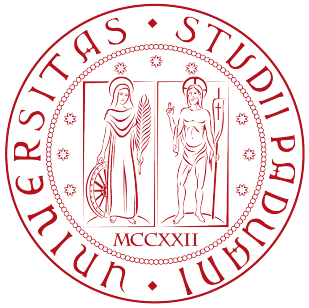
Università degli studi di Padova
Dipartimento di Scienze Statistiche
Corso di Laurea Magistrale in

Scienze statistiche



Metodi non parametrici per la combinazione di test diagnostici

Relatore: Ch.mo Prof. Monica Chiogna
Dipartimento di Scienze Statistiche

Laureando: Davide Bonanno Consiglio
Matricola N. 1131516

Anno Accademico 2017/2018

Alla mia famiglia,
colei che ha creduto sempre in me
e mi ha dato la forza di affrontare i momenti più difficili.

Indice

Introduzione

In ambito medico si utilizzano sempre più spesso test diagnostici e biomarcatori che danno specifiche informazioni riguardo lo status di una certa condizione di un paziente. Per valutare la loro efficacia, lo strumento più utilizzato è la curva ROC (*Receiving Operating Characteristic*) spesso sintetizzata in un unico valore, ovvero l'AUC (*Area Under the Curve*), che misura l'accuratezza dei tests e dei biomarcatori.

Non sempre, però, un test diagnostico o un biomarcatore è sufficientemente accurato e la sua capacità previsiva può essere potenziata se usato in combinazione con altri test. Per questo motivo, al fine di incrementare la precisione e l'informazione in possesso, si cerca di combinare in maniera opportuna più tests e biomarcatori.

Un approccio molto utile è quello di individuare la combinazione che permette di ottenere la curva ROC uniformemente più alta, massimizzando, di conseguenza, anche l'AUC. Il rapporto di verosimiglianza, strumento noto in ambito di verifica di ipotesi, individua la combinazione ottimale secondo questo criterio, come garantito dal lemma di Neyman-Pearson. Per ottenere una combinazione ottimale è possibile utilizzare anche altri approcci il cui criterio di ottimalità è relativo alla massimizzazione della sola AUC.

In questa tesi l'obiettivo è quello di individuare un'opportuna combinazione di test diagnostici basandosi sul rapporto di verosimiglianza e, in particolare, su una sua trasformazione monotona crescente (per cui gode delle stesse proprietà di ottimalità), ovvero il *risk score*, definito come probabilità di malattia condizionata ai risultati dei tests. A differenza del rapporto di verosimiglianza, il *risk score* ha una maggiore semplicità applicativa e interpretativa. La stima del *risk score* può essere effettuata all'interno di approcci

di tipo parametrico, come la regressione logistica, sebbene la specificazione del predittore lineare possa essere complessa. In questa tesi, ci si baserà su approcci di tipo non parametrico che garantiscono una maggiore flessibilità rispetto a modelli parametrici. Nello specifico, si stimerà il *risk score* per mezzo di modelli GAM (*Generalized Additive Model*) e di modelli MARS (*Multivariate Adaptive Regression Spline*).

Il primo capitolo richiama la teoria sottostante alla curva ROC e l'AUC e delinea il problema della combinazione di test diagnostici al fine di incrementare l'accuratezza nella diagnosi rispetto all'utilizzo dei singoli tests. Nel secondo capitolo si passa in rassegna gli approcci esistenti che individuano la combinazione ottimale massimizzando l'AUC. Dopo di ciò, ci si sofferma sulla combinazione per mezzo del *risk score* e si descrivono gli approcci di stima non parametrica, ovvero il GAM e il MARS, delineandone la teoria sottostante. Il capitolo 3 è dedicato ai risultati derivanti da svariati studi di simulazione effettuati con l'obiettivo di verificare quanto le curve ROC del *risk score* stimato con tali modelli si avvicinano alla vera curva. Sempre in questo capitolo, si riportano i risultati ottenuti dalle simulazioni effettuate per valutare l'adeguatezza dell'utilizzo di un approccio bootstrap per ottenere intervalli di confidenza sull'AUC relativo alla curva ROC del *risk score* stimato con i modelli non parametrici. Il capitolo 4 analizza tre insiemi di dati reali. Infine, il capitolo 5 è dedicato a delle considerazioni finali sui risultati prodotti in questa tesi con possibili proposte per sviluppi futuri sull'argomento.

Capitolo 1

I test diagnostici

1.1 Aspetti principali dei test diagnostici

In medicina, per avere informazioni riguardo una determinata condizione (soprattutto patologie), utilizzano test diagnostici oppure biomarcatori (particolare sostanza utilizzata come indicatore di un certo processo biologico).

Per dire che un test è accurato, si valuta se la sua capacità discriminante permette di dare risultati quanto più simili a quelli che si osservano utilizzando esami diagnostici di riferimento, i cosidetti *gold standard*.

Dato un insieme di n soggetti, si definisce una variabile D_i, $i = 1, ..., n$ tale per cui il valore 1 indica che l'i-esimo paziente è malato, mentre il valore 0 se il paziente è sano; si definisce, inoltre, un vettore di osservazioni $T = (T_1, ..., T_n)$ dove l'i-esimo elemento di T indica il risultato del test (che si ipotizza essere su scala continua) per l'i-esimo soggetto. Stabilito un valore soglia c, si dice che il paziente è malato se $T_i > c$ e che è sano se vale il viceversa. La sensibilità del test rappresenta la probabilità che identifichi il paziente come malato quando è effettivamente malato, mentre la specificità è la probabilità che il test identifichi il soggetto come sano quando è effettivamente sano, ovvero

$$\begin{cases} \text{Sensibilità:} & P\left(T_i > c | D_i = 1\right) \\ \text{Specificità:} & P\left(T_i < c | D_i = 0\right) \end{cases}$$

1.1.1 La curva ROC e l'AUC

Considerando che sensibilità e specificità dipendono dal valore soglia scelto, è possibile costruire uno strumento ampiamente utilizzato in ambito medico, ovvero la curva ROC (*Receiver Operating Characteristic*). Esso è uno strumento grafico che mette in relazione la sensibilità e la proporzione di falsi positivi (1-Specificità) e misura la capacità del test per tutti i valori soglia assumibili. La Figura 1.1 riporta diverse curve ROC. Più la curva si allontana

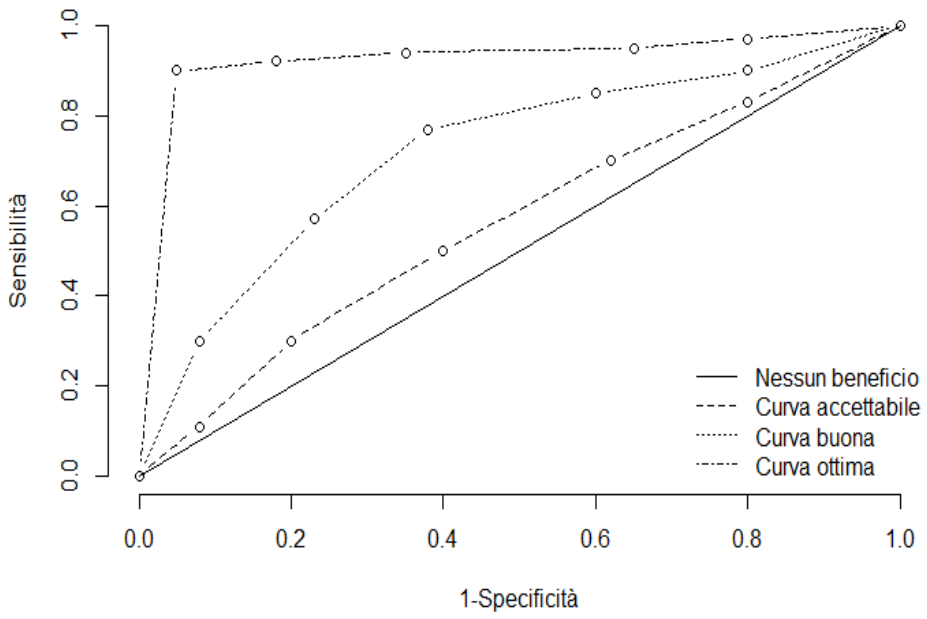


Figura 1.1: Esempi di curve ROC

dalla bisettrice (caso in cui i veri e falsi positivi sono in proporzioni uguali), più il test è accurato poichè sensibilità e specificità saranno massimizzate (e la curva raggiunge l'angolo sinistro in alto del grafico, dove entrambe le probabilità avranno valore 1). Sulla base di tale curva, un modo per sintetizzare con un unico valore la precisione del test è l'AUC (*Area Under the Curve*).

Siano, pertanto, T^0 e T^1 i punteggi ottenuti da un test diagnostico o biomarcatore e S_0 ed S_1 le relative funzioni di sopravvivenza[1], rispettivamente per i sani e per i malati.

L'AUC può essere definito come

$$AUC = \int_0^1 ROC(t)dt = \int_0^1 S_1\left(S_0^{-1}(t)\right)dt \tag{1.1}$$

Secondo i risultati riportati in Bamber (1975), è possibile esprimere (1.1) come $P(T^0 < T^1)$. Tale quantità è stimabile secondo un approccio non parametrico tramite la statistica U di Mann-Whitney,

$$\widehat{AUC} = \frac{1}{n_0 n_1}\sum_{i=1}^{n_0}\sum_{j=1}^{n_1} I(T_i^0 < T_j^1) \tag{1.2}$$

dove n_0 e n_1 sono le dimensioni campionarie rispettivamente dei sani e dei malati, mentre $I(.)$ è una funzione indicatrice che ha valore 1 se l'evento di interesse si verifica e 0 altrimenti.

Pepe (2003) fornisce una stima parametrica dell'AUC, assumendo che $T^d \sim N(\mu_d, \sigma_d^2)$ (i cui parametri sono stimati con i dati), $d = 0, 1$, esprimibile in

$$\widehat{AUC} = \Phi\left(\frac{\mu_1 - \mu_0}{\sqrt{\sigma_0^2 + \sigma_1^2}}\right) \tag{1.3}$$

Sulla base del valore assunto dall'AUC, è possibile dire quanto sia preciso il test diagnostico o il biomarcatore in questione: valori prossimi a 0.5 indicano un test poco accurato poichè la curva ROC è vicina alla bisettrice, mentre valori prossimi a 1 indicano un test altamente accurato poichè la curva ROC si avvicina all'angolo sinistro in alto, massimizzando sensibilità e specificità.

1.2 Combinazione di test diagnostici

Al fine di ottenere maggiori informazioni riguardo l'effettiva presenza di una certa condizione, spesso si fa ricorso a più di un test diagnostico, non sempre tutti dotati di elevata precisione quando considerati singolarmente.

[1] Probabilità che una variabile aleatoria (in questo caso T) sia superiore a un certo valore x. Rispettivamente $S_1(x) = \Pr(T \geq x|D = 1)$ e $S_0(x) = \Pr(T \geq x|D = 0)$.

Interessa in questi casi valutare se un'opportuna combinazione di tutte le informazioni consenta di ottenere una accuratezza nella diagnosi superiore a quella che si ottiene utilizzando i test singolarmente. Un criterio molto utile per ottenere una combinazione ottimale dei tests può essere quello di individuare quella combinazione tale per cui la curva ROC corrispondente sia uniformemente più alta rispetto a tutte le altre che si possono ottenere con altre combinazioni: ciò implica anche che l'AUC corrispondente è in assoluto il più alto.

Tale criterio può essere utilizzato in pratica grazie al lemma di Neyman-Pearson.

Si definisce, quindi, un vettore di p test diagnostici, $\mathbb{T}$, di cui $\mathbb{T}_i = (T_{1,i}, T_{2,i}, ..., T_{p,i})$ indica il vettore di risultati dei p tests per l'i-esima osservazione. Il rapporto di verosimiglianza può essere espresso come

$$LR(\mathbb{T}) = \frac{\Pr(\mathbb{T}|D=1)}{\Pr(\mathbb{T}|D=0)} \tag{1.4}$$

dove $\Pr(\mathbb{T}|D=1)$ indica la probabilità che $\mathbb{T}$ assume determinati valori nel gruppo dei malati, mentre $\Pr(\mathbb{T}|D=0)$ indica la probabilità che $\mathbb{T}$ assume determinati valori nel gruppo dei sani. Grazie al lemma di Neyman-Pearson, è possibile utilizzare (1.4) come regola decisionale dato un valore soglia c

$$LR(\mathbb{T}) > c$$

Si stabilisce tale soglia c come il più piccolo valore di $LR(\mathbb{T})$ per il quale la proporzione di falsi positivi è pari a un valore $t \in (0,1)$; per valori di $LR(\mathbb{T})$ superiori alla soglia il soggetto è classificato come malato, altrimenti come sano. Questa regola è ottima perchè il lemma di Neyman-Pearson garantisce che è possibile ottenere la più alta sensibilità per una qualsiasi proporzione di falsi positivi considerata (Green e Swets (1966)) e, quindi, tra tutte le possibili combinazioni che si possono considerare, questa restituisce la curva ROC uniformemente più alta.

Data tale regola decisionale, il problema che ci si pone è come stimare $LR(\mathbb{T})$: è possibile usare approcci di tipo parametrico, attribuendo una certa distribuzione di probabilità a $\mathbb{T}$ sia nel gruppo dei sani che in quello dei malati. Un altro modo per poter stimare $LR(\mathbb{T})$ è per via semi-parametrica,

come mostrato in Chen et al. (2016); è possibile anche seguire approcci di tipo non parametrico cercando di stimare la distribuzione dei tests nei sani e nei malati attraverso opportuni *kernel* e un esempio di stima di distribuzioni di probabilità condizionate attraverso tali *kernel* è dato in Hansen (2004).

Tuttavia, il rapporto di verosimiglianza di per sé non ha un riscontro pratico rilevante: ciò è dovuto dal fatto che da un punto di vista medico vi sono delle difficoltà interpretative, oltre al fatto che $LR(\mathbb{T})$ è adatto quando le densità nei sani e nei malati sono correttamente specificate (nell'ambito parametrico). Alla luce di tali problematiche, in Pepe (2003) si discute la possibilità di utilizzare una trasformazione monotona crescente quale il *risk score* che è la probabilità di malattia condizionandosi ai risultati dei tests ($\Pr(D = 1|\mathbb{T})$). Essendo una trasformazione monotona crescente di $LR(\mathbb{T})$, anche il *risk score* permette di ottenere una combinazione ottimale di test diagnostici tale per cui la curva ROC derivante è uniformemente più alta tra tutte le curve ROC ottenute con altre combinazioni (e, pertanto, anche l'AUC sarà il più alto).

Un altro modo per combinare i tests è quello di focalizzarsi nell'individuare quale tra le possibili combinazioni lineari di tests rendono l'AUC massimo. Considerando, quindi, il vettore di p test diagnostici $\mathbb{T}$, si definisce un vettore di parametri $\beta = (\beta_1, \beta_2, ..., \beta_p)$: l'AUC corrispondente alla combinazione lineare dei tests (distinguendo tra valori dei tests ottenuti nei sani e nei malati, rispettivamente $\mathbb{T}^0$ e $\mathbb{T}^1$) è definito come

$$AUC(\beta) = \Pr\left(\beta^T\mathbb{T}^0 < \beta^T\mathbb{T}^1\right)$$

Data tale definizione dell'AUC, gli approcci esistenti descrivono come individuare una stima di β, indicata con $\hat{\beta}$, tale per cui l'AUC risulti massimizzata.

Si rimanda al Capitolo 2 per approfondimenti riguardo combinazioni ottimali utilizzando il *risk score* e la massimizzazione dell'AUC.

1.3 Il rapporto di verosimiglianza tra verifica di ipotesi e test diagnostici

$LR(\mathbb{T})$ trova ragione di essere utilizzato come regola decisionale nell'ambito dei test diagnostici poichè i risultati ad esso legati sono noti in ambito

di verifica di ipotesi che è un punto di vista parallelo a quello di interesse.

Nell'ambito della verifica di ipotesi, il lemma di Neyman-Pearson pone come obiettivo quello di individuare il vero modello statistico per i dati a disposizione, ovvero $p^0(y)$. Si avrà quindi un modello sotto ipotesi nulla, $p_0(y)$, e un modello sotto ipotesi alternativa, $p_1(y)$: ciò equivale nell'ambito dei test diagnostici a individuare quale sia la vera distribuzione di $\mathbb{T}$ al fine di determinare univocamente il vero stato D: si ha, quindi, un modello sotto ipotesi nulla, $\Pr(\mathbb{T}|D=0)$, e un modello sotto ipotesi alternativa, $\Pr(\mathbb{T}|D=1)$. Tutto ciò si traduce nel seguente sistema di ipotesi

$$\begin{cases} H_0 : p^0(y) = p_0(y) \\ H_1 : p^0(y) = p_1(y) \end{cases} \equiv \begin{cases} H_0 : \mathbb{T} \sim \Pr(\mathbb{T}|D=0) \\ H_1 : \mathbb{T} \sim \Pr(\mathbb{T}|D=1) \end{cases} \tag{1.5}$$

Dato il sistema in (1.5), il lemma definisce una regione di rifiuto R^* e una regione di accettazione A^*, nel seguente modo

$$\begin{aligned} A^* &= \{y \in \mathcal{Y} : p_1(y) \leq c p_0(y)\} \\ R^* &= \{y \in \mathcal{Y} : p_1(y) > c p_0(y)\} \end{aligned} \tag{1.6}$$

Tale $c > 0$ corrisponde a una costante che permette di fissare l'errore di primo tipo α (espressa come probabilità di rifiutare l'ipotesi nulla quando questa è vera).

Nell'ambito dei test diagnostici, l'errore di primo tipo α corrisponde alla proporzione di falsi positivi; R^* corrisponde alla regione costituita da tutti quegli eventi (ordinati secondo il valore che assume $LR(\mathbb{T})$ in corrispondenza di ciascun evento) per cui si classifica un soggetto come malato; A^*, invece, corrisponde a quella regione costituita da tutti quegli eventi per cui si classifica un soggetto come sano. Data tale equivalenza, le regioni (1.6), quindi, possono essere riscritte nel seguente modo

$$\begin{aligned} A^* &= \{\mathbb{T} \in \mathcal{T} : \Pr(\mathbb{T}|D=1) \leq c \Pr(\mathbb{T}|D=0)\} \\ R^* &= \{\mathbb{T} \in \mathcal{T} : \Pr(\mathbb{T}|D=1) > c \Pr(\mathbb{T}|D=0)\} \end{aligned} \tag{1.7}$$

Dalle regioni (1.6) e (1.7) si evince che il test di riferimento è il rapporto di verosimiglianza, con un valore soglia c

$$t^*(y) = \frac{p_1(y)}{p_0(y)} \equiv LR(\mathbb{T}) = \frac{\Pr(\mathbb{T}|D=1)}{\Pr(\mathbb{T}|D=0)}$$

Poichè il lemma nell'ambito della verifica di ipotesi garantisce la più alta potenza del test $t^*(y)$ per un errore α fissato, per le equivalenze esposte consegue che tale lemma ha validità anche nel caso dei test diagnostici, restituendo per $LR(\mathbb{T})$ la più alta sensibilità (ovvero la potenza del test) per una proporzione di falsi positivi fissata, il che comporta che, per p test diagnostici, la regola decisionale basata su $LR(\mathbb{T})$, al variare di c è caratterizzata dalla curva ROC uniformemente più alta (e, quindi, anche l'AUC) tra tutte le regole possibili.

Capitolo 2

Approcci per la combinazione di test diagnostici

Nel Capitolo 1 si è discusso dell'utilità di considerare una combinazione di test diagnostici e da ciò si è delineato come è possibile fare ciò tramite l'approccio del rapporto di verosimiglianza poichè, grazie al lemma di Neyman-Pearson, è possibile ottenere una combinazione tale per cui la curva ROC è massimizzata, ovvero si ottiene la curva uniformemente più alta. Oltre al rapporto di verosimiglianza, vi sono altre due alternative per ottenere una combinazione ottimale: la prima individua una combinazione lineare ottimale secondo il criterio di massimizzazione dell'AUC, mentre l'altra consiste nell'utilizzare trasformazioni monotone crescenti del rapporto di verosimiglianza come il *risk score*.

Tale capitolo è suddiviso in due parti principali: nella prima, infatti, si passano in rassegna alcuni tra gli approcci esistenti che individuano la combinazione lineare ottimale massimizzando l'AUC. La seconda parte, invece, è dedicata alla descrizione del *risk score* e, in particolare, come stimarlo attraverso approcci non parametrici. Essendo il valore atteso della variabile D_i che determina lo status di malattia condizionandosi ai risultati dei tests, in questa tesi si propone l'utilizzo di due modelli non parametrici noti in letteratura per stimare il *risk score*: ciò trova giustificazione nel fatto che tramite tali modelli è possibile essere molto più flessibili di quanto lo si faccia stimando il *risk score* per mezzo di un modello logistico.

I modelli non parametrici utilizzati per tale scopo sono costruiti per gestire un numero elevato di covariate: essi sono il MARS (*Multivariate Adaptive Regression Spline*) il cui approccio è descritto in Hastie, Tibshirani e Friedman (2009), e il GAM (*Generalized Additive Models*) descritto in Wood (2006). Entrambi i modelli possono essere utilizzati semplicemente al posto di una regressione lineare, e con altrettanta semplicità possono essere estesi anche nell'ambito dei GLM, con un'opportuna specificazione della funzione *link* che, nel caso trattato in questa tesi, è di tipo logit per entrambi.

I due modelli sono accomunati da uno strumento di tipo non parametrico che può essere utilizzato per una o al più due covariate, ovvero le *spline*. Esse possono essere di diverso genere ed estendibili al caso multidimensionale, sebbene debbano far fronte a elevati costi computazionali, soprattutto quando il numero di covariate è maggiore di 2 (motivo per cui si usano *spline* al massimo per due covariate). Infatti, il MARS si presenta come una particolare specificazione di quelle che prendono il nome di *spline* di regressione; nel GAM, invece, si utilizzano delle funzioni non parametriche delle singole covariate nella costruzione del modello. La tipologia delle funzioni non parametriche per ciascuna covariata non è cruciale, e possono essere diverse per ciascuna di esse: tuttavia, in questa tesi sono considerate per tutte le covariate delle *spline* cubiche di regressione.

2.1 Combinare test diagnostici massimizzando l'AUC

Gli approcci che verranno passati in rassegna in seguito sono alternative al rapporto di verosimiglianza e il criterio utilizzato per individuare la combinazione di test diagnostici non è più la massimizzazione della curva ROC, ma solo dell'AUC. La combinazione risultante da questo criterio è di tipo lineare e ciascuna componente di $\beta = (\beta_1, ..., \beta_p)$ indica il peso che ha un determinato test all'interno di tale combinazione. Come si vedrà, le stime di questi parametri, $\hat{\beta}$, sono ottenute in differenti modi tramite specifiche assunzioni.

2.1.1 Approccio di Su e Liu

L'assunzione basilare di tale approccio è che i valori dei test ottenuti sui sani e sui malati, $\mathbb{T}^d$, $d = 0, 1$, si distribuiscono secondo una normale multivariata, $N_p(\mu_d, \Sigma_d)$ con $d = 0, 1$. Data tale assunzione, la combinazione lineare ottimale ottenuta da Su e Liu (1993) è data da

$$\beta = (\Sigma_0 + \Sigma_1)^{-1} (\mu_1 - \mu_0) \tag{2.1}$$

Le matrici di varianze e covarianze e i vettori delle medie possono essere stimate tramite i dati stessi, e saranno indicati come $\hat{\Sigma}_d$ e $\hat{\mu}_d$, $d = 0, 1$: sostituendo tali quantità in (2.1), lo stimatore $\hat{\beta}$ risultante è consistente. A questo punto, rifacendosi alla formula (1.3), l'AUC sotto ipotesi di normalità sarà

$$\begin{aligned} \widehat{AUC} &= \Phi\left(\frac{\hat{\beta}^T(\hat{\mu}_1 - \hat{\mu}_0)}{\sqrt{\hat{\beta}^T\left(\hat{\Sigma}_1 + \hat{\Sigma}_0\right)\hat{\beta}}}\right) \\ &= \Phi\left(\sqrt{(\hat{\mu}_1 - \hat{\mu}_0)^T\left(\hat{\Sigma}_1 + \hat{\Sigma}_0\right)^{-1}(\hat{\mu}_1 - \hat{\mu}_0)}\right) \end{aligned} \tag{2.2}$$

Tuttavia, in assenza di normalità oppure quando n_0 ed n_1 (rispettivamente il numero di sani e di malati) sono piccoli (per cui la normalità asintotica non è valita), il risultato (2.1) ottenuto non è ottimale.

2.1.2 Approccio di Pepe e Thompson

L'approccio proposto da Pepe e Thompson (2000), a differenza di Su e Liu (1993), rilassa l'ipotesi di normalità. Si pone nel caso in cui si hanno $p = 2$ test e sfrutta anche il fatto che la curva ROC è invariante rispetto a trasformazioni di scala, il che significa che si può porre $\beta = (\beta_1, \beta_2) = (1, \alpha)$, con $\alpha = \frac{\beta_2}{\beta_1}$. Per ottenere la combinazione ottima $\hat{\beta}$, viene utilizzata la formula (1.2) che, in questo caso, è data da

$$\widehat{AUC}(\beta) = \frac{1}{n_0 n_1} \sum_{i=1}^{n_0} \sum_{j=1}^{n_1} I\left(T^0_{1,i} + \alpha T^0_{2,i} < T^1_{1,i} + \alpha T^1_{2,i}\right) \tag{2.3}$$

Alla luce del fatto che non è possibile usare approcci basati sulle derivate (la funzione non è continua), in Pepe e Thompson (2000) si propone una procedura secondo la quale vengono considerati 201 valori α nell'intervallo $[-1, 1]$, e 201 valori $\gamma = \frac{1}{\alpha}$ nel medesimo intervallo: il coefficiente ottimale $\hat{\alpha}$ o $\hat{\gamma}^{-1}$ è quello che massimizza la formula (2.3). Inoltre, sono state proposte alcune estensioni tra cui quella di considerare delle *spline* cubiche su α (che permette di inserire covariate che possono influenzare sulla presicione della diagnosi), e la possibilità di massimizzare l'AUC parziale (pAUC).

Sebbene sia un approccio più robusto rispetto a quello proposto in Su e Liu (1993), c'è da tenere in considerazione che per $p \geq 3$ il costo computazionale per individuare i coefficienti ottimali è notevolmente elevato.

2.1.3 Approccio di Ma e Huang

Alla luce delle difficoltà insorte in Pepe e Thompson (2000), in Ma e Huang (2007) si propone una modifica della funzione obiettivo (2.3), approssimando la funzione indicatrice attraverso la funzione sigmoide $s(x) = \frac{1}{1+\exp(-x)}$: in questa maniera, piuttosto che utilizzare un metodo di ricerca globale, si utilizza un approccio basato sulle derivate per trovare la combinazione ottimale di interesse. Nello specifico, $\hat{\beta}$ è ottenuto massimizzando l'AUC sigmoidale (SAUC)

$$\hat{\beta} = \arg\max_{\beta \in C} \left\{ \frac{1}{n_0 n_1} \sum_{i=1}^{n_0} \sum_{j=1}^{n_1} s_n \left(\beta^T (\mathbb{T}_j^1 - \mathbb{T}_i^0) \right) \right\} \tag{2.4}$$

dove $C = \{\beta \in \mathbb{R}^p, \beta_1 = 1\}$; $s_n(x) = s(\frac{x}{\lambda_n})$ con $\lambda_n > 0$ e $\lim_{n\to\infty} \lambda_n = 0$.

Sotto opportune assunzioni, si fa vedere che, quando $n_0 + n_1 = n \to \infty$, $\hat{\beta}$ è uno stimatore consistente e assume una distribuzione normale con una certa media e una certa matrice di varianze e covarianze.

2.1.4 Approccio di Liu e Halabi

Una proposta alternativa a Ma e Huang (2007) è quella data da Liu e Halabi (2011) che modificano la formula (2.3), sostituendo ai singoli valori dei test per ogni soggetto i massimi e minimi valori tra tutti i p tests.

Il processo di stima di α rimane sempre la ricerca globale tale che l'AUC risulti massimizzato, come descritto in Pepe e Thompson (2000).

$$AUC(\beta) = \frac{1}{n_0 n_1} \sum_{i=1}^{n_0} \sum_{j=1}^{n_1} I\left(T^0_{i,\max} + \alpha T^0_{i,\min} < T^1_{j,\min} + \alpha T^1_{j,\max}\right) \tag{2.5}$$

dove

$$\begin{cases} T^0_{i,\max} = \max_{1\leq k\leq p} T^0_{i,k} & T^0_{i,\min} = \min_{1\leq k\leq p} T^0_{i,k} \\ T^1_{i,\max} = \max_{1\leq k\leq p} T^1_{i,k} & T^1_{i,\min} = \min_{1\leq k\leq p} T^1_{i,k} \end{cases}$$

Sebbene sia un approccio facilmente implementabile con un carico computazione ridotto poichè si deve stimare solo 1 parametro, è necessario anche evidenziare alcuni aspetti negativi. Il primo tra questi è la necessità di standardizzare i valori dei test quando questi presentano differenti unità di misura. Inoltre, non si garantisce in questo modo che tutta l'informazione data dai tests (biomarcatori) disponibile sia utilizzata.

2.1.5 Approccio di Kang et Al.

L'approccio definito da Kang, Liu e Tian (2016) consiste nell'inserire i vari tests (biomarcatori) secondo una procedura *stepwise* e i parametri sono stimati secondo quanto descritto in Pepe e Thompson (2000). In particolare, dato il vettore $\beta = (1, \alpha_2, ..., \alpha_p)$ la procedura si articola nel seguente modo:

Passo	**Descrizione**
1.	Stimare gli AUC per i p tests (biomarcatori) secondo la formula (1.2);
2.	Ordinare i tests da 1 a p dall'AUC più grande al più piccolo;
3.	Inserire i primi due test e stimare α_2;
4.	Inserire il terzo test con il più grande AUC;
5.	Continuare fino a quando non è stato inserito l'ultimo test.

L'algoritmo può anche essere riformulato modificando il passo 2: infatti, piuttosto che ordinare in senso decrescente, è possibile ordinare anche in senso crescente. Tale approccio (indipendentemente dall'ordinamento) risulta

robusto poichè non vi sono assunzioni sulle distribuzioni dei tests; inoltre, è di facile implementazione con un carico computazionale ridotto ed è possibile interpretare facilmente i parametri stimati (diversamente da quanto visto in Liu e Halabi (2011)).

2.2 Stima non parametrica del risk score

Come già detto nel Capitolo 1, il *risk score* è definito in Pepe (2003) come la probabilità di avere la malattia condizionata ai risultati dei tests ($RS(\mathbb{T}) = \Pr(D = 1|\mathbb{T})$). Come il rapporto di verosimiglianza, il criterio di ottimalità col quale si individua la combinazione ottimale di tests è la massimizzazione della curva ROC (che è uniformemente più alta di tutte quelle che si ottengono con altre combinazioni) e ciò è reso possibile perchè $RS(\mathbb{T})$ è una trasformazione monotona crescente di $LR(\mathbb{T})$. Si seguito se ne da la dimostrazione

Dimostrazione 1 *Data la definizione del risk score*

$$RS(\mathbb{T}) = \Pr(D = 1|\mathbb{T}) \tag{2.6}$$

Tramite il Teorema di Bayes, è possibile riscrivere l'equazione (2.6) *nel seguente modo*

$$\begin{aligned} \Pr(D = 1|\mathbb{T}) &= \frac{\Pr(\mathbb{T}|D = 1)\Pr(D = 1)}{\Pr(\mathbb{T})} \\ &= \frac{\Pr(\mathbb{T}|D = 1)\Pr(D = 1)}{\Pr(\mathbb{T}|D = 1)\Pr(D = 1) + \Pr(\mathbb{T}|D = 0)\Pr(D = 0)} \\ &= \frac{LR(\mathbb{T})\Pr(D = 1)}{LR(\mathbb{T})\Pr(D = 1) + \Pr(D = 0)} \end{aligned} \tag{2.7}$$

Il risultato appena ottenuto è, infatti, una funzione monotona crescente di $LR(\mathbb{T})$.

□

Tale *risk score* è una quantità che può essere facilmente stimata attraverso i dati e avendo le stesse proprietà di $LR(\mathbb{T})$, $RS(\mathbb{T})$ può essere utilizzato come regola decisionale che dipende da una soglia c^*

$$RS(\mathbb{T}) > c^*$$

Per valori superiori a c^*, un soggetto sarà classificato come malato, altrimenti come sano.

$RS(\mathbb{T})$ può essere modellizzato nel seguente modo

$$RS(\mathbb{T}) = g(\eta) = g(\beta_0 + h(\beta, \mathbb{T}))$$

dove $g(.)$ è una funzione monotona crescente generalmente incognita; β_0 è un'intercetta e $h(\beta, \mathbb{T})$ rappresenta la combinazione ottimale dei tests (o biomarcatori). Un modo semplice di stimare $RS(\mathbb{T})$ è dato dalla regressione logistica, la cui funzione *link* è di tipo logit: in questo caso, η costituisce la combinazione lineare dei tests con $g(.)$ funzione logistica

$$\begin{gathered} \eta = \log\left(\frac{RS(\mathbb{T})}{1-RS(\mathbb{T})}\right) = \beta_0 + \beta_1 T_1 + ... + \beta_p T_p \\ RS(\mathbb{T}) = \Pr(D = 1|\mathbb{T}) = g(\eta) = \frac{\exp(\eta)}{1+\exp(\eta)} \end{gathered} \tag{2.8}$$

Stimare il *risk score* tramite la regressione logistica è un approccio che è stato proposto in McIntosh e Pepe (2002) poichè tale modello ha il pregio di poter essere stimato sia in studi di coorte che in studi caso-controllo: in particolare, hanno proposto di stimare tale modello entro un certo *range* di falsi positivi, poichè tramite studi di simulazione hanno osservato che la curva ROC corrispondente si avvicina di più a quella vera, più di quella ottenuta stimando il *risk score* su tutto il *range* di falsi positivi, nonostante entrambe le curve ROC stimate fossero superiori a quelle associate ai singoli tests.

La stima parametrica di $RS(\mathbb{T})$ in (2.8) impone che η (ovvero la combinazione dei test diagnostici) abbia una certa forma funzionale e la sua specificazione, inoltre, può essere complessa. Alla luce di ciò, si propone di definire η in maniera più flessibile e ciò è fatto tramite l'utilizzo di modelli non parametrici poichè, diversamente da (2.8), η è definito nel seguente modo

$$\begin{gathered} \eta = \log\left(\frac{RS(\mathbb{T})}{1-RS(\mathbb{T})}\right) = f(\mathbb{T}) \\ RS(\mathbb{T}) = \Pr(D = 1|\mathbb{T}) = g(\eta) = \frac{\exp(\eta)}{1+\exp\eta} \end{gathered} \tag{2.9}$$

dove $f(.)$ è una generica funzione non parametrica dei tests. Di seguito si mostrano due modelli noti in letteratura per definire (2.9), ovvero il modello GAM (*Generalized Additive Model*) e il MARS (*Multivariate Adaptive*

Regression Spline): questi modelli definiscono η in maniera differente, ma entrambi sono accomunati dall'utilizzo delle *spline*, strumento non parametrico che per via dell'eccessivo carico computazionale è possibile utilizzare al più per due covariate.

Essendo entrambi accomunati dalle *spline*, prima di presentare il metodo di stima non parametrica del *risk score* tramite questi due modelli, si descrivono le caratteristiche principali di esse.

2.2.1 Le spline

Supponendo di avere n coppie (x, y), il termine *spline* è utilizzato in ambito matematico per indicare una tecnica di interpolazione di coppie di punti (x, y) facendo uso di funzioni polinomiali a tratti (ciascuna di grado d). La funzione che interpola tali coppie, $f(x)$, è ottenuta mediante la definzione di K punti ξ_j, $j = 1, ..., K$, ordinati tra loro ($\xi_1 < \xi_2 < ... < \xi_K$) all'interno del supporto dei valori di x e prendono il nome di nodi.

Un tipo di *spline* piuttosto comune è la *spline* cubica naturale il cui grado delle funzioni polinomiali a tratti è $d = 3$. Per poter parlare di ciò, $f(x)$ deve interpolare esattamente ciascun nodo e in corrispondenza di essi, deve avere derivate continue fino all'ordine 2 (per un generico grado d deve essere $d - 1$ l'ordine di derivabilità); infine, si pone che la derivata seconda in corrispondenza del primo e dell'ultimo nodo sia nulla, così che il numero totale di parametri sia $4(K - 1)$.

Tale tipo di *spline* sono impiegate nell'ambito statistico in due modi, ovvero come *spline* cubiche di regressione e *spline* di lisciamento. Supponendo sempre di avere n coppie di osservazioni (x, y), le *spline* di regressione sono legate a un modello del tipo

$$y = f(x) + \varepsilon \tag{2.10}$$

Indicando con ε un termine di errore, $f(x)$ è espressa come una combinazione di basi di funzioni che ha la seguente forma

$$f(x) = \sum_{j=1}^{K+4} \beta_j b_j(x) \tag{2.11}$$

dove β è un vettore di $K+4$ componenti di cui K è il numero di nodi e 4 sono i polinomi elementari: infatti

$$b_j(x) = \begin{cases} x^{j-1} & \text{per } j = 1, ..., 4 \\ (x - \xi_j)_+^3 = \max\left(0, (x - \xi_{j-4})^3\right) & \text{per } j = 5, ..., K+4 \end{cases}$$

Sebbene i parametri β possano essere stimati tramite minimi quadrati, non sempre si ha una conoscenza a priori di quanti e quali nodi considerare poichè questi determinano il grado di flessibilità del modello finale.

L'altro impiego delle *spline* cubiche naturali sono le *spline* di lisciamento che, per stimare $f(x)$ utilizzano i minimi quadrati penalizzati

$$\sum_{i=1}^{n} (y_i - f(x_i))^2 + \lambda \int_{-\infty}^{\infty} f''(t)^2 dt \tag{2.12}$$

Diversamente dalle *spline* di regressione, tramite (2.12) la flessibilità del modello è riassunta in un'unica componente, ovvero $\lambda \int_{-\infty}^{\infty} f''(t)^2 dt$: λ rappresenta il parametro di lisciamento, mentre $\int_{-\infty}^{\infty} f''(t)^2 dt$ rappresenta il grado di irregolarità di $f(x)$. Se $\lambda \to 0$, allora la penalizzazione è nulla e, pertanto, il risultato finale è la media di y in corrispondenza di una determinata ascissa x; se $\lambda \to \infty$ allora la penalizzazione è massima e ciò comporta ad ottenere come risultato finale la retta dei minimi quadrati poichè $f''(x) = 0$. La minimizzazione di (2.12) restituisce come risultato finale una *spline* cubica naturale il cui numero di nodi è pari al numero di valori distinti di x. Le stime del vettore di parametri, $\hat{\theta}$, possono essere ottenute in forma esplicita, riscrivendo (2.12) in forma matriciale: tali stime hanno la seguente forma

$$\hat{\theta} = (B^T B + \lambda \Omega)^{-1} B^T y$$

dove B è la matrice dei valori assunti dalle basi di funzioni in corrispondenza di ciascun nodo, mentre $\Omega = \int B_j''(t) B_j''(t) dt$.

Per ottenere il valore del parametro di lisciamento λ, esso sarà quello per la quale la *cross validation leave-one-out* (CV_{loo}) è minimizzata

$$CV_{loo} = \frac{1}{n}\sum_{i=1}^{n}\left(\hat{f}^{[-i]} - y_i\right)^2 \tag{2.13}$$

dove $\hat{f}^{[-i]}$ indica la stima di $f(x)$ ottenuta escludendo l'i-esima osservazione.

Nel caso di più covariate, la generalizzazione delle *spline* è complessa. Tuttavia, esistono due possibili estensioni, ovvero le *thin-plate spline* e i *tensor product spline*: tali generalizzazioni, però, soffrono di problemi di natura computazionale e di interpretazione.

2.2.2 Modello Additivo Generalizzato (GAM)

Un modello additivo generalizzato (GAM) è un'estensione del GLM poichè la forma assunta dal predittore η è svincolata da qualunque assunzione parametrica, a differenza del GLM che ha una forma ben specifica e definita in (2.8). Anche qui, la specificazione di un'opportuna funzione *link* è necessaria e dipende dalla natura della variabile di risposta: essendo interessati a stimare il *risk score* per mezzo del GAM, si considera un *link* di tipo logit, considerando che la variabile D_i è dicotomica. Da qui deriva che il modello è specificato nel seguente modo

$$\begin{gathered}\eta = \log\left(\frac{RS(\mathbb{T})}{1-RS(\mathbb{T})}\right) = f\left(\mathbb{T}\right) = \alpha + \textstyle\sum_{j=1}^{p} f_j(T_j) \\ RS(\mathbb{T}) = \Pr(D=1|\mathbb{T}) = g(\eta) = \frac{\exp(\eta)}{1+\exp\eta}\end{gathered} \tag{2.14}$$

Si nota in (2.14) che il predittore η è costituito da un'intercetta e da una somma di funzioni non parametriche delle singole covariate. Esse possono essere di diverso tipo e diverse tra loro per le varie covariate ma generalmente se ne sceglie una uguale per tutte quante: il tipo di funzione non è cruciale, ma per stimare il *risk score* si utilizzano delle *spline* cubiche di regressione per ciascuna covariata.

Per stimare tale modello, si fa ricorso a un approccio di stima noto come *Penalized-Iterative Reweighted Least Squares* (P-IRLS) che è una modifica dell'IRLS utilizzato nell'ambito dei GLM. Tale algoritmo è così articolato

Penalized-Iterative Reweighted Least Squares

1. Dato il predittore al passo K, $\eta^{[k]}$, e il valore atteso stimato $\mu^{[k]}$, calcolare i pesi e la variabile di risposta aggiustata

$$\omega_i \propto \frac{1}{V\left(\mu_i^{[k]}\right) g'\left(\mu_i^{[k]}\right)^2} = \mu_i^{[k]}(1-\mu_i^{[k]})$$

$$z_i = g'\left(\mu_i^{[k]}\right)\left(y_i - \mu_i^{[k]}\right) + \eta_i^{[k]} = \eta_i^{[k]} + \frac{y_i - \mu_i^{[k]}}{\mu_i^{[k]}(1-\mu_i^{[k]})}$$

2. Data la matrice triangolare W tale che $W_{[i,i]} = \omega_i$ e Z il vettore della variabile di risposta aggiustata, minimizzare i minimi quadrati penalizzati pesati per la variabile di risposta aggiustata

$$||\sqrt{W}(Z - f(X))||^2 + \sum_{j=1}^{p} \lambda_j \int f_j''(x_j)^2 dx$$

   Da ciò si ottiene la nuova stima del predittore, $\eta^{[k+1]}$ e, di conseguenza, $\mu^{[k+1]}$;
3. Continuare fino a quando la devianza residua (RSS) non è stabile.

I parametri di lisciamento $\lambda = (\lambda_1, \lambda_2, ..., \lambda_p)$ non sono più ottenuti minimizzando la CV_{loo} in (2.13) poichè il carico computazionale è molto elevato. Si preferisce, invece, utilizzare la *Generalized Cross Validation* (GVC) espressa come

$$GCV(\lambda) = \frac{\sum_{i=1}^{n}(y_i - \hat{f}_\lambda(x_i))^2}{\left(1 - \frac{d(\lambda)}{n}\right)^2} \tag{2.15}$$

dove $\hat{f}_\lambda(x_i)$ rappresenta la stima del *risk score* in corrispondenza di un preciso valore λ, mentre $d(\lambda)$ rappresenta i gradi di libertà del modello. Sul GAM, c'è anche da tenere in considerazione alcuni aspetti negativi: infatti, proprio per la sua natura additiva, eventuali effetti di interazione rilevanti tra

covariate non sono considerati dal modello in maniera automatica e, per questo motivo, tali effetti possono essere considerati solo se inseriti manualmente. Nel caso in cui fossero inserite delle interazioni

$$f(x_1) + f(x_2) + f(x_1, x_2)$$

allora è necessario imporre dei vincoli di identificabilità: in particolare, si propone un approccio secondo il quale si identificano numericamente le dipendenze lineari delle basi di un lisciatore preso in considerazione con quelle di un altro che condivide le stesse covariate ed eliminarle nel lisciatore preso in considerazione. Per dettagli è possibile fare riferimento a Wood (2006) (§4.10.2 p. 206).

Inoltre, nel GAM si parla anche di concurvità, che è un analogo della collinearità nel caso di modelli lineari, che accade quando si osserva tra le covariate una relazione di tipo non lineare. Le conseguenze derivanti dalla concurvità si osservano nelle stime del modello: infatti, nel caso di un modello additivo (la cui funzione *link* è l'identità), i risultati provenienti da un algoritmo di *backfitting* (metodo di stima del modello additivo) non sono unici (Morlini (2006)).

2.2.3 Multivariate Adaptive Regression Spline (MARS)

Il modello MARS (*Multivariate Adaptive Regression Spline*), proposto per la prima volta da Friedman (1991), è l'altro metodo non parametrico con il quale si vuole stimare il *risk score*. Tale modello è una particolare specificazione delle *spline* di regressione che permette di superare i limiti dettati dal carico computazionale e difficoltà di interpretazione che si ha nelle *spline* multivariate (*thin-plate* o *tensor product*). Sebbene anche il GAM utilizzi lisciatori univariati per ciascuna covariata, ciò che contraddistingue il MARS è che per ciascuna covariata si utilizzano coppie di basi di funzioni opposte tra loro, che non sono altro che delle *spline* di ordine $d = 1$. Si indica con $\mathcal{C}$ l'insieme di tutte le coppie di basi, ovvero

$$\mathcal{C} = \{(X_j - \xi)_+, (\xi - X_j)_+\} \quad \begin{matrix} \xi \in x_{1j}, ..., x_{nj} \\ j = 1, 2, ..., p \end{matrix}$$

di cui

$$(x_j - \xi)_+ = \begin{cases} x_j - \xi & \text{se } x_j > \xi \\ 0 & \text{altrimenti} \end{cases} \quad (\xi - x_j)_+ = \begin{cases} \xi - x_j & \text{se } x_j < \xi \\ 0 & \text{altrimenti} \end{cases}$$

Con ξ si indica un generico nodo.

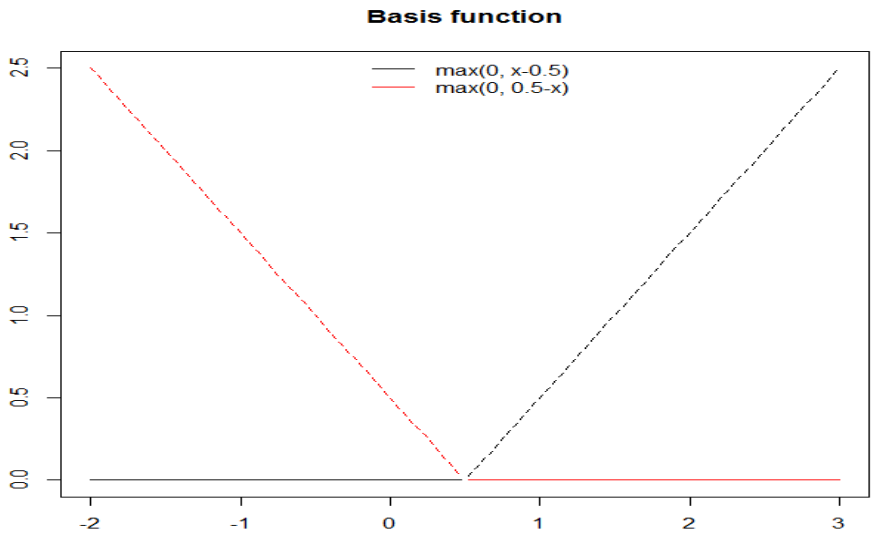


Figura 2.1: Esempio di base di funzione utilizzata nel modello MARS con $\xi = 0.5$

Per stimare il *risk score* tramite un modello MARS, si fa ricorso alla procedura descritta in Leathwick et al. (2005) che permette di generalizzare tale modello al caso dei GLM. Sotto un modello MARS, il predittore in (2.9) assume la seguente forma

$$\begin{gathered} \eta = \log\left(\frac{RS(\mathbb{T})}{1-RS(\mathbb{T})}\right) = f(\mathbb{T}) = \beta_0 + \sum_{k=1}^{K} \beta_k h_k(\mathbb{T}) \\ RS(\mathbb{T}) = \Pr(D = 1|\mathbb{T}) = g(\eta) = \frac{\exp(\eta)}{1+\exp\eta} \end{gathered} \tag{2.16}$$

dove K indica il numero di basi, mentre $h_k(.)$ sono le varie funzioni del modello che possono essere le singole basi appartenenti all'insieme $\mathcal{C}$ o prodotti tra loro. Il primo passo per stimare il modello specificato in (2.16) consiste

nel modellare il *risk score* secondo un modello MARS lineare, poichè tramite esso è possibile selezionare quelle funzioni (compresi eventuali prodotti tra basi) che hanno un contributo consistente nei confronti della precisione del modello: il risultato finale sarà una matrice chiamata bx di dimensione $n \times K$ (con n si indica il numero di osservazioni) che contiene tutti i valori assunti dalle funzioni che sono state selezionate e inserite nel modello. Tale selezione è articolata in due fasi: la prima è la crescita (*forward*), a cui segue la potatura (*pruning*). La fase *forward* è così articolata

Passo *forward*

1. Passo $K = 0$: si introduce la prima base $h_0(x) = 1$;
2. Passo generico $K = K + 1$: supposto che siano presenti già K basi nell'insieme $\mathcal{M}$ delle funzioni già inserite nel modello, la nuova coppia sarà quella data dal prodotto con una delle basi in $\mathcal{M}$ che minimizza la devianza residua e, dunque, al modello si aggiungerà il seguente termine

$$\hat{\beta}_{K+1} h_m(x)(x_j - \xi)_+ + \hat{\beta}_{K+2} h_m(x)(\xi - x_j)_+$$

   $h_m(x)$ indica una funzione in $\mathcal{M}$, mentre $\hat{\beta}_{K+1}$ e $\hat{\beta}_{K+2}$ sono parametri che vengono stimati insieme a tutti gli altri che sono già presenti nel modello per mezzo dei minimi quadrati;
3. Il processo continua fino a quando non si è raggiunto il numero massimo di termini.

Il motivo per cui alla fase *forward* segue la fase *pruning* è dovuto dal fatto che il modello così costruito si sovra-adatta ai dati e, pertanto, la fase *pruning* ha come obiettivo quello di eliminare quelle funzioni (o prodotti) che hanno un contributo trascurabile nei confronti della precisione del modello, e ciò verrà fatto per mezzo della GVC in (2.15) dove $d(\lambda)$ rappresenta in questo caso il numero di parametri impiegati nel modello inteso come somma del numero di termini che costituiscono il modello e dal numero di parametri per individuare i vari nodi.

La matrice bx risultante da questo passo è considerata come la nuova matrice delle covariate che viene utilizzata per stimare i relativi parametri β nel passo successivo. Per fare ciò, il *risk score* è modellato secondo un modello

logistico, con funzione *link* di tipo logit, la cui forma del predittore η è definita in (2.16): i parametri, quindi, sono stimati attraverso l'approccio classico dei GLM. La Figura 2.2 esemplifica tutta la procedura appena descritta.

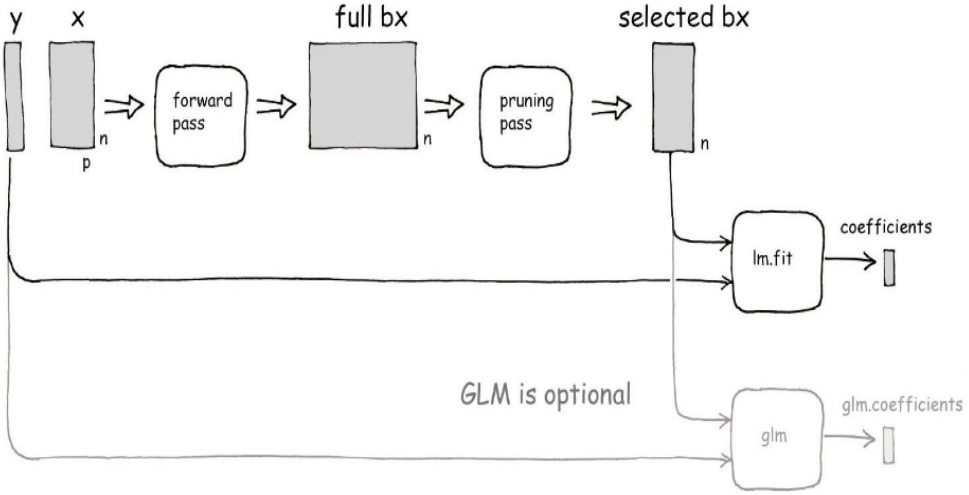


Figura 2.2: Processo di costruzione del modello MARS (immagine estratta da Milborrow (2017))

Il modello MARS presenta numerosi pregi tra cui il fatto di operare localmente: infatti, le basi di funzioni sono nulle per un certo *range* e, quando si considerano i prodotti tra essi, sono diverse da 0 solo per un *range* ristretto. Attraverso tale logica, le superfici vengono costruite in maniera parsimoniosa, impiegando un numero ridotto di parametri anche quando il numero di covariate è alto. Inoltre, il prodotto di basi di funzioni come queste permette di ridurre il carico computazionale per via della loro semplicità. Dal passo *forward*, inoltre, si delinea una costruzione di natura gerarchica, secondo la quale un'interazione (data dal prodotto delle basi di funzioni) di un certo ordine può esistere solo quando esiste l'interazione di un'ordine inferiore: si nota come il concetto delle interazioni è, diversamente dal GAM, qualcosa che è considerato automaticamente ed è possibile anche stabilire durante la costruzione del modello l'ordine massimo delle interazioni (anche per garantire l'interpretazione dei parametri ad essi legati). Infine, l'inserimento nel modello di covariate di tipo qualitativo non implica nessuna modifica per la

stima del modello poichè esse seguono lo stesso principio dettato dalle basi di funzioni.

Sebbene vi siano numerosi vantaggi, è necessario tenere in considerazione i problemi derivanti dalla collinearità: quando le covariate sono correlate tra loro e con la variabile di risposta, infatti, il modello nel passo *forward* sceglie in maniera del tutto arbitraria (poichè la devianza residua in corrispondenza delle due covariate è pressocchè uguale) l'assegnazione di un nodo a una delle due covariate che presentano tale caratteristica. Scegliendo in maniera arbitraria vi sono delle conseguenze incisive nei confronti del modello finale (Morlini (2006)).

Capitolo 3

Le simulazioni

Avendo definito nel Capitolo 2 i modelli non parametrici che si vogliono utilizzare per stimare il *risk score*, in questo capitolo si effettuano degli studi di simulazione atti a valutare quanto le curve ROC ottenute dalle stime di esso sono simili alla curva relativa al vero *risk score*.

Per tali valutazioni, sono stati posti cinque scenari, ovvero

- 2 tests che provengono da una distribuzione normale bivariata con parametri differenti tra soggetti malati e sani;
- 2 tests che provengono da una distribuzione esponenziale bivariata con parametri differenti tra soggetti sani e malati;
- 4 tests che provengono da una distribuzione normale a dimensione quattro con matrice di varianze e covarianze comune tra i sani e i malati, ma con differenti vettori di medie;
- 4 tests che provengono da una distribuzione normale a dimensione quattro le cui matrici di varianze e covarianze e i vettori delle medie sono differenti nei sani e nei malati;
- 2 tests indipendenti di cui il primo proviene da una distribuzione normale standard, mentre l'altro da una distribuzione uniforme nell'intervallo $(-1, +1)$.

Per ciascuno scenario sono state ottenute 1000 replicazioni, e le valutazioni sono fatte per differenti dimensioni campionarie e differenti specificazioni del set di dati utilizzato per stimare i modelli: ciò significa che sono state considerate anche trasformazioni di covariate. La curva ROC finale riportata per ciascun approccio di stima del *risk score* è ottenuta come media delle 1000 curve ROC ottenute per ciascun modello.

La stima del *risk score* tramite il modello GAM è fatta per mezzo del pacchetto R `mgcv` (Wood (2011)): per ciascuna covariata, si utilizza una *spline* cubica di regressione, ognuna costituita da 10 nodi (valore di *defeault*). La stima del *risk score* tramite il modello MARS, invece, è fatta per mezzo del pacchetto R `earth` (Milborrow (2017)).

Questo capitolo è suddiviso in tre parti principali, di cui la prima è dedicata a descrivere i risultati ottenuti nei cinque scenari considerati; nella seconda parte, invece, si discute dell'utilizzo dell'approccio di tipo bootstrap per ottenere intervalli di confidenza sugli AUC delle curve ROC del *risk score* stimato con i modelli non parametrici. Infine, la terza parte è dedicata a riepilogare i risultati caratterizzanti ottenuti tramite le simulazioni.

3.1 Scenario 1: test da normale bivariata

Sulla base di Chen et al. (2016), si indica con $\mathbb{T}|D = 1 \sim N_2(\mu_1, \Sigma_1)$ la distribuzione normale bivariata dei tests nel gruppo dei malati, e con $\mathbb{T}|D = 0 \sim N_2(\mu_0, \Sigma_0)$ la distribuzione normale bivariata dei tests nel gruppo dei sani. I rispettivi parametri delle due distribuzioni sono fissati nel seguente modo:

$$\mu_1 = (5, 4)^T \; \mu_0 = (1, 0)^T$$

$$\Sigma_1 = \begin{bmatrix} 10 & 0 \\ 0 & 8 \end{bmatrix}^{1/2} \begin{bmatrix} 1 & 0.5 \\ 0.5 & 1 \end{bmatrix} \begin{bmatrix} 10 & 0 \\ 0 & 8 \end{bmatrix}^{1/2}$$

$$\Sigma_0 = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}^{1/2} \begin{bmatrix} 1 & 0.4 \\ 0.4 & 1 \end{bmatrix} \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}^{1/2}$$

Sotto tali condizioni, il vero *risk score* ha la seguente forma

$$\text{logit}(RS(\mathbb{T})) = \alpha + \beta_1 T_1 + \beta_2 T_2 + \beta_3 T_1^2 + \beta_4 T_2^2 + \beta_5 T_1 T_2 \tag{3.1}$$

Non disponendo dei veri valori dei parametri in (3.1), sono state generate $n_1 = 10^4$ osservazioni da $N_2(\mu_1, \Sigma_1)$ e $n_0 = 10^4$ osservazioni da $N_2(\mu_0, \Sigma_0)$ e (3.1) è stato stimato come un modello logistico e da questo è stata ottenuta la corrispondente curva ROC. In tali simulazioni sono coinvolti i modelli GAM e MARS, di cui uno vincolato a non avere alcun effetto di interazione ($d = 1$) e un altro che include al massimo interazioni di primo ordine ($d = 2$), e un modello logistico non correttamente specificato (No-logit) che considera solo gli effetti lineari dei due test

$$\text{logit}(RS(\mathbb{T})) = \alpha + \beta_1 T_1 + \beta_2 T_2$$

I valori scelti per (n_1, n_0) per le simulazioni sono (45, 35), (90, 70), (180, 140) e (360, 280).

3.1.1 Specificazioni differenti dei parametri

Per queste simulazioni, si considerano differenti specificazioni delle matrici di varianze e covarianze mantenendo sempre gli stessi i vettori relativi alle medie: la prima simulazione considera le matrici definite in §3.1. Nelle altre due simulazioni che seguono, le matrici sono modificate in maniera tale che risulti accentuato il peso di β_5 associato al termine di interazione e quello del termine quadratico. Le stime dei parametri per le tre differenti specificazioni del modello teorico sono riportate in Appendice, nella Tabella B.1.

Simulazione 1

Secondo le condizioni iniziali, il vero valore dell'AUC è pari a 0.9465. La Figura 3.1 mostra che le curve ROC del *risk score* stimato tramite i MARS, diversamente da quella ottenuta stimandolo con il GAM, sono più vicine alla vera curva anche quando si hanno poche osservazioni (pannello (a)). Nei restanti pannelli, si nota che le curve ROC ottenute stimando il *risk score* con il GAM e i MARS sono indistinguibili sia tra loro che rispetto alla vera curva. La curva ROC del *risk score* stimato tramite il No-logit è

sempre lontana dalla vera curva poichè non coglie gli effetti quadratici e di interazione presenti in (3.1).

AUC (n_1, n_0)	**GAM**	**MARS** $(d = 1)$	**MARS** $(d = 2)$	**No-Logit**
(45, 35)	0.9649	0.9510	0.9523	0.9237
	(0.0217)	(0.0284)	(0.0282)	(0.0325)
(90, 70)	0.9534	0.9498	0.9257	0.9228
	(0.0170)	(0.0184)	(0.0178)	(0.0219)
(180, 140)	0.9479	0.9469	0.9493	0.9221
	(0.0121)	(0.0129)	(0.0136)	(0.0153)
(360, 280)	0.9448	0.9443	0.9467	0.9214
	(0.0090)	(0.0099)	(0.0096)	(0.0113)

Tabella 3.1: Simulazione 1, scenario 1. AUC medi con deviazione standard tra parentesi. Il vero AUC è 0.9465.

Simulazione 2

Le matrici di varianze e covarianze che danno luogo ad un valore β_5 più elevato in (3.1), sono le seguenti

$$\Sigma_1 = \begin{bmatrix} 10 & 0 \\ 0 & 10 \end{bmatrix}^{1/2} \begin{bmatrix} 1 & 0.9 \\ 0.9 & 1 \end{bmatrix} \begin{bmatrix} 10 & 0 \\ 0 & 10 \end{bmatrix}^{1/2}$$

$$\Sigma_0 = \begin{bmatrix} 2 & 0 \\ 0 & 1 \end{bmatrix}^{1/2} \begin{bmatrix} 1 & 0.8 \\ 0.8 & 1 \end{bmatrix} \begin{bmatrix} 2 & 0 \\ 0 & 1 \end{bmatrix}^{1/2}$$

In questo scenario, il vero AUC è 0.9337. Dalla Figura 3.13 in §3.8 si nota come la curva ROC del *risk score* stimato con il MARS $(d = 2)$ è sempre vicina a quella vera qualunque sia (n_1, n_0) mentre quelle ottenute stimandolo con il GAM e il MARS $(d = 1)$ differiscono leggermente da quella vera. La curva ROC ottenuta stimando il *risk score* con il No-logit presenta le stesse caratteristiche osservate in Figura 3.1.

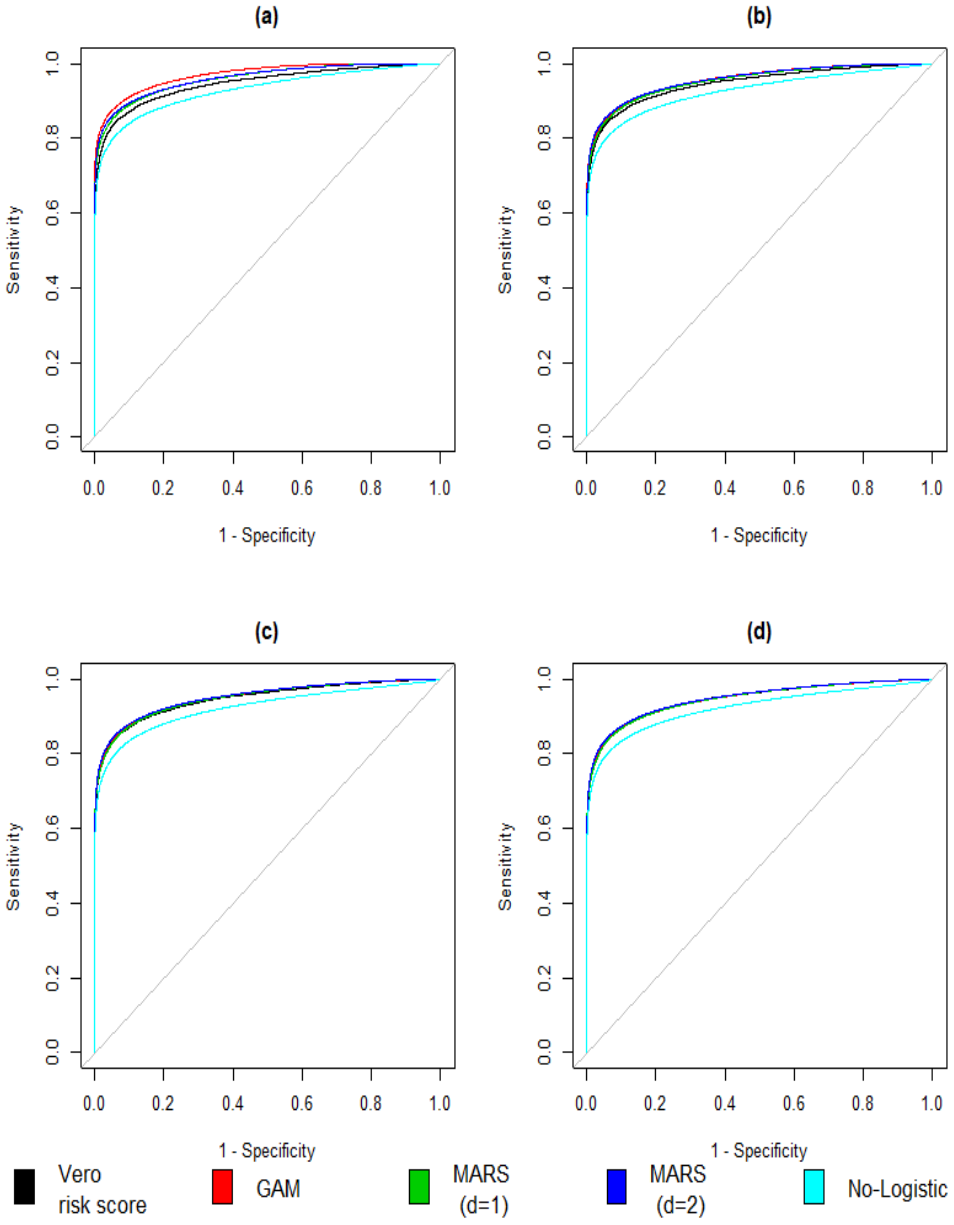


Figura 3.1: Simulazione 1, scenario 1. Confronto tra curve ROC stimate e curva vera per (a) $(n_1, n_0) = (45, 35)$, (b) $(n_1, n_0) = (90, 70)$, (c) $(n_1, n_0) = (180, 140)$ e (d) $(n_1, n_0) = (360, 280)$.

AUC (n_1, n_0)	**GAM**	**MARS** $(d = 1)$	**MARS** $(d = 2)$	**No-Logit**
(45, 35)	0.9531	0.9309	0.09363	0.8947
	(0.0267)	(0.0330)	(0.0338)	(0.0388)
(90, 70)	0.9368	0.9264	0.9407	0.8918
	(0.0204)	(0.0233)	(0.0246)	(0.0271)
(180, 140)	0.9290	0.9242	0.9387	0.8893
	(0.0144)	(0.0169)	(0.0184)	(0.0195)
(360, 280)	0.9267	0.9231	0.9370	0.8904
	(0.0105)	(0.0129)	(0.0159)	(0.0136)

Tabella 3.2: Simulazione 2, scenario 1: AUC medi con deviazione standard tra parentesi. Il vero AUC è 0.9337.

Simulazione 3

Le matrici di varianze e covarianze che danno luogo a un valore di β_4 più alto, rendendo minimo il valore di β_5, sono le seguenti

$$\Sigma_1 = \begin{bmatrix} 12 & 0 \\ 0 & 8 \end{bmatrix}^{1/2} \begin{bmatrix} 0.5 & 0.01 \\ 0.01 & 0.5 \end{bmatrix} \begin{bmatrix} 12 & 0 \\ 0 & 8 \end{bmatrix}^{1/2}$$

$$\Sigma_0 = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}^{1/2} \begin{bmatrix} 2 & 0.1 \\ 0.1 & 0.5 \end{bmatrix} \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}^{1/2}$$

In questo scenario, il vero AUC è 0.9861. Dalla Figura 3.2 si nota come le curve ROC del *risk score* stimato con i MARS e il GAM si avvicinano alla vera curva con l'aumentare della dimensione campionaria, a differenza di quella ottenuta stimandolo con il No-logit che, invece, è sempre vicina a quella vera.

3.1.2 Trasformazioni di covariate

Le simulazioni seguenti prevedono delle trasformazioni dei dati simulati utilizzati per stimare i modelli proposti. In particolare, si trasforma uno dei due tests lasciando immutata la specificazione del regressore. Si considerano

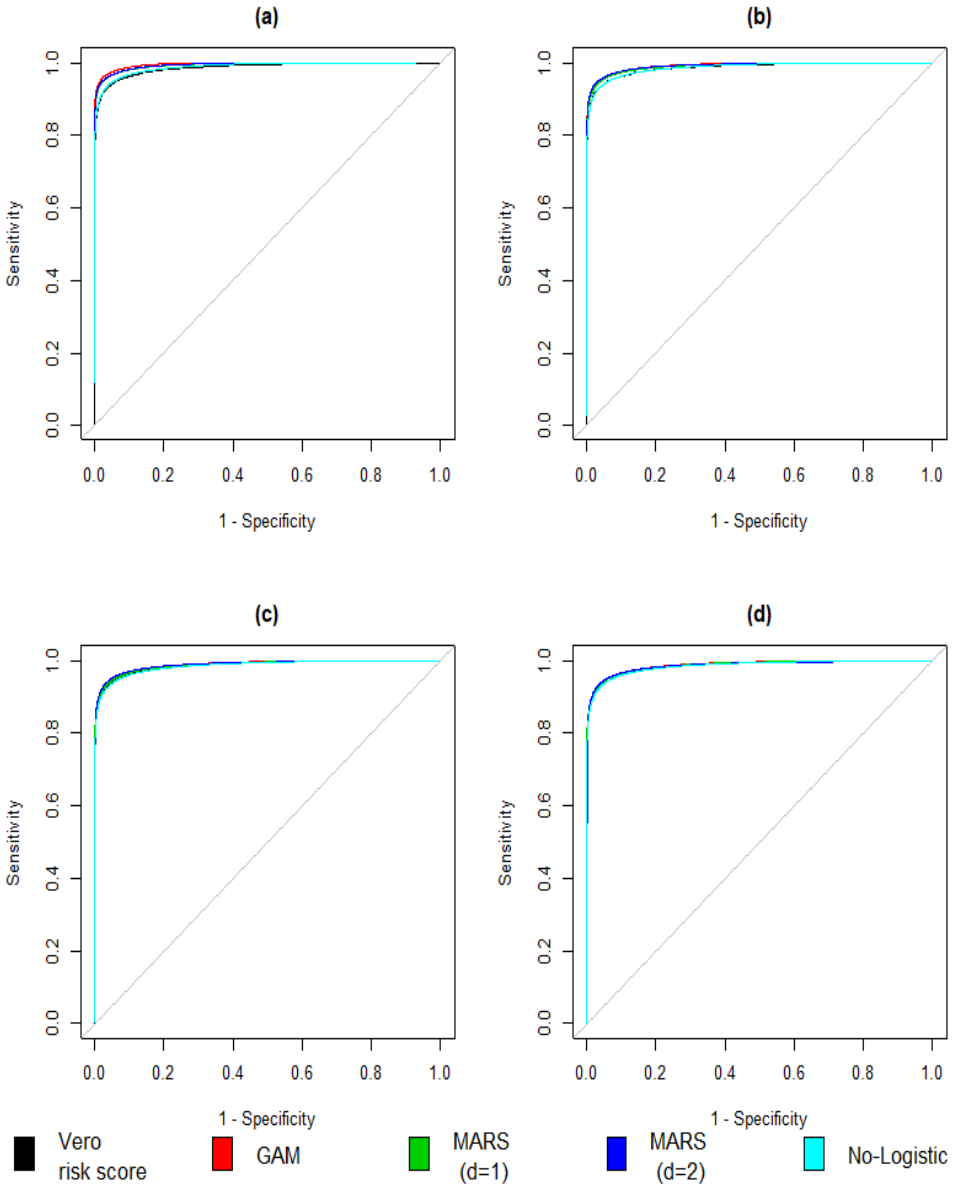


Figura 3.2: Simulazione 3, scenario 1. Confronto tra curve ROC stimate e curva vera per (a) $(n_1, n_0) = (45, 35)$, (b) $(n_1, n_0) = (90, 70)$, (c) $(n_1, n_0) = (180, 140)$ e (d) $(n_1, n_0) = (360, 280)$.

AUC (n_1, n_0)	**GAM**	**MARS** $(d = 1)$	**MARS** $(d = 2)$	**No-Logit**
(45, 35)	0.9935	0.9898	0.9896	0.9856
	(0.0070)	(0.0099)	(0.0112)	(0.0113)
(90, 70)	0.9901	0.9888	0.9888	0.9850
	(0.0070)	(0.0071)	(0.0086)	(0.0083)
(180, 140)	0.9880	0.9871	0.9871	0.9844
	(0.0051)	(0.0054)	(0.0090)	(0.0057)
(360, 280)	0.9872	0.9867	0.9855	0.9842
	(0.0037)	(0.0037)	(0.0107)	(0.0041)

Tabella 3.3: Simulazione 3, scenario 1. AUC medi con deviazione standard tra parentesi. Il vero AUC è 0.9861.

tre trasformazioni e per ciascuna di esse si effettuano delle simulazioni per le tre specificazioni delle matrici di varianze e covarianze. Ciò ha lo scopo di valutare come i modelli utilizzati rispondono ad errate specificazioni delle covariate.

Simulazione 4, 5 e 6

La prima trasformazione considerata è ($T_2 = \frac{1}{T_2}$). È possibile notare dalla Figura 3.3, 3.14 e 3.15 (di cui le ultime due riportate in §3.8) che la curva ROC del *risk score* stimato con il MARS ($d = 2$) è più vicina a quella reale quando si hanno poche osservazioni; quella ottenuta stimandolo con il GAM, invece, è più vicina a quella vera solo quando si hanno più osservazioni. Le altre, infine, sono sempre lontane dalla vera curva.

Simulazione 7, 8 e 9

Considerando la trasformazione $T_2 = T_2^3$, le Figure 3.16, 3.17 (riportate in §3.8) mostrano gli stessi risultati ottenuti nelle simulazioni 4 e 5: l'unica differenza è che qui la curva ROC del *risk score* stimato con il MARS ($d = 2$) è vicina a quella reale tanto quanto quella ottenuta stimandolo con il modello

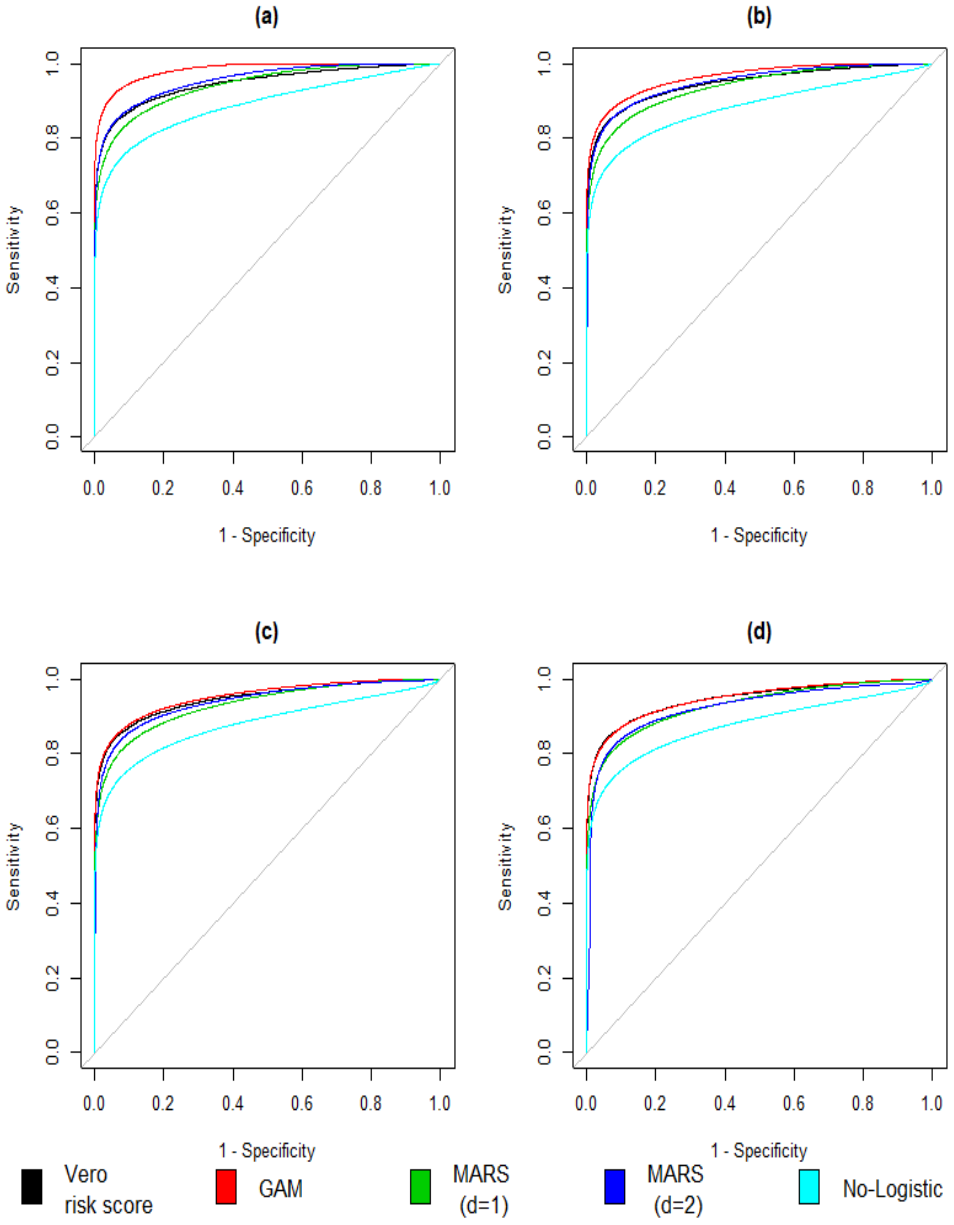


Figura 3.3: Simulazione 4, scenario 1. Confronto tra curve ROC stimate e curva vera per (a) $(n_1, n_0) = (45, 35)$, (b) $(n_1, n_0) = (90, 70)$, (c) $(n_1, n_0) = (180, 140)$ e (d) $(n_1, n_0) = (360, 280)$. Nei dati simulati $T_2 = \frac{1}{T_2}$.

AUC (n_1, n_0)	**GAM**	**MARS** $(d = 1)$	**MARS** $(d = 2)$	**No-Logit**
(45, 35)	0.9765	0.9292	0.9431	0.8833
	(0.0234)	(0.0429)	(0.0415)	(0.0387)
(90, 70)	0.9602	0.9288	0.9443	0.8798
	(0.0167)	(0.0337)	(0.0290)	(0.0278)
(180, 140)	0.9510	0.9258	0.9364	0.8777
	(0.0124)	(0.0289)	(0.0271)	(0.0196)
(360, 280)	0.9463	0.9255	0.9252	0.8759
	(0.0088)	(0.0254)	(0.0315)	(0.0138)

Tabella 3.4: Simulazione 4, scenario 1. AUC medi con la deviazione standard tra parentesi. Il vero AUC è 0.9465. Nei dati simulati $T_2 = \frac{1}{T_2}$.

AUC (n_1, n_0)	**GAM**	**MARS** $(d = 1)$	**MARS** $(d = 2)$	**No-Logit**
(45, 35)	0.9666	0.9141	0.9316	0.8797
	(0.0257)	(0.0418)	(0.0429)	(0.0399)
(90, 70)	0.9450	0.9135	0.9337	0.8771
	(0.0196)	(0.0314)	(0.0286)	(0.0265)
(180, 140)	0.9339	0.9110	0.9263	0.8768
	(0.0147)	(0.0252)	(0.0224)	(0.0199)
(360, 280)	0.9279	0.9097	0.9175	0.8759
	(0.0109)	(0.0204)	(0.0183)	(0.0142)

Tabella 3.5: Simulazione 5, scenario 1. AUC medi con la deviazione standard tra parentesi. Il vero AUC è 0.9337. Nei dati simulati $T_2 = \frac{1}{T_2}$.

AUC (n_1, n_0)	**GAM**	**MARS** $(d = 1)$	**MARS** $(d = 2)$	**No-Logit**
(45, 35)	0.9960	0.9851	0.9851	0.9011
	(0.0107)	(0.0407)	(0.0218)	(0.0340)
(90, 70)	0.9930	0.9649	0.9831	0.8987
	(0.0074)	(0.0366)	(0.0192)	(0.0242)
(180, 140)	0.9903	0.9646	0.9781	0.8977
	(0.0047)	(0.0349)	(0.0230)	(0.0168)
(360, 280)	0.9881	0.9609	0.9703	0.8972
	(0.0035)	(0.0353)	(0.0296)	(0.0120)

Tabella 3.6: Simulazione 6, scenario 1. AUC medi con le deviazioni standard tra parentesi. Il vero AUC è 0.9861. Nei dati simulati $T_2 = \frac{1}{T_2}$.

GAM anche quando le osservazioni sono tante. Sulla Figura 3.18 in §3.8 è possibile fare le stesse considerazioni fatte per la Figura 3.2.

Simulazione 10, 11 e 12

L'ultima trasformazione considerata è $T_2 = \exp(T_2)$. Dalla Figura 3.4, 3.19 e 3.20 (di cui le ultime due riportate in §3.8) si nota come solo la curva ROC del *risk score* stimato con il GAM si avvicina sempre più alla vera curva all'aumentare del numero di osservazioni, sebbene nella simulazione 11 tenda a essere leggermente distante da essa. Nella simulazione 12, invece, si ha anche che la curva ROC del *risk score* stimato con il MARS $(d = 1)$ è vicina alla vera curva all'aumentare del numero di osservazioni.

AUC (n_1, n_0)	**GAM**	**MARS** $(d = 1)$	**MARS** $(d = 2)$	**No-Logit**
(45, 35)	0.9681	0.9391	0.9432	0.9230
	(0.0221)	(0.0311)	(0.0313)	(0.0340)
(90, 70)	0.9544	0.9399	0.9467	0.9214
	(0.0171)	(0.0200)	(0.0194)	(0.0234)
(180, 140)	0.9471	0.9366	0.9438	0.9198
	(0.0133)	(0.0151)	(0.0146)	(0.0176)
(360, 280)	0.9445	0.9355	0.9423	0.9194
	(0.0085)	(0.0115)	(0.0116)	(0.0115)

Tabella 3.7: Simulazione 7, scenario 1. AUC medi con le deviazioni standard tra parentesi. Il vero AUC è 0.9465. Nei dati simulati $T_2 = T_2^3$.

AUC (n_1, n_0)	**GAM**	**MARS** $(d = 1)$	**MARS** $(d = 2)$	**No-Logit**
(45, 35)	0.9535	0.8996	0.9178	0.8930
	(0.0282)	(0.0370)	(0.0373)	(0.0405)
(90, 70)	0.9356	0.8971	0.9231	0.8899
	(0.0213)	(0.0274)	(0.0268)	(0.0288)
(180, 140)	0.9276	0.8972	0.9247	0.8886
	(0.0155)	(0.0192)	(0.0209)	(0.0208)
(360, 280)	0.9244	0.8941	0.9260	0.8876
	(0.0111)	(0.0137)	(0.0159)	(0.0144)

Tabella 3.8: Simulazione 8, scenario 1. AUC medi con le deviazioni standard tra parentesi. Il vero AUC è 0.9337. Nei dati simulati $T_2 = T_2^3$.

AUC (n_1, n_0)	**GAM**	**MARS** $(d = 1)$	**MARS** $(d = 2)$	**No-Logit**
(45, 35)	0.9941	0.9874	0.9867	0.9857
	(0.0070)	(0.0114)	(0.0124)	(0.0117)
(90, 70)	0.9904	0.9865	0.9863	0.9850
	(0.0066)	(0.0085)	(0.0093)	(0.0085)
(180, 140)	0.9882	0.9854	0.9860	0.9844
	(0.0050)	(0.0059)	(0.0069)	(0.0058)
(360, 280)	0.9870	0.9845	0.9851	0.9842
	(0.0038)	(0.0065)	(0.0066)	(0.0043)

Tabella 3.9: Simulazione 9, scenario 1. AUC medi con le deviazioni standard tra parentesi. Il vero AUC è 0.9861. Nei dati simulati $T_2 = T_2^3$.

AUC (n_1, n_0)	**GAM**	**MARS** $(d = 1)$	**MARS** $(d = 2)$	**No-Logit**
(45, 35)	0.9663	0.8934	0.9141	0.9254
	(0.0280)	(0.0409)	(0.0433)	(0.0312)
(90, 70)	0.9531	0.8961	0.9157	0.9258
	(0.0274)	(0.0306)	(0.0367)	(0.0222)
(180, 140)	0.9460	0.8982	0.9188	0.9235
	(0.0209)	(0.0282)	(0.0320)	(0.0154)
(360, 280)	0.9441	0.9001	0.9220	0.9237
	(0.0089)	(0.0260)	(0.0281)	(0.0113)

Tabella 3.10: Simulazione 10, scenario 1. AUC medi con le deviazioni standard tra parentesi. Il vero AUC è 0.9465. Nei dati simulati $T_2 = \exp(T_2)$.

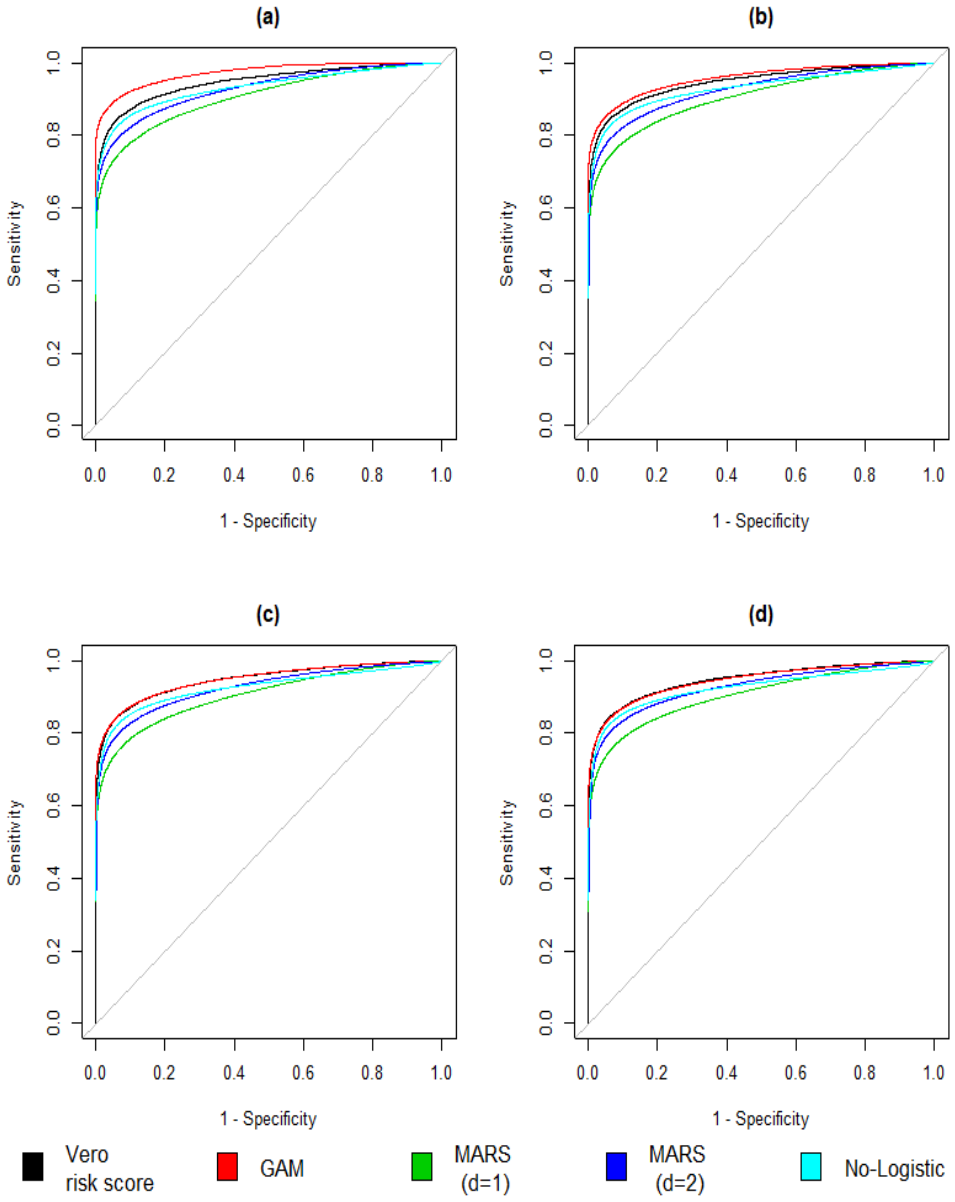


Figura 3.4: Simulazione 10, scenario 1. Confronto tra curve ROC stimate e curva vera per (a) $(n_1, n_0) = (45, 35)$, (b) $(n_1, n_0) = (90, 70)$, (c) $(n_1, n_0) = (180, 140)$ e (d) $(n_1, n_0) = (360, 280)$. Nei dati simulati $T_2 = \exp(T_2)$.

AUC (n_1, n_0)	**GAM**	**MARS** $(d = 1)$	**MARS** $(d = 2)$	**No-Logit**
(45, 35)	0.9446	0.8901	0.8949	0.9028
	(0.0471)	(0.0384)	(0.0407)	(0.0377)
(90, 70)	0.9256	0.8866	0.8977	0.8969
	(0.0521)	(0.0279)	(0.0306)	(0.0268)
(180, 140)	0.9175	0.8865	0.9016	0.8927
	(0.0503)	(0.0184)	(0.0215)	(0.0198)
(360, 280)	0.9195	0.8871	0.9034	0.8913
	(0.0342)	(0.0142)	(0.0166)	(0.0147)

Tabella 3.11: Simulazione 11, scenario 1. AUC medi con le deviazioni standard tra parentesi. Il vero AUC è 0.9337. Nei dati simulati $T_2 = \exp(T_2)$.

AUC (n_1, n_0)	**GAM**	**MARS** $(d = 1)$	**MARS** $(d = 2)$	**No-Logit**
(45, 35)	0.9935	0.9585	0.9795	0.9863
	(0.0091)	(0.0455)	(0.0255)	(0.0114)
(90, 70)	0.9903	0.9749	0.9824	0.9865
	(0.0083)	(0.0327)	(0.0194)	(0.0074)
(180, 140)	0.9880	0.9808	0.9797	0.9856
	(0.0049)	(0.0223)	(0.0189)	(0.0053)
(360, 280)	0.9870	0.9831	0.9718	0.9855
	(0.0036)	(0.0163)	(0.0285)	(0.0038)

Tabella 3.12: Simulazione 12, scenario 1. AUC medi con le deviazioni standard tra parentesi. Il vero AUC è 0.9861. Nei dati simulati $T_2 = \exp(T_2)$.

3.2 Scenario 2: test da esponenziale bivariata

Sulla base di Chen et al. (2016), sia $T_1|D = 1 \sim \text{Exp}(\xi_1)$ la distribuzione esponenziale del primo test in caso di malattia, e $T_2|D = 1 \sim \text{Exp}(\xi_2)$ la distribuzione esponenziale del secondo test, anche questa nel gruppo dei malati.

La funzione di densità congiunta, per una correlazione tra i tests pari a $\frac{\xi_0}{\xi_1+\xi_2-\xi_0}$ con $\xi_0 \in [0, \min(\xi_1, \xi_2))$ è espressa come

$$f(\mathbb{T}|D) = \begin{cases} \xi_1(\xi_2 - \xi_0)e^{(-\xi_1 t_1 - (\xi_2-\xi_0)t_2)} & \text{se } t_1 \leq t_2, \\ \xi_2(\xi_1 - \xi_0)e^{(-\xi_2 t_2 - (\xi_1-\xi_0)t_1)} & \text{se } t_1 > t_2. \end{cases}$$

In Chen et al. (2016) viene dato il seguente algoritmo per poter generare da tale distribuzione:

Generazione da esponenziale bivariata
1. Generare Y_1 da una distribuzione esponenziale di parametro $\xi_1 - \xi_0$ e Y_2 da una distribuzione esponenziale di parametro $\xi_2 - \xi_0$;
2. Generare Z da una distribuzione esponenziale di parametro ξ_0;
3. Definire $T_1 = \min(Y_1, Z)$ e $T_2 = \min(Y_2, Z)$.

Dato ciò, la distribuzione bivariata dei test nel gruppo dei malati avrà parametri $\xi_0 = 1$ e $\xi_1 = \xi_2 = 2$, mentre nel gruppo dei sani la distribuzione avrà parametri $\xi_0 = 1$ e $\xi_1 = \xi_2 = 10$.

Per tale simulazione, il vero *risk score* ha la seguente forma

$$\text{logit}(RS(\mathbb{T})) = \alpha + \beta_1 T_1 + \beta_2 T_2 \tag{3.2}$$

Come in §3.1 non si dispongono dei veri valori β e, pertanto, essi sono stimati secondo un modello logistico, generando 10^4 osservazioni da entrambi i gruppi: le stime di (3.2) sono riportate in Appendice, in Tabella B.2. Da ciò è stata ricavata la vera curva ROC, da cui il vero AUC è pari a 0.8790. Le simulazioni effettuate sono valutate per le stesse coppie (n_1, n_0) utilizzate in §3.1.

I modelli considerati sono il modello GAM, MARS (senza termini di interazioni includibili nel modello) e un modello logistico sovraspecificato (No-logit) espresso come

$$\text{logit}(RS(\mathbb{T})) = \alpha + \beta_1 T_1 + \beta_2 T_2 + \beta_3 T_1^2 + \beta_4 T_2^2$$

Simulazione 1

Considerando le condizioni iniziali, la Figura 3.5 mostra che la curva ROC del *risk score* stimato con il MARS è la più vicina a quella vera indipendentemente dalla dimensione campionaria e lo stesso risultato lo si ottiene anche quando lo si stima con il No-logit (essendo sovraspecificato). Stimando il *risk score* con il GAM, invece, risulta che la curva ROC derivante si avvicina a quella vera solo all'aumentare del numero di osservazioni.

AUC (n_1, n_0)	**GAM**	**MARS**	**No-logit**
(45, 35)	0.9159	0.8810	0.8851
	(0.0401)	(0.0398)	(0.0388)
(90, 70)	0.8977	0.8806	0.8821
	(0.0280)	(0.0281)	(0.0289)
(180, 140)	0.8883	0.8777	0.8793
	(0.0193)	(0.0189)	(0.0196)
(360, 280)	0.8846	0.8775	0.8795
	(0.0145)	(0.0139)	(0.0146)

Tabella 3.13: Simulazione 1, scenario 2. AUC medi con le deviazioni standard tra parentesi. Il vero AUC è 0.8790.

3.2.1 Trasformazioni di covariate

Le simulazioni seguenti prevedono delle trasformazioni dei dati simulati utilizzati per stimare i modelli proposti. In particolare, al fine di verificare come i modelli utilizzati rispondono ad errate specificazioni delle covariate, si trasforma uno dei due tests, in particolare il secondo come in §3.1, lasciando immutata la specificazione del regressore. Le trasformazioni considerate sono $T_2 = \frac{1}{T_2}$, $T_2 = T_2^3$ e $T_2 = \exp(T_2)$. I risultati che si ottengono trasformando entrambi i test sono gli stessi ai risultati di seguito descritti e, pertanto, non sono stati riportati.

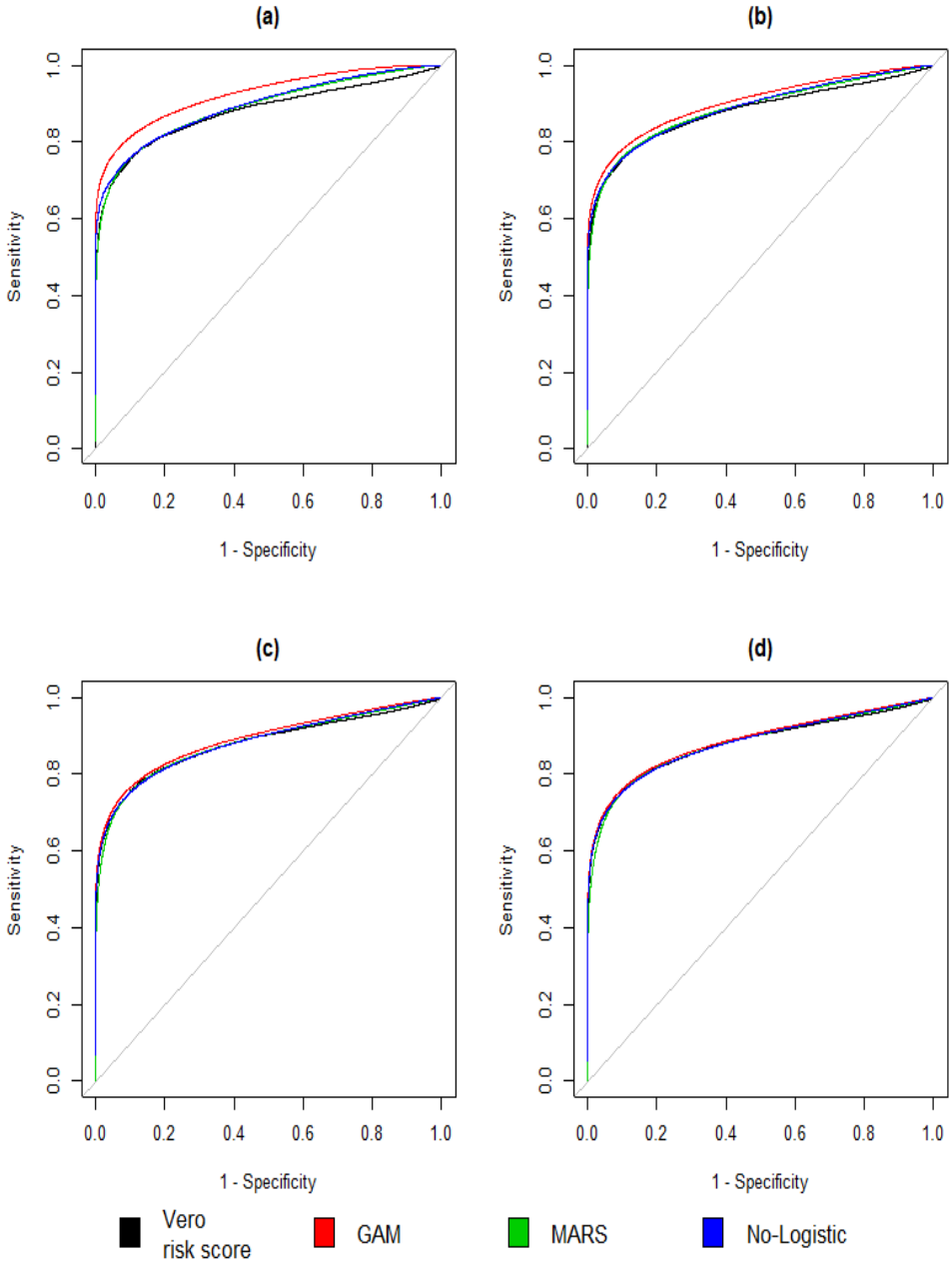


Figura 3.5: Simulazione 1, scenario 2. Confronto tra curve ROC stimate e curva vera per (a) $(n_1, n_0) = (45, 35)$, (b) $(n_1, n_0) = (90, 70)$, (c) $(n_1, n_0) = (180, 140)$ e (d) $(n_1, n_0) = (360, 280)$.

Simulazione 2

Utilizzando la prima trasformazione sul test, è possibile notare dalla Figura 3.6 che solo la curva ROC del *risk score* stimato con il GAM è molto vicina a quella reale: ciò, però, lo si osserva solo all'aumentare del numero di osservazioni. Per numerosità inferiori, la stima del *risk score* con gli altri modelli restituisce curve ROC più vicine a quella vera, diversamente da quanto si ottiene stimandolo con il GAM.

AUC (n_1, n_0)	**GAM**	**MARS**	**No-logit**
(45, 35)	0.9293	0.8602	0.8704
	(0.0379)	(0.0494)	(0.0500)
(90, 70)	0.9058	0.8613	0.8594
	(0.0265)	(0.0363)	(0.0439)
(180, 140)	0.8931	0.8568	0.8501
	(0.0194)	(0.0274)	(0.0406)
(360, 280)	0.8862	0.8562	0.8435
	(0.0142)	(0.0217)	(0.0383)

Tabella 3.14: Simulazione 2, scenario 2. AUC medi con le deviazioni standard tra parentesi. Il vero AUC è 0.8790. Nei dati simulati $T_2 = \frac{1}{T_2}$.

Simulazione 3

In Figura 3.21 (riportata in in §3.8) si mostra che, anche considerando la seconda trasformazione, i risultati sono analoghi a quelli della Figura 3.6, anche se, in questo caso, le differenze tra le curve ROC del *risk score* stimato con i modelli MARS e No-logit e quella vera sono meno marcate.

Simulazione 4

Considerando una trasformazione di tipo esponenziale, i risultati della Figura 3.22 (riportata in §3.8) sono gli stessi della Figura 3.5.

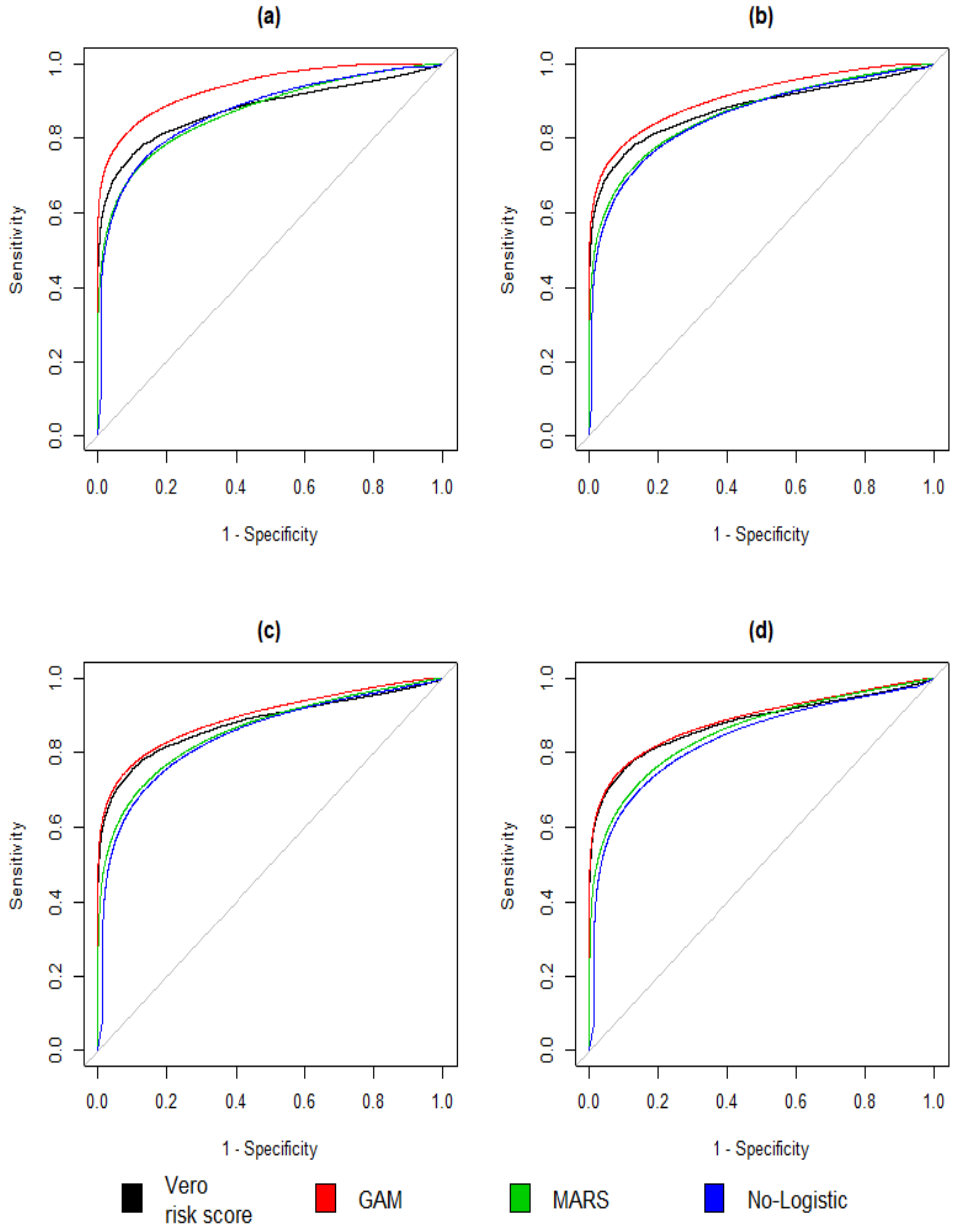


Figura 3.6: Simulazione 2, scenario 2. Confronto tra curve ROC stimate e curva vera per (a) $(n_1, n_0) = (45, 35)$, (b) $(n_1, n_0) = (90, 70)$, (c) $(n_1, n_0) = (180, 140)$ e (d) $(n_1, n_0) = (360, 280)$. Nei dati simulati $T_2 = \frac{1}{T_2}$.

AUC (n_1, n_0)	**GAM**	**MARS**	**No-logit**
(45, 35)	0.9233	0.8497	0.8806
	(0.0471)	(0.0495)	(0.0387)
(90, 70)	0.8986	0.8662	0.8754
	(0.0321)	(0.0350)	(0.0292)
(180, 140)	0.8878	0.8715	0.8736
	(0.0224)	(0.0222)	(0.0214)
(360, 280)	0.8836	0.8733	0.8727
	(0.0152)	(0.0144)	(0.0186)

Tabella 3.15: Simulazione 3, scenario 2. AUC medi con le deviazioni standard tra parentesi. Il vero AUC è 0.8790. Nei dati simulati $T_2 = T_2^3$.

AUC (n_1, n_0)	**GAM**	**MARS**	**No-logit**
(45, 35)	0.9165	0.8779	0.8854
	(0.0420)	(0.0417)	(0.0387)
(90, 70)	0.8979	0.8789	0.8816
	(0.0278)	(0.0281)	(0.0277)
(180, 140)	0.8904	0.8785	0.8800
	(0.0199)	(0.0200)	(0.0215)
(360, 280)	0.8834	0.8759	0.8778
	(0.0147)	(0.0147)	(0.0155)

Tabella 3.16: Simulazione 4, scenario 2. AUC medi con le deviazioni standard tra parentesi. Il vero AUC è 0.8790. Nei dati simulati $T_2 = \exp(T_2)$.

3.3 Scenario 3: simulazioni con 4 test

In tale scenario ci si pone nel caso in cui il numero di test è superiore a 2. Qui, lo status di malattia D è stato generato da una distribuzione di Bernoulli con un parametro θ (probabilità di malattia a priori) che è stato fissato a 0.1; si indica, inoltre, con $\mathbb{T}|D = 0 \sim N_4(0, \Sigma)$ la distribuzione normale a quattro dimensioni dei test relativa al gruppo dei sani, mentre $\mathbb{T}|D = 1 \sim N_4(\mu, \Sigma)$ la distribuzione normale a quattro dimensioni dei test relativa al gruppo dei malati. I parametri sono i seguenti

$$\mu = (1, 1, 1, 0.5)^T$$

$$\Sigma = \begin{bmatrix} 2.0 & 0.1 & -0.2 & 0.5 \\ 0.1 & 2.5 & 0.5 & -0.3 \\ -0.2 & 0.5 & 1.0 & 0.7 \\ 0.5 & -0.3 & 0.7 & 1.2 \end{bmatrix}$$

Il vero *risk score* relativo a tale scenario è il seguente

$$\text{logit}(RS(\mathbb{T})) = \alpha + \beta_1 T_1 + \beta_2 T_2 + \beta_3 T_3 + \beta_4 T_4 \tag{3.3}$$

Le stime dei parametri (riportate in Appendice, nella Tabella B.3), ottenute secondo l'approccio di stima del modello logistico, e la relativa curva ROC sono ottenute generando $n = 10^4$ osservazioni: il vero AUC è pari a 0.8642. Tutte le varie simulazioni effettuate in tale scenario sono state valutate in corrispondenza di differenti valori di n, ovvero 80, 160, 320 e 640. I modelli che vengono considerati in tale simulazione sono sempre il GAM, MARS (senza termini di interazioni includibili nel modello) e un modello logistico sovraspecificato (No-logit) avente la seguente forma

$$\text{logit}(RS(\mathbb{T})) = \alpha + \sum_{j=1}^{4} \beta_j T_j + \sum_{j=1}^{4} \beta_{4+j} T_j^2$$

Tutte le simulazioni che sono descritte in seguito sono state ripetute anche per un θ pari a 0.3 e hanno prodotto dei risultati piuttosto simili a quelli per $\theta = 0.1$. Pertanto, tali risultati non sono stati riportati.

Simulazione 1

Mantenendo le condizioni iniziali dello schema di simulazione, si osserva dalla Figura 3.7 che le curve ROC del *risk score* stimato con i vari modelli si avvicinano sempre più alla vera curva ROC solo quando aumenta n.

AUC (n)	**GAM**	**MARS**	**No-logit**
(80)	0.9947	0.8894	0.9363
	(0.0120)	(0.1050)	(0.0545)
(160)	0.9579	0.9001	0.9058
	(0.0328)	(0.0635)	(0.0404)
(320)	0.9128	0.8987	0.8894
	(0.0313)	(0.0344)	(0.0306)
(640)	0.8903	0.8897	0.8811
	(0.0234)	(0.0227)	(0.0222)

Tabella 3.17: Simulazione 1, scenario 3. AUC medi con le deviazioni standard tra parentesi. Il vero AUC è 0.8642.

3.3.1 Trasformazioni di covariate

La seconda parte delle simulazioni in tale scenario consiste, come già visto prima, nel trasformare una covariata nei set di dati simulati per stimare i modelli per capire come questi rispondono a una errata specificazione delle covariate; la specificazione del regressore è sempre la stessa. La trasformazione è fatta su T_3 poichè dalla Tabella B.3 si osserva che è il parametro con l'effetto più alto.

Simulazione 2

In tale simulazione, la trasformazione considerata è $T_3 = \frac{1}{T_3}$. Dai risultati riportati in Figura 3.8 emerge che le curve ROC del *risk score* stimato con il MARS e il No-logit sono vicine a quella vera solo quando si hanno poche osservazioni (pannelli (a) e (b)). Quando n aumenta, invece, solo quella ottenuta stimando il *risk score* con il GAM è vicina a quella vera.

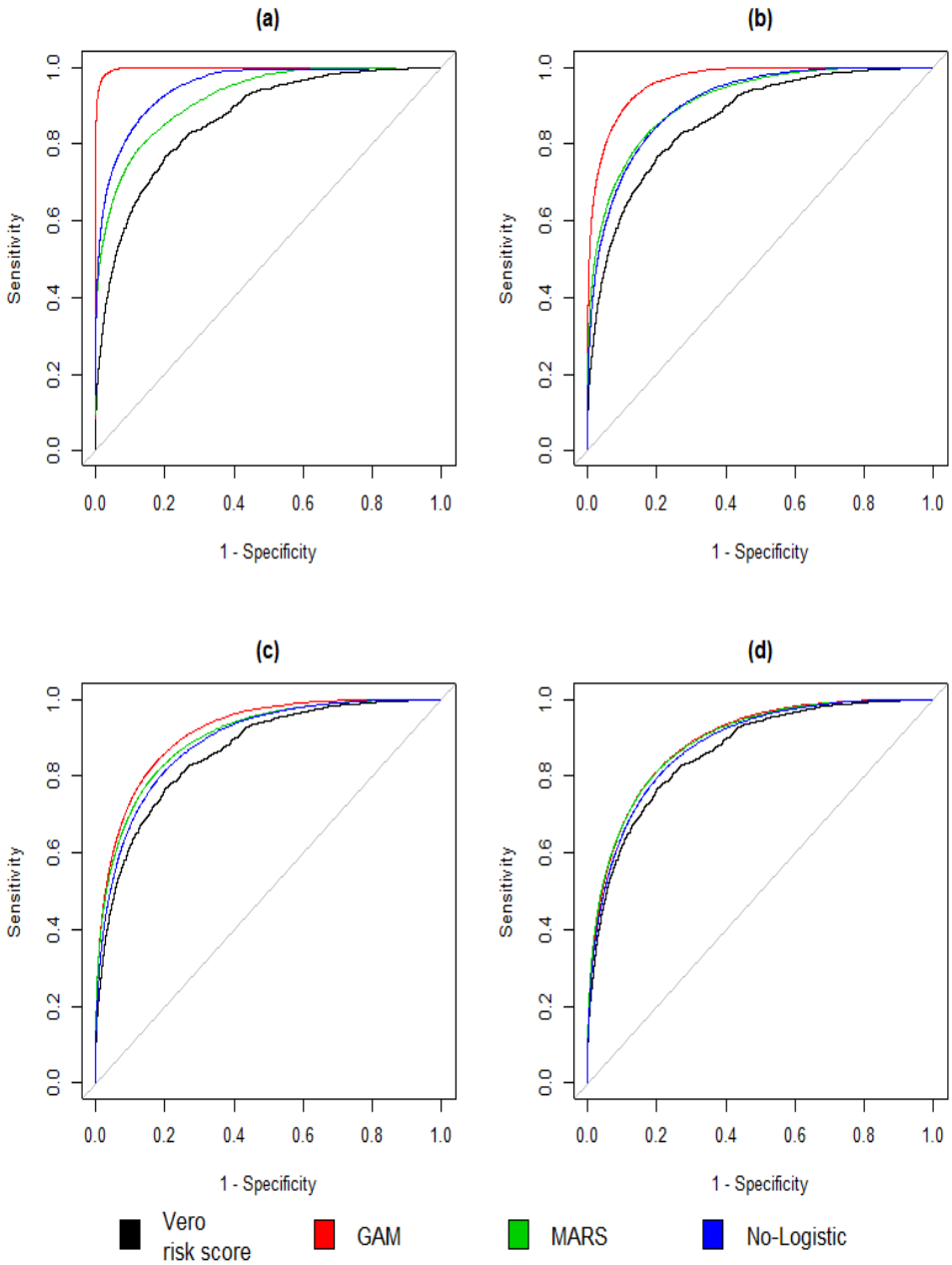


Figura 3.7: Simulazione 1, scenario 3. Confronto tra curve ROC stimate e curva vera per (a) $n = 80$, (b) $n = 160$, (c) $n = 320$ e (d) $n = 640$.

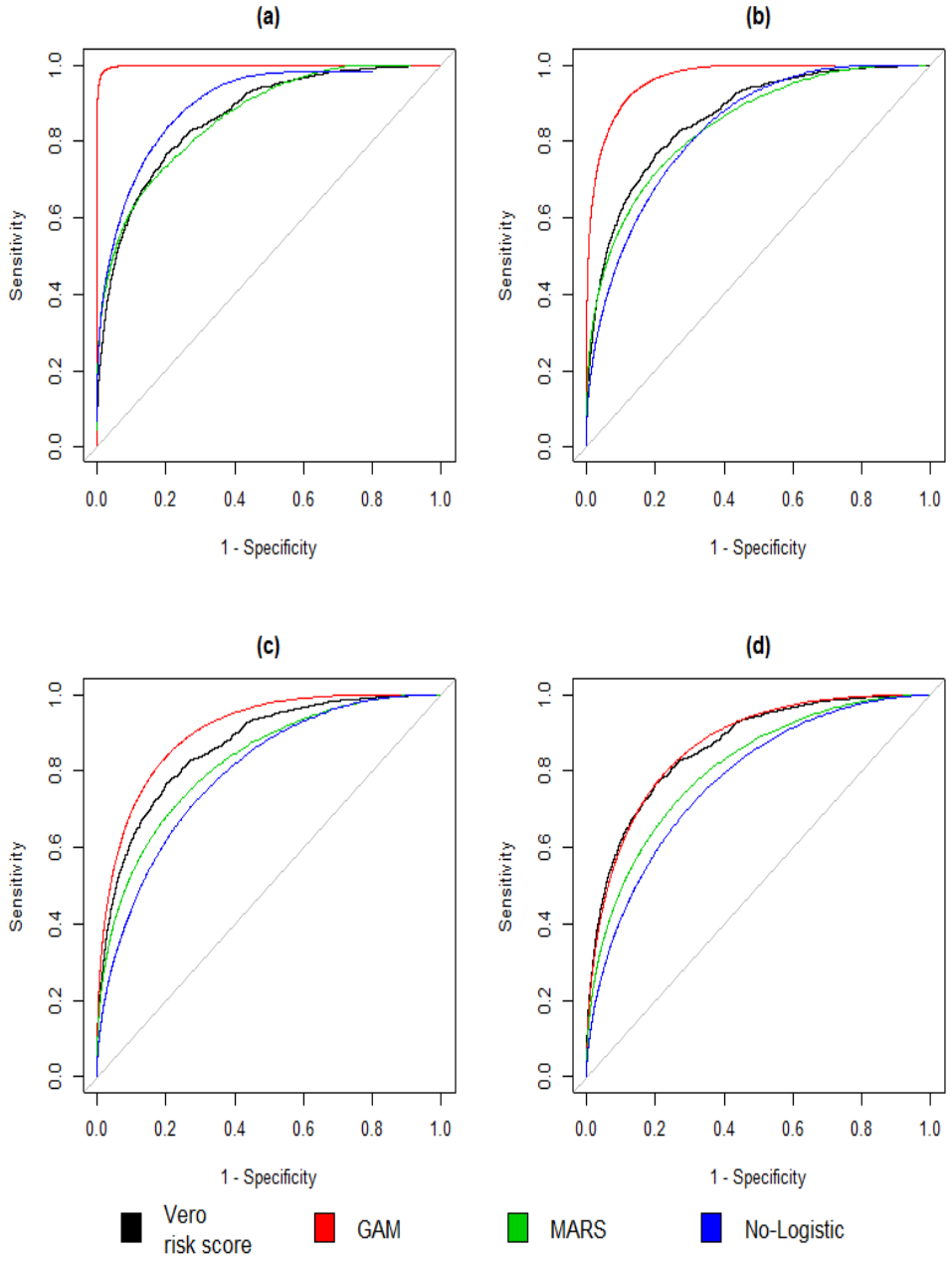


Figura 3.8: Simulazione 2, scenario 3. Confronto tra curve ROC stimate e curva vera per (a) $n = 80$, (b) $n = 160$, (c) $n = 320$ e (d) $n = 640$. Nei dati simulati $T_3 = \frac{1}{T_3}$.

AUC (n)	**GAM**	**MARS**	**No-logit**
(80)	0.9937	0.8209	0.8858
	(0.0208)	(0.1438)	(0.0838)
(160)	0.9549	0.8237	0.8212
	(0.0446)	(0.1010)	(0.0599)
(320)	0.8991	0.8160	0.7947
	(0.0432)	(0.0618)	(0.0443)
(640)	0.8660	0.8048	0.7773
	(0.0340)	(0.0489)	(0.0314)

Tabella 3.18: Simulazione 2, scenario 3. AUC medi con le deviazioni standard tra parentesi. Il vero AUC è 0.8642. Nei dati simulati $T_3 = \frac{1}{T_3}$.

Simulazione 3

Per tale simulazione, la trasformazione considerata è $T_3 = T_3^3$. In Figura 3.9 si nota che all'aumentare di n, le curve ROC del *risk score* stimato con i modelli GAM e MARS sono sempre più vicine a quella reale, a differenza di quando lo si stima con il No-logit. La curva ROC del *risk score* stimato con il GAM, però, è leggermente diversa da quella reale, diversamente da quella ottenuta stimandolo con il MARS (pannello (d)).

Simulazione 4

L'ultima trasformazione considera è $T_3 = \exp(T_3)$. I risultati in Figura 3.23 (riportata in §3.8) sono molto simili a quelli osservati in Figura 3.9. L'unica differenza è che la curva ROC del *risk score* stimato con il MARS è leggermente più lontata da quella vera rispetto a prima, mentre la curva ottenuta stimando il *risk score* con il No-logit è più vicina.

AUC (n)	**GAM**	**MARS**	**No-logit**
(80)	0.9964	0.8734	0.9096
	(0.0111)	(0.1215)	(0.0694)
(160)	0.9603	0.8875	0.8699
	(0.0328)	(0.0656)	(0.0605)
(320)	0.9121	0.8801	0.8382
	(0.0324)	(0.0390)	(0.0553)
(640)	0.8907	0.8733	0.8182
	(0.0235)	(0.0258)	(0.0644)

Tabella 3.19: Simulazione 3, scenario 3. AUC medi con le deviazioni standard tra parentesi. Il vero AUC è 0.8642. Nei dati simulati $T_3 = T_3^3$.

AUC (n)	**GAM**	**MARS**	**No-logit**
(80)	0.9966	0.8843	0.9247
	(0.0115)	(0.1128)	(0.0572)
(160)	0.9592	0.8984	0.8838
	(0.0354)	(0.0614)	(0.0463)
(320)	0.9135	0.8956	0.8606
	(0.0310)	(0.0354)	(0.0348)
(640)	0.8923	0.8860	0.8467
	(0.0227)	(0.0240)	(0.0275)

Tabella 3.20: Simulazione 4, scenario 3. AUC medi con le deviazioni standard tra parentesi. Il vero AUC è 0.8642. Nei dati simulati $T_3 = \exp(T_3)$.

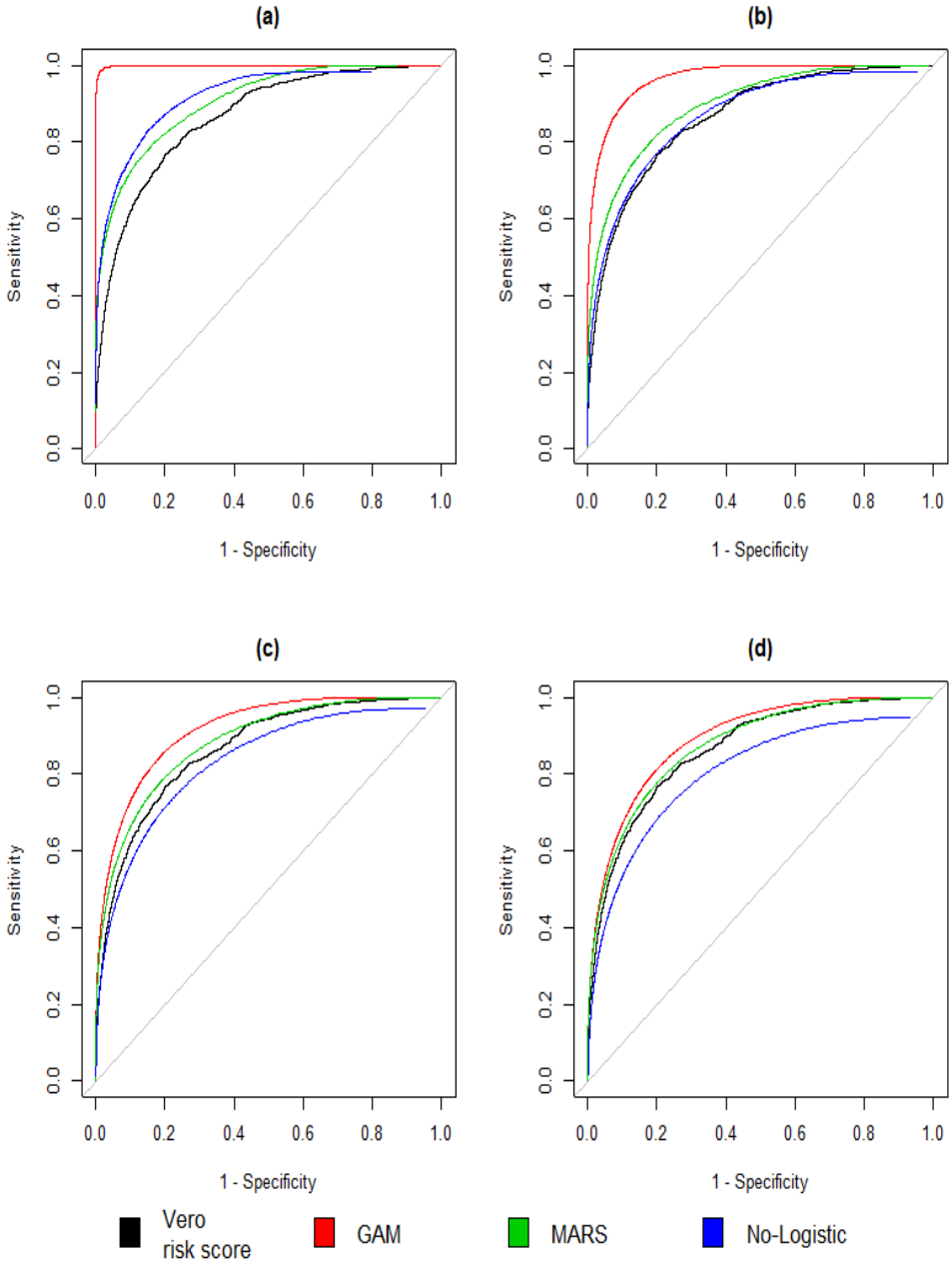


Figura 3.9: Simulazione 3, scenario 3. Confronto tra curve ROC stimate e curva vera per (a) $n = 80$, (b) $n = 160$, (c) $n = 320$ e (d) $n = 640$. Nei dati simulati $T_3 = T_3^3$.

3.4 Scenario 4: simulazione con 4 test - seconda parte

Basandosi sullo scenario in §3.3, se ne costruisce uno nuovo dove vengono modificati alcuni elementi. In particolare, si considerano matrici di varianze e covarianze differenti nelle due condizioni: tali matrici sono le seguenti

$$\Sigma_0 = \begin{bmatrix} 1.00 & -0.10 & 0.20 & -0.25 \\ -0.10 & 1.25 & 0.25 & -0.30 \\ 0.20 & 0.25 & 2.00 & -0.35 \\ -0.25 & -0.30 & -0.35 & 1.50 \end{bmatrix}$$

$$\Sigma_1 = \begin{bmatrix} 2.0 & 0.1 & -0.2 & -0.5 \\ 0.1 & 2.5 & 0.5 & -0.3 \\ -0.2 & 0.5 & 1.0 & 0.7 \\ -0.5 & -0.3 & 0.7 & 1.2 \end{bmatrix}$$

Si ha, dunque, $\mathbb{T}|D = 1 \sim N_4(\mu, \Sigma_1)$ e $\mathbb{T}|D = 0 \sim N_4(0, \Sigma_0)$. Lo status di malattia, invece, è generato da una distribuzione di Bernoulli con parametro θ (probabilità di malattia a priori) pari a 0.3. Il vero *risk score* assume in questo caso la seguente forma

$$\text{logit}(RS(\mathbb{T})) = \alpha + \sum_{j=1}^{4} \beta_j T_j + \sum_{j=1}^{4} \beta_{4+j} T_j^2 + \sum_{j=1}^{4} \sum_{i>j} \beta_{8+j} T_j T_i \qquad (3.4)$$

Le stime dei parametri in (3.4) sono riportate in Appendice, nella Tabella B.4: per ricavare tali stime (e, di conseguenza, anche la curva ROC) sono state generate $n = 10^4$ osservazioni e stimando (3.4) come un modello logistico, e risulta che il vero AUC è pari a 0.9242.

In tale scenario sono considerati il modello GAM, due modelli MARS di cui il primo è vincolato a non inserire alcun effetto di interazione ($d = 1$) e il secondo ad avere solo interazioni di primo ordine ($d = 2$), e un modello logistico non correttamente specificato che contiene i soli effetti lineari di (3.4). Le valutazioni sono fatte per gli stessi n utilizzati in §3.3.

Simulazione 1

Sotto le condizioni iniziali, è possibile osservare dalla Figura 3.10 che per poche osservazioni le curve ROC del *risk score* stimato con i modelli MARS sono sempre le più vicine alla vera curva; all'aumentare di n, però, solo la curva ottenuta stimando il *risk score* con il MARS ($d = 2$) è vicina a quella reale, mentre le altre curve sono simili tra loro ma diverse da quella reale.

AUC (n)	**GAM**	**MARS** ($d = 1$)	**MARS** ($d = 2$)	**No-Logit**
(80)	0.9794	0.9109	0.9191	0.9138
	(0.0228)	(0.0536)	(0.0594)	(0.0392)
(160)	0.9312	0.9043	0.9264	0.8998
	(0.0281)	(0.0318)	(0.0289)	(0.0286)
(320)	0.9036	0.8951	0.9231	0.8917
	(0.0220)	(0.0216)	(0.0190)	(0.0206)
(640)	0.8949	0.8892	0.9214	0.8894
	(0.0158)	(0.0157)	(0.0121)	(0.0154)

Tabella 3.21: Simulazione 1, scenario 4. AUC medi con le deviazioni standard tra parentesi. Il vero AUC è 0.9242.

3.4.1 Trasformazioni di covariate

Come già fatto negli altri scenari, le simulazioni seguenti prevedono trasformazioni di covariate nei set di dati usati per stimare i modelli utilizzati al fine di verificare la risposta di questi in presenza di un'errata specificazione delle covariate; anche qui la specificazione del regressore rimane la stessa. In questo scenario, come già fatto in §3.3, si trasforma il terzo test per lo stesso motivo esposto precedentemente.

Simulazione 2, 3 e 4

I risultati riportati in Figura 3.11 ($T_3 = \frac{1}{T_3}$) e 3.24 ($T_3 = T_3^3$) (di cui l'ultima in §3.8) mostrano che la curva ROC del *risk score* stimato con i

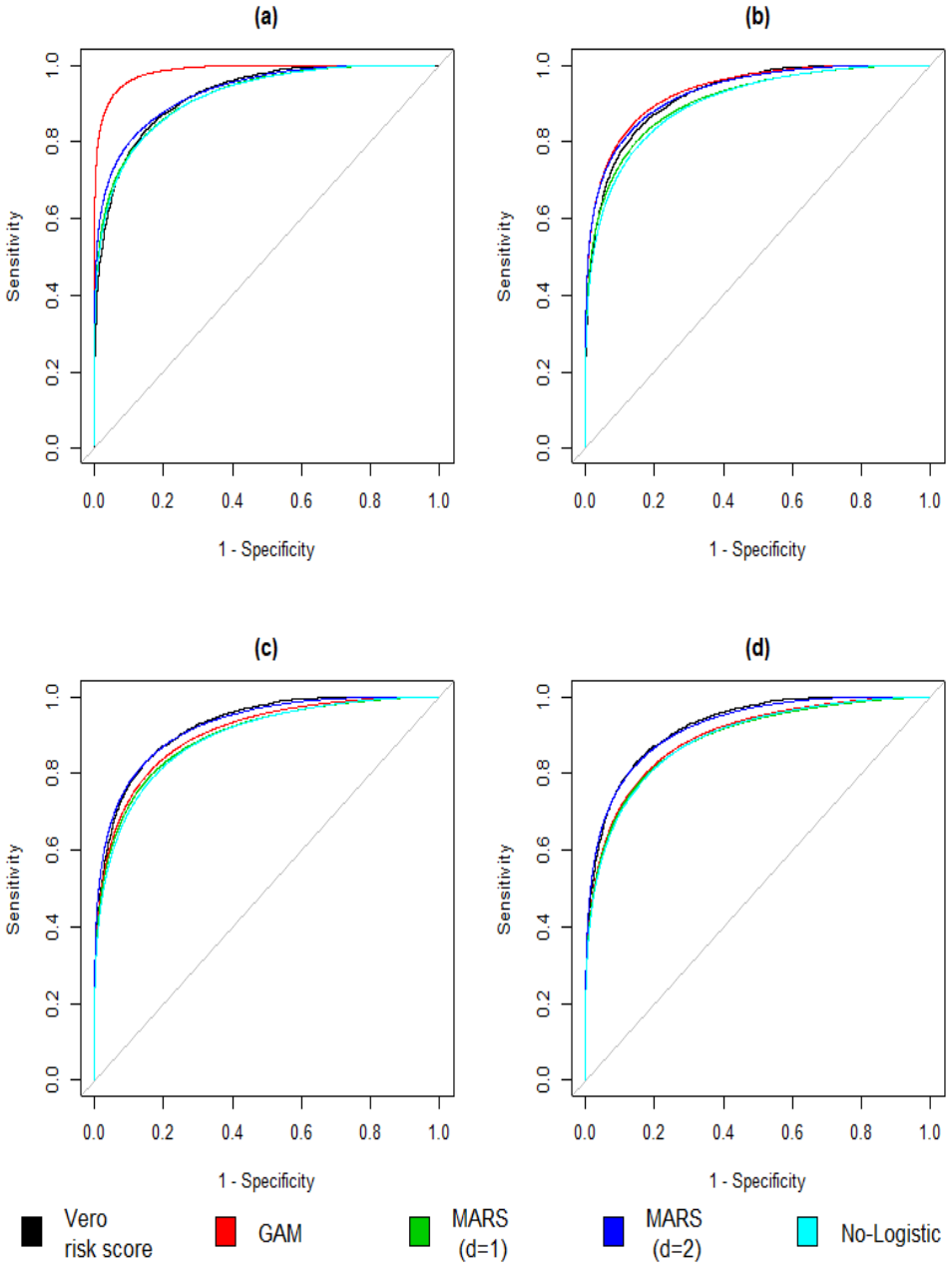


Figura 3.10: Simulazione 1, scenario 4. Confronto tra curve ROC stimate e curva vera per (a) $n = 80$, (b) $n = 160$, (c) $n = 320$ e (d) $n = 640$.

MARS e il No-logit sono sempre vicine alla vera curva per poche osservazioni. Quando n aumenta, risulta che le curve ottenute stimando il *risk score* con il GAM e il MARS ($d = 2$) sono vicine a quella vera pur rimanendo leggermente differenti; non è possibile dire altrettanto nel caso del No-logit poichè la curva risultante è molto diversa da quella vera. La Figura 3.25 ($T_3 = \exp(T_3)$) in §3.8 mostra risultati molto simili a quelli visti in Figura 3.10.

AUC (n)	**GAM**	**MARS** ($d = 1$)	**MARS** ($d = 2$)	**No-Logit**
(80)	0.9858	0.8995	0.9004	0.8930
	(0.0257)	(0.0581)	(0.0664)	(0.0474)
(160)	0.9346	0.8953	0.9000	0.8737
	(0.0291)	(0.0361)	(0.0410)	(0.0362)
(320)	0.9083	0.8868	0.8922	0.8636
	(0.0212)	(0.0248)	(0.0277)	(0.0261)
(640)	0.8963	0.8782	0.8834	0.8584
	(0.0154)	(0.0202)	(0.0223)	(0.0185)

Tabella 3.22: Simulazione 2, scenario 4. AUC medi con le deviazioni standard tra parentesi. Il vero AUC è 0.9242. Nei dati simulati $T_3 = \frac{1}{T_3}$.

3.5 Scenario 5: due test indipendenti

In tale scenario, si considerano due test indipendenti che provengono da distribuzioni differenti e non sono condizionate dallo status di malattia. In tale scenario, il vero *risk score* ha la seguente forma

$$\text{logit}(RS(\mathbb{T})) = -1 + T_1 + T_2 + 0.5T_1^2 - T_2^2 - T_1T_2 \tag{3.5}$$

A differenza degli altri scenari, i veri parametri di (3.5) sono noti e sono $\beta = (-1, 1, 1, 0.5, -1, -1)$. Per generare la variabile D, si procede con il seguente schema

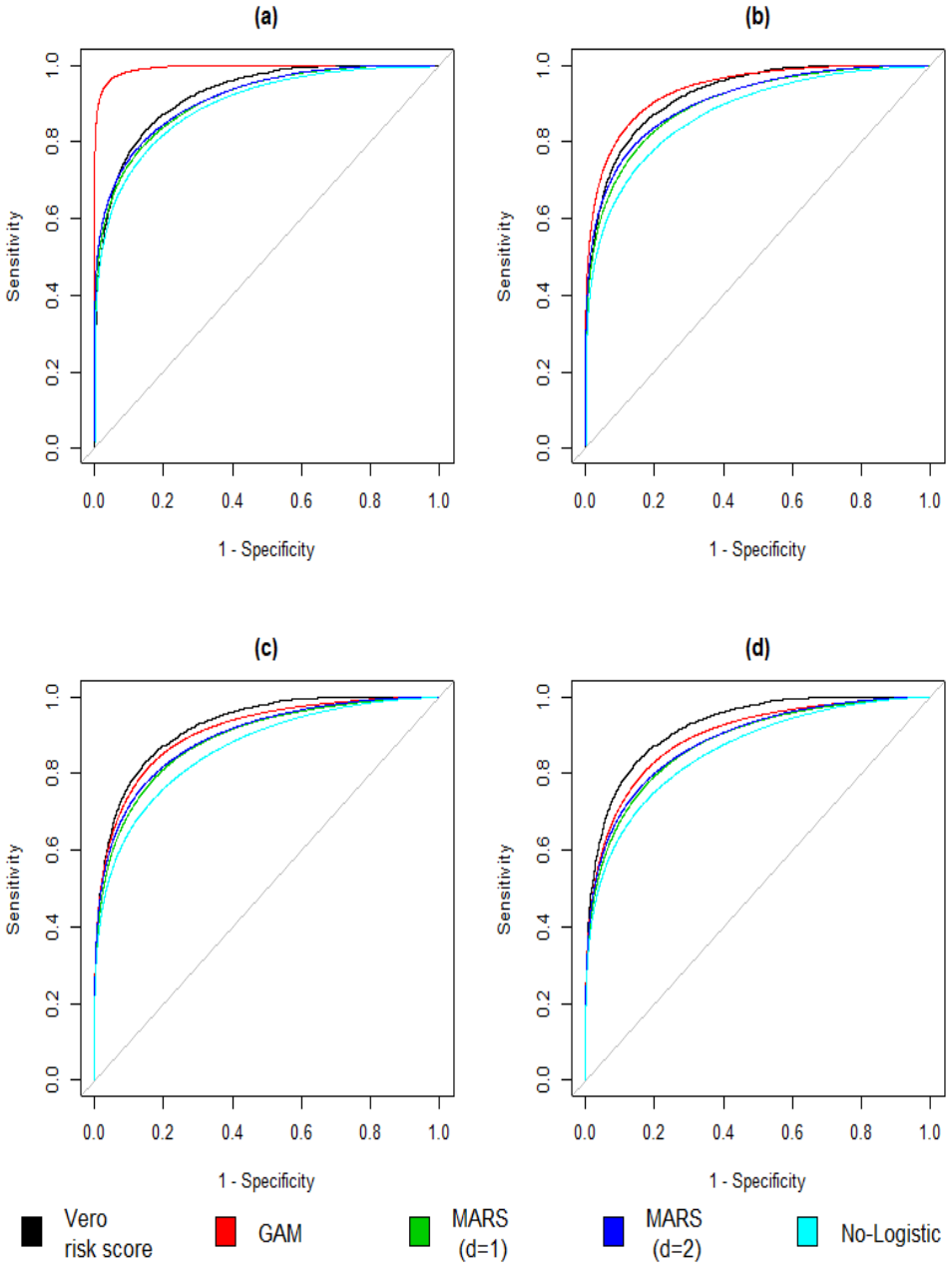


Figura 3.11: Simulazione 2, scenario 4. Confronto tra curve ROC stimate e curva vera per (a) $n = 80$, (b) $n = 160$, (c) $n = 320$ e (d) $n = 640$. Nei dati simulati $T_3 = \frac{1}{T_3}$.

AUC (n)	**GAM**	**MARS** ($d = 1$)	**MARS** ($d = 2$)	**No-Logit**
(80)	0.9832	0.9028	0.8967	0.9021
	(0.0240)	(0.0547)	(0.0641)	(0.0442)
(160)	0.9345	0.8983	0.9069	0.8879
	(0.0275)	(0.0339)	(0.0357)	(0.0315)
(320)	0.9056	0.8903	0.9028	0.8777
	(0.0213)	(0.0217)	(0.0226)	(0.0229)
(640)	0.8949	0.8848	0.9013	0.8728
	(0.0153)	(0.0153)	(0.0166)	(0.0192)

Tabella 3.23: Simulazione 3, scenario 4. AUC medi con le deviazioni standard tra parentesi. Il vero AUC è 0.9242. Nei dati simulati $T_3 = T_3^3$.

AUC (n)	**GAM**	**MARS** ($d = 1$)	**MARS** ($d = 2$)	**No-Logit**
(80)	0.9845	0.9120	0.9199	0.9002
	(0.0237)	(0.0520)	(0.0565)	(0.0414)
(160)	0.9309	0.9016	0.9252	0.8797
	(0.0305)	(0.0337)	(0.0313)	(0.0335)
(320)	0.9046	0.8927	0.9234	0.8684
	(0.0212)	(0.0223)	(0.0187)	(0.0262)
(640)	0.8949	0.8877	0.9214	0.8639
	(0.0152)	(0.0160)	(0.0121)	(0.0206)

Tabella 3.24: Simulazione 4, scenario 4. AUC medi con le deviazioni standard tra parentesi. Il vero AUC è 0.9242. Nei dati simulati $T_3 = \exp(T_3)$.

Generazione di D
1. Generare $n = 10^4$ coppie di osservazioni (T_1, T_2) dalle rispettive distribuzioni;
2. Sostituire la generica coppia (T_{i1}, T_{i2}) in (3.5), dalla quale, utilizzando l'anti trasformata, si ottiene la stima del *risk score* in corrispondenza dell'i-esima coppia, $\hat{\theta}_i$;
3. Dato $\hat{\theta}_i$, D_i è generato da una distribuzione di Bernoulli con probabilità di successo pari a $\hat{\theta}_i$.

Da tale procedura risulta che la probabilità di malattia a priori sia stimata a 0.34 circa, mentre il vero AUC è pari a 0.7904.

In questo scenario si considerano sempre il modello GAM, due MARS di cui uno non vincolato a non avere interazioni ($d = 1$) mentre l'altro interazioni di primo ordine solamente ($d = 2$), e un modello logistico non correttamente specificato (No-logit) espresso come

$$\text{logit}(RS(\mathbb{T})) = \alpha + \beta_1 T_1 + \beta_2 T_2$$

Le valutazioni sono fatte con gli stessi valori di n considerati in §3.3 e §3.4.

Simulazione 1

In tale simulazione, sono mantenute le condizioni iniziali, e i risultati sono riportati in Figura 3.12, da cui si mostra ancora una volta come la curva ROC del *risk score* stimato con il GAM si avvicini alla vera curva solo quando n aumenta; le curve ottenute stimando il *risk score* con i due modelli MARS, invece, sono molto simili alla curva reale anche quando n è piccolo. La curva ROC ottenuta stimando il *risk score* con il No-logit, infine, è sempre differente dalla curva reale indipendentemente da n.

3.5.1 Trasformazioni di covariate

Si effettuano adesso altre simulazioni che prevedono trasformazioni di covariate nei set di dati utilizzati per stimare i modelli usati e, nello specifico, il secondo test, con lo scopo di valutare la risposta di tali modelli in presenza di

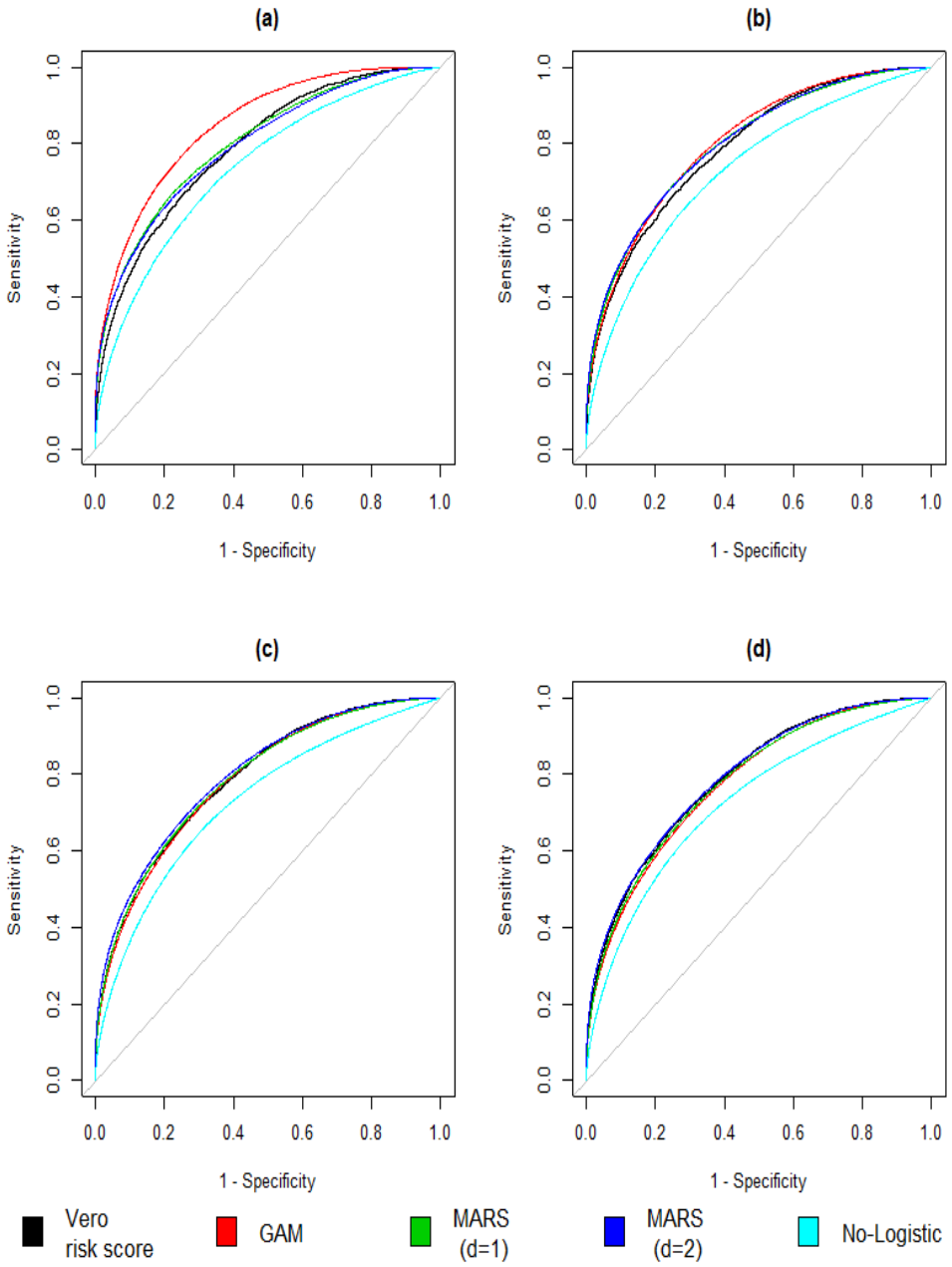


Figura 3.12: Simulazione 1, scenario 5. Confronto tra curve ROC stimate e curva vera per (a) $n = 80$, (b) $n = 160$, (c) $n = 320$ e (d) $n = 640$.

AUC (n)	**GAM**	**MARS** ($d = 1$)	**MARS** ($d = 2$)	**No-Logit**
(80)	0.8445	0.7837	0.7789	0.7372
	(0.0541)	(0.0786)	(0.0802)	(0.0597)
(160)	0.8018	0.7903	0.7931	0.7316
	(0.0398)	(0.0526)	(0.0524)	(0.0426)
(320)	0.7861	0.7866	0.7970	0.7285
	(0.0285)	(0.0332)	(0.0308)	(0.0308)
(640)	0.7784	0.7807	0.7924	0.7262
	(0.0186)	(0.0197)	(0.0199)	(0.0206)

Tabella 3.25: Simulazione 1, scenario 5. AUC medi con le deviazioni standard tra parentesi. Il vero AUC è 0.7904.

una non corretta specificazione delle covariate; la specificazione del regressore è sempre la stessa. Le trasformazioni sono sempre le stesse che sono state utilizzate per gli scenari precedenti.

Simulazione 2, 3 e 4

I risultati delle simulazioni presenti in Figura 3.26, 3.27 e 3.28 (riportate in §3.8) sono molto simili a quelli descritti in Figura 3.12.

3.6 Intervalli di confidenza bootstrap

Oltre che a valutare tramite simulazioni l'adeguatezza dei modelli non parametrici come stima del *risk score* rispetto alla sua vera forma nei differenti scenari, un altro aspetto rilevante considerato è relativo all'inferenza sull'AUC.

A tale scopo, si propone di utilizzare un intervallo di confidenza bootstrap di tipo percentile, utilizzando un campionamento stratificato per status della malattia in maniera tale da mantenere la proporzione di sani e malati nei campioni bootstrap estratti e utilizzati per stimare i modelli. Per valutare se l'approccio bootstrap è adeguato per costruire tali intervalli di confidenza,

AUC (n)	**GAM**	**MARS** ($d = 1$)	**MARS** ($d = 2$)	**No-Logit**
(80)	0.8534	0.7617	0.7540	0.7176
	(0.0580)	(0.0781)	(0.0832)	(0.0657)
(160)	0.8067	0.7588	0.7567	0.7071
	(0.0401)	(0.0571)	(0.0592)	(0.0478)
(320)	0.7894	0.7629	0.7679	0.7068
	(0.0291)	(0.0427)	(0.0436)	(0.0332)
(640)	0.7797	0.7561	0.7709	0.7035
	(0.0193)	(0.0316)	(0.0288)	(0.0231)

Tabella 3.26: Simulazione 2, scenario 5. AUC medi con le deviazioni standard tra parentesi. Il vero AUC è 0.7904. Nei dati simulati $T_2 = \frac{1}{T_2}$.

AUC (n)	**GAM**	**MARS** ($d = 1$)	**MARS** ($d = 2$)	**No-Logit**
(80)	0.8589	0.7738	0.7695	0.7285
	(0.0512)	(0.0811)	(0.0859)	(0.0603)
(160)	0.8088	0.7825	0.7833	0.7270
	(0.0378)	(0.0537)	(0.0539)	(0.0423)
(320)	0.7874	0.7831	0.7906	0.7223
	(0.0278)	(0.0319)	(0.0310)	(0.0294)
(640)	0.7792	0.7803	0.7893	0.7223
	(0.0199)	(0.0204)	(0.0202)	(0.0216)

Tabella 3.27: Simulazione 3, scenario 5. AUC medi con le deviazioni standard tra parentesi. Il vero AUC è 0.7904. Nei dati simulati $T_2 = T_2^3$.

AUC (n)	**GAM**	**MARS** ($d = 1$)	**MARS** ($d = 2$)	**No-Logit**
(80)	0.8452	0.7826	0.7791	0.7323
	(0.0551)	(0.0788)	(0.0852)	(0.0586)
(160)	0.8027	0.7891	0.7926	0.7285
	(0.0403)	(0.0515)	(0.0536)	(0.0428)
(320)	0.7867	0.7863	0.7974	0.7240
	(0.0277)	(0.0315)	(0.0305)	(0.0307)
(640)	0.7796	0.7799	0.7932	0.7229
	(0.0188)	(0.0207)	(0.0195)	(0.0221)

Tabella 3.28: Simulazione 4, scenario 5. AUC medi con le deviazioni standard tra parentesi. Il vero AUC è 0.7904. Nei dati simulati $T_2 = \exp T_2$.

si è verificato quanto le stime degli AUC (e la relativa variabilità) ottenute tramite tale approccio sono simili a quelle che si ottengono dalle simulazioni Monte Carlo effettuate nelle sezioni precedenti. Per fare ciò, sono stati considerati i primi due scenari, effettuando 1000 replicazioni, per ciascuna delle quali sono stati estratti 200 campioni bootstrap. Per il primo scenario, sono stati considerati i parametri definiti in §3.1.1, mentre per il secondo scenario i parametri sono quelli definiti in §3.2.

In Tabella 3.29 si riportano i risultati relativi al primo scenario: si nota che per $(n_1, n_0) = (45, 35)$, un numero di nodi $k = 10$ per le *spline* di ogni test nel GAM porta a un sovra-adattamento ai dati poichè la variabilità delle stime bootstrap è nettamente inferiore a quella ottenuta dalle simulazioni Monte Carlo. A parità di dimensione campionaria e dimezzando il numero di nodi, però, tale aspetto è più attenuato. Tale effetto non si nota per $(n_1, n_0) = (90, 70)$ e per $k = 10$. Nel caso del modello MARS, la variabilità delle stime bootstrap è simile a quelle Monte Carlo indipendentemente dalla dimensione campionaria.

Anche ponendosi sotto il secondo scenario, è possibile notare le stesse caratteristiche notate per il primo scenario, come mostrano i risultati riportati in Tabella 3.30.

Dai risultati ottenuti, è possibile concludere che l'utilizzo di un approccio

		AUC Monte Carlo	**Dev. standard Monte Carlo**	**Dev. standard bootstrap**
(45, 35) $k = 5$	GAM	0.9539	0.0243	0.0198
	MARS ($d = 1$)	0.9522	0.0283	0.0225
	MARS ($d = 2$)	0.9533	0.0272	0.0222
(45, 35) $k = 10$	GAM	0.9656	0.0218	0.0126
	MARS ($d = 1$)	0.9522	0.0283	0.0225
	MARS ($d = 2$)	0.9533	0.0272	0.0222
(90, 70) $k = 10$	GAM	0.9527	0.0174	0.0141
	MARS ($d = 1$)	0.9490	0.0185	0.0163
	MARS ($d = 2$)	0.9521	0.0190	0.0166

Tabella 3.29: Stime Monte Carlo dell'AUC (deviazioni standard Monte Carlo e bootstrap) per il primo scenario. k indica il numero di nodi nella spline cubica di regressione per ogni test nel GAM.

		AUC Monte Carlo	**Dev. standard Monte Carlo**	**Dev. standard bootstrap**
(45, 35) $k = 5$	GAM	0.8890	0.0368	0.0351
	MARS	0.8796	0.0409	0.0386
(45, 35) $k = 10$	GAM	0.9148	0.0406	0.0281
	MARS	0.8796	0.0409	0.0386
(90, 70) $k = 10$	GAM	0.8963	0.0280	0.0253
	MARS	0.8792	0.0272	0.0273

Tabella 3.30: Stime Monte Carlo dell'AUC (deviazioni standard Monte Carlo e bootstrap) per il secondo scenario. k indica il numero di nodi nella spline cubica di regressione per ogni test nel GAM.

bootstrap per l'inferenza sull'AUC quando il *risk score* è modellato tramite un GAM è condizionato dal numero di osservazioni e dal numero di nodi scelto per ciascuna spline utilizzata sui tests, a differenza del MARS.

3.7 Conclusioni

Alla luce dei risultati ottenuti dalle varie simulazioni, è possibile notare un elemento chiave, ovvero che la curva ROC ottenuta stimando il *risk score* con il modello GAM tende ad essere vicina alla vera curva solo all'aumentare del numero di osservazioni; se il *risk score* è stimato con il MARS (con/senza effetti di interazione inseribili), invece, la curva ROC risultante è vicina a quella vera anche quando il numero di osservazioni è ridotto.

Ponendosi nel caso di una non corretta specificazione delle covariate, è possibile notare che stimando il *risk score* con il modello GAM, la curva ROC derivante è nella maggior parte dei casi vicina a quella vera, sempre per elevate dimensioni campionarie. Solo in un ridotto numero di casi ciò accade quando si stima il *risk score* con il MARS (soprattutto quando può inserire effetti di interazione). Quando le dimensioni campionarie sono abbastanza elevate, nella gran parte dei casi si osserva che il *risk score* stimato con entrambi i modelli restituisce delle curve ROC che sono molto simili sia tra di loro che con la vera curva ROC.

Come già stato evidenziato in §3.6, la costruzione di un intervallo di confidenza bootstrap per l'inferenza sull'AUC quando il *risk score* è modellato attraverso il GAM è condizionato dalla dimensione campionaria e dal numero di nodi scelto per ciascuna spline utilizzata sui tests, a differenza del modello MARS.

3.8 Appendice: grafici

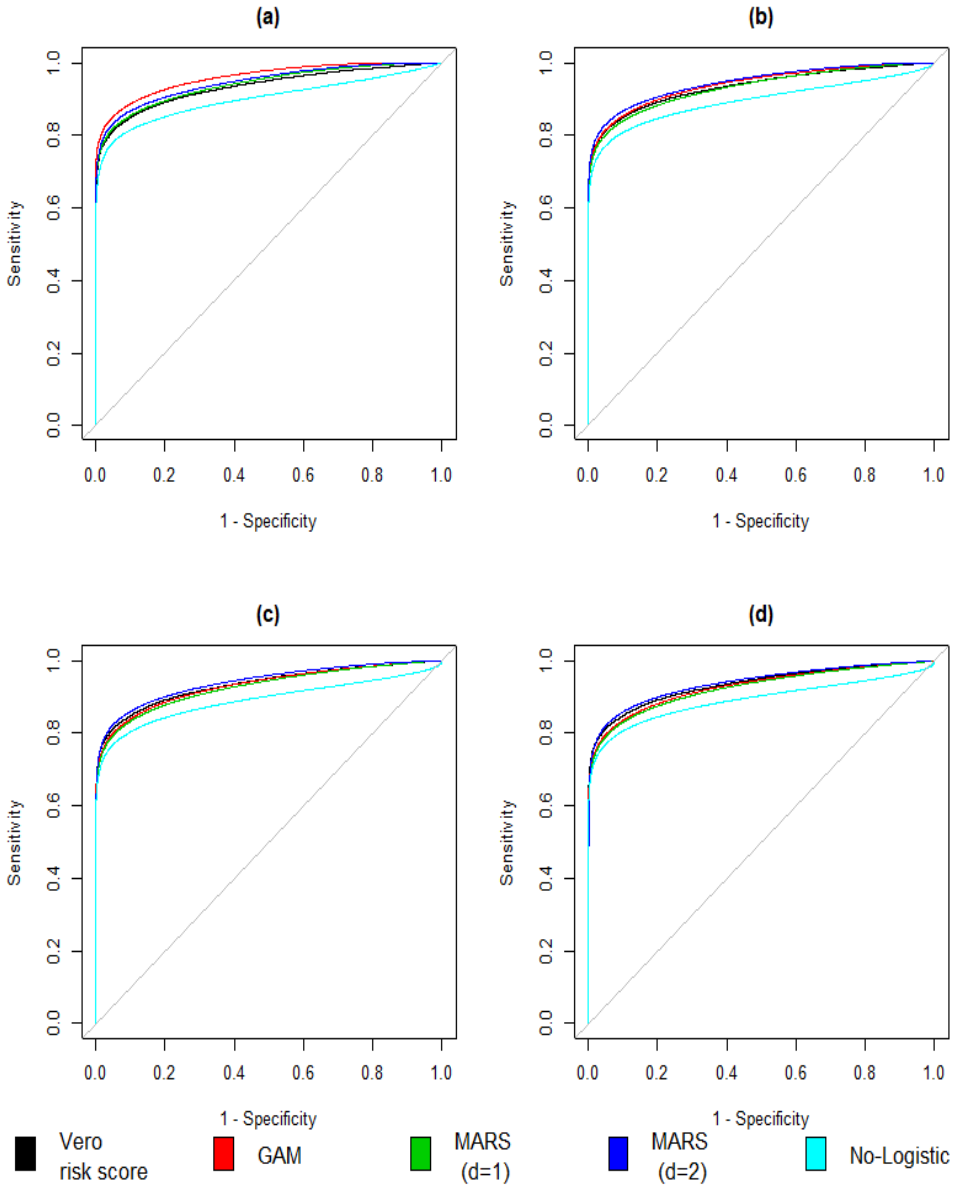


Figura 3.13: Simulazione 2, scenario 1. Confronto tra curve ROC stimate e curva vera per (a) $(n_1, n_0) = (45, 35)$, (b) $(n_1, n_0) = (90, 70)$, (c) $(n_1, n_0) = (180, 140)$ e (d) $(n_1, n_0) = (360, 280)$.

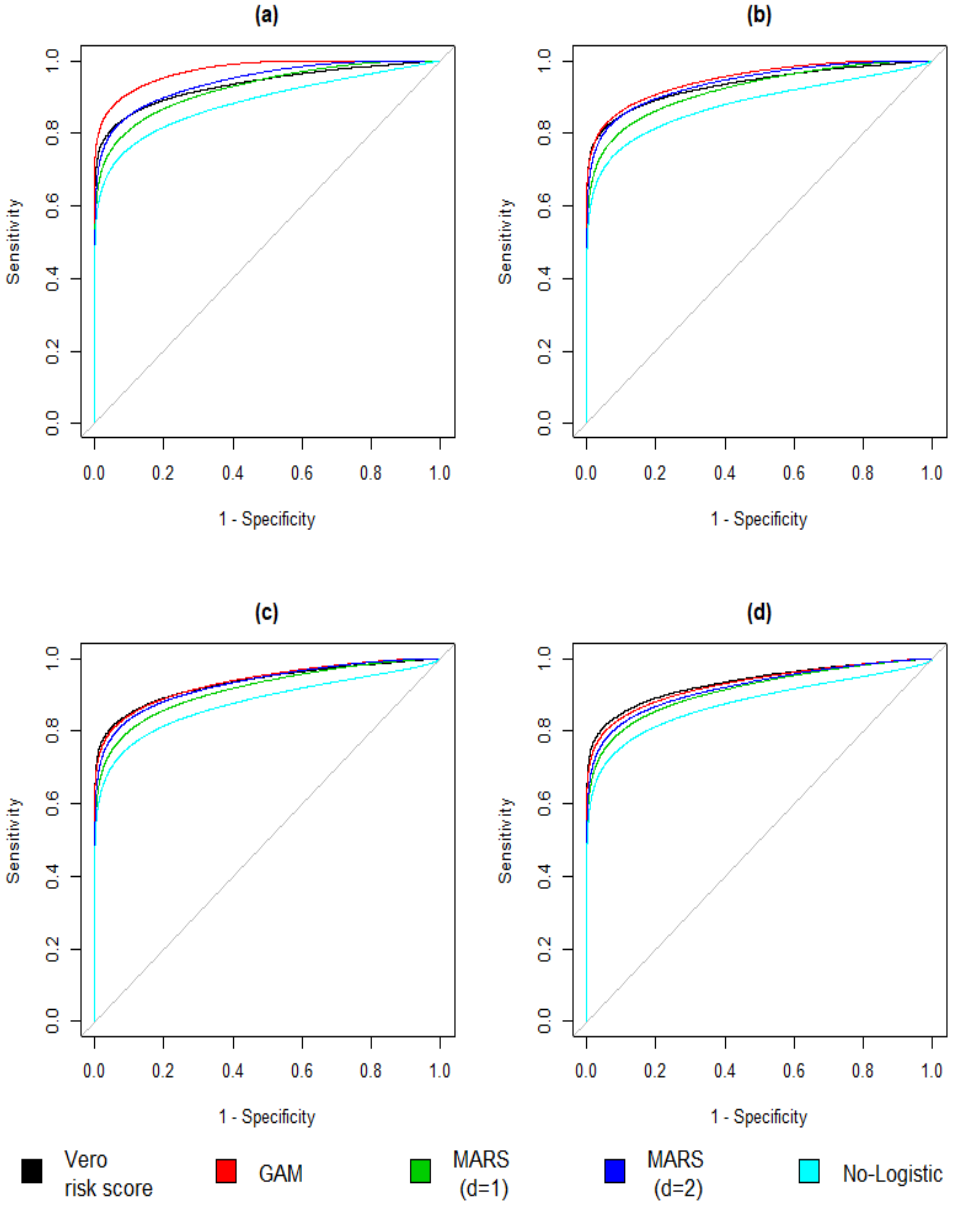


Figura 3.14: Simulazione 5, scenario 1. Confronto tra curve ROC stimate e curva vera per per (a) $(n_1, n_0) = (45, 35)$, (b) $(n_1, n_0) = (90, 70)$, (c) $(n_1, n_0) = (180, 140)$ e (d) $(n_1, n_0) = (360, 280)$. Nei dati simulati $T_2 = \frac{1}{T_2}$.

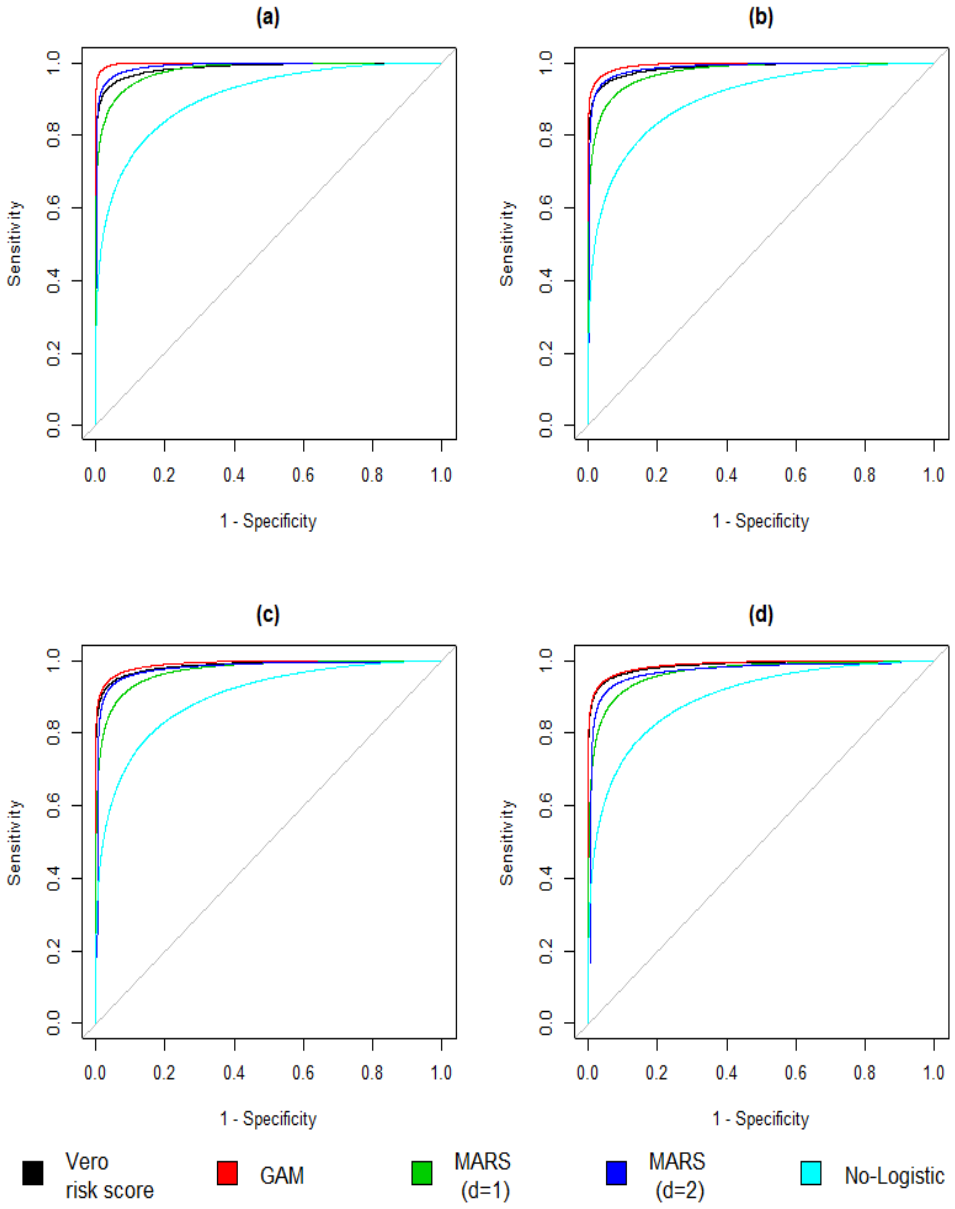


Figura 3.15: Simulazione 6, scenario 1. Confronto tra curve ROC stimate e curva vera per per (a) $(n_1, n_0) = (45, 35)$, (b) $(n_1, n_0) = (90, 70)$, (c) $(n_1, n_0) = (180, 140)$ e (d) $(n_1, n_0) = (360, 280)$. Nei dati simulati $T_2 = \frac{1}{T_2}$.

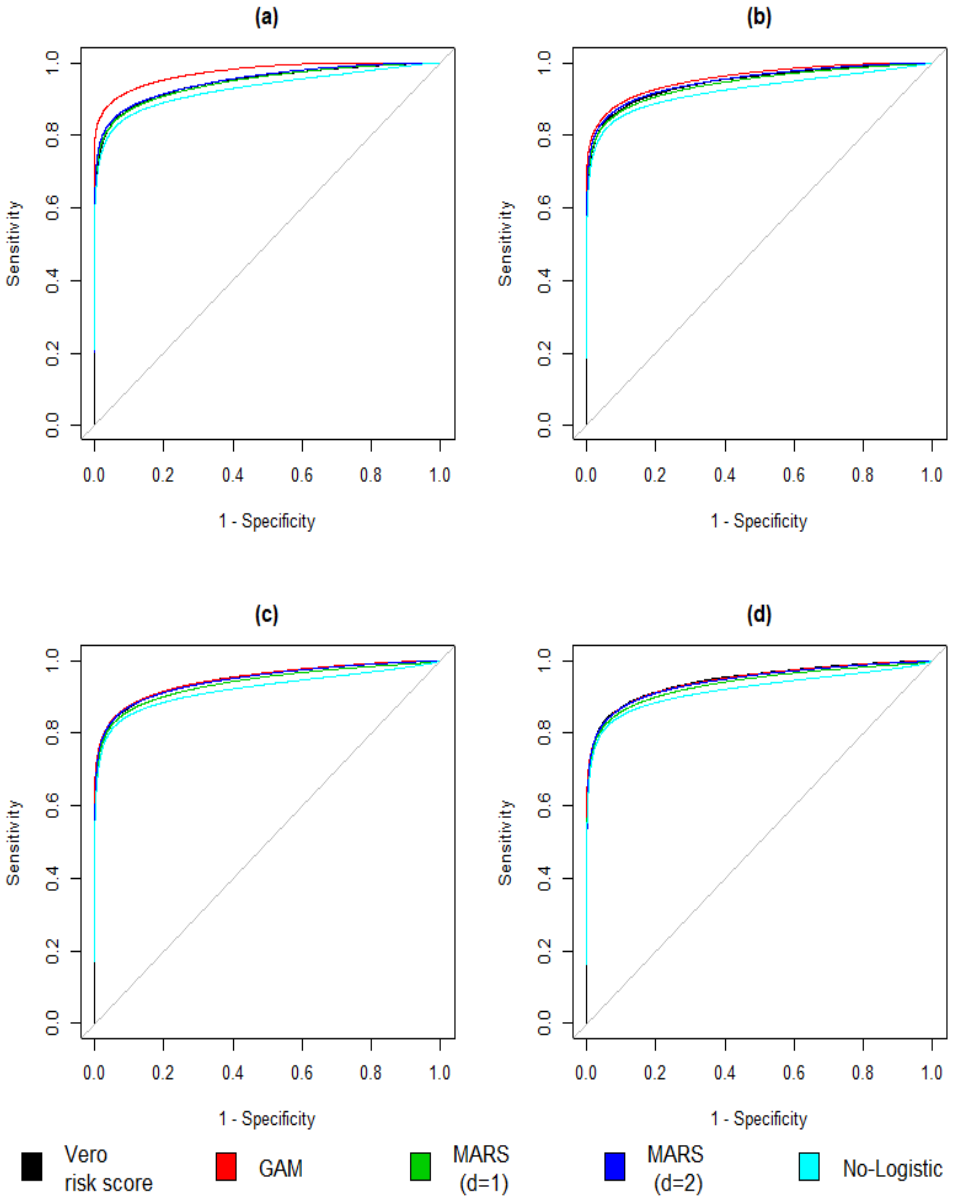


Figura 3.16: Simulazione 7, scenario 1. Confronto tra curve ROC stimate e curva vera per per (a) $(n_1, n_0) = (45, 35)$, (b) $(n_1, n_0) = (90, 70)$, (c) $(n_1, n_0) = (180, 140)$ e (d) $(n_1, n_0) = (360, 280)$. Nei dati simulati $T_2 = T_2^3$.

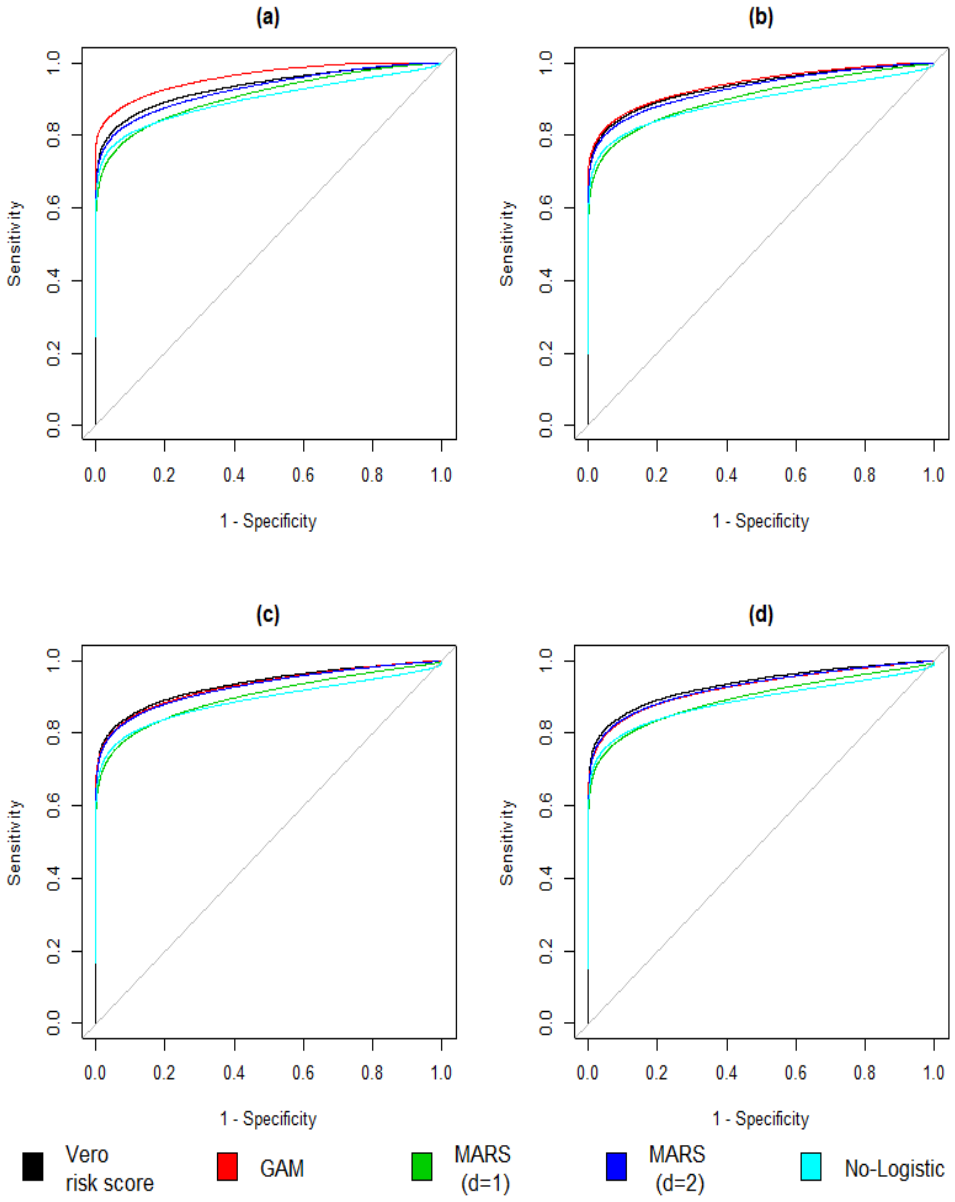


Figura 3.17: Simulazione 8, scenario 1. Confronto tra curve ROC stimate e curva vera per (a) $(n_1, n_0) = (45, 35)$, (b) $(n_1, n_0) = (90, 70)$, (c) $(n_1, n_0) = (180, 140)$ e (d) $(n_1, n_0) = (360, 280)$. Nei dati simulati $T_2 = T_2^3$.

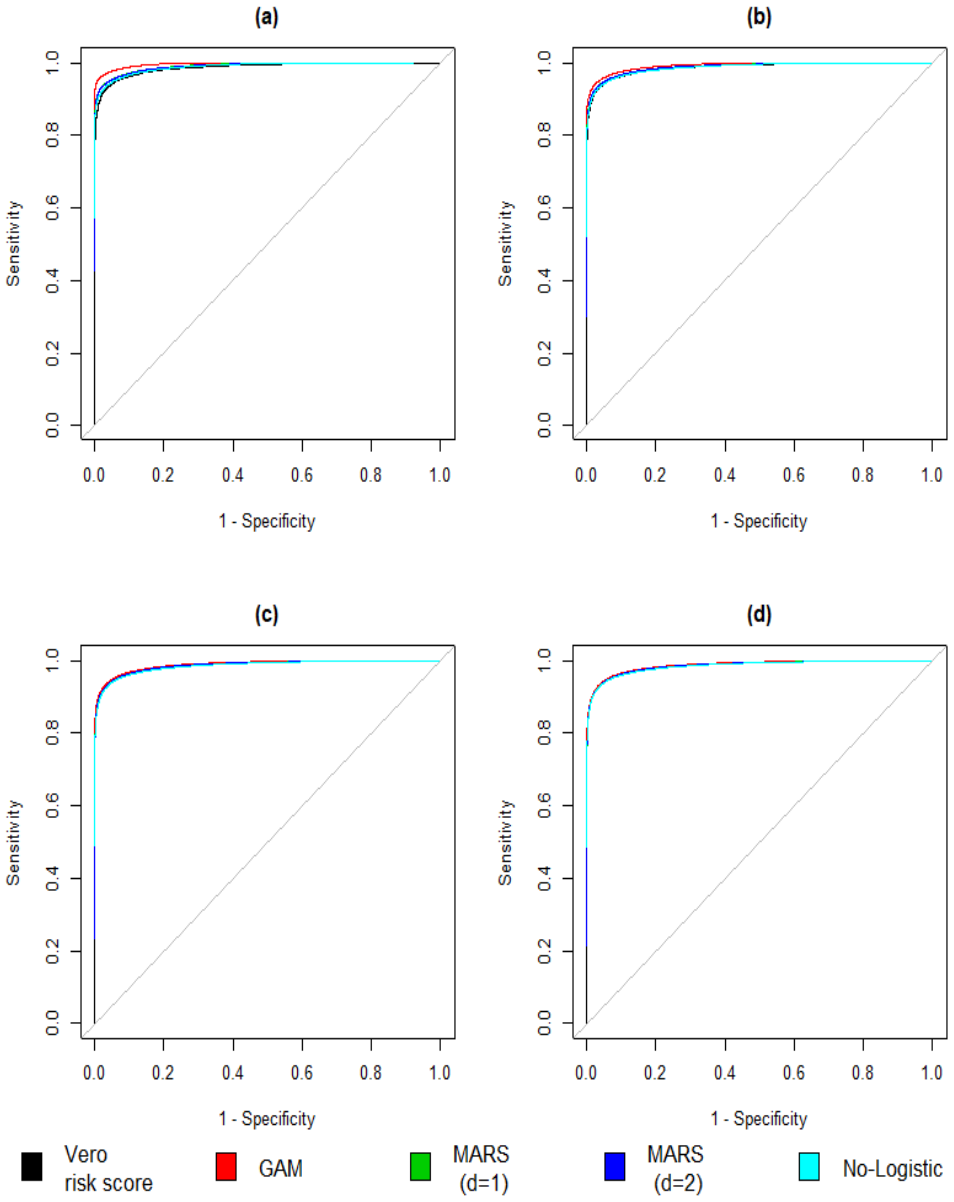


Figura 3.18: Simulazione 9, scenario 1. Confronto tra curve ROC stimate e curva vera per (a) $(n_1, n_0) = (45, 35)$, (b) $(n_1, n_0) = (90, 70)$, (c) $(n_1, n_0) = (180, 140)$ e (d) $(n_1, n_0) = (360, 280)$. Nei dati simulati $T_2 = T_2^3$.

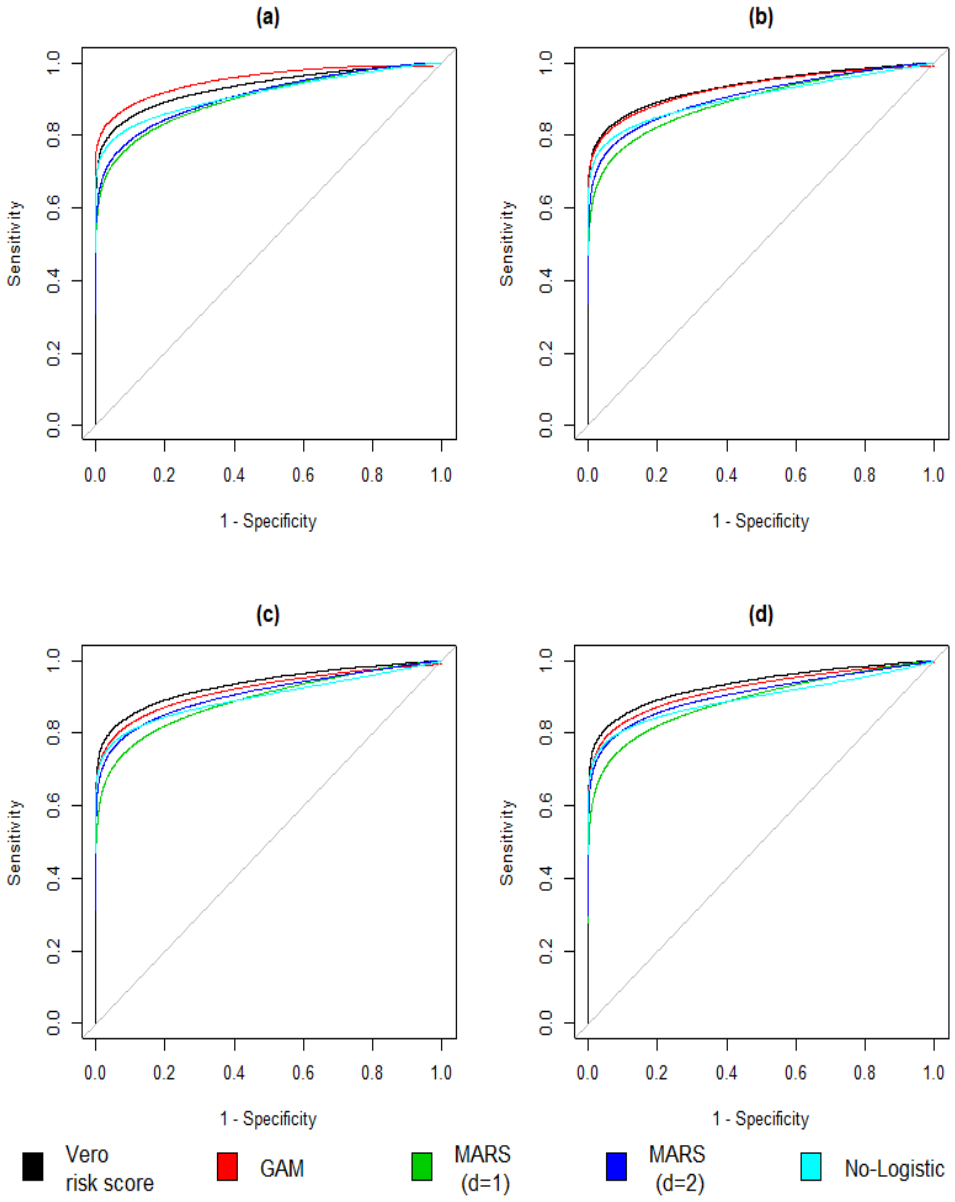


Figura 3.19: Simulazione 11, scenario 1. Confronto tra curve ROC stimate e curva vera per (a) $(n_1, n_0) = (45, 35)$, (b) $(n_1, n_0) = (90, 70)$, (c) $(n_1, n_0) = (180, 140)$ e (d) $(n_1, n_0) = (360, 280)$. Nei dati simulati $T_2 = \exp(T_2)$.

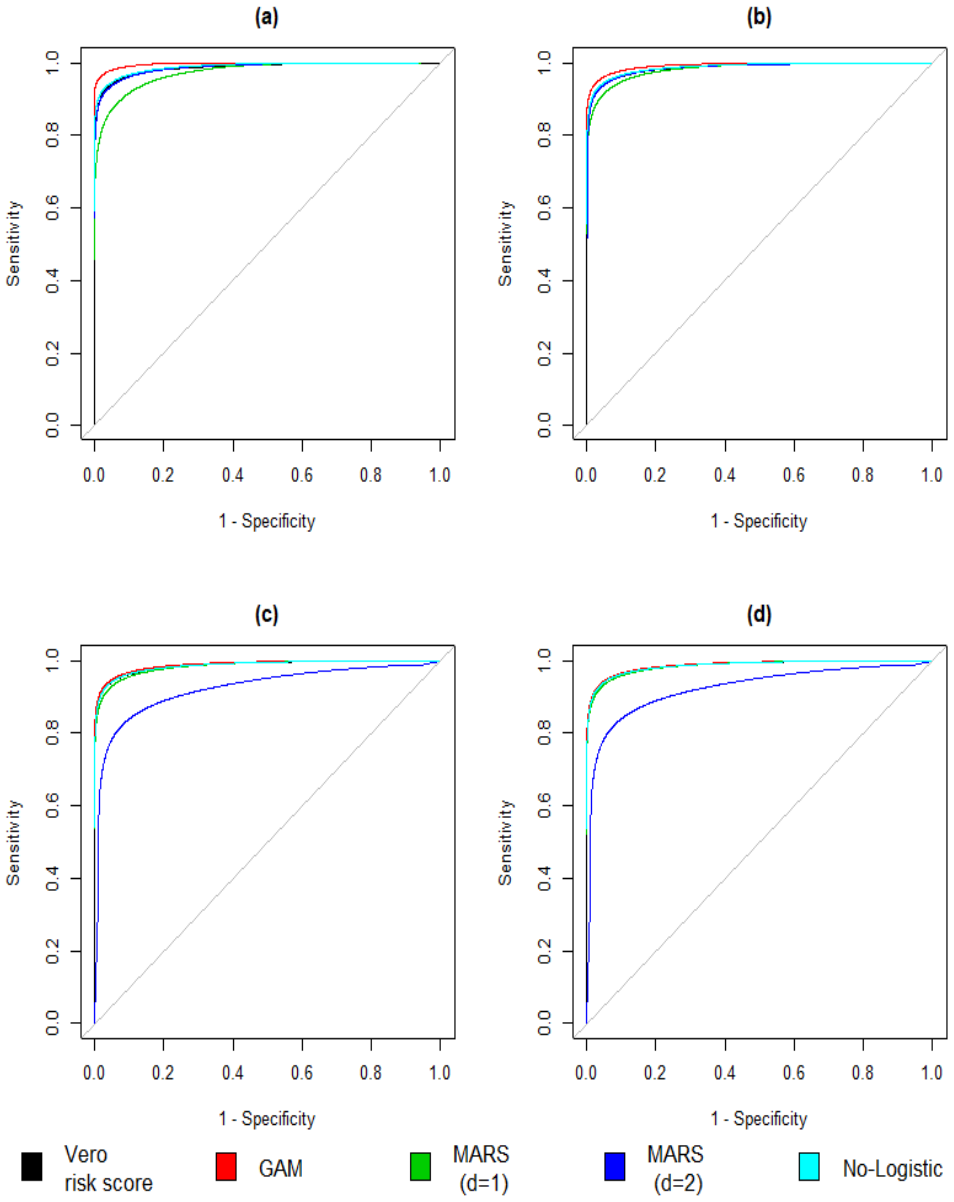


Figura 3.20: Simulazione 12, scenario 1. Confronto tra curve ROC stimate e curva vera per (a) $(n_1, n_0) = (45, 35)$, (b) $(n_1, n_0) = (90, 70)$, (c) $(n_1, n_0) = (180, 140)$ e (d) $(n_1, n_0) = (360, 280)$. Nei dati simulati $T_2 = \exp(T_2)$.

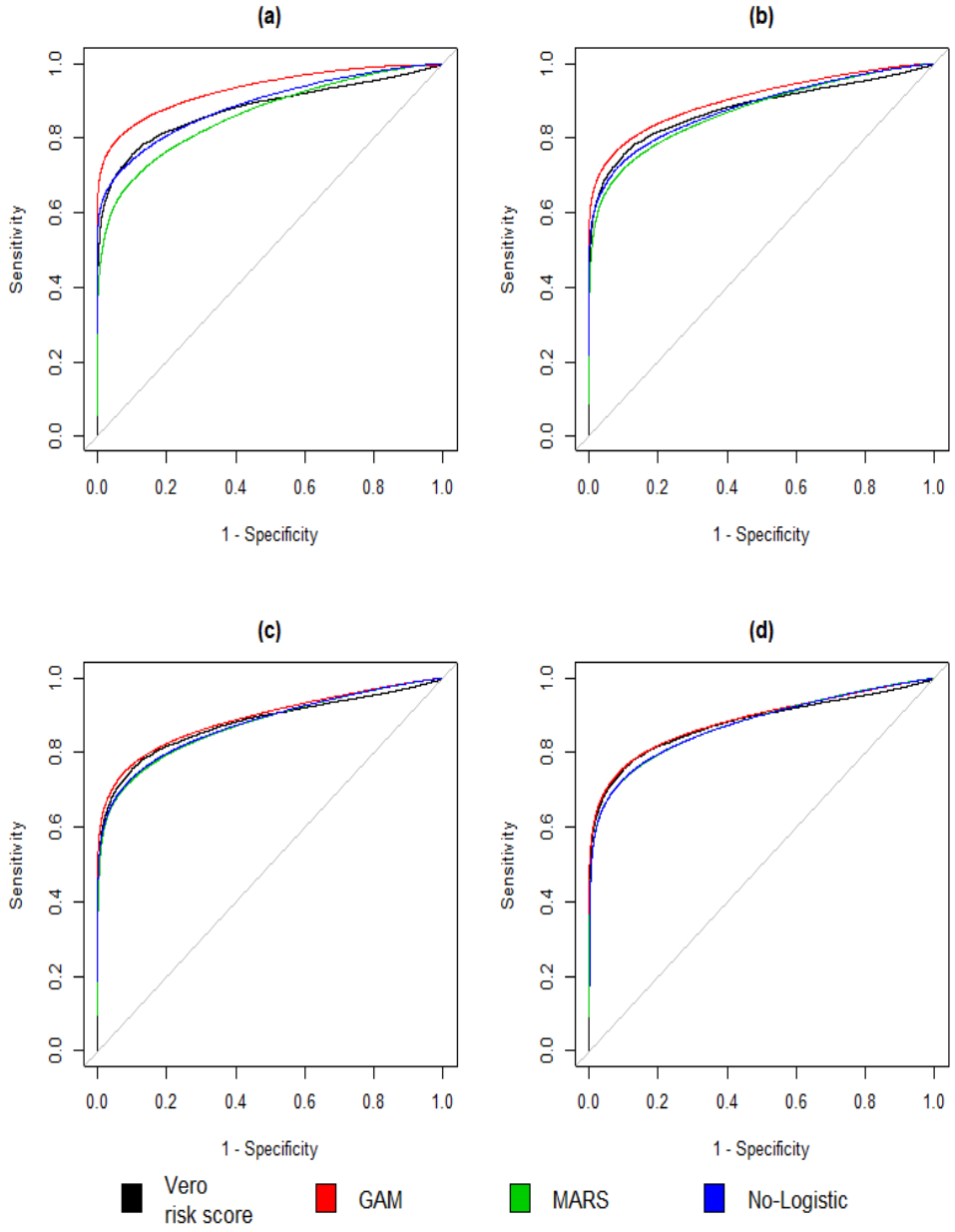


Figura 3.21: Simulazione 3, scenario 2. Confronto tra curve ROC stimate e curva vera per (a) $(n_1, n_0) = (45, 35)$, (b) $(n_1, n_0) = (90, 70)$, (c) $(n_1, n_0) = (180, 140)$ e (d) $(n_1, n_0) = (360, 280)$. Nei dati simulati $T_2 = T_2^3$.

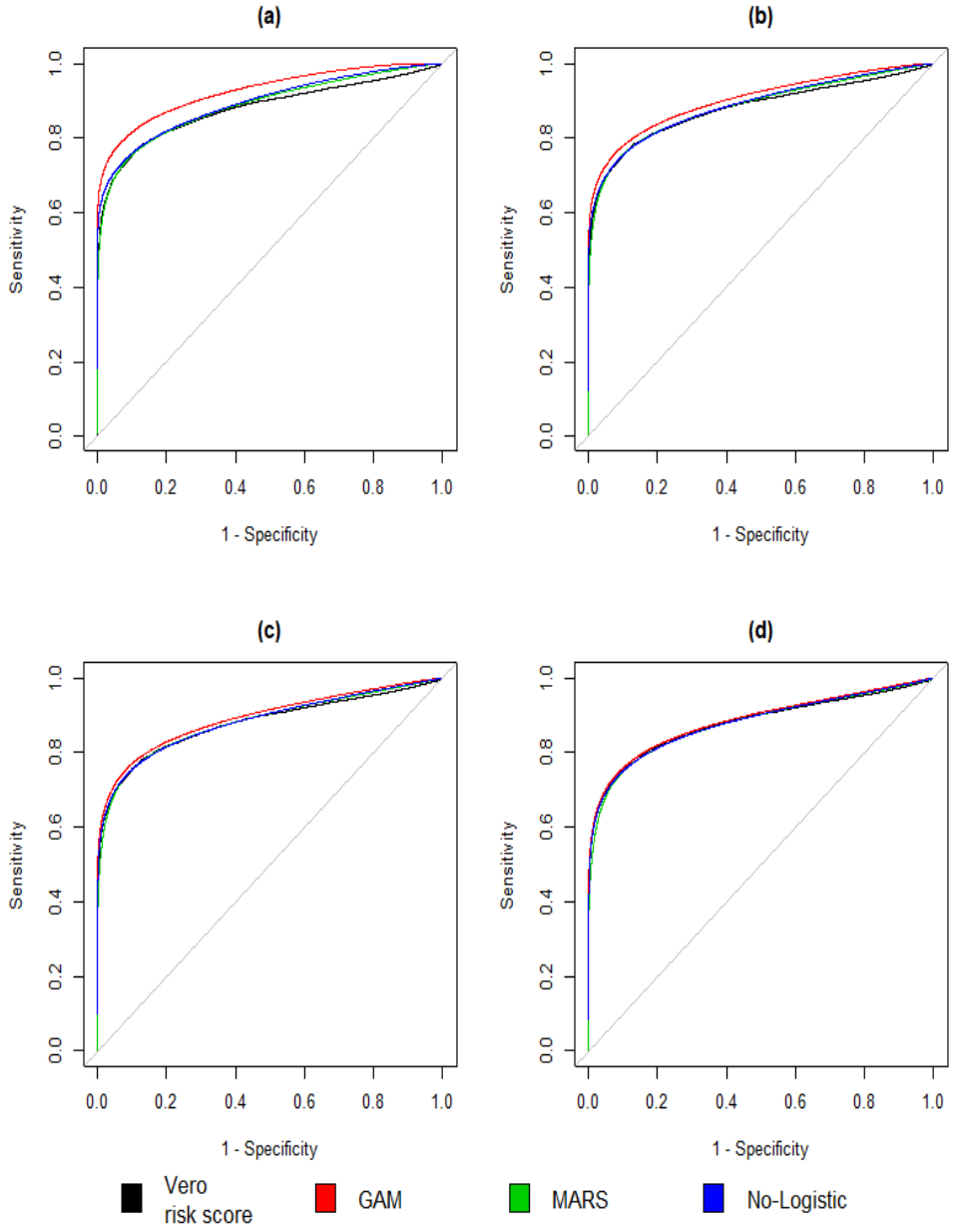


Figura 3.22: Simulazione 4, scenario 2. Confronto tra curve ROC stimate e curva vera per (a) $(n_1, n_0) = (45, 35)$, (b) $(n_1, n_0) = (90, 70)$, (c) $(n_1, n_0) = (180, 140)$ e (d) $(n_1, n_0) = (360, 280)$. Nei dati simulati $T_2 = \exp(T_2)$.

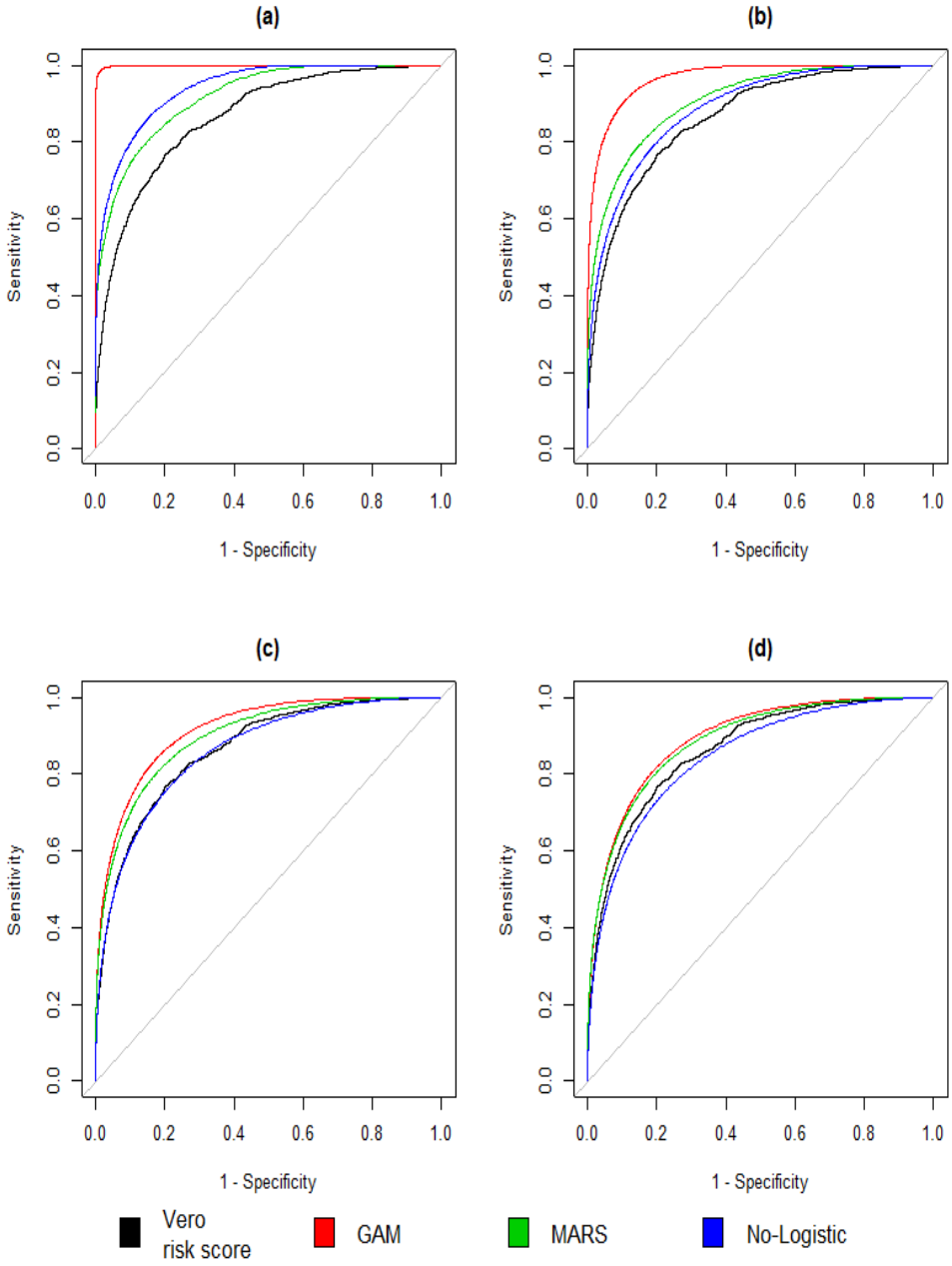


Figura 3.23: Simulazione 4, scenario 3. Confronto tra curve ROC stimate e curva vera per (a) $n = 80$, (b) $n = 160$, (c) $n = 320$ e (d) $n = 640$. Nei dati simulati $T_3 = \exp(T_3)$.

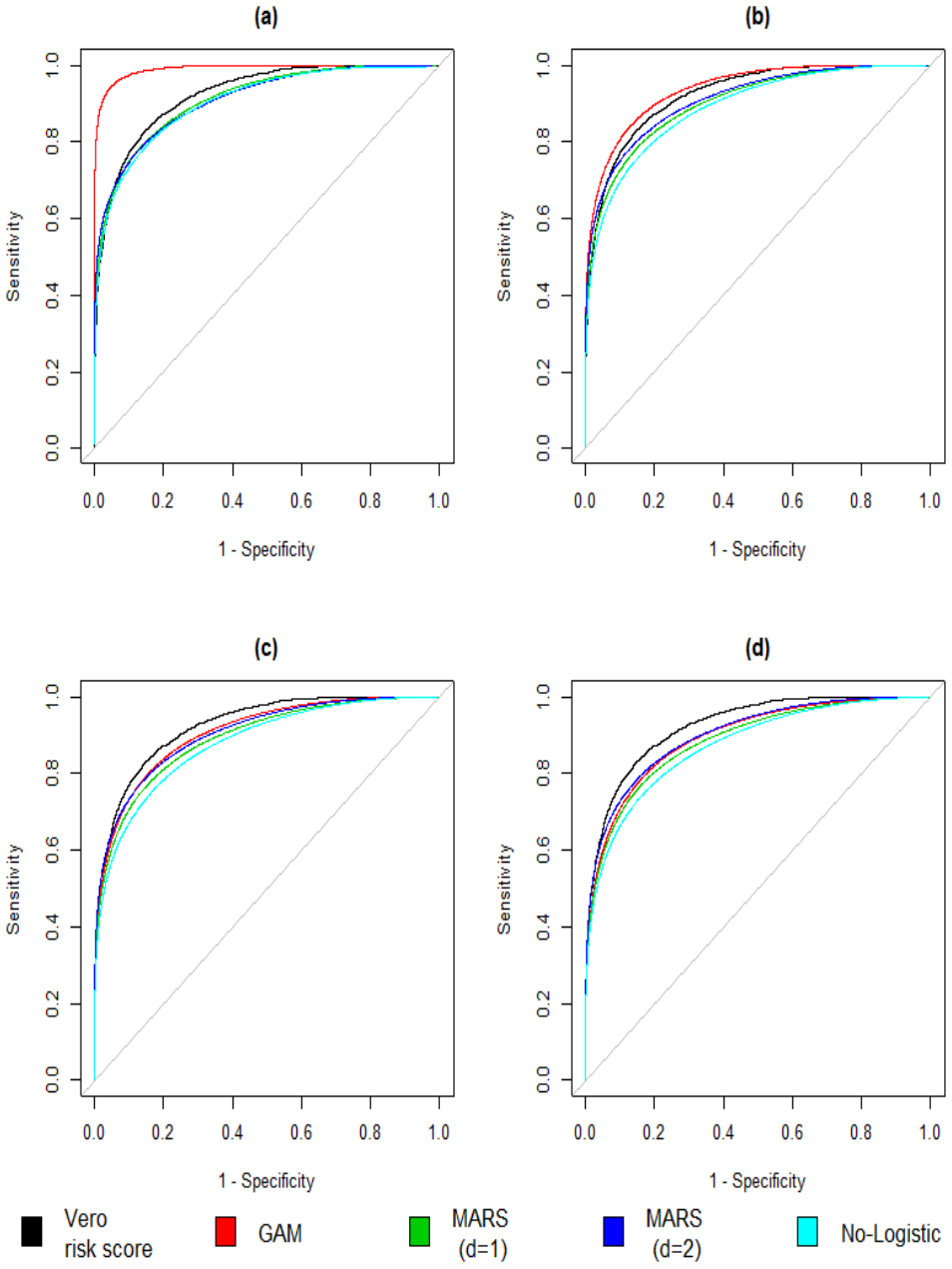


Figura 3.24: Simulazione 3, scenario 4. Confronto tra curve ROC stimate e curva vera per (a) $n = 80$, (b) $n = 160$, (c) $n = 320$ e (d) $n = 640$. Nei dati simulati $T_3 = T_3^3$.

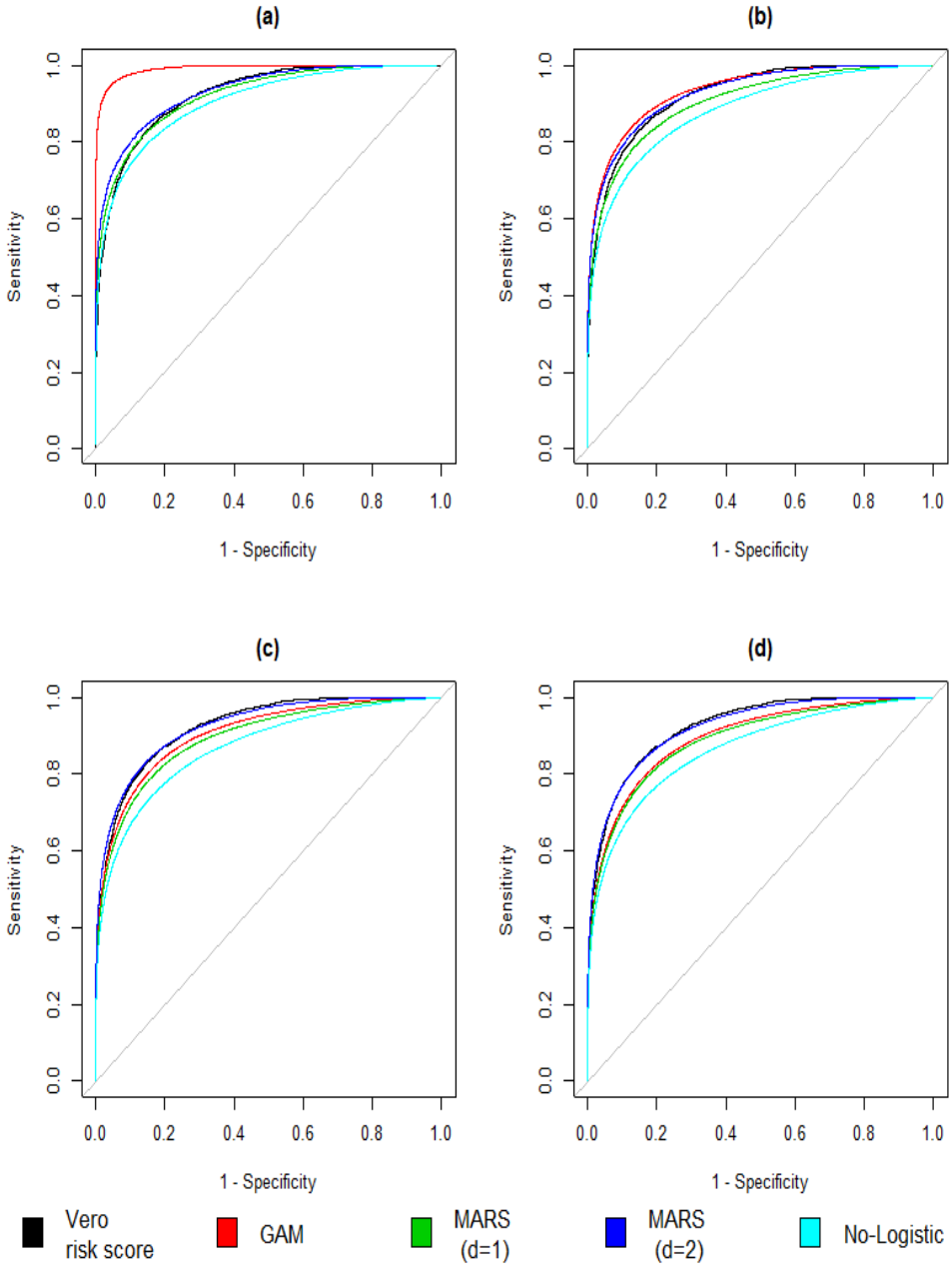


Figura 3.25: Simulazione 4, scenario 4. Confronto tra curve ROC stimate e curva vera per (a) $n = 80$, (b) $n = 160$, (c) $n = 320$ e (d) $n = 640$. Nei dati simulati $T_3 = \exp(T_3)$.

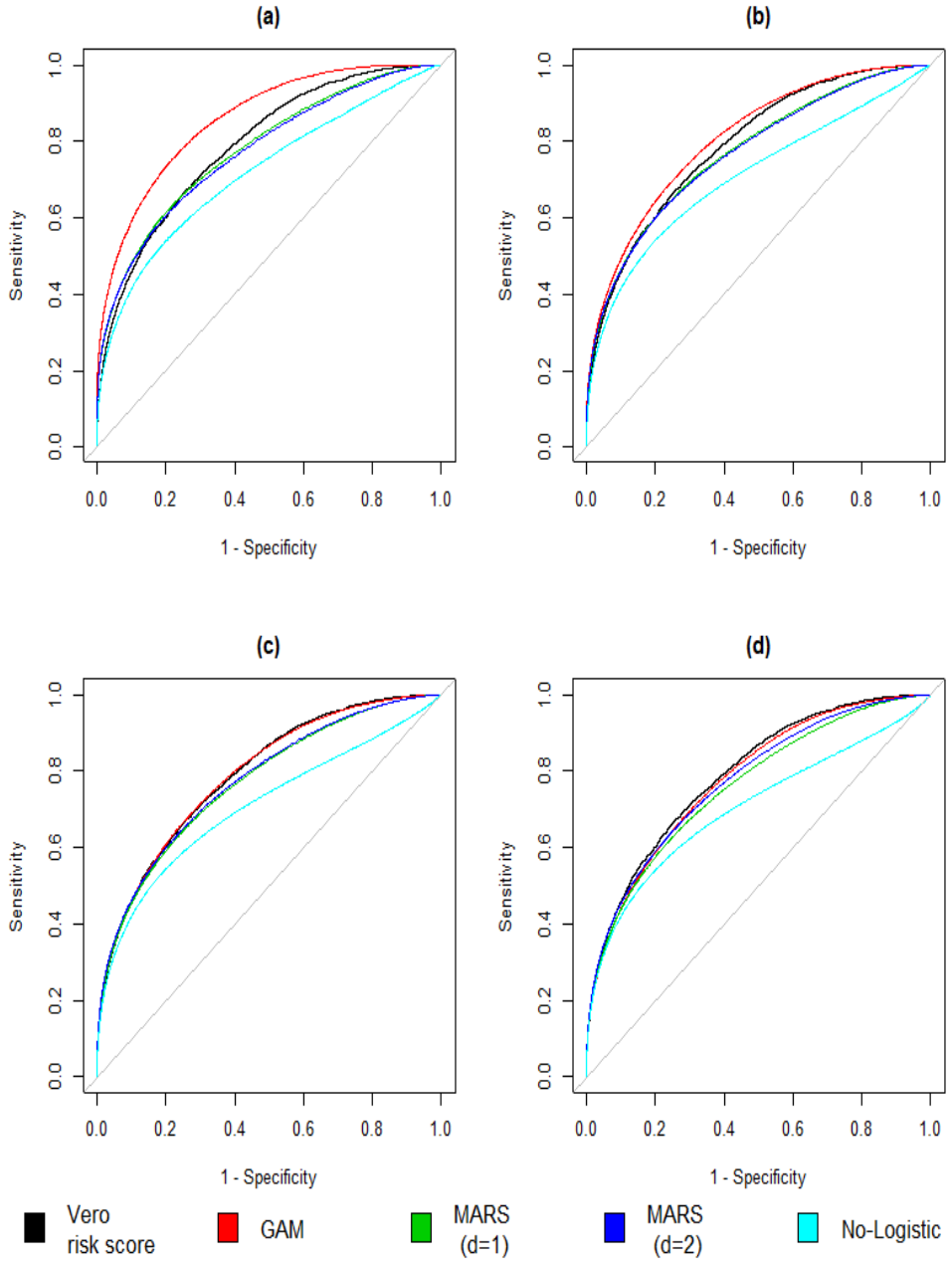


Figura 3.26: Simulazione 2, scenario 5. Confronto tra curve ROC stimate e curva vera per (a) $n = 80$, (b) $n = 160$, (c) $n = 320$ e (d) $n = 640$. Nei dati simulati $T_2 = \frac{1}{T_2}$.

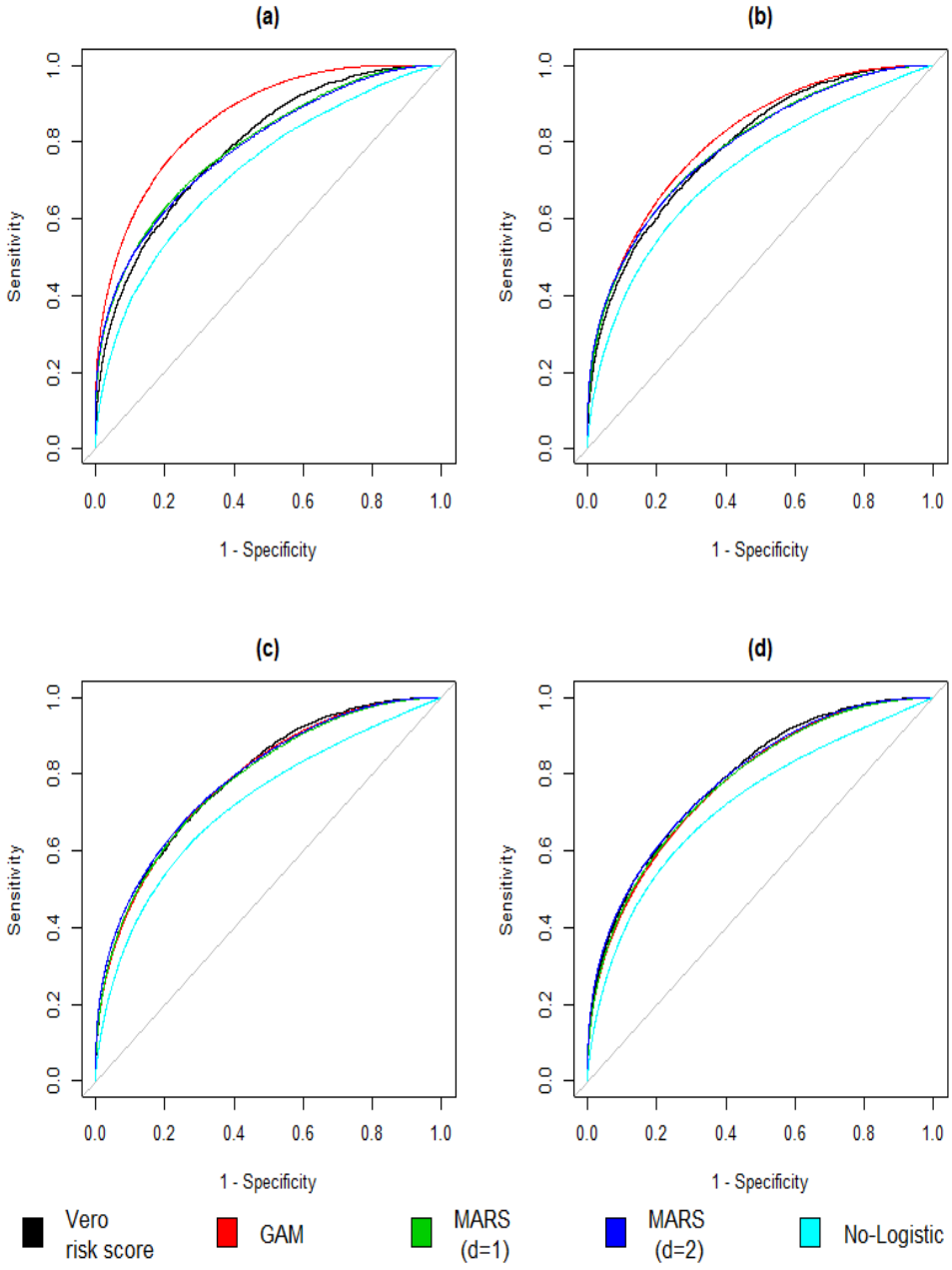


Figura 3.27: Simulazione 3, scenario 5. Confronto tra curve ROC stimate e curva vera per (a) $n = 80$, (b) $n = 160$, (c) $n = 320$ e (d) $n = 640$. Nei dati simulati $T_2 = T_2^3$.

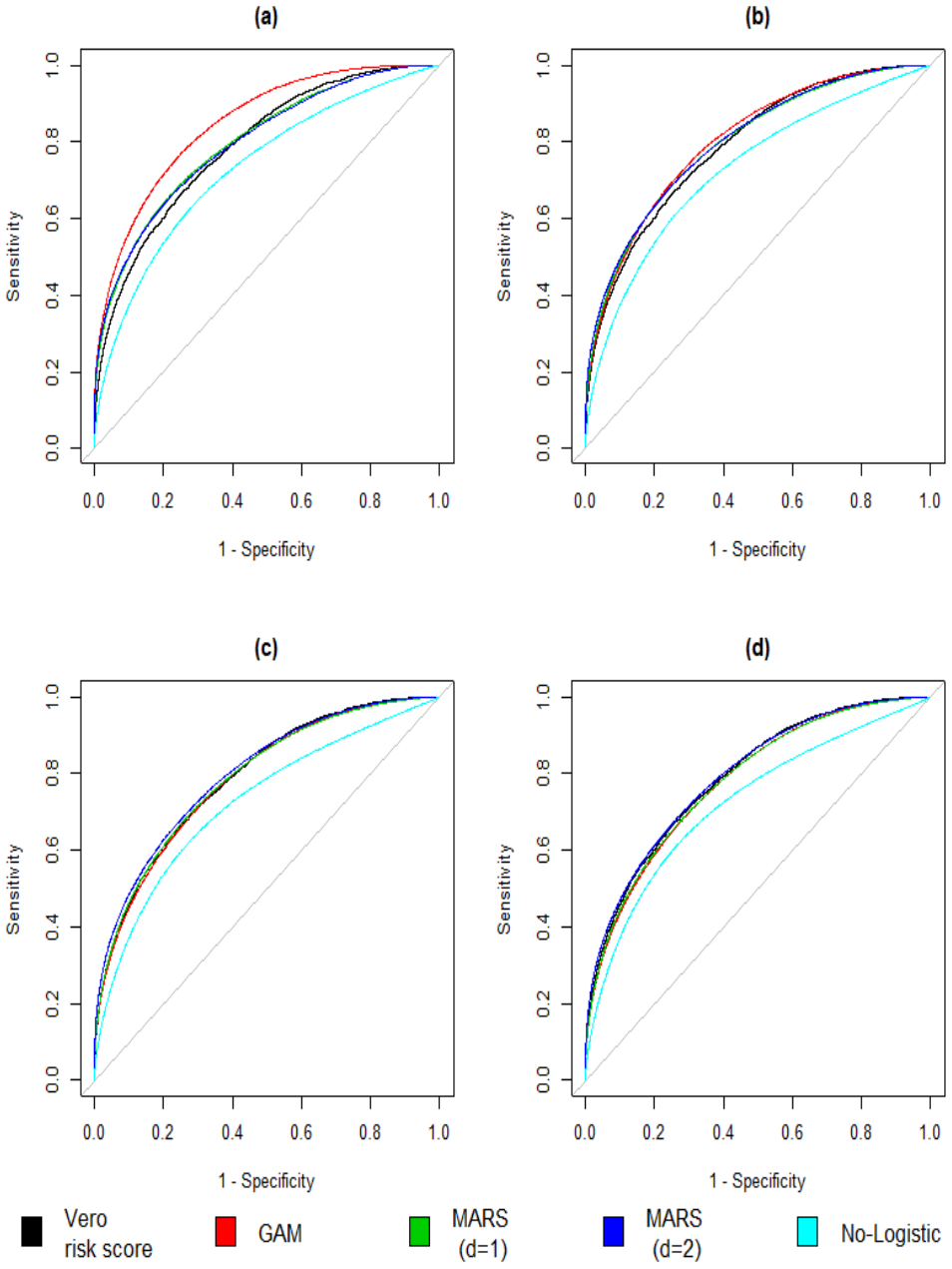


Figura 3.28: Simulazione 4, scenario 5. Confronto tra curve ROC stimate e curva vera per (a) $n = 80$, (b) $n = 160$, (c) $n = 320$ e (d) $n = 640$. Nei dati simulati $T_2 = \exp T_2$.

Capitolo 4

Applicazione a dati reali

In questo capitolo si analizzano due dataset frequentemente utilizzati nell'ambito del *machine learning* per la valutazione dei classificatori, ovvero i dataset *Pima Indians Diabetes*[1] e *Heart Disease Cleveland*[2] e dati relativi allo studio ADNI (*Alzheimer's Disease Neuroimaging Initiative*)[3]. Scopo delle analisi è quello di sperimentare gli approcci non parametrici per la stima dell'AUC della combinazione dei marcatori presenti negli studi e confrontarli con la stima ottenuta mediante l'utilizzo di più tradizionali strumenti parametrici.

Essendo in questa tesi interessati a valutare test su scala continua, negli studi considerati sono state utilizzate solamente le variabili che presentano tale natura. I modelli parametrici utilizzati per stimare il *risk score* sono due specificazioni differenti di un modello logistico, di cui il primo con solo effetti lineari (GLM(1)), e il secondo con effetti di interazione di primo ordine (GLM(2)). I modelli non parametrici, invece, sono il modello GAM e due modelli MARS di cui il primo senza effetti di interazione (MARS(1)), mentre l'altro con effetti di interazione fino al primo ordine (MARS(2)).

Alla stima dell'AUC è stato associato un intervallo di confidenza bootstrap di tipo percentile (ottenuto da 1000 campioni bootstrap) per un livello di confidenza pari a 0.95. Il ricampionamento effettuato è basato su una

[1] http://www.cit.ctu.edu.vn/~dtnghi/detai/PimaIndiansDiabetes.html

[2] http://sci2s.ugr.es/keel/dataset.php?cod=57

[3] http://www.adni.loni.usc.edu

stratificazione per status di malattia in maniera tale che la proporzione tra sani e malati rimanga invariata per tutti i campioni bootstrap estratti. Una stima basatta sulla convalida incrociata è stata ottenuta utilizzando il protocollo definito in Ma e Huang (2007) che stima i modelli su un insieme di stima (2/3 dei dati) e calcola l'AUC su un insieme di verifica (1/3 dei dati). Tale procedura è ripetuta 1000 volte: l'AUC finale, quindi, è la media dei 1000 AUC ottenuti, e la stima della sua variabilità ottenuta come deviazione standard delle 1000 stime.

Pima Indians Diabetes

Lo studio *Pima Indians Diabetes* (Smith et al. (1988)) ha come obiettivo quello di identificare le variabili che meglio identificano la condizione di diabete in 768 pazienti donna di almeno 21 anni degli Indiani Pima. Sono disponibili 7 variabili biomediche e l'età (usata in questa analisi come covariata per migliorare la precisione delle stime dei modelli). Come in Ma e Huang (2007), le variabili sono state standardizzati.

In Tabella 4.1 si riportano gli AUC calcolabili dai modelli utilizzati per stimare il *risk score*. Si nota immediatamente che l'utilizzo della convalida incrociata come descritto in Ma e Huang (2007) è essenziale per ottenere stime non inflazionate dell'AUC (cfr. le stime dell'AUC per modelli non parametrici con e senza convalida incrociata). Inoltre, è stato osservato che l'utilizzo di 10 nodi per ogni *spline* nei test nel modello GAM porta a un sovra-adattamento ai dati, come dimostrato dalla non appratenenza dell'AUC ottenuto (0.8744) all'interno dell'intervallo di confidenza bootstrap costruito (0.8804, 0.9300). Per questo motivo i risultati del GAM riportati in Tabella 4.1 sono ottenuti ponendo 5 nodi per ogni *spline*. Il confronto tra risultati ottenuti utilizzando l'approccio parametrico e l'approccio non parametrico non fa emergere discrepanze evidenti tra le due strategie.

Hearth Disease Cleveland

Lo studio su *Hearth Disease Cleveland*, attraverso diverse variabili biomediche (tra cui anche l'età utilizzata con lo stesso fine descritto nell'analisi precedente), ha come obiettivo quello identificare la presenza di malattia

	(1)	(2)
GLM(1)	0.8394	0.8298
	(0.8121, 0.8709)	(0.0222)
GLM(2)	0.8638	0.8210
	(0.8514, 0.9024)	(0.0224)
GAM	0.8678	0.8400
	(0.8556, 0.9062)	(0.0219)
MARS(1)	0.8659	0.8317
	(0.8557, 0.9049)	(0.0225)
MARS(2)	0.8712	0.8183
	(0.8644, 0.9131)	(0.0255)

Tabella 4.1: AUC stimati sui dati Pima in due modi differenti: (1) senza convalida incrociata (con intervalli bootstrap per un livello di confidenza 0.95) e (2) con convalida incrociata (deviazione standard delle stime in 1000 repliche)

cardiaca nei pazienti. Qui si utilizza un sottoinsieme dell'intero set di dati costituito da 297 osservazioni. La variabile risposta indica la presenza di malattia cardiaca ed è composta da più modalità che corrispondono a specifici status della malattia individuati per mezzo dell'angiografia: il valore 0 indica l'assenza di malattia, mentre i valori $\{1, 2, 3, 4\}$, invece, indicano la presenza della malattia. Per gli scopi dell'analisi da svolgere, tale variabile di risposta è stata riclassificata indicando con 0 l'assenza di malattia e con 1 il contrario. Nell'insieme di dati risultante, 160 pazienti non hanno la malattia e 137 la presentano.

In Tabella 4.2 si riportano gli AUC calcolabili dai vari modelli utilizzati per stimare il *risk score*. L'analisi condotta ha portato ad escludere il modello MARS(2) per via del sovra-adattamento ai dati: infatti l'AUC stimato (0.8269) non rientra all'interno dell'intervallo bootstrap costruito (0.8348, 0.9312). La stessa cosa è stata osservata per un GAM con 10 nodi per ogni *spline* (AUC pari a 0.8358 e un intervallo bootstrap (0.8514, 0.9505)) e per questo motivo, come già fatto per il set di dati precedente, i risultati del GAM riportati in Tabella 4.2 sono ottenuti ponendo 5 nodi per *spline*. An-

che qui, analizzando gli AUC ottenuti nei due metodi, è possibile fare le stesse considerazioni fatte utilizzando il dataset *Pima Indians Diabetes*.

	(1)	**(2)**
GLM(1)	0.8007	0.7846
	(0.7548, 0.8542)	(0.0377)
GLM(2)	0.8212	0.7711
	(0.7972, 0.8872)	(0.0387)
GAM	0.8170	0.7791
	(0.7959, 0.8930)	(0.0413)
MARS(1)	0.8210	0.7716
	(0.8154, 0.9124)	(0.0431)

Tabella 4.2: AUC stimati sui dati Cleveland in due modi differenti: (1) senza convalida incrociata (con intervalli bootstrap per un livello di confidenza 0.95) e (2) con convalida incrociata (deviazione standard delle stime in 1000 repliche)

Dati ADNI

Lo studio ADNI (*Alzheimer's Disease Neuroimaging Initiative*) pone come obiettivo, tra gli altri, quello di diagnosticare mediante l'uso di vari *marker* se un paziente presenta l'AD (*Alzheimer's Disease*) oppure l'MCI (*Mild Cognitive Impairment*), condizione associata a problemi con la memoria ma non tali da compromettere le funzioni quotidiane come può accadere per la demenza. L'AD è la forma più comune di demenza ed è una malattia progressiva, ovvero che i sintomi associati a tale patologia peggiorano nel tempo[4]. I *marker* utilizzati in questa applicazione sono rispettivamente il $t\tau$ (*Total Tau*), il $p\tau_{181p}$ (*Tau Phosphorylated at the Threonine 181*) e $A\beta_{1-42}$ (*Amolyd-β 1-42 peptide*). Si dispone di 846 osservazioni di cui 618 MCI e 228 AD.

L'analisi su tali dati mette a confronto gli AUC stimati per diverse specificazioni del regressore. La prima specificazione, infatti, considera soltanto

[4] https://www.alz.org/nca/in_my_community_22013.asp

la variabile T ottenuta come rapporto tra $t\tau$ e $A\beta_{1-42}$. La seconda specificazione, invece, è costituita dalla somma dei due *marker* che compongono T; infine, la terza specificazione non è altro che la seconda, alla quale, però, è stato aggiunto anche $p\tau$.

In Tabella 4.3 si riportano i risultati dei modelli utilizzati per stimare il *risk score* in termini di AUC

	T		$\tau + \beta_{1-42}$		$\tau + p\tau + \beta_{1-42}$	
	(1)	**(2)**	**(1)**	**(2)**	**(1)**	**(2)**
GLM(1)	0.7504 (0.7149, 0.7843)	0.7498 (0.0244)	0.7411 (0.7052, 0.7785)	0.7374 (0.0263)	0.7482 (0.7168, 0.7861)	0.7431 (0.0251)
GLM(2)			0.7460 (0.7136, 0.7850)	0.7411 (0.0259)	0.7631 (0.7338, 0.7996)	0.7520 (0.0250)
GAM	0.7503 (0.7157, 0.7848)	0.7482 (0.0244)	0.7529 (0.7254, 0.7925)	0.7432 (0.0256)	0.7628 (0.7360, 0.8046)	0.7507 (0.0250)
MARS(1)	0.7504 (0.7178, 0.7946)	0.7436 (0.0244)	0.7520 (0.7511, 0.8190)	0.7334 (0.0268)	0.7632 (0.7582, 0.8257)	0.7318 (0.0280)
MARS(2)			0.7694 (0.7505, 0.8281)	0.7347 (0.0273)		

Tabella 4.3: AUC stimati sui dati dello studio ADNI in due modi differenti: (1) senza convalida incrociata (con intervalli bootstrap per un livello di confidenza 0.95) e (2) con convalida incrociata (deviazione standard delle stima in 1000 repliche)

I risultati relativi al modello GAM in Tabella 4.3 sono ottenuti utilizzando 5 nodi per *spline* poichè è stato osservato che l'AUC stimato con la convalida incrociata era molto simile a quello ottenuto con un modello GAM con 10 nodi per *spline* (e altrettanto è stato osservato anche confrontando gli AUC stimati senza convalida incrociata). Osservando gli AUC stimati tramite convalida incrociata per ciascuna specificazione del regressore, è possibile osservare che essi sono simili tra loro qualunque sia il modello considerato. Come per l'analisi dei dati Cleveland, è stato osservato anche qui un problema di sovra-adattamento in corrispondenza del modello MARS(2) con la terza specificazione: l'AUC stimato (0.7620), infatti, non rientra all'interno dell'intervallo bootstrap costruito (0.7683, 0.8351). Confrontando, invece, le stime dell'AUC per diverse specificazioni, è possibile osservare che gli AUC

calcolabili dal GAM nella seconda e nella terza specificazione non sono tanto differenti dall'AUC nella prima specificazione: ciò significa che il GAM coglie la trasformazione T anche se si utilizzano i due *marker* che definiscono T marginalmente (ed anche quando si aggiunge anche $p\tau$). Questo, però non lo si nota per il modello MARS(1).

Capitolo 5

Considerazioni finali

La ricerca della combinazione ottima (secondo il criterio definito nel Capitolo 1) di marcatori attraverso la modellazione del *risk score* è di notevole utilità poichè è di semplice interpretazione e gode anche delle stesse proprietà di ottimalità del rapporto di verosimiglianza (strumento che, invece, è più complesso da interpretare da un punto di vista medico) poichè è una sua trasformazione monotona crescente.

L'analisi dei dati reali ha mostrato che i modelli non parametrici proposti per stimare il *risk score* presentano un comportamento molto simile ai modelli parametrici, lasciando intendere che nelle tre situazioni considerate, la combinazione ottima è verosimilmente una combinazione lineare. In questi casi, quindi, la maggiore flessibilità dello strumento non parametrico non esprime il suo potenziale. Uno dei vantaggi è la possibilità di cogliere eventuali effetti non lineari dei tests. Oltre a ciò, nei modelli non parametrici è possibile anche definire agevolmente effetti di interazione complicati: questo è un aspetto caratteristico del modello MARS che coglie in automatico quelle interazioni (intese come prodotti di basi di funzioni) che hanno un contributo consistente nei confronti della previsione del modello. Tale flessibilitàà dell'interazione può essere raggiunta anche dal modello GAM se si inseriscono *spline* multivariate o, più in generale, lisciatori multivariati.

Sebbene la modellazione non parametrica offra questi potenziali vantaggi, è necessario sottolineare un aspetto negativo, evidente sia nelle simulazioni che nell'analisi dei dati reali: infatti, un elevato grado di flessibilità porta a un

sovra-adattamento ai dati. Nelle simulazioni, tale aspetto è stato evidenziato solo in corrispondenza della modellazione GAM dove il sovra-adattamento è essenzialmente legato al numero di nodi per ciascuna *spline*, che impatta fortemente sui risultati. Dall'analisi dei dati reali è stato possibile riscontrare che un elevato grado di flessibilità (10 nodi per *spline*) ha portato a stimare intervalli bootstrap per l'AUC non contenenti la stima. L'analisi di tali dati ha anche evidenziato tale problema anche per i modelli MARS con interazioni fino al primo ordine. Mentre per il GAM è stato possibile controllare il grado di sovra-adattamento riducendo il numero di nodi per *spline* a 5, il modello MARS, invece, è più difficile da gestire.

Riguardo gli intervalli bootstrap, sarebbe di rilievo verificare la copertura di tali intervalli con opportuni studi di simulazione.

In questa tesi, ci si è posti nel caso in cui per ciascun paziente si ha un *gold standard*, ma futuri sviluppi su tale argomento potrebbero incentrarsi sul verificare l'adeguatezza dei modelli non parametrici proposti ponendosi nel caso in cui tale *gold standard* non fosse disponibile per tutti i pazienti, problema noto come *verification bias*. Ciò accade quando gli esami diagnostici di riferimento non possono essere effettuati per differenti motivi: essi possono essere di natura economica (perchè troppo costosi), oppure perchè questi esami possono essere effettuati solo dopo l'insorgenza della patologia o dopo il decesso del paziente (rivelandosi poco informativi) o ancora perchè sarebbero delle procedure troppo invasive.

Bibliografia

Bamber, D. (1975). «The area above the ordinal dominance graph and the area below the receiver operating characteristic graph.» In: *J Mathematical Psychology* 12, pp. 387–415.

Chen, B. et al. (2016). «Using a monotonic density ratio model to find the asimptotically optimal combination of multiple diagnostic tests». In: *Jounal of american statistical association* 111(514), pp. 861–874.

Friedman, J.H. (1991). «Multivariate Adaptive Regression Splines». In: *Annals of statistics* 19(1), pp. 1–141.

Green, D. M. e J. A. Swets (1966). *Signal detection theory and psychophysics*. Los Altos, California USA: Peninsula Publishing.

Hansen, B.E. (2004). «Nonparametric estimation of smooth conditional distributions». In: *University of Winsconsin*.

Hastie, T.J., R.J. Tibshirani e J.H. Friedman (2009). *The elements of statistical learning: Second edition*.

Kang, L., A. Liu e L. Tian (2016). «Linear combination methods to improve diagnostic/prognostic accuracy on future observations.» In: *Statistical methods in medical research* 25(4), pp. 1359–1380.

Leathwick, J.R. et al. (2005). «Using multivariate adaptive regression splines to predict the distributions of New Zealand's freshwater diadromous fish». In: *Freshwater biolgy* 50, pp. 2034–2052.

Liu, C. e S. Halabi (2011). «A min-max combination of biomarkers to improve diagnostic accuracy.» In: *Statistics in medicine* 30(16), pp. 2005–2014.

Ma, S. e J. Huang (2007). «Combining multiple markers for classification using ROC». In: *Biometrics* 63, pp. 751–757.

McIntosh, M. W. e M. S. Pepe (2002). «Combining several screening tests: optimality of the risk score». In: *Biometrics* 58, pp. 657–664.

Milborrow, S. (2017). «`earth`: Multivariate Adaptive Regression Splines». In: R package. URL: `\url{https://CRAN.R-project.org/package=earth}`.

Morlini, I. (2006). «On multicollinearity and concurvity in some nonlinear multivariate models». In: *Statistical methods and application* 15, pp. 3–26.

Pepe, M. S (2003). *The statistical evaluation of medical tests for classification and prediction.* Oxford University Press.

Pepe, M. S. e M. L. Thompson (2000). «Combining diagnostic test results to increase accuracy.» In: *Biostatistics* 1(2), pp. 123–140.

Robin, X. et al. (2011). «pROC: an open-source package for R and S+ to analyze and compare ROC curves». In: *Bioinformatics* 12.

Smith, J.W. et al. (1988). «Using the ADAP learning algorithm to forecast the onset of diabetes mellitus». In: *Proceedings of the symposium on computer applications and medical care*, pp. 261–265.

Su, J. Q. e J. S. Liu (1993). «Linear combinations of multiple diagnostic markers.» In: *Journal of the American Statistical Association* 88(424), pp. 1350–1355.

Wood, S. N. (2006). *Generalized Additive Models: An introduction with R.* Chapman Hall.

— (2011). «Fast stable restricted maximum likelihood and marginal likelihood estimation of semiparametric generalized linear models». In: *Journal of the Royal Statistical Society (B)* 73(1), pp. 3–36.

Appendice A

Codice R utilizzato

Codice A.1: Funzione che calcola la curva ROC nonchè piccola modifica di alcune funzioni presenti nel pacchetto `pROC` (Robin et al. 2011)

```
##Funzione che calcola sensitivita' e specificita'
roc.utils.perfs <- function(threshold, controls, cases, direction) {
  if (direction == '>') {
    tp <- sum(cases <= threshold) #Veri positivi
    tn <- sum(controls > threshold) #Veri negativi
  }
  else if (direction == '<') {
    tp <- sum(cases >= threshold);tn <- sum(controls < threshold)
  }
  return(c(sp=tn/length(controls), se=tp/length(cases)))
}
roc.utils.perfs.all.safe <- function(thresholds, controls, cases,
    direction) {
  #Per ogni valore soglia calcolo i veri positivi e negativi
  perf.matrix <- sapply(thresholds, roc.utils.perfs, controls=controls,
      cases=cases, direction=direction)
  return(list(se=perf.matrix[2,], sp=perf.matrix[1,]))
}
##Funzione che calcola la curva ROC
myROC<-function(response, predicted, direction='<', plot=FALSE)
```

```
{
  #Divido i valori predetti dal modello sulla base delle classi della
      risposta
  splitted<- split(predicted, response)
  #Livelli della risposta
  levels=base::levels(as.factor(response))
  #Definisco i controlli (response=0) e i casi (response=1)
  controls <- splitted[[as.character(levels[1])]]
  cases <- splitted[[as.character(levels[2])]]
  #Definisco i valori soglia che voglio utilizzare
  threshold<- c(-Inf, seq(10^-8, 1-10^-8, length=998), Inf)
  #Calcolo della sensibilita' e della specificita'
  sesp<- roc.utils.perfs.all.safe(threshold, controls, cases, direction)
  if(plot==TRUE)
  {
    plot(1-sesp$sp,sesp$se,type='l',xlab='1-Specificity',ylab='Sensibility
        ',main='ROC')
  }
  invisible(list(Sensitivities=sesp$se, Specificities=sesp$sp))
}
```

Codice A.2: Funzione che simula il primo scenario in cui si hanno due tests che provengono da una distribuzione normale bivariata

```
sim1<-function(Nsim, m1, m0, s1, s0, n1, n0, type=1)
{
 ###Info su parametri
 # 1. Nsim sono il numero di simulazioni che si vogliono fare;
 # 2. m1 e m0 sono i vettori di medie nei malati (m1) e nei sani (m0);
 # 3. s1 e s0 sono le matrici di varianze e covarianze nei malati (s1) e
     nei sani (s0);
 # 4. n1 e n0 sono il numero di malati (n1) e di sani (n0);
 # 5. type indica se si deve considerare una trasformazione delle
     covariate oppure no:
 #  - type = 1 -> Nessuna trasformazione
 #  - type = 2 -> x2 = 1/x2
 #  - type = 3 -> x2 = x2^3
```

```
#  - type = 4 -> x2 = exp(x2)
require(pROC);require(mvtnorm);require(mgcv);require(earth)
#Creazione della barra del progresso
pb<- txtProgressBar(min=1, max=Nsim, style=3)
#Vettori per convergenze
convGAM<- convMARS<- convMARS2<- convLOG1<- rep(NA, Nsim)
#Definisco la matrice dove vanno a finire tutti gli AUC stimati
auc_val<- matrix(NA, ncol=4, nrow=Nsim)
colnames(auc_val)<- c('GAM', 'LMARS (d=1)', 'LMARS (d=2)', 'LOGISTIC')
#Definisco la matrice dove vanno a finire tutte le sensibilita' e
    specificita'
##GAM
seGAM_val<- spGAM_val<- matrix(NA, ncol=Nsim, nrow=1000)
##MARS (degree=1)
seMARS_val<- spMARS_val<- matrix(NA, ncol=Nsim, nrow=1000)
##MARS (degree=2)
seMARS2_val<- spMARS2_val<- matrix(NA, ncol=Nsim, nrow=1000)
##Logistic
seLOG_val<- spLOG_val<- matrix(NA, ncol=Nsim, nrow=1000)
#Inizio le simulazioni per GAM, LMARS (degree 1 e 2) e Logistic
for(i in 1:Nsim)
{
 #Generazione dei dati
 x1<- rbind(rmvnorm(n1,mean=m1,sigma=s1),rmvnorm(n0,mean=m0,sigma=s0))
 #Modifiche sulla seconda covariata?
 if(type==1) x1<- x1 #Nessuna
 if(type==2) x1[,2]<- x1[,2]^(-1) #Reciproco
 if(type==3) x1[,2]<- x1[,2]^3 #Cubo
 if(type==4) x1[,2]<- exp(x1[,2]) #Esponenziale
 y1 <- c(rep(1, n1), rep(0, n0));y1 <- factor(y1)
 simdat<- data.frame(y1, x1)
 colnames(simdat)<- c('y', 'x1', 'x2')
 #Stima dei modelli
 #GAM
 gam1<- gam(y~s(x1,bs='cr')+s(x2,bs='cr'),data=simdat,family=binomial,
     optimizer = c('outer','nlm'))
```

```
p_gam<- predict(gam1, type='response')
roc_gam<- myROC(as.vector(simdat$y), p_gam)
convGAM[i]<- gam1$converged
#MARS (degree=1)
mars1<- earth(y~x1+x2,data=simdat,glm=list(family=binomial,control=list(
    maxit=10^4)),trace=0)
p_mars<- predict(mars1, type='response')
roc_mars<- myROC(as.vector(simdat$y), as.vector(p_mars))
convMARS[i]<- mars1$glm.list[[1]][19]
#MARS (degree=2)
mars2<- earth(y~x1+x2,data=simdat,glm=list(family=binomial,control=list(
    maxit=10^4)),trace=0,degree=2)
p_mars2<- predict(mars2, type='response')
roc_mars2<- myROC(as.vector(simdat$y), as.vector(p_mars2))
convMARS2[i]<- mars2$glm.list[[1]][19]
#Logistic (non correttamente specificato)
log1<- glm(y~x1+x2,data=simdat,family=binomial,control=list(maxit=10^4))
p_log<- predict(log1, type='response')
roc_log<- myROC(as.vector(simdat$y), p_log)
convLOG1[i]<- log1$converged
#Salvo gli AUC
auc_val[i,]<-c(as.numeric(auc(as.vector(simdat$y),p_gam)),
               as.numeric(auc(as.vector(simdat$y),as.vector(p_mars))),
               as.numeric(auc(as.vector(simdat$y),as.vector(p_mars2))),
               as.numeric(auc(as.vector(simdat$y),p_log)))
#Salvo le sensibilita' e le specificita'
##GAM
seGAM_val[,i] <- roc_gam$Sensitivities
spGAM_val[,i] <- roc_gam$Specificities
##MARS (degree=1)
seMARS_val[,i] <- roc_mars$Sensitivities
spMARS_val[,i] <- roc_mars$Specificities
##MARS (degree=1)
seMARS2_val[,i] <- roc_mars2$Sensitivities
spMARS2_val[,i] <- roc_mars2$Specificities
##Logistic
```

```
 seLOG_val[,i] <- roc_log$Sensitivities
 spLOG_val[,i] <- roc_log$Specificities
 Sys.sleep(0.001)
 setTxtProgressBar(pb, i)
}
#Sensibilita' e specificita' medie
##GAM
seGAM<-apply(seGAM_val,1,mean);spGAM<-apply(spGAM_val,1,mean)
##MARS (degree=1)
seMARS<-apply(seMARS_val,1,mean);spMARS<-apply(spMARS_val,1,mean)
##MARS (degree=2)
seMARS2<-apply(seMARS2_val,1,mean);spMARS2<-apply(spMARS2_val,1,mean)
##Logistic
seLOG<- apply(seLOG_val,1,mean);spLOG<-apply(spLOG_val,1,mean)
#Convergenze dei modelli
ris<-list(GAM=table(convGAM),MARS1=table(unlist(convMARS)),MARS2=table(
    unlist(convMARS2)),LOG1=table(convLOG1))
close(pb) #Chiusura della barra del progresso
#Ritorno delle quantita' utili a fine processo
list(AUC=auc_val,AUC_mean=apply(auc_val,2,mean),AUC_sd=apply(auc_val,2,sd
    ),spGAM_val=spGAM_val,seGAM_val=seGAM_val,spMARS_val=spMARS_val,
    seMARS_val=seMARS_val,spMARS2_val=spMARS2_val,seMARS2_val=seMARS2_val
    ,spLOG_val=spLOG_val,seLOG_val=seLOG_valM,spGAM=spGAM,seGAM=seGAM,
    spMARS=spMARS,seMARS=seMARS,spMARS2=spMARS2,seMARS2=seMARS2,spLOG=
    spLOG,seLOG=seLOG,GAM=convGAM,MARS1=unlist(convMARS),MARS2=unlist(
    convMARS2),LOG1=convLOG1,ris=ris)
}
```

Codice A.3: Funzione che simula il secondo scenario in cui si hanno due tests che provengono da una distribuzione esponenziale bivariata

```
#Funzione che genera dati dalla distribuzione esponenziale bivariata
rbivexp<- function(R, e0, e1, e2)
{
 x1<- x2<- rep(NA, R)
 for(i in 1:R)
 {
```

```
 z<-rexp(1,e0);x1[i]<-min(rexp(1,e1-e0),z);x2[i]<-min(rexp(1,e2-e0),z)
}
cbind(x1, x2)
}
sim2<-function(Nsim, eps, n1, n0, type=1)
{
###Info su parametri
#1. Nsim e' il numero di simulazioni da fare
#2. eps sono i parametri per generare dall'esponenziale bivariata
#   di cui i primi 3 sono per i malati, mentre gli altri per i sani
#3. n0 e n1 sono il numero di sani (n0) e di malati (n1) nella
    simulazione
#4. type indica se si deve considerare una trasformazione delle covariate
     oppure no:
#  - type = 1 -> Nessuna trasformazione
#  - type = 2 -> x2 = 1/x2
#  - type = 3 -> x2 = x2^3
#  - type = 4 -> x2 = exp(x2)
require(pROC);require(mgcv);require(earth)
pb<- txtProgressBar(min=1, max=Nsim, style=3)
epsDIS<- eps[1:3] #Parametri per i malati malati
epsNDIS<- eps[4:6] #Parametri per i sani
#Definisco la matrice dove vanno a finire tutti gli AUC stimati
auc_val<- matrix(NA, ncol=3, nrow=Nsim)
colnames(auc_val)<- c('GAM', 'LMARS', 'NO-LOGISTIC')
#Definisco la matrice dove vanno a finire tutte le sensibilita' e
    specificita'
##GAM
seGAM_val<- spGAM_val<- matrix(NA, ncol=Nsim, nrow=1000)
convGAM<- rep(NA, Nsim)
##MARS
seMARS_val<- spMARS_val<- matrix(NA, ncol=Nsim, nrow=1000)
convMARS<- rep(NA, Nsim)
##Logistic (Non specificato correttamente)
seLOG_val<- spLOG_val<- matrix(NA, ncol=Nsim, nrow=1000)
convLOG1<- rep(NA, Nsim)
```

```
#Inizio le simulazioni per GAM, LMARS e Logistic
for(i in 1:Nsim)
{
 #Generazione dei dati
 x1<- rbind(rbivexp(n1, epsDIS[1], epsDIS[2], epsDIS[3]), #Malati
            rbivexp(n0, epsNDIS[1], epsNDIS[2], epsNDIS[3])) #Sani
 #Modifica delle covariate?
 if(type==1) x1<- x1 #Nessuna
 if(type==2) x1[,2]<- x1[,2]^(-1) #Reciproco
 if(type==3) x1[,2]<- x1[,2]^3 #Cubo
 if(type==4) x1[,2]<- exp(x1[,2]) #Esponenziale
 y1 <- c(rep(1, n1), rep(0, n0)); y1 <- factor(y1)
 simdat<- data.frame(y1, x1)
 colnames(simdat)<- c('y', 'x1', 'x2')
 #Stima dei modelli
 #GAM
 gam1<-gam(y~s(x1,bs='cr')+s(x2,bs='cr'),data=simdat,family=binomial,
     optimizer=c('outer', 'nlm'))
 convGAM[i]<- gam1$converged
 p_gam<- predict(gam1, type='response')
 roc_gam<- myROC(as.vector(simdat$y), p_gam)
 #MARS
 f_mars<- as.formula('y~x1+x2')
 mars1<- earth(f_mars, data=simdat, glm=list(family=binomial,control=list
     (maxit=10^4)), trace=0)
 convMARS[i]<- mars1$glm.list[[1]][19]
 p_mars<- predict(mars1, type='response')
 roc_mars<- myROC(as.vector(simdat$y), as.vector(p_mars))
 ##Logistic (Non correttamente specificato)
 f_glm<- as.formula('y~x1+x2+I(x1^2)+I(x2^2)')
 log1<- glm(f_glm, data=simdat, family=binomial, control=list(maxit=10^4)
     )
 convLOG1[i]<- log1$converged
 p_log<- predict(log1, type='response')
 roc_log<- myROC(as.vector(simdat$y), p_log)
 #Salvo gli AUC
```

```
 auc_val[i,]<- c(as.numeric(auc(as.vector(simdat$y), p_gam)),
                 as.numeric(auc(as.vector(simdat$y),as.vector(p_mars))),
                   as.numeric(auc(as.vector(simdat$y), p_log)))
 #Salvo le sensibilita' e le specificita'
 ##GAM
 seGAM_val[,i]<-roc_gam$Sensitivities
 spGAM_val[,i]<-roc_gam$Specificities
 ##MARS
 seMARS_val[,i]<-roc_mars$Sensitivities
 spMARS_val[,i]<-roc_mars$Specificities
 ##Logistic (Non correttamente specificato)
 seLOG_val[,i]<-roc_log$Sensitivities
 spLOG_val[,i]<-roc_log$Specificities
 Sys.sleep(0.001)
 setTxtProgressBar(pb, i)
}
#Sensibilita' e specificita' medie
##GAM
seGAM<-apply(seGAM_val,1,mean);spGAM<-apply(spGAM_val,1,mean)
##MARS
seMARS<- apply(seMARS_val,1,mean);spMARS<-apply(spMARS_val,1,mean)
##Logistic (Non correttamente specificato)
seLOG<- apply(seLOG_val,1, mean);spLOG<-apply(spLOG_val,1,mean)
#Convergenze dei modelli
ris<-list(GAM=table(convGAM),MARS1=table(unlist(convMARS)),LOG1=table(
    convLOG1))
close(pb) #Chiusura della barra del progresso
#Riporto gli oggetti di interesse dell'intero processo
list(AUC=auc_val,AUC_mean=apply(auc_val,2,mean),AUC_sd=apply(auc_val,2,sd
    ),spGAM_val=spGAM_val,seGAM_val=seGAM_val,spMARS_val=spMARS_val,
    seMARS_val=seMARS_val,spLOG_val=spLOG_val,seLOG_val=seLOG_val,spGAM=
    spGAM,seGAM=seGAM,spMARS=spMARS,seMARS=seMARS,spLOG=spLOG,seLOG=seLOG
    ,GAM=convGAM,MARS=unlist(convMARS),LOG1=convLOG1,ris=ris)
}
```

Codice A.4: Funzione che simula il terzo scenario in cui si hanno quattro tests che provengono da una distribuzione normale a quattro dimensioni

```
#Funzione che genera i dati
normdatsim<- function(N, theta, mu, sigma)
{
 require(mvtnorm)
 d<- rbinom(N, size=1, theta) #Status di malattia
 val<- table(d);x<- matrix(NA, nrow=N, ncol=4) #Matrice di covariate
 x[d==1,]<- rmvnorm(val[2], mean=mu, sigma=sigma)
 x[d==0,]<- rmvnorm(val[1], mean=rep(0, 4), sigma=sigma)
 datasim<- data.frame(d, x)
 colnames(datasim)<- c('y', 'x1', 'x2', 'x3', 'x4')
 datasim
}
sim3<-function(Nsim, mu, sigma, theta, n, type=1)
{
 ###Info su parametri
 #1. Nsim e' il numero di simulazioni da fare
 #2. mu e' il vettore di medie da inserire per simulare i dati dei malati
     dalla normale a 4 dimensioni
 #3. sigma e' la matrice di varianze e covarianze per simulare i dati
     dalla normale a 4 dimensioni
 #4. theta e' la probabilita' a priori di malattia
 #5. n a' il numero di osservazioni generate nelle simulazioni
 #6. type indica se si deve considerare una trasformazione delle covariate
      oppure no:
 #  - type = 1 -> Nessuna trasformazione
 #  - type = 2 -> x2 = 1/x2
 #  - type = 3 -> x2 = x2^3
 #  - type = 4 -> x2 = exp(x2)
 require(pROC);require(mgcv);require(earth)
 pb<- txtProgressBar(min=1, max=Nsim, style=3)
 #Definisco la matrice dove vanno a finire tutti gli AUC stimati
 auc_val<- matrix(NA, ncol=3, nrow=Nsim)
 colnames(auc_val)<- c('GAM', 'LMARS', 'NO-LOGISTIC')
 #Definisco la matrice dove vanno a finire tutte le sensibilita' e
```

```
    specificita'
##GAM
seGAM_val<- spGAM_val<- matrix(NA, ncol=Nsim, nrow=1000)
convGAM<- rep(NA, Nsim)
##MARS
seMARS_val<- spMARS_val<- matrix(NA, ncol=Nsim, nrow=1000)
convMARS<- rep(NA, Nsim)
##Logistic Regression (Non specificato correttamente)
seLOG_val<- spLOG_val<- matrix(NA, ncol=Nsim, nrow=1000)
convLOG1<- rep(NA, Nsim)
#Inizio le simulazioni per GAM, LMARS e Logistic
for(i in 1:Nsim)
{
 #Generazione dei dati
 simdat<- normdatsim(n, theta, mu, sigma)
 simdat$y<- factor(simdat$y)
 if(type==1) simdat<- simdat
 if(type==2) simdat$x3<- simdat$x3^(-1)
 if(type==3) simdat$x3<- simdat$x3^3
 if(type==4) simdat$x3<- exp(simdat$x3)

 #Stima dei modelli
 #GAM
 gam1<-gam(y~s(x1,bs='cr')+s(x2,bs='cr')+s(x3,bs='cr')+s(x4,bs='cr'),data
    =simdat,family=binomial,optimizer=c('outer','nlm'))
 convGAM[i]<- gam1$converged
 p_gam<- predict(gam1, type='response')
 roc_gam<- myROC(as.vector(simdat$y), p_gam)
 #MARS
 f_mars<- as.formula('y~x1+x2+x3+x4')
 mars1<-earth(f_mars,data=simdat,glm=list(family=binomial,control=list(
    maxit=10^4)), trace=0)
 convMARS[i]<- mars1$glm.list[[1]][19]
 p_mars<- predict(mars1, type='response')
 roc_mars<- myROC(as.vector(simdat$y), as.vector(p_mars))
 ##Logistic (Sovraspecificato)
```

```
 f_glm<-as.formula('y~x1+x2+x3+x4+I(x1^2)+I(x2^2)+I(x3^2)+I(x4^2)')
 log1<- glm(f_glm, data=simdat, family=binomial, control=list(maxit=10^4)
     )
 convLOG1[i]<- log1$converged
 p_log<- predict(log1, type='response')
 roc_log<- myROC(as.vector(simdat$y), p_log)
 #Salvo gli AUC
 auc_val[i,]<- c(as.numeric(auc(as.vector(simdat$y), p_gam)),
                 as.numeric(auc(as.vector(simdat$y),as.vector(p_mars))),
                   as.numeric(auc(as.vector(simdat$y), p_log)))
 #Salvo le sensibilita' e le specificita'
 ##GAM
 seGAM_val[,i] <- roc_gam$Sensitivities
 spGAM_val[,i] <- roc_gam$Specificities
 ##MARS
 seMARS_val[,i] <- roc_mars$Sensitivities
 spMARS_val[,i] <- roc_mars$Specificities
 ##Logistic (Sovraspecificato)
 seLOG_val[,i] <- roc_log$Sensitivities
 spLOG_val[,i] <- roc_log$Specificities
 Sys.sleep(0.001)
 setTxtProgressBar(pb, i)
}
#Sensibilita' e specificita' medie
##GAM
seGAM<-apply(seGAM_val,1,mean);spGAM<-apply(spGAM_val,1,mean)
##MARS
seMARS<-apply(seMARS_val,1,mean);spMARS<-apply(spMARS_val,1,mean)
##Logistic (Sovraspecificato)
seLOG<-apply(seLOG_val,1,mean);spLOG<-apply(spLOG_val,1,mean)
#Convergenze dei modelli
ris<-list(GAM=table(convGAM),MARS1=table(unlist(convMARS)),LOG1=table(
    convLOG1))
close(pb) #Chiusura della barra del progresso
#Riporto gli oggetti rilevanti dell'intero processo
list(AUC=auc_val,AUC_mean=apply(auc_val,2,mean),AUC_sd=apply(auc_val,2,sd
```

```
    ),spGAM_val=spGAM_val,seGAM_val=seGAM_val,spMARS_val=spMARS_val,
    seMARS_val=seMARS_val,spLOG_val=spLOG_val,seLOG_val=seLOG_val,spGAM=
    spGAM,seGAM=seGAM,spMARS=spMARS,seMARS=seMARS,spLOG=spLOG,seLOG=seLOG
    ,GAM=convGAM,MARS=unlist(convMARS),LOG1=convLOG1,ris=ris)
}
```

Codice A.5: Funzione che simula il quarto scenario in cui si hanno quattro tests che provengono da una distribuzione normale a quattro dimensioni

```
#Funzione che genera i dati
normdatsim2<- function(N, theta, mu, sigma1, sigma0)
{
 require(mvtnorm)
 d<- rbinom(N, size=1, theta)
 val<- table(d)
 x<- matrix(NA, nrow=N, ncol=4)
 x[d==1,]<- rmvnorm(val[2], mean=mu, sigma=sigma1)
 x[d==0,]<- rmvnorm(val[1], mean=rep(0, 4), sigma=sigma0)
 datasim<- data.frame(d, x)
 colnames(datasim)<- c('y', 'x1', 'x2', 'x3', 'x4')
 datasim
}
sim4<-function(Nsim, mu, theta, sigma1, sigma0, n, type=1)
{
 ###Info su parametri
 #1. Nsim e' il numero di simulazioni da fare
 #2. mu e' il vettore di medie da inserire per simulare i dati dalla
    normale a 4 dimensioni
 #3. sigma1 e sigma0 sono le matrici di varianze e covarianze per simulare
     i dati dalla normale a 4 dimensioni in malati e sani
 #4. theta e' la probabilita' a priori di malattia
 #5. n e' il numero di osservazioni generate nelle simulazioni
 #6. type indica se si deve considerare una trasformazione delle covariate
     oppure no:
 #  - type = 1 -> Nessuna trasformazione
 #  - type = 2 -> x3 = 1/x3
 #  - type = 3 -> x3 = x3^3
```

```
#  - type = 4 -> x3 = exp(x3)
require(pROC);require(mgcv);require(earth)
pb<- txtProgressBar(min=1, max=Nsim, style=3)
#Definisco la matrice dove vanno a finire tutti gli AUC stimati
auc_val<- matrix(NA, ncol=4, nrow=Nsim)
colnames(auc_val)<- c('GAM', 'LMARS(d=1)', 'LMARS(d=2)', 'NO-LOGISTIC')
#Definisco la matrice dove vanno a finire tutte le sensibilita' e
    specificita'
##GAM
seGAM_val<- spGAM_val<- matrix(NA, ncol=Nsim, nrow=1000)
convGAM<- rep(NA, Nsim)
##MARS
seMARS_val<- spMARS_val<- matrix(NA, ncol=Nsim, nrow=1000)
convMARS<- rep(NA, Nsim)
##MARS (degree=2)
seMARS2_val<- spMARS2_val<- matrix(NA, ncol=Nsim, nrow=1000)
convMARS2<- rep(NA, Nsim)
##Logistic Regression (Non specificato correttamente)
seLOG_val<- spLOG_val<- matrix(NA, ncol=Nsim, nrow=1000)
convLOG1<- rep(NA, Nsim)
#Inizio le simulazioni per GAM, LMARS e Logistic
for(i in 1:Nsim)
{
 #Generazione dei dati
 simdat<- normdatsim2(n, theta, mu, sigma1, sigma0)
 simdat$y<- factor(simdat$y)
 if(type==1) simdat<- simdat
 if(type==2) simdat$x3<- simdat$x3^(-1)
 if(type==3) simdat$x3<- simdat$x3^3
 if(type==4) simdat$x3<- exp(simdat$x3)
 #Stima dei modelli
 #GAM
 gam1<-gam(y~s(x1,bs='cr')+s(x2,bs='cr')+s(x3,bs='cr')+s(x4,bs='cr'),data
     =simdat,family=binomial,optimizer=c('outer','nlm'))
 convGAM[i]<- gam1$converged
 p_gam<- predict(gam1, type='response')
```

```
roc_gam<- myROC(as.vector(simdat$y), p_gam)
#MARS (degree=1)
f_mars<- as.formula('y~x1+x2+x3+x4')
mars1<-earth(f_mars,data=simdat,glm=list(family=binomial,control=list(
    maxit=10^4)),trace=0)
convMARS[i]<- mars1$glm.list[[1]][19]
p_mars<- predict(mars1, type='response')
roc_mars<- myROC(as.vector(simdat$y), as.vector(p_mars))
#MARS(degree=2)
mars2<-earth(f_mars,data=simdat,glm=list(family=binomial,control=list(
    maxit=10^4)),trace=0,degree=2)
convMARS2[i]<- mars2$glm.list[[1]][19]
p_mars2<- predict(mars2, type='response')
roc_mars2<- myROC(as.vector(simdat$y), as.vector(p_mars2))
#Logistic (Non correttamente specificato)
f_glm<- as.formula('y~x1+x2+x3+x4+I(x1^2)+I(x2^2)+I(x3^2)+I(x4^2)')
log1<- glm(f_glm,data=simdat,family=binomial,control=list(maxit=10^4))
convLOG1[i]<- log1$converged
p_log<- predict(log1, type='response')
roc_log<- myROC(as.vector(simdat$y), p_log)
#Salvo gli AUC
auc_val[i,]<-c(as.numeric(auc(as.vector(simdat$y), p_gam)),
               as.numeric(auc(as.vector(simdat$y),as.vector(p_mars))),
              as.numeric(auc(as.vector(simdat$y),as.vector(p_mars2))),
              as.numeric(auc(as.vector(simdat$y), p_log)))

#Salvo le sensibilita' e le specificita'
##GAM
seGAM_val[,i] <- roc_gam$Sensitivities
spGAM_val[,i] <- roc_gam$Specificities
##MARS (degree=1)
seMARS_val[,i] <- roc_mars$Sensitivities
spMARS_val[,i] <- roc_mars$Specificities
##MARS (degree=2)
seMARS2_val[,i] <- roc_mars2$Sensitivities
spMARS2_val[,i] <- roc_mars2$Specificities
```

```
 ##Logistic (Non correttamente specificato)
 seLOG_val[,i] <- roc_log$Sensitivities
 spLOG_val[,i] <- roc_log$Specificities
 Sys.sleep(0.001)
 setTxtProgressBar(pb, i)
}
#Sensibilita' e specificita' medie
##GAM
seGAM<-apply(seGAM_val,1,mean);spGAM<-apply(spGAM_val,1,mean)
##MARS (degree=1)
seMARS<-apply(seMARS_val,1,mean);spMARS<-apply(spMARS_val,1,mean)
##MARS (degree=2)
seMARS2<-apply(seMARS2_val,1,mean);spMARS2<-apply(spMARS2_val,1,mean)
##Logistic (Non correttamente specificato)
seLOG<-apply(seLOG_val,1,mean);spLOG<-apply(spLOG_val,1,mean)
#Convergenze dei modelli
ris<-list(GAM=table(convGAM),MARS1=table(unlist(convMARS)),MARS2=table(
    unlist(convMARS2)),LOG1=table(convLOG1))
close(pb) #Chiusura della barra del progresso
#Riporto gli oggetti rilevanti dell'intero processo
list(AUC=auc_val,AUC_mean=apply(auc_val,2,mean),AUC_sd=apply(auc_val,2,sd
    ),spGAM_val=spGAM_val,seGAM_val=seGAM_val,spMARS_val=spMARS_val,
    seMARS_val=seMARS_val,spMARS2_val=spMARS2_val, seMARS2_val=seMARS2_
    val,spLOG_val=spLOG_val,seLOG_val=seLOG_val,spGAM=spGAM,seGAM=seGAM,
    spMARS=spMARS,seMARS=seMARS,spMARS2=spMARS2,seMARS2=seMARS2,spLOG=
    spLOG,seLOG=seLOG,GAM=convGAM,MARS1=unlist(convMARS),MARS2=unlist(
    convMARS2),LOG1=convLOG1,ris=ris)
}
```

Codice A.6: Funzione che simula il quinto scenario con due test di cui uno proveniente da una normale standard mentre l'altro da una distribuzione uniforme

```
#Funzione che genera i dati
logitsim<-function(N, beta)
{
 t1<- rnorm(N) #Test 1
```

```
 t2<- runif(N, -1, 1) #Test 2
 dat<- data.frame(t1, t2)
 eta<- beta[1]+beta[2]*dat$t1+beta[3]*t2+beta[4]*dat$t1^2+beta[5]*dat$t2
     ^2+beta[6]*dat$t1*dat$t2 #Predittore lineare
 prob<- exp(eta)/(1+exp(eta)) #Probabilita'
 d<- rbinom(N, size=1, prob=prob) #Status di malattia
 dat<- data.frame(d, t1, t2)
 colnames(dat)<- c('y', 'x1', 'x2')
 list(P=prob, dati=dat)
}

sim5<-function(Nsim, n, beta, type=1)
{
 ###Info su parametri
 #1. Nsim e' il numero di simulazioni da fare
 #2. n e' il numero di osservazioni simulate nelle simulazioni
 #3. beta e' il vettore di parametri relativi al vero risk score
 #4. type indica se si deve considerare una trasformazione delle covariate
      oppure no:
 #  - type = 1 -> Nessuna trasformazione
 #  - type = 2 -> x2 = 1/x2
 #  - type = 3 -> x2 = x2^3
 #  - type = 4 -> x2 = exp(x2)
 require(pROC);require(mgcv);require(earth)
 pb<- txtProgressBar(min=1, max=Nsim, style=3)
 #Definisco la matrice dove vanno a finire tutti gli AUC stimati
 auc_val<- matrix(NA, ncol=4, nrow=Nsim)
 colnames(auc_val)<- c('GAM', 'LMARS(d=1)', 'LMARS(d=2)', 'NO-LOGISTIC')
 #Definisco la matrice dove vanno a finire tutte le sensibilita' e
     specificita'
 ##GAM
 seGAM_val<- spGAM_val<- matrix(NA, ncol=Nsim, nrow=1000)
 convGAM<- rep(NA, Nsim)
 ##MARS (degree=1)
 seMARS_val<- spMARS_val<- matrix(NA, ncol=Nsim, nrow=1000)
 convMARS<- rep(NA, Nsim)
```

```
##MARS (degree=2)
seMARS2_val<- spMARS2_val<- matrix(NA, ncol=Nsim, nrow=1000)
convMARS2<- rep(NA, Nsim)
##Logistic (Non specificato correttamente)
seLOG_val<- spLOG_val<- matrix(NA, ncol=Nsim, nrow=1000)
convLOG1<- rep(NA, Nsim)
#Inizio le simulazioni per GAM, LMARS (degree 1 e 2) e Logistic
for(i in 1:Nsim)
{
 simdat<- logitsim(n, beta)
 simdat$dati$y<- factor(simdat$dati$y)
 if(type==1) simdat<- simdat
 if(type==2) simdat$dati[,3]<- simdat$dat[,3]^(-1)
 if(type==3) simdat$dati[,3]<- simdat$dati[,3]^3
 if(type==4) simdat$dati[,3]<- exp(simdat$dati[,3])
 #Stima dei modelli
 #GAM
 gam1<-gam(y~s(x1, bs='cr')+s(x2, bs='cr'),data=simdat$dati,family=
     binomial,optimizer=c('outer','nlm'))
 convGAM[i]<- gam1$converged
 p_gam<- predict(gam1, type='response')
 roc_gam<- myROC(as.vector(simdat$dati$y), p_gam)
 #MARS (degree=1)
 f_mars<- as.formula('y~x1+x2')
 mars1<-earth(f_mars,data=simdat$dati,glm=list(family=binomial,control=
     list(maxit=10^4)),trace=0)
 convMARS[i]<- mars1$glm.list[[1]][19]
 p_mars<- predict(mars1, type='response')
 roc_mars<- myROC(as.vector(simdat$dati$y), as.vector(p_mars))
 #MARS(degree=2)
 mars2<-earth(f_mars,data=simdat$dati,glm=list(family=binomial,control=
     list(maxit=10^4)),trace=0,degree=2)
 convMARS2[i]<- mars2$glm.list[[1]][19]
 p_mars2<- predict(mars2, type='response')
 roc_mars2<- myROC(as.vector(simdat$dati$y), as.vector(p_mars2))
 #Logistic (Non correttamente specificato)
```

```
 f_glm<- as.formula('y~x1+x2')
 log1<- glm(f_glm, data=simdat$dati, family=binomial, control=list(maxit
     =10000))
 convLOG1[i]<- log1$converged
 p_log<- predict(log1, type='response')
 roc_log<- myROC(as.vector(simdat$dati$y), p_log)
 #Salvo gli AUC
 auc_val[i,]<-c(as.numeric(auc(as.vector(simdat$dati$y),p_gam)),
                as.numeric(auc(as.vector(simdat$dati$y),as.vector(p_mars)
                    )),
            as.numeric(auc(as.vector(simdat$dati$y),as.vector(p_mars2))),
            as.numeric(auc(as.vector(simdat$dati$y),p_log)))
 #Salvo le sensibilita' e le specificita'
 ##GAM
 seGAM_val[,i] <- roc_gam$Sensitivities
 spGAM_val[,i] <- roc_gam$Specificities
 ##MARS (degree=1)
 seMARS_val[,i] <- roc_mars$Sensitivities
 spMARS_val[,i] <- roc_mars$Specificities
 ##MARS (degree=2)
 seMARS2_val[,i] <- roc_mars2$Sensitivities
 spMARS2_val[,i] <- roc_mars2$Specificities
 ##Logistic (Non correttamente specificato)
 seLOG_val[,i] <- roc_log$Sensitivities
 spLOG_val[,i] <- roc_log$Specificities
 Sys.sleep(0.001)
 setTxtProgressBar(pb, i)
}
#Sensibilita' e specificita' medie
##GAM
seGAM<-apply(seGAM_val,1,mean);spGAM<-apply(spGAM_val,1,mean)
##MARS (degree=1)
seMARS<-apply(seMARS_val,1,mean);spMARS<-apply(spMARS_val,1,mean)
##MARS (degree=2)
seMARS2<-apply(seMARS2_val,1,mean);spMARS2<-apply(spMARS2_val,1,mean)
##Logistic (Non correttamente specificato)
```

```
 seLOG<-apply(seLOG_val,1,mean);spLOG<-apply(spLOG_val,1,mean)
 #Convergenze dei modelli
 ris<-list(GAM=table(convGAM),MARS1=table(unlist(convMARS)),MARS2=table(
     unlist(convMARS2)),LOG1=table(convLOG1))
 close(pb) #Chiusura della barra del progresso
 #Riporto gli oggetti rilevanti dell'intero processo
 list(AUC=auc_val,AUC_mean=apply(auc_val,2,mean),AUC_sd=apply(auc_val,2,sd
     ),spGAM_val=spGAM_val,seGAM_val=seGAM_val,spMARS_val=spMARS_val,
     seMARS_val=seMARS_val,spMARS2_val=spMARS2_val,seMARS2_val=seMARS2_val
     ,spLOG_val=spLOG_val,seLOG_val=seLOG_val,spGAM=spGAM,seGAM=seGAM,
     spMARS=spMARS,seMARS=seMARS,spMARS2=spMARS2,seMARS2=seMARS2,spLOG=
     spLOG,seLOG=seLOG,GAM=convGAM,MARS1=unlist(convMARS),MARS2=unlist(
     convMARS2),LOG1=convLOG1,ris=ris)
}
```

Codice A.7: Funzione che effettua le simulazioni sul bootstrap nel primo scenario

```
#Funzione che calcola gli AUC per i vari modelli
theta1<-function(data, i)
{
 mgam<-gam(y~s(x1,bs='cr')+s(x2,bs='cr'),family=binomial,data=data[i,],
     optimizer=c('outer', 'nlm'))
 mmars<-earth(y~x1+x2,data=data[i,],glm=list(family=binomial,control=list(
     maxit=10000)),trace=0)
 mmars2<-earth(y~x1+x2,data=data[i,],glm=list(family=binomial,control=list
     (maxit=10000)),trace=0,degree=2)
 pgam<-predict(mgam,type='response')
 pmars<-as.vector(predict(mmars, type='response'))
 pmars2<-as.vector(predict(mmars2, type='response'))
 aucs<-c(auc(data[i,1],pgam),auc(data[i,1],pmars),auc(data[i,1],pmars2),
     mgam$converged,unlist(mmars$glm.list[[1]][19]),unlist(mmars2$glm.list
     [[1]][19]))
 names(aucs)<- c('GAM','MARS (d=1)','MARS(d=2)','ConvGAM','ConvMARS1','
     ConvMARS2')
 aucs
}
#Funzione che effettua il bootstrap stratificato
```

```
auc.boot1<- function(data, nboot, theta, conf.level)
{
 datab<-matrix(NA, nrow=nboot, ncol=nrow(data))
 datab[,1:nrow(data[data$y==0,])]<-sample(1:nrow(data[data$y==0,]),replace
     =T,size=nrow(data[data$y==0,])*nboot)
 datab[,(nrow(data[data$y==0,])+1):nrow(data)]<-sample((nrow(data[data$y
     ==0,])+1):nrow(data),replace=T,size=nrow(data[data$y==1,])*nboot)
 t0<- theta(data, 1:nrow(data))
 risboot <- apply(datab, 1, function(x) theta(data[x,], 1:ncol(datab)))
 t<- risboot[1:3,]
 ts<- apply(t, 1, mean)
 conv<- risboot[4:6,]
 bias<- apply(t, 1, mean)-t0
 se<- apply(t, 1, sd)
 ic<- apply(t, 1, quantile, c(0+(1-conf.level)/2, 1-(1-conf.level)/2))
 return(list(t0=t0, t=t, conv=conv, ts=ts, B=bias, se=se, ic=ic))
}
#Funzione che effettua la simulazione nel primo scenario
simboot1<- function(Nsim,n1,n0,m1,m0,s1,s0,nboot,theta,conf.level=.95)
{
 require(mgcv);require(earth);require(pROC);require(mvtnorm)
 pb<- txtProgressBar(min=1, max=Nsim, style=3)

 t_boot<-matrix(NA,ncol=3,nrow=Nsim);colnames(t_boot)<-c('GAM','MARS(d=1)'
     ,'MARS(d=2)')
 sd_boot<-matrix(NA,ncol=3,nrow=Nsim);colnames(sd_boot)<-c('GAM','MARS(d
     =1)','MARS(d=2)')
 ic_boot<-matrix(NA,ncol=6,nrow=Nsim);colnames(ic_boot)<-c('GAM_LOW','GAM_
     UPPER','MARS(d=1)_LOW','MARS(d=1)_UPPER','MARS(d=2)_LOW','MARS(d=2)_
     UPPER')
 for(i in 1:Nsim)
 {
  #Creo i dati per l'i-esimo ciclo
  xtest<-rbind(rmvnorm(n1, mean=m1, sigma=s1),rmvnorm(n0, mean=m0, sigma=
      s0))
  ytest<-c(rep(1, n1),rep(0, n0))
```

```
  ytest<- factor(ytest)
  dat<- data.frame(ytest, xtest)
  colnames(dat)<-c('y', 'x1', 'x2')
  iboot<-auc.boot2(dat, nboot, theta, conf.level)
  t_boot[i,]<- iboot$ts
  sd_boot[i,]<- iboot$se
  ic_boot[i,]<- as.vector(iboot$ic)
  Sys.sleep(0.001)
  setTxtProgressBar(pb, i)
 }
 close(pb)
 list(t_boot=t_boot,sd_boot=sd_boot,ic_boot=ic_boot)
}
```

Codice A.8: Funzione che effettua le simulazioni sul bootstrap nel secondo scenario

```
#Funzione che calcola gli AUC per i vari modelli
theta2<-function(data, i)
{
 mgam  <- gam(y~s(x1, bs='cr')+s(x2, bs='cr'), family=binomial, data=data[
     i,], optimizer=c('outer', 'nlm'))
 mmars <- earth(y~x1+x2, data=data[i,], glm=list(family=binomial, control=
     list(maxit=10000)), trace=0)
 pgam  <- predict(mgam, type='response')
 pmars <- as.vector(predict(mmars, type='response'))
 aucs<-c(auc(data[i,1], pgam),auc(data[i,1], pmars),mgam$converged,unlist(
     mmars$glm.list[[1]][19]))
 names(aucs)<- c('GAM', 'MARS', 'ConvGAM','ConvMARS1')
 aucs
}
#Funzione che effettua il bootstrap stratificato
auc.boot2<- function(data, nboot, theta, conf.level)
{
 datab<-matrix(NA, nrow=nboot, ncol=nrow(data))
 datab[, 1:nrow(data[data$y==0,])]<-sample(1:nrow(data[data$y==0,]),
     replace=T,size=nrow(data[data$y==0,])*nboot)
```

```
 datab[, (nrow(data[data$y==0,])+1):nrow(data)]<- sample((nrow(data[data$y
     ==0,])+1):nrow(data),replace=T,size=nrow(data[data$y==1,])*nboot)
 t0<- theta(data, 1:nrow(data))
 risboot <- apply(datab, 1, function(x) theta(data[x,], 1:ncol(datab)))
 t<- risboot[1:2,]
 ts<- apply(t, 1, mean)
 conv<- risboot[3:4,]
 bias<- apply(t, 1, mean)-t0
 se<- apply(t, 1, sd)
 ic<- apply(t, 1, quantile, c(0+(1-conf.level)/2, 1-(1-conf.level)/2))
 return(list(t0=t0, t=t, conv=conv, ts=ts, B=bias, se=se, ic=ic))
}
#Funzione che effettua la simulazione nel secondo scenario
simboot2<- function(Nsim, n1, n0, eps, nboot, theta, conf.level=.95)
{
 require(mgcv);require(earth);require(pROC)
 pb<- txtProgressBar(min=1, max=Nsim, style=3)

 t_boot<- matrix(NA, ncol=2, nrow=Nsim);colnames(t_boot)<- c('GAM', 'MARS'
     )
 sd_boot<- matrix(NA, ncol=2, nrow=Nsim);colnames(sd_boot)<- c('GAM', '
    MARS')
 ic_boot<- matrix(NA, ncol=4, nrow=Nsim);colnames(ic_boot)<- c('GAM.LOW',
     'GAM.UPPER', 'MARS.LOW','MARS.UPPER')
 epsDIS<- eps[1:3];epsNDIS<- eps[4:6]
 for(i in 1:Nsim)
 {
  xtest<-rbind(rbivexp(n1, epsDIS[1], epsDIS[2], epsDIS[3]),
               rbivexp(n0, epsNDIS[1], epsNDIS[2], epsNDIS[3]))
  ytest<-c( rep(1, n1),rep(0, n0) )
  ytest<- factor(ytest)
  dat<- data.frame(ytest, xtest)
  colnames(dat)<-c('y', 'x1', 'x2')
  iboot<-auc.boot2(dat, nboot, theta, conf.level)
  t_boot[i,]<- iboot$ts
  sd_boot[i,]<- iboot$se
```

```
  ic_boot[i,]<- as.vector(iboot$ic)
  Sys.sleep(0.001)
  setTxtProgressBar(pb, i)
  }
 close(pb)
 list(t_boot=t_boot,sd_boot=sd_boot,ic_boot=ic_boot)
}
```

Appendice B

Tabelle

	Matrici 1		**Matrici 2**		**Matrici 3**	
	$\hat{\beta}$	SE($\hat{\beta}$)	$\hat{\beta}$	SE($\hat{\beta}$)	$\hat{\beta}$	SE($\hat{\beta}$)
Intercept	-2.6929	0.0419	-2.2626	0.0416	-4.8001	0.1122
T_1	-0.2730	0.0232	-0.6019	0.0421	-0.5591	0.0518
T_2	0.5387	0.0207	1.2026	0.0512	1.0948	0.0772
T_1^2	0.2462	0.0072	0.4148	0.0160	0.0453	0.0070
T_2^2	0.2207	0.0075	1.1083	0.0304	0.3711	0.0223
T_1T_2	-0.1666	0.0105	-1.0546	0.0393	-0.0501	0.0180

Tabella B.1: Stime dei parametri e relativi standard error del vero *risk score* relativo al primo scenario con matrici di varianze e covarianze specificate in maniera differente. In particolare Matrici 1 (colonne 1 e 2) sono quelle definite in §3.1, Matrici 2 (colonne 3 e 4) accentuano il termine di interazione e Matrici 3 (colonne 5 e 6) il termine quadratico riducendo al minimo l'effetto interazione

	$\hat{\beta}$	SE($\hat{\beta}$)
Intercept	-2.3680	0.0354
T_1	5.9824	0.1461
T_2	6.0498	0.1476

Tabella B.2: Stime dei parametri e relativi standard error del vero *risk score* relativo al secondo scenario

	$\theta = 0.1$		$\theta = 0.3$	
	$\hat{\beta}$	SE($\hat{\beta}$)	$\hat{\beta}$	SE($\hat{\beta}$)
Intercept	-3.4665	0.0675	-2.1615	0.0438
T_1	1.2156	0.0429	1.2218	0.0316
T_2	-0.4094	0.0355	-0.3539	0.0258
T_3	2.6462	0.0894	2.6474	0.0672
T_4	-1.7384	0.0787	-1.7245	0.0576

Tabella B.3: Stime dei parametri e relativi standard error del vero *risk score* relativo al terzo scenario per due differenti probabilità di malattia a priori

	$\hat{\beta}$	SE($\hat{\beta}$)		$\hat{\beta}$	SE($\hat{\beta}$)
Intercept	-1.4825	0.0582			
T_1	0.6595	0.0380	T_1^2	0.2519	0.0211
T_2	0.2509	0.0342	T_2^2	0.1303	0.0178
T_3	0.6741	0.0616	T_3^2	-0.9880	0.0465
T_4	0.3356	0.0549	T_4^2	-0.6609	0.0406
T_1T_2	0.0552	0.0281	T_2T_3	0.6455	0.0421
T_1T_3	0.1128	0.0419	T_2T_4	-0.4029	0.0403
T_1T_4	-0.1554	0.0415	T_3T_4	1.7753	0.0745

Tabella B.4: Stime dei parametri e relativi standard error del vero *risk score* relativo al quarto scenario