National Technical University of Athens
School of Electrical and Computer Engineering
Division of Computer Science

fMRI-Based Classification and Visual Explanation of Dyslexia and Spelling Disorder using Machine & Deep Learning

Diploma Thesis

DIMITRIOS GEORGIOU

Supervisor: **Andreas-Georgios Stafylopatis**
Professor, NTUA

Athens, October 2022

National Technical University of Athens
School of Electrical and Computer Engineering
Division of Computer Science

fMRI-Based Classification and Visual Explanation of Dyslexia and Spelling Disorder using Machine & Deep Learning

Diploma Thesis

DIMITRIOS GEORGIOU

Supervisor: **Andreas-Georgios Stafylopatis**
Professor, NTUA

Approved by the examination committee on 3th October 2022.

(Signature)	*(Signature)*	*(Signature)*
.....	.....	.....
Andreas-Georgios Stafylopatis	**Georgios Stamou**	**Georgios Siolas**
Professor, NTUA	Professor, NTUA	Associate Professor, NTUA

Athens, October 2022

National Technical University of Athens
School of Electrical and Computer Engineering
Division of Computer Science

DISCLAIMER ON ACADEMIC ETHICS AND INTELLECTUAL PROPERTY RIGHTS

Being fully aware of the implications of copyright laws, I expressly state that this diploma thesis, as well as the electronic files and source codes developed or modified in the course of this thesis, are solely the product of my personal work and do not infringe any rights of intellectual property, personality and personal data of third parties, do not contain work / contributions of third parties for which the permission of the authors / beneficiaries is required and are not a product of partial or complete plagiarism, while the sources used are limited to the bibliographic references only and meet the rules of scientific citing. The points where I have used ideas, text, files and / or sources of other authors are clearly mentioned in the text with the appropriate citation and the relevant complete reference is included in the bibliographic references section. I fully, individually and personally undertake all legal and administrative consequences that may arise in the event that it is proven, in the course of time, that this thesis or part of it does not belong to me because it is a product of plagiarism.

(Signature)

.....

Dimitrios Georgiou
Electrical and Computer Engineering Graduate of the National Technical University of Athens

Περίληψη

Στόχος της παρούσας διπλωματικής εργασίας είναι η διάγνωση και η οπτική επεξηγισιμότητα της δυσλεξίας και της ορθογραφικής διαταραχής, με χρήση βασικών ταξινομητών μηχανικής μάθησης καθώς και μοντέλων βαθιάς μάθησης σε δεδομένα fMRI. Τα δεδοδομένα που χρησιμοποιήθηκαν ελήφθησαν από τη βάση δεδομένων του MRI Lab Gras, η οποία έχει συλλέξει δεδομένα απεικόνισης νευροεγκεφάλου από εργαστήρια σε όλο τον κόσμο. Μετά από 3 διαφορετικές διαδικασίες επιλογής, οι επιστήμονες κατέληξαν στο σύνολο δεδομένων που περιέχει Μαγνητικές Τομογραφίες (fMRI, φαινοτυπικά) από 58 παιδιά ηλικίας 8 με 13 (16 με ορθογραφική διαταραχή, 20 με δυσλεξία και 22 υγιή παιδιά). Για κάθε παιδί εξαγάγαμε έναν συμμετρικό πίνακα αλληλοσχέτισης (correlation matrix) $(39, 39)$ των περιοχών του εγκεφάλου όπως αυτές ορίζονται από τον άτλαντα MSDL, ενώ κρατήσαμε το άνω τριγωνικό κομμάτι (μονοδιάστατο διάνυσμα) για τα πειράματα μας. Για την μηχανική μάθηση εφαρμόσαμε μία εξαντλητική προσέγγιση 3 βελτιστοποιήσεων (ως προς μετασχηματιστές, υπεραπαμέτρους, συλλογικών ταξινομητών) για όλους τους 16 βασικούς ταξινομητές που επιλέχθηκαν και καταλήξαμε στον καλύτερο συνδυασμό συλλογικών ταξινομητών που περιέχει τα βελτιστοποιημένα pipelines των ταξινομητών MLP, Logistic, Random Forest, Ridge, Extra Tree με test score $F1 = 83.5\%$. Επιπλεόν, εφαρμόσσαμε μια παραμετροποιημένη μορφή του αλγορίθμου LIME για να επεξήγησουμε την σημαντικότητα των 741 χαρακτηριστικών, δηλαδή ποιες εγκεφαλικές συνδέσεις συμβάλλον περισσότερο στις προβλέψεις, ενώ παράξαμε γραφήματα που διευκολύνουν την κατανόηση των σημαντικοτήτων αυτών. Τέλος, πειραματίστηκαμε με διάφορες παραλλαγές της αρχιτεκτονικής του συνελεκτικού δικτύου γράφου (G-CNN), τον γράφου του οποίου κατασκευάσαμε με τα διαθέσιμα φαινοτυπιικά δεδομένα των συμμετεχόντων (ηλικία, φύλο). Βελτιστοποιήσαμε τις υπερπαραμέτρους και καταλήξαμε στην αρχιτεκτονική που λαμβάνει το καλύτερο test score $F1 = 83.5\%$ χωρίς υπερεκπαίδευση overfitting.

Λέξεις Κλειδιά

νευρωνικά δίκτυα, μηχανική μάθηση, ταξινομητές, δυσλεξία, ορθογραφική διαταραχής, fMRI, μάθηση γνωρισμάτων, βαθιά μάθηση, μοντέλα δικτύων, πίνακας αλληλοσυσχέτισης, περιοχές ενδιαφέροντος, lime, ανάλυση άτλαντα

Abstract

The goal of this thesis is the diagnosis and visual explainability of dyslexia and orthographic disorder, using basic machine learning classifiers as well as deep learning models on fMRI data. The data used were obtained from the MRI Lab Gras database, which has collected neurobrain imaging data from laboratories around the world. After 3 different selection procedures, the scientists concluded on the dataset which contains data (fMRI, phenotypic) from 58 children aged 8 to 13 (16 with spelling disorder, 20 with dyslexia and 22 healthy children). For each child we extracted a symmetric correlation matrix $(39, 39)$ of the brain regions defined by the MSDL atlas, keeping the upper triangular piece (one-dimensional vector) for our experiments. For the machine learning part we apply 3-step optimization exhaustive approach (in terms of transformations, hyperparameters, ensembling classifiers) for all 16 base classifiers selected and we concluded upon the best ensembling classifier which contains the optimized pipelines MLP classifiers , Logistic, Random Forest, Ridge, Extra Tree achieving a test score $F1 = 83.5\%$. Additionally, we apply a parameterized form of the LIME algorithm to illustrate the importance of the 741 features, to find which brain connections contribute most to our predictions. Finally, we experimented with several variants of the Graph Convolutional Network (G-CNN) architecture, where the graph was constructed with the available participant phenotypic data. We optimized the hyperparameters and cconcluded upon the architecture that achieves the best test score $F1 = 83.3\%$ without leading to overfitting.

Keywords

machine learning, deep learning, g-cnn, classification, multiclass, dyslexia, spelling disorer, fMRI, feature extraction, regions-of-interest, correlation matrices, atlas analysis, glm analysis, lime, feature importance

Ευχαριστίες

Η συγγραφή της παρούσας Διπλωματικής Εργασίας σηματοδοτεί την ολοκλήρωση των προπτυχιακών μου σπουδών. Προτού όμως ολοκληρωθεί το κεφάλαιο αυτό της ζωής μου, θα ήθελα να ευχαριστήσω όσους με στήριξαν στην μέχρι τώρα πορεία μου και με ενθάρρυναν να αναλάβω πρωτοβουλίες επαγγελματικού και εθελοντικού χαρακτήρα.

Αρχικά, είμαι ευγνώμων που ως μέλος του Εργαστηρίου Συστημάτων Τεχνητής Νοημοσύνης και Μάθησης έλαβα πολύτιμη βοήθεια και καθοδήγηση από τον Καθηγητή μου, κ. Ανδρέα Στραφυλοπάτη και τον κ. Γεώργιο Σιόλα στην εκπόνηση της διπλωματικής μου εργασίας. Οι ιδέες τους καθώς και η άμεση ανταπόκριση τους ήταν καταλυτικές ώστε να μείνω εστιασμένος, ενώ παράλληλα με ενέπνευσαν να εξετάσω το ζήτημα της δυσλεξίας από πολλές επιστημονικές πτυχές.

Δεν θα μπορούσα να μην ευχαριστήσω τους φίλους μου, που μοιραστήκαμε αξέχαστες στιγμές με έντονο συναίσθημα. Άγχη, πάθος, χαμόγελα. Ήταν και ελπίζω να είναι πάντα δίπλα μου, όπως θα είμαι εγώ για αυτούς. Είναι εξαιρετικά παιδιά διψασμένα για ζωή και πάντα θα με εμπνέουν να γίνομαι η καλύτερη εκδοχή του ευατού μου και να εξελίσσομαι.

Τέλος, η φροντίδα της οικογένειας μου ήταν και είναι η κινητήρια δύναμη μου για να αγαπήσω τον εαυτό μου, να πιστέψω σε μένα και να κάνω τα όνειρα μου πραγματικότητα. Νιώθω ότι χωρίς την συναισθηματική στήριξη της αδερφής μου δεν θα είχα την ψυχική ανθεκτικότητα που έχω σήμερα.

Περιεχόμενα

Κεφάλαιο 1

Εκτεταμένη Περίληψη

1.1 Εισαγωγή

1.1.1 Μαθησιακές Δυσκολίες

Η δυσλεξία δεν είναι ασθένεια και δεν έχει καμία σχέση με την νοητική ικανότητα, την ηλικία, ή τις εκπαιδευτικές ευκαιρίες του παιδιού. Προκαλεί δυσκολία στο χειρισμό των λέξεων και συμβόλων και βρίσκεται κάτω από την ομπρέλα των **Μαθησιακών Δυσκολιών**.

Ο όρος μαθησιακές δυσκολίες περικλίει διαταρχές γλώσσας, διαταραχές γραφής, ανάγνωσης, αριθμητικής και επικοινωνίας. Κάτω από την ίδια ομπρέλα βρίσκονται και το σύνδρομο της υπερκινητικότητας, της διάσπασης προσοχής, οι γενικά προβληματικές ψυχολογικές καταστάσεις και οι ψευδοδιαταραχές.

Με τον όρο μαθησιακές δυσκολίες εννοούμε μια ομάδα διαταραχών που αφορούν την *γλώσσα, την αντίληψη, την ομιλία, τη γραφή, την αριθμητική, τη μάθηση και τη μνήμη, την κίνηση, την προσοχή, την σκέψη, το συλλογισμό και τους μηχανισμούς επίλυσης* που οφείλονται σε διαταραχές του κεντρικού νευρικού συστήματος.

Σύμφωνα με εκτιμήσεις που βασίζονται σε παραμέτρους και κριτήρια διάγνωσης το 15-20% του πληθυσμού αντιμετωπίζει κάποια μαθησιακή δυσκολία. Περίπου το 20-30% αυτών (3-6% του συνολικού πληθυσμού) έχουν δυσκολία στη ταχύτητα και την κατανόηση ενώ το 70-80% αυτών έχει δυσλεξία.

1.1.2 Ορισμοί για τη δυσλεξία

Η δυσλεξία είναι μία μαθησιακή δυσκολία νευροβιολογικής προέλευσης που γίνεται αντιληπτή λόγω δυσκολιών στην εκμάθηση της ανάγνωσης.

Έχουν διατυπωθεί πολλοί ορισμοί που προσπαθούν να απαντήσουν στο ερώτημα ¨Τι είναι η δυσλεξία·¨. Θα παραθέσουμε τον πρώτο ορισμού που δόθηκε αλλά και τον πιο πρόσφατο για να αναδείξουμε την πρόοδο της επιστημονικής κοινότητας στην κατανόηση της συγκεκριμένηςς μαθησιακής δυσκολίας. Σύμφωνα με τον Pringle Morgan (1896) ¨Η δυσλεξία είναι μία περίπτωση συγγενικής λεξικήςς τύφλωσης¨ (British Dyslexia Association). Τον πιο αποδεκτό ορισμό έως σήμερα έδωσσε ο Rose Jim (2009) και έπειτα υιοθετήθηκε από την Βρετανική Εταιρεία Δυσλεξίας : ¨Η δυσλεξία είναι μια μαθησιακή δυσκολία που επηρεάζει κυρίως τι δεξιότητες που εμπλέκονται στην ακριβή και με ευχέρεια ανάγνωση και ορθογραφία των λέξεων¨ (British Dyslexia Association).

1.1.3 Τα είδη της δυσλεξίας & Συμπτώματα

Κάτω από την ομπρέλα της δυσλεξίας, οι ερευνητές έχουν εντοπίσει διαφορετικούς τύπους με βάση την αιτία. Η δυσλεξία χωρίζεται σε δύο μεγάλες κατηγορίες:

- **Αναπτυξιακή ή Γενετική** : Η αναπτυξιακή δυσλεξία υπάρχει από τη γέννηση, έχει νευροβιολογική

βάση και χαρακτηρίζεται από έλλειψη στην απόκτηση των δεξιοτήτων της ανάγνωσης και της ορθογραφίας συγκριτικά με τις γενικές διανοητικές ικανότητες του ατόμου. Καθοριστικό χαρακτηριστικό είναι η δυσκολία στη **φωνολογική επεξεργασία** (Phonological Deficit) [Βούλγαρης ημήτριος, 2010] και άρα στην ανάλυση των λέξεων σε μεμονωμένους ήχους. Τα άτομα με αυτό το είδος δυσλεξίας μπορούν συχνά να επεξεργαστούν και να κατανοήσουν ολόκληρες λέξεις, αλλά όχι τους μεμονωμένους ήχους που τις απαρτίζουν. Δυσκολεύονται να αποκωδικοποιήσουν λέξεις.

Η αναπτυξιακή δυσλεξία περιλαμβάνει **πρωτοπαθή** και **δευτεροπαθή** δυσλεξία. Ορισμένες εκτιμήσεις υποδηλώνουν, ότι το 40-60% των παιδιών, των οποίων οι γονείς έχουν δυσλεξία, θα αναπτύξουν επίσης αυτή τη μαθησιακή δυσκολία.

- Η πρωτοπαθής δυσλεξία οφείλεται σε κληρονομικά γονίδια ή σε μια γενετική μετάλλαξη, που εμφανίζεται πρώτα στο ίδιο το άτομο. Η δυσλειτουργία βρίσκεται στην αριστερή πλευρά του εγκεφάλου, η οποία εμπλέκεται στην ανάγνωση, και επηρεάζει την ικανότητα ενός ατόμου να επεξεργάζεται τη γλώσσα. Είναι πιο συχνό στους άντρες παρά στις γυναίκες.
- Η δευτερογενής δυσλεξία προκαλείται από προβλήματα με τη νευρολογική ανάπτυξη κατά την εμβρυϊκή περίοδο (στη μήτρα). Όπως και με την πρωτοπαθή δυσλεξία, τα συμπτώματα της δευτερογενούς δυσλεξίας είναι παρόντα ξεκινώντας από την πρώιμη παιδική ηλικία.

- **Επίκτητη** : Η επίκτητη δυσλεξία, γνωστή και ως τραυματική δυσλεξία, εμφανίζεται στην παιδική ή ενήλικη ζωή ως αποτέλεσμα τραυματισμού ή ασθένειας. Αυτό μπορεί να είναι εγκεφαλικό τραύμα, εγκεφαλικό επεισόδιο (εγκεφαλικός τραυματισμός λόγω απόφραξης αιμοφόρου αγγείου ή αιμορραγίας στον εγκέφαλο) ή άνοια (προοδευτική μείωση της μνήμης, της ικανότητας σκέψης και της συμπεριφοράς).

  Η επίκτητη δυσλεξία διακρίνεται βάσει της αδυναμίας που εμφανίζει το ατόμο :

  - **Αδυναμία Ακουστικής Σύλληψης** : Σχετίζεται με την **Βαθιά Δυσλεξία** . Το ατόμο αδυνατεί να διακρίνετι **φθόγγους**, με αποτέλεσμα να δυσκολεύεται να ξεχωρίσει φθόγγους που μοιάζουν ηχητικά μεταξύ τους, όπως α-ο, ο-ου, ε-ι. Πρόβλημα μπορεί να παρουσιάζουν και με την οξύτητα άλλων φθόγγων, όπως: Β-Φ: βάρος - φάρος, φόβος - βαφή ή Δ-Θ: δάσος - Θάσος, δεσμός - θεσμός. Οι ασθενείς με βαθιά δυσλεξία:
    - Κάνουν σημασιολογικά λάθη στην ανάγνωση μεμονωμένων λέξεων (π.χ. δέντρα αντί δάσος)
    - Κάνουν οπτικά λάθη (π.χ. κατσαρό-κάστανο)
    - Κάνουν παράγωγα λάθη (οδηγώ-οδηγός)
    - Έχουν δυσκολίες στην ανάγνωση αφηρημένων λέξεων
    - Είναι σχεδόν αδύνατη η ανάγνωση ψευδολέξεων
  - **Αδυναμία Οπτικής Σύλληψης** : Σχετίζεται με την **Επιφανειακή Δυσλεξία**. Τα άτομα με επιφανειακή δυσλεξία δυσκολεύονται να αναγνωρίσουν οικείες λέξεις στη σελίδα και να αντιστοιχίσουν τις έντυπες λέξεις με τους ήχους τους. Αυτό τους δυσκολεύει να απομνημονεύσουν και να θυμηθούν λέξεις, ακόμα και αυτές που έχουν ήδη μάθει. Δεν μπορούν να συγκρατήσουν στη μνήμη τους τις λεπτές διαφορές μεταξύ μερικών γραμμάτων, αριθμών, μορφών ή σχημάτων, π.χ **γράμματα**: α - ο, ε - ω, β - φ, β - θ, γ - χ, ζ - ξ, ει - ιε, αι - ια, **αριθμοί**: 7 - 1, 6 - 9, 5 - 3, 2 - 3, 8 - 4, 45 - 54, 23 - 32, 36 - 63. Επίσης αντιμεταθέτουν γράμματα, συλλαβές, λέξεις (π.χ. πότρα αντί πόρτα), ή παραλείπουν γράμματα, συλλαβές, λέξεις (π.χ. πόρι αντί ποτήρι) κλπ.
  - **Αδυναμία Ακουστικής και Οπτικής Σύλληψης**

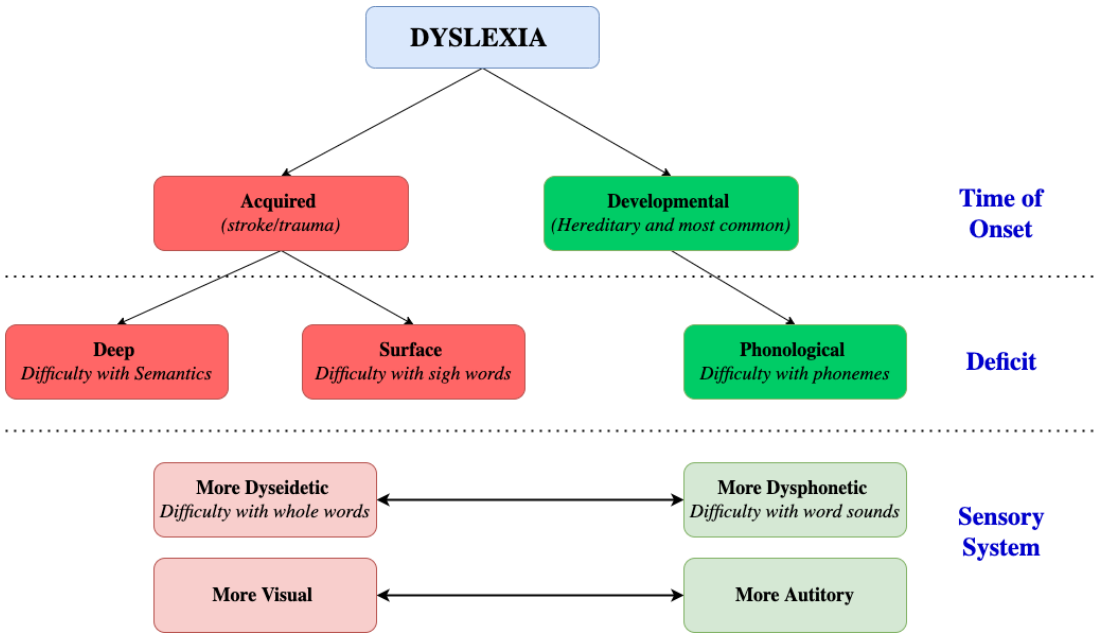


Figure 1.1. *Μορφές και τύποι Δυσλεξίας (International Dyslexia Association)*

1.1.4 Αιτιολογία

Οι μαθησιακές δυσκολίες μπορεί να αφορούν παιδιά οποιαδήποτε ηλικίας και εφυίας . Δεν οφείλονταιο πρωτογενώς σε προβλήματα ακοής, όρασης ή κίνησης, νοητικής καθυστερήσηςη, συγκινησιακής διαταραχής ή περιβαλλοντικής αποστέρησης. Προέρχονται από (1) **γενετικές αλλοιώσεις, βιοχημικές ανωμαλίες, από προ-γεννητικές καταστάσεις** (πχ. ιογενείς λοιμώξεις της εγκύου, ακτινοβολοία X, διαβήτης, νεφρική ανεπάρχεια, ...) (2) **περιγεννητικούς παράγοντες** (3) **μεταγεννητικούς παράγοντες** (πχ. μικροτραυματισμοί του παιδιού στο κεφάλι, όγκοι και φλεγμονές στον εγκέφαλο, ...) (4) Τέλος, οι **βιολογικοί, ψυχολογικοί και κοινωνικοί**παράγοντες προβληματίζουν για την σσυμμετοχή τους στην αιτιολογία των μαθησιακών δυσκολιών.

Εδώ και δεκαετίες, έχουν γίνει πολλές έρευνες οι οποίες αναζητούν την αιτία ή τα αίτια της δυσλεξίας. Πολλές κατέληξαν στα ίδια συμπεράσματα, ενώ άλλες είχαν τελείως διαφορετικά και αντικρουόμενα αποτελέσματα. Τελικά, μετά από πολλές μελέτες, οι ερευνητές κατέληξαν πως η δυσλεξία έχουν **βιολογική βάση** και οφείλεται σε εγκεφαλικές δυσλειτουργίες στο επίπεδο των νευρικών συνδέσεων [ρεφ, ς].

- **Η υπόθεση της νευρολογικής υπολειτουργίας** : Αρκετοί ερευνητές (Kinsbourne & Warrington, 1963 - Naidoo, 1972) έχουν υποστηρίξει την άποψη ότι η δυσλεξία εκδηλώνεται σε άτομα με διάφορα νευρολογικά ελλείμματα, όπως δυσκολίες στην χωρική αντίληψη και οργάνωση, στην διάκριση των αντικειμένων από τον περιβάλλοντα χώρο και ακόμα στην άρθρωση του προφορικού λόγου. Υπάρχουν 2 θεωρίες

  1. Η **πρώτη** θεωρία υποστηρίζει την ύπαρξη μιας μιας αμφίπλευρης ελαττωματικής ανάπτυξης των πίσω περιοχών του εγκεφάλου, η οποία μπορεί να προκλήθηκε από κάποια ασθένεια ή να είναι κληρονομική. Η διαταραχή αυτή προκαλεί αναγνωστικές δυσκολίες. Σύμφωνα με τον Geschwind η περιοχή της συμβολής του κροταφικού, ινιακού και βρεγματικού λοβού είναι υπεύθυνη για την επεξεργασία του γραπτού λόγου και ανώμαλη ανάπτυξη της οδηγεί σε αναγνωστικές δυσκολίες.
  2. Σύμφωνα με την **δεύτερη** θεωρία, που προτάθηκε κυρίως από τον Orton, η δυσλεξία μπορεί να οφείλεται σε ελλιπή οργάνωση του εγκεφάλου. Σε αυτή τη θεωρία βασίστηκαν οι απόψεις που αποδίδουν την δυσλεξία σε «καθυστέρηση ωρίμανσης» και «ελαφριά εγκεφαλική δυσλειτουργία», οι οποίες όμως δεν έχουν υποστηριχθεί.

- **Ελλιπής ημισφαιρική κυριαρχία & Παθολογικές Δυσλειτουργίες** στην ανατομία όσο και στην οργάνωση και την λειτουργία του εγκεφάλου που μπορεί να μεταβιβάζονται κληρονομικά. Τα άτομα που έχουν δυσλεξία δεν έχουν κυρίαρχο ημισφαίριο ή αυτό εκδηλώνεται καθυστερημένα. Οι υποστηρικτές της άποψης αυτής βασίστηκαν στα παρακάτω:

  1. Το αριστερό ημισφαίριο είναι υπεύθυνο για την παραγωγή λόγου.
  2. Η δεξιοχειρία και η μονόπλευρη ευθύνη της ομιλίας οφείλονταν σε έμφυτη λειτουργική υπεροχή του αριστερού εγκεφαλικού ημισφαιρίου.
  3. Η δυσκολία κατάκτησης των εννοιών «δεξί – αριστερό» από τους δυσλεξικούς φανερώνει την ελλιπή εγκεφαλική κυριαρχία. Η δυσκολία αυτή είναι η αιτία των καθρεπτικών λαθών στη γραφή και ανάγνωση των δυσλεξικών μαθητών (Πόρποδας, 1997).

- **Κληρονομικότητα Παράγοντες γενετικών ανωμαλιών** : Έχει παρατηρηθεί ότι η δυσλεξία μπορεί να χαρακτηρίζει τα άτομα της ίδιας οικογένειας σ' αυτή την περίπτωση ενοούμε το ευρύτερο οικογενειακό περιβάλλον και όχι μόνο τους γονείς με τα παιδιά τους. Μελέτες έχουν υποδείξει ότι πιθανώς η δυσλεξία να συνδέεται με τα γονίδια του γενετικού μας υλικού. Η ανάλυση σύνδεσης έχει εντοπίσει 9 περιοχές χρωμοσωμάτων (Schumacher 2007) και 4 υποψήφια γονίδια

  Οι ερευνητές του Τμήματος Ψυχολογικής Ιατρικής του Κολεγίου Ιατρικής της Ουαλίας ανέλυσαν τα γενετικά χαρακτηριστικά 300 οικογενειών από την Ουαλία και την δυτική Αγγλία που είχαν τουλάχιστον ένα παιδί με δυσλεξία. Απομόνωσαν το γονίδιο KIAA0319 (του χρωμοσώματος 6) συγκρίνοντας τα αποτελέσματα των 300 οικογενειών με αυτά οικογενειών χωρίς μέλος με δυσλεξία. Οι επιστήμονες επικεντρώθηκαν στο πως το KIAA0319 λειτουργεί μέσα στον εγκέφαλο και διακόπτει τις ικανότητες ανάγνωσης και γραφής του ατόμου.

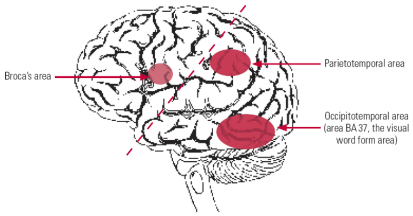
- **Η υπόθεση των Nicolson και Fawcett** : Δυσλειτουργία της παρεγκεφαλίτιδας. Η παρεγκεφαλίτιδα είναι μια περιοχή στο πίσω μέρος του εγκεφάλου η οποία είναι υπεύθυνη πρώτον, για τον **κινητικό έλεγχο** και συνεπώς για την άρθρωση του λόγου και δεύτερον για την **αυτοματοποίηση των καθηκόντων** που μαθαίνονται συνεχώς, η οποία επηρεάζει την μάθηση της αντιστοιχίας του γραφήματος-φωνήματος (Nicolson & Fawcett, 2011). Συμπερασματικά, ένας δυσλεξικός μειονεκτεί στην αυτοματοποίηση, δηλαδή στο να μάθει να εκτελεί κάποιες δεξιότητες με ευχέρεια, επιδεξιότητα και χωρίς λάθη είτε αυτές είναι κινητικές είτε γνωστικές (Μαυρομμάτη, 2004).

1.1.5 Φυσιοπαθολογία

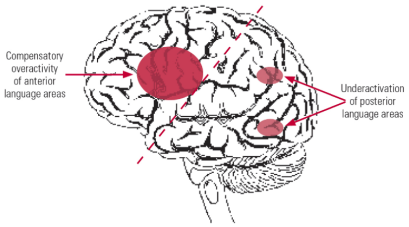
Σήμερα οι ραγδαίες εξελίξεις στην τεχνολογία όπως η μαγνητική τομογραφία (MRI) και η λειτουργική απεικόνικη μαγνητικού συντονισμού (fMRI), επιτρέπουν την οπτικοποίηση της λειτουργίαςς και της ανταπόκρισσης του εγκεφάλου κατά την διεξαγωγή γνωστικών διαδικασιών, με αποτέλεσμα οι ερευνητές να πλησιάζουν αρκετά στην εξήγηση της σύνδεσης του φαινομένου της **δυσλεξίας** με την νευρολογία. Οι Rumsey, Horwitz, Donohue, Nace, 1999) χρησιμοποίησαν τεχνικές απεικόνισης για να αναδείξουν ότι **η ροή του αίματος προς την γωνιακή έλικα (η περιοχή του εγκεφάλου που είναι υπεύθυνση για την μετάφραση του γραπτού λόγου σε προφορικού) είναι σημαντικά μειωμένη σε άτομα με δυσλεξία** Zambo Debby, 2004)

Επιπλέον, τεχνικές απεικόνισης πραγματοποιήθηκαν για την μελέτη εγκεφάλων δυσλεκτικών και μη δυσλεκτικών ατόμων κατά την διαδικασία της *ανάγνωσης, φωνολογικής και σημασιολογικής επεξεργασίας*. Πιο συγκεκριμένα, ο Shaywitz et. Al (1998) χρησιμοποιώνταςς δείγμα δυσλεκτικών και μη-δυσλεκτικών ενηλίκων διαπίστωσε ότι κατά την διαδικασία φωνολογικής επεξεργασίας η ενεργοποίηση του **αριστερού ημισφαιρίου** είναι μειωμένη σε σχέση με αυτή του αριστερού ημισφαιρίου σε έναν εγκέφαλο μη δυσλεκτικού Landi Nicole

et al., 2009 . Πλέον είναι ευρέως γνωστό ότι ο εγκέφαλος των δυσλεκτικών παρουσιάζει διαφορές σε κυτταρικό και ανατομικό επίπεο Zambo Debby, 2004



Brain Region	Brain Areas	Activation	Role
Anterior	Broca's Area	**Normal** (left)	Production of speech (phonemic word analysis and articulation during reading and naming)
Posterior (Parieto-Temporal)	Wernicke's area, Superior Temporal Gyrus, the Angular Gyrus)	**Normal** (left)	Phonological processing and in mapping letters to sounds (decoding)
Posterior (Occipito-Termporal)	Brodmann's area	**Normal** (left)	Word Recognition



Brain Region	Brain Areas	Activation
Anterior	Broca's Area	**Over-activation** (left & right)
Posterior (Parieto-Temporal)	Wernicke's area, Superior Temporal Gyrus, the Angular Gyrus)	**Under-activation**
Posterior (Occipito-Termporal)	Brodmann's area	**Under-activation**

Figure 1.2. *Σύγκριση μεταξύ του αριστερού ημισφαιριού του δυσλεκτικού και μη δυσλεκτικού εγκεφάλου με βάση τις περιοχές που ενεργοποιούνται κατά την εκτέλεσ η μαθησιακών δεξιοτήτων*

1.1.6 Διάγνωση δυσλεξίας

Η διαγνωστική διαδικασία κατά κανόνα περιλαμβάνει μια σειρά ειδικών αξιολογήσεων , στην οποία εμπλέκονται διάφοροι ειδικοί επιστήμονες. Οι ειδικότητες που συνήθως καλούνται να συμμετάσχουν άμεσα στη διαγνωστική διαδικασία είναι ο σχολικός ψυχολόγος, ο ειδικός παιδαγωγός και ο λογοπεδικός. Για την επίτευξη μίας έγκυρης διάγνωσης, θα πρέπει να ληφθούν υπόψη τα εξής:

- Το ιστορικό του παιδιού: η βιο-ιατρική προϊστορία, εγκυμοσύνη, γέννηση, ασθένειες, κ.α.
- Ατομική εξέλιξη του παιδιού τόσο συναισθηματικά όσο και κοινωνικά ,συνήθειες διατροφής, κοινωνικές επαφές.
- Ανάπτυξη των αδρών κινήσεων- μπουσούλημα στα τέσσερα, περπάτημα, τρέξιμο και λεπτών κινήσεων, κόψιμο με ψαλίδι, σχέδιο, ζωγραφική.
- Ανάπτυξη των οπτικών – χωρικών ικανοτήτων - οξύτητα όρασης, οπτική ανάλυση και σύνθεση, προσανατολισμός στο χώρο.
- Πλευρίωση, αριστεροχειρία, αμφιχειρία η δεξιοχειρία.
- Αντίληψη χρονικής διαδοχής – ημέρες της εβδομάδας, μήνες καθώς και των διατροπικών λειτουργιών, όπως ικανότητα συγκράτησης σειρών, εύρεση ονομάτων προς αντίστοιχες εικόνες και αντίστροφα.
- Γλωσσική εξέλιξη , εξέλιξη στο επίπεδο της αυθόρμητης επικοινωνίας.
- Σχολικό προϊστορικό, πότε αντιληφθήκαμε τη στασιμότητα στην ανάγνωση και γραφή.
- Ψυχοδιαγνωστική και ορθο-παιδαγωγική εξέταση. (Αναστασίου,1998)

Ο ψυχολόγος είναι εκείνος που διενεργεί πρώτος την εξέταση του παιδιού ,προκειμένου να αποφανθεί για την ύπαρξη μιας σοβαρής αναγνωστικής του δυσκολίας.

1. Ένα σταθμισμένο τεστ νοημοσύνης, κατά προτίμηση η αναθεωρημένη Κλίμακα Νοημοσύνης του Wechsler για παιδιά για τον προσδιορισμό του γνωστικού πεδίου του δυσλεξικού παιδιού

2. Ένα σταθμισμένο τεστ αναγνωστικής ικανότητας. Επίσης θεωρείται αναγκαίο η χορήγηση ενός τεστ ορθογραφημένης γραφής σε περίπτωση που το δυσλεξικό παιδί είναι ενδεχόμενο να παρουσιάζει σοβαρές δυσκολίες μόνο στην ορθογραφημένη γραφή (Στασινός, 2003).

1.2 Στόχος

Ο στόχος αυτής της διπλωματικής εργασίας είναι η κατασκευή ενός βελτιστοποιημένου μοντέλου μηχανικής μάθησης που θα προβλέπει αν ένας/μία συμμετέχων/ουσα είναι *υγιής*, έχει *Δυσλεξία* ή έχει κάποια άλλη *Ορθογραφική διαταραχή* με χρήδη δεδομένων fMRI(Λειτουργική απεικόνιση μαγνητικού συντονισμού). Παράλληλα στοχεύουμε στην εμπλούτιση της πληροφορίας που παρέχεται μέσω κατασκευής διαγραμμάτων στα οποία μπορούν να παρατηρηθούν οι περιοχές του εγκεφάλου (και συνδέσεις αυτών) που συμβάλλουν σε μεγαλύτερο βαθμό στις προβλέψεις του μοντέλου.

Με τα χρόνια η τεχνητή νοημοσύνη έχει χρησιμοποιηθεί για την ανίχνευση ψυχικών ασθενειών και γλωσσικών δυσσκολιών με διαφορετικά μέσα. Οι πιο κοινές τεχνικές για την ανίχνευση της σχιζοφρένειας με χρήση τεχνητής νοημοσύνης περιλαμβάνουν σαρώσεις PET, EEG τεχνικές που περιλαμβάνουν ταξινόμηση γονιδίων και πρωτεϊνών και απεικόνιση μαγνητικού συντονισμού (MRI)

1.3 Θεωρητικό Υπόβαθρο

1.3.1 Μηχανική Μάθηση

Η Μηχανική Μάθηση είναι η τεχνολογία ανάπτυξης αλγορίθμων υπολογιστών που μπορούν να μιμηθούν την ανθρώπινη νοημοσύνη. Ως κλάδος της επιστήμης των υπολογιστών η μηχανική μάθηση προέρχεται από τη μελέτη της αναγνώρισης προτύπων και της υπολογιστικής θεωρίας μάθησης στην τεχνητή νοημοσύνη. Είναι ένα νέο πεδίο το οποίο εξελίσσεται συνεχώς και βρίσκει εφαρμογές σε πολλούς τομείς όπως η βιοπληροφορική, η οικονομία, το μάρκετινγκ, η χημειοπληροφορική κ.α

Αυτοί οι αλγόριθμοι έχουν κατασκευαστεί για να μπορούν να βελτιώνονται αυτόματα μέσω της εμπειρίας και με τη χρήση δεδομένων, γνωστών ως δεδομένα εκπαίδευσης. Οι αλγόριθμοι μηχανικής μάθησης μπορούν να ταξινομηθούν σε ξεχωριστές κατηγορίες ανάλογα με τη φύση των δεδομένων, τη διαδικασία εκμάθησης και τον τύπο μοντέλου. Η Μηχανική Μάθηση θεωρείται ως μέρος της Τεχνητής Νοημοσύνης και χρησιμοποιείται για τη λήψη προβλέψεων ή αποφάσεων με βάση τη μαθησιακή εμπειρία που απέκτησε το μοντέλο μέσω της εκπαίδευσης

Μέχρι σήμερα η τεχνολογία μηχανικής μάθησης έχει εφαρμοστεί σε διαφορετικά πεδία όπως η αναγνώριση προτύπων, υπολογιστική όραση, μηχανική διαστημικών σκαφών , χρηματοδότηση, ψυχαγωγία, οικολογία,υπολογιστική βιολογία και βιοϊατρικές και ιατρικές εφαρμογές .Υπάρχουν τρεις τύποι μηχανικής μάθησης: η **επιβλεπόμενη μάθηση**, η **μάθηση χωρίς επίβλεψη** και η **ενισχυτική μάθηση**. Σε αυτή τη διπλωματική εργασία, η επιβλεπόμενη μάθηση χρησιμοποιείται για την πολλαπλή ταξινόμηση.

- **Επιβλεπόμενη Μάθηση** : Κατά τη διάρκεια της εκμάθησης, μία συνάρτηση αντι- στοιχεί μία είσοδο σε μία έξοδο, βασισμένη σε ζευγάρια εισόδου - εξόδου. Ο αλγόριθμος αναλύει τα δεδομένα εκπαίδευσης και παράγει ένα μοντέλο, το οποίο μπορεί να χρησιμοποιηθεί για την αντιστοίχιση νέων δεδομένων. Αυτό επιτυγχάνεται με τη βελτιστοποίηση μίας συνάρτησης κόστους που συσχετίζει την είσοδο με την έξοδο. Προβλήματα επιβλεπόμενης μάθησης είναι η **ταξινόμηση**, η **πρόγνωση** και η **διερμηνεία**.

- **Μη-Επιβλεπόμενη Μάθηση** : Στην μη-επιβλεπόμενη μάθηση ο αλγόριθμος μαθαίνει μοτίβα από δεδομένα χωρίς ετικέτες, δηλαδή από δεδομένα που σε αντίθεση με πριν η έξοδος δεν παρέχεται στον αλγόριθ-

μο. Σε αυτό το είδος μηχανικής μάθησης η μηχανή χτίζει μία εσωτερική αναπαράσταση του κόσμου της. Δύο προβλήματα μη-επιβλεπόμενης μάθησης αποτελούν η ανάλυση συσχετισμών και η **ομαδοποίηση**.

- **Ενισχυτική Μάθηση** : Με την ενισχυτική μάθηση το πρόβλημα μοντελοποιείται ως πράκτορες που λαμβάνουν αποφάσεις σε ένα περιβάλλον, το οποίο πολλές φορές αλλάζει κατάσταση. Σκοπός τους είναι να μεγιστοποιηθεί το συνολικό κέρδος από τις επιβραβεύσεις που λαμβάνουν παίρνοντας μία απόφαση. Η παρούσα κατηγορία προσπαθεί να βρει την χρυσή τομή ανάμεσα στην εξερεύνηση του χώρου διερεύνησης του προβλήματος, και στην εκμετάλλευση της γνώσης που έχει συσσωρευθεί.

1.3.2 Αλγόριθμοι Ταξινόμησης

Σε αυτή την ενότητα περιγράφονται θεωρητικά οι αλγόριθμοι που χρησιμοποιήθηκαν για την ταξινόμηση των δεδομένων σε άτομα με **δυσλεξία**, με **ορθογραφική διαταραχή** και σε άτομα **χωρίς**. Με κάθε έναν από τους αλγορίθμους που αναφέρονται παρακάτω, κάποιος μπορεί να εξάγει συμπεράσματα και για το κατά πόσο σημαντικά ήταν τα χαρακτηριστικά του συνόλου δεδομένων στην ταξινόμηση. Στην παρούσα εργασία χρησιμοποιήθηκαν 16 ταξινομητές όπως αυτοί παρέχονται από την βιβλιοθήκη sklearn . Αυτοί είναι με την σειρά : *Λογιστική Παλινδρόμηση (Logistic Regression), Δέντρα Απόφασης (Decision Tree), Τυχαία Δάση (Random Forest), Δάση Ενίσχυσης (Ada Boost), Μηχανές Διανυσμάτων Υποστήριξης (Support Vector Machttps://www.overleaf.com/project/622549ce096bd9b22853d906hine (SVM), Ενίσχυσης Κλίσης* (Gradient Boosting), XGboost, Extra Tree, Linear Discriminant Analysis (LDA), Ridge, Stochastic Gradient Descent (SGD), Multi-layer Perceptron (MLP)

Λογιστική παλινδρόμηση

Η γραμμική παλινδρόμηση χρησιμοποιείται για την εκτίμηση πραγματικών τιμών με βάση συνεχείς μεταβλητές. Για παράδειγμα, η πρόβλεψη της τιμής ενός σπιτιού, των συνολικών πωλήσεων κ.λπ. Ο στόχος είναι να βρεθεί η γραμμή που ταιριάζει καλύτερα, γνωστή ως γραμμή παλινδρόμησης που αναπαρίσταται στην εξίσωση

$$y = \beta_0 + \beta_1 \cdot x + \epsilon$$

όπου ϵ είναι το σφάλμα, επομένως η διαφορά μεταξύ της παρατηρούμενης τιμής y και της ευθείας γραμμής $\beta_0 + \beta_1 \cdot x + \epsilon$. Υπάρχουν δύο τύποι γραμμικής παλινδρόμησης: η απλή γραμμική παλινδρόμηση και η πολλαπλή γραμμική παλινδρόμηση. Η πρώτη χαρακτηρίζεται από μία ανεξάρτητη μεταβλητή, ενώ η δεύτερη από πολλαπλάσια (πάνω από 1). Στο παρακάτω Σχήμα 1 παρουσιάζεται ένα παράδειγμα απλής γραμμικής παλινδρόμησης με μία ανεξάρτητη μεταβλητή.

Η εκτίμηση μέγιστης πιθανότητας (MLE) είναι ένας αλγόριθμος που χρησιμοποιε- ίται από τον αλγόριθμο λογιστικής παλινδρόμησης προκειμένου οι συντελεστές (τιμές βήτα) του αλγορίθμου να εκτιμηθούν από τα δεδομένα εκπαίδευσης. Ο αλγόριθμος (MLE) αναζητά τιμές συντελεστών που ελαχιστοποιούν το σφάλμα στις πιθανότητες που προβλέπονται από το μοντέλο αυτές στα δεδομένα. Αυτό υλοποιείται κυρίως χρησι- μοποιώντας αποδοτικούς αλγόριθμους αριθμητικής βελτιστοποίησης. Τέτοιοι αλγόριθμοι μπορούν να επιλεγούν ως παράμετροι στον ταξινομητή λογιστικής παλινδρόμησης χρησιμοποιώντας τη βιβλιοθήκη sklearn, επομένως είναι ένας επιπλέον συντονισμός υπερ-παραμέτρου στη διαδικασία ταξινόμησης. Υπάρχουν αρκετοί βελτιστοποιητές που μπορούν να χρησιμοποιηθούν στον ταξινομητή, αλλά δεν είναι όλοι κατάλληλοι για ένα πρόβλημα πολλαπλών κλάσεων όπως το τρέχον. Ως αποτέλεσμα, αναφέρονται μόνο αυτά που χρησιμοποιούνται για τα πειράματα.

Κ-πιο κοντινός γείτονας (k-Nearest Neighbor)

Ο k-nearest neighbor είναι ένας μη παραμετρικός εποπτευόμενος αλγόριθμος που χρησιμοποιείται είτε σε προβλήματα παλινδρόμησης είτε σε προβλήματα ταξινόμησης, ο οποίος χρησιμοποιεί την εγγύτητα για να κάνει προβλέψεις σχετικά με την ταξινόμηση ενός σημείου δεδομένων. Χρησιμοποιείται περισσότερο για προβλήματα ταξινόμησης, δουλεύοντας με την υπόθεση ότι παρόμοια σημεία μπορούν να βρεθούν το ένα κοντά στο άλλο. Θα συζητήσουμε την ταξινόμηση k-NN και όχι την παλινδρόμηση καθώς εξυπηρετεί το εύρος του τρέχοντος προβλήματος. Η είσοδος αποτελείται από τα κ πιο κοντινά παραδείγματα εκπαίδευσης σε ένα σύνολο δεδομένων και η έξοδος είναι μια ιδιότητα μέλους κλάσης. Η διαδικασία κατά την οποία ταξινομείται ένα σημείο ονομάζεται πλειοψηφία των γειτόνων του. Κάθε αντικείμενο εκχωρείται στην κλάση που είναι πιο κοινή μεταξύ των k πλησιέστερων γειτόνων του, όπου κ είναι ένας θετικός ακέραιος, συνήθως ένας μικρός ακέραιος. Επομένως, αν $k = 1$,τότε το αντικείμενο απλώς εκχωρείται στην κλάση αυτού του απλού πλησιέστερου γείτονα.

Για να ρυθμιστεί ποια σημεία δεδομένων είναι πιο κοντά σε ένα δεδομένο σημείο δεδομένων, πρέπει να υπολογιστεί η απόσταση μεταξύ αυτών των σημείων δεδομένων. Υπάρχουν πολλές μετρήσεις απόστασης από τις οποίες μπορούμε να επιλέξουμε, με την Ευκλείδεια απόσταση να είναι η πιο κοινή για συνεχείς μεταβλητές και η απόσταση Χαμμινγ για τις διακριτές μεταβλητές.

Μια άλλη κρίσιμη παράμετρος για τον αλγόριθμο k-NN που πρέπει να συντονιστεί είναι η τιμή k. Η τιμή κ στον αλγόριθμο k-NN καθορίζει πόσοι γείτονες θα ελεγχθούν πριν την ταξινόμηση του σημείου ερωτήματος. Η επιλογή του κ μπορεί να καθορίσει εάν ο αλγόριθμος θα υπερπροσαρμόζεται ή όχι. Οι χαμηλότερες τιμές του k μπορεί να έχουν υψηλή διακύμανση, αλλά χαμηλή προκατάληψη, ενώ μεγαλύτερες τιμές του k μπορεί να έχουν υψηλή διακύμανση και χαμηλότερη διακύμανση.

Δέντρα απόφασης (Decision Trees)

Τα δέντρα απόφασηςδιαχωρίζουν τον χώρο των δεδομένων σε ορθογωνικές δομές και εκπαιδεύουν ένα απλό μοντέλο, όπως είναι μία σταθερά, σε κάθε μία από αυτές. Συμπληρώνοντας, κάθε διαμέριση αντιστοιχεί σε έναν κόμβο ενός δυαδικού δέντρου, οι ακμές που οδηγούν στα παιδιά του είναι υποχώροι, και τα παιδιά του είναι διαμερίσεις αυτών των υποχώρων. Θεωρώντας N παρατηρήσεις της μορφής $(x_{i1}, x_{i2}, ..., x_{ip}, yi)$ όπου $i = 1, 2, ..., N$, p το σύνολο των εισόδων, y_i μία από K κλάσεις και M διαμερίσεις $R_1, ..., R_m, ..., R_M$ του χώρου δεδομένων από το δέντρο, ορίζουμε:

$$\hat{p}_{mk} = \frac{1}{N_m} \sum_{x_i \in R_m} I(y_i = k)$$

Οπου I η δείκτρια συνάρτηση και $\hat{p}^m k$ η ποσότητα των παρατηρήσεων της κλάσης kστη διαμέριση m προς το πλήθος των στοιχείων της διαμέρισης. Όσες παρατηρήσεις βρίσκονται στην διαμέριση μ ταξινομούνται με βάση την πλειονότητα των παρατηρήσεων που αυτή περιέχει, δηλαδή:

$$k(m) = arg \max_k \hat{p}_{mk}$$

Εδώ αξίζει να αναφερθεί η διαδικασία εύρεσης των διαμερίσεων $R_1, ..., R_M$. Θεωρώντας μία συνάρτηση $Q_m(T)$ ως την συνάρτηση απώλειας του δέντρου για τη διαμέριση m, μία μεταβλητή j και ένα σημείο διαμέρισης s ορίζονται οι υποχώροι:

$$R_1(j, s) = \{X|X_j \leq s\} \text{ και } R_2(j, s) = \{X|X_j < s\}$$

Αναζητούμε το ζεύγος των j και s που ικανοποιούν την:

$$min_{j,s}[Q1(T) + Q2(T)]$$

Για κάθε j η επιλογή του ς μπορεί να γίνει εύκολα, ψάχνοντας όλα τα δεδομένα εισόδου, και άρα η επιλογή του ζεύγους (j, s) είναι εύκολη. Αναδρομικά επαναλαμβάνουμε σε κάθε μία από τις περιοχές που χωρίσαμε.

Στην παραπάνω διαδικασία, για μεγάλο μέγεθος δέντρου, ελλοχεύει ο κίνδυνος της υπερεκπαίδευσης στα δεδομένα. Ετσι η βιβλιογραφία προτείνει την εξής στρατηγική για την αποφυγή του παραπάνω προβλήματος. Αρχικά η διαδικασία διαμερισμού του χώρου δεδομένων σταματά όταν οι κόμβοι φύλλα έχουν έναν ελάχιστο αριθμό από στοιχεία, π.χ. 5 και έτσι δημιουργείται ένα δέντρο,έστω T_0. Τότε χρησιμοποιείται μία τεχνική κόστους - πολυπλοκότητας για το κλάδεμα του δέντρου. Ορίζουμε ένα υποδέντρο $T \subset T0$, που προκύπτει αν κλαδέψουμε το δέντρο, δηλαδή αν αφαιρέσουμε έναν οποιοδήποτε αριθμό κόμβων του. Αν ένας κόμβος φύλλο μ όπως παραπάνω, αντιστοιχεί σε μία περιοχή Ρμ, και συνολικά υπάρχουν $|T|$ περιοχές στο υποδέντρο, τότε το κριτήριο κόστους - πολυπλοκότητας παίρνει τη μορφή:

$$C_a(T) = \sum_{m=1}^{|T|} N_m Q_m(t) + a|T|$$

Εξετάζονται πολλές τιμές για το $a \geq 0$, το οποίο ερμηνεύεται ως η ανταλλαγή μεταξύ του μεγέθους του δέντρου, και της ικανότητας του να μάθει τα δεδομένα. Τέλος το $T \subset T0$ δέντρο που επιλέγεται, πρέπει ελαχιστοποιεί το $C_a(T)$ για το εκάστοτε a

Οι διαφορετικές $Q_m(T)$ που προτείνονται για ταξινόμηση είναι:

$$\text{Misclassification Error} = \frac{1}{N_m} \sum_{i \in R_m} I(y_i \neq k(m)) = 1 - \hat{p}_{m,k}$$

$$\text{Gini Index} : \sum_{k \neq k'} p_{m,k} p'_{m,k} = \sum_{k=1}^{K} \hat{p}_{m,k}(1 - \hat{p}_{m,k})$$

$$\text{Cross entropy or deviance} : - \sum_{k=1}^{K} \hat{p}_{m,k} log \hat{p}_{m,k}$$

Ενθυλάκωση

Για τη διαδικασία της ενθυλάκωσης (Bagging), γίνεται αρχικά η δημιουργία ενός προκαθορισμένου πλήθους B συνόλων δεδομένων εκπαίδευσης, με βάση το αρχικό. Κάθε σύνολο δημιουργείται δειγματοληπτικά με επανατοποθέτηση από το αρχικό (βοοτστραπ σαμπλες), και καθένα από αυτά χρησιμοποιείται για την εκπαίδευση ενός μοντέλου. Τελικά όταν το νέο μο- ντέλο δεχτεί δεδομένα, το αποτέλεσμα που παράγει (στην περίπτωση της ταξινόμησης) είναι η κατηγορία που ψήφισε η πλειοψηφία των μοντέλων. Σκοπός της διαδικασίας είναι η μείωση της διασποράς του αποτελέσματος του μοντέλου, μέσα από τον συνδυασμό μοντέλων που έχουν θόρυβο, αλλά είναι αμερόληπτα

Τυχαία Δάση (Random Forest)

Συνδυάζοντας τις παραπάνω δύο μεθόδους προκύπτει ο εξής αλγόριθμος.

Το μοντέλο του βήματος 2 ονομάζεται τυχαίο δάσος (Ρανδομ Φορεστ) καθώς αποτελείται από πολλά δέντρα αποφάσεων, εκπαιδευμένα σε τυχαία υποσύνολα του συνόλου εκπαίδευσης.

Αλγοριθμος 1.1: *Τυχαία δάση*

1. Για $b \in \{0, 1, ...B\}$:
(α) Δημιούργησε ένα νέο σύνολο εκπαίδευσης από το αρχικό παίρνοντας δείγματα με επανατοποθέτηση
(β) Δημιούργησε ένα δέντρο απόφασης και εκπαίδευσέ το στο νέο σύνολο δεδομένων
2. Επίστρεψε το νέο συνολικό μοντέλο που προέκυψε
3. Για κάθε πρόβλεψη σε νέο σημείο βγάλε ως αποτέλεσμα την κλάση που προέκυψε στην πλειοψηφία των δέντρων.

Δέντρα ενίσχυσης (Ada Boost)

Διατηρώντας τη σημειολογία των δέντρων αποφάσεων, σε κάθε περιοχή του χώρου ενός δέντρου απόφασης R_j αντιστοιχούμε μία σταθερά γ_j ώστε οι προβλέψεις να γίνονται ως:

$$x \in R_j \rightarrow f(x) = \gamma_j$$

Και το δέντρο πλέον ορίζεται ως:

$$T(x; \partial) = \sum_{j=1}^{J} \gamma_j I(x \in R_j)$$

όπου J το πλήθος των τελικών περιοχών διαμέρισης του δέντρου και $\Theta = \{R_j, \gamma_j\}_{j=1}^{J}$ μετα- παράμετροι. Αυτές οι παράμετροι βρίσκονται ελαχιστοποιώντας το εμπειρικό ρίσκο:

$$\hat{\Theta} = arg \min_{\Theta} \sum_{j=0}^{J} \sum_{x \in R_j} L(y_i, \gamma_j)$$

όπου $L(y_{i,j})$ ησυνάρτηση σφάλματος μεταξύτων πραγματικών τιμών $\hat{y}_i$ καιτωνπροβλέψεων του μοντέλου $gamma_j$. Εδώ αναφέρουμε τον αλγόριθμο AdaBoost ο οποίος είναι ο εξής:

Αλγοριθμος 1.2: *Δέντρα Ενίσχυσης*

1. Αρχικοποίησε τα βάρη $w_i = 1/N, i = 1, 2, ..., N$, όπου N ο αριθμός των δεδομένων εκπαίδευσης.
2. Για $m = 1$ μέχρι το M:
(α) Εκπαίδευσε το μοντέλο ταξινόμησης G_m στα δεδομένα εκπαίδευσης με βάρη w_i
(β) Υπολόγισε το σφάλμα

$$err_m = \frac{\sum_{i=1}^{N} w_i I(y_i \neq G_m(x_i))}{\sum_{i=1}^{N} w_i}$$

(γ) Υπολόγισε $a_m = log((1err_m)/err_m)$
(δ) Θέσε

$$w_i \leftarrow w_i \dot{e}^{[a_m \dot{I}(y_i \neq G_m(x_i))]}, i = 1, 2, ..., N$$

3. Επίστρεψε ως αποτέλεσμα $G(x) = sign[\sum_{m=1} Ma_m G_m m(x)]$

Μηχανές Διανυσμάτων Υποστήριξης (Support Vector Machine)

Υποστήριξη διανυσματικού μηχανήματος (SVM) είναι ένας αλγόριθμος υψηλής χρήσης που χρησιμοποιείται τόσο για προβλήματα παλινδρόμησης όσο και για προβλήματα ταξινόμησης. Ο στόχος του αλγόριθμου της μηχανής διανυσμάτων υποστήριξης είναι να βρει ένα υπερεπίπεδο σε ένα χώρο N-διάστασης, όπουN είναι ο

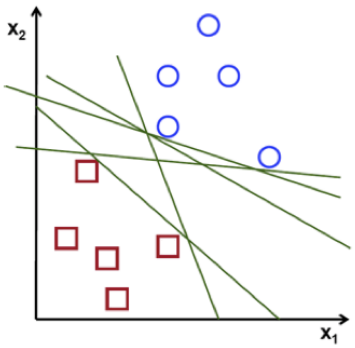


Figure 1.3. *Πιθανά υπερεπίπεδα*

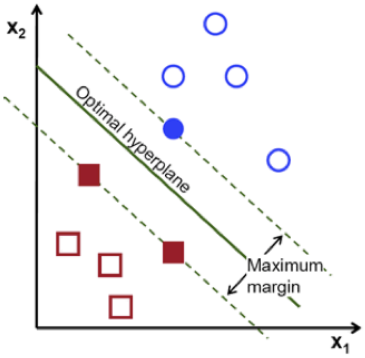


Figure 1.4. *Βέλτιστο υπερεπίπεδο*

αριθμός των χαρακτηριστικών που μπορούν να ταξινομήσουν τα σημεία δεδομένων. Χρησιμοποιεί ένα απλό μαθηματικό μοντέλο $y = w \cdot x + \gamma$ και το χειρίζεται για να επιτρέψει τη γραμμική διαίρεση τομέα. Το SVM μπορεί να χωριστεί σε γραμμικά και μη γραμμικά μοντέλα . Το μηχάνημα διανυσμάτων γραμμικής υποστήριξης μπορεί να διαιρέσει τα δεδομένα με μια γραμμική γραμμή ή υπερεπίπεδο για να διαχωρίσει τις κλάσεις στον αρχικό τομέα. Από την άλλη πλευρά, το μη γραμμικό μηχάνημα διανυσμάτων υποστήριξης υποδεικνύει ότι ο τομέας δεδομένων δεν μπορεί να διαιρεθεί γραμμικά και μπορεί να μετατραπεί σε ένα χώρο που ονομάζεται χώρος χαρακτηριστικών όπου τα δεδομένα μπορούν να διαιρεθούν γραμμικά.

Όπως φαίνεται στο Figure 1.3, υπάρχουν πολλά πιθανά υπερεπίπεδα που μπορούν να επιλεγούν, προκειμένου να διαχωριστούν οι δύο κατηγορίες σημείων δεδομένων. Ο σκοπός είναι να βρεθεί το επίπεδο που έχει το μέγιστο περιθώριο, άρα τη μέγιστη, απόσταση μεταξύ σημείων δεδομένων και από τις δύο κατηγορίες, όπως φαίνεται στο Figure 1.4. Τα σημεία δεδομένων με την ελάχιστη απόσταση από το υπερεπίπεδο ονομάζονται διανύσματα υποστήριξης και επηρεάζουν τη θέση και τον προσανατολισμό του υπερεπίπεδου. Εάν διαγράψουμε τα διανύσματα στήριξης, η θέση του υπερεπίπεδου θα αλλάξει. Χρησιμοποιώντας αυτά τα διανύσματα υποστήριξης, μεγιστοποιούμε το περιθώριο του ταξινομητή, όπως απεικονίζεται στο Figure 1.5

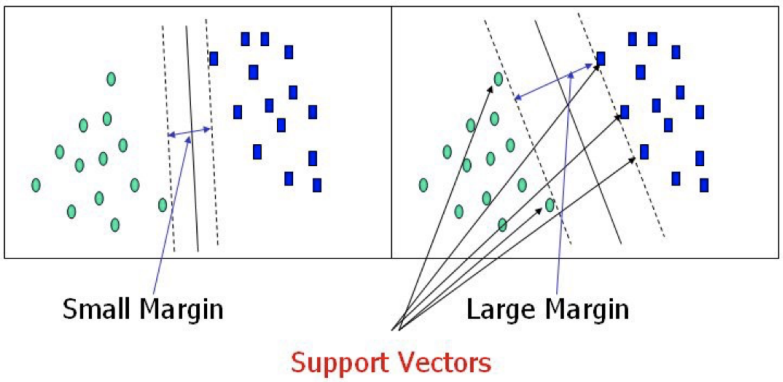


Figure 1.5. *Διανύσματα υποστήριξη*

Ενίσχυσης Κλίσης (Gradient Boosting)

Ο αλγόριθμος **Ενίσχυσης Κλίσης** (Gradient Boosting), βασίζεται στα δέντρα απόφασης και χρησιμοποιεί την τεχνική ενίσχυσης κλίσης για την βελτίωση της επίδοσης του μοντέλου. Ο αλγόριθμος αυτός παρουσιάζει αρκετές ομοιότητες με την Προσαρμοσμένη Ενίσχυση που είδαμε προηγουμένως. Γίνεται χρήση αδύναμων λεαρνερς οι οποίοι συνδυάζονται με σταθμισμένα βάρη κατά την εκπαίδευση και συγκεντρώνονται με παρόμοιο τρόπο για την δημιουργία του δυνατού λεαρνερ που παράγει την τελική τιμή εξόδου.

Αρχικά, δημιουργείται ένα μοντέλο βασισμένο σε ένα υποσύνολο των δεδομένων. Με αυτό το μοντέλο γίνονται προβλέψεις σε ολόκληρο το σύνολο των δεδομένων εκπαίδευσης και κατόπιν υπολογίζεται το σφάλμα. Το αδύναμο μοντέλο παράγει ανεπαρκείς προβλέψεις και έτσι πρέπει να ενισχυθεί σε μεταγενέστερες επαναλήψεις. Έτσι δημιουργείται ένα καινούριο μοντέλο το οποίο λαμβάνει υπόψιν τα σφάλματα που υπολογίστηκαν ήδη και επιχειρεί να εξαλείψει τα λάθη του προηγούμενου. Οι προβλέψεις αυτής της νέας επανάληψης συνδυάζονται με τις προβλέψεις της προηγούμενης

Η διαφορά της Ενίσχυσης Κλίσης από την Προσαρμοσμένη Ενίσχυση είναι ότι αυτή τη φορά, αντί να αλλάξει η βαρύτητα των στιγμιότυπων όπου ταξινομήθηκαν λάθος, γίνεται εκπαίδευση κάθε νέου μοντέλου αξιοποιώντας τα υπολειπόμενα σφάλματα του προηγούμενου. Η επαναληπτική αυτή διαδικασία σταματά είτε όταν το σφάλμα δεν αλλάζει, ή αν επιτευχθεί το μέγιστο όριο του αριθμού των μοντέλων.

XGBoost

Σε προβλήματα πρόβλεψης όπου τα δεδομένα είναι αδόμητα (εικόνες, κείμενο) τα Τεχνητά Νευρωνικά Δίκτυα (ΤΝΔ) τείνουν να επικρατούν των υπόλοιπων αλγορίθμων. Ωστόσο σε δεδομένα που είναι δομημένα, οι αλγόριθμοι που βασίζονται στα δέντρα απόφασης είναι συνήθως καλύτεροι. Ο ΞΓΒοοστ είναι μια εξέλιξη των δέντρων απόφασης που χρησιμοποιεί αρκετές έξυπνες βελτιστοποιήσεις για να πετυχαίνει καλύτερο αποτέλεσμα. Κάποιες από τις βελτιστοποιήσεις που χρησιμοποιεί είναι παραλληλοποίηση, κλάδεμα δέντρων, και βελτιστοποίηση υπολογιστικών πόρων.

Ο αλγόριθμος αυτός δημιουργήθηκε ως μέρος έρευνας στο Πανεπιστήμιο της Ωασηινγτον. Οι Tianqi Chen και Carlos Guestrin παρουσίασαν τη δουλειά τους στο συνέδριο SIGKDD 2016

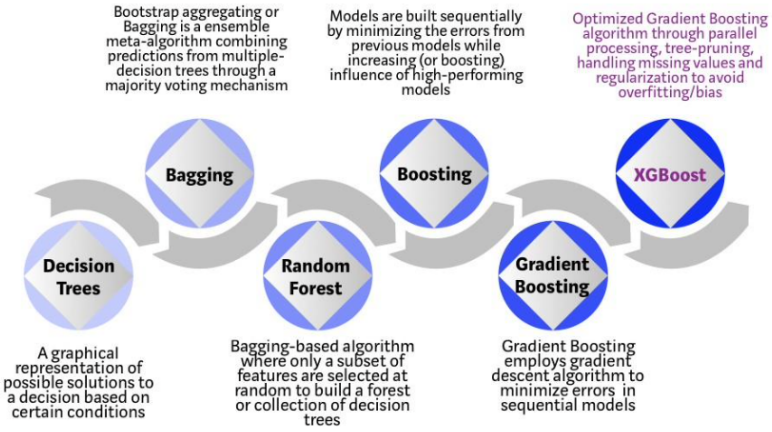


Figure 1.6. *Εξέλιξη αλγόριθμων βασισμένων σε δέντρα αποφάσεων*

1.3.3 Βελτιστοποίηση

Ένα κομμάτι της παρούσας διπλωματικής που μπορεί να προσφέρει επιστημονική αξία είναι ότι δημιουργήσαμε ένα εργαλείο για εξαντλητική βελτιστοποίηση σε 3 διαφορετικά επίπεδα, όπως φαίνεται και στο Figure 1.8

Βελτιστοποίηση no. 1 - Επιλογή βέλτιστων Μετασχηματιστών

Στόχος μας στο βήμα αυτό είναι να επιλέξουμε τον καταλληλότερο συνδυασμό προεπεξεργαστών που οδηγούν στο καλύτερο f1 score (μετρική η οποία θα αναλυθεί παρακάτω). Χρησιμοποιήθηκε η βιβλιοθήκη scikit-learn καθώς παρέχει πληθώρα μετασχηματιστών. Όπως θα αναφέρουμε και παρακάτω, τα δεδομένα

Figure 1.7. *XGBoost*

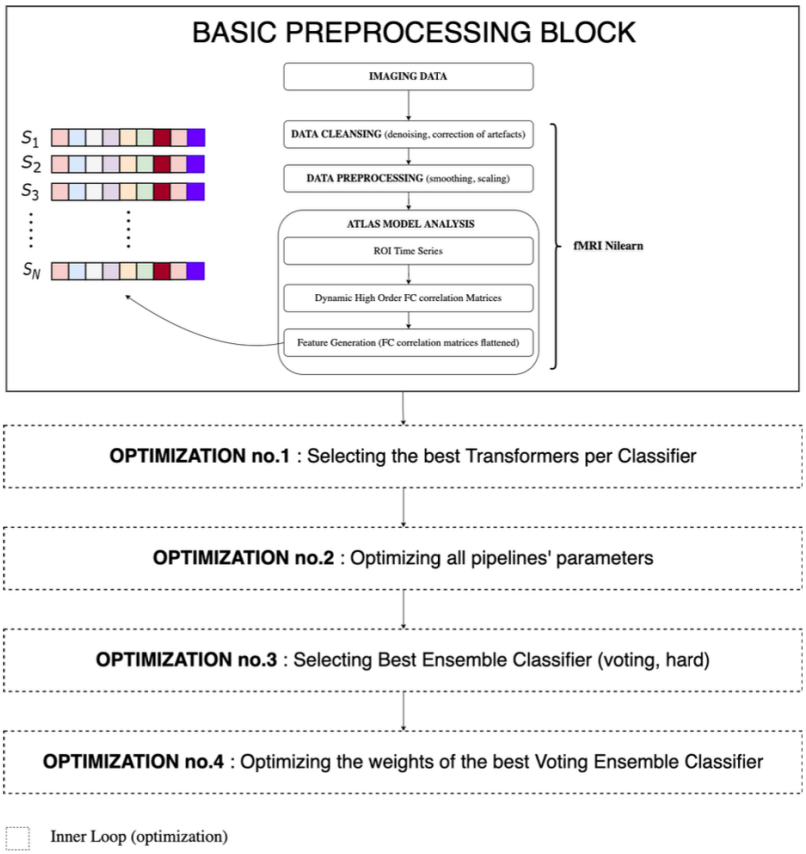


Figure 1.8. *Διάγραμμα ροής που απεικονίζει τα είδη βελτιστοποιήσεων μηχανικής μάθησης*

χωρίστηκαν σε σύνολα **εκπαίδευσης** και **ελέγχου** με αναλογία 80/20. Απλώς αξίζει να επισημάνουμε ότι κατά την διάρκεια των πειραμάτων χρησιμοποιούμε τον ίδο ¨σπόρο¨ (seed) για τον διαχωρισμό, ώστε να υπάρχει ¨αναπαραγωγικότητα¨ (reproducibility) ως προς τα αποτελέσματα, δηλάδη με ίδιο configuration πειράματος να οδηγούμαστε στα ακριβώς ίδια αποτελέσματα.

Στην παρούσα διπλωματική προσπαθούμε να διερευνήσουμε ένα προβλήματα πολλαπλής ταξινόμησης 3 κλάσεων, όπου τα χαρακτηριστικά (features) του κάθε συμμετέχοντα είναι ένα μοναδιαίο διάνυσμα D_i μεγέθους 751 που είναι το αποτέλσμα ισοπέδωσης (flattenting) του εκάστοτε πίνακα αλληλοσχέτισης. Επομένως όλα τα χαρακτηριστικά που χρησιμοποιούνται είναι **αριθμητικά** χαρακτηριστικά. Σε κλασικά προβλήματα μηχανικής μάθηδης συναντάμε μια ποικιλία τεχνικών μετασχηματισμού τόσο για **αριθμητικά** όσο και για **κατηγορικά**

χαρακτηριστικά. Παρ'όλα αυτά εστιάζουμε μόνο στις τεχνικές που αφορούν τα **αριθμητικά** ενώ παραλείπονται άλλες τεχνικές που αφορούν τα **κατηγορικά**, όπως πχ κωδικοποίηση (encoding). Παραθέτουμε τους αριθμητικούς μετασχηματιστές καθώς στον πίνακα

Scaler	**Undersampler**	**Feature Selector**	**Feature Extractor**
StandardScaler	*RandomUnderSampler*	*SelectKBest*	*PCA*
MinMaxScaler	*NearMiss*	*VarianceThreshold*	*LDA*
MaxAbsScaler	*CondensedNearestNeighbour*	*SelectPercentile*	
PowerTransformer	*TomekLinks*	*SelectFromModel(LogisticRegression)*	
QuantileTransformer	*EditedNearestNeighbours*	*GenericUnivariateSelect*	
		SelectFpr	
		SelectFdr	
		SelectFwe	

Εφόσον εξετάζουμε 16 ταξινομητές, στοχεύουμε στην εύρεση του καλύτερου συνδυασμού μεταχηματιστών, δηλαδή στην εύρεση των 16 καλύτερων (με μεγαλύτερο f1 test score) pipelines. Περιληπτικά :

- **Κλιμάκωση** (Scaling) : Αλλαγή του εύρους τιμών αλλά χωρίς αλλαγή του σχήματος κατανομής. Το εύρος συχνά ορίζεται από 0 έως 1. Οι αλγόριθμοι μηχανικής μάθησης λειτουργούν καλύτερα όταν τα χαρακτηριστικά είναι σε παρόμοια κλίμακα και προσεγγίζουν στην Κανονική Κατανομή.
- **Επιλογή Χαρακτηρστικών** (Feature Selection) : Η επιλογή χαρακτηριστικών είναι η διαδικασία απομόνωσης των πιο συνεπών, μη περιττών και σχετικών χαρακτηριστικών για χρήση στην κατασκευή μοντέλων μηνανικής μάθησης. Το νέο μέγεθος του συνόλου X θα είναι (εφόσον θα έχουμε λιγότερα χαρακτηριστικά) :

$$(n_{samples}, n'_{features}) = (n_{samples}, n_{features} \downarrow)$$

- **Εξαγωγή Χαρακτηριστικών** (Feature Extraction) : Οι πιο κοινές γραμμικές μέθοδοι για την εξαγωγή χαρακτηριστικών είναι η Ανάλυση Κύριων Συνιστωσών (PCA) και η Γραμμική Διακριτική Ανάλυση (LDA).
- **Υποδειγματοληψία** (Undersampling) : Αποφασίσαμε να μην κάνουμε καθόλου υπερδειγματολογηψία (Overrsampling) γιατί μπορεί να αυξήσει την πιθανότητα υπερπροσαρμογής (overfitting), καθώς δημιουργεί ακριβή αντίγραφα των παραδειγμάτων της μειοψηφίας, παρόλο που το dataset μας είναι αρκετά ισορροπημένο balanced μεταξύ των 3 κλάσεων $SpD \sim 21, DL \sim 23, TL \sim 27$. Οι μέθοδοι υπερδειγματοληψίας αντιγράφουν ή δημιουργούν νέα συνθετικά παραδείγματα στην κατηγορία μειοψηφίας, ενώ οι μέθοδοι υποδειγματοληψίας διαγράφουν ή συγχωνεύουν παραδείγματα στην κλάση πλειοψηφία. Οπότε εφόσον η υποδειγματοληψία δεν ασχολείται καθόλου με συνθετικά δεδομένα, προβαίνουμε στην εξέταση τεχνικών. Το νέο μέγεθος του συνόλου X θα είναι (εφόσον θα έχουμε λιγότερα δείγματα) :

$$(n'_{samples}, n_{features}) = (n_{samples} \downarrow, n_{features})$$

Βελτιστοποίηση no. 2 : Εύρεση Υπερ-Παραμέτρων με χρήση HalvingGridsearchCV

Έχοντας λοιπόν στα χέρια έναν συνδυασμό μετασχηματιστών (4 σε πλήθος) για κάθε έναν από τους 16 ταξινομητές, στο βημα αυτό αναζητούμε τις βέλτιστες υπερπαραμέτρους όλων των αλγορίθμων.

Οι μετασχηματιστές χρησιμοποιούνται για να κάνουν την προεπεξεργασία (μέσω μετασχηματισμού) των δεδομένων. Οι μετασχηματιστές γενικά έχουν και αυτοί υπερ-παραμέτρους που επηρρεάζουν τη λειτουργία τους πχ ο VarianceThreshold είχε το κατώτερο κατώφλι διακύμανσης, ο PCA τον αριθμό των κύριων συνιστωσών. Η επιλογή των υπερ-παραμέτρων (όπως το k του kNN) γίνεται μόνο εμπειρικά μέσω διασταυρούμενης επικύρωσης (cross-validation). Οι μετασχηματιστές και οι υπερπαράμετροι τους επιδρούν λοιπόν στη μορφή των δεδομένων. Επομένως στο βήμα αυτό στοχεύουμε στην εύρεση των βέλτιστων παραμέτρων για το καθε ένα από τα 16 pipelines (μετασχηματιστές και ταξινομητής)

Η απόδοση όλων των πιθανών συνδυασμών υπερ-παραμέτρων στο εκάστοτε μοντέλο γίνεται με τη μέθοδο της αναζήτησης πλέγματος (Grid Search). Η αναζήτηση πλέγματος είναι απλά η εξαντλητική αναζήτηση όλων των συνδυασμών ενός ορισμένου συνόλου τιμών για κάθε υπερ-παράμετρο του μοντέλου. Οι τιμές αυτές ορίζονται χειροκίνητα βάσει δοκιμών και εμπειρικής γνώσης. Η μέθοδος πρέπει να καθοδηγείται από μια μετρική επίδοσης η οποία αποτιμάται πάνω στο σύνολο ελέγχου (test set) με τη χρήση διασταυρωμένης επικύρωσης (Cross Validation). Στην περίπτωση μας χρησιμοποιήθηκε το F1 test score. Η υλοποίηση της μεθόδου έγινε μέσω της συνάρτησης GridSearchCV της βιβλιοθήκης Sci-kit Learn

Κατά την τεχνική αυτή τα δεδομένα εκπαίδευσης (training set) χωρίζονται σε ένα σταθερό αριθμό πτυχών στον οποίο θα γίνει το Cross Validation. Σε κάθε επανάληψη της τεχνικής, χρησιμοποιείται μία πτυχή των δεδομένων για εκτίμηση των αποτελεσμάτων της εκπαίδευσης και οι εναπομείναντες πτυχές σαν δεδομένα εκπαίδευσης. Χρησιμοποιώντας αυτή τη μέθοδο έχουμε αποτελεσματικότερη εκπαίδευση και αποφεύγουμε το πρόβλημα της υπερπροσαρμογής (overfitting). Σε όλα τα πειράματα χρησιμοποιήθηκε 10-Φολδ ῎ροος ἄλιδατιον για τη διασφάλιση της σωστής εκπαίδευσης.

Στην συγκεκριμένη διπλώματική η βελτιστοποίηση πραγματοποιείται σε επίπεδο pipelines, δηλαδή στην εύρεση των καλύτερων υπερ-παραμέτρων τόσο για τους 16 ταξινομητές όσο και για τους αντίστοιχους μετασχηματιστές που προηγούνται αυτών, όπως αυτοί προέκυψαν στο προηγούυμενο βήμα.

Μάλιστα, αντί να χρησιμοποιήσουμε GridsearchCV, χρησιμοποιήσαμε GridsearchCV, που έχει αποδειχθεί ότι είναι $x5-10$ πιο γρήγορο (για περισσότερα δείτε το Figure 1.9). Εφόσον το project αυτό είναι αφιερωμένο στην παροχή ενός ολοκληρωμένου εξαντλητικού σχήματος βελτιστοποίησης,η κατανομή των πόρων δεν μπορεί να παραμεληθεί. Ο αλγόριθμος HalvingGridsearchCV αποτελεί μια στρατηγική αναζήτησης γνωστή ως "συνεχόμενη διαίρεση" (Successive Halving που (1) χρησιμοποιεί ένα υποσύνολο δεδομένων και λίγους πόρους στην αρχή της διαδικασίας για να βρεθούν μερικοί από τους συνδυασμούς παραμέτρων με την καλύτερη απόδοση (καλύτεροι υποψήφιοι) και στη συνέχεια **αυξάνει σταδιακά** τον όγκο των δεδομένων και τον αριθμό των πόρων που χρησιμοποιούντα καθώς εστιάζουμε μόνο στους καλύτερους υποψήφιους.

Βελτιστοποίηση no.3 : Εύρεση του καλύτερου hard συλλογικού Ταξινομητή (Ensemble)

Έπειτα από 2 βήματα βελτιστοποιήσεων έχουμε pipelines βελτιστοποιημένα τόσο σε επίπεδο επιλογής μετασχηματιστών όσο και σε επίπεδο υπερπαραμέτρων. Μπορούμε με κάποιοπ τρόπο να συνδυάσουμε την γνώση και τον τρόπο λειτουργίας που παρέχει το κάθε pipeline για να δημιουργήσουμε ένα συνδυασμό pipeline με ακόμα υψηλότερο f1 test score. Αυτό μπορεί να γίνει με την μέθοδο ensembling

Πολλοί ερευνητές έχουν διερευνήσει την τεχνική συνδυασμού των προβλέψεων πολλώνταξινομητών για τη δημιουργία ενός ενιαίου ταξινομητή (Breiman-1996, Clemen-1989,Perrone-1993, Wolpert—1992). Ο ταξινομητής που προκύπτει ονομάζεται συλλογικός ταξινομητής (ensemble classifier) και είναι γενικά περισσότερο ακριβής από κάθε ταξινομητή που συμμετέχει στην ομάδα σχηματισμού του.

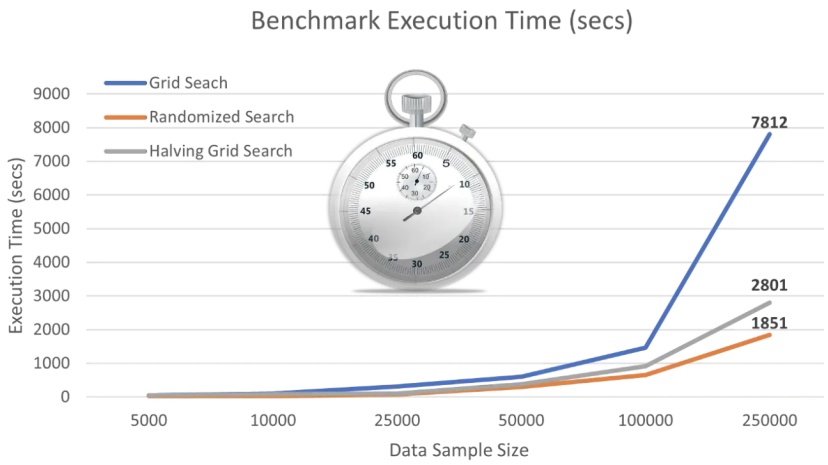


Figure 1.9. *HalvingGridsearchCV*

Η βασική ιδέα της συλλογικής ταξινόμησης είναι η στάθμιση διαφορετικών ταξινομητών και ο συνδυασμός τους σε ένα ενιαίο ταξινομητή, ο οποίος αποδίδει καλύτερα από κάθε ένα από τους επιμέρους ταξινομητές. Κατά τη λήψη μιας απόφασης οι άνθρωποι ακολουθούν την ίδια τεχνική, ακούγοντας διάφορες απόψεις και στη συνέχεια αξιολογώντας τις απόψεις αυτές για τη λήψη της τελικής απόφασης.

Στην παρούσα διπλωματική , επιχειρείται η βελτιστοποίηση της επιλογής των ταξινομητών που θα συμμετέχουν στον ευρέως χρησιμοποιούμενο **συνδυαστικό αλγόριθμο ψηφοφορίας** (voting algorithm) . Μία απλή προσέγγιση είναι η μέθοδος ¨σκληρής¨ πλειοψηφίας (hard majority voting), όπου μία τάξη αποτελεί πρόβλεψη μόνο αν την έχουν προβλέψει («ψηφίσει») τουλάχιστον $(k + 1)/2$ ταξινομητές . Στο διάγραμμα Figure 6.4 φαίνεται η σύγκριση ενός soft και ενός hard συλλογικού ταξινομητή.

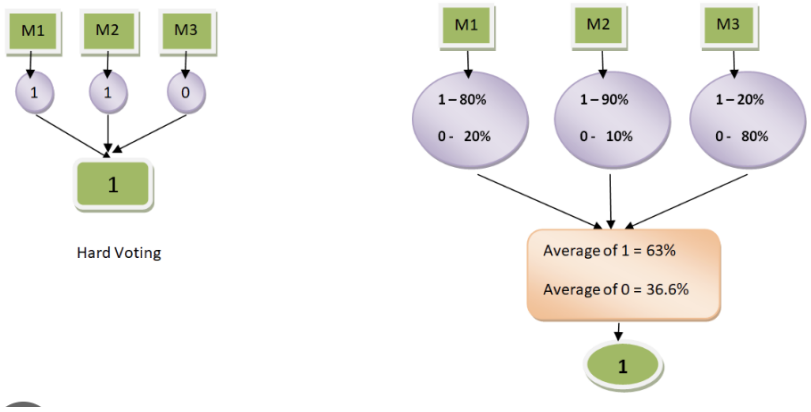


Figure 1.10. *Soft and Hard Voting Classifier*

Επεξηγησιμότητα στην μηχανική μάθηση με τον αλγόριθμο LIME

LIME (Τοπικό ερμηνευτικό μοντέλο-Agnostic Explainer) είναι ένας αλγόριθμος που μπορεί να εξηγήσει τις προβλέψεις οποιουδήποτε αλγόριθμου μηχανικής μάθησης. Όπως υποδηλώνει το όνομα, το LIME είναι αγνωστικό μοντέλο. Αυτό σημαίνει ότι το LIME θα μπορούσε να λειτουργήσει για οποιοδήποτε τύπο μοντέλου. Το μοντέλο είναι επίσης τοπικά ερμηνεύσιμο, πράγμα που σημαίνει ότι μπορούμε να εξηγήσουμε το μοντέλο χρησιμοποιώντας τοπικά αποτελέσματα αντί να εξηγήσουμε ολόκληρο το μοντέλο.

Ακόμα κι αν το μοντέλο που εξηγείται είναι ένα μαύρο κουτί, το LIME δημιουργεί ένα τοπικό γραμμικό μοντέλο γύρω από σημεία κοντά σε μια συγκεκριμένη θέση. Το LIME παρέχει ένα γραμμικό μοντέλο που

προσεγγίζει το μοντέλο κοντά σε μια πρόβλεψη αλλά όχι απαραίτητα συνολικά. Επίσης λειτουργεί τοπικά, δηλαδή παράγουμε την σημαντικότητα και τα βάρη $w1, ..., w_{n_{features}}$ για κάθε δείγμα $x_i \in X_{test}$ ξεχωριστά.

$$explanation(x) = [w_1, ..., w_{n_{features}}] = \arg\min_{g \in G} L(F, g, \pi_x) + \Omega(g) \quad (1.1)$$

Το μοντέλο εξήγησης για παράδειγμα x είναι το μοντέλο g (π.χ. μοντέλο γραμμικής παλινδρόμησης) που ελαχιστοποιεί την απώλεια L (π.χ. μέσο τετραγωνικό σφάλμα), το οποίο μετρά πόσο κοντά είναι η εξήγηση στην πρόβλεψη του αρχικού μοντέλου f (π.χ. ένα μοντέλο *xgboost*), ενώ η πολυπλοκότητα του μοντέλου $\Omega(g)$ διατηρείται σε χαμηλά επίπεδα (π.χ. λιγότερα χαρακτηριστικά). Το G είναι η οικογένεια των πιθανών εξηγήσεων, για παράδειγμα όλα τα πιθανά μοντέλα γραμμικής παλινδρόμησης. Το μέτρο εγγύτητας πξ ορίζει πόσο μεγάλη είναι η γειτονιά γύρω από το παράδειγμα x που θεωρούμε για την εξήγηση. Στην πράξη, το *LIME* βελτιστοποιεί μόνο το τμήμα απώλειας. Ο χρήστης πρέπει να προσδιορίσει την πολυπλοκότητα, π.χ. επιλέγοντας τον μέγιστο αριθμό χαρακτηριστικών που μπορεί να χρησιμοποιήσει το μοντέλο γραμμικής παλινδρόμησης.

Στην παρούσα διπλωματική παρουσιάζουμε την δική μας προσέγγιση, και το πως χρησιμοποιήσαμε τον LIME για να εξάγουμε συμπεράσματα για όλο τον πληθυσμού του συνόλου **ελέγχου**.

Αλγοριθμος 1.3: *Παραμετροποιημένος LIME αλγόριθμος για επεξήγηση μοντέλων μηχανικής μάθησης*

1. Υπολόγισε όλες τις $N = 11$ προβλέψεις και αποθήκευσε τες στο μοναδιαίο διάνυσμα y_pred.
2. Κράτα μόνο τις $M \leq N$ προβλέψεις που είναι σωστές, δηλαδή κρατάμε την $i-th$ πρόβλεψη αν ικανοποιείται η συνθήκη $y\_pred[i] = y\_test[i]$
3. Αρχικοποιούμε ένα αντικείμενο LIME το οποίο δέχεται ως ορίσματα τα X_train, feature names, class names.
4. Για $i \in \{0, 1, ...M\}$:
(α) Χρησιμοποίησε την συνάρτηση *explain_distance*() για τον υπολογισμό των βαρών w_i ως εξής

$$w_i = explain\_distance(X\_test_i)$$

όπου το w_i είναι μονοδιάστατο διάνυσμα μεγέθους $(n_{features},) = (741,)$ ενώ το w_{ij} δείχνει την ¨συνεισφορά¨ (ή βάρος) του $j-th$ χαρακτηριστικού στο $i-th$ δείγμα του συνόλου δοκιμών X_test, οπότε η ακόλουθη σχέση πρέπει να ικανοποιείται :

$$\sum_{i=1}^{n_{features}} w_{ij} = 1$$

5. Υπολογίζουμε τον μέσο όρο όλων των προβλέψεων για κάθε χαρακτηριστικό και προκύπτει το τελικό διάνυσμα συνεισφορών/βαρών w' ως εξής :

$$w'_j = \frac{w_{ij}}{M} \quad \forall i \in \{1, ..., M\}$$

1.4 Βαθιά Μάθηση (Deep Learning)

Περιγράφουμε το μοντέλο βαθιάς μάθησης, ένα Συνελεκτικό Νευρωνικό Δίκτυο γράφων (G-CNN) που υποστηρίζει τη διάγνωση της Δυσλεξίας ταξινομώντας πίνακες συσχέτισης των συνδεδεμένων περιοχών του εγκεφάλου που προέκυψαν μετά την προεπεξεργασία του συνόλου δεδομένων fMRI

1.4.1 Γενική περιγραφή του μοντέλου

Σε αυτό το έργο εφαρμόσαμε το μοντέλο βαθιάς μάθησης του G-CNN, όπως περιγράφει η (Παρισοτ) στις εργασίες της. Εφαρμόστηκε ένα αραιό γράφημα για να αναπαραστήσει το δίκτυο αλληλεπίδρασης του συνόλου δεδομένων των 58 συμμετεχόντων με βάση τα φαινοτυπικά δεδομένα (ηλικία, φύλο). Ακριβέστερα :

- **Κόμβοι** V : Κάθε κόμβος αντιπροσωπεύει έναν μόνο συμμετέχοντα και αντιστοιχεί στο μοναδιαίο διάνυσμα χαρακτηριστικών D_i
- **Βάρη και Ακμές** W,E : Το βάρος w_i κάθε ακμής$e = (v_1, v_2)$ που συνδέει 2 κόμβους, αντιπροσωπεύει το βαθμό ομοιότητας μεταξύ 2 συμμετεχόντων και υπολογίζεται με βάση τα διαθέσιμα φαινοτυπικά δεδομένα

Εστιάσαμε στην αξιοποίηση όλων των διαθέσιμων δεδομένων που παρέχονται από την βάση δεδοδμένων (δεδομένα fMRI, φαινοτυπικά). Στη συνέχεια, το αραιό γράφημα εισάγεται στο μοντέλο G-CNN ως το πρώτο επίπεδο, ενώ εκπαιδεύεται με (ημι)-εποπτευόμενο (semi-supervised) τρόπο χρησιμοποιώντας ένα σύνολο **εκπαίδευσης** με ετικέτα και ένα σύνολο **δοκιμών** χωρίς ετικέτα. Αυτό που κάνει αυτό το μοντέλο βαθιάς μάθησης ξεχωριστό είναι η ικανότητά του να χρησιμοποιεί τόσο **απεικονιστικές** όσο και **μη απεικονιστικές** πληροφορίες που πιστεύουμε ότι είναι χρήσιμες για τη διάγνωση της δυσλεξίας, εκφράζοντας την **ομοιότητα** μεταξύ των συμμετεχόντων. Με αυτόν τον τρόπο, σκοπεύουμε να επιτύχουμε αποτελέσματα υψηλότερης ακρίβειας σε σύγκριση με τα μοντέλα μηχανικής μάθησης που χτίσαμε προηγουμένως

1.4.2 Δομή του γράφου

Έχουμε ($n_{samples} = 58$) συμμετέχοντες, ενώ ο στόχος είναι να προβλέψουμε την κατάσταση κάθε συμμετέχοντα βάσει της απεικόνισης φMPI και των φαινοτυπικών δεδομένων συνδυαστικά (που συνδέουν τα άτομα του πληθυσμού).

- Το σύνολο των συμμετεχόντων αναπαρίσταται ως ένα αραιό γράφημα $G = (V, E)$, συνδεδεμένο με σταθμισμένες ακμές, όπου ο πίνακας βαρών W είναι ο «διπλανός πίνακας» (adjacent table του γραφήματος.
- Ο i-th συμμετέχων αντιστοιχεί σε έναν συγκεκριμένο κόμβο γραφήματος «1-1», ο οποίος συσχετίζεται με ένα ενιαίο διάνυσμα χαρακτηριστικών D_i που εξάγεται από το σύνολο δεδομένων fMRI (όπως έχουμε δείξει)
- Το σύνολο ακμών E του γραφήματος αντιπροσωπεύει τις ομοιότητες μεταξύ των συμμετεχόντων βάσει των φαινοτυπικών δεδομένων

Στην παρούσα προσέγγιση βαθιάς μάθησης, το πρόβλημα διάγνωσης της δυσλεξίας μετατρέπεται σε πρόβλημα ταξινόμησης κόμβων. Ο στόχος είναι να δοθεί μια ετικέτα $l \in 0, 1, 2$ σε κάθε κόμβο, όπου l σημαίνει την κατάσταση κάθε συμμετέχοντα, $l = 0$ για τους συμμετέχοντες **ελέγχου** (typical reader), $l = 1$ για τους συμμετέχοντες με **ορθογραφική διαταραχή** και $l = 2$ για συμμετέχοντες με **δυσλεξία** Η στρατηγική εκπαίδευσης που ακολουθείται εδώ είναι **εποπτευόμενη**, που σημαίνει ότι **όλοι** οι κόμβοι επισημαίνονται και εισάγονται στο μοντέλο G-CNN για την εκπαίδευση.

Προκειμένου αυτή η προσέγγιση να επιτύχει υψηλότερα και ακριβή αποτελέσματα σε σύγκριση με την προσέγγιση μηχανικής μάθησης, πρέπει να κατασκευάσουμε το γράφημα με **ακρίβεια**, ακόμη και όταν χρησιμοποιούμε μόνο **δύο** 2 φαινοτυπικά χαρακτηριστικά. Ένα ανεπαρκώς καθορισμένο γράφημα μπορεί να έχει χειρότερη απόδοση από έναν γραμμικό ταξινομητή. Υπάρχουν 2 στοιχεία που επικυρώνουν τη σωστή κατασκευή του γραφήματος:

- Το μοναδιαίο διάνυσμα χαρακτηριστικών D_i

- Ο τρόπος που συνδέονται οι κόμβοι-συμμετέχοντες μεταξύ τους, δηλαδή η ύπαρξη (ή όχι) άκρης μεταξύ δύο κόμβους και το βάρος του. Αυτός είναι ο τρόπος για να μοντελοποιήσουμε και να εκφράσουμε την **ομοιότητα** δύο κόμβων-συμμετεχόντων.

1.4.3 Το σύνολο ακμών E του γράφου G

Στα κλασικά συνελεκτικ΄λα μοντέλα CNN, οι γειτονιές των εικονοστοιχείων επηρεάζουν τη διαδικασία συνέλιξης. Στα μοντέλα G-CNN, η επιλογή των E, W του γραφήματος επηρεάζει επίσης τη διαδικασία συνέλιξης. Επομένως, θα πρέπει να χρησιμοποιήσουμε τα διαθέσιμα φαινοτυπικά δεδομένα με τέτοιο τρόπο που να εκφράζουν και να εξηγούν πιο αποτελεσματικά τις ομοιότητες μεταξύ των μεταξύ των $n_{samples}$ κόμβων-συμμετεχόντων του γραφήματος. Τα διαθέσιμα φαινοτυπικά δεδομένα είναι 2 (φύλο, ηλικία)

Φύλο (Gender)

Το **φύλο** επιλέγεται επειδή οι άνδρες διαγιγνώσκονται με δυσλεξία πιο συχνά από τις γυναίκες, ακόμη και σε επιδημιολογικά δείγματα. Η ίδια έρευνα έδειξε ότι η ταχύτητα επεξεργασίας βοηθά στην εξήγηση της διαφοράς φύλου στη δυσλεξία [Döpfner et al., 2008]

Ηλικία (Age)

Η ηλικία επιλέγεται επειδή όπως και κάθε άλλο φυσιολογικό χαρακτηριστικό, ο δυσλεξικός **γνωστικός** και **νευρικός** φαινότυπος είναι πιθανό να επηρεάσει τη διαδικασία γήρανσης [ref, 2022a]

Το βάρος $w(u, v)$ μεταξύ των κόμβων u και v μπορεί να οριστεί ως εξής :

$$w(\boldsymbol{u}, \boldsymbol{v}) = sim(x(\boldsymbol{u}), x(\boldsymbol{u})) \cdot \sum_{i=0}^{H} \gamma(M_i(\boldsymbol{u}), M_i(\boldsymbol{v}))$$

Παράγοντας	Επεξήγηση
$\boldsymbol{sim}(x(v), x(u))$	ο συντελεστής ομοιότητας μεταξύ του διανύσματος χαρακτηριστικών $x(u)$ και $x(u)$ των κόμβων-συμμετεχόντων u, v. Για παράδειγμα, η συνάρτηση εκθετικής ομοιότητας που χρησιμοποιείται είναι : $$sim(x(\boldsymbol{u}), x(\boldsymbol{v})) = e^{\frac{[\rho(x(\boldsymbol{u}), x(\boldsymbol{v})]^2}{2\sigma^2}}$$ όπου $\boldsymbol{\rho}$ είναι η συσχετίση και $\boldsymbol{\sigma}$ καθορίζει το μέγεθος του πυρήνα
H	Ο συνολικός αριθμός φαινοτυπικών δεδομένων διαθέσιμων για τον υπολογισμό του βάρους της ακμής. Εδώ $H = 2$
M_i	Η συνάρτηση που χρησιμοποιείται για τη μέτρηση των φαινοτυπικών πληροφοριών π.χ. $M_i = 9$ ώ
$\boldsymbol{\gamma}$	Η συνάρτηση για σύγκριση φαινοτυπικών πληροφοριών • **Κατηγορικά Χαρακτηριστικά** (πχ. Φύλο) $\gamma(M_i(\boldsymbol{u}), M_i(\boldsymbol{v}))$ = Kronecker $\delta(M_i(\boldsymbol{u}), M_i(\boldsymbol{v})) = \begin{cases} 1, & M_i(\boldsymbol{u}) = M_i(\boldsymbol{v}) \\ 0, & M_i(\boldsymbol{u}) \neq M_i(\boldsymbol{v}) \end{cases}$ • **Αριθμητικά Χαρακτηριστικά** (πχ. Ηλικία) $\gamma(M_i(\boldsymbol{u}), M_i(\boldsymbol{v})) = \begin{cases} 1, & \lvert M_i(\boldsymbol{u}) - M_i(\boldsymbol{v})\rvert < \partial \\ 0, & \lvert M_i(\boldsymbol{u}) - M_i(\boldsymbol{v})\rvert \geq \partial \end{cases}$ όπου ∂ είναι ένα κατώφλι (που μπορεί να βελτιστοποιηθεί) που ορίζει πότε πχ 2 ηλικίες θεωρούνται "κοντά"

Επομένως, 2 κόμβοι γραφήματος-συμμετέχοντες συνδέονται με μια ακμή με βάρος $w_0 \in \{0, 1, 2\}$, με βάση το πόσες από τις φαινοτυπικές τους πληροφορίες θεωρούνται "κοντά". Στη συνέχεια, αυτό το w_0 πολλαπλασιάζεται με τον συντελεστή ομοιότητας κόμβων, δίνοντας το τελικό βάρος w της ακμής ως $w = sim(x(\boldsymbol{u}), x(\boldsymbol{v})) \cdot w_0$

1.4.4 Η αρχιτεκτονική του μοντέλου G-CNN

Η αρχιτεκτονική του μοντέλου μας απεικονίζεται στο Figure 1.11). Το μοντέλο αποτελείται από ένα πλήρως συνελικτικό CCN με L κρυφά στρώματα (hidden layers) που ενεργοποιούνται χρησιμοποιώντας τη λειτουργία Rectified Linear Unit (ReLU). Το επίπεδο εξόδου ακολουθείται από μια λειτουργία ενεργοποίησης softmax (επειδή έχουμε πρόβλημα ταξινόμησης). Το γράφημα εκπαιδεύεται χρησιμοποιώντας ολόκληρο το γράφημα του πληθυσμού ως είσοδο. Επιπλέον, χρησιμοποιούμε μια συνάρτηση απώλειας διασταυρούμενης εντροπίας (cross entropy loss) για τη διαδικασία βελτιστοποίησης.

1.4.5 Συνάρτηση Κόστους (Loss), Αλγόριθμο Βελτιστοποίησης (Optimizer), Μετρικές Αξιολόγησης (Metrics)

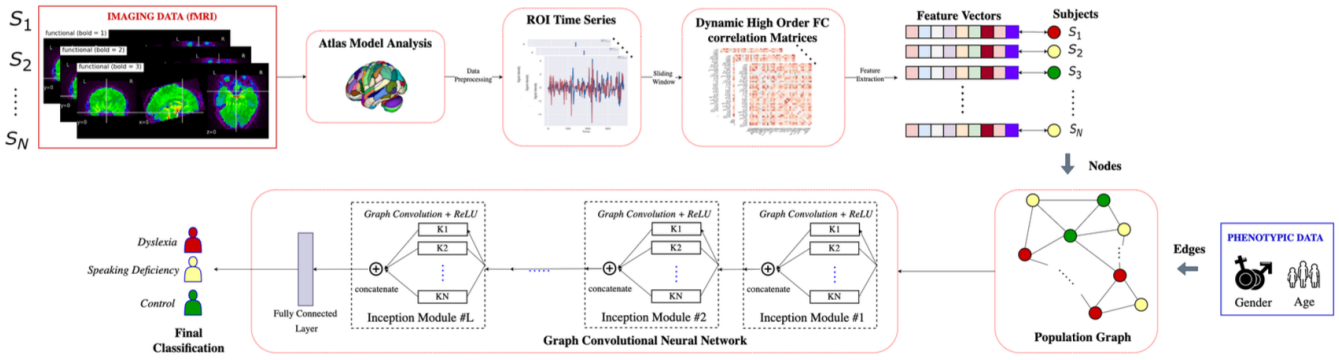


Figure 1.11. *Η αρχιτεκτονική του μοντέλου G-CNN*

Optimizer	**keras.optimizers**.*Adam*
Loss Function	**keras.losses**.*SparseCategoricalCrossentropy(from_logits = True)*
Metics	**keras.metrics**.*SparseCategoricalAccuracy(name = "acc")*

Για να μάθει το G-CNN μοντέλο τις βέλτιστες παραμέτρους, αξιοποιούμε έναν αλγόριθμο βελτιστοποίησης (optimizer) των παραμέτρων, με βάση τον υπολογισμό του ανάδελτα (gradient) $\nabla_\theta L(y, \hat{y})$ για την εύρεση ενός τοπικού ελάχιστου. Χρησιμοποίησαμε τον ADAM optimizer (ADAptive Momentum) ο οποίος υπολογίζει όσα και ο ADAGRAD και επίσης υπολογίζει για κάθε παράμετρο το ρυθμό μάθησης και τις αλλαγές της ορμής (Momentum). Ο Adam είναι ένας υπολογιστικά αποδοτικός αλγόριθμος, εύκολος στην υλοποίηση και οι απαιτήσεις του για μνήμη είναι περιορισμένες. Λειτουργεί καλά σε μεγάλα σετ δεδομένων με μεγάλες παραμέτρους καθώς και σε προβλήματα με θορυβώδεις ή αραιές (sparse) κλίσεις, όπως στην παρούσα διπλωματική.

Για την συνάρτηση κόστους : Η cross-εντροπία αποτελεί το μέσο ελάχιστο μέγεθος κωδικοποίησης της πληροφορίας του να επικοινωνήσουμε ένα γεγονός από μία κατανομή πιθανότητας σε μία άλλη. Η διακριτή συνάρτηση κόστους cross-entropy (αναφέρεται και ως η αρνητική λογιστική συνάρτηση πιθανοφάνειας (negative log likelihood) χρησιμοποιείται όταν είναι επιθυμητή μια πιθανοτική ερμηνεία των αποτελεσμάτων. Επειδή έχουμε αραιές κλίσεις χρησιμοποιούμε το *SparseCategoricalCrossentropy*

1.5 fMRI & Δεδομένα

Για τους σκοπούς της διπλωματικής εργασίας χρησιμοποιήθηκε ένα σύνολο δεδομένων ανοιχτού κώδικα που ελήφθη από το OpenNeuro. Προκειμένου να δημιουργηθεί το σύνολο δεδομένων, ελήφθησαν σαρωμένες εικόνες του εγκεφάλου τόσο από ασθενείς με διάγνωση σχιζοφρένειας όσο και από υγιείς εξεταζόμενους.

Η βάση δεδομένων που χρησιμοποιήθηκε ονομάζεται **MRI Lab Graz**, ενώ παρακάτω εξετάζουμε τα παιδιά-συμμετέχοντες καθώς και την διαδικασία επιλογής. Εξηγούμε πως κατεβάσαμε τα δεδομένα και πως το [Banfi et al., 2020] προεπεξεργάστηκε τα δεδομένα fMRI. Επίσης παρουσιάζουμε την κατανομή των φαινοτύπων (*φύλο, ηλικία*).

1.5.1 Συμμετέχοντες και έρευνα

[Banfi et al., 2020] πραγματοποίησε την έρευνα σε πλήρη συμφωνία με την τελευταία έκδοση του Declaration of Helsinki και της εθνικης νομοθεσίας και είναι νομικά αποδεχτό από την επιτροπή ηθών του **Πανεπιστημίου Γραζ** στην Αυστρία. Η διαδικασία επιλογής συμμετεχότνων περιγράφεται παρακάτω.

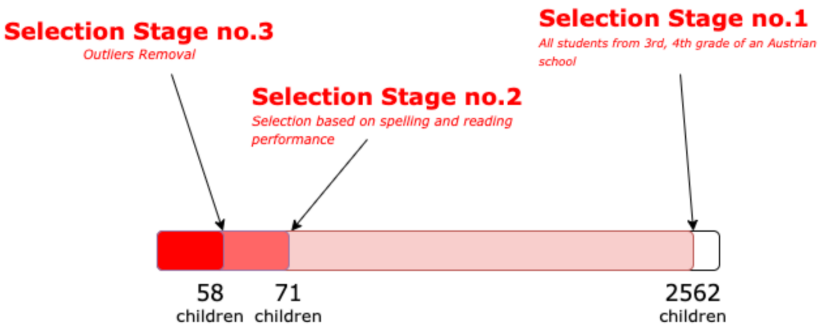


Figure 1.12. *Επιλογή συμμετεχόντων ανά στάδιο*

Group	Stage no.1	Stage no.2	Stage no.3
Spelling Disorder (SpD)		21	
Dyslexia (DL)		23	
Skilled (TD)		27	
Total	2562	71	58

Figure 1.13. *Επιλογή συμμετεχόντων ανά στάδιο και ανά κλάση*

1ο Στάδιο Επιλογής

Συνολικά 2562 παιδιά ηλικίας 9-13 (από 3η έως 4η Δημοτικού) επιλέχθηκαν να εξεταστούνε σε ενα 3-μερών τεστ (1) Τυποποιημένο τεστ στην τάξη για την εξέταση της ευχέρειας ανάγνωσης προτάσεων [Wimmer et al., 2014] (2) Τυποποιημένο τεστ στην τάξη για την εξέταση της ευχέρειας στην ορθογραφία [Müller and R., 2004] (3) Εξατομικευμένο τυποποιημένο τεστ ευχέρειας ανάγνωσης λέξεων και ψευδολέξεων διάρκειας ενός λεπτού [Moll et al., 2014]

2ο Στάδιο Επιλογής

Από αυτό το μεγάλο δείγμα τνων 2562 ο [Banfi et al., 2020] επέλεξε τρεις ομάδες με βάση την απόδοση τους στην **ορθογραφία** και την **ανάγνωση** (μέση απόδοση και στα 3 τεστ ανάγνωσης)

GROUP	**Spelling performance**	**Mean Reading performance (3 tests)**
Spelling Disorder (SpD)	$performance \leq 20\%$	$performance_{mean} \geq 25\%$
Dyslexia (DL)		$performance_{2outof3} \leq 20\%$ and $performance_{3rd} \leq 43\%$
Skilled (TD)	$performance \in (25\%, 85\%()$	$performance_{mean} \in (25\%, 85\%()$

Όλα τα παιδιά-συμμετέχοντες πρέπει να τηρούν με τις ακόλουθες προϋποθέσεις

1. Η πρώτη τους γλώσσα να είναι τα γερμναικά
2. Απαιτείται τουλάχιστον $IQ_{non-verbal} \geq 85$ στο IX τεστ [Weiß and R., 2006]
3. Δεν έχουν εντοπιστεί αισθητηριακά ή νευρολογικά ελλείμματα

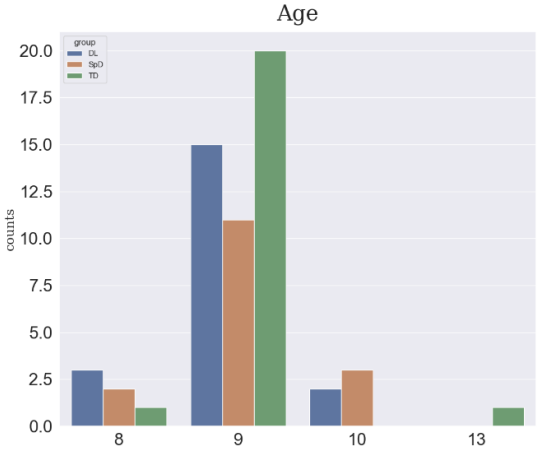


Figure 1.14. *i*

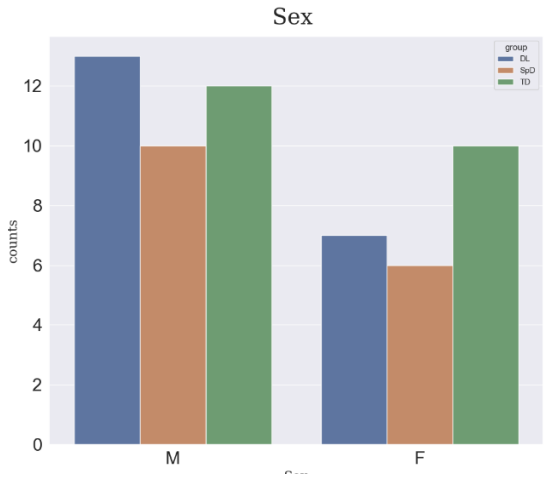


Figure 1.15. *ύ*

Figure 1.16. *Κατανομή φαινοτυπικών δεδομένων των συμμετεχόντων*

4. καμία κλινική διάγνωση ΔΕΠΥ

5. Βαθμολογία πάνω από το όριο σε ένα τυποποιημένο γονικό ερωτηματολόγιο για ελλείμματα προσοχής [Döpfner et al., 2008]

Αφού έγινε μια πρώτη επιλογή, παρατηρήθηκε ότι μόνο 70% των παιδιών στο δείγμα Spelling Disorder (SpD) και 90% στην ομάδα Dyslexia (DL) συμμορφώνονταν με τις γερμανικές διαγνωστικές οδηγίες για επίσημη κλινική διάγνωση [Galuschka et al., 2016]. Συεμπερασματικά, επιλέχθηκαν 71 παιδιά

3ο Στάδιο Επιλογής

Από αυτά τα 71 παιδιά που επιλέχθηκαν, αρκετά εξαιρέθηκαν από την ανάλυση και το τελικό σύνολο δεδομένων για τους ακόλουθους λόγους:

- 12 (από τα 71) αποκλείστηκαν λόγω υπερβολικής κίνησης κατά τη διάρκεια της συνεδρίας fMRI.
- 1 (από τα 71) αποκλείστηκε επίσης επειδή το οπτικό πεδίο (ΦΟ") είχε οριστεί λανθασμένα, οδηγώντας στην κοπή ενός σημαντικού τμήματος του μετωπιαίου λοβού

1.5.2 Κατανομή Φαινοτύπων

Στη βάση δεδομένων μας, υπάρχει ένα αρχείο *.tσ* που περιγράφει 3 διαφορετικά φαινοτυπικά δεδομένα. Αυτά είναι: **Ηλικία, Φύλο** και **Ομάδα Νοσημάτων**. Ο φαινότυπος Disease Group είναι η ετικέτα κλάσης που χρησιμοποιείται για τα πλαίσια μηνχαικής βαθιάς μάθησης. Τα υπόλοιπα 2, **Αγε** και **Γενδερ** χρησιμοποιούνται για την κατασκευή του αρχικού γραφήματος του μοντέλου **Γ-"ΝΝ** στα πλάισια της βαθιάς μάθησης.

Παρατηρούμε ότι το σύνολο δεδομένων είναι ισορροπημένο μεταξύ των δύο **φύλων**, ενώ ο φαινότυπος **Αγε** είναι μη ισορροπημένος. Οι ηλικίες των συμμετεχόντων κατανέμονται μεταξύ 8 και 13, με μέση τιμή $\mu_{age} = 9.05$ και τυπική απόκλιση $\sigma_{age} = 0.68$.

1.5.3 Πειραματική Διαδικασία

Η όλη διαδικασία μπορεί να διαρκέσει έως και 1040 δευτερόλεπτα για κάθε παιδί-συμμετέχων (κατά μέσο όρο, όπως εξηγείται αργότερα) και συνολικά έως $58 \times 1040 \sim 60320sec \sim 17$ ώρες για όλα τα παιδιά-

συμμετέχοντες. Συνοπτικά μπορεί να αποτυπωθεί στο διάγραμμα που δημιουργήσαμε Figure 1.17. Πιο αναλυτικά :

1. Κάθε παιδί καλείται να διαβάσει 120 λέξεις (λέξεις 60 που συμβολίζονται ως "W" και 60 ψευδο-λέξεις που συμβολίζονται ως "PH"). Οι ψευδο-λέξεις κατασκευάστηκαν από την αρχική λέξη με ανταλλαγή ενός φωνολογικά πανομοιότυπου γραφήματος, ενώ αποτελούνται από 3 έως 8 γράμματα

2. Και οι 120 λέξεις χωρίστηκαν σε 3 διαδοχικές σειρές των 40 λέξεων η καθεμία, χωρισμένες με μικρά διαλείμματα 3-5 λεπτών, ώστε τα παιδιά να μην κουραστούν και να παραμείνουν συγκεντρωμένα

3. Σε κάθε επανάληψη : Κάθε λέξη παρουσιάζεται με λευκή γραμματοσειρά σε μαύρο φόντο για 3 δευτερόλεπτα σε σχέδιο σχετικό με την εκδήλωση. Αφού εξαφανιστεί, εφμανίζεται ένας σταυρός στερέωσης για 2000 και 6000 ms (μέσος όρος = 4000 ms)

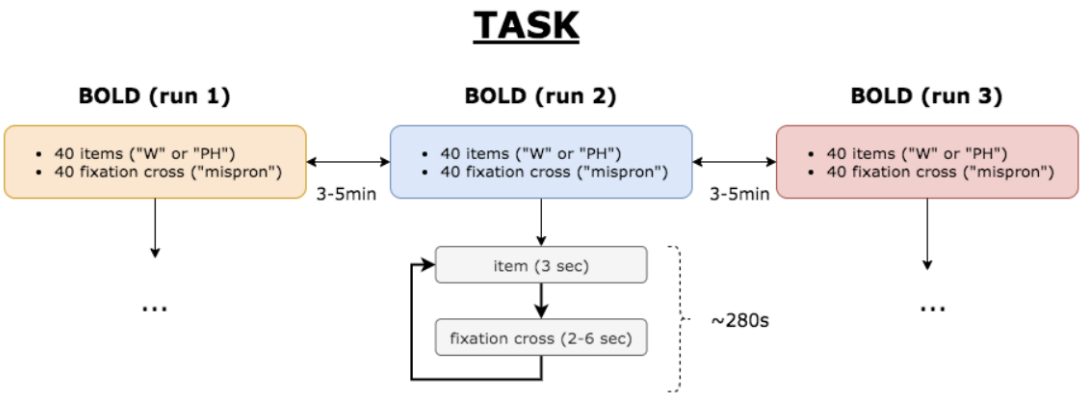


Figure 1.17. *Η δομή της πειραματικής δικαδικασίας*

1.6 Απόκτηση Δεδομένων fMRI

Σε κάθε ένα από τα 58 παιδιά δόθηκαν αναλυτικές οδηγίες σε ένα αθόρυβο δωμάτιο, ώστε να αισθάνονται ήρεμα και έτσι να διασφαλιστεί η σωστή εκτέλεση του πειράματος. Κατά τη διάρκεια του πειράματος, πραγματοποιήθηκε απεικόνιση σε σαρωτή Skyra 3.0T (Siemens Healthineers, Erlangen, Γερμανία) χρησιμοποιώντας ένα πηνίο κεφαλής 20 καναλιών. Καταγράφηκαν δύο (2) τύποι εικόνων:

- Υψηλής ποιότητας 3D-T1 MPRAGE **ανατομική απεικόνιση εγκεφάλου** (TR = 1600 ms, TE = 1,81 ms, FOV = 224 mm, $angle_{flip}$ = 8 degrees, 176 slices, voxel resolution $1 \times 1 \times 1$ mm^3) . Παράδειγμα αποτελεί το Figure 1.18

- Χαμηλότερης ποιότητας BOLD-sensitive T2*-weighted **λειτουργικές απεικονίσεις εγκεφάλου** were acquired using a single shot gradient-echo EPI pulse sequence (TR = 2340 ms, TE = 33 ms, FOV = 192 mm, $angle_{flip}$ = 90 degrees, 34 slices with 0.3 mm gap, voxel resolution $3 \times 3 \times 3 mm^3$, descending acquisition order) . Παράδειγμα αποτελεί το Figure 1.19

Προκειμένου οι σαρώσεις του εγκεφάλου και οι εικόνες να είναι σταθερές και όχι θολές, η κίνηση του κεφαλιού σταθεροποιήθηκε χρησιμοποιώντας σταθερές στηρίξεις γύρω από το κεφάλι. Οι λεκτικές απαντήσεις των συμμετεχόντων καταγράφηκαν μέσω μικροφώνου συμβατού με μαγνητική τομογραφία (FOMRI-III, OptoacousticsLtd., Moshav Mazor, Ισραήλ). Τα ερεθίσματα παρουσιάστηκαν χρησιμοποιώντας την Software Presentation (Neurobehavioral Systems, Albany, CA) [Banfi et al., 2020].

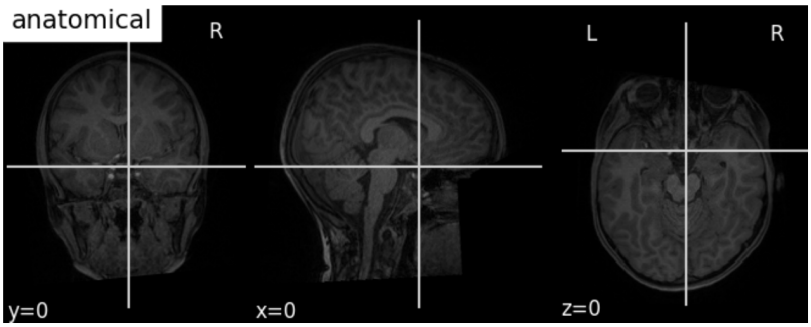


Figure 1.18. *Υψηλής ποιότητας 3D-T1 MPRAGE ανατομική απεικόνιση εγκεφάλου*

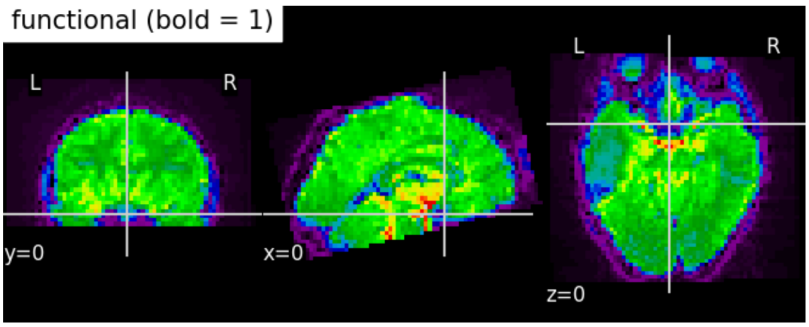


Figure 1.19. *Χαμηλής ποιότητας BOLD-sensitive T2*-weighted λειτουργικές απεικονίσεις εγκεφάλου*

1.7 Προεπεξεργασία Δεδομένων fMRI

Kind of Image	Image Shape
3D Anatomical T1w	$176 \times 224 \times 224$
4D Functional (3 BOLDS) T2w	$(64 \times 64 \times 34) \times 130$

Για την προεπεξεργασία της υψηλής ποιότητας **ανατομική απεικόνιση** T1-weigthed (T1w) , εφαρμόζονται οι εξής τεχνικές με την σειρά : *Διόρθωση για ανομοιομορφία έντασης, Απογύμνωση κρανίου με χρήση Νιπυπε, Χωρική Καυονικοποίηση, Τμηματοποίηση ιστού εγκεφάλου*

Για την προεπεξεργασία των χαμηλής ποιότητας **λειτουργικών απεικονίσεων** T2*-weighted (T2w*) , εφαρμόζονται οι εξής τεχνικές με την σειρά : *Απογύμνωση κρανίου, Συνεγγραφή, Εκτίμηση παραμέτρων κίνησης κεφαλής, Διόρθωση τομής-χρόνου, Επαναδειγματοληψία της ΒΟΛΔ χρουοσειράς, υπολογισμός χρονοσειρών των confounds, Διόρθωση θορύβου, ογκομετρική επαναδειγματοληψία, Εξομάλυνση*

1.8 Εξαγωγή Χαρακτηριστικών από fMRI

Έχουμε 58 παιδιά-συμμετέχοντες. Για τον κάθε έναν έχουμε τα εξής δεδομένα. Την 1 ανατομική απεικόνιση εγεκφάλου με $(176 \times 224 \times 224)$ σε μορφή "nii.gz". Τις 3 λειτουργικές εικόνες $3 \times (64 \times 64 \times 64 \times 130)$ σε μορφή "nii.gz"

Σε κάθε ένα από αυτά τα 3 σεσσιονς συναντάμε τις ακόλουθες 3 καταστάσεις : "W" για κανονική λέξη, "PH" για ψευδολέξη και "mispron" για ενδιάμεση παύση, ενώ ο χρόνος επανάληψης είναι $tr = 3$

Αν και τα δεδομένα έχουν ήδη υποβληθεί σε **προεπεξεργασία**, στόχος μας είναι να παράγουμε **πίνακες συσχέτιοης** που θα εισαχθούν ως είσοδος στα μοντέλα μηχανικής μάθησης και βαθιάς μάθησης. Υπάρχουν πολλά διαφορετικά είδη μετα-ανάλυσης που μπορούν να εφαρμοστούν για την εξαγωγή αυτών των χαρακτηριστικών (πίνακες συσχέτισης). Εν συντομία, αυτά τα είδη είναι :

- Ανάλυση πρώτου επιπέδου (GLM First Level Analysis)
- Λεξικό Μάθησης Ανάλυση (Dictionary Learning Analysis)
- Μοντέλο Ανάλυσης Άτλαντα (Atlas Model Analysis)

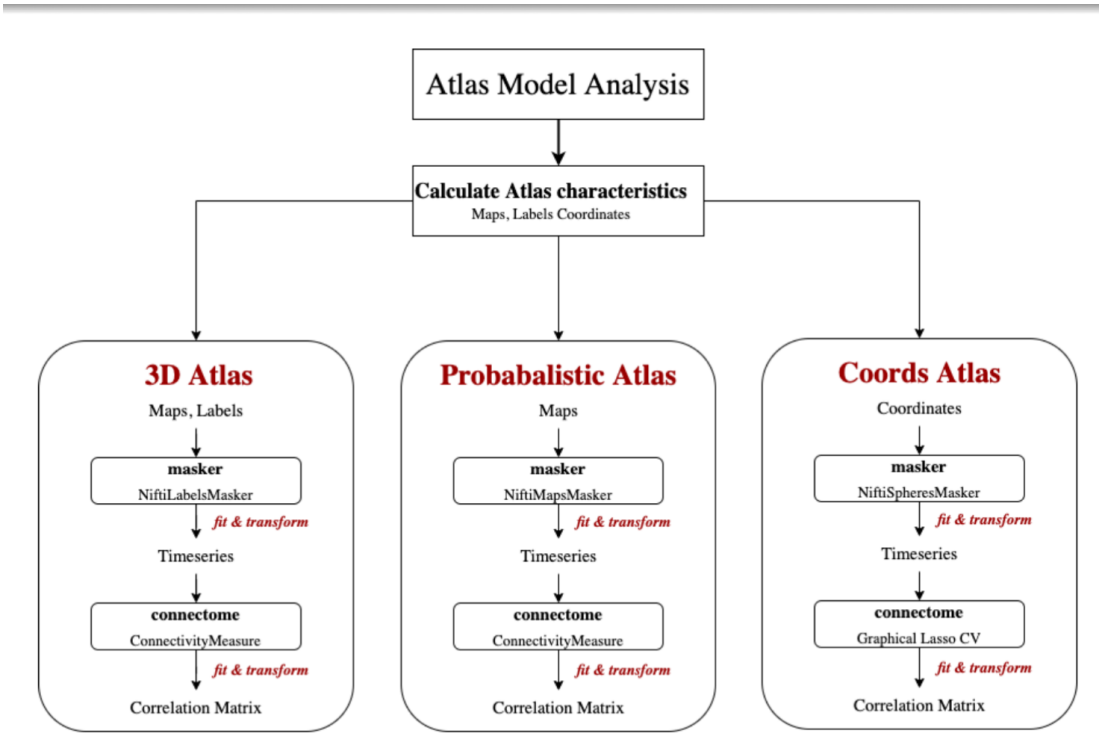


Figure 1.20. *Τεχνικές Εξαγωγής Χαρακτηριστικών για δεδομένα fMRI*

Στην παρούσα διπλωματική εργασία, οι πίνακες συσχέτισης που χρησιμοποιούνται είναι αυτοί που παράγονται από το *Μοντέλο Ανάλυσης Άτλαντα*.

1.8.1 Μοντέλο Ανάλυσης Άτλαντα

Οι χάρτες ολόκληρου του εγκεφάλου (whole-brain maps) μπορούν να κρύψουν σημαντικές λεπτομέρειες σχετικά με τις επιπτώσεις που μελετάμε. Μπορεί να βρεθεί ότι μια επίδραση του ασύμφωνου-σύμφωνου (incongruent-congruent) είναι σημαντική, αλλά στην πραγματικότητα θα μπορούσε να συμβαίνει επειδή το ασύμφωνο είναι μεγαλύτερο πό το σύμφνο ή επειδή το σύμφωνο είναι πολύ πιο αρνητικό από το ασύμφωνο, ή κάποιος συνδυασμός των 2. Ο μόνος τρόπος να προσδιορίσουμε την κατεύθυνση της επίδρασης είναι μέσω των **περιοχών Ενδιάφεροντος** (Regions of Interest (ROI)). Έτσι, πλέον δεν μελετάμε τον εγκέφαλος ως ολοκληρωτική οντότητα, αλλά αντιθέτως εστιάζουμε σε περιοχές του με σημαντική δραστηριότητα ή σε περιοχές όπως αυτές έχουν οριστεί από συγκεκριμένους άτλαντες όπως θα δούμε παρακάτω.

Υπάρχει πληθώρα διαφορετικών άτλαντων που μπορούν να χρησιμοποιηθούν για την εξαγωγή των περιοχών ενδιαφέροντος, οι οποίοι παρατίθενται παρακάτω. Ο κώδικάς μας είναι δομημένος έτσι ώστε ο άτλαντας να επιλέγεται δυναμικά, ενώ η βιβλιοθήκη πψτηον που χρησιμοποιείται είναι η nilearn.datasets. Ωστόσο, για τους σκοπούς αυτής της διπλωματικής εργασίας θα χρησιμοποιηθεί μόνο ο άτλας εκμάθησης λεξικών πολλαπλών θεμάτων Multi-Subject Dictionary Learning (MSDL), ο οποίος έχει 39 διακριτές εγκεφαλικές περιοχές)

Kind	Atlas Name	Method
3D	cortical	**datasets.***fetch_atlas_harvard_oxford('cort-maxprob-thr25-2mm')*
3D	subcortical	**datasets.***fetch_atlas_harvard_oxford('sub-maxprob-thr25-2mm')*
3D	yeo7	**datasets.***fetch_atlas_yeo_2011()*
3D	aal	**datasets.***fetch_atlas_aal()*
Probabalistic	aal	**datasets.***fetch_atlas_msdl()*
Coords	seitzman	**datasets.***fetch_coords_seitzman_2018()*
Coords	power	**datasets.***fetch_coords_power_2011()*
Coords	dosenbach	**datasets.***fetch_coords]_dosenbach_2010()*

Με βάση την βιβλιοθήκη nilearn, ο άτλας MSDL είναι ένας πιθανολογικός άτλαντας. Αυτού του είδους οι άτλαντες ορίζουν μαλακά τμήματα του εγκεφάλου στα οποία οι περιοχές μπορεί να επικαλύπτονται. Σε αντίθεση με τους ντετερμινιστικούς άτλαντες, ένα voxel μπορεί να ανήκει σε παραπάνω από μία. Αυτοί οι άτλαντες αντιπροσωπεύονται από εικόνες 4D όπου τα 3D στοιχεία, που ονομάζονται επίσης «χωρικοί χάρτες», στοιβάζονται κατά μήκος μιας διάστασης (συνήθως της 4ης διάστασης). Σε κάθε περιοχή 3D, η τιμή σε ένα συγκεκριμένο voxel υποδεικνύει πόσο ισχυρά σχετίζεται αυτό το voxel με αυτό την περιοχή. Η οπτικοποίηση ενός πιθανολογικού άτλαντα απαιτεί την οπτικοποίηση των διαφορετικών χαρτών που τον συνθέτουν. Εδώ απεικονίζουμε τον άτλαντα MSDL με ¨περιγράμματα¨, που σημαίνει ότι τα ROI εμφανίζονται ως περιγράμματα που οριοθετούνται από χρωματιστές γραμμές, όπως φαίνεται στο Figure 1.21. Τέλος οπτικοποιήσαμε τα ROI και με μορφή κόμβων, όπου κάθε κόμβος αντιπροσωπεύει το κέντρο της εκάστοτε περιοχής όπως φαίνεται στο Figure 1.22

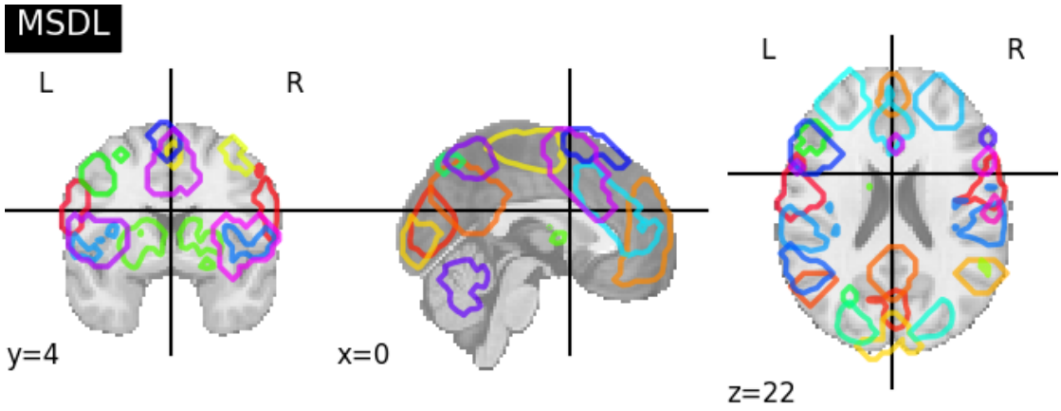


Figure 1.21. *MSDL Άτλαντας σε x, y,z άξονες με τα αντίστοιχα επικαλυπτόμενα ¨περιγράμματα¨*

1.8.2 Εξαγωγή Χαρακτηριστικών

Τα αρχεία NifTi που περιέχουν τις σαρώσεις fMRI για κάθε θέμα πρέπει να μετατραπούν σε πίνακες συσχέτισης που θα τροφοδοτηθούν στα μοντέλα. Ένας **πίνακας συσχέτισης** είναι ένας πίνακας που εμφανίζει τους συντελεστές συσχέτισης για διαφορετικές μεταβλητές. Ο πίνακας απεικονίζει τη συσχέτιση μεταξύ όλων των πιθανών ζευγών τιμών σε έναν πίνακα. Είναι ένα ισχυρό εργαλείο για τη σύνοψη ενός μεγάλου συνόλου δεδομένων και τον εντοπισμό και την οπτικοποίηση μοτίβων στα δεδομένα. Όλα τα βήματα για την εξαγωγή

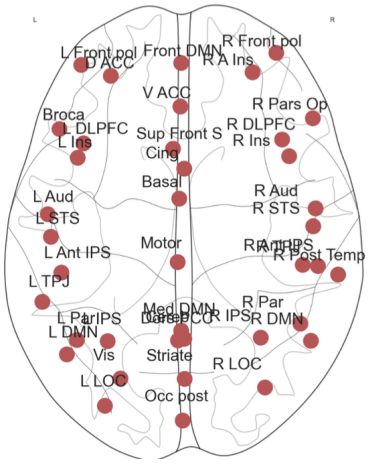


Figure 1.22. *Οι περιοχές ROI του MSDL Άτλαντα*

χαρακτηριστικών έχουν περιγραφεί στο Figure 3.27 . Στην περίπτωση μας μας ενδαφιέρουν τα βήματα για τον χάρτη MSDL που είναι ένας **πιθανολογικός χάρτης**

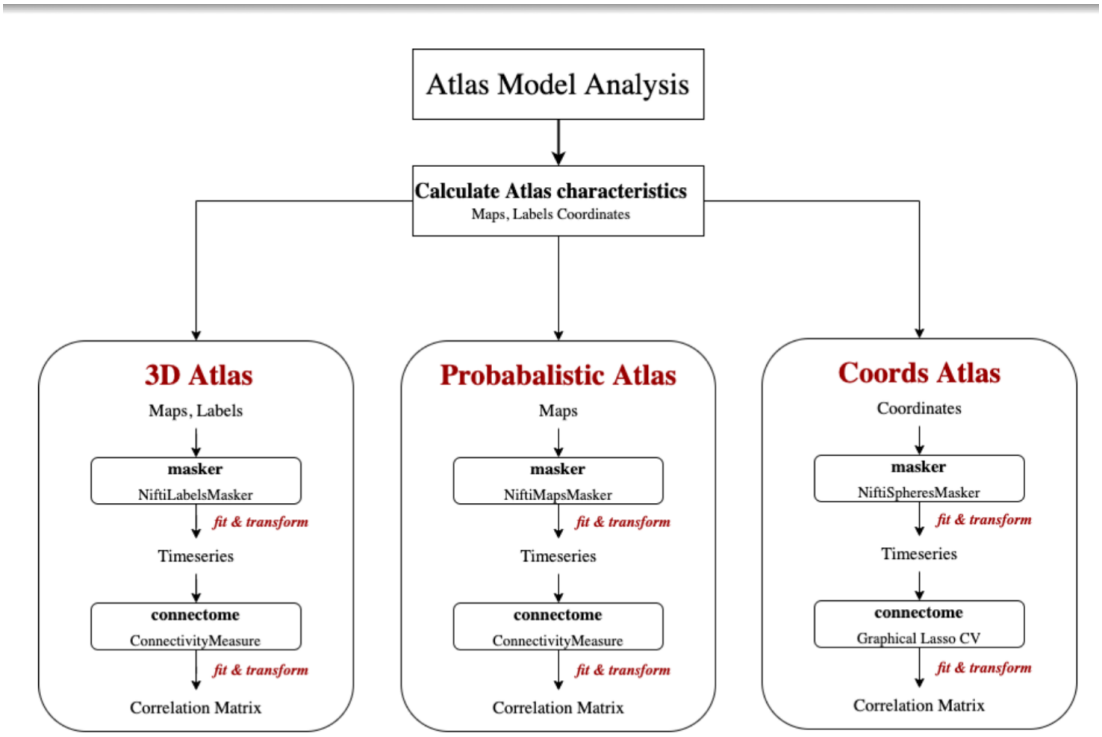


Figure 1.23. *Atlas Analysis Framework*

1. **Φόρτωση του χάρτη** : Χρησιμοποιήσαμε τη συνάρτηση **datasets**.*fetch_atlas_msdl()* για να ανακτήσουμε τον άτλαντα. Αυτό επιστρέφει ένα dictionary, με attributes:

   - *maps* : str, διαδρομή προς το αρχείο τλΝιΦτι που περιέχει την εικόνα του Πιθανολογικού Άτλαντα. Έχουμε 39 περιοχές ενδιαφέροντος. Οι διστάσεις του *maps* είναι $(40, 48, 35, 39)$
   - *labels* : list of str, η λίστα περιέχει τις ετικέτες των 39 περιοχών, ώστε το *data.labels*$[i]$ να αντιστοιχεί στην i-th περιοχή .
   - *region_coords* : list of tuples , η λίστα περιέχει τις συντεταγμένες POI, ώστε το *data.region_coords*$[i]$ να περιέχει τις συντενταγμένες (x, y, z) του κέντρου της i-th περιοχή i-th περιοχής.

2. **Καθορισμός του Masker** : Ορίζουμε ένα *NiftiMapsMasker()*, ένα ισχυρό εργαλείο για τη φόρτωση εικόνων και την εξαγωγή σημάτων voxel στην περιοχή που ορίζεται από τη μάσκα. Δόθηκαν οι παρακάτω παράμετροι

Parameter	**Value**	**Explanation**
maps_img	atlas_maps	The Nifti1Image object of the atlas maps
t_r	3	Έχουμε 3 καταστάσεις "PH', "W", "mispron"
detrend	True	
standardize	True	το σήμα είναι z-scored. Οι χρονοσειρές μετατοπίζονται στο μηδέν μέσο όρο και κλιμακώνονται με μοναδιαία διακύμανση
high_pass	0.0135	

3. **Εξαγωγή χρονοσειρών**: Χρησιμοποιούμε τη μέθοδο *fit_transform()* του καθορισμένου Masker για να εκπαιδεύσουνε το μοντέλο στις **λειγουργικές απεικονίσεις ιμαγες** και στα αντίστοιχα ¨**περιγράμματα**¨, με σκοπό να υπολογίσουμε το τις **χρονοσειρές** Ωστόσο, υπάρχουν 2 προσεγγίσεις εδώ από αυτές που μπορούν να ακολουθηθούν. Επιλέγουμε την 1η, αλλά για λόγους πληρότητας παραθέτουμε και τις 2

   - **1η προσέγγιση** :

     Συνένωση (concat) όλων των λειτουργικών απεικονίσεων σε μια ενιαία λειτουργική απεικόνιση με μέγεθος

     $$(64, 64, 64, 130 + 130 + 123 \text{ timestamps}) = (64, 64, 64, 383 \text{ timestamps})$$

     και στη συνέχεια υπολογίζουμε τα ¨περιγράμματα¨ (confounds). Εκπαιδεύουμε και μεταμορφώνουμε την μάσκα με την απεικόνιση και τα ¨περιγράμματα¨. Η έξοδος είναι η **χρονοσειρά** με μέγεθος

     $$(383 \text{ timestamps }, 39 \text{ regions })$$

   - Εκπαιδεύουμε και μεταμορφώνουμε την μάσκα 3 φορές ξεχωριστά, μία για κάθε λειγουργικλή απεικόνιση, εξάγουμε τις 3 διαφορετικές χρονοσειρές με μέγεθος

     $$(X \text{ timestamps }, 39 \text{ regions })$$

     όπου $X = \{130, 130, 123\}$. Τέλος συνενώνουμε τις χρονοσειρές και προκύπτει η τελική χρονοσιειρά με μέγεθος

     $$(383 \text{ timestamps }, 39 \text{ regions })$$

4. **Πίνακες αλληλοσχέτισης**: Οι πίνακες συσχέτισης υπολογίζονται χρησιμοποιώντας τις χρονοσειρές από το προηγούμενο βήμα. Ορίζουμε ένα αντικείμενο *δννεστιτψΜεασυρε()* που δέχεται ως παράμετρο τον τύπο συνδεσιμότητας, όπου και επιλέγουμε **"correlation"**. Στη συνέχεια χρησιμοποιούμε τη μέθοδο *fit_transfsorm()* για να εκπαιδεύσουμε το μοντέλο στις **χρονοσειρές** του προηγούμενου βήματος. Ο τελικός πίνακας συσχέτισης έχει μέγεθος

(39 regions , 39 regions)

Δείτε ένα παράδειγμα στο Figure 1.26 . Όπως μπορείτε να παρατηρήσετε οι συσχετίσεις είναι **πολύ έντονες**, αλλά εξακολουθούν να είναι αισθητές. Ο λόγος είναι ότι καταργήσαμε το **confounds** που προσθέτουν **θόρυβο** στις **χρονοσειρές**

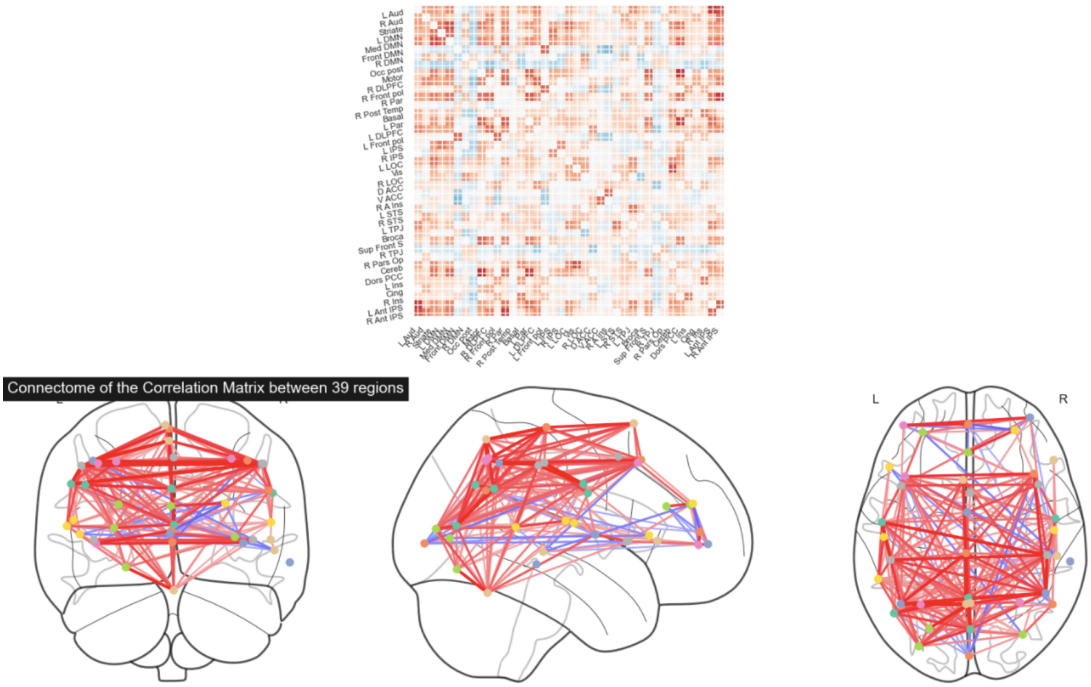


Figure 1.24. *Ο πίνακας αλληλοσχέτισης και οι συνδέσεις των 39 περιοχών του εγκεφάλου (με κατάλληλη χρωματική ένταση) για τον συμμετέχων 047EPKL011005 όπως αυτός προκύπτει από το παραπάνω pipeline προεπεξεργασίας*

1.9 Πειράματα

Μετά από το θεωρητικό πλαίσιο και την ανάλυση της κάθε μεθόδου που αναφέρθηκαν στα προηγούμενα κεφάλαια, καθώς και κατόπιν της γεφύρωσης αυτών των μεθόδων με το πρόβλημα που κλήθηκε να λύσει η παρούσα εργασία, φυσικό είναι να παρατεθούν στον αναγνώστη και τα αποτελέσματα των πειραμάτων που πραγματοποιήθηκαν για τη διάγνωση της μαθησιακής δυσκολίας.

1.9.1 Ορισμός των συνόλων X,y

Για κάθε έναν από τους 58 συμμετέχοντες έχουμε έναν συμμετρικό πίνακα αλληλοσχέτισης μεγέθους $(39, 39)$. Όμως οι αλγόριθμοι μηχανικής μάθησης απαιτούν διανύσματας 1 διάστασης. Επομένως, και επειδή το πίνακας είναι συμμετρικός, κρατάμε μόνο το πάνω τριγωνικό τμήμα, το οποίο στη συνέχεια ισοπεδώνεται (**flatten**). Το τελικό διάνυσμα του i-th συμμετέχοντας D_i θα έχει μέγεθος

$$n_{features} = \frac{N_{regions} \cdot (N_{regions} - 1)}{2} = \frac{39 \cdot (39 - 1)}{2} = 741$$

όπου κάθε χαρακτηριστικό (feature) $x \in D_i$ αντιπροσωπεύει τη συσχέτιση ενός ζεύγους περιοχών ενδιαφέροντος (ROI). Όλα τα $\forall x \in D_i$, είναι **αριθμητικά**. Δεν χρησιμοποιήθηκαν **κατηγορικά χαρακτηριστικά**, με

Optimization Framework	Explanation	Training set	Validation set	Test set
Transformers Optimization	Best Transformers (per Classifier)	(47, 741)		(11, 741)
Gridsearch Optimization	Best Parameters (per pipeline)	(43, 741)	(4, 741)	(11, 741)
Transformers Optimization	Best Hard Voting Ensemble Classifier	(47, 741)		(11, 741)

εξαίρεση τη χρήση φαινοτυπικών δεδομένων για τη δομή του γραφήματος του μοντέλου βαθιάς μάθησης Γ"ΝΝ

Έχουμε 58 συμμετέχοντες οπότε $n_{samples} = 58$. Ο τελικός πίνακας χαρακτηριστικών *tlX* και το διάνυσμα ετικέτας *tly* θα έχουν μεγέθη $(n_{features}, n_{samples}) = (58, 741)$ και $(n_{samples},) = (58,)$ αντίστοιχα.

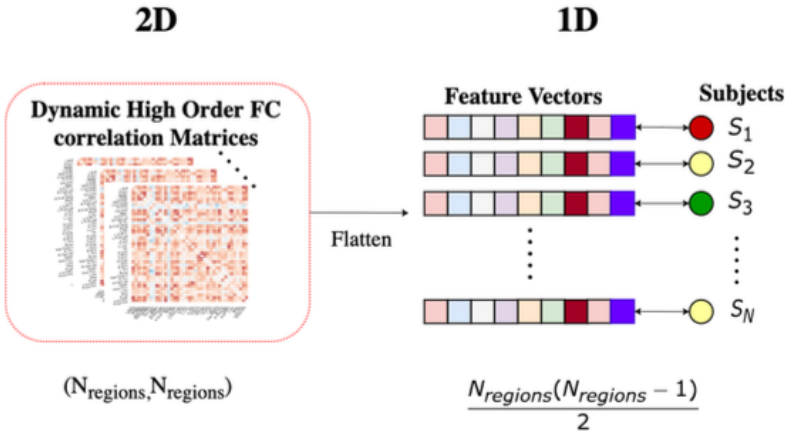


Figure 1.25. *"Ισοπέδωση" (flatten) πινάκων αλληλοσχέτισης*

1.9.2 Ορισμός των συνόλων εκπαίδευσης (train), επικύρωσης (validation) και ελέγχου (test)

Το παρόν dataset έχει 58 δείγματα, τα οποία πρέπει να διαχωριστούν σε σύνολο **εκπαίδευσης** και σύνολο **ελέγχου**. Έγινε χρήση της μεθόδου train_test_split από τη βιβλιοθήκη Sci-kit learn. Το σύνολο ελέγχου αποτελεί το 20% των δεδομένων (ή συνολικά ~ 11 δείγματα), και το υπόλοιπο 80% χρησιμοποιήθηκαν για την εκπαίδευση (ή συνολικά ~ 47 δείγματα).

Στην παρούσα διπλωματιή θα προβούμε σε διάφορα είδη βελτιστοποιήσεων. Συγκεκριμένα, κάτα την **βελτιστοποίηση πλέγματος** (Gridsearch Optimization), το σύνολο **εκπαίδευσης** χωρίστηκε περαιτέρω σε σύνολο **εκπαίδευσης** και σύνολο **επικύρωσης** . Εφόσον επιλέγουμε $cv = 10$ έχουμε

- **Εκπαίδευση** : Ένα μοντέλο εκπαιδεύεται χρησιμοποιώντας $k = 9$ από τα folds → Επομένως τελικός αριθμός δειγμάτων εκπαίδευσης $\frac{9}{10}47 \sim 43$.

- **Επικύωρησ** : Το εκπαιδευμένο μοντέλο επικυρώνεται χρησιμοποιώντας τα υπόλοιπα $k = 1$ από τα folds → Επομένως τελικός αριθμός δειγμάτων επικύρωσης $\frac{1}{10}47 \sim 4$.

Ο τελικός πίνακας με την κατανομή του πλήθους δειγμάτων στα διάφορα σύνολα μπορεί να βρεθεί παρακάτω

1.9.3 Μετρικές Αξιολόγησης

Η παρούσα διπλωματική πραγματεύεται με το πρόλβημα της **ταξινόμησης**, δηλαδή την εκπαίδευση μοντέλων με σκοπό τον διαμοιρασμό των δειγμάτων στις 3 κλάσεις **Δυσλεξια, Ορθογραφική Διαταρχή, Υγιές**. Μετά την εκπαίδευσή τους θα πρέπει κάποιος να είναι σε θέση να αξιολογήσει την ικανότητα των μοντέλων στο να εκτελούν αυτή την εργασία με επιτυχία. Για αυτό το σκοπό έχει οριστεί ένα σύνολο από ευρέως χρησιμοποιούμενες **μετρικές** οι οποίες εκφράζουν την ακρίβεια των προβλέψεων ενός μοντέλου σε σχέση με τις πραγματικές τιμές των δεδομένων αξιολόγησης. Έτσι γίνεται αισθητό πως αυτές δεν σχετίζονται με τη συνάρτηση κόστους που αναφέρθηκε στα μοντέλα ταξινόμησης.

Για τη δυαδική αλλά και πολλαπλή ταξινόμηση των δεδομένων ορίζονται ως: αληθώς θετική (True Positive) μία πρόβλεψη που ήταν θετική και η πραγματική τιμή των δεδομένων ήταν θετική, ψευδώς θετική (False Positive) μία πρόβλεψη που ήταν θετική και η πραγματική τιμή των δεδομένων ήταν αρνητική, αληθώς αρνητική (True Negative) μία πρόβλεψη που ήταν αρνητική και η πραγ- ματική τιμή των δεδομένων ήταν αρνητική, ψευδώς αρνητική (False Negative) μία πρόβλεψη που ήταν αρνητική και η πραγματική τιμή των δεδομένων ήταν θετική. Συμβολίζεται TP το σύνολο των True Positive τιμών, FP το σύνολο των False Positive τιμών, TN το σύνολο των True Negative τιμών, FN το σύνολο των False Negative τιμών. Οι μετρικές ορίζονται ως:

- Ορθότητα (Accuracy): Εκφράζει το ποσοστό επιτυχίας του μοντέλου, δηλαδή το πλήθος των δειγμάτων που το μοντέλο ταξινόμησε σωστά προς το σύνολο όλων των δειγμάτων.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

- Ακρίβεια (Precision): Εκφράζει το ποσοστό των επιτυχημένων προβλέψεων μίας κλάσης προς τις συνολικές προβλέψεις της κλάσης.

$$Precision = \frac{TP}{TP + FP}$$

- Ανάκληση (Recall): Εκφράζει το ποσοστό των επιτυχημένων προβλέψεων μίας κλάσης προς το πραγματικό πλήθος των παρατηρήσεων που ανήκουν σε αυτή.

$$Recall = \frac{TP}{TP + FN}$$

- F1 score: Αποτελεί τον αρμονικό μέσο των Precision και Recall.

$$F1 = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall}$$

1.9.4 Machine Learning

Βελτιστοποίηση no. 1

Για κάθε έναν από αυτούς τους 16 ταξινομητές, εκτελούμε μια εξαντλητική βελτιστοποίηση με 4 ένθετους βρόχους (1 για κάθε είδος μετασχηματιστών). Ο συνολικός αριθμός των συνδυασμών που εξετάστηκαν (καθώς και ο απαιτούμενος χρόνος εκτέλεσης) είναι

No. Experiments	**Duration**	**Duration per Experiment**
$\|E\| = \|S\| \times \|U\| \times \|FT\| \times \|FE\| \times \|C\| \sim 7680$	3047 sec	0.4 sec

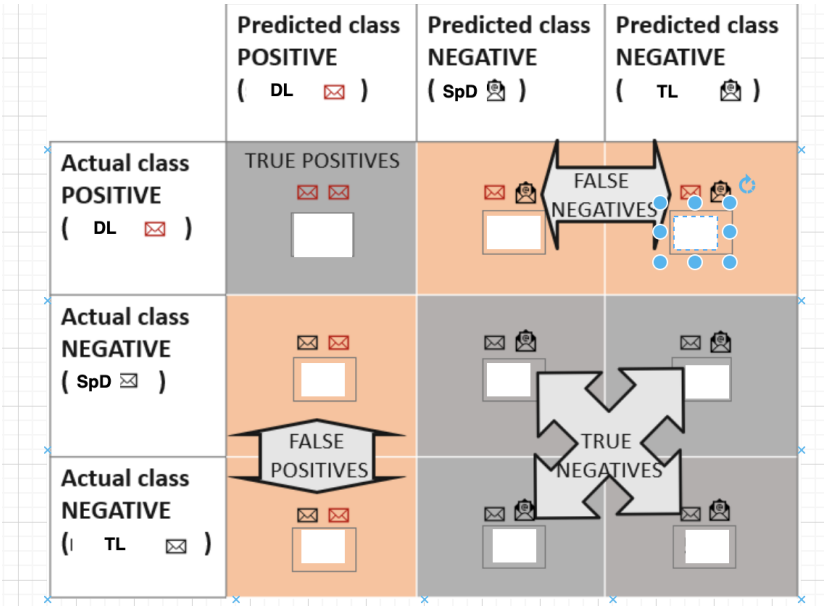


Figure 1.26. *Πίνακας Σύγχυσης για ταξινόμηση 3 κλάσεων*

Τα αποτελέσματα των πειραματών βρίσκονται στο Πίνακα ~ **1.9.4**

Classifier	Scaler	Undersampler	Feature Selector	Feature Extractor	F1 score
adaboost	quantile	nearmiss	selectpercentile	pca	0.624
calibrated	quantile	tomek	selectfpr	lda	0.574
decisiontree	power	random	selectfpr	pca	0.666
extratree	quantile	nearmiss	selectpercentile	lda	0.604
gradientboosting	powerpower	random	selectpercentile	pca	0.648
knn	minmax	condensed	selectkbest	lda	0.562
lda	standard	tomek	variancethreshold	lda	0.583
linearsvc	quantile	tomek	selectfpr	lda	0.573
logistic	standard	editednearest	selectkbest	lda	0.654
mlp	standard	condensed	variancethreshold	lda	0.569
nusvc	power	condensed	selectfpr	pca	0.653
randomforest	power	nearmiss	variancethreshold	lda	0.586
ridge	maxabs	random	variancethreshold	pca	0.587
sgd	quantile	tomek	selectfpr	lda	0.573
svc	power	condensed	selectfrommodel	lda	0.623
xgboost	power	editednearest	selectfpr	pca	0.764

Βελτιστοποίηση no. 2

Ο συνολικός αριθμός των πειραμάτων χρησιμοποιώντας το HalvingGridsearchCV μπορεί να βρεθεί στον πίνακα ~ **1.9.4**. Ο *Αριθμός Δοκιμών* είναι το γινόμενο του πλήθους όλων των παραμέτρων των 4 μετασχηματιστών και του ταξινομητή, ενώ *Αριθμός Πειραμάτων* είναι το γινόμενο του Αριθμού Δοκιμών με τοcv όπου $cv = 10$.

Classifier	No. Configurations
adaboost	14112
calibrated	4480
decisiontree	11520
extratree	54432
gradientboosting	69984
knn	268800
lda	1152
linearsvc	20160
logistic	32256
mlp	89600
nusvc	18144
randomforest	1512
ridge	53760
sgd	48384
svc	89600
xgboost	110592

Ένα παράδειγμα βελτιστοποιημένου pipeline μπορεί βρεθεί στο Figure 1.27 ενώ τα συνολικά αποτελέσματα μετά την εφαρμογή των 2 παραπάνω βελτιστοποιήσεων βρίσκονται στο Figure 1.28

- Ο καλύτερος βελτιστοποιημένος αλγόριθμος pipeline είναι ο **XGBoost Classifier** με test score F_1 = 76.5%
- Ο χειρότερος βελτιστοποιημένος pipeline είναι ο **SVC** και **NuSVC** με test score F_1 = 36.1% F_1 = 27.8% που είναι λιγότερο ακόμα και από έναν baseline classifier. Μάλιστα στην ίδια οικογένεια ταξινομητών ανήκει και ο **LinearSVC** με test score F_1 = 57.4% . Γενικά, οι μη γραμμικοί ταξινομητές είναι πιο κατάλληλοι για τη μοντελοποίηση πιο περίπλοκων συναρτήσεων από τις γραμμικές, αλλά εξαρτάται από τα δεδομένα, τις επιλεγμένες υπερπαραμέτρους (π.χ. ποινή και πυρήνα) και τον τρόπο ερμηνείας των αποτελεσμάτων .

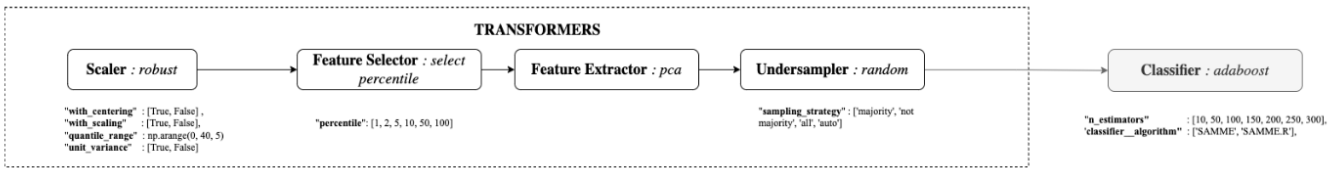


Figure 1.27. *Pipeline του Adaboost Classifier*

Βελτιστοποίηση no. 3

Ο συνολικός αριθμός των πειραμάτων αφορά όλους τους πιθανούς συνδυασμούς των βελτισοποιημένων pipeline (ως προς μετασχηματιστές και υπεραπαραμέτρους). Γενικά, υπάρχουν πολλά διαφορετικά είδη συλλογικών ταξινομητών (Ensembling). Σε αυτή τη διπλωματική εργασία χρησιμοποιούμε μόνο την προσέγγιση της ¨σκληρής¨ (hard) ψηφοφορίας, ωστόσο μελλοντικά σκοπεύουμε να πειραματιστούμε και με την προσέγγιση του soft voting.

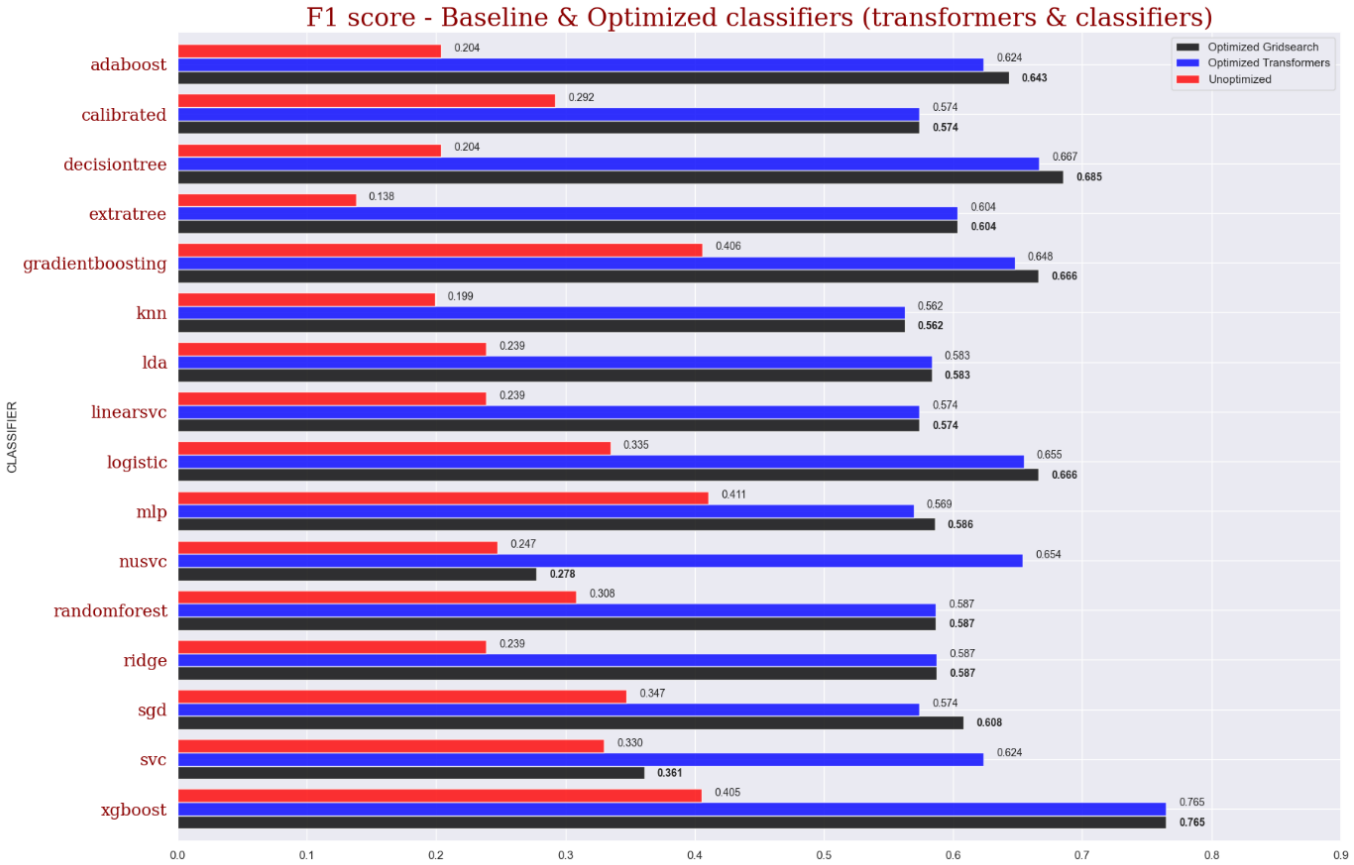


Figure 1.28. *Τα αποτελέσματα για το F1 test score μετά τις βελτιστοποιήσεις no.1 , no.2*

Στην συγκεκριμένη βελτιστοποίηση ψάχνουμε τον συνδυασμό που θα οδηγήσει στο καλύτερο test f1 score. Δοκιμάζουζουμε όλους τους συνδυασμούς για όλους τα 16 βελτιστοποημένα πιπελινες (εκπαίδευση και αξιολόγηση). Τα συνολικά αποτελέσματα μετά την εφαρμογή των 3 παραπάνω βελτιστοποιήσεων βρίσκονται στο Figure 6.3, όπου αναδεικνύουμε τους πρώτους 20 συνδυασμούς. Στο αριστερό διάγραμματα παρατηρείται ένας πίνακας αληθείας που υποδεικνύει ποια βελτιστοποιημένα μοντέλα συμμετέχουν στον εκάστοτε συνδυασμό (με έντονο μπλε συμβολίζεται η συμμετοχή ενώ με ανοιχτό μπλε η μη συμμετοχή).Τέλος, ο αριθμώς των συνδυασμών-πειραμάτων είναι :

$$\sum_{k=1}^{16} \binom{16}{k} = 65535$$

Μερικές παρατηρήσεις :

- Ο καλύτερος βελτιστοποιημένος συνδυασμός είναι αυτός που περιέχει τα βελτιστοποιημένα pipelines των ταξινομητών **MLP, Logistic, Random Forest, Ridge, Extra Tree** με test score $F_1 = 83.5\%$
- Ενδιαφέρον είναι ότι το pipeline του ταξινομητή **XGBoost**, που οδηγούσε στο καλύτερο test score F_1 μετά την βελτιστοποίηση no.2, δεν συμμετέχει σε κανέναν από τους πρώτους 20 συνδυασμούς. Μια πιθανή εξήγηση είναι ο **XGBoost** λειτουργεί πολύ διαφορετικά από όλους τους υπόλοιπους ταξινομητές.
- Ένα ακόμα ενδιαφέρουν εύρημα είναι το pipeline του ταξινομητή **Extra Tree** συμμετέχει σε 19 από τους 20 πρώτους συνδυασμούς, ενώ τα pipelines των ταξινομητών **Linear SVC, SVC, kNN, LDA, Decision Tree** σε κανέναν
- Τα αποτελέσματα μπορεί να είναι λίγο βιασεδ, καθώς μπορείνα υπάρχει overfitting σε κάποιους απο

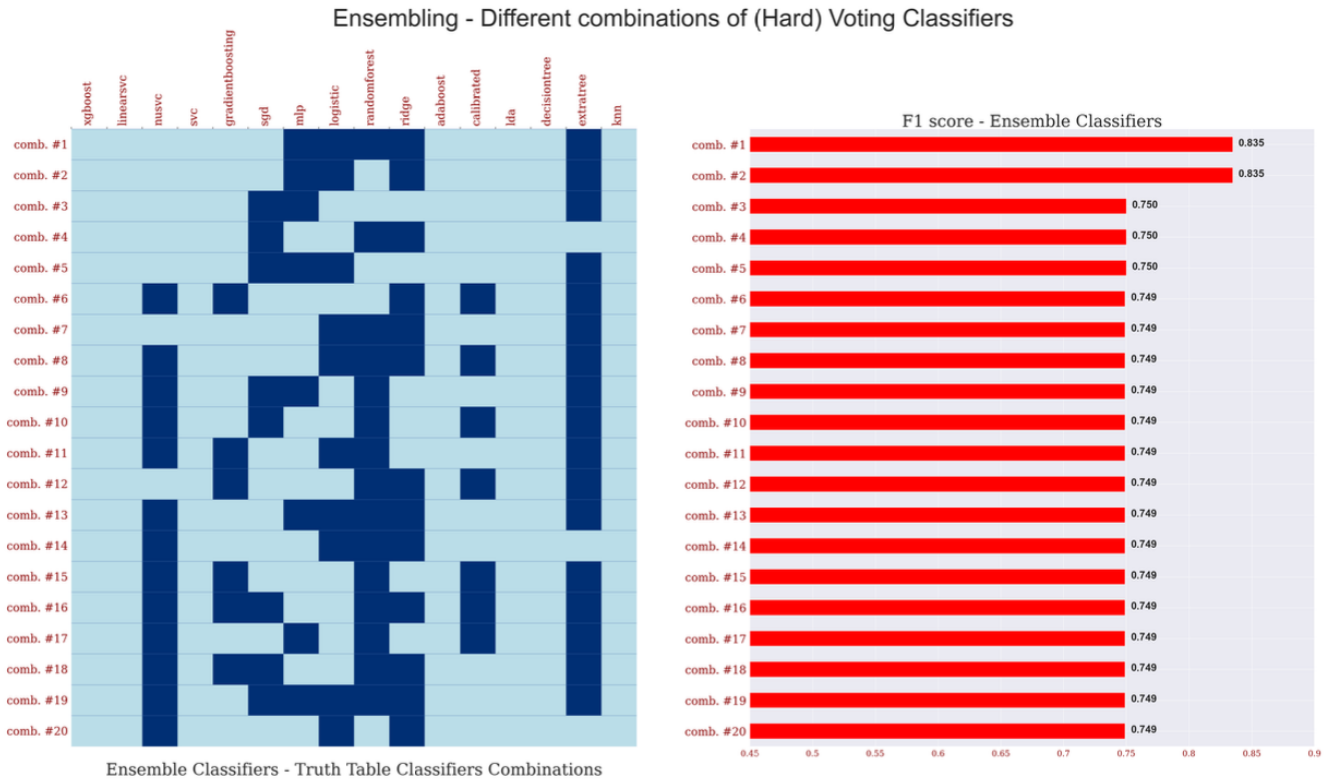


Figure 8.4. *Top 20 (F1 score) combinations of Hard Voting Ensemble Classifiers*

Figure 1.29. *Τα αποτελέσματα για το F1 test score μετά τις βελτιστοποιήσεις no.1 , no.2, no. 3*

τους συνδυασμούς το οποίο πρέπει να ελεγχθεί. Αυτό μπορεί να γίνει κάνοντας visualize τις γραφικές train/validation loss, learning curve.

- Τέλος, η συγκεκριμένα βελτιστοποίηση έλαβε χώρα μόνο για σκοπούς απόδοσης. Επειδή οι συλλογικοί ταξινομητές είναι "άφηρημένοι" ταξινομητές υψηλού επιπέδου, δεν μπορούμε να ερμηνεύσουμε ξεκάθαρα τα "μαύρα κουτιά" τους και τον τρόπου λειτουργίας τους. Για αυτό τον λόγο, δεν θα τους χρησιμοποιήσουμε για να ελέγξουμε τις σημαντικότητες των χαρακτηριστικών, αντιθέτως θα κινηθούμε με τα βελτιστοποιημένα pipelines του προηγούμενου βήματος.

Επεξηγισιμότητα με LIME

Χρησιμοποιώντας τον παραμετροποιημένο αλγόριθμο που παρουσιάσαμε σε προηγούμενη ενότητα, εδώ δείχνουμε τα αποτελέσματα της εφαρμογής στον καλύτερο (ως προς test score F_1) βελτιστοποιημένο pipeline που προκύπτει μετά τα βήματα no. 1 no.2. Μάλιστα, είναι σημαντικό να τονίσουμε ότι εφαρμόζουμε τον παραπάνω αλγόριθμο για κάθε κλάση ξεχωριστά. Επομένως προκύπτουν 3 διαφορετικά δισνύσματα βαρών w' μεγέθους (741,) ένα για κάθε κλάση, δηλαδή $w'_{SpD}, w'_{DL}, w'_{TD}$. Οπτικοποιούμε τα 20 πρώτα για κάθε κλάση όπως φαίνεται στο στο Figure 1.31

Μερικές παρατηρήσεις :

- Οι περιοχές του **πρόσθιου φλοιού του προσαγωγίου** (Dorsal Anterior Cingulate Cortex, **πίσω φλοιού του προσαγωγίου** (Ventral Anterior Cingulate Cortex), **δεξιού νησιωτικού φλοιού** (Right Anterior Insula) ενεργοποιούνται ιδιαίτερα σε υγιή παιδιά (typical readers) ενώ για τις υπόλοιπες 2 ομάδες παιδιών δεν ενεργοποιούνται. Ο ρόλος αυτών σχετίζεται στενά με την επεξεργασία ανταμοιβής στην κοινωνική

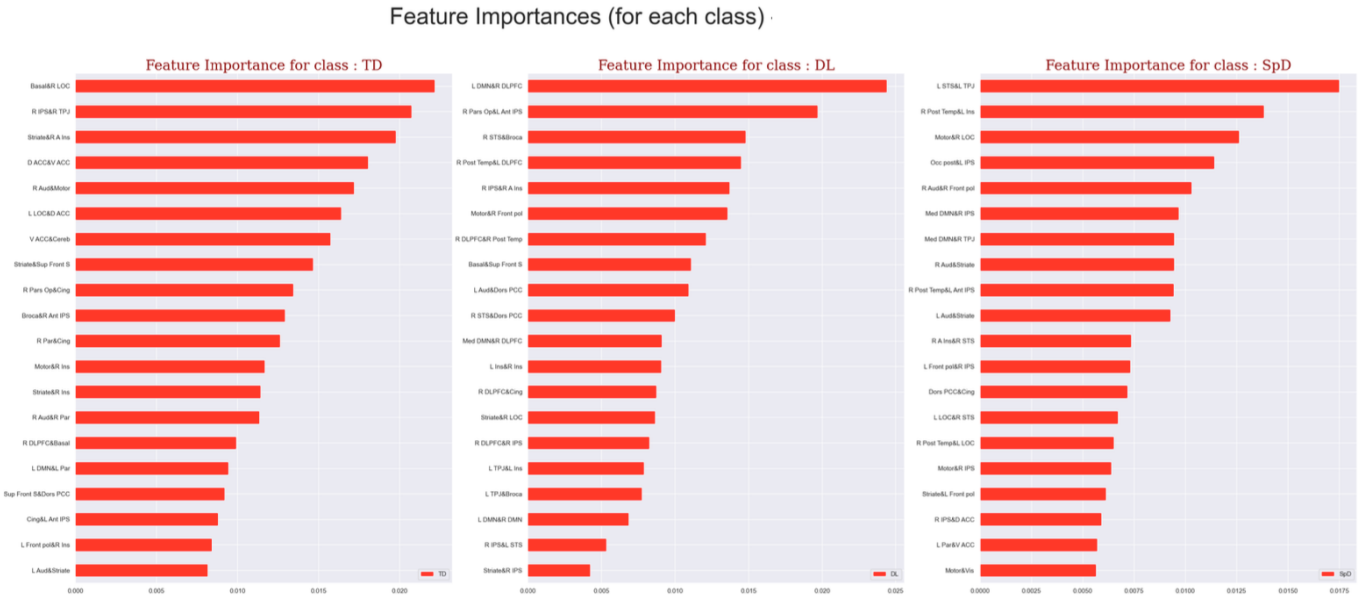


Figure 1.30. *Διάγραμμα μπαρών που υποδεικνύει τις σημαντικότητες των 20 μη κοινών χαρακτηριστικών για τις 3 κλάσεις*

αξιολόγηση και ενσυναίσθηση

- Η περιοχή του Broca παίζει βασικό ρόλο στα παιδιά με Δυσλεξία, εύρημα που είναι απόλυτα σύμφωνα με τη βιβλιογραφία μας. Για τις υπόλοιπες 2 ομάδες, η περιοχή του Broca δεν συμμετέχει σε κανένα χαρακτηριστικό, ούτε καν στα top 20 τους

- Οι δυσλεκτικοί έχουν δυσλειτουργία των συστημάτων μετατροπής φωνολογικού και ορθογραφίας σε φωνολογία. Πράγματι, η περιοχή του **ινιακού λοβού** (Occipital Lobe) δεν είναι ενεργοποιημένη και δεν συμμετέχει σε καμία από τα 20 κορυφαία χαρακτηριστικά.

- Η ενεργοποίηση του **ραβδωτού** (Stratium), που συντονίζει τη λήψη αποφάσεων, τα κίνητρα, την ενίσχυση και την αντίληψη της ανταμοιβής είναι υψηλότερη (αλλά συγκρίσιμη) σε υγιή παιδιά (typical readers) σε σχέση με την αντίστοιχη ενεργοποίηση στις υπόλοιπες 2 ομάδες.

- Η περιοχή του **Αριστερού Προσθίου Ενδοβεγματικού Αυλίου** (Left Anterior Intraparietal Sulsus) είναι ιδιαίτερα ενεργοποιημένη σε παιδιά με **Δυσλεξία** και **Ορθογραφική Διαταραχή**, όπως αναφέρεται και σε σχετική βιβλιογραφία.

- Τέλος, βρήκαμε μειωμένη εγκεφαλική δραστηριότητα στις περιοχές του **κυκλικού φλοιού** (Cing) στα παιδιά με **ορθογραφική διαταραχή** στο ΣπΔ σε σύγκριση με τις υπόλοιπες 2 ομάδες

Ένα ακόμα διάγραμμα που προσφέρει πολλή αξία στην παρούσα διπλωματική είναι το Figure **;;**
Μερικές παρατηρήσεις :

- Το διάγραμμα όχι μόνο δείχνει τα top 20 κοινά χαρακτηριστικά πάνω στον πανακα συσχέτισης αλλά δείχνει και την ένταση τους (θετική - αρνητική) πάνω στο σχετικό connectome εγκεφάλου, το οποίο βοηθάει να καταλάβουμε την επίδραση των χαρακτηριστικών σε κάθε μία από τις κλάσεις ξεχωριστά. Σημειώνεται ότι επιλέξαμε να οπτικοποιήσουμε κοινά χαρακτηριστικά για να εξάγουμε πιο ακριβή συμπεράσματα για τις 3 ομάδες που μελετάμε.

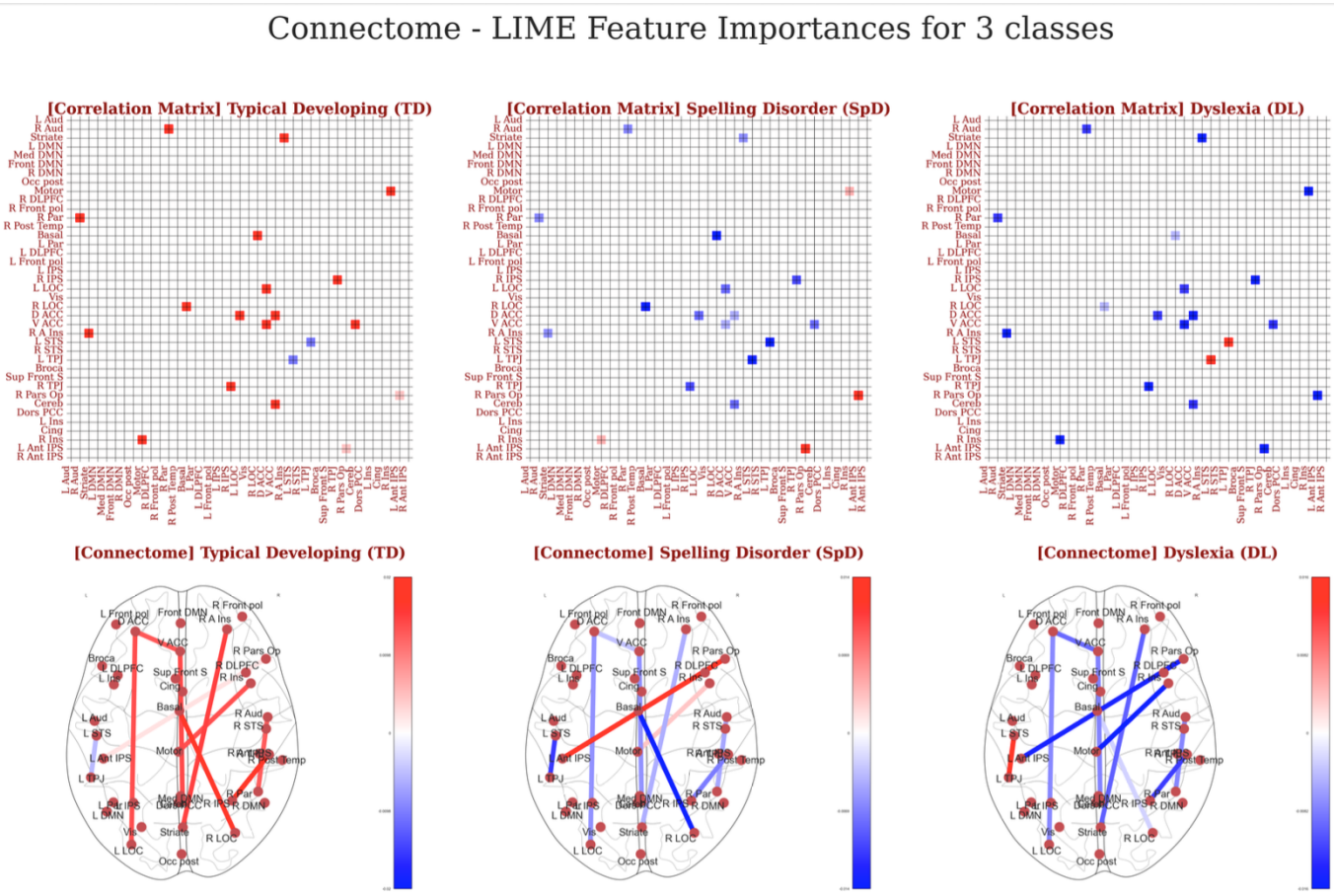


Figure 1.31. *Διάγραμμα μπαρών που υποδεικνύει τις σημαντικότητες των 20 κοινών χαρακτηριστικών για τις 3 κλάσεις καθώς και τις σχετικές εντάσεις των ακμών που συνδέετι τις περιοχές του εγκεφάλου*

- Όλες οι ακμές/συνδέσεις έχουν **θετική επίδραση** για να πούμε ότι ένα παιδί δεν έχει καμία **γλωσσική δυσκολία** ενώ οι ίδιες συνδέσεις έχουν **αρνητική επίδραση** για να πούμε ότι ένα παιδί έχει **δυσλεξία** ή **ορθογραφική διαταραχή**. Το οποίο φαίνεται σχεδόν σε όλα τα χαρακτηριστικά

1.9.5 Βαθιά Μάθηση

Πειράματα

Έχοντας δομήσει την αρχιτεκτονική του μοντέλου μας, θέλουμε να βελτιστοποιήσουμε τις υπεραπαρμέτρους τρέχοντας κατάλληλα πειράματα, με σκοπό να πετύχουμε το καλύτερο test f1 score αλλά ταυτόχρονα να απο-

Table 1.4. *Όλες οι τιμές των υπερπαραμέτρων που ελέγχθηκαν κατά την διάρκεια των πειραμάτων με το G-CNN (Hyperparameter Tuning)*

Hyperparameter	Value Options	Best Value
Epochs	$e \in (20, 240)$ with step = 20	80
κ	$k \in (1, 14)$	4
∂	$\partial \in (1, 5)$	3
Dropout	$d \in (0.05, 0.8)$	0.6
Hidden Units	$h \in \{[2], [8], ..., [4, 8, 16, 32, 64]\}$	[8]
Learning Rate Units	$lr \in ([1e-1, 1e-8])$	$1e-4$

φύγουμε το overfitting, πρόβλημα που είναι πολύ σύνηθες σε μοντέλα βαθιάς μάθησης που εκπαιδεύονται με λίγα δεδομένα. Τα αποτελέσματα των πειραμάτων φαίνονται στον πίνακα **Table 6.2**. Συνολικά, ο αριθμός των πειραμάτων ανέρχονται :

$$|E| = |epochs| \times |\kappa| \times |\partial| \times |d| \times |h| \times |lr| \sim 985600$$

Απώλεια (loss), Ακρίβεια (accuracy)

Από όλα τα παραπάνω πειράματα, ξεκινάμε από αυτό με το καλύτερο test f1 score = 83.3% και προχωράμε σε φθίνουσα σειρά μέχρι να βρούμε κάποιο πείραμα στο οποίο δεν συναντάμε το φαινόμενο του overfitting. Στην συγκεκριμένα περίπτωση ήταν το πρώτο, και οι τιμές των καλύτερων τιμών για τις υπερπαραμέτρους φαίνεται επίσης στο **Table 6.2**. Οι καπύλες απώλειας (loss), ακρίβειας (accuracy) τόσο για το σύνολο εκπαίδευσης όσο και για το σύνολο επαλήθευσης φαίνονται στο Figure 1.32 . Παρατηρούμε ότι ενώ το test f1 score = 83.3%, το validation accuracy score ≤60 %

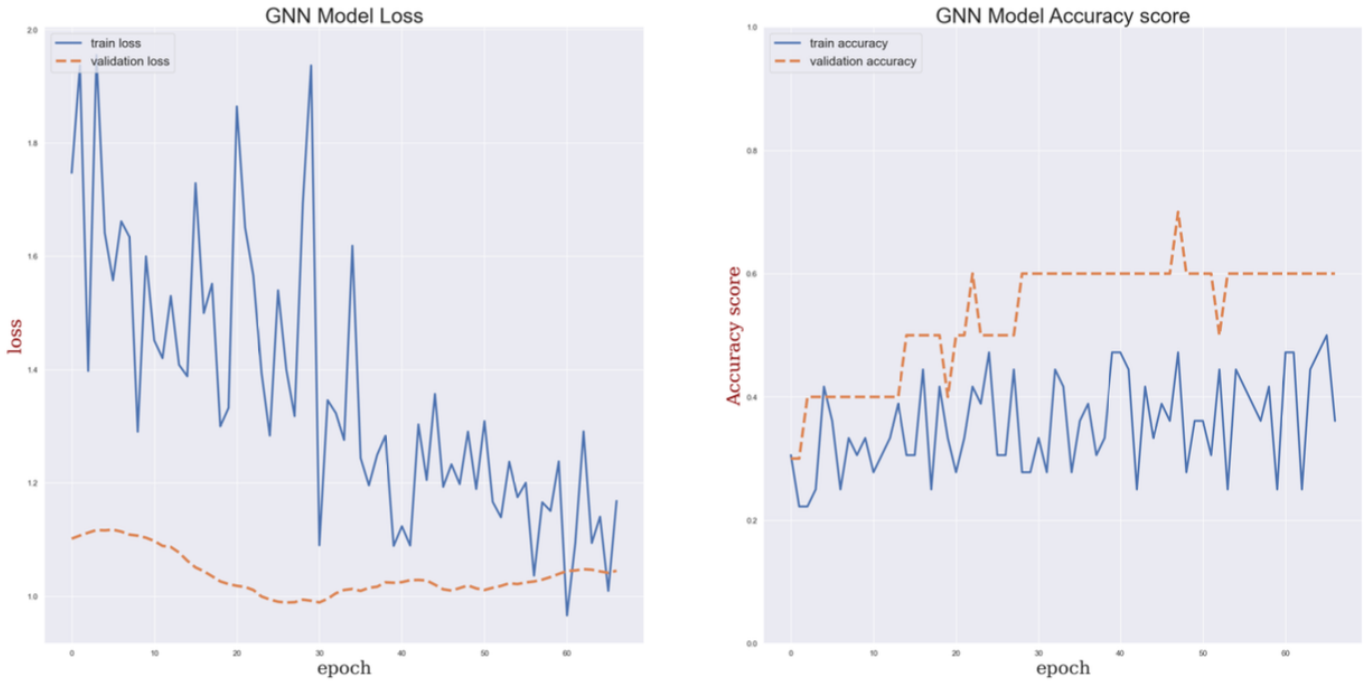


Figure 1.32. *Καμπύλες απώλειας και ακρίβειας για το σύνολο εκπαίδευσης και επαλήθευσης*

Chapter 2

Dyslexia & fMRI

2.1 Definition of Dyslexia

Dyslexia has been around for a long time and has been defined in different ways.

- Based on **International Dyslexia Association** : *Dyslexia is a specific learning disability that is neurobiological in origin. It is characterized by difficulties with accurate and/or fluent word recognition and by poor spelling and decoding abilities. These difficulties typically result from a deficit in the phonological component of language that is often unexpected in relation to other cognitive abilities and the provision of effective classroom instruction. Secondary consequences may include problems in reading comprehension and reduced reading experience that can impede growth of vocabulary and background knowledge*
- Based on [Organization, 2010] : *Dyslexia is a specific reading disorder characterized by a specific and significant impairment in the development of reading skills that are unrelated to problems with visual acuity, schooling or overall mental development*

2.2 General Symptoms

For groups of people with some reading disability, 80 % of them have a **phonological difficulties** and are considered **dyslectic**, and the rest 20 % have **speed & comprehensions difficulties** and they are not technically characterized as **dyslectic** but reading help will be required. Reading difficulties have been related to various symptoms:

- **Phonological Difficulties** (most frequently reported). Concerns around **the 80%** Brain is unable to process phonemes, ie. the smallest units of speech that make words different from each other. So there is difficulty decoding words based on their sounds.
- **Visual/Attentional** deficits (frequently reported as well) [Ramus and Ahissar, 2012]
- **Speed/Naming Deficit** Slow reading; poor use oof sight words, words which are instantly recognized by a control subject (not sounded out, no effort to understand)
- **Comprehension Deficit** Poor understanding of what was just read

2.3 Types

Figure 2.1 describes a taxonomy of only the most recognized and discussed types of dyslexia [ref, 2022b]. Generally, there are even more subtypes and categories identified by researchers. Deep diving into those is

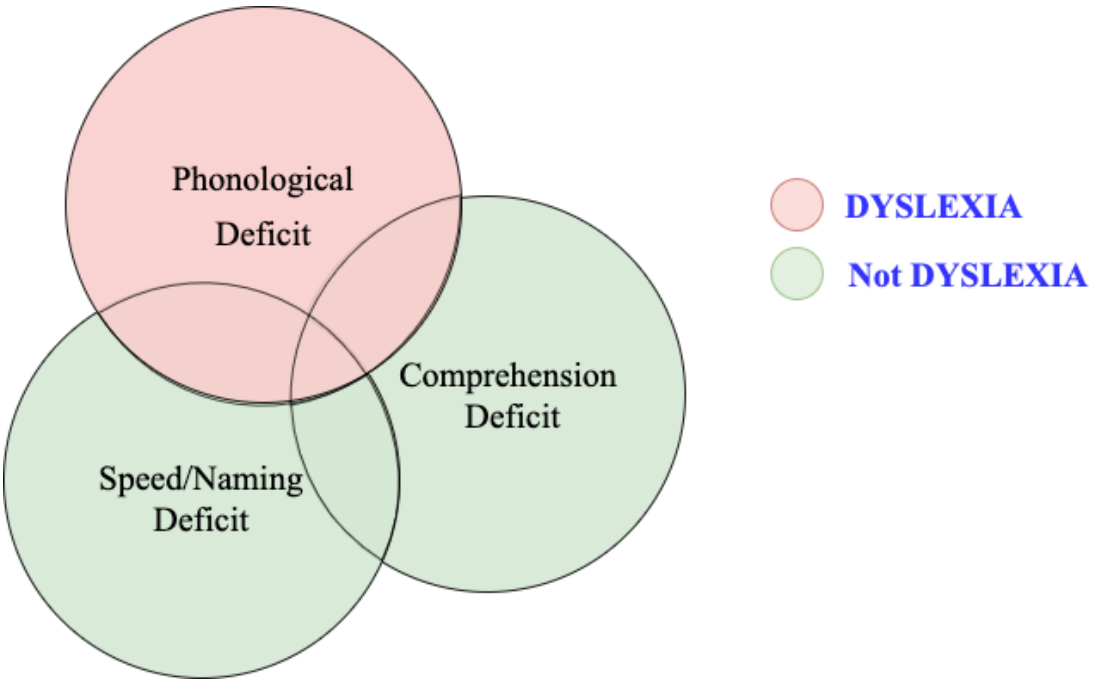


Figure 2.1. *Dyslexia Reading Deficits*

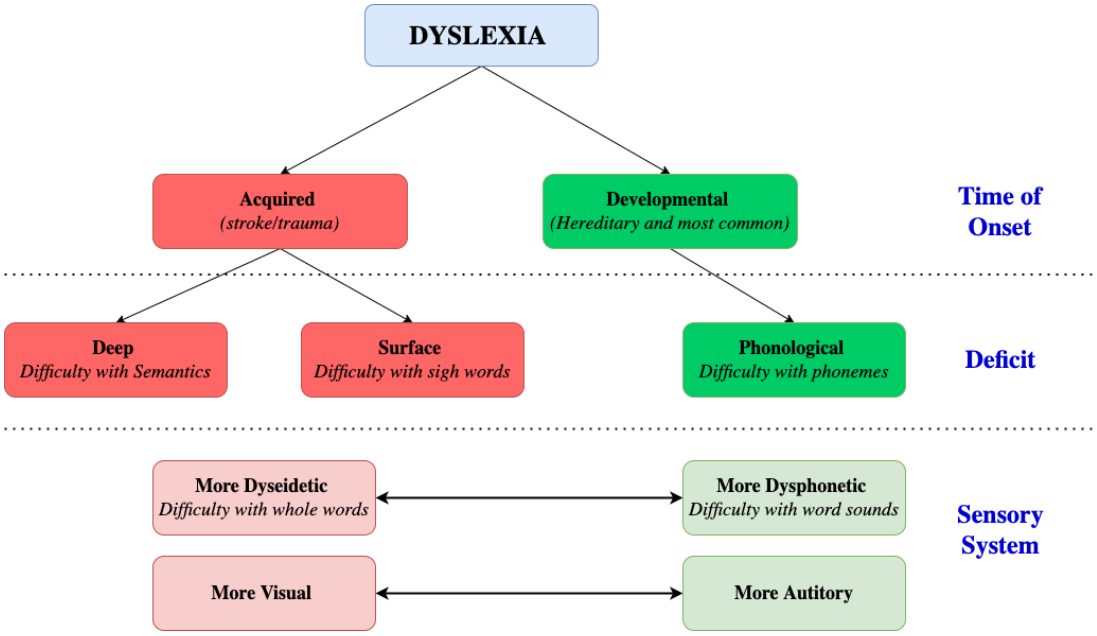


Figure 2.2. *Dyslexia Types*

out of the scope of this project.

2.3.1 Dyslexia by Time of Onset

Developmental Dyslexia

This is the **most common** type of dyslexia, giving the definition to the term **dyslexia** itself, which is due to the fact that 80% of dyslectics belong to this category, having a phonological **hereditary** brain based deficit/disability. So, this dyslexia is genetic and present from birth, subsequently '**developing**' over the course of time. Often co-exists with **ADHD, dysgraphia, dyscalculia and dyspraxia**. Classic dyslexia symptoms (which can be seen as early as the first 6 **months** of age) vary in degree and number accordingly

and include :

- Slow reading
- Very poor spelling and weak phonemic awareness resulting in great difficulty sounding out words (especially unfamiliar ones)
- Pre-school warning signs : delayed speech, difficulty learning the alphabet, inability to rhyme words, confusion of left right, before and after and other directional or relational words, poor pencil grip and messy writing
- Difficulty tying shoe laces and reading a clock with hands

In our research, the participants are 58 and all are children aging between 9 and 13. So it would be wise to deep dive into **the Elementary School Dyslexia Symptoms** :

- **Struggles with directions (Directional Dyslexia)** : *Left/right*, *east/west*, *before/after*, *in front of/behind*, are all difficult abstractions for the dyslexic.
- **Difficulty telling time on a clock with hands** : Dyslectic children are unable sometimes to tell time on clocks. This may result from **prepositional confusion** (*before/after*) or **directional confusion** (*left, right*). As a result children end up having a problem with sequencing and ordering of things, in all levels (sounds, space, time).
- **Difficulty learning to tie shoes** : tying knots also constitutes an abstract sequence of steps, therefore can be a trouble for dyslectic children and reliable sign of dyslexia.
- **Difficulty with cursive writing** : Children tend to see words as entire shapes/objects, therefore they are prone to struggle with basic letter formations and cursive writing .
- **Inability to rhyme** : Rhyming involves swapping out one sound (phoneme) for another, but most of dyslectics face a difficulty parsing (break/reassemble) words in that way. It also constitutes a reliable indicator of dyslexia because it is a pure phoneme task.
- **Struggles to find the right words when speaking** : A lot of dyslectic children have a difficulty recalling words, events and stories since they are also sequences in a specific order. Therefore most of them tend to **pause** or use a lot of **"ums"** and **"ams"**
- **Difficulty recalling the phone number, address** : These also constitute abstract sequences of information not frequently used, making it difficult for dyslectic children to recall.
- **Difficulty Organizing** Dyslectic children have a difficulty organizing some everyday stuff eg. bedrooms, closets, school bags, lockers.
- **Lower self esteem** : Dyslectic children experience daily stress and anxiety, because they tend to face all the previous problems, lowering their confidence and their mood.

Acquired Dyslexia

Compared to the Developmental Dyslexia, only 20% of dyslectics belong to this category, which is a result from trauma or injury to that part of the brain which is directly connected to reading and writing. Late in life this can be the result of a tumor or stroke.

Since developmental dyslexia is more common, and many times identical to the term **dyslexia**, many people argue that acquired dyslexia is not a category of dyslexia at all, but rather a different kind of reading disability because it does not have any genetic background. However, when we examine the etymology of the word **dyslexia** = *dys* · *lexia*, it can literally be defined as the *difficulty with words* and so the term can be widely used for reading difficulties, eliminating any confusion.

2.3.2 Dyslexia by Deficit

Phonological Dyslexia

This is the **most common** type of dyslexia giving the definition to the term **dyslexia** itself. Subjects are unable to manipulate the basic sounds of languages (especially nonsense words), resulting in a *'sticky'* sound ad with repetitions. This is an **auditory** processing problem (a brain-based disorder). Typically it can be considered as **developmental (genetic/inherited)** but some cases can be an acquired type as a result of a stroke or Alzheimer's disease.

Surface Dyslexia

Surface dyslexics may have a visual recognition problem linking words to sounds. Typically it can be considered as **acquired**, ie. not genetic, developing later in life as a loss of former capacity, but some cases cases can be considered as **developmental**. This is a **challenge for the current understanding of dyslexia**. According to Nancy Mather and Barbara Wendling (TODO) "**surface dyslexia** is characterized by difficulty with whole word recognition and spelling, especially when the words have irregular spelling-sound correspondences". This is a paradox because in general in people's mind dyslectics tend to struggle with parts of the words and not the whole words. This is due to the fact that most of of dyslectics (around 80 %) have phonological deficits.Examples of irregular words

Enough, colonel, debt, pretty, they, island, yacht, chaos, wednesday, father, comfortable, answer, earth, friends, have... and hundreds others

Deep Dyslexia

Deep dyslectics subjects may suffer from a loss of capacity to read. Typically it can be considered as **acquired**, ie. not genetic, developing later in life. Often it is because of a trauma or stroke, **affecting the left side of the brain**. There are 2 types of symptoms here :

- **Frequent semantic errors** : Subjects tend to guess at words (entirely replacing the correct ones with related) based on context clues or word shape and size. For example */table/* may be read for */chair/*, */road/*, */dog/*, */canine/*

- **Extreme difficulty reading nonsense words** : Subjects tend to make up words that do not exist at all like */bluck/*, */zub/*. Some of them can not decode or sound out these words. [ad Rhonda B.Friedman12, 1990 proposes that deep and phonological dyslexia may be **opposite endpoints** on a continuum of reading disability, with the surface dyslexia would fall in between.

2.3.3 Dyslexia by Sensory System

Auditory Dyslexia

- **Sound-symbol association problems** The subject is unable to associate **phonemes** with **graphemes**. There is difficulty processing sounds of letters (or groups), realizing that combination of consonants and vowels produce entirely different sounds; that hard, soft and silent sounds are different depending on the position of the letters in the word

- **Auditory discrimination difficulties** : Based on (Critchley), the subject has difficulty differentiating between similarities and differences in the sounds of letters and words, especially in eg. certain consonantal sounds, such as /b/ and /p/, /m/ and /n/ or /d/ and /t/.

- **Difficulties in auditory analysis and synthesis**. Unfamiliar words are difficult to be processed because the subjects lacks of structural analytical skill, so they can not identify morphemes (prefixes, suffixes, root words); ie. they can't establish a connection between these new words and others with similar etymology

- **Auditory sequencing difficulties** The subjects are unable to retain the sequence of sounds long enough in their short-term memory, so they can reproduce them in the correct order ; eg. */ememy/* in place of */enemy/*, */familiar/* in place of */falimiar/*

- **Auditory memory problems** The subjects have difficulty in recalling specific sounds of letters and words which are used in everyday-spontaneous converstations ! For example */dad/* for */father/*. The dyslectic can remember the former easier than the latter one, because they are easier to pronounce and have specific phonics principles. All this difficulty is due to the inefficient system of processing in long-term memory storage.

- **Omissions, additions, substitutions**

  - The subjects tend to **omit** single phonemes or syllabus in word pronunciation. For example */box/* for */boxes/*, */walk/* for */walking/*. This type does affect the meaning derived significantly. But there is another type where the meaning is affected. eg. */bet/* for */bent/*.The most frequent words in which letters are omitted are those starting from */s/* for example, */sit/* for */split/*.
  - Some subjects tend to **add** words or sound units, frequently heard phrases that are automatically associated with each other. For example */the little baby/* for */baby sitter/*
  - Finally, some subjects tend to **substitute** because they think that sound like the correct words but are not. For example */optimist/* for */optometrist/*

- **Mispronunciations** Auditory dyslectic subjects pronounce some vowels incorrectly in some occasions, especially when these words include short vowels. This is happening because they can fully understand the difference between short and long vowel sound. For example, there might a confusion between */mat/* and */mate/* and between */hat/* and */hate/*. Moreover, words that have different initial consonants and the the rest is the same are baffling. For example, words */hut/*, */mut/*, */put/* have identical sound properties and entirely different meaning.

- **Hesitations and repetitions** A lot of subjects may sometimes pause incorrectly between words, thus repeating the preceding phrase or word several times before attempting the problem word.

Visual Dyslexia (common)

Diagnosis a visual dyslexia tends to be **difficult** because the young subjects do not have the experience to perceive that words or phrases are differently written or read from what they know. They believe what they see, they do not have a knowledge baseline to compare and diagnose themselves. Visual dyslexia is reading difficulty resulting from either optical visual problems (**physical causes**) or visual processing disorders (**cognitive/neurological causes**). Common symptoms are :

- Skipping words or lines while reading
- Squinting
- Preference to read in low light
- Rubbing eyes or blinking frequently
- Discomfort reading from monitors and screens
- Headaches / migraines
- Balance or coordination issues
- Poor fluency and comprehension

Therefore, apart from the general symptoms that apply to both subtypes, we are mentioning here more special symptoms of each subtype

Visual Dyslexia (physical) ~ Problems with the eyes

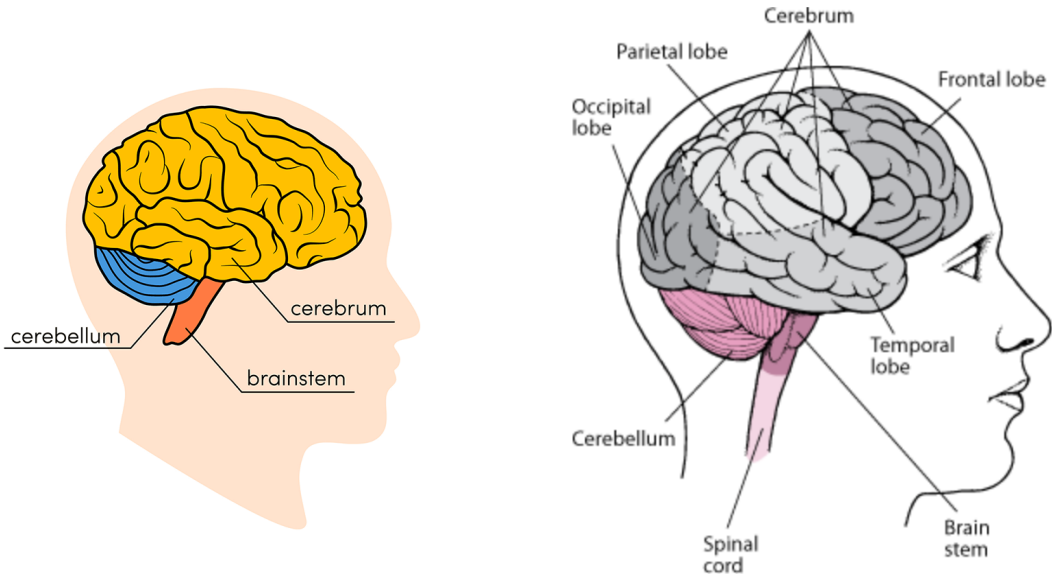
- Distant letters and objects appear blurry
- Nearby letters and objects appear blurry
- An inability to otherwise normally focus your eyes

Visual Dyslexia (cognitive) ~ Visual Stress

- **Instability of text** : Characters appear distorted or shaking.
- **Illusions of light and colour** : Colour may appear between text lines or in the text background
- **Depth perception difficulties** : The depth of words is not accurately perceived by the subjects, leading to balance motion or motion issues
- **Sensitivity to light** : Subjects may suffer from headaches, migraines or fatigues when exposed to different light sources

2.4 Causes

Causes of dyslexia vary based on the type. For **Developmental Dyslexia**, researchers focused on specific hereditary factors. Specific genes are identified to contribute to the development of dyslexia, which is useful and insightful in the direction of early diagnosis children at risk and faster intervening through special education/teaching programs. However, lets deep dive into the brain basics, the complexity of the reading skill, the dyslectic brain and finally the causes.



(a) *Main Areas of the brain 1* **(b)** *Brain Lobes*

Figure 2.3. *Main Areas of the brain*

2.4.1 Brain Basics

There are three main parts of the brain:

- **The Cerebrum** : This is the big wrinkled part of the brain which is divided into **left** and **right** hemisphere. The **left** is directly involved into reading and includes two small areas which are linked to language : **Broca's area** and **Wernicke's area**. Both left and right part are further divided into four (4) **lobes** :
  - Occipital toward the back of the head
  - Frontal near the forehead. **Brocas's area** is here
  - Parietal lobe which is in between
  - Temporal lobe which is also in between. **Wernicke's area** is here
- **The Cerebellum** : It is located at the back of the head. This part of the brain coordinates **physical motor control** (muscle movements, posture, balance). It is slightly implicated in reading
- **The Brainstem** The smallest area among the 3, which is connected to **Spinal Cord**, and coordinates sensory input (hot, cold, pain, bright, loud), breathing, hunger, consciousness, cardiac function, body temperature and involuntary movements (coughing, sneezing). It is not directly implicated in reading process

2.4.2 The Dyslectic Brain

Reading is a **complex** skill. The reader should decode the words, understand the meaning (semantics) and finally read fluently. Neuroimaging and advances in this direction have unraveled the complexity of reading. Reading is cultural invention, so our brain was not neurally wired fore that, so new neural connections were formed, creating a specific circuit just for this complex skill. The success of this circuit is due to the combination of different brain areas (TODO : https://www.omoguru.com/omoblog/lexie/dyslexia-and-reading-in-the-brain/)

Research in neuroscience reveals that brain functions differently in people with dyslexia than those without it. Actually all these structural and neural differences make dyslectic people unable to read, spell, write in an efficient 100% way as a typical reader.

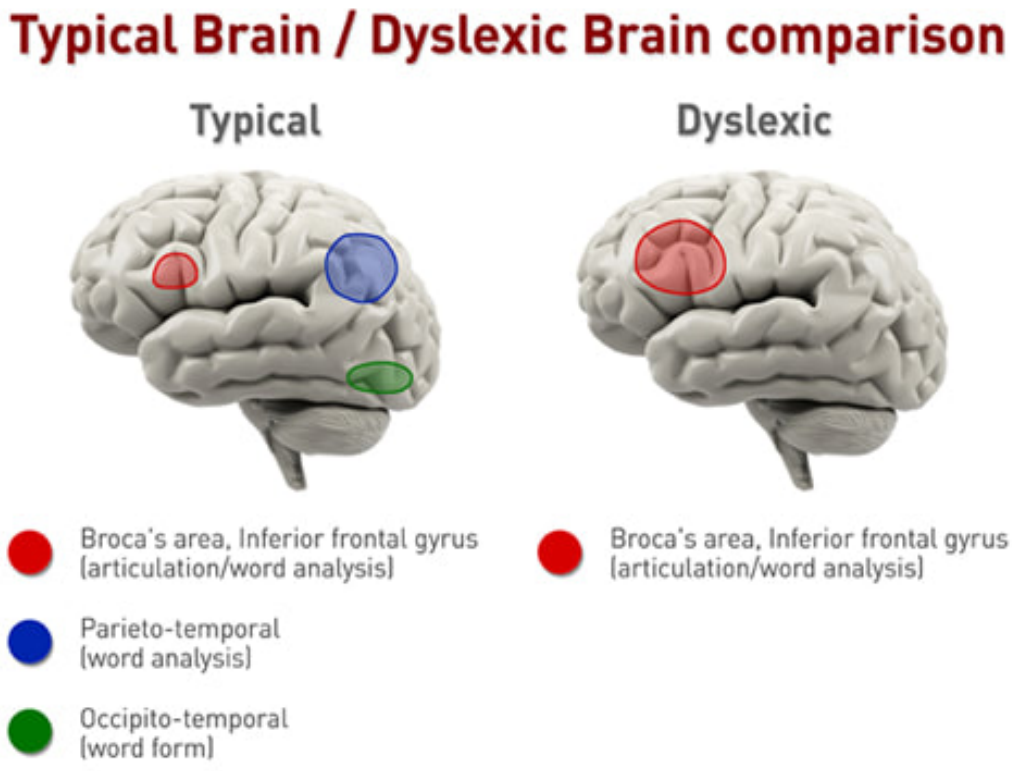


Figure 2.4. *Dyslectic vs Non-Dyslectic brain*

Category	Brain areas
Non-dyslectic brain	The efficient, non-dyslexic readers uses the back areas (parieto-termporal, occipito-termporal) in a very active way, making their fluent since the words are reccognized at ligthing speed. • **Parieto-Temporal area** : The novice reader uses this area in combination with **Broca's area** to **slowly analyze new words**. This is the area that also processes **articulation** • **Occipito-Temporal area** : This is the area where the **words form**, the brain creates new neural connections for these including *spelling, pronunciation, syntax and semantics* • **Broca's area / Inferior Frontal Gyrus**.
Dyslectic Brain	The dyslectic readers do not tap these high powered areas at the back (parieto-termporal, occipito-termporal). Instead the **dyslectic** brain uses only one brain area : • **Broca's area** : This is the area that processes **articulation** and help readers connect sounds with letters and phonemes. In the dyslectic brain, this area is over-utilized and over-activated in order to compensate because the other 2 regions are not effectively fired. This over-utilization leads to inefficiencies. Over time, dyslectic readers tap parts of the right hemisphere to support their reading skills. Since these areas are not created to effectively support reading, the dyslectic readers use a lot of brain effort for less results (inefficient brain activity patterns)

2.4.3 Causes of Dyslexia

Having analyzed the areas which are fired at both dyslectic and non-dyslectic brains, we notice that **Broca's area** is the only area being activated at both type of readers. On the other hand in the non-dyslectic, Parieto-Temporal and Occipito-Termporal areas are not activated; leading to the activation of the right hemisphere to support the reading skills. Even during today, researchers still do not know why **dyslectic-brains** use different parts of their brain to accomplish the same function. Moreover, since **reading** is a complex skill, including the firing of overlapping regions, decoding their interaction constitutes also a complex task.

Therefore, we still do not know the ultimate **root cause** of dyslexia, but we are aware of a specific **intermediate cause**. According to the **"Phonological Processing Impairment Theory"**, the underlying dyslectic brains are not skillful at processing phonemes ; the basic sounds of language, the smallest units of speech that make words different from each other. Therefore, the reader has a difficulty decoding words based on their sounds, since these become "sticky" unable to be broken apart and manipulated easily.

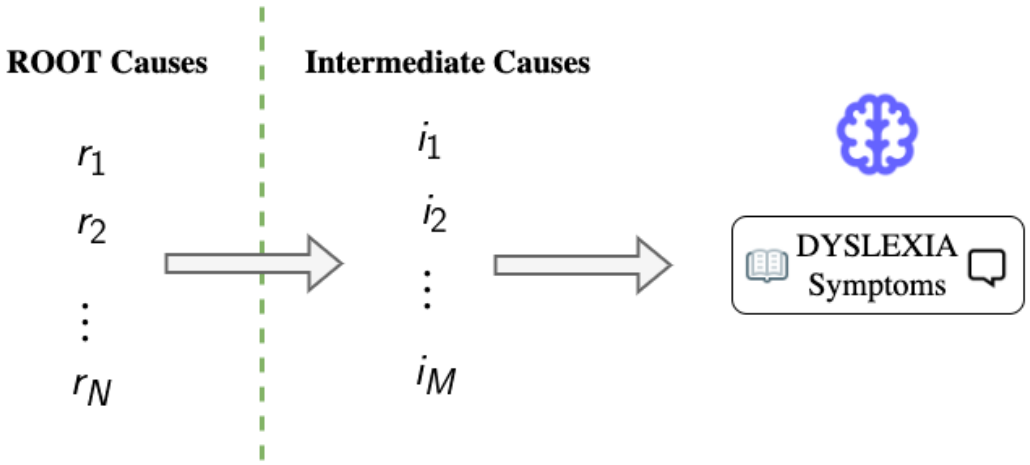


Figure 2.5. *Causes, Symptoms*

- **Genetic Causes** : Discovering the exact genes involved and completely comprehending their activation and expression (or inhibition) is extremely difficult.

  According to the UK's Dyslexia Research Trust, chromosomes 6 and 18 are directly linked to dyslexia. In their research, 50 genetic biomarkers (within 15 brain-expressed genes) were examined. There are strong associations between gene named **KIAA0319** and low performance in tests (*reading, spelling, orthography*) ; a result which is also supported also by other independent studies. Although the function of the **KIAA0319** gene is poorly understood, several studies have shown that it is related to impaired neuronal migration and intercellular adhesion [Zhao et al., 2016]

- **Environmental Causes** :**Developmental dyslexia** (DD) is a multifactorial, specific learning disorder. Susceptibility genes have been identified, but there is growing evidence that environmental factors, and especially stress, may act as triggering factors that determine an individual's risk of developing DD. [Theodoridou et al., 2021]. Moreover, acquired forms of dyslexia (stroke/trauma) can be considered environmental in origin.

2.5 Treatment

Dyslexia is not caused by low intelligence, lack of motivation, laziness or bad parenting. Dyslexic readers, who estimated to represent 5% to 15% of the population, simply need effective reading interventions. Through special remedial teaching programs, they can overcome early learning delays but not completely [P.Tamboer et al., 2016].

2.5.1 Developmental Dyslexia - Phonological

All the teaching programs specified for the most established category of dyslexia, Developmental Dyslexia, include the following methods and content. Taking into consideration principles of the **Orton-Gillingham approach** developed in the 1930's by Samuel Torrey Orton and Anna Gillingham, it has been observed that the most effective teaching programs includes all the following six (6) elements, which follow a specific order of difficulty, starting from simpler structural elements (phonemes, syllables and words, vowels, digraph bllends) and moving on to more advanced (syllable types, roots, preffixes, and suffixes)

- **Personalized** : Respects the specific language needs of the learner

- **Multisensory** : Uses a variety of learning pathways: *seeing, hearing, touching, and awareness of motion*. For example, the dyslectic reader, simultaneously, sees letter *A*, repeat its name and sound, and finally write in the air. Through this way, the readers enhance memory storage and retrieval.

- **Structured, Systematic, Sequential, and Cumulative** : Follows a sequential learning pathway; always acknowledging the past material taught and presenting the new material. Language elements and rules are introduced in a linguistically logical, comprehensive order.

- **Incremental** : From simple well-learned material to more complex, well-structured material, mastering each along the way.

- **Cognitive** : The readers are exposed to a plethora of generalizations and rules directly related to the structure of language.

- **Flexible** :

5) Flexible: Instructors ensure the learner is not simply recognising a pattern and applying it without understanding. When confusion of a previously taught rule is discovered, it is re-taught from the beginning.

6) Personal and Direct: Building a close teacher-student relationship with continuous feedback and positive reinforcement leading to success and self confidence.

2.6 Diagnosis of Dyslexia & Official Diagnostic methods

The *Diagnostic and Statistical Manual of Mental Disorders* (DSM) is the handbook used by health care experts as sourcce of truth in North America, South America, Australia, and many other European countries for diagnosis of mental disorders. It enables professionals to communicate about mental health disorders using a common language. So, even every expert should made a diagnosis based on the criteria presented in DSM-5 (5th version of the handbook). Most of the times, a patient maye present overlapping symptoms and the diagnosis itself is a complex task. Experts use their knowledge, experience and intuition to conclude on a patient's diagnosis.

DSM category for learning disorders, such as **dyslexia** and **dyscalculia**, underwent a significant change between DSM-4 and DSM-5, sparking a lot of controversy and criticism from dyslexia advocacy groups due to its failure to code both as distinct types of Specific Learning Disability (SLD).

Based on on DSM-5, the criteria for diagnosing **dyslexia, dyscalculia and other learning disorders** are :

1. **Criteria A (Key Characteristics)** : Difficulty in mastering reading, writing, arithmetic skills, number sense, number facts or calculation and mathematical reasoning in school age years. The difficulty persisted for at least 6 months and is failed to improve despite the provision of intervention that deal with these difficulties.

2. **Criteria B (Measurement Analysis)** : The difficulty led to the affected academic skill to be substantially and quantifiably elow those expected for the individual's chronological age. It will be assessed through a standardised achievement tests and the analysis is done based on cut off. For individuals aged 17 and older, a documented history of impairing learning difficulties may be substituted for the standardized assessment.

3. **Criteria C (Age of Onset)** : The learning difficulty may begin in the early years of schooling, but may not fully manifest until young adulthood in some individuals until the demands for those affected academic skills exceed the individual's abilities.

4. **Criteria D** : Any other disorder, such as Intellectual Disabilities, auditory or visual acuity problems, other mental or neurological disorders or adverse conditions, such as psychosocial adversity, lack of proficiency in the language of instruction, inadequate instruction that may have plausible explanation for the difficulties being experienced by the individual must be taken into account first before confirming the diagnosis.

2.7 functional Magnetic Resonance Imaging (fMRI)

2.7.1 Brain Sizes

We should understand specific brain sizes before deep diving into the concept of fMRI as presented in Figure 2.6. A Voxel is the unit block of a 3D brain image and represents mm^3 (isotropic voxel resolution) of space in the brain. Each voxel may contain millions of neurons and billions of neural synapses depending on the problem. A Slice is one plane of the brain (x, y, z), while the volume is a 3D image of the brain, recorded at one single timepoint

2.7.2 Brain Sizes Quality Metrics

The quality of the measured data depends on the resolution and the following parameters : The repetition time (TR) is the time required to scan one volume, the field of view (FOV) defines the extent of a slice, eg. $256mm \times 256mm$ while the acquisition time (TA) defines the time required to scan one slice

$$TA = TR - \frac{TR}{n_{slices}}$$

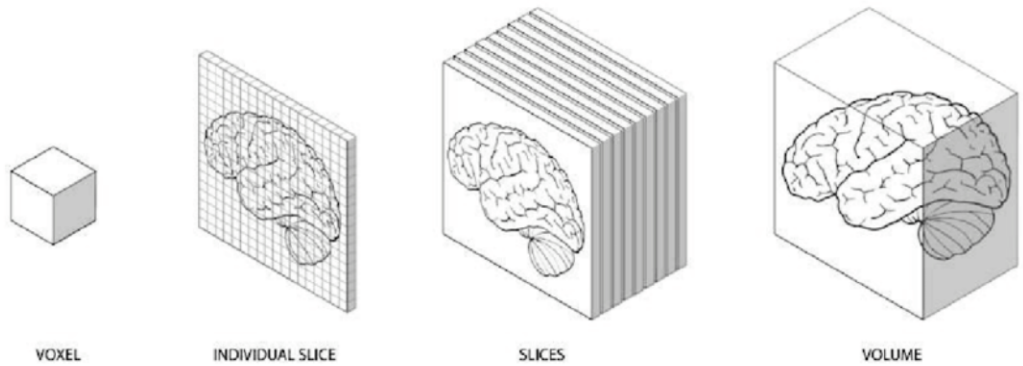


Figure 2.6. *Brain Sizes*

2.7.3 Magnetic Resonance Imaging (MRI) Data Specifics

Magnetic Resonance Imaging (MRI) scanners output their neuroimaging data in a raw data format called **DICOM**, with which most analysis packages cannot work (all the underlying format can be found in Figure 2.8). Special software is used to convert them to a standard format for further neuroimaging analysis. The current standard is **NIFTI (.nii)**, which was also used for all participants files.

- **Image** : The actual data and is represented by a 3D matrix (or 4D for fMRI) that contains a value (e.g. gray value) for each voxel

- **Header** : contains information about the data like voxel dimension, voxel extend etc

Format	Header	Data Types
DICOM	Variable length binary format	Signed and unsigned integer, (8-, 16-bit; 32-bit only allowed for radiotherapy dose)
NIFTI	Fixed-length: 352 byte binary formata	Signed and unsigned integer (from 8- to 64-bit), float(from 32- to 128-bit), complex (from 64- to 256-bit)
NRRD	Extensible with Attached and detatched	
ANALYZE	Fixed-length: 348 byte binary format	Unsigned integer (8-bit), signed integer (16-, 32-bit),float (32-, 64-bit), complex (64-bit)
MINC	Extensible binary format	Signed and unsigned integer (from 8- to 32-bit),float (32-, 64-bit), complex (32-, 64-bit)

Figure 2.7. *MRI file formats*

2.7.4 functional Magnetic Resonance Imaging (fMRI) Data Specifics

Functional magnetic resonance imaging or functional MRI (fMRI) measures brain activity by detecting changes associated with blood flow. This technique relies on the fact that cerebral blood flow and neuronal activation are coupled. When an area of the brain is in use, blood flow to that region also increases.

However, fMRI does not exactly measure electrical activity (compare EEG, MEG, intracranial neurophysiology); but rather it measures the indirect consequences of neural activity (the *haemodynamic response*) as also shown in Figure 2.8.

1. Our brain needs a lot of energy to sustain its functionality

2. Increased function results in increased **blood flow** (oxygen O_2) towards the energy consuming location

3. Immediately after neural activity the blood oxygen level decreases (**initial dip**)

4. Increased flow of new and oxygen-rich blood towards the energy consuming region. After $4-6$ seconds a peak of blood oxygen level is reached

5. After no further neuronal activation takes place the signal decreases again to to the baseline level

2.7.5 MRI vs fMRI

While an MRI scan allows doctors to examine a patient's organs, tissue, or bones, "an fMRI looks at the function of the brain. To better understand the 2 imaging approaches, you can counsel table **??**

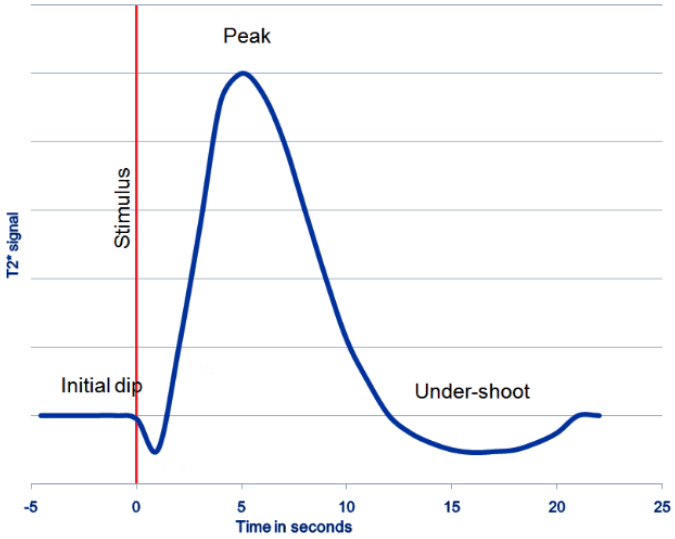


Figure 2.8. *Haemodynamic Response neural activity*

Characteristic	MRI	fMRI
Brain Focus	**Studies Brain anatomy**	**Studies Brain function**
Temporal Resolution	**Low**	**Low**
Spatial Resolution	**High**	**High**
Measures Brain Activity	**Structural/Anatomical**	**Functional (BOLD response)**
Image	High Resolution 3D	Low Resolution 4D

2.7.6 Related Work : Dyslexia, fMRI and Machine Learning

In the case of neurodevelopmental disorders, finding patterns of brain-related features that generalize from smaller samples of data to larger ones, can help understand differences in brain function and development that underpin early signs of risk for **developmental dyslexia** or **spelling disorder**.

In [Laura Tomaz Da Silva and Buchweitz, 2021], the researches used fMRI data from children diagnosed with developmental dyslexia, and typical reader children, which were provided as input to convolution neural networks for learning high-level features. Their accurate and opaque results from the models showed accurate classification of developmental dyslexia, with an *accuraccy* = 94.8% from the brain imaging alone. They also provided automatic visualizations of the features involved that match contemporary neuroscientific knowledge (brain regions involved in the reading process for the dyslexic reader group and brain regions associated with strategic control and attention processes for the typical reader group), which consists their main contribution and allows neuroscience domain experts to interpret the resulting models

In [Yolanda García Chimenoa and Fernandez-Ruanova, 2014], the objective was to ultimately introduce a group of monocular vision subjects, in order to assess whether these subjects are more akin to dyslexic or control subjects. They constructed a tool to compare LDA, MLP, SVM, kNN, Ada Boost machine learning classifiers and classify subjects with dyslexia and monocular vision was obtained, achieving a success rate of 94.8718%.

In [Sofia Zahiaa and Fernandez-Ruanovac, 2020], Statistical Parametric Maps (SPMs) were usedd and a total of 165 3D volumes containing brain activation of 55 children were created. For the classification part, they used three parallel 3D Convolutional Neural Network (3D CNN), one for each reading task which were eventually converged to a single architecture and by using 4-fold cross validation approach they

achieved an F1-score of 67% in dyslexia detection. They conclude that the recognition of dyslexic children is feasible using deep learning and functional magnetic resonance Imaging when performing phonological and orthographic reading tasks.

Chapter 3

Dataset

In this chapter we describe the **MRI Lab Graz** database, by examining the participants and the selection process. We explain how these data are retrieved, and how [Banfi et al., 2020] preprocessed the fMRI data. Last but not least, we present the distribution of the phenotypes (gender, age), or the metadata used for the modeling in the next chapters.

3.1 Participants & Study

[Banfi et al., 2020] performed a study in accordance with the latest version of the Declaration of Helsinki and the national legislation and legally approved by the ethics committee of the University of Graz in Austria. Parents were informed and asked to give a written consent on behalf of their children participated in the study.

A total of 2562 children at the end of 3rd or beginning of 4th Grade (studying in a Austrian school) were selected to be tested under a three-dimensional assessment (1) Standardised classroom tests of sentence reading fluency [Wimmer et al., 2014] (2) Standardised classroom tests of sentence spelling fluency [Müller and R., 2004]) (3) individually administered standardised one-minute word and pseudo-word reading fluency test [Moll et al., 2014]. To sum up, the children undertook 1 spelling test and 3 reading tests in a *sentence* level and a *word or pseudo-word* level

3.1.1 Participants 1st selection

From this large sample [Banfi et al., 2020] selected three groups based on their spelling and reading performance (mean performance on all 3 reading tests)

GROUP	**Spelling performance**	**Mean Reading performance (3 tests)**
Spelling Disorder (SpD)	$performance \leq 20\%$	$performance_{mean} \geq 25\%$
Dyslexia (DL)		$performance_{2outof3} \leq 20\%$ and $performance_{3rd} \leq 43\%$
Skilled (TD)	$performance \in (25\%, 85\%()$	$performance_{mean} \in (25\%, 85\%()$

Along with the previous assessment, all participants must be compliant to the following requirements.

1. Chidrens' first language should be German
2. A $IQ_{non-verbal} \geq 85$ is required [Weiß and R., 2006]

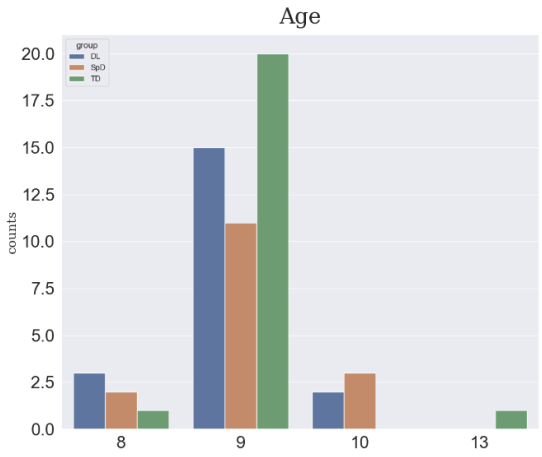


Figure 3.1. *Age*

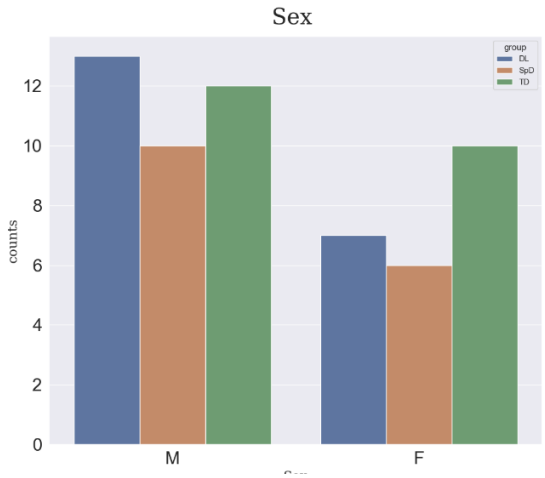


Figure 3.2. *Sex*

Figure 3.3. *Phenotypic Data Distribution*

3. No identified sensory or neurological deficits
4. no clinical ADHD diagnosis
5. Above threshold score on a standardized pareental questionnaire for attention deficits [Döpfner et al., 2008]

According to this screening process, specific children were initially selected to be assessed. However, only 70% of the children in the SpD sample and 90% in the DL group were in compliance with the German diagnostic guidelines for an official clinical diagnosis [Galuschka et al., 2016]. Altogether 71 children were assessed

$$participants \sim (21\ \text{SpD}, 23\ \text{DL}, 27\ \text{TD})$$

3.1.2 Participants 2nd selection

From these 71 children assessed, several were excluded from the analysis and the final dataset for the following reasons :

- 12 (out of 71) were excluded due to excessive movement during the fMRI session.
- 1 (out of 71) was also excluded because the field of view (FOV) had been wrongly defined, leading to the cutting of an important part of the frontal lobe

$$participants \sim (16\ \text{SpD}, 20\ \text{DL}, 22\ \text{TD})$$

3.1.3 Participants Phenotypes Distribution

In our database, there is one *.tsv* file describing 3 different phenotypic data. These are : **Age, Gender** and **Disease Group**. Disease Group is considered as the class label used for the machine and deep learning classification frameworks built. The rest 2, **Age** and **Gender** are used to construct the initial graph of the **G-CNN** model of the underlying deep learning framework

As we can clearly the dataset is balanced among both **Genders**, while the **Age** phenotype is imbalanced. The ages of the participants are distributed between 8 and 13, with a mean value of $\mu_{age} = 9.05$ and a standard deviation of $\sigma_{age} = 0.68$.

3.2 Experimental Stimuli

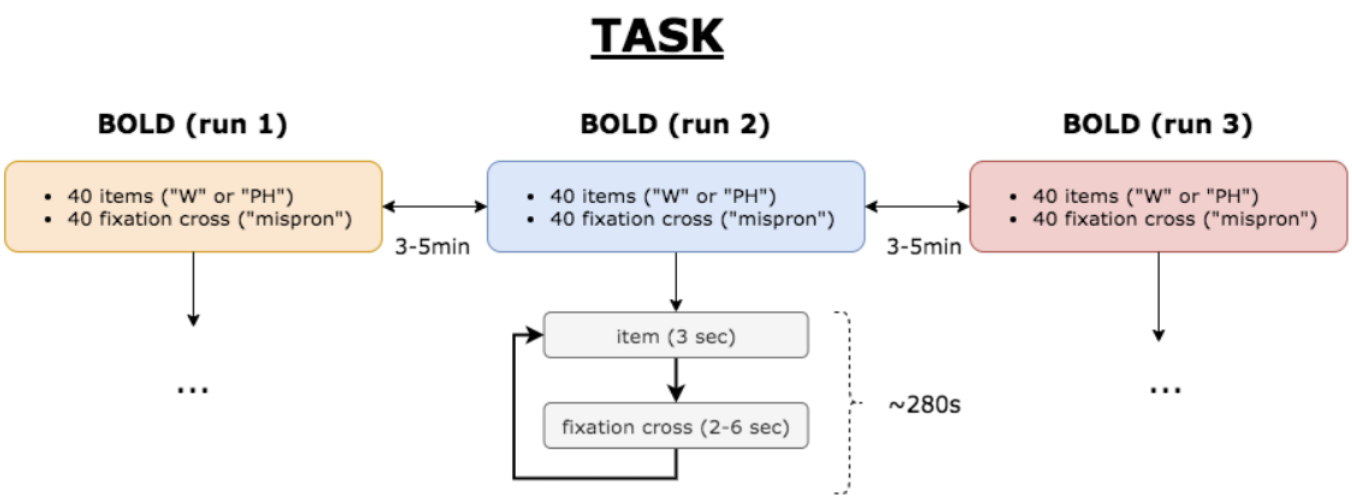


Figure 3.4. *The structure of the experiment* [1]

The whole task process can last up to 1040s for each subject (on average as explained later on) and in total up to $58 \times 1040 \sim 60320 sec \sim 17 hours$ for all the subjects

1. Every subject is asked to read 120 items (60 words denoted as "W" and 60 pseudohomophones denoted as "PH").
   - Pseudohomophones were built from the base word by exchanging one phonologically identical grapheme, and are consisted of 3 to 8 letters
   - Each word or pseudohomophone was shown at least 30 items after its paired stimulus had been presented
2. All 120 items were split into 3 consecutive runs of 40 items each, separated by short breaks of 3-5 min in order to prevent fatigue effects in children
3. For each run
   (a) Each item was presented white in a black background for 3 s in an event-related design.
   (b) After it disappeared a fixation cross was displayed for a duration in the range of 2000 and 6000 ms (average = 4000ms)

3.3 fMRI Data Acquisition

Advances in brain imaging techniques have provided a significant opportunity for the study of brain development in humans. Functional magnetic resonance imaging (fMRI) is the most promising and broadly used imaging technology that is also safe for use in pediatric populations [Thomason, 2009]

All these 58 children were given direct analytical instructions in a silent room so that they feel calm and that the correct execution of the experiment is validated. During the task, imaging was performed on 3.0T Skyra scanner (Siemens Healthineers, Erlangen, Germany) using a 20-channel head coil. Two (2) types of images were captured :

- High-resolution 3D-T1 MPRAGE **structural scans** (TR = 1600 ms, TE = 1,81 ms, FOV = 224 mm, $angle_{flip}$ = 8 degrees, 176 slices, voxel resolution $1 \times 1 \times 1\ mm^3$)

- BOLD-sensitive T2*-weighted **functional images** were acquired using a single shot gradient-echo EPI pulse sequence (TR = 2340 ms, TE = 33 ms, FOV = 192 mm, $angle_{flip}$ = 90 degrees, 34 slices with 0.3 mm gap, voxel resolution $3 \times 3 \times 3mm^3$, descending acquisition order).

In order for the brain scans and the images to be stable and not blury, head motion was limited using firm paddings around the head. Verbal responses of participants were recorded via an MR compatible microphone (FOMRI-III, OptoacousticsLtd., Moshav Mazor, Israel). Stimuli were presented using the Software Presentation (Neurobehavioral Systems, Albany, CA). [Banfi et al., 2020]

3.4 fMRI Data Preprocessing

Kind of Image	Image Shape
3D Anatomical T1w	$176 \times 224 \times 224$
4D Functional (3 BOLDS) T2w	$(64 \times 64 \times 34) \times 130$

3.4.1 Anatomical/Structural Data Preprocessing (3D-T1 MPRAGE)

According to [Banfi et al., 2020], the T1-weigthed (T1w) images were preprocessed based on the following preprocessing pipeline

- **Correction for intensity non-uniformity** (INU) by using N4BiasFieldCorrection (Tustison et al. 2010), distributed with ANTs 2.2.0 (Avants et al. 2008, RRID:SCR_004757)

- **Skull-stripping by using a Nipype** implementation of the "antsBrainExtraction.sh" workflow (from ANTs), using OASIS30ANTs as target template

- **Spatial Normalisation** to the ICBM 152 Nonlinear Asymmetrical template version 2009c (Fonov et al. 2009, RRID:SCR_008796) by using Nonlinear registration "antsRegistration.sh" (ANTs 2.2.0) (with the brain-extracted versions of both T1w volume and template)

- **Brain Tissue Segmentation** of (a) cerebrospinal fluid (CSF) (b) white-matter (WM) and (c) gray-matter (GM) was performed on the brain-extracted T1w using fast (FSL 5.0.9, RRID: SCR_002823, Zhang et al., 2001).

3.4.2 Functional Data Preprocessing

For each subject, there are 3 BOLD runs available. According to [Banfi et al., 2020], the following data preprocessing was performed.

- **Skull-stripping** the reference volume by using a custom methodology of fMRIPrep

- **Co-registration** (freedom degrees = 9 for the distortions of the remaining in the BOLD reference) to the T1w by using flirt [Jenkinson et al., 2001] with the boundary-based registration [Greve et al., 2009] cost-function

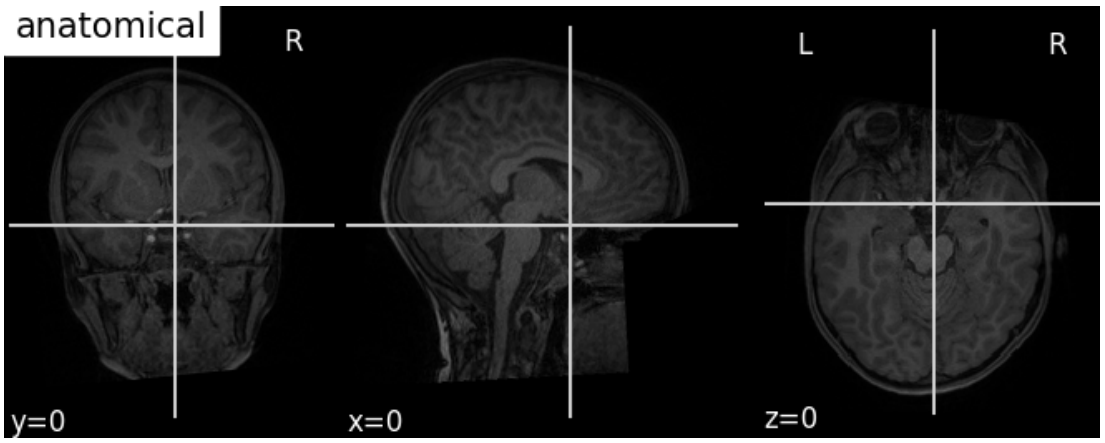


Figure 3.5. *Subject 047EPKL011005's raw anatomical T1w image. The dimensions are* $176 \times 224 \times 224$

- **Head-motion parameters estimation** (transformation matrices, 6 rotation/translation parameters) with respect to the BOLD reference by using mcflirt [Jenkinson et al., 2002]. This step was conducted before any spatiotemporal filtering
- **Slice-time correction** using 3DTshift from AFNI 20160207 [Cox et al., 1997]
- **Resampling of the BOLD time-series** onto their original space through composite transformation in order to correct head-motion and susceptibility distortions.
- **Resampling of the BOLD time-series** onto MNI152NLin2009cAsym standard space, generating a preprocessed BOLD run in MNI152NLin2009cAsym space.
- **Time-series confounding calculation** based on the previous preprocessed BOLD
  - **framewise displacement (FD), DVARS** were calculated by using their implementations in Nipype (following the definitions by [Power et al., 2014]).
  - **3 region-wise global signals**, extracted within the (a) CSF, (b) the WM, and (c) the whole-brain masks.
- **Component-based noise correction** [Behzadi et al., 2007] through the extraction of a set of physiological regressors
- **Gridded (volumetric) resampling** by using "antsApplyTransformation.sh" (ANTS), configured with Lanczos interpolation to minimise the smoothing effects of other kernels [Lanczos and C., 2007]
- **Smoothing** (Gaussian kernel of 8 mm) using SPM12 software (v6906; Wellcome Department of Imaging Neuroscience, London, UK), in a MATLAB 2016a environment (MathWorks Inc., Natick, MA).

3.5 Phenotypes

3.6 fMRI Feature Extraction

Before deep diving into the feature extraction, it is of utmost importance to state the number of samples and data we have in our hands are the pre-processing step. We deal with 58 subject, with 3 sessions fMRI data. For each subject and session we have the following data :

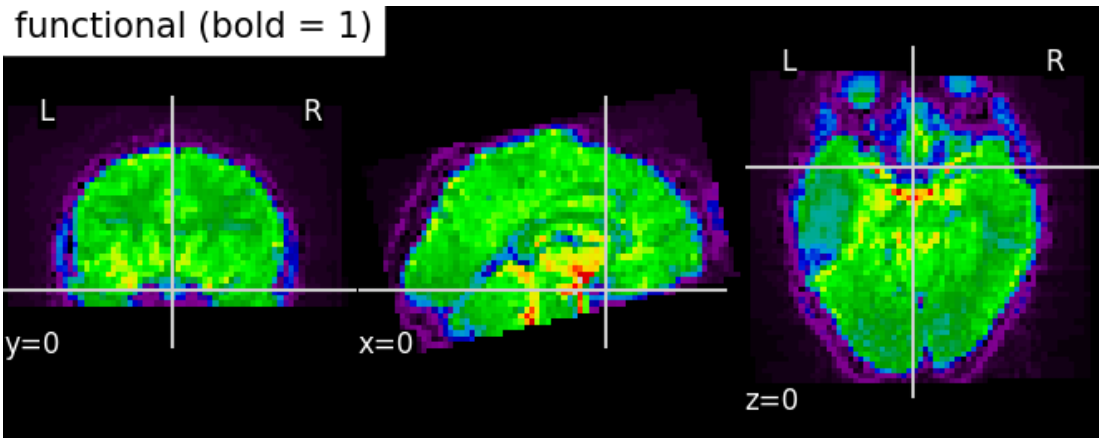


Figure 3.6. *Subject 047EPKL011005's EPI functional (BOLD 1) T2w image. The dimensions are* $(64 \times 64 \times 34) \times 130$. *Timestamps* $= 130$

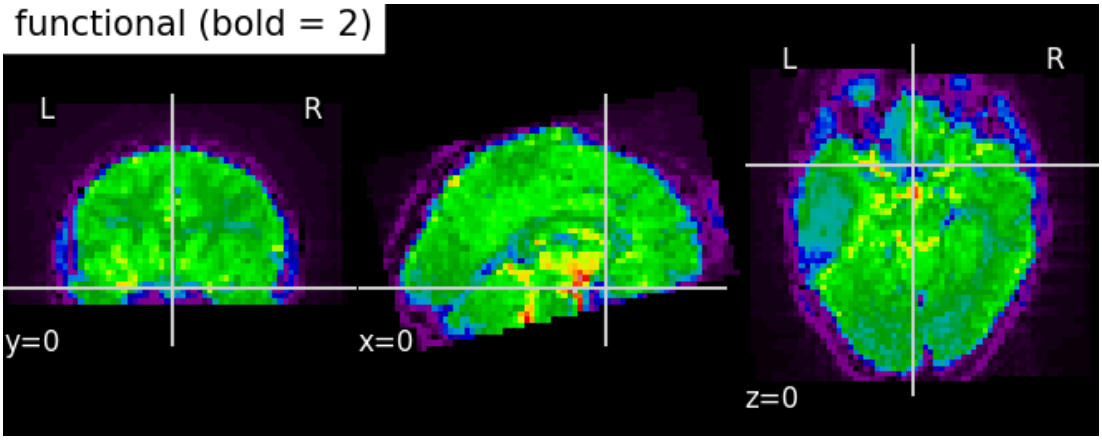


Figure 3.7. *Subject 047EPKL011005's raw functional (BOLD 2) T2w image. The dimensions are* $(64 \times 64 \times 34) \times 130$. *Timestamps* $= 130$

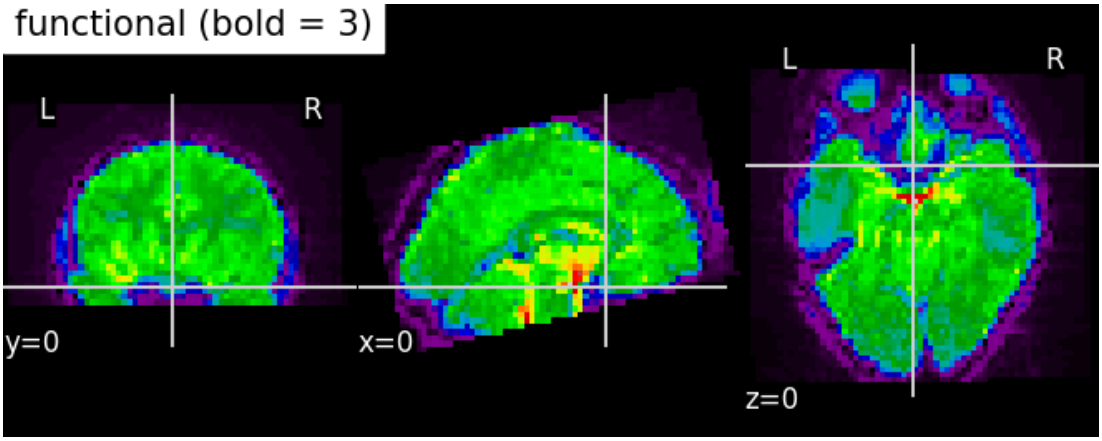


Figure 3.8. *Subject 047EPKL011005's raw functional (BOLD 3) T2w image. The dimensions are* $(64 \times 64 \times 34) \times 123$. *Timestamps* $= 123$

- The single 1 anatomical high-resolution mask image ($176 \times 224 \times 224$) in "nii.gz" format
- The 3 functional images, once for each BOLD run ($64 \times 64 \times 64 \times 130$) in "nii.gz" format
- The 3 events files, one for each BOLD run, in ".tsv" format

According to SPM documentation, each and every one of the sessions had a repetition time of $TR = 3$s.

The task comprises 3 conditions

- **"W"** for word
- **“PH”** for pseudohomophone
- **“mispron”** for a fixation cross

Although the data is already preprocessed, there are still many steps needed for the data to be in an appropriate form for input in our models. Our goal is to produce correlation matrices that will be fed into our **machine learning** and **deep learning** models. There are many different kinds of post-analysis that can be applied to extract these features (correlation matrices). Briefly, these kinds are :

- GLM First Level Analysis
- Dictionary Learning Analysis
- Atlas Model Analysis

In this diploma thesis, the correlation matrices used are the ones produced by the *Atlas Model Analysis*. However for the sake of completeness, in this diploma thesis all methods are presented in Figure !3.9

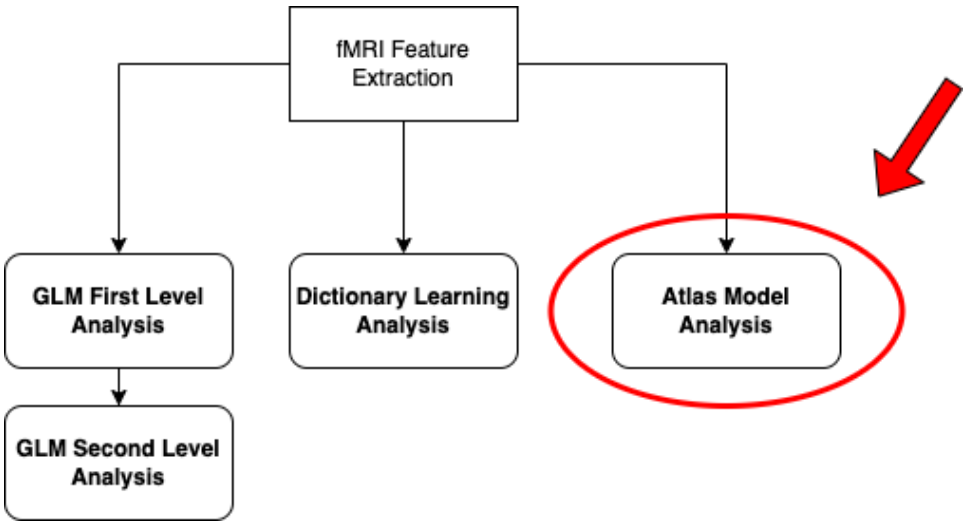


Figure 3.9. *fMRI Feature Extraction techniques*

3.6.1 GLM First Level Analysis

Theoretical Background

The GLM stands for **General Linear Model** and has been the most widely used technique for analyzing task-based fMRI experiments for the past 30 years and is considered as the default method provided by vendors for their clinical fMRI packages. The purpose of this analysis is to take each subject in turn, realigning, smoothing and standardising subject’ scans so that we can identify **regions of activation** in relation to e.g. MNI space.

The First Level Analysis is a **single-session, single-subject** GLM level analysis. In this diploma thesis, we deal with 58 subjects, 3 sessions and 3 voxels. We used the nilearn GLM 1st Level Model library for all the underlying work presented here.

The General Linear Model (GLM) representation of an fMRI experiment retains the same basic form

$$Y = X \cdot \beta + \epsilon$$

Stated in words, the GLM says that Y (the measured fMRI signal from a single voxel as a function of time) can be expressed as the sum of one or more experimental design variables X, each multiplied by a weighting factor (β), plus random error (ϵ). In GLM both Y and remain as single column vectors containing fMRI signal data (y_i) or error estimates ($_i$) respectively for a single voxel at successive time points ($i = 1ton$). The experimental **design matrix** (X), however, is typically much more complex, consisting of perhaps 5-10 columns. Each new column of X would be constructed by the investigator to reflect a specific factor ("**regressor**") thought to influence the outcome of the experiment.

- **Essential regressors** (also known as regressors of interest) are a set of idealized predictions of what the hemodynamic response function (HRF) should look like if a voxel of interest became activated due to a task or stimulus.
- **Nuisance regressors** – experimental factors that confound the analysis (such as head motion or signal drifts) but which are of no particular interest by themselves

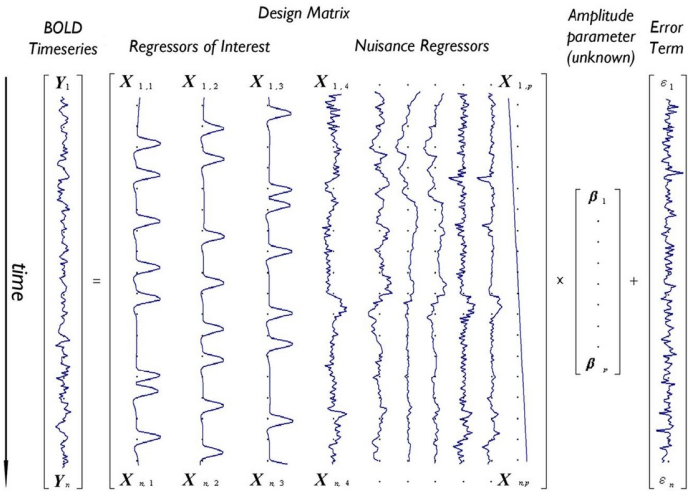


Figure 3.10. *Depiction of the General Linear Model (GLM) for a voxel with time-series Y predicted by a design matrix X including 10 effects (three regressors of interest - e.g., tasks A,B,C - and seven nuisance regressors - e.g., six motion parameters and one linear drift).*

Performing GLM First Level Analysis

The before-mentioned approach is a **single-session** approach. In our case we have 3 sessions, so we propose the following framework as you can see in Figure 3.11

For each **subject-session** :

1. We create a *FirstLevelModel()*, which will produce the design matrix using the information provided through the **events** file. The following parameters are given

Parameter	Value	Explanation

t_r	3	According to SPM documentation, each and every one of the sessions had a repetition time of 3 sec.
noise_model	"ar1"	This specifies the noise covariance model: a lag-1 dependence. An alternative choice is to use an ordinary least squares model (ols) that assumes no temporal structure (time-independent noise). While the difference is not obvious we should rather stick to the ar(1) model, which is arguably more accurate.
drift_model	polynomial	
drift_order	10	Here we specify a set of polynomials (or regressors). We used 10 polynomials.
standardize	False	This means that we do not want to rescale the time series to mean 0, variance 1
hrf_model	"spm"	Glover canonical model of spm. The hemodynamic response model (filter) is used to convert the event sequence into a reference BOLD signal for the design matrix. We could try “spm + derivative”. In that case we still perform the contrasts and obtain statistical significance for the main effect —not the time derivative => The only effect of the derivatives inclusion is discounting timing misspecification from the error term, leading to a variance decrease and statistical significance increase
high_pass	0.0135	• To remove spurious low-frequency effects related to heart rate, breathing and slow drifts in the scanner signal, the standard cutoff frequency is $1/128Hz \sim 0.01$Hz. • However it is relative to the problem. The cutoff period should be set as the longest period between 2 trials of the same condition multiplied by 2. We customized the function ‘calculate_longest_period_between_conditions()‘ that finds the longest duration period between **"PH"** (or **"W"**) among all subject. You can inspect an events dataframe sample in the Figure 3.12 • Therefore the longest period is $t_{longest} = 37.0481$ sec so $$f_{cutoff} = \frac{1}{2 \cdot t_{longest}} = 0.0135Hz$$

2. The .fit() functionality of *FirstLevelModel()* function creates the design matrix and the beta maps. Moreover, the following parameters were configured

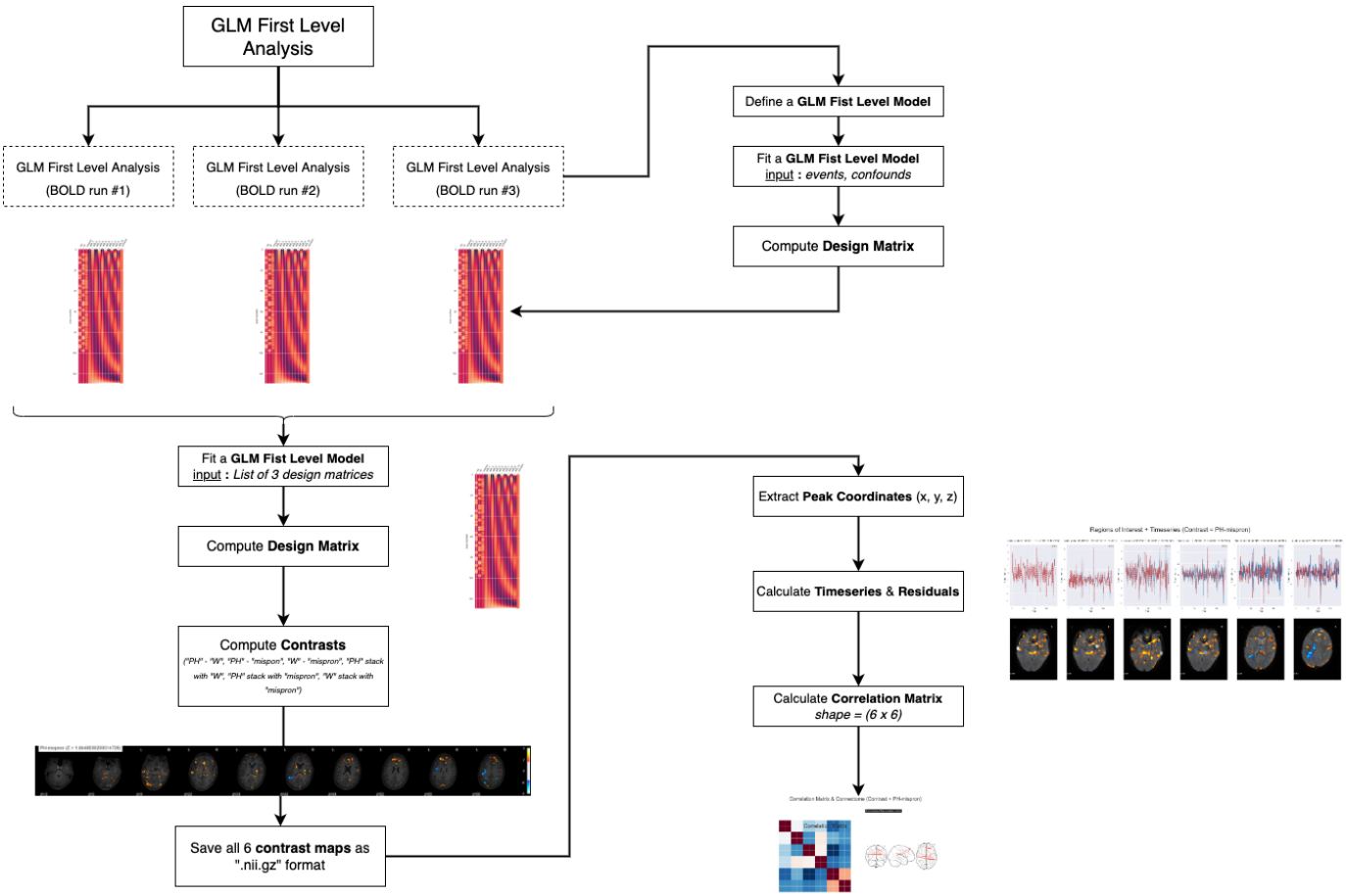


Figure 3.11. *GLM First Level Analysis (per session). This diagram was built in draw.io*

	longest period
sub-047EPKL011005	24.765299999999968
sub-047EPKL011011	24.765299999999968
sub-047EPKL011012	24.765299999999968
sub-047EPKL011041	24.765299999999968
sub-047EPKL011047	24.7654
sub-047EPKL011049	24.7654
sub-047EPKL011056	24.765199999999993
sub-047EPKL011051	24.765199999999993

Figure 3.12. *Sample of events dataframe*

Parameter	Value	Explanation
events	events	This is the events dataframe. One for each BOLD run (session)

confounds	confounds	This is the confounds datafarme • A problematic feature of fMRI is the presence of uncontrolled confounds in the data, due to scanner instabilities (spikes) or physiological phenomena, such as motion, heart and respiration-related blood oxygenation fluctuations. Side measurements are sometimes acquired to characterize these effects. Here we don't have access to those. • In the dataset, specific confounds are not given. However, the function *nilearn.image.high_variance_confounds()* can be used, which calculates the confounds of signals extracted from input signals (functional BOLD image) with highest variance, given the percentile of signals with highest variance (percentile = 1%). • Given that the number of signals are 130 (runs 1,2) and 123 (run 3) respectively, the algorithm (a) Computes the sum of squares for each signal (no mean removal) (b) Keeps a given percentile of signals with highest variance (percentile). (c) Computes an SVD of the extracted signals. (d) Returns a given number (*n_confounds*) of signals from the SVD with highest singular values. You can inspect a confounds dataframe sample in the Figure 3.13

	0	1	2	3	4
0	-0.03474278	-0.022853563	0.050185133	-0.14320432	0.09307721
1	-0.05076641	0.06351458	0.022782475	0.0048133135	0.08592908
2	0.0748547	-0.01224988	0.16533768	0.13849506	0.058348186
3	0.13041061	0.09265873	-0.023908492	0.10159505	0.04723967
4	0.08631882	-0.00814835	-0.09345957	0.075335	0.01623372
5	-0.078255296	0.056549486	-0.021983052	0.06659957	0.019388057
6	-0.059309285	0.0566825	-0.01270308	-0.015607471	0.0069111655
7	0.030656075	-0.12669355	-0.004967785	-0.013116378	-0.06369121
8	0.10007117	-0.010011288	0.018034713	-0.11510462	0.15913163
9	0.041908897	0.008061619	0.034373313	-0.07829267	0.12184648
10	0.10418947	-0.07670601	0.10875339	-0.06616994	0.11699247
11	0.0070005013	0.047626596	-0.06435491	0.058852993	0.0061871153
...					
126	-0.041240092	0.17864566	0.020255709	-0.04379363	-0.1200203
127	0.028409224	-0.046743225	-0.085570335	-0.011576328	-0.095304176
128	-0.15333112	0.06302357	-0.25614765	-0.10529339	-0.008536898
129	-0.13500485	-0.079087004	0.27684906	-0.020455725	-0.15768296

Figure 3.13. *Sample of confounds dataframe*

3. **Design Matrix** During the analysis, 3 experimental conditions are met and 10 polynomials are used. The design matrix includes 3 main columns corresponding to 3 experimental conditions (“W”, “PH”, “mispron”) followed by 11 (10 + 1 constant) columns describing low-frequency signals (drifts) and a constant regressor (10 polynomials are used). The function *nilearn.plotting.plot_design_matrix()* is also used to plot the respective design matrix. A sample design matrix is presented in the Figure 3.14

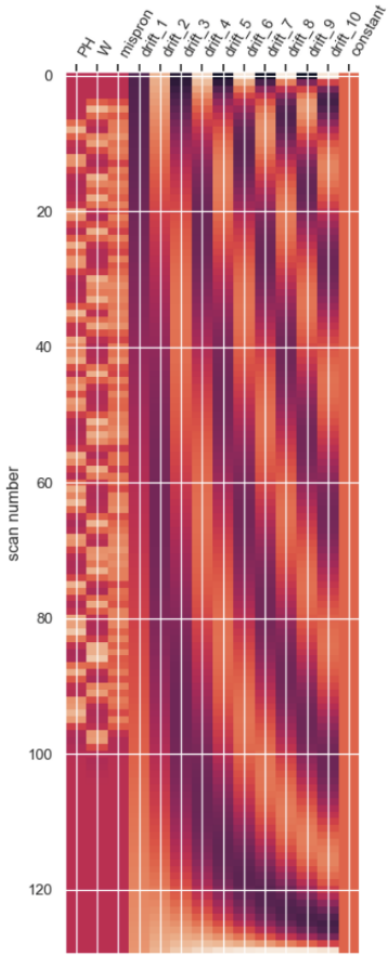


Figure 3.14. *Subject 047EPKL011005’s design matrix of functional (BOLD 3) T2w image. (123 Timestamps, 10 polynomials)*

4. **Compute contrasts (detecting voxels with significant effects)** : To access the estimated coefficients (β of the GLM model), we create a dictionary of 6 contrasts by properly manipulating the design matrix.

   - **pseudohomophone** substract **word** (“PH” - “W”)
   - **pseudohomophone** substract **fixation** (“PH” - “mispron”)
   - **word** substract fixation (“W” - “mispron”)
   - **pseudohomophone** stack with **word** (“PH” stack with “W”)
   - **pseudohomophone** stack with **fixation** (“PH” stack with “mispron”)
   - **word** stack with **fixation** (“W” stack with “mispron”)

   To compute the contrast map for these 3 contrasts, the function *compute_contrasts()* of *FirstLevelModel* was used. The parameters were configured as following :

Parameter	Value	Explanation
contrast_def	contrast	The contrast dataframe
stat_type	"t"	Can be “t”, “F”. Here “t” was used for linear t-contrasts

The function returns **Nifti1Image** objects.

5. **Visualize the thresholded statistical map** : Having calculated the contrast maps 6× **Nifti1Image** objects , it is of utmost importance to visualize them through the use of the function *plotting.plot_stat_map()*. In the beginning, we decided to display them on top of the **average functional** image, but for higher resolution we selected the **anatomical** image. Please refer to Figures 3.15, 3.16 , 3.17, 3.18, 3.19 , 3.20

Parameter	Value	Explanation
stat_map_img		The Nifti1Image object of the contrast map
bg_img		The anatomical image was used
threshold	3	• Initially, an arbitrary threshold of 3.0 was used. But he threshold should provide some guarantees on the risk of false detections (aka type-1 errors in statistics). • One suggestion is to control the false positive rate (fpr, denoted by alpha) at a certain level, e.g. 0.001: this means that there is 0.1% chance of declaring an inactive voxel, active. We used the function *glm.threshold_stats_img()* which given the following, returns the optimized threshold – the 10% chance of declaring (1) a “word” as a “pseudohomophone”(2) a “fixation” as a “word” (3) a “fixation” as a “pseudohomophone” – the number of clusters = 10 (10 voxels are reported)
display_mode	"z"	z-scale is always used for comparable results and optimized inspection
cut_coords	10	The MNI coordinates of the point where the cut is performed. We selected “10” brain scans to be produced
black_bg	True	Used to set black background

6. The previous analysis is for a **single** subject and a **single** BOLD run (session). However in this diploma thesis we deal with a dataset which has 3 BOLD runs (sessions). Here

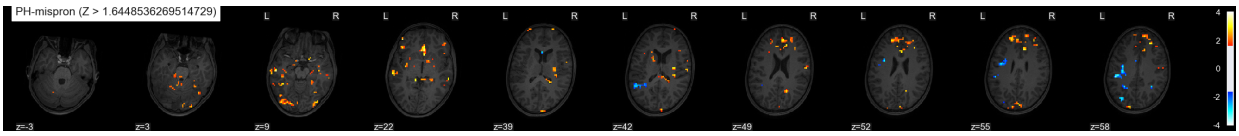


Figure 3.15. *Subject 047EPKL011005's statistical map of functional (BOLD 3) T2w image for the contrast map ("PH" - "mispron"). Contrast map produced with "effect_size". Optimized z threshold is calculated*

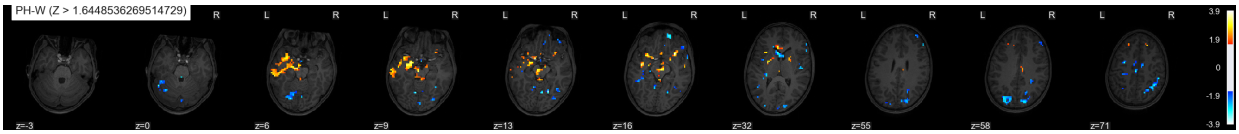


Figure 3.16. *Subject 047EPKL011005's statistical map of functional (BOLD 3) T2w image for the contrast map ("PH" - "W")*

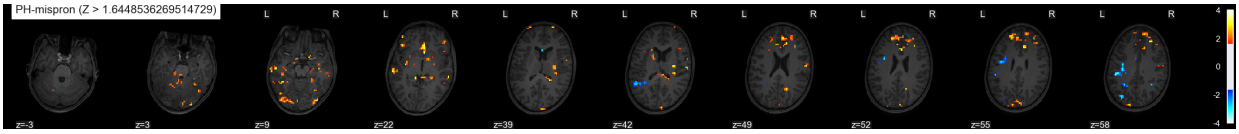


Figure 3.17. *Subject 047EPKL011005's statistical map of functional (BOLD 3) T2w image for the contrast map ("W" - "mispron")*

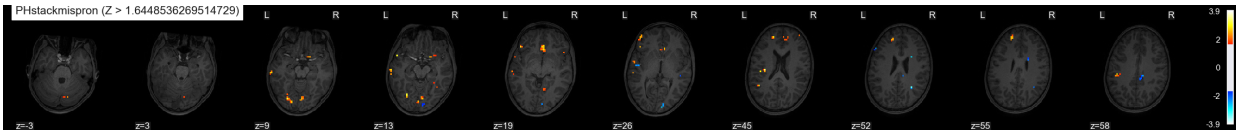


Figure 3.18. *Subject 047EPKL011005's statistical map of functional (BOLD 3) T2w image for the contrast map ("PH" stack "mispron"). Contrast map produced with "effect_size". Optimized z threshold is calculated*

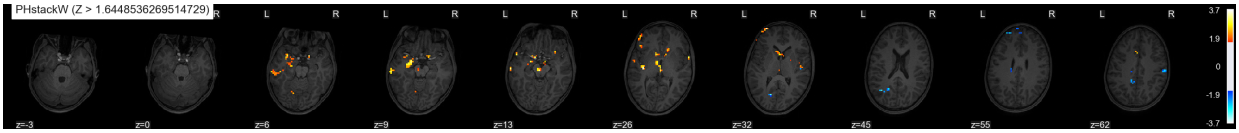


Figure 3.19. *Subject 047EPKL011005's statistical map of functional (BOLD 3) T2w image for the contrast map ("PH" stack "W")*

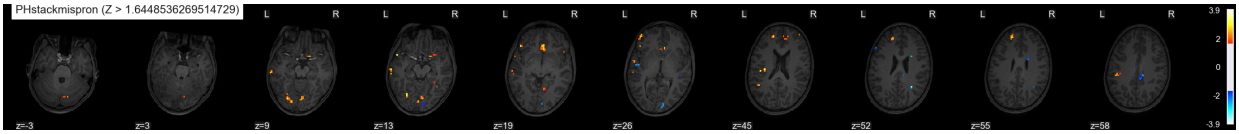


Figure 3.20. *Subject 047EPKL011005's statistical map of functional (BOLD 3) T2w image for the contrast map ("W" stack "mispron")*

(a) The function *first_level_analysis* is implemnted that calls the function *first_level_analysis_one_session* and retrieve the **design matrix** of each session separately (we are not interested in the contrast maps of each session)

(b) All 3 design matrices are combined into a list

(c) A *FirstLevelModel()* model object is defined and fit by providing as argument the list with the 3 design matrices. After fitting the **new design matrix** is retrieved

(d) All the 6 contrasts maps (subtracting conditions, stacking conditions) are then calculateed

(e) All 6 contrast maps are saved as ".nii.gz" files (features extracted)

7. **Extract peak coordinates & timeseries & residuals from the contrast z maps** : The method *extract]_peak_coordinates_and_timeseries* was implemented which accepts as parameters (1) the trained GLM model and (2) the contrast maps and calculates

- **PEAKS COORDINATES (x,y,z)**
  - For each one of the 6 contrast maps, the peak coordinates (for all 3 axis x,y,z) were retrieved and stored as *"coordinates_[contrast_map].csv"* file in the **features** directory
  - For those peak coordinates (that may vary in length for each one of the 6 contrast maps) we keep only the top clusters. Here *top_clusters* = 6.
  - All peak coordinates subsets were concatenated into a final dataframe which was also stored as *"coordinates_ALL.csv"* file in the **features** "directory. Its shape is

$$(6 \text{ contrast maps} \times 6 \text{ top clusters}, 3 + 1) = (36, 4)$$

  - We have 3 axis and 1 final column indicating the task (the contrast map type), as you can clearly see in Figure 3.21

Cluster ID	X	Y	Z	task
1	-39.44595680013299	-12.796430915594101	21.575822584331036	PH-W
2	32.79323948547244	12.63066628575325	13.257118605077267	PH-W
3	2.4862493500113487	-42.09604287147522	13.95265257358551	PH-W
4	-0.35328948497772217	24.945840939879417	31.58453032746911	PH-W
5	26.580127831548452	-4.272488832473755	23.789004415273666	PH-W
6	-30.343546010553837	5.451224282383919	21.332487534731627	PH-W
1	-57.43560720235109	-21.04721885919571	16.657799318432808	PH-mispron
2	29.58878179639578	-12.637071833014488	19.112279411405325	PH-mispron
3	59.492118172347546	-39.44438727200031	15.092543620616198	PH-mispron
4	11.46615519747138	-36.75701351463795	18.28232367709279	PH-mispron
5	35.60506881028414	35.52636103332043	40.450699616223574	PH-mispron
6	-3.591756910085678	44.3802635371685	61.48061067610979	PH-mispron
1	-0.3758070506155491	7.715930685400963	25.418518278747797	W-mispron
2	-0.6621478125452995	46.258020743727684	68.51278111711144	W-mispron
3	11.46615519747138	-36.75701351463795	18.28232367709279	W-mispron
4	50.751476123929024	11.999870538711548	16.720326006412506	W-mispron
5	23.55840663984418	54.71531067788601	53.47652620449662	W-mispron

Figure 3.21. *Subject 047EPKL011011's extracted peak coordinates for all 6 contrast maps*

- **TIMESERIES & RESIDUALS**
  - For each coordinates subset $(6, 3 + 1) = (6, 4)$ we define a *input_data.NiftiSpheresMasker()* object which will be used to extract the timeseries from concatenated functional imaging within the sphere.

Parameter	Value	Explanation
detrend	True	
standard-ize	True	normalized to variance 1
low_pass	0.1	

high_pass	0.01	both used so → band passed
smoothing_fwhm	6	
t_r	3	equals to the number of conditions

- **Timeseries Truth array** :The Sphere Masker Model is fit by using as input, the single functional image (after preprocess and concatenate all functional images for all 3 sessions). The image has a shape of

$$(64, 64, 64, 130 + 130 + 123 \text{ timestamps}) = (64, 64, 64, \text{ timestamps})$$

The output of this fit and transform operation is the **timeseries truth array**, with a shape of

$$(381 \text{ timestamps }, 6 \text{ top clusters })$$

- **Timeseries predictions array** : The attribute *fmri_glm.predicted* gives the timeseries predictions array for each session separately. The result is concatenated in a timestamp level $(127, 6) \times 3 \rightarrow (381, 6)$
- **Residuals** : The attribute *fmri_glm.residuals* gives the residuals between thee timesries truth and predictions array for each session separately. The result is concatenated in a timestamp level

$$(127, 6) \times 3 \rightarrow (381, 6)$$

- We then call the implemented functions *_visualize_timeseries()* and *_visualize_residuals()* to visualize the timeseries and the residuals respectively for all 6 contrast maps. Please see Figures 3.22, 3.23

8. **Calculate Correlation Matrix** [2]:

- Correlation matrices are calculated using the timeseries from the previous step. Since we have 6 contrast maps, we will produce 6 different correlation matrices, one per each contrast map.
- We initialize an *ConnectivityMeasure()* object by providing as single parameter the **kind**. For the purpose of this diploma thesis we use *"correlation"*, however more options are available ('partial correlation', 'tangent','covariance','precision').
- Then the model is fit and transformed by providing as input the **Timeseries truth array**
- Each correlation matrix has a shape of

$$(top\_clusters, top\_clusters) = (6, 6)$$

as you can see in Figure 3.26

- After inspecting all 6 correlation matrices, we conclude that those with the higher intensity are concerning the contrast maps of ("PH" stack "W") and ("W" stack "mispron")

[2]The same logic will be applied to the Atlas Level Analysis

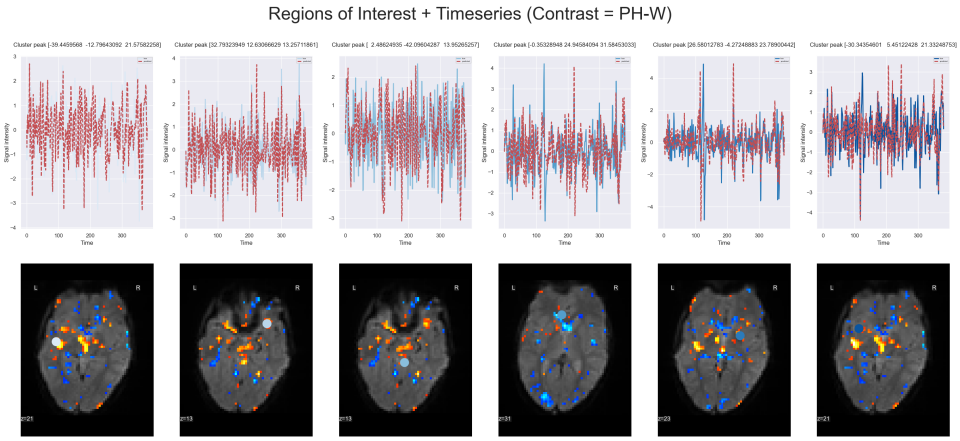


Figure 3.22. *Subject 047EPKL011011's timeseries and statistical map of ALL functional T2w images for the contrast map ("PH" - "W") for the top 6 clusters. Contrast map produced with "z$_s$core"*

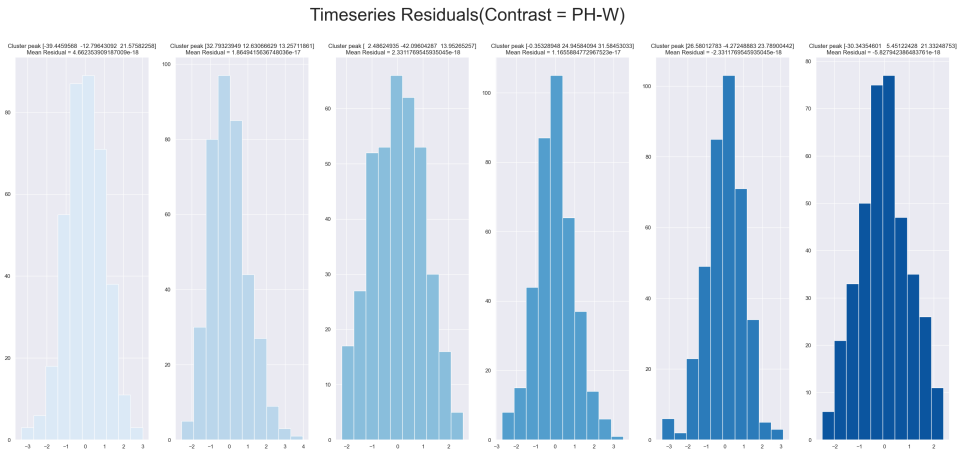


Figure 3.23. *Subject 047EPKL011011's residuals of ALL functional T2w images for the contrast map ("PH" - "W") for the top 6 clusters. Contrast map produced with "z$_s$core"*

3.6.2 Dictionary Learning Model

3.6.3 Atlas Model Analysis

Theoretical Background

Whole-brain maps can hide important details about the effects that we're studying. We may find a significant effect of incongruent-congruent, but the reason the effect is significant could be because incongruent **is greater** than congruent, or because congruent **is much more negative** than congruent, or some combination of the two. The only way to determine what is driving the effect is with **Regions of Interest** (ROI) analysis, and this is especially important when dealing with interactions and more sophisticated designs. [ref, a]

One way to create a region for our ROI analysis is to use an atlas, or a map that partitions the brain into anatomically distinct regions. There are plethora of different atlases that can be used to extract the regions of interest, which will be all listed below. Our code is structured in this way so that the atlas is dynamically selected (entered by the used), while the python library used is *nilearn.datasets*. However, for the purpose of this diploma thesis only **multi-subject dictionary learning** (MSDL) atlas will be used, which has 39 distinct brain regions (nodes)

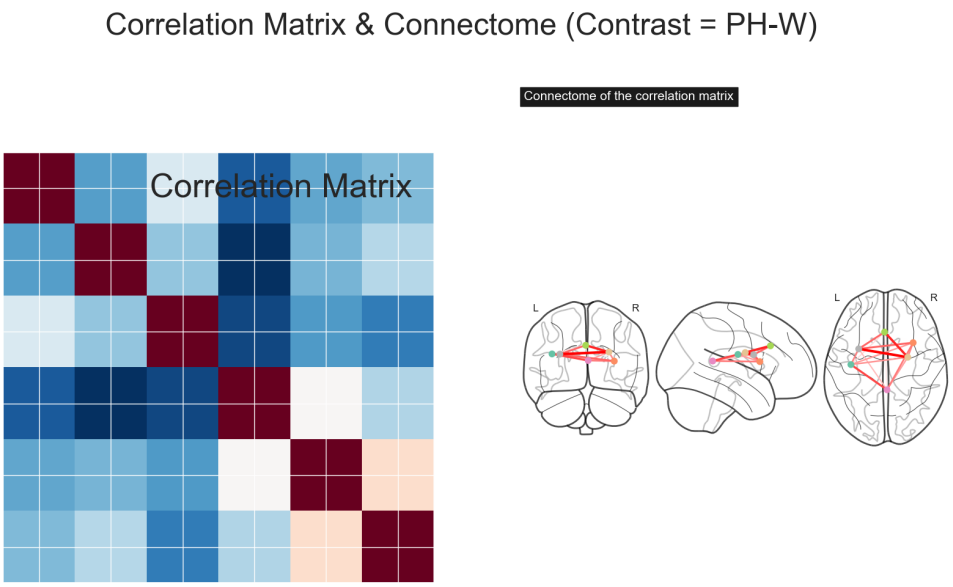
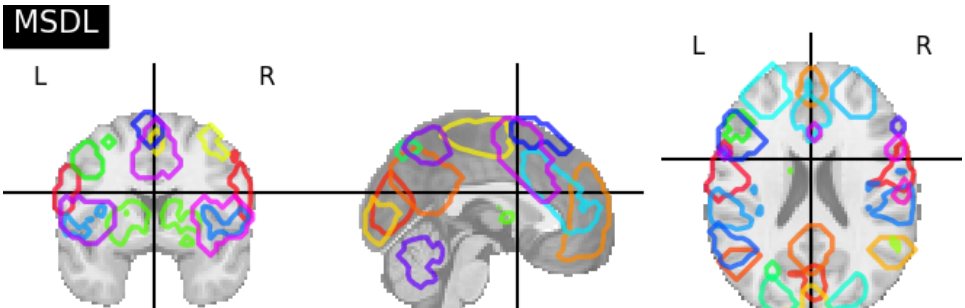


Figure 3.24. *Subject 047EPKL011011's correlation matrix of ALL functional T2w images for the contrast map ("PH" - "W") for the top 6 clusters. Contrast map produced with "z_score"*

Kind	Atlas Name	Method
3D	cortical	**datasets.***fetch_atlas_harvard_oxford('cort-maxprob-thr25-2mm')*
3D	subcortical	**datasets.***fetch_atlas_harvard_oxford('sub-maxprob-thr25-2mm')*
3D	yeo7	**datasets.***fetch_atlas_yeo_2011()*
3D	aal	**datasets.***fetch_atlas_aal()*
Probabalistic	aal	**datasets.***fetch_atlas_msdl()*
Coords	seitzman	**datasets.***fetch_coords_seitzman_2018()*
Coords	power	**datasets.***fetch_coords_power_2011()*
Coords	dosenbach	**datasets.***fetch_coords]_dosenbach_2010()*

Based on nilearn, MSDL atlas is a **probabilistic** atlas. These kind of atlases define soft parcellations of the brain in which the regions may overlap. Contrary to **deterministic** atlases, a voxel can belong to several components. These atlases are represented by 4D images where the 3D components, also called *'spatial maps'*, are stacked along one dimension (usually the 4th dimension). In each 3D component, the value at a given voxel indicates how strongly this voxel is related to this component.

Visualizing a probabilistic atlas requires visualizing the different maps that compose it. Here we visualize the **MSDL** atlas with **"contours"**, which means ROIs are shown as contours delineated by colored lines.



Striate (Striatum)	Striate (Striatum)	the striatum coordinates multiple aspects of cognition, including both motor and action planning, decision-making, motivation, reinforcement, and reward perception.
Default Mode Network (DMN)	Left Default Mode Network (L DMN)	The default mode network is active during passive rest and mind-wandering which usually involves thinking about others, thinking about one's self, remembering the past, and envisioning the future rather than the task being performed
	Right Default Mode Network (R DMN)	__"__
	Front Default Mode Network (Front DMN)	__"__
	Medial Default Mode Network (Med DMN)	__"__
Occipital Lobe (Occ post)	Occipital Lobe (Occ post)	The occipital lobe is the visual processing area of the brain. It is associated with visuospatial processing, distance and depth perception, color determination, object and face recognition, and memory formation
Motor Cortext (Motor)	Motor	The primary function of the motor cortex is to generate signals to direct the movement of the body.
Basal Ganglia (Basal)	Basal Ganglia (Basal)	The "basal ganglia" refers to a group of subcortical nuclei responsible primarily for motor control, as well as other roles such as motor learning, executive functions and behaviors, and emotions.
Right Visual Attention (R V Att)	Right Dorsolateral Prefrontal Cortex (R DLPFC)	It is considered as a brain area associated with domain general executive control functions such as task switching and task-set reconfiguration, prevention of interference, inhibition, planning, and working memory
	Right Frontal Pole (R Front pol)	__"__
	Right Parietal Lobe (R par)	The parietal lobe is vital for sensory perception and integration, including the management of taste, hearing, sight, touch, and smell. The right side, is believed to be important in helping us keep track of the space around us.
	Right Posterior Temporal Lobe (R Post Temp)	The non-dominant lobe, which is typically the right temporal lobe, is involved in learning and remembering non-verbal information (e.g. visuo-spatial material and music)
Left Visual Attention (L V Att)	Left Dorsolateral Prefrontal Cortex (L DLPFC)	

	Left Frontal Pole (L Front pol)	The frontal lobe is responsible for higher cognitive functions such as memory, emotions, impulse control, problem solving, social interaction, and motor function.
	Left Parietal Lobe (L par)	The left side is believed to be important in keeping track of the location of parts of the body which are moving
	Left Posterior Temporal Lobe (L Post Temp)	The dominant temporal lobe, which is the left side in most people, is involved in understanding language and learning and remembering verbal information
Attention (D Att)	Right Intraparietal Sulcus (R IPS)	The areas within the intraparietal sulcus (IPS), in particular, serve as interfaces between the perceptive and motor systems for controlling arm and eye movements in space.
	Left Intraparietal Sulcus (L IPS)	_”_
Visual Seconary (Vis Sec)	Left Lateral Occipital Complex (L LOC)	a key brain region in object and shape processing
	Visual Network (Vis)	
	Right Lateral Occipital Complex(R LOC)	a key brain region in object and shape processing
Salience	Dorsal Anterior Cingulate Cortex (D ACC)	It is a brain region that subserves cognition and motor control, but the mechanisms of these functions remain unknown
	Ventral Anterior Cingulate Cortex (V ACC)	The ventral Anterior Cingulate Cortex (vACC) has been robustly implicated in reward processing in social evaluation, but its precise role remains poorly understood.
	Right Anterior Insula (R A INS)	It is associated with the affective-perceptual form of empathy, while the left insula was associated with both the affective-perceptual and cognitive-evaluative forms of empathy
Temporal	Left Superior Temporal Sulcus (L STS)	Plays a role in phonological processing,
	Right Superior Temporal Sulcus (R STS)	A key role in perception of voice and the prosody of speech.
Language	Left Tempoparietal Junction (L TPJ)	The left temporoparietal junction (lTPJ) contains both Wernicke’s area and the angular gyrus, both prominent anatomical structures of the brain that are involved in language cognition, processing, and comprehension of both written and spoken language

	Right Tempoparietal Junction (R TPJ)	The right temporoparietal junction (rTPJ) is frequently associated with different capacities that to shift attention to unexpected stimuli (reorienting of attention) and to understand others' (false) mental state [theory of mind (ToM), typically represented by false belief tasks]
	Broca's Area (Broca) Superior Frontal Sulcus (Sup Front S)	Broca's area is recognized as one of the main language centers of the brain. This region is associated with the production of speech and written language, as well as being linked with the processing and comprehension of language The superior frontal gyrus (SFG) is thought to contribute to higher cognitive functions and particularly to working memory (WM), although the nature of its involvement remains a matter of debate.
	Right Pars Opercularis (R Pars Op)	The pars opercularis (BA44) is involved in language production and phonological processing due to its connections with motor areas of the mouth and tongue
Cerebellum (Cereb)	Cereb	helps coordinate and regulate a wide range of functions and processes in both your brain and body. While it's very small compared to your brain overall, it holds more than half of the neurons (cells that make up your nervous system) in your whole body
Dorsal Posterior Cingulate Cortex (Dors PCC)	Dorsal Posterior Cingulate Cortex (Dors PCC)	It is involved in the dorsal attention network (a top-down control of visual attention and eye movement)
Cingulate Cortex Insula (Cing-Ins)	Cingulate Cortex (Cing)	A plethora of research has implicated the cingulate cortex in the processing of social information (i.e., processing elicited by, about, and directed toward others) and reward-related information that guides decision-making.
	Left Insula (L ins)	
	Rigt Insula (R ins)	
Anterior Intraparietal Sulsus (Ant IPS)	Left Anterior Intraparietal Sulsus (L Ant IPS)	Its ts principal functions are related to perceptual-motor coordination (e.g., directing eye movements and reaching) and visual attention, which allows for visually-guided pointing, grasping, and object manipulation that can produce a desired effect
	Right Anterior Intraparietal Sulsus (R Ant IPS)	

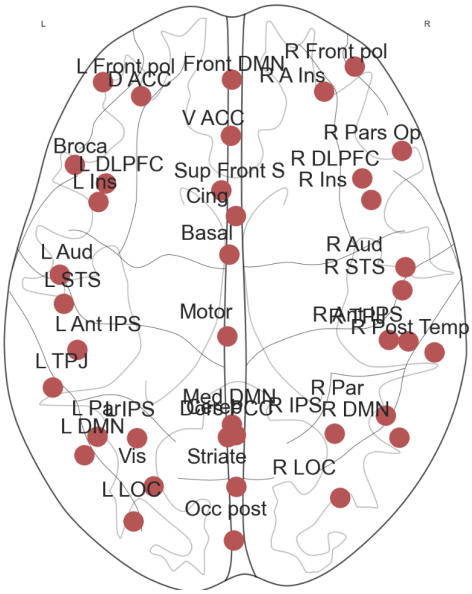


Figure 3.26. *MSDL Atlas Regions. Genarated with the function plot_connectome()*

Performing Model Atlas Analysis

The whole framework with all the steps combined are presented in the Figure 3.27. In our case we are only interested in the middle subbox of operations, since **MSDL** maps is a probabilistic maps.

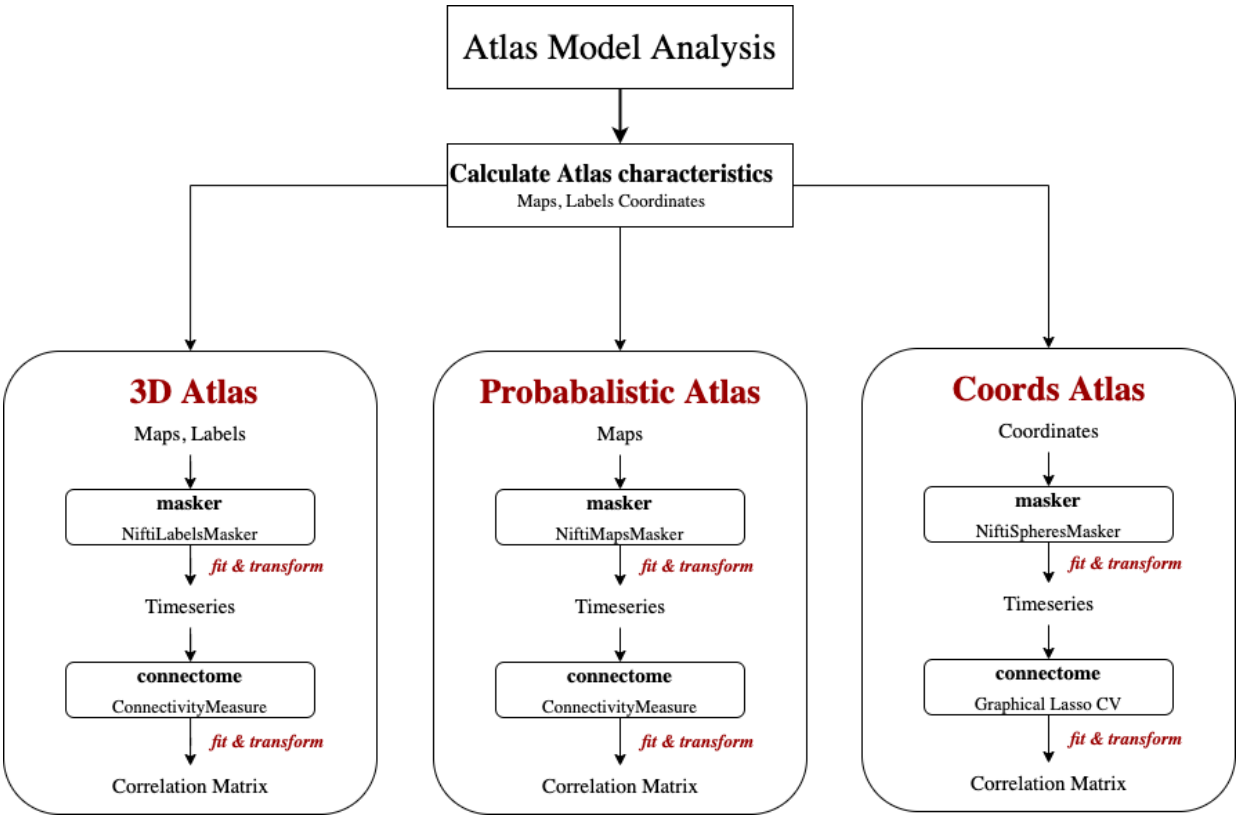


Figure 3.27. *Atlas Analysis Framework*

1. **MSDL Atlas Loading** We used the function **datasets**.*fetch_atlas_msdl()* to retrieve the atlas. This returns a Dictionary-like object, the interest attributes (characteristics) in which we are interested are

:

- *maps* : str, path to NiFti file containing the Probabilistic atlas image. We have 39 regions of interest. The actual *maps* shape is :

$$(40, 48, 35, 39)$$

- *labels* : list of str, list containing the labels of the regions. There are 39 labels such that *data.labels*[i] corresponds to the $i-th$ map
- *region_coords*: list of length-3 tuple, *data.region_coords*[i] contains the coordinates (x, y, z) of $i-th$ region in MNI space.

2. **Define the Masker** : We define a *NiftiMapsMasker()* which accepts as arguments

Parameter	Value	Explanation
maps_img	atlas_maps	The Nifti1Image object of the atlas maps
t_r	3	We have 3 different conditions
detrend	True	This parameter is passed to signal.clean
standardize	True	the signal is z-scored. Timeseries are shifted to zero mean and scaled to unit variance
high_pass	0.0135	The explanation is given before

3. We use the method *fit_transform()* of the previously defined masker to fit the model in the **functional images** and the respective **confounds**, in order to finally calculate the **timeseries** (performing **signal extraction**). However there are 2 approaches here than can be followed. We proceeded with the first one, however for seek completeness we mention both of them

- **First Approach** : Concat all BOLD runs (sessions) into a single functional image with shape

$$(64, 64, 64, 130 + 130 + 123 \text{ timestamps}) = (64, 64, 64, \text{ timestamps})$$

and then calculate the confounds. Then, we fit and transform the masker with this image and these confounds. The output is the timeseries signal which has a shape of

$$(381 \text{ timestamps }, 39 \text{ regions })$$

- **Second Approach** : We fit and transform the masker 3 times separately, one per each BOLD run (session), and we extract 3 different timeseries signal which has a shape of

$$(130 \text{ timestamps }, 39 \text{ regions })$$

$$(130 \text{ timestamps }, 39 \text{ regions })$$

$$(123 \text{ timestamps }, 39 \text{ regions })$$

Then we concat the 3 different timeseries numpy array and get the final output timeseries signal which has a shape [3] of

(381 timestamps , 39 regions)

4. We define a *ConnectivityMeasure()* which accepts as a parameter the matrix kind connectivity that should be established. We choose a **"correlation"** kind of connectivity. Then we use the method *fit_transfsorm()* to fit the model in the **timeseries** calculated in the previous step. After squeezing the result we get a numpy array, the final **correlation matrix** which has a shape of

   (39 regions , 39 regions)

   Please see the Figure 3.28. As you may notice the correlations are **very intense**, but they are still noticeable. The reason is that we have removed the **confounds** that add **noise** to the **timeseries** signal

5. Finally we save the **correlation matrix** as a *".nii.gz"* format under the folder of the respective subjects

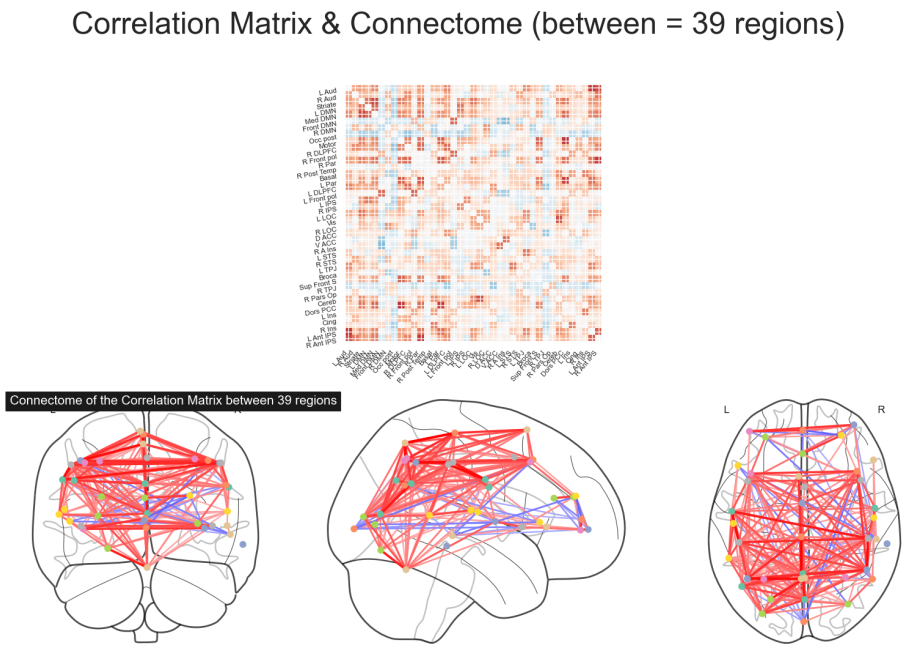


Figure 3.28. *Subject 047EPKL011005's correlation matrix derived from the MSDL Maps after applying all preprocessing*

All these steps are performed for each subject separately. So we end up with 58 different correlation matrices, which will be therefore used as input in our **machine learning** and **deep learning** frameworks

[3] same as in the first approach

Chapter 4

Theoretical Background

4.1 Machine Learning

Machine Learning (ML) is the technology of developing computer algorithms that are able to emulate human intelligence. These algorithms are built as to be able to improve automatically through experience and by the use of data, known as training data. Machine learning algorithms can be classified into separate classes according to the nature of the data, the learning process, and the model type [Naqa and Murphy, 2015]. ML is seen as a part of Artificial Intelligence and it is used to make predictions or decisions based on the learning experience the model gained through training. To this day ML technology has been applied to such diverse fields as pattern recognition [Bishop and Nasrabadi, 2006] , computer vision [B. Apolloni and Patnaik, 2005], spacecraft engineering [S.-I. Ao and Amouzegar, 2010], finance [L. Gy¨orfi and Walk, 2012], entertainment [Gong and Xu, 2007], [Yu and Tao, 2013], ecology [Fielding, 1999], computational biology [S. Mitra and Michailidis, 2008], [Yang, 2010], and biomedical and medical applications [T. J. Cleophas and Cleophas-Allers, 2013], [J. D. Malley and Pajevic, 2011].

4.2 Machine Learning Methods

There are **three** methods of machine learning: *supervised learning, unsupervised learning, and reinforcement learning*. Although in this diploma thesis, *supervised learning* is used for the multi-classification, a brief introduction of both methods follows for better understanding.

4.2.1 Supervised learning

The defining characteristic of supervised learning is the availability of annotated data. To be more precise, supervised learning entails learning a mapping between a set of input variables and an output variable, called label and after that this mapping is applied to predict the values for the unseen data [P. Cunningham and Delany, 2008]. Having the data labeled, hence knowing the correct output for each input the model will be trained over time, measuring the accuracy through a loss function adjusting until the error has been sufficiently minimized.

There are **two** types of supervised learning techniques in machine learning: *regression* and *classification*

- Regression : is used for prediction of a continuous variable based on the relationship between the input variables and output variable learned during the training. For example, regression can be useful for house price prediction, with the price of the house as output and inputs could be variables such as locality, size of house etc.

- Classification : is used when the output variable is categorical. Thus, it is useful for grouping the output inside a class. If the algorithm tries to label input into two distinct classes, it is called binary classification. Selecting between more than two classes is referred to as multi-class classification, which is the case in this diploma thesis

4.2.2 Unsupervised learning

On the contrary of the aforementioned method, there are cases in which labeled data are not possible to be obtained or very strenuous to create. To solve these type of cases, unsupervised learning techniques are used to find hidden patterns from the given dataset. This type of learning can be compared with the procedure taking place in the human brain while learning new things. As there is no corresponding output data for the input data unsupervised learning cannot be directly applied to regression or classification problems. The goal of unsupervised learning is to find the underlying structure of dataset, group that data according to similarities, and represent that dataset in a compressed format. The unsupervised algorithm can be further categorized into clustering and association problems [T. Hastie and Friedman, 2009].

- Clustering is a method that attempts to group the objects based on the similarity between them, such that objects with most similarities remain into a group and have less or no similarities with objects in another group.

- Association is used to detect the relationships between variables in a large database. It is commonly used for marketing strategies, such as people who buy X item are more likely to buy item Y

4.2.3 Reinforcement learning

Reinforcement learning is a sub-field of machine learning that deals with the problem of training an agent to maximize a reward signal while acting in an environment. This method is based on rewarding desired behaviors and/or punishing undesired ones, hence a reinforcement learning agent is able to learn through trial and error. The main goal of reinforcement learning is to define the best sequence of decisions the agent has to follow to solve a problem while maximizing a long-term reward. This is why it is primarily applied for motion planning, dynamic pathing, controller optimization, scenario-based learning policies for highways etc. A characteristic example of the adequacy of the method is its use for parking that can be achieved by learning automatic parking policies.

4.3 Machine Learning Approacch

Machine learning is the general term to describe the procedure followed by a computer to learn from data. A machine learning algorithm is a computational process that uses the input data to perform a task without being explicitly programmed to do so, instead they recognize patterns in data when it arrives and make predictions. As mentioned earlier machine learning algorithms can be divided into *supervised, unsupervised and reinforcement learning*. The broad categories of classification and regression algorithms were mentioned before as well. Although in this diploma thesis we are dealing with a classification problem, some regression algorithms are also worth mentioning.

Before deep diving into the mathematical models of the different machine learning, it is of utmost importance to mention that since the data samples use are small (only 58 subjects), we preferred to construct an exhaustive pipeline of work which will be thouroughfully analyzed in the next chapter. In this direction, 16 sklearn classifiers were used, as well as a plethora of transformers.

4.3.1 Classifiers

All the Classifiers used

Sklearn Class	Classifier
ensemble	*AdaBoostClassifier*
ensemble	*RandomForestClassifier*
ensemble	*GradientBoostingClassifier*
discriminant_analysis	*LinearDiscriminantAnalysis*
svm	*SVC*
svm	*NuSVC*
neighbors	*KNeighborsClassifier*
linear_model	*LogisticRegression*
linear_model	*SGDClassifier*
linear_model	*RidgeClassifier*
naive_bayes	*GaussianNB*
tree	*DecisionTreeClassifier*
tree	*ExtraTreeClassifier*
neural_network	*MLPClassifier*
calibration	*CalibratedClassifierCV*
xgboost	*XGBClassifier*

Logistic Regression

Despite its name the logistic regression [D. G. Kleinbaum and Klein, 2002] algorithm is used in classification problems and not in regression problems as the linear regression. The logistic function, a core part of the LR, is a sigmoid function that can take a real number and transforms it into a value between 0 and 1, producing a "S"-shaped curve. LR uses the maximum likelihood estimation method to estimate the model coefficients. The goal is to model the probability of a random variable Y being 0 or 1 (for binary classification) given experimental data. It is typically used for binary classification problems, but a multiclass classification is also possible. The logistic function is defined as Equation 1 and it can be considered as a generalized linear model function parameterized by ∂. The logistic function is a sigmoid function and it is illustrated in the Figure 4.1

$$h_{\partial}(X) = \frac{1}{1 + e^{-\partial^T X}} = Pr(Y = 1|X; \partial) \tag{1}$$

The equation used by logistic regression for representation is very similar to the linear regression's. Therefore, the input values (x) are combined linearly using weights or coefficient values to predict an output variable. The difference with the linear regression model is that the output value is modeled as a binary output (0 or 1) rather than a numeric value. Thus, the logistic regression equation is defined as follows

$$y = \frac{e^{\beta_0 + \beta_1 x}}{1 + e^{\beta_0 + \beta_1 x}} \tag{2}$$

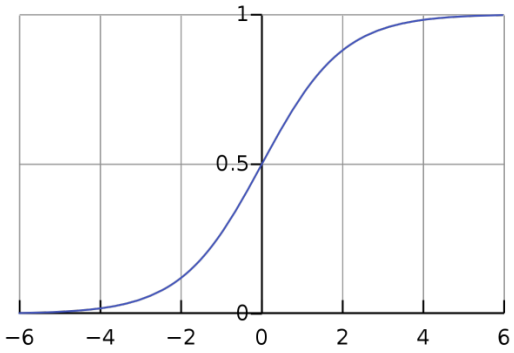


Figure 4.1. *Logistic Function (sigmoid)*

,where y is the predicted output, β_0 is the bias or intercept term and b1 is the coefficient for the single input value (x). In other words, logistic regression is a linear model but the output is transorformed using the logistic function.

Maximum-likelihood estimation (MLE) is an algorithm used by the logistic regression algorithm in order for the coefficients (beta values) of the algorithm to be estimated from the training data. The MLE algorithm searches for coefficient values that minimize the error in the probabilities predicted by the model those in the data. This is mainly implemented using efficient numerical optimization algorithms. Such algorithms can be selected as a parameter in the logistic regression classifier using the **sklearn** library, therefore is an extra hyper-parameter tuning in the classification procedure. There are several optimizers that can be used in the classifier but not all of them are suitable for a multiclass problem like the current one. As a result, only the ones used for the experiments are referred.

Before we dive in to the optimizer functions there are some terms that need to be clarified first.

- The Hessian of a function f(x,y) is a matrix of second order partial derivatives formed from all pairs of variables in the domain of f. Suppose we have a function f of n variables $f : R^n$ßR. The Hessian of the function f is given in the following figure Figure 4.2

$$H(f) = \begin{pmatrix} \frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_1 \partial x_n} \\ \frac{\partial^2 f}{\partial x_2 \partial x_1} & \frac{\partial^2 f}{\partial x_2^2} & \cdots & \frac{\partial^2 f}{\partial x_2 \partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_n \partial x_1} & \frac{\partial^2 f}{\partial x_n \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_n^2} \end{pmatrix}$$

Figure 4.2. *Hessian Matrix H_f*

- A quadratic function is one of the form $f(x) = ax^2 + bx + c$, where $a, b and c$ are numbers with a not equal to zero.

- Linear approximation : Given a function, f(x), we can find its tangent at x = a. The equation of the tangent line L(x) is: $L(x) = f(a) + f(a)(xa)$ A shown in the Figure 4.3 near $x = a$ the tangent line and the function have nearly the same graph. On occasion, we will use the tangent line, $L(x)$, as an

approximation to the function, $f(x)$, near x = a. In these cases, we call the tangent line the *Linear Approximation* to the function at $x = a$.

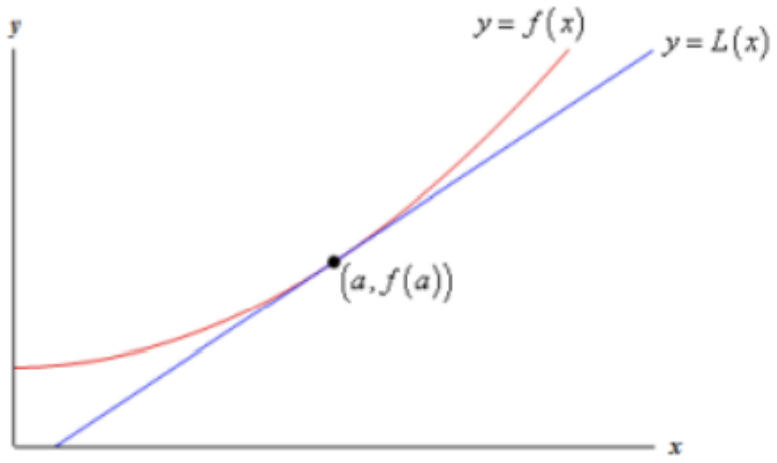


Figure 4.3. *A function and its tangent*

- Quadratic approximation : Same like linear approximation, yet this time we are dealing with a curve where we cannot find the point near to 0 by using only the tangent line. The reason we need a point near zero because the slope of the tangent line in the minimum cost point (global optima) is zero. In this case we use the parabola, as shown in Figure 2.5. In order to fit a good parabola, both parabola and quadratic function should have same value, same first derivative, and the same second derivative.
- Penalization is used to avoid overfitting by penalizing the algorithm for fitting a model that fits the training data tightly. L1 regularization penalizes the sum of absolute values of the weights, whereas L2 regularization penalizes the sum of squares of the weights.

Optimizer Functions

- Newton's method : Known as ("newton-cg" in sklearn) is using the quadratic approximation in order to mimic the loss function, thus the quadratic function that is to be minimized in order to minimize the error. It is like a twisted Gradient Descent with the Hessian. The disadvantages of the Newton's method are that it is computationally expensive due to the computation of the Hessian and that it attracts saddle points, which are stable points where the function has a local maximum in one direction, but a local minimum in another direction. Although, Newton's method is expensive at each iteration, it has very fast convergence rates.
- Limited-memory Broyden–Fletcher–Goldfarb–Shanno Algorithm : Limited-memory Broyden–Fletcher–Goldfarb–S Algorithm ("lbfgs" in sklearn) is analogue to the Newton's method, with the difference that the Hessian matrix is approximated using an estimation to the inverse Hessian matrix. With the term Limited-memory it is indicated that it stores only a few vectors that represent the approximation implicitly. The most important disadvantage of this solver is that it may not coverage to anything
- Stochastic Average Gradient : Stochastic average gradient is a method for optimizing the sum of a finite number of smooth convex functions. The SAG method's iteration cost is independent of the number of terms in the sum [M. Schmidt and Bach, 2017]. It is faster than other solvers for large datasets, when both the number of samples and the number of features are large. A drawback is that it only supports L2 penalization

- SAGA : The SAGA solver is a variant of SAG that also supports the non-smooth penalty L1 option (i.e. L1 Regularization). This is therefore the solver of choice for sparse multinomial logistic regression and it's also suitable for very large dataset. So it will be not selected by our gridsearch optimization process (it will be analyzed in the the **results** chapter)

Support Vector Machine (SVM)

Support vector machine (SVM) [21] is a highly used algorithm that is used for both regression and classification problems. The objective of the support vector machine algorithm is to find a hyperplane in a N-dimensional space, where N is the number of features that can classify the data points. It uses a simple mathematical model $y = w \cdot x' + \gamma$, and manipulates it to allow linear domain division[22]. SVM can be divided into linear and non-linear models [23]. Linear support vector machine can divide the data with a linear line or hyperplane to separate the classes in the original domain. On the other side, non-linear support vector machine indicates that the data domain cannot be divided linearly and can be transformed to a space called the feature space where the data can be divided linearly

As shown in the Figure 4.4 there are many possible hyperplanes that can be chosen, in order to separate the two classes of data points. The purpose is to find the plane that has the maximum margin, thus the maximum, distance between data points from both classes, as illustrated in the Figure 4.5. The data points with the minimum distance to the hyperplane are called Support Vectors and influence the position and the orientation of the hyperplane. If we delete the support vectors the position of the hyperplane will change. Using these support vectors we maximize the margin of the classifier, as depicted in Figure 4.6

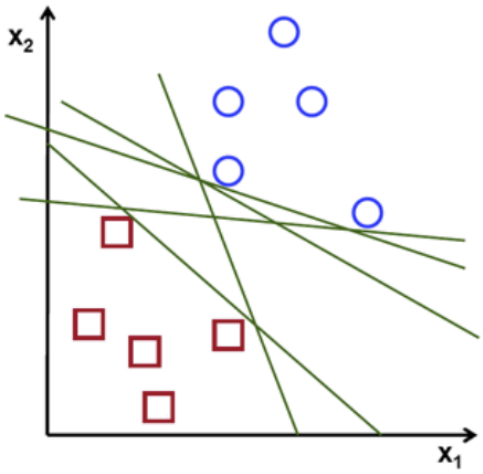


Figure 4.4. *Support vector machine possible hyperplanes*

The simplest type of SVM as explained before is used for binary classification problems dividing the data point into two categories 0 or 1. In order to perform multiclass classification, which is the scope of this diploma thesis, we use the same principle as in binary classification. To be more precise, the multiclass problem is broken down into multiple binary classification problems, using a heuristic function. There are two types of **heuristic functions**:

- One-vs-Rest : This method simple splits the multiclass data into binary classification data so the binary classification algorithms can be applied to the binary classification data. In this technique the N-class instances are divided into N binary classifier models.

- One-vs-One : This method is similar to the One-vs-Rest method as it is based on splitting the multiclass

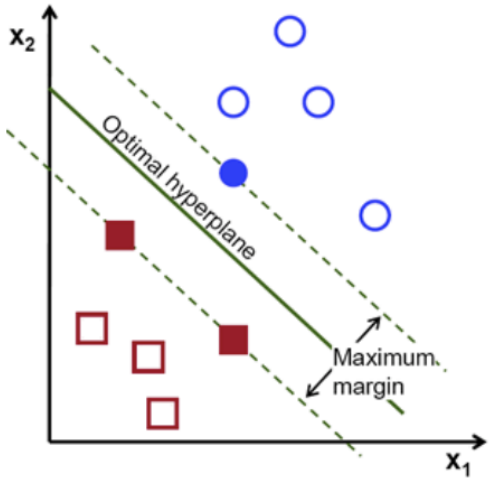


Figure 4.5. *Support vector machine optimal plane*

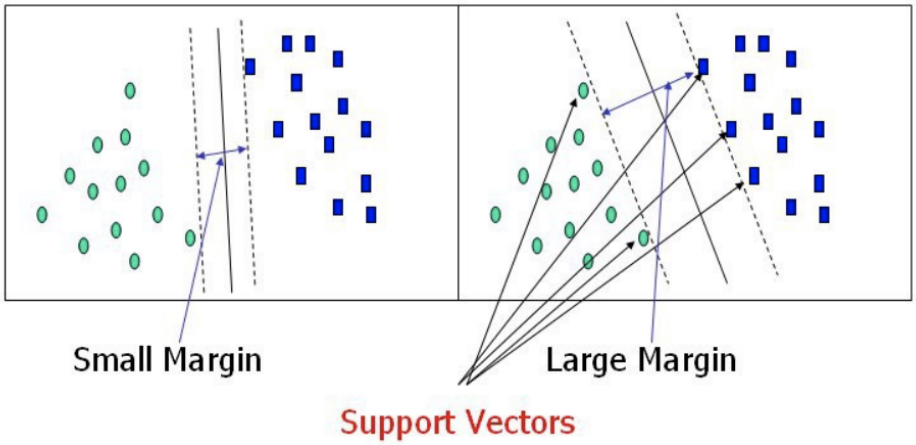


Figure 4.6. *: Support vectors*

data into binary classification data, but the splitting behaviour is different. In this technique the N-class instances are divided into $\frac{N(N-1)}{2}$ binary classifier models.

SVM is also called kernelized SVM as it uses the kernel function method to take data as input and transform it so that a non-linear decision surface is able to transform to a linear equation in a higher number of dimension spaces. The transformation is possible if we use the score calculated by calculating the distance score of two datapoints. This score is higher for closer datapoints and vice versa. We will discuss the most popular kernel functions that are used in the experiments and are available in **scikit-learn**

Kernels

- Linear Function : Linear function is used when data is linearly separable,that is, it can be separated using as single line. It is referred for the sake of completeness as it cannot be used for multiclass classification problems. The kernel function is defined as:

$$k(x, y) = xy$$

- Polynomial Function : Polynomial function represents the similarity of the training samples in a feature space over polynomials of the original variables used in the kernel, allowing learning of non-linear models. For degree-d polynomials, the polynomial kernel is defined as:

$$k(x, y) = (x^T y * c)^d$$

where x and y are vector in the input space, and $c \geq 0$ is a free parameter trading off the influence of higher-order versus lower-order terms in the polynomial.

- Gaussian Kernel Radial Basis Function (RBF) : The value of the RBF kernel decreases with distance and ranges between zero and one it can be portrayed as a similarity measure, thus real-valued function that quantifies the similarity between two objects. The RBF kernel on two samples x and y, represented as feature vectors in some input space, is defined as:

$$k(x, y) = e^{-\gamma \|x-y\|^2}$$

- Sigmoid Function Is similar to the sigmoid function in logistic regression. The sigmoid kernel comes from the neural network field, where the bipolar sigmoid function is often used as an activation function for artificial neurons. An SVM model using a sigmoid kernel function is equivalent to a twolayer, perceptron neural network. The definition of the sigmoid kernel function is:

$$k(x, y) = tanh(\gamma x^T y + r)$$

k-Nearest Neighbor

The k-nearest neighbor is a non-parametric supervised algorithms used in either regression or classification problems, which uses proximity to make predictions about the classification of a data point. It is most used for classification problems, working with the assumption that similar point can be found near one another. We will discuss k-NN classification and not regression as it serves the scope of the current problem. The input consists of the k closest training examples in a data set and the output is a class membership. The procedure in which a point is classified is called majority vote of its neighbors. Each object is assigned in the class that is more common among its k nearest neighbors, where k is a positive integer, typically a small integer. Therefore, if $k = 1$, then the object is simply assigned to the class of that single nearest neighbor

In order to regulate which data points are closest to a given data point, the distance between these data point must be calculated. There are several distance metrics that we can choose from with the Euclidean distance being the most common for continuous variables and Hamming distance for discrete variables.

Parameter no.1 (Distance)

- Euclidean Distance : Euclidean distance is limited in real-valued vectors and it measures a straight line between the query point (data point that needs to be labeled) and the other points being measured. The euclidean distance is calculated using the following formula:

$$d(x, y) = \sqrt{\sum_{i=1}^{n} (y_i - x_i)^2} \tag{3}$$

- Hamming Distance This technique is most commonly used with Boolean or string vectors and it measures the minimum number of errors that could transformed one string into the other. The above can be represented by the following formula:

$$D_H = \sum_{i=1}^{k} |x_i - y_i| \tag{4}$$

Parameter no.2 (k)

Another crucial parameter for the k-NN algorithm that needs to be tuned is the k value. The k value in the k-NN algorithm defines how many neighbors will be checked before the query point is classified. The choice of k can define whether the algorithm will overfit or underfit. Lower values of k can have high variance, but low bias, whilst larger values of k may have high bias and lower variance. Let's dive in a little bit more in theory.

Decision Tree

The DT method uses a non-parametric supervised learning approach to solve both regression and classification problems. DT uses a tree representation where each test on an attribute is represented by an internal node, and each leaf denotes a class label. Thus, DT can learn certain decision rules inferred from the features used to build a model that predicts the value of the target variable.

Random Forest

RF is an ensemble learning method, consisting of several DTs, that can be used for classification and regression. Ensemble methods are algorithms that incorporate more than one type of algorithm. RF works by constructing a number of DT classifiers which learn and make predictions independently, and outputs a combined single prediction that is the same or better than the output made by the previously constructed DT classifiers.

Adaptive Boosting

AdaBoost is an ensemble boosting algorithm that combines a set of "weak" classifiers into a weighted sum to create a stronger more accurate "boosted" classifier. AdaBoost starts by fitting a classifier on the dataset, and then fits the same version of that classifier on the same dataset where the weights of the misclassified instances are modified so that the next classifier is improved to work on the more challenging instances.

4.3.2 Definition of feature array X and label vector y

The correlation matrices are **symmetrical** so we only keep one triangular sub-matrix, which is also flattened (1D) giving a feature vector of size

$$n_{features} = \frac{N_{regions} \cdot (N_{regions} - 1)}{2} = \frac{39 \cdot (39 - 1)}{2} = 741$$

Every feature represent the connection between 2 regions of the brain (as defined in the MSDL atlas). We have 58 participants so $n_{samples} = 58$. The final feature array X and label vector y have a shape of $(n_{samples}, n_{features}) = (58, 741)$ and $(n_{samples},) = (58,)$ respectively.

4.3.3 Definition of Training, Validation and Test Set

The current dataset has 58 subjects/samples and and we split them into **training set** and a **test set** with an analogy of *80/20*. So we end up with 47 training samples and 11 testing samples. For all the

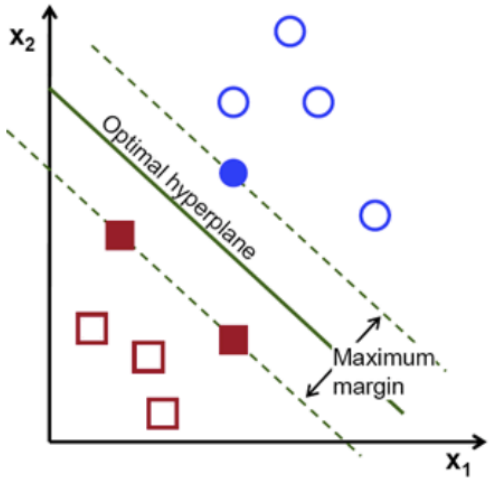


Figure 4.7. *Flattening Correlation Matrices*

underlying analysis, the **test set** is held back and is only used as **real data** for a final evaluation of our models.

Moreover, during the GridSearch optimization, the sklearn further splits **training set** into a new **training set** and a **validation set**, which is used to give an estimate of model skill while tuning pipeline's parameters. The final size of the **training** and **validation** set is dependent on the number of splits we chose. During this diploma thesis we chose $cv = 10$. This mean, that we have a *10-fold Cross Validation* scheme so :

- A model is trained using $k-1 = 9$ of the folds as training data. So the training samples are $\frac{9}{10} * 47 \sim 43$
- The resulting model is validated on the remaining part of the data (i.e., it is used as a validation set to compute a performance measure such as f1). So the total validation samples are $\frac{1}{10} * 47 \sim 4$

So, below you can find the shapes of all data sets used in each optimization framework

Optimization Framework	Explanation	Train set	Validation set	Test set
Transformers Optimization	Best Transformers Combination (per Classifier)	$(47, 741)$		$(11, 741)$
Gridsearch Optimization	Best Parameters (per pipeline)	$(43, 741)$	$(4, 741)$	$(11, 741)$
Ensemble Optimization	Best Voting Ensemble Classifier	$(47, 741)$		$(11, 741)$

4.3.4 Metrics

This diploma thesis deals with the problem of classification, that is, the training of models with the aim of dividing the samples into the 3 classes of **Dyslexia**, **Spelling Disorder** and **Typical Reader**. After training them one should be able to assess the models ability to perform this task successfully. For this purpose, a set of widely used metrics has been defined which express the accuracy of a model's predictions in relation to the actual values of the evaluation data.

For the binary and multiple classification of the data the pairwise (of 2 classes) relationship between prediction and ground truth is defined as following :

- **True Positive (TP)** : a prediction that was positive and the actual value of the data was positive
- **False Positive (FP)** : a prediction that was positive and the actual value of the data was negative
- **True Negative (TN)** : a prediction that was negative and the actual value of the data was negative
- **False Negative (FN)** : a prediction that was negative and the actual value of the data was positive.

So based on the previous abbreviations we can define our metrics

- **Accuracy** : Expresses the success rate of the model, i.e. the number of samples that the model correctly classified against the total of all samples

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

- **Precision** : It expresses the percentage of successful predictions of a class to the total predictions of the class.

$$Precision = \frac{TP}{TP + FP}$$

- **Recall** : It expresses the percentage of successful predictions of a class to the actual one number of observations belonging to it.

$$Recall = \frac{TP}{TP + FN}$$

- **F1 score** : It is the harmonious medium of Precision and Recall.

$$F1 = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall}$$

In this diploma thesis the instances are not equally distributed in the **4** classes, so we will use F1 score. F1 Score is used for imbalanced datasets because it balances precision and recall. Precision measures the proportion of true positive predictions, while recall measures the proportion of actual positive cases that are correctly identified. In imbalanced datasets, there may be a skewed class distribution, where one class is more prevalent than the other. In such cases, using accuracy as a metric may not give an accurate representation of the model's performance, as it would be biased towards the majority class. F1 Score provides a harmonic mean of precision and recall, taking into account both false negatives and false positives, making it a better metric for evaluating the performance of a model on imbalanced datasets.

4.3.5 Optimizations

Experimental Set Up

The Machine learning exploratory pipeline was implemented in IPython using Sklearn library. We followed an exhaustive optimization framework in 3 different levels (See Figure 4.8)

1. **Transformers Optimization** : Selecting the best Transformers per Classifier
2. **Hyper-parameter Tuning** : Optimizing all 16 Pipelines' parameters

3. **Ensemble Optimization** : Selecting the Best Ensemble Hard Voting Classifier

On top of that, a 4th optimization layer was considered to be implemented. Having selected the best Ensemble Hard Voting classifier, where each pipeline has **equal weight**, in soft voting, the classifiers are **weighted** based on their individual accuracy or confidence levels. Therefore, we consider to keep the pipelines selected by the 3rd optimization layer and optimize their **weights** by experimenting on different sets of discrete weights in the process of determining the contribution of each individual pipeline in the ensemble to the final prediction. However this is out of the scope of this diploma thesis and was left as future work.

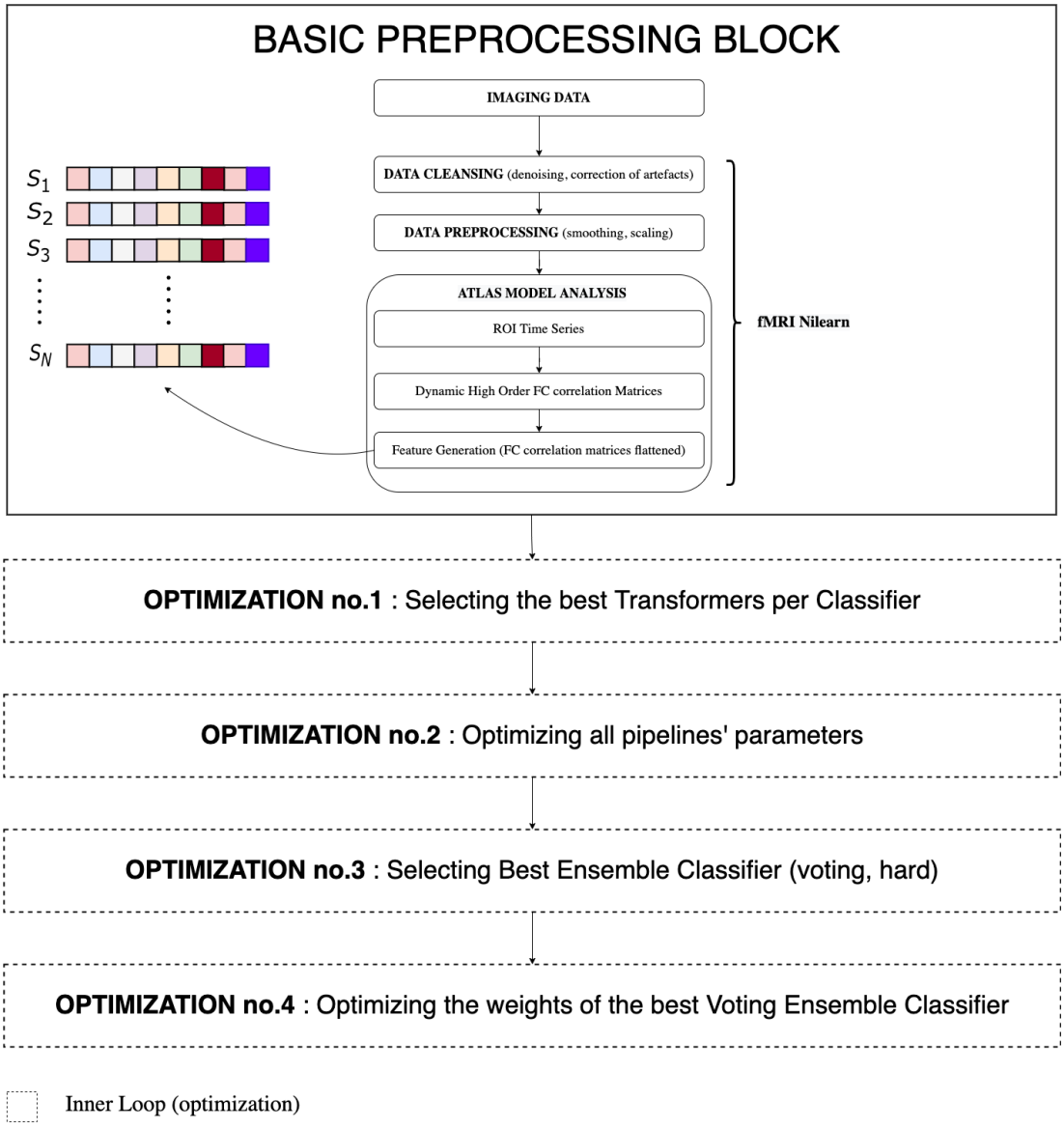


Figure 4.8. *Flow Diagram illustrating the machine learning workflow*

Optimization no.1 : Selecting the best Transformers per Classifier

This is a multi-class classification problem, and as you know all the features are the result of flattening the correlation matrices. So, all features used are **numerical features**. In classical machine learning projects we come across a variety of transforming techniques for both **numerical** and **categorical features**

Transformer	Numerical Features	Categorical Features
Imputer	*Yes*	*Yes*
Scaler	*Yes*	*No*
Encoder	*No*	*Yes*
Undersampler	*Yes*	*Yes*
Oversampler	*Yes*	*Yes*
Feature Selector	*Yes*	*Yes*

However for the purpose of this diploma thesis, and since we deal with **numerical features** and no missing values, only the following kind of transformers are used and optimized :

Scaler	Undersampler	Oversampler	Feature Selector	Feature Extractor
StandardScaler	*RandomUnder-Sampler*	*SMOTE*	*SelectKBest*	*PCA*
MinMaxScaler	*NearMiss*		*VarianceThreshold*	*LDA*
MaxAbsScaler	*CondensedNear-estNeighbour*		*SelectPercentile*	
PowerTrans-former	*TomekLinks*		*SelectFrom-Model(LogisticRegression)*	
QuantileTrans-former	*EditedNearest-Neighbours*		*GenericUnivariateSelect*	
			SelectFpr	
			SelectFdr	
			SelectFwe	

Since we have 16 classifiers, we aim to find the best combination of transformers, that is, finding the 16 best pipelines that leads to the highest f1 test score. In summary :

- **Scaling** : Changing the value range without changing the distribution pattern. The range is often set from 0 to 1. Machine learning algorithms work best when the features are on a similar scale and approach the Normal Distribution.

- **Feature Selection** : Feature selection is the process of isolating the most consistent, non-redundant and relevant features for building machine learning models. The new size of the set X will be (fewer features) :

$$(n_{samples}, n'_{features}) = (n_{samples}, n_{features} \downarrow)$$

- **Feature Extraction** : The most common linear methods for feature extraction are Principal Component Analysis (PCA) and Linear Discriminant Analysis (LDA). "Feature extraction fills this requirement: it builds valuable information from raw data — the features — by reformatting, combining, transforming primary features into new ones... until it yields a new set of data that can be consumed by the Machine Learning models to achieve their goals" [**?**]

- **Undersampling** : We will not apply any **oversampling** method at all because it will possibly increase the probability of overfitting, as it creates exact copies of the minority samples, even though our dataset is fairly balanced between the 3 classes SpD ~ 21, DL ~ 23, TL ~ 27. **Oversampling** methods copy or create new composite examples in the minority class, while **undersampling** methods delete or merge examples in the majority class. So since **undersampling** does not deal with synthetic data at all, we proceed to the examination of difference undersampling aglrothms. The new size of the set X will be (fewer samples):

$$(n'_{samples}, n_{features}) = (n_{samples} \downarrow, n_{features})$$

Optimization no.2 : Optimizing all 16 pipelines' parameters

HalvingGridsearchCV

This is a **hyperparameter tuning** step. For each classifier, we already have the best combination of transformers (4 transformers per classifier). However, hyperparameters are not specifically learned during the training process but can be adjusted to optimize how a model performs. Transformers in general also have hyper-parameters that affect their operation, eg **VarianceThreshold** had the lower variance threshold, **PCA** the number of principal components. The selection of hyper-parameters (such as k of kNN) is only done empirically through **cross-validation**. Transformers and their hyperparameters therefore affect the format of the data. In this optimization step, we search the best **parameter values kernel** for each pipeline (*pipeline = transformers + classifier*).

The performance of all possible combinations of hyper-parameters in each pipeline is examined through the method of grid search (Grid Search). Grid search is simply the exhaustive search of all combinations of a given set of values for each hyper-parameter of the pipeline. These values are set manually based on trial-error and empirical knowledge. The method must be driven by a performance metric that is evaluated on the test set using cross validation. In our case we use **F1 score**. The implementation of the method was done through the GridSearchCV function of the Sci-kit Learn library.

For this optimization step we used **HalvingGridsearchCV** which is **x5-10 times faster**. Moreover, this project is dedicated to provide a well-rounded exhaustive optimization scheme, so resources distribution can not be neglected. This HalvingGridsearchCV is a *search strategy* and known as *Successive Halving* that utilizes (a) a subset of the data and (b) **small amount of resources** early in the process to find some of the best performing parameter combinations (best candidates) and then gradually increases the amount of data used and the number of resources as it narrows in on the best combinations.

Parameter Repository

The parameter repository for all transformers and classifiers used can be found below :

Scaler

Scaler	Parameter	Values

StandardScaler	*with_mean*	[True, False]
	with_std	[True, False]
RobustScaler	*with_centering*	[0, 1,2,3,4]
	with_scaling	[True, False]
	quantile_range	[0,25,50,75]
	unit_variance	[True, False]
MinMaxScaler	*feature_range*	[0, 1,2,3,4]
	clip	[True, False]
PowerTransformer	*method*	["yeo-johnson", "box-cox"]
	standardize	[True, False]
QuantileTransformer	*n_quantiles*	[5,10,15,20,25,30,35]
	output_distribution	["uniform", "normal"]
MaxAbsScaler		

Undersampler

Undersampler	Parameter	Values
RandomUnderSampler	*sampling_strategy*	’majority’, ’not majority’, ’all’, ’auto’]
NearMiss	*with_centering*	[0, 1,2,3,4]
	sampling_strategy	’majority’, ’not majority’, ’all’, ’auto’]
	n_neighbors	[1,2,3,4,5,6,7]
CondensedNearestNeighbour	*sampling_strategy*	’majority’, ’not majority’, ’all’, ’auto’]
	n_neighbors	[1,2,3,4,5,6,7]
TomekLinks	*sampling_strategy*	’majority’, ’not majority’, ’all’, ’auto’]
	standardize	[True, False]
EditedNearestNeighbours	*sampling_strategy*	’majority’, ’not majority’, ’all’, ’auto’]
	n_neighbors	[1,2,3,4,5,6,7]
	kind_sel	["all", "mode"]

Feature Selector

Feature Selector	Parameter	Values

SelectKBest	*score_func*	[mutual_info_classif,chi2,f_-classif,f_regression, mutual_info_regression]
	k	[100, 200, 500, 1000, 1400]
VarianceThreshold	*threshold*	[0.003, 0.005, 0.01, 0.05, 0.1, 0.2]
SelectPercentile	*percentile*	[1, 2, 5, 10, 50, 100]
SelectFromModel	*threshold*	[0.003, 0.005, 0.01, 0.05, 0.1, 0.2]
GenericUnivariateSelect	*score_func*	[mutual_info_classif,chi2,f_-classif,f_regression, mutual_info_regression]
	mode	["percentile", "k_best", "fpr", "fdr", "fwe"]
SelectFpr	*score_func*	[mutual_info_classif,chi2,f_-classif,f_regression, mutual_info_regression]
	alpha	[0.002, 0.005, 0.01, 0.015]
SelectFdr	*score_func*	[mutual_info_classif,chi2,f_-classif,f_regression, mutual_info_regression]
	alpha	[0.002, 0.005, 0.01, 0.015]
SelectFwe	*score_func*	[mutual_info_classif,chi2,f_-classif,f_regression, mutual_info_regression]
	alpha	[0.002, 0.005, 0.01, 0.015]

Feature Extractor

Feature Extractor	**Parameter**	**Values**
PCA	*whiten*	[False, True]
	n_components	[0.5 , 0.625, 0.75 , 0.875, 1.]
	svd_solver	["auto", "full", "arpack", "randomized"]
LinearDiscriminantAnaly-sis	*shrinkage*	["auto"]
	store_covariancebool	[True, False]
	n_components	[0.5 , 0.625, 0.75 , 0.875, 1.]
	solver	["svd", "lsqr", "eigen"]

Classifier

Classifier	Parameter	Values
AdaboostClassifier	*n_estimators*	[10, 50, 100, 150, 200, 250, 300]
	learning_rate	['constant','adaptive']
	algorithm	['SAMME', 'SAMME.R']
CalibratedClassifierCV	*method*	['sigmoid', 'isotonic']
	ensemble	[True, False]
DecisionTreeClassifier	*criterion*	["gini", "entropy"]
	splitter	["best", "random"]
	max_depth	[3,4,5,6,7]
	min_samples_split	[2,3,4]
	min_samples_leaf	[1,2,3]
ExtraTreeClassifier	*criterion*	["gini", "entropy"]
	max_depth	[3,4,5,6,7,8,9,10,11,12,13,14]
	min_samples_split	[2,3,4]
	min_samples_leaf	[1,2,3]
	max_features	["auto", "sqrt", "log2"]
GradientBoostingClassifier	*loss*	["deviance"]
	learning_rate	[0.1,0.2,0.3]
	n_estimators	[50,100,150]
	min_samples_leaf	[1,2,3]
	min_samples_split	[2,3,4]
	max_depth	[2,3,4]
	max_subsample	[0.9,0.8,0.7]
KNeighborsClassifier	*weights*	["uniform", "distance"]
	n_neighbors	[3,4,5,6,7,8,9,10,11,12,13,14]
	algorithm	["auto", "ball_tree", "kd_tree", "brute"]
KNeighborsClassifier	*weights*	["uniform", "distance"]
	n_neighbors	[3,4,5,6,7,8,9,10,11,12,13,14]
	algorithm	["auto", "ball_tree", "kd_tree", "brute"]
LinearDiscriminantAnalysis	*weights*	["uniform", "distance"]
	n_components	[6, 8, 10, 12]
	solver	['svd', 'lsqr', 'eigen']
LinearSVC	*penalty*	['l2']
	loss	['hinge', 'squared_hinge']

	C	[1, 0.1, 0.5, 5, 2]
	fit_intercept	['True','False']
	intercept_scaling	[1,0.5,2]
LogisticRegression	*C*	[1.00000000e-10, 2.15443469e-07, 4.64158883e-04, 1.00000000e+00, 2.15443469e+03, 4.64158883e+06, 1.00000000e+10]
	solver	['lbfgs', 'liblinear', 'saga']
MLPClassifier	*activation*	['tanh', 'relu']
	solver	['adam', 'sgd']
	alpha	[0.0001, 0.05]
	learning_rate	['constant','adaptive']
	hidden_layer_sizes	[(50,50,50), (50,100,50), (100,)]
NuSVC	*nu*	[0.1, 0.3, 0.5, 0.7, 0.9]
	gamma	["auto","scale"]
	kernel	["poly", "rbf", "sigmoid", "precomputed"]
RandomForestClassifier	*n_estimators*	[100, 200, 300, 400, 500, 600, 700]
	max_depth	[6,7,8,9,10,11,12,13,14]
RidgeClassifier	*alpha*	[0.001, 0.1, 1.0]
	tol	[0.1, 0.01, 0.001]
	solver	['auto', 'svd', 'cholesky','lsqr', 'sparse_cg', 'sag', 'saga']
SGDClassifier	*loss*	["hinge", "log", "squared_hinge", "modified_huber"]
	alpha	[0.0001, 0.001, 0.01, 0.1]
	penalty	["l2", "l1", "none"]
SVC	*C*	[0.5, 1, 1.5]
	kernel	['poly', 'linear', 'rbf', 'sigmoid']
	degree	[2, 3, 4],
	gamma	['scale', 'auto'],
XGBClassifier	*gamma*	[0.1, 0.25, 0.5]
	max_depth	[4, 5, 6, 7]
	colsample_bytree	[0.2, 0.4, 0.6, 0.8],
	min_child_weight	[1, 2, 3, 4]
	subsample	[0.8, 0.9, 1]

Optimization no.3 : Selecting Best Ensemble pipeline

Our approach to using the voting hard ensemble classifier involves a comprehensive evaluation of all possible combinations of the 16 different pipelines that have been previously implemented using grid search

cross-validation. Each pipeline consists of 5 stages, including a scaler, feature selector, feature extractor, under sampler, and classifier. These pipelines have been carefully designed to address specific challenges in the data, such as class imbalance and high dimensionality. By combining these pipelines, we aim to achieve a more robust and improved performance compared to the individual pipelines.

Voting ensemble classifiers are a popular machine learning technique for improving the performance of individual classifiers (or pipelines). This approach works by combining the predictions of multiple base classifiers to make a final prediction.

In the case of a **hard voting classifier**, the final prediction is based on the majority vote of the base classifiers. This approach is well suited to problems where the base classifiers are highly accurate and diverse, as the ensemble classifier can take advantage of the strengths of each individual classifier.

Soft voting classifiers, on the other hand, weight the predictions of each base classifier based on their accuracy. This approach is well suited to problems where the base classifiers are not equally accurate, as the ensemble classifier can give more weight to the more accurate base classifiers. In general, soft voting classifiers have been found to perform better than hard voting classifiers in many machine learning problems, as they can effectively capture the strengths of multiple base classifiers while mitigating their weaknesses.

However, in practice, the choice of whether to use a hard or soft voting classifier will depend on the specifics of the data and the base classifiers being used. In our case we have 16 highly accurate and diverse pipelines (as we will see in the Experimentation section) so we will proceed with the **Hard Voting Ensemble Optimization** step by finding the one (hard combination of the optimized pipelines) that lead to the highest **F1 test score**.

4.3.6 Feature Importance Algorithms

Feature (variable) importance indicates how much each feature contributes to the model prediction. Basically, it determines the degree of usefulness of a specific variable for a current model and prediction.

Feature importance refers to techniques that assign a score to input features based on how useful they are at predicting a target variable. Interpretation of a machine learning model is the process wherein we try to understand the predictions of a machine learning model. Predictive modeling lifecycle involved 2 stages:

1. Pre-Analysis : Where we keep an eye on the evaluation metric and experiment with different feature engineering, feature selection, and algorithm selection ideas in order to develop and create more robust models.

2. Post-Analysis : The second stage, is to analyze the models using the predictions and parameters to figure out why the classifier, for example, chose a particular class.

List of libraries for ML model interpretation with the respective documentation can be found below :

Model Interpreter	Documentation Link
ELI5	ELI5 Documentation
LIME	
SHAP	SHAP Documentation

ELI5

ELI5 is an acronym for *Explain like I am a 5-year old*. Python has ELI5 methods to show the functionality for both: (1) **Global interpretation-Look** at a model's parameters and figure out at a global level how the model works and (2) **Local interpretation-Look** : at a single prediction and identify features leading to that prediction.

SHAP

SHAP is an acronym for **SHapley Additive exPlanations** (SHAP) [Lundberg and Lee, 2017].The SHAP library uses Shapley values at its core and is aimed at explaining **individual predictions**. Simply put, Shapley values are derived from Game Theory, where each feature in our data is a player, and the final reward is the prediction. Depending on the reward, Shapley values tell us how to distribute this reward among the players fairly. The best part about SHAP is that it offers a special module for tree-based models. Considering how popular tree-based models are in hackathons and in the industry, this module makes fast computations, even considering dependent features.

LIME

The idea behind **Local Interpretable Model-Agnostic Explanation** (LIME) is to provide the reasons why a prediction was made [Marco Tulio Ribeiro and Guestrin, 2016]. Taking the same example, if a machine learning model predicts that a movie is going to be a blockbuster, LIME highlights the characteristics of the movie that would make it a super hit. Features like genre and actor might contribute to the movie doing well, while others like running time, director, etc. might work against it.

Mathematically, local surrogate models with interpretability constraint can be expressed as follows:

$$explanation(x) = \arg\min_{g \in G} L(F, g, \pi_x) + \Omega(g) \tag{4.1}$$

The explanation model for instance x is the model g (e.g. linear regression model) that minimizes loss L (e.g. mean squared error), which measures how close the explanation is to the prediction of the original model f (e.g. an xgboost model), while the model complexity $\Omega(g)$ is kept low (e.g. prefer fewer features) [ref, b]. G is the family of possible explanations, for example all possible linear regression models. The proximity measure π_x defines how large the neighborhood around instance x is that we consider for the explanation. In practice, LIME only optimizes the loss part. The user has to determine the complexity, e.g. by selecting the maximum number of features that the linear regression model may use.

In this thesis we present our own **LIME** approach, and how it was configured to make inferences about the entire population of the control set

μ 4.1: *The modified version of LIME algorithm for explaining machine learning models*

1. Calculate all $N = 11$ predictions and store the result in the unit vector *y_pred*.
2. Keep only the $M \leq N$ predictions that are correct, that is, we keep the $i-th$ prediction if the condition $y\_pred[i] = y\_test[i]$
3. A LIME object is initialized LIME with arguments the *X_train, feature names, class names*.
4. **For** $i \in \{0, 1, ...M\}$:

(a) The method *explain_distance*() is used to calculate the weights w_i :

$$w_i = explain\_distance(X\_test_i)$$

where w_i is the unit vector of size $(n_{features},) = (741,)$ while the w_{ij} shows the "contribution" of the $j-th$ feature in predicting the $i-th$ sample of the *X_test*, therefore the following formula should stand:

$$\sum_{i=1}^{n_{features}} w_{ij} = 1$$

5. The average weight w' for all correct predictions is finally calculated :

$$w'_j = \frac{w_{ij}}{M} \quad \forall i \in \{1, ..., M\}$$

Chapter 5

G-CNN model

In this chapter we describe the deep learning model, a Graph Convolution Neural Network (G-CNN) set to support the diagnosis of Dyslexia by classifying correlation matrices of the brain regions connected occurred after preprocessing the fMRI dataset

5.1 General Description of the Model

In this project we implemented the G-CNN deep learning model (*gcnn.py*), as (Parisot) described in her papers. A sparse graph was implemented to represent the interaction network of the 58-participants dataset based on the phenotypic data (age, gender). More precisely :

- **Nodes /** Each node represents a single participant and correspond to a single vector of features (after flattening the correlation matrices) extracted from the fMRI dataset.
- **Weights /** the weight of each edge - which connects 2 nodes) represent the similarity degree between 2 participants, and it is calculated based on the phenotypic data available

Our well-rounded approach was established in order to exploit all the available data provided (fMRI data, phenotypic). The sparse graph is then inserted in the G-CNN model as the first layer, trained in a (semi)-supervised manner by using a labeled training set and a non-labeled testing set . What makes this deep learning model special is its ability to use both imaging and non-imaging information which -we believe- are also useful for diagnosing dyslexia, by expressing the inter-between participant similarity. Through this way, we intend to achieve higher accuracy and precision results compared to our machine learning approach.

5.2 Graph Structure

The sample population consists of N participants ($N = 58$) and the goal is to predict the condition of each participant from the fMRI imaging and phenotypic data combined (which connect the individuals of the population).

- The set of participants is represented as a sparse graph $G = (V, E)$, connected with weighted edges, where the W table is the adjacent table of the graph.
- Each participant is assigned to a specific graph node in a *"1-1"* way, which is associated with a single vector of features D extracted from the fMRI dataset.
- The edges set E of the graph represents the similarities among the participants with the help of phenotypic data

In this deep learning model approach, the dyslexia diagnosis problem is reduced to resolving a node classification problem. The goal is to give a label $l \in \{0, 1, 2\}$ to every node, where l stands for the state of each participant, $l = 0$ for the **control** participants, $l = 1$ for the speaking deficiency participants and $l = 2$for the dyslectic participants . The training strategy followed here is a supervised, meaning that all the nodes are labeled and inserted in the G-CNN model for the training.

In order for this approach to deliver and achieve higher and precision results compared to our machine learning approach, we need to construct the graph accurately, even when we use only two phenotypic features. An inadequately defined graph can perform worse than a linear classifier. There are 2 elements that validate thee correct construction of the graph:

1. The single feature vector D_i of the i-th participant

2. The way the nodes-participants are connected to each other, ie. the existence of an edge between two nodes and its weight. This is the way to model and express the similarity of two nodes-participants.

5.2.1 Vector of Features D of the nodes N

The single feature vector D_i of the i-th participant is extracted purely from the imaging fMRI data, as described in Chapter 4. We are performing connectivity analyses, so BOLD timeseries (extracted from the imaging fMRI data) are compared across regions (usually with correlation) and the strength of the relationship determines their functional connectivity [REFERENCE]. In our case, we have concatenated all the ($t = 3$) BOLD fMRI images, and then calculated the BOLD timeseries on the concatenated fMRI image.

We followed 2 different **data preprocessing frameworks** to perform a connectivity analysis and acquire the correlation matrices and consequently the single feature vector D_1. The first one is the **GLM 1st Level Analysis**(TODO) an approach to extract correlation matrices based on specific contrast maps. In our case the contrast maps concern contrast difference among words, pseudowords and pauses. The second apparoc is the **Atlas Model Analysis**. Instead of creating our own seed ROIs, we can use available atlases to extract ROIs. The python library *Nilearn* provides an easy way to accomplish this.

1. The module *datasets* of the *Nilearn* library is used to import the different atlases. There are 3 kinds of atlases that can be fetched : *3D* atlases, *probabilistic* atlases and *coordinates* atlases.

2. The atlas **maps, labels** attributes are directly retrieved as attribute of the atlas object while the atlas **coordinates** are calculated depending on the atlas kind

Atlas Kind	**Coordinates**
3D	*plotting.find_parcellation_cut_coords()*
Probabilistic	*atlas attribute .coords*
Coords	*np.vstack((atlas.rois['x'], atlas.rois['y'], atlas.rois['z'])).T*

Moreover the atlas labels, as discussed in Chapter (TODO), represents all the brain regions in which our BOLD signals will be projected and have the following 1D shape

$$(n\_regions) = (R \text{ non-overlapping regions})$$

, where

$R =$ the number of non-overlapping regions of the respective atlas

3. The masker is selected and initiated differently based on the atlas kind

Atlas Kind	Masker	Initialization
3D	**Nifti Labels** masker	*maps, labels*
Probabilistic	**Nifti Maps** masker	*maps*
Coords	**Nifti Spheres** masker	*coordinates*

4. In each BOLD run session, the selected masker is fit with the fMRI BOLD data, which are then tran sformed to their final timeseries format. For the $i-th$ session the timeseries shape is :

$$(TS_i \times n\_regions) = (127 \times R)$$

Since the number of BOLD sessions are 3, all 3 timeseries are concatenated on a timestamp level. The final timeseries shape is

$$\left(\sum_{i=1}^{n=3} TS_i \times n\_regions\right) = ((127 \times 3) \times R) = (381 \times R)$$

5. Last but not least, we fit and transform the timeseries into the final correlation matrix. Each cell in the matrix reflects the correlation of the BOLD timeseries between a pair regions (of the corresponding atlas). The final data shape is

$$(n\_regions \times n\_regions) = (R \times R)$$

The final correlation matrix, of each participant, is saved as a *.nii.gz* file, for future use. All the (6) steps are time consuming. Accessing directly the correlation matrix will be helpful for our numerous **machine learning** and **deep learning** experiments.

6. During our **machine learning** and **deep learning** experiments, the final D_i single feature vector of the i-th participant is calculated based on the correlation matrix. The matrix is a symmetrical $(R \times R)$ matrix, so we only keep the upper triangular part, which is then flattened. The shape of the D_i single feature vector is :

$$\left(\frac{R \times (R-1)}{2}\right)$$

We used the specific data type to construct the single feature vector D_i as there is evidence that **dyslexia** and **speaking deficiency** in general are associated with disorders in the structural and functional organization of the brain. Due to the large dimensionality of the vector [1], we use dimensional reduction techniques and test the **machine learning** and **deep learning** models with both the whole feature vector and with vectors of reduced dimensionality.

5.2.2 The edges E of the graph G

The G-CNN model efficiency is determined by the accurate definition of the edges E and their weights W. So, both E, W should be defined in such a way as to reflect as accurately as possible the similarities

[1] $R \geq 48$, so the we have at least $\frac{48 \times 47}{2} \sim 1128$ features

among the nodes-participants N of the graph. On classic CNN models, the pixel neighbourhoods affect the convolution process. On G-CNN models, the selection of E, W of the graph also affects the convolutions process. Therefore, we should use the phenotypic data available in such a way that express and explain more efficiently the similarities among the examine-es and their fMRI images.

In this project, the available phenotypic data were : (1) gender (2) age.

- The gender is selected because males are diagnosed with dyslexia more frequently than females, even in epidemiological samples. The same research showed that processing speed helps explain the sex difference in dyslexia [Döpfner et al., 2008]

- The age is selected because just as any other physiologic trait, the dyslexic cognitive and neural phenotype is likely to influence the aging process [ref, 2022a]

The weight $w(u, v)$ between the nodes u and v can be defined as followed :

$$w(\boldsymbol{u}, \boldsymbol{v}) = sim(x(\boldsymbol{u}), x(\boldsymbol{u})) \cdot \sum_{i=0}^{H} \gamma(M_i(\boldsymbol{u}), M_i(\boldsymbol{v}))$$

where

Factor	Explanation
$\boldsymbol{sim}(x(v), x(u))$	the similarity coefficient between feature vector $x(u)$ and $x(u)$ of the nodes-participants u, v. In this project, we used many different similarity functions for our experiment, as they are described in Chapter (TODO). For reference, the exponential similarity function used is : $$sim(x(\boldsymbol{u}), x(\boldsymbol{v})) = e^{\frac{[\rho(x(\boldsymbol{u}), x(\boldsymbol{v})]^2}{2\sigma^2}}$$ where $\boldsymbol{\rho}$ is the correlation function and $\boldsymbol{\sigma}$ determines the kernel's range
H	The total number of phenotypic data available for the calculation of the edge weigh. Here $H = 2$
M_i	The function used to measure the phenotypic information eg. M_i = 9 years old
$\boldsymbol{\gamma}$	is a function to compare phenotypic information • **Categorical Features** (eg. Gender) $\gamma(M_i(\boldsymbol{u}), M_i(\boldsymbol{v})) =$ Kronecker $\delta(M_i(\boldsymbol{u}), M_i(\boldsymbol{v})) = \begin{cases} 1, & M_i(\boldsymbol{u}) = M_i(\boldsymbol{v}) \\ 0, & M_i(\boldsymbol{u}) \neq M_i(\boldsymbol{v}) \end{cases}$ • **Numerical Features** (eg. Age) $\gamma(M_i(\boldsymbol{u}), M_i(\boldsymbol{v})) = \begin{cases} 1, & \lvert M_i(\boldsymbol{u}) - M_i(\boldsymbol{v})\rvert < \partial \\ 0, & \lvert M_i(\boldsymbol{u}) - M_i(\boldsymbol{v})\rvert \geq \partial \end{cases}$ where ∂ is a threshold (than can be optimized) defining when 2 eg. Ages are considered to be close

Therefore, 2 graph nodes-participants are connected with an edge weighing

$$w_0 \in \{0, 1, 2\}$$

based on **how many of their phenotypic information are considered close**. Then, this w_0 is multiplied by the nodes **similarity coefficient**, giving the final weight w of the edge

$$w = sim(x(\boldsymbol{u}), x(\boldsymbol{v})) \cdot w_0$$

5.2.3 The architecture of the G-CNN model

Our model architecture is illustrated in Figure 5.1. The model consists of a fully convolutional GCN with L hidden layers activated using the Rectified Linear Unit (ReLU) function. The output layer is followed by a softmax activation function. The graph is trained using the whole population graph as input. In addition we use a **cross entropy loss** function for the optimisation process. After training the G-CNN model, the softmax activations are computed on the test set, and the unlabelled nodes are assigned the labels maximising the softmax output.

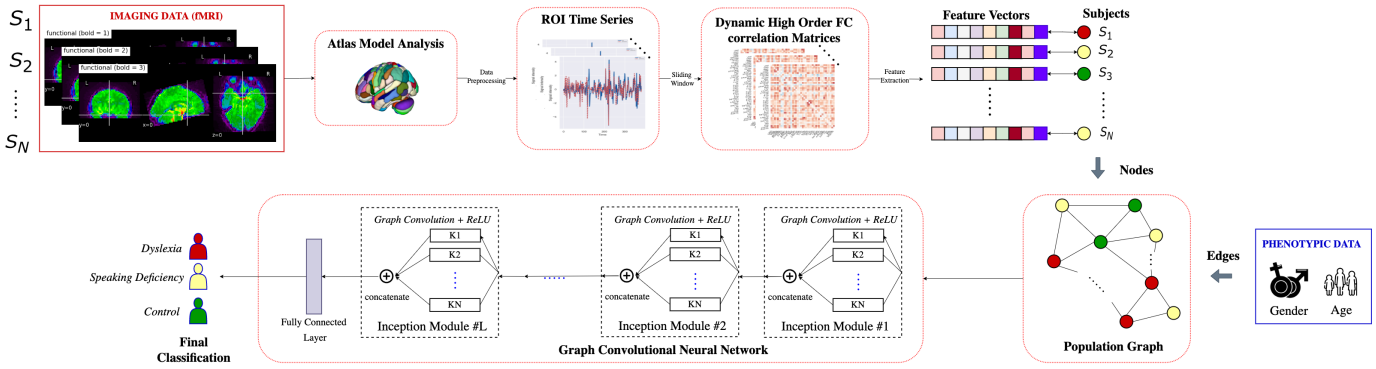


Figure 5.1. *The architecture of the G-CNN model for the multi-class classification problem of dyslexia*

5.2.4 Cost Function, Optimizer, Metrics

In order for the G-CNN model to learn the optimal parameters, we use an optimizer of the parameters, based on the calculation of the gradient $\nabla_\partial L(y, \hat{y})$ to find a local minimum. We specifically used **ADAM optimizer (ADAptive Momentum)** which calculates as much as ADAGRAD and also calculates the learning rate and momentum changes for each paramete. **Adam** is a computationally efficient algorithm, easy to implement and has limited memory requirements. It works well on large datasets with large parameters as well as problems with noisy or sparse gradients, as in the present diploma.

For the cost function : The cross-entropy is the average minimum information encoding size of communicating an event from one probability distribution to another. The discrete cost function cross-entropy (also referred to as the negative log likelihood function) is used when a probabilistic interpretation of the results is desired. Since we have sparse gradients we use *SparseCategoricalCrossentropy*

Table 5.1. *Optimizer, Cost Function and Metric used for the Deep Learning approach*

Optimizer	**keras.optimizers**.*Adam*
Loss Function	**keras.losses**.*SparseCategoricalCrossentropy(from_logits = True)*
Metics	**keras.metrics**.*SparseCategoricalAccuracy(name = "acc")*

Chapter 6

Experiments & Results

We consider a problem of diagnostic pattern recognition/classification from neuroimaging data. We propose a common data analysis pipeline for neuroimaging-based diagnostic classification problems using various ML algorithms and processing toolboxes for brain imaging. We already analyzed the theoretical framework behind our machine and deep learning approach, as well as the modified feature importance algorithm which will be used to extract insights for the learning disability of **dyslexia**. Finally we present the experimental set-up. In this chapter we present the result of all the underlying experiments.

The results of the experiments showed that the optimization techniques used in the framework provided accurate results in terms of classifying participants into the correct group. The results were statistically significant and demonstrated the potential of machine learning in the analysis of fMRI data for the classification of neurological conditions. The findings from this study have important implications for the field of neuroscience and could be used to develop new and more effective diagnostic tools for dyslexia, spelling disorder, and other neurological conditions. Additionally, the results of this study demonstrate the potential of using brain connectivity patterns in the analysis of fMRI data and could lead to new approaches in the field of brain imaging.

6.1 Machine Learning Framework

6.1.1 Selecting the best Transformers per Classifier

No. Experiments

For each one of these 16 classifiers, we run an exhaustive optimization with 4 nested loops (1 for each type of transformers). The total number of combinations examined are

$$|E|_{optimization_1} = |S| \times |U| \times |FT| \times |FE| \times |C| \sim 7680$$

No. Experiments	Duration	Duration per Experiment
7680	3047 sec	0.4 sec

According to the proposed schema for the optimization no.1 (see (See Figure 6.1) for details), scikit-learn was used as it provides a plethora of transformers. Moreover, in order to validate the consistency of the experiment set-up and the reproducibility of its results, we use the same SEED. In each iteration, the respective combination of transformers was used to initialize the pipeline, which was fit on the **training** set and evaluated on the **test** set by measuring **F1 score**. The results are shown below in table 6.1

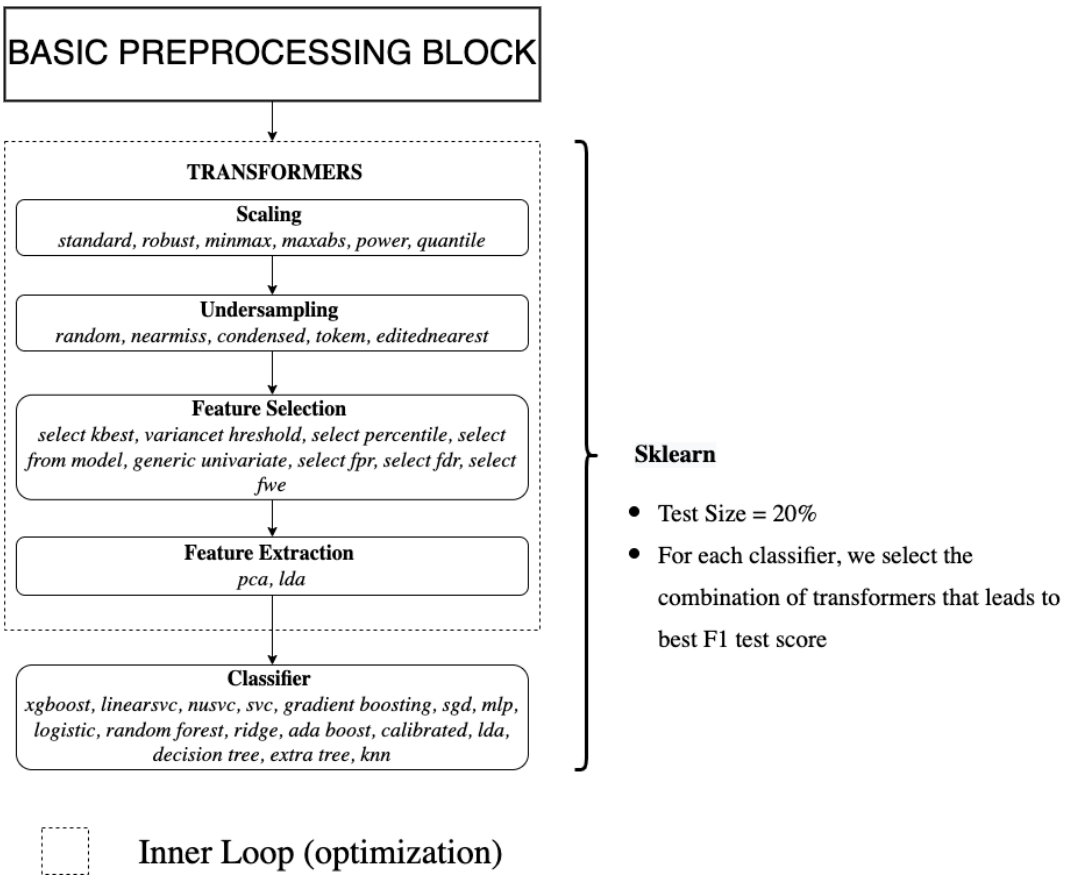


Figure 6.1. *Flow Diagram illustrating Optimization no.1*

Results

Some useful insights that can be extracted are

- The best F1 score on the test set was achieved by the **XGBoost Classifier** and it is $F1 = 76.48\%$
- The performance results should not be taken into consideration to professionally evaluate the models. This optimization step is purely dedicated to selecting the best **transformers combination** and build a proper pipeline for each classifier

6.1.2 Optimizing all 16 Pipelines' parameters

No. Experiments

In this optimization step, we tune our hyperparameters for all 16 pipelines. In order to do that we used the HalvingGridsearchCV which was well-described in chapter 4, which is faster, computationally lighter and occasionally more efficient than the classic GridsearchCV algorithm.

The total number of experiments can be found below. To find the total number of fits in the first iteration of the algorithm, we multiply the number of experiments by the $cv = 10$. The total number of experiments is the **product** of the number of values for all underlying parameters of the pipeline.

$$|fits| = |E|_{optimization_2} \times cv = |E|_{optimization_2} \times 10$$

where

$$|E|_{optimization_2} = |S_{parameter\_values}| \times |U_{parameter\_values}| \times |FT_{parameter\_values}| \times |FE_{parameter\_values}| \times |C_{parameter\_values}|$$

Table 6.1. *F1 test score for all 16 classifiers afters Optimization no.1*

Classifier	Scaler	Undersampler	Feature Selector	Feature Extractor	F1 score
adaboost	quantile	nearmiss	selectpercentile	pca	0.6238636364
calibrated	quantile	tomek	selectfpr	lda	0.5737433862
decisiontree	power	random	selectfpr	pca	0.6666666667
extratree	quantile	nearmiss	selectpercentile	lda	0.603505291
gradientboosting	powerpower	random	selectpercentile	pca	0.6478174603
knn	minmax	condensed	selectkbest	lda	0.5625
lda	standard	tomek	variancethreshold	lda	0.5833333333
linearsvc	quantile	tomek	selectfpr	lda	0.5737433862
logistic	standard	editednearest	selectkbest	lda	0.6547619048
mlp	standard	condensed	variancethreshold	lda	0.5694444444
nusvc	power	condensed	selectfpr	pca	0.6537037037
randomforest	power	nearmiss	variancethreshold	lda	0.5865800866
ridge	maxabs	random	variancethreshold	pca	0.5873015873
sgd	quantile	tomek	selectfpr	lda	0.5737433862
svc	power	condensed	selectfrommodel	lda	0.6238636364
xgboost	power	editednearest	selectfpr	pca	0.7648809524

Classifier	No. Fits
adaboost	14112
calibrated	4480
decisiontree	11520
extratree	54432
gradientboosting	69984
knn	268800
lda	1152
linearsvc	20160
logistic	32256
mlp	89600
nusvc	18144
randomforest	1512
ridge	53760
sgd	48384
svc	89600
xgboost	110592

The results of the whole gridesearch analysis

AdaBoost Classifier

The best parameters are presented below

Scaler	Undersampler	Feature Selector	Feature Extractor	Classifier
robust	random	selectpercentile	pca	adaboost
with_centering *True*	**sampling_strategy** *'majority'*	**percentile** *50*	**whiten** *True*	**n_estimators** *50*
with_scaling *True*			**n_components** *0.875*	**algorithm** *'SAMME'*
quantile_range *40*			**svd_solver** *"auto"*	
unit_variance *True*				

Calibrated Classifier

The best parameters are presented below

Scaler	Undersampler	Feature Selector	Feature Extractor	Classifier
quantile	tomek	selectfpr	lda	adaboost
n_quantiles *150*	**sampling_strategy** *'majority'*	**score_func** *'mutual_info_classif'*	**store_covariancebool** *True*	**method** *sigmoid*
output_distribution *"uniform"*		**alpha** *0.01*	**n_components** *0.875*	**ensemble** *True*
		solver *svd*		

Decision Tree Classifier

The best parameters are presented below

Scaler	Undersampler	Feature Selector	Feature Extractor	Classifier
power	random	selectfpr	pca	decisiontree
				criterion *"entropy"*
				max_depth *10*
				min_samples_leaf *1*
				min_samples_split *4*
				splitter *"random"*

Extra Tree Classifier

Scaler	Undersampler	Feature Selector	Feature Extractor	Classifier
quantile	nearmiss	selectpecentile	pca	extratree
				criterion *"gini"*
				max_depth *10*
				min_samples_leaf *1*
				min_samples_split *4*
				max_features *"log2"*

Gradient Boosting Classifier

Scaler	Undersampler	Feature Selector	Feature Extractor	Classifier
power	random	selectperentile	pca	gradientboosting
				learning_rate *0.2*
				loss *"deviance"*
				max_depth *2*
				min_samples_split *2*
				n_estimators *150*

K Neighbor Classifier

Scaler	Undersampler	Feature Selector	Feature Extractor	Classifier
minmax	condensed	selectkbest	lda	knn
				algorithm *"ball_tree*
				n_neighbors *3*
				weights *"distance"*

LinearSVC

Scaler	Undersampler	Feature Selector	Feature Extractor	Classifier
quantile	tomek	selectfpr	lda	linearsvc
				C *1*
				fit_intercept *True*
				intercept_scaling *1*
				loss *hinge*
				penalty *l2*

Logistic Regression

Scaler	Undersampler	Feature Selector	Feature Extractor	Classifier
standard	editednearest	selectkbest	lda	logistic
				C *1e-10*
				solver *"lbfgs"*

MLP Classifier

Scaler	Undersampler	Feature Selector	Feature Extractor	Classifier
standard	condensed	variancethreshold	lda	mlp
				activation *"tanh"*
				alpha *0.0001*
				hidden_layer_sizes *[50, 100, 50]*
				learning_rate *"constant"*
				solver *"adam"*

NuSVC

Scaler	Undersampler	Feature Selector	Feature Extractor	Classifier
power	condensed	selectfpr	pca	nusvc
				gamma *"auto"*
				kernel *"poly"*
				nu *0.5*

Random Forest Classifier

Scaler	Undersampler	Feature Selector	Feature Extractor	Classifier
power	nearmiss	variancethreshold	lda	randomforest
				max_depth *6*
				n_estimators *500*

Ridge Classifier

Scaler	Undersampler	Feature Selector	Feature Extractor	Classifier
maxabs	random	variancethreshold	pca	ridge
				alpha *0.001"*
				solver *"saga"*
				tol *0.01*

SGD Classifier

Scaler	Undersampler	Feature Selector	Feature Extractor	Classifier
quantile	tomek	selectfpr	lda	sgd
				alpha *0.1"*
				loss *"modified_huber"*
				penalty *none*

SVC

Scaler	Undersampler	Feature Selector	Feature Extractor	Classifier
power	condensed	selectfrommodel	lda	svc
				C *1.5*
				degree *4*
				gamma *"auto"*
				kernel *"linear"*

XGBoost Classifier

Scaler	Undersampler	Feature Selector	Feature Extractor	Classifier
power	editednearest	selectfpr	pca	xgboost
				colsample_bytree *0.2*
				max_depth *4*
				gamma *0.1*
				min_child_weight *1*
				subsample *0.8*

Results

In Figure 6.2 we visualize the summary of the structure of the optimized pipeline for the Adaboost Classifier, while in Figure 6.3 we present the results for the 2 optimizations 2 different optimization steps combined. The key metric is **F1 score**.

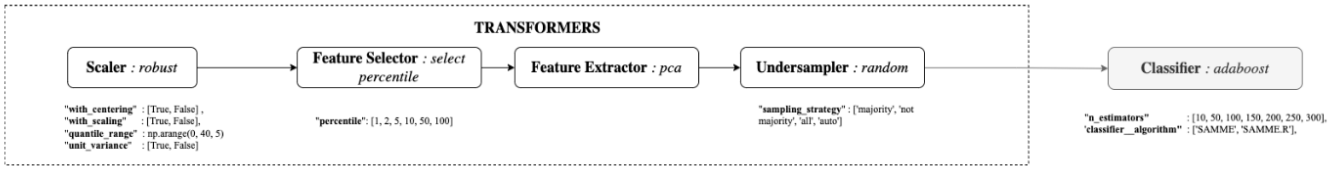


Figure 6.2. *Pipeline for Adaboost Classifier*

Some useful insights that can be extracted are

- The best F1 score on the test set was achieved by the **XGBoost Classifier** and is $F_1 = 76.5\%$

- The worst optimized pipeline is SVC and NuSVC with test score $F_1 = 36.1\%$ and $F_1 = 27.8\%$ which is even less than a baseline classifier. In fact, LinearSVC also belongs to the same family of classifiers and achieved a test score of $F_1 = 57.4\%$. In general, non-linear classifiers are more suitable for modeling more complex functions than linear ones, but depends on the data, the selected hyperparameters (e.g. penalty and kernel) and the interpretation of the results

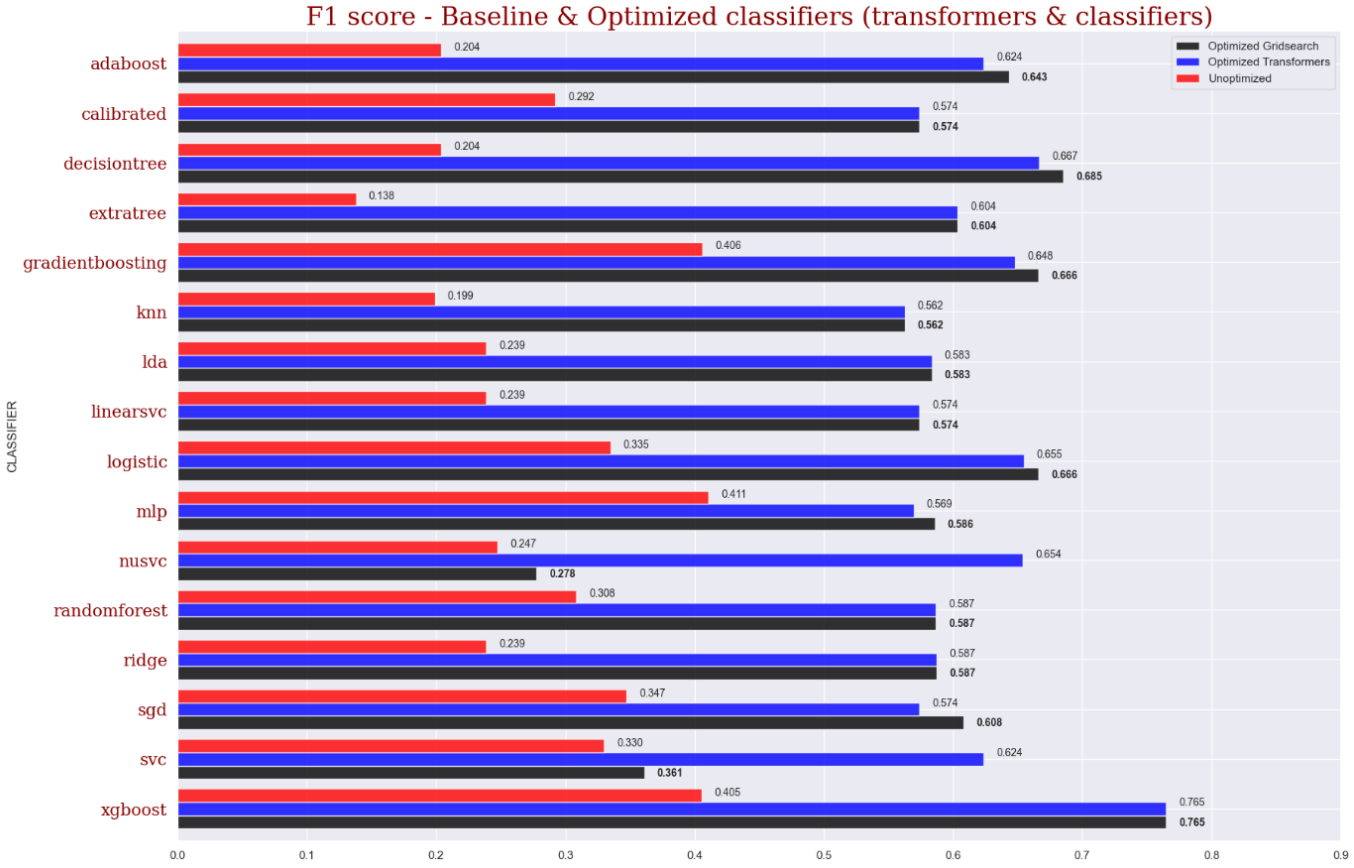


Figure 6.3. *The performance results for the F1 test score for the 2 different optimization steps*

6.1.3 Selecting Best Ensemble Classifier

No. Experiments

These optimized in terms of *transformers combination* and *parameters* pipelines, are given as input to the Ensemble Voting Classifiers. As we analyzed in the previous chapter, there are many different kind of Ensemble classifiers, eg. **Hard Voting Ensemble Classifier** which employs the majority rule, **Soft Voting Ensemble Classifier** which is based on probability averaging etc. In this diploma thesis we only use the Hard Voting approach, however experimenting with the Soft Voting approach and optimizing the underlying weights can be part of our future work.

In this Hard Voting approach, we are searching for the best combination of optimized pipelines which leads to the best F1 score on the test set. We experiment with all different combinations of these optimized pipelines, so the actual number of experiments can be found as following :

$$|E|_{optimization_3} = \sum_{k=1}^{16} \binom{16}{k} = 65535$$

Results

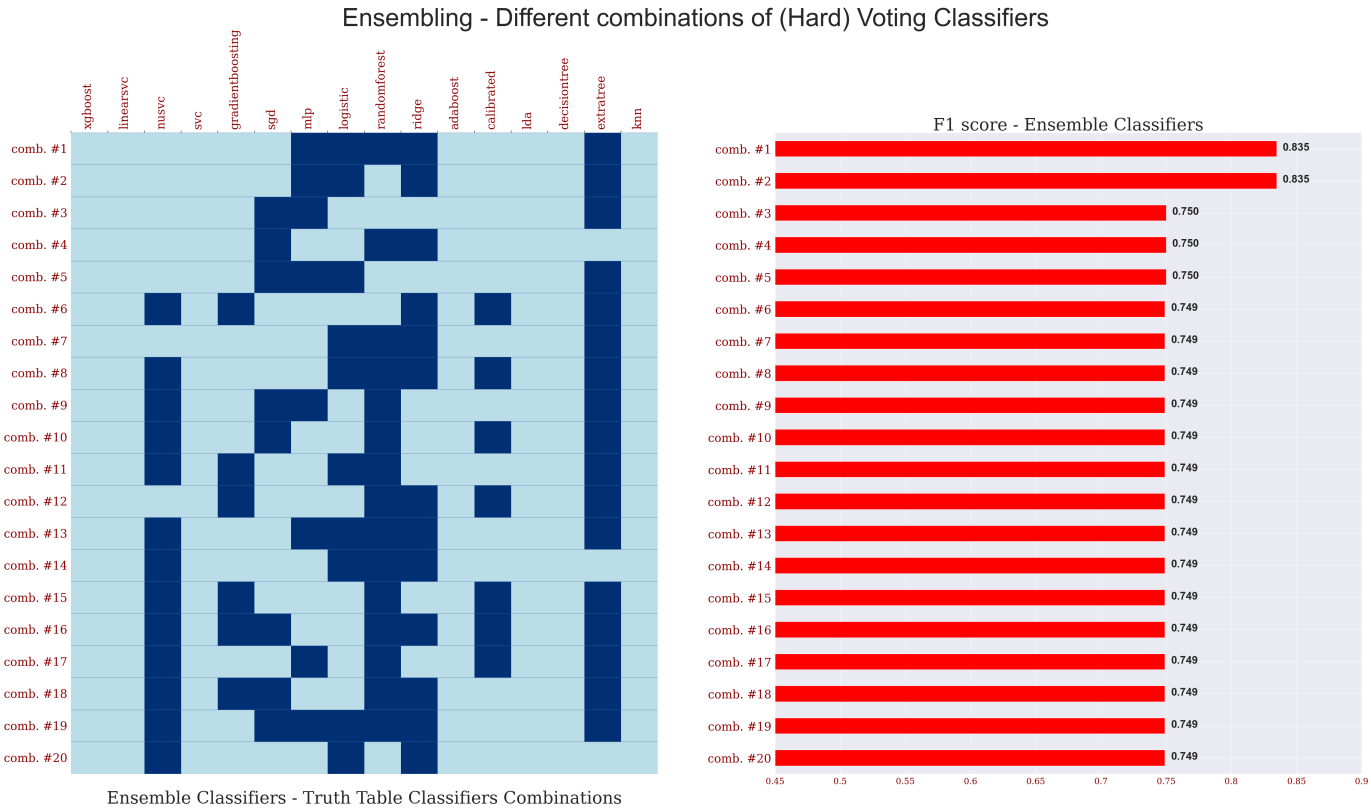


Figure 6.4. *Top 20 (F1 score) combinations of Hard Voting Ensemble Classifiers*

- This optimization step is purely executed in order to optimize our models performance. Since Ensemble Classifiers are high level abstract classifiers (abstract structure), we can NOT clearly and easily interpret their black boxes. For this reason, we will not take them into consideration for extracting feature importance and insights on the dyslexia learning disability.
- Interesting is that the best optimized (from optimization step no.2) pipeline is **XGBoost Classifier** and was not selected in any of the top combinations in this optimization step.
- The best F1 test score achieved is $F_1 = 83.5\%$ by the combination of : *MLP Classifier, Logistic Classifier, Random Forest Classifier, Ridge Classifier, Extra Tree Classifier*

6.1.4 Feature Importance

In this section we use the modified version of the LIME algorithm. We apply the algorithm to the best optimized pipeline (in terms of test score F1), which is XGBoost classifier that achieved $F_1 = 76.5\%$. We will not use any of the Ensemble Classifiers because of their high level structure and the difficulty in their interpretation. This modified version takes into consideration only the correct predicted test samples, minimizing the inclusion of bias from faulty predictions, therefore our insights are more useful and accurate.

The advantage of LIME algorithm is that provides feature impotance for all features of the test sample for each class separately. In our case we have 3 classes (*Dyslexia, Speaking Deficiency, Control*). So, the algorithm returns the weight 3D vector w, where the value $w_{ijk} \in \mathcal{R}$ shows the importance of the *j-th* feature for the model to correctly predict the *k-th* class in the *i-th* test sample , where $i = 1, 2, ..., N$, N the number of the correct predictions, $j = 1, 2, ..., 741$ and $k \in \{SpD, TD, DL\}$.

Top 20 non common features for each class

In order to retrieve insights in a population level we need to focus on the average feature importance . For this section we calculate the average absolute impact of the *j-th* feature in the *k-th* class as following :

$$|\overline{w_{jk}}| = \frac{\sum_{i=1}^{N} |w_{ijk}|}{N}$$

We used the absolute value, because we are interested in the **power** of the respective feature and not its *positive or negative impact* on our predictions. In the Figure 6.5, we can clearly inspect the **top** 20 features (in terms of weight/importance $\overline{w_{kl}}$) separately for each class. Some features with high importance may appear to more than 1 class (overlapping) indicating their interpretablity power.

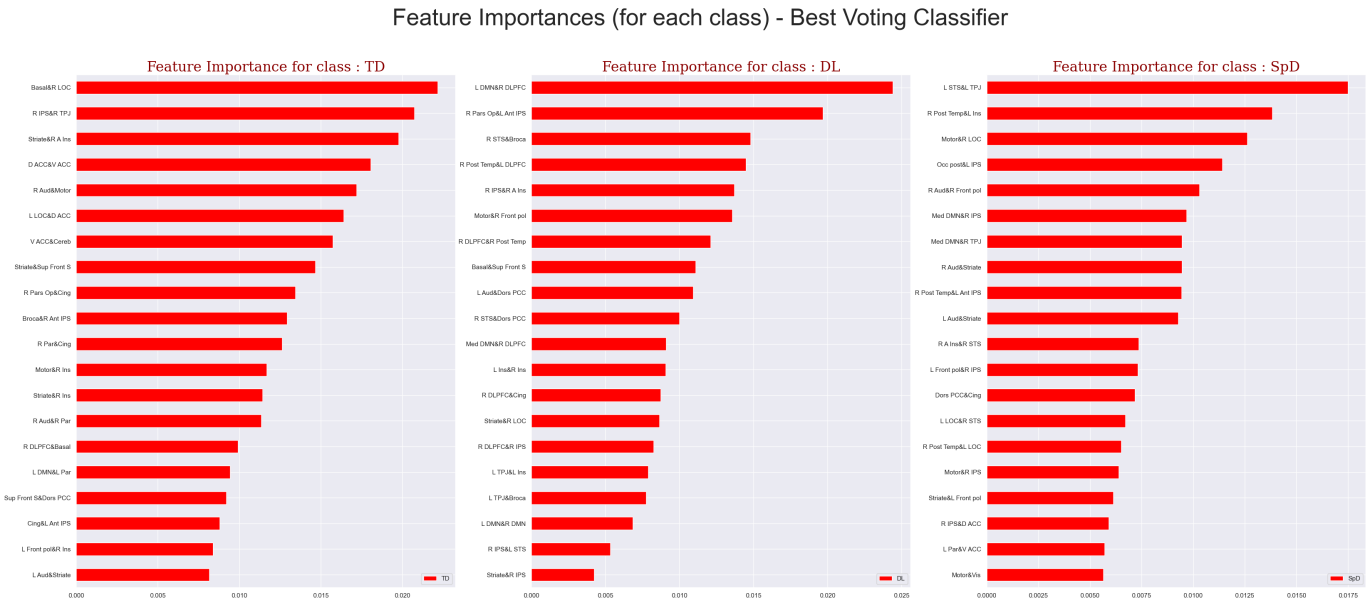


Figure 6.5. *Barplot of the feature importance for the top 20 non-common features for all 3 classes*

Skilled (TD)

Let's focus on the top 5 features. These are

- The connection of the regions **Basal** <> **R LOC**)
- The connection of the regions **R IPS** and **R TPJ**
- The connection of the regions **Striate** and **R A INS**
- The connection of the regions **D ACC** and **V ACC**
- The connection of the regions **R Aud** and **Motor**

Dyslexia (DL)

Let's focus on the top 5 features. These are

- The connection of the regions **L DMN** <> **R DLPFC**
- The connection of the regions **R Pars Op** and **L Ant IPS**
- The connection of the regions **R STS** and **Broca**

- The connection of the regions **R Post Temp** and **L DLPFC**
- The connection of the regions **R IPS** and **R A Ins**

Speaking Deficiency (SpD)

Let's focus on the top 5 features. These are

- The connection of the regions **L STS <> L TPJ**
- The connection of the regions **R Post Temp** and **L ins**
- The connection of the regions **Motor** and **R LOC**
- The connection of the regions **Occ post** and **L IPS**
- The connection of the regions **R Aud** and **R Front pol**

Results

Some useful insights that can be extracted

- The regions **D ACC,V ACC, R A INS** are highly activated in **skilled** children while for the rest 2 groups of children they are not. The role of these are closely related to **reward processing in social evaluation and empathy**
- We have clear indication that the **Broca's area** plays a key role in children with **Dyslexia**, which is totally according to our bibliography. For the rest 2 groups , **Broca's area** does not participate in any feature even in their top 20. Based on , dyslexics activated severely restricted areas of the language cortex, in fact Broca's area alone [Lishman, 2003]
- Typical developmental dyslexics have a dysfunction of the phonological and orthography to phonology conversion systems (particularly for reading and reading-like behaviors and for visuo-phonological tasks), in which the left [Eraldo Paulesu and Berlingeri, 2014]. Indeed here the brain region **Occ post** is not activated and not participate in any of the top 20 features.
- The activation of **Stratium**, which coordinates decision-making, motivation, reinforcement, and reward perception is higher (but comparable) to the activation in the rest 2 groups.
- Another interesting observation is that the region of **Left Anterior Intraparietal Sulsus**(L Ant IPS) is highly activated in children with **dyslexia** and **speaking deficiency**, which is according to the bibliography. More specifically, dyslexics show consistently higher activation in the left precentral gyrus (PRG) [Richlan F, 2009]
- Last but not least we list all conclusions of the initial paper [Banfi et al., 2020], on which our diploma thesis analysis is based, that overlap with our conclusions
  - We found reduced functional activity **cingulate cortices** (Cing) regions in **SpD** compared to the rest 2 groups
  - Children with dyslexia showed a more widespread alteration of brain activity compared to the other two groups in our ROIs.

Top 20 common features for all classes

In this section, we intend to bring more useful insights about dyslexia, by interpreting the *positive* or *negative* impact of the features. For this purpose, we calculate the top 20 common features among all ($M = 741$) for all 3 classes. For the *j-th* feature we calculate the sum of the absolute values in all 3 classes, indicating the total importance of the feature, as following :

$$|\overline{w_j}| = |\overline{w_{jSpD}}| + |\overline{w_{jTD}}| + |\overline{w_{jDL}}|$$

Finally,we select the top 20 features with the highest weight/importance $|\overline{w_j}|$. In this point,the selection process is over. For these 20 features and from now on, we are using their normal weight $\overline{w_{jk}}$ which clearly indicates the *positibe* or *negative* importance of the *j-th* feature in the *k-th* class separately. In the Figure 6.6, we can clearly inspect these 20 top features projected differently into 2 subfigures (a) Correlation Matrix and a (b) Brain CConnectome for each one of these 3 classes.

- Correlation Matrix : The first one is the $(39, 39)$ correlation matrix in which only the top 20 features are visualized. A scalable **red-blue** color map is also presented. The red color indicates the **positive** impact of the respective feature (connection between 2 regions), while the blue indicates the **negative impact**.
- Brain Connectome : The MSDL Atlas brain connectome which purely and clearly visualizes the features as edges between the respective regions. The colormap is also applied here to show the impact.

Results

The Figure 6.6 is very powerful and many insights can be extracted in a **class** but also in a **intra-class** level.

- The very first and most important insight is that all edges/connections have a clear **positive** impact to determine that a child is *skilled* and a clear **negative** impact to determine that a child has **speaking deficiency** or **dyslexia**

6.2 Deep Learning Framework : G-CNN Model

6.2.1 No. Experiments

Having structured the architecture of our model, we want to tune/optimize the hyperparameters by running proper experiments, in order to achieve the best test f1 score and avoid overfitting, a problem that is very common the field of deep learning when the models are trained with a small number of samples. The hyperparameter repositoryory can be found in the **Table 6.2**, while the number of total experiments can be calculated as following :

$$|E| = |epochs| \times |\kappa| \times |\partial| \times |d| \times |h| \times |lr| \sim 985600$$

6.2.2 Loss, Accuracy

Both loss and accuracy lines for validation and training sets can be found in the Figure 6.7. The G-CNN deep learning model was trained on only 55 samples using correlation matrices from fMRI data to predict

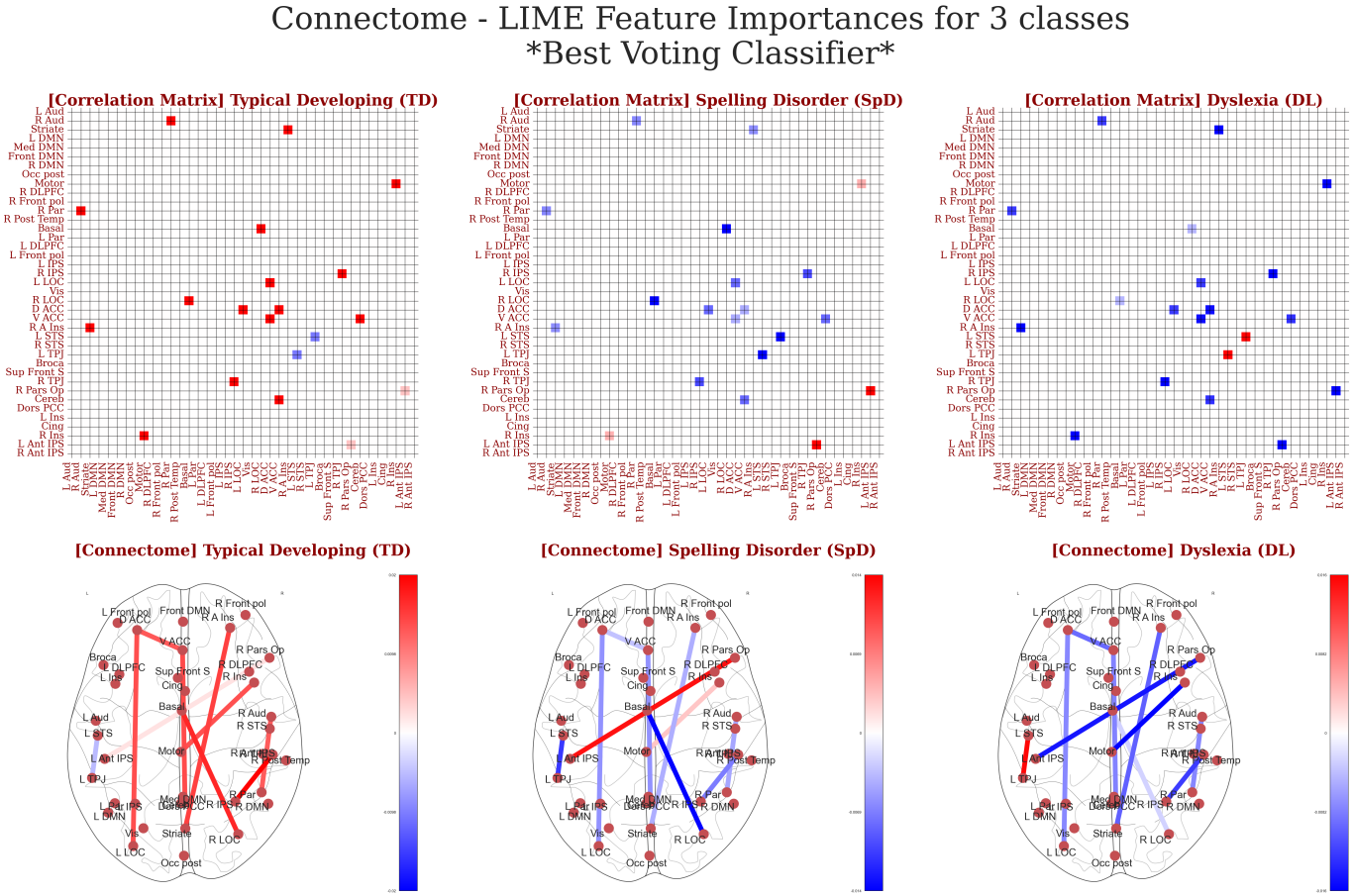


Figure 6.6. *Correlation Matrix & Connectome for the feature importance of the top* 20 *common features for all* 3 *classes*

among 3 classes of Dyslexia, Spelling Disorder, and Control. Despite the small training set, the model showed promising results. The validation loss decreased with the increase of epochs, indicating no signs of overfitting. Additionally, the validation accuracy of the model increased and reached a peak of 60%, which suggests that the model was able to successfully differentiate between the three classes. While the accuracy may not be ideal, considering the small training set and the complexity of the problem, the results are still impressive and demonstrate the potential of the G-CNN model in predicting learning disorders. Further research is needed to improve the accuracy and validate the model on larger datasets.

Table 6.2. *All the hyperparameters used for the tuning of the G-CNN deep learning model*

Hyperparameter	**Value Options**	**Best Value**
Epochs	$e \in (20, 240)$ with step = 20	80
κ	$k \in (1, 14)$	4
∂	$\partial \in (1, 5)$	3
Dropout	$d \in (0.05, 0.8)$	0.6
Hidden Units	$h \in \{[2], [8], ..., [4, 8, 16, 32, 64]\}$	[8]
Learning Rate Units	$lr \in ([1e-1, 1e-8])$	$1e-4$

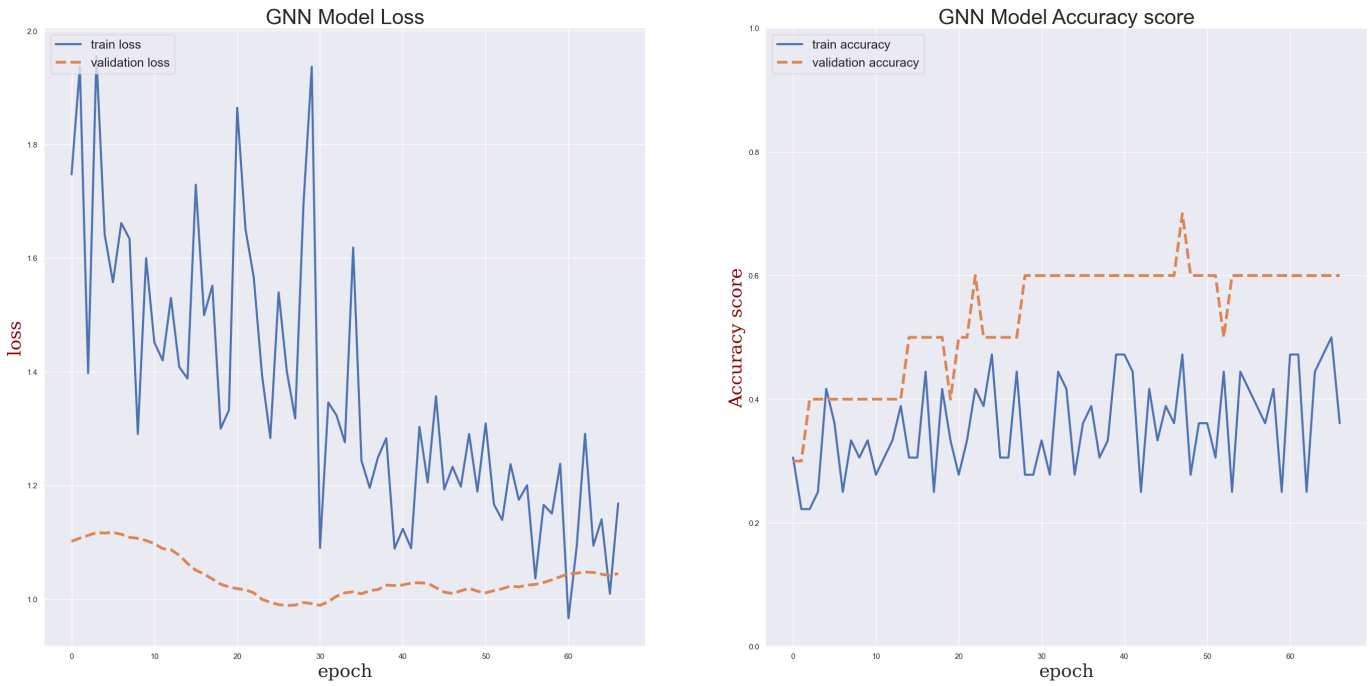


Figure 6.7. *Loss & Accuracy curves for the training and the validation set*

Bibliography

[ref, a] fmri_09_roianalysis. https://andysbrainbook.readthedocs.io/en/latest/fMRI_Short_Course/fMRI_09_ROIAnalysis.html.

[ref, b] Local surrogate (lime). https://christophm.github.io/interpretable-ml-book/lime.html.

[ref, c] Αίτια της δυσλεξίας. https://www.evrymathia.com.gr/dislexia/Αίτια-της-δυσλεξίας. Ημερομηνία πρόσβασης: 20-08-2022.

[ref, 2022a] (2022a). Adults and aging. http://www.latex-project.org. Online;.

[ref, 2022b] (2022b). Types of dyslexia. https://www.dyslexia-reading-well.com/types-of-dyslexia.html. Online;.

[ad Rhonda B.Friedman12, 1990] ad Rhonda B.Friedman12, G. G. (1990). The continuum of deep/phonological alexia*. *Cortex*, 26(3):343–359.

[B. Apolloni and Patnaik, 2005] B. Apolloni, A. Ghosh, F. A. and Patnaik, S. (2005). *Machine learning and robot perception*, volume 7. Springer Science Business Media.

[Banfi et al., 2020] Banfi, C., Koschutnig, K., Moll, K., Schulte-Körne, G., Fink, A., and Landerl, K. (2020). Reading-related functional activity in children with isolated spelling deficits and dyslexia. In *Language, Cognition and Neuroscience*.

[Behzadi et al., 2007] Behzadi, Y., Restom, K., Liau, Liu, J. ., and T. (2007). A component based noise correction method (compcor) for bold and perfusion based fmri. *NeuroImage*, 37(1):90–101.

[Bishop and Nasrabadi, 2006] Bishop, C. M. and Nasrabadi, N. M. (2006). *Pattern recognition and machine learning*, volume 4. Springer.

[Cox et al., 1997] Cox, Hyde, R. W. ., and S., J. (1997). Software tools for analysis and visualization of fmri data. *NMR in Biomedicine*, 10(4):171–178.

[D. G. Kleinbaum and Klein, 2002] D. G. Kleinbaum, K. Dietz, M. G. M. K. and Klein, M. (2002). *Logistic regression*. Springer.

[Döpfner et al., 2008] Döpfner, M., Görtz-Dorten, A., Lehmkuhl, G., Breuer, Goletz, D. ., and H. (2008). *DISYPS-II. Diagnostik-System für psychische Störungen nach ICD-10 und DSM-IV für Kinder und Jugendliche-II [DISYPS-II. Diagnostic assessment system for mental disorders in children and adolescents according to ICD-10 and DSM-IV, 2nd ed.]*. Hogrefe.

[Eraldo Paulesu and Berlingeri, 2014] Eraldo Paulesu, L. D. and Berlingeri, M. (2014). Reading the dyslexic brain: multiple dysfunctional routes revealed by a new meta-analysis of pet and fmri activation studies. *Frontiers in Human Neurosciense, Section Speech and Language*.

[Fielding, 1999] Fielding, A. (1999). *Machine learning methods for ecological applications*. Springer Science Business Media.

[Galuschka et al., 2016] Galuschka, Schulte-Körne, K. ., and G. (2016). The diagnosis and treatment of reading and / or spelling disorders in children and adolescents. *Deutsches Arzteblatt international*, 113:279–286.

[Gong and Xu, 2007] Gong, Y. and Xu, W. (2007). *Machine learning for multimedia content analysis*, volume 30. Springer.

[Greve et al., 2009] Greve, D., N., . F., and B. (2009). Accurate and robust brain image alignment using boundary-based registration. *NeuroImage*, 48(1):63–72.

[J. D. Malley and Pajevic, 2011] J. D. Malley, K. G. M. and Pajevic, S. (2011). *Statistical learning for biomedical data*. Cambridge University Press.

[Jenkinson et al., 2002] Jenkinson, M., Bannister, P., M., B., and S., S. (2002). Improved optimization for the robust and accurate linear registration and motion correction of brain images. *NeuroImage*, 17(2):825–841.

[Jenkinson et al., 2001] Jenkinson, M., . S., and S. (2001). A global optimisation method for robust affine registration of brain images. *Medical Image Analysis*, 5(2):143–156.

[L. Gy¨orfi and Walk, 2012] L. Gy¨orfi, G. O. and Walk, H. (2012). *Machine learning for financial engineering*.

[Lanczos and C., 2007] Lanczos and C. (2007). Evaluation of noisy data. *Journal of the Society for Industrial and Applied Mathematics Series B Numerical Analysis*, 1(1):76–85.

[Laura Tomaz Da Silva and Buchweitz, 2021] Laura Tomaz Da Silva, Nathalia Bianchini Esper, D. D. R. F. M. and Buchweitz, A. (2021). Visual explanation for identification of the brain bases for developmental dyslexia on fmri data. *Frontiers in Computational Neuroscience*.

[Lishman, 2003] Lishman, W. A. (2003). Developmental dyslexia. *Journal of Neurology, Neurosurgery and Psychiatry*.

[Lundberg and Lee, 2017] Lundberg, S. and Lee, S.-I. (2017). A unified approach to interpreting model predictions. *Computer Science, Artificial Intelligence*.

[M. Schmidt and Bach, 2017] M. Schmidt, N. L. R. and Bach, F. (2017). *inimizing finite sums with the stochastic average gradient*, volume 162. Mathematical Programming.

[Marco Tulio Ribeiro and Guestrin, 2016] Marco Tulio Ribeiro, S. S. and Guestrin, C. (2016). "why should i trust you?": Explaining the predictions of any classifier. *Computer Science, Machine Learning*.

[Moll et al., 2014] Moll, K., Kunze, S., Neuhoff, N., Bruder, J., Schulte-Körne, and G. (2014). Specific learning disorder: Prevalence and gender differences. *PLoS ONEs*, 9(7).

[Müller and R., 2004] Müller and R. (2004). *DRT 3: Diagnostischer Rechtschreibtest für 3. Klassen : In neuer Rechtschreibung*. Beltz.

[Naqa and Murphy, 2015] Naqa, I. E. and Murphy, M. J. (2015). What is machine learning? *Machine Learning in Radiation Oncology: Theory and Applications*, pages 3–11.

[Organization, 2010] Organization, W. H. (2010). *International statistical classification of diseases and related health problems*. World Health Organization, Geneva, Switserland, 10 edition.

[P. Cunningham and Delany, 2008] P. Cunningham, M. C. and Delany, S. J. (2008). Supervised learning. *Machine Learning Techniques for Multimedia: Case Studies on Organization and Retrieval*, pages 21–49.

[Power et al., 2014] Power, Mitra, J. D., A., Laumann, T. O., S., Z., A., Schlaggar, Petersen, B. L. ., and E., S. (2014). Methods to detect, characterize, and remove motion artifact in resting state fmri. *NeuroImage*, 84:320–341.

[P.Tamboer et al., 2016] P.Tamboer, H.C.M.Vorst, S.Ghebreab, and H.S.Scholte (2016). Machine learning and dyslexia: Classification of individual structural neuroimaging scans of students with and without dyslexia. *NeuroImage: Clinical*, 11:508–514.

[Ramus and Ahissar, 2012] Ramus, F. and Ahissar, M. (2012). Developmental dyslexia: the difficulties of interpreting poor performance, and the importance of normal performance. *Cognitive Neuropsychology*, 29(1):104–122.

[Richlan F, 2009] Richlan F, K. M. a. H. (2009). Functional abnormalities in the dyslexic brain: A quantitative meta-analysis of neuroimaging studies. *Hum Brain Mapp*.

[S.-I. Ao and Amouzegar, 2010] S.-I. Ao, B. B. R. and Amouzegar, M. (2010). *Machine learning and systems engineering*, volume 68. Springer Science Business Media.

[S. Mitra and Michailidis, 2008] S. Mitra, S. Datta, T. P. and Michailidis, G. (2008). *Introduction to machine learning and bioinformatics*. CRC Press.

[Sofia Zahiaa and Fernandez-Ruanovac, 2020] Sofia Zahiaa, Begonya Garcia-Zapiraina, I. S. and Fernandez-Ruanovac, B. (2020). Dyslexia detection using 3d convolutional neural networks and functional magnetic resonance imaging. *Computer Methods and Programs in Biomedicine*, 197.

[T. Hastie and Friedman, 2009] T. Hastie, R. T. and Friedman, J. (2009). *Unsupervised learning*. The elements of statistical learning.

[T. J. Cleophas and Cleophas-Allers, 2013] T. J. Cleophas, A. H. Z. and Cleophas-Allers, H. I. (2013). *Machine learning in medicine*, volume 9. Springer.

[Βούλγαρης ημήτριος, 2010] Βούλγαρης ημήτριος (2010). Ηυπόθεση φωνολογικού ελλείμματος στη δυσλεξία-Εκπαιδευτικές προεκτάσεις.

[Theodoridou et al., 2021] Theodoridou, D., Christodoulides, P., Zakopoulou, V., and Syrrou1, M. (2021). Developmental dyslexia: Environment matters. *Brain Science*, 11(6):782.

[Thomason, 2009] Thomason, M. E. (2009). Children in non-clinical functional magnetic resonance imaging (fmri) studies give the scan experience a “thumbs up”. *American Journal of Bioethics*, 9(1):25–27.

[Weiß and R., 2006] Weiß and R. (2006). *CFT 20-R: Grundintelligenztest Skala 2-Revision[Culture fair intelligence test-scale 2]*. Hogref.

[Wimmer et al., 2014] Wimmer, Mayringer, H., and H. (2014). *Salzburger Lese-screening für die Schulstufen 2-9. SLS 2-9*. Hogrefe.

[Yang, 2010] Yang, Z. R. (2010). *Machine learning approaches to bioinformatics*, volume 4. World scientific.

[Yolanda García Chimenoa and Fernandez-Ruanova, 2014] Yolanda García Chimenoa, Begonya García Zapiraina, I. S. P. and Fernandez-Ruanova, B. (2014). Automatic classification of dyslexic children by applying machine learning to fmri images. *Bio-Medical Materials and Engineering*.

[Yu and Tao, 2013] Yu, J. and Tao, D. (2013). *Modern machine learning techniques and their applications in cartoon animation research*, volume 4. John Wiley Sons.

[Zhao et al., 2016] Zhao, H., Chen, Y., ping Zhang, B., and xiang Zuo, P. (2016). Kiaa0319 gene polymorphisms are associated with developmental dyslexia in chinese uyghur children. *Nature Journal of Human Genetics*, 61:745–752.