την καλύτερη δυνατή τιμή που επιτεύχθηκε με το Logistic Regression και το SVM, 0.794.

Με παρόμοιο τρόπο εργάστηκα και για το LDA. Ακολουθούν τα αποτελέσματα:

	Hyperparameters	**Results**
Logistic Regression	'C': np.float64(0.0001), 'class_weight': 'balanced', 'max_iter': 100, 'penalty': None, 'solver': 'lbfgs'	Recall_0: 0.67 Test Accuracy: 0.8141025641025641
SVM	'C': 1, 'class_weight': 'balanced', 'gamma': 0.0001, 'kernel': 'rbf'	Recall_0: 0.93 Test Accuracy: 0.4807692307692308
K-NN	'metric': 'euclidean', 'n_neighbors': np.int64(2), 'weights': 'uniform'	Recall_0: 0.76 Test Accuracy: 0.7435897435897436
Naive Bayes	'var_smoothing': np.float64(2.782559402207126e-06)	Recall_0: 0.60 Test Accuracy: 0.8076923076923077

5.1.b Νευρωνικά δίκτυα

Για τα νευρωνικά δίκτυα, πραγματοποιήθηκε προεπεξεργασία των δεδομένων, μετατροπή των εικόνων και των ετικετών σε κατάλληλη μορφή (δίχως flattening και προεπεξεργασία όπως έγινε για τους άλλους αλγηορίθμους) και διαχωρισμός σε σύνολα εκπαίδευσης, επικύρωσης και δοκιμής, ορίζοντας το αρχικό σύνολο επικύρωσης ως το σύνολο δοκιμής και το αρχικό σύνολο επικύρωσης ως δοκιμής. Με αυτόν τον τρόπο εξασφαλίστηκε προτιμότερος διαχωρισμός δεδομένων για την εκπαίδευση των νευρωνικών δικτύων, καθώς με την ανταλλαγή αυτή δημιουργήθηκαν ένα σύνολο επικύρωσης 20% του συνολικού και ένα ελέγχου 10% του συνόλου.
Για το συνελικτικο νευρωνικο δίκτυο(CNN), σχεδιάστηκε η αρχιτεκτονική με τρία συνελικτικά επίπεδα, ενεργοποιήσεις (ReLU, LeakyReLU ή ELU), max pooling και πλήρως συνδεδεμένα επίπεδα, ενώ εφαρμόστηκε dropout για regularization και χρησιμοποιήθηκε cross-entropy loss για την εκπαίδευση. Πραγματοποιήθηκαν πειράματα με διαφορετικούς υπερπαραμέτρους (learning rate, weight decay, activation function) και για κάθε συνδυασμό εκπαιδεύτηκε το μοντέλο, καταγράφοντας την απόδοση σε κάθε εποχή. Τέλος, επιλέχθηκαν τα καλύτερα μοντέλα με βάση 3 κριτήρια: validation accuracy, recall of 0 και custom metric, το οποίο υπολογιζόταν από το 70% από το recall 0 και κατά 30% από το validation accuracy σε περίπτωση που εμφανιζόταν ένας συνδιασμός υπερπαραμέτρων που είχε καλή ισορροπία μεταξύ recall και validation acc. Αξιολογήθηκε η τελική τους απόδοση σε validation και test set με αναλυτικά γραφήματα και μετρικές.
Ακολουθούν τα αποτελέσματα από τις δοκιμές:

	Hyperparameters	**Results**
Best Validation Accuracy	'lr': 0.0001, 'activation_fn': 'ReLU', 'weight_decay': 0.001	Test Loss: 0.4457, Accuracy: 91.03%, Recall 0: 0.7619
Best Recall 0	'lr': 0.001, 'activation_fn': 'LeakyReLU', 'weight_decay': 0.01	Test Loss: 0.2699, Test Accuracy: 92.31% Recall 0: 0.83
Best custom metric	'lr': 0.0001, 'activation_fn': 'ReLU', 'weight_decay': 0.01	Test Loss: 0.3670, Accuracy: 91.03%, Recall 0: 0.7619

Για το πολυεπίπεδο νευρωνικό δίκτυο (MLP) πραγματοποιήθηκε προεπεξεργασία των δεδομένων και μετατροπή των εικόνων και των ετικετών σε κατάλληλη μορφή για το PyTorch, με διαχωρισμό σε σύνολα εκπαίδευσης, επικύρωσης και δοκιμής. Σχεδιάστηκε η αρχιτεκτονική του MLP με δύο ή περισσότερα κρυφά επίπεδα, συναρτήσεις ενεργοποίησης (ReLU, LeakyReLU ή Tanh), dropout για regularization και πλήρως συνδεδεμένα επίπεδα. Εφαρμόστηκε σταθμισμένη συνάρτηση κόστους (CrossEntropyLoss με βάρη) για αντιμετώπιση ανισορροπίας μεταξύ των κλάσεων και χρησιμοποιήθηκε ο Adam optimizer με L2 regularization. Για να καταπολεμηθεί η ανισορροπία των δεδομένων αναζητήθηκε και καλύτερο βαρος για την τάξη 0 ώστε να βελτιστοποιηθεί η ανάκληση στην τιμή 0. Πραγματοποιήθηκαν πειράματα με διαφορετικούς συνδυασμούς υπερπαραμέτρων (αριθμός κρυφών νευρώνων, ρυθμός μάθησης, dropout, συνάρτηση ενεργοποίησης) και για κάθε συνδυασμό εκπαιδεύτηκε το μοντέλο, καταγράφοντας την απόδοση σε κάθε εποχή. Τέλος, επιλέχθηκε το καλύτερο μοντέλο με βάση την ειδική μετιρκή που αξιοποιήθηκε και στο CNN που συνδιάζει 70% recall 0 και 30% validation accuracy, και αξιολογήθηκε η τελική του απόδοση σε validation και test set με αναλυτικά γραφήματα και μετρικές.

	Hyperparameters	**Results**
Best Recall 0	weight_for_0 = 3.0 , weight_for_1 = 1.0, 'lr': 0.0001, 'activation_fn': 'ReLU', 'weight_decay': 0.001	Test Accuracy: 34.62% Recall 0: 1.0
Custom Metric	Weight for 0 = 3.0, Str(hidden_layers): '[256, 128]', Activation function: 'ReLU', Learning rate: 0.0001, Dropout rate: 0.3	Test Accuracy: 0.6795 Recall 0: 0.81

5.2 Medical MNIST

Στα πλαίσια αυτής της εργασίας γενικά δόθηκε βάση στην εκμάθηση και σωστή χρήση των υπαρχόντων εργαλείων και βιβλιοθηκών και όχι τόσο στη δημιουργία προσωποποιημένων λύσεων. Με αυτόν τον γνώμονα προχώρησα στους παρακάτω αλγόριθμους κατηγοριοποίησης.
Σχετικά με το χώρισμα των δεδομένων σε 2 σετ train & test, χρησιμοποιήθηκε η στάνταρ βιβλιοθήκη της Scikit learn και βασιζόμενος στη βιβλιογραφία αποφασίστηκαν τα ποσοστά να είναι **70% το test και 30% το train.**

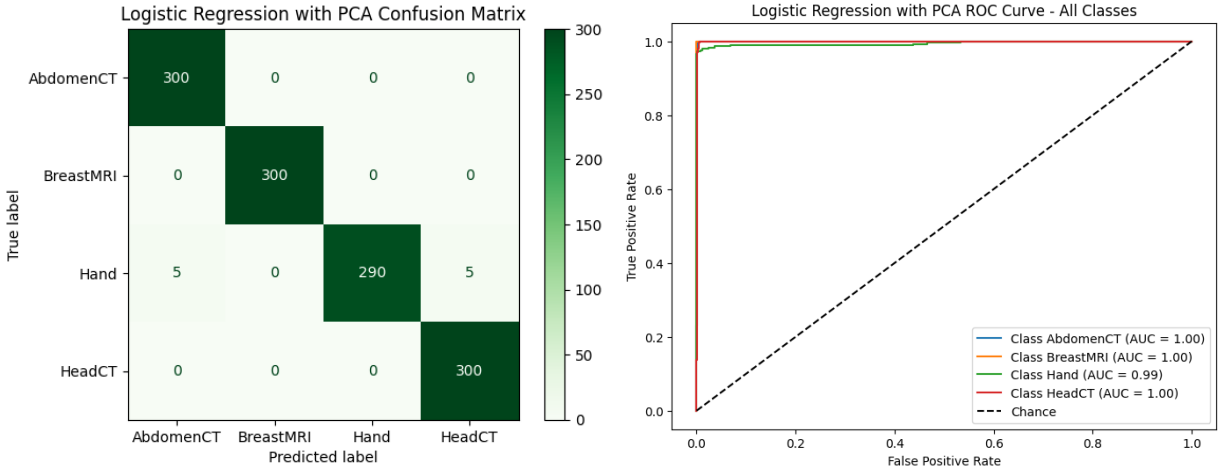
Σχετικά με τους αλγορίθμους κατηγοριοποίησης και τα νευρωνικά δίκτυα τύπου Multi-layer Perceptron και πάλι προτιμήθηκε η βιβλιοθήκη Scikit learn ενώ για τα συλλεκτικά νευρωνικά δίκτυα χρησιμοποιήθηκε η Tensorflow μαζί με το Keras API.
Σε όλα τα μοντέλα δεν δημιουργήθηκε ξεχωριστό σετ validation. **Η μέθοδος k-fold της Scikit learn**, χρησιμοποιήθηκε εκτενώς όπου υπήρχε ανάγκη βελτιστοποίησης παραμέτρων.
Ομοίως , σχετικά με τα δεδομένα από τις κατηγορίες BreastMRI, ChestCT, CXR και Hand, ο διαχωρισμός έγινε για το σύνολο εκπαίδευσης (70%) και το σύνολο ελέγχου (30%), χρησιμοποιώντας την τεχνική train_test_split με την παράμετρο stratify, ώστε να διασφαλιστεί η ισοκατανομή των κατηγοριών. Δεν δημιουργήθηκε ξεχωριστό σύνολο επικύρωσης, καθώς για την επικύρωση των μοντέλων και τη βελτιστοποίηση των υπερπαραμέτρων χρησιμοποιήθηκε η μέθοδος k-fold cross-validation.

5.2.a Logistic regression

5.2.a.1 Οδυσσέας

- Από την αρχή και χωρίς καμία μείωση των διαστάσεων το μοντέλο αποδίδει πολύ καλά με τιμές τις ακρίβειας ταξινόμησης πάνω από 0,99. Συνολικά 10 λάθη γίνονται και όλα παρατηρούνται στην κλάση “hand”
- Με τη χρήση της PCA η ακρίβεια παραμένει πρακτικά ίδια, αλλά μειώνεται σίγουρα το υπολογιστικό κόστος οπότε είναι δόκιμη χρήση της.
- Οι μέθοδοι βελτιστοποίησης των παραμέτρων δεν δίνουν κάποια αξιόλογη αύξηση ήδη τα νούμερα της ακρίβειας προσεγγίζουν το 1.
- Όσον αφορά την μέθοδο LDA και αυτή έχει πολύ αξιόλογα αποτελέσματα με ακρίβεια της τάξης του 0,9175 με βελτιστοποίηση των υπερ παραμέτρων και η ευαισθητοποίησή τους φτάνει το 0,94 όμως το αυξημένο υπολογιστικό κόστος δεν δικαιολογείται από τέτοιες μικρές αυξήσεις ακριβείας οπότε δεν θα τη χρησιμοποιούσα.

Δίνονται ως παραδείγματα εικόνες από το αντίστοιχο Jupiter notebook:



5.2.a.2 Δήμητρα

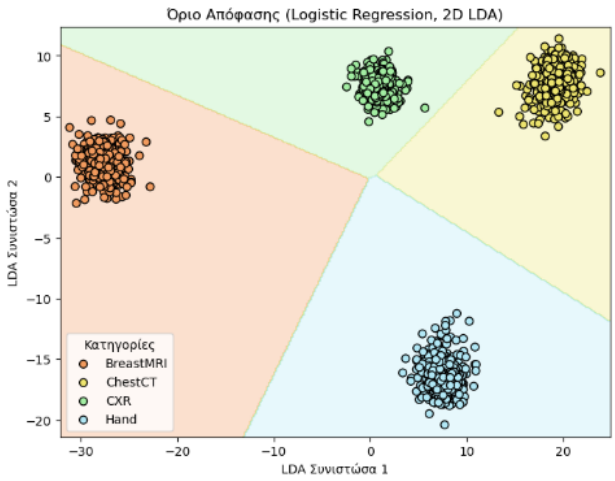
Για τις κατηγορίες BreastMRI, ChestCT, CXR, Hand έχουμε τον ακόλουθο πίνακα συγκριτικά με τις μεθόδους μείωσης διαστάσεων και την απόδοσή τους στα μοντέλα ταξινόμησης.
Πιο συγκεκριμένα:
Logistic Regression [7] χωρίς μείωση διαστάσεων: ακρίβεια 99.25%
PCA

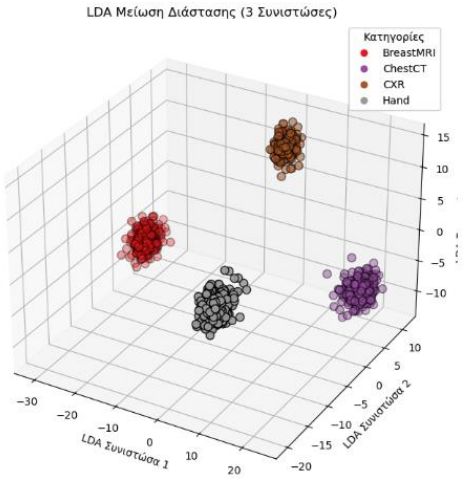
- Πέτυχε ακρίβεια 99.33%, ελαφρώς βελτιωμένη από το baseline χωρίς μείωση.
- Τα σφάλματα ήταν ελάχιστα, κυρίως σε ChestCT και Hand.
- Ο ROC-AUC ήταν 1.0 σε όλες τις κατηγορίες, δείχνοντας τέλεια διάκριση.
- Η βελτιστοποίηση με GridSearchCV δεν άλλαξε την ακρίβεια, επιβεβαιώνοντας τη σταθερότητα του μοντέλου.

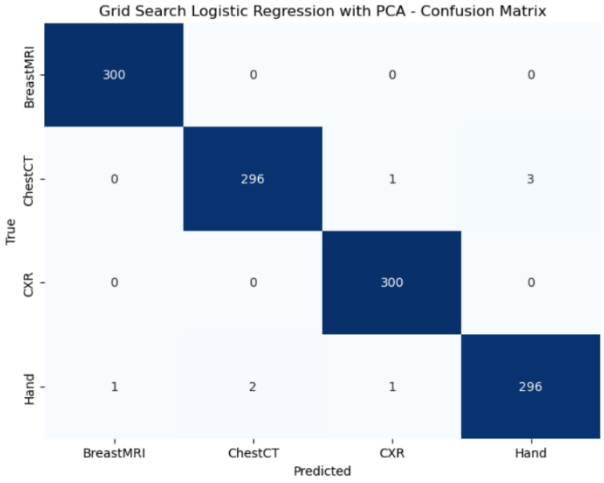
LDA [8]

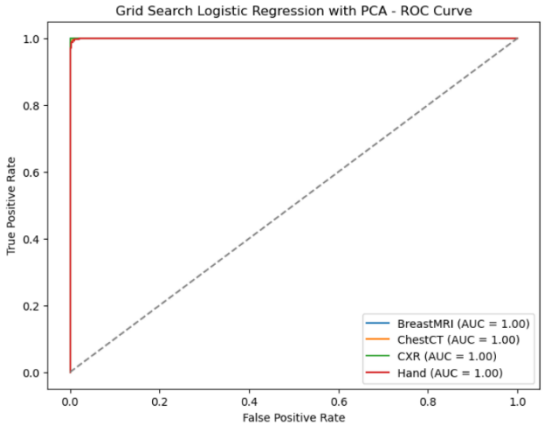
- Η ακρίβεια μειώθηκε σε 97.17%, με ελαφρά βελτίωση σε 97.5% μετά από GridSearchCV.
- Παρατηρήθηκαν περισσότερα λάθη σε ChestCT, CXR, Hand.
- Παρόλο που οι διαστάσεις μειώθηκαν σε 3, η οπτική διάκριση ήταν καλή.
- Η συνολική απόδοση ήταν καλή, αλλά υστερούσε ελαφρώς έναντι του PCA.

Ενδεικτικά από το Jupyter Notebook:







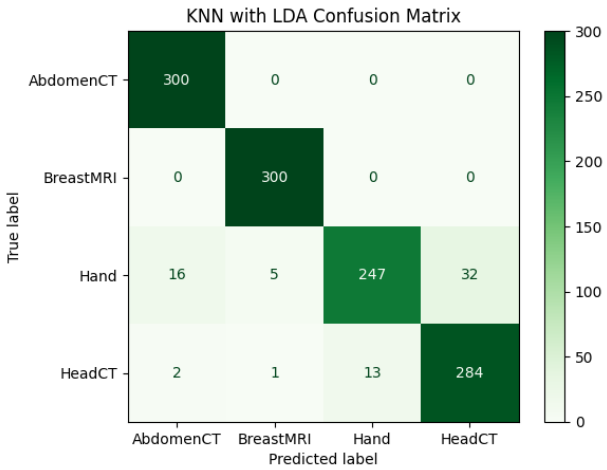


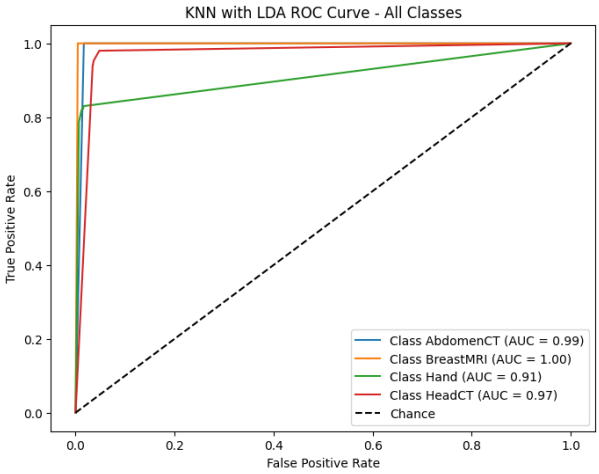
5.2.b K nearest neighbor

5.2.b.1 Οδυσσέας

- Εξαιρετικά αποτελέσματα με ακρίβεια μεγαλύτερη του 0,99 και συνολικά 11 λάθη στο σύνολο των 1200 εικόνων του τεστ σετ.
- Με την με την χρήση της PCA καταφέρνουμε ταυτόχρονα αύξηση της ακρίβειας στο 0,993 και σημαντική μείωση στο υπολογιστικό φορτίο.
- Με αναζήτηση βέλτιστων παραμέτρων μέσω της Grid search η ακρίβεια αυξάνεται στο 0,996 αλλά μαζί της αυξάνονται και οι απαραίτητοι υπολογισμοί. Λόγω της μικρής μόνο αύξησης της ακρίβειας δεν έχει ιδιαίτερο νόημα η χρήση του.
- Εκτέλεσα και μια custom βελτιστοποίηση των υπέρ παραμέτρων της KNN την οποία έγραψα με βοήθεια από διαδικτυακές πηγές, η οποία καταλήγει στο ίδιο συμπέρασμα με την αλλά με μικρότερο υπολογιστικό κόστος για περισσότερα μπορείτε να δείτε στο αντίστοιχο Jupiter notebook.
- Το μοντέλο με μείωση διαστάσεων LDA χάνει λίγο από την ακρίβεια = 0,9425 αλλά παραμένει πολύ αξιόλογο.
- Οι κλάσεις που το δυσκολεύουν περισσότερο είναι αυτές του χεριού και του κεφαλιού, πιθανώς λοιπόν οι τιμές των πίξελ σε αυτές τις εικόνες να μην ακολουθούν τόσο κανονική κατανομή και να μην είναι τόσο κατάλληλες για τη μέθοδο LDA.
- Η αναζήτηση για βέλτιστους υπερ παραμέτρους δεν ευδοκίμησε καθώς η ακρίβεια παρέμεινε f πρακτικά σταθερή (=0,93920 και το μόνο που αυξήθηκε είναι το υπολογιστικό κόστος.

Παρατίθενται εικόνες από αντίστοιχο Jupiter notebook:





5.2.b.2 Δήμητρα

5.2.c K nearest neighbor [9] χωρίς μείωση διαστάσεων: ακρίβεια 99.25%

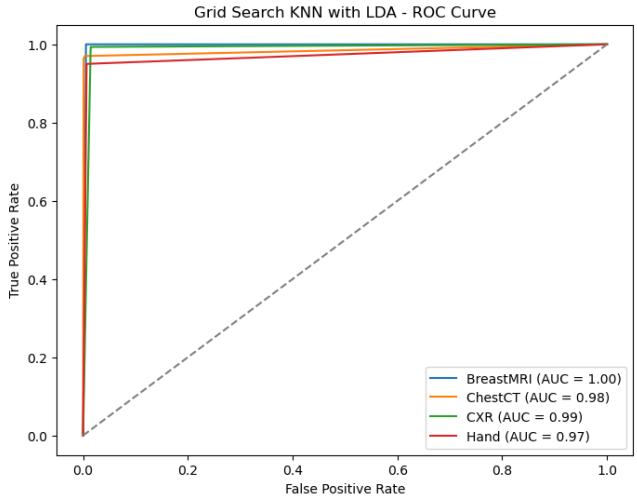
PCA

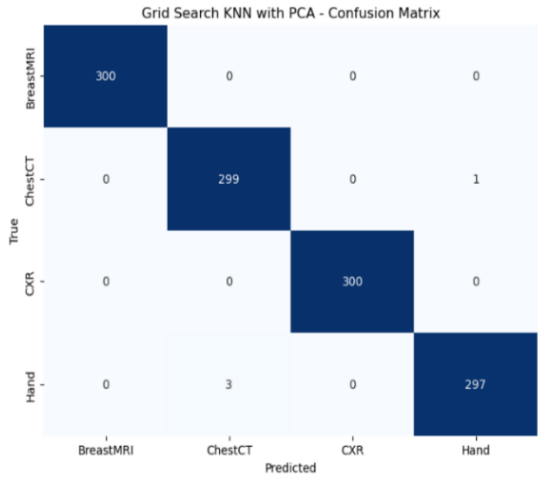
- Η ακρίβεια ανέβηκε από 99.25% σε 99.42%, και έφτασε 99.67% με GridSearchCV.
- Παρατηρήθηκαν λίγα σφάλματα, κυρίως σε ChestCT και Hand.
- Ο αλγόριθμος με PCA απέδωσε εξαιρετικά μετά το tuning.

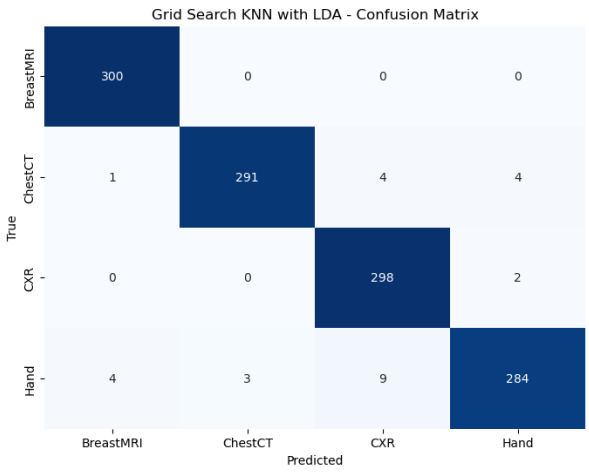
LDA

- Η ακρίβεια ήταν 97.67%, οριακά βελτιωμένη σε 97.75% μετά από GridSearchCV.
- Λάθη εντοπίστηκαν σε ChestCT, CXR, Hand.
- Η ROC-AUC ήταν χαμηλότερη για την κατηγορία Hand (0.97).
- Συνολικά, η απόδοση με LDA ήταν υποδεέστερη από αυτή του PCA.

Ενδεικτικά από το Jupiter Notebook





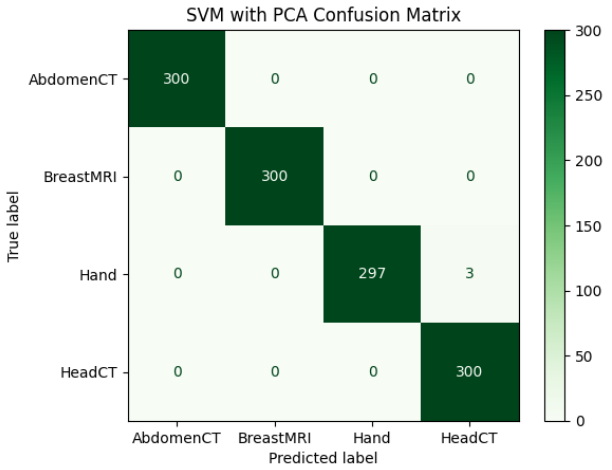


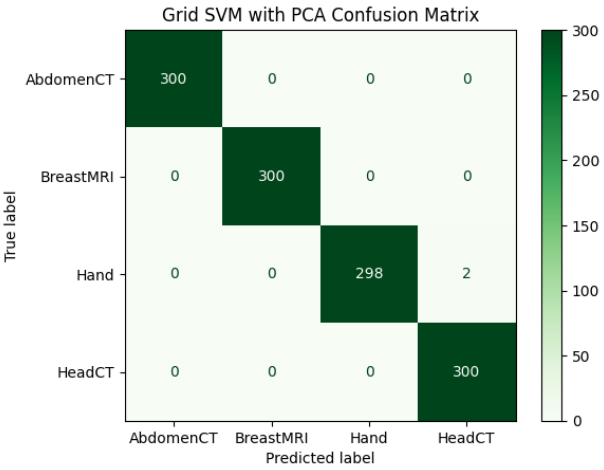
5.2.d Support vector machine

5.2.d.1 Οδυσσέας

- Το συγκεκριμένο μοντέλο αποδείχθηκε εξαιρετικό και ταχύτατο. Η πρώτη εκτέλεση δίχως καμία μείωση των διαστάσεων κατάφερε ακρίβεια ίση με 0,9975!
- Με μείωση των διαστάσεων με τη μέθοδο PCA η ακρίβεια παρέμεινε ολόιδια (0,9975) και ταυτόχρονα μειώθηκε σημαντικά και ο υπολογιστικός φόρτος. Φυσικά και αξίζει η χρήση της.
- Βελτιστοποίηση των υπέρ παραμέτρων με τη μέθοδο Grid Search έφερε μια αύξηση στην ακρίβεια η οποία πλέον είναι ίση με 0,9983 αλλά φυσικά αύξησε το υπολογιστικό φορτίο με αποτέλεσμα να μην αξίζει η χρήση της.
- Και με τη χρήση της μεθόδου LDA για τη μείωση των διαστάσεων το μοντέλο πετυχαίνει ακρίβεια 0,9425 ελάχιστα μικρότερη από αυτήν με το μοντέλο PCA. Πάλι και εδώ τη μεγαλύτερη δυσκολία την αντιμετωπίζει στις κλάσεις Hand και HeadCT.
- Με βελτιστοποίηση των παραμέτρων η ακρίβεια αυξάνεται ελάχιστα στο 0,9441 πράγμα που σημαίνει ότι δεν αξίζει ιδιαίτερα και η χρήση της πιστοποίησης.

Παρατίθενται εικόνες από αντίστοιχο Jupiter notebook:





5.2.d.2 Δήμητρα

Χωρίς μείωση διαστάσεων: ακρίβεια 99.75%

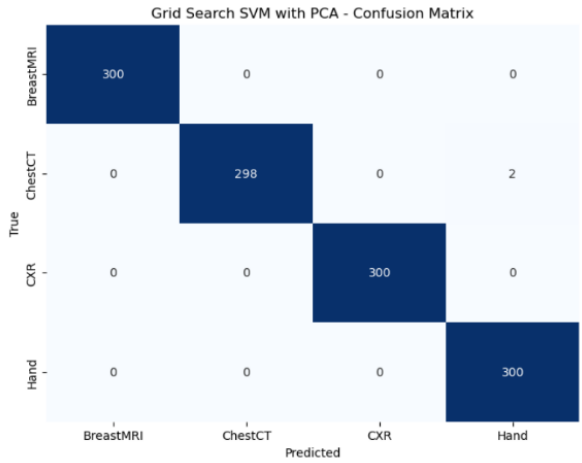
PCA

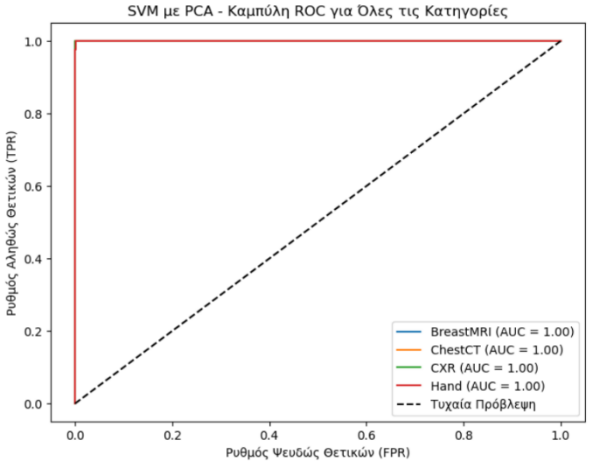
- Πέτυχε ακρίβεια 99.83%, ελαφρώς καλύτερη από χωρίς μείωση.
- Τα λάθη περιορίστηκαν σχεδόν μόνο στην κατηγορία ChestCT.
- Η χρήση GridSearchCV επιβεβαίωσε τη βέλτιστη απόδοση με τον πυρήνα RBF.

LDA

- Η ακρίβεια μειώθηκε σε 97.58%, με μικρή βελτίωση σε 97.67% μετά από tuning.
- Παρουσιάστηκαν λάθη στις ίδιες ευαίσθητες κατηγορίες (ChestCT, CXR, Hand).
- Η LDA πρόσφερε καλή οπτική διάκριση αλλά με μικρό κόστος στην ακρίβεια.
- Το μοντέλο ήταν λιγότερο σταθερό σε σύγκριση με το PCA.

Ενδεικτικά από το Jupyter Notebook:



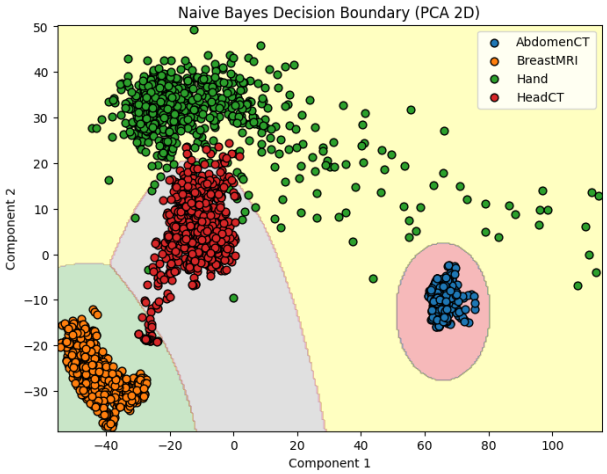


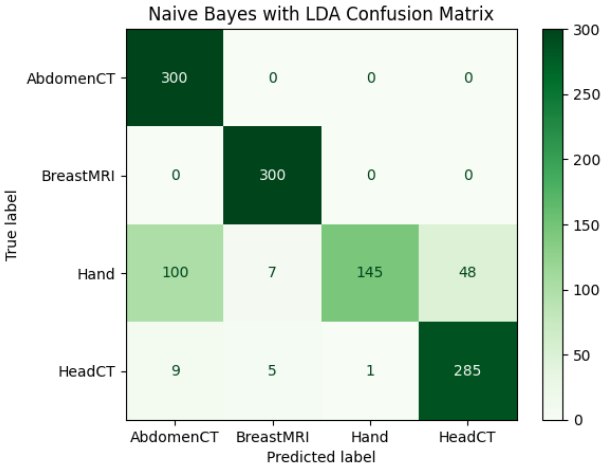
5.2.e Naive bayes

5.2.e.1 Οδυσσέας

- Χωρίς κάποια μείωση διαστάσεων το μοντέλο επιτυγχάνει αξιοσημείωτη ακρίβεια 0,9975. Αυτό πρακτικά μεταφράζεται σε 3 μόνο λάθη από τις 1200 φωτογραφίες του test set όλα στην κλάση HeadCT
- Με χρήση της μεθόδου PCA για μείωση των διαστάσεων η ακρίβεια πέφτει στο 0,985, μικρή μείωση πράγμα που σημαίνει ότι δικαιολογείται η χρήση της μεθόδου καθώς συνοδεύεται με μείωση του υπολογιστικού φορτίου
- Η προσπάθεια βελτιστοποίησης των παραμέτρων δεν πετυχαίνει αύξηση της ακρίβειας = 0,985 και έτσι δεν δικαιολογεί την χρησιμοποίηση της.
- Με μείωση των διαστάσεων χρησιμοποιώντας τη μέθοδο LDA, η ακρίβεια πέφτει σημαντικά = 0,8583. Η κλάση Hand είναι αυτή που επηρεάζεται περισσότερο αρνητικά και περιέχει τις περισσότερες λάθος εκτιμήσεις από το μοντέλο.
- Βελτιστοποίηση παραμέτρων το μοντέλο που χρησιμοποιεί LDA βελτιώνεται σημαντικά με την ακρίβεια να ανεβαίνει στο 0,9425 και τα λάθη να μειώνονται σημαντικά.

Παρατίθενται εικόνες από αντίστοιχο Jupiter notebook:





5.2.e.2 Δήμητρα

Naïve bayes [10] Χωρίς μείωση διαστάσεων: Ακρίβεια 96.83%

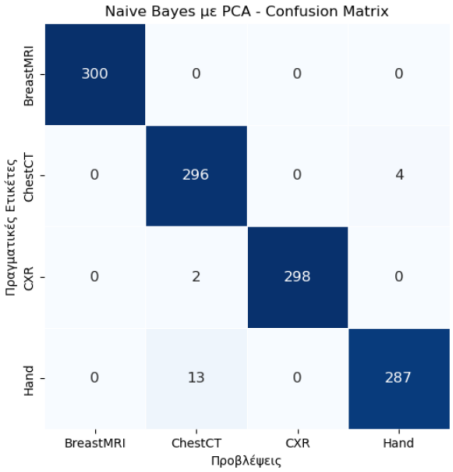
PCA

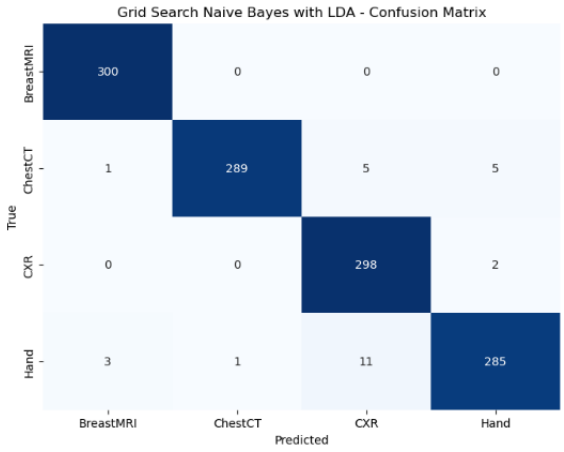
- Η ακρίβεια αυξήθηκε από 96.83% σε 98.42%.
- Το confusion matrix έδειξε λιγότερα σφάλματα από ό,τι χωρίς μείωση.
- Η βελτιστοποίηση (var_smoothing) δεν άλλαξε την απόδοση, σταθερή στο 98.4%.

LDA

- Η ακρίβεια έφτασε 98.17%, και έπεσε ελαφρώς σε 97.67% με tuning.

- Παρατηρήθηκαν πιο συχνά λάθη σε ChestCT, CXR, Hand.
- Αν και προσέφερε καλή οπτική διάκριση, η συνολική απόδοση υστέρησε του PCA.



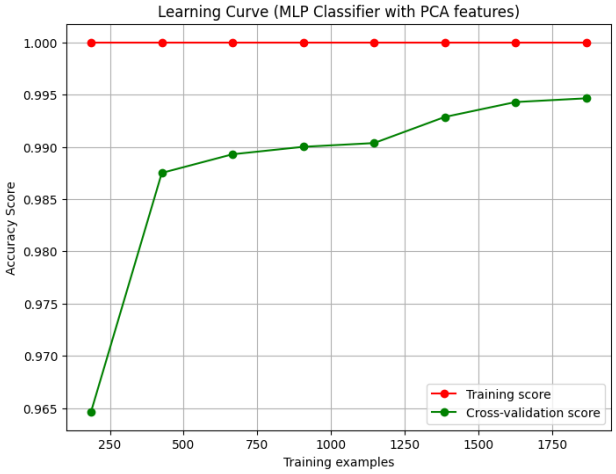


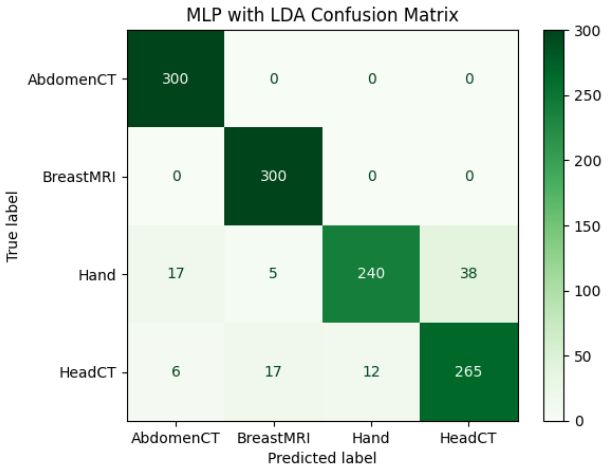
5.2.f Multi-Layer Perceptron neural network

5.2.f.1 Οδυσσέας

- Για χωρίς καμία μείωση διαστάσεων η μέθοδος πετυχαίνει ακρίβεια 0.993 για το υπολογιστικό κόστος τουλάχιστον σε χρόνο φάνηκε να μην διαφέρει ιδιαίτερα από τις άλλες μεθόδους.
- Η προσπάθεια βελτιστοποίησης των παραμέτρων όμως αύξησε κατακόρυφα το υπολογιστικό κόστος το οποίο χρονικά έφτασε τα 60 λεπτά περίπου με την ακρίβεια να αγγίζει το 0,994, ελάχιστη λοιπόν η διαφορά της και σίγουρα δεν ναι δικαιολογεί την χρήση της.
- Με μείωση διαστάσεων χρησιμοποιώντας τη μέθοδο PCA, το νευρωνικό δίκτυο MLP για πετυχαίνει ακρίβεια 0,994 μετά από 34 iterations loss =0.0009947, ακόμα όμως και από τη $3^{η}$ iteration, το loss=0.08. Επομένως πολύ γρήγορα φτάνει σε πολύ καλά επίπεδα. Και ταυτόχρονα δεν είναι ιδιαίτερα υπολογιστικά «βαρύ».
- Προσπάθεια βελτιστοποίησης υπέρ παραμέτρων στο μοντέλο με μείωση διαστάσεων PCA δίνει τα αποτελέσματα {'activation': 'relu', 'alpha': 0.0001, 'hidden_layer_sizes': (100,), 'learning_rate': 'constant', 'max_iter': 300, 'solver': 'sgd'} και πρακτικά δεν αλλάζει την τιμή της ακρίβειας, υπάρχει μεγάλη διαφορά όμως στο υπολογιστικό κόστος καθώς έτρεξε σε χρόνο γύρω στα 10 λεπτά σε αντίθεση με τα 60 λεπτά που χρειάστηκε χωρίς μείωση διαστάσεων.
- Οι μείωση των διαστάσεων με τη χρήση της μεθόδου LDA επηρεάζει ελαφρώς αρνητικά την ακρίβεια του μοντέλου =0.92.
- Η προσπάθεια βελτιστοποίησης του μοντέλου που χρησιμοποιεί τη μέθοδο LDA καταφέρνει να αυξήσει ελαφρώς την ακρίβεια στην τιμή 0,93. Η μικρή αυτή αύξηση δεν δικαιολογεί το αυξημένο υπολογιστικό κόστος που έρχεται μαζί της. Και πάλι όμως η μείωση του υπολογιστικού κόστους ήταν μεγάλη με το μοντέλο να τρέχει σε συγκρίσιμους χρόνους με τη μέθοδο PCA γύρω στα 10 λεπτά.

Παρατίθενται εικόνες από αντίστοιχο Jupiter notebook:





5.2.f.2 Δήμητρα

Αρχική αρχιτεκτονική (baseline):

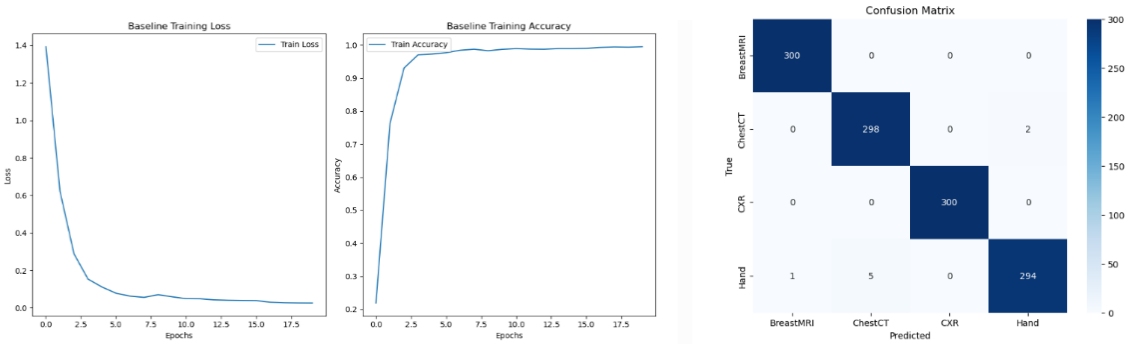
- Τα δεδομένα εκπαίδευσης και δοκιμής μετατράπηκαν σε PyTorch tensors για να χρησιμοποιηθούν στο νευρωνικό δίκτυο, χωρίς να γίνει μείωση διαστάσεων.
- Αξιοποιήθηκαν δύο κρυφά επίπεδα των 256 νευρώνων, με ReLU και dropout 0.3. Η επιλογή της συγκεκριμένης αρχιτεκτονικής βασίστηκε στην ανάγκη για μια σχετικά απλή αλλά αποδοτική λύση για την κατηγοριοποίηση των εικόνων.
- Η εκπαίδευση πραγματοποιήθηκε για 20 εποχές
- Ακρίβεια στο test set: 99.17% . Πολύ ικανοποιητική για απλό MLP, με ελάχιστα σφάλματα (κυρίως σε ChestCT και Hand).
- ROC-AUC: 1.0 για όλες τις κατηγορίες
- Δείχνει ότι το μοντέλο διαχωρίζει άριστα τις κλάσεις.
- Πειραματισμός με υπερπαραμέτρους:
- Δοκιμάστηκαν διαφορετικά μεγέθη κρυφών επιπέδων, συναρτήσεις ενεργοποίησης (ReLU, Sigmoid, Tanh), dropout, batch norm, learning rates.
- Χρήση GridSearch & Stratified Cross-Validation (5 folds):
- Εξασφαλίστηκε σταθερότητα και γενίκευση σε διαφορετικά subsets.

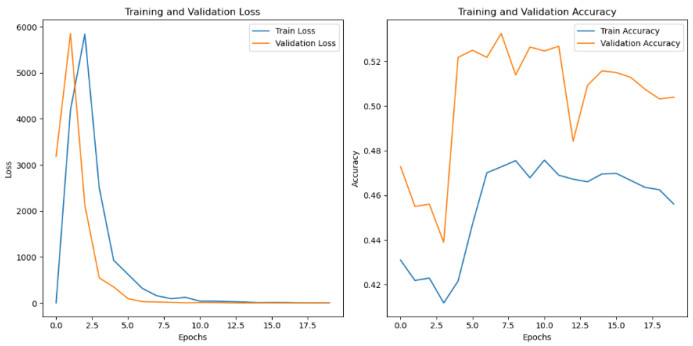
Βελτιστοποιημένο μοντέλο:

- Πειραματιστήκαμε με κρυφά επίπεδα με διαφορετικά μεγέθη νευρώνων, ρυθμούς dropout, batch normalization και διαφορετικές συναρτήσεις ενεργοποίησης (ReLU, Sigmoid, Tanh).
- Κρυφά επίπεδα: [512, 256, 128], χωρίς dropout/batch norm, learning rate 0.001, 20 εποχές.
- Ακρίβεια CV: 99.70%, Ακρίβεια στο test set: 99.33%.

Συνολικά, το fine-tuned MLP ξεπέρασε κατά πολύ το baseline, με άριστη γενίκευση και εξαιρετική απόδοση ταξινόμησης.

Ενδεικτικά από το Jupyter Notebook:






```
Best Params: {'hidden_sizes': [512, 256, 128], 'dropout_rate': 0.0, 'use_batchnorm': False, 'lr': 0.001, 'num_epochs': 20}
Best Cross-Validation Accuracy: 99.70%
```

5.2.g Convolutional neural network

5.2.g.1 Οδυσσέας

- Επιλέχθηκε η βιβλιοθήκη tensorflow για την δημιουργία του CNN, παρότι έγιναν δοκιμές και με την Pytorch, η επιλογή έγινε από προσωπική προτίμηση και όχι με κάποιο αντικειμενικό κριτήριο.
- Η αρχιτεκτονική του μοντέλου βασίστηκε σε ένα βασικό πρότυπο από το βιβλίο “Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow” [11].

model = Sequential([Conv2D(32, (3,3), activation='relu', input_shape=input_shape)MaxPooling2D(2,2),Conv2D(64, (3,3), activation='relu'),MaxPooling2D(2,2),Flatten(),Dense(128, activation='relu'),Dropout(0.5),Dense(num_classes, activation='softmax')

- Χρησιμοποιήθηκαν 10 εποχές και από την 3η κιόλας η ακρίβεια είχε γίνει πρακτικά ≈1
- Ταυτόχρονα και ο χρόνος εκτελέσεις ήταν ικανοποιητικός.
- Χρησιμοποιήθηκε και το keras tuner για τη βελτιστοποίηση των υπερ παραμέτρων του μοντέλου όμως όπως είδαμε το μοντέλο ήδη ήταν σε άριστο επίπεδο και έτσι η βελτιστοποίηση δεν απέφερε κάποια αύξηση στην ακρίβεια παρά μόνο αύξησε τον όγκο υπολογισμών. Εξ αυτού η χρήση της δεν έχει κάποιο νόημα.
- Για εκπαιδευτικούς σκοπούς έγινε και μια δοκιμή διαφορετικών configurations.

{'conv1_filters': 32, 'conv2_filters': 64, 'dense_units': 128, 'dropout_rate': 0.5},
{'conv1_filters': 64, 'conv2_filters': 128, 'dense_units': 256, 'dropout_rate': 0.3},
{'conv1_filters': 16, 'conv2_filters': 32, 'dense_units': 64, 'dropout_rate': 0.7},

Παρατίθενται εικόνες από αντίστοιχο Jupiter notebook:

5.2.g.2 Δήμητρα

Αρχιτεκτονική baseline:

- Τα δεδομένα χρησιμοποιήθηκαν χωρίς να γίνει μείωση διαστάσεων.
- Αξιοποιήθηκαν 3 συνελικτικά επίπεδα με φίλτρα (32, 64, 128), ReLU + MaxPooling + Dropout 0.5.Η επιλογή της συγκεκριμένης αρχιτεκτονικής βασίστηκε στην ικανότητά της να μαθαίνει σταδιακά όλο και πιο σύνθετα χαρακτηριστικά των εικόνων. Ακολουθούν 2 πλήρως συνδεδεμένα επίπεδα για την τελική ταξινόμηση.

self.conv1 = nn.Conv2d(1, 32, kernel_size=3, padding=1)
self.conv2 = nn.Conv2d(32, 64, kernel_size=3, padding=1)
self.conv3 = nn.Conv2d(64, 128, kernel_size=3, padding=1)
*self.fc1 = nn.Linear(128 * 8 * 8, 512)*
self.fc2 = nn.Linear(512, num_classes)
self.activation_fn = getattr(nn, activation_fn)()
self.dropout = nn.Dropout(dropout_rate)
self.maxpool = nn.MaxPool2d(2)

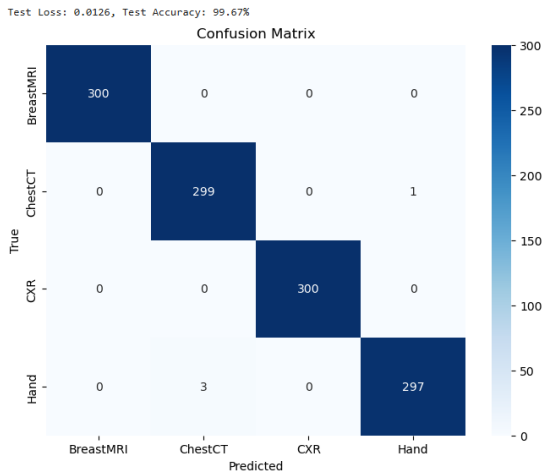
- Εκπαίδευση: 20 εποχές με Adam optimizer και Cross-Entropy Loss.
- Μεγαλύτερη από 99% ακρίβεια εκπαίδευσης από νωρίς (ήδη από τη 2η εποχή).
- Ακρίβεια στο test set: 99.67%, με χαμηλό loss (0.0126), με σφάλματα μόνο σε 1 δείγμα ChestCT και 1 Hand.
- ROC-AUC: 1.0 σχεδόν παντού (0.99 μόνο στο Hand), άριστη ικανότητα διάκρισης κατηγοριών.

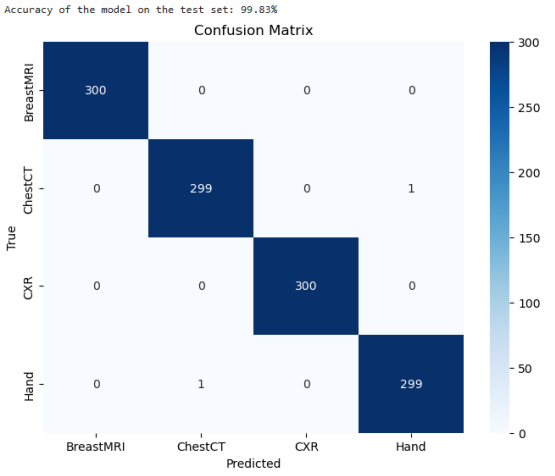
Βελτιστοποίηση:

- Grid Search + Stratified Cross-Validation (5 folds)**:** Εξετάστηκαν αριθμός επιπέδων, νευρώνες, dropout, learning rate, συναρτήσεις ενεργοποίησης.

- Καλύτερες υπερπαράμετροι: Learning_rate: 0.001, dropout_rate: 0.3, layers: 2, neurons/layer: 64, activation: ReLU.
- Τελική ακρίβεια test set: 99.83% Με precision & recall = 1.00 για όλες τις κατηγορίες.

Συνολικά, το CNN ήταν το καλύτερο σε απόδοση μοντέλο, με ελάχιστα λάθη και μέγιστη γενίκευση.





5.3 Σχολιασμός, συγκριτική Ανάλυση για όλα τα μοντέλα

5.3.a Οδυσσέας

- Κατά γενική ομολογία η μέθοδος μείωσης διαστάσεων PCA έκανε τα μοντέλα να εκτελούνται σε σημαντικά μικρότερο χρόνο και είτε επηρέαζε θετικά είτε δεν επηρέαζε την ακρίβεια, πράγμα που στην πραγματικότητα μάλλον την κάνει απαραίτητη για οποιαδήποτε εφαρμογή.
- Ταυτόχρονα και η μέθοδος LDA έδωσε πολύ καλά αποτελέσματα και η μείωση του υπολογιστικού φορτίου ήταν αισθητή. Η κλάσεις Hand και HeadCT είναι αυτές που την «δυσκόλεψαν» περισσότερο.
- Το μοντέλο SVM είχε πολύ εντυπωσιακά αποτελέσματα ακόμα και χωρίς κάποια βελτιστοποίηση και με ταχύτατους χρόνους εκτέλεσης. Έδωσε την αίσθηση ότι ήταν το μόνο σε παρόμοιο επίπεδο με τα νευρωνικά δίκτυα.
- Τα νευρωνικά δίκτυα MLP και CNN, πέτυχαν εντυπωσιακή ακρίβεια ακόμα και στο basic configuration τους. Το υπολογιστικό φορτίο δεν ήταν ιδιαίτερα μεγαλύτερο από άλλες μεθόδους στην εκδοχή δίχως μείωση διαστάσεων.
- Ή βελτιστοποίηση παραμέτρων για το νευρωνικά δίκτυα MLP δίχως μείωση διαστάσεων επέφερε πολύ μεγάλο υπολογιστικό κόστος!
- Ή βελτιστοποίηση παραμέτρων για το νευρωνικά δίκτυα CNN δίχως μείωση διαστάσεων επέφερε και αυτή υπολογιστικό κόστος, αλλά όχι με τόση μεγάλη διαφορά όπως η αντίστοιχη για τα MLP.

Accuracy	Χωρίς μείωση διαστάσεων	PCA	Grid search PCA	LDA	Grid search LDA
Logisitic regression	0.992	0.992	0.991	0.918	0.936
SVM	0.998	0.998	0.998	0.943	0.944
KNN	0.991	0.993	0.995	0.943	0.940
Naïve Bayes	0.998	0.985	0.985	0.858	0.943
MLP (10 iter.)	0.993	0.994	0.994	0.921	0.930
CNN	1	-	-	-	-

5.3.b Δήμητρα

Test Accuracy	Χωρίς μείωση διαστάσεων	PCA	Grid Search PCA	LDA	Grid Search LDA
Logistic Regression	99.25%.	99.33%	99.33%	97.17%	97.5%.
Support Vector Machines	99.75%	99.83%	99.83%	97.58%	97.67%.
k-Κοντινότεροι Γείτονες (k-NN)	99.25%	99.42%	99.67%	97.67%	97.75%

Naive Bayes	96.83%	98.42%	98.4%	98.17%	97.67%
MLP / MLP (GridSearch)	99.17% / 99.33%	-	-	-	-
CNN / CNN (GridSearch)	99.67% / 99.83%	-	-	-	-

- Όλα τα μοντέλα πέτυχαν υψηλή ακρίβεια, με καλύτερες επιδόσεις τα SVM, CNN και MLP.
- Η εφαρμογή PCA βελτίωσε την ακρίβεια σε όλα σχεδόν τα μοντέλα και μείωσε την πολυπλοκότητα.
- Το CNN είναι το πιο αποδοτικό για εικόνες, αλλά έχει υψηλό υπολογιστικό κόστος.
- Απλά μοντέλα (Logistic, KNN) είχαν καλή απόδοση και είναι κατάλληλα για ταχύτερη ταξινόμηση.
- Η πολυπλοκότητα βελτιώνει την απόδοση, αλλά χρειάζεται σωστή ρύθμιση για αποφυγή υπερπροσαρμογής.
- Η σωστή επιλογή υπερπαραμέτρων και η Cross-Validation οδηγούν σε πιο σταθερά και γενικεύσιμα μοντέλα.
- Η μείωση διαστάσεων βοηθά στη σταθερότητα και απόδοση, χωρίς σημαντική απώλεια πληροφορίας.

5.3.c Βασίλης

	PCA	**LDA**	**Custom Metric**	**Best Recall 0**	**Best Validation Accuracy**
Logistic Regression	Recall_0: 0.67 Test Accuracy: 0.7948717948717948	Recall_0: 0.67 Test Accuracy: 0.8141025641025641	-	-	-
SVM	Recall_0: 0.67 Test Accuracy: 0.7948717948717948	Recall_0: 0.93 Test Accuracy: 0.4807692307692308	-	-	-
K-NN	Recall_0: 0.76 Test Accuracy: 0.7051282051282052	Recall_0: 0.76 Test Accuracy: 0.7435897435897436	-	-	-
Naive Bayes	Recall_0: 0.76 Test Accuracy: 0.6602564102564102	Recall_0: 0.60 Test Accuracy: 0.8076923076923077	-	-	-
MLP	-	-	Test Accuracy: 0.6795 Recall 0: 0.81	Test Accuracy: 34.62% Recall 0: 1.0	-
CNN	-	-	Test Loss: 0.3670, Accuracy: 91.03%, Recall 0: 0.7619	Test Loss: 0.2699, Test Accuracy: 92.31% Recall 0: 0.83	Test Loss: 0.4457, Accuracy: 91.03%, Recall 0: 0.7619

Η αναζήτηση με βάση το recall 0 έγινε επειδή η αναζήτηση του μοντέλου με την μεγαλύτερη ακρίβεια στο σύνολο δοκιμής δεν θα είχε ιδιαίτερο νόημα αν έβγαζε αδίκως άτομα δίχως κακοήθεις καρκίνους καρκινοπαθείς. Ένα τέτοιο μοντέλο θα ήταν επικίνδυνο και δυστυχώς από τα δεδομένα και τα μοντέλα που σχεδιάστηκαν κανένα δεν κατάφερε να φτάσει σε state of the art αποτελέσματα. Το πιο αξιόπιστο είναι το best recall MLP μοντέλο, αλλά έχει μόλις 34% ακρίβεια το οποίο ναι μεν καλύτερο από το τίποτα, δεν είναι ικανοποιητικό. Πολλές φορές μάλιστα τα μοντέλα νευρωνικών δικτύων έβγαζαν αποτελέσματα που ελαχιστοποιούσε το recall 0 για να μεγιστοποιήσει το recall 1 δίχως καμία αλλαγή στον κώδικα καθαυτό αλλα τις υπερπαραμέτρους, και αυτό έκανε τον σχεδιασμό των μοντέλων υπερβολικά κουραστικό και συγχηστικό.
Ιδανικά στο μέλλον θα θέλαμε να είχε γίνει πιο οργανωμένη προσπάθεια αξιοποίησης των στοιχείων από την εξαγωγή στοιχείων ή έστω να βρεθεί κάποιος τρόπος να μπορούν να οπτικοποιηθούν τα δεδομένα με πιο σαφή τρόπο, καθώς ως έχει τα δεδομένα είναι πολύ δύσκολο να αναγνωριστούν. Επίσης, μπορεί να είχε νόημα η χρήση τρισδιάστατων plot ή γενικότερα πιο ενδελεχής αξιοποίηση του PCA και του NDA ή ακόμα να ερευνηθούν διαφορές μεταξύ της αποτελεσματικότηατας των μοντέλων με τα αντίστοιχα σύνολα του BreastMNIST που είχαν αλλες διαστάσεις (πχ 28 ή 112). Παράλληλα, θα μπορούσε να ερευνηθεί περαιτέρω το πώς δουλεύουν οι encoders και να αναζητηθούν περαιτέρουσες τεχνικές διαχείρισης δεδομεών που δεν έχουν καλή ισορροπία.
Τέλος, θεωρούμε πως το ίδιο το σύνολο των δεδομένων θα μπορούσε να είναι καλύτερο. Θα ήταν πολύτιμο να υπήρχαν αναλυτικές εξηγήσεις για το τι αλλαγές έγιναν και γιατί, καθώς και πόσο σκόπιμος ήταν ο χωρισμός των δεδομένων. Επίσης, παρότι οι ακτινογραφίες είναι χωρισμένες σε normal, benign και malignant, το ίδιο το σύνολο δεν έχει ετικέτα για τα benign, το οποίο μπορέι και να διασάφνιζε τις διαφορές μεταξύ των κατηγοριών.

6. References

[1] R. C. H. K. &. A. R. Jalloul, "A Review of Machine Learning Techniques for the Classification and Detection of Breast Cancer from Medical Images.," *Diagnostics,* pp. 13-14, 2023.

[2] M. G. H. K. A. F. Walid Al-Dhabyani, "Dataset of breast ultrasound images," *Data in Brief,* vol. 28, 2020.

[3] S. learn, "StandardScaler," [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.StandardScaler.html.

[4] Γ. Παναγάκης, L06-Διαστατικότητα των δεδομένων και PCA (Σειμειώσεις μαθήματος "Μηχανική μάθηση"), 2025.

[5] A. Géron, Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow, O'Reilly, 2022, p. 247.

[6] R. Pramoditha, "Towards data science," [Online]. Available: https://towardsdatascience.com/lda-is-highly-effective-than-pca-for-dimensionality-reduction-in-classification-datasets-4489eade632/.

[7] S. learn, "Logistic regression," [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html.

[8] S. learn, "LDA_QDA," [Online]. Available: https://scikit-learn.org/stable/modules/lda_qda.html .

[9] W3schools, "python_ml_knn," [Online]. Available: https://www.w3schools.com/python/python_ml_knn.asp.

[10] S. learn, "Naive Bayes," [Online]. Available: https://scikit-learn.org/stable/modules/naive_bayes.html.

[11] A. Géron, Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow, O'Reilly, 2022, p. 312.