ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ

UNIVERSITY OF PIRAEUS

ΤΜΗΜΑ ΨΗΦΙΑΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

Π.Μ.Σ «ΠΛΗΡΟΦΟΡΙΑΚΑ ΣΥΣΤΗΜΑΤΑ & ΥΠΗΡΕΣΙΕΣ»

ΚΑΤΕΥΘΥΝΣΗ: "ΠΡΟΗΓΜΕΝΑ ΠΛΗΡΟΦΟΡΙΑΚΑ ΣΥΣΤΗΜΑΤΑ"

Διπλωματική Εργασία

«Πειραματική Αξιολόγηση Αλγορίθμων Επιβλεπόμενης Μηχανικής Μάθησης σε Δεδομένα Υγείας»

Γιαννόπουλος Δημήτριος
Αριθμός Μητρώου: ΜΕ2131

Επιβλέπων: Μιχαήλ Φιλιππάκης, Καθηγητής

Αθήνα, 2023

ΠΕΡΙΛΗΨΗ

Τα δεδομένα ηλεκτροεγκεφαλογραφήματος (ΗΕΓ) αποτελούν ένα μη επεμβατικό μέσο παρακολούθησης της ηλεκτρικής δραστηριότητας του εγκεφάλου. Εδώ και χρόνια, οι επιστήμονες και οι επαγγελματίες του ιατρικού κλάδου τα χρησιμοποιούν για να αποκτήσουν γνώσεις σχετικά με διάφορες νευρολογικές καταστάσεις και φαινόμενα. Μια τέτοια εφαρμογή είναι και η αξιολόγηση των οφθαλμικών καταστάσεων των ασθενών, όπου οι μικροσκοπικές αλλαγές στα μοτίβα του ΗΕΓ μπορούν να παράσχουν πολύτιμες διαγνωστικές ενδείξεις. Τα τελευταία χρόνια η ραγδαία ανάπτυξη των τεχνολογιών απόκτησης δεδομένων έχει οδηγήσει σε έκρηξη του όγκου των δεδομένων και, κατ' επέκταση, των δεδομένων ΗΕΓ που είναι διαθέσιμα για ανάλυση. Οι αλγόριθμοι μηχανικής μάθησης (ΜΜ), μέσω των προσαρμοστικών και προβλεπτικών τους δυνατοτήτων, είναι σε θέση να διακρίνουν μοτίβα και σχέσεις σε δεδομένα που μπορεί να διαφεύγουν από τις παραδοσιακές αναλυτικές μεθόδους. Όταν εφαρμόζονται σωστά, όχι μόνο μπορούν να ενισχύσουν την ακρίβεια των διαγνώσεων, αλλά και να προβλέψουν πιθανές μελλοντικές εξελίξεις. Ο συνδυασμός δεδομένων ΗΕΓ και ΜΜ αντιπροσωπεύει ένα αναπτυσσόμενο πεδίο με ποικίλες μελέτες που εξετάζουν αλγορίθμους από στατιστικά μοντέλα έως προηγμένα νευρωνικά δίκτυα. Η ανάδειξη της χρησιμότητας αυτής της έρευνας στον συνδυασμό των τεχνικών ΜΜ με τα δεδομένα ΗΕΓ είναι στο επίκεντρο της παρούσας διπλωματικής εργασίας. Συγκεκριμένα, στην παρούσα εργασία, εφαρμόστηκε πειραματική αξιολόγηση αλγορίθμων επιβλεπόμενης ΜΜ, σε ένα σύνολο δεδομένων ΗΕΓ που επικεντρώνεται στις καταστάσεις των οφθαλμών. Για το σκοπό αυτό, αρχικά, μελετήθηκαν και παρουσιάστηκαν συνοπτικά τεχνικές από παρόμοιες προηγμένες έρευνες, όπου ανάλογα με τον τομέα εφαρμογής τους διακρίθηκαν σε κατηγορίες. Στη συνέχεια πραγματοποιήθηκε εύρεση των βέλτιστων παραμέτρων και σύγκριση των αλγορίθμων στο αρχικό σύνολο δεδομένων χωρίς περαιτέρω προεπεξεργασία προκειμένου να αποκτηθεί ένα μέτρο σύγκρισης. Η ίδια διαδικασία υλοποιήθηκε εκ νέου, αφού πρώτα εφαρμόστηκαν τεχνικές προεπεξεργασίας, καθώς και έγινε παράθεση των αποτελεσμάτων και των συμπερασμάτων που προέκυψαν. Τέλος, εφαρμόστηκε η μέθοδος ΑΚΣ για τη μείωση της διαστασιμότητας του συνόλου δεδομένων και τη διερεύνηση της απόδοσης της ταξινόμησης μέσα από τις νέες διαστάσεις.

Λέξεις κλειδιά: ηλεκτροεγκεφαλογράφημα, αλγόριθμοι, μηχανική μάθηση, προεπεξεργασία, ρύθμιση υπερπαραμέτρων, διασταυρούμενη επικύρωση, ανάλυση κυρίων συνιστωσών

ABSTRACT

Electroencephalogram (EEG) data is a non-invasive method of monitoring the brain's electrical activity. In years, scientists and medical professionals have been using them to gain insights into various neurological conditions and phenomena. One such application is the evaluation of patients' eye states, where tiny changes in EEG patterns can provide valuable diagnostic clues. In recent years the rapid development of data acquisition technologies has led to an explosion in the amount of data and, by extension, the EEG data available for analysis. Machine learning (ML) algorithms, through their adaptive and predictive capabilities, are able to discern patterns and relationships in data that may elude traditional analytical methods. When applied correctly, they can not only enhance the accuracy of diagnoses, but also predict possible future developments. The combination of EEG and ML data represents a growing field with a variety of studies examining algorithms from statistical models to advanced neural networks. Highlighting the usefulness of this research in the combination of ML techniques with EEG data is the focus of this thesis. Specifically, in the present work, an experimental evaluation of supervised ML algorithms was applied to an EEG dataset focusing on eye states. For this purpose, initially, techniques from similar advanced researches were studied and briefly presented, where depending on the field of application they were distinguished into categories. The optimal parameters were then found and the algorithms compared to the original data set without further preprocessing in order to obtain an overview. The same process was implemented again, after preprocessing techniques were first applied, and the results and conclusions obtained were presented. Finally, PCA method was applied to reduce the dimensionality of the data set and investigate the classification’s performance through the new dimensions.

Key words: electroencephalogram, algorithms, machine learning, preprocessing, hyperparameter tuning, cross validation, principal component analysis

Πίνακας Περιεχομένων

Κατάλογος Σχημάτων

Κατάλογος Πινάκων

Κατάλογος Συντομογραφιών

Συντ/φία	*Επεξήγηση*
ΜΜ	Μηχανική Μάθηση
ΕΜΜ	Επιβλεπόμενη Μηχανική Μάθηση
ΗΕΓ	Ηλεκτροεγκεφαλογράφημα
ΑΚΣ	Ανάλυση Κύριων Συνιστωσών
EEG	Electroencephalogram
PCA	Principal Component Analysis

1 Εισαγωγή

Η περίπλοκη και δυναμική φύση του ανθρώπινου εγκεφάλου αντικατοπτρίζεται συχνά στα πολύπλοκα μοτίβα που εντοπίζονται στα εγκεφαλογραφικά δεδομένα. Τα ηλεκτροεγκεφαλογραφήματα (ΗΕΓ), κοινώς γνωστά ως EEG, προσφέρουν ένα μη επεμβατικό μέσο παρακολούθησης της ηλεκτρικής δραστηριότητας του εγκεφάλου. Εδώ και χρόνια, οι επιστήμονες και οι επαγγελματίες του ιατρικού κλάδου χρησιμοποιούν τα ΗΕΓ για να αποκτήσουν γνώσεις σχετικά με διάφορες νευρολογικές καταστάσεις και φαινόμενα. Μια τέτοια εφαρμογή είναι η αξιολόγηση των οφθαλμικών καταστάσεων των ασθενών, όπου οι μικροσκοπικές αλλαγές στα μοτίβα του ΗΕΓ μπορούν να παράσχουν πολύτιμες διαγνωστικές ενδείξεις.

Τον τελευταίο καιρό, η ραγδαία ανάπτυξη των τεχνολογιών απόκτησης δεδομένων έχει οδηγήσει σε έκρηξη του όγκου των δεδομένων ΗΕΓ που είναι διαθέσιμα για ανάλυση. Αυτή η μεγάλης κλίμακας και υψηλής ταχύτητας εισροή πληροφοριών, που στον κόσμο της πληροφορικής αποκαλείται "Big Data" (μεγάλα δεδομένα), παρουσιάζει τόσο ευκαιρίες όσο και προκλήσεις. Από τη μία πλευρά, προσφέρει έναν θησαυρό πληροφοριών που θα μπορούσαν ενδεχομένως να ξεκλειδώσουν νέες γνώσεις σχετικά με τις παθολογικές καταστάσεις των ανθρώπων και επιπλέον να εδραιώσουν τη βάση για τη δημιουργία νέων εφαρμογών επωφελών για τον άνθρωπο. Από την άλλη πλευρά, ο μεγάλος όγκος και η πολυπλοκότητά τους θέτουν σημαντικές προκλήσεις που αφορούν τους υπολογιστικούς πόρους και κόστη που χρειάζονται, ταυτόχρονα με την ανάγκη για την πολύπλοκη ανάλυσή τους για εξαγωγή συμπερασμάτων.

Στο πλαίσιο αυτού του κατακλυσμού δεδομένων, η μηχανική μάθηση (ML) έχει αναδειχθεί σε φάρο ελπίδας, επιτρέποντας στους ερευνητές και τους κλινικούς γιατρούς να εμβαθύνουν στα μοτίβα και τις παραλλαγές που κρύβονται στις τεράστιες εκτάσεις των δεδομένων EEG. Οι αλγόριθμοι μηχανικής μάθησης, μέσω των προσαρμοστικών και προβλεπτικών τους δυνατοτήτων, είναι σε θέση να διακρίνουν μοτίβα και σχέσεις σε δεδομένα που μπορεί να διαφεύγουν από τις παραδοσιακές αναλυτικές μεθόδους. Αυτοί οι αλγόριθμοι, όταν εφαρμόζονται σωστά, όχι μόνο ενισχύουν την ακρίβεια των διαγνώσεων, αλλά και προβλέπουν πιθανές μελλοντικές εξελίξεις, ένα όφελος για προληπτικές ιατρικές παρεμβάσεις. Η προσαρμοστικότητα και η ακρίβειά τους, τους έχουν καταστήσει απαραίτητα εργαλεία στην πληροφορική της υγείας, εγκαινιάζοντας μια νέα εποχή λήψης αποφάσεων με βάση τα δεδομένα στην υγειονομική περίθαλψη.

Ο όρος "πειραματική αξιολόγηση" στον τομέα της ML σημαίνει μια αυστηρή φάση δοκιμών κατά την οποία οι αλγόριθμοι υποβάλλονται σε διάφορα σύνολα δεδομένων για να μετρηθεί η αποτελεσματικότητα, η ακρίβεια και η αξιοπιστία τους. Τέτοιες αξιολογήσεις είναι ζωτικής σημασίας, πολύ περισσότερο στον τομέα των δεδομένων υγείας, όπου το διακύβευμα είναι υψηλό και τα περιθώρια λάθους ελάχιστα. Με τη συστηματική αξιολόγηση των επιδόσεων αυτών των αλγορίθμων, οι ερευνητές μπορούν να εντοπίσουν τις βέλτιστες πρακτικές, να βελτιώσουν τις μεθοδολογίες και να διασφαλίσουν ότι τα συμπεράσματα που προκύπτουν είναι τόσο ισχυρά όσο και αξιόπιστα. Η σημασία αυτής της προσέγγισης στον τομέα της υγειονομικής περίθαλψης δεν μπορεί να υποτιμηθεί. Τα δεδομένα υγείας, από τη φύση τους, είναι τόσο προσωπικά όσο και κρίσιμα. Η χρησιμοποίηση της μηχανικής μάθησης για την άντληση πληροφοριών από αυτά τα δεδομένα συνεπάγεται την ευθύνη να διασφαλιστεί ότι οι αλγόριθμοι αυτοί δεν είναι απλώς αποτελεσματικοί, αλλά και αξιόπιστοι και συνεπείς στις προβλέψεις τους.

Ωστόσο, ακόμη και με τις εξελιγμένες δυνατότητες των αλγορίθμων μηχανικής μάθησης, ο όγκος και η πολυπλοκότητα των δεδομένων EEG μπορεί να είναι δύσκολα διαχειρίσιμα. Καθώς τα σύνολα

δεδομένων αυξάνονται σε διαστασιμότητα, αυξάνονται και οι υπολογιστικές απαιτήσεις και ο κίνδυνος να πέσουν στην "κατάρα της διαστασιμότητας", μια κατάσταση όπου τα δεδομένα γίνονται τόσο πολυδιάστατα, που η ανάλυσή τους γίνεται εξαιρετικά αναξιόπιστη. Εδώ είναι που οι τεχνικές απλοποίησης και συμπύκνωσης των δεδομένων, καθίστανται ανεκτίμητες, χωρίς να θυσιάζονται οι εγγενείς πληροφορίες. Η μείωση της διαστασιμότητας αναδεικνύεται ως ένα ζωτικής σημασίας βήμα προεπεξεργασίας σε τέτοια ερευνητικά προβλήματα, ανοίγοντας το δρόμο για πιο αποτελεσματικές και προσοδοφόρες αναλύσεις. Μεταξύ της πληθώρας των τεχνικών μείωσης της διαστασιμότητας, η Ανάλυση Κύριων Συνιστωσών (Principal Component Analzsis(PCA)) ξεχωρίζει λόγω της αποδεδειγμένης αποτελεσματικότητας και της ευρείας υιοθέτησής της.

Συγκεκριμένα, η μείωση της διαστασιμότητας είναι μια τεχνική που χρησιμοποιείται συνήθως στον τομέα της μηχανικής μάθησης και της ανάλυσης δεδομένων για την αντιμετώπιση των προκλήσεων που θέτουν τα δεδομένα υψηλής διάστασης. Με τη μείωση του αριθμού των εξεταζόμενων μεταβλητών και την εξαγωγή μόνο των πιο σημαντικών χαρακτηριστικών, η τεχνική αυτή μπορεί να εξορθολογήσει τους υπολογισμούς, να μειώσει τον θόρυβο και να ενισχύσει τη σαφήνεια των μοτίβων μέσα στα δεδομένα. Η τεχνική PCA, μια κλασική γραμμική τεχνική, έχει αναδειχθεί ως μια από τις κορυφαίες μεθόδους για την επίτευξη μείωσης των πολλών διαστάσεων. Η ικανότητά της να μετασχηματίζει δεδομένα υψηλής διάστασης σε μορφή χαμηλότερης διάστασης χωρίς ουσιαστική απώλεια πληροφοριών την καθιστά ελκυστική επιλογή για πολλούς ερευνητές.

1.1 Ερευνητικά Ερωτήματα

Στην παρούσα μελέτη, εφαρμόζεται μια πειραματική αξιολόγηση διαφόρων αλγορίθμων επιβλεπόμενης ΜΜ σε ένα σύνολο δεδομένων εγκεφαλογραφήματος που επικεντρώνεται στις καταστάσεις (ανοιχτοί ή κλειστοί) των οφθαλμών των ασθενών. Επιπλέον, στο επίκεντρο της έρευνας βρίσκεται η διερεύνηση της τεχνικής PCA. Στόχος μας είναι να αξιολογήσουμε κατά πόσον η PCA μπορεί να μειώσει αποτελεσματικά τη διαστασιμότητα των δεδομένων ΗΕΓ, χωρίς να διακυβεύεται η ακεραιότητα και η χρησιμότητα των πληροφοριών. Επιπλέον, με την εισαγωγή της PCA στον τομέα των μεγάλων δεδομένων, σκοπεύουμε να διερευνήσουμε την επεκτασιμότητα και την αποτελεσματικότητά της, εκτιμώντας ουσιαστικά αν υπόσχεται να αποτελέσει μια βιώσιμη λύση για την ανάλυση δεδομένων EEG μεγάλης κλίμακας.

Τα ερευνητικά ερωτήματα που καλείται να διερευνήσει η παρούσα διπλωματική είναι τα ακόλουθα δύο και συγκεκριμένα:

i. Μπορούμε με τη πειραματική αξιολόγηση αλγορίθμων Μηχανικής Μάθησης να προβλέψουμε με μεγάλη ακρίβεια την κατάσταση των οφθαλμών των ασθενών;

ii. Μπορούν τα αποτελέσματα της πειραματικής αξιολόγησης να επεκταθούν σε Μεγάλα Δεδομένα Ηλεκτροεγκεφαλογραφήματος; Ποια είναι η αποτελεσματικότητα της μεθόδου PCA και κατά πόσο επιτυγχάνει την ταυτόχρονη μείωση των διαστάσεων των Μεγάλων Δεδομένων με υψηλά ποσοστά ταξινόμησης;

Μέσα από την υλοποίηση αυτής της έρευνας, σκοπός είναι να υπάρξει μία συμβολή στην ενίσχυση των διαγνωστικών δυνατοτήτων της ανάλυσης του ΗΕΓ και γενικότερα η αναζήτηση πιο εξορθολογισμένων και αποτελεσματικών μεθοδολογιών επεξεργασίας μεγάλων δεδομένων στον τομέα της νευροπληροφορικής.

2 Βιβλιογραφική Ανασκόπηση

Το ηλεκτροεγκεφαλογράφημα (ΗΕΓ) αποτελεί, εδώ και καιρό, ένα ανεκτίμητο εργαλείο για τους νευροεπιστήμονες, προσφέροντας ένα παράθυρο στις περίπλοκες ηλεκτρικές δραστηριότητες του ανθρώπινου εγκεφάλου. Αποτελεί μια μη επεμβατική μέθοδο μέτρησης της ηλεκτρικής δραστηριότητας στον εγκέφαλο. Εδώ και δεκαετίες, παρέχει στους ερευνητές και τους κλινικούς γιατρούς ανεκτίμητες πληροφορίες για τη λειτουργία του εγκεφάλου, οδηγώντας στην πρόοδο της κατανόησης των νευρολογικών διεργασιών, του ύπνου, της νόησης και άλλων φαινομένων. Μέσω του ΗΕΓ, οι ερευνητές μπόρεσαν να εμβαθύνουν σε μια πληθώρα νευρολογικών φαινομένων, από την κατανόηση των γνωστικών διεργασιών έως την παρακολούθηση των προτύπων ύπνου και ακόμη και την αποκωδικοποίηση των συναισθημάτων.

Τα τελευταία χρόνια, ωστόσο, έχει προκύψει μια συνέργεια μεταξύ της έρευνας του ΗΕΓ και του τομέα της Μηχανικής Μάθησης (ΜΜ). Καθώς το EEG παρέχει τεράστιες ποσότητες χρονικών δεδομένων, οι παραδοσιακές αναλυτικές προσεγγίσεις συχνά δεν μπορούν να αξιοποιήσουν πλήρως τις δυνατότητές του. Εδώ είναι που η μηχανική μάθηση, με την ικανότητά της να αποκρυπτογραφεί σύνθετα μοτίβα και σχέσεις λαμβάνει μετασχηματιστικό ρόλο.

Η εφαρμογή της μηχανικής μάθησης σε δεδομένα ΗΕΓ αντιπροσωπεύει μια βαθιά εξέλιξη στον τρόπο με τον οποίο ερμηνεύουμε και χρησιμοποιούμε αυτά τα σήματα εγκεφαλικών κυμάτων. Είτε πρόκειται για τον εντοπισμό ενδείξεων νευρολογικών διαταραχών, είτε για την πρόβλεψη γνωστικών καταστάσεων, είτε για τη διάκριση μεταξύ ανεπαίσθητων αλλαγών όπως η κατάσταση των ματιών, τα μοντέλα ML έχουν αυξήσει την ικανότητά μας να εξάγουμε σημαντικές πληροφορίες από τα δεδομένα ΗΕΓ. Η προσαρμοστικότητά τους, η επεκτασιμότητά τους και οι δυνατότητες πρόβλεψης έχουν μετατρέψει τις μελέτες ΗΕΓ από παθητικές παρατηρήσεις σε δυναμικές αναλύσεις, όπου η ανατροφοδότηση σε πραγματικό χρόνο και η διάγνωση ακριβείας έχουν καταστεί εφικτές.

Μεταξύ των πολύπλευρων χρήσεων του ΗΕΓ, η μελέτη των οφθαλμικών καταστάσεων - συγκεκριμένα των μεταβάσεων μεταξύ ανοικτών και κλειστών ματιών - κατέχει μια μοναδική θέση. Η παρακολούθηση της κατάστασης των ματιών είναι κάτι περισσότερο από μια απλή αξιολόγηση της οπτικής ενεργοποίησης, καθώς χρησιμεύει ως πύλη για την κατανόηση των μεταβολών της εγκεφαλικής δραστηριότητας, των επιπέδων κόπωσης, την παρακολούθηση της διάρκειας της προσοχής, την ανίχνευση των επιπέδων εγρήγορσης ή υπνηλίας, ακόμη και ορισμένων παθολογικών καταστάσεων. Η σημασία αυτού του συγκεκριμένου τομέα μελέτης έγκειται στις πιθανές εφαρμογές του, που κυμαίνονται από συστήματα παρακολούθησης της ασφάλειας των οδηγών, όπου μπορεί να ανιχνευθεί η υπνηλία, έως την εκπαίδευση νευροανάδρασης, όπου τα άτομα μαθαίνουν να ρυθμίζουν την εγκεφαλική τους δραστηριότητα.

Ενώ η τεχνολογία και οι μεθοδολογίες πίσω από το ΗΕΓ έχουν εξελιχθεί με την πάροδο του χρόνου, η βασική αρχή παραμένει αμετάβλητη: η σύλληψη της ηλεκτρικής λειτουργίας του εγκεφάλου για την αποκάλυψη των μυστηρίων του. Η συνάφεια και η χρησιμότητα των συνόλων δεδομένων ΗΕΓ γίνεται ακόμη πιο κατανοητή όταν εντάσσεται στο ευρύτερο τοπίο της έρευνας και της ανάλυσης του ΗΕΓ. Η ερευνητική διασταύρωση δεδομένων ΗΕΓ και ΜΜ, είναι ένας αναπτυσσόμενος τομέας, πλούσιος σε μελέτες που διερευνούν ποικίλους αλγορίθμους, από παραδοσιακά στατιστικά μοντέλα έως προηγμένα νευρωνικά δίκτυα.

Η κατανόηση της προϋπάρχουσας έρευνας, των μεθοδολογιών και των ευρημάτων που σχετίζονται με δεδομένα EEG, καθώς και των τάσεων και των ανακαλύψεων στην εφαρμογή τεχνικών ML σε δεδομένα EEG, αποσκοπεί στην ανάδειξη της χρησιμότητας της παρούσας διπλωματικής. Μέσω της διερεύνησης της υπάρχουσας έρευνας, στόχος είναι μια ολοκληρωμένη επισκόπηση της εξέλιξης στην εφαρμογή τεχνικών ML σε δεδομένα EEG.

2.1 Μηχανική Μάθηση σε δεδομένα Ηλεκτροεγκεφαλογραφήματος

Στις παρακάτω υποενότητες έχει γίνει η διάκρισή τους ανάλογα με τον τομέα εφαρμογής των ερευνών. Αυτό πραγματοποιήθηκε για την αντιπροσωπευτικότερη παρουσίαση των μελετών και κυρίως για να γίνει ευκολότερη για τον αναγνώστη, η ανάδειξη των εφαρμογών της ΜΜ σε δεδομένα ΗΕΓ.

2.1.1 Ασθενείς με Επιληψία

Αναλυτικότερα, ο Hafeez κ.ά. [1] ασχολήθηκαν με την ανίχνευση επιληψίας σε σήματα EEG. Πρότειναν μία νέα προσέγγιση που βασίζεται στην ανάλυση κυματιδίων και την αριθμητική κωδικοποίηση για την αυτοματοποιημένη ανίχνευση και διάγνωση επιληπτικής κρίσης σε σήματα EEG χρησιμοποιώντας τεχνικές μηχανικής μάθησης. Μέχρι τώρα η παραδοσιακή οπτική ερμηνεία των σημάτων EEG είναι αργή και επιρρεπής σε σφάλματα. Για να αντιμετωπιστεί αυτό, παρουσιάζουν μια νέα διαγνωστική μέθοδο CAD (Computer-Aided Diagnostic) για την ανίχνευση επιληψίας σε σήματα EEG, εισάγοντας μια τεχνική που συνδυάζει τον διακριτό μετασχηματισμό κυματιδίων (Discrete Wavelet Transform (DWT)) και την αριθμητική κωδικοποίηση. Η διαδικασία περιλαμβάνει τη διάσπαση των σημάτων EEG με χρήση DWT, τη μετατροπή των σημαντικών συντελεστών σε ροές bit και τη χρήση ταξινομητών μηχανικής μάθησης για τη διάκριση μεταξύ επιληπτικών κρίσεων και τυπικών σημάτων. Χρησιμοποιώντας μια βάση δεδομένων αναφοράς από το Πανεπιστήμιο της Βόννης για επικύρωση, η μέθοδος πέτυχε τέλεια ακρίβεια 100%. Ο συνδυασμός DWT και αριθμητικής κωδικοποίησης εξάγει αποτελεσματικά, σημαντικά χαρακτηριστικά του ΗΕΓ, με την προτεινόμενη τεχνική να αποδεικνύεται υπολογιστικά γρήγορη, επιταχύνοντας έτσι αντίστοιχες κλινικές εφαρμογές που μπορούν να διαγνώσουν την επιληψία σε πραγματικό χρόνο.

Αντιστοίχως, η Savadkoohi κ.ά. [2] διερευνούν τη χρήση της μηχανικής μάθησης στην πρόβλεψη επιληπτικών κρίσεων μέσω σημάτων ηλεκτροεγκεφαλογραφήματος (EEG). Στην έρευνά τους, συνέλεξαν καταγραφές EEG τόσο από υγιείς εθελοντές, όσο και από ασθενείς με επιληψία, αναλύοντας αυτές τις καταγραφές στα πεδία του χρόνου, της συχνότητας και της χρονικής συχνότητας. Η εξαγωγή χαρακτηριστικών πραγματοποιήθηκε μέσω του φίλτρου Butterworth, του μετασχηματισμού Fourier και του μετασχηματισμού Wavelet και στη συνέχεια βελτιστοποιήθηκε με τη χρήση του T-test και του Sequential Forward Floating Selection (SFFS) για την επιλογή χαρακτηριστικών. Εφάρμοσαν τους αλγορίθμους SVM και KNN για ταξινόμηση, διαπιστώνοντας ότι ο SVM υπερτερεί ελαφρώς έναντι του KNN με ποσοστό ακρίβειας 100% έναντι 99,5%. Η σημασία αυτής της μελέτης έγκειται στη δημιουργία ενός αξιόπιστου και αποτελεσματικού μοντέλου για την ανίχνευση επιληπτικών κρίσεων με τη χρήση δεδομένων EEG. Η προσέγγισή τους εμφανίζει ισχυρή απόδοση τόσο σε βιολογικά σήματα υψηλής όσο και χαμηλής συχνότητας. Η τεχνική σχεδιασμού χαρακτηριστικών που χρησιμοποιείται διασφαλίζει την εξαγωγή των σχετικών πληροφοριών χωρίς την ανάγκη για πρόσθετα στάδια επεξεργασίας, εξοικονομώντας έτσι υπολογιστικούς πόρους. Επιπλέον, το μοντέλο τους φανερώνει και επεκτασιμότητα σε άλλα ιατρικά σήματα, όπως το ηλεκτρομυογράφημα (EMG) και το ηλεκτροκαρδιογράφημα (ECG).

Σε άλλη μια σημαντική προσπάθεια, ο Jaiswal κ.ά. [3] επικεντρώθηκαν στην ανάπτυξη αυτοματοποιημένων συστημάτων ανίχνευσης επιληπτικών κρίσεων, δεδομένης της χρονοβόρας φύσης της παραδοσιακής ανάλυσης ΗΕΓ. Η μελέτη προτείνει δύο νέες μεθόδους, την PCA με βάση τα υποδείγματα (Subpattern based PCA (SpPCA) και την PCA με βάση τη συσχέτιση μεταξύ των υποδειγμάτων (cross-subpattern correlation-based PCA (SubXPCA), σε συνδυασμό με τον αλγόριθμο SVM για την ανίχνευση επιληπτικών κρίσεων. Αυτές οι τεχνικές εστιάζουν στη συσχέτιση υποδειγμάτων στα σήματα EEG, βοηθώντας τη διαδικασία λήψης αποφάσεων. Η εξαγωγή χαρακτηριστικών επιτεύχθηκε με τη χρήση των SpPCA και SubXPCA, ακολουθούμενη από ταξινόμηση με SVM. Σύμφωνα με τα ευρήματά τους, τόσο η SpPCA όσο και η SubXPCA υπερτερούσαν της τυπικής PCA, όσον αφορά την ακρίβεια ταξινόμησης. Χρησιμοποίησαν ένα σύνολο δεδομένων EEG επιληψίας που περιείχε 500 σήματα EEG και διεξήγαγε επτά διαφορετικές πειραματικές περιπτώσεις ταξινόμησης, αξιολογώντας τις επιδόσεις με τη χρήση δεκαπλής διασταυρούμενης επικύρωσης (10-fold Cross Validation). Από τα αποτελέσματα, τόσο η SpPCA όσο και η SubXPCA πέτυχαν 100% ακρίβεια στην ταξινόμηση φυσιολογικών και επιληπτικών σημάτων ΗΕΓ σε ορισμένα σενάρια, με την SpPCA να αποδίδει τα καλύτερα αποτελέσματά της χρησιμοποιώντας 40-80 χαρακτηριστικά και την SubXPCA με 18-40 χαρακτηριστικά. Μια ενδιαφέρουσα παρατήρηση είναι πως παρά τις αρκετές υπάρχουσες μεθόδους στη βιβλιογραφία, καμία δεν ασχολήθηκε με το ζήτημα της συσχέτισης των επιμέρους προτύπων μεταξύ των σημάτων ΗΕΓ, το οποίο, το 2018, η παρούσα μελέτη ανέδειξε ως κρίσιμο. Η προσδοκία τους είναι ότι η αναγνώριση και η διερεύνηση των συσχετίσεων υποπροτύπων στα σήματα EEG θα μπορούσε να ωφελήσει περαιτέρω την επεξεργασία άλλων βιοϊατρικών σημάτων.

2.1.2 Ασθενείς με Κατάθλιψη

Ο Hosseinifard κ.ά. [4] εξέτασαν την εφαρμογή της μη γραμμικής ανάλυσης σήματος ΗΕΓ για τη διάκριση μεταξύ ασθενών με κατάθλιψη και φυσιολογικών ατόμων. Κατέγραψαν σήματα ΗΕΓ από 45 ασθενείς με κατάθλιψη χωρίς φαρμακευτική αγωγή και 45 φυσιολογικούς συμμετέχοντες. Στη μελέτη εξήχθησαν τόσο η ισχύς από τέσσερις ζώνες ΗΕΓ, όσο και μη γραμμικά χαρακτηριστικά, συμπεριλαμβανομένης της ανάλυσης Detrended Fluctuation (DFA), του fractal Higuchi, της συσχέτισης των διαστάσεων και του εκθέτη Lyapunov. Για την ταξινόμηση χρησιμοποιήθηκαν τεχνικές μηχανικής μάθησης, όπως ο k-κοντινότερος γείτονας (k-nearest neighbor), η Linear Discriminant Analysis (LDA) και η λογιστική παλινδρόμηση. Η συσχέτιση των διαστάσεων χρησιμοποιήθηκε με τη λογιστική παλινδρόμηση και απέδωσε την υψηλότερη ακρίβεια σε ποσοστό 83,3%. Με το συνδυασμό όλων των μη γραμμικών χαρακτηριστικών, η ακρίβεια αυξήθηκε στο 90%. Ο γενετικός αλγόριθμος χρησιμοποιήθηκε για την επιλογή κομβικών χαρακτηριστικών, επιβεβαιώνοντας ότι η μη γραμμική ανάλυση του ΗΕΓ θα μπορούσε να χρησιμεύσει ως πολύτιμο διαγνωστικό εργαλείο για την κατάθλιψη. Αναλυτικότερα, κατά τη διάρκεια της ανάλυσης, παρατηρήθηκε ότι η ισχύς της ζώνης άλφα συχνοτήτων στο ΗΕΓ διέφερε σημαντικά μεταξύ καταθλιπτικών και φυσιολογικών ατόμων, ιδίως στο αριστερό ημισφαίριο. Αυτή η διάκριση στη ζώνη άλφα ήταν σύμφωνη με προηγούμενα ευρήματα. Από τα μη γραμμικά χαρακτηριστικά, η διάσταση της συσχέτισης αποδείχθηκε πιο αποτελεσματική για την ανάλυση του ΗΕΓ, όσον αφορά τη διάκριση μεταξύ των δύο ομάδων. Η μελέτη υπογραμμίζει την αποτελεσματικότητα των μη γραμμικών χαρακτηριστικών, ιδίως όταν συνδυάζονται, για την ενίσχυση της ακρίβειας ταξινόμησης. Συμπερασματικά, τονίζουν τις δυνατότητες της ανάλυσης σήματος EEG ως μεθόδου για τη μελέτη και τη διάγνωση της κατάθλιψης.

Αντιστοίχως, η Jaworska κ.ά. [5] χρησιμοποίησαν τεχνικές μηχανικής μάθησης (ML), ιδίως Random Forests, για να εξακριβώσουν τις δυνατότητες του EEG και των κλινικών δεδομένων στην πρόβλεψη της ανταπόκρισης των ασθενών με μείζονα καταθλιπτική διαταραχή (ΜΚΔ) - Major Depressive Disorder (MDD) - στα αντικαταθλιπτικά. Κατά τη διάρκεια μιας δοκιμής 12 εβδομάδων με 51

ασθενείς με MDD, συγκεντρώθηκαν δεδομένα EEG και κλινικοί δείκτες, όπως η κλίμακα κατάθλιψης Montgomery-Asberg (MADRS). Μέσω μιας λεπτομερούς προσέγγισης ML, 50 χαρακτηριστικά eLORETA και 88 χαρακτηριστικά EEG του τριχωτού της κεφαλής αναδείχθηκαν ως προβλέψιμα. Η μελέτη τους ενσωμάτωσε επίσης άλλους αλγορίθμους για σύγκριση. Όταν ενοποιήθηκαν, αυτά τα χαρακτηριστικά πέτυχαν ακρίβεια πρόβλεψης 88%, ενώ ένα μοντέλο που επικεντρώθηκε στα 12 πιο κρίσιμα χαρακτηριστικά έφτασε το 78%. Τα αποτελέσματα υπογραμμίζουν την υπόσχεση των μοντέλων ML, συμπεριλαμβανομένων των Random Forests, της ανάλυσης κύριων συνιστωσών και άλλων συγκριτικών αλγορίθμων, στην αξιοποίηση των δεδομένων EEG και των κλινικών δεδομένων για την πρόβλεψη της αποτελεσματικότητας των αντικαταθλιπτικών, θέτοντας τα θεμέλια για εξατομικευμένες θεραπευτικές προσεγγίσεις με βάση βιοδείκτες.

2.1.3 Ασθενείς με Σχιζοφρένεια

Η Shim κ.ά. [6] είχαν ως στόχο την ανάπτυξη ενός διαγνωστικού εργαλείου για τη σχιζοφρένεια χρησιμοποιώντας τεχνικές μηχανικής μάθησης που εφαρμόζονται σε βιοδείκτες EEG. Ενώ η πλειοψηφία προηγούμενων έρευνών επικεντρώνονταν κυρίως σε χαρακτηριστικά ΗΕΓ, σε επίπεδο αισθητήρα, για τη διάγνωση της σχιζοφρένειας, αυτή η μελέτη ενσωμάτωσε χαρακτηριστικά τόσο σε επίπεδο αισθητήρα, όσο και σε επίπεδο πηγής από σήματα ΗΕΓ που καταγράφηκαν. Οι καταγραφές EEG συγκεντρώθηκαν από 34 ασθενείς με σχιζοφρένεια και 34 υγιείς συμμετέχοντες. Τα ευρήματα αποκάλυψαν ότι η συνδυασμένη χρήση χαρακτηριστικών σε επίπεδο αισθητήρα και σε επίπεδο πηγής είχε ως αποτέλεσμα υψηλότερη ακρίβεια ταξινόμησης, με το συνδυασμένο σύνολο χαρακτηριστικών να φτάνει σε μέγιστη ακρίβεια 88,24%. Η χωρική κατανομή αυτών των χαρακτηριστικών έδειξε ότι τα επιλεγμένα χαρακτηριστικά σε επίπεδο αισθητήρα βρέθηκαν κυρίως στη μετωπιαία περιοχή, ευθυγραμμισμένα με γνωστές περιοχές παθολογίας για σχιζοφρένεια, ενώ τα χαρακτηριστικά σε επίπεδο πηγής προέκυψαν κυρίως από την κροταφική περιοχή. Παρατήρησαν ότι η συνδυασμένη χρήση χαρακτηριστικών τόσο του αισθητήρα όσο και των χαρακτηριστικών σε επίπεδο πηγής ξεπέρασε την ακρίβεια της χρήσης οποιουδήποτε συνόλου χαρακτηριστικών. Μια εξέταση των επιλεγμένων χαρακτηριστικών αποκάλυψε ότι οκτώ χαρακτηριστικά σε επίπεδο αισθητήρα εντοπίστηκαν στην μετωπιαία περιοχή, μια περιοχή όπου τα πλάτη P300 μειώθηκαν σημαντικά σε ασθενείς με σχιζοφρένεια. Αντίθετα, τα επτά χαρακτηριστικά σε επίπεδο πηγής βρίσκονταν κυρίως στον αριστερό κροταφικό φλοιό.

Επιπρόσθετα, οι Buettner, Ricardo κ.ά. [7] παρουσίασαν μια νέα προσέγγιση μηχανικής μάθησης για τη διάγνωση της σχιζοφρένειας χρησιμοποιώντας καταγραφές ηλεκτροεγκεφαλογραφίας (EEG). Χρησιμοποίησαν τον ταξινομητή Random Forest σε συνδυασμό με διαιρέσεις φασμάτων EEG για τη διάκριση μεταξύ σχιζοφρενικών και μη σχιζοφρενικών ατόμων. Η έρευνα τους διεξήχθη σε 499 εγγραφές ΗΕΓ ενός λεπτού από 28 συμμετέχοντες, που περιλάμβαναν 14 παρανοϊκούς σχιζοφρενείς ασθενείς και 14 υγιείς συμμετέχοντες. Η μέθοδός τους πέτυχε ένα εντυπωσιακό ποσοστό ακρίβειας 96,77%, καθιστώντας την ένα πολλά υποσχόμενο εργαλείο για γρήγορη και αξιόπιστη διάγνωση της σχιζοφρένειας. Τα αποτελέσματα διευκρίνισαν περαιτέρω ειδικές υποζώνες συχνοτήτων στα φάσματα EEG που είναι κρίσιμα για τη διάγνωση της σχιζοφρένειας. Εξετάζοντας έως και συχνότητες 100 Hz στο φάσμα EEG, προσδιόρισαν τις τέσσερις πιο προγνωστικές ζώνες συχνοτήτων. Η υψηλότερη προγνωστική τιμή αποδόθηκε στη ζώνη 50-50,5 Hz. Είναι ενδιαφέρον ότι από αυτές τις ζώνες, τρεις βρίσκονταν στην περιοχή γάμμα και μία στην περιοχή άλφα. Σε σύγκριση με άλλες μελέτες, η ισορροπημένη ακρίβεια αυτής της προσέγγισης ξεπέρασε σημαντικά τα προηγούμενα αποτελέσματα. Επιπλέον, η αποτελεσματικότητα της προσέγγισης, η οποία απαιτεί εγγραφή ΗΕΓ μόνο ενός λεπτού, θα μπορούσε να κάνει τη διάγνωση της σχιζοφρένειας πιο γρήγορη και ακριβή, μειώνοντας τα ανθρώπινα λάθη σε κλινικές συνθήκες.

2.1.4 Ασθενείς με Ανίατες Ασθένειες (Parkinson, Alzheimer)

Σε αυτήν την ενότητα παρουσιάζονται έρευνες που σχετίζονται με ανίατες ασθενείς όπως είναι το Parkinson και Alzheimer και φανερώνουν σημαντικά αποτελέσματα στην «εκμετάλλευση» δεδομένων ΗΕΓ και την εμπλοκή τεχνικών ΜΜ για την εξαγωγή πολύ χρήσιμων συμπερασμάτων.

Η Oliveira κ.ά. [8] διερεύνησαν τις δυνατότητες χρήσης σημάτων ΗΕΓ, σε συνδυασμό με τεχνικές μηχανικής μάθησης, για τη διάγνωση της νόσου του Πάρκινσον (ΝΠ) στα αρχικά της στάδια. Αξιοποιώντας τη σημαντική ανατομική αναπαράσταση του οπτικού συστήματος στον εγκεφαλικό φλοιό, πρότειναν έναν βιοδείκτη για την ΝΠ που χρησιμοποιούσε σήματα EEG και φωτοδιέγερση για την εξαγωγή χαρακτηριστικών. Η μέθοδός τους αποσκοπούσε στην ταξινόμηση των συμμετεχόντων σε τρεις κατηγορίες: ασθενείς με φαρμακευτική αγωγή με ΝΠ, σε ασθενείς με ΝΠ χωρίς φαρμακευτική αγωγή και υγιή άτομα. Πέτυχαν αξιοσημείωτα αποτελέσματα, με ακρίβεια που ξεπέρασε το 99% και στατιστική kappa που έφτασε έως και 0,98 χρησιμοποιώντας τον αλγόριθμο Random Forest και τεχνικές επιλογής χαρακτηριστικών (feature selection). Τα ευρήματα τους ανέδειξαν τη σημασία του αριστερού εγκεφαλικού ημισφαιρίου, ιδίως σε περιοχές όπως ο νησιωτικός, ο οσφρητικός και ο κογχομετωπιαίος φλοιός, στην ανίχνευση της νόσου σε πρώιμο στάδιο. Διαπίστωσαν ότι ο Random Forest, επέδειξε ανώτερες επιδόσεις σε σύγκριση με άλλους αλγορίθμους, επιτυγχάνοντας ακρίβεια έως και 99,22% στην κατηγοριοποίηση των σημάτων ΗΕΓ. Η σημασία αυτών των αποτελεσμάτων υπογραμμίζεται από προηγούμενες έρευνες που δείχνουν ότι το αριστερό ημισφαίριο του εγκεφάλου υφίσταται αλλαγές κατά τα αρχικά στάδια της νόσου PD. Στην έρευνά τους, τόνισαν επίσης τις δυνατότητες της μηχανικής μάθησης, ειδικά με δεδομένα EEG, στη διάκριση ασθενών με πρώιμο στάδιο ΝΠ, μέσω της παρακολούθησης των μεταβολών στην αισθητηριακή επεξεργασία. Κάνοντας παραλληλισμούς με άλλες μελέτες, υπέθεσαν ότι οι περιοχές που σχετίζονται με την οπτική επεξεργασία, συμπεριλαμβανομένου του βρεγματικού και του μετωπιαίου λοβού, μπορεί να μεταβάλλονται στους ασθενείς με ΝΠ. Συμπερασματικά, η αξιοσημείωτη ακρίβεια που επιτεύχθηκε μέσω της χρήσης του Random Forest υποδηλώνει υποσχόμενες κλινικές εφαρμογές για την έγκαιρη ανίχνευση της ΝΠ, η οποία θα μπορούσε να οδηγήσει σε βελτιωμένη πρόγνωση και βελτιωμένη ποιότητα ζωής για τους ασθενείς.

Ο Betrouni κ.α [9] χρησιμοποίησαν δεδομένα ΗΕΓ ασθενών με Πάρκινσον (ΝΠ) που βρισκόταν σε κατάσταση ηρεμίας, σε συνδυασμό με τεχνικές εξόρυξης δεδομένων για τη σκιαγράφηση γνωστικών καταστάσεων. Ανέλυσαν τα δεδομένα EEG από 118 ασθενείς με PD, οι οποίοι κατηγοριοποιήθηκαν σε πέντε ομάδες με βάση τη σοβαρότητα των γνωστικών διαταραχών τους. Πραγματοποιήσαν φασματική ανάλυση ισχύος σε επτά ζώνες συχνοτήτων των σημάτων ΗΕΓ, ενώ τα βασικά ποσοτικά χαρακτηριστικά ΗΕΓ από 100 ασθενείς αξιολογήθηκαν με τη χρήση του αλγορίθμου SVM και του αλγορίθμου k-κοντινότερων γειτόνων (KNN). Στη συνέχεια, τα μοντέλα δοκιμάστηκαν σε 18 ασθενείς, επιτυγχάνοντας ακρίβεια ταξινόμησης 84% (SVM) και 88% (KNN). Τα ευρήματα τους υποδεικνύουν ότι η σοβαρότητα της γνωστικής εξασθένησης σε ασθενείς με PD μπορεί να χαρακτηριστεί χρησιμοποιώντας μια 10λεπτη καταγραφή ΗΕΓ σε κατάσταση ηρεμίας. Αναλυτικότερα, έδειξαν ότι η άνοια στους ασθενείς με ΝΠ συνδέεται με μια αξιοσημείωτη επιβράδυνση της εγκεφαλικής δραστηριότητας σε κατάσταση ηρεμίας, σε σύγκριση με τους ασθενείς με ΠΣ που δεν πάσχουν από άνοια και με τα υγιή άτομα που είναι στην ίδια ηλικία. Συνήθως, οι γνωστικοί τύποι των ασθενών με ΝΠ προσδιορίζονται μέσω νευροψυχολογικής αξιολόγησης. Ωστόσο, η παρούσα μελέτη υποδηλώνει τη σκοπιμότητα της χρήσης ποσοτικών χαρακτηριστικών του ΗΕΓ ως προκαταρκτικού εργαλείου για την αξιολόγηση του γνωστικού προφίλ ενός ασθενούς. Η προσέγγιση αυτή προσφέρει έναν γρήγορο αρχικό έλεγχο, καθοδηγώντας τους ασθενείς προς πιο ολοκληρωμένες εξετάσεις με βάση την κατάστασή τους. Ωστόσο, οι συγγραφείς τονίζουν ότι η μέθοδός τους παρέχει μόνο χαρακτηρισμό κατά τη στιγμή της απόκτησης του ΗΕΓ και δεν προβλέπει τη γνωστική εξέλιξη. Συμπερασματικά καταλήγουν, πως το ΗΕΓ, όντας μια οικονομικά αποδοτική και ευρέως προσβάσιμη μέθοδος, έχει δυνατότητες ως εργαλείο διαλογής

ασθενών, για τον προσδιορισμό της βαρύτητας της γνωστικής εξασθένησης σε ασθενείς με νόσο Parkinson.

Στην άλλη κατηγορία ανίατης ασθένειας, που είναι η νόσος Alzheimer, η Simpraga κ.ά. [10] διερεύνησαν τις δυνατότητες χρήσης πολλαπλών βιοδεικτών ηλεκτροεγκεφαλογραφίας (EEG) για την ανίχνευση των επιπτώσεων ασθενειών ή φαρμακολογικών παρεμβάσεων. Η έρευνα τους επικεντρώθηκε στη χρήση μηχανικής μάθησης για την ενίσχυση της απόδοσης ταξινόμησης των δεδομένων EEG, με έμφαση στις επιδράσεις της σκοπολαμίνης - ενός ανταγωνιστή του υποδοχέα muscarinic acetylcholine (mAChR). Ανέλυσαν δεδομένα EEG για να εντοπίσουν βιοδείκτες ενδεικτικούς της χολινεργικής δυσλειτουργίας, χαρακτηριστικό γνώρισμα της νόσου Αλτσχάιμερ. Ανέπτυξαν έναν ειδικό δείκτη, τον δείκτη mAChR, που προέρχεται από 14 βιοδείκτες EEG, ο οποίος παρουσίασε ανώτερη ακρίβεια ταξινόμησης, ευαισθησία, ειδικότητα και ακρίβεια που κυμαίνεται μεταξύ 88% και 92%. Ειδικότερα, ο δείκτης mAChR ήταν αποτελεσματικός στη διάκριση μεταξύ υγιών ηλικιωμένων ατόμων και ασθενών με νόσο του Alzheimer (AD). Στα αποτελέσματα, έδειξαν ότι με την ενσωμάτωση διαφόρων βιοδεικτών, η διαδικασία ταξινόμησης βελτιώθηκε. Χρησιμοποίησαν έναν αλγόριθμο elastic net για να καθορίσουν τη σημασία 40 πιθανών βιοδεικτών. Η επιλογή χαρακτηριστικών με τη χρήση των σταθμισμένων εξόδων του αλγορίθμου και της μεθόδου Elbow ανέδειξε 14 βέλτιστα χαρακτηριστικά για να σχηματίσουν έναν ολοκληρωμένο δείκτη που ονομάζεται δείκτης mAChR. Χρησιμοποιώντας αυτόν τον ολοκληρωμένο δείκτη, η εκπαίδευση στα ίδια δεδομένα απέδωσε ακρίβεια 95 %, με AUC 0,98. Μετά από 100 επαναλήψεις διασταυρούμενης επικύρωσης, η ακρίβεια ήταν 90 ± 2%. Αντίθετα, ο μοναδικός καλύτερος βιοδείκτης (σχετικό δέλτα) εμφάνισε ακρίβεια εκπαίδευσης 82% και AUC 0,87 και ακρίβεια διασταυρούμενης επικύρωσης 79 ± 2%. Οι συγγραφείς υποστηρίζουν πώς η ερευνητική τους μέθοδος θα μπορούσε να αποδειχθεί ανεκτίμητη στην ανάπτυξη φαρμάκων, ιδίως στον τομέα των φαρμάκων που αποσκοπούν στην αντιμετώπιση των επιδράσεων παραγόντων όπως η σκοπολαμίνη, προσφέροντας δυνητικά θεραπείες για τη νόσο του Αλτσχάιμερ και τις γνωστικές διαταραχές που σχετίζονται με τη σχιζοφρένεια.

Τέλος, οι Fouad και Labib [11] παρουσίασαν ένα αυτοματοποιημένο σύστημα υπολογιστή που έχει σχεδιαστεί για την ανίχνευση της νόσου Alzheimer χρησιμοποιώντας σήματα ΗΕΓ που λαμβάνονται από τρία μόνο ηλεκτρόδια του κεντρικού λοβού. Μετά την προεπεξεργασία και την εφαρμογή μετασχηματισμών wavelet στα δεδομένα ΗΕΓ για την εξαγωγή στατιστικών χαρακτηριστικών, αξιολόγησαν διάφορους παραδοσιακούς ταξινομητές μηχανικής μάθησης, με τους ταξινομητές Naïve Bayes και LSVM να έχουν εξαιρετικά καλές επιδόσεις, επιτυγχάνοντας ακρίβεια 96,55% και 95,69% στο πρώτο σύνολο δεδομένων, αντίστοιχα. Ειδικότερα, η μελέτη τους καταδεικνύει τις ανώτερες διαγνωστικές δυνατότητες της προσέγγισης βαθιάς μάθησης χρησιμοποιώντας τον ταξινομητή "ResNet-50" Convolutional Neural Network, ο οποίος πέτυχε αξιοσημείωτη ακρίβεια 97,8261% στο πρώτο σύνολο δεδομένων. Τα αποτελέσματα τους υπογραμμίζουν την αποτελεσματικότητα της βαθιάς μάθησης στην ενίσχυση της έγκαιρης ιατρικής διάγνωσης, καθώς το ResNet-50 ξεπέρασε ακόμη και τον κορυφαίο παραδοσιακό ταξινομητή με τις καλύτερες επιδόσεις. Επισημαίνουν τη σημασία της έγκαιρης διάγνωσης στη νόσο Αλτσχάιμερ, λαμβάνοντας υπόψη ότι ακόμη και έμπειροι ειδικοί μπορεί να χάσουν το 10-15% των περιπτώσεων. Υποστηρίζουν ότι ενώ άλλες διαγνωστικές διαδικασίες, όπως η αξονική τομογραφία, η μαγνητική τομογραφία και το PET, μπορεί να είναι χρονοβόρες και δαπανηρές, η χρήση των σημάτων ΗΕΓ προσφέρει μια εναλλακτική λύση με χαμηλότερο κόστος και ταχύτερο χρόνο. Η μεθοδολογία τους διακρίνει με επιτυχία τα σήματα EEG από ασθενείς με Αλτσχάιμερ, από εκείνα των υγιών ατόμων ελέγχου, υποδεικνύοντας επίσης ότι η εφαρμογή αλγορίθμων συνόλου (ensemble) θα μπορούσε να ενισχύσει περαιτέρω την απόδοση ταξινόμησης.

2.1.5 Παρακολούθηση των σταδίων ύπνου

Όπως προαναφέρθηκε και στην εισαγωγή της παρούσας ενότητας, η ανάλυση των δεδομένων ΗΕΓ με τεχνικές μηχανικής μάθησης χρησιμοποιείται για την εξέταση των καταστάσεων του ύπνου. Ο Abdulla κ.ά. [12] εμβαθύνουν στη σημασία του ύπνου για τη διατήρηση της ψυχικής και σωματικής υγείας του ανθρώπου. Επισημαίνουν την ανάγκη για μια αποτελεσματική μέθοδο βαθμολόγησης των σταδίων ύπνου του ηλεκτροεγκεφαλογραφήματος (ΗΕΓ), ώστε να βοηθηθούν οι επαγγελματίες του ιατρικού κλάδου στην έγκαιρη διάγνωση των διαταραχών του ύπνου. Η έρευνα τους παρουσίασε μια νέα προσέγγιση για την ταξινόμηση των σημάτων ΗΕΓ που αντιστοιχούν σε στάδια ύπνου χρησιμοποιώντας γραφήματα συσχέτισης. Η βασική μεθοδολογία τους περιλαμβάνει την τμηματοποίηση μιας καταγραφής ΗΕΓ διάρκειας 30 δευτερολέπτων σε υποτμήματα, τη μείωση της διαστασιμότητας κάθε υποτμήματος με τη χρήση ενός στατιστικού μοντέλου και στη συνέχεια, τη μετατροπή ολόκληρου του τμήματος ΗΕΓ σε γράφημα. Για να γίνει αυτό, κάθε υποτμήμα λειτουργεί ως κόμβος στο γράφημα και οι συνδέσεις μεταξύ αυτών των κόμβων δημιουργούνται με βάση το συντελεστή συσχέτισής τους. Η μορφή του γράφου (επίπεδα, βάθος) χρησιμεύει ως χαρακτηριστικά εισόδου για έναν ταξινομητή συνόλου (ensemble learning). Προκειμένου να διακριθούν τα πιο αποτελεσματικά χαρακτηριστικά, διερεύνησαν διάφορους αλγόριθμους που βασίζονται σε γράφους συσχέτισης. Πολλαπλές τεχνικές ταξινόμησης, συμπεριλαμβανομένων του Least Square Vector Machine (LS-SVM), του k-means, του Naïve Bayes, του Fuzzy C-means, του k-nearest και της λογιστικής παλινδρόμησης αξιολογήθηκαν με τη μέθοδο χρήσης αποφάσεων με πολλαπλά κριτήρια (Multi-Criteria Decision-Making (MCDM)). Μετά από αυτή την αξιολόγηση, οι τέσσερις κορυφαίες μέθοδοι - LS-SVM, Naïve Bayes, λογιστική παλινδρόμηση και k-nearest - συγχωνεύθηκαν σε ένα σύνολο ταξινομητών που σχεδιάστηκε για την ταξινόμηση των χαρακτηριστικών του γραφήματος. Στη συνέχεια, τα αποτελέσματα από αυτή τη μέθοδο συνόλου αντιπαραβλήθηκαν με τα αποτελέσματα των μεμονωμένων ταξινομητών. Τα πειραματικά τους ευρήματα απέδειξαν ότι η προτεινόμενη μέθοδος, η οποία ταξινομεί σήματα ύπνου EEG με βάση γραφήματα συσχέτισης, υπερτερεί των υφιστάμενων σύγχρονων τεχνικών στην ταξινόμηση σταδίων ύπνου.

Μία αντίστοιχη έρευνα στην ίδια κατηγορία εκμεταλλεύεται διαφορετικά τα δεδομένα ΗΕΓ για την εξαγωγή συμπερασμάτων. Συγκεκριμένα ο Musa Peker [13], εισήγαγε μια υβριδική μεθοδολογία μηχανικής μάθησης για την αυτόματη βαθμολόγηση του ύπνου με χρήση σημάτων ηλεκτροεγκεφαλογραφίας (ΗΕΓ) ενός καναλιού. Η προσέγγιση του συνδυάζει μη γραμμικά χαρακτηριστικά που έχουν σύνθετη τιμή (complex-valued nonlinear features (CVNF)) με ένα νευρωνικό δίκτυο που έχει σύνθετες τιμές (complex-valued neural network (CVANN)). Αρχικά, προσδιορίστηκαν εννέα μη γραμμικά χαρακτηριστικά που συνήθως προτιμώνται για την ταξινόμηση σήματος ΗΕΓ. Στη συνέχεια, τα χαρακτηριστικά αυτά μετατράπηκαν σε μορφή αριθμών μέσω μιας μεθόδου κωδικοποίησης φάσης, με αποτέλεσμα ένα νέο σύνολο χαρακτηριστικών σύνθετων τιμών, προσαρμοσμένο για τη βαθμολόγηση του ύπνου. Αυτό το σύνολο χαρακτηριστικών χρησιμοποιήθηκε ως είσοδος για τον αλγόριθμο CVANN. Η διαδικασία αξιολόγησης ενσωμάτωσε διάφορες στατιστικές παραμέτρους και βασίστηκε σε δύο πρότυπα ύπνου: Rechtschaffen & Kales (R&K) και American Academy of Sleep Medicine (AASM). Η καινοτομία της έρευνας του έγκειται στην υβριδική μέθοδο CVNF+CVANN, που συνδυάζει έναν ισχυρό ταξινομητή με ένα ισχυρό σύνολο χαρακτηριστικών. Η ανάπτυξη ενός συνόλου χαρακτηριστικών σύνθετης αξίας από μη γραμμικά χαρακτηριστικά πραγματικής αξίας με χρήση κωδικοποίησης φάσης αποτέλεσε σημαντική καινοτομία. Στη φάση της ταξινόμησης χρησιμοποίησε τον αλγόριθμο CVANN. Η προτεινόμενη μέθοδος πέτυχε ποσοστά ακρίβειας 91,57% και 93,84% για τα πρότυπα R&K και AASM, αντίστοιχα. Το αποτέλεσμα αυτής της μελέτης είναι ένα πολλά υποσχόμενο σύστημα πρόβλεψης που μπορεί να ενσωματωθεί σε διαγνωστικά συστήματα με τη βοήθεια υπολογιστή, ενώ παράλληλα ανέδειξαν την υψηλή ακρίβεια της μεθόδου και τη δημιουργία ενός νέου συνόλου χαρακτηριστικών με σύνθετες τιμές για την διάκριση των σταδίων του ύπνου.

2.1.6 *Παρακολούθηση διάρκειας προσοχής και ανίχνευση των επιπέδων εγρήγορσης ή υπνηλίας*

Η Inan Aci κ.ά. [14] προσπάθησαν να διερευνήσουν τις προκλήσεις που προκύπτουν σε περιβάλλοντα όπου οι άνθρωποι έχουν σε μεγάλο βαθμό παθητικό ρόλο, γεγονός που συχνά οδηγεί σε μειωμένη εστίαση και συγκέντρωση. Ανέπτυξαν μια παθητική διεπαφή εγκεφάλου-υπολογιστή (Brain Compuer Interface (BCI)) σχεδιασμένη να παρακολουθεί τις καταστάσεις νοητικής προσοχής ενός ατόμου, συγκεκριμένα τις καταστάσεις "εστιασμένη", "μη εστιασμένη" και "νυσταγμένη". Αυτό πραγματοποιήθηκε χρησιμοποιώντας δεδομένα ηλεκτροεγκεφαλογραφικής απεικόνισης (ΗΕΓ) και ανάλυση δεδομένων μηχανικής μάθησης. Για τη συλλογή των σχετικών δεδομένων, χρησιμοποιήσαν ένα συμβατικό μηχάνημα καταγραφής ΗΕΓ δραστηριότητας, συγκεντρώνοντας 25 ώρες καταγραφών ΗΕΓ από πέντε συμμετέχοντες. Εντόπισαν ότι οι μεταβολές της δραστηριότητας του ΗΕΓ σε συγκεκριμένες περιοχές του εγκεφάλου και ζώνες συχνοτήτων συσχετίστηκαν με τις αλλαγές στις καταστάσεις προσοχής των συμμετεχόντων. Χρησιμοποιώντας τη μέθοδο SVM απέδειξαν τη δυνατότητα εντοπισμού αυτών των καταστάσεων προσοχής με ποσοστά ακρίβειας έως και 96,70% και μέσο όρο 91,72% σε ελεγχόμενες συνθήκες. Σε σύγκριση με άλλες μεθόδους, όπως η μέθοδος k-Nearest Neighbor και το Adaptive Neuro-Fuzzy System, οι επιδόσεις της SVM ήταν αξιοσημείωτες. Τα ευρήματα αυτά μπορούν να επηρεάσουν συστήματα όπως οι εφαρμογές ασφάλειας οδηγών βοηθώντας στη διατήρηση της συνεχούς προσοχής. Η προσέγγιση τους είναι ευέλικτη και προσφέρει ιδέες για την αναπαράσταση τέτοιων καταστάσεων σε σήματα EEG, ενώ μπορεί επίσης να επεκταθεί σε κλινικές εφαρμογές, όπως η παρακολούθηση του διφασματικού δείκτη (Bispectral Index Monitoring (BIS)) που χρησιμοποιείται για την παρακολούθηση του βάθους αναισθησίας κατά τη διάρκεια χειρουργικών επεμβάσεων.

2.1.7 *Ανάλυση Συναισθήματος*

Ο Wang κ.ά. [15] ασχολήθηκαν με την ταξινόμηση συναισθημάτων από δεδομένα EEG μέσω μηχανικής μάθησης. Αρχικά, αξιολόγησαν τρία χαρακτηριστικά του EEG: φάσμα ισχύος, κυματομορφή και η μη γραμμική δυναμική ανάλυσή του. Μετέπειτα χρησιμοποίησαν μια μέθοδο εξομάλυνσης των χαρακτηριστικών με τη χρήση ενός γραμμικού δυναμικού συστήματος για τη μείωση του θορύβου. Τέλος, εφάρμοσαν πολλαπλή μάθηση για την ανίχνευση και την οπτικοποίηση των αλλαγών της συναισθηματικής κατάστασης με την πάροδο του χρόνου. Τα πειράματά τους, που περιλάμβαναν αποσπάσματα ταινιών για την πρόκληση συναισθημάτων σε έξι συμμετέχοντες, αποκάλυψαν το χαρακτηριστικό «φάσμα ισχύος» ως το πιο χρήσιμο. Επίσης, η χρήση του γραμμικού δυναμικού συστήματος βελτίωσε σημαντικά την ακρίβεια ταξινόμησης. Για τη μείωση της διαστασιμότητας, χρησιμοποίησαν τις μεθόδους PCA, LDA και CFS. Η μέθοδος LDA πέτυχε την υψηλότερη ακρίβεια ταξινόμησης 91,77% όταν ο αριθμός των διαστάσεων μειώθηκε σε 30 χαρακτηριστικά. Η μέθοδος CFS εντόπισε συγκεκριμένες περιοχές EEG που συνδέονται με συναισθηματικές αντιδράσεις, αναδεικνύοντας μοτίβα σε όλους τους λοβούς του εγκεφάλου και τις ζώνες συχνοτήτων. Οι συγγραφείς προτείναν ότι η μεθοδολογία τους μπορεί να επιτρέψει την απεικόνιση της συναισθηματικής κατάστασης σε πραγματικό χρόνο μέσω του EEG και υπαινίσσονται την επέκταση της έρευνας με περισσότερα υποκείμενα και πολυτροπική ανάλυση.

Επιπρόσθετα, σε μία σημαντική έρευνα που εμπλέκει Βαθιά Νευρωνικά Δίκτυα, η Hassouneh κ.ά. [16], παρουσίασαν ένα σύστημα αναγνώρισης συναισθημάτων σε πραγματικό χρόνο, το οποίο στοχεύει κυρίως στην παροχή βοήθειας σε άτομα με σωματική αναπηρία και παιδιά με αυτισμό, αναλύοντας τα σημεία αναφοράς του προσώπου και τα σήματα ηλεκτροεγκεφαλογραφήματος (EEG). Χρησιμοποιώντας έναν μοναδικό αλγόριθμο που αναγνωρίζει τα συναισθήματα μέσω εικονικών σημείων μέσω ενός αλγορίθμου οπτικής ροής, το σύστημα τους επιδεικνύει ευελιξία, λειτουργώντας αποτελεσματικά σε διαφορετικές συνθήκες φωτισμού, τόνους δέρματος και ακόμη και με περιστροφές

του κεφαλιού έως και 25°. Μέσω της χρήσης των ταξινομητών νευρωνικών δικτύων συνελικτικού τύπου (CNN) και LSTM, η μελέτη τους πέτυχε ένα εντυπωσιακό ποσοστό αναγνώρισης 99,81% για τα συναισθήματα με χρήση χαρακτηριστικών του προσώπου και ένα ποσοστό 87,25% για την ανίχνευση συναισθημάτων με βάση το EEG. Το σύστημα λειτουργεί σε πραγματικό χρόνο για τα χαρακτηριστικά προσώπου και σε περιβάλλον εκτός σύνδεσης για τα ακατέργαστα δεδομένα EEG, καθιστώντας αναγκαία τη χρήση συσκευών εγκεφαλικών καταγραφών EPOC+ για τα υποκείμενα. Η ολοκληρωμένη αξιολόγηση αυτού του συστήματος περιελάμβανε δεδομένα που συλλέχθηκαν από προπτυχιακούς φοιτητές του Πανεπιστημίου του Κουβέιτ. Τα ποσοστά αναγνώρισης αναδεικνύουν τις δυνατότητες του συστήματος σε διάφορες εφαρμογές, από την υποβοήθηση της κατανόησης των συναισθημάτων των ατόμων με αναπηρία, έως τη μέτρηση των συναισθηματικών αντιδράσεων του κοινού, ακόμη και σε εξατομικευμένα περιβάλλοντα ηλεκτρονικής μάθησης. Ενώ η ανίχνευση με βάση τα χαρακτηριστικά του προσώπου απέδωσε μεγαλύτερη ακρίβεια, η ανίχνευση συναισθημάτων μέσω σημάτων EEG μπορεί να βελτιωθεί περαιτέρω με τη συλλογή περισσότερων δεδομένων και τη ταυτόχρονη βελτίωση των τεχνικών εξαγωγής χαρακτηριστικών.

2.1.8 Μαθησιακές Δυσκολίες (Δυσλεξία)

Οι μαθησιακές δυσκολίες αποτελούν ένα αντικείμενο μελέτης στην επόμενη έρευνα που χρησιμοποιεί διαφορετικές μορφές του αλγορίθμου SVM. Ο Parmar κ.ά. [17] προτείναν μια καινοτόμο μέθοδο για την ανίχνευση της δυσλεξίας, μιας νευροαναπτυξιακής διαταραχής, χρησιμοποιώντας σήματα ΗΕΓ. Η μεθοδολογία τους περιλαμβάνει την προεπεξεργασία των ακατέργαστων δεδομένων EEG, την εξαγωγή χαρακτηριστικών, την ομαδοποίησή τους και στη συνέχεια, τη χρήση αυτών των χαρακτηριστικών σε έναν ταξινομητή βασισμένο στη μηχανική μάθηση. Χρησιμοποίησαν έναν συνδυασμό χαρακτηριστικών Statistical, Hjorth, Frequency και Katz Fractal Dimension για την ομαδοποίηση των χαρακτηριστικών. Αντί της αξιοποίησης δεδομένων από όλα τα κανάλια EEG, ομαδοποιούνται και αναλύονται επιλεγμένα κανάλια, μειώνοντας έτσι τον αριθμό των απαιτούμενων ηλεκτροδίων. Στη συνέχεια, εφάρμοσαν έναν ταξινομητή SVM με διαφορετικούς μη γραμμικούς πυρήνες, συμπεριλαμβανομένου του πυρήνα Gauss (RBF), του πολυωνυμικού πυρήνα και του σιγμοειδούς πυρήνα, για την ανίχνευση της δυσλεξίας. Τα πιο ελπιδοφόρα αποτελέσματα, με ακρίβεια 62,4%, επιτεύχθηκαν με τη χρήση του πυρήνα RBF Kernel σε μη γλωσσικά ερεθίσματα για την παραγωγή των σημάτων ΗΕΓ. Αυτό κρίθηκε σημαντικό, ιδίως όταν δοκιμάστηκε σε ένα μεγάλο σύνολο δεδομένων 391 συμμετεχόντων. Η έρευνα υπογραμμίζει περαιτέρω τη σημασία της έγκαιρης ανίχνευσης της δυσλεξίας, ώστε να παρέχεται εξειδικευμένη υποστήριξη στους μαθητές που πάσχουν. Ο πυρήνας RBF Kernel, έδειξε τις περισσότερες δυνατότητες με τα μειωμένα ποσοστά σφαλμάτων του. Τα ευρήματα τους υποδηλώνουν ότι, ενώ η ομαδοποίηση των καναλιών EEG μπορεί να απλοποιήσει τη διαδικασία, δεν μπορεί απαραίτητα να επηρεάσει θετικά την ορθότητα. Η παρούσα έρευνα προσφέρει ένα σημαντικό βήμα προόδου στον τομέα της ανίχνευσης της δυσλεξίας με τη χρήση σημάτων EEG, ενώ η μελλοντική εργασία μπορεί να ωφεληθεί από τη διερεύνηση άλλων μοναδικών χαρακτηριστικών και τη βελτιστοποίηση των υπολογιστικών προσεγγίσεων.

Σε ακόμα μία έρευνα που ασχολείται με τη Δυσλεξία, ο Perera κ.ά. [18], μελέτησαν τη χρήση σημάτων ΗΕΓ για τον εντοπισμό μοναδικών μοτίβων εγκεφαλικών κυμάτων σε ενήλικες με δυσλεξία κατά τη διάρκεια δραστηριοτήτων γραφής και πληκτρολόγησης. Χρησιμοποιώντας μηχανική μάθηση, συγκεκριμένα Cubic SVM, ανέλυσαν τα δεδομένα EEG, τα οποία υποβλήθηκαν σε προεπεξεργασία για τη μείωση των ανωμαλιών και στη συνέχεια αποσυντέθηκαν σε συγκεκριμένες υπο-ζώνες συχνοτήτων. Κατά την εξέταση των εργασιών γραφής, παρατηρήθηκε ότι οι ενήλικες με δυσλεξία παρήγαγαν ξεχωριστά μοτίβα εγκεφαλικών κυμάτων σε σύγκριση με τα υγιή άτομα. Ο βέλτιστος ταξινομητής για αυτή την εργασία απέδωσε μέγιστη ακρίβεια επικύρωσης (VA) 71,88%, ευαισθησία 76,47% και ειδικότητα 66,67% για τη γραφή. Για την εργασία πληκτρολόγησης, εξετάστηκε ένα σύνολο ταξινομητών σε διαφορετικές περιοχές του εγκεφάλου επιτυγχάνοντας το υψηλότερο VA

78,13%. Τα πιο κρίσιμα κανάλια ΗΕΓ που διέκριναν τα άτομα με δυσλεξία κατά τη διάρκεια της πληκτρολόγησης ήταν τα F5, F3, Fz, F4 και F6. Συμπερασματικά, η έρευνα αποκάλυψε περιοχές και κανάλια στην ανάλυση σήματος ΗΕΓ που δεν είχαν αναφερθεί προηγουμένως και τα οποία προσδιορίζουν σαφώς τους ενήλικες με δυσλεξία κατά τη διάρκεια εργασιών γραφής και πληκτρολόγησης, αποκαλύπτοντας την εμπρόσθια μετωπιαία περιοχή ως τη σημαντικότερη στην ανάλυση τέτοιων σημάτων. Η μελέτη τους προτείνει την επέκταση αυτής της έρευνας σε διαφορετικές δημογραφικές ομάδες, όπως αριστερόχειρες ή άτομα κάτω των 18 ετών και τη διερεύνηση άλλων μαθησιακών δυσκολιών, όπως η δυσκολίες στη γραφή και την αρίθμηση.

2.1.9 Τεχνικές αναλύσεις για μελλοντικές εφαρμογές

Ο Amin κ.ά. [19] εισήγαγαν μια μέθοδο ταξινόμησης σημάτων EEG με χρήση του διακριτού μετασχηματισμού κυματιδίων (wavelets) για την εξαγωγή χαρακτηριστικών. Η σχετική ενέργεια κυματιδίου υπολογίζεται από τα σήματα ΗΕΓ μετά την εφαρμογή του διακριτού μετασχηματισμού κυματιδίου. Η έρευνα εξετάζει συγκεκριμένα τα σήματα ΗΕΓ κατά τη διάρκεια μιας σύνθετης γνωστικής εργασίας, του προηγμένου προοδευτικού μετρικού τεστ του Raven και τα αντιπαραβάλλει με τα σήματα ΗΕΓ σε κατάσταση ηρεμίας με ανοιχτά μάτια. Για την ταξινόμηση αυτών των σημάτων EEG χρησιμοποίησαν τέσσερις ταξινομητές μηχανικής μάθησης: Support Vector Machine (SVM), Multi-Layer Perceptron (MLP), K-Nearest Neighbor (K-NN) και Naïve Bayes. Μεταξύ αυτών των ταξινομητών, οι SVM, MLP και K-NN πέτυχαν ποσοστό ακρίβειας άνω του 98% όταν εξέτασαν τις υποζώνες συχνοτήτων A4 (0,53-3,06 Hz) και D4 (3,06-6,12 Hz) των εγκεφαλικών σημάτων. Τα ευρήματα τους υποδηλώνουν ότι η ενέργεια κυματιδίου είναι αποτελεσματική στην ταξινόμηση σημάτων EEG κατά τη διάρκεια σύνθετων γνωστικών εργασιών. Τα αποτελέσματα έχουν δυνητικές εφαρμογές σε κλινικά περιβάλλοντα για τη διάγνωση καταστάσεων όπως η επιληψία, η κατάθλιψη και το άγχος. Τονίζεται ιδιαίτερα η αποτελεσματικότητα της τεχνικής στον εντοπισμό αλλαγών σε μη στάσιμα σήματα ΗΕΓ λόγω των χαρακτηριστικών εντοπισμού του μετασχηματισμού wavelet. Η μελέτη υπογραμμίζει επίσης τις δυνατότητες της προτεινόμενης προσέγγισης εξαγωγής χαρακτηριστικών για χρήση σε εφαρμογές διεπαφής εγκεφάλου-υπολογιστή, όπου ασθενείς με κινητική αναπηρία θα μπορούσαν να ελέγχουν εξωτερικές συσκευές χρησιμοποιώντας τη γνωστική ισχύ.

Ο Al Zoubi κ.ά. [20] ασχολήθηκαν με την πρόβλεψη της ηλικίας ενός ατόμου μέσω των σημάτων EEG του εγκεφάλου, χρησιμοποιώντας μια προσέγγιση μηχανικής μάθησης. Η εκτίμηση του εγκεφαλικού χάσματος ηλικίας (BrainAGE), η οποία αντιπροσωπεύει τη διαφορά μεταξύ της εκτιμώμενης και της χρονολογικής ηλικίας ενός ατόμου, μελετήθηκε κυρίως με τη χρήση τεχνικών μαγνητικής τομογραφίας. Στην παρούσα έρευνα, τα σήματα EEG χρησιμοποιήθηκαν σε συνδυασμό με μια εκτεταμένη διαδικασία εξαγωγής χαρακτηριστικών και ένα ισχυρό πλαίσιο μηχανικής μάθησης για την πρόβλεψη της χρονολογικής ηλικίας. Χρησιμοποιώντας δεδομένα από 468 συμμετέχοντες, τα οποία περιλάμβαναν υγιή άτομα και άτομα με διαταραχές διάθεσης, άγχους, διατροφής και χρήσης ουσιών, χρησιμοποίησαν μια προσέγγιση εμφωλευμένης διασταυρούμενης επικύρωσης και μάθηση stack-ensemble. Από τα μοντέλα μηχανικής μάθησης που χρησιμοποιήσαν, ο καλύτερος μεμονωμένος αλγόριθμος ήταν η παλινδρόμηση διανυσμάτων υποστήριξης (Support Vector Regression (SVR)), επιτυγχάνοντας τιμή $R^2 = 0,34$. Ωστόσο, η τιμή αυτή βελτιώθηκε ελαφρώς με την προσέγγιση stack-ensemble, η οποία πέτυχε τιμή $R^2 = 0,37$. Η συσχέτιση μεταξύ της προβλεπόμενης και της πραγματικής ηλικίας ήταν σημαντική με $r = 0,6$. Η μελέτη εντόπισε ότι οι προγνωστικοί παράγοντες ηλικίας ήταν διασκορπισμένοι σε διάφορους τύπους χαρακτηριστικών, καταδεικνύοντας τις δυνατότητες των σημάτων EEG για την αξιόπιστη εκτίμηση της χρονολογικής ηλικίας και του BrainAGE. Μια αξιοσημείωτη παρατήρηση ήταν ότι οι παράγοντες πρόβλεψης της ηλικίας κατανέμονται σε διάφορους τύπους χαρακτηριστικών και ζώνες συχνοτήτων, γεγονός που υποδηλώνει ότι κανένα μοναδικό χαρακτηριστικό δεν κυριαρχεί στην πρόβλεψη της ηλικίας. Επιπλέον οι διάφοροι

προγνωστικοί παράγοντες EEG εκδηλώνουν διαφορετικές επιδράσεις ανάλογα με την ηλικία. Η ικανότητα ακριβούς προσδιορισμού του BrainAGE από σήματα EEG μπορεί να είναι καθοριστική σε διάφορες κλινικές και ερευνητικές εφαρμογές.

2.2 Έρευνες για οφθαλμική κατάσταση ασθενών

Στη συγκεκριμένη ενότητα παρουσιάζονται υπάρχουσες έρευνες που σχετίζονται με το σύνολο δεδομένων που χρησιμοποιείται στην παρούσα διπλωματική και αφορά την οφθαλμική κατάσταση ασθενών. Παρουσιάζονται εφαρμογές Μηχανικής Μάθησης σε δεδομένα ΗΕΓ που μελετούν εάν οι ασθενείς έχουν τους οφθαλμούς τους κλειστούς ή ανοιχτούς.

Οι Reddy και Behera [21] ασχολήθηκαν με εφαρμογές για άτομα με κινητικά προβλήματα και κυριότερα την ανίχνευση της προσοχής των οδηγών. Εξέτασαν τη χρήση αρχιτεκτονικών βαθιάς μάθησης για την ταξινόμηση δεδομένων EEG με έμφαση στην ανίχνευση της κατάστασης των ματιών. Ενώ οι τεχνολογικές εξελίξεις έχουν επιτρέψει την ενισχυμένη επεξεργασία σήματος σε πραγματικό χρόνο σε συστήματα Brain Computer Interface (BCI), η συστηματική εφαρμογή της Βαθιάς Μάθησης (Deep Learning) στην ανάλυση δεδομένων EEG ήταν περιορισμένη. Στη μελέτη τους, χρησιμοποιήθηκαν διάφορα μοντέλα Deep Learning, με στόχο όχι μόνο να επιτευχθούν ακρίβειες παρόμοιες με κορυφαίους ταξινομητές, όπως οι ταξινομητές K* του Roesler και οι ταξινομητές των Cameron et al. αλλά και να διασφαλιστεί η ταχεία επεξεργασία που είναι κατάλληλη για ένα online πλαίσιο BCI. Οι αρχιτεκτονικές που διερευνήθηκαν περιελάμβαναν νευρωνικά δίκτυα πολλαπλών επιπέδων, βαθιά δίκτυα γνώσεων και μάσκες απόκλισης σε βαθιά νευρωνικά δίκτυα. Τα ευρήματά τους υπογραμμίζουν ότι οι βαθιές νευρωνικές αρχιτεκτονικές που διερεύνησαν, παρουσιάζουν σημαντικά ταχύτερες ταχύτητες ταξινόμησης με συγκρίσιμα, αν όχι καλύτερα, ποσοστά ακρίβειας. Για παράδειγμα, η αρχιτεκτονική με τις καλύτερες επιδόσεις από την έρευνά τους απέδωσε ακρίβεια 97,5% και ταχύτητα ταξινόμησης τουλάχιστον 1000 δειγμάτων ανά δευτερόλεπτο. Αυτός ο γρήγορος ρυθμός ταξινόμησης είναι ιδιαίτερα υποσχόμενος για εφαρμογές που απαιτούν γρήγορους χρόνους απόκρισης, όπως ο έλεγχος BCI με βάση το κλείσιμο των ματιών. Για μελλοντικές έρευνες, οι συγγραφείς προτείνουν τη διερεύνηση συνόλων βαθιών νευρωνικών μοντέλων για την ταξινόμηση άλλων βιολογικών δραστηριοτήτων και την πιθανή ενσωμάτωση χρονικών RBM για να ληφθεί υπόψη η φύση των χρονοσειρών των δεδομένων. Οι προσπάθειές τους θέτουν ένα θεμελιώδες βήμα προς την ανάπτυξη προηγμένων λύσεων BCI για διάφορες εφαρμογές, όπως συστήματα ανίχνευσης υπνηλίας οδηγών.

Ο Narejo κ.ά. [22] επικεντρώθηκα στην πρόβλεψη των καταστάσεων των ματιών από σήματα EEG χρησιμοποιώντας αρχιτεκτονικές βαθιάς μάθησης για τη βελτίωση των μοντέλων ταξινομητών. Για το σκοπό αυτό χρησιμοποίησαν το Deep Belief Network (DBN) και τους Stacked AutoEncoders (SAE). Ενώ και οι δύο ταξινομητές παρουσίασαν ενθαρρυντικές επιδόσεις, ένα συγκεκριμένο μοντέλο SAE ξεπέρασε τα άλλα, συμπεριλαμβανομένου του μοντέλου DBN, επιτυγχάνοντας εντυπωσιακή ακρίβεια 98,9% με ποσοστό σφάλματος μόλις 1,1% στο σύνολο δοκιμών. Αυτή η ακρίβεια ξεπέρασε τα μοντέλα που είχαν καταγραφεί προηγουμένως σε υπάρχουσες έρευνες. Η μελέτη υπογραμμίζει ότι η συγκεκριμένη προσέγγιση της βαθιάς μάθησης της μη επιβλεπόμενης προεκπαίδευσης που ακολουθείται από επιβλεπόμενη βελτιστοποίηση προσφέρει καλύτερη γενίκευση σε σύγκριση με μοντέλα που αρχικοποιούνται με τυχαίο τρόπο. Κορυφαίο στοιχείο της έρευνάς τους είναι η ανώτερη απόδοση του μοντέλου SAE 2. Παρείχαν επίσης αποδείξεις για τις δυνατότητες των βαθιών αρχιτεκτονικών για την ταξινόμηση της κατάστασης των ματιών με βάση το EEG και υπέδειξαν ότι θα μπορούσαν να γίνουν περαιτέρω βελτιώσεις με τη χρήση Denoising και Contrastive Autoencoders. Η εργασία αυτή θέτει τα θεμέλια για μελλοντικές διερευνήσεις στο

πεδίο των εφαρμογών διεπαφής εγκεφάλου-υπολογιστή (BCI), ιδίως σε σενάρια όπου ακούσιες κινήσεις των ματιών ή ανοιγοκλείσιμο των ματιών εισάγουν κρίσιμα σφάλματα μέτρησης.

Σε μία διαφορετική προσέγγιση, η Wang κ.ά. [23] εξέτασαν τις δυνατότητες της Incremental Attribute Learning (IAL) για την αναγνώριση της κατάστασης των ματιών του EEG. Η IAL είναι μια νέα προσέγγιση μηχανικής μάθησης που εισάγει προοδευτικά και εκπαιδεύει διαδοχικά χαρακτηριστικά. Ενώ έχει διαπιστωθεί η σκοπιμότητα της IAL για την αναγνώριση προτύπων, η εφαρμογή της σε προβλήματα χρονοσειρών, όπως η ταξινόμηση της κατάστασης των ματιών EEG, παρέμεινε ανεξερεύνητη. Για να προσδιοριστεί η αποτελεσματικότητα του IAL, αρχικά εξήχθησαν χαρακτηριστικά από ακατέργαστα δεδομένα EEG. Αυτά τα χαρακτηριστικά στη συνέχεια οργανώθηκαν σε μια συγκεκριμένη ακολουθία χρησιμοποιώντας τη μέθοδο ταξινόμησης χαρακτηριστικών IAL, η οποία βασίζεται στη διακριτή τιμή κάθε χαρακτηριστικού. Κατά τη φάση της εκπαίδευσης, τα χαρακτηριστικά εισήχθησαν διαδοχικά σε ένα νευρωνικό σύστημα πρόβλεψης, σύμφωνα με αυτήν την προκαθορισμένη σειρά. Συγκρίνοντας το IAL με τις συμβατικές μεθόδους εκπαίδευσης, τα αποτελέσματα ήταν σαφώς υπέρ του IAL. Συγκεκριμένα, η πρώτη προσέγγιση IAL πέτυχε το χαμηλότερο ποσοστό σφάλματος ταξινόμησης, σημαντικά καλύτερο από τη συμβατική μέθοδο εκπαίδευσης κατά παρτίδες, η οποία είχε το υψηλότερο ποσοστό σφάλματος 30,63%. Σε σύγκριση με το πείραμα του Rösler χρησιμοποιώντας ένα πολυστρωματικό perceptron, το οποίο είχε ποσοστό σφάλματος άνω του 30%, οι προσεγγίσεις IAL είχαν όλες καλύτερες επιδόσεις, με ποσοστά σφάλματος κάτω από 30%. Τα αποτελέσματα τους υπογράμμισαν τρεις βασικές ιδέες: το IAL ξεπέρασε τις παραδοσιακές μεθόδους εκπαίδευσης κατά παρτίδες, η εξαγωγή χαρακτηριστικών με ιδιότητες χρονοσειρών βελτίωσε σημαντικά τα αποτελέσματα ταξινόμησης και η σειρά με την οποία εισήχθησαν τα χαρακτηριστικά στο IAL ήταν κρίσιμη. Διαπίστωσαν ότι η σχέση μεταξύ των δεδομένων χρονοσειρών είναι απαραίτητη για την ανάλυση δεδομένων σε τέτοια προβλήματα ταξινόμησης. Τα ευρήματα της μελέτης αποκαλύπτουν ότι με την κατάλληλη εξαγωγή και οργάνωση των χαρακτηριστικών, η IAL μπορεί να χειριστεί αποτελεσματικά την ταξινόμηση χρονοσειρών και μάλιστα ξεπερνά τις παραδοσιακές μεθόδους, όσον αφορά τα ποσοστά σφάλματος ταξινόμησης.

Ο Chirra κ.ά. [24] εισήγαγαν ένα νέο πλαίσιο που χρησιμοποιεί τη βαθιά μάθηση για την ανίχνευση της υπνηλίας του οδηγού εξετάζοντας την κατάσταση των ματιών του. Χρησιμοποίησαν τον αλγόριθμο ανίχνευσης προσώπου Viola-Jones για να αναγνωρίσουν και να εξαγάγουν την περιοχή των ματιών από τις εικόνες του προσώπου. Στη συνέχεια χρησιμοποίησαν ένα Stacked deep convolution neural network (CNN) για να αντλήσουν χαρακτηριστικά από δυναμικά πλαίσια κλειδιών, τα οποία προσδιορίζονται από ακολουθίες κάμερας, για τη φάση εκμάθησης. Αυτό το πλαίσιο ενσωματώνει ένα επίπεδο SoftMax στο CNN για να κατηγοριοποιήσει την κατάσταση του οδηγού είτε σε κατάσταση ύπνου είτε σε κατάσταση μη ύπνου. Όταν το σύστημα διαπιστώσει ότι ο οδηγός φαίνεται νυσταγμένος, ηχεί συναγερμός. Η αποτελεσματικότητα του προτεινόμενου συστήματος αξιολογήθηκε σε ένα σύνολο δεδομένων, επιτυγχάνοντας ένα εντυπωσιακό ποσοστό ακρίβειας 96,42%, ξεπερνώντας τα παραδοσιακά CNN. Η έρευνα περιελάμβανε δύο είδη πειραμάτων. Το πρώτο πείραμα περιελάμβανε ένα σύνολο δεδομένων 2850 εικόνων με 1450 «νυσταγμένες» και 1400 «μη νυσταγμένες» εικόνες. Για εκπαίδευση, δοκιμή και επικύρωση, επέλεξαν έναν συγκεκριμένο αριθμό εικόνων και από τις δύο κατηγορίες. Το προτεινόμενο μοντέλο πέτυχε ακρίβεια 96,42% στο σύνολο δεδομένων δοκιμής. Στο δεύτερο πείραμα, το εκπαιδευμένο μοντέλο δοκιμάστηκε σε καρέ βίντεο που καταγράφηκαν μέσω κάμερας, με βασικά καρέ που εξήχθησαν από συνεχείς αλληλουχίες βίντεο και δοκιμάστηκαν έναντι των εκπαιδευμένων στατικών εικόνων. Τα αποτελέσματα ήταν ελπιδοφόρα, καθώς το σύστημα ειδοποιούσε συνεχώς με συναγερμό όταν ανίχνευε καταστάσεις υπνηλίας στα μάτια στο βίντεο. Συμπερασματικά, οι συγγραφείς τόνισαν τις δυνατότητες της μεθόδου τους για ανίχνευση υπνηλίας σε πραγματικό χρόνο με βάση την κατάσταση των ματιών.

Παρομοίως, αλλά με διαφορετικό αντικείμενο μελέτης και μεθοδολογία, ο Zeng κ.ά. [25] ασχολήθηκαν με το κρίσιμο ζήτημα της κόπωσης των οδηγών, μια εξέχουσα αιτία οδικών ατυχημάτων. Εισήγαγαν δύο προηγμένα μοντέλα βαθιάς μάθησης, το EEG-Conv και το EEG-Conv-R, για να προβλέψουν την ψυχική κατάσταση των οδηγών χρησιμοποιώντας σήματα ηλεκτροεγκεφαλογραφίας (EEG). Αναλύοντας σενάρια τόσο εντός υποκειμένου (χρησιμοποιώντας δεδομένα από το ίδιο άτομο για εκπαίδευση και δοκιμές), όσο και μεταξύ υποκειμένων (χρησιμοποιώντας δεδομένα από διαφορετικά άτομα), οι ερευνητές διαπίστωσαν ότι τόσο το EEG-Conv όσο και το EEG-Conv-R ξεπέρασαν τους παραδοσιακούς LSTM και SVM ταξινομητές. Συγκεκριμένα, το EEG-Conv-R βρέθηκε ότι είναι πιο ικανό για την πρόβλεψη της νοητικής κατάστασης μεταξύ των υποκειμένων και έδειξε ταχύτερη σύγκλιση σε σύγκριση με το EEG-Conv. Τα ευρήματά τους υποστηρίζουν ότι αυτοί οι προτεινόμενοι ταξινομητές προσφέρουν βελτιωμένες δυνατότητες πρόβλεψης και υπόσχονται εφαρμογές αλληλεπίδρασης εγκεφάλου-υπολογιστή στον πραγματικό κόσμο. Για την ταξινόμηση εντός υποκειμένου, και τα δύο προτεινόμενα μοντέλα, EEG-Conv και EEG-Conv-R, παρείχαν παρόμοια απόδοση, επιτυγχάνοντας μέση ακρίβεια 91,788% και 92,682% αντίστοιχα. Συγκριτικά, το μοντέλο LSTM πέτυχε μέση ακρίβεια 85,132%, και ο ταξινομητής SVM, με εξαγωγή χαρακτηριστικών CSP, διαχειρίστηκε το 88,070%. Για την ταξινόμηση μεταξύ των θεμάτων, το μοντέλο EEG-Conv-R, ειδικότερα, επέδειξε ανώτερη απόδοση με μέση ακρίβεια 84,38%, ξεπερνώντας τα EEG-Conv, SVM και LSTM. Επιπλέον, το EEG-Conv-R εμφάνισε ταχύτερη σύγκλιση στη φάση εκπαίδευσης, που αποδίδεται στα πρόσθετα στρώματα συνέλιξης και στην ενσωμάτωση της υπολειπόμενης μάθησης. Συμπερασματικά, τόνισαν τους περιορισμούς που οφείλονται στην ανεπάρκεια των δειγμάτων εντός του υποκειμένου και σηματοδοτούσαν τη μελλοντική τους πρόθεση να εξερευνήσουν ταξινομήσεις σημάτων EEG πολλαπλών ετικετών αξιοποιώντας τις τεχνικές βαθιάς μάθησής.

Ο Medeiros κ.ά. [26] εστίασαν στην ανάπτυξη ενός αποτελεσματικού συστήματος μηχανικής μάθησης για την ανίχνευση εκούσιων ανοιγμάτων οφθαλμών σε πραγματικό χρόνο, χρησιμοποιώντας μια τυπική web κάμερα. Αυτό το σύστημα είναι ζωτικής σημασίας για την παροχή βοήθειας σε άτομα με Αμυοτροφική Πλάγια Σκλήρυνση (Amyotrophic Lateral Sclerosis -ALS), μια κατάσταση που μειώνει σημαντικά τις κινητικές ικανότητες και παρεμποδίζει την επικοινωνία. Το προτεινόμενο σύστημα αντιμετωπίζει την ανίχνευση βλεφαρίδων ως επέκταση της ταξινόμησης κατάστασης ματιών και χρησιμοποιεί έναν αγωγό (pipeline) που περιλαμβάνει ανίχνευση προσώπου, ευθυγράμμιση προσώπου, εξαγωγή της περιοχής ενδιαφέροντος (ROI) και ταξινόμηση κατάστασης ματιού. Για τη βελτίωση αυτού του αγωγού, εισήχθησαν πρόσθετα μοντέλα: ένας αντισταθμιστής περιστροφής, ένας αξιολογητής της εξεταζόμενης περιοχής ενδιαφέροντος (Region-Of-Interest - ROI) και ένα φίλτρο κινητού μέσου όρου. Αναπτύχθηκαν επίσης δύο νέα σύνολα δεδομένων: το Youtube Eye-state Classification (YEC) και το Autonomous Blink Dataset (ABD). Χρησιμοποίησαν ένα Συνελικτικό Νευρωνικό Δίκτυο (CNN) τον αλγόριθμο (SVM) για τη μοντελοποίηση, με τα αποτελέσματα να φανερώνουν ακρίβεια 97,44% για την ταξινόμηση κατάστασης ματιών και 92,63% F1-Score για ανίχνευση βλεφαρίσματος. Το προτεινόμενο σύστημα ανίχνευσης βλεφαρίσματος έχει σχεδιαστεί για να είναι ανθεκτικό, να λειτουργεί σε πραγματικό χρόνο και να είναι οικονομικά αποδοτικό. Η κύρια λειτουργία του είναι η απρόσκοπτη ενσωμάτωση με τα βοηθητικά συστήματα επικοινωνίας, μετατρέποντας τα ανιχνευμένα βλεφαρίσματα σε σήματα επικοινωνίας. Η απόδοση του συστήματος χαρακτηρίζεται από το χαμηλό ποσοστό σφάλματος και τις δυνατότητές του σε πραγματικό χρόνο και τα ευρήματα της μελέτης επικυρώνουν την αποτελεσματικότητά του στο προτεινόμενο πλαίσιο εφαρμογής. Τέλος, το σύστημα προσφέρει πολλαπλά οφέλη για τους ασθενείς με ALS, συμπεριλαμβανομένης της βελτίωσης της ποιότητας ζωής τους, της διατήρησης των γνωστικών τους δεξιοτήτων και της προώθησης της ανεξαρτησίας, ειδικά σε προχωρημένα στάδια της νόσου. Οι μελλοντικές προσπάθειες σε αυτή τη γραμμή έρευνας μπορεί να περιλαμβάνουν τη διεξαγωγή πειραμάτων σε ασθενείς με ALS και την εστίαση σε περαιτέρω βελτιστοποίηση του λανθάνοντος χρόνου απόκρισης.

Η έρευνα των Karina κ.ά. [27] αφορά την εκπαιδευτική διαδικασία και αποτελεί μία προσπάθεια για τη διερεύνηση της συνεργατικής μάθησης. Συγκεκριμένα, χρησιμοποίησαν πολυτροπικές αναλύτικές μάθησης (Multimodal Learning Analytics - MMLA), όπως είναι η παρακολούθηση των οφθαλμών, φυσιολογικά δεδομένα και ανίχνευση κίνησης, για να προσδιορίσουν τις καταστάσεις συνεργατικής μάθησης μεταξύ ομάδων μαθητών που βρίσκονται μαζί. Χρησιμοποιώντας τεχνικές μηχανικής μάθησης χωρίς επίβλεψη σε ένα σύνολο δεδομένων 84 συμμετεχόντων, εντόπισαν τρεις διακριτές καταστάσεις συνεργασίας, καθεμία από τις οποίες συνδέεται σημαντικά με την απόδοση της εργασίας, την ποιότητα της συνεργασίας και τα μαθησιακά κέρδη. Ο K-Means Clustering εμφανίστηκε ως μια αποτελεσματική μέθοδος για την οριοθέτηση αυτών των συνεργατικών καταστάσεων. Συγκεκριμένα, η κοινή οπτική προσοχή (Joint Visual Attention) ενίσχυσε τη συνεργασία διευκρινίζοντας την εστίαση ή την πρόθεση της επικοινωνίας. Επιπλέον, ο φυσιολογικός συγχρονισμός μέσα στα ζεύγη μαθητών συνδέθηκε με βελτιωμένη ποιότητα συνεργασίας. Οι διαφορές στην κίνηση και τη θέση μέσα σε μια δυάδα βρέθηκε να επηρεάζουν αρνητικά την ποιότητα της συνεργασίας. Μια μελέτη περίπτωσης που συνέκρινε τις περισσότερες και λιγότερο συνεργατικές ομάδες απέδειξε ότι η απλή παρατήρηση θα μπορούσε να οδηγήσει σε υψηλά μαθησιακά κέρδη, δεδομένου του σωστού μαθησιακού πλαισίου. Οι ερευνητές κατέληξαν στο συμπέρασμα ότι ο συνδυασμός της παρακολούθησης των ματιών προσφέρει ένα πιο πλούσιο προγνωστικό μοντέλο για την ποιότητα της συνεργασίας από ό,τι όταν αυτοί οι τύποι δεδομένων εξετάζονται χωριστά. Συγκεκριμένα, η μειωμένη κοινή οπτική προσοχή και ο φυσιολογικός αποσυγχρονισμός μεταξύ των συνεργαζόμενων μαθητών οδήγησαν σε χαμηλότερη απόδοση εργασίας και μειωμένα μαθησιακά αποτελέσματα. Η ικανότητα αναγνώρισης καταστάσεων συνεργασίας χρησιμοποιώντας δεδομένα αισθητήρων θα μπορούσε να επιτρέψει την παρακολούθηση της συνεργασίας σε πραγματικό χρόνο, βοηθώντας στον εντοπισμό και την αντιμετώπιση μη παραγωγικών αλληλεπιδράσεων. Η μελλοντική έρευνα ενθαρρύνεται να διερευνήσει διάφορα μέτρα κίνησης για να αποτυπώσει με μεγαλύτερη ακρίβεια τις στάσεις και τις κινήσεις των συμμετεχόντων.

Τέλος, ο Zhao κ.ά. [28] εμβαθύναν στην αναγνώριση της κατάστασης των ματιών, με εφαρμογή της μεθόδου Transfer Learning, προχωρημένων τεχνικών βαθιάς μάθησης, πολύπλοκων νευρωνικών δικτύων με χρήση τριών (3) διαφορετικών συνόλων δεδομένων. Για την ταξινόμηση των καταστάσεων των ματιών από στατικές εικόνες προσώπου, πρότειναν ένα μοναδικό πλαίσιο βασισμένο στη βαθιά μάθηση. Αυτό το πλαίσιο συγχωνεύει ένα βαθύ νευρωνικό δίκτυο (deep neural network (DNN)) και ένα βαθύ συνελικτικό νευρωνικό δίκτυο (Deep Convolutional Neural Network (DCNN)) για να δημιουργήσει ένα βαθιά ολοκληρωμένο νευρωνικό δίκτυο (Deep Integrated Neural Network (DINN)). Αυτό το DINN βελτιστοποιήθηκε για να εξάγει πολύτιμες πληροφορίες από την περιοχή των ματιών. Για να ενισχυθούν οι ικανότητες ταξινόμησης του μοντέλου, ειδικά με μικρότερα σύνολα δεδομένων δειγμάτων, χρησιμοποιήθηκε μια στρατηγική μεταφοράς μάθησης. Μέσω αυστηρών πειραμάτων που χρησιμοποιούν σύνολα δεδομένων, όπως τα σύνολα δεδομένων Closed Eyes in the Wild (CEW) και τα σύνολα δεδομένων Eyeblink του Πανεπιστημίου Zhejiang, η προτεινόμενη μέθοδος βρέθηκε να είναι ανώτερη από τις υπάρχουσες τεχνικές αιχμής. Επιπλέον, οι συγγραφείς κατασκεύασαν ένα σύνολο δεδομένων αναγνώρισης υπνηλίας οδηγού για να αξιολογήσουν τη δυνατότητα εφαρμογής της μεθόδου σε συνθήκες οδήγησης, όπου το προτεινόμενο μοντέλο έδειξε ισχυρή και σταθερή απόδοση. Συμπερασματικά, αυτή η έρευνα παρουσίασε μια πρωτοποριακή προσέγγιση βαθιάς μάθησης προσαρμοσμένη για την ταξινόμηση της κατάστασης των ματιών κάτω από πραγματικές, μη ελεγχόμενες συνθήκες. Με τη συγχώνευση DNN και DCNN στο μοντέλο DINN χρησιμοποιώντας κοινή βελτιστοποίηση, οι θα μπορούσαν να συλλάβουν περίπλοκες λεπτομέρειες από τις περιοχές των ματιών. Παρατήρησαν επίσης ότι η αύξηση του όγκου των δεδομένων προεκπαίδευσης ενίσχυσε την απόδοση του DNN αλλά δεν επηρέασε σημαντικά τις δυνατότητες αναγνώρισης του DCNN. Παρά τα πλεονεκτήματα, η μελέτη αναγνώρισε την υπολογιστική επιβάρυνση που εισάγεται από τις τεχνικές βαθιάς μάθησης. Για να το αντιμετωπίσουν αυτό, χρησιμοποίησαν μια GPU για την επιτάχυνση της λειτουργίας του DINN σε πραγματικό χρόνο. Ωστόσο, αναγνωρίζοντας ότι το μοντέλο τους είναι πιο περίπλοκο από ορισμένα συμβατικά

συστήματα, οι μελλοντικές εργασίες θα επικεντρωθούν στην απλοποίηση και την επιτάχυνση του μοντέλου διατηρώντας παράλληλα την ακρίβεια. Τα σχέδια περιλαμβάνουν επίσης τη βελτίωση της εκπαιδευτικής προσέγγισης και τελικά τον σχεδιασμό ενός συστήματος ανίχνευσης οφθαλμικής κατάστασης σε πραγματικό χρόνο κατάλληλο, τόσο για έρευνα, όσο και για πρακτικές εφαρμογές.

3 Επιβλεπόμενη Μηχανική Μάθηση

Στην παρούσα διπλωματική χρησιμοποιείται Επιβλεπόμενη Μηχανική Μάθηση (ΕΜΜ) για την ταξινόμηση των περιστατικών (instances) σε ανοιχτούς ή κλειστούς οφθαλμούς. Η ΕΜΜ υλοποιείται μέσα από τη χρήση αλγορίθμων εφαρμογής, οι οποίοι στα πλαίσια διερεύνησης του ερευνητικού σκοπού της διπλωματικής, εφαρμόζονται στο ίδιο σύνολο δεδομένων, με διαφορετικούς τρόπους, ώστε να επιτευχθεί η συγκριτική διερεύνηση των αποτελεσμάτων τους.

Πριν όμως παρουσιαστούν στην τελευταία υποενότητα οι αλγόριθμοι που εφαρμόζονται, κρίνεται απαραίτητη μία γενική αναφορά στην Μηχανική Μάθηση, στις υποκατηγορίες της με περισσότερη εμβάθυνση στην υποκατηγορία της ΕΜΜ, η οποία και υλοποιείται. Επιπρόσθετα, θα παρουσιαστούν γενικά οι τομείς εφαρμογής της ΕΜΜ σε δεδομένα EEG και οι δυνατότητες που υπάρχουν για πιθανές εφαρμογές.

Αρχικά, η μηχανική μάθηση είναι ένα υποσύνολο της τεχνητής νοημοσύνης που περιλαμβάνει την ανάπτυξη αλγορίθμων που επιτρέπουν στους υπολογιστές να μαθαίνουν και να κάνουν προβλέψεις ή να παίρνουν αποφάσεις με βάση δεδομένα. Αντί να προγραμματίζεται άμεσα για να εκτελέσει μια εργασία, μια μηχανή μαθαίνει από μεγάλες ποσότητες δεδομένων χρησιμοποιώντας στατιστικές τεχνικές.

3.1 Χαρακτηριστικά

Η μηχανική μάθηση μπορεί να κατηγοριοποιηθεί σε γενικές γραμμές σε τρεις τύπους:

- Μάθηση με επίβλεψη (Supervised Learning)
- Μάθηση χωρίς επίβλεψη (Unsupervised Learning)
- Ενισχυτική μάθηση (Reinforcement Learning)

Η επιβλεπόμενη μάθηση είναι η πιο κοινή τεχνική μεταξύ των μεθοδολογιών μηχανικής μάθησης. Σε αυτή την προσέγγιση, ένας αλγόριθμος μαθαίνει από επισημασμένα δεδομένα εκπαίδευσης και κάνει προβλέψεις με βάση αυτά τα δεδομένα. Αυτή η διαδικασία μάθησης είναι "εποπτευόμενη", επειδή ο αλγόριθμος κάνει προβλέψεις και διορθώνεται από τις σωστές απαντήσεις που παρέχονται από τον άνθρωπο, όποτε κάνει λάθος. Ουσιαστικά, είναι σαν ένας μαθητής που μαθαίνει υπό την επίβλεψη ενός δασκάλου ή όπως μαθαίνει ένα παιδί από έναν γονέα.

Βασικές έννοιες στην επιβλεπόμενη μάθηση:

- Δεδομένα εκπαίδευσης: Αυτά είναι τα δεδομένα στα οποία εκπαιδεύεται ο αλγόριθμος. Αποτελούνται από ζεύγη εισόδου-εξόδου, όπου η είσοδος είναι ένα διάνυσμα χαρακτηριστικών και η έξοδος είναι η αντίστοιχη ετικέτα (σωστή απάντηση).
- Μοντέλο: Αφού εκπαιδευτεί στα δεδομένα εκπαίδευσης, ο αλγόριθμος επιβλεπόμενη μάθησης δημιουργεί ένα μοντέλο. Αυτό το μοντέλο αντιπροσωπεύει ό,τι έχει μάθει ο αλγόριθμος από τα δεδομένα εκπαίδευσης.
- Πρόβλεψη: Αφού εκπαιδευτεί το μοντέλο, μπορεί να χρησιμοποιηθεί για την πραγματοποίηση προβλέψεων σε νέα, άγνωστα δεδομένα.

- Συνάρτηση απώλειας: Πρόκειται για ένα μέτρο του πόσο απέχουν οι προβλέψεις του μοντέλου από τις πραγματικές τιμές. Ο στόχος κατά την εκπαίδευση είναι η ελαχιστοποίηση αυτής της απώλειας.
- Υπερπροσαρμογή: Αυτό συμβαίνει όταν το μοντέλο μαθαίνει πολύ καλά, υπερπροσαρμόζεται στα τα δεδομένα εκπαίδευσης, συμπεριλαμβανομένου του θορύβου και των ακραίων τιμών τους, γεγονός που το κάνει να έχει κακή απόδοση σε άγνωστα δεδομένα.
- Κανονικοποίηση: Τεχνικές για την αποτροπή της υπερπροσαρμογής με την προσθήκη κάποιας μορφής ποινής στη συνάρτηση απώλειας.

3.2 Αλγόριθμοι Εφαρμογής

Οι αλγόριθμοι που χρησιμοποιήθηκαν στην πειραματική αξιολόγηση και αποτελούν χαρακτηριστικούς αλγορίθμους της EMM είναι οι Naive Bayes, Logistic Regression, Decision Tree, KNN, Random Forest, MLP (νευρωνικό δίκτυο), SVM, AdaBoost και Bagging. Παρακάτω παρουσιάζονται συνοπτικά τα χαρακτηριστικά για κάθε αλγόριθμο.

Naive Bayes

Ο Naive Bayes είναι μια στατιστική μέθοδος ταξινόμησης που βασίζεται στο θεώρημα του Bayes. Είναι ένας από τους απλούστερους αλγορίθμους επιβλεπόμενης μάθησης και είναι πιθανοτικός ταξινομητής. Οι ταξινομητές Naive Bayes έχουν υψηλή ακρίβεια και ταχύτητα σε μεγάλα σύνολα δεδομένων.

Ο ταξινομητής Naive Bayes υποθέτει ότι η επίδραση ενός συγκεκριμένου χαρακτηριστικού σε μια κλάση είναι ανεξάρτητη από άλλα χαρακτηριστικά. Για παράδειγμα, ένα φρούτο μπορεί να θεωρηθεί μήλο εάν είναι κόκκινο, στρογγυλό και έχει διάμετρο περίπου 3 ίντσες. Ακόμη και αν αυτά τα χαρακτηριστικά εξαρτώνται το ένα από το άλλο ή από την ύπαρξη των άλλων χαρακτηριστικών, ένας ταξινομητής Naive Bayes θεωρεί ότι όλες αυτές οι ιδιότητες συμβάλλουν ανεξάρτητα στην πιθανότητα ότι το φρούτο αυτό είναι μήλο.

Το μοντέλο Naive Bayes είναι εύκολο στην κατασκευή και ιδιαίτερα χρήσιμο για πολύ μεγάλα σύνολα δεδομένων. Μαζί με την απλότητα, το μοντέλο Naive Bayes είναι γνωστό ότι υπερτερεί ακόμη και σε σχέση με πολύ εξελιγμένες μεθόδους ταξινόμησης.

Ο Naive Bayes είναι επίσης γνωστός ως generative model επειδή προβλέπει την κοινή πιθανότητα, p(x, y), των χαρακτηριστικών εισόδου, x, και της εξόδου, y, και στη συνέχεια το χρησιμοποιεί για να υπολογίσει το p(y|x), την πιθανότητα του y δεδομένου του x.

Λογιστική Παλινδρόμηση (Logistic Regression)

Η λογιστική παλινδρόμηση είναι ένας τύπος ανάλυσης παλινδρόμησης που χρησιμοποιείται για την πρόβλεψη της πιθανότητας ενός δυαδικού αποτελέσματος. Είναι ένας τρόπος μοντελοποίησης της σχέσης μεταξύ μιας δυαδικής εξαρτημένης μεταβλητής και μιας ή περισσότερων ανεξάρτητων μεταβλητών.

Η έξοδος της λογιστικής παλινδρόμησης είναι μια πιθανότητα το δεδομένο σημείο εισόδου να ανήκει σε μια συγκεκριμένη κλάση, η οποία μετατρέπεται σε δυαδικό αποτέλεσμα μέσω ενός κατωφλίου (π.χ., εάν η πιθανότητα εξόδου είναι μεγαλύτερη από 0,5, ταξινομείται ως κλάση 1, διαφορετικά κλάση 0). Η κεντρική παραδοχή της λογιστικής παλινδρόμησης είναι η υπόθεση ότι ο χώρος εισόδου μπορεί να διαχωριστεί σε δύο "περιοχές", μία για κάθε κλάση, μέσω ενός γραμμικού ορίου.

Δέντρα Απόφασης (Decision Trees)

Ένα δέντρο αποφάσεων είναι μια δομή που μοιάζει με διάγραμμα ροής, στην οποία κάθε εσωτερικός κόμβος αντιπροσωπεύει ένα χαρακτηριστικό (ή ιδιότητα), κάθε κλάδος αντιπροσωπεύει έναν κανόνα απόφασης και κάθε κόμβος φύλλου αντιπροσωπεύει το αποτέλεσμα. Ο κόμβος ρίζας είναι ο κορυφαίος κόμβος απόφασης σε ένα δέντρο που αντιστοιχεί στον καλύτερο προγνωστικό που ονομάζεται κόμβος ρίζας. Τα δέντρα αποφάσεων μπορούν να χειριστούν τόσο κατηγορικά όσο και αριθμητικά δεδομένα.

Οι κανόνες απόφασης έχουν γενικά τη μορφή δηλώσεων if-then-else. Όσο βαθύτερο είναι το δέντρο, τόσο πιο πολύπλοκοι είναι οι κανόνες και τόσο πιο κατάλληλο είναι το μοντέλο.

KNN

Οι K-κοντινότεροι γείτονες (KNN) είναι ένας τύπος μάθησης βασισμένης σε παραδείγματα ή τεμπέλικης (lazy) μάθησης, όπου η συνάρτηση προσεγγίζεται μόνο τοπικά και όλος ο υπολογισμός μετατίθεται μέχρι την αξιολόγηση της συνάρτησης. Είναι μια μη παραμετρική μέθοδος που χρησιμοποιείται για ταξινόμηση και παλινδρόμηση.

Αντίστοιχα και στις δύο περιπτώσεις, η είσοδος αποτελείται από τα k πλησιέστερα παραδείγματα εκπαίδευσης στο χώρο των χαρακτηριστικών. Η έξοδος εξαρτάται από το αν ο KNN χρησιμοποιείται για ταξινόμηση ή παλινδρόμηση:

Στην ταξινόμηση με KNN, η έξοδος είναι η ένταξη σε κλάση. Ένα αντικείμενο ταξινομείται από μια πληθώρα ψήφων των γειτόνων του, με το αντικείμενο να κατατάσσεται στην κλάση που είναι πιο κοινή μεταξύ των k πλησιέστερων γειτόνων του (το k είναι ένας θετικός ακέραιος αριθμός, συνήθως μικρός). Εάν k = 1, τότε το αντικείμενο απλώς κατατάσσεται στην κλάση του συγκεκριμένου κοντινότερου γείτονα.

Από την άλλη στην παλινδρόμηση με KNN, η έξοδος είναι η τιμή της ιδιότητας για το αντικείμενο. Η τιμή αυτή είναι ο μέσος όρος των τιμών των k πλησιέστερων γειτόνων.

Random Forest

Ο Random Forest είναι μια μέθοδος μάθησης συνόλου για ταξινόμηση, παλινδρόμηση και άλλες εργασίες, η οποία περιλαμβάνει την κατασκευή ενός πλήθους δέντρων απόφασης κατά τη στιγμή της εκπαίδευσης και την εξαγωγή της που είναι ο συνδυασμός των κλάσεων (ταξινόμηση) ή η μέση πρόβλεψη (παλινδρόμηση) των μεμονωμένων δέντρων. Είναι μια επέκταση της μεθόδου bagging που εκτός από την κατασκευή δέντρων σε τυχαία υποσύνολα των δεδομένων, χρησιμοποιεί επίσης τυχαία υποσύνολα των χαρακτηριστικών για τον διαχωρισμό των κόμβων.

Multi-layer Perceptron (MLP)

Ο Multi-layer Perceptron (MLP) είναι ένας τύπος νευρωνικού δικτύου, ο οποίος από μόνος του είναι ένα σύνολο αλγορίθμων που αποσκοπούν στην ανίχνευση υποκείμενων σχέσεων σε ένα σύνολο δεδομένων μέσω μιας διαδικασίας που μιμείται τον τρόπο λειτουργίας του ανθρώπινου εγκεφάλου.

Ο MLP είναι ένα βαθύ, τεχνητό νευρωνικό δίκτυο. Αποτελείται από περισσότερα του ενός επίπεδα (εξού και το "multi-layer"). Συγκεκριμένα, αποτελείται από ένα επίπεδο εισόδου για τη λήψη του σήματος, ένα επίπεδο εξόδου που λαμβάνει μια απόφαση ή πρόβλεψη σχετικά με την είσοδο, και μεταξύ αυτών των δύο, έναν αυθαίρετο αριθμό κρυφών επιπέδων που αποτελούν την πραγματική υπολογιστική μηχανή του MLP. Τα MLP με ένα κρυφό επίπεδο είναι ικανά να προσεγγίσουν οποιαδήποτε συνεχή συνάρτηση.

SVM

Ο Support Vector Machines (SVM) είναι ένας αλγόριθμος επιβλεπόμενης μάθησης που χρησιμοποιείται για προβλήματα ταξινόμησης ή παλινδρόμησης. Είναι ιδιαίτερα χρήσιμος για την ταξινόμηση σύνθετων μεν, αλλά μικρού ή μεσαίου μεγέθους συνόλων δεδομένων.

Η βασική ιδέα του SVM στην εύρεση ενός υπερεπιπέδου που χωρίζει καλύτερα ένα σύνολο δεδομένων σε δύο κλάσεις. Τα διανύσματα υποστήριξης είναι τα σημεία δεδομένων που βρίσκονται πλησιέστερα στην επιφάνεια απόφασης (ή στο υπερεπίπεδο). Είναι τα σημεία δεδομένων που είναι πιο δύσκολο να ταξινομηθούν και έχουν άμεση σχέση με τη βέλτιστη θέση της επιφάνειας απόφασης.

AdaBoost

Ο AdaBoost, ή Adaptive Boosting, είναι μια τεχνική ενίσχυσης που χρησιμοποιείται για την κατασκευή ενός ισχυρού ταξινομητή από έναν αριθμό αδύναμων ταξινομητών. Ένας αδύναμος ταξινομητής είναι απλά ένας ταξινομητής που έχει κακή απόδοση, αλλά έχει καλύτερη απόδοση από την τυχαία επιλογή (μαντεψιά). Ένας συνηθισμένος αδύναμος ταξινομητής που χρησιμοποιείται με το AdaBoost είναι ένα δέντρο απόφασης με μία μόνο διαίρεσή του (ένα κλαδί).

Bagging

Ο όρος Bagging, που αναλύεται σε Bootstrap AGGregatING, είναι μια μέθοδος μάθησης μέσω συνόλων (ensemble learning) που χρησιμοποιείται για τη μείωση της διακύμανσης ενός βασικού εκτιμητή (π.χ. ένα δέντρο απόφασης). Αυτό το επιτυγχάνει εκπαιδεύοντάς τον τελευταίο σε τυχαία υποσύνολα των δεδομένων εκπαίδευσης και στη συνέχεια, συγκεντρώνοντας τις επιμέρους προβλέψεις για να σχηματιστεί μια τελική πρόβλεψη.

3.3 Εφαρμογές σε ιατρικά δεδομένα

Τα ιατρικά δεδομένα είναι περίπλοκα και περιέχουν πληθώρα πληροφοριών. Η ανάλυση αυτών των δεδομένων έχει τη δυνατότητα να φέρει επανάσταση στην υγειονομική περίθαλψη, από τη βελτίωση της διάγνωσης έως την εξατομίκευση της θεραπείας. Παρακάτω, παρουσιάζονται κάποιες ενδεικτικές κατηγορίες που μπορεί να εφαρμοστεί και να φέρει αποτελέσματα στον τομέα της υγείας.

- **Αναγνώριση και διάγνωση ασθενειών**: Οι αλγόριθμοι μπορούν να εκπαιδευτούν σε δεδομένα όπως ακτίνες Χ, μαγνητικές τομογραφίες, ακόμη και γονιδιωματικές ακολουθίες για τον εντοπισμό και τη διάγνωση ασθενειών, μερικές φορές με μεγαλύτερη ακρίβεια από τους ανθρώπους.
- **Εξατομικευμένη θεραπεία και ανάπτυξη φαρμάκων**: Με βάση ιστορικά δεδομένα ασθενών με παρόμοια συμπτώματα ή παθήσεις, οι αλγόριθμοι με επίβλεψη μπορούν να καθορίσουν ποιες θεραπείες είναι πιθανό να είναι οι πιο αποτελεσματικές για μεμονωμένους ασθενείς. Αυτό το είδος εξατομικευμένης ιατρικής λαμβάνει υπόψη γενετικούς, περιβαλλοντικούς παράγοντες και παράγοντες του τρόπου ζωής. Επιπλέον, αναλύοντας δεδομένα από δοκιμές φαρμάκων, οι αλγόριθμοι μπορούν να βοηθήσουν στην πρόβλεψη του πόσο αποτελεσματικό θα είναι ένα νέο φάρμακο για έναν ευρύτερο πληθυσμό.
- **Πρόβλεψη επιδημικής έξαρσης**: Με την εκπαίδευση σε ιστορικά δεδομένα που σχετίζονται με επιδημίες ασθενειών, τα μοντέλα με επίβλεψη μπορούν να προβλέψουν

πιθανές μελλοντικές επιδημίες. Αυτό επιτρέπει στους οργανισμούς υγείας να είναι προληπτικοί και όχι ανασταλτικοί, σώζοντας δυνητικά πολλές ζωές.

- **Πρόβλεψη πρώιμων ασθενειών**: Η μηχανική μάθηση, ειδικά όταν συνδυάζεται με φορητή τεχνολογία (wearables), μπορεί να βοηθήσει στην έγκαιρη ανίχνευση ασθενειών. Τα μοντέλα αυτά μπορούν να προβλέψουν διάφορα αποτελέσματα, όπως η ευαισθησία σε ασθένειες, η επανεισαγωγή ασθενών ή η εξέλιξη της νόσου. Για παράδειγμα, με την ανάλυση δεδομένων από φορητές συσκευές, μπορεί να είναι δυνατή η ανίχνευση πρώιμων ενδείξεων παθολογικών καταστάσεων, όπως οι καρδιακές παθήσεις.
- **Ανάλυση ιατρικών εικόνων**: Πέρα από την απλή ανίχνευση ανωμαλιών σε ιατρικές εικόνες, οι προηγμένοι αλγόριθμοι μπορούν να παρέχουν λεπτομερή ανάλυση, να τμηματοποιούν εικόνες και ακόμη και να προβλέπουν την πιθανότητα εξάπλωσης ή επανεμφάνισης μιας ασθένειας. Αυτό επεκτείνεται σε αξονικές τομογραφίες, μαστογραφίες και άλλες αντίστοιχες εξετάσεις.
- **Ανάλυση γονιδιωματικής αλληλουχίας**: Τα τελευταία χρόνια ο τομέας που ασχολείται με τις αναλύσεις γονιδίων έχει περισσότερα δεδομένα στη διάθεσή τους σε σχέση με τα προηγούμενα χρόνια, που η εξαγωγή τέτοιων δεδομένων ήταν πολύ ακριβή. Η επιβλεπόμενη μάθηση μπορεί να βοηθήσει στην κατανόηση δεδομένων που αφορούν τη γονιδιακή αλληλουχία, στον προσδιορισμό της λειτουργίας των γονιδίων και στην πρόβλεψη της ευαισθησίας σε ασθένειες με βάση γενετικούς παράγοντες.

3.3.1 Εφαρμογές με δεδομένα ΗΕΓ

Το ηλεκτροεγκεφαλογράφημα (ΗΕΓ) καταγράφει την ηλεκτρική δραστηριότητα του εγκεφάλου, εξάγοντας δεδομένα που είναι πολύ σημαντικά στην παροχή πληροφοριών. Τα δεδομένα αυτά είναι πλούσια και πολύπλοκα, με πιθανές εφαρμογές που κυμαίνονται από τη διάγνωση νευρολογικών διαταραχών, έως τις διεπαφές εγκεφάλου-υπολογιστή. Παρακάτω παρουσιάζονται οι εφαρμογές που μπορούν να γίνουν με εφαρμογή και ανάλυση δεδομένων ΗΕΓ.

- **Ανίχνευση επιληπτικών κρίσεων**: Οι επιβλεπόμενοι αλγόριθμοι μπορούν να εκπαιδευτούν σε δεδομένα EEG για να ανιχνεύσουν την έναρξη επιληπτικών κρίσεων σε πραγματικό χρόνο, βοηθώντας στην έγκαιρη ιατρική παρέμβαση.
- **Αξιολόγηση γνωστικού φόρτου**: Με την εκπαίδευση σε επισημασμένα δεδομένα EEG, οι αλγόριθμοι μπορούν να προσδιορίσουν πότε ένα άτομο βρίσκεται υπό σημαντικό γνωστικό φορτίο. Αυτή η αξιολόγηση μπορεί να είναι χρήσιμη σε διάφορα σενάρια, όπως η αξιολόγηση της αποτελεσματικότητας των εκπαιδευτικών σχεδίων ή η αξιολόγηση της κόπωσης των πιλότων ή των οδηγών.
- **Νευροανάδραση**: Αυτό περιλαμβάνει την παροχή ανατροφοδότησης σε πραγματικό χρόνο σχετικά με τη δραστηριότητα των εγκεφαλικών κυμάτων. Η επιβλεπόμενη μάθηση μπορεί να χρησιμοποιηθεί για την καθοδήγηση των χρηστών ώστε να προκαλέσουν συγκεκριμένα μοτίβα ΗΕΓ, τα οποία μπορεί να είναι θεραπευτικά για καταστάσεις όπως η ΔΕΠΥ ή το άγχος.
- **Διαφορική διάγνωση**: Ορισμένες νευρολογικές παθήσεις μπορεί να έχουν παρόμοια εξωτερικά συμπτώματα αλλά διαφορετικά μοτίβα ΗΕΓ. Οι αλγόριθμοι μπορούν να εκπαιδευτούν ώστε να διακρίνουν μεταξύ αυτών των καταστάσεων, βοηθώντας στην ακριβή διάγνωση.

- **Πρόβλεψη της ηλικίας του εγκεφάλου**: Χρησιμοποιώντας δεδομένα EEG, μοντέλα μάθησης με επίβλεψη μπορούν να εκπαιδευτούν για να εκτιμήσουν τη "λειτουργική ηλικία" ενός εγκεφάλου, η οποία μπορεί να διαφέρει από τη χρονολογική ηλικία. Αυτό μπορεί να είναι χρήσιμο για την αξιολόγηση καταστάσεων, όπως η νόσος Αλτσχάιμερ ή άλλες μορφές άνοιας.
- **Παρακολούθηση των επιδράσεων των φαρμάκων**: Η ανταπόκριση του εγκεφάλου σε ορισμένα φάρμακα, ιδίως σε εκείνα για νευρολογικές παθήσεις, μπορεί να παρακολουθείται με τη χρήση του ΗΕΓ. Μπορούν να εκπαιδευτούν αλγόριθμοι με επίβλεψη για να ανιχνεύουν αλλαγές στα μοτίβα του ΗΕΓ που είναι ενδεικτικές της αποτελεσματικότητας του φαρμάκου ή των πιθανών παρενεργειών.
- **Ανίχνευση επιπέδων συνείδησης**: Στην εντατική περίθαλψη ή σε καταστάσεις που αφορούν ασθενείς σε κώμα, τα δεδομένα EEG σε συνδυασμό με τη μηχανική μάθηση μπορούν να βοηθήσουν στον προσδιορισμό των επιπέδων συνείδησης ή στην πρόβλεψη των πιθανοτήτων ανάκαμψης.
- **Νευρωνική αποκωδικοποίηση**: Πρόκειται για την αποκωδικοποίηση των συγκεκριμένων σκέψεων ή των προβλεπόμενων ενεργειών ενός ατόμου με βάση τα δεδομένα του ΗΕΓ. Αφορά ένα πολύπλοκο έργο, αλλά έχει πιθανές εφαρμογές σε διεπαφές εγκεφάλου-υπολογιστή και υποστηρικτικές τεχνολογίες για παράλυτα άτομα.
- **Διεπαφές εγκεφάλου-υπολογιστή**: Μοντέλα μηχανικής μάθησης μπορούν να ερμηνεύσουν δεδομένα EEG για να επιτρέψουν στους χρήστες να ελέγχουν συσκευές χρησιμοποιώντας τις σκέψεις τους.
- **Ανάλυση ύπνου**: Χρησιμοποιώντας το EEG, οι αλγόριθμοι μπορούν να τμηματοποιήσουν τον ύπνο σε διάφορα στάδια και να διαγνώσουν διαταραχές του ύπνου.
- **Παρακολούθηση της ψυχικής κατάστασης**: Τα δεδομένα EEG μπορούν να χρησιμοποιηθούν για τη μέτρηση της προσοχής, της χαλάρωσης ή ακόμη και για την ανίχνευση ενδείξεων κατάθλιψης.

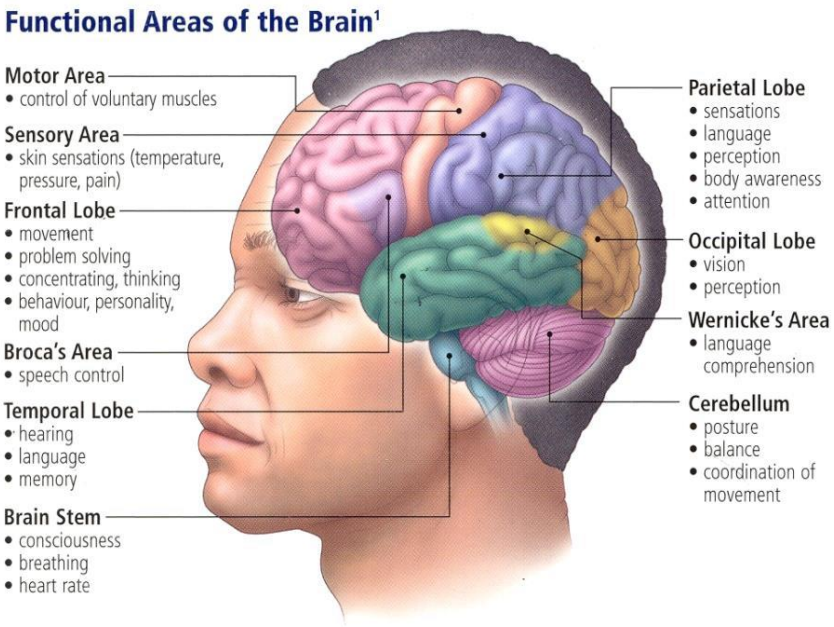
Συμπερασματικά, η μηχανική μάθηση και ειδικότερα η επιβλεπόμενη ΜΜ διαδραματίζει καθοριστικό ρόλο στην εξαγωγή ουσιαστικών πληροφοριών από πολύπλοκα σύνολα δεδομένων. Όπως παρουσιάστηκε και στο κεφάλαιο 2 της βιβλιογραφικής ανασκόπησης, οι έρευνες σε αυτόν τον τομέα βρίσκονται σε συνεχή εξέλιξη παράγοντας αξιοσημείωτα αποτελέσματα. Καθώς η συλλογή δεδομένων βελτιώνεται και οι αλγόριθμοι γίνονται πιο εξελιγμένοι, οι εφαρμογές θα συνεχίσουν να αυξάνονται και να ενσωματώνονται περισσότερο στην καθιερωμένη ιατρική πρακτική. Στο πεδίο των ιατρικών δεδομένων, και ειδικότερα του EEG, οι γνώσεις αυτές έχουν τη δυνατότητα να επιφέρουν ουσιαστικές εξελίξεις στη φροντίδα των ασθενών, τη διάγνωση και τις θεραπευτικές μεθόδους.

4 Σύνολο Δεδομένων Ηλεκτροεγκεφαλογραφήματος (ΗΕΓ)

Στην παρούσα ενότητα παρουσιάζεται η περιγραφή του προβλήματος μελέτης και του συνόλου δεδομένων που χρησιμοποιείται. Περιγράφονται τα επιμέρους χαρακτηριστικά των δεδομένων ΗΕΓ και επιχειρείται η σύνδεση εφαρμογών της Μηχανικής Μάθησης με δεδομένα ΗΕΓ, είτε μέσα από τη σκοπιά των αλγορίθμων, είτε μέσα από τα πεδία εφαρμογής.

4.1 Περιγραφή Προβλήματος Μελέτης

Το πρόβλημα που καλούμαστε να επιλύσουμε ασχολείται με την καταγραφή εγκεφαλικών σημάτων και την κατηγοριοποίηση τους ανάλογα με το αν το μετρούμενο άτομο είχε ανοιχτά (τιμή κλάσης '0') ή κλειστά (τιμή κλάσης '1'). Αυτές οι μετρήσεις γίνονται σε συγκεκριμένα μέρη του εγκεφάλου, αναλόγως τη στόχευση της μέτρησης, όπως φαίνεται στο **Σχήμα 1**. Η καταγραφή των ηλεκτρικών σημάτων του εγκεφάλου γίνεται με ηλεκτρόδια που τοποθετούνται στην επιφάνεια του κρανίου. Ηλεκτρικά σήματα συλλέγονται από τον φλοιό του εγκεφάλου (αφού αυτός είναι πιο κοντά στο κρανίο σε σχέση με άλλες δομές του φλοιού που είναι στο εσωτερικό του εγκεφάλου).



Σχήμα 1: Λειτουργικές Περιοχές του Εγκεφάλου (Πηγή: https://t.ly/RvOuU)

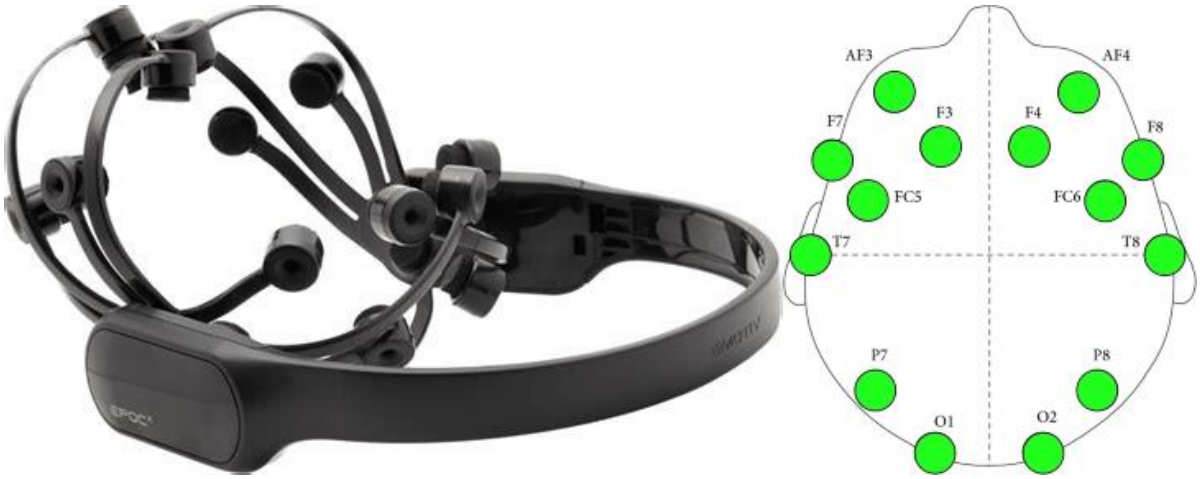
Συνήθως αυτά τα δεδομένα περιλαμβάνουν θόρυβο ο οποίος μπορεί να προέρχεται από παρεμβολές λόγω του ηλεκτρικού δικτύου (50Hz ή 60 Hz), την κίνηση οφθαλμών / βλεφάρων, τη μυϊκή δραστηριότητα, τη καρδιακή λειτουργία, κ.α..

Παράλληλα, το πρόβλημα αφορά και μπορεί να ενταχθεί στην κατηγορία των Μεγάλων Δεδομένων, στην οποία θα έχει καταγραφεί πολύ μεγάλος αριθμός εγκεφαλικών σημάτων, καθώς οι ολοκληρωμένες μετρήσεις εγκεφαλικών σημάτων ασθενών διαρκούν αρκετά, τόσο ώστε να παράγουν χιλιάδες αν όχι εκατομμύρια καταγραφές της ηλεκτρικής δραστηριότητας του ανθρωπίνου εγκεφάλου. Συνεπαγωγικά, το σύνολο δεδομένων θα έχει τέτοια διαστασιμότητα που καλούμαστε να διαχειριστούμε, να επεξεργαστούμε και να εξάγουμε νέα γνώση. Επομένως, θα πρέπει να βρεθεί

ένας τρόπος που θα οδηγήσει σε μεγαλύτερη αποδοτικότητα. Ταυτόχρονα, κοιτώντας την κατανομή του συνόλου δεδομένων, παρατηρούμε πολλές ακραίες τιμές, οι οποίες εικάζουμε και περιμένουμε να επηρεάζουν αρνητικά την απόδοση των αλγορίθμων μας στην απόδοση τους κατά την διαδικασία της κατηγοριοποίησης. Επιπρόσθετα, αφού αυτά τα δεδομένα μπορούν να αφορούν και μεγάλα δεδομένα, υπάρχει η επιτακτική ανάγκη να μειωθούν και οι διαστάσεις και να αυξηθεί η αποδοτικότητα. Συμπερασματικά, εξετάζουμε την προεπεξεργασία του συνόλου δεδομένων μας για να πραγματοποιηθούν οι παραπάνω στόχοι.

4.2 Παρουσίαση Συνόλου Δεδομένων και των χαρακτηριστικών του

Το σύνολο δεδομένων που εξετάζεται αφορά δεδομένα εγκεφαλογραφήματος, συνολικού μεγέθους 14980 στοιχείων και δεκατεσσάρων χαρακτηριστικών τα οποία είναι ανωνυμοποιημένα. Τα χαρακτηριστικά αντιστοιχούν σε 14 μετρήσεις ΗΕΓ (EEG) από το τη συσκευή καταγραφής που φαίνεται στο **Σχήμα 2**, με την αρχική σήμανση AF3, F7, F3, FC5, T7, P7, O1, O2, P8, T8, FC6, F4, F8, AF4, με αυτή τη σειρά.



Σχήμα 2: Emotiv EEG Neuroheadset (Πηγή: https://www.emotiv.com/epoc/)

Όπως αναφέρεται στη σελίδα του UCI, απ' όπου έγινε η άντληση, *«όλα τα δεδομένα προέρχονται από μία συνεχή μέτρηση Εγκεφαλογραφήματος (ΗΕΓ - EEG) με το Emotiv EEG Neuroheadset, διάρκειας 117 δευτερολέπτων. Η κατάσταση των ματιών ανιχνεύθηκε μέσω κάμερας κατά τη μέτρηση ΗΕΓ και προστέθηκε αργότερα χειροκίνητα στο αρχείο μετά την ανάλυση των καρέ βίντεο».* Η κλάση παίρνει τιμές '1' που υποδηλώνει την κατάσταση με τα μάτια κλειστά και το '0' που σημαίνει την κατάσταση με τα μάτια ανοιχτά. Όλες οι τιμές είναι με χρονολογική σειρά με την πρώτη μετρούμενη τιμή στην αρχή του συνόλου δεδομένων.

Επιπλέον, λόγω της φύσης αυτών των δεδομένων, αναμένονται να παρουσιάζουν κάποια χαρακτηριστικά που αφορούν την:

- ***Ύπαρξη Θορύβου***

Τα δεδομένα ΗΕΓ είναι συνήθως πολύ θορυβώδη, καθώς είναι επιρρεπή σε ευρήματα όπως ανοιγοκλείσιμο των ματιών, μυϊκές κινήσεις ή περιβαλλοντικός θόρυβος. Αυτό είναι σημαντικό να

σημειωθεί, καθώς οι πιο εξελιγμένοι ταξινομητές, όπως οι Random Forest ή Bagging, μπορούν ενδεχομένως να χειριστούν καλύτερα αυτόν τον θόρυβο λόγω της φύσης του συνόλου τους.

- ***Εξάρτηση στο Χρόνο***

Τα δεδομένα ΗΕΓ έχουν χρονικές εξαρτήσεις, που σημαίνει ότι η τιμή τη χρονική στιγμή *t* μπορεί να εξαρτάται από τις τιμές τις χρονικές στιγμές *t−1, t−2,* κ.λπ.

4.3 Σύνδεση Μηχανικής Μάθησης με Δεδομένα Υγείας ΗΕΓ

Παρακάτω παρουσιάζεται η χρησιμότητα των ταξινομητών σε σχέση με τα επιμέρους χαρακτηριστικά του EEG συνόλου δεδομένων. Αυτή η κατηγοριοποίηση βοηθάει ώστε να βρεθούν οι συνδέσεις των αποδόσεων των ταξινομητών σε σχέση με τα μετέπειτα αποτελέσματα που παρουσιάζονται στην ενότητα Αποτελέσματα.

- ***Random Forest σε σχέση με τα θορυβώδη δεδομένα***

  Δεδομένης της θορυβώδους φύσης των δεδομένων EEG, η ισχυρή απόδοση του ταξινομητή Random Forest είναι αξιοσημείωτη, όπως σημειώθηκε στη βιβλιογραφική ανασκόπηση. Η συνδυαστική του φύση, όπου πολλά δέντρα αποφάσεων συμμετέχουν για το τελικό αποτέλεσμα, μπορεί να του δώσει ένα πλεονέκτημα στη διάκριση των υποκείμενων μοτίβων από το θόρυβο.

- ***Χρονική εξάρτηση***

  Παρόλο που κανένα από τα μοντέλα των ταξινομητών που χρησιμοποιούνται δεν καλύπτει αποκλειστικά δεδομένα χρονοσειρών, η απόδοση ορισμένων μοντέλων μπορεί να υποδηλώνει την ικανότητά τους να συλλαμβάνουν τα υποκείμενα μοτίβα ανεξάρτητα από αυτό.

- ***Ερμηνευσιμότητα***

  Σε ιατρικά ή ερευνητικά σενάρια, η ερμηνευσιμότητα είναι συχνά ζωτικής σημασίας. Τα Δέντρα Αποφάσεων, αν και δεν είναι τόσο ακριβή όσο το Random Forest, είναι ιδιαίτερα ερμηνεύσιμα. Μπορεί εύκολα να οπτικοποιηθεί ένα δέντρο αποφάσεων για την ανίχνευση και κατανόηση ποιων χαρακτηριστικών (εύρη συχνοτήτων ΗΕΓ, θέσεις ηλεκτροδίων κ.λπ.) παίζουν σημαντικό ρόλο στον καθορισμό της κατάστασης των ματιών (ανοιχτά ή κλειστά).

- ***Προβλέψεις σε πραγματικό χρόνο***

  Αυτή η κατηγορία ίσως αποτελεί και την σημαντικότερη από άποψη ερμηνείας για εμπορικές εφαρμογές. Εάν ο στόχος είναι η πρόβλεψη της κατάστασης των ματιών σε πραγματικό χρόνο (για εφαρμογές όπως η ανίχνευση υπνηλίας), η ταχύτητα του μοντέλου είναι απαραίτητη. Εδώ, τα δέντρα απόφασης ή το KNN μπορεί να είναι καταλληλότερα λόγω της ισορροπίας μεταξύ ταχύτητας και απόδοσης.

Επιπροσθέτως, θα μπορούσαμε να εξετάσουμε τη σύνδεση αυτή ανά κατηγορία εφαρμογής και στόχευσης. Σχετικά παραδείγματα είναι οι ερευνητικοί σκοποί, η στόχευση αμεσότητας σε πραγματικό χρόνο, η ακρίβεια στις προβλέψεις ή ακόμα η εφαρμογή σε συνδυαστικά πολύπλοκα μοντέλα.

- ***Για ερευνητικούς σκοπούς***

Αν ο πρωταρχικός στόχος είναι να κατανοήσουμε ποια χαρακτηριστικά των δεδομένων ΗΕΓ επηρεάζουν περισσότερο την κατάσταση των ματιών, ένα ερμηνεύσιμο μοντέλο όπως τα δέντρα απόφασης μπορεί να είναι χρήσιμο.

- ***Για εφαρμογές πραγματικού χρόνου***

Η ταχύτητα καθίσταται προτεραιότητα, επομένως εξετάζεται ο χρόνο εκτέλεσης. Ο KNN, το Δέντρο Απόφασης ή ο Ensemble - AdaBoost μπορεί να επιτύχει μια ισορροπία εδώ.

- ***Για μέγιστη ακρίβεια***

Για αυτό το σκοπό, εξετάζεται ο Random Forest ή Bagging. Ωστόσο, συνίστανται βήματα προεπεξεργασίας για περαιτέρω καθαρισμό των δεδομένων ΗΕΓ και επίσης η εξέταση μοντέλων βαθιάς μάθησης που είναι προσαρμοσμένα για χρονοσειρές ή δεδομένα ακολουθιών.

- ***Πολύπλοκα μοντέλα***

Αξίζει να σημειωθεί ότι τα μοντέλα βαθιάς μάθησης, ειδικά οι πιο σύνθετες αρχιτεκτονικές όπως Αναδρομικά Νευρωνικά Δίκτυα (Recursive Neural Networks (RNN), Long-Short Term Memory networks(LSTM), ή ακόμα Συνελικτικά Νευρωνικά Δίκτυα (CNN (Convolutional Neural Networks)) που είναι προσαρμοσμένα για δεδομένα χρονοσειρών, θα μπορούσαν να παρέχουν ακόμη καλύτερα αποτελέσματα. Ωστόσο, πιθανότατα θα είναι ακόμα πιο υπολογιστικά «ακριβά» από άποψη χρόνου εκτέλεσης και των πόρων που θα χρειαστούν.

5 Μεθοδολογία

Στην παρούσα ενότητα παρουσιάζεται η μεθοδολογία που ακολουθήθηκε για την πραγματοποίηση της πειραματικής αξιολόγησης και την παραγωγή αποτελεσμάτων. Αναλύονται τα βήματα που εκτελέστηκαν στις επιμέρους ενότητες με την αντίστοιχη συμπερίληψη και ανάλυση του κώδικα που χρησιμοποιήθηκε. Ο κώδικας παρατίθεται αναλυτικά στο ΠΑΡΑΡΤΗΜΑ στο τέλος της διπλωματικής.

Συνοπτικά, αφού το σύνολο δεδομένων μετατραπεί στην κατάλληλη μορφή csv, αρχικά εφαρμόζεται πειραματική αξιολόγηση στο αρχικό σύνολο δεδομένων, που δεν έχει υποστεί καμία επεξεργασία. Αυτό είναι κρίσιμο ώστε να έχουμε μέτρο σύγκρισης των αποδόσεων για τα επόμενα βήματα. Στη συνέχεια εφαρμόζεται προεπεξεργασία του συνόλου δεδομένων και πραγματοποιείται βελτιστοποίηση παραμέτρων για να βρεθούν οι κατάλληλες για κάθε αλγόριθμο (ταξινομητή). Στο επόμενο βήμα εφαρμόζεται πειραματική αξιολόγηση στο επεξεργασμένο σύνολο δεδομένων και τα αποτελέσματα συγκρίνονται σε αντιπαραβολή με εκείνα της πειραματικής αξιολόγησης στο αρχικό σύνολο δεδομένων. Τέλος, σαν ξεχωριστό κομμάτι και επέκταση της διαδικασίας αυτής, εφαρμόζεται Ανάλυση Κύριων Συνιστωσών για να βρεθούν αρχικά οι συνιστώσες που αντιπροσωπεύουν την ανάλογη διακύμανση των δεδομένων. Στη συνέχεια πραγματοποιείται ΑΚΣ πριν την εφαρμογή του ταξινομητή που παρουσίασε τα καλύτερα αποτελέσματα στο προηγούμενο στάδιο. Τα αποτελέσματα στη συνέχεια αναλύονται για να γίνει μία προβολή σε Μεγάλα Δεδομένα και να εξεταστεί σε ποιο βαθμό μπορούν να μειωθούν τα δεδομένα, ενώ ταυτόχρονα επιτυγχάνουν βέλτιστα αποτελέσματα πρόβλεψης.

5.1 Εργαλεία και αρχικά βήματα

Η γλώσσα προγραμματισμού που χρησιμοποιήθηκε είναι η Python 3.11.4 και το περιβάλλον εργασίας που πραγματοποιήθηκαν οι επιμέρους πειραματισμοί είναι το Visual Studio Code. Το σύνολο δεδομένων όπως αναφέρθηκε και παραπάνω είναι σε μορφή arff από την πηγή που ανακτήθηκε. Για να χρησιμοποιηθούν στη Python στο περιβάλλον εργασίας που αναφέρθηκε και να μπορούν μετέπειτα να ανακτηθούν εύκολα πρέπει να μετατραπούν σε μορφή csv. Το πρώτο βήμα ήταν η δημιουργία ενός σεναρίου-αρχείου κώδικα (ΠΑΡΑΡΤΗΜΑ 9.1) που πραγματοποιεί αυτή τη μετατροπή και μετατρέπει το σύνολο δεδομένων ΗΕΓ στην κατάλληλη μορφή προς περαιτέρω επεξεργασία. Συνοπτικά, το σενάριο (τμήμα κώδικα) μετατρέπει ένα αρχείο ARFF (το eeg-eye-state.arff) σε μορφή CSV (eeg-eye-state.csv) παραλείποντας τα μεταδεδομένα του ARFF και αντιγράφοντας μόνο τις γραμμές δεδομένων στο CSV, των οποίων προηγείται μια καθορισμένη επικεφαλίδα. Παρακάτω περιγράφεται συνοπτικά η διαδικασία που εκτελείται:

- Το σενάριο αρχικά ανοίγει δύο αρχεία:
  - Το αρχείο 'eeg-eye-state.arff' για ανάγνωση.
  - Το αρχείο 'eeg-eye-state.csv' για εγγραφή. Αυτό θα είναι το αρχείο εξόδου σε μορφή CSV (Comma-Separated Values).
  - Η σημαία data_flag αρχικοποιείται σε False. Χρησιμοποιείται για να προσδιορίσει πότε αρχίζει το τμήμα δεδομένων του αρχείου ARFF.
  - Για κάθε γραμμή στο αρχείο ARFF εισόδου:
    - Εάν η σημαία data_flag είναι False (που σημαίνει ότι το σενάριο δεν έχει φτάσει ακόμη στο τμήμα δεδομένων).

- Ελέγχει αν η γραμμή περιέχει το '@DATA'. Αυτός είναι ο διαχωριστής στα αρχεία ARFF που υποδεικνύει την αρχή του πραγματικού τμήματος δεδομένων.
- Εάν βρεθεί το '@DATA', θέτει το data_flag σε True (υποδεικνύοντας ότι οι επόμενες γραμμές θα είναι δεδομένα) και γράφει τη γραμμή επικεφαλίδας ('AF3,F7,...,eyeDetection\n') στο αρχείο CSV εξόδου.
- Εάν η σημαία data_flag είναι True (που σημαίνει ότι το σενάριο διαβάζει τώρα το τμήμα δεδομένων του αρχείου ARFF), απλώς γράφει τη γραμμή στο αρχείο CSV εξόδου.

5.2 Προεπεξεργασία Δεδομένων

Η προεπεξεργασία δεδομένων στη Μηχανική Μάθηση (ΜΜ) αναφέρεται στη διαδικασία καθαρισμού και μετασχηματισμού των ακατέργαστων δεδομένων σε μορφή κατάλληλη για τη δημιουργία και την εκπαίδευση μοντέλων ΜΜ. Τα ακατέργαστα δεδομένα που συλλέγονται από διάφορες πηγές συχνά περιέχουν θόρυβο, ελλιπείς τιμές και άσχετες πληροφορίες. Η προεπεξεργασία βοηθά στο να γίνουν αυτά τα δεδομένα πιο δομημένα, «καθαρά» και αποτελεσματικά. Η προεπεξεργασία είναι ένα κρίσιμο βήμα επειδή η ποιότητα και η ποσότητα των δεδομένων που χρησιμοποιείται για την εκπαίδευση του μοντέλου επηρεάζει άμεσα και την απόδοση των υπό-διερεύνηση μοντέλων αξιολόγησης. Η διερευνητική ανάλυση δεδομένων είναι παρούσα σε όλα τα σενάρια που παρατίθενται στο ΠΑΡΑΡΤΗΜΑ (9.2, 9.3, 9.4), ενώ η προεπεξεργασία δεδομένων είναι παρούσα, σύμφωνα με τη μεθοδολογία που εφαρμόζεται, στα σενάρια 3 και 4 (ΠΑΡΑΡΤΗΜΑ 9.3 και 9.4). Παρακάτω παρουσιάζονται οι κατηγορίες προεπεξεργασίας που εφαρμόστηκαν στο σύνολο δεδομένων.

♦ **Διερευνητική ανάλυση δεδομένων**

Πραγματοποιείται βασική διερευνητική ανάλυση δεδομένων για την κατανόηση των δεδομένων. Αυτό περιλαμβάνει την εκτύπωση των πρώτων σειρών των δεδομένων, τον έλεγχο για μηδενικές τιμές και την οπτικοποίηση της κατανομής των κλάσεων και του πίνακα συσχέτισης.

♦ **Καθαρισμός δεδομένων**

Αυτό το αρχικό βήμα περιλαμβάνει τον καθαρισμό των δεδομένων με τον χειρισμό των ελλιπών τιμών και την αφαίρεση των διπλοτύπων.

♦ **Αφαίρεση ακραίων τιμών**

Η μέθοδος που ακολουθήθηκε χρησιμοποιεί το Z-score για κάθε τιμή και αφαιρεί τις τιμές με Z-score μεγαλύτερο από 3. Το Z-score είναι ο αριθμός των τυπικών αποκλίσεων ενός σημείου δεδομένων από το μέσο όρο. Ένα Z-score 0 υποδηλώνει ότι το σημείο δεδομένων βρίσκεται ακριβώς στη μέση τιμή του συνόλου δεδομένων. Ένα Z-score 1, αφορά μια τιμή που απέχει μία τυπική απόκλιση από το μέσο όρο. Έτσι, όταν το σενάριο υπολογίζει το Z-score για κάθε τιμή και αφαιρεί εκείνα με Z-score μεγαλύτερο από 3, αφαιρεί όλα τα σημεία δεδομένων που απέχουν περισσότερο από 3 τυπικές αποκλίσεις από το μέσο όρο. Αυτή είναι μια κοινή πρακτική, καθώς σε μια κανονική κατανομή, περίπου το 99,7% των δεδομένων εμπίπτει σε τρεις τυπικές αποκλίσεις από τον μέσο όρο. Έτσι, κάθε σημείο δεδομένων με Z-score μεγαλύτερο από 3 ή μικρότερο από -3 θεωρείται εξαιρετικά ασυνήθιστο και, ως εκ τούτου, ακραίο.

♦ **Υπερδειγματοληψία**

Για το σκοπό αυτό εφαρμόστηκε η SMOTE (Synthetic Minority Over-sampling Technique), που είναι μία μέθοδος για τον χειρισμό της ανισορροπίας κλάσεων σε ένα σύνολο δεδομένων. Σε πολλά προβλήματα ταξινόμησης του πραγματικού κόσμου, οι κλάσεις δεν είναι ομοιόμορφα κατανεμημένες. Για παράδειγμα, σε ένα σύνολο ιατρικών δεδομένων όπου προσπαθούμε να προβλέψουμε μια σπάνια ασθένεια, τα περισσότερα δείγματα θα είναι αρνητικά (καμία ασθένεια) και μόνο λίγα θα είναι θετικά (ασθένεια). Αυτό αποτελεί πρόβλημα επειδή οι περισσότεροι αλγόριθμοι μηχανικής μάθησης έχουν σχεδιαστεί για να λειτουργούν καλύτερα όταν ο αριθμός των δειγμάτων σε κάθε κλάση είναι περίπου ίσος. Η μέθοδος SMOTE λειτουργεί δημιουργώντας συνθετικά δείγματα από την μειοψηφική κλάση αντί για τη δημιουργία αντιγράφων. Ο αλγόριθμος επιλέγει δύο ή περισσότερα παρόμοια δείγματα (χρησιμοποιώντας ένα μέτρο απόστασης) και μεταβάλλει κάθε φορά ένα δείγμα κατά ένα τυχαίο ποσό εντός της διαφοράς από τα γειτονικά δείγματα. Αν υποθέσουμε ότι έχουμε ένα σύνολο δεδομένων με δύο χαρακτηριστικά, X_1και X_2, και χρησιμοποιούμε το SMOTE για να κάνουμε υπερδειγματοληψία της κλάσης της μειοψηφίας. Πρώτον, για κάθε δείγμα στην κλάση της μειοψηφίας, ο αλγόριθμος βρίσκει τους k κοντινότερους γείτονές του (το k είναι μια παράμετρος που μπορεί να οριστεί εκ των προτέρων, συνήθως είναι 5). Στη συνέχεια, επιλέγει τυχαία έναν από αυτούς τους γείτονες. Ας πούμε ότι για ένα συγκεκριμένο δείγμα, x, ο επιλεγμένος γείτονας είναι το x_n. Στη συνέχεια, θα δημιουργήσει ένα σύνθετο δείγμα μεταβάλλοντας το x. Αυτό πραγματοποιείται παίρνοντας έναν τυχαίο αριθμό μεταξύ 0 και 1, πολλαπλασιάζοντάς τον με τη διαφορά μεταξύ του x και του x_nκαι προσθέτοντάς τον στο x. Έτσι, για κάθε χαρακτηριστικό X_1, η τιμή X_1του συνθετικού δείγματος θα είναι:

$$x_{(i)} + (\tau\upsilon\chi\alpha\acute{\iota}o\varsigma\ \alpha\rho\iota\theta\mu\acute{o}\varsigma\ \mu\varepsilon\tau\alpha\xi\acute{\upsilon}\ 0\ \kappa\alpha\iota\ 1) * (x_{n(i)} - x_{(i)})$$

Για παράδειγμα, εάν $x = (2, 3)$ και $x_n = (3, 5)$, ένα συνθετικό δείγμα μπορεί να είναι:

$X_1 = 2 + (0.5) * (3 - 2) = 2.5$

$X_2 = 3 + (0.1) * (5 - 3) = 3.2$

Έτσι, το σύνθετο δείγμα θα γίνει $(2{,}5, 3{,}2)$.

Αυτή η διαδικασία επαναλαμβάνεται μέχρι να δημιουργηθεί ο επιθυμητός αριθμός σύνθετων δειγμάτων. Με αυτόν τον τρόπο, αντί να αντιγράφει τα ίδια δείγματα της μειοψηφικής κατηγορίας, δημιουργεί "νέα" δείγματα που είναι παρόμοια αλλά όχι πανομοιότυπα με αυτά του συνόλου δεδομένων. Αυτό μπορεί να οδηγήσει σε ένα καλύτερο μοντέλο, επειδή αναγκάζει τον αλγόριθμο να μάθει πιο γενικά μοτίβα στα δεδομένα, αντί να απομνημονεύει απλώς τα δείγματα της μειοψηφικής κλάσης.

5.3 Επόμενα βήματα

♦ **Συλλογή Αποτελεσμάτων**

Στη συνέχεια συλλέγονται επίσης διάφορες πληροφορίες, όπως ο χρόνος εύρεσης των βέλτιστων παραμέτρων για κάθε αλγόριθμο ξεχωριστά, ο χρόνος εκτέλεσης του μοντέλου, οι καλύτερες παράμετροι και η αναφορά ταξινόμησης για κάθε μοντέλο. Οι πληροφορίες αυτές αποθηκεύονται στη συνέχεια σε αρχεία κειμένου για μεταγενέστερη ανάλυση.

♦ **Βελτιστοποίηση αποδόσεων**

Η εφαρμογή των αλγορίθμων από μόνη της δεν αρκεί, καθώς σε κάθε νέο πρόβλημα προς εξέταση χρειάζεται να βρεθούν οι βέλτιστοι παράμετροι που θα ικανοποιούν κατά το μέγιστο το υπό-εξέταση

πρόβλημα. Η διαδικασία της βελτιστοποίησης παραμέτρων αποτελεί μία χρονοβόρα και λεπτομερή διαδικασία, καθώς πρέπει μέσα από αλλεπάλληλες δοκιμές να βρεθούν εκείνες οι παράμετροι για τον κάθε αλγόριθμο ξεχωριστά, που θα δώσουν τα βέλτιστα αποτελέσματα για τον κάθε αλγόριθμο αντίστοιχα. Μόνο αφού βρεθούν οι κατάλληλες παράμετροι μπορεί να πραγματοποιηθεί η πειραματική αξιολόγηση επί ίσοις όροις για όλους τους αλγορίθμους. Στην περίπτωσή μας αυτή η διαδικασία πραγματοποιήθηκε σε ξεχωριστό σενάριο κώδικα. Παρακάτω στην ενότητα 5.6, αρχικά αναλύεται ο κώδικας που περιέχεται στο ΠΑΡΑΡΤΗΜΑ και στη συνέχεια αναλύονται οι παράμετροι που εξετάστηκαν ανά αλγόριθμο. Τα αποτελέσματα της βελτιστοποίησης παρουσιάζονται στην ενότητα 6.

- **Εύρεση καλύτερου μοντέλου**

Τέλος, πραγματοποιείται ο εντοπισμός του καλύτερου μοντέλου. Αυτό πραγματοποιείται από το αποτελέσματα της υψηλότερης βαθμολογίας διασταυρούμενης επικύρωσης (cross validation), η οποία κατατάσσει το καλύτερο μοντέλο για κάθε αλγόριθμο. Το σενάριο τελειώνει με την εμφάνιση των λεπτομερειών του επικρατέστερου μοντέλου.

5.4 Μετρικές

Οι μετρικές που χρησιμοποιήθηκαν για την αξιολόγηση των αποτελεσμάτων είναι οι Accuracy, Precision, Recall, F1-score, ROC-AUC, PRC-AUC και MCC. Αυτές οι μετρικές χρησιμοποιούνται συχνά σε συνδυασμό για να παρέχουν μια ολοκληρωμένη επισκόπηση της απόδοσης του μοντέλου ταξινόμησης. Στο παρακάτω τμήμα γίνεται μία συνοπτική παρουσίαση για την καθεμία από αυτές:

- ***Ορθότητα (Accuracy)***

Αυτή είναι μία από τις πιο απλές και ευρέως χρησιμοποιούμενη μορφή μετρήσεων που χρησιμοποιούνται στην ταξινόμηση. Είναι ο λόγος του αριθμού των σωστών προβλέψεων προς τον συνολικό αριθμό των προβλέψεων που έγιναν. Υπολογίζεται χρησιμοποιώντας τον τύπο:

$$\text{Ορθότητα} = \frac{\text{Αριθμός σωστών προβλέψεων}}{\text{Συνολικός αριθμός προβλέψεων}}$$

- ***Ακρίβεια (Precision)***

Η ακρίβεια είναι ο λόγος των σωστά προβλεπόμενων θετικών παρατηρήσεων προς το σύνολο των προβλεπόμενων θετικών παρατηρήσεων. Ονομάζεται επίσης θετική προβλεπτική αξία. Υπολογίζεται ως εξής:

$$\text{Ακρίβεια} = \frac{\text{Αληθώς Θετικά}}{\text{Αληθώς Θετικά} + \text{Ψευδώς Θετικά}}$$

- ***Ανάκληση (Ευαισθησία) (Recall)***

Η ανάκληση είναι ο λόγος των σωστά προβλεπόμενων θετικών παρατηρήσεων προς το σύνολο των παρατηρήσεων της πραγματικής θετικής κλάσης. Ονομάζεται επίσης ευαισθησία (Sensitivity), ποσοστό επιτυχίας ή ποσοστό αληθώς θετικών αποτελεσμάτων. Υπολογίζεται ως εξής:

$$\text{Ανάκληση} = \frac{\text{Αληθώς Θετικά}}{\text{Αληθώς Θετικά} + \text{Ψευδώς Αρνητικά}}$$

- ***F1-Score***

Το F1-Score είναι ο σταθμισμένος μέσος όρος της Ακρίβειας και της Ανάκλησης. Είναι ένας καλός τρόπος για αποδειχθεί ότι ένας ταξινομητής έχει καλή τιμή τόσο για την ανάκληση όσο και για την ακρίβεια. Το F1-Score είναι ο αρμονικός μέσος όρος της ακρίβειας και της ανάκλησης:

$$\text{F1} - \text{Score} = \frac{2 * \text{Ακρίβεια} * \text{Ανάκληση}}{\text{Ακρίβεια} + \text{Ανάκληση}}$$

- ***Συντελεστής συσχέτισης Matthews (Matthews Correlation Coefficient (MCC))***

Ο συντελεστής συσχέτισης Matthews χρησιμοποιείται στη μηχανική μάθηση ως μέτρο της ποιότητας των δυαδικών ταξινομήσεων (δύο κλάσεων). Λαμβάνει υπόψη τα αληθώς και ψευδώς θετικά και αρνητικά αποτελέσματα και θεωρείται γενικά ως ένα ισορροπημένο μέτρο που μπορεί να χρησιμοποιηθεί ακόμη και αν οι κλάσεις έχουν πολύ διαφορετικά μεγέθη. Το MCC είναι στην ουσία μια τιμή του συντελεστή συσχέτισης μεταξύ -1 και +1. Ένας συντελεστής +1 αντιπροσωπεύει μια τέλεια πρόβλεψη, 0 μια μέση τυχαία πρόβλεψη και -1 μια αντίστροφη πρόβλεψη. Υπολογίζεται ως εξής:

$$MCC = \frac{\text{Αληθώς Θετικά} * \textit{Αληθώς} \text{ Αρνητικά} - \text{Ψευδώς Θετικά} * \textit{Ψευδώς} \text{ Αρνητικά}}{\sqrt{(\textit{Αληθώς Θετικά} + \textit{Ψευδώς Θετικά}) * (\textit{Αληθώς Θετικά} + \textit{Ψευδώς Αρνητικά}) * (\textit{Αληθώς Αρνητικά} + \textit{Ψευδώς Θετικά}) * (\textit{Αληθώς Αρνητικά} + \textit{Ψευδώς Αρνητικά})}}$$

- ***ROC-AUC***

Το ROC σημαίνει Receiver Operating Characteristic και AUC σημαίνει Area Under the Curve. Η ROC είναι μια γραφική παράσταση του ποσοστού αληθώς θετικών (Recall) έναντι του ποσοστού ψευδώς θετικών (1-Specificity) για τα διάφορα πιθανά σημεία αποκοπής μιας διαγνωστικής δοκιμής. Η AUC (Area Under Curve) είναι η περιοχή κάτω από την καμπύλη ROC. Μια AUC με τιμή 1 υποδηλώνει τέλεια ταξινόμηση, ενώ μια AUC 0,5 υποδηλώνει ένα μοντέλο που δεν είναι καλύτερο από την τυχαία εικασία.

- ***PRC-AUC***

Η PRC σημαίνει Precision-Recall Curve και αφορά τη καμπύλη ακρίβειας-ανάκλησης που δείχνει το συμβιβασμό μεταξύ ακρίβειας και ανάκλησης για διαφορετικά κατώτατα όρια. Μια υψηλής τιμής περιοχή κάτω από την καμπύλη αντιπροσωπεύει τόσο υψηλή ανάκληση όσο και υψηλή ακρίβεια, όπου η υψηλή ακρίβεια σχετίζεται με χαμηλό ποσοστό ψευδώς θετικών αποτελεσμάτων και η υψηλή ανάκληση σχετίζεται με χαμηλό ποσοστό ψευδώς αρνητικών αποτελεσμάτων.

5.5 Ανάλυση Κύριων Συνιστωσών – Principal Component Analysis(PCA)

Ως τελευταίο βήμα, εφαρμόστηκε Ανάλυση Κύριων Συνιστωσών (PCA), μιας στατιστικής τεχνικής που χρησιμοποιείται για την εξαγωγή και τον προσδιορισμό των κύριων συνιστωσών σε ένα σύνολο δεδομένων με πολλές μεταβλητές, μειώνοντας τη διαστασιμότητα των δεδομένων χωρίς να χάνεται πολύ πληροφορία. Η τεχνική αυτή περιλαμβάνει τον υπολογισμό της ιδιοτιμής και του ιδιοδιανύσματος για μια δεδομένη συνδιακύμανση ή συσχέτιση μεταξύ των μεταβλητών. Οι κύριες

συνιστώσες είναι ένα νέο σύνολο μεταβλητών που προκύπτει από τις αρχικές μεταβλητές και είναι γραμμικά ανεξάρτητες μεταξύ τους. Ο στόχος της PCA είναι να βρει μια νέα βάση για τα δεδομένα, όπου η πρώτη συνιστώσα έχει την μεγαλύτερη δυνατή διασπορά, η δεύτερη συνιστώσα (που είναι ορθογώνια προς την πρώτη) έχει τη δεύτερη μεγαλύτερη διασπορά και ούτω καθεξής. Η PCA είναι ευρέως χρησιμοποιημένη στη στατιστική, στην επεξεργασία σήματος, στην υπολογιστική όραση και στην μηχανική μάθηση για την προεπεξεργασία δεδομένων. Παρακάτω παρουσιάζονται τα βήματα υλοποίησης της μεθοδολογίας για την εφαρμογή του PCA όπως εφαρμόστηκαν στο ΠΑΡΑΡΤΗΜΑ 9.5.

- **Προεπεξεργασία δεδομένων - Μείωση δεδομένων**

Αυτό περιλαμβάνει τη μείωση της διαστασιμότητας των δεδομένων, δηλαδή τη μείωση του αριθμού των υπό εξέταση τυχαίων μεταβλητών και μπορεί να χωριστεί σε επιλογή χαρακτηριστικών και εξαγωγή χαρακτηριστικών.

- **Συλλογή αποτελεσμάτων**

Στο σενάριο αυτό συλλέγονται επίσης διάφορες πληροφορίες, όπως ο χρόνος εύρεσης των βέλτιστων παραμέτρων για κάθε αλγόριθμο ξεχωριστά, ο χρόνος εκτέλεσης του μοντέλου, οι καλύτερες παράμετροι και η αναφορά ταξινόμησης για κάθε μοντέλο. Οι πληροφορίες αυτές αποθηκεύονται στη συνέχεια σε αρχεία κειμένου για μεταγενέστερη ανάλυση.

- **Εύρεση καλύτερου μοντέλου**

Τέλος, το σενάριο εντοπίζει το μοντέλο με την υψηλότερη βαθμολογία διασταυρούμενης επικύρωσης (cross validation) ως το καλύτερο μοντέλο και εκτυπώνει τις λεπτομέρειές του.

5.6 Επεξήγηση Κώδικα - Εφαρμογή Σεναρίων στα Σύνολα Δεδομένων

Τα παρακάτω σενάρια (τμήματα κώδικα) όπως παρουσιάζονται κατά σειρά, εφαρμόζουν τη μεθοδολογία για να επιτευχθεί η πειραματική αξιολόγηση και να εξαχθούν τα αποτελέσματα που θα οδηγήσουν στα τελικά συμπεράσματα. Σε αυτό το τμήμα γίνεται αρχικά συνολική και μετέπειτα λεπτομερή παρουσίαση των σεναρίων κώδικα, για την καλύτερη κατανόηση τους, από τον αναγνώστη. Αρχικά παρουσιάζονται οι κοινές βιβλιοθήκες, μεταβλητές και συναρτήσεις που χρησιμοποιήθηκαν στα σενάρια, ώστε να είναι ευκολότερη η πλοήγηση του αναγνώστη και να μπορεί να γίνει πιο κατανοητή η δομή των προγραμμάτων. Στη συνέχεια αναλύονται κατά σειρά τα σενάρια, όπως εφαρμόστηκαν κατά την εκτέλεση της μεθοδολογίας.

5.6.1 Βιβλιοθήκες

- ***Pandas***

Χρησιμοποιείται για τη διαχείριση και την επεξεργασία δεδομένων, σε μορφή DataFrame. Συγκεκριμένα, προσφέρει δομές δεδομένων και λειτουργίες για τον χειρισμό αριθμητικών πινάκων και χρονοσειρών.

- ***Matplotlib και Seaborn***

Χρησιμοποιούνται για την οπτικοποίηση των δεδομένων. Η matplotlib.pyplot είναι μια ενότητα της Matplotlib που παρέχει μια διεπαφή που μοιάζει με το MATLAB. Η Matplotlib έχει σχεδιαστεί για τη δημιουργία όλων των ειδών των γραφικών παραστάσεων και διαγραμμάτων με την Python.

Επιπλέον, η Seaborn είναι μια βιβλιοθήκη οπτικοποίησης δεδομένων Python που βασίζεται στην Matplotlib. Παρέχει μια διεπαφή υψηλού επιπέδου για τη σχεδίαση ελκυστικών και κατατοπιστικών στατιστικών γραφημάτων.

- ***time***

Χρησιμοποιείται για τη χρονομέτρηση των διαφόρων διαδικασιών. Παρέχει διάφορες συναρτήσεις που σχετίζονται με τον χρόνο. Εδώ, χρησιμοποιείται για την καταγραφή του χρόνου εκτέλεσης τμημάτων του κώδικα.

- ***sklearn***

Το Scikit-learn (sklearn) είναι μια βιβλιοθήκη μηχανικής μάθησης ελεύθερου λογισμικού για τη γλώσσα προγραμματισμού Python. Διαθέτει διάφορους αλγορίθμους ταξινόμησης, παλινδρόμησης και ομαδοποίησης και έχει σχεδιαστεί για να συνεργάζεται με τις αριθμητικές και επιστημονικές βιβλιοθήκες NumPy και SciPy της Python. Χρησιμοποιείται επίσης για την προεπεξεργασία των δεδομένων, το διαχωρισμό τους σε σύνολα εκπαίδευσης και ελέγχου, την εφαρμογή των αλγορίθμων εκπαίδευσης και ταξινόμησης και την αξιολόγηση των μοντέλων.

- ***scipy***

Αυτή η βιβλιοθήκη περιέχει έναν μεγάλο αριθμό κατανομών πιθανοτήτων καθώς και μια βιβλιοθήκη στατιστικών συναρτήσεων. Στην περίπτωσή μας, χρησιμοποιείται για την εισαγωγή της zscore για τον υπολογισμό του Z-Score στην ενότητα της προεπεξεργασίας δεδομένων.

- ***imblearn***

Αυτή η βιβλιοθήκη παρέχει μεθόδους για τον χειρισμό μη ισορροπημένων συνόλων δεδομένων. Η SMOTE (Synthetic Minority Over-sampling Technique) χρησιμοποιείται για την αντιμετώπιση της ανισορροπίας κλάσεων στην ενότητα της προεπεξεργασίας. Στην περίπτωσή μας χρησιμοποιείται για την υπερδειγματοληψία των δεδομένων ώστε να επιτευχθεί ισορροπία μεταξύ των δύο κλάσεων.

5.6.2 Συναρτήσεις και μεταβλητές

- ***train_test_split***

Χρησιμοποιείται για το διαχωρισμό των δεδομένων σε σύνολα εκπαίδευσης και ελέγχου.

- ***df.apply()***

Αυτή η συνάρτηση εφαρμόζει μια συνάρτηση κατά μήκος ενός άξονα του DataFrame. Εδώ, εφαρμόζει τη συνάρτηση zscore σε όλες τις στήλες.

- ***GridSearchCV***

Χρησιμοποιείται για τη βελτιστοποίηση των παραμέτρων των μοντέλων.

- ***classification_report, roc_curve, auc, precision_recall_curve, matthews_corrcoef***

Χρησιμοποιούνται για την αξιολόγηση των μοντέλων και τον υπολογισμό των εκατέρωθεν μετρικών.

- ***df***

Αυτή η μεταβλητή αποθηκεύει το DataFrame που δημιουργείται από το αρχείο 'eeg-eye-state.csv'.

- ***features, labels***

Τα δεδομένα διαχωρίζονται σε χαρακτηριστικά (features) και ετικέτες (labels). Πρόκειται για αντικείμενα DataFrame που αποθηκεύουν τα χαρακτηριστικά (ανεξάρτητες μεταβλητές) και τις ετικέτες (εξαρτημένες μεταβλητές), αντίστοιχα.

- ***X_train, X_test, y_train, y_test***

Αυτές οι μεταβλητές αποθηκεύουν τα σύνολα εκπαίδευσης και ελέγχου που δημιουργούνται από τη μέθοδο train_test_split. Πρόκειται για τα σύνολα δεδομένων εκπαίδευσης και ελέγχου για τα χαρακτηριστικά και τις ετικέτες, αντίστοιχα, τα οποία λαμβάνονται από τη διάσπαση του αρχικού συνόλου δεδομένων.

- ***classifiers***

Λίστα που περιέχει τους ταξινομητές. Πρόκειται για έναν κατάλογο πλειάδων, όπου κάθε πλειάδα περιέχει μια περίπτωση ενός ταξινομητή, ένα λεξικό υπερπαραμέτρων για τον εν λόγω ταξινομητή που πρέπει να βελτιστοποιηθεί με τη χρήση του GridSearchCV και το όνομα του ταξινομητή.

- ***pipe***

Αυτό είναι ένα αντικείμενο pipeline που πρώτα μεταβάλει τα δεδομένα χρησιμοποιώντας StandardScaler και στη συνέχεια εφαρμόζει έναν ταξινομητή.

- ***grid***

Πρόκειται για ένα αντικείμενο GridSearchCV που χρησιμοποιείται για τη ρύθμιση των υπερπαραμέτρων.

- ***best_model***

Πρόκειται για το μοντέλο με τις καλύτερες παραμέτρους που λαμβάνονται από το αντικείμενο GridSearchCV.

- ***predictions and probabilities***

Αυτές είναι οι προβλεπόμενες κλάσεις και οι πιθανότητες για κάθε περίπτωση ελέγχου, αντίστοιχα.

- ***fpr, tpr, _***

Πρόκειται για τα ποσοστά ψευδώς θετικών και τα ποσοστά αληθώς θετικών, αντίστοιχα, που χρησιμοποιούνται για τον υπολογισμό της καμπύλης ROC (Receiver Operating Characteristic).

- ***roc_auc***

Πρόκειται για την περιοχή κάτω από την καμπύλη ROC.

- ***mcc***

Πρόκειται για τον συντελεστή συσχέτισης Matthews (MCC).

- ***precision, recall***

Αυτές είναι οι τιμές ακρίβειας και ανάκλησης, αντίστοιχα, που χρησιμοποιούνται για τον υπολογισμό της καμπύλης ακρίβειας-ανάκλησης (PRC).

- ***prc_auc***

Αφορά την περιοχή κάτω από την καμπύλη PRC.

- ***model_scores, model_times, accuracy_dict***

Πρόκειται για λίστες που αποθηκεύουν το όνομα, τη βαθμολογία και το χρόνο εκτέλεσης, καθώς και ένα λεξικό που αποθηκεύει την ακρίβεια κάθε μοντέλου, αντίστοιχα.

- ***hyper_tuning_time, model_exec_time, best_params***

Πρόκειται για λεξικά που αποθηκεύουν τον χρόνο που απαιτείται για τον συντονισμό των υπερπαραμέτρων, τον χρόνο που απαιτείται για την εκτέλεση του μοντέλου και τις καλύτερες παραμέτρους για κάθε μοντέλο, αντίστοιχα.

5.6.3 Αρχικό Σύνολο Δεδομένων

Όπως προαναφέρθηκε, εφαρμόζεται η μεθοδολογία για την εξαγωγή των πρώτων αποτελεσμάτων που θα χρησιμοποιηθούν ως αναφορά. Το σύνολο δεδομένων είναι το αρχικό στο οποίο δεν έχει εφαρμοστεί καμία προεπεξεργασία. Αυτό το σενάριο (ΠΑΡΑΡΤΗΜΑ 9.2) ασχολείται κυρίως με τη διαδικασία αξιολόγησης και σύγκρισης πολλαπλών ταξινομητών μηχανικής μάθησης για ένα πρόβλημα ταξινόμησης. Το πρόβλημα αφορά την ανίχνευση της κατάστασης των ματιών ασθενών (ανοιχτά ή κλειστά) με βάση δεδομένα ΗΕΓ (ηλεκτροεγκεφαλογραφήματος). Συνολικά, το σενάριο αποτελεί μια ολοκληρωμένη λύση για την αξιολόγηση πολλαπλών ταξινομητών για το πρόβλημα ανίχνευσης της κατάστασης των ματιών EEG και παρέχει λεπτομερείς οπτικοποιήσεις και μετρήσεις για τη σύγκριση και την επιλογή του καλύτερου ταξινομητή. Το σενάριο μπορεί να γίνει κατανοητό με τα ακόλουθα βήματα:

- ***Εισαγωγή των απαραίτητων βιβλιοθηκών***

Το σενάριο ξεκινά με την εισαγωγή των απαραίτητων βιβλιοθηκών Python, οι οποίες περιλαμβάνουν βιβλιοθήκες χειρισμού δεδομένων (pandas), οπτικοποίησης δεδομένων (matplotlib, seaborn) και μηχανικής μάθησης (sklearn) και άλλες.

- ***Φόρτωση δεδομένων***

Τα δεδομένα φορτώνονται από ένα αρχείο CSV eeg-eye-state.csv σε ένα πλαίσιο δεδομένων pandas DataFrame (df).

- ***Διερευνητική ανάλυση δεδομένων***

- Εκτυπώνονται οι πέντε πρώτες σειρές του συνόλου δεδομένων με τη χρήση του df.head().

- Εμφανίζονται τοι πληροφορίες του συνόλου δεδομένων (τύποι δεδομένων, αριθμός μη μηδενικών τιμών για κάθε στήλη κ.λπ.) χρησιμοποιώντας την df.info().
- Ελέγχεται και εκτυπώνεται ο αριθμός των ελλιπών τιμών σε κάθε στήλη.
- Εκτυπώνονται περιγραφικά στατιστικά στοιχεία (όπως μέσος όρος, τυπική απόκλιση, ελάχιστο, μέγιστο κ.λπ.) για κάθε στήλη χρησιμοποιώντας την df.describe().
- Οπτικοποιείται η κατανομή της κλάσης στόχου (eyeDetection) χρησιμοποιώντας το countplot του seaborn.

♦ ***Προετοιμασία δεδομένων***

Το σύνολο δεδομένων χωρίζεται σε χαρακτηριστικά (features) και στη μεταβλητή-στόχο (labels). Το χαρακτηριστικό με την ονομασία "eyeDetection" είναι η μεταβλητή-στόχος, υποδεικνύοντας την κατάσταση των ματιών των ασθενών (ανοικτά/κλειστά). Στη συνέχεια χωρίζονται τα δεδομένα σε σύνολα εκπαίδευσης και ελέγχου.

♦ ***Ρύθμιση υπερπαραμέτρων***

Ορίζεται ένας κατάλογος ταξινομητών μαζί με τις υπερπαραμέτρους τους που πρέπει να βελτιστοποιηθούν. Οι ταξινομητές περιλαμβάνουν τους Naive Bayes, Logistic Regression, Decision Tree, KNN, Random Forest, MLP (νευρωνικό δίκτυο), SVM, AdaBoost και Bagging.

Για κάθε ταξινομητή:

- Δημιουργείται ένας αγωγός (pipeline) ο οποίος πρώτα κλιμακώνει τα δεδομένα χρησιμοποιώντας το StandardScaler και στη συνέχεια εφαρμόζει τον ταξινομητή.
- Χρησιμοποιείται αναζήτηση πλέγματος (GridSearchCV) για την εύρεση των καλύτερων υπερπαραμέτρων για τον ταξινομητή.
- Εκπαιδεύεται το καλύτερο μοντέλο και υπολογίζονται και εκτυπώνονται οι μετρικές απόδοσης (όπως ακρίβεια, ανάκληση, βαθμολογία F1, ROC AUC, MCC).
- Οι καμπύλες ROC και Precision-Recall σχεδιάζονται για οπτικοποίηση.
- Καταγράφεται ο χρόνος εκτέλεσης για την εύρεση των υπερπαραμέτρων και την προσαρμογή του μοντέλου.

♦ ***Αποτελέσματα και οπτικοποίηση***

- Οι καλύτερες παράμετροι για κάθε μοντέλο αποθηκεύονται σε αρχεία κειμένου.
- Εντοπίζεται το μοντέλο με την καλύτερη βαθμολογία (απόδοση).
- Δημιουργείται μια ενοποιημένη καμπύλη ROC για όλους τους ταξινομητές για την οπτική σύγκριση των επιδόσεών τους.
- Οι λεπτομέρειες του καλύτερου μοντέλου αποθηκεύονται σε αρχείο κειμένου.

5.6.4 Βελτιστοποίηση Παραμέτρων

Σε αυτό το στάδιο προχωράμε στην προεπεξεργασία του αρχικού συνόλου δεδομένων (ΠΑΡΑΡΤΗΜΑ 9.3). Αφού εφαρμοστούν οι κατάλληλες επεξεργασίες αρχίζει η αναζήτηση των βέλτιστων παραμέτρων με cross validation για να αποθηκευτούν οι βέλτιστες παράμετροι που θα εφαρμοστούν στη συνέχεια. Αναλυτικότερα, το σενάριο φορτώνει τα δεδομένα από ένα csv αρχείο, εντοπίζει και αφαιρεί τις ακραίες τιμές, και στη συνέχεια εφαρμόζει την τεχνική SMOTE για την

υπερδειγματοληψία των δεδομένων. Μετά από αυτή την προεπεξεργασία, τα δεδομένα εξετάζονται και οπτικοποιούνται. Το επόμενο στάδιο του σεναρίου είναι η βελτιστοποίηση των υπερπαραμέτρων για διάφορους ταξινομητές, συγκεκριμένα των Gaussian Naive Bayes, Logistic Regression, Decision Tree, KNN, Random Forest, MLP, SVM, AdaBoost και Bagging. Οι ταξινομητές αυτοί εκπαιδεύονται και αξιολογούνται με τη χρήση των δεδομένων εκπαίδευσης και ελέγχου. Αρχικά ορίζεται ένας κατάλογος των ανωτέρω ταξινομητών με τις αντίστοιχες υπερπαραμέτρους τους για βελτιστοποίηση. Ακολουθεί περιγραφή κάθε ταξινομητή και των υπερπαραμέτρων του:

- ***Naive Bayes (GaussianNB)***

  Αυτός ο ταξινομητής δεν διαθέτει υπερπαραμέτρους για βελτιστοποίηση.

- ***Λογιστική παλινδρόμηση (Logistic Regression)***

  C: Η υπερπαράμετρος 'C' στη Λογιστική Παλινδρόμηση συμβολίζει το αντίστροφο της δύναμης κανονικοποίησης. Η κανονικοποίηση χρησιμοποιείται για την αποτροπή της υπερπροσαρμογής με την προσθήκη ενός όρου ποινής στη συνάρτηση απώλειας. Μια μικρότερη τιμή του C καθορίζει ισχυρότερη κανονικοποίηση. Οι τιμές που εξετάζονται είναι οι 0,01, 0,1 και 1.

  Penalty: Αυτή καθορίζει τη ρύθμιση που χρησιμοποιείται στην εφαρμογή ποινών. Οι ποινές που εξετάστηκαν είναι η 'l2' ή καμία (none).

- ***Δέντρα απόφασης (Decision Trees)***

  max_depth: Το μέγιστο βάθος του δέντρου. Οι τιμές που εξετάζονται είναι οι 5,10 ή χωρίς περιορισμό στο βάθος (none).

- ***K-Nearest Neighbors (KNN)***

  n_neighbors: Αφορά τον αριθμό των γειτόνων που θα εξετάσει ο αλγόριθμος. Το εύρος του αριθμού θα είναι για τιμές από 1 έως 30.

  weights: Συνάρτηση βάρους που χρησιμοποιείται στην πρόβλεψη. Οι τιμές που δοκιμάζονται είναι οι 'uniform' και 'distance'.

- ***Random Forest***

  n_estimators: Ο αριθμός των δέντρων στο «δάσος» από δέντρα που δημιουργείται. Οι τιμές που εξετάζονται είναι 50, 100 και 200 δέντρα.

  max_features: Ο αριθμός των χαρακτηριστικών που πρέπει να ληφθούν υπόψη κατά την αναζήτηση του καλύτερου διαχωρισμού. Οι συναρτήσεις που εξετάζονται είναι οι 'sqrt' και 'log2'.

- ***Multi-layer Perceptron (MLP)***

  alpha: Είναι η παράμετρος ποινής L2 (όρος κανονικοποίησης). Οι τιμές προς δοκιμή είναι 0,0001, 0,001 και 0,01.

  learning_rate_init: Ο αρχικός ρυθμός μάθησης που χρησιμοποιείται. Οι τιμές που εξετάζονται είναι 0,001, 0,01 και 0,1.

  momentum: Το momentum για την ανανέωση της gradient descent. Οι τιμές προς δοκιμή είναι 0,2, 0,5 και 0,9.

hidden_layer_sizes: Ο αριθμός των νευρώνων στα κρυφά επίπεδα του νευρωνικού δικτύου. Οι τιμές προς εξέταση είναι (10,), (50,), (100,) και (50,50).

activation: Αφορά τη συνάρτηση ενεργοποίησης για το κρυφό επίπεδο. Οι συναρτήσεις που εξετάζονται είναι οι "tanh" και "relu".

- ***Support Vector Machine (SVM)***

  C: Παράμετρος κανονικοποίησης. Η ισχύς της κανονικοποίησης είναι αντιστρόφως ανάλογη του C. Οι τιμές προς δοκιμή είναι 0,01, 0,1 και 1.

  Kernel: Καθορίζει τον τύπο πυρήνα που θα χρησιμοποιηθεί στον αλγόριθμο. Οι διαφορετικές μορφές πυρήνα που θα δοκιμαστούν είναι οι γραμμικός ('linear'), πολυωνυμικός('poly') και ο γκαουσιανός ('rbf').

- ***AdaBoost***

  n_estimators: Ο μέγιστος αριθμός εκτιμητών στον οποίο τερματίζεται η ενίσχυση. Οι τιμές που εξετάζονται είναι οι 50 και 100.

  learning_rate: Ο ρυθμός μάθησης συρρικνώνει τη συνεισφορά κάθε ταξινομητή. Οι τιμές που δοκιμάζονται είναι οι 0,01, 0,1 και 1.

- ***Bagging***

  n_estimators: Ο αριθμός των βασικών εκτιμητών στο σύνολο. Οι τιμές προς εξέταση είναι οι 5, 10, 15, 20 και 30.

Κάθε ταξινομητής δοκιμάζεται με διαφορετικούς συνδυασμούς αυτών των υπερπαραμέτρων για να βρεθεί ο συνδυασμός που αποδίδει καλύτερα.

5.6.5 K-Fold Cross Validation

Για την πραγματοποίηση της βελτιστοποίησης παραμέτρων πραγματοποιείται η διασταυρούμενη επικύρωση Κ αναδιπλώσεων, ώστε να επιτευχθεί η όσο το δυνατόν πιο έγκυρη βέλτιστη εύρεση των κατάλληλων υπερπαραμέτρων. Στην περίπτωση μας χρησιμοποιήθηκε διασταυρούμενη επικύρωση 5 αναδιπλώσεων (ομάδων δεδομένων).

Συνοπτικά, η διασταυρούμενη επικύρωση Κ αναδιπλώσεων είναι μια διαδικασία επαναδειγματοληψίας που χρησιμοποιείται για την αξιολόγηση μοντέλων μηχανικής μάθησης σε ένα περιορισμένο δείγμα δεδομένων. Η διαδικασία έχει μία μόνο παράμετρο που ονομάζεται "k" και αναφέρεται στον αριθμό των ομάδων στις οποίες θα χωριστεί ένα δεδομένο δείγμα δεδομένων. Μόλις τα δεδομένα χωριστούν σε k ομάδες (ή "αναδιπλώσεις"), ο αλγόριθμος εκπαιδεύεται σε k-1 από αυτές τις αναδιπλώσεις και δοκιμάζεται στις υπόλοιπες. Αυτή η διαδικασία επαναλαμβάνεται k φορές, με κάθε αναδίπλωση να χρησιμοποιείται ακριβώς μία φορά ως σύνολο δοκιμής. Τα k αποτελέσματα από τις αναδιπλώσεις μπορούν στη συνέχεια να υπολογιστούν κατά μέσο όρο για να προκύψει ένα ενιαίο σκορ απόδοσης.

Η διασταυρούμενη επικύρωση Κ-πτυχών χρησιμοποιείται συχνά στη διαδικασία επιλογής μοντέλου όπου προσπαθεί κανείς να καθορίσει τις βέλτιστες υπερπαραμέτρους. Με την εκπαίδευση του μοντέλου σε k-1 αναδιπλώσεις και την επικύρωση στην υπόλοιπη αναδίπλωση, μπορεί κανείς να

λάβει ένα πιο αξιόπιστο μέτρο της απόδοσης του μοντέλου, καθώς δίνει μια εικόνα για το πώς το μοντέλο αποδίδει σε διαφορετικά υποσύνολα των δεδομένων. Αυτό είναι ιδιαίτερα σημαντικό, εάν τα δεδομένα έχουν κάποιου είδους δομή ή διάταξη που το μοντέλο θα μπορούσε να μάθει κατά λάθος, οδηγώντας έτσι σε ένα μεροληπτικό ή υπερβολικά προσαρμοσμένο μοντέλο.

5.6.6 Επεξεργασμένο Σύνολο Δεδομένων

Σε αυτό το σενάριο (ΠΑΡΑΡΤΗΜΑ 9.4) γίνεται εφαρμογή των βέλτιστων υπερπαραμέτρων που βρέθηκαν στο προηγούμενο σενάριο για την εύρεση και οπτικοποίηση των τελικών αποτελεσμάτων. Όπως προαναφέρθηκε τα βήματα υλοποίησης είναι κοινά, για λόγους απλοποίησης, κατανόησης και ευκολότερης σύγκρισης, με την εκάστοτε προσθήκη των απαραίτητων λειτουργιών. Σε αυτό το σενάριο συγκεκριμένα η στόχευση είναι η παραγωγή των τελικών αποτελεσμάτων, η αποθήκευσή τους και η συγκριτική οπτικοποίησή τους.

- ***Φόρτωση βιβλιοθηκών***

Το πρόγραμμα αρχικά φορτώνει διάφορες βιβλιοθήκες, μεταξύ των οποίων βιβλιοθήκες για την επεξεργασία και ανάλυση δεδομένων (pandas, matplotlib, scipy), βιβλιοθήκες για την εκπαίδευση και ελέγχου μηχανικών μοντέλων μάθησης (sklearn) και μια βιβλιοθήκη για την επεξεργασία μη ισορροπημένων δεδομένων (imblearn).

- ***Προεπεξεργασία δεδομένων***

Τα δεδομένα φορτώνονται από ένα CSV αρχείο με την pd.read_csv(). Εκτελείται εντοπισμός και αφαίρεση ακραίων τιμών (outliers) με τη βοήθεια της στατιστικής της z-score. Έπειτα, γίνεται υπερδειγματοληψία των μειοψηφικών κλάσεων με την τεχνική SMOTE για να διορθώσει τυχόν ανισορροπίες στα δεδομένα. Τέλος, τα δεδομένα διαχωρίζονται σε σύνολα εκπαίδευσης και ελέγχου.

- ***Φόρτωση αποθηκευμένων υπερπαραμέτρων***

Δημιουργείται ένα λεξικό που περιέχει τις βέλτιστες υπερπαραμέτρους για την εκπαίδευση κάθε μοντέλου.

- ***Κατηγοριοποίηση***

Εκπαιδεύονται διάφορα μοντέλα με τη βοήθεια ενός pipeline, που πρώτα εκτελεί κανονικοποίηση των δεδομένων με το StandardScaler(). Το script εκπαιδεύει και αξιολογεί διάφορα μοντέλα ταξινόμησης όπως τα Naive Bayes, Logistic Regression, Decision Tree, KNN, Random Forest, MLP, SVM και δύο ensemble learning μοντέλα, το AdaBoost και το Bagging. Κάθε μοντέλο εκπαιδεύεται και αξιολογείται χρησιμοποιώντας τις βέλτιστες υπερπαραμέτρους που αποθηκεύονται σε ένα λεξικό. Κατά την εκπαίδευση, το αντίστοιχο τμήμα του κώδικα χρονομετρά τη διάρκεια της διαδικασίας για κάθε μοντέλο και εμφανίζει τη βαθμολογία (accuracy) του μοντέλου, την αναφορά (report) ταξινόμησης, τον δείκτη AUC της καμπύλης ROC και τον δείκτη MCC. Μετά την εκπαίδευση, γίνονται προβλέψεις βάσει του συνόλου ελέγχου και υπολογίζονται διάφορες μετρικές απόδοσης, όπως το ROC AUC και το MCC.

- ***Οπτικοποίηση***

Οπτικοποιούνται τα τελικά αποτελέσματα που βρέθηκαν μετά την εφαρμογή των βέλτιστων υπερπαραμέτρων. Οι οπτικοποιήσεις αυτές περιλαμβάνουν τις οπτικοποίησης των καμπυλών ROC και Precision-Recall για μεταγενέστερη ερμηνεία-ανάλυση των αποτελεσμάτων.

5.6.7 Ανάλυση Κύριων Συνιστωσών

Σε αυτό το τμήμα κώδικα (ΠΑΡΑΡΤΗΜΑ 9.5) εφαρμόζεται η μέθοδος ανάλυσης κύριων συνιστωσών. Ο σκοπός είναι η μείωση της διαστασιμότητας των δεδομένων μας, η εύρεση των κατάλληλων συνιστωσών για την βέλτιστη ακρίβεια του ταξινομητή που θα επιλεχθεί και η τελική εξαγωγή συμπερασμάτων για πιθανή εφαρμογή σε ανάλογα σύνολα δεδομένων ΗΕΓ μεγαλύτερης κλίμακας.

- ***Εισαγωγή βιβλιοθηκών***

Το σενάριο ξεκινάει με την εισαγωγή διαφόρων βιβλιοθηκών Python που απαιτούνται για διάφορες εργασίες, συμπεριλαμβανομένης της επεξεργασίας δεδομένων, της οπτικοποίησης και της μηχανικής μάθησης.

- ***Φόρτωση δεδομένων***

Τα δεδομένα ΗΕΓ φορτώνονται από ένα αρχείο CSV με όνομα "eeg-eye-state.csv" χρησιμοποιώντας το pandas.

- ***Προεπεξεργασία δεδομένων:***

- **Υπολογισμός του IQR (διατεταρτημοριακό εύρος)**: Το σενάριο υπολογίζει το διατεταρτημοριακό εύρος (IQR) για κάθε χαρακτηριστικό στο σύνολο δεδομένων.

- ***Προσδιορισμός ακραίων τιμών:*** Εντοπίζει τις ακραίες τιμές στα δεδομένα με βάση το υπολογισμένο IQR.

- ***Αφαίρεση των ακραίων τιμών:*** Αφαιρεί τις γραμμές που περιέχουν ακραίες τιμές από το σύνολο δεδομένων.

- ***Εξερεύνηση και οπτικοποίηση δεδομένων:*** Εκτυπώνονται οι πρώτες σειρές του συνόλου δεδομένων, πληροφορίες σχετικά με τους τύπους δεδομένων και τις ελλείπουσες τιμές, καθώς και μια συνοπτική στατιστική του συνόλου δεδομένων. Επιπλέον, δημιουργείται ένα διάγραμμα καταμέτρησης για την οπτικοποίηση της κατανομής των κλάσεων.

- ***Διαχωρισμός δεδομένων***

Το σύνολο δεδομένων χωρίζεται σε σύνολα εκπαίδευσης και δοκιμής χρησιμοποιώντας το train_test_split από το scikit-learn.

- ***Τυποποίηση***

Εφαρμόζεται τυποποίηση με τη StandardScaler στα χαρακτηριστικά των συνόλων εκπαίδευσης και ελέγχου, για μετασχηματισμό των δεδομένων έτσι ώστε η μορφή τους να έχει μέση τιμή 0 και τυπική απόκλιση 1.

- ***Οπτικοποίηση της εξηγούμενης διακύμανσης PCA***

Υπολογίζεται η αθροιστική εξηγούμενη διακύμανση για διαφορετικούς αριθμούς συνιστωσών PCA και την οπτικοποιεί χρησιμοποιώντας ένα γραμμικό διάγραμμα. Στη συνέχεια επιλέγει τον αριθμό των συνιστωσών που εξηγούν τουλάχιστον το 92% της διακύμανσης.

- ***Εφαρμογή PCA***

Εφαρμόζεται η PCA με τον αριθμό συνιστωσών που επιλέχθηκε σε προηγούμενο βήμα.

- ***Εκπαίδευση***

Για τον βέλτιστο ταξινομητή που βρέθηκε στο προηγούμενο σενάριο κώδικα, προσαρμόζεται το καλύτερο μοντέλο στα δεδομένα εκπαίδευσης, για τις παραμέτρους που οδήγησαν σε αυτό μοντέλο. Αξιολογείται η απόδοση του μοντέλου στο σύνολο ελέγχου και εκτυπώνονται η αναφορά ταξινόμησης, η ROC AUC, ο MCC (Matthews Correlation Coefficient) και η PRC AUC (Precision-Recall Curve Area Under the Curve). Αποθηκεύονται οι υπερπαραμέτρου του μοντέλου, η αναφορά ταξινόμησης, η καμπύλη ROC και η καμπύλη PRC σε ξεχωριστά αρχεία.

- ***Περίληψη και αποτελέσματα***

Εμφανίζονται οι αποδόσεις και οι υπερπαράμετροι του μοντέλου και εκτυπώνεται ο χρόνος εκπαίδευσης για τον ταξινομητή, όπου επίσης αποθηκεύεται σε αρχείο όπως και οι πληροφορίες για το μοντέλο σε ξεχωριστά αρχεία.

- ***Κλείσιμο αρχείων***

Τέλος, κλείνουν τα αρχεία που ανοίχτηκαν κατά τη διάρκεια της διαδικασίας και ολοκληρώνεται το πρόγραμμα.

6 Αποτελέσματα - Συμπεράσματα

Σε αυτήν την ενότητα παρουσιάζονται τα σημαντικότερα στοιχεία της ερευνητικής εφαρμογής. Ξεχωριστά, για την ορθότερη κατανόηση των πεπραγμένων, αναφέρονται και μετέπειτα αναλύονται τα αποτελέσματα για κάθε κομμάτι της μεθοδολογίας με τη σειρά. Συγκεκριμένα, στην αρχή παρουσιάζονται τα αποτελέσματα για το αρχικό σύνολο δεδομένων και στη συνέχεια για το επεξεργασμένο. Στη συνέχεια γίνεται αντιπαραβολή, εξαγωγή συμπερασμάτων και σύγκριση των αποτελεσμάτων στα δύο σύνολα. Τέλος, παρουσιάζονται τα αποτελέσματα μετά την εφαρμογή της τεχνικής Ανάλυσης Κύριων Συνιστωσών και επεξηγείται η σημασία τους.

Να σημειωθεί πως όλη η εφαρμογή της μεθοδολογίας με τα αλλεπάλληλα, διαδοχικά πειράματα εφαρμογής των διαφορετικών σεναρίων (τμήματα κώδικα), την καταμέτρηση των χρόνων εκτέλεσης και των αποτελεσμάτων από την εφαρμογή των αλγορίθμων ΜΜ πραγματοποιήθηκαν σε υπολογιστή με λειτουργικό σύστημα Windows 11 στα 64-bit, επεξεργαστή Intel(R) Core(TM) i7-11800H CPU με χρονισμό στα 2.30GHz, εγκατεστημένη RAM στα 32,0 GB και κάρτα γραφικών NVIDIA GeForce RTX 3070 με 8,0 GB αυτόνομη μνήμη.

6.1 Αρχικό Σύνολο Δεδομένων

Αυτό το σενάριο ασχολείται κυρίως με τη διαδικασία αξιολόγησης και σύγκρισης πολλαπλών ταξινομητών μηχανικής μάθησης για ένα πρόβλημα ταξινόμησης. Το πρόβλημα αφορά την ανίχνευση της κατάστασης των ματιών με βάση δεδομένα ΗΕΓ (ηλεκτροεγκεφαλογραφήματος). Να σημειωθεί πως αρχικά, στα πλαίσια της πειραματικής αξιολόγησης εξετάζονται τα δεδομένα στο αρχικό σύνολο δεδομένων. Αυτό γίνεται για να παραχθούν τα αρχικά αποτελέσματα αφενός για να δημιουργηθεί μία πρώτη εικόνα και αφετέρου για να χρησιμοποιηθούν αργότερα ως βάση για σύγκριση με τα αποτελέσματα των μετέπειτα επεξεργασμένων συνόλων δεδομένων. Ακολούθως εξάγονται τα αντίστοιχα συμπεράσματα που παρουσιάζονται στο τέλος του παρόντος κεφαλαίου.

- **Γενική επισκόπηση**

Είναι προφανές ότι οι διάφοροι ταξινομητές έχουν διαφορετικές επιδόσεις στο συγκεκριμένο σύνολο δεδομένων. Δεδομένων των ολοκληρωμένων αποτελεσμάτων που περιλαμβάνουν τις μετρικές, το χρόνο εύρεσης των βέλτιστων παραμέτρων, το χρόνο εκτέλεσης του μοντέλου και τις καλύτερες παραμέτρους, μπορεί να διατυπωθεί μια λεπτομερής ανάλυση για τον προσδιορισμό του καλύτερου ταξινομητή. Οι ταξινομητές Random Forest, MLP και Bagging φαίνεται να επιτυγχάνουν τις καλύτερες συνολικές επιδόσεις, με βάση τις διάφορες μετρικές. Παρακάτω παρουσιάζονται τα παραγόμενα διαγράμματα, οπτικοποιήσεις και συγκριτικοί πίνακες με τα αποτελέσματα.

	Naive Bayes	*Logistic Regression*	*Decision Trees*	*KNN*	*Random Forest*	*MLP*	*SVM*	*AdaBoost*	*Bagging*
Χρόνος 1 (Sec)	0.063	0.55	1.00	73.90	67.69	2151.77	930.52	3.34	28.90
Χρόνος 2 (Sec)	0.01	0.06	0.10	0.04	4.32	4.10	36.00	0.13	2.46
Accuracy	0.468	0.569	0.841	0.829	**0.925**	0.887	0.618	0.837	0.919
Precision	*0.451*	0.570	0.841	0.829	**0.927**	0.887	0.651	0.837	0.920
Recall	*0.468*	0.569	0.841	0.829	**0.925**	0.887	0.618	0.837	0.919
F1-score	*0.315*	0.544	0.841	0.829	**0.925**	0.887	0.581	0.837	0.918
Mcc	*-0.027*	0.125	0.682	0.657	**0.851**	0.774	0.250	0.673	0.838
Roc_auc	*0.500*	0.594	0.840	0.910	**0.982**	0.960	0.709	0.836	0.975
Prc_auc	*0.473*	0.582	0.872	0.902	**0.981**	0.958	0.693	0.868	0.974

Πίνακας 1: Συγκεντρωτικά αποτελέσματα Πειραματικής Αξιολόγησης για Αρχικό Σύνολο Δεδομένων

Από τις παραπάνω τιμές στον **Πίνακα 1** εξάγεται το συμπέρασμα πως ο αλγόριθμος Random Forest υπερτερεί των άλλων ταξινομητών όσον αφορά τις μετρικές της ορθότητας (accuracy), της ακρίβειας (precision), της ανάκλησης (ή ευαισθησίας) (recall), του F1-score, της Roc_auc, της Prc_auc και του συντελεστή Mcc. Ο Χρόνος 1, αναφέρεται στον χρόνο εύρεσης του μοντέλου με τις βέλτιστες παραμέτρους σε δευτερόλεπτα (seconds) μέσα από τη διαδικασία της διασταυρούμενης επικύρωσης, ενώ ο Χρόνος 2 στο χρόνο εκτέλεσης του ταξινομητή στο σύνολο εκπαίδευσης.

- **Ανάλυση με βάση τους χρόνους εύρεσης των βέλτιστων παραμέτρων**

Η ανάλυση αφορά το χρόνο που απαιτείται για την εύρεση των καλύτερων παραμέτρων για κάθε ταξινομητή. Συνοπτικά, ο MLP χρειάζεται τον μεγαλύτερο χρόνο, δεδομένου βέβαια και του ότι εξετάστηκαν περισσότεροι παράμετροι σε σχέση με του υπόλοιπους, καταναλώνοντας 2151,77 δευτερόλεπτα. Ο SVM είναι επίσης χρονοβόρος, καθώς χρειάζεται 930,52 δευτερόλεπτα. Οι Random Forest και KNN χρειάζονται επίσης σημαντικό χρόνο, με 67,69 και 73,90 δευτερόλεπτα, αντίστοιχα. Αντιθέτως, οι Naive Bayes και Logistic Regression είναι μεταξύ των ταχύτερων.

- **Ανάλυση με βάση τους χρόνους εκπαίδευσης των ταξινομητών**

Ο χρόνος εκπαίδευσης είναι ο χρόνος που απαιτείται για την πραγματοποίηση προβλέψεων στο σύνολο ελέγχου. Ο SVM χρειάζεται τον μεγαλύτερο χρόνο με 36,01 δευτερόλεπτα. Οι αλγόριθμοι Random Forest και MLP έπονται με περίπου 4 δευτερόλεπτα ο καθένας. Ο αλγόριθμος Naive Bayes είναι σχεδόν στιγμιαίος, υπερτερεί όλων των άλλων αν και αποτυγχάνει σε όλες τις κύριες μετρικές.

- **Ανάλυση με βάση τις Μετρικές**

Στα παρακάτω τμήματα γίνεται μία σύντομη αναφορά στην σημασία της κάθε μετρικής και στα χαρακτηριστικότερα αποτελέσματα που εξήγαγε η πειραματική αξιολόγηση των αλγορίθμων:

- ***Accuracy***

Δίνει τις συνολικές σωστές προβλέψεις επί του συνόλου των προβλέψεων. Η υψηλότερη ακρίβεια επιτυγχάνεται από τον ταξινομητή Random Forest (92,5%), ακολουθούμενος στενά από τον ταξινομητή Bagging (91,9%). Ο ταξινομητής Naive Bayes έχει τη χειρότερη επίδοση από αυτή την άποψη (46,8%).

- ***Precision***

Είναι ο λόγος των σωστά προβλεπόμενων θετικών παρατηρήσεων προς το σύνολο των προβλεπόμενων θετικών παρατηρήσεων. Η υψηλότερη ακρίβεια παρατηρείται στον ταξινομητή Random Forest (92,7%).

- ***Recall***

Δείχνει πόσες από τις πραγματικές θετικές περιπτώσεις μπορέσαμε να προβλέψουμε σωστά. Ο ταξινομητής Random Forest βρίσκεται και εδώ στην κορυφή με 92,5%.

- ***F1-Score***

Το F1-Score είναι ο σταθμισμένος μέσος όρος των Precision και Recall. Προσπαθεί να βρει την ισορροπία μεταξύ ακρίβειας και ανάκλησης. Και πάλι, ο ταξινομητής Random Forest κυριαρχεί με F1-Score 92,5%.

- ***MCC (Συντελεστής συσχέτισης Matthews)***

Αυτή η μετρική παρέχει ένα ισορροπημένο μέτρο, ακόμη και όταν οι κλάσεις έχουν πολύ διαφορετικά μεγέθη. Κυμαίνεται από -1 (απόλυτη διαφωνία) έως 1 (απόλυτη συμφωνία), με το 0 να υποδηλώνει ότι δεν είναι καλύτερο από την τυχαία πρόβλεψη. Ο Random Forest προηγείται με 85,1%, που είναι μια πολύ καλή βαθμολογία.

- ***ROC AUC***

Αυτή η μετρική παρέχει ένα συνολικό μέτρο της απόδοσης σε όλα τα πιθανά κατώφλια ταξινόμησης. Ένα AUC 1,0 υποδηλώνει τέλεια ταξινόμηση, ενώ ένα AUC 0,5 υποδηλώνει ένα μοντέλο που δεν είναι καλύτερο από την τυχαία εικασία. Ο ταξινομητής Random Forest υπερέχει με πολύ υψηλό ROC AUC 98,2%.

- ***PRC AUC***

Αυτή η μετρική παρέχει ένα συνολικό μέτρο της ακρίβειας ενός μοντέλου σε διαφορετικά κατώτατα όρια. Ο Random Forest βρίσκεται και εδώ στην κορυφή με βαθμολογία 98,1%.

Κατάταξη με βάση τους Ταξινομητές

Σε αυτήν την υποενότητα παρουσιάζεται η κατάταξη με βάση τους Ταξινομητές. Εξετάζεται η απόδοση του κάθε αλγορίθμου και μετέπειτα σχολιάζονται οι τιμές των μετρικών που εξήχθησαν.

- ***Naive Bayes***

Αρκετά κακές επιδόσεις σε όλες τις μετρήσεις. Η MCC του είναι ακόμη και αρνητική, υποδηλώνοντας ότι η απόδοσή του είναι χειρότερη από την τυχαία πρόβλεψη.

- ***Λογιστική παλινδρόμηση***

Μέτρια απόδοση. Όχι η καλύτερη, αλλά σίγουρα όχι και η χειρότερη σε σχέση με τα αποτελέσματα που παρουσιάστηκαν παραπάνω.

- ***Δέντρο αποφάσεων***

Σημειώνει αξιοσημείωτες επιδόσεις. Είναι ιδιαίτερα άξιο αναφοράς πόσο κοντά είναι οι βαθμολογίες του σε όλες τις μετρήσεις, γεγονός που υποδηλώνει ισορροπημένη απόδοση.

- ***KNN***

Έχει αρκετά καλές επιδόσεις, ιδίως αν λάβουμε υπόψη τις βαθμολογίες ROC AUC και PRC AUC. Ωστόσο, οι πραγματικές μετρικές ταξινόμησής του (όπως η ακρίβεια και το F1-score) είναι ελαφρώς χαμηλότερες από το Decision Tree.

- ***Random Forest***

Σαφώς ο καταλληλότερος αυτής της κατηγορίας. Δίνει τις υψηλότερες βαθμολογίες σε όλες σχεδόν τις μετρικές.

- ***MLP***

Άλλη μια ισχυρή επίδοση, ιδίως αν λάβουμε υπόψη τις τιμές ROC AUC και PRC AUC.

- ***SVM***

Οι επιδόσεις του είναι κάπως μέτριες, τείνοντας προς το χαμηλότερο άκρο του εύρους.

- ***AdaBoost***

Παρόμοια με το Δέντρο Αποφάσεων σε απόδοση, αλλά με ελαφρώς χαμηλότερη ROC AUC.

- ***Bagging***

Μία από τις κορυφαίες επιδόσεις, με μικρή μόνο διαφορά πίσω από τον Random Forest.

6.2 Συμπεράσματα

Εάν η απόδοση είναι η απόλυτη προτεραιότητα και υπάρχει ο αντίστοιχος υπολογιστικός χρόνος, ο ταξινομητής Random Forest είναι η καλύτερη επιλογή. Από την άλλη, εάν υπάρχει ανάγκη να γίνονται συχνές εκπαιδεύσεις ή εκπαίδευση σε πραγματικό χρόνο, θα πρέπει να ληφθεί υπόψη ο χρόνος εκπαίδευσης του μοντέλου. Εδώ, ενώ ο Random Forest είναι κορυφαίος ταξινομητής, ο χρόνος πρόβλεψής του δεν είναι ο ταχύτερος. Ο Decision Tree ή ο KNN μπορεί να επιτύχουν καλύτερη ισορροπία μεταξύ ταχύτητας και απόδοσης σε τέτοιες περιπτώσεις. Τέλος, εάν ο στόχος είναι η γρήγορη δοκιμή μοντέλων και η επανάληψη, ο χρόνος ρύθμισης των υπερπαραμέτρων καθίσταται σημαντικός. Ο MLP και ο SVM μπορεί να μην είναι οι καλύτερες επιλογές, λόγω των πολύ χρονοβόρων συντονισμών τους στην εύρεση των κατάλληλων παραμέτρων και στην εκτέλεση των αλγορίθμων. Παρακάτω παρουσιάζονται τα συμπεράσματα για τις επιμέρους κατηγορίες.

- **Για υψηλότερη συνολική απόδοση**

Αν η προτεραιότητα είναι να επιτευχθούν τα καλύτερα αποτελέσματα σε όλες τις μετρικές, ο Random Forest θα πρέπει να είναι η κύρια επιλογή, με τον Bagging και τον MLP να αποτελούν εξαιρετικές εναλλακτικές λύσεις.

- **Αποτελεσματικότητα χρόνου και ταχύτητα**

Εάν η πολυπλοκότητα του μοντέλου ή η ταχύτητα εκπαίδευσης/πρόβλεψης αποτελεί ανησυχία, θα μπορούσαν να εξεταστούν απλούστερα μοντέλα όπως τα δέντρα απόφασης, ιδίως δεδομένης της ικανοποιητικής απόδοσής τους. Αν και ο Random Forest αποδίδει εξαιρετικά καλά, χρειάζεται αρκετό χρόνο για τον συντονισμό των υπερπαραμέτρων και την εκτέλεση του μοντέλου. Αντίθετα, τα απλούστερα μοντέλα όπως τα δέντρα απόφασης (Decision Trees) και η Λογιστική Παλινδρόμηση (Logistic Regression) είναι πολύ ταχύτερα. Οι MLP και SVM, από την άλλη πλευρά, είναι αρκετά χρονοβόροι.

- **Για ισορροπία**

Εάν υπάρχει ανάγκη για ισορροπημένη απόδοση μεταξύ ακρίβειας και ανάκλησης, θα πρέπει να δοθεί προτεραιότητα σε μοντέλα με υψηλές βαθμολογίες F1 (όπως οι Random Forest, MLP και Bagging).

- **Πολυπλοκότητα**

Οι βέλτιστες παράμετροι που βρέθηκαν ανά μοντέλο αποκαλύπτουν ότι ορισμένα μοντέλα όπως το MLP και το Random Forest απαιτούν πιο πολύπλοκες διαμορφώσεις για να επιτύχουν τα καλύτερα αποτελέσματά τους, ενώ άλλα όπως το Naive Bayes και η Logistic Regression είναι πιο απλά.

6.2.1 Αποτελέσματα Βελτιστοποίησης

Τα αποτελέσματα από τη διαδικασία της εύρεσης των βέλτιστων παραμέτρων στο επεξεργασμένο σύνολο δεδομένων (ΠΑΡΑΡΤΗΜΑ 9.3) παρουσιάζονται αναλυτικά παρακάτω. Μαζί εμπεριέχεται και μία επεξήγηση για τις παραμέτρους που επικράτησαν και τις αναδειχθείσες τιμές που προέκυψαν.

- ***Naive Bayes***

Για τον Naive Bayes δεν δόθηκαν ούτε ρυθμίστηκαν υπερπαράμετροι, καθώς αυτός ο ταξινομητής δεν διαθέτει υπερπαραμέτρους για βελτιστοποίηση.

- ***Λογιστική παλινδρόμηση***

C: Αυτή η υπερπαράμετρος ρυθμίζει το αντίστροφο της ισχύος της κανονικοποίησης. Μια μικρότερη τιμή του C υποδηλώνει ισχυρότερη κανονικοποίηση. Στην προκειμένη περίπτωση, η βέλτιστη τιμή που βρέθηκε για το C 0,1, γεγονός που υποδηλώνει κάποιο επίπεδο κανονικοποίησης.

Penalty: Αυτή καθορίζει τη ρύθμιση που χρησιμοποιείται στην εφαρμογή ποινών. Η ποινή 'l2' είναι η κανονικοποίηση Ridge, η οποία τείνει να ενθαρρύνει μικρά βάρη για όλα τα χαρακτηριστικά.

- ***Δέντρο απόφασης***

Max Depth: Αυτή η υπερπαράμετρος αντιπροσωπεύει το μέγιστο βάθος του δέντρου. Όταν έχει οριστεί σε Καμία (None), σημαίνει ότι οι κόμβοι επεκτείνονται έως ότου όλα τα φύλλα είναι καθαρά (χωρίς άλλο βάθος) ή έως ότου όλα τα φύλλα περιέχουν λιγότερα από τα ελάχιστα δείγματα που απαιτούνται για τη διάσπαση ενός επιπλέον κλαδιού του δέντρου. Ουσιαστικά, με αυτήν τη τιμή που βρέθηκε, η βέλτιστη απόδοση επιτυγχάνεται όταν δεν υπάρχει κανένας καθορισμένος περιορισμός βάθους.

- ***K-Nearest Neighbors (KNN)***

N_neighbors: Αυτό καθορίζει τον αριθμό των γειτόνων που θα χρησιμοποιηθούν για το ερώτημα kneighbors. Βρέθηκε η τιμή 1, πράγμα που σημαίνει ότι για κάθε δεδομένο παράδειγμα δοκιμής, ο αλγόριθμος θα αναζητήσει το μοναδικό πλησιέστερο παράδειγμα εκπαίδευσης.

Weights: Η υπερπαράμετρος weights μπορεί να οριστεί σε 'uniform' (όλα τα σημεία σε κάθε γειτονιά σταθμίζονται εξίσου) ή 'distance' (σταθμίζονται τα σημεία με το αντίστροφο της απόστασής τους). Η βέλτιστη τιμή που βρέθηκε είναι η 'uniform'.

- ***Random Forest***

Max Features: Η βελτιστοποίηση σε αυτήν την παράμετρο ορίζει τον αριθμό των χαρακτηριστικών που θα λαμβάνονται υπόψη κατά την αναζήτηση του καλύτερου διαχωρισμού. Η επιλογή 'sqrt' σημαίνει ότι χρησιμοποιείται η τετραγωνική ρίζα του συνολικού αριθμού των χαρακτηριστικών.

N_estimators: Η τιμή αυτή καθορίζει τον αριθμό των δέντρων στο «δάσος». Τα περισσότερα δέντρα οδηγούν συνήθως σε καλύτερη απόδοση αλλά και αναλόγως σε μεγαλύτερο υπολογιστικό κόστος. Η διαδικασία της βελτιστοποίησης βρήκε τα 200 δέντρα ως τη βέλτιστη τιμή της παραμέτρου N_estimators, όπου ήταν και η μεγαλύτερη που εξετάστηκε.

- ***Multi-layer Perceptron (MLP)***

Activation: Η συνάρτηση ενεργοποίησης για το κρυφό στρώμα. Επιλέχθηκε η "tanh" ή υπερβολική εφαπτομένη, η οποία εξάγει τιμές μεταξύ -1 και 1.

Alpha: Πρόκειται για την παράμετρο της ποινής L2 (όρος κανονικοποίησης). Διαμορφώθηκε σε 0,0001.

Hidden Layer Sizes: Αυτή η παράμετρος καθορίζει τον αριθμό των νευρώνων στα κρυφά στρώματα. Ως βέλτιστα επιλέχθηκαν τα δύο κρυφά στρώματα, το καθένα με 50 νευρώνες.

Learning Rate Init: Ο αρχικός ρυθμός μάθησης που χρησιμοποιείται. Ελέγχει το μέγεθος βήματος στην ενημέρωση των βαρών. Η τιμή που βρέθηκε είναι 0,01.

Momentum: Αυτή η παράμετρος βοηθά στην επιτάχυνση της σύγκλισης και η τιμή που βρέθηκε ως βέλτιστη είναι η 0,5.

- ***Support Vector Machine (SVM)***

C: Όπως και στη λογιστική παλινδρόμηση, το C ρυθμίζει το αντίστροφο της ισχύος της κανονικοποίησης και η βέλτιστη τιμή βρέθηκε να είναι 1.

Kernel: Η συνάρτηση πυρήνα που βρέθηκε ως βέλτιστη για τον αλγόριθμο SVM είναι η "rbf".

- ***Ensemble - AdaBoost***

Learning Rate:: Αυτή η παράμετρος συρρικνώνει τη συνεισφορά κάθε ταξινομητή και η βέλτιστη τιμή συρρίκνωσης ορίστηκε σε 0,1.

N estimators: Αφορά τον μέγιστο αριθμό εκτιμητών (δηλ. ασθενών ταξινομητών) στον οποίο τερματίζεται η ενίσχυση και βρέθηκε να είναι ο αριθμός 50.

- ***Bagging***

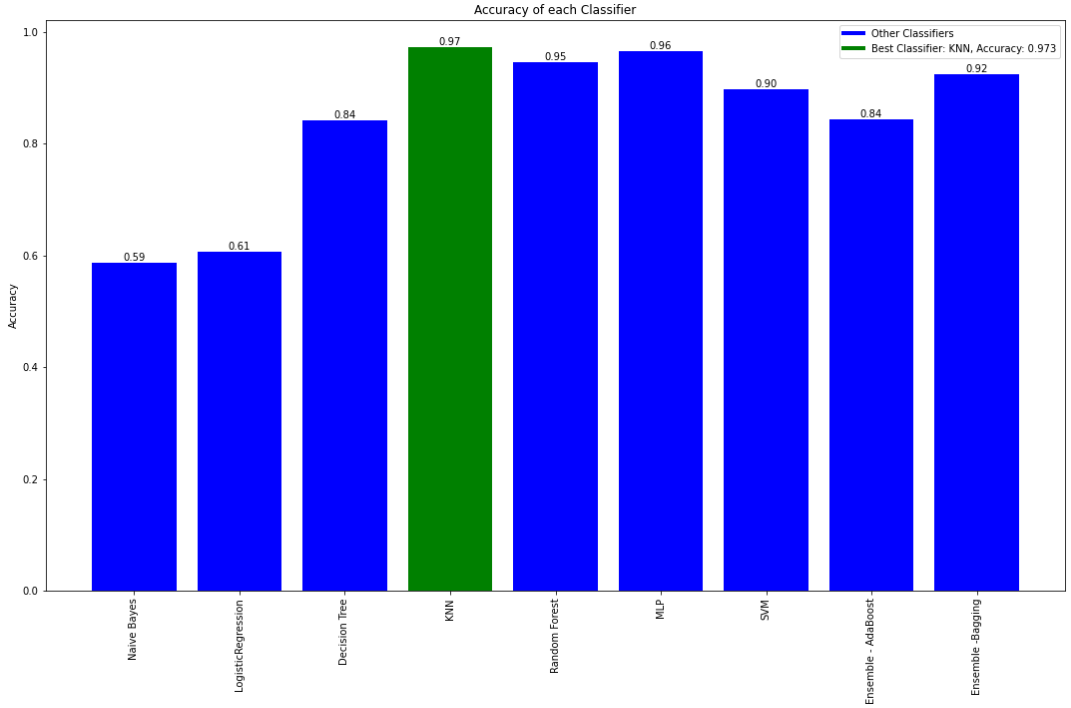
N estimators: Καθορίζει τον αριθμό των βασικών εκτιμητών στο σύνολο. Ως εκτιμητές βάσης επιλέχθηκαν οι 30 ως βέλτιστος αριθμός.

6.3 Επεξεργασμένο Σύνολο Δεδομένων

Σε αυτήν την ενότητα παρουσιάζονται τα πιο σημαντικά αποτελέσματα, σύμφωνα με το 1ο ερευνητικό ερώτημα, που αφορά τη βελτιστοποίηση των αποδόσεων της ταξινόμησης για την πρόβλεψη της κατάστασης των οφθαλμών των ασθενών. Αφορά το επεξεργασμένο σύνολο δεδομένων μετά την προεπεξεργασία και την εφαρμογή των βέλτιστων παραμέτρων, που οδήγησε όπως βλέπουμε στον **Πίνακα 2**, στην βελτίωση των μετρικών σε όλους τους ταξινομητές. Παράλληλα, βλέπουμε στο **Σχήμα 3** πως ο KNN σημείωσε τη μεγαλύτερη πρόοδο βάσει ορθότητας με αποτέλεσμα να βρίσκεται αποτελεί την καλύτερη επιλογή.

	Naive Bayes	*Logistic Regression*	*Decision Trees*	*KNN*	*Random Forest*	*MLP*	*SVM*	*AdaBoost*	*Bagging*
Χρόνος 1 (Sec)	0.05	0.49	1.23	111.87	115.61	2188.83	1049.31	4.39	37.90
Χρόνος 2 (Sec)	0.010	0.03	0.15	0.04	5.42	3.39	23.90	0.16	2.99
Accuracy	0.587	0.608	0.842	**0.973**	0.946	0.965	0.897	0.844	0.925
Precision	*0.606*	0.608	0.842	**0.973**	0.946	0.965	0.897	0.844	0.925
Recall	*0.587*	0.608	0.842	**0.973**	0.946	0.965	0.897	0.844	0.925
F1-score	*0.569*	0.607	0.842	**0.973**	0.946	0.965	0.897	0.844	0.925
Mcc	*0.193*	0.215	0.684	**0.946**	0.892	0.930	0.794	0.689	0.850
Roc_auc	*0.649*	0.658	0.842	0.973	0.989	**0.995**	0.962	0.844	0.978
Prc_auc	*0.644*	0.658	0.882	0.979	0.990	**0.995**	0.963	0.883	0.980

Πίνακας 2: Συγκεντρωτικά αποτελέσματα Πειραματικής Αξιολόγησης για Επεξεργασμένο Σύνολο Δεδομένων



Σχήμα 3: Συγκεντρωτικό Bar Chart αποδόσεων Ορθότητας Ταξινομητών στο Επεξεργασμένο Σύνολο Δεδομένων

Ανάλυση σε σχέση με το Αρχικό σύνολο Δεδομένων

Μέσα από την προεπεξεργασία των δεδομένων και τη βελτιστοποίηση των παραμέτρων για κάθε ταξινομητή, η διαδικασία επανεκτελείται, αυτή τη φορά με ορίσματα τις υπερπαραμέτρους που βρέθηκαν, δηλαδή χωρίς τη χρήση πλέγματος για να την αναζήτηση βέλτιστων υπερπαραμέτρων. Οι ταξινομητές αξιολογούνται με βάση τις τιμές των μετρικών που εξάγονται και αναλύονται τα τελικά αποτελέσματα της ταξινόμησης. Με βάση αυτά τα αποτελέσματα, συνοπτικά, βλέπουμε πολύ καλύτερες αποδόσεις στο προεπεξεργασμένο σύνολο δεδομένων με χαρακτηριστικότερα αυτά του ταξινομητή K-Nearest Neighbors (KNN) ο οποίος επιτυγχάνει τιμές ορθότητας 0,973 με πολύ χαμηλό χρόνο εκτέλεσης 0,04 δευτερόλεπτα. Αναλόγως μεγάλες αποδόσεις πέτυχαν και οι αλγόριθμοι Random Forest και Multi-Layer Perceptron (MLP) με τιμές ορθότητας 0,946 και 0,965 αντίστοιχα, όμως με σημαντικά υψηλότερους χρόνους εκτελέσεις σε σχέση με τον KNN. Ο Random Forest και ο MLP είχαν χρόνους εκτέλεσης 5.49 και 3.39 δευτερόλεπτα αντίστοιχα, τουλάχιστον 300% πιο αργοί σε σχέση με το 0,04 του επικρατέστερου KNN.

Στις αποδόσεις των ταξινομητών στο προεπεξεργασμένο σύνολο δεδομένων συγκριτικά με αυτές στο αρχικό, βλέπουμε πως υπάρχει μεγάλη βελτίωση, καθώς η βέλτιστη τιμή ορθότητας στο αρχικό σύνολο δεδομένων φτάνει το 0.925 με τον Random Forest να υπολείπεται αρκετά σε σχέση με τον KNN που είναι ορθότερος και γρηγορότερος σε προεπεξεργασμένο σύνολο.

6.3.1 Ανάλυση για κάθε ταξινομητή

Αναλυτικότερα, παρακάτω παρουσιάζονται, αρχικά συνοπτικά και μετέπειτα πιο αναλυτικά, τα τελικά αποτελέσματα στο επεξεργασμένο σύνολο δεδομένων μετά την εφαρμογή των βέλτιστων παραμέτρων. Αρχικά παρουσιάζονται συγκεντρωτικά τα αποτελέσματα για τους ταξινομητές που εφαρμόστηκαν και μετέπειτα γίνεται αναφορά στις τιμές των επιμέρους μετρικών.

Συνοπτική Παρουσίαση

Οι ταξινομητές Naive Bayes και Logistic Regression έχουν τις χαμηλότερες τιμές ορθότητας, ακρίβειας, ανάκλησης, F1-Score και περιοχών κάτω από την καμπύλη (AUC) ROC και PRC μεταξύ όλων των ταξινομητών, υποδεικνύοντας ότι δεν είναι οι καλύτεροι για τα δεδομένα ΗΕΓ.

Οι Decision Tree και AdaBoost έχουν παρόμοιες επιδόσεις, με ακρίβεια και άλλες μετρικές γύρω στο 0,84 - 0,85.

Οι K-Nearest Neighbors (KNN), Random Forest και Multi-Layer Perceptron (MLP) είχαν τις καλύτερες επιδόσεις, με accuracy, precision, recall και F1-Score όλα πάνω από 0,94. Ειδικότερα, ο KNN έχει την υψηλότερη ορθότητα (0,973), ακρίβεια (0,973), ανάκληση (0,973) και F1-Score (0,973) μεταξύ όλων των ταξινομητών.

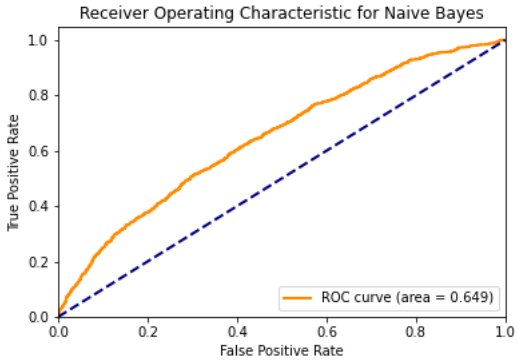
Ο SVM και Ensemble - Bagging είχαν επίσης καλές επιδόσεις, με ακρίβεια και άλλες μετρικές πάνω από 0,89.

Συμπερασματικά, οι KNN, Random Forest και MLP φαίνεται να είναι οι καλύτεροι ταξινομητές για τα δεδομένα ΗΕΓ, ακολουθούμενοι από τους SVM και Ensemble - Bagging. Οι Naive Bayes και Logistic Regression είχαν τις χειρότερες επιδόσεις και ενδέχεται να μην είναι κατάλληλοι για αυτό τον τύπο δεδομένων.

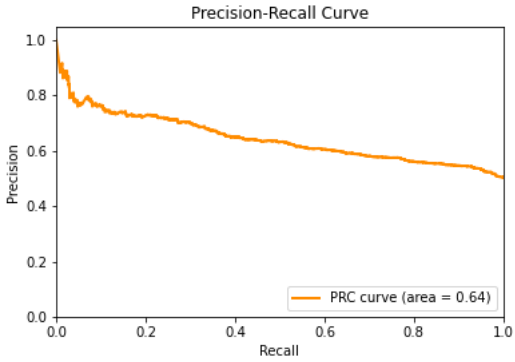
Αναλυτική Παρουσίαση

Naive Bayes

- *Ορθότητα* 0.587 σημαίνει ότι το 58,7% των συνολικών προβλέψεων του μοντέλου είναι σωστές.
- *Ακρίβεια* : 0,606 σημαίνει ότι από το σύνολο των θετικών προβλέψεων του μοντέλου, το 60,6% είναι πράγματι θετικές.
- *Ανάκληση*: 0,587 σημαίνει ότι από όλες τις πραγματικές θετικές περιπτώσεις, το μοντέλο μπόρεσε να προβλέψει σωστά το 58,7% αυτών.
- *F1-score*: 0,569 είναι ο αρμονικός μέσος όρος της ακρίβειας και της ανάκλησης και είναι ένας καλός τρόπος για να συνοψίσουμε την απόδοση του μοντέλου σε μια ενιαία τιμή.
- *ROC_AUC*: 0,649 είναι το εμβαδόν κάτω από την καμπύλη ROC όπως βλέπουμε στο **Σχήμα 4**, η οποία συνοψίζει την ικανότητα του μοντέλου να διακρίνει μεταξύ θετικών και αρνητικών κλάσεων.
- *PRC_AUC*: 0,644 είναι η περιοχή κάτω από την καμπύλη ακρίβειας-ανάκλησης όπως βλέπουμε στο **Σχήμα 5**, η οποία συνοψίζει την ικανότητα του μοντέλου να προβλέπει σωστά τη θετική κλάση.
- *MCC*: 0,193 είναι ο συντελεστής συσχέτισης Matthews, ο οποίος αποτελεί μέτρο της ποιότητας των δυαδικών ταξινομήσεων. Η τιμή 0.193 είναι και αυτή πολύ χαμηλή.



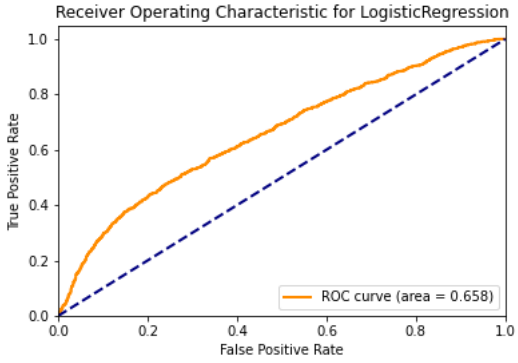
Σχήμα 4: ROC Curve Naive Bayes



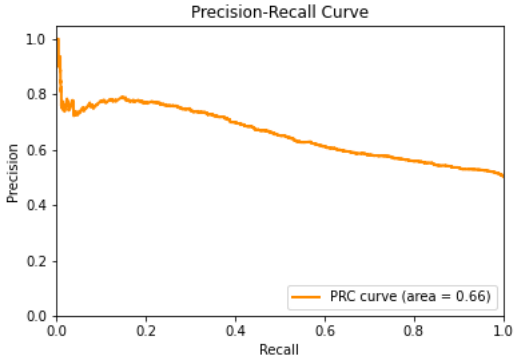
Σχήμα 5: Precision - Recall Curve Naive Bayes

Λογιστική παλινδρόμηση

Η Λογιστική Παλινδρόμηση έδωσε παρόμοια ποσοστά με τον Naive Bayes, αλλά με ελαφρώς καλύτερες επιδόσεις σε όλες τις μετρήσεις. Είναι ένας άλλος βασικός αλγόριθμος ταξινόμησης που φαίνεται να μην αποδίδει πολύ καλά στα δεδομένα. Στο **Σχήμα 6** και **Σχήμα 7** μπορούμε να διακρίνουμε την καμπύλη ROC και την καμπύλη PRC αντίστοιχα.



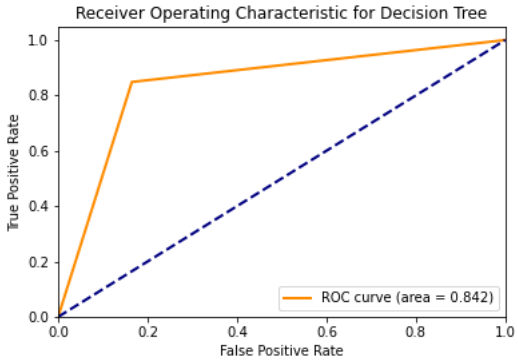
Σχήμα 6: ROC Curve Λογιστική Παλινδρόμηση



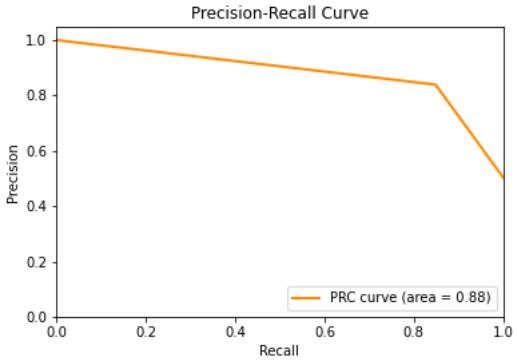
Σχήμα 7: Precision - Recall Curve Λογιστική Παλινδρόμηση

Δέντρα Απόφασης

Η ορθότητα, η ακρίβεια, η ανάκληση και το F1-Score είναι όλα 0,842, που σημαίνει ότι το μοντέλο ταξινόμησε σωστά το 84,2% των περιπτώσεων και από όλες τις θετικές προβλέψεις που έκανε το μοντέλο, το 84,2% είναι πραγματικά θετικές και το μοντέλο μπόρεσε να προβλέψει σωστά το 84,2% των πραγματικών θετικών περιπτώσεων. Η PRC_AUC είναι 0,882, που είναι αρκετά καλή, και η MCC είναι 0,684, που είναι επίσης σχετικά καλή. Στο **Σχήμα 8** και **Σχήμα 9** μπορούμε να διακρίνουμε την καμπύλη ROC και την καμπύλη PRC αντίστοιχα.



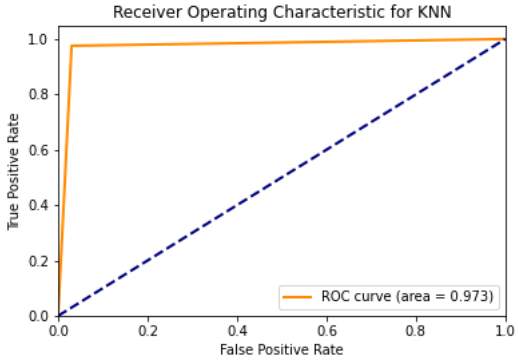
Σχήμα 8: ROC Curve Δέντρα Απόφασης



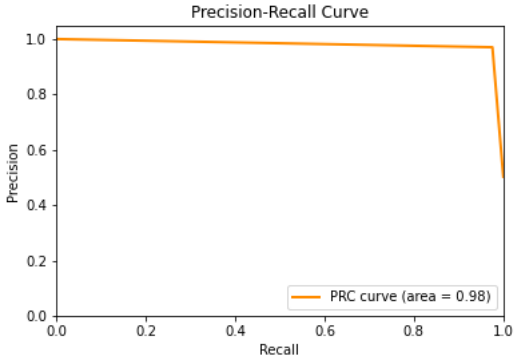
Σχήμα 9: Precision - Recall Curve Δέντρα Απόφασης

K-Nearest Neighbors (KNN)

Αυτό το μοντέλο είχε την καλύτερη απόδοση μεταξύ όλων των ταξινομητών με ακρίβεια, ακρίβεια, ανάκληση και f1-score που είναι όλα 0,973. Τα ROC_AUC και PRC_AUC είναι επίσης πολύ υψηλά, υποδεικνύοντας εξαιρετική απόδοση του μοντέλου. Το MCC 0,946 είναι επίσης εξαιρετικό και υποδεικνύει δυαδικές ταξινομήσεις πολύ υψηλής ποιότητας. Στο **Σχήμα 10** και **Σχήμα 11** μπορούμε να διακρίνουμε την καμπύλη ROC και την καμπύλη PRC αντίστοιχα, με τιμές 0,973 και 0,979.



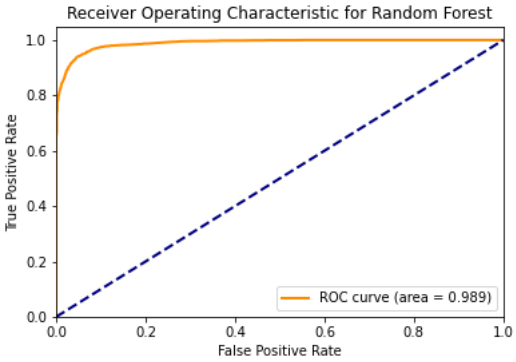
Σχήμα 10: ROC Curve KNN



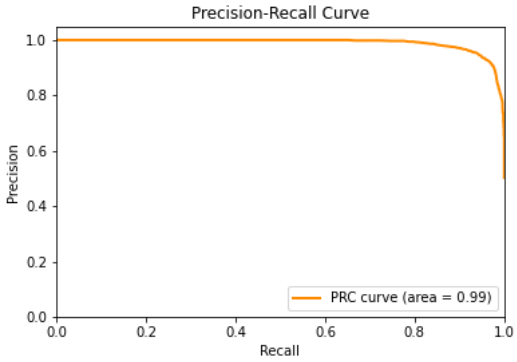
Σχήμα 11: Precision - Recall Curve KNN

Random Forest

Αυτό το μοντέλο είχε επίσης πολύ καλές επιδόσεις με όλες τις μετρικές πάνω από 0,94. Η ROC_AUC του 0,989 και η PRC_AUC του 0,990, όπως βλέπουμε στο **Σχήμα 13** και **Σχήμα 12** αντίστοιχα, είναι εξαιρετικές και υποδεικνύουν ότι το μοντέλο έχει εξαιρετική ικανότητα διάκρισης μεταξύ των θετικών και των αρνητικών κλάσεων.



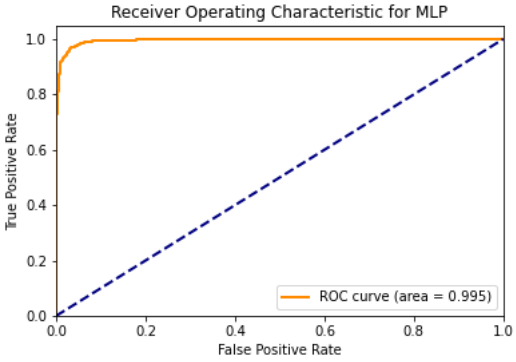
Σχήμα 12: ROC Curve Random Forest



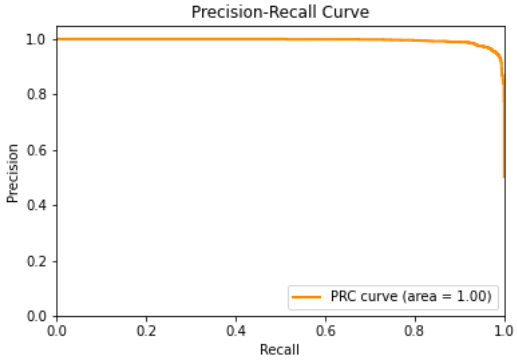
Σχήμα 13: Precision - Recall Curve Random Forest

Multi-layer Perceptron (MLP)

Το νευρωνικό δίκτυο είχε πολύ καλές επιδόσεις με όλες τις μετρικές να είναι 0,965. Οι τιμές ROC_AUC και PRC_AUC 0,995 είναι εξαιρετικές όπως διακρίνουμε επίσης στο **Σχήμα 14** και **Σχήμα 15** αντίστοιχα.



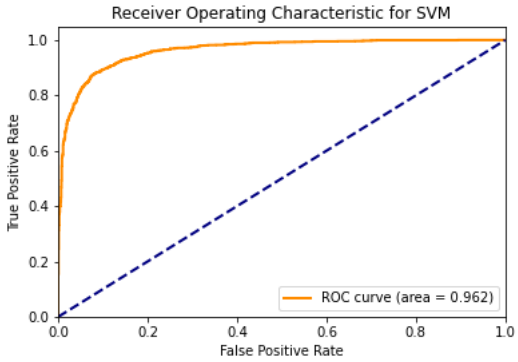
Σχήμα 14: ROC Curve MLP



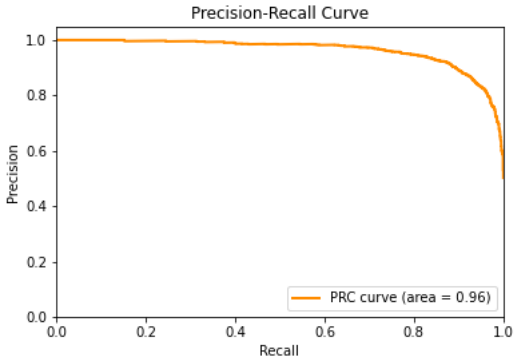
Σχήμα 15: Precision - Recall Curve MLP

Support Vector Machine (SVM)

Αυτό το μοντέλο απέδωσε αρκετά καλά με όλες τις μετρικές να είναι 0,897. Η ROC_AUC του 0,962 και η PRC_AUC του 0,963 είναι πολύ καλές όπως βλέπουμε στο **Σχήμα 16** και **Σχήμα 17** αντίστοιχα.



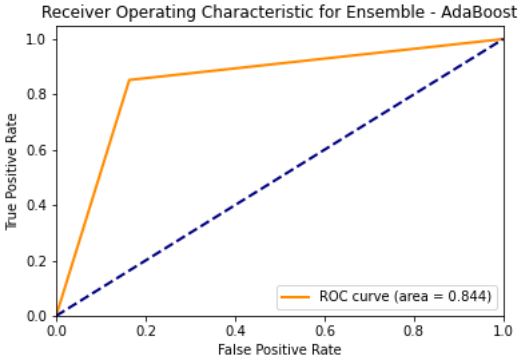
Σχήμα 16: ROC Curve SVM



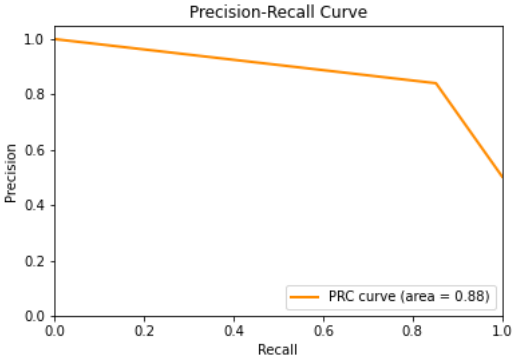
Σχήμα 17: Precision - Recall Curve SVM

AdaBoost

Οι επιδόσεις αυτού του μοντέλου είναι παρόμοιες με του μοντέλου Decision Tree, με όλες τις μετρικές να είναι γύρω στο 0,84 - 0,85, αλλά με καλύτερες τιμές στην καμπύλη ROC (0,844) και PRC (0,883) όπως βλέπουμε στο **Σχήμα 18** και **Σχήμα 19** αντίστοιχα.



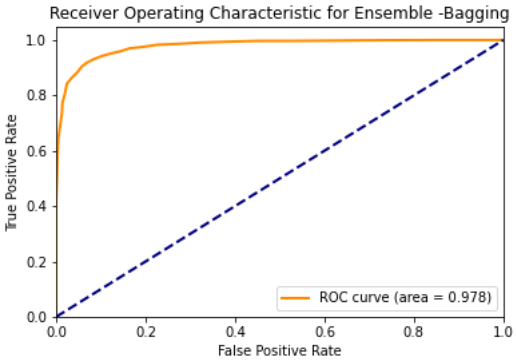
Σχήμα 18: ROC Curve Εικόνα AdaBoost



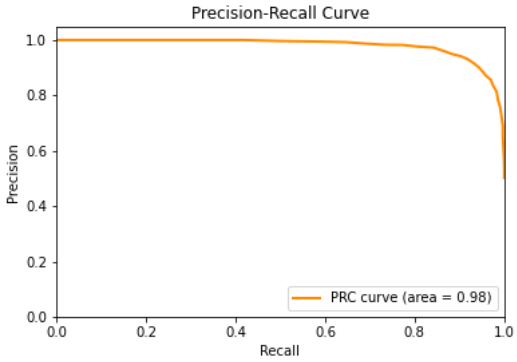
Σχήμα 19: Precision - Recall Curve Adaboost

Bagging

Αυτό το μοντέλο είχε πολύ καλή απόδοση με όλες τις μετρικές να είναι 0,925. Η ROC_AUC του 0,978 και η PRC_AUC του 0,980 είναι εξαιρετικές, όπως φαίνονται στο **Σχήμα 20** και **Σχήμα 21** αντίστοιχα.



Σχήμα 20: ROC Curve Bagging

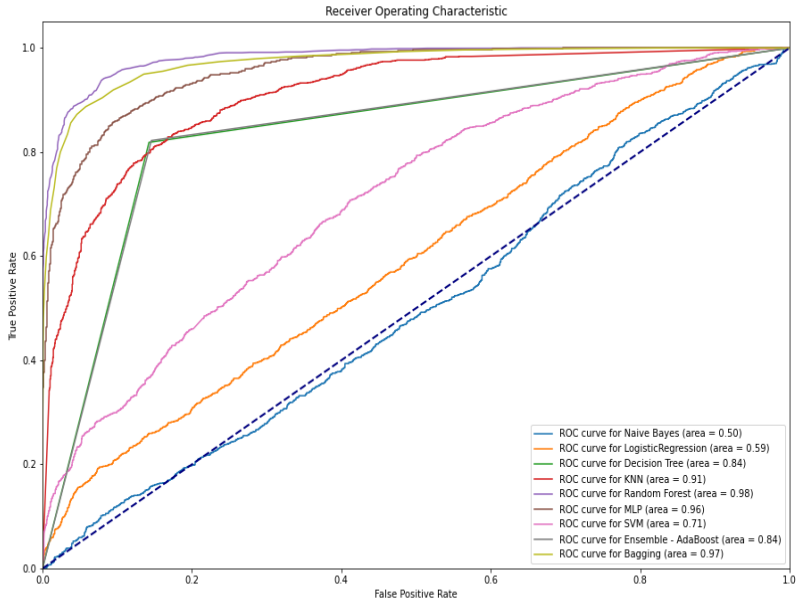


Σχήμα 21: Precision - Recall Curve Bagging

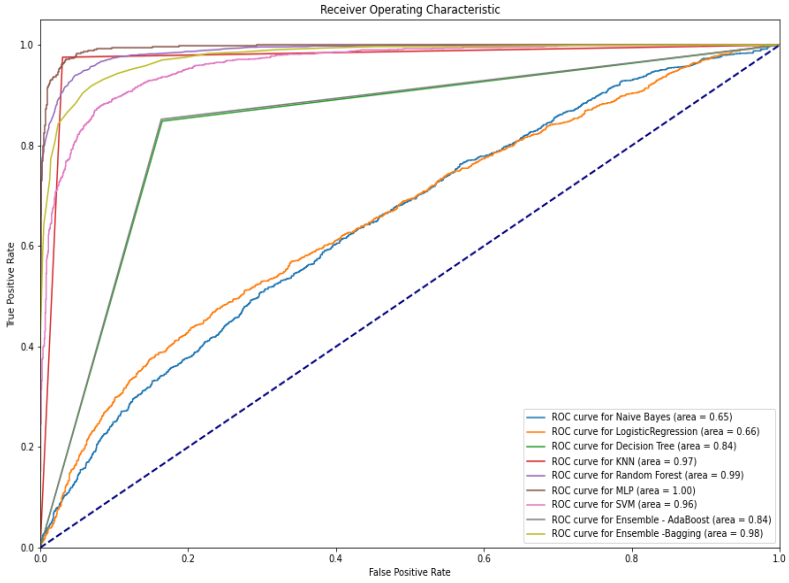
Συμπερασματικά, οι KNN, Random Forest και MLP είναι οι κορυφαίοι στο σύνολο δεδομένων ΗΕΓ, παρουσιάζοντας υψηλή ορθότητα, ακρίβεια, ανάκληση και F1-score. Αυτά τα μοντέλα παρουσιάζουν επίσης υψηλές τιμές AUC τόσο για τις καμπύλες ROC όσο και για τις καμπύλες precision-recall, υποδεικνύοντας ισχυρή ικανότητα διάκρισης μεταξύ κλάσεων και σωστής πρόβλεψης της θετικής κλάσης. Από την άλλη πλευρά, τα μοντέλα Naive Bayes και Logistic Regression έχουν τις χαμηλότερες επιδόσεις, παρουσιάζοντας τις χαμηλότερες τιμές σε όλες τις μετρικές. Άλλα μοντέλα όπως τα SVM, AdaBoost και Bagging παρουσιάζουν αρκετά καλές επιδόσεις, αλλά υπολείπονται των KNN, Random Forest και MLP. Ως επικρατέστερος ταξινομητής κατατάσσεται ο KNN με τις υψηλότερες αποδόσεις συνολικά.

6.3.2 Συγκριτικη Παρουσίαση Αποδόσεων

Στα παρακάτω διαγράμματα παρουσιάζονται σε συγκεντρωτικά γραφήματα οι αποδόσεις των μοντέλων μέσα από τις καμπύλες ROC. Η αντιπαραβολή των διαγραμμάτων γίνεται τόσο για το αρχικό σύνολο δεδομένων, όσο και για το επεξεργασμένο σύνολο δεδομένων. Έτσι λοιπόν, στο **Σχήμα 22** βλέπουμε συνολικά που κυμαίνονται οι καμπύλες ROC για κάθε ταξινομητή στο αρχικό σύνολο δεδομένων, ενώ αντίστοιχα στο **Σχήμα 23** παρατηρούμε πως οι καμπύλες αυτές μεταβλήθηκαν, σχεδόν για όλους τους ταξινομητές προς το καλύτερο, όταν το σύνολο υπέστη κάποιους είδους προεπεξεργασία.



Σχήμα 22: Συγκεντρωτικό -Συγκριτικό Σχήμα για ROC Curves στο Αρχικό Σύνολο Δεδομένων



Σχήμα 23: Συγκεντρωτικό -Συγκριτικό Σχήμα για ROC Curves στο Επεξεργασμένο Σύνολο Δεδομένων

Συγκριτικός Σχολιασμός Συγκεντρωτικών Γραφημάτων

Αναλύοντας τις επιδόσεις των διαφόρων ταξινομητών και παρατηρώντας τις συγκριτικές καμπύλες ROC, τόσο στο αρχικό όσο και στο προεπεξεργασμένο σύνολο δεδομένων, μπορούν να γίνουν ορισμένες αρχικές παρατηρήσεις. Ξεκινώντας με τον Naive Bayes, κατέγραψε βαθμολογία ROC AUC 0,500 στο αρχικό σύνολο δεδομένων, υποδεικνύοντας ότι οι προβλέψεις του αντανακλούσαν τυχαιότητα. Ωστόσο, στα προεπεξεργασμένα δεδομένα, υπήρξε μια αισθητή άνοδος της απόδοσής του, αν και η μετρική μετατοπίστηκε στην ακρίβεια, καταγράφοντας 0,587. Η λογιστική παλινδρόμηση σημείωσε πρόοδο στη βαθμολογία ROC AUC από 0,594 στο αρχικό σύνολο δεδομένων σε 0,658 στο προεπεξεργασμένο, ενισχύοντας την σημαντική επίδραση της προεπεξεργασίας στην ενίσχυση της ικανότητας διάκρισης ενός ταξινομητή σε δυαδικές κλάσεις. Η μετατόπιση για τα δέντρα απόφασης μετά την προεπεξεργασία ήταν οριακή, από 0,840 σε 0,842, υποδηλώνοντας ότι είναι εκ των πραγμάτων λιγότερο ευαίσθητα στην προπεξεργασία που έγινε.

Άξια προσοχής είναι η απόδοση του KNN - ο καταλληλότερος αλγόριθμος σύμφωνα με την πειραματική αξιολόγηση - που παρουσίασε ένα εξαιρετικό άλμα στην απόδοση. Στο αρχικό σύνολο δεδομένων, παρουσίασε μια ισχυρή βαθμολογία ROC AUC 0,910. Ωστόσο, μετά την προεπεξεργασία, η βαθμολογία αυτή αυξήθηκε κατακόρυφα στο εκπληκτικό 0,973. Αυτή η εμφανής βελτίωση υπογραμμίζει την εγγενή εξάρτηση του KNN από την ποιότητα των δεδομένων. Δεδομένου ότι ο KNN είναι ένας αλγόριθμος που βασίζεται στην απόσταση, είναι εξαιρετικά ευαίσθητος στις διαφοροποιήσεις της διαβάθμισης και της ποιότητας των δεδομένων. Η προεπεξεργασία προφανώς βελτιστοποίησε τα δεδομένα με τρόπο που αύξησε σημαντικά τη διακριτική ικανότητα του KNN, καθιστώντας τον πρωταγωνιστή μεταξύ των ταξινομητών που αξιολογήθηκαν.

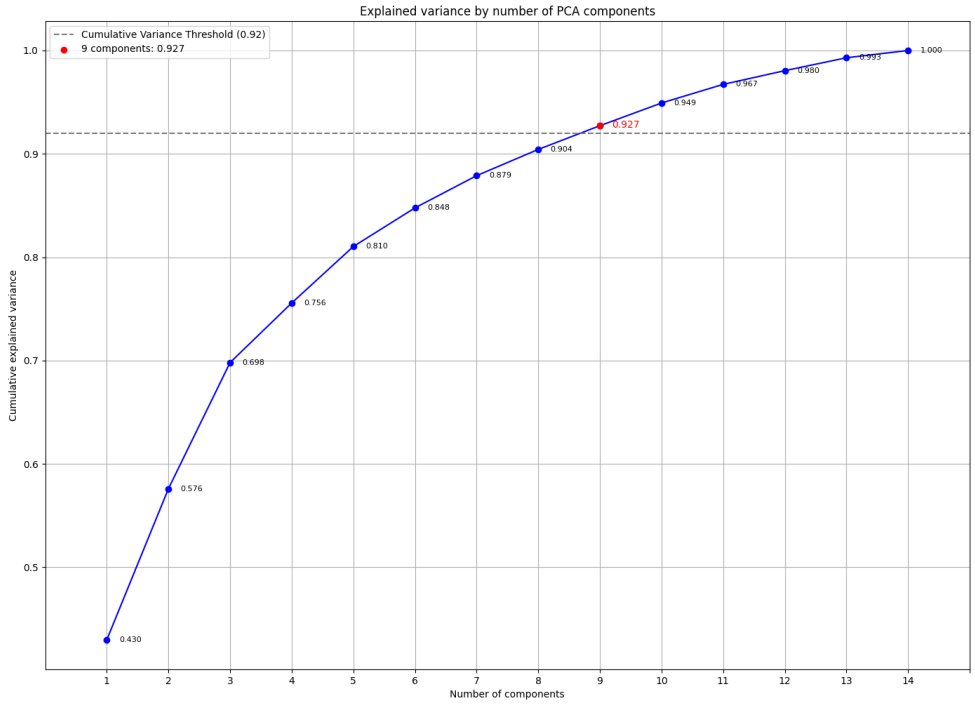
Στην ουσία, ενώ ορισμένοι ταξινομητές παρουσίασαν οριακές βελτιώσεις, άλλοι, ιδίως οι αλγόριθμοι που βασίζονται στην απόσταση και στο περιθώριο διαχωρισμού ανάμεσα στα στιγμιότυπα (margin), παρουσίασαν σημαντικές βελτιώσεις στην απόδοση μετά την προεπεξεργασία.

Συνοψίζοντας, η προεπεξεργασία επιφέρει προφανώς σημαντική θετική μεταβολή στην αποτελεσματικότητα των περισσότερων ταξινομητών, υπογραμμίζοντας τη σημασία της για την ενίσχυση της αξιοπιστίας και της ακρίβειας της προγνωστικής μοντελοποίησης, ιδίως όταν πρόκειται για σύνθετα σύνολα δεδομένων ΗΕΓ.

6.4 Ανάλυση Κύριων Συνιστωσών

Αφού εκτελέστηκε η μεθοδολογία μας στο επεξεργασμένο σύνολο δεδομένων και εφαρμόστηκε η βελτιστοποίηση των παραμέτρων για κάθε ταξινομητή ξεχωριστά, προέκυψε ο αλγόριθμος KNN, ως ο καταλληλότερος για το σύνολο δεδομένων μας. Οι βέλτιστες παράμετροι που προέκυψαν από τη βελτιστοποίηση είναι kneighbors = 1 και Weights = 'uniform'. Όπως προαναφέρθηκε και στη μεθοδολογία, επιλέγεται ο βέλτιστος αλγόριθμος με τις αντίστοιχες παραμέτρους του, καθώς ο σκοπός είναι η προβολή της μεθοδολογίας και τα αντίστοιχα αποτελέσματα σε Μεγάλα Δεδομένα ΗΕΓ. Επομένως, ζητούμενο είναι η αποφυγή του υπολογιστικού κόστους που θα επέφερε μία πειραματική αξιολόγηση με ταυτόχρονη βελτιστοποίηση πολλών ταξινομητών σε Μεγάλα Δεδομένα. Συνεπώς, καταλήγουμε στην χρήση του KNN με τις ίδιες υπερπαραμέτρους που εξήχθησαν από την προηγούμενη διαδικασία. Επιπλέον, θα θέσουμε ένα όριο συνολικής διακύμανσης (cumulative variance) για την επιλογή των συνιστωσών στο 0.92. Με αυτά τα δεδομένα, τα αποτελέσματα μετά την εφαρμογή της Ανάλυσης Κύριων Συνιστωσών είναι όπως παρουσιάζονται στα παρακάτω τμήματα.

Οι συνολικές διαστάσεις του συνόλου δεδομένων μας είναι 14 και εφαρμόζοντας PCA, θέλουμε να εξετάσουμε το ποσοστό της συνολικής διακύμανσης των δεδομένων που επεξηγούνται, καθώς μειώνουμε τις διαστάσεις των δεδομένων. Όπως βλέπουμε στο παρακάτω γράφημα, μειώνοντας τις διαστάσεις κατά 35,7%, δηλαδή σε 9 κύριες συνιστώσες, συμπεριλαμβάνεται 92,7% των αρχικών δεδομένων (εξηγούμενη διακύμανση 0.927). Αυτό πρακτικά σημαίνει πως παρόλο που μειώνουμε τις σημαντικά τις διαστάσεις των δεδομένων, οι νέες διαστάσεις που προκύπτουν, καταφέρνουν να συμπεριλάβουν το 92,7% της αρχικής πληροφορίας, έχοντας απωλέσει μόνο 7,3%. Φυσικά, μπορούμε να μειώσουμε και άλλο τις διαστάσεις και να κρατήσουμε λιγότερες συνιστώσες, πετυχαίνοντας μεγαλύτερη μείωση, καθώς αυτό που είναι το ζητούμενο στα Μεγάλα Δεδομένα, είναι να τα μετατρέψουμε σε λειτουργικότερα από πλευράς υπολογιστικού κόστους. Όμως, όπως θα βλέπουμε και παρακάτω στο **Σχήμα 24**, όσο μειώνονται οι διαστάσεις και η συμπερίληψη της αρχικής πληροφορίας, τόσο χάνεται πληροφορία που είναι χρήσιμη για την εφαρμογή του αλγορίθμου ΜΜ (KNN), που θα χρησιμοποιήσει αυτά τα δεδομένα για να ταξινομήσει (προβλέψει) την κατάσταση των οφθαλμών των ασθενών. Αυτό έχει ως συνέπεια χαμηλότερες τιμές στις μετρικές που αξιολογούνται και κατ' επέκταση στην αποτελεσματικότητα του μοντέλου που χρησιμοποιείται.



Σχήμα 24: Γράφημα Αθροιστικής Διακύμανσης ανά αριθμό Συνιστωσών

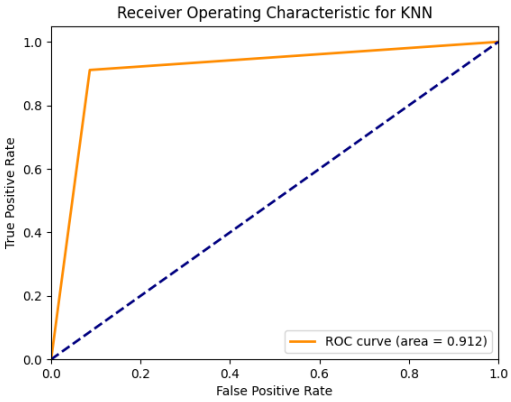
Διατηρώντας λοιπόν μία συντηρητική χρήση της μείωσης των διαστάσεων σε 9 κύριες συνιστώσες, εφαρμόζουμε τον αλγόριθμο KNN για την εξαγωγή των μετρικών. Τα αποτελέσματα που προκύπτουν είναι αξιοσημείωτα και φαίνονται στον **Πίνακα 3**.

	Χρόνος (Sec)	*Accuracy*	*Precision*	*Recall*	*F1-score*	*Mcc*	*ROC_AUC*	*PRC_AUC*
KNN	0.008	0.913	0.913	0.913	0.913	0.824	0.912	0.924

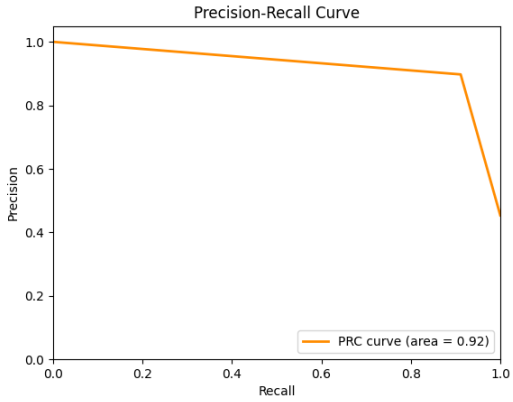
Πίνακας 3: Αποτελέσματα εφαρμογής του αλγορίθμου KNN

Ο ταξινομητής KNN επέδειξε αξιοσημείωτη απόδοση στο σύνολο δεδομένων EEG και μετά την εφαρμογή της PCA, επιτυγχάνοντας ορθότητα 91,3%. Αυτό σημαίνει ότι, από όλες τις προβλέψεις που έγιναν, περίπου το 91,3% ήταν σωστές. Η ακρίβεια, η ανάκληση και το F1-score συμπίπτουν όλα στο 91,3%, υποδηλώνοντας μια ισορροπημένη απόδοση. Συγκεκριμένα, η ακρίβεια μας λέει ότι όταν το μοντέλο προβλέπει μια θετική περίπτωση, είναι σωστό στο 91,3% των περιπτώσεων. Η ανάκληση, από την άλλη πλευρά, δείχνει ότι το μοντέλο αναγνώρισε σωστά το 91,3% όλων των πραγματικών θετικών περιπτώσεων. Το F1-score, το οποίο είναι ο αρμονικός μέσος όρος της ακρίβειας και της ανάκλησης, επιβεβαιώνει την ισορροπία στην ικανότητα του μοντέλου να προβλέπει θετικές περιπτώσεις και την ακρίβειά του σε αυτές τις προβλέψεις.

Αυτό που είναι ιδιαίτερα εντυπωσιακό σε αυτό το μοντέλο είναι η βαθμολογία ROC_AUC του 91,2% και η βαθμολογία PRC_AUC AUC του 92,4%, όπως φαίνονται στο **Σχήμα 25** και **Σχήμα 26** αντίστοιχα. Η βαθμολογία ROC_AUC, η οποία αξιολογεί την ικανότητα του ταξινομητή να διακρίνει μεταξύ των θετικών και των αρνητικών κλάσεων, υποδηλώνει ότι ο ταξινομητής KNN έχει υψηλή διακριτική ικανότητα. Μια βαθμολογία κοντά στο 100% υποδηλώνει σχεδόν τέλειο διαχωρισμό των κλάσεων. Η βαθμολογία PRC_AUC συμπληρώνει αυτό το αποτέλεσμα αντικατοπτρίζοντας την απόδοση του μοντέλου ειδικά στη θετική κλάση, γεγονός που είναι ιδιαίτερα σημαντικό για ανισοβαρή σύνολα δεδομένων. Τέλος, ο συντελεστής συσχέτισης Matthews (MCC) 0,824, ο οποίος κυμαίνεται μεταξύ -1 και 1, υποδηλώνει μια καλής ποιότητας δυαδική ταξινόμηση. Μια βαθμολογία MCC πιο κοντά στο 1 υποδηλώνει τέλεια πρόβλεψη, οπότε η τιμή 0.824 είναι πολύ σημαντική, ιδίως αν ληφθεί υπόψη συντελεστής MCC κοντά στο μηδέν θα σήμαινε τυχαία πρόβλεψη.



Σχήμα 25: ROC Curve KNN



Σχήμα 26: Precision - Recall Curve KNN

Συμπληρωματικά, να σημειωθεί η σημαντική μείωση των χρόνων βελτιστοποίησης και εκτέλεσης του αλγορίθμου. Στο επεξεργασμένο σύνολο δεδομένων, στα αποτελέσματα στον **Πίνακα 2**, ο χρόνος εκτέλεσης του αλγορίθμου 0,04. Στο σύνολο δεδομένων μετά την εφαρμογή της PCA ο αντίστοιχος χρόνος είναι 0,008 δευτερόλεπτα. Παρατηρείται μία μείωση περίπου 80% στο χρόνο εκτέλεσης του αλγορίθμου. Αυτό πρακτικά σημαίνει πως εάν κάνουμε μία προβολή των χρόνων εκτέλεσης της μεθοδολογίας σε Μεγάλα Δεδομένα EEG, εκτός από τη μείωση των διαστάσεων θα επιτυγχανόταν και σημαντική μείωση του χρόνου εκτέλεσης επεξεργασίας και ανάλυσης αυτών των δεδομένων. Οπότε, το συγκριτικό όφελος από τη χρήση αυτού του μοντέλου θα ήταν χαρακτηριστικό.

Επομένως, ο ταξινομητής KNN επιβεβαιώνει πως είναι ένας πολύ αποτελεσματικός αλγόριθμος για τη διάγνωση των οφθαλμικών καταστάσεων των ασθενών, παρουσιάζοντας ισορροπημένη και ισχυρή απόδοση σε πολλαπλές μετρικές αξιολόγησης, με μεγάλη ταχύτητα στο χρόνο εκτέλεσης. Το πιο σημαντικό όμως συμπέρασμα, είναι η αναγωγή των αποτελεσμάτων σαν προβολή σε Μεγάλα Δεδομένα ΗΕΓ. Τα παραπάνω αποτελέσματα απαντούν στο 2^{ο} ερευνητικό ερώτημα με θετικό τρόπο και επιβεβαιώνουν τη αποτελεσματικότητα της μεθοδολογίας. Μέσα από τη χρήση της πειραματικής αξιολόγησης, της βελτιστοποίησης παραμέτρων, την ανάδειξη του καλύτερου ταξινομητή και την εφαρμογή της τεχνικής PCA για τη μείωση των διαστάσεων, επιτυγχάνεται όχι μόνο η ορθή εκτίμηση ότι η συγκεκριμένη μεθοδολογία μπορεί να χρησιμοποιηθεί σε Μεγάλα Δεδομένα, αλλά επίσης ότι μπορεί να εφαρμοστεί με επιτυχία, γρήγορα και με υψηλή ακρίβεια για εφαρμογές Υγείας.

7 Σύνοψη – Μελλοντική εργασία

Στις ενότητες που προηγήθηκαν εξετάστηκε το πρόβλημα της ανίχνευσης της κατάστασης των οφθαλμών ασθενών μέσω Επιβλεπόμενης Μηχανικής Μάθησης, με υλοποίηση πειραματικής αξιολόγησης μέσω εφαρμογής αλγορίθμων Μηχανικής Μάθησης. Για να επιτευχθεί αυτό πρώτα πραγματοποίθηκε σύγκριση των αλγορίθμων στο αρχικό σύνολο δεδομένων χωρίς καμία περαιτέρω προεπεξεργασία. Στην συνέχεια η ίδια διαδικασία υλοποιήθηκε ξανά, αφού πρώτα εφαρμόστηκαν τεχνικές προεπεξεργασίας και βελτιστοποίηση παραμέτρων των αλγορίθμων. Οι βέλτιστες παράμετροι που εξήχθησαν μέσω της διασταυρούμενης επικύρωσης στο προεπεξεργασμένο σύνολο δεδομένων χρησιμοποιήθηκαν για την δημιουργία των μοντέλων που εφαρμόστηκαν, με την παράθεση των αποτελεσμάτων και των συμπερασμάτων που προέκυψαν να ακολουθούν. Στο τέλος, εφαρμόστηκε η μέθοδος PCA για τη μείωση της διαστασιμότητας του συνόλου δεδομένων και τη διερεύνηση της απόδοσης της ταξινόμησης μέσα από τις νέες διαστάσεις. Τα αποτελέσματα που προέκυψαν, διερευνήθηκαν και αναλύθηκαν για την προβολή της μεθόδου PCA σε αντίστοιχα Μεγάλα Δεδομένα ΗΕΓ και την ανάδειξη ή μη της χρησιμότητάς της σε πραγματικά δεδομένα ΗΕΓ μεγάλου μεγέθους.

Η εφαρμογή της μεθοδολογίας στο αρχικό σύνολο δεδομένων ανέδειξε ως επικρατέστερο ταξινομητή για τα δεδομένα μας, τον Random Forest με τιμή ορθότητας 0.925. Σε δεύτερη θέση ακολούθησε ο αλγόριθμος Bagging με τιμή 0.919 και το Πολυστρωματικό Νευρωνικό Δίκτυο (MLP) με τιμή 0.887.

Επεκτείνοντας την έρευνά μας στο επεξεργασμένο σύνολο δεδομένων ο αλγόριθμος KNN προκρίθηκε ως ο καταλληλότερος με τιμή ορθότητας 0.973. Στις επόμενες θέσεις βρέθηκε ο MLP με 0.965, τρίτος ο Random Forest με 0.946 και στη συνέχεια ο Bagging με τιμή 0.925. Συνυπολογίζοντας τους χρόνους εκπαίδευσης, μόλις 0,04 δευτερόλεπτα, ο KNN κατατάχθηκε με διαφορά ως ο βέλτιστος αλγόριθμος που ανέδειξε η πειραματική αξιολόγηση για τη προγνωστική μοντελοποίηση στο σύνολο δεδομένων ΗΕΓ.

Πηγαίνοντας ένα βήμα παρακάτω διερευνήθηκε η εφαρμογή της μεθόδου PCA, για να εξεταστεί εάν και πώς μέσα από τη μείωση των διαστάσεων των δεδομένων μας μπορεί να επιτευχθεί ταυτόχρονα και η διατήρηση υψηλών ποσοστών ακρίβειας ταξινόμησης, όπως εφαρμόστηκε στο επεξεργασμένο σύνολο δεδομένων. Το ερευνητικό ερώτημα από την εφαρμογή της PCA, ήταν η διερεύνηση της επιτυχίας εφαρμογής της μεθόδου σε δεδομένα ΗΕΓ και εάν ήταν επιτυχής η εξέταση της εφαρμογής της σε αντίστοιχα Δεδομένα ΗΕΓ μεγαλύτερης κλίμακας, ανάλογα με τα αποτελέσματα που θα προέκυπταν.

Τα αποτελέσματα ήταν πολύ σημαντικά, καθώς η τεχνική της Ανάλυσης Κύριων Συνιστωσών (PCA) κατάφερε να επιτύχει μία σημαντική μείωση των διαστάσεων των δεδομένων, με συνολική διακύμανση 0.927 και με ταυτόχρονη επίτευξη τιμής ορθότητας 0.913, με τη χρήση του αλγορίθμου KNN. Αυτό σημαίνει πως σε αντίστοιχα Μεγάλα Δεδομένα ΗΕΓ θα μπορούσαμε με την τεχνική της PCA, να επιτύχουμε εξαρχής μια αντίστοιχη μείωση, ενώ παράλληλα να διατηρήσουμε την πληροφορία των αρχικών δεδομένων μας σε μεγάλα ποσοστά όπως αυτό του 92,7% και να επιτύχουμε ορθή πρόβλεψη της κατάστασης των οφθαλμών σε ποσοστά που θα κυμαίνονται στο 91,3%.

Ωστόσο, θα υπήρχαν ενδεχομένως περιθώρια βελτίωσης των αποτελεσμάτων με καταλληλότερη προεπεξεργασία και υπό την προϋπόθεση ότι η μελέτη θα ενσωμάτωνε παρόμοιου είδους, αλλά και παράλληλα πιο απαιτητικών ως προς την επεξεργασία συνόλων δεδομένων. Η προεπεξεργασία των ΗΕΓ, τα οποία απεικονίζουν τη νευρική δραστηριότητα του εγκεφάλου, αποτελεί βασικό βήμα πριν την εφαρμογή αλγορίθμων μηχανικής μάθησης. Τα σήματα EEG είναι συχνά επηρεασμένα από θόρυβο, επομένως, η χρήση ψηφιακών υψηλοπερατών και χαμηλοπερατών φίλτρων, καθώς και των notch φίλτρων, θα ήταν ιδιαίτερα χρήσιμη για την αφαίρεση περιττών θορύβων και παρεμβολικών συχνοτήτων. Επιπλέον, τεχνικές όπως η Ανάλυση Ανεξάρτητων Συνιστωσών (Independent Component Analysis), μπορούν να αντιμετωπίσουν την παρουσία αρτηριακών παλμών και άλλων μυοηλεκτρικών διαταραχών. Λόγω της χρονικής φύσης των EEG δεδομένων, τεχνικές όπως η ανάλυση χρονοσειρών ή ανάλυση της κατανομής συχνοτήτων μπορούν να αποκαλύψουν σημαντικές πληροφορίες. Επιπρόσθετα, χαρακτηριστικά όπως ο μέσος, η διασπορά, ο συντελεστής διακύμανσης, η ενέργεια στις διάφορες ζώνες συχνοτήτων (θ, α, β, γ), καθώς και η συνδυαστική ανάλυση συχνοτήτων και χρονικών σημάτων μπορούν να ενισχύσουν την πληροφοριακή περιεκτικότητα του συνόλου δεδομένων και να οδηγήσουν στην εξαγωγή νέων χαρακτηριστικών. Για το λόγο αυτό ενδείκνυνται πιο προηγμένες τεχνικές για την προεπεξεργασία δεδομένων τέτοιας φύσης, συγκριτικά με αυτές που χρησιμοποιήθηκαν καθώς και βαθύτερη ερμηνεία/κατανόηση και ανάλυση για την ενδεχόμενη εξαγωγή νέων χαρακτηριστικών. Επομένως, η αναζήτηση και η εφαρμογή τέτοιων μεθόδων θα μπορούσε να αποτελέσει έναν νέο ορίζοντα σε συνέχεια της μελέτης που πραγματοποιήσαμε.

Στην προσπάθειά μας να εξετάσουμε και να αναλύσουμε τα δεδομένα, χρησιμοποιήσαμε μια σειρά από αλγόριθμους μηχανικής μάθησης που έδωσαν ενδιαφέροντα αποτελέσματα. Παρ' όλα αυτά, η ταχύτατη εξέλιξη της τεχνολογίας στον τομέα της βαθιάς μάθησης μας παρέχει μια ακόμα διάσταση που θα μπορούσαμε να εξερευνήσουμε. Η βαθιά μάθηση, με τη χρήση νευρωνικών δικτύων πολλαπλών επιπέδων, μπορεί να ανακαλύψει πολύπλοκες δομές και σχέσεις στα δεδομένα, που πολλές φορές ξεπερνούν τη δυνατότητα των παραδοσιακών μεθόδων. Ειδικά για τα δεδομένα που περιέχουν χρονοσειρές ή ακόμα και για εικόνες, τα συνελικτικά νευρωνικά δίκτυα (CNN) και τα αναδρομικά νευρωνικά δίκτυα (RNN) μπορούν να προσφέρουν ιδιαίτερα αποτελεσματικές λύσεις. Επιπλέον, η αναγνώριση μοτίβων και η αυτόματη εξαγωγή χαρακτηριστικών μέσω της βαθιάς μάθησης μπορεί να μας απαλλάξει από το βάρος της χειροκίνητης επιλογής και μετασχηματισμού χαρακτηριστικών. Ωστόσο, είναι σημαντικό να λάβουμε υπόψη ότι η εφαρμογή της βαθιάς μάθησης απαιτεί μεγαλύτερους υπολογιστικούς πόρους και εξειδικευμένη γνώση, αλλά τα αποτελέσματα μπορούν να είναι πραγματικά εντυπωσιακά, ανάλογα με το πρόβλημα που αντιμετωπίζουμε.

Επομένως μία κατεύθυνση μελλοντικής επέκτασης της παρούσας διπλωματικής θα ήταν επίσης και η εφαρμογή Βαθιών Νευρωνικών Δικτύων. Αυτή η υλοποίηση θα εξυπηρετούσε τον στόχο της μέγιστης ακρίβειας σε Μεγάλα Δεδομένα, παραβλέποντας όμως το υπολογιστικό κόστος και τους χρόνους εκτέλεσης. Συγκεκριμένα, εάν η στόχευση αφορά ακριβείς εφαρμογές Υγείας, που δεν απαιτούν προβλέψεις σε πραγματικό χρόνο, τότε ενδείκνυται η ανάπτυξη των ανωτέρω νευρωνικών δικτύων για την μεγιστοποίηση της αποτελεσματικότητας των προβλέψεων.

8 Βιβλιογραφία

[1] Hafeez Ullah Amin, Mohd Zuki Yusoff, Rana Fayyaz Ahmad, ***"A novel approach based on wavelet analysis and arithmetic coding for automated detection and diagnosis of epileptic seizure in EEG signals using machine learning techniques"***, Biomedical Signal Processing and Control, Volume 56, 2020, 101707, ISSN 1746-8094, Ανακτήθηκε στις 11.09.2023 από https://doi.org/10.1016/j.bspc.2019.101707 .

[2] Marzieh Savadkoohi, Timothy Oladunni, Lara Thompson, ***"A machine learning approach to epileptic seizure prediction using Electroencephalogram (EEG) Signal"***, Biocybernetics and Biomedical Engineering, Volume 40, Issue 3, 2020, Pages 1328-1341, ISSN 0208-5216, Ανακτήθηκε στις 11.09.2023 από https://doi.org/10.1016/j.bbe.2020.07.004 .

[3] Jaiswal, Abeg Kumar., Banka, Haider., ***"Epileptic seizure detection in EEG signal using machine learning techniques"***. *Australas Phys Eng Sci Med* **41**, 81–94 (2018). Ανακτήθηκε στις 11.09.2023 από https://doi.org/10.1007/s13246-017-0610-y .

[4] Behshad Hosseinifard, Mohammad Hassan Moradi, Reza Rostami, ***"Classifying depression patients and normal subjects using machine learning techniques and nonlinear features from EEG signal"***, Computer Methods and Programs in Biomedicine, Volume 109, Issue 3, 2013, Pages 339-345, ISSN 0169-2607, Ανακτήθηκε στις 11.09.2023 https://doi.org/10.1016/j.cmpb.2012.10.008 .

[5] Jaworska N, de la Salle S, Ibrahim M-H, Blier P and Knott V (2019), ***"Leveraging Machine Learning Approaches for Predicting Antidepressant Treatment Response Using Electroencephalography (EEG) and Clinical Data"***. Front. Psychiatry 9:768. Ανακτήθηκε στις 16.09.2023 από doi: https://doi.org/10.3389/fpsyt.2018.00768 .

[6] Miseon Shim, Han-Jeong Hwang, Do-Won Kim, Seung-Hwan Lee, Chang-Hwan Im, ***"Machine-learning-based diagnosis of schizophrenia using combined sensor level and source-level EEG features",*** Schizophrenia Research, Volume 176, Issues 23, 2016, Pages 314-319, ISSN 0920-9964, Ανακτήθηκε στις 16.09.2023 https://doi.org/10.1016/j.schres.2016.05.007 .

[7] Buettner, Ricardo, et al. ***"Development of a machine learning based algorithm to accurately detect schizophrenia based on one-minute EEG recordings."*** (2020). Ανακτήθηκε στις 16.09.2023 από https://scholarspace.manoa.hawaii.edu/handle/10125/64135 .

[8] Ana Paula S. de Oliveira, Maíra Araújo de Santana, Maria Karoline S. Andrade, Juliana Carneiro Gomes, Marcelo C. A. Rodrigues & Wellington P. dos Santos. ***"Early diagnosis of Parkinson's disease using EEG, machine learning and partial directed coherence***. Res. Biomed. Eng. 36, 311–331 (2020). Ανακτήθηκε στις 16.09.2023 από https://doi.org/10.1007/s42600-020-00072-w

[9] Betrouni, N., Delval, A., Chaton, L., Defebvre, L., Duits, A., Moonen, A., Leentjens, A.F.G. and Dujardin, K. (2019), ***"Electroencephalography-based machine learning for cognitive profiling in Parkinson's disease: Preliminary results"***. Mov Disord., 34: 210-217. Ανακτήθηκε στις 16.09.2023 από https://doi.org/10.1002/mds.27528

[10] Sonja Simpraga, Ricardo Alvarez-Jimenez, Huibert D. Mansvelder, Joop M. A. van Gerven, Geert Jan Groeneveld, Simon-Shlomo Poil & Klaus Linkenkaer-Hansen. ***"EEG machine learning for accurate detection of cholinergic intervention and Alzheimer's disease"***. *Sci Rep* **7**, 5775 (2017). Ανακτήθηκε στις 17.09.2023 από https://doi.org/10.1038/s41598-017-06165-4 .

[11] Islam A. Fouad, Fatma El-Zahraa M. Labib, ***"Identification of Alzheimer's disease from central lobe EEG signals utilizing machine learning and residual neural network"***,

Biomedical Signal Processing and Control, Volume 86, Part B, 2023, 105266, ISSN 1746-8094, Ανακτήθηκε στις 21.09.2023 από https://doi.org/10.1016/j.bspc.2023.105266 .

[12] Shahab Abdulla, Mohammed Diykh, Raid Luaibi Laft, Khalid Saleh, Ravinesh C Deo, ***"Sleep EEG signal analysis based on correlation graph similarity coupled with an ensemble extreme machine learning algorithm"***, Expert Systems with Applications, Volume 138, 2019, 112790, ISSN 0957-4174, Ανακτήθηκε στις 21.09.2023 από https://doi.org/10.1016/j.eswa.2019.07.007 .

[13] Musa Peker, ***"An efficient sleep scoring system based on EEG signal using complex-valued machine learning algorithms"***, Neurocomputing, Volume 207, 2016, Pages 165-177, ISSN 0925-2312, Ανακτήθηκε στις 21.09.2023 από https://doi.org/10.1016/j.neucom.2016.04.049 .

[14] Çiğdem İnan Acı, Murat Kaya, Yuriy Mishchenko, ***"Distinguishing mental attention states of humans via an EEG-based passive BCI using machine learning methods"***, Expert Systems with Applications, Volume 134, 2019, Pages 153-166, ISSN 0957-4174, Ανακτήθηκε στις 21.09.2023 από https://doi.org/10.1016/j.eswa.2019.05.057 .

[15] Xiao-Wei Wang, Dan Nie, Bao-Liang Lu, ***"Emotional state classification from EEG data using machine learning approach"***, Neurocomputing, Volume 129, 2014, Pages 94-106, ISSN 0925-2312, Ανακτήθηκε στις 21.09.2023 από https://doi.org/10.1016/j.neucom.2013.06.046 .

[16] Aya Hassouneh, A.M. Mutawa, M. Murugappan, ***"Development of a Real-Time Emotion Recognition System Using Facial Expressions and EEG based on machine learning and deep neural network methods"***, Informatics in Medicine Unlocked, Volume 20, 2020, 100372, ISSN 2352-9148, Ανακτήθηκε στις 21.09.2023 από https://doi.org/10.1016/j.imu.2020.100372 .

[17] S. K. Parmar, O. A. Ramwala and C. N. Paunwala, ***"Performance Evaluation of SVM with Non-Linear Kernels for EEG-based Dyslexia Detection,"*** *2021 IEEE 9th Region 10 Humanitarian Technology Conference (R10-HTC)*, Bangalore, India, 2021, pp. 1-6, doi: Ανακτήθηκε στις 22.09.2023 από https://doi.org/10.1109/R10-HTC53172.2021.9641696 .

[18] Perera P, Harshani H, Shiratuddin MF, Wong KW, Fullarton K. ***"EEG signal analysis of writing and typing between adults with dyslexia and normal controls."*** (2018). Ανακτήθηκε στις 22.09.2023 από https://doi.org/10.9781/ijimai.2018.04.005.

[19] Hafeez Ullah Amin, Aamir Saeed Malik, Rana Fayyaz Ahmad, Nasreen Badruddin, Nidal Kamel, Muhammad Hussain & Weng-Tink Chooi, ***"Feature extraction and classification for EEG signals using wavelet transform and machine learning techniques"***. Australas Phys Eng Sci Med 38, 139–149 (2015). Ανακτήθηκε στις 22.09.2023 https://doi.org/10.1007/s13246-015-0333-x .

[20] Al Zoubi Obada, Ki Wong Chung, Kuplicki Rayus T., Yeh Hung-wen, Mayeli Ahmad, Refai Hazem, Paulus Martin, Bodurka Jerzy, ***"Predicting Age From Brain EEG Signals—A Machine Learning Approach"***, Frontiers in Aging Neuroscience, Volume 10, 2018, ISSN=1663-4365, Ανακτήθηκε στις 22.09.2023 https://doi.org/10.3389/fnagi.2018.00184 .

[21] T. K. Reddy and L. Behera, ***"Online Eye state recognition from EEG data using Deep architectures"***, *2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, Budapest, Hungary, 2016, pp. 000712-000717, Ανακτήθηκε στις 24.09.2023 από doi: https://doi.org/10.1109/SMC.2016.7844325 .

[22] Narejo, Sanam; Pasero, Eros; Kulsoom, Farzana (2016). ***"EEG Based Eye State Classification using Deep Belief Network and Stacked AutoEncoder"***. In: INTERNATIONAL JOURNAL OF ELECTRICAL AND COMPUTER ENGINEERING,

vol. 6 n. 6, pp. 3131-3141. - ISSN 2088-8708, doi: 10.11591/ijece.v6i6.12967. Ανακτήθηκε στις 24.09.2023 από https://core.ac.uk/download/pdf/76533865.pdf .

[23] Ting Wang, Sheng-Uei Guan, Ka Lok Man, T. O. Ting, ***"EEG Eye State Identification Using Incremental Attribute Learning with Time-Series Classification"***, *Mathematical Problems in Engineering*, vol. 2014, Article ID 365101, 9 pages, 2014. Ανακτήθηκε στις 24.09.2023 από https://doi.org/10.1155/2014/365101 .

[24] Chirra, V.R.R., Uyyala, S.R., Kolli, V.K.K. ***"Deep CNN: A machine learning approach for driver drowsiness detection based on eye state".*** Revue d'Intelligence Artificielle, Vol. 33, No. 6, pp. 461-466. (2019). Ανακτήθηκε στις 24.09.2023 από https://doi.org/10.18280/ria.330609 .

[25] Zeng, H., Yang, C., Dai, G. et al.. ***"EEG classification of driver mental states by deep learning".*** Cogn Neurodyn 12, 597–606 (2018). Ανακτήθηκε στις 27.09.2023 από https://doi.org/10.1007/s11571-018-9496-y .

[26] Paulo Augusto de Lima Medeiros, Gabriel Vinícius Souza da Silva, Felipe Ricardo dos Santos Fernandes, Ignacio Sánchez-Gendriz, Hertz Wilton Castro Lins, Daniele Montenegro da Silva Barros, Danilo Alves Pinto Nagem, Ricardo Alexsandro de Medeiros Valentim, ***"Efficient machine learning approach for volunteer eye-blink detection in real-time using webcam"***, Expert Systems with Applications, Volume 188, 2022, 116073, ISSN 0957-4174, Ανακτήθηκε στις 27.09.2023 από https://doi.org/10.1016/j.eswa.2021.116073 .

[27] Huang, Karina, Tonya Bryant, and Bertrand Schneider. ***"Identifying Collaborative Learning States Using Unsupervised Machine Learning on Eye-Tracking, Physiological and Motion Sensor Data."*** International Educational Data Mining Society (2019). Ανακτήθηκε στις 27.09.2023 από https://eric.ed.gov/?id=ED599214 .

[28] Zhao, L., Wang, Z., Zhang, G. et al.., "Eye state recognition based on deep integrated neural network and transfer learning". Multimed Tools Appl 77, 19415–19438 (2018). Ανακτήθηκε στις 27.09.2023 από https://doi.org/10.1007/s11042-017-5380-8 .

9 ΠΑΡΑΡΤΗΜΑ

9.1 Προετοιμασία Δεδομένων

```
##**** Μεταφόρτωση και Μετατροπή Δεδομένων από .arff σε .csv***
with open('eeg-eye-state.arff', 'r') as input_file, open('eeg-eye-state.csv',
'w') as output_file:
    data_flag = False
    for line in input_file:
        if not data_flag:
            if '@DATA' in line:
                data_flag = True
                output_file.write('AF3,F7,F3,FC5,T7,P7,O1,O2,P8,T8,FC6,F4,F8,
AF4,eyeDetection\n')
        else:
            output_file.write(line)
```

9.2 Αρχικό σύνολο δεδομένων

```
######## 2. Πειραματική Αξιολόγηση σε Αρχικό Σύνολο Δεδομένων ########

# Φόρτωση Βιβλιοθηκών
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import time
from sklearn.model_selection import train_test_split, GridSearchCV
from sklearn.preprocessing import StandardScaler
from sklearn.pipeline import Pipeline
from sklearn.metrics import classification_report, roc_curve, auc,
precision_recall_curve, matthews_corrcoef
from sklearn.naive_bayes import GaussianNB
from sklearn.linear_model import LogisticRegression
from sklearn.tree import DecisionTreeClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.neural_network import MLPClassifier
from sklearn.svm import SVC
from sklearn.ensemble import AdaBoostClassifier, BaggingClassifier


# Φόρτωση δεδομένων
df = pd.read_csv('eeg-eye-state.csv')

# Διερεύνηση Δεδομένων
print(df.head())
```

```
print("")
print(df.info())

print("")
# Αναζήτηση Ελλειπόντων Τιμών
print(df.isnull().sum())

print("")
# Περιγραφική Στατιστική
print(df.describe())


# # Κατανομή Κλάσεων
sns.countplot(x='eyeDetection', data=df)
plt.savefig("Classes Distribution.png")
plt.show()

print("")
# Οπτικοποίηση κατανομή Κλάσης στόχου
sns.countplot(df['eyeDetection'])
plt.savefig("Target Class Distribution.png")
plt.show()


# Διαχωρισμός δεδομένων σε σύνολο εκπαίδευσης και ελέγχου
features = df.drop('eyeDetection', axis=1)
labels = df['eyeDetection']

X_train, X_test, y_train, y_test = train_test_split(features, labels,
test_size=0.2, random_state=42)


#****** ΒΕΛΤΙΣΤΟΠΟΙΗΣΗ ΠΑΡΑΜΕΤΡΩΝ *******

# Ορισμός τη λίστας ταξινομητών και των παραμέτρων για βελτιστοποίηση

classifiers = [
    (GaussianNB(), {}, "Naive Bayes"),
    (LogisticRegression(), {
        'classifier__C': [0.01, 0.1, 1]}, "LogisticRegression"),
        'classifier__penalty': ['none', 'l2']}, "LogisticRegression"),
    (DecisionTreeClassifier(), {
        'classifier__max_depth': [None, 5, 10]}, "Decision Tree"),
    (KNeighborsClassifier(), {
       # 'classifier__n_neighbors': [3, 5, 7, 10, 15, 20],
        'classifier__n_neighbors': range(1, 31),
        'classifier__weights': ['uniform', 'distance']}, "KNN"),
```

```
    (RandomForestClassifier(), {
        'classifier__n_estimators': [50, 100, 200],
        'classifier__max_features': ['sqrt', 'log2']}, "Random Forest"),
    (MLPClassifier(max_iter=100), {
        'classifier__alpha': [0.0001, 0.001, 0.01],
        'classifier__learning_rate_init': [0.001, 0.01, 0.1],
        'classifier__momentum': [0.2, 0.5, 0.9],
        'classifier__hidden_layer_sizes': [(10,), (50,), (100,), (50,50)],
        'classifier__activation': ['tanh', 'relu']
    }, "MLP"),
    (SVC(probability=True), { # probability=True χρειάζεται για το
predict_proba
        'classifier__C': [0.01, 0.1, 1],
        'classifier__kernel': ['linear', 'poly', 'rbf'],
    }, "SVM"),
    (AdaBoostClassifier(estimator=DecisionTreeClassifier()), {
        'classifier__n_estimators': [50, 100],
        'classifier__learning_rate': [0.01, 0.1, 1]
        }, "Ensemble - AdaBoost"),
    (BaggingClassifier(estimator=DecisionTreeClassifier()), {
        'classifier__n_estimators': [5, 10, 15, 20, 30]}, "Bagging")
]


# Προετοιμασία λεξικών για συλλογή πληροφοριών
hyper_tuning_time = {}
model_exec_time = {}
best_params = {}
metrics_dict = {}

# Δημιουργία κενής λίστας για την αποθήκευση των πληροφοριών της καμπύλης ROC
roc_info = []

# Λίστες για την αποθήκευση των αποτελεσμάτων του εκάστοτε μοντέλου και των
χρόνων εκτέλεσης
model_scores = []
model_times = []

### ***** ΒΕΛΤΙΣΤΟΠΟΙΗΣΗ - ΣΥΓΚΡΙΤΙΚΗ ΔΟΚΙΜΗ - ΤΑΞΙΝΟΜΗΣΗ **********
for classifier, params, name in classifiers:
    pipe = Pipeline(steps=[('scaler', StandardScaler()),
                           ('classifier', classifier)])

    # Δημιουργία του πλέγματος παραμέτρων
    param_grid = params

    # Χρονομέτρηση της διαδικασίας βελτιστοποίησης των υπερπαραμέτρων
```

```
    start_hyper = time.time()


    grid = GridSearchCV(pipe, param_grid, cv=5)
    grid.fit(X_train, y_train)
    end_hyper = time.time()
    print(f"Hyperparameter tuning for {name} took {end_hyper -
start_hyper:.2f} seconds.")

    hyper_tuning_time[name] = end_hyper - start_hyper


    best_model = grid.best_estimator_

    print("Best parameters: ", grid.best_params_)
    print(f"Best parameters for {name}: ", grid.best_params_)
    with open(f"{name}_best_params.txt", 'w') as file:
        file.write(str(grid.best_params_))

    # Χρονομέτρηση της εκπαίδευσης του μοντέλου
    start_model = time.time()
    best_model.fit(X_train, y_train)
    end_model = time.time()
    print(f"Model fitting for {name} took {end_model - start_model:.2f}
seconds.")



    # Εκπαίδευση και πρόβλεψη με το καλύτερο μοντέλο
    predictions = best_model.predict(X_test)
    probabilities = best_model.predict_proba(X_test)[:, 1]
    print(f"{name} Model score: {best_model.score(X_test, y_test):.3f}")
    print(classification_report(y_test, predictions))

    fpr, tpr, _ = roc_curve(y_test, probabilities)
    roc_auc = auc(fpr, tpr)
    print("ROC AUC: %.3f" % roc_auc)
    roc_info.append((fpr, tpr, roc_auc, name))

    mcc = matthews_corrcoef(y_test, predictions)
    print("MCC: %.3f" % mcc)

    # Οπτικοποίηση της καμπύλης ROC
    plt.figure()
    plt.plot(fpr, tpr, color='darkorange', lw=2, label='ROC curve (area =
%0.2f)' % roc_auc)
    plt.plot([0, 1], [0, 1], color='navy', lw=2, linestyle='--')
```

```
    plt.xlim([0.0, 1.0])
    plt.ylim([0.0, 1.05])
    plt.xlabel('False Positive Rate')
    plt.ylabel('True Positive Rate')
    plt.title(f'Receiver Operating Characteristic for {name}')
    plt.legend(loc="lower right")
    plt.savefig(f"ROC_curve_{name}.png")
    plt.show()

    # Καμπύλη Precision-Recall και AUC
    precision, recall, _ = precision_recall_curve(y_test, probabilities)
    prc_auc = auc(recall, precision)
    print("PRC AUC: %.3f" % prc_auc)

    # Οπτικοποίηση της καμπύλης Precision-Recall
    plt.figure()
    plt.plot(recall, precision, color='darkorange', lw=2, label='PRC curve
(area = %0.2f)' % prc_auc)
    plt.xlim([0.0, 1.0])
    plt.ylim([0.0, 1.05])
    plt.xlabel('Recall')
    plt.ylabel('Precision')
    plt.title('Precision-Recall Curve')
    plt.legend(loc="lower right")
    plt.savefig(f"PRC_curve_{name}.png")
    plt.show()

    # Εξαγωγή Μετρικών
    report = classification_report(y_test, predictions, output_dict=True)
    roc_auc_value = auc(fpr, tpr)
    prc_auc_value = auc(recall, precision)
    mcc_value = matthews_corrcoef(y_test, predictions)

    # Αποθήκευση μετρικών
    metrics_dict[name] = {
        'accuracy': best_model.score(X_test, y_test),
        'precision': report['weighted avg']['precision'],
        'recall': report['weighted avg']['recall'],
        'f1-score': report['weighted avg']['f1-score'],
        'roc_auc': roc_auc_value,
        'prc_auc': prc_auc_value,
        'mcc': mcc_value
        }

    model_scores.append((name, grid.best_score_, grid.best_params_))
    model_times.append((name, end_model - start_model))
```

```
    hyper_tuning_time[name] = end_hyper - start_hyper
    model_exec_time[name] = end_model - start_model

    # Καταγραφή καλύτερων παραμέτρων
    best_params[name] = grid.best_params_


# Αποθήκευση συλλεγόμενων πληροφοριών
with open("hyper_tuning_time.txt", 'w') as file:
    file.write(str(hyper_tuning_time))

with open("model_exec_time.txt", 'w') as file:
    file.write(str(model_exec_time))

with open("best_params.txt", 'w') as file:
    file.write(str(best_params))

with open("model_metrics.txt", 'w') as file:
    for classifier, metrics in metrics_dict.items():
        file.write(f"Classifier: {classifier}\n")
        for metric, value in metrics.items():
            file.write(f"{metric}: {value:.3f}\n")
        file.write("\n")

# Εντοπισμός του καλύτερου μοντέλου
best_model_name, best_model_score, best_model_params = max(model_scores,
key=lambda item:item[1])
print(f"Best Model: {best_model_name}")
print(f"Best Score: {best_model_score}")
print(f"Best Parameters: {best_model_params}")

# Εκτύπωση του χρόνου εκτέλεσης για κάθε μοντέλο
for name, execution_time in model_times:
    print(f"Execution time for {name}: {execution_time:.3f} seconds")

# Οπτικοποίηση για όλες τις καμπύλες ROC μαζί
plt.figure(figsize=(15, 10))
for fpr, tpr, roc_auc, name in roc_info:
    plt.plot(fpr, tpr, label='ROC curve for %s (area = %0.2f)' % (name,
roc_auc))
plt.plot([0, 1], [0, 1], color='navy', lw=2, linestyle='--')
plt.xlim([0.0, 1.0])
plt.ylim([0.0, 1.05])
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('Receiver Operating Characteristic')
```

```
plt.legend(loc="lower right")
plt.savefig("Comparative ROC.png")
plt.show()

# Αποθήκευση των λεπτομερειών του καλύτερου μοντέλου
with open("best_model.txt", 'w') as file:
    file.write(f"Best Model: {best_model_name}\n")
    file.write(f"Best Score: {best_model_score}\n")
    file.write(f"Best Parameters: {best_model_params}\n")

# Κλείσιμο Φακέλου
file.close()
```

9.3 Βελτιστοποίηση Παραμέτρων

```
############ 3. ΒΕΛΤΙΣΤΟΠΟΙΗΣΗ ΠΑΡΑΜΕΤΡΩΝ  ############

# Φόρτωση Βιβλιοθηκών

import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import time
from sklearn.model_selection import train_test_split, GridSearchCV
from sklearn.preprocessing import StandardScaler
from sklearn.pipeline import Pipeline
from sklearn.metrics import classification_report, roc_curve, auc,
precision_recall_curve, matthews_corrcoef
from sklearn.naive_bayes import GaussianNB
from sklearn.linear_model import LogisticRegression
from sklearn.tree import DecisionTreeClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.neural_network import MLPClassifier
from sklearn.svm import SVC
from sklearn.ensemble import AdaBoostClassifier, BaggingClassifier
from scipy.stats import zscore
from imblearn.over_sampling import SMOTE


# Φόρτωση δεδομένων
df = pd.read_csv('eeg-eye-state.csv')

#******  ΠΡΟΕΠΕΞΕΡΓΑΣΙΑ ΔΕΔΟΜΕΝΩΝ **************

# Εντοπισμός και Αφαίρεση ακραίων τιμών
```

```
z_scores = df.apply(zscore)
df = df[(z_scores < 3).all(axis=1)]

# Υπερδειγματοληψία
smote = SMOTE()
df, df['eyeDetection'] = smote.fit_resample(df.iloc[:,:-1],
df['eyeDetection'])

# Διερεύνηση Δεδομένων
print(df.head())
print(df.info())
print(df.isnull().sum())
print(df.describe())

# Οπτικοποίηση της κατανομής των κλάσεων και του πίνακα συσχέτισης
sns.countplot(x='eyeDetection', data=df)
plt.show()

plt.figure(figsize=(12,10))
sns.heatmap(df.corr(), annot=True, cmap=plt.cm.Blues)
plt.show()

# Διαχωρισμός δεδομένων σε σύνολο εκπαίδευσης και ελέγχου
features = df.drop('eyeDetection', axis=1)
labels = df['eyeDetection']
X_train, X_test, y_train, y_test = train_test_split(features, labels,
test_size=0.2, random_state=42)


#****** ΒΕΛΤΙΣΤΟΠΟΙΗΣΗ ΠΑΡΑΜΕΤΡΩΝ *******

# Ορισμός τη λίστας ταξινομητών και των παραμέτρων για βελτιστοποίηση
classifiers = [
    (GaussianNB(), {}, "Naive Bayes"),
    (LogisticRegression(), {
        'classifier__C': [0.01, 0.1, 1],
        'classifier__penalty': ['none', 'l2']}, "LogisticRegression"),
    (DecisionTreeClassifier(), {
        'classifier__max_depth': [None, 5, 10]}, "Decision Tree"),
    (KNeighborsClassifier(), {
        'classifier__n_neighbors': range(1, 31),
        'classifier__weights': ['uniform', 'distance']}, "KNN"),
    (RandomForestClassifier(), {
        'classifier__n_estimators': [50, 100, 200],
        'classifier__max_features': ['sqrt', 'log2']}, "Random Forest"),
    (MLPClassifier(max_iter=100), {
        'classifier__alpha': [0.0001, 0.001, 0.01],
```

```
        'classifier__learning_rate_init': [0.001, 0.01, 0.1],
        'classifier__momentum': [0.2, 0.5, 0.9],
        'classifier__hidden_layer_sizes': [(10,), (50,), (100,), (50,50)],
        'classifier__activation': ['tanh', 'relu']
    }, "MLP"),
    (SVC(probability=True), {
        'classifier__C': [0.01, 0.1, 1],
        'classifier__kernel': ['linear', 'poly', 'rbf'],
    }, "SVM"),
    (AdaBoostClassifier(estimator=DecisionTreeClassifier()), {
        'classifier__n_estimators': [50, 100],
        'classifier__learning_rate': [0.01, 0.1, 1]
        }, "Ensemble - AdaBoost"),
    (BaggingClassifier(estimator=DecisionTreeClassifier()), {
        'classifier__n_estimators': [5, 10, 15, 20, 30]}, "Bagging")
]

# Προετοιμασία λεξικών για συλλογή πληροφοριών
hyper_tuning_time = {}
model_exec_time = {}
best_params = {}

# Δημιουργία κενής λίστας για την αποθήκευση των πληροφοριών της καμπύλης ROC
roc_info = []

# Λίστες για την αποθήκευση των αποτελεσμάτων του εκάστοτε μοντέλου και των
χρόνων εκτέλεσης
model_scores = []
model_times = []

# Λεξικό για την αποθήκευση της ακρίβειας κάθε ταξινομητή
accuracy_dict = {}

# Δημιουργία φακέλου για αποθήκευση μετρικών
file = open("classification_reports.txt", 'w')


### ***** ΒΕΛΤΙΣΤΟΠΟΙΗΣΗ - ΣΥΓΚΡΙΤΙΚΗ ΔΟΚΙΜΗ - ΤΑΞΙΝΟΜΗΣΗ ***********
for classifier, params, name in classifiers:
    pipe = Pipeline(steps=[('scaler', StandardScaler()), ('classifier',
classifier)])
    param_grid = params

    # Χρονομέτρηση της διαδικασίας βελτιστοποίησης των υπερπαραμέτρων
    start_hyper = time.time()
    grid = GridSearchCV(pipe, param_grid, cv=5)
    grid.fit(X_train, y_train)
```

```
    end_hyper = time.time()
    print(f"Hyperparameter tuning for {name} took {end_hyper -
start_hyper:.2f} seconds.")

    hyper_tuning_time[name] = end_hyper - start_hyper


    best_model = grid.best_estimator_
    print(f"Best parameters for {name}: ", grid.best_params_)
    with open(f"{name}_best_params.txt", 'w') as file:
        file.write(str(grid.best_params_))

    # Χρονομέτρηση της εκπαίδευσης του μοντέλου
    start_model = time.time()
    best_model.fit(X_train, y_train)
    end_model = time.time()
    print(f"Model fitting for {name} took {end_model - start_model:.2f}
seconds.")

    # Εκπαίδευση και πρόβλεψη με το καλύτερο μοντέλο
    predictions = best_model.predict(X_test)
    probabilities = best_model.predict_proba(X_test)[:, 1]
    print(f"{name} Model score: {best_model.score(X_test, y_test):.3f}")
    print(classification_report(y_test, predictions))

    fpr, tpr, _ = roc_curve(y_test, probabilities)
    roc_auc = auc(fpr, tpr)
    print("ROC AUC: %.3f" % roc_auc)
    roc_info.append((fpr, tpr, roc_auc, name))

    mcc = matthews_corrcoef(y_test, predictions)
    print("MCC: %.3f" % mcc)

    # Οπτικοποίηση της καμπύλης ROC
    plt.figure()
    plt.plot(fpr, tpr, color='darkorange', lw=2, label='ROC curve (area =
%0.2f)' % roc_auc)
    plt.plot([0, 1], [0, 1], color='navy', lw=2, linestyle='--')
    plt.xlim([0.0, 1.0])
    plt.ylim([0.0, 1.05])
    plt.xlabel('False Positive Rate')
    plt.ylabel('True Positive Rate')
    plt.title(f'Receiver Operating Characteristic for {name}')
    plt.legend(loc="lower right")
    plt.show()

    # Καμπύλη Precision-Recall και AUC
```

```
    precision, recall, _ = precision_recall_curve(y_test, probabilities)
    prc_auc = auc(recall, precision)
    print("PRC AUC: %.3f" % prc_auc)

    # Οπτικοποίηση της καμπύλης Precision-Recall
    plt.figure()
    plt.plot(recall, precision, color='darkorange', lw=2, label='PRC curve
(area = %0.2f)' % prc_auc)
    plt.xlim([0.0, 1.0])
    plt.ylim([0.0, 1.05])
    plt.xlabel('Recall')
    plt.ylabel('Precision')
    plt.title('Precision-Recall Curve')
    plt.legend(loc="lower right")
    plt.show()


    model_scores.append((name, grid.best_score_, grid.best_params_))
    model_times.append((name, end_model - start_model))


    hyper_tuning_time[name] = end_hyper - start_hyper
    model_exec_time[name] = end_model - start_model

    # Καταγραφή καλύτερων παραμέτρων
    best_params[name] = grid.best_params_



    # Άνοιγμα του αρχείου για την αποθήκευση των μετρικών
    with open("classification_reports.txt", 'a') as file:
        report = classification_report(y_test, predictions)
        print(report)
        file.write(f"{name} Model:\n")
        file.write(report)
        file.write("\n")

    # Αποθήκευση accuracy
    accuracy_dict[name] = best_model.score(X_test, y_test)


# Αποθήκευση συλλεγόμενων πληροφοριών
with open("hyper_tuning_time.txt", 'w') as file:
    file.write(str(hyper_tuning_time))

with open("model_exec_time.txt", 'w') as file:
    file.write(str(model_exec_time))
```

```
with open("best_params.txt", 'w') as file:
    file.write(str(best_params))

# Εντοπισμός του καλύτερου μοντέλου
best_model_name, best_model_score, best_model_params = max(model_scores,
key=lambda item:item[1])
print(f"Best Model: {best_model_name}")
print(f"Best Score: {best_model_score}")
print(f"Best Parameters: {best_model_params}")

# Εκτύπωση του χρόνου εκτέλεσης για κάθε μοντέλο
for name, execution_time in model_times:
    print(f"Execution time for {name}: {execution_time:.3f} seconds")


# Αποθήκευση των λεπτομερειών του καλύτερου μοντέλου
with open("best_model.txt", 'w') as file:
    file.write(f"Best Model: {best_model_name}\n")
    file.write(f"Best Score: {best_model_score}\n")
    file.write(f"Best Parameters: {best_model_params}\n")


# Κλείσιμο Φακέλου
file.close()
```

9.4 Επεξεργασμένο Σύνολο Δεδομένων

```
############## 4. ΕΠΕΞΕΡΓΑΣΜΕΝΟ ΣΥΝΟΛΟ ΔΕΔΟΜΕΝΩΝ ##############

# Φόρτωση Βιβλιοθηκών
import pandas as pd
import matplotlib.pyplot as plt
from matplotlib.lines import Line2D
import time
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.pipeline import Pipeline
from sklearn.metrics import classification_report, roc_curve, auc,
precision_recall_curve, matthews_corrcoef
from sklearn.naive_bayes import GaussianNB
from sklearn.linear_model import LogisticRegression
from sklearn.tree import DecisionTreeClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.neural_network import MLPClassifier
```

```
from sklearn.svm import SVC
from sklearn.ensemble import AdaBoostClassifier, BaggingClassifier
from scipy.stats import zscore
from imblearn.over_sampling import SMOTE


#******  ΠΡΟΕΠΕΞΕΡΓΑΣΙΑ ΔΕΔΟΜΕΝΩΝ **************

# Φόρτωση δεδομένων
df = pd.read_csv('eeg-eye-state.csv')

# Εντοπισμός και Αφαίρεση ακραίων τιμών
z_scores = df.apply(zscore)
df = df[(z_scores < 3).all(axis=1)]

# Υπέρδειγματοληψία
smote = SMOTE()
df, df['eyeDetection'] = smote.fit_resample(df.iloc[:,:-1],
df['eyeDetection'])

# Διαχωρισμός δεδομένων σε σύνολο εκπαίδευσης και ελέγχου
features = df.drop('eyeDetection', axis=1)
labels = df['eyeDetection']
X_train, X_test, y_train, y_test = train_test_split(features, labels,
test_size=0.2, random_state=42)


# Φόρτωση αποθηκευμένων υπερπαραμέτρων
best_params = {'Naive Bayes': {},
               'LogisticRegression': {'C': 0.1, 'penalty': 'l2'},
               'Decision Tree': {'max_depth': None},
               'KNN': {'n_neighbors': 1, 'weights': 'uniform'},
               'Random Forest': {'max_features': 'sqrt', 'n_estimators':
200},
               'MLP': {'activation': 'tanh', 'alpha': 0.0001,
'hidden_layer_sizes': (50, 50), 'learning_rate_init': 0.01, 'momentum': 0.5},
               'SVM': {'C': 1, 'kernel': 'rbf'},
               'Ensemble - AdaBoost': {'learning_rate': 0.1, 'n_estimators':
50},
               'Ensemble -Bagging': {'n_estimators': 30}}

# Φόρτωση λίστας υπερπαραμέτρων
classifiers = [
    (GaussianNB(), "Naive Bayes"),
    (LogisticRegression(), "LogisticRegression"),
    (DecisionTreeClassifier(), "Decision Tree"),
    (KNeighborsClassifier(), "KNN"),
```

```
    (RandomForestClassifier(), "Random Forest"),
    (MLPClassifier(max_iter=100, random_state=42), "MLP"),
    (SVC(probability=True), "SVM"),
    (AdaBoostClassifier(estimator=DecisionTreeClassifier()), "Ensemble -
AdaBoost"),
    (BaggingClassifier(estimator=DecisionTreeClassifier()), "Ensemble -
Bagging")
]

# Προετοιμασία λεξικών για συλλογή πληροφοριών
model_exec_time = {}
roc_info = []
accuracy_dict = {}
metrics_dict = {}  # αποθήκευση μετρικών για κάθε ταξινομητή


### ***** ΤΑΞΙΝΟΜΗΣΗ ************
for classifier, name in classifiers:
    # Δημιουργία pipeline με τις βελτιστοποιημένες υπερπαραμέτρους
    pipe = Pipeline(steps=[('scaler', StandardScaler()), ('classifier',
classifier.set_params(**best_params[name]))])

    # Χρονομέτρηση εκπαίδευσης του μοντέλου
    start_model = time.time()
    pipe.fit(X_train, y_train)
    end_model = time.time()
    print(f"Ο χρόνος εκπαίδευσης για το μοντέλο {name} ήταν {end_model -
start_model:.2f} δευτερόλεπτα.")

    predictions = pipe.predict(X_test)
    probabilities = pipe.predict_proba(X_test)[:, 1]
    print(f"Η βαθμολογία του μοντέλου {name}: {pipe.score(X_test,
y_test):.3f}")
    print(classification_report(y_test, predictions))

    fpr, tpr, _ = roc_curve(y_test, probabilities)
    roc_auc = auc(fpr, tpr)
    print("ROC AUC: %.3f" % roc_auc)
    roc_info.append((fpr, tpr, roc_auc, name))

    mcc = matthews_corrcoef(y_test, predictions)
    print("MCC: %.3f" % mcc)

    # Οπτικοποίηση της καμπύλης ROC
    plt.figure()
    plt.plot(fpr, tpr, color='darkorange', lw=2, label='ROC curve (area =
%0.3f)' % roc_auc)
```

```
    plt.plot([0, 1], [0, 1], color='navy', lw=2, linestyle='--')
    plt.xlim([0.0, 1.0])
    plt.ylim([0.0, 1.05])
    plt.xlabel('False Positive Rate')
    plt.ylabel('True Positive Rate')
    plt.title(f'Receiver Operating Characteristic for {name}')
    plt.legend(loc="lower right")
    plt.savefig("roc_curve_" + name + ".png")  # Αποθήκευση των καμπυλών ROC
    plt.show()

    # Καμπύλη Precision-Recall και AUC
    precision, recall, _ = precision_recall_curve(y_test, probabilities)
    prc_auc = auc(recall, precision)
    print("PRC AUC: %.3f" % prc_auc)

    # Οπτικοποίηση της καμπύλης Precision-Recall
    plt.figure()
    plt.plot(recall, precision, color='darkorange', lw=2, label='PRC curve
(area = %0.2f)' % prc_auc)
    plt.xlim([0.0, 1.0])
    plt.ylim([0.0, 1.05])
    plt.xlabel('Recall')
    plt.ylabel('Precision')
    plt.title('Precision-Recall Curve')
    plt.legend(loc="lower right")
    plt.savefig("prc_curve_" + name + ".png")  # Αποθήκευση των καμπυλών
Precision-Recall
    plt.show()

    # Εξαγωγή μετρικών
    report = classification_report(y_test, predictions, output_dict=True)
    roc_auc_value = auc(fpr, tpr)
    prc_auc_value = auc(recall, precision)
    mcc_value = matthews_corrcoef(y_test, predictions)

    # Αποθήκευση μετρικών
    metrics_dict[name] = {
        'accuracy': pipe.score(X_test, y_test),
        'precision': report['weighted avg']['precision'],
        'recall': report['weighted avg']['recall'],
        'f1-score': report['weighted avg']['f1-score'],
        'roc_auc': roc_auc_value,
        'prc_auc': prc_auc_value,
        'mcc': mcc_value
        }
    # Εμφάνιση και αποθήκευση των αποτελεσμάτων σε αρχεία
```

```
    # Άνοιγμα του αρχείου για την αποθήκευση των μετρικών
    with open("classification_reports.txt", 'a') as file:
        report = classification_report(y_test, predictions)
        print(report)
        file.write(f"{name} Model:\n")
        file.write(report)
        file.write("\n")


    # Αποθήκευση accuracy
    accuracy_dict[name] = pipe.score(X_test, y_test)

    # Υπολογισμός και αποθήκευση χρόνου εκτέλεσης μοντέλου
    model_exec_time[name] = end_model - start_model

# Αποθήκευση συλλεγμένων πληροφοριών


# Αποθήκευση των accuracies
with open('accuracy_dict.txt', 'w') as file:
    for key, value in accuracy_dict.items():
        file.write(f"{key}: {value}\n")


# Αποθήκευση των χρόνων εκτέλεσης για κάθε μοντέλο σε ένα αρχείο κειμένου
.txt
with open("model_exec_time.txt", 'w') as file:
    file.write(str(model_exec_time))

with open("model_metrics.txt", 'w') as file:
    for classifier, metrics in metrics_dict.items():
        file.write(f"Classifier: {classifier}\n")
        for metric, value in metrics.items():
            file.write(f"{metric}: {value:.3f}\n")
        file.write("\n")


# Εντοπισμός του καλύτερου μοντέλου
best_model_name = max(accuracy_dict, key=accuracy_dict.get)
best_model_accuracy = accuracy_dict[best_model_name]
print(f"Best Model: {best_model_name}, Accuracy: {best_model_accuracy:.3f}")

# Εκτύπωση του χρόνου εκτέλεσης για κάθε μοντέλο
for name, execution_time in model_exec_time.items():
    print(f"Execution time for {name}: {execution_time:.3f} seconds")

# Συγκριτική οπτικοποίηση όλων των καμπυλών ROC
```

```
plt.figure(figsize=(15, 10))
for fpr, tpr, roc_auc, name in roc_info:
    plt.plot(fpr, tpr, label='ROC curve for %s (area = %0.2f)' % (name,
roc_auc))
plt.plot([0, 1], [0, 1], color='navy', lw=2, linestyle='--')
plt.xlim([0.0, 1.0])
plt.ylim([0.0, 1.05])
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('Receiver Operating Characteristic')
plt.legend(loc="lower right")
plt.savefig("combined_roc_curves.png")  # Αποθήκευση του συνδυαστικού
γραφήματος για τις καμπύλες ROC
plt.show()

# Δημιουργία διαγράμματος bar chart
plt.figure(figsize=(15, 10))

# Δημιουργία λίστας κλειδιών
keys = list(accuracy_dict.keys())

# Ορισμός του χρώματος για την μέγιστη τιμή accuracy σε πράσινο
colors = ['b' if key != max(accuracy_dict, key=accuracy_dict.get) else 'g'
for key in keys]

plt.bar(keys, accuracy_dict.values(), color=colors)
plt.xticks(rotation=90)
plt.ylabel('Accuracy')
plt.title('Accuracy of each Classifier')

# Εκτύπωση της ακρίβειας στα bars
for i, v in enumerate(accuracy_dict.values()):
    plt.text(i, v, "{:.2f}".format(v), ha='center', va='bottom')

legend_elements = [
    Line2D([0], [0], color='b', lw=4, label='Other Classifiers'),
    Line2D([0], [0], color='g', lw=4,
           label=f'Best Classifier: {best_model_name}, Accuracy:
{best_model_accuracy:.3f}')
]

plt.legend(handles=legend_elements)

plt.tight_layout()
plt.savefig("classifier_accuracies_bar_chart.png")  # Αποθήκευση του
συγκριτικού Ιστογράμματος
plt.show()
```

```
# Κλείσιμο Φακέλου
file.close()
```

9.5 PCA

```
########## 5. Ανάλυση Κύριων Συνιστωσών ###############

# Φόρτωση Βιβλιοθηκών
import pandas as pd
import matplotlib.pyplot as plt
from matplotlib.lines import Line2D
import seaborn as sns
import time
import numpy as np
from sklearn.model_selection import train_test_split, GridSearchCV
from sklearn.preprocessing import StandardScaler
from sklearn.pipeline import Pipeline
from sklearn.metrics import classification_report, roc_curve, auc,
precision_recall_curve, matthews_corrcoef
from sklearn.neighbors import KNeighborsClassifier
from sklearn.decomposition import PCA

# Φόρτωση Δεδομένων
df = pd.read_csv('eeg-eye-state.csv')

#******  ΠΡΟΕΠΕΞΕΡΓΑΣΙΑ ΔΕΔΟΜΕΝΩΝ **************

# Υπολογισμός του IQR
Q1 = df.quantile(0.25)
Q3 = df.quantile(0.75)
IQR = Q3 - Q1

# Εντοπισμός ακραίων τιμών
outliers = ((df < (Q1 - 1.5 * IQR)) | (df > (Q3 + 1.5 * IQR))).any(axis=1)

# Αφαίρεση ακραίων τιμών
df = df[~outliers]

# Διερεύνηση Δεδομένων
print(df.head())
print(df.info())
print(df.isnull().sum())
print(df.describe())

# Οπτικοποίηση της κατανομής των κλάσεων και του πίνακα συσχέτισης
```

```
sns.countplot(x='eyeDetection', data=df)
plt.savefig("Classes Distribution.png")
plt.show()

# Διαχωρισμός δεδομένων σε σύνολο εκπαίδευσης και ελέγχου
features = df.drop('eyeDetection', axis=1)
labels = df['eyeDetection']
X_train, X_test, y_train, y_test = train_test_split(features, labels,
test_size=0.3, random_state=42)

# Κανονικοποίηση στα σύνολα δεδομένων
scaler = StandardScaler()
scaler.fit(X_train)
X_train = scaler.transform(X_train)
X_test = scaler.transform(X_test)

# Ορισμός κατωφλιού για τη συνολική διακύμανση
threshold = 0.92

# Εφαρμογή της PCA και απεικόνιση του ποσοστού εξηγούμενης διακύμανσης
pca = PCA()
pca.fit(X_train)
explained_variance = np.cumsum(pca.explained_variance_ratio_)
explained_variance = np.insert(explained_variance, 0, 0)
n_components_selected = np.argmax(explained_variance >= threshold)

# Διάγραμμα εξηγούμενης διακύμανσης ανά αριθμό συνιστωσών PCA
fig, ax = plt.subplots(figsize=(17, 12))
ax.plot(range(1, len(explained_variance)), explained_variance[1:],
marker='o', color='blue', linestyle='-')
ax.axhline(y=0.92, color='gray', linestyle='--', label='Cumulative Variance
Threshold (0.92)')
ax.set_xlim(1, len(explained_variance) - 1)
ax.set_xticks(range(0, len(explained_variance)))
ax.set_xlabel('Number of components')
ax.set_ylabel('Cumulative explained variance')
ax.set_title('Explained variance by number of PCA components')
ax.grid()

# Απεικόνιση της αθροιστικής τιμής της διακύμανσης για κάθε component με ένα
bullet point επάνω στην καμπύλη
for i, var in enumerate(explained_variance[1:], start=1):
    if i == n_components_selected:
        plt.annotate(f'{var:.3f}', xy=(i + 1, var), xytext=(i + 0.2, var -
0.002), ha='left',color='red', fontsize=10)

    else:
```

```
        plt.annotate(f'{var:.3f}', xy=(i + 1, var), xytext=(i + 0.2, var -
0.002), ha='left', fontsize=8)

# Ορισμός του χρώματος bullet point του 9ου component σε κόκκινο
plt.scatter(n_components_selected, explained_variance[n_components_selected],
color='red', marker='o', zorder=5,
            label=f'{n_components_selected} components:
{explained_variance[n_components_selected]:.3f}')

# Ορισμός των σημείων του άξονα x για να εμφανίζονται όλοι οι αριθμοί των
components
tick_labels = list(range(1, len(explained_variance) + 1))
tick_labels[-1] = ''
plt.xticks(range(1, len(explained_variance) + 1), tick_labels)

plt.legend()
plt.savefig("Explained Cumulative Variance.png")
plt.show()

# Εκτύπωση αριθμού συνιστωσών για τουλάχιστον 92% διακύμανση
n_components_92 = n_components_selected
print(f"Αριθμός συνιστωσών για τουλάχιστον 92% διακύμανσης:
{n_components_selected}")

# Φόρτωση αποθηκευμένων υπερπαραμέτρων
best_params = {
               'n_neighbors': 1,
               'weights': 'uniform'
               }

# Προετοιμασία λεξικών για συλλογή πληροφοριών
model_exec_time = {}
metrics_dict = {}

# Δημιουργία κενής λίστας για την αποθήκευση των πληροφοριών της καμπύλης ROC
roc_info = []

# Εφαρμογή PCA στο σύνολο εκπαίδευσης και μετασχηματισμός των δεδομένων
pca = PCA(n_components=n_components_92)
pca.fit(X_train)
X_train_pca = pca.transform(X_train)
X_test_pca = pca.transform(X_test)


### ***** ΕΦΑΡΜΟΓΗ ΣΤΑ ΝΕΑ ΔΕΔΟΜΕΝΑ ΚΑΙ ΑΞΙΛΟΓΗΣΗ ΤΟΥ ΤΑΞΙΝΟΜΗΤΗ **********
knn_classifier = KNeighborsClassifier(**best_params)  # Φόρτωση παραμέτρων
```

```
start_model = time.time()
knn_classifier.fit(X_train_pca, y_train)
end_model = time.time()

# Έυρεση χρόνου εκτέλεσης του KNN
model_exec_time = end_model - start_model
print(f"Χρόνος εκπαίδευσης του KNN μετά την εφαρμογή PCA στα δεδομένα:
{model_exec_time:.2f} seconds")

# Αξιολόγηση του μοντέλου μετά το μετασχηματισμό των δεδομένων
predictions = knn_classifier.predict(X_test_pca)
probabilities = knn_classifier.predict_proba(X_test_pca)[:, 1]
print(f"Η βαθμολογία του μοντέλου KNN: {knn_classifier.score(X_test_pca,
y_test):.3f}")
print(classification_report(y_test, predictions))


fpr, tpr, _ = roc_curve(y_test, probabilities)
roc_auc = auc(fpr, tpr)
print("ROC AUC: %.3f" % roc_auc)
roc_info.append((fpr, tpr, roc_auc, "KNN"))

mcc = matthews_corrcoef(y_test, predictions)
print("MCC: %.3f" % mcc)

# Οπτικοποίηση της καμπύλης ROC
plt.figure()
plt.plot(fpr, tpr, color='darkorange', lw=2, label='ROC curve (area = %0.3f)'
% roc_auc)
plt.plot([0, 1], [0, 1], color='navy', lw=2, linestyle='--')
plt.xlim([0.0, 1.0])
plt.ylim([0.0, 1.05])
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title(f'Receiver Operating Characteristic for KNN')
plt.legend(loc="lower right")
plt.savefig("roc_curve_" + "KNN" + ".png")  # Αποθήκευση των καμπυλών ROC
plt.show()

# Καμπύλη Precision-Recall και AUC
precision, recall, _ = precision_recall_curve(y_test, probabilities)
prc_auc = auc(recall, precision)
print("PRC AUC: %.3f" % prc_auc)

# Οπτικοποίηση της καμπύλης Precision-Recall
plt.figure()
```

```
plt.plot(recall, precision, color='darkorange', lw=2, label='PRC curve (area
= %0.2f)' % prc_auc)
plt.xlim([0.0, 1.0])
plt.ylim([0.0, 1.05])
plt.xlabel('Recall')
plt.ylabel('Precision')
plt.title('Precision-Recall Curve')
plt.legend(loc="lower right")
plt.savefig("prc_curve_" + "KNN" + ".png")  # Αποθήκευση των καμπυλών
Precision-Recall
plt.show()

# Εξαγωγή Μετρικών
report = classification_report(y_test, predictions, output_dict=True)
roc_auc_value = auc(fpr, tpr)
prc_auc_value = auc(recall, precision)
mcc_value = matthews_corrcoef(y_test, predictions)

# Αποθήκευση μετρικών
metrics_dict["KNN"] = {
    'accuracy': knn_classifier.score(X_test_pca, y_test),
    'precision': report['weighted avg']['precision'],
    'recall': report['weighted avg']['recall'],
    'f1-score': report['weighted avg']['f1-score'],
    'roc_auc': roc_auc_value,
    'prc_auc': prc_auc_value,
    'mcc': mcc_value
    }

# Εμφάνιση και αποθήκευση των αποτελεσμάτων σε αρχεία

# Άνοιγμα του αρχείου για την αποθήκευση των πληροφοριών
with open("classification_reports.txt", 'a') as file:
    report = classification_report(y_test, predictions)
    print(report)
    file.write(f"KNN Model:\n")
    file.write(report)
    file.write("\n")

with open("model_exec_time.txt", 'w') as file:
    file.write(str(model_exec_time))

with open("best_params.txt", 'w') as file:
    file.write(str(best_params))

with open("model_metrics.txt", 'w') as file:
    for classifier, metrics in metrics_dict.items():
```

```
        file.write(f"Classifier: {classifier}\n")
        for metric, value in metrics.items():
            file.write(f"{metric}: {value:.3f}\n")
        file.write("\n")

file.close()
```