Università degli Studi di Verona

DIPARTIMENTO DI INFORMATICA

Laurea in Bioinformatica

Approcci basati su Random Forest per la caratterizzazione della Sclerosi Multipla

Candidati:
Raniero Matteo
Mendo Lorenzo Antonio

Relatore:
Bicego Manuele

Anno Accademico 2018-2019

Indice

1 Introduzione **3**
- 1.1 La malattia . . . 3
- 1.2 Rete di connettività . . . 4
- 1.3 Classificazione e Analisi . . . 4

2 Background **5**
- 2.1 Rappresentazione dati nelle reti di connettività . . . 5
- 2.2 Classificazione . . . 5
  - 2.2.1 Tecniche di classificazione . . . 6
  - 2.2.2 Random Forest . . . 6
  - 2.2.3 Cross Validation . . . 8

3 Metodologia Proposta **11**
- 3.1 Perchè le random forest? . . . 11
- 3.2 Parametri delle RF . . . 11
- 3.3 Feature Selection nella RF . . . 12
- 3.4 Pipeline . . . 12

4 Risultati Sperimentali **13**
- 4.1 Il dataset . . . 13
- 4.2 Risultati . . . 14
  - 4.2.1 Analisi del software . . . 14
  - 4.2.2 Analisi del numero di alberi e features . . . 17
  - 4.2.3 Confonti tipologia di rete e rappresentazione . . . 17
- 4.3 Feature Selection . . . 18
- 4.4 Significativà statistica . . . 18

5 Conclusioni **23**

A Elenco Feature **25**

Sommario

Questo studio si colloca nell'ambito della Pattern Recognition, in particolare alla sua applicazione su dati biomedici. L'obiettivo dell'analisi è di caratterizzare un dataset di pazienti malati di sclerosi multipla, ognuno descritto con una rete di connettività ottenuta con tecniche di elaborazione d'immagini effettuate dopo l'acquisizione tramite risonanza magnetica. L'analisi si basa su una tecnica di classificazione chiamata Random Forest: si vuole evidenziare per quali feature applicate alle reti di connettività sia possibile discriminare tra pazienti sani e malati. Nel complesso è previsto l'impiego di 79 pazienti rappresentati secondo due diverse tipologie di rete di connettività e tre tipi di rappresentazione. Infine è stata utilizzata una tecnica per ricavare quelli che potrebbero essere i fasci neurali più importanti per caratterizzare la malattia; questi dati necessitano tuttavia di essere confrontati con analisi di tipo differente per essere confermati ed eventualmente assumere importanza medica.

CAPITOLO 1

Introduzione

1.1 La malattia

La sclerosi multipla (SM) è una malattia neurodegenerativa demielinizzante [1], cioè che provoca lesioni a carico del sistema nervoso centrale (SNC). Essa è una malattia autoimmune, una condizione che si verifica quando il sistema immunitario di un individuo attacca erroneamente tessuti e organi del proprio organismo scambiandolo per una minaccia esterna; infatti, nella SM, le difese immunitarie del paziente vanno ad attaccare le guaine mieliniche presenti attorno agli assoni, compromettendo quindi la corretta trasmissione dei segnali. La malattia può manifestarsi con una vastissima gamma di sintomi neurologici e può progredire fino alla disablità fisica e cognitiva. Esistono tre tipologie principali di pazienti malati di sclerosi multipla:

1. Pazienti affetti da sclerosi multipla a decorso recidivante-remittente (SM-RR); questa è la forma più comune di SM caratterizzata da episodi acuti alternati a periodi di parziale benessere.

2. Pazienti affetti da sclerosi multipla secondariamente progressiva (SM-SP); questa rappresenta l'evoluzione della precedente, caratterizzata da una disabilità persistente che progredisce gradualmente nel tempo.

3. L'ultima variante (SM-PP) è invece caratterizzata dal peggioramento delle funzioni neurologiche fin dai primi sintomi.

Generalmente dopo la comparsa della malattia, le aree colpite variano da persona a persona, ma si presentano sempre come delle zone indurite rispetto al resto del SNC; questa caratteristica risulta essere molto importante per l'identificazione tramite tecniche automatiche.

1.2 Rete di connettività

Recentemente, la sclerosi multipla è stata studiata applicando ad ogni paziente delle tecniche di analisi automatica per caratterizzare la malattia ed identificare particolari pattern con cui essa si presenta. Una possibilità è quella di calcolare per ogni paziente una rete di connettività, cioè una mappa che descrive le connessioni neurali del cervello, tramite tecniche di risonanza magnetica e di trattografia. Queste strutture permettono una descrizione in piccola scala, simile ad una mappa dettagliata dell'insieme di neuroni e sinapsi di tutto il sistema nervoso o solamente di una sua parte. In questo modo si identificano le varie zone del cervello ed è possibile ricavare quali fasci sono danneggiati o assenti a causa della malattia. Utilizzando un approccio di questo tipo nello studio potrebbe essere più semplice comprendere l'organizzazione e le interazioni dei neuroni all'interno del cervello, favorendo l'identificazione dei fasci neurali maggiormente coinvolti nella malattia.

1.3 Classificazione e Analisi

Una delle possibili analisi che si possono effettuare sulle reti di connettività consiste nel derivare un modello di classificazione basato sulla rete intesa come un grafo, con nodi e archi colleganti i vari punti del cervello analizzato. Per far questo si possono utilizzare tecniche di pattern recognition, una disciplina che include diverse tecniche per identificare pattern all'interno di dati grezzi al fine di identificarne la classificazione. In particolare, dato un campione di training relativo a pazienti affetti da SM e non, il nostro scopo diventa quello di imparare le regole di predizione della classe di una data rete ed interpretarle nel modo migliore possibile. Indagando sul risultato di questa operazione di classificazione, si possono ottenere diverse informazioni legate ai pazienti analizzati, come ad esempio i possibili fasci implicati nella malattia.

CAPITOLO 2

Background

In questa sezione verranno introdotte le informazioni di base necessarie per comprendere l'analisi proposta. In particolare verrà descritta la struttura su cui sono organizzati i dati e la tecnica di classificazione usata per l'analisi dei dati stessi.

2.1 Rappresentazione dati nelle reti di connettività

La tecnica di risonanza usata in queste analisi è la cosiddetta DTI, *Diffusion Tensor Imaging* [2]; essa sfrutta il processo di diffusione di molecole d'acqua in vivo e in modo non invasivo. Secondo questo processo è possibile visualizzare le interazioni tra gli ostacoli, come i fasci di materia bianca, ricostruendo in un'immagine 3D la struttura del cervello. La creazione di una rete di connettività avviene attraverso diverse fasi seguite da *trattografia*; questa è una tecnica di modellazione 3D che ha lo scopo di rappresentare visivamente i tratti neurali, elaborando i dati raccolti dalla DTI. Una *rete di connettività* di un cervello umano può essere intesa come una mappa delle connessioni neurali: più nel dettaglio essa rappresenta un grafo, dove i nodi rappresentano le sinapsi mentre gli archi indicano i fasci caratteristici di materia bianca. Da questo grafo si possono estrarre diverse caratteristiche rappresentanti i pazienti in analisi; su questi dati possono essere applicate le tecniche di classificazione usate in quest'analisi.

2.2 Classificazione

La *classificazione* è una delle problematiche più studiate nel contesto della *pattern recognition* (PR). In particolare, con questo termine si vuole indicare un processo per l'analisi di dati che ha come obiettivo la caratterizzazione in classi/categorie [3]; nello specifico, classificare significa assegnare un oggetto ad una data classe di appartenenza, sulla base di un modello costruito a partire da caratteristiche estratte dall'oggetto (features). Per costruire un classificatore quindi, occorre definire un modello, che viene poi istanziato tramite un procedimento di addestramento basato su esempi presi come riferimento, il cui insieme è

definito training set. Questo classificatore verrà successivamente testato per verificarne la bontà e la capacità di generalizzazione, con dati non presenti nel dataset di training, ma in quello di testing. Dopo aver effettuato la classificazione sul testing set, possiamo identificare un errore che indica l'efficacia del classificatore.

2.2.1 Tecniche di classificazione

Esistono svariate tipologie di classificatori ognuno basato su assunzioni e tecniche differenti, ma in generale essi possono essere raggruppati in due grandi categorie: i classificatori generativi ed i classificatori discriminativi. Questa distinzione si basa sul modo in cui i classificatori modellano il problema di classificare. I classificatori *generativi* mirano a modellare le probabilità a posteriori delle classi (ottenuta considerando sia la probabilità a priori che quella condizionale) per descrivere al meglio come questi oggetti si distribuiscono; ogni oggetto viene poi assegnato ad una classe tramite la regola di decisione di Bayes. Purtroppo, le probabilità non sono conosciute a priori e devono essere stimate dai dati di training e sulla base di queste stime si distinguono altre due grandi famiglie di classificatori generativi; quelli che effettuano *stime parametriche* e quelli che effettuano *stime non parametriche*. Nelle prime, si assume di conoscere la forma della densità di probabilità e quindi si calcolano direttamente i parametri della distribuzione. Nelle seconde, invece, non si fa nessuna assunzione ma la probabilità viene stimata direttamente dai dati.
I classificatori *discriminativi*, invece, non cercano di descrivere come gli oggetti si distribuiscono ma puntano a trovare un confine di separazione, cioè un criterio col quale poter assegnare alle classi i vari oggetti senza considerarne la distribuzione. Questi operano principalmente stimando direttamente la probabilità a posteriori o affidandosi a concetti geometrici. Le tecniche utilizzate in questo lavoro appartengono a questa seconda classe e sono definite con il termine *Random Forest* (RF).

2.2.2 Random Forest

Le RF introdotte da Leo Breiman [4], sono un metodo di classificazione definito da un insieme di *alberi di decisione* binari, detto *ensemble*. In particolare una RF combina la predizione di un insieme di alberi di decisione, definendo la classe con un maggior numero di voti come predizione della foresta. Per comprendere al meglio le RF è quindi fondamentale comprendere il funzionamento dei sottomodelli usati, gli alberi di decisione basati su di un procedimento definito CART; essi costituiscono un modo molto efficace per classificare degli oggetti in un numero finito di classi. Ad ogni nodo dell'albero (split) corrisponderà una domanda su una particolare feature estratta dai dati in esame. Da ogni nodo si deciderà se proseguire a destra o a sinistra sulla base della risposta data, fino ad arrivare alle foglie. Questi alberi vengono costruiti suddividendo ripetutamente i dati del problema sulla base di caratteristiche precise. In questo modo si otterranno gruppi al loro interno omogenei e quanto più differenziati.
In figura 2.1 è rappresentato un esempio di albero di decisione; Y' e Y" indicano le classi e y l'oggetto. Il controllo sulle feature avviene su ogni nodo dell'albero. L'idea del CART sta nel valutare secondo un approccio greedy, quali sono le caratteristiche migliori per ogni split, considerando una funzione di costo che si deriva dall'indice di eterogeneità di Gini. L'unione di più alberi di classificazione tra loro porta alla formazione di una Random Forest. Una struttura di questo tipo viene definita per risolvere i problemi di varianza e di overfitting

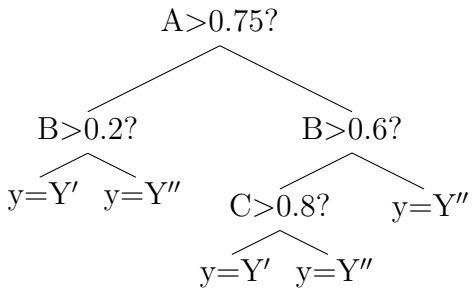

Figura 2.1: Esempio di Decision Tree

derivanti dall'applicazione di singoli alberi di classificazione su dati, ottenendo risultati più precisi combinati su diverse features e su diversi livelli di importanza. Questo tipo di applicazione segue il concetto di *bagging*, ossia aggregazione di tipo bootstrap di diversi modelli, per ridurre la varianza delle predizioni di un singolo modello. Per far in modo di evitare l'overfitting, ogni componente viene allenato su un campione di *bootstrap*. Questo termine indica un campione della stessa dimensione del dataset originale costruito con il replacement. Le Random Forest possono essere istanziate in molti modi possibili; ad esempio, è possibile modificare il modo in cui gli alberi binari vengono uniti tra loro e il modo in cui gli split vengono definiti.

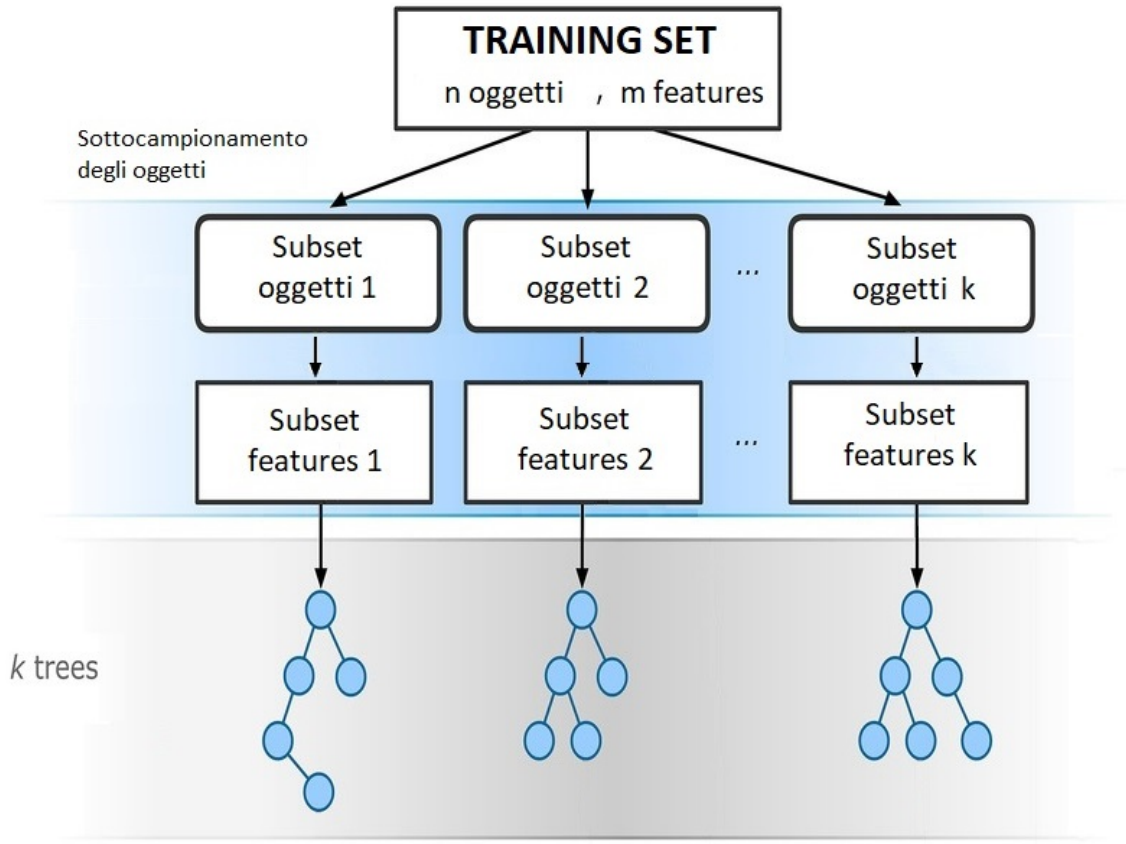

Figura 2.2: Training

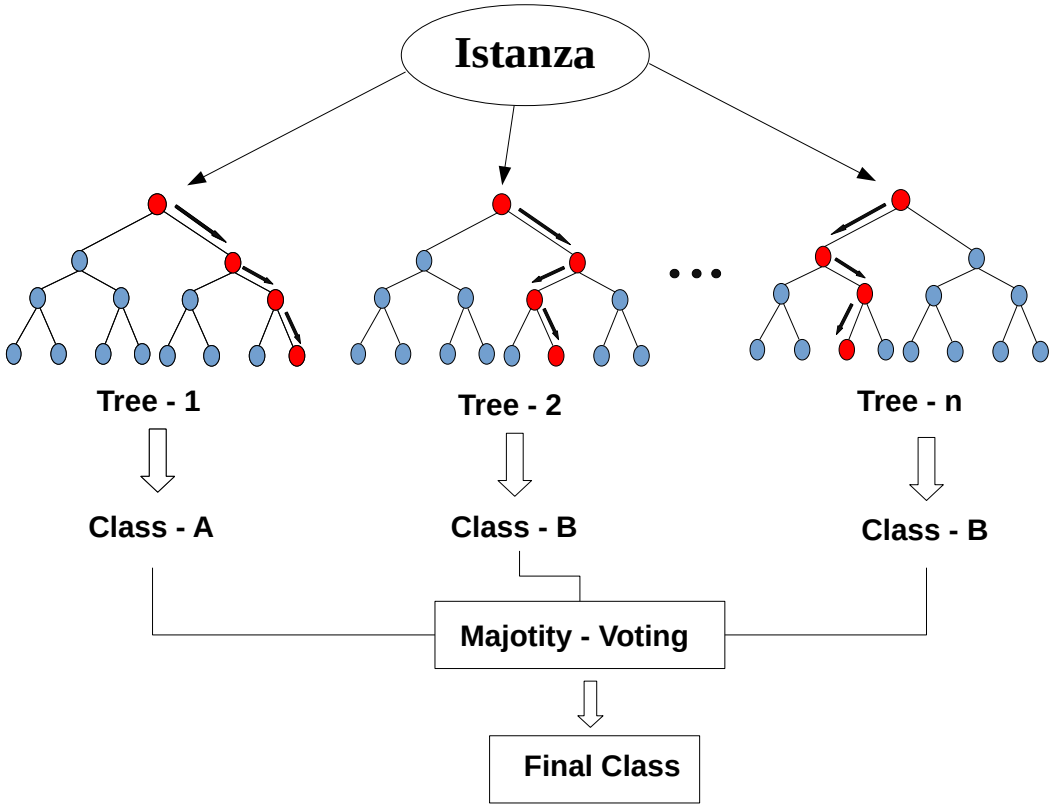

Figura 2.3: Testing

2.2.3 Cross Validation

Come detto in precedenza, per valutare correttamente un classificatore occorre testarlo su dati che non ha mai utilizzato in fase di addestramento. Per far questo si ricorre a tecniche note con il nome di *cross validation* che permettono di derivare training e testing set dal dataset originale. Nella sua versione più semplice (Holdout crossvalidation), il dataset di partenza viene suddiviso in due parti uguali:

- *Training set*: viene utilizzato dal classificatore per costruire il modello con cui discriminare sugli oggetti nelle varie classi (Addestramento del modello).
- *Testing set*: l'insieme di elementi non utilizzati nella fase di training, che vengono utilizzati per verificare l'efficacia del classificatore in esame.

Ovviamente più questi due dataset sono grandi, migliore sarà la capacità di classificazione e migliore sarà la stima dell'errore. Spesso però accade che il dataset di partenza non sia sufficientemente grande e spezzarlo in due parti uguali non sia la scelta migliore. Si ricorre quindi ad altre varianti di cross validation:

- *Averaged Holdout*: per evitare che il risultato sia dipendente dalla partizione casuale fatta con il metodo holdout, si mediano i risultati calcolati su più partizioni holdout. Queste partizioni sono costruite casualmente oppure in modo esaustivo.

- *Leave One-Out*: si seleziona un solo elemento del dataset che verrà utilizzato come dataset di test ed i restanti verranno utilizzati cone dataset di training. Una volta testato il classificatore, si sceglie un altro elemento diverso dal precedente e si ripete l'operazione per tutti gli elementi del dataset. Dopodiché, si mediano i risultati ottenuti.

- *Leave K-Out*: generalizzazione della tecnica leave one-out; l'idea è quella di suddividere il dataset in N segmenti ciascuno formato da k elementi distinti e casuali. I primi k elementi vengono utilizzati per il dataset di testing ed i rimanenti per il training. Questa operazione viene effettuata N volte variando a turno il segmento del testing set e la capacità di generalizzazione viene mediata tra gli N risultati.

In questo lavoro abbiamo utilizzato il leave one-out (LOO): questa è la tecnica più utilizzata per dataset piccoli, in quanto massimizza sia la dimensione del training set che che quella del testing set.

CAPITOLO 3

Metodologia Proposta

In questo studio abbiamo applicato le RF al problema della classificazione della SM attraverso le reti di connettività. Prima di descrivere le varie parametrizzazioni utilizzate e la pipeline di analisi, presentiamo alcune considerazioni sul perchè abbiamo utilizzato le RF.

3.1 Perchè le random forest?

Abbiamo scelto di studiare le RF per diversi motivi: innanzitutto, le Random Forest rappresentano un classificatore molto accurato, la cui efficacia è stata dimostrata in diverse applicazioni. Secondo, ben si prestano a dati altamente dimensionali (come questi); infatti, il criterio di randomicità con cui vengono scelte le features per il dataset di training assicura che tutte le feature siano incluse in modo randomico dal modello e che nessuna feature venga esclusa durante l'addestramento, pur costruendo ogni albero su un insieme potenzialmente ridotto di features. Questo perchè il sottoinsieme di feature coinvolte nell'addestramento ad ogni ciclo è diverso, e quindi si ha una maggiore variabilità per la classificazione. Infine, le RF permettono di determinare quali sono le feature più importanti per la classificazione. Questo risulta essere fondamentale in ambito medico per capire quale caratteristica derivata dai dati è più influente nel classificare i pazienti in esame.

3.2 Parametri delle RF

Il primo parametro importante delle RF è il numero d'alberi che fanno parte della foresta; scegliere il numero ottimale di alberi è chiaramente un problema: un numero troppo basso di alberi riduce notevolmente i tempi di allenamento del modello, ma porta ad una classificazione poco precisa. Allo stesso tempo, un numero elevato di alberi, oltre ad impiegare più tempo per l'allenamento del modello, non assicura una classificazione migliore; anzi, spesso, il risultato non varia più all'aumentare del numero di alberi usati. In questo studio abbiamo testato le RF con un numero variabile di alberi, al fine di trovare il valore ottimale tra velocità di allenamento e capacità di classificazione.

Il secondo parametro fondamentale riguarda il numero di feature considerate nella costruzione dell'albero. Nel nostro studio abbiamo considerato due opzioni: una che considera tutte le features e l'altra che considera la metà delle features. Considerando un numero troppo basso la classificazione può risultare non efficiente, in quanto l'informazione fornita non è sufficiente, mentre con un numero troppo elevato posso avere problemi di rumore. Come detto sopra, i sottoinsiemi di features considerati per la costruzione degli alberi sono sempre diversi, quindi ho più variabilità e meno perdita di informazione (includo tutte le features), il che comporta generalmente una classificazione più efficiente.

3.3 Feature Selection nella RF

Una delle caratteristiche più importanti delle RF è la possibilità di determinare quali sono le features più importanti per il task in questione. L'approccio per l'estrazione dei predittori migliori si basa sulla tecnica Out-Of-Bag (OOB), utilizzabile già durante la fase di training. L'OOB in particolare utilizza gli elementi non impiegati per creare ed allenare il modello per testarne subito la capacità di classificazione. L'approccio per l'estrazione dei predittori migliori caratterizzanti il modello è caratterizzato da questa tecnica. Nelle RF già durante la fase di training, è possibile stimare l'importanza delle features. Gli elementi OOB testati forniscono le informazioni che ci servono: se questi risultano corretti allora le features utlizzate per la classificazione ricevono un punteggio buono, altrimenti no. In questo modo, possiamo determinare le features più importanti come quelle che hanno totalizzato un punteggio maggiore. Andando a mediare tutti i punteggi che ci vengono ritornati dal protocollo LOO, per il numero di pazienti in analisi, si può ottenere una classifica più robusta dei predittori migliori del modello. Si può quindi pensare di effettuare una nuova classificazione, usando solo le feature migliori con lo scopo di ottenere un modello più efficiente.

3.4 Pipeline

La pipeline qui descritta rappresenta l'intero procedimento effettuato per l'analisi:

1. Rappresentazione Anatomica di un cervello umano con MR.
2. Rete di connettività ricavata attraverso un processo di Imaging e trattografia.
3. Rappresentazione. Estrazione delle features dal connettoma.
4. Per ogni suddivisione training/testing data dal protocollo LOO addestramento di una RF con il training set e validazione.
5. Estrazione delle features migliori e nuova classificazione LOO con le feature selezionate.
6. Analisi della significativià statistica dei risultati del modello costruito.

CAPITOLO 4

Risultati Sperimentali

Gli esperimenti effettuati all'interno dello studio sono stati realizzati con due diversi tool in Matlab: Prtools e Treebagger. Inizialmente è stato utilizzato Prtools, un tool che garantisce semplicità di utilizzo anche se risulta essere non troppo flessibile; per questo motivo i primi esperimenti sono quindi stati effettuati tramite una semplice random forest con suddivisione del dataset di training in stile LOO. Successivamente lo studio è stato continuato con Treebagger, grazie al quale è stato possibile introdurre alcune parametrizzazioni più specifiche e ricavare l'importanza dei predittori (Feature Selection). Con entrambi i software utilizzati, l'analisi ha previsto il test di varie combinazioni di parametri, come il numero di alberi e la suddivisione delle features. Un'ultima nota: in tutti gli esperimenti è stato applicato il paired t-test per misurare la significatività statistica dei confonti.

4.1 Il dataset

Il dataset utilizzato nello studio, è ricavato dalle reti di connettività di 79 soggetti, di cui 24 sani mentre 55 che presentano diverse forme di SM:

- 13 RR affetti da SM di tipo Recidivante Remittente
- 20 SP affetti da SM di tipo Secondario Progressivo
- 22 PP affetti da SM di tipo Primario Progressivo

Le reti sono ottenute dall'analisi di trattografia sulle immagini 3D ricavate dalla DTI. Le tecniche di elaborazione d'immagine applicate e la costruzione delle reti è stata effettuata in un altro studio dal professore Alessandro Daducci, il quale ha elaborato un modello, il COMMIT, per far risaltare alcune caratteristiche precise dalle reti di connettività [5]. Nella rete ci sono 85 regioni cerebrali, rappresentate come nodi del grafo rappresentante il cervello, le cui connessioni neurali sono rappresentate dagli archi.
In generale, da un grafo è possibile ricavare diverse proprietà come il peso di un nodo o la distanza tra due diversi nodi utilizzabili come features, che possono essere suddivise in

feature locali e globali. Le prime indicano una misura legata ad ogni nodo (regione) della rete analizzata, mentre le seconde rispecchiano le proprietà di una rete nel suo complesso (una per ogni rete). Nel dataset analizzato sono presenti due tipologie diverse di reti ognuna delle quali rappresentata tramite insiemi di feature. Due delle tre diverse modalità di rappresentazione consistono in 85 features (una per ogni regione) corrispondenti alle feature locali, l'altra ne presenta 3570 essendo derivata da un'analisi più specifica che include tutte le connessioni neurali della rete. Le tre rappresentazioni sono LocalEff (85), NodeStrength (85),WholeNet (3570), mentre le due reti analizzate sono XbyLen_MeanL e Raw.

- Raw: Dati grezzi riguardanti le connessioni tra nodi, numero di fibre non filtrati da COMMIT.
- XbyLen_meanL: Feature elaborate dal modello COMMIT per far risaltare alcune proprietà specifiche della rete.

	Local_Eff	Node_Strenght	WholeNet
Raw	79*85	79*85	79*3570
XByLenMean_L	79*85	79*85	79*3570

Tabella 4.1: Struttura Dataset

4.2 Risultati

Date le diverse tipologie di reti e di rappresentazioni, abbiamo verificato l'accuratezza LOO delle RF al variare del numero di alberi (100,200,300,400,500) e del numero di features utilizzate nella costruzione del modello (metà o tutte). Come prima analisi abbiamo confrontato i due software utilizzati per l'analisi: Prtools e Treebagger. In secondo luogo abbiamo confrontato risultati per le diverse reti, rappresentazioni, e parametrizzazioni utilizzando il solo software Treebagger. Infine, vengono presentati i risultati della feature selection e della predizione dell'importanza dei predittori del modello.

4.2.1 Analisi del software

La prima analisi è stata condotta sui due software utilizzati (Prtools e Treebagger) in modo da comprenderne al meglio le caratteristiche. Per confrontarli è considerata l'accuratezza ottenuta dalla classificazione fatta dai due software per gli stessi parametri (stesse reti, numero di features, numero di alberi) assegnando un punto al software con l'accuratezza maggiore e zero in caso di pareggio; considerando tutti i test possibili si identifica il software migliore. Dai risultati ottenuti, Treebagger ha classificato con una accuratezza maggiore di Prtools 33 volte su 60 confronti, mentre Prtools si è dimostrato migliore di Treebagger 14 volte su 60 confronti. Nonostante Treebagger si presenti notevolente superiore a Prtools in questa analisi, è importante notare che ciò è dovuto alle pessime prestazioni di Prtools

Rete	**Rappr.**	**NumAlberi**	**Features**	**Accuracy**
XbyLenmeanL	**LocalEff**	100	42	0.797
		100	85	0.823
		200	42	0.835
		200	85	0.823
		300	42	0.835
		300	85	0.810
		400	42	0.835
		400	85	0.835
		500	42	0.835
		500	85	0.797
	NodeStreght	100	42	0.772
		100	85	0.785
		200	42	0.772
		200	85	0.734
		300	42	0.785
		300	85	0.759
		400	42	0.772
		400	85	0.759
		500	42	0.772
		500	85	0.734
	WholeNet	100	1785	0.861
		100	3570	0.785
		200	1785	0.823
		200	3570	0.785
		300	1785	0.835
		300	3570	0.823
		400	1785	0.861
		400	3570	0.810
		500	1785	0.848
		500	3570	0.823

Tabella 4.2: XbyLen meanL

Rete	Rappr.	NumAlberi	Features	Accuracy
Raw	**LocalEff**	100	42	0.709
		100	85	0.696
		200	42	0.696
		200	85	0.696
		300	42	0.722
		300	85	0.709
		400	42	0.722
		400	85	0.709
		500	42	0.696
		500	85	0.696
	NodeStrenght	100	42	0.747
		100	85	0.759
		200	42	0.784
		200	85	0.759
		300	42	0.747
		300	85	0.759
		400	42	0.772
		400	85	0.759
		500	42	0.784
		500	85	0.772
	WholeNet	100	1785	0.861
		100	3570	0.823
		200	1785	0.823
		200	3570	0.797
		300	1785	0.823
		300	3570	0.797
		400	1785	0.848
		400	3570	0.823
		500	1785	0.810
		500	3570	0.823

Tabella 4.3: Raw

sulla rete WholeNet, che contiene un numero di features superiore alle 3000. Infatti, su questo tipo di rete l'errore di classificazione rimane fissa ad un valore di 0.696 nonostante i parametri considerati varino notevolmente. Si ipotizza quindi che Prtools non sia in grado di gestire al meglio dataset di grandi dimensioni per creare modelli efficienti basati sulle random forest, probabilmente per qualche dettaglio implementativo non efficiente. Se i confronti sulla WholeNet non vengono considerati, si ottiene che Treebagger ha classificato con una accuratezza maggiore di Prtools 13 volte su 40 confronti e Prtools si è dimostrato migliore di Treebagger 14 volte su 40 confronti. Nonostante Prtools sia comunque valido per analisi basilari, è abbastanza limitato, quindi si è scelto di continuare le analisi con Treebagger, che permette anche di settare più parametri e di ricavare più informazioni. Nella tabella 4.2 sono rappresentati i risultati completi ottenuti con Treebagger; nella quarta colonna è specificata l'accuratezza del modello su una scala da 0 a 1. Come prima osservazione generale, risulta evidente una maggiore accuratezza della classificazione nel caso della rappresentazione WholeNet; si ha una correttezza massima dell'86% con 11 errori su 79. Un'altra osservazione riguarda la maggior efficacia della rete XByLen_meanL rispetto a Raw con un errore medio dell' 80.4% nel primo caso e del 76.4% nel secondo.

4.2.2 Analisi del numero di alberi e features

Un altro aspetto rilevante è il confronto tra risultati sulla base del numero di alberi ed il numero di features con cui il modello è stato addestrato. Guardando le tabelle 4.2 e 4.3, non è possibile estrarre una forte relazione tra l'errore di classificazione e il numero di alberi; più precisamente, sembra che il numero di alberi non sia così fondamentale. Una possibile spiegazione è che il range di alberi utilizzato per l'analisi non sia sufficientemente ampio per poter notare questo fenomeno.
Sucessivamente, si è analizzata la variazione dell'errore di classificazione al variare del numero di features usate nella costruzione del modello (metà o tutte). Qui si notano due comportamenti diversi in base al dataset usato; nei dataset di piccole dimensioni (quelli relativi a LocalEff e NodeStrenght) non si nota una grande differenza tra i due casi. Invece, nei dataset di grandi dimenensioni, come la WholeNet, si ha una visibile diminuzione dell'errore di classificazione utilizzando metà delle features; questo conferma le intuizioni del capitolo precedente e in generale le idee alla base delle RF, dato che per allenare il dataset viene preso un sottoinsieme delle features totali e questo permette di avere alberi diversi e quindi una migliore generalizzazione.

4.2.3 Confonti tipologia di rete e rappresentazione

Utilizzando lo stesso approccio introdotto precedentemente nel confronto dei software, abbiamo analizzato il diverso comportamento delle reti XbyLen_MeanL e Raw. Su un totale di 30 confronti effettuati, si sono verificati 6 pareggi; la rete XbyLen_MeanL risulta aver classificato con una accuratezza maggiore di Raw 17 volte su 30 confronti, mentre Raw si è dimostrato migliore in totale di 7 confronti su 30. Il fatto che XbyLen_MeanL sia più descrittiva conferma che il filtraggio COMMIT è estremamente utile per migliorare le caratteristiche della rete di connettività.
Allo stesso modo sono stati analizzati i comportamenti delle diverse rappresentazioni in esame. Nella rete Raw, WholeNet risulta essere la rappresentazione migliore dato che vince in

Rete	Rappr.1	Punteggio	Rappr.2
XbyLenmeanL	LocalEff	10-0	NodeStrenght
	LocalEff	4-5	WholeNet
	NodeStrenght	0-9	WholeNet
Raw	LocalEff	0-9	NodeStrenght
	LocalEff	0-10	WholeNet
	NodeStrenght	0-10	WholeNet

Tabella 4.4: Confronto Rappresentazioni

entrambi i confronti con LocalEff e NodeStrenght per 10-0. In XbyLen_MeanL, NodeStrenght si è rilevata essere la peggiore ed invece WholeNet risulta essere leggermente migliore rispetto a LocalEff, anche se in misura meno significativa rispetto a Raw (5-4). Visti questi risultati, per la rete XbyLen_MeanL si può concludere che per una prima analisi può essere sufficiente l'utilizzo della rappresentazione LocalEff, dato che il numero ridotto di features implica un tempo di classificazione minore e si ottengono comunque buoni risultati. Per analisi più approfondite è invece opportuno utilizzare WholeNet in quanto offre complessivamente risultati migliori ma a scapito di un tempo di training maggiore.

4.3 Feature Selection

In questa sezione riportiamo i risultati dell'analisi volta ad identificare i predittori migliori per ogni coppia Rete-Rappresentazione, ovvero quelle features più influenti nella classificazione dei pazienti. I grafici riportati nelle figure 4.1-4.6 mostrano i predittori delle varie reti e rappresentazioni in esame ordinati per importanza. I nomi delle aree evidenziati in rosso sono associate ad aree della corteccia motoria (che si suppone sia la più coinvolta nella malattia). Si può notare che molti dei predittori che hanno un ruolo fondamentale nel discriminare pazienti sani da pazienti affetti da SM, sono associati ad aree della corteccia motoria, fenomeno ben visibile sulla rete WholeNet.
Usando questi predittori migliori è stata effettuata un'ulteriore classificazione per vedere l'accuratezza delle feature stimate, in particolare abbiamo selezionato le migliori 25%,40%,75%, 85% features della rapresentazione WholeNet. In questo modo si può arrivare fino ad un'accuratezza del 94.9% nel caso del 40% con la rete XbyLen_MeanL (figura 4.5).

4.4 Significativà statistica

Per verificare la correttezza dei confronti effettuati a livello di analisi, occorre in generale utilizzare un test statistico come il paired t-test. Questo test statistico, date due popolazioni di campioni appaiati (cioè relativi allo stesso esperimento), testa l'ipotesi che la differenza tra i campioni venga da una gaussiana a media nulla, ovvero che le due distribuzioni siano statisticamente significative. Il valore ritornato dal test permette il rigetto o il non rigetto dell'ipotesi nulla, ossia l'uguaglianza o la diversità delle medie dei dati. Nel caso del confronto

tra le reti il risultato del paired t-test ci permette di rigettare l'ipotesi nulla ottenendo così la diversità dei dati in analisi (non essendo uguali, sono per forza diversi). La stessa analisi è stata effettuata sulle rappresentazioni e possiamo anche qui rigettare l'ipotesi nulla, tranne in un caso specifico: nel confronto tra le rappresentazioni NodeStrenght e LocalEff. Osservando i risultati, si nota che nella rete Raw, LocalEff è più efficiente di NodeStrenght mentre in COMMIT i ruoli si invertono. Questo può spiegare il risultato ottenuto dal paired t-test; infatti effettuando un ulteriore paired t-test confrondanto LocalEff e NodeStrenght solamente sulla singola rete, il risultato di questo permette il rigetto dell'ipotesi nulla e conferma quindi la diversità tra le rappresentazioni analizzate.

Rappr.	Soglia	Alberi	Features	Raw	XbyLenMeanL
WholeNet	**25**	100	metà	0.886	0.899
		100	tutte	0.886	0.899
		200	metà	0.873	0.911
		200	tutte	0.886	0.886
		300	metà	0.886	0.911
		300	tutte	0.886	0.899
		400	metà	0.886	0.873
		400	tutte	0.886	0.899
		500	metà	0.873	0.911
		500	tutte	0.886	0.899
	40	100	metà	0.873	0.949
		100	tutte	0.861	0.899
		200	metà	0.911	0.924
		200	tutte	0.873	0.899
		300	metà	0.911	0.911
		300	tutte	0.873	0.899
		400	metà	0.911	0.924
		400	tutte	0.873	0.899
		500	metà	0.899	0.937
		500	tutte	0.873	0.899
	75	100	metà	0.861	0.899
		100	tutte	0.873	0.873
		200	metà	0.886	0.924
		200	tutte	0.873	0.886
		300	metà	0.873	0.924
		300	tutte	0.886	0.899
		400	metà	0.886	0.924
		400	tutte	0.873	0.886
		500	metà	0.899	0.924
		500	tutte	0.873	0.886
	85	100	metà	0.861	0.911
		100	tutte	0.873	0.873
		200	metà	0.886	0.886
		200	tutte	0.848	0.886
		300	metà	0.873	0.899
		300	tutte	0.861	0.886
		400	metà	0.873	0.911
		400	tutte	0.861	0.886
		500	metà	0.873	0.899
		500	tutte	0.873	0.886

Tabella 4.5: Feature Selection

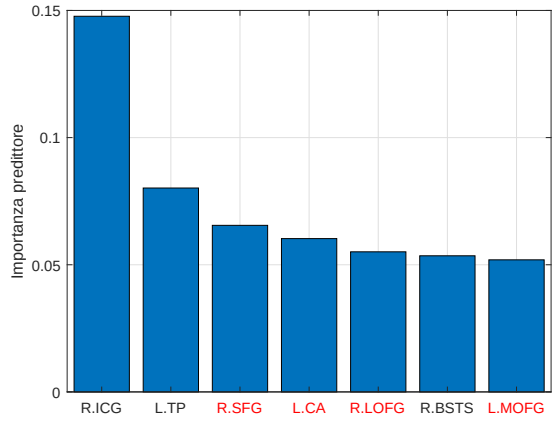

Figura 4.1: Raw-LocalEff

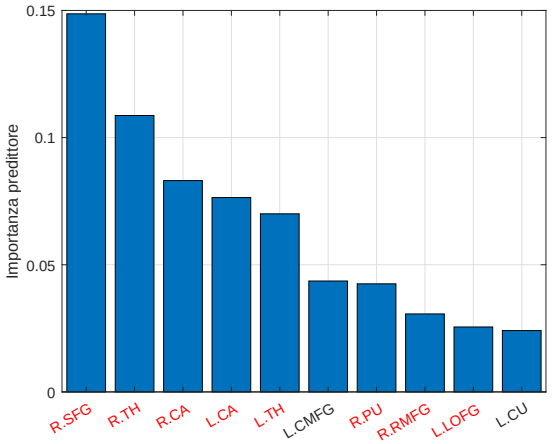

Figura 4.2: XbyLenmeanL-LocalEff

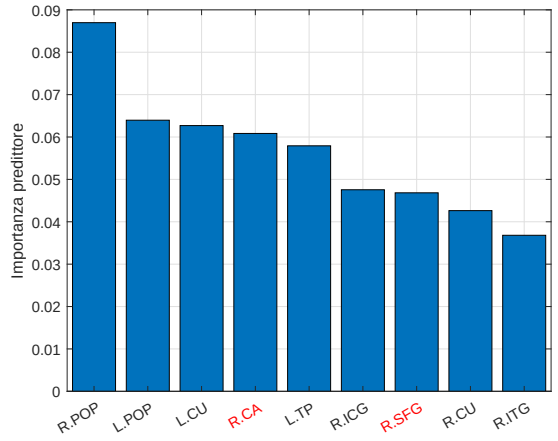

Figura 4.3: Raw-NodeStrength

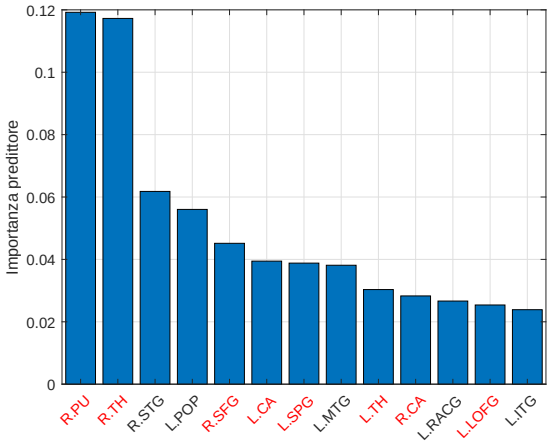

Figura 4.4: XbyLenMeanL-NodeStrength

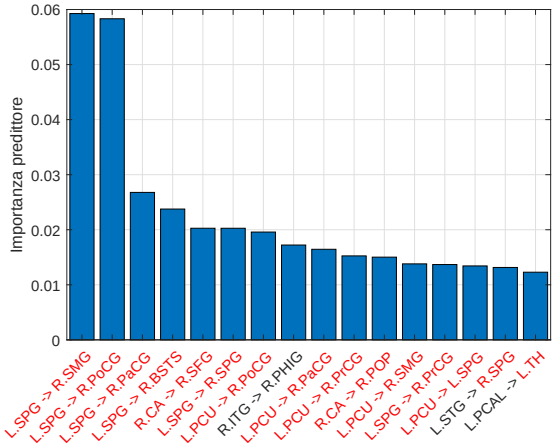

Figura 4.5: Raw-WholeNet

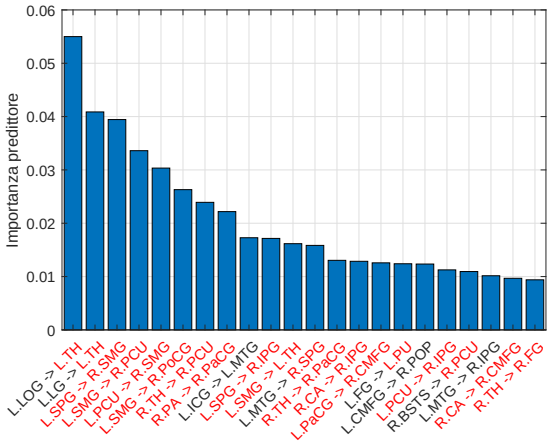

Figura 4.6: XbyLenMeanL-WholeNet

CAPITOLO 5

Conclusioni

In questo studio abbiamo investigato l'efficienza delle RF per la classificazione della SM a partire da reti di connettività. In primo luogo si è verificata l'efficacia delle RF come metodo di classificazione, che risultano essere metodi veloci efficaci e parametrizzabili in diversi modi. Come osservazione generale abbiamo notato che in un'analisi di questo tipo non è semplice ricavare il numero di parametri corretto per una classificazione ottimale. Inoltre, notiamo che sarebbe utile effettuare ulteriori classificazioni dello stesso tipo per certificare il risultato ottenuto ed eliminare possibili differenze di risultato (seppur minime nelle RF) a livello di accuratezza del modello. Per quanto riguarda l'applicazione specifica è stata verificata l'efficacia della rappresentazione WholeNet, che porta ad una classificazione migliore. Con questa rappresentazione otteniamo l'errore di classificazione più basso; in più notiamo che i predittori migliori del modello corrispondono spesso a fasci neurali della corteccia motoria. Quest'ipotesi necessita di essere verificata ulteriormente attraverso ulteriori classificazioni con metodi differenti. A livello di rappresentazione si vuole far notare come LocalEff per la rete XbyLen_MeanL sia efficace per una prima classificazione, cosa non vera per Raw (dove è più affidabile WholeNet). Nel complesso come risultato dell'analisi si è verificata la miglior efficacia dei dati filtrati da COMMIT; questo certifica che il lavoro di estrapolazione dei dati fatto a partire dalle risonanze magnetiche dei pazienti è stato efficace e porta effettivamente vantaggio.

APPENDICE A

Elenco Feature

#	acronym	structure_name	#	acronym	structure_name
1	**L.BSTS**	ctx-lh-bankssts	41	**L.AM**	Left-Amygdala
2	**L.CACG**	ctx-lh-caudalanteriorcingulate	42	**L.AC**	Left-Accumbens-area
3	**L.CMFG**	ctx-lh-caudalmiddlefrontal	43	**R.TH**	Right-Thalamus-Proper
4	**L.CU**	ctx-lh-cuneus	44	**R.CA**	Right-Caudate
5	**L.EC**	ctx-lh-entorhinal	45	**R.PU**	Right-Putamen
6	**L.FG**	ctx-lh-fusiform	46	**R.PA**	Right-Pallidum
7	**L.IPG**	ctx-lh-inferiorparietal	47	**R.HI**	Right-Hippocampus
8	**L.ITG**	ctx-lh-inferiortemporal	48	**R.AM**	Right-Amygdala
9	**L.ICG**	ctx-lh-isthmuscingulate	49	**R.AC**	Right-Accumbens-area
10	**L.LOG**	ctx-lh-lateraloccipital	50	**R.BSTS**	ctx-rh-bankssts
11	**L.LOFG**	ctx-lh-lateralorbitofrontal	51	**R.CACG**	ctx-rh-caudalanteriorcingulate
12	**L.LG**	ctx-lh-lingual	52	**R.CMFG**	ctx-rh-caudalmiddlefrontal
13	**L.MOFG**	ctx-lh-medialorbitofrontal	53	**R.CU**	ctx-rh-cuneus
14	**L.MTG**	ctx-lh-middletemporal	54	**R.EC**	ctx-rh-entorhinal
15	**L.PHIG**	ctx-lh-parahippocampal	55	**R.FG**	ctx-rh-fusiform
16	**L.PaCG**	ctx-lh-paracentral	56	**R.IPG**	ctx-rh-inferiorparietal
17	**L.POP**	ctx-lh-parsopercularis	57	**R.ITG**	ctx-rh-inferiortemporal
18	**L.POR**	ctx-lh-parsorbitalis	58	**R.ICG**	ctx-rh-isthmuscingulate
19	**L.PTR**	ctx-lh-parstriangularis	59	**R.LOG**	ctx-rh-lateraloccipital
20	**L.PCAL**	ctx-lh-pericalcarine	60	**R.LOFG**	ctx-rh-lateralorbitofrontal
21	**L.PoCG**	ctx-lh-postcentral	61	**R.LG**	ctx-rh-lingual
22	**L.PCG**	ctx-lh-posteriorcingulate	62	**R.MOFG**	ctx-rh-medialorbitofrontal
23	**L.PrCG**	ctx-lh-precentral	63	**R.MTG**	ctx-rh-middletemporal
24	**L.PCU**	ctx-lh-precuneus	64	**R.PHIG**	ctx-rh-parahippocampal
25	**L.RACG**	ctx-lh-rostralanteriorcingulate	65	**R.PaCG**	ctx-rh-paracentral
26	**L.RMFG**	ctx-lh-rostralmiddlefrontal	66	**R.POP**	ctx-rh-parsopercularis
27	**L.SFG**	ctx-lh-superiorfrontal	67	**R.POR**	ctx-rh-parsorbitalis
28	**L.SPG**	ctx-lh-superiorparietal	68	**R.PTR**	ctx-rh-parstriangularis
29	**L.STG**	ctx-lh-superiortemporal	69	**R.PCAL**	ctx-rh-pericalcarine
30	**L.SMG**	ctx-lh-supramarginal	70	**R.PoCG**	ctx-rh-postcentral
31	**L.FP**	ctx-lh-frontalpole	71	**R.PCG**	ctx-rh-posteriorcingulate
32	**L.TP**	ctx-lh-temporalpole	72	**R.PrCG**	ctx-rh-precentral
33	**L.TTG**	ctx-lh-transversetemporal	73	**R.PCU**	ctx-rh-precuneus
34	**L.IN**	ctx-lh-insula	74	**R.RACG**	ctx-rh-rostralanteriorcingulate
35	**L.CER**	Left-Cerebellum-Cortex	75	**R.RMFG**	ctx-rh-rostralmiddlefrontal
36	**L.TH**	Left-Thalamus-Proper	76	**R.SFG**	ctx-rh-superiorfrontal
37	**L.CA**	Left-Caudate	77	**R.SPG**	ctx-rh-superiorparietal
38	**L.PU**	Left-Putamen	78	**R.STG**	ctx-rh-superiortemporal
39	**L.PA**	Left-Pallidum	79	**R.SMG**	ctx-rh-supramarginal
40	**L.HI**	Left-Hippocampus	80	**R.FP**	ctx-rh-frontalpole
			81	**R.TP**	ctx-rh-temporalpole
			82	**R.TTG**	ctx-rh-transversetemporal
			83	**R.IN**	ctx-rh-insula
			84	**R.CER**	Right-Cerebellum-Cortex
			85	**B.Stem**	Brainstem

Bibliografia

[1] Muthuraman Muthuraman, Vinzenz Fleischer, Pierre Kolber, Felix Luessi, Frauke Zipp and Sergiu Groppa, "Structural Brain Network Characteristics Can Differentiate CIS from Early RRMS", Frontiers in Human Neuroscience ·Volume 10-Article 14-January 2016.

[2] Patric Hagmann , Lisa Jonasson, Philippe Maeder, Jean-Philippe Thiran, Van J. Wedeen, Reto Meuli, Understanding Diffusion MR Imaging Techniques: From Scalar Diffusion-weighted Imaging to Diffusion Tensor Imaging and Beyond.

[3] Duda, Richard O., Hart, Peter E. and Stork, David G.. Pattern Classification. 2 New York: Wiley, 2001.

[4] Zachary Jones and Fridolin Linder , "Exploratory Data Analysis using Random Forests", prepared for the 73rd annual MPSA conference, April 16-19, 2015-¿Proc MPSA, 2015.

[5] Alessandro Daducci, Alessandro Dal Palù, Alia Lemkaddem, and Jean-Philippe Thiran, "COMMIT: Convex Optimization Modeling for Microstructure Informed Tractography", IEEE TRANSACTIONS ON MEDICAL IMAGING, VOL. 34, NO. 1, 246-257, JANUARY 2015.