**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. 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.

**Medical MNIST**

Στο Medical MNIST, παρατηρήθηκαν βασικά στατιστικά των pixel, με μέση τιμή 94,18 και διάμεσο 123. Η τυπική απόκλιση ήταν 78,6, που δείχνει σημαντική διακύμανση στις τιμές. Οι τιμές των pixel κυμαίνονται από 0 (σκοτάδι) έως 255 (πλήρης φωτεινότητα).

**Κανονικοποίηση/Τυποποίηση χαρακτηριστικών:**

Εφαρμόσαμε τη μέθοδο StandardScaler για τυποποίηση των χαρακτηριστικών, με αποτέλεσμα κάθε pixel να έχει μέση τιμή 0 και τυπική απόκλιση 1. 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.

**Επιλογή και μετασχηματισμός χαρακτηριστικών:**

Εφαρμόσαμε τον μετασχηματισμό διακύμανσης (VarianceThreshold) για να αφαιρέσουμε χαρακτηριστικά με μικρή διακύμανση (κάτω από το 0.01), κρατώντας μόνο τα σημαντικά χαρακτηριστικά που επηρεάζουν την ανάλυση των εικόνων. 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.

**Διαχωρισμός σε σύνολα εκπαίδευσης, επικύρωσης και ελέγχου**

Σχετικά με τις κατηγορίες BreastMRI, ChestCT, CXR και Hand, επιλέξαμε 1000 εικόνες από κάθε κατηγορία, προκειμένου να διατηρηθεί μια ισχυρή και ισότιμη εκπροσώπηση των δεδομένων και να αποφευχθεί η μεροληψία λόγω ασυμμετρίας στα δεδομένα. Στη συνέχεια, τα δεδομένα χωρίστηκαν σε δύο σύνολα: το σύνολο εκπαίδευσης (70%) και το σύνολο ελέγχου3 (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.

**Μείωση Διαστάσεων και Κλασικοί Αλγόριθμοι Ταξινόμησης:**

**PCA & Λογιστική Παλινδρόμηση (Logistic Regression)**

Αρχικά εφαρμόσαμε Logistic Regression απευθείας στα δεδομένα χωρίς μείωση διάστασης, για να αξιολογήσουμε την απόδοση του μοντέλου. Η ακρίβεια ήταν **99.25%.** Στο δεύτερο πείραμα, χρησιμοποίησα την τεχνική PCA (Principal Component Analysis) για να μειώσουμε τις διαστάσεις των δεδομένων πριν την εκπαίδευση του μοντέλου. Το PCA βοήθησε να μειωθεί η πολυπλοκότητα των δεδομένων και η εκπαίδευση ήταν ταχύτερη χωρίς να επηρεαστεί η απόδοση, παραμένοντας η ακρίβεια υψηλή **99.33%** και ελάχιστα σφάλματα στο confusion matrix (**ChestCT, Hand**). Οπτικοποιήσαμε τα όρια απόφασης για το training set. Έπειτα, κάναμε χρήση του GridSearchCV για να βρούμε τις βέλτιστες υπερπαραμέτρους του Logistic Regression με τα δεδομένα που είχαν μειωθεί σε διάσταση με το PCA. Στο GridSearchCV για την παράμετρο C χρησιμοποιήσαμε τις τιμές από το λογαριθμικό διάστημα μεταξύ 10−410^{-4}10−4 και 10410^{4}104, για τον solver επιλέξαμε τις τιμές 'lbfgs', 'saga' και 'liblinear', για την παράμετρο penalty ορίσαμε την τιμή 'l2', και για την παράμετρο max\_iter δοκιμάσαμε τις τιμές 1000, 2500 και 5000. Ο αλγόριθμος μάς έδωσε τον καλύτερο συνδυασμό παραμέτρων ('C': np.float64(0.23357214690901212), 'max\_iter': 1000, 'penalty': 'l2', 'solver': 'lbfgs') επιτυγχάνοντας την ίδια ακρίβεια (**99.33%)** . Ο Confusion Matrix δείχνει σχεδόν τέλεια ακρίβεια και το ROC-AUC για όλες τις κατηγορίες είναι 1.0, δείχνοντας ότι το μοντέλο κάνει εξαιρετική διάκριση μεταξύ των κατηγοριών.

**PCA & Support Vector Machines**

Εφαρμόσαμε τον ταξινομητή SVM απευθείας στα δεδομένα χωρίς μείωσης διαστάσεων. Η ακρίβεια ήταν υψηλή **99.75%.** Με PCAδόθηκε η δυνατότητα στο μοντέλο να επικεντρωθεί στις κύριες διαστάσεις των δεδομένων και να επιτύχει υψηλότερη ακρίβεια **99.83%** και ελάχιστα σφάλματα στο confusion matrix (**ChestCT**). Οπτικοποιήσαμε τα όρια απόφασης για το training set. Χρησιμοποίησαμε GridSearchCV για την εύρεση των βέλτιστων υπερπαραμέτρων του SVM. Δοκιμάσαμε για την παράμετρο C τις τιμές 0.1, 1, 10, 100, 1000, οι οποίες ελέγχουν την ποινή για τα λάθη του μοντέλου. Για την παράμετρο gamma, δοκιμάσαμε την τιμή 'scale' (προεπιλογή) καθώς και τιμές όπως 0.0001, 0.001, 0.01, 0.1, 1, οι οποίες ελέγχουν την απόσταση του πυρήνα. Για την παράμετρο kernel, δοκιμάσαμε δύο τύπους πυρήνα: 'rbf' και 'linear', οι οποίοι καθορίζουν τη μορφή του πυρήνα που χρησιμοποιείται στο SVM. Ο αλγόριθμος μάς έδωσε τον καλύτερο συνδυασμό παραμέτρων {'C': 10, 'gamma': 'scale', 'kernel': 'rbf'} επιτυγχάνοντας την ίδια ακρίβεια (**99.83%)**

**PCA & k-Κοντινότεροι Γείτονες (k-NN)**

Εφαρμόσαμε το KNN χωρίς μείωση διαστάσεων και έδωσε ακρίβεια **99.25%.** Στη συνέχεια, με την εφαρμογή PCA το μοντέλο πέτυχε ακρίβεια **99.42%** με καλύτερη διάκριση των κατηγοριών (όπως μας δείχνει το classification report). Στο confusion matrix βλέπουμε ελάχιστα σφάλματα στις κατηγορίες **ChestCT** και **Hand**. Οπτικοποιήσαμε τα όρια απόφασης για το training set. Χρησιμοποιήσαμε GridSearchCV για δοκιμάζοντας υπερπαραμέτρους για n\_neighbors (τιμές από 2 έως 29, με βήμα 1, οι οποίες καθορίζουν τον αριθμό των γειτόνων που χρησιμοποιούνται για την ταξινόμηση). Για weights, δοκιμάσαμε τις επιλογές 'uniform' (όλες οι γειτονικές κλάσεις έχουν την ίδια βαρύτητα) και 'distance' (η βαρύτητα των γειτόνων εξαρτάται από την απόσταση τους από το δεδομένο σημείο). Για metric, δύο μετρικές: 'euclidean' (ευκλείδεια απόσταση) και 'manhattan' (μανχαττανική απόσταση), οι οποίες καθορίζουν πώς μετράμε την απόσταση μεταξύ των σημείων. Ο αλγόριθμος μάς έδωσε τον καλύτερο συνδυασμό παραμέτρων {'metric': 'euclidean', 'n\_neighbors': np.int64(2), 'weights': 'distance'} επιτυγχάνοντας υψηλότερη ακρίβεια (**99.67%**)κάνοντας λιγότερα λάθη στις κατηγορίες ChestCT και Hand (confusion matrix).

**PCA & Naive Bayes**

Χρησιμοποιήσαμε τον ταξινομητή Naive Bayes και το μοντέλο εκπαιδεύτηκε αρχικά χωρίς την εφαρμογή PCA, επιτυγχάνοντας ακρίβεια περίπου **96.83%.** Στη συνέχεια, εφαρμόσαμεPCA, διατηρώντας τις πιο σημαντικές πληροφορίες. Η χρήση του PCA οδήγησε σε αύξηση της απόδοσης του μοντέλου, με την ακρίβεια να φτάνει το **98.42%** και στο confusion matrix έδειξε ότι το μοντέλο έκανε κάποια λάθη στις κατηγορίες **ChestCT, CXR, Hand**. Οπτικοποιήσαμε τα όρια απόφασης για το training set. Επίσης χρησιμοποιήθηκε το εργαλείο GridSearchCV για την αναζήτηση των βέλτιστων υπερπαραμέτρων του μοντέλου, με επικέντρωση στην παράμετρο var\_smoothing του Naive Bayes, που βελτιώνει τη σταθερότητα του μοντέλου σε δεδομένα με μικρές διακυμάνσεις. Η βέλτιστη παράμετρος που βρέθηκε ήταν var\_smoothing = 2.78 \times 10^{-6}. Ωστόσο, η ακρίβεια παρέμεινε στα ίδια επίπεδα, περίπου **98.4%**.

**LDA & Λογιστική Παλινδρόμηση (Logistic Regression)**

Χρησιμοποιήσαμε Logistic Regression χωρίς καμία τεχνική μείωσης διάστασης, με αποτέλεσμα μια ακρίβεια **99.25%** στο test set. Στη συνέχεια, την τεχνική Linear Discriminant Analysis (LDA). Μετά τη μείωση διάστασης, το μοντέλο Logistic Regression πέτυχε ακρίβεια **97.17%** (κάνοντας λάθη στις κατηγορίες **ChestCT, CXR, Hand**), το οποίο είναι λίγο χαμηλότερο σε σχέση με το πρώτο πείραμα, αλλά επιτρέπει καλύτερη διακριτικότητα μεταξύ των κατηγοριών, όπως φαίνεται και στο διάγραμμα 3D που δημιουργήσαμε καθώς και στα όρια απόφασης για το training set. Στη συνέχεια, χρησιμοποιήσαμε Grid SearchCV για την εύρεση των καλύτερων υπερπαραμέτρων του μοντέλου. Για την παράμετρο C ως logarithmic scale δοκιμάσαμε τιμές από 10−410^{-4}10−4 έως 10410^{4}104 σε 20 σημεία. Επιπλέον, εξετάσαμε δύο διαφορετικούς solvers: lbfgs και saga. Τέλος, ορίσαμε διάφορες τιμές για το max\_iter (100, 1000, 2500, και 5000), για να αξιολογήσουμε την απόδοση του μοντέλου με διαφορετικούς αριθμούς επαναλήψεων κατά την εκπαίδευση. Το Grid SearchCV με τις βέλτιστες παραμέτρους ({'C': np.float64(0.0001), 'max\_iter': 100, 'solver': 'lbfgs'}) βελτίωσε ελαφρώς την ακρίβεια, φτάνοντας στο **97.5%.** Συνολικά, τα αποτελέσματα αποδεικνύουν ότι η συνδυασμένη χρήση της LDA με την Logistic Regression και η βελτιστοποίηση υπερπαραμέτρων μέσω Grid SearchCV παρέχουν ένα ισχυρό μοντέλο ταξινόμησης με υψηλή απόδοση, χωρίς όμως να παρουσιάζεται σημαντική βελτίωση σε σχέση με την αρχική προσέγγιση Logistic Regression χωρίς LDA. Η μειωμένη ακρίβεια μπορεί να οφείλεται στη μείωση του αριθμού των διαστάσεων, καθώς η LDA έχει περιορίσει τις πληροφορίες στις 3 συνιστώσες που θεωρεί πιο διακριτικές. Αυτό μπορεί να οδηγήσει σε μια μικρή σύγχυση στα όρια μεταξύ κάποιων κατηγοριών, αλλά η γενική διάκριση παραμένει πολύ καλή.

**LDA & Support Vector Machines**

Εκπαιδεύσαμε το SVM χωρίς μείωση διαστάσεων, με ακρίβεια **99.75%** και άριστη διάκριση όλων των κατηγοριών. Στη συνέχεια, χρησιμοποιήσαμε LDA για μείωση των διαστάσεων, η οποία προσέφερε καλύτερη διάκριση των κατηγοριών, αν και η ακρίβεια μειώθηκε ελαφρώς στο **97.58%** (με κάποια λάθη στις κατηγορίες ChestCT, CXR, Hand). Οπτικοποιήσαμε τα όρια απόφασης, τα οποία δείχνουν σαφή διάκριση μεταξύ των κατηγοριών. Mέσω GridSearchCV βρήκαμε τις βέλτιστες παραμέτρους για τις τιμές που δοκιμάσαμε [**C** (0.1, 1, 10, 100, 1000), Gamma ('scale', 0.0001, 0.001, 0.01, 0.1, 1), Kernel (rbf και linear). Βέλτιστες βρέθηκαν οι {'C': 0.1, 'gamma': 'scale', 'kernel': 'rbf'} με ακρίβεια **97.67%**. Παρατηρήθηκαν παρόμοια λάθη, όπως προηγουμένως, στις κατηγορίες ChestCT, CXR, Hand).

**LDA & k-Κοντινότεροι Γείτονες (k-NN)**

Εκπαιδεύσαμε το ΚΝΝ χωρίς μείωση διαστάσεων, με ακρίβεια **99.25%** . Στη συνέχεια, χρησιμοποιήσαμε LDA για μείωση των διαστάσεων, η οποία έδωσε ακρίβεια **97.67%** (με κάποια λάθη στις κατηγορίες ChestCT, CXR, Hand). Οπτικοποιήσαμε τα όρια απόφασης, τα οποία δείχνουν σαφή διάκριση μεταξύ των κατηγοριών. Mέσω GridSearchCV βρήκαμε τις βέλτιστες παραμέτρους για τις τιμές που δοκιμάσαμε ['n\_neighbors': np.arange(2, 30, 1), 'weights' ( 'uniform', 'distance'), 'metric' ('euclidean', 'manhattan')Βέλτιστες βρέθηκαν οι {'metric': 'euclidean', 'n\_neighbors': np.int64(2), 'weights': 'uniform'} με ακρίβεια **97.75%** Παρατηρήθηκαν ομοίως λάθη, όπως προηγουμένως, στις κατηγορίες ChestCT, CXR, Hand. Βλέπουμε και στην ROC-AUC πως η AUC για την κατηγορία Hand είναι συγκριτικά η μικρότερη από τις 3 κατηγορίες, συνεπώς το μοντέλο δυσκολεύεται λίγο. Έχουμε μέχρι στιγμής λίγο μικρότερες τιμές AUC για τις κατηγορίες μας σε σχέση με όλα τα προηγούμενα μοντέλα και τεχνικές μείωσης διαστάσεων που δοκιμάσαμε (BreastMRI: 1.0, ChestCT: 0.98, CXR: 0.99, Hand: 0.97)

**LDA & Naive Bayes**

Εκπαιδεύσαμε το Naïve Bayes χωρίς μείωση διαστάσεων, με ακρίβεια **96.83%.** Στη συνέχεια, χρησιμοποιήσαμε LDA για μείωση των διαστάσεων, η οποία έδωσε ακρίβεια **98.17%** (με κάποια λάθη στις κατηγορίες ChestCT, CXR, Hand). Οπτικοποιήσαμε τα όρια απόφασης, τα οποία δείχνουν σαφή διάκριση μεταξύ των κατηγοριών. Mέσω GridSearchCV βρήκαμε τη βέλτιστη τιμή για την παράμετρο var\_smoothing (1.0) με ακρίβεια **97.67%** Παρατηρήθηκαν περισσότερα λάθη, στις κατηγορίες ChestCT, CXR, Hand με τη var\_smoothing.

**t-SNE**

Εφαρμόζοντας την τεχνική t-SNE για μείωση διαστάσεων, απλώς για οπτικοποίηση, παρατηρήσαμε ότι οι κατηγορίες διαχωρίζονται σαφώς σε δύο διαστάσεις για τo σύνολο εκπαίδευσης.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 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% |

Συνολικά, η **PCA** φαίνεται να είναι πιο αποδοτική για την πλειονότητα των μοντέλων, ενώ η **LDA** προσφέρει καλύτερη διάκριση για τα δεδομένα αλλά σε βάρος της ακρίβειας σε συγκεκριμένες κατηγορίες. Στην πραγματικότητα, με την LDA αν χάνονται χρήσιμες πληροφορίες, η ακρίβεια μπορεί να μειωθεί. Τα μοντέλα με PCA συχνά διατηρούν περισσότερη πληροφορία (ή τουλάχιστον, την πιο "γενική" πληροφορία), και γι' αυτό μπορεί να οδηγούν σε υψηλότερη ακρίβεια.

**Βαθιά Νευρωνικά Δίκτυα**

**Απλό πολυεπίπεδο νευρωνικό δίκτυο (MLP)**

Τα δεδομένα εκπαίδευσης και δοκιμής μετατράπηκαν σε PyTorch tensors για να χρησιμοποιηθούν στο νευρωνικό δίκτυο. Το πρώτο (baseline) μοντέλο που δημιουργήσαμε ήταν ένα απλό MLP με δύο κρυφά επίπεδα των 256 νευρώνων το καθένα. Η επιλογή της συγκεκριμένης αρχιτεκτονικής βασίστηκε στην ανάγκη για μια σχετικά απλή αλλά αποδοτική λύση για την κατηγοριοποίηση των εικόνων. Το μοντέλο εκπαιδεύτηκε με την ReLU συνάρτηση ενεργοποίησης και χρησιμοποιήθηκε dropout 0.3 για αποφυγή υπερπροσαρμογής (overfitting). Η απόδοση στο test set ήταν υψηλή, με την ακρίβεια να φτάνει το 99.17%. Τα classification report, confusion matrix και ROC-AUC επιβεβαίωσαν την καλή απόδοση του μοντέλου, καθώς οι κατηγορίες ταξινομήθηκαν με υψηλή ακρίβεια και ελάχιστα λάθη. Πειραματιστήκαμε με διάφορες υπερπαραμέτρους προκειμένου να βελτιστοποιηθεί η απόδοσή του MLP στην ταξινόμηση ιατρικών εικόνων. Στην αρχιτεκτονική του μοντέλου, χρησιμοποιήσαμε κρυφά επίπεδα με διαφορετικά μεγέθη νευρώνων, ρυθμούς dropout, batch normalization και διαφορετικές συναρτήσεις ενεργοποίησης (ReLU, Sigmoid, Tanh). Χρησιμοποιήσαμε, επίσης, Grid Search για να εξετάσουμε διάφορους συνδυασμούς υπερπαραμέτρων και Stratified Cross-Validation με 5 folds για να διασφαλιστεί η σταθερότητα του μοντέλου σε διαφορετικά υποσύνολα των δεδομένων. Οι καλύτερες υπερπαράμετροι που προέκυψαν ήταν τα κρυφά επίπεδα [512, 256, 128], χωρίς dropout, χωρίς batch normalization, με ρυθμό μάθησης 0.001 και 20 εποχές, καταλήγοντας σε 99.70% ακρίβεια στην Cross-Validation. Αξιολογώντας το μοντέλο στο test set, πέτυχε ακρίβεια 99.33%. Τα αποτελέσματα από το classification report και την confusion matrix δείχνουν εξαιρετική ακρίβεια και πολύ μικρό αριθμό λανθασμένων ταξινομήσεων (στις κατηγορίες ChestCT και Hand) και η ROC curve έδειξε AUC 1.0 για όλες τις κατηγορίες, επιβεβαιώνοντας την άριστη απόδοση του μοντέλου. Συνολικά, η βελτίωση από το baseline μοντέλο αποδεικνύει την αποτελεσματικότητα του πειραματισμού με τις υπερπαραμέτρους και την ικανότητα του μοντέλου να γενικεύει σε νέα δεδομένα.

**Συνελικτικό Νευρωνικό Δίκτυο (CNN) για δεδομένα εικόνας**

Τα δεδομένα εκπαίδευσης και δοκιμής φορτώθηκαν ως tensors με κατάλληλο σχήμα (1 κανάλι, 64x64 pixels) για την είσοδο του CNN. Το baseline μοντέλο αποτελείται από τρία συνελικτικά επίπεδα με αυξανόμενο αριθμό φίλτρων (32, 64, 128), τα οποία ακολουθούνται από λειτουργίες ReLU, max pooling για μείωση διάστασης και dropout 0.5 για αποφυγή υπερπροσαρμογής. Το μοντέλο ολοκληρώνει την ταξινόμηση με δύο πλήρως συνδεδεμένα επίπεδα. Η επιλογή της συγκεκριμένης αρχιτεκτονικής βασίστηκε στην ικανότητά της να μαθαίνει σταδιακά όλο και πιο σύνθετα χαρακτηριστικά των εικόνων. Η εκπαίδευση έγινε για 20 εποχές με χρήση Adam optimizer και cross-entropy loss, πετυχαίνοντας υψηλή ακρίβεια εκπαίδευσης (>99%) ήδη από τις πρώτες εποχές. Η αξιολόγηση στο test set έδειξε μέση ακρίβεια 99.67% και χαμηλή απώλεια (0.0126). Η confusion matrix και το classification report επιβεβαίωσαν την εξαιρετική ταξινόμηση με ελάχιστα λάθη κυρίως στις κατηγορίες ChestCT και Hand. Η ROC-AUC για τις περισσότερες κατηγορίες ήταν 1.0, με εξαίρεση το Hand που είχε 0.99, δείχνοντας την ισχυρή απόδοση του μοντέλου. Πειραματιστήκαμε δοκιμάζοντας με Grid Search διαφορετικό αριθμό συνελικτικών επιπέδων, νευρώνων ανά επίπεδο, ποσοστό dropout, ρυθμό μάθησης και συναρτήσεις ενεργοποίησης. Για το σκοπό αυτό, χρησιμοποιήσαμε επίσης, Stratified Cross-Validation με 5 folds, ώστε να εξασφαλίσουμε την ακρίβεια των αποτελεσμάτων σε διάφορα υποσύνολα των δεδομένων. Ο καλύτερος συνδυασμός παραμέτρων προέκυψε με learning\_rate 0.001, dropout\_rate 0.3, num\_layers 2, neurons\_per\_layer 64 και activation\_function ReLU, δίνοντας ακρίβεια 99.83% στο test set. Η confusion matrix και οι τιμές precision και recall για όλες τις κατηγορίες ήταν 1.00, επιβεβαιώνοντας την άριστη απόδοση του μοντέλου σχεδόν χωρίς κανένα λάθος στις ταξινομήσεις (εκτός από 1 λάθος για την κατηγορία ChestCT και 1 για την κατηγορία Hand). Επιπλέον, το ROC-AUC ήταν 1.0 για όλες τις κατηγορίες, καταδεικνύοντας τη δυνατότητα του μοντέλου να διακρίνει τέλεια τις κατηγορίες. Συνολικά, η χρήση του cross-validation και του grid search απέδωσε εξαιρετικά αποτελέσματα, επιβεβαιώνοντας ότι το μοντέλο είναι αξιόπιστο και έχει πολύ υψηλές επιδόσεις.

**Συγκριτική Ανάλυση Αλγορίθμων**

1. **Σύγκριση επιδόσεων των αλγορίθμων σε κάθε σύνολο δεδομένων:**
   * *Logistic Regression*: Εξαιρετική απόδοση με υψηλή ακρίβεια (99.25%) και σταθερότητα, ακόμη και με χρήση PCA (99.33%). Ελάχιστα λάθη παρατηρήθηκαν σε ChestCT και Hand.
   * *SVM*: Πέτυχε την υψηλότερη ακρίβεια χωρίς μείωση διαστάσεων (99.75%) και βελτιώθηκε με PCA (99.83%).
   * *KNN*: Επίδοση παρόμοια με την Logistic Regression (99.25%), αλλά βελτιώθηκε με PCA (99.42%), με καλύτερη διάκριση κατηγοριών.
   * *Naive Bayes*: Χαμηλότερη απόδοση χωρίς PCA (96.83%), αλλά αύξηση με PCA (98.42%).
   * *MLP*: Αρχική απόδοση 99.17%, βελτιώθηκε στο 99.33% με βελτιστοποίηση υπερπαραμέτρων και Cross-Validation.
   * *CNN*: Άριστη απόδοση με ακρίβεια 99.67%, φτάνοντας με βελτιστοποίηση υπερπαραμέτρων ακρίβεια 99.83% .Εξαιρετική διάκριση κατηγοριών, αποδεικνύοντας ότι το CNN είναι πολύ καλό μοντέλο για εικόνες.
2. **Πλεονεκτήματα και Μειονεκτήματα κάθε προσέγγισης:**
   * *PCA*: Μειώνει την πολυπλοκότητα των δεδομένων και βελτιώνει την ταχύτητα εκπαίδευσης, χωρίς σημαντική μείωση στην ακρίβεια. Ωστόσο, μπορεί να χάσει πληροφορίες αν οι διαστάσεις που μειώνονται είναι κρίσιμες.
   * *LDA*: Προσφέρει εξαιρετική διάκριση κατηγοριών, αλλά μειώνει την ακρίβεια λόγω της απώλειας πληροφορίας στις διαστάσεις που θεωρεί πιο διακριτικές.
   * *Logistic Regression*: Απλός και γρήγορος, αλλά ενδέχεται να μην επιτυγχάνει την καλύτερη διάκριση για πιο πολύπλοκα δεδομένα.
   * *SVM*: Πολύ ισχυρός για δεδομένα με μεγάλο εύρος χαρακτηριστικών, αλλά απαιτεί μεγάλη υπολογιστική ισχύ για μεγάλα σύνολα δεδομένων.
   * *KNN*: Απλός και ευέλικτος, αλλά η απόδοσή του εξαρτάται από τον αριθμό γειτόνων και τη σωστή επιλογή παραμέτρων.
   * *Naive Bayes*: Ιδανικός για γρήγορη εκπαίδευση και ταξινόμηση, αλλά έχει περιορισμούς σε περίπλοκα δεδομένα με αλληλεξαρτήσεις.
   * *MLP*: Εξαιρετική απόδοση, αλλά μπορεί να υποφέρει από υπερπροσαρμογή χωρίς σωστή ρύθμιση υπερπαραμέτρων.
   * *CNN*: Πολύ αποδοτικό για εικόνες, αλλά απαιτεί μεγάλες υπολογιστικές πόρους και δεδομένα για να επιτύχει τα βέλτιστα αποτελέσματα.
3. **Σχέση μεταξύ πολυπλοκότητας του μοντέλου και απόδοσης:**
   * Η πολυπλοκότητα αυξάνει συνήθως την απόδοση σε πιο σύνθετα σύνολα δεδομένων (όπως τα CNN), αλλά μπορεί να οδηγήσει σε υπερπροσαρμογή ή αυξημένο χρόνο εκπαίδευσης. Απλοί αλγόριθμοι όπως η Logistic Regression είναι γρήγοροι και αποδοτικοί, αλλά ενδέχεται να μην πετύχουν την καλύτερη απόδοση σε πιο σύνθετα σύνολα δεδομένων. Η χρήση τεχνικών μείωσης διάστασης (PCA, LDA) μπορεί να μειώσει την πολυπλοκότητα χωρίς να επηρεάζει αρνητικά την ακρίβεια.
4. **Προτάσεις για βελτιώσεις ή εναλλακτικές προσεγγίσεις:**
   * *Βελτίωση υπερπαραμέτρων*: Η χρήση μεθόδων όπως το Random Search ή Hyperopt μέσω της τεχνικής Bayesian Optimization, θα μπορούσε να εξετάσει περισσότερους συνδυασμούς παραμέτρων πιο αποδοτικά.
   * *Αντιμετώπιση της υπερπροσαρμογής*: Για μοντέλα όπως το MLP ή CNN, η χρήση τεχνικών όπως το early stopping μπορεί να μειώσει την υπερπροσαρμογή και να ενισχύσει την ικανότητα του μοντέλου να γενικεύει σε νέα δεδομένα.
   * *Συνδυασμός αλγορίθμων*: Εξερεύνηση συνδυασμένων προσεγγίσεων όπως το ensemble learning, για παράδειγμα Random Forest ή Boosting, μπορεί να συνδυάσει τα πλεονεκτήματα πολλών αλγορίθμων και να προσφέρει καλύτερα αποτελέσματα.

Βιβλιογραφία

<https://scikit-learn.org/stable/modules/lda_qda.html>

<https://www.jmlr.org/papers/volume9/vandermaaten08a/vandermaaten08a.pdf>

<https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html>

<https://www.w3schools.com/python/python_ml_knn.asp>

<https://scikit-learn.org/stable/modules/naive_bayes.html>