

Università degli Studi di Padova
Dipartimento di Scienze Statistiche
Corso di Laurea Magistrale in
Scienze Statistiche



**Pattern analysis per la risonanza magnetica funzionale
tramite rotazione di Procruste generalizzata**

Relatore Prof. Livio FINOS
Dipartimento di Psicologia

Laureando: Fabio CACCO
Matricola N. 1082619

Anno Accademico 2015/2016

Indice

1	Risonanza magnetica funzionale	3
1.1	Effetto BOLD	4
1.2	Disegno sperimentale	5
1.3	Dati	6
1.3.1	Soggetti	6
1.3.2	Acquisizione dei dati	7
1.3.3	Paradigma sperimentale	7
1.4	Pre-processing dei dati	8
1.4.1	Normalizzazione dell'intensità	8
1.4.2	Skull-Strip delle immagini strutturali	9
1.4.3	Motion Correction	9
1.4.4	Normalizzazione spaziale	9
1.4.5	Coregistrazione	10
2	Analisi di Procruste	11
2.1	Hyperalignment	12
2.2	Orthogonal Procruste Analysis	13
2.3	Extended Orthogonal Procruste Analysis	15
2.4	Generalized Orthogonal Procruste Analysis	19
2.5	Simulazione	21
3	Multi-Voxel Pattern Analysis	29
3.1	Esempio	29
3.2	Pre-elaborazione dati	31

3.3	Modelli	31
3.3.1	Modello MARS	32
3.3.2	Modello Lasso	39
3.3.3	Modello Albero di classificazione	43
3.3.4	Modello Random Forest	49
3.3.5	Modello Rete Neurale	57
4	Conclusioni	63

Elenco delle tabelle

2.1	Matrice di rotazione R_1 , per la creazione della matrice B	22
2.2	Matrice di rotazione R_2 , per la creazione della matrice C	22
3.1	Tabella di errata classificazione (caso dati allineati), modello MARS.	36
3.2	Tabella di errata classificazione (caso dati non allineati), modello MARS.	36
3.3	Errore percentuale del modello MARS.	39
3.4	(dati allineati) Tabella di errata classificazione, modello lasso.	40
3.5	(dati non allineati) Tabella di errata classificazione, modello lasso.	40
3.6	Errore percentuale del modello.	43
3.7	(caso dati allineati) Tabella di errata classificazione, modello albero di classificazione.	46
3.8	(caso dati non allineati) Tabella di errata classificazione, modello albero di classificazione.	46
3.9	Errore percentuale del modello.	49
3.10	(caso dati allineati) Tabella di errata classificazione, modello Random forest.	54
3.11	(caso dati non allineati) Tabella di errata classificazione, modello Random forest.	54
3.12	Errore percentuale del modello.	57
3.13	(caso dati allineati) Tabella di errata classificazione, modello Rete neurale.	58

3.14 (caso dati non allineati) Tabella di errata classificazione, modello Rete neurale.	58
3.15 Errore percentuale del modello.	61
4.1 Riduzione percentuale dell'errore nel passare dai dati non allineati a dati allineati.	66
4.2 Errore di classificazione commesso dai modelli nei due differenti insiemi di verifica dati non allineati a dati allineati.	66

Elenco delle figure

1.1	Questa immagine rappresenta la differenza tra risonanza magnetica a sinistra e risonanza magnetica funzionale a destra.	4
1.2	Effetto <i>BOLD</i>	5
1.3	Acquisizione delle immagini durante la fase di task (rosso) e la fase di rest (verde).	6
2.1	In questo grafico ogni singola spezzata rappresenta una delle 100 iterazioni eseguite, ogni singolo punto rappresenta il valore del baricentro in funzione dell'iterazione eseguita all'interno della funzione procruste. L'iterazione $i0$ rappresenta il valore iniziale del baricentro, quindi non è altro che la media aritmetica iniziale delle tre matrici (A, B, C).	23
2.2	Errori per ogni matrice generata rispetto alla iterazione eseguita all'interno della funzione di procruste, il grafico mostra gli errori raggruppati per matrice, da sinistra verso destra troviamo (A, B, C).	24
2.3	In questo grafico ogni singola spezzata rappresenta una delle 100 iterazioni eseguite, ogni singolo punto rappresenta l'errore medio delle tre matrici in funzione all'iterazione eseguita all'interno della funzione procruste.	25
2.4	Rappresentazione del fattore di scala medio in funzione della matrice associata. Ogni punto è la media degli N fattori di scala calcolati per matrice a ogni singola iterazione	26

2.5	Rappresentazione degli errori ottenuti all'ultima iterazione della funzione procruste per ogni matrice	27
3.1	(Caso dati allineati) In questo grafico sono riportate le variabili che fanno parte del modello, e la funzione che lega le variabili con la variabile risposta.	34
3.2	(Caso dati non allineati) In questo grafico sono riportate le variabili che fanno parte del modello, e la funzione che lega le variabili con la variabile risposta.	35
3.3	Rappresentazione grafica della tabella di errata classificazione (caso dati allineati), modello MARS.	37
3.4	Rappresentazione grafica della tabella di errata classificazione (caso dati non allineati), modello MARS.	38
3.5	(caso dati allineati) Rappresentazione grafica della tabella di errata classificazione, modello lasso.	41
3.6	(caso dati non allineati) Rappresentazione grafica della tabella di errata classificazione, modello lasso.	42
3.7	(Caso dati allineati) In questo grafico viene mostrato l'errore di classificazione commesso dal modello, sui dati T2 in funzione del numero di nodi. La retta di color rosso tratteggiata rappresenta il numero di nodi che riduce al minimo gli errori di classificazione in T2, che corrisponde a $N = 47$	44
3.8	(Caso dati non allineati) In questo grafico viene mostrato l'errore di classificazione commesso dal modello, sui dati T2 in funzione del numero di nodi. La retta di color rosso tratteggiata rappresenta il numero di nodi che riduce al minimo gli errori di classificazione in T2, che corrisponde a $N = 40$	45
3.9	(Caso dati allineati) Rappresentazione grafica della tabella di errata classificazione, modello albero di classificazione.	47
3.10	(Caso dati non allineati) Rappresentazione grafica della tabella di errata classificazione, modello albero.	48

3.11 (Caso dati allineati) Questo grafico mostra l'errore OOB in funzione del numero di variabili che usiamo nei nodi, in questo caso il numero di variabili che minimizza l'errore OOB è risultato quattro.	50
3.12 (Caso dati allineati) Questo grafico mostra l'errore OOB in funzione del numero di variabili che usiamo nei nodi, in questo caso il numero di variabili che minimizza l'errore OOB è risultato quattro.	51
3.13 (Caso dati allineati) Nel grafico vediamo l'importanza della variabile in termini di decremento medio di accuratezza, (ad esempio eliminando la variabile che risulta essere più importante ai fini della classificazione vR_{229} , commetto circa 9.5 errori in più in media).	52
3.14 (Caso dati non allineati) Nel grafico vediamo l'importanza della variabile in termini di decremento medio di accuratezza, (ad esempio eliminando la variabile che risulta essere più importante ai fini della classificazione vR_{203} , commetto circa 10 errori in più in media).	53
3.15 (Caso dati allineati) Rappresentazione grafica della tabella di errata classificazione, modello Random forest	55
3.16 (Caso dati non allineati) Rappresentazione grafica della tabella di errata classificazione, modello Random forest.	56
3.17 (caso dati allineati) Rappresentazione grafica della tabella di errata classificazione, modello Rete neurale.	59
3.18 (caso dati non allineati) Rappresentazione grafica della tabella di errata classificazione, modello Rete neurale.	60
4.1 Rappresentazione delle curve ROC dei modelli.	64
4.2 Rappresentazione grafica degli errori commessi dai modelli. . .	65
4.3 Rappresentazione grafica 3D dei voxel attivi e i rispettivi livelli di intensità del primo soggetto.	67

Introduzione

Il presente lavoro è nato per migliorare lo studio eseguito da Haxby e presentato nel suo articolo "*A common high-dimensional model of the representational space in human ventral temporal cortex*" [9]. Haxby ha proposto un nuovo metodo di analisi per i dati provenienti da esperimenti fMRI.

L'esperimento fMRI è una metodologia applicata soprattutto nell'ambito delle neuroscienze, infatti attraverso essa è possibile acquisire immagini funzionali sulla attività neurale del cervello. Questi esperimenti stanno prendendo sempre più piede grazie alla loro principale caratteristica di essere non invasivi. Il metodo di Haxby introduce una nuova fase prima della costruzione dei modelli, questa fase è denominata Hyperalignment. La tecnica del Hyperalignment si basa sulla analisi di procruste, metodo tipicamente utilizzato in *shape analysis*, Haxby ha utilizzato una delle possibili varianti della procruste la Orthogonal Procruste Analysis (OPA). L'OPA è la più semplice delle varianti della analisi di procruste, questa variante risulta essere la meno flessibile, come si potrà vedere nel secondo capitolo dove descriveremo l'analisi di procruste.

Quello che si è fatto in questo lavoro per migliorare il metodo proposto da Haxby è stato generalizzare il problema utilizzando la Generalized Orthogonal Procruste che risulta essere più flessibile rispetto alla OPA, creando una funzione apposita. Inoltre ci siamo chiesti se l'uso dell'Hyperalignment migliorasse o meno le prestazioni dei modelli di classificazione, che nell'ambito delle neuroscienze vengono detti MVPA.

Per valutare se c'erano o meno miglioramenti si sono eseguiti gli stessi modelli sia sui dati originali (dati dove non è stata eseguita la procruste) sia sui dati allineati (dati dove si è eseguita la procruste), i modelli saranno presen-

tati nel terzo capitolo con i rispettivi risultati modello per modello. Infine si sono confrontati i risultati ottenuti dove verranno esposti nel quarto capitolo, come metodo di confronto per i modelli si è preso in considerazione l'errore di classificazione valutando anche le rispettive curve ROC.

Capitolo 1

Risonanza magnetica funzionale

Una tecnica di neuro-imaging sviluppata negli ultimi trent'anni è la risonanza magnetica funzionale (fMRI¹), con la quale si possono ricavare informazioni sull'attività neuronale durante una stimolazione. L'fMRI è diventata un'importante tecnica di ricerca utilizzata in vari campi che vanno dalle neuroscienze di base alla pratica clinica, come nel caso di pianificazione pre-chirurgica o allo studio del recupero delle funzioni cerebrali dopo una lesione. Lo studio delle funzioni cerebrali con il sistema a risonanza magnetica funzionale sfrutta le capacità di localizzazione spaziale degli scanner utilizzati per la risonanza magnetica ma in più rileva le variazioni di flusso, del volume e della ossigenazione del sangue in seguito ad una attivazione cerebrale, sfruttando quello che viene chiamato l'effetto BOLD², in figura 1.1 possiamo vedere la differenza tra la risonanza magnetica e l'fMRI. Più precisamente il segnale *BOLD* non offre una misura diretta dell'attività cerebrale, ma ci indica solo la variazione di attività cerebrale. Per questo negli esperimenti fMRI vengono utilizzati almeno due tipi di stimoli, uno stimolo di attivazione (task) ed uno di riposo (rest), in modo da avere due segnali dell'attività cerebrale diversi. La caratteristica principale di questo metodo è la sua non invasività, nel senso che non viene somministrato nessun mezzo di contrasto paramagnetico.

¹fMRI: functional Magnetic Resonance Imaging

²BOLD: Blood Oxygenation Level Dependent

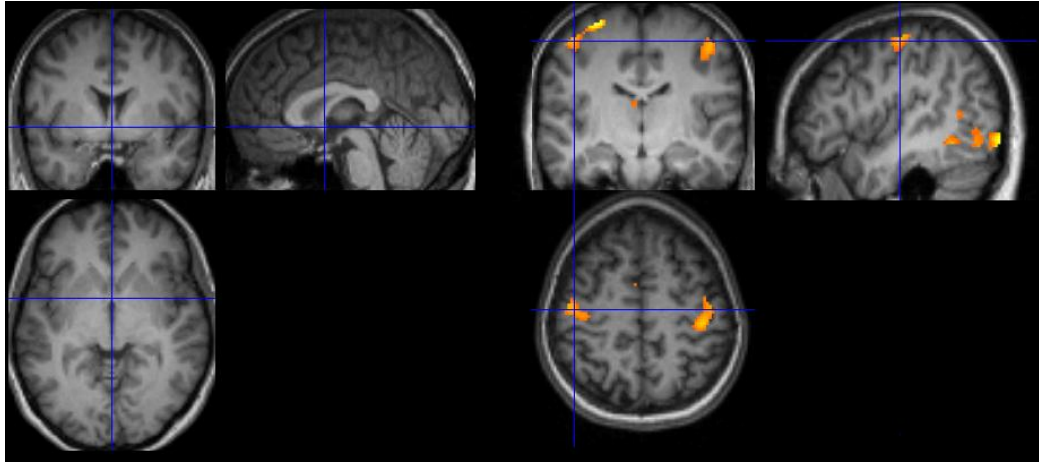


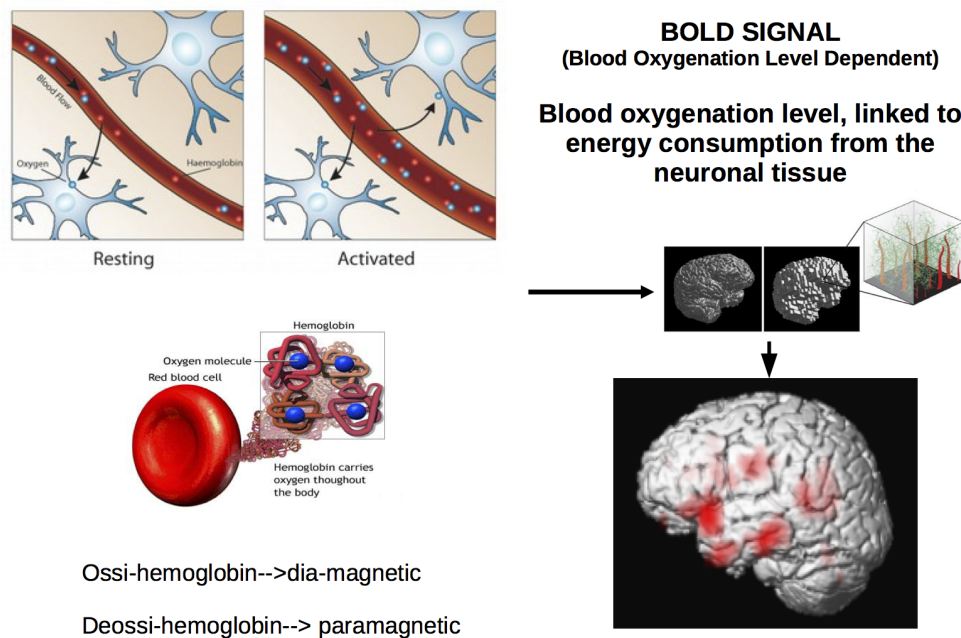
Figura 1.1: Questa immagine rappresenta la differenza tra risonanza magnetica a sinistra e risonanza magnetica funzionale a destra.

1.1 Effetto BOLD

L'fMRI è in grado di visualizzare la risposta emodinamica correlata all'attività neurale del cervello. Durante un'attività cerebrale, i neuroni attivi consumano ossigeno, con un conseguente aumento di flusso e volume sanguigno finalizzato all'apporto di sangue ricco di ossigeno. Questo comporta un cambiamento del rapporto fra la concentrazione di ossiemoglobina e de-ossiemoglobina. Poiché le due molecole hanno proprietà magnetiche diverse, ossiemoglobina è diamagnetica³ e de-ossiemoglobina è paramagnetica⁴, si ha una variazione del campo magnetico, cioè c'è una variazione dell'iterazione del materiale con il campo magnetico. Minore è la concentrazione di de-ossiemoglobina e maggiore sarà l'intensità del segnale misurato, come si può vedere in figura 1.2.

³Diamagnetica: Sostanze le cui molecole non hanno un momento magnetico proprio.

⁴Paramagnetiche: Sostanze le cui molecole hanno un momento magnetico proprio.

Figura 1.2: Effetto *BOLD*

1.2 Disegno sperimentale

Il tipico esperimento fMRI consiste nella presentazione di una serie di task (stimoli) durante l'acquisizione del segnale *BOLD*. L'acquisizione riguarda una serie di immagini tridimensionali che comprendono tutto il cervello. Nel corso dell'esperimento lo stesso stimolo viene presentato più volte e vengono acquisite centinaia di immagini *BOLD*. L'analisi statistica dei dati è finalizzata a stabilire se e dove nel cervello si verificano variazioni sistematiche del segnale *BOLD* in corrispondenza della presentazione degli stimoli. Fino ad oggi ci sono due classi di disegni sperimentali sull'fMRI, event-related designs e block designs. I dati che analizzeremo più avanti provengono da un esperimento dove il disegno sperimentale era block designs, detto anche

disegno a blocchi. Nel disegno sperimentale a blocchi vengono alternati nel tempo degli intervalli di attivazione e degli intervalli di riposo. Il blocco di attivazione è composto da una serie di stimoli uguali e successivi, che possono essere motori, sensitivi o cognitivi, mentre durante il blocco di rest il soggetto rimane in condizioni di riposo, come si vede in figura 1.3.

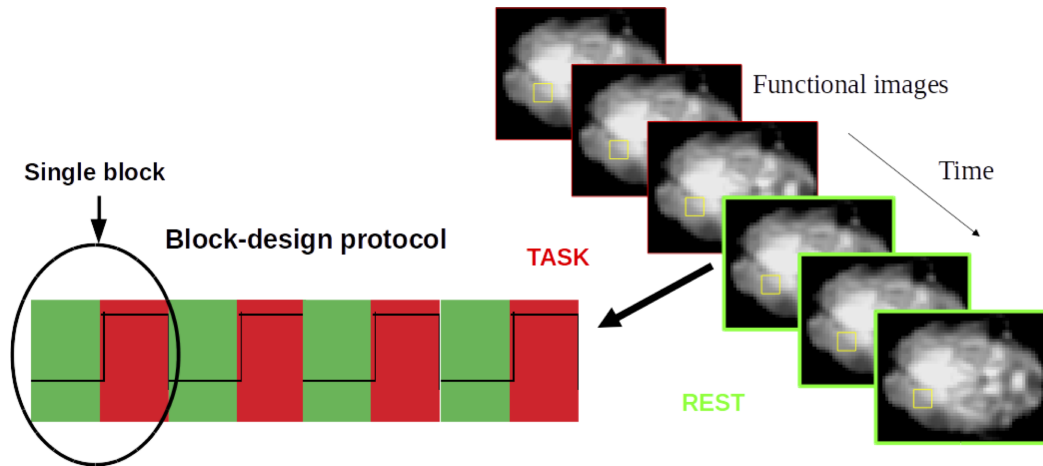


Figura 1.3: Acquisizione delle immagini durante la fase di task (rosso) e la fase di rest (verde).

Queste due condizioni sono necessarie per poter ottenere delle informazioni sulle aree che si attivano, confrontando il *BOLD* acquisito tra la fase di attivazione con quella di riposo. Il disegno sperimentale a blocchi, grazie al fatto che le risposte ai singoli stimoli ripetuti numerosamente si sommano fra loro in modo lineare, permette di avere un elevato rapporto segnale-rumore (SNR) e, quindi, un alto potere di discriminazione dei voxel.

1.3 Dati

1.3.1 Soggetti

I dati che verranno utilizzati nel seguito fanno parte di uno studio più ampio presentato da nell'articolo di Gorgolewski [5][6]. Inizialmente i dati erano

riferiti a undici soggetti volontari, rimasti in dieci dato che tutte le risposte di un soggetto sono state scartate per un problema di esecuzione dei compiti. Una sola sessione riferita ad un altro soggetto è stata scartata. I dieci volontari includono quattro soggetti di sesso maschile e sei soggetti di sesso femminile, tre mancini e sette destrimani. L'età media è 52.55 anni con un range d'età che varia tra i 50-58 anni.

1.3.2 Acquisizione dei dati

Tutte le scansioni sono state acquisite su un scanner GE Signa HDxt 1.5 T, e sono avvenute al Brain Research Imaging Centre University of Edinburgh. Il Protocollo sperimentale degli studi fMRI prevede l'acquisizione di immagini funzionali del cervello (T2-weighted) acquisite usando sequenze di imaging gradient echo, del tipo echo planar (EPI), sono immagini funzionali a bassa risoluzione spaziale. Per avere una dettagliata anatomia del cervello, per ogni soggetto vengono inoltre acquisite immagini strutturali ad alta risoluzione spaziale (T1-weighted) del tipo spoiled gradient recall (SPGR). Per l'acquisizione di immagini T1 sono stati impostati questi parametri; field of view (*FOV*): 256×256 mm, Slice thickness: 1.3 mm, Number of slices: 156, Slice acquisition order: front to back, Slice orientation: coronal, tempo di Ripetizione (*TR*): 10 sec., tempo d'eco (*TE*): 4 sec., No. of volumes in time series: 1, Excitation flip angle (deg): 8. Invece per l'acquisizione delle immagini funzionali sono stati impostati questi parametri: field of view (*FOV*): 256×256 mm, In-plane matrix: 64×64 , slice thickness: 4mm, Number of slices: 30, slice acquisition order: interleaved, Slice orientation: axial, tempo di ripetizione (*TR*): 2.5 sec., tempo d'eco (*TE*): 50 ms., No. of volumes in time series: 173, Excitation flip angle (deg): 90.

1.3.3 Paradigma sperimentale

L'esperimento iniziale prevedeva diversi task che comprendevano le tematiche di natura linguistica, natura motoria e di natura visiva. Ora descriveremo solo il task riferito ai dati analizzati che risulta essere di natura linguistica. Il task "*Covert verb generation task*" chiedeva ai soggetti di pensare a un

verbo e completare una frase con un sostantivo a loro presentato visivamente, immagine presentata a schermo. Le istruzioni che dovevano eseguire i soggetti sono state: "Quando viene visualizzata una parola questa sarà un sostantivo. Pensate a quello che potete fare con questa e immaginate di dire 'Con questo io posso ...' o 'quello che posso ...'". Il disegno comprende un blocco di stimolo della durata di 30 sec. e un blocco di riposo della stessa durata, per ogni blocco si acquisivano 12 volumi, immagini tridimensionali del cervello, quindi ogni 2.5 secondi si acquisiva una immagine dell'attività funzionale del cervello. Si sono eseguite otto prove utilizzate per l'addestramento. Durante i blocchi di attivazione, dieci nomi sono stati presentati per un secondo ciascuno seguito da un breve periodo di stop durante il quale il soggetto doveva generare la risposta. I nomi sono stati scelti a caso da una serie di 70 nomi. Si sono eseguite sette prove ognuna delle quali conteneva un blocco di attivazione ed uno di riposo. Per ogni soggetto si sono salvati 173 immagini funzionali.

1.4 Pre-processing dei dati

I dati che verranno analizzati non avevano bisogno di questa fase, dato che era già stata eseguita a monte, ma giusto per capire cosa comprende questa fase la descriveremo in maniera breve e abbastanza superficiale dato che non fa parte dello scopo di questo lavoro. Quando le immagini cerebrali MRI vengono acquisite hanno formato un *DICOM*, e risulta necessario eseguire convertirle nel formato *NII*, formato standard per le immagini medicali e che può essere utilizzato nei programmi di analisi di immagini.

1.4.1 Normalizzazione dell'intensità

Le immagini acquisite presentano delle imperfezioni tipiche delle acquisizioni MRI, la variazione dell'intensità luminosa dei pixel in funzione della posizione. Questo è dovuto principalmente alle disomogeneità del campo magnetico e alla variabilità dell'anatomia dei soggetti, causando una variazione di intensità luminosa e del contrasto nell'immagine. La variazione di intensità

ottenuta nei diversi pixel provoca che uno stesso tipo di tessuto abbia un valore di intensità significativamente diverso in funzione della posizione spaziale. Eseguire una fase di correzione dell'intensità del segnale risulta quindi necessario e di notevole importanza.

1.4.2 Skull-Strip delle immagini strutturali

Nella maggior parte delle applicazioni siamo interessati solo alla parte cerebrale delle immagini. In questa fase si vanno ad eliminare dalle immagini, attraverso software appositi, tutte parti non cerebrali come ad esempio lo scalpo, occhi, collo e altre parti a cui non siamo interessati.

1.4.3 Motion Correction

La correzione dei movimenti della testa è una delle procedure principali. In generale si cerca di limitare la fonte principale degli errori di acquisizione poiché anche il più piccolo movimento del soggetto può indurre significativi errori di acquisizione. Tali errori sono più frequenti nei confini dei tessuti e nel bordo del cervello. Per limitare ma non eliminare del tutto i movimenti, anche involontari della testa esistono diversi metodi per immobilizzare la testa, ma questo non basta. La fase motion correction prevede di scegliere un volume funzionale come riferimento al quale verranno allineati tutti i restanti volumi, attraverso l'uso di algoritmi di allineamento prevedono l'uso di sei parametri, tre di traslazione lungo i tre assi (X, Y, Z) e tre di rotazione (roll, pitch, yaw).

1.4.4 Normalizzazione spaziale

Negli studi di fMRI si acquisiscono immagini di un certo numero di soggetti diversi. Il cervello umano si differenzia da un soggetto all'altro sia per forma che dimensioni. In questa fase l'obiettivo è quello di deformare le singole acquisizioni in modo che una regione nella scansione di un soggetto corrisponda alla medesima regione nella scansione di un altro soggetto, in

modo da rendere le acquisizioni confrontabili tra loro. Generalmente tutte le acquisizioni sono allineate su di un unico specifico atlante.

1.4.5 Coregistrazione

Come già detto abbiamo molteplici immagini funzionali dalla scarsa risoluzione e l'immagine strutturale ad alta risoluzione. Per riconoscere con precisione le regioni di particolare interesse nelle immagini funzionali è necessario allineare queste con l'immagine strutturale. La fase di coregistrazione prevede di allineare le immagini funzionali sull'immagine strutturale. L'obiettivo è quello di sovrapporre le informazioni contenute nelle immagini funzionali su un'immagine dove è possibile discriminare le regioni anatomiche.

Capitolo 2

Analisi di Procruste

Il nome Procuste si riferisce a un bandito della mitologia greca, Procuste era il soprannome del bandito, il suo vero nome era Damaste, che appostato sul monte Coridallo, nell'Attica, lungo la via sacra tra Eleusi e Atene, aggrediva i viandanti. I malcapitati venivano fatti sdraiare su un letto, ricavato nella roccia, e stirati a forza se essi erano più corti rispetto al letto, altrimenti amputati nel caso contrario, si narra che possedesse due letti uno molto corto ed uno più lungo.

Con l'Analisi di Procruste si fa riferimento a un tipo particolare di shape analysis.

Immaginiamo di avere a disposizione due o più forme, solo i contorni degli oggetti, la maniera più semplice ed intuita per vedere i tratti comuni è quello di sovrapporle. La sovrapposizione può essere eseguita utilizzando rotazioni, traslazioni e cambiamenti di scala che vanno sotto un unico nome trasformazioni di similarità, con l'obiettivo di allineare le forme nel miglior modo possibile, riducendo al minimo l'errore. Nel seguito utilizzeremo il termine "matrice", concetto più familiare, al posto del termine "forma dell'oggetto", dato che possiamo riassumere le coordinate spaziali dell'oggetto in una matrice. Per eseguire le analisi di Procrustes non è richiesta nessuna informazione a priori sulla relazione geometrica esistente tra le diverse forme. L'analisi di Procrustes è una tecnica geometrica utilizzata per allineare le varie matrici,

che stima i parametri di trasformazione ottimali per ridurre al minimo l'errore.

Il metodo denominato Orthogonal Procrustes Problem, formulato da Schoenemann nel 1966 [11], dà una soluzione diretta usando i minimi quadrati al problema che risultava essere trasformare una data matrice A in una data matrice B utilizzando una matrice trasformazione ortogonale T in modo tale da minimizzare la somma dei quadrati dei residui $E = AT - B$.

La prima generalizzazione del Orthogonal Procrustes Problem è stata data da Schoenemann e Carroll nel 1970 [12] quando per adattate una matrice A a una matrice B furono aggiunti una matrice di rotazione, un parametro per la traslazione e un fattore di scala, esso viene chiamato Extended Orthogonal Procrustes problem, altra generalizzazione utilizzata nel caso si abbiano più di due matrici prende il nome di Generalized Orthogonal Procrustes problem [7][2]. Il capitolo prosegue con la presentazione del metodo utilizzato da Haxby, Hyperalignment, e con la spiegazione matematica delle differenti versione della analisi di Procruste.

2.1 Hyperalignment

La tecnica dell'hyperalignment introdotta nell'articolo di Haxby [9] per allineare dati fMRI, non è altro che l'allineamento dell'intensità dei segnali dei singoli soggetti, utilizzando l'analisi Procruste. Nel suo lavoro Haxby sceglie di utilizzare la Orthogonal Procrustes Analysis e come matrice di riferimento la matrice di un soggetto, ad esempio il primo. In questo lavoro si è optato per delle scelte differenti, che potrebbero migliorare ulteriormente il metodo rendendolo ancora più generale e quindi anche più fruibile. Abbiamo scelto di usare la versione generalizzata della procruste dato che questa offre maggiori parametri per l'allineamento ed abbiamo scelto come matrice di riferimento una matrice media, costruita come valore medio voxel per voxel di tutti i soggetti. Per eseguire la procruste come si può capire dobbiamo prima passare da una dimensione 4D, sequenze di volumi 3D nel tempo, a delle matrici riferite ad ogni soggetto. Questa trasformazione porta a considerare volta per volta un singolo voxel nel tempo. Nel senso che se prendiamo un

singolo voxel e valutiamo i suoi segnali nel tempo quello che otteniamo sarà un vettore. Eseguiamo questa operazione per ogni singolo voxel e quello che si ottiene è una matrice che ci dice come ogni singolo voxel varia nel tempo, in questo modo ora si potrà eseguire la Procruste.

2.2 Orthogonal Procruste Analysis

Nel problema Orthogonal Procrustes lo scopo è allineare una matrice A su una matrice B , oppure viceversa, utilizzando matrici di rotazione ortogonali T . Ipotizziamo di voler allineare A a B , entrambi di dimensione $(n \times p)$. Applicando il criterio dei minimi quadrati, minimizzando la somma dei quadrati dei residui, $E = AT - B$, si trova la stima della matrice di rotazione T . Per via matematica il problema può essere impostato nel seguente modo, teniamo presente che d'ora in poi la matrice di riferimento sarà B , mentre la matrice A sarà quella che dovrà allinearsi a B grazie all'uso di matrice di rotazione T :

$$AT = B + E \quad (2.2.0.1)$$

sotto la condizione:

$$TT^T = T^T T = I \quad (2.2.0.2)$$

La funzione obiettivo che dovrà essere minimizzata risulta essere:

$$tr(E^T E) = tr\left[(AT - B)^T (AT - B)\right] = tr(T^T A^T AT - 2T^T A^T B + B^T B) = \min \quad (2.2.0.3)$$

L'equazione (2.2.0.3) non tiene conto del vincolo (2.2.0.2), quindi possiamo scrivere il tutto in una unica funzione:

$$F = tr(E^T E) + tr(L(T^T T - I)) \quad (2.2.0.4)$$

dove la matrice L , di dimensione $(p \times p)$, è una matrice di moltiplicatori di Lagrange, ora deriviamo la (2.2.0.4) rispetto a T e poniamo la derivata uguale a 0:

$$\frac{\partial F}{\partial T} = 2A^T AT - 2A^T B + T(L + L^T) = 0 \quad (2.2.0.5)$$

le matrici, $(A^T A)$ e $(L + L^T)$, sono matrici simmetriche. Pre-moltiplicando per T^T la (2.2.0.5) otteniamo:

$$T^T A^T A T - T^T A^T B + \frac{L + L^T}{2} = 0 \quad (2.2.0.6)$$

$$\frac{L + L^T}{2} = T^T (A^T B) - T^T (A^T A) T = \left[\frac{(L + L^T)}{2} \right]^T \quad (2.2.0.7)$$

Il termine $T^T (A^T A) T$ risulta essere simmetrico, mentre il termine $T^T (A^T B)$ dobbiamo imporre una condizione di simmetria:

$$T^T (A^T B) = (A^T B)^T T \quad (2.2.0.8)$$

Quello che stiamo cercando è la matrice T che soddisfi la condizione di simmetria, quindi pre-moltiplichiamo per T e otteniamo:

$$(A^T B) = T (A^T B) T \quad (2.2.0.9)$$

mentre post-moltiplicando per T^T si ottiene:

$$T^T (A^T B) T^T = (A^T B)^T \quad (2.2.0.10)$$

Quindi usando le equazioni appena trovate, (2.2.0.9) e (2.2.0.10), e dato che siamo sotto una condizione di simmetria otteniamo la seguente equazione:

$$(A^T B) (A^T B)^T = T (A^T B)^T (A^T B) T^T \quad (2.2.0.11)$$

le matrici $\left[(A^T B) (A^T B)^T \right]$ e $\left[(A^T B)^T (A^T B) \right]$ sono simmetriche, entrambe hanno gli stessi autovalori, applicando la decomposizione a valori singolari nota anche come *SVD* a entrambi i membri della (2.2.0.11):

$$svd \left[(A^T B) (A^T B)^T \right] = T svd \left[(A^T B)^T (A^T B) \right] T^T \quad (2.2.0.12)$$

si ottiene:

$$V D_S V^T = T W D_S W^T T^T \quad (2.2.0.13)$$

Al fine che si verifichi l'uguaglianza dell'equazione (2.2.0.13) dobbiamo imporre:

$$V = TW \quad (2.2.0.14)$$

La soluzione al problema, trovare la matrice T che minimizzi (2.2.0.3) sotto la condizione (2.2.0.2) è:

$$T = VW^T \quad (2.2.0.15)$$

2.3 Extended Orthogonal Procruste Analysis

Una prima generalizzazione al Orthogonal Procrustes di Schönemann del 1966 [11] fu proposta dallo stesso Schönemann e da Carrol nel 1970 [12]. Rispetto alla proposta iniziale di Schönemann aggiunsero dei parametri per migliorare l'allineamento tra matrici, un parametro di traslazione t e un fattore di scala c , il seguente approccio prende il nome di Extended orthogonal procrustes analysis.

La differenza tra la matrice da allineare A e la matrice di riferimento B la esprimiamo come:

$$E = cAT + jt^T - B \quad (2.3.0.16)$$

dove $j^T = [1, 1, \dots, 1]$ vettore unitario di dimensione $(1 \times n)$, le matrici A e B hanno dimensione $(n \times p)$. La matrice di rotazione ortogonale T ha dimensione $(p \times p)$, il vettore di traslazione t ha dimensione $(p \times 1)$ e c è un fattore di scala. Per ottenere la stima dei parametri di interesse, (T, t, c) , scriviamo la "funzione lagrangiana"/ funzione obiettivo:

$$F = tr [E^T E] + tr [L (T^T T - I)] \quad (2.3.0.17)$$

sostituendo E , (2.3.0.16), alla (2.3.0.17) si ottiene:

$$F = tr \left[(cAT + jt^T - B)^T (cAT + jt^T - B) \right] + tr [L (T^T T - I)] \quad (2.3.0.18)$$

$$\begin{aligned}
\text{tr} [E^T E] &= \text{tr} \left[(cAT + jt^T - B)^T (cAT + jt^T - B) \right] = \\
&= \text{tr} [B^T B] + c^2 \text{tr} [T^T A^T AT] + nt^T t - \\
&\quad + 2ctr [B^T AT] - 2tr [B^T jt^T] + 2ctr [T^T A^T jt^T]
\end{aligned}$$

la quantità $n = j^T j$ è uno scalare e indica il numero di righe delle matrici. Eseguendo le derivate parziali rispetto a tre parametri ignoti (T, t, c) otteniamo:

$$\frac{\partial F}{\partial T} = 2c^2 A^T AT - 2cA^T B + 2cA^T jt^T + T(L + L^T) = 0 \quad (2.3.0.19)$$

$$\frac{\partial F}{\partial t} = 2pt - 2B^T j + 2cT^T A^T j = 0 \quad (2.3.0.20)$$

$$\frac{\partial F}{\partial c} = 2ctr [T^T A^T AT] - 2tr [B^T AT] + 2tr [T^T A^T jt^T] = 0 \quad (2.3.0.21)$$

Dalla seconda equazione (2.3.0.20), ricaviamo il vettore delle traslazioni:

$$t = (B - cAT)^T \frac{j}{n} \quad (2.3.0.22)$$

I termini ($A^T A$) e ($L + L^T$) della (2.3.0.19), sono matrici simmetriche, premoltiplicandola per T^T si ottiene:

$$c^2 T^T (A^T A) T - cT^T A^T B + cT^T A^T jt^T + \frac{L + L^T}{2} = 0 \quad (2.3.0.23)$$

per la condizione di simmetria otteniamo:

$$\frac{L + L^T}{2} = cT^T A^T B - cT^T A^T jt^T - c^2 T^T (A^T A) T = \left[\frac{L + L^T}{2} \right]^T \quad (2.3.0.24)$$

Si noti che $T^T (A^T A) T$ risulta simmetrica, mentre ($cT^T A^T B - cT^T A^T jt^T$) deve risultare simmetrica, quindi usando la definizione di matrice simmetrica:

$$T^T A^T B - T^T A^T jt^T = (T^T A^T B - T^T A^T jt^T)^T \quad (2.3.0.25)$$

usando la soluzione precedentemente ricavata per t , (2.3.0.22) e sostituendola all'interno della (2.3.0.25) si ottiene la seguente condizione di simmetria:

$$\begin{aligned} T^T A^T B - T^T A^T \left(\frac{jj^T}{p} \right) (B - cAT) &= \\ = T^T A^T B - T^T A^T \left(\frac{jj^T}{p} \right) B + cT^T A^T \left(\frac{jj^T}{p} \right) AT &= \text{symm}. \end{aligned} \quad (2.3.0.26)$$

si noti che $T^T A^T \left(\frac{jj^T}{p} \right) AT$ è simmetrica. Per i termini rimanenti imponiamo la condizione di simmetria:

$$\begin{aligned} T^T A^T B - T^T A^T \left(\frac{jj^T}{p} \right) B &= \\ = T^T A^T \left(B - \left(\frac{jj^T}{p} \right) B \right) &= \\ = T^T A^T \left(I - \frac{jj^T}{p} \right) B &= \text{symm}. \end{aligned} \quad (2.3.0.27)$$

Poniamo:

$$S = A^T \left(I - \frac{jj^T}{p} \right) B \quad (2.3.0.28)$$

dove S è una matrice ($p \times p$). Riscriviamo (2.3.0.27) sostituendo la matrice S al suo interno e usando la definizione di matrice simmetrica, per il vincolo di simmetria imposto:

$$T^T S = S^T T \quad (2.3.0.29)$$

pre-moltiplichiamo per T l'equazione (2.3.0.29):

$$S = T S^T T \quad (2.3.0.30)$$

mentre post-moltiplicando per T^T :

$$T^T S T^T = S^T \quad (2.3.0.31)$$

Utilizzando (2.3.0.30) e (2.3.0.31), possiamo scrivere la seguente equazione:

$$S S^T = T S^T S T^T \quad (2.3.0.32)$$

le matrici SS^T ed $S^T S$ sono simmetriche ed entrambi hanno gli stessi autovalori. Applicando la decomposizione a valori singolari alle matrici SS^T ed $S^T S$:

$$svd(SS^T) = Tsvd(S^T S)T^T \quad (2.3.0.33)$$

quello che si ottiene è il seguente:

$$VD_S V^T = TWD_S W^T T^T \quad (2.3.0.34)$$

dove le matrici V e W sono matrici ortonormali e D_S è una matrice diagonale di autovalori. Al fine che si verifichi la condizione di uguaglianza della, (2.3.0.34) occorre:

$$V = TW \quad (2.3.0.35)$$

risolvendo rispetto alla matrice di rotazione ignota T otteniamo:

$$T = VW^T \quad (2.3.0.36)$$

Secondo Schönemann e Carroll [12] nella fase di calcolo si dovrebbe prendere in considerazione questa equazione:

$$svd(S) = svd\left(A^T \left(I - \frac{jj^T}{p}\right) B\right) = VDWT^T \quad D \neq D_S \quad (2.3.0.37)$$

Una volta trovato il parametro T , possiamo risolvere le altre equazioni trovando gli altri parametri ignoti (t, c), in particolare il fattore di scala, c , sarà:

$$c = \frac{tr\left(T^T A^T \left(I - \frac{jj^T}{p}\right) B\right)}{tr\left(A^T \left(I - \frac{jj^T}{p}\right) A\right)} \quad (2.3.0.38)$$

sostituendo, T e c , nell'equazione, 2.3.0.22 si trova il parametro di traslazione t .

2.4 Generalized Orthogonal Procruste Analysis

Nella realtà accade molto spesso che non si hanno solo due matrici da allineare, ma un insieme di matrici ed i metodi Orthogonal Procrustes Analysis ed Extended Procrustes Analysis non sono adatti allo scopo. La Generalized Orthogonal Procrustes Analysis è adatta a questo scopo[7][2][8]. In questo caso la funzione obbiettivo che si deve minimizzare è:

$$\text{tr} \left[\sum_{i=1}^m \sum_{j=i+1}^m [(c_i A_i T_i + j t_i^T) - (c_j A_j T_j + j t_j^T)]^T [(c_i A_i T_i + j t_i^T) - (c_j A_j T_j + j t_j^T)] \right] = \min \quad (2.4.0.39)$$

dove (A_1, A_2, \dots, A_m) sono matrici ed ognuna delle quali ha dimensioni $(n \times p)$. Nell'impostazione Goodall [4] esiste una matrice Z , detta matrice consenso, la quale contiene le vere coordinate degli n punti definiti in un sistema di coordinate comune[4].

La soluzione del problema può essere sintetizzata come la ricerca della matrice ignota Z . La matrice consenso la possiamo scrivere come:

$$Z + E_i = \hat{A}_i = c_i A_i T_i + j t_i^T \quad i = 1, 2, \dots, m \quad (2.4.0.40)$$

dove E_i è una matrice di errori. L'equazione (2.4.0.39) la possiamo scrivere come segue:

$$\sum_{i=1}^m \sum_{j=i+1}^m \|\hat{A}_i - \hat{A}_j\|^2 = \min \quad (2.4.0.41)$$

Definiamo una matrice C , che risulterà essere il baricentro geometrico delle matrici trasformate \hat{A}_i :

$$C = \frac{1}{m} \sum_{i=1}^m \hat{A}_i \quad (2.4.0.42)$$

Da un punto di vista computazionale la soluzione al problema di minimo della, (2.4.0.41) risulta essere molto oneroso e complicato.

$$\sum_{i=1}^m \sum_{j=i+1}^m \|\hat{A}_i - \hat{A}_j\|^2 = \sum_{i=1}^m \sum_{j=i+1}^m \text{tr} \left[\left(\hat{A}_i - \hat{A}_j \right)^T \left(\hat{A}_i - \hat{A}_j \right) \right] \quad (2.4.0.43)$$

L'equazione (2.4.0.41) riscritta anche come (2.4.0.43) la possiamo scrivere utilizzando il baricentro C :

$$m \sum_{i=1}^m \|\hat{A}_i - C\|^2 = m \sum_{i=1}^m \text{tr} \left[\left(\hat{A}_i - C \right)^T \left(\hat{A}_i - C \right) \right] \quad (2.4.0.44)$$

la (2.4.0.43) e (2.4.0.44) sono equivalenti ma dati i problemi computazionali della (2.4.0.43) è consigliabile usare (2.4.0.44). [10][8]

Si noti che per entrambe le funzioni obiettivo la soluzione la si raggiunge mediante un algoritmo iterativo. Si è deciso di esporre solo il metodo che prevede l'equazione (2.4.0.44). Quindi la nuova funzione obiettivo che vogliamo minimizzare è:

$$\sum_{i=1}^m \|\hat{A}_i - C\|^2 = \sum_{i=1}^m \text{tr} \left[\left(\hat{A}_i - C \right)^T \left(\hat{A}_i - C \right) \right] = \min \quad (2.4.0.45)$$

L'algoritmo iterativo per risolvere il problema di minimo è:

- Definiamo un baricentro iniziale C , una scelta sensata potrebbe essere la matrice media delle matrici iniziali.
- Calcoliamo i parametri di trasformazione (T, t, c) per ogni matrice A_i rispetto al baricentro C , dopo aver calcolato ogni matrice \hat{A}_i , aggiorniamo il baricentro C .
- Eseguiamo questa ultima operazione fino ad arrivare alla stabilizzazione di C .

Il baricentro C corrisponde alla stima, \hat{Z} , dei minimi quadrati di Z . La dimostrazione di questa definizione è stata data da Crosilla e Beinat.[3]

$$C = \hat{Z} = \frac{1}{m} \sum_{i=1}^m \hat{A}_i \quad (2.4.0.46)$$

2.5 Simulazione

Per verificare il funzionamento della procruste, e della funzione appositamente creata, abbiamo eseguito una simulazione, nella quale si è verificato se il baricentro C andava a stabilizzarsi dopo le varie iterazioni dove veniva aggiornamento e come si comportava l'errore $tr([E^T E])$. La funzione che abbiamo scritto esegue l'algoritmo presentato nel sezione Generalized Orthogonal Procruste Analysis, con un'unica variante che nel nostro caso non si è avuto bisogno di calcolare il fattore di traslazione, dato che i dati erano già centrati.

Si è deciso di procedere nella seguente maniera, si sono generate tre matrici, (A, B, C) , create attraverso rotazioni, (R, R_1, R_2) , di una matrice precedentemente creata in maniera del tutto casuale, (Z) , e aggiunto un errore casuale, (E_A, E_B, E_C) , come possiamo vedere nelle equazioni 2.5.0.47, 2.5.0.48 e 2.5.0.49.

$$A = ZR + E_A \quad (2.5.0.47)$$

$$B = ZR_1 + E_B \quad (2.5.0.48)$$

$$C = ZR_2 + E_C \quad (2.5.0.49)$$

Le matrici di rotazione scelte sono state una matrice di rotazione uguale alla matrice identità $R = I$ per la creazione della matrice A e due rotazioni arbitrarie per la creazione della matrice B (2.1) e della matrice C (2.2).

0.4554696	0.8902513
-0.8902513	0.4554696

Tabella 2.1: Matrice di rotazione R_1 , per la creazione della matrice B .

-0.2439426	-0.9697897
-0.9697897	0.2439426

Tabella 2.2: Matrice di rotazione R_2 , per la creazione della matrice C .

Una volta eseguita questa operazione è stata eseguita la Generalized Orthogonal Procrustes Analysis sulle matrici (A, B, C) . Questa operazione è stata eseguita 100 volte, tenendo fisse solo le rotazioni e creando ad ogni iterazione una nuova matrice casuale Z . Quindi ad ogni iterazione si venivano a creare tre nuove matrici (A, B, C) dove eseguire la procruste. Ad ogni iterazione si salvano i baricentri, l'errore, le matrici di rotazione, il fattore di scala e le nuove matrici trasformate. Finita la fase di simulazione si sono eseguiti dei grafici per valutare il comportamento del baricentro a ogni iterazione (figura: 2.1), l'errore di ogni matrice generata in funzione della matrice di rotazione e del numero di iterazione eseguito all'interno della funzione procruste (figura: 2.2), l'errore medio commesso dalle tre matrici per ogni singola iterazione (figura: 2.3) ed infine la media di fattori di scala stimati ad ogni iterazione (figura: 2.4).

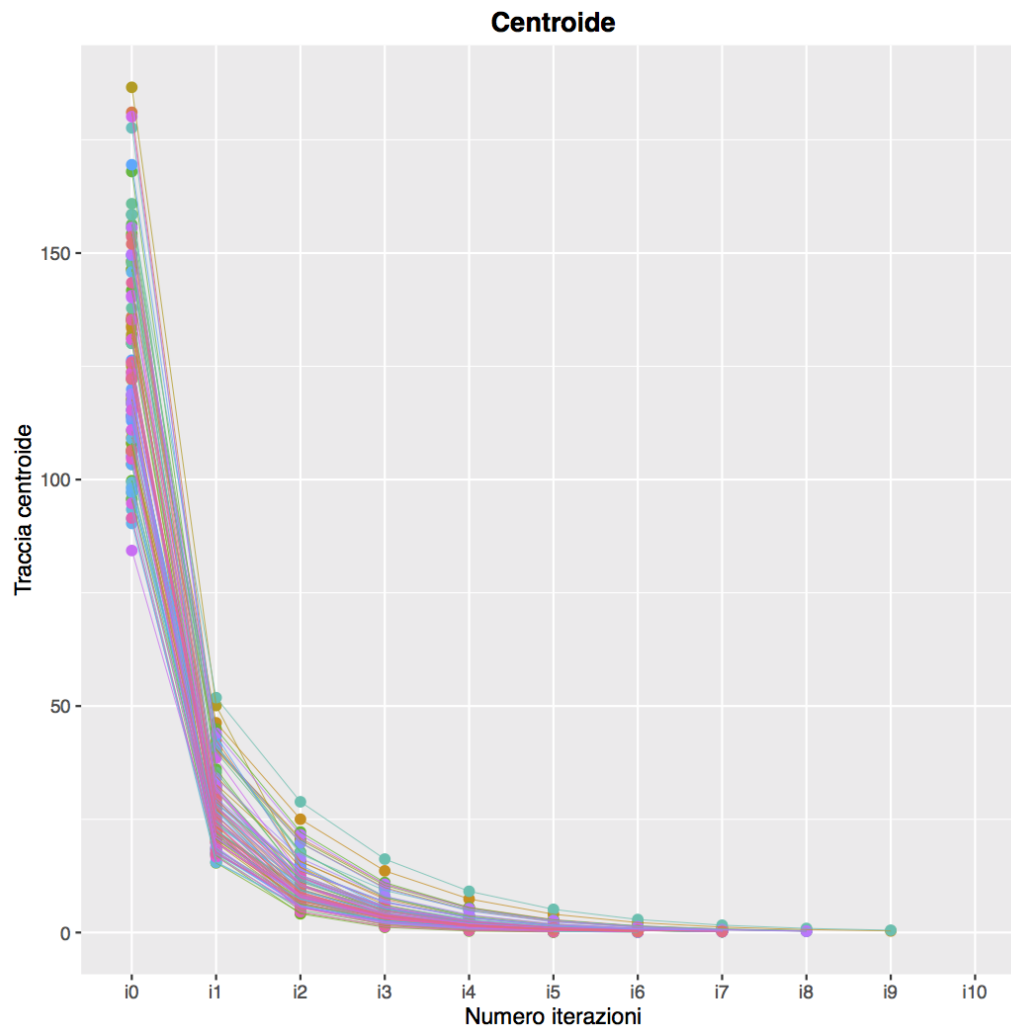


Figura 2.1: In questo grafico ogni singola spezzata rappresenta una delle 100 iterazioni eseguite, ogni singolo punto rappresenta il valore del baricentro in funzione dell'iterazione eseguita all'interno della funzione procruste. L'iterazione i_0 rappresenta il valore iniziale del baricentro, quindi non è altro che la media aritmetica iniziale delle tre matrici (A, B, C) .

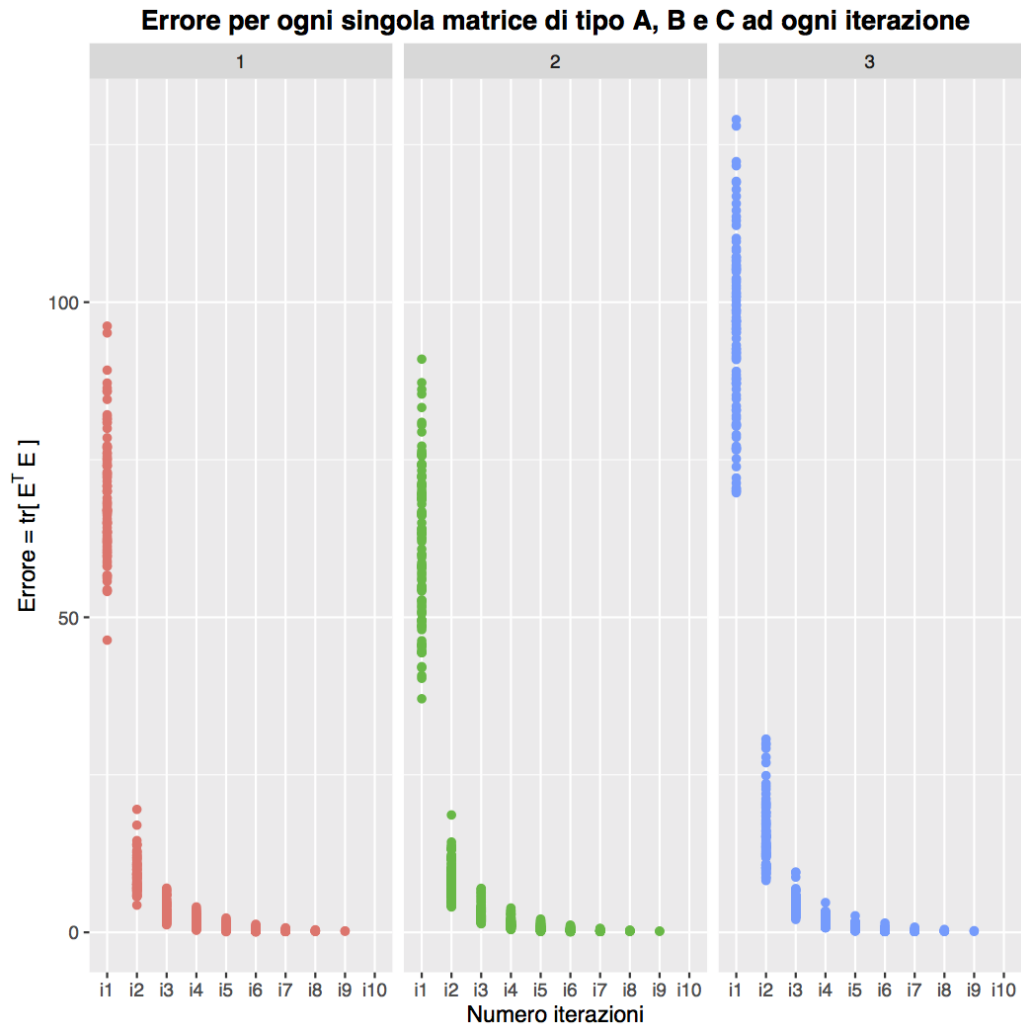


Figura 2.2: Errori per ogni matrice generata rispetto alla iterazione eseguita all'interno della funzione di procruste, il grafico mostra gli errori raggruppati per matrice, da sinistra verso destra troviamo (A, B, C) .

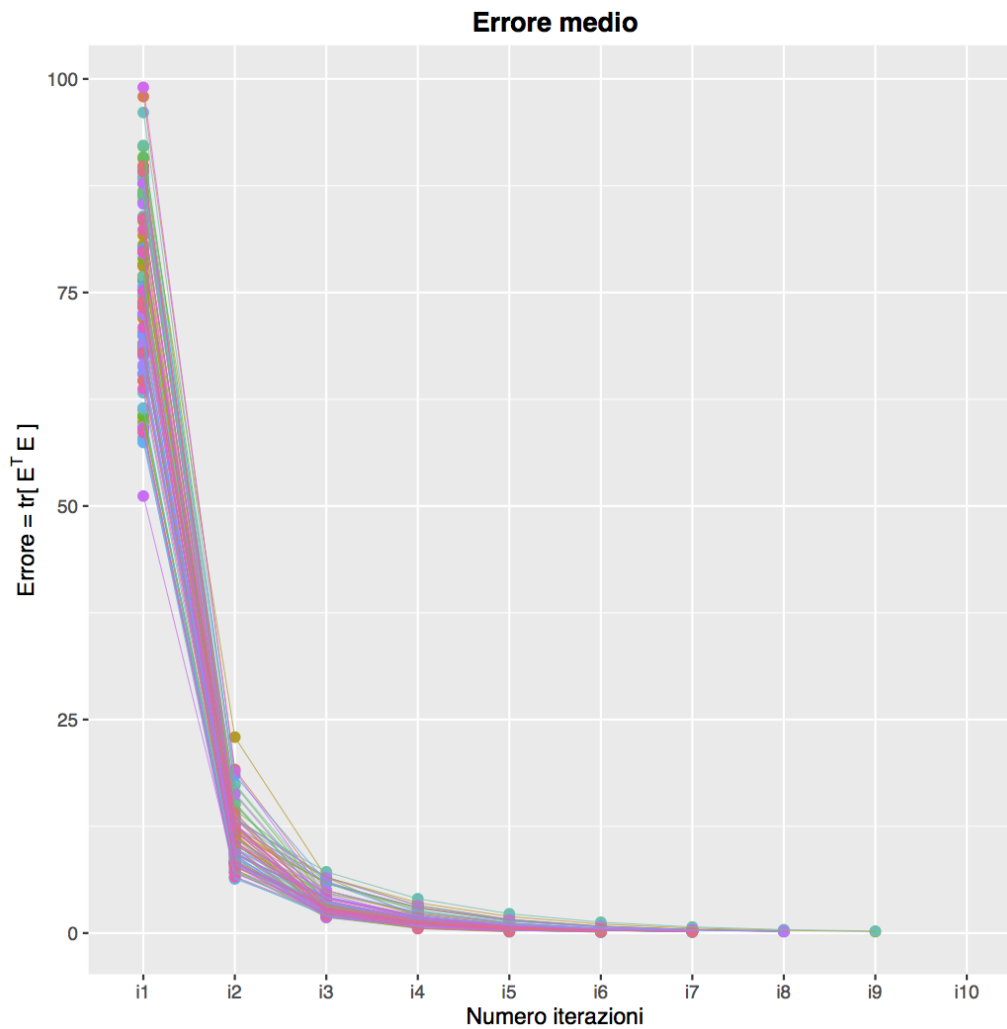


Figura 2.3: In questo grafico ogni singola spezzata rappresenta una delle 100 iterazioni eseguite, ogni singolo punto rappresenta l'errore medio delle tre matrici in funzione all'iterazione eseguita all'interno della funzione procruste.

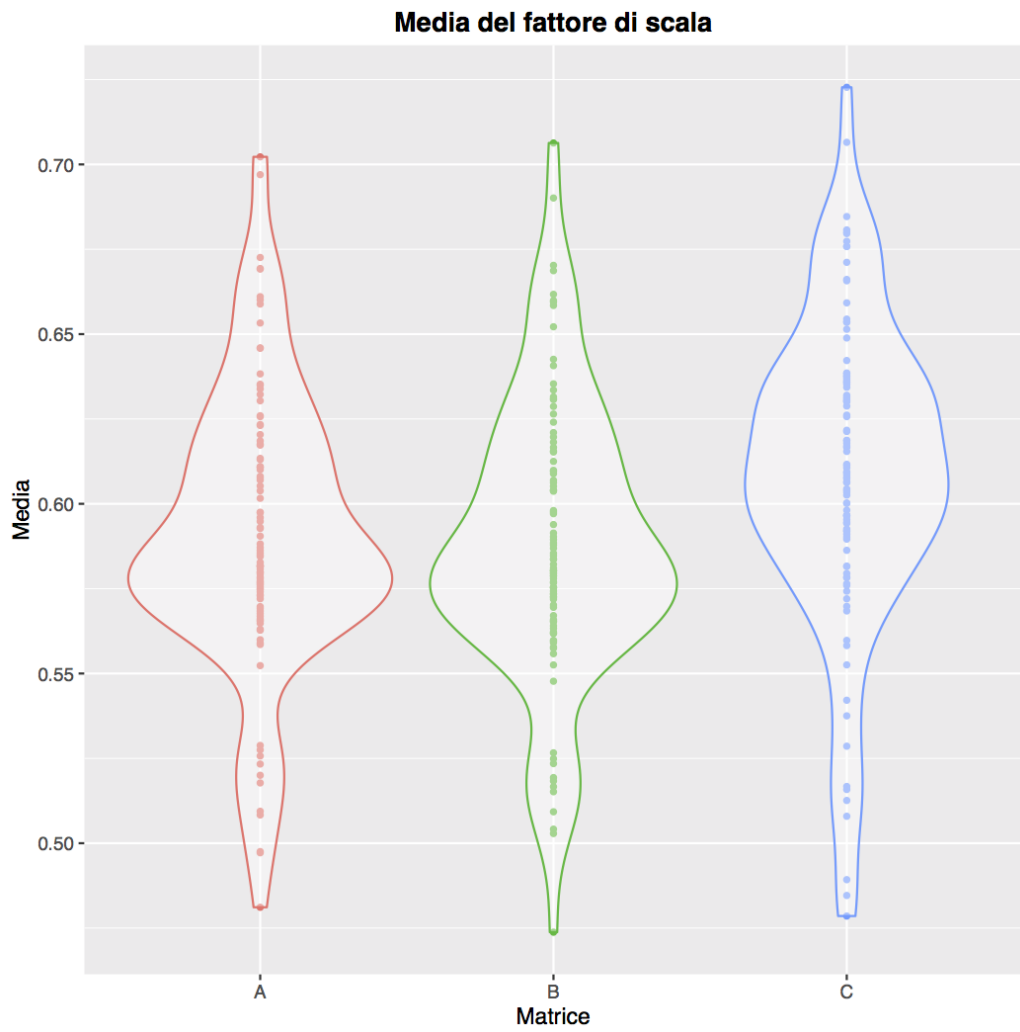


Figura 2.4: Rappresentazione del fattore di scala medio in funzione della matrice associata. Ogni punto è la media degli N fattori di scala calcolati per matrice a ogni singola iterazione

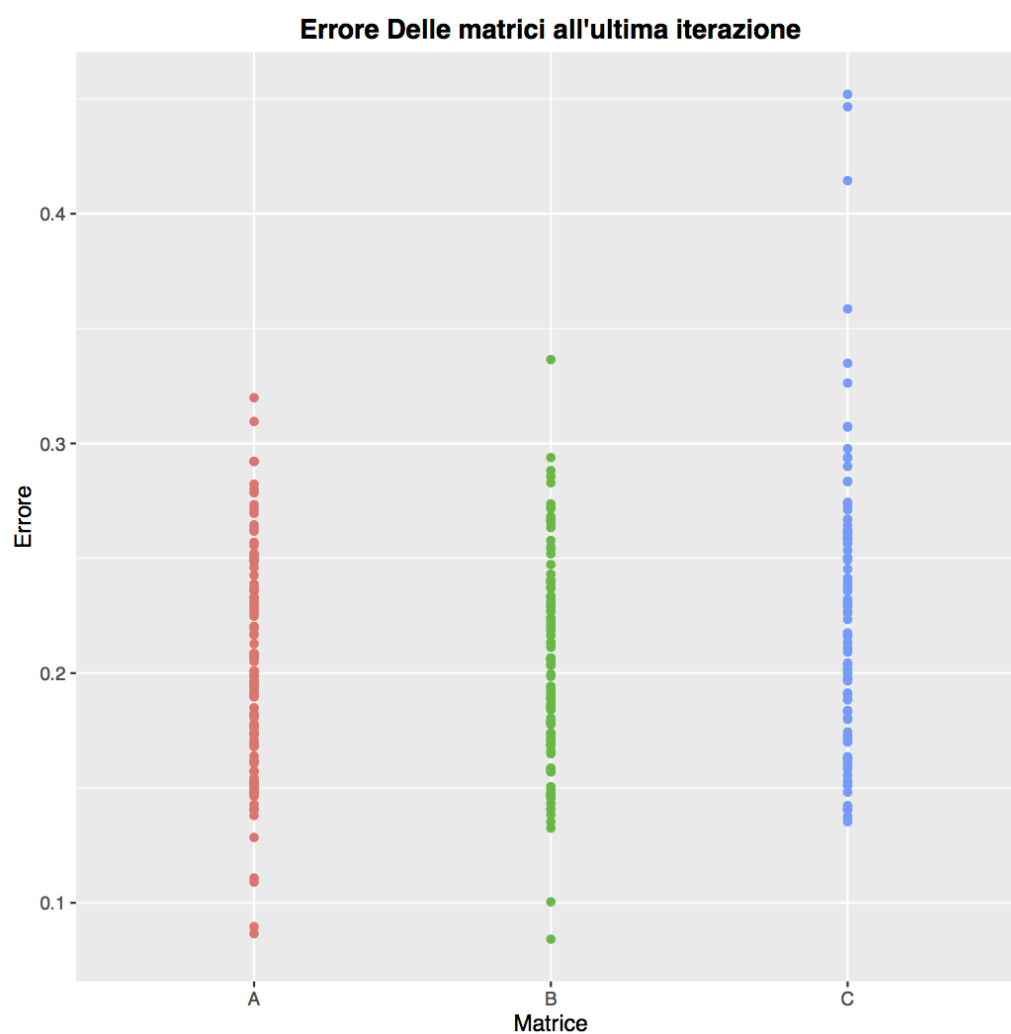


Figura 2.5: Rappresentazione degli errori ottenuti all'ultima iterazione della funzione procruste per ogni matrice

La funzione creata sembra funzionare in particolare nella figura 2.2 possiamo vedere come gli errori diminuiscono indipendentemente dalla matrice di rotazione utilizzata per la generazione dei dati. Si può vedere come la matrice C pur avendo nelle prime iterazioni errori maggiori, rispetto ad A e B , nelle iterazioni successive gli errori tendano ad assumere gli stessi valori. Questo vuol dire che le rotazioni eseguite all'interno della funzione per

allineare i valori funzionali. Altro grafico significativo è 2.4 dove si vedono le medie dei fattori di scala per ogni matrice, si può notare come la media sia molto diversa da uno. La cosa da rilevare è che Haxby utilizza la *OPA*, la quale non tiene conto del fattore di scala. Ciò equivale nel nostro caso ad imporre il vincolo di un fattore di scala sempre uguale a uno. Questo ci porta alla conclusione che il fattore di scala è rilevante. In figura 2.5 sono visibili gli errori ottenuti all'ultima iterazione della funzione di procruste per ogni matrice, si noti come gli errori siano simili anche avendo diverse matrici di rotazioni.

Capitolo 3

Multi-Voxel Pattern Analysis

La Multivariate pattern analysis (MVPA) o detta anche Multivoxel pattern analysis è un metodo sempre più diffuso per analizzare i dati fMRI. La MVPA sta guadagnando sempre maggior importanza nello studio delle neuro-immagini, questo è dovuto al fatto che risulta più sensibile rispetto alle classiche analisi uni-variate. Il metodo viene utilizzato per analizzare l'esperienza percettiva del soggetto prodotta dalla attività neuronale in alcune regioni circoscritte del cervello.

3.1 Esempio

Per comprendere meglio come funziona la MVPA e come differisce dalle analisi convenzionali ci conviene spiegarlo attraverso un semplice esempio. Ad un soggetto vengono presentati due stimoli video differenti, l'immagine di una arancia e l'immagine di una mela. Entrambi gli stimoli, visione di una arancia o di una mela, producono un segnale specifico di attività neurale, questo significa che avremmo i vari voxel¹ con diverse intensità di segnale. Arrivati a questo punto nelle procedure convenzionali si può procedere in due modi:

¹Voxel: Il più piccolo elemento di volume che si riesce a differenziare in una determinata immagine si chiama voxel

- si esegue la media del segnale dei due stimoli, $(\bar{X}_{arancia}, \bar{X}_{mela})$, si esegue la differenza tra le medie se questa risulta significativa si riusciranno a distinguere i due stimoli.
- si stabilisce un contrasto sottrazione per ogni voxel, $x_i^{mela} - x_i^{arancia}$, il livello di attivazione durante la condizione di "mela" è sottratto dal livello di attivazione durante la condizione di "arancio". Come per il caso precedente in caso di differenze significative si riuscirà a distinguere i due differenti stimoli ma in caso contrario i due stimoli non saranno distinguibili.

Il vantaggio di usare la MVPA deriva dal fatto che esso considera tutti i voxel contemporaneamente, ed è quindi possibile rilevare un modello sottostante. Altra differenza sostanziale risulta che i metodi "convenzionali" tipicamente cercano di mostrare una dipendenza statistica tra certi stimoli e le attività neuronali, in altre parole ci si chiede se due stimoli differenti come in questo caso la visione di una arancia e di una mela producono differenti livelli di intensità in una specifica regione cerebrale. Mentre con l'utilizzo della MVPA si vuole rispondere a domande come: sulla base di una attività neuronale in una specifica regione di interesse si può prevedere che stimolo percepisce il soggetto, la possiamo pensare come ad una operazione di decodifica.

Nel seguito mostriamo come il metodo MVPA affronta il problema se lo stimolo di vedere la mela induce una diversa attività neuronale rispetto allo stimolo di vedere una arancia.

1. Vengono letti i segnali fMRI del soggetto mentre vede diversi stimoli tra mela ed arancia.
2. I dati salvati e vengono divisi in due insiemi, training set e test set, tipicamente il training e due o tre volte la numerosità del test.
3. I dati del training vengono classificati attraverso un modello di classificazione, in questo si cerca di rilevare le caratteristiche nei segnali neurali che distinguono i due tipi di stimolo l'uno dall'altro.

4. Una volta scelto il modello lo si testa sul test set e si valuta la sua performance, che in questo caso la possiamo definire in termini di errori di classificazione.

Quindi la MVPA si riduce ad essere quello che in statistica si chiama problema di classificazione, i modelli scelti per le analisi sono stati, modello Lasso, modello albero di classificazione, Random forest, modello MARS e modello reti neurali.

3.2 Pre-elaborazione dati

In questa fase abbiamo eseguito alcune elaborazioni sui dati, dato che la loro struttura originaria era difficilmente analizzabile. Si è scelto di non analizzare ogni singolo voxel ma di raggruppare voxel vicini per crearne uno unico, assegnandoli la media dei voxel che lo costituiscono, come numero di raggruppamento si è scelto $N = 10$. Questo si è fatto principalmente per due motivi: uno legato alla dimensionalità, così facendo abbiamo ridotto il numero di voxel da prendere in considerazione che altrimenti sarebbe stato troppo grande e poi se pensiamo ad una zona del cervello che si attiva è più grande rispetto ad un singolo voxel iniziale. Dato lo scopo del lavoro si è creata una variabile Y , variabile risposta, che indica se il soggetto durante quell'istante temporale è in una fase di stimolo $Y = 1$ o in fase di riposo $Y = 0$, questa è stata poi aggiunta ai nostri dati. La dimensione del data set dopo tutta questa fase è risultata essere $[1680 \times 312]$ dove la prima dimensione indica gli istanti temporali, e la seconda indica voxel dopo averli raggruppati, ricordiamo che i primi quattro volumi e l'ultimo sono stati eliminati per ogni soggetto.

3.3 Modelli

Per verificare se l'analisi di Procrustes eseguita sui dati migliorava o meno la predizione dello stimolo si è eseguito un confronto tra le capacità predittive dei modelli per i due data-set, quelli allineati tramite la analisi di procrustes

e quelli originali e quindi non allineati. Si sono eseguiti diversi modelli di classificazione valutandone la loro bontà, come termine di paragone si è utilizzato l'errore di classificazione e la curva ROC. In ogni modello si è eseguita una fase di training ed una di test. Nella fase di training si procedeva alla creazione del modello, utilizzando come dati solo una parte di tutti di essi, questo insieme di dati viene detto training set (insieme di stima). Nella fase di test si valutava l'errore di classificazione commesso dal modello nella parte nella parte di dati esclusa nella fase precedente che prende il nome di test-set (insieme di verifica). Nei prossimi paragrafi presenteremo i modelli utilizzati, descrivendoli in maniera non approfondita dato che non è lo scopo di questo lavoro, ma saremo più interessati a confrontare i risultati ottenuti. Tutte le analisi sono state svolte con il software statistico R.

3.3.1 Modello MARS

I modelli MARS acronimo di Multivariate Adaptive Regression Spline, fanno parte dei modelli non parametrici, cioè non considerano la funzione che stima il modello appartenente ad una predeterminata classe parametrica di funzioni, ma lascia che sia la stessa funzione ad adattarsi ai dati. Precisiamo che questo non significa che la funzione seguirà perfettamente tutte le osservazioni, perché se questo accadesse si verificherebbe il problema del sovra-adattamento ai dati. Per evitare ciò si vanno ad imporre dei vincoli, fuori dai quali la funzione può liberalmente muoversi. Altro problema che si può verificare molto spesso è causato dal fatto che all'aumentare del numero di dimensioni, la distanza tra le osservazioni cresce in maniera esponenziale, questo aumenta la difficoltà per i modelli di cogliere le relazioni esistenti. Questo problema prende il nome di maledizione della dimensionalità, e nell'analisi dei modelli non si potrà prescindere da tale problema. Modelli come i MARS sono particolarmente adatti quando si lavora con molte variabili dato che forniscono una procedura di selezione automatica delle variabili. Il

Modello MARS ha la seguente forma:

$$f(x) = \beta_o + \sum_{j=1}^J \beta_j h_j(x) \quad (3.3.1.1)$$

dove le h_j rappresentano le diverse funzioni, dette basi. Una volta ottenute le basi i parametri β_j sono stimati minimizzando la somma dei quadrati dei residui. Dato che si tratta di un modello adattivo e quindi andiamo ad aggiungere maniera progressiva nuove funzioni ad ogni passaggio si rischia sovra-adattare il modello ai dati. Per non andare incontro a questo problema bisogna scegliere il valore ottimale J , numero basi da fare entrare nel modello. Un modo per scegliere il parametro J è la convalida incrociata generalizzata (GCV). Per la costruzione di tale modello si è scelto di utilizzare tutte le variabili esplicative a nostra disposizione, il numero di dimensioni sarà elevato, ma la procedura iterativa per la costruzione del modello eseguirà una selezione delle variabili da utilizzare. Durante la fase di stima, si è tenuto conto oltre alle relazioni delle singole variabili con la variabile risposta anche delle possibili iterazioni tra le variabili con la risposta, in figura (3.1) e (3.2) vengono mostrate le funzioni associate alle variabili rispetto alla variabile risposta.

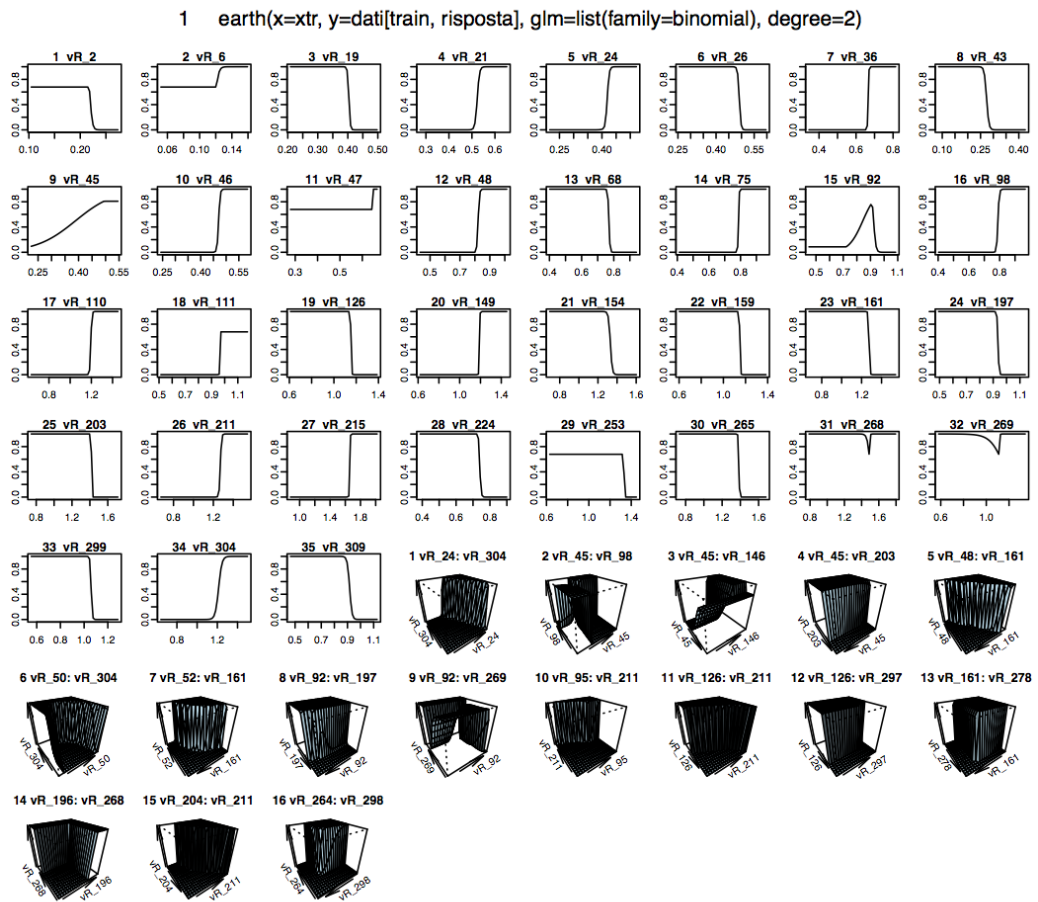


Figura 3.1: (Caso dati allineati) In questo grafico sono riportate le variabili che fanno parte del modello, e la funzione che lega le variabili con la variabile risposta.

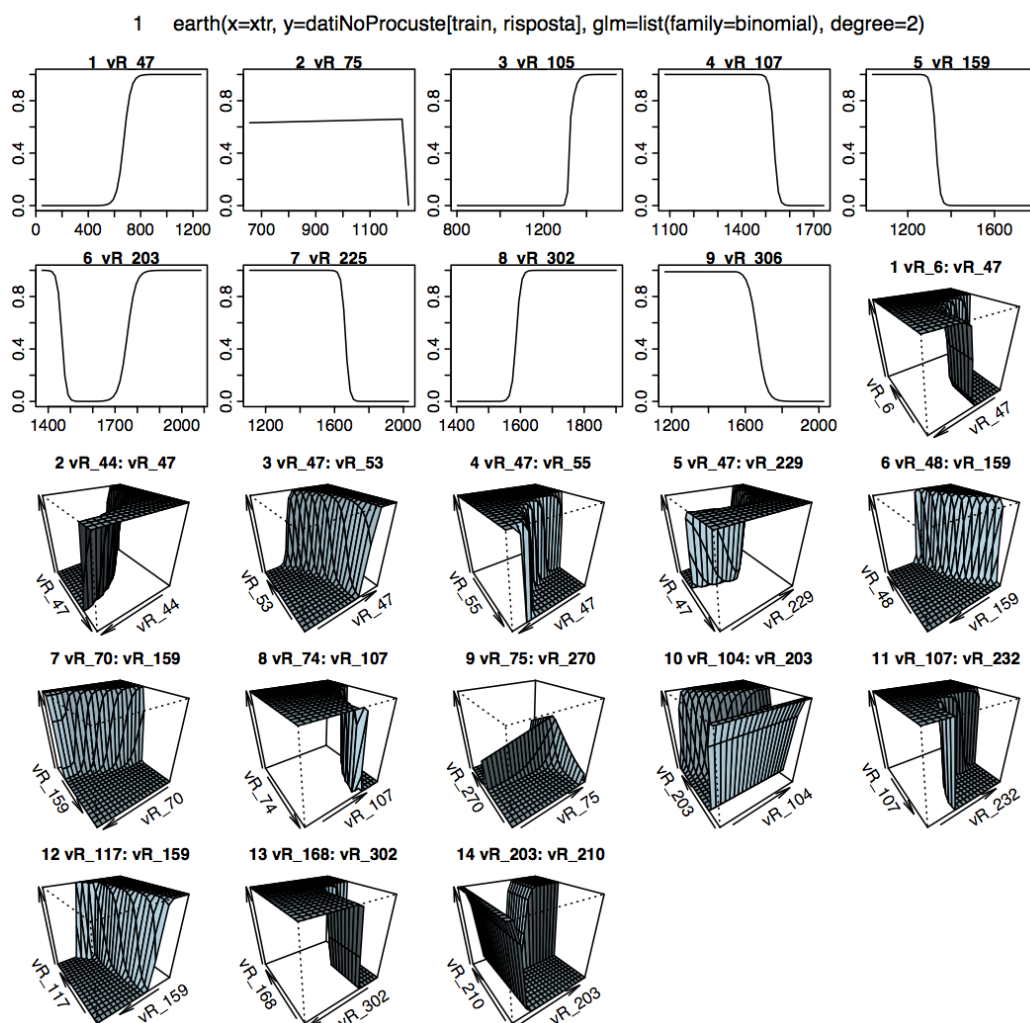


Figura 3.2: (Caso dati non allineati) In questo grafico sono riportate le variabili che fanno parte del modello, e la funzione che lega le variabili con la variabile risposta.

Una volta eseguita l'operazione di stima dei due differenti modelli, abbiamo valutato gli errori sugli insiemi di verifica corrispettivi, ottenendo le tabelle di errata classificazione 3.1 e 3.2 e la loro rappresentazione grafica in figura 3.3 e figura 3.4.

		<i>Verità</i>	
		<i>0</i>	<i>1</i>
<i>Previsione</i>	<i>0</i>	180	25
	<i>1</i>	23	192

Tabella 3.1: Tabella di errata classificazione (caso dati allineati), modello MARS.

		<i>Verità</i>	
		<i>0</i>	<i>1</i>
<i>Previsione</i>	<i>0</i>	171	29
	<i>1</i>	32	188

Tabella 3.2: Tabella di errata classificazione (caso dati non allineati), modello MARS.

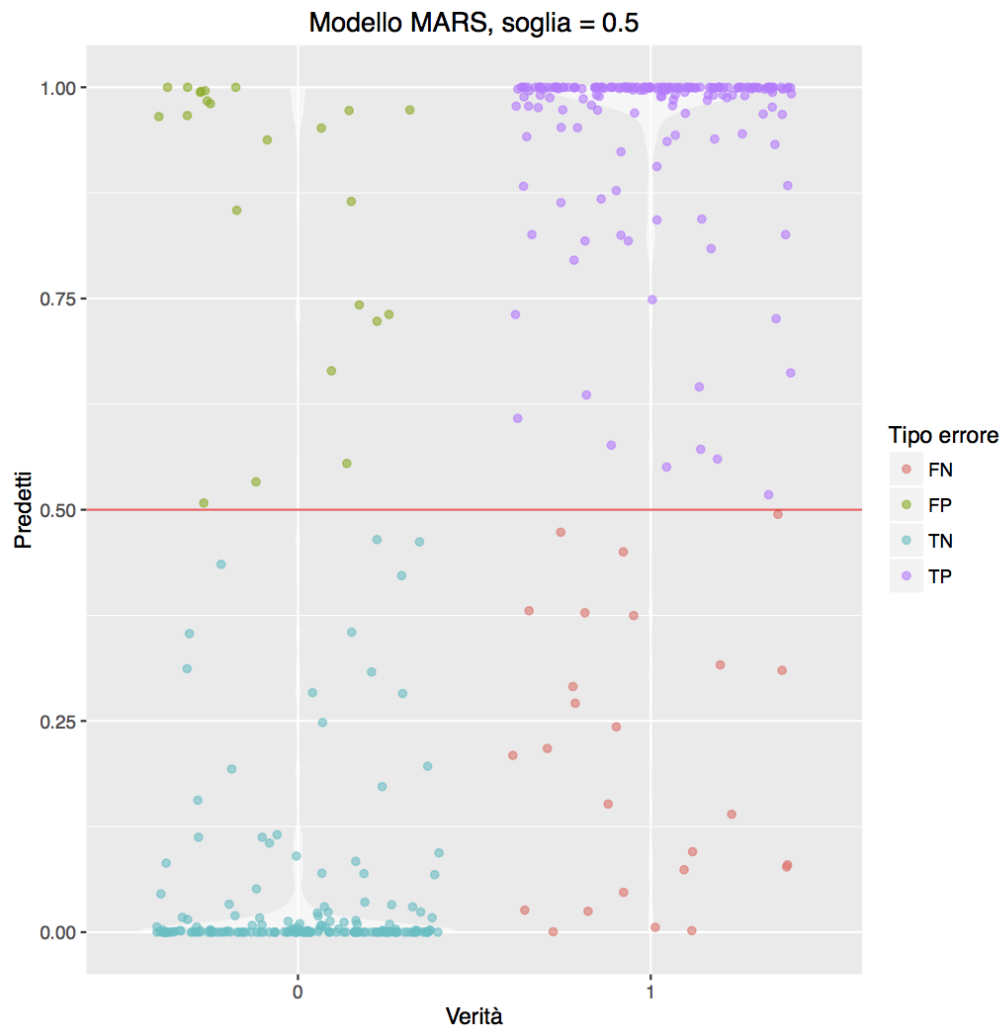


Figura 3.3: Rappresentazione grafica della tabella di errata classificazione (caso dati allineati), modello MARS.

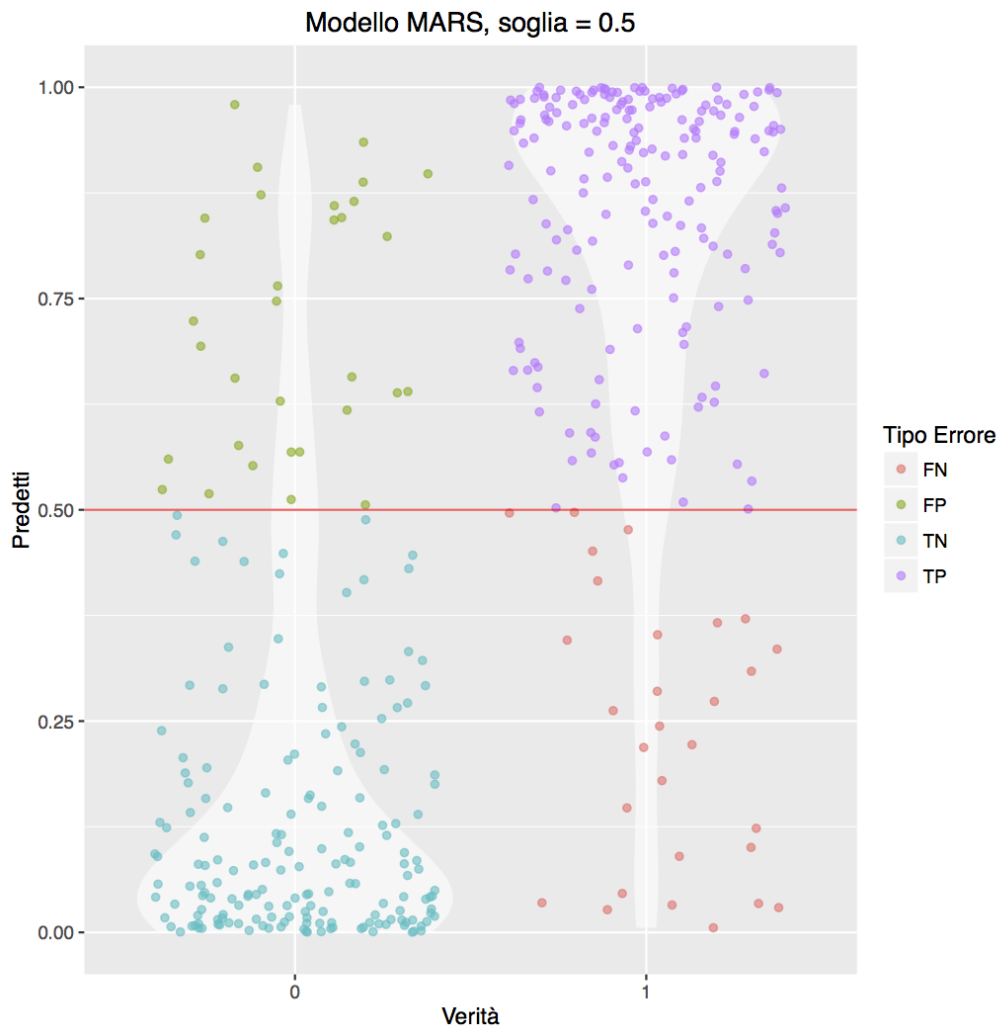


Figura 3.4: Rappresentazione grafica della tabella di errata classificazione (caso dati non allineati), modello MARS.

L'errore commesso dal modello nei rispettivi insiemi di verifica è mostrato nella tabella 3.3.

	Errore percentuale
Modello MARS dati allineati	11.42 %
Modello MARS dati non allineati	14.52%

Tabella 3.3: Errore percentuale del modello MARS.

Guardando gli errori percentuali in tabella 3.3, si vede che eseguendo la procruste si ha una riduzione dell'errore del 21%.

3.3.2 Modello Lasso

Il modello lasso è una versione penalizzata dei minimi quadrati ordinari. Esso minimizza la somma dei quadrati dei residui più un termine di penalizzazione sui parametri del modello, chiamato *L1-penalty*. Le stime lasso sono definite da:

$$\hat{\beta}_{lasso} = \arg \min_{\beta} \sum_{i=1}^N \left(y_i - \beta_0 - \sum_{j=1}^p x_{ij} \beta_j \right)^2 \quad (3.3.2.1)$$

sotto il vincolo

$$\sum_{j=1}^p |\beta_j| \leq t \quad (3.3.2.2)$$

La precedente formula può essere riscritta in forma di Lagrange come:

$$\hat{\beta}_{lasso} = \left[\frac{1}{2} \arg \min_{\beta} \sum_{i=1}^N \left(y_i - \beta_0 - \sum_{j=1}^p x_{ij} \beta_j \right)^2 + \lambda \sum_{j=1}^p |\beta_j| \right] \quad (3.3.2.3)$$

Data la natura del vincolo (3.3.2.2) scelte di t sufficientemente piccole porteranno alcuni coefficienti ad essere esattamente zero, mentre per scelte di t grandi, $t_0 = \sum_1^p |\hat{\beta}_j|$ si otterranno le stime dei minimi quadrati, $\hat{\beta}_j = \hat{\beta}_j^{ls}$. Quindi in base alla scelta di t si ottengono diversi livelli di selezione delle variabili. La scelta di λ nella (3.3.2.3) chiamato parametro di regolazione, ha lo stesso significato di t , influisce sul numero di variabili che entrano nel modello, si vede che per $\lambda = 0$ si ha il modello lineare, mentre all'aumentare

di λ si avranno sempre meno variabili che entreranno a far parte del modello, come si è potuto capire t e λ sono legati da una proporzionalità inversa. Per selezionare il parametro λ si è usata la convalida incrociata, su entrambi gli insiemi di stima. La procedura che si è usata ci fornisce di default due λ differenti; uno λ_{min} che ci fornisce il minor errore di classificazione e λ_{1se} definito come il più grande valore di λ tale per cui l'errore di classificazione è compreso in uno standard error del λ_{min} . Per entrambi i modelli si è scelto di usare λ_{1se} , il motivo che ci ha portato a questa scelta è stato che il numero delle variabili selezionate con λ_{min} è risultato maggiore rispetto alle variabile selezionate con λ_{1se} , con una differenza in termini di errore di classificazione non così diversa. La scelta dei due parametri nei due insiemi è stata: $\lambda_P = 0.0001234098$ e $\lambda_{NP} = 0.0005768794$ nel caso di dati allineati e non allineati. Una volta stimato il modello lo abbiamo valutato sull'insieme di verifica, ottenendo le tabelle di errata classificazione tabella 3.4 e tabella 3.5 ed una rappresentazione grafica 3.5 e 3.6.

		<i>Verità</i>	
		<i>0</i>	<i>1</i>
<i>Previsione</i>	<i>0</i>	183	23
	<i>1</i>	20	194

Tabella 3.4: (dati allineati) Tabella di errata classificazione, modello lasso.

		<i>Verità</i>	
		<i>0</i>	<i>1</i>
<i>Previsione</i>	<i>0</i>	168	39
	<i>1</i>	35	178

Tabella 3.5: (dati non allineati) Tabella di errata classificazione, modello lasso.

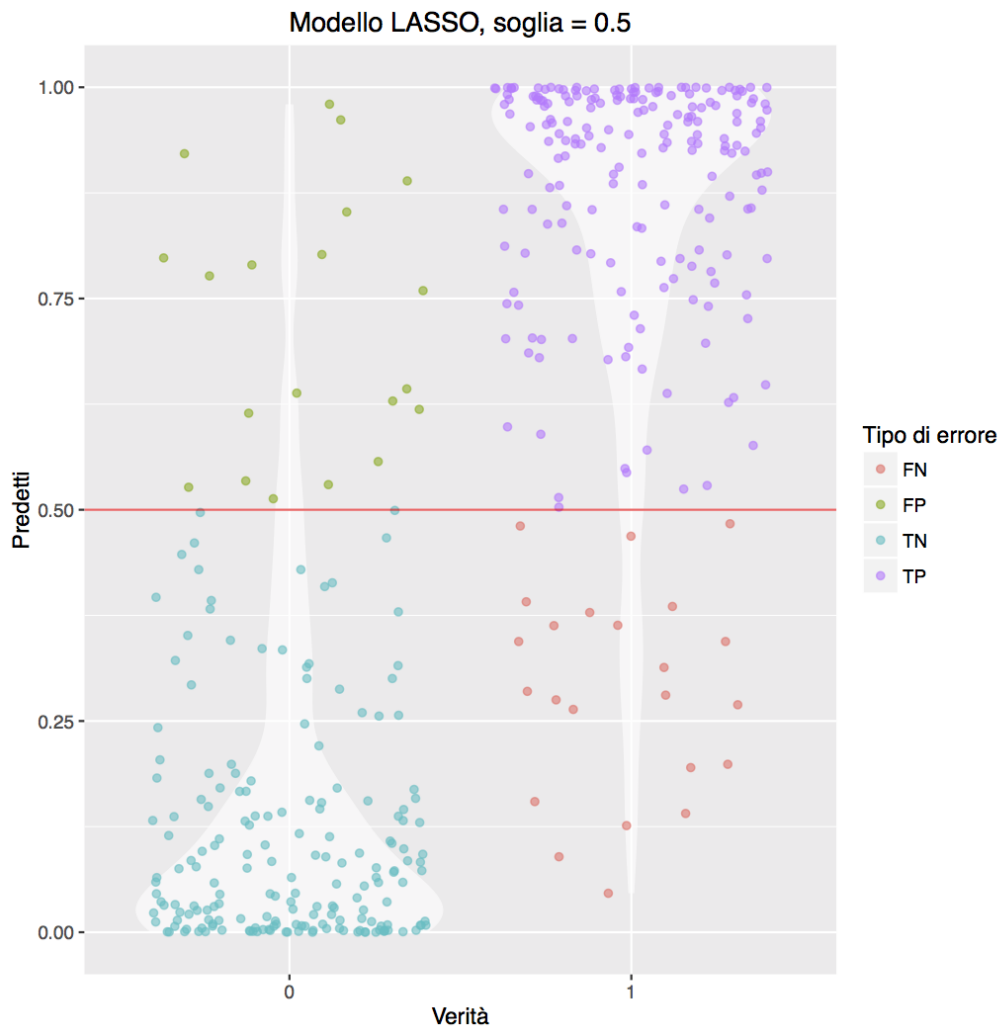


Figura 3.5: (caso dati allineati) Rappresentazione grafica della tabella di errata classificazione, modello lasso.

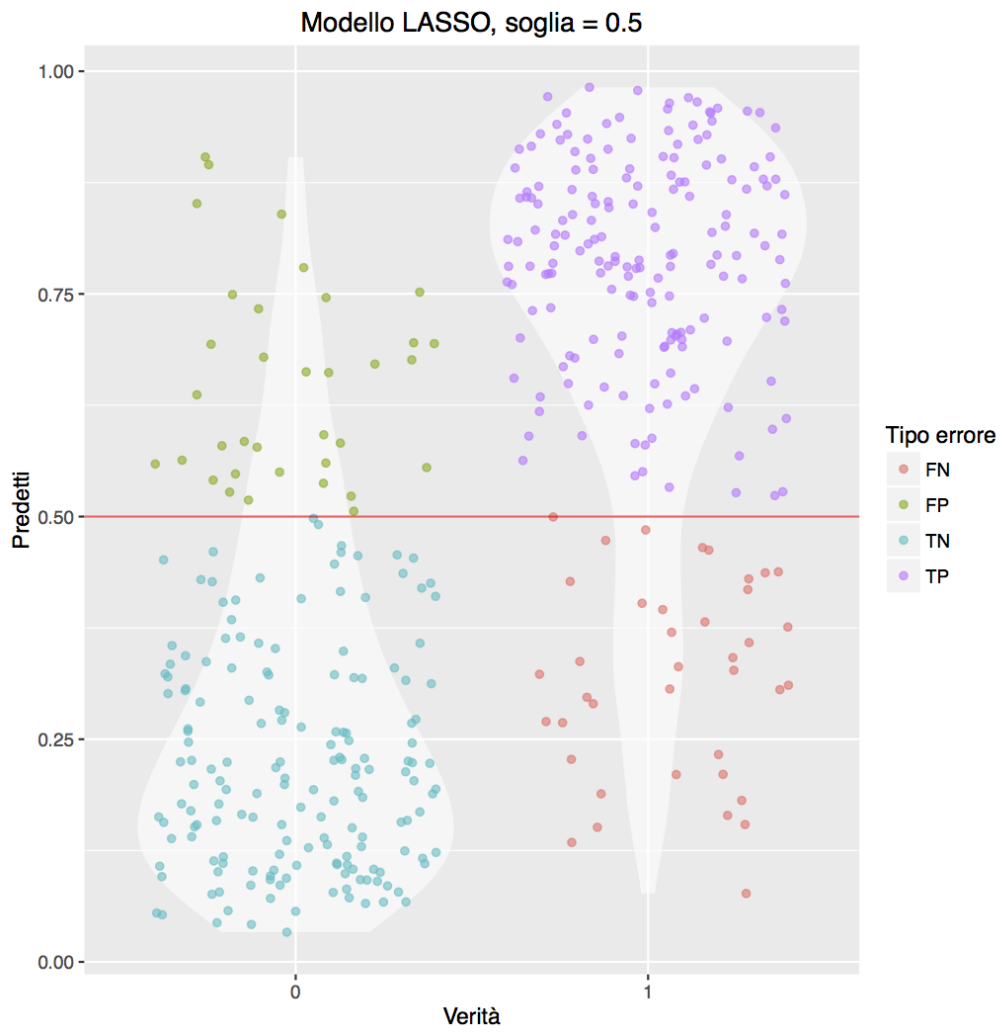


Figura 3.6: (caso dati non allineati) Rappresentazione grafica della tabella di errata classificazione, modello lasso.

L'errore totale commesso dal modello nei due insiemi di verifica è mostrato in tabella 3.6.

	Errore percentuale
Modello Lasso dati allineati	10.23 %
Modello Lasso dati non allineati	17.61 %

Tabella 3.6: Errore percentuale del modello.

Guardando gli errori percentuali in tabella 3.6, si vede che eseguendo la procruste si ha una riduzione dell'errore del 41%.

3.3.3 Modello Albero di classificazione

Uno dei metodi più intuitivi per approssimare la probabilità che un'osservazione appartenga a una classe è quello di utilizzare una funzione a gradini, questa è l'idea alla base degli alberi di classificazione. Questo metodo consente di classificare le osservazioni presenti nel data-set attraverso un apposito algoritmo, che consiste nel descrivere una partizione dello spazio mediante successive divisioni dello spazio stesso questo produce una gerarchia dell'albero, dove i sottoinsiemi di osservazioni vengono chiamati nodi, mentre il risultato finale viene rappresentato dalle foglie, che contengono la classe di appartenenza delle osservazioni. Ogni elemento è classificato seguendo uno dei tanti percorsi lungo l'albero che parte dalla radice ² ad una foglia ³. I percorsi possibili sono rappresentati dai rami ⁴ dell'albero, che forniscono una serie di regole, espresse in funzione delle variabili esplicative. In questi modelli bisogna fare attenzione al numero di foglie finali N , trovare un corretto numero di foglie ci permette di non avere problemi di sovra-adattamento. Per non andare incontro a tale problema si è diviso casualmente l'insieme di stima in due sottoinsiemi: uno su cui verrà costruito il modello ($T1$) ed uno su cui l'albero verrà potato ($T2$). Nella scelta di N abbiamo eseguito la procedura descritta sopra scegliendo quello che minimizzava l'errore di

²Nodo radice: estremo superiore dell'albero.

³Nodo foglia: sono i nodo che sono nella parte inferiore dell'albero.

⁴Rami: vengono utilizzati per collegare nodi diversi.

classificazione, abbiamo ottenuto che il numero ottimale di nodi è risultato essere $N_P = 47$ e $N_{NP} = 40$ come mostrano le figure 3.7 e 3.8, nel caso di dati allineati e non allineati.

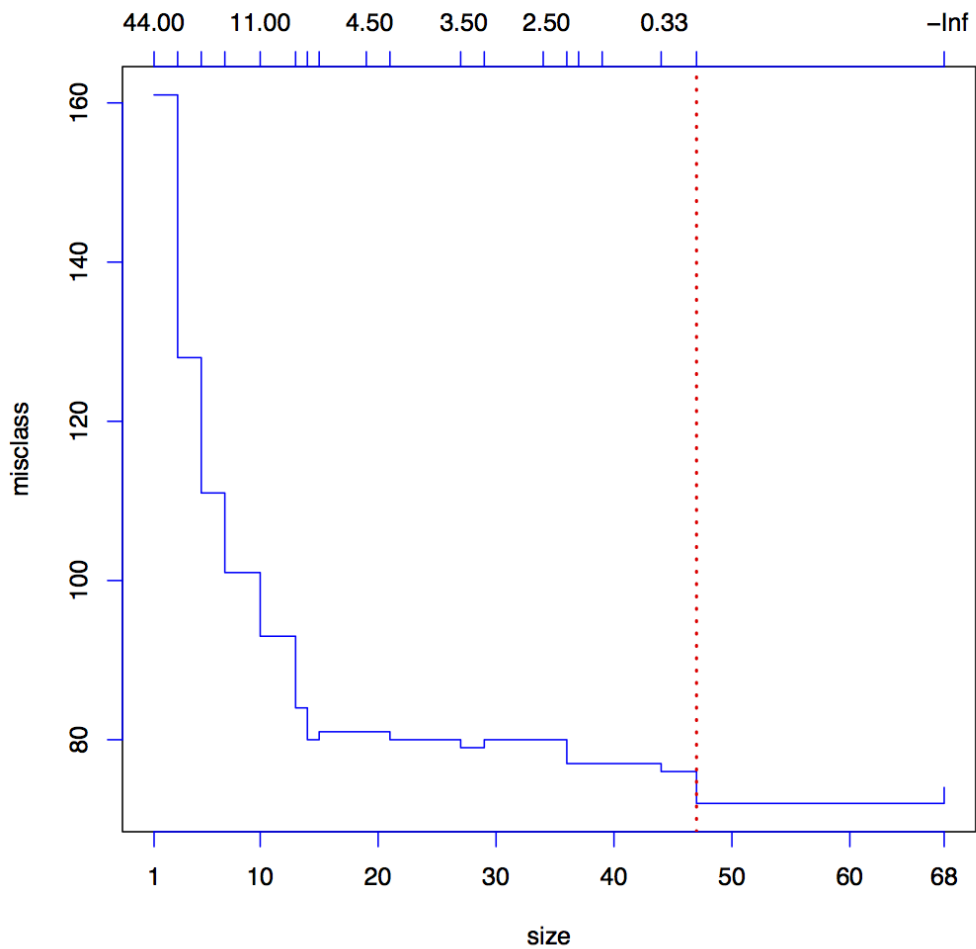


Figura 3.7: (Caso dati allineati) In questo grafico viene mostrato l'errore di classificazione commesso dal modello, sui dati T2 in funzione del numero di nodi. La retta di color rosso tratteggiata rappresenta il numero di nodi che riduce al minimo gli errori di classificazione in T2, che corrisponde a $N = 47$.

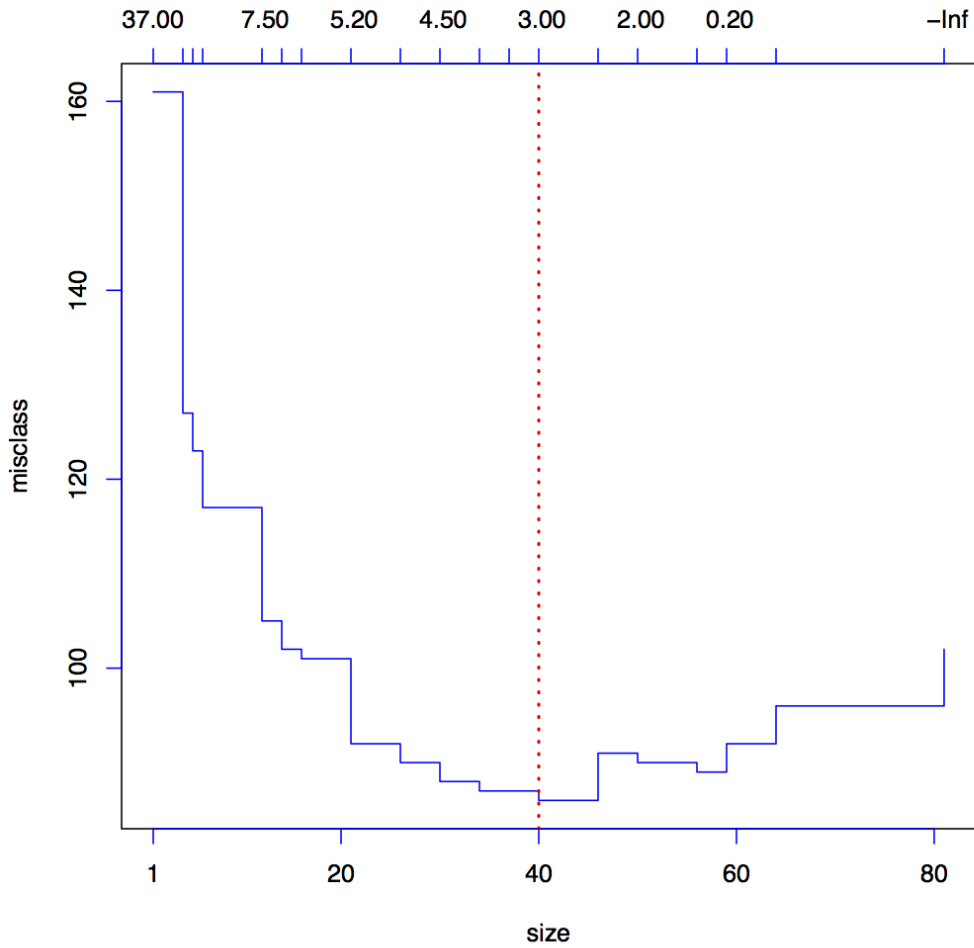


Figura 3.8: (Caso dati non allineati) In questo grafico viene mostrato l'errore di classificazione commesso dal modello, sui dati T2 in funzione del numero di nodi. La retta di color rosso tratteggiata rappresenta il numero di nodi che riduce al minimo gli errori di classificazione in T2, che corrisponde a $N = 40$.

Una volta calcolati N_P e N_{NP} , abbiamo costruito l'albero completo ⁵ sull'insieme di stima lo si è potato con utilizzando il numero di nodi ottimali calcolati precedentemente, infine abbiamo valutato ogni singolo modello nel suo insieme di stima corrispettivo, ottenendo le rispettive tabelle di errata classificazione, tabella 3.7 e tabella 3.8 e le loro rappresentazioni grafiche in figura 3.9 e figura 3.10.

		<i>Verità</i>	
		<i>0</i>	<i>1</i>
<i>Previsione</i>	<i>0</i>	168	39
	<i>1</i>	35	178

Tabella 3.7: (caso dati allineati) Tabella di errata classificazione, modello albero di classificazione.

		<i>Verità</i>	
		<i>0</i>	<i>1</i>
<i>Previsione</i>	<i>0</i>	153	66
	<i>1</i>	50	151

Tabella 3.8: (caso dati non allineati) Tabella di errata classificazione, modello albero di classificazione.

⁵Numero di foglie uguale al numero di osservazioni

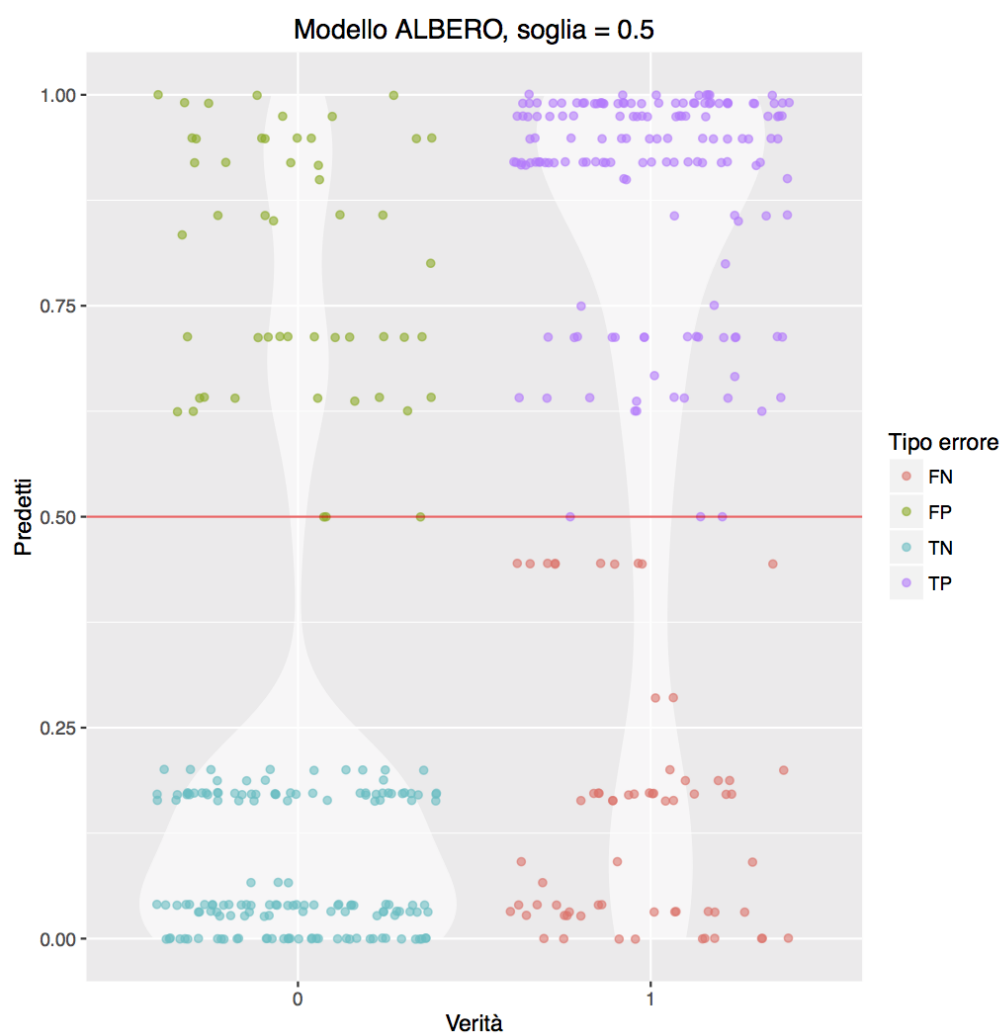


Figura 3.9: (Caso dati allineati) Rappresentazione grafica della tabella di errata classificazione, modello albero di classificazione.

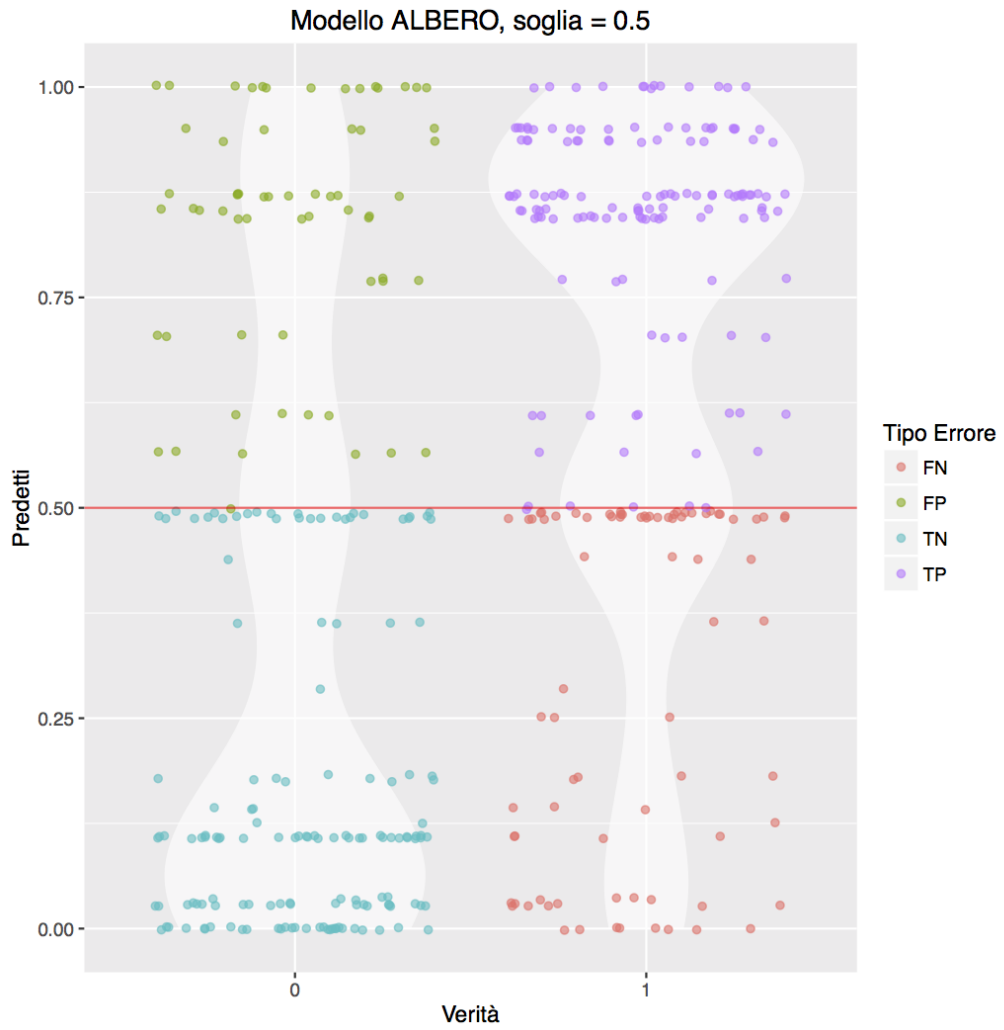


Figura 3.10: (Caso dati non allineati) Rappresentazione grafica della tabella di errata classificazione, modello albero.

L'errore totale commesso dal modello nei due insiemi di verifica è mostrato in tabella 3.9.

	Errore percentuale
Modello Albero dati allineati	27.61 %
Modello Albero dati non allineati	34.04 %

Tabella 3.9: Errore percentuale del modello.

Guardando gli errori percentuali in tabella 3.9, si vede che eseguendo la procruste si ha una riduzione dell'errore del 18%.

3.3.4 Modello Random Forest

Le foreste casuali sono una tecnica di classificazione utilizzate per migliorare le prestazioni di un classificatore semplice⁶. Se si decide di utilizzare come classificatore un albero si fa riferimento alle combinazioni di alberi ottenuti utilizzando la selezione casuale delle variabili esplicative. La procedura consiste nel selezionare in modo casuale, ad ogni nodo di un albero, un sottoinsieme di variabili esplicative di grandezza K , costante in tutti i nodi, che verranno poi analizzate per trovare lo “split” ottimale. L'albero viene fatto crescere fino alla grandezza massima e non viene potato come un normale albero di classificazione, sarà infatti l'operazione di combinazione dei diversi alberi che permetterà di evitare problemi di sovra adattamento. Questa procedura viene applicata per tutti i B alberi della foresta. Ciascun albero viene fatto crescere su un campione bootstrap diverso utilizzando per ciascun nodo un numero K di variabili selezionate casualmente. Nel nostro caso si sono impostati i seguenti parametri $B = 500$, mentre per la scelta di $K_P = 4$ e $K_{NP} = 4$, nel caso di dati allineati e non allineati, come possiamo vedere nelle figure 3.11 e 3.12. Questi K si sono calcolati utilizzando gli errori di classificazione *out-of-bag* (OOB)⁷

⁶Per classificatore semplice intendiamo un singolo modello di classificazione.

⁷L'utilizzo di campioni bootstrap permette di utilizzare la tecnica del *out-of-bag*. In ciascun campione bootstrap una parte dei dati rimane fuori dall'insieme di stima, quindi la parte esclusa può essere utilizzata come insieme di verifica stimando l'errore di classificazione su questi dati.

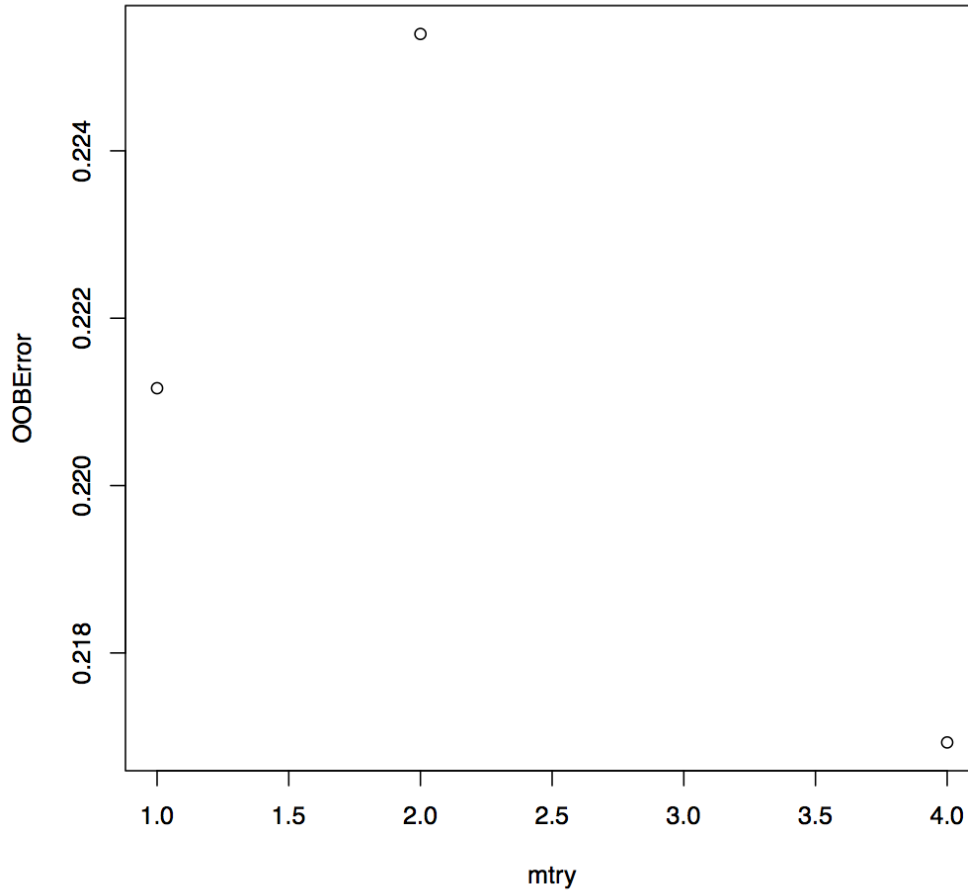


Figura 3.11: (Caso dati allineati) Questo grafico mostra l'errore OOB in funzione del numero di variabili che usiamo nei nodi, in questo caso il numero di variabili che minimizza l'errore OOB è risultato quattro.

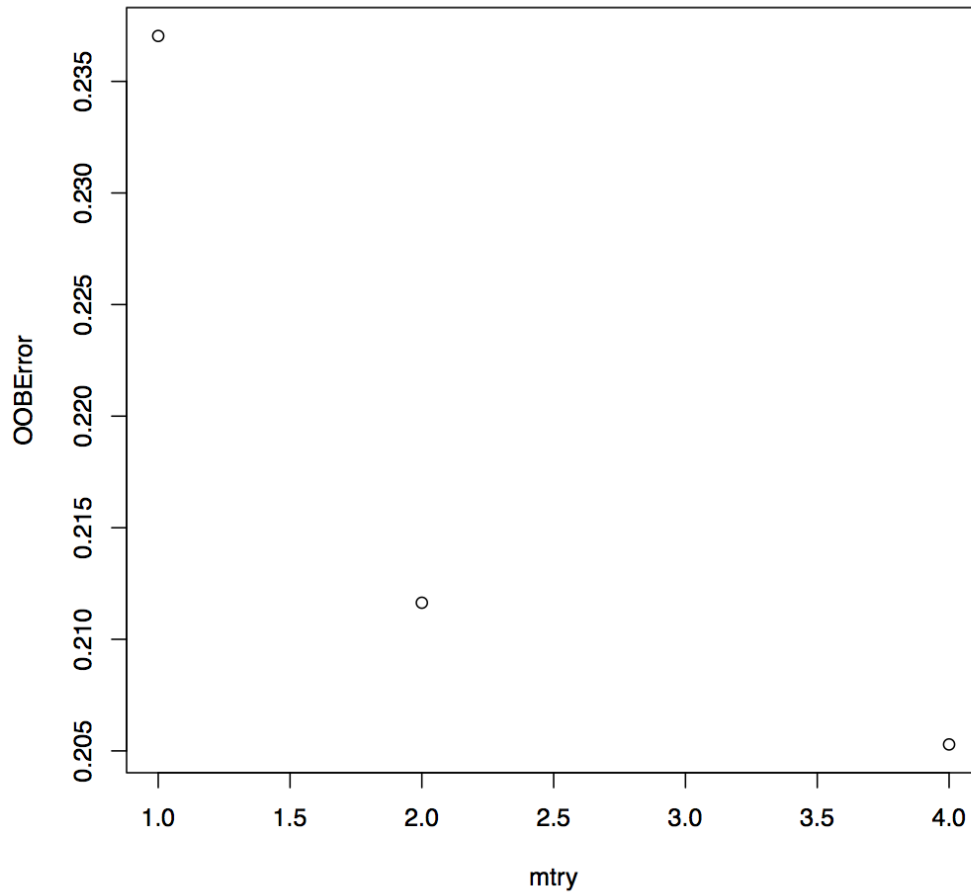


Figura 3.12: (Caso dati allineati) Questo grafico mostra l'errore OOB in funzione del numero di variabili che usiamo nei nodi, in questo caso il numero di variabili che minimizza l'errore OOB è risultato quattro.

Dalla figura 3.13 e 3.14 possiamo vedere l'importanza delle variabili ai fini della classificazione nei due differenti set di dati.

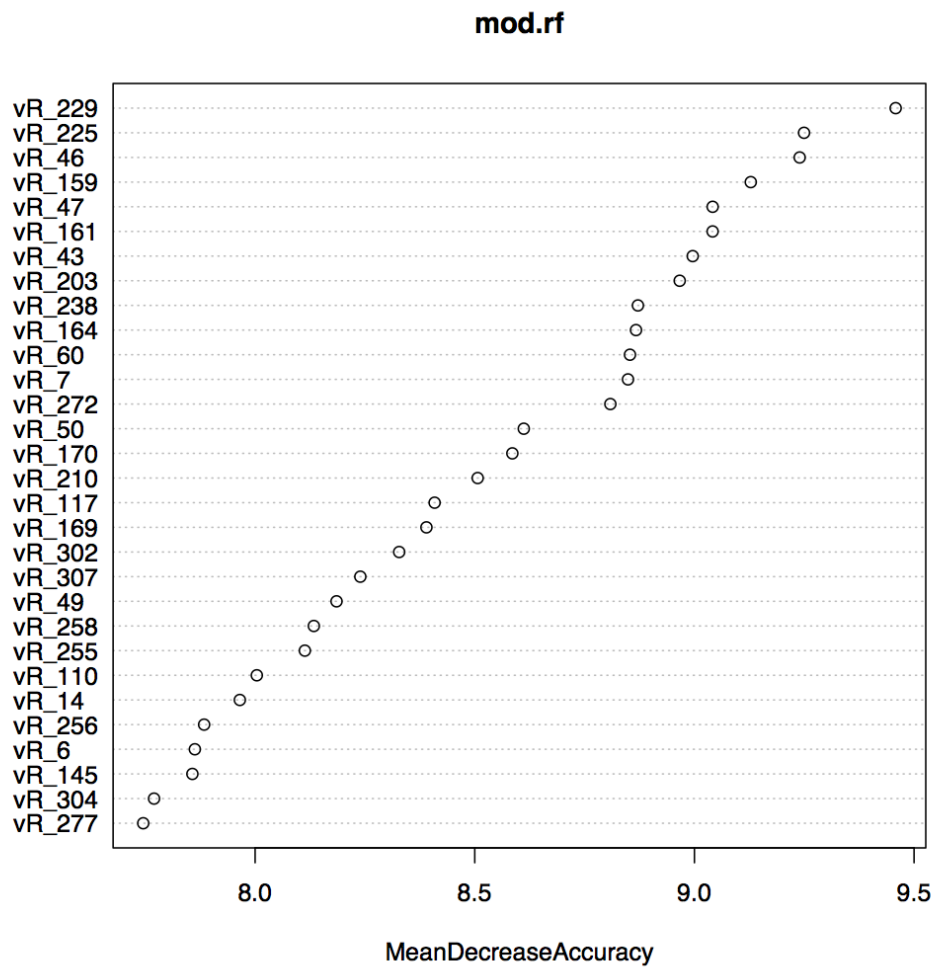


Figura 3.13: (Caso dati allineati) Nel grafico vediamo l'importanza della variabile in termini di decremento medio di accuratezza, (ad esempio eliminando la variabile che risulta essere più importante ai fini della classificazione vR_{229} , commetto circa 9.5 errori in più in media).

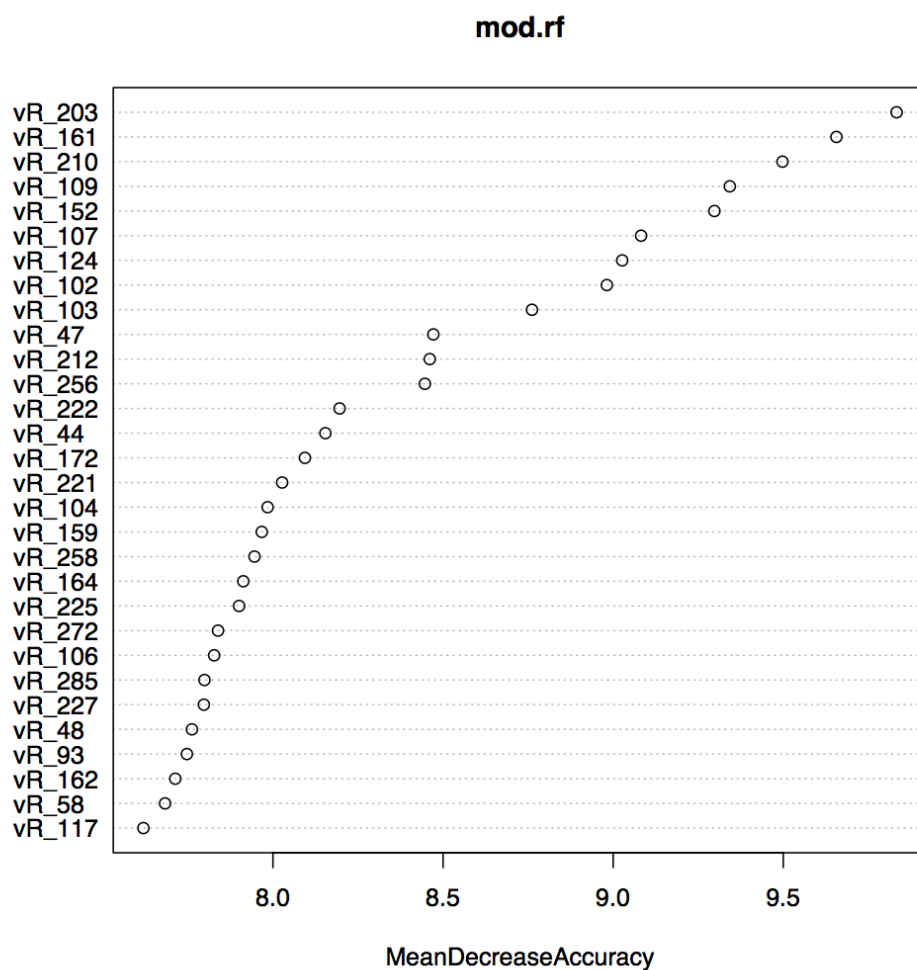


Figura 3.14: (Caso dati non allineati) Nel grafico vediamo l'importanza della variabile in termini di decremento medio di accuratezza, (ad esempio eliminando la variabile che risulta essere più importante ai fini della classificazione vR_{203} , commetto circa 10 errori in più in media).

Una volta eseguita l'operazione di stima dei due differenti modelli, abbiamo valutato gli errori sugli insiemi di verifica corrispettivi, ottenendo le tabelle di errata classificazione tabella 3.10 e tabella 3.11 e la loro rappresentazione grafica in figura 3.15 e figura 3.16.

		<i>Verità</i>	
		<i>0</i>	<i>1</i>
<i>Previsione</i>	<i>0</i>	171	35
	<i>1</i>	32	182

Tabella 3.10: (caso dati allineati) Tabella di errata classificazione, modello Random forest.

		<i>Verità</i>	
		<i>0</i>	<i>1</i>
<i>Previsione</i>	<i>0</i>	174	44
	<i>1</i>	29	173

Tabella 3.11: (caso dati non allineati) Tabella di errata classificazione, modello Random forest.

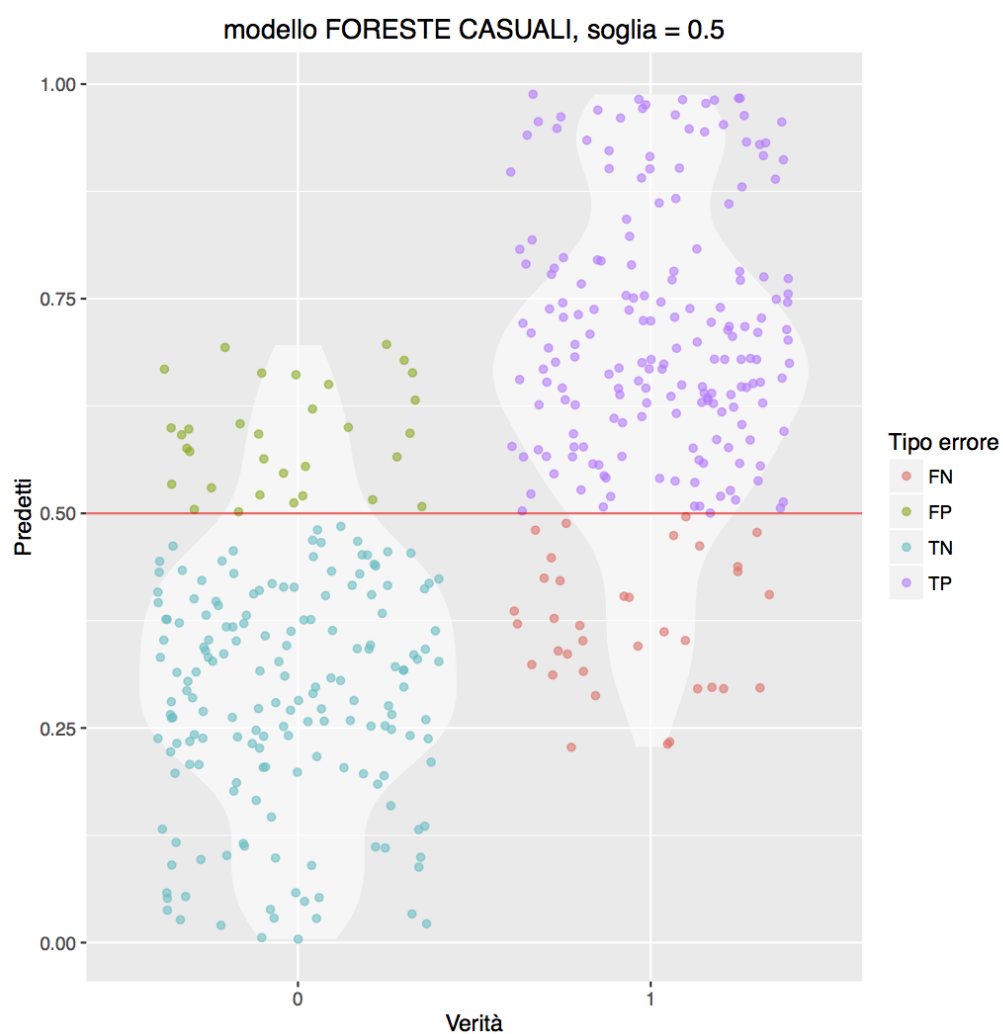


Figura 3.15: (Caso dati allineati) Rappresentazione grafica della tabella di errata classificazione, modello Random forest .



Figura 3.16: (Caso dati non allineati) Rappresentazione grafica della tabella di errata classificazione, modello Random forest.

L'errore commesso dal modello nei rispettivi insiemi di verifica è mostrato in tabella 3.12.

	Errore percentuale
Modello foreste casuali dati allineati	15.95 %
Modello foreste casuali dati non allineati	17.38%

Tabella 3.12: Errore percentuale del modello.

Guardando gli errori percentuali in tabella 3.12, si vede che eseguendo la procruste si ha una riduzione dell'errore del 8%.

3.3.5 Modello Rete Neurale

Una rete neurale è uno schema di regressione a due stadi, dove sono messe in relazione le p variabili esplicative (input) con q variabili risposta (output), attraverso uno strato latente di w variabili. In sostanza abbiamo che le p variabili di input influenzano le w variabili dello strato latente che a loro volta influenzano le q variabili di output. Indichiamo con x_h la generica variabile di input, z_j la generica variabile latente ed con y_k la generica variabile di output, e poniamo pesi $x_0 = 1$ e $z_0 = 1$, la relazione tra i diversi strati è espressa come:

$$z_j = f_0 \left(\sum_{k \rightarrow j} \alpha_{kj} x_k \right) \quad (3.3.5.1)$$

$$y_k = f_1 \left(\sum_{j \rightarrow k} \beta_{jk} z_j \right) \quad (3.3.5.2)$$

dove i parametri α_{kj} e β_{jk} sono da stimare. Le funzioni f_0 e f_1 sono dette funzioni di attivazione e sono poste uguali alla funzione logistica. Prima di procedere alla costruzione del modello, si deve impostare il parametro w , cioè quante variabili vogliamo all'interno dello strato latente, non esiste un criterio per trovare in numero ottimale w , la cosa più semplice è stimare differenti modelli ognuno con un w diverso e confrontare i risultati. I parametri (α, β) sono stimanti attraverso la minimizzazione di una opportuna funzione

obiettivo:

$$D = d + \lambda J(\alpha, \beta) \quad (3.3.5.3)$$

dove d risulta essere l'entropia, λ è un parametro di regolazione positivo compreso tra 10^{-3} e 10^{-1} e $J(\alpha, \beta)$ è una funzione di penalizzazione, tutto questo serve per evitare il problema del sovra-adattamento. La funzione di penalizzazione da noi utilizzata è il weight decay:

$$J(\alpha, \beta) = \|\alpha\|^2 + \|\beta\|^2 \quad (3.3.5.4)$$

Il numero di unità w nello strato latente e il *weight decay*, come si è visto, devono essere impostati dall'analista, per ottenere i valori ottimali sono state stimate più reti neurali con valori differenti di w unità nello strato latente e di *weight decay*, ed è stato scelto il miglior modello in termini di minor errore di classificazione, ottenendo $w = 2$ e *weight decay* = 10^{-2} per entrambi i modelli. Fissati i parametri si sono stimati i modelli nei rispettivi insiemi di stima, e si sono stati valutati nell'insieme di test, si sono ottenute le tabelle di classificazione tabella 3.13 e tabella 3.14 e le loro rappresentazioni grafiche figura 3.17 e figura 3.18.

		<i>Verità</i>	
		<i>0</i>	<i>1</i>
<i>Previsione</i>	<i>0</i>	185	21
	<i>1</i>	18	196

Tabella 3.13: (caso dati allineati) Tabella di errata classificazione, modello Rete neurale.

		<i>Verità</i>	
		<i>0</i>	<i>1</i>
<i>Previsione</i>	<i>0</i>	113	141
	<i>1</i>	90	76

Tabella 3.14: (caso dati non allineati) Tabella di errata classificazione, modello Rete neurale.

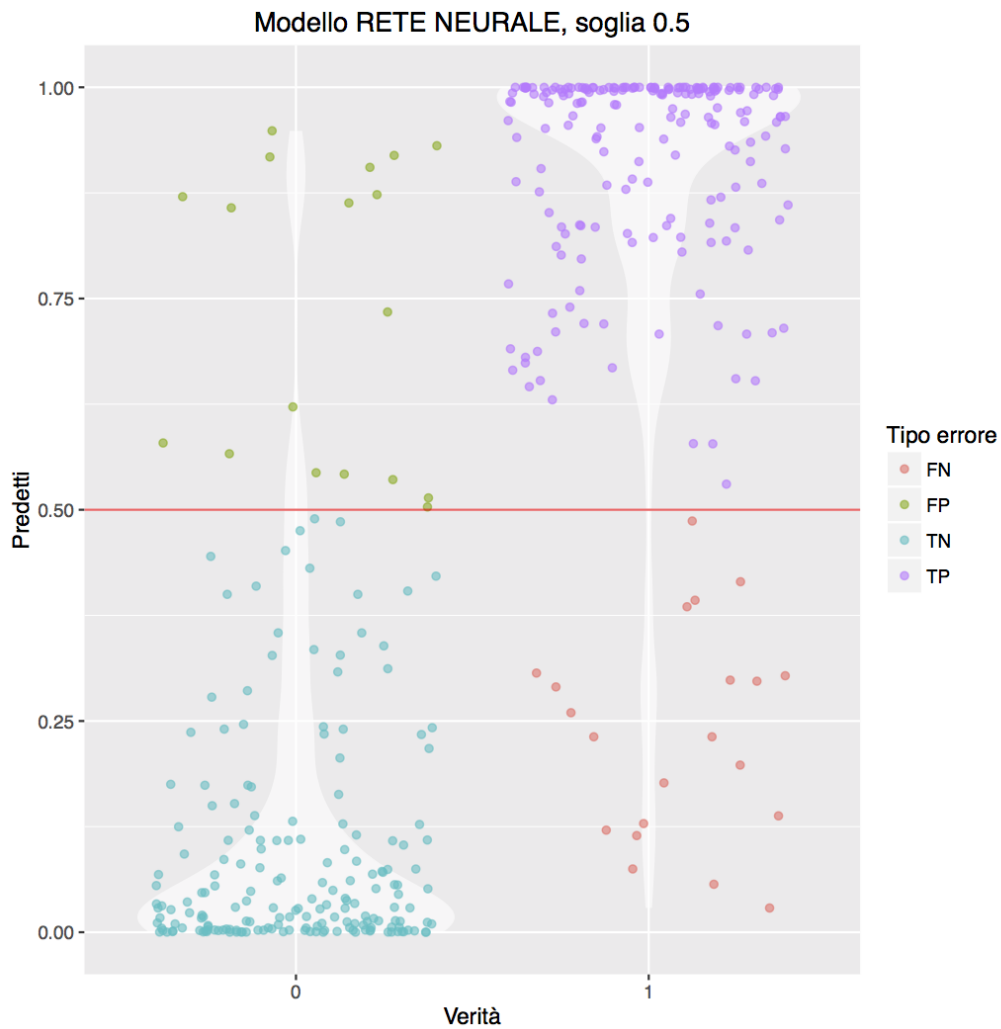


Figura 3.17: (caso dati allineati) Rappresentazione grafica della tabella di errata classificazione, modello Rete neurale.

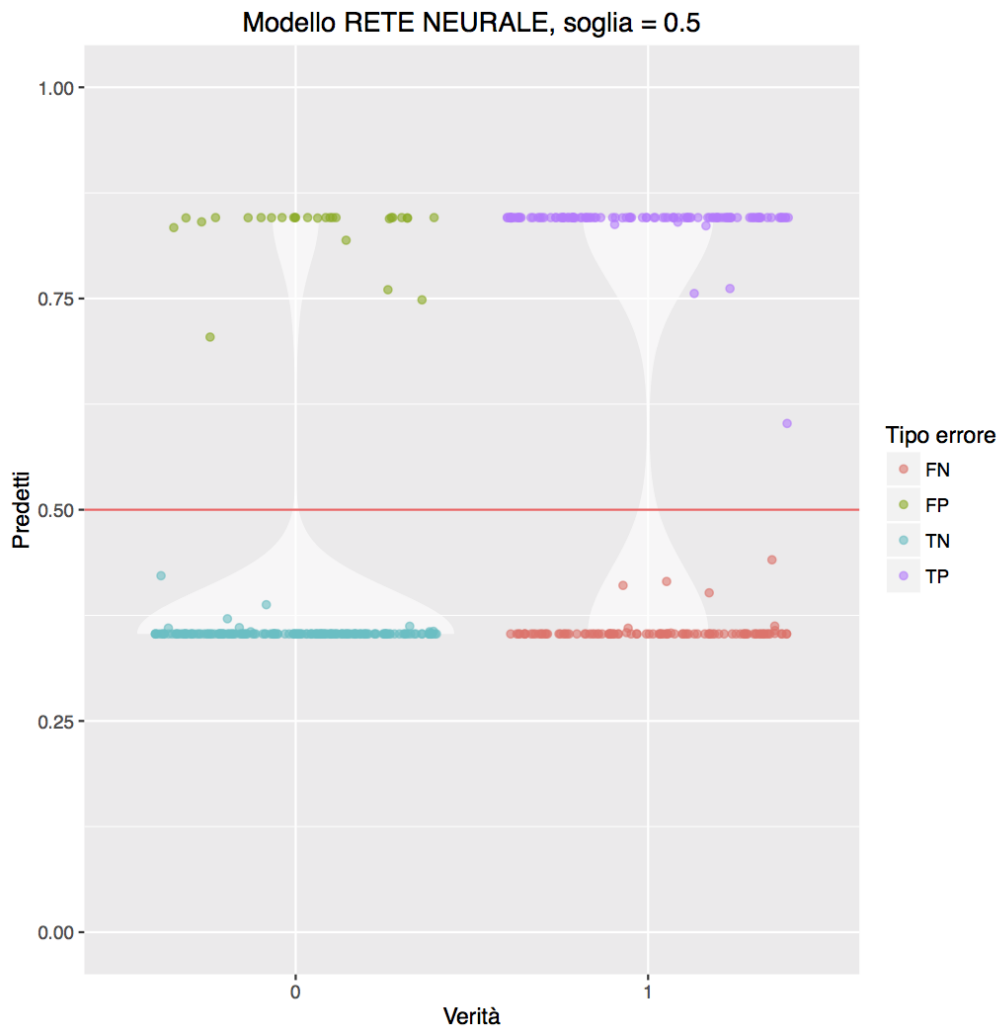


Figura 3.18: (caso dati non allineati) Rappresentazione grafica della tabella di errata classificazione, modello Rete neurale.

L'errore commesso dal modello nei rispettivi insiemi di verifica è mostrato in tabella 3.15.

	Errore percentuale
Modello rete neurale dati allineati	9.28 %
Modello rete neurale dati non allineati	31.42%

Tabella 3.15: Errore percentuale del modello.

Guardando gli errori percentuali in tabella 3.15, si vede che eseguendo la procruste si ha una riduzione dell'errore del 70%.

Capitolo 4

Conclusioni

I risultati ottenuti sono stati confrontati, rappresentando gli errori percentuali dei modelli nei due differenti set di dati come si vede in figura 4.2 e le curve roc rappresentate in figura 4.1.

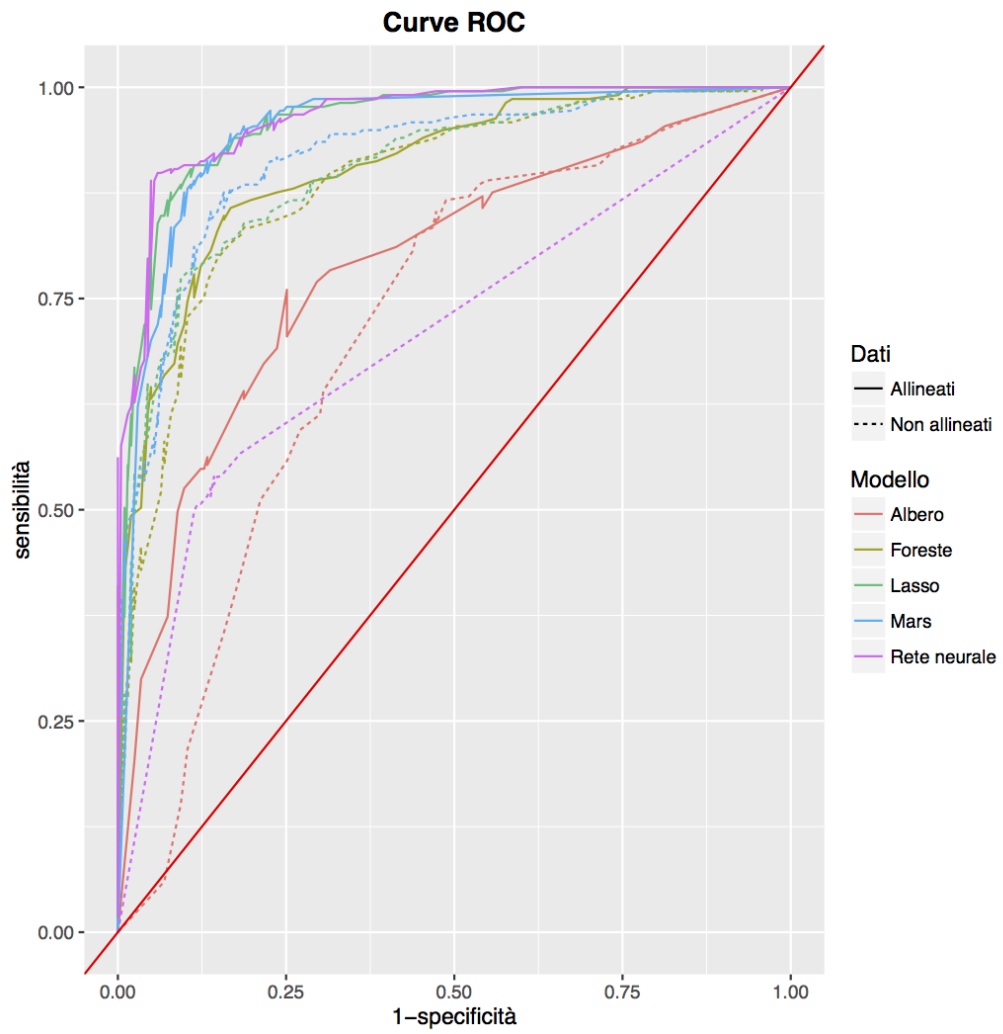


Figura 4.1: Rappresentazione delle curve ROC dei modelli.

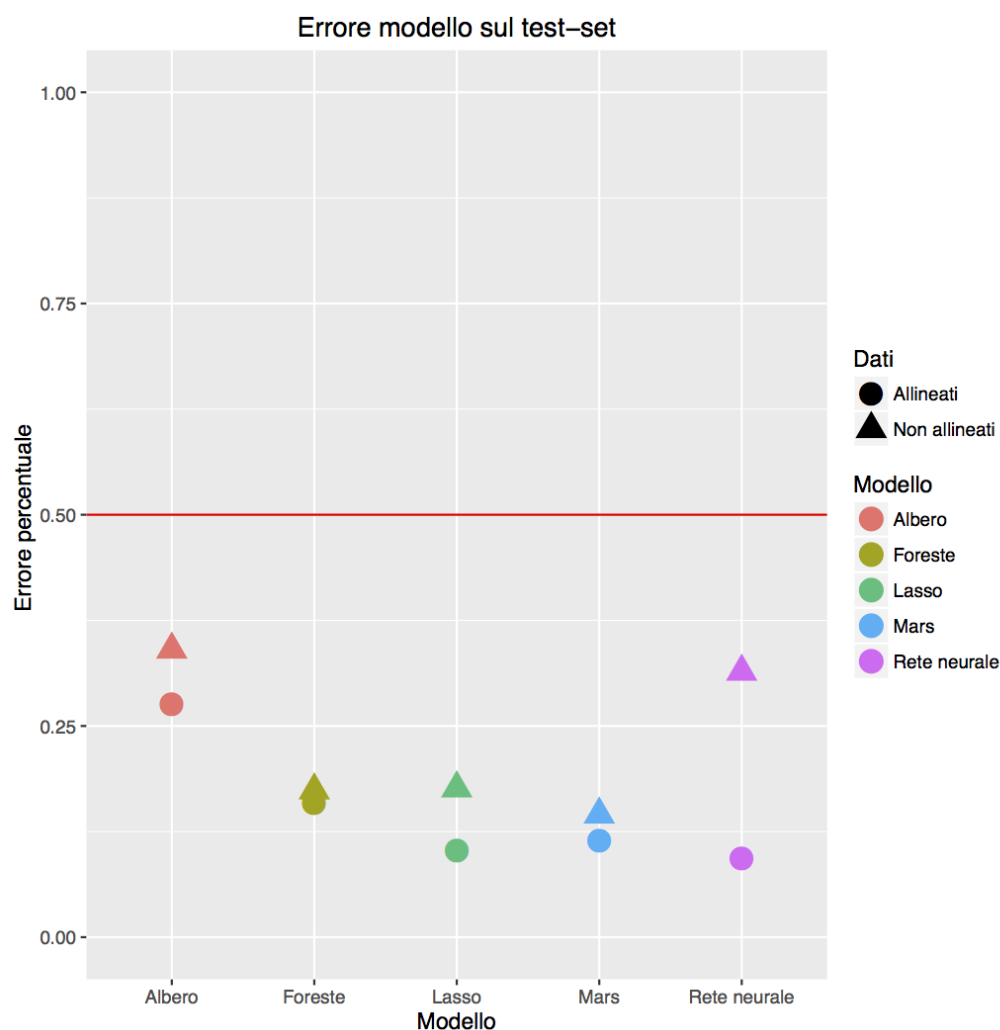


Figura 4.2: Rappresentazione grafica degli errori commessi dai modelli.

Escludendo dai commenti finali il modello reti neurali, dove servirebbero maggiori approfondimenti per capire il loro comportamento anomalo, vista la troppa differenza delle prestazioni tra i due casi. Esaminiamo i risultati ottenuti dei modelli restanti; modello albero, foreste casuali, modello lasso e modello MARS si ha sempre un miglioramento della previsione se viene effettuata l'analisi di procruste. In termini di riduzione percentuale dell'errore i modelli che prevedono meglio sono riportati nella seguente tabella 4.1:

	Riduzione percentuale dell'errore
Albero	18%
Foreste casuali	8%
Lasso	41%
Mars	21%
Rete neurale	70%

Tabella 4.1: Riduzione percentuale dell'errore nel passare dai dati non allineati a dati allineati.

Da questa tabella (4.1) possiamo notare come il passare da dati non allineati a dati allineati migliori di molto tutti i modelli ad esclusione del modello foreste casuali, anche si ha comunque un miglioramento. Analizzando invece la percentuale di errori di classificazione (4.2) si può vedere che il modello che commette la minor percentuale di errori.

	Errore di classificazione percentuale nell'insieme di verifica	
	Dati non allineati	Dati allineati
Albero	34.04%	27.61%
Random forest	17.38%	15.95%
Lasso	17.61%	10.23%
Mars	14.52%	11.42%
Rete neurale	31.42%	9.28%

Tabella 4.2: Errore di classificazione commesso dai modelli nei due differenti insiemi di verifica dati non allineati a dati allineati.

Il figura 4.3 sono rappresentati in forma 3D i voxel attivi e il loro livello di intensità del primo soggetto. I livelli di intensità e la selezione dei voxel sono quelli stimati attraverso il modello lasso. Ogni singolo punto identifica l'area attivata del cervello, si parla di area dato che in fase di pre elaborazione si erano raggruppati i voxel.

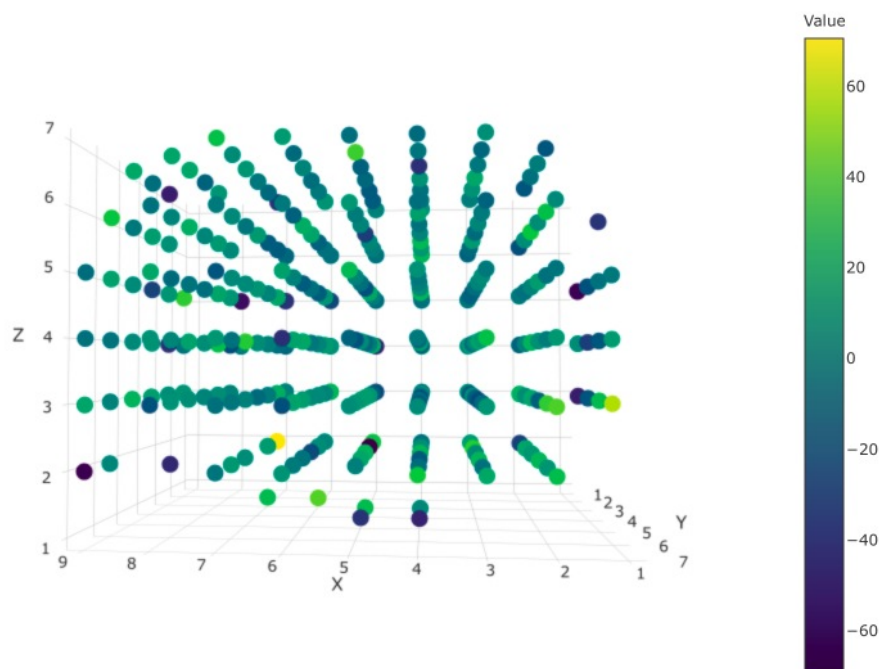


Figura 4.3: Rappresentazione grafica 3D dei voxel attivi e i rispettivi livelli di intensità del primo soggetto.

Bibliografia

- [1] Fabio Crosilla Alberto Beinat. Generalised procrustes analysis for size and shape 3-d object reconstructions. pages 345–353.
- [2] Jos M. F. Ten Berge. Orthogonal procrustes rotation for two or more matrices. *Psychometrika - VOL. 42, NO. 2.* 12, 19
- [3] Alberto Beinat Fabio Crosilla. Use of generalised procrustes analysis for the photogrammetric block adjustment by independent models. 56(3):195–209. 21
- [4] Colin Goodall. Procrustes methods in the statistical analysis of shape. *Journal of the Royal Statistical Society. Series B (Methodological), Vol.53, NO.2, 285-339.* 19
- [5] Amos Gorgolewski, Krzysztof J. and Storkey, Mark E. Bastin, Ian R. Whittle, Joanna M. Wardlaw, and Cyril R. Pernet. A test-retest fmri dataset for motor, language and spatial attention functions. *GigaScience*, 2(1):1–4, 2013. 6
- [6] Krzysztof J. Gorgolewski, Amos J. Storkey, Mark E. Bastin, Ian Whittle, and Cyril Pernet. Single subject fmri test–retest reliability metrics and confounding factors. *NeuroImage*, 69:231 – 243, 2013. 6
- [7] Gower. Generalized procrustes analysis. *Psychometrika - VOL. 40, NO. 1.* 12, 19
- [8] Patrick J.F. Groenen Ingwer Borg. *Modern Multidimensional Scaling.* Springer, 2005. 19, 20

- [9] Andrew C. Connolly Yaroslav O. Halchenko Bryan R. Conroy M. Ida Gobbini Michael Hanke Peter J. Ramadge James V. Haxby, J. Swaroop Guntupalli. A common high-dimensional model of the representational space in human ventral temporal cortex. *Neuron*, 72:404 – 416. 1, 12
- [10] Wingersky Kristof. Generalization of the orthogonal procrustes rotation procedure for more than two matrices. pages 89–90. 20
- [11] Peter H. Schonemann. A generalized solution of the orthogonal procrustes problem. 12, 15
- [12] Robert M. Schönemann, Peter H. and Carroll. Fitting one matrix to another under choice of a central dilation and a rigid motion. *Psychometrika*, 35(2):245–255, 1970. 12, 15, 18