



Politecnico di Torino

Laurea Magistrale in Ingegneria Informatica

Tesina di Data Spaces

01RLPOV

Docente:

Prof. Francesco Vaccarino

Studente:

Montarolo Marco, S266608

## Introduzione

Lo scopo di questo elaborato è l'analisi di un set di dati medici attraverso algoritmi di classificazione per evidenziare le condizioni e i fattori di rischio collegati.

I dati sono relativi a problemi cardiaci originariamente raccolti all'interno di una ricerca condotta presso Government College University (Faisalabad, Pakistan) nel 2017 da Tanvir Ahmad, Assia Munir, Sajjad Haider Bhatti, Muhammad Aftab, e Muhammad Ali Raza.

La versione utilizzata in questo scritto tuttavia è quella elaborata da Davide Chicco (Krembil Research Institute, Toronto, Canada) e donata alla University of California Irvine Machine Learning Repository<sup>1</sup> nel gennaio 2020.

Gli stessi dati sono stati utilizzati all'interno dell'articolo *Machine learning can predict survival of patients with heart failure from serum creatinine and ejection fraction alone*<sup>1</sup>.

L'insieme dei dati è composto da 13 variabili:

Nome	Descrizione	Tipo	Unità di misura
Age	Età	Numerico	Anni
Anaemia	Anemia	Booleana	0,1
High blood pressure	Ipertensione	Booleana	0,1
Creatinine phosphokinase	Livello dell'enzima CPK	Numerico	mcg/L
Diabetes	Diabete	Booleana	0,1
Ejection fraction	Percentuale di sangue espulso dal cuore	Numerico	%
Sex	Sesso	Binario	0 (Donna), 1 (Uomo)
Plateles	Piastrine nel sangue	Numerico	Kiloplatelets/mL
Serum creatinine	Concentrazione creatinina nel sangue	Numerico	mg/dL
Serum sodium	Concentrazione di sodio nel sangue	Numerico	mEq/L
Smoking	Fumatore	Booleana	0,1
Time	Tempo di monitoraggio	Numerico	Giorni
Death event	Morte del paziente	Booleana	0,1

Ogni elemento del dataset rappresenta un paziente che è descritto da 13 variabili; i casi riportati sono 299.

All'interno dell'insieme delle caratteristiche la variabile *target* è *death event*, ovvero il risultato atteso, quindi le rimanenti saranno considerate come predittori, cioè come elementi caratterizzanti del risultato.

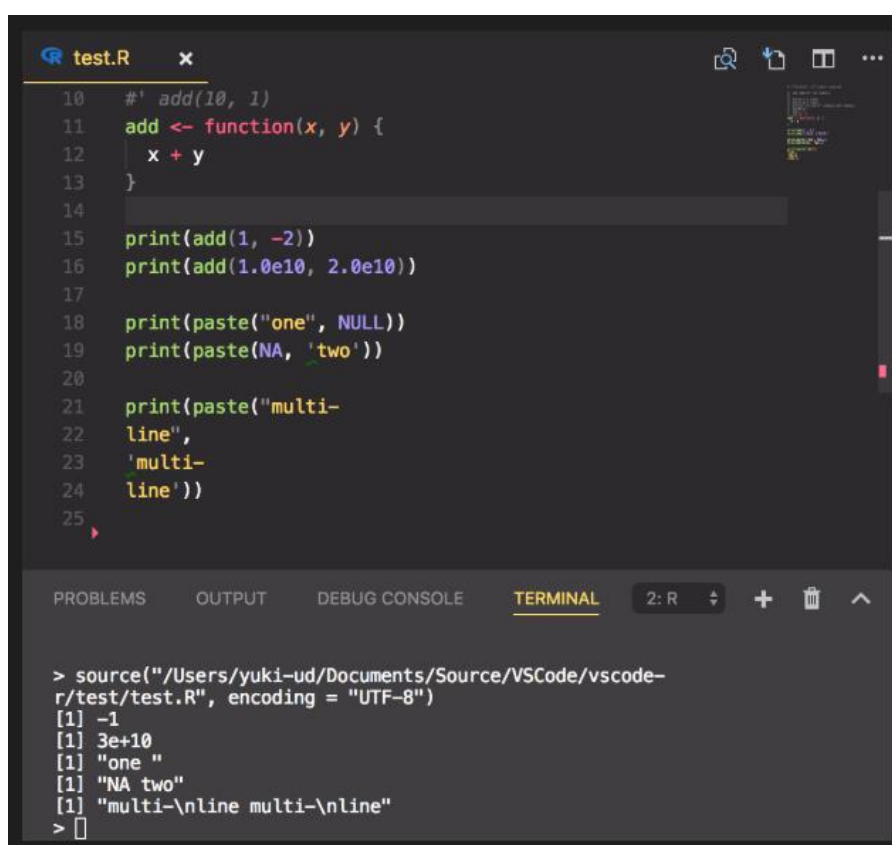
---

<sup>1</sup> <https://doi.org/10.1186/s12911-020-1023-5>

## Gli strumenti

L'analisi dei dati è stata eseguita utilizzando il linguaggio di programmazione R, specifico per l'analisi statistica, disponibile sotto licenza GNU GPL. Questo offre diverse opzioni per lo sviluppo sia tramite linea di comando sia attraverso ambienti integrati di sviluppo (IDE) quali Visual Studio Code, R Studio oppure strumenti che permettono la collaborazione come Jupyter Notebook.

Nella analisi qui illustrata si è preferito utilizzare R tramite linea di comando in combinazione con Visual Studio Code per la redazione del codice. Infatti all'interno del prodotto Microsoft è disponibile un'estensione<sup>2</sup> che agevola lo sviluppo tramite suggerimenti e scorciatoie tramite tastiera.



```
test.R
10 #' add(10, 1)
11 add <- function(x, y) {
12   x + y
13 }
14
15 print(add(1, -2))
16 print(add(1.0e10, 2.0e10))
17
18 print(paste("one", NULL))
19 print(paste(NA, 'two'))
20
21 print(paste("multi-
22 line",
23 'multi-
24 line'))
25

> source("/Users/yuki-ud/Documents/Source/VSCode/vscode-
r/test/test.R", encoding = "UTF-8")
[1] -1
[1] 3e+10
[1] "one "
[1] "NA two"
[1] "multi-\nline multi-\nline"
>
```

Figura 1 Screenshot dell'ambiente di sviluppo

Nella fase di sviluppo del codice sono state utilizzate funzionalità per la rappresentazione grafica e l'applicazione degli algoritmi forniti da terze parti per R, ma opensource: *tidyverse*, *ggpubr*, *tree*, *boot*, *MASS*, *ROCR*, *pROC*, *class*, *rpart*, *rpart.plot*.

I dati sono stati acquisiti in formato CSV (Comma Separated Value) e sommariamente analizzati per avere maggiore comprensione del contesto in cui si svilupperanno le successive analisi.

<sup>2</sup> <https://github.com/ikuyadeu/vscode-R/wiki>

## Le variabili qualitative

Le prime osservazioni si possono fare attraverso un'analisi della distribuzione dei valori qualitativi di tutto il dataset, ovvero di quelle variabili che assumono valori binari.

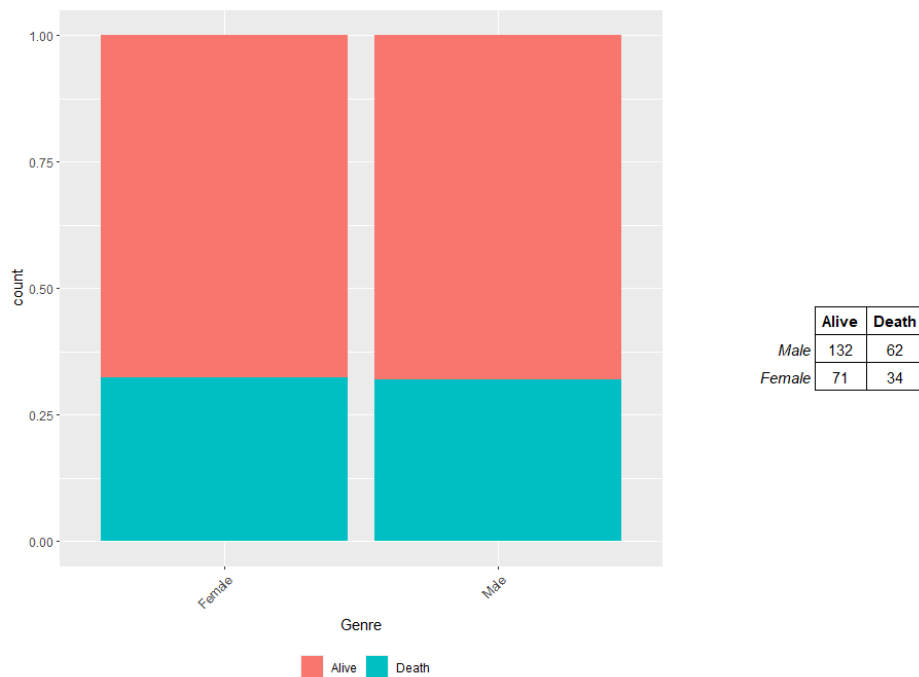


Figura 2 Rappresentazione della variabile sex

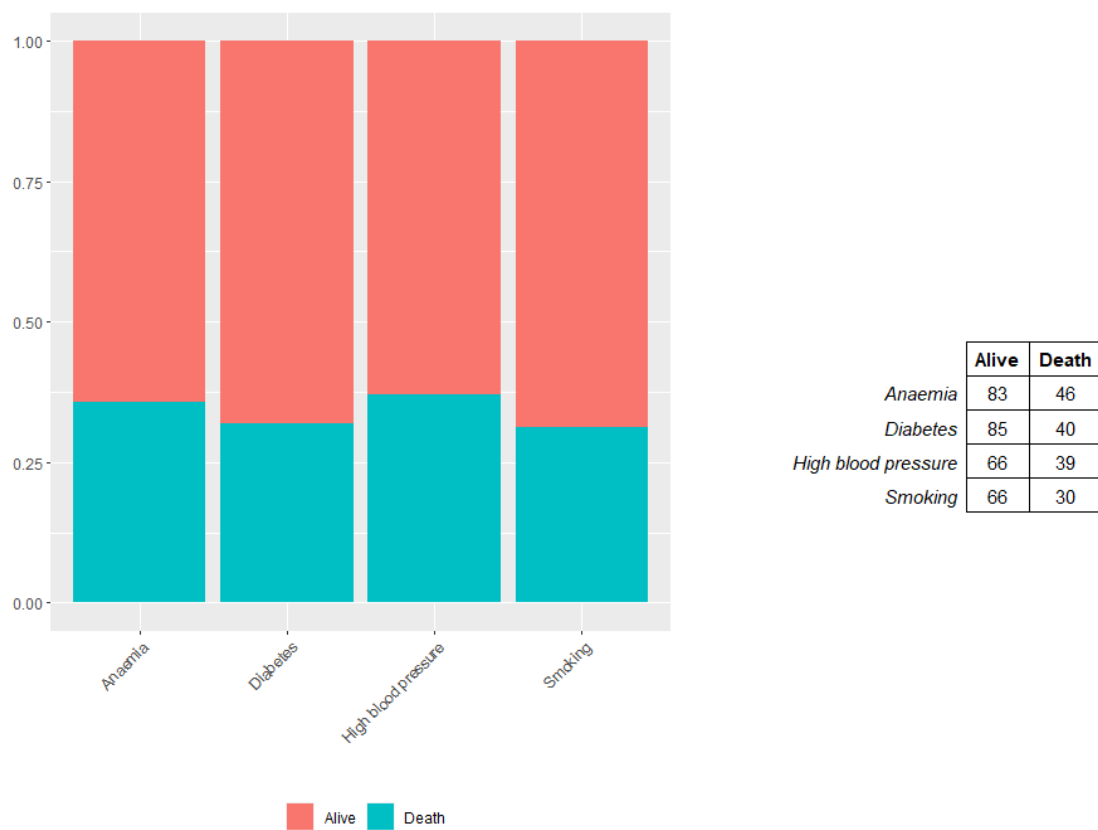


Figura 3 Rappresentazione delle patologie/abitudini

La variabile relativa al sesso degli elementi appartenenti al dataset è stata considerata separatamente perché non semanticamente correlata con le altre caratteristiche degli individui.

Dalla figura 3 si nota come non ci sia una caratteristica tra quelle rappresentate che possa corrispondere ad una maggiore predisposizione per problemi cardiaci, infatti ciascuna corrisponde ad una percentuale di rischio comparabile.

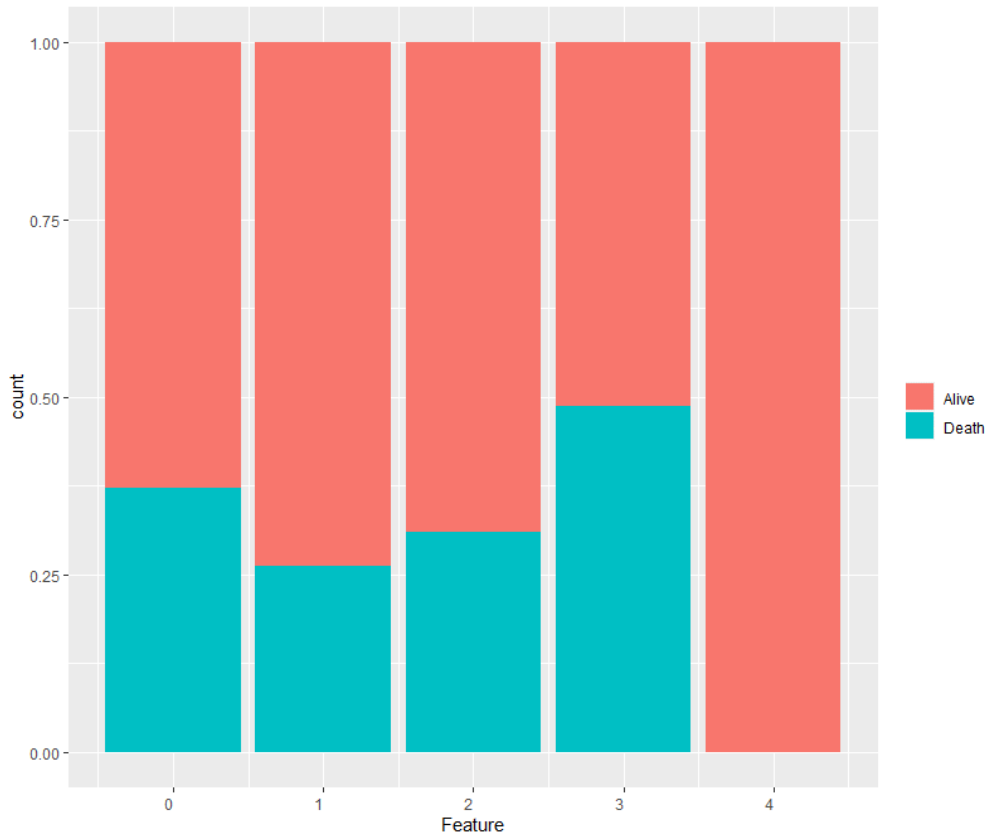


Figura 4

Si nota come la combinazione di tre fattori comporti un maggiore numero di casi in cui l’esito delle problematiche è risultato negativo.

### Le variabili quantitative

Ora verranno considerate solamente le variabili numeriche, definite quantitative.

age	creatinine_phosphokinase	ejection_fraction	platelets	serum_creatinine	serum_sodium	time
Min. :40.00	Min. : 23.0	Min. :14.00	Min. : 25100	Min. :0.500	Min. :113.0	Min. : 4.0
1st Qu.:51.00	1st Qu.: 116.5	1st Qu.:30.00	1st Qu.:212500	1st Qu.:0.900	1st Qu.:134.0	1st Qu.: 73.0
Median :60.00	Median : 250.0	Median :38.00	Median :262000	Median :1.100	Median :137.0	Median :115.0
Mean :60.83	Mean : 581.8	Mean :38.08	Mean :263358	Mean :1.394	Mean :136.6	Mean :130.3
3rd Qu.:70.00	3rd Qu.: 582.0	3rd Qu.:45.00	3rd Qu.:303500	3rd Qu.:1.400	3rd Qu.:140.0	3rd Qu.:203.0
Max. :95.00	Max. :7861.0	Max. :80.00	Max. :850000	Max. :9.400	Max. :148.0	Max. :285.0

Tabella 1 Caratteristiche descrittive delle variabili

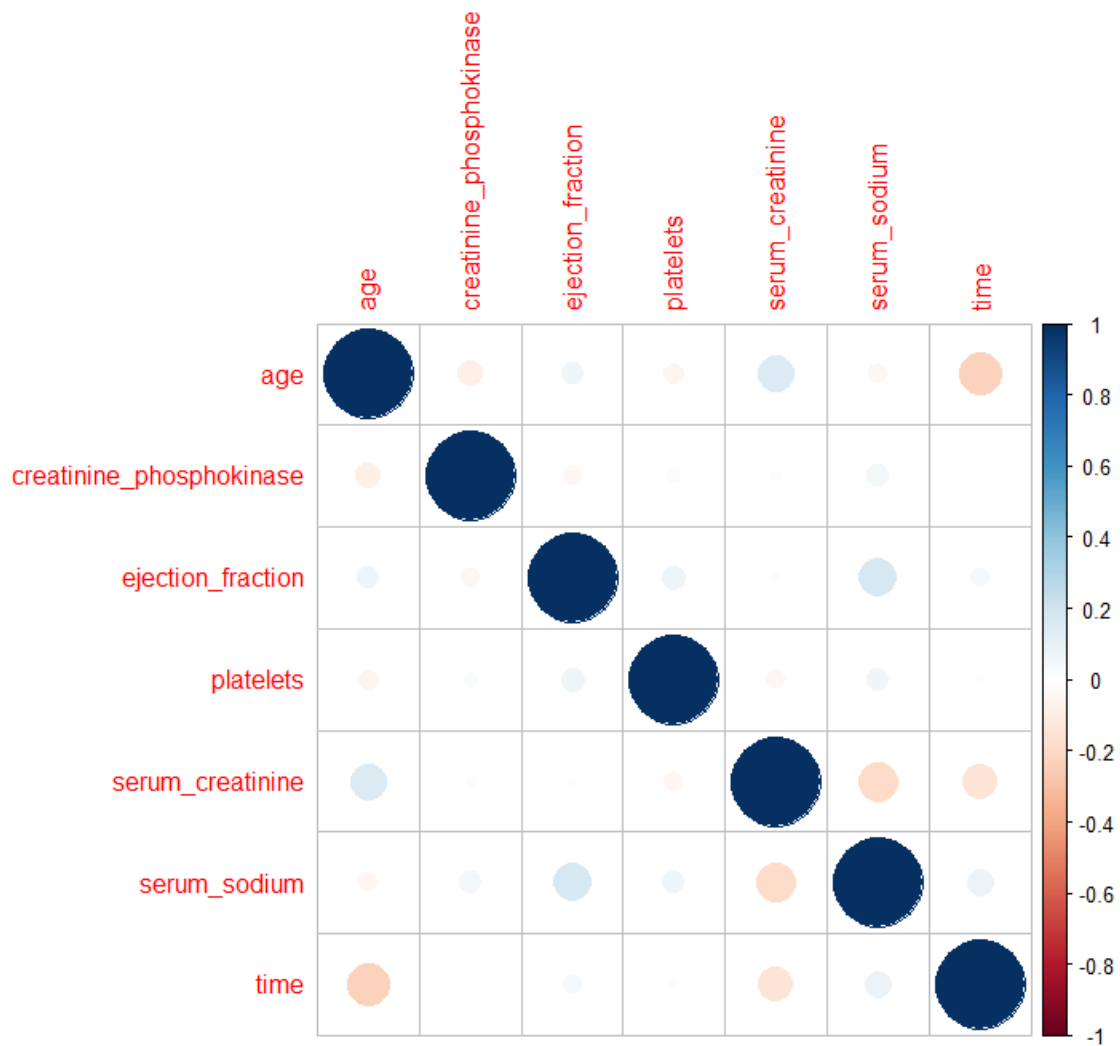


Figura 5

Dalla heatmap riportata è possibile dedurre che ci sono delle correlazioni sia positive sia negative. Quelle positive sono riconducibili alla coppia di variabili serum\_creatinine/ejection\_fraction; quelle negative invece sono age/time e serum\_sodium/serum\_creatinine. Alcune di queste correlazioni però potrebbero essere influenzate dalla presenza di outliers come si potrebbe pensare dalla figura nella pagina seguente, in cui sono rappresentate le singole misurazioni.

L'entità di questi valori però le rende molto poco influenti sui risultati successivi, come evidenziato dalla Tabella 2, in cui sono rappresentati numericamente i valori delle correlazioni.

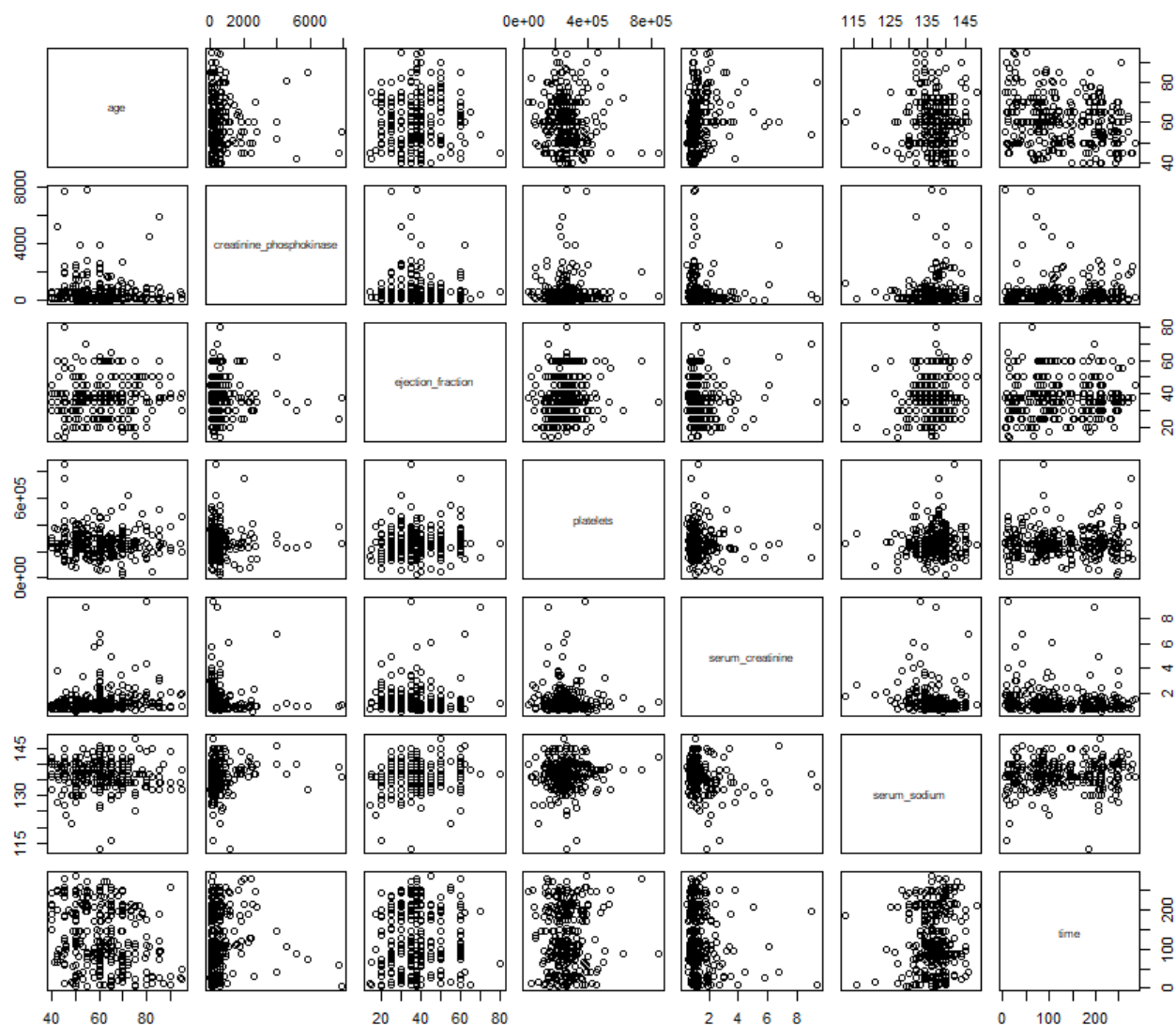


Figura 6 Rappresentazione grafica dei rapporti tra variabili

	age	creatinine_phosphokinase	ejection_fraction	platelets	serum_creatinine	serum_sodium	time
age	1	-0.08	0.06	-0.05	0.16	-0.05	-0.22
creatinine_phosphokinase	-0.08	1	-0.04	0.02	-0.02	0.06	-0.01
ejection_fraction	0.06	-0.04	1	0.07	-0.01	0.18	0.04
platelets	-0.05	0.02	0.07	1	-0.04	0.06	0.01
serum_creatinine	0.16	-0.02	-0.01	-0.04	1	-0.19	-0.15
serum_sodium	-0.05	0.06	0.18	0.06	-0.19	1	0.09
time	-0.22	-0.01	0.04	0.01	-0.15	0.09	1

Tabella 2 Valori della matrice di correlazione tra le variabili

Per analizzare le singole variabili sono stati calcolati i valori di varianza e deviazione standard (Tabella 3) per poter valutare la variabilità, in combinazione con box plot per esaminare la dispersione dei singoli valori; così è possibile individuare le variabili in grado di fornire un sufficiente spettro di valori in grado di creare un modello efficace.

	variance	std dev
<i>platelets</i>	9565668749.44888	97804.2368685983
<i>creatinine_phosphokinase</i>	941458.571457431	970.287880712436
<i>time</i>	6023.96527575139	77.6142079502934
<i>age</i>	141.486482907971	11.8948090740445
<i>ejection_fraction</i>	140.063455365761	11.8348407410392
<i>serum_sodium</i>	19.469955781015	4.41247728390923
<i>serum_creatinine</i>	1.07021107270319	1.03451006408985

Tabella 3 Valori di varianza e deviazione standard

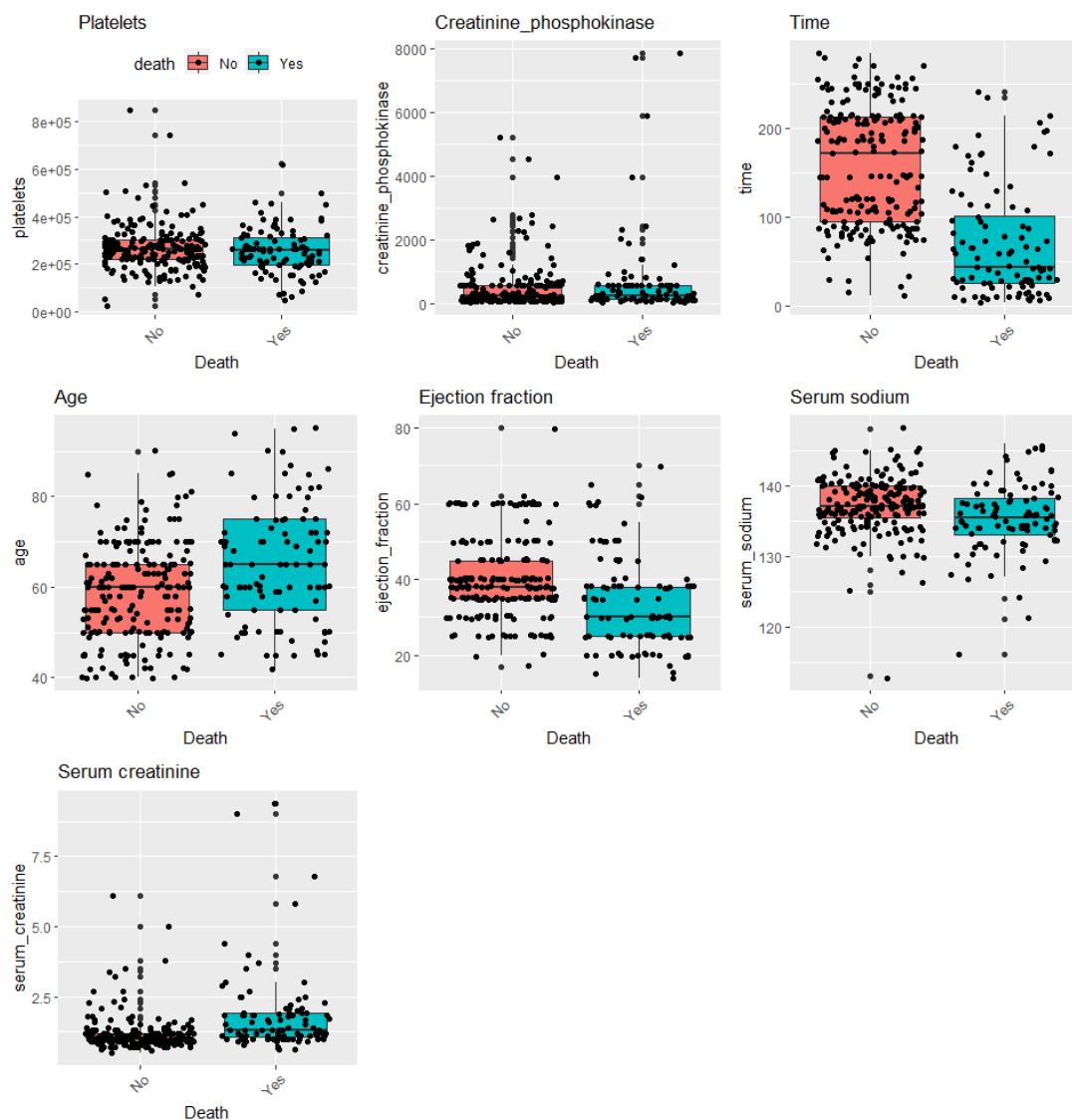


Figura 7 Boxplot per le variabili quantitative



Come è possibile notare dalla figura 7 alcune variabili hanno una varietà di valori così esigua da non essere considerabili come fattori determinanti per il modello che si andrà a sviluppare di seguito. Pertanto ho deciso di eliminare dall'insieme le seguenti variabili:

- Serum creatinine
- Serum sodium
- Creatinine phosphokinase

In aggiunta a queste anche il sesso non verrà considerato all'interno del modello in quanto vi è una quasi perfetta proporzione tra i casi in individui di sesso differente, l'unica differenza è sul numero di osservazioni (Uomini 65%, Donne 35%), come mostrato dalla figura 2.

## Analisi tramite tecniche di classificazione

Lo scopo di questa analisi è quello di ottenere un modello in grado di predire se un individuo sarà in grado di sopravvivere o no, per farlo sono state implementate diverse tecniche di classificazione che meglio si adattano al dataset impiegato.

Poiché il dataset a disposizione non contiene un numero molto elevato di osservazioni, è necessario ricorrere a tecniche di manipolazioni dei dati per poter ottenere una validazione efficace dei modelli.

La scelta per questo tipo di manipolazione è ricaduta sulla k-fold cross validation, che consiste nella suddivisione del dataset originale in k parti uguali in numero di osservazioni, iterando sulle k partizioni si ottiene che ad ogni passo dell'algoritmo la k-esima parte è il test set mentre le restanti parti rappresentano il training set (Figura 8). Al termine delle iterazioni viene calcolato un modello medio di cui viene valutata l'accuratezza. L'utilizzo di questa tecnica garantisce una valutazione più precisa del modello in quanto permette di evitare problemi di over/underfitting.

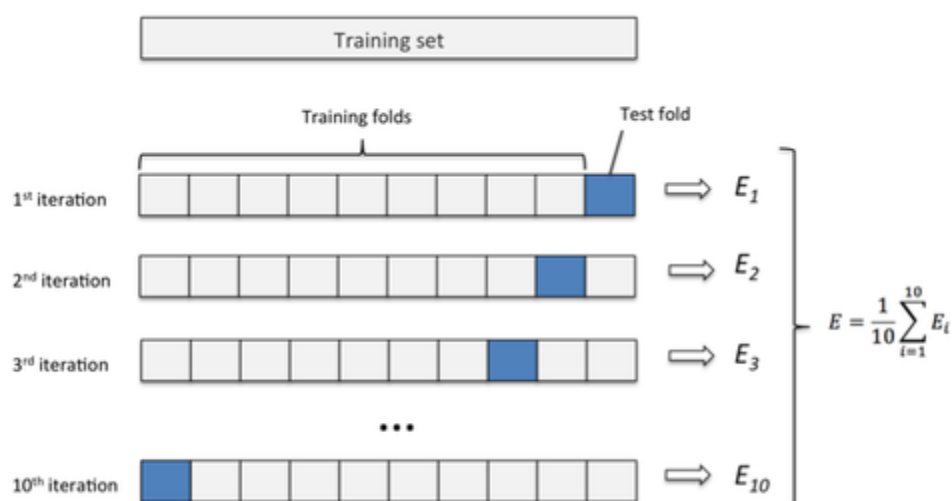


Figura 8 Rappresentazione dell'algoritmo di k-fold cross validation

Il dataset preso in considerazione in questo caso verrà suddiviso in due sezioni, una di testing e una di training. La prima costituirà il 30% del dataset in maniera casuale per ciascuna analisi.

Gli algoritmi di classificazione che verranno utilizzati saranno:

- Logistic regression
- Linear Discriminant Analysis
- Decision Tree
- Random Forest
- Support Vector Machine

Ciascun algoritmo verrà brevemente introdotto da alcune note teoriche per meglio comprendere l'analisi di un problema di classificazione.

### Logistic Regression

Nel caso di un problema a risposta binaria 0/1, l'utilizzo di modelli in grado di produrre uno spettro di valori infinito, come la regressione lineare, risulta inappropriato.

I modelli di regressione logistica si basano sulla formula logistica:

$$P(Y = 1) = \frac{e^{\beta_0 + \beta_1 X}}{1 + e^{\beta_0 + \beta_1 X}}$$

Dopo alcune manipolazioni si ottiene:

$$\frac{P(Y = 1)}{1 - P(Y = 1)} = e^{\beta_0 + \beta_1 X}$$

La quantità ottenuta nel membro sinistro dell'equazione viene definita *odds*. Questa è in grado, dato un valore compreso tra 0 e  $+\infty$ , di restituire un valore compreso tra 0 e 1.

Al fine di ottenere una funzione lineare in X, applicando il logaritmo naturale si ottiene la funzione *logit* o *log-odds*

$$\ln\left(\frac{P(Y = 1)}{1 - P(Y = 1)}\right) = \beta_0 + \beta_1 X$$

Nel caso specifico di questo dataset verrà considerato come esito negativo un valore  $\leq 0.5$ , con esito positivo altrimenti.

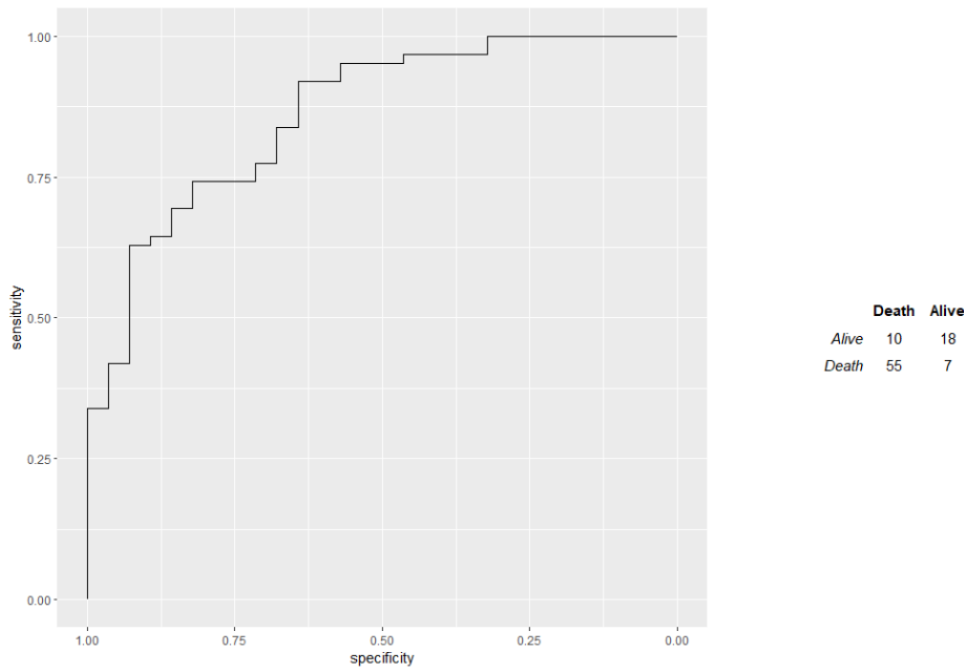


Figura 9

Come si può vedere dalla tabella che rappresenta la confusion matrix per il modello ottenuto, le osservazioni correttamente classificate per il test set sono l'81,1% con AUC pari a 0.86.

### Linear Discriminant Analysis

Il secondo metodo per poter modellare la probabilità analizzato è la Linear Discriminant Analysis. Questo approccio si basa sull'analisi separata di ciascun predittore  $X$  per ciascuna delle classi della variabile target, utilizzando il teorema di Bayes. Durante queste valutazioni si assume che le variabili abbiano distribuzione normale, di conseguenza il modello risulta molto vicino alla regressione logistica.

Per completezza viene riportato il teorema di Bayes:

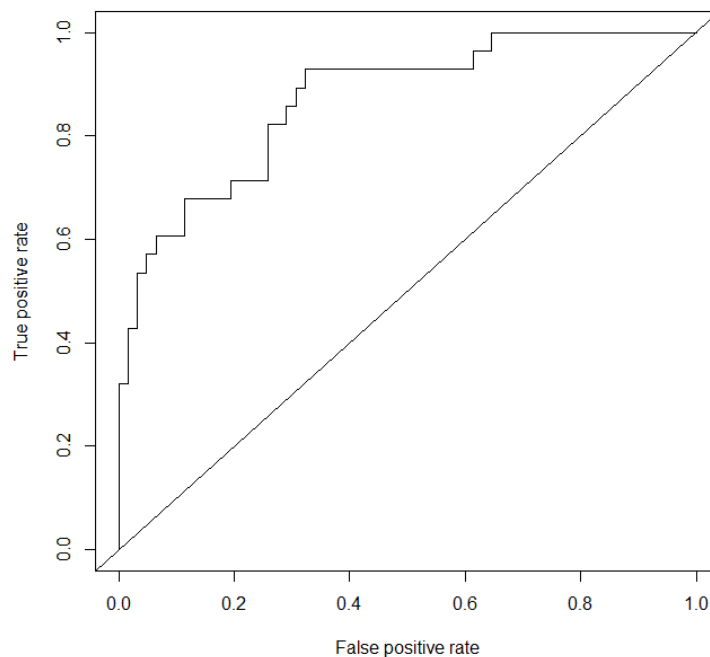
$$P(Y = k|X = x) = \frac{P(X = x|Y = k)}{P(X = x)}$$

da cui si ricava la forma che viene utilizzata all'interno de LDA:

$$P(Y = k|X = x) = \frac{\pi_k f_k(x)}{\sum_{l=0}^K \pi_l f_l(x)}$$

dove

- $f_k(x) = P(X = x|Y = k)$  corrisponde alla densità di  $X$  nella classe  $k$
- $\pi_k = P(Y = k)$  è la probabilità a priori per la classe  $k$



	Death	Alive
Death	55	7
Alive	9	19

Figura 10

Nella confusion matrix si può notare un buon grado di accuratezza pari al 82,2% e AUC di 0.87.

### Decision trees

L'algoritmo alla base del Decision Tree consiste nella classificazione di un dato sulla base di un albero decisionale a profondità finita. Infatti ogni nodo è un test che viene effettuato su una caratteristica del dataset, dal risultato si originano i rami discendenti. La misura dell'efficacia di test si possono utilizzare diverse misure, la più comune è il GINI index. Questo è in grado di valutare la purezza/ordine del dataset ad un determinato nodo. L'indice in questione è definito come:

$$GINI(t) = 1 - \sum_j [p(j|t)]^2$$

in cui  $p(j|t)$  rappresenta la frequenza relativa di  $j$  rispetto al nodo  $t$  considerato.

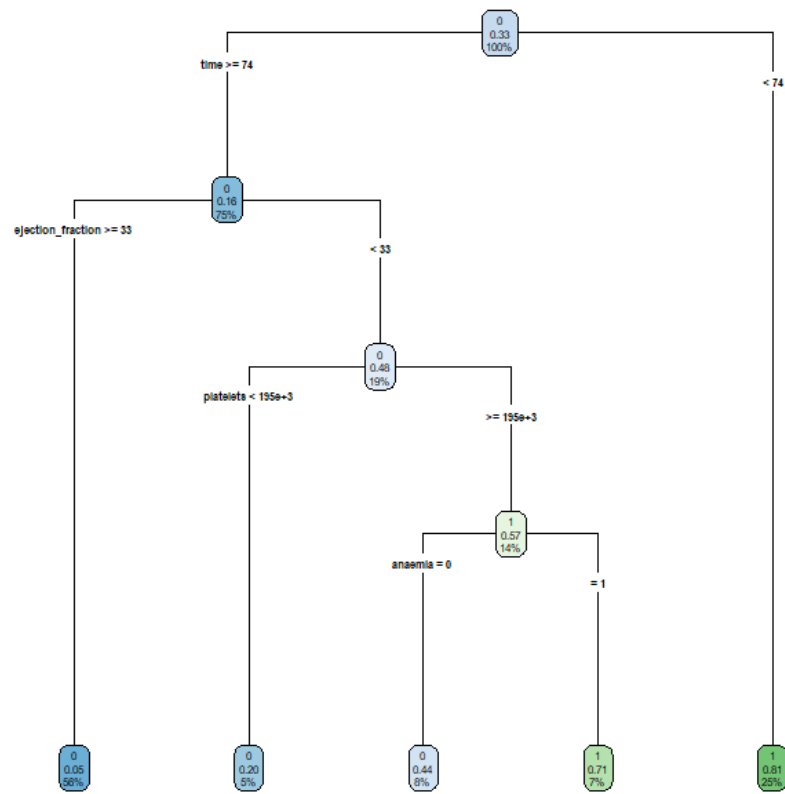


Figura 11 Rappresentazione grafica del decision tree considerato

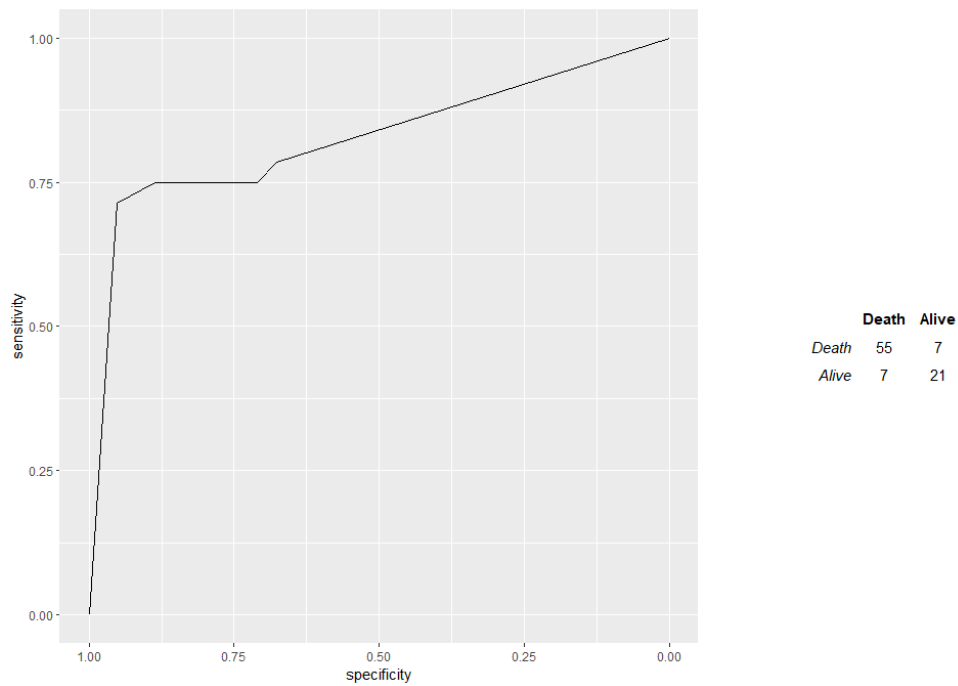


Figura 12 Risultati della classificazione tramite decision tree

I risultati presentanti nella figura 12 risultano essere in linea con le precedenti analisi e si può osservare come l'accuratezza sia del 84.4%, tuttavia l'AUC nonostante sia buona con un valore di 0.82 presenta una forma molto diversa rispetto alle precedenti poiché degrada molto velocemente per valori di sensibilità crescenti.

### Random forest

La random forest è una tecnica che si è sviluppata come estensione del più semplice decision tree. Infatti questo tipo di analisi è nato per migliorare l'efficienza dei decision tree.

L'algoritmo si basa su una tecnica per ridurre la varianza di un metodo statistico, il Bagging. Questo afferma che date  $N$  osservazioni indipendenti con varianza  $\sigma^2$  la loro varianza è data da  $\sigma^2/n$ . Questa assunzione è derivata dal teorema del limite centrale. Da questo deriva che la media delle osservazioni riduce la varianza.

L'algoritmo di generazione si articola nella selezione di un sotto insieme casuale di variabili e la creazione di un decision tree. L'operazione si ripete per un numero di alberi definito, si procede quindi al calcolo del modello medio.

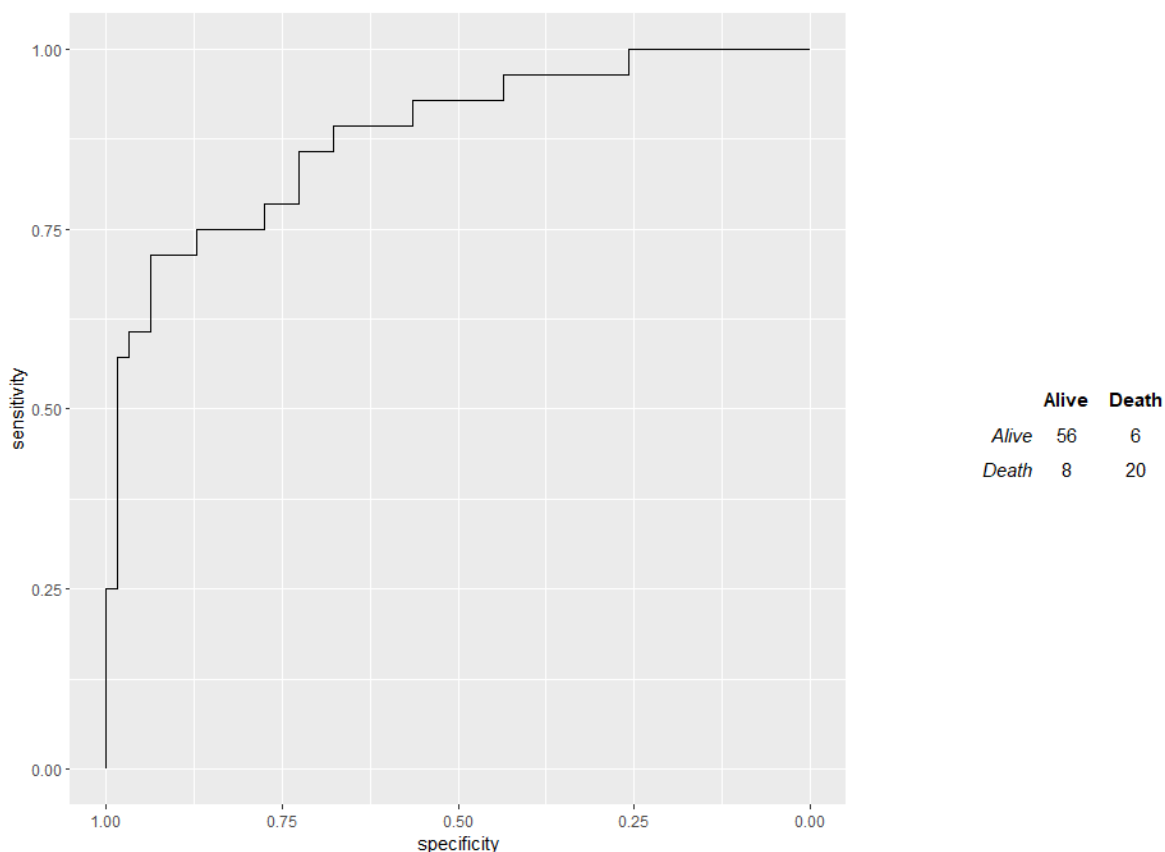


Figura 13 Risultati analisi con random forest

Come si può vedere dalla confusion matrix i risultati sono molto simili a quelli ottenuti precedentemente con un decision tree, in particolare l'accuratezza risulta la stessa 84.4%. La differenza è molto più marcata per quanto riguarda AUC che in questo caso è di 0.88. Si può quindi concludere che la complessità introdotta dalla random forest garantisce una maggiore stabilità nel modello.

## Support Vector Machine

Le tecniche basate su Support Vector Machine utilizzano come principio fondante la costruzione di macchine di apprendimento in grado di generare degli iperpiani per separare i punti nel migliore modo possibile.

Nel caso di una classificazione binaria, come quella presa in considerazione in questo elaborato, si otterranno due valori possibili per le funzioni  $f(X)$ .

L'obiettivo per il modello sarà quello di selezionare tra gli infiniti iperpiani possibili quello che massimizza la separazione tra le classi delle osservazioni, che si può modellizzare come un modello di massimizzazione.

$$\text{Max}(\beta_0, \beta_1, \dots, \beta_p, M) \text{ con } \sum_{j=1}^p \beta_j^2 = 1$$

$$y_i(\beta_0 + \beta_1 x_{i1} + \dots + \beta_p x_{ip}) \geq M \quad \forall i = 1, \dots, n$$

Tale problema può essere risolto tramite moltiplicatori di Lagrange, tuttavia il risultato sarebbe troppo rumoroso perciò si è optato per utilizzare una struttura di supporto in grado di creare un margine sufficientemente ampio così da poter considerare anche i punti all'interno dell'area delimitata dal Support Vector.

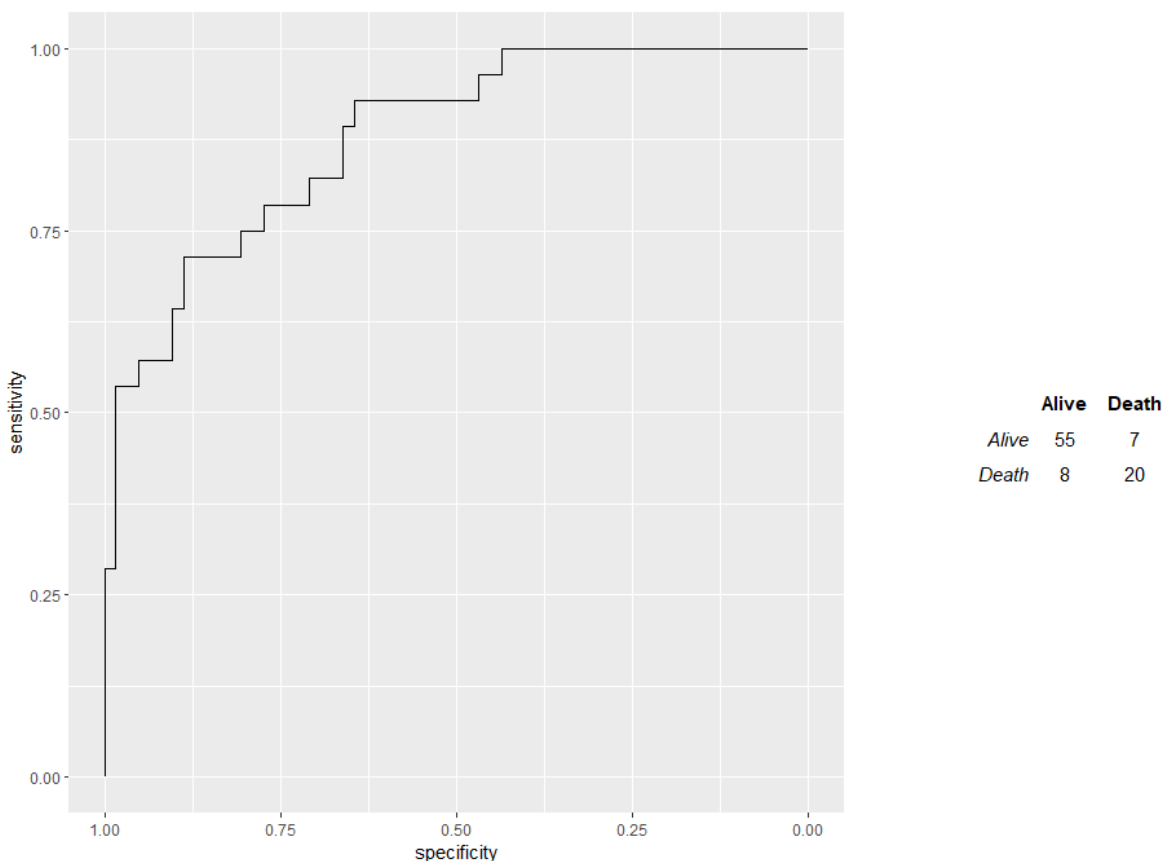


Figura 14 Risultati con modello basato su Support Vector Machine

I risultati dati dal modello in figura 14 mostrano un modello con una buona accuratezza pari al 83.3%, interessante è AUC che mostra una buona resa del modello con un valore di 0.87.

## Conclusioni

Modello	Accuratezza	AUC
Logistic regression	81.1%	0.86
Linear Discriminant Analysis	82.2%	0.87
Decision trees	84.4%	0.82
Random Forest	84.4%	0.88
Support Vector Machine	83.3%	0.87

A seguito delle scelte fatte in fase preliminare sul dataset, sono stati riportati le statistiche per ciascun modello analizzato.

Si può notare come i risultati ottenuti dai modelli basati su alberi abbiano generato delle percentuali di accuratezza migliori, a fronte di un impiego computazionale leggermente superiore. Il decision tree però ha mostrato una leggera tendenza al deterioramento delle prestazioni con errori nella classificazione rispetto alla random forest.

I modelli lineari hanno dimostrato di essere molto efficaci, ma meno precisi nella predizione dei valori. Rimangono però molto affidabili e poco influenzabili come dimostrano i valori di AUC.

La support vector machine mostra un risultato bilanciato, con una buona affidabilità e un'accuratezza nella media.

In generale si può concludere che i modelli analizzati siano stati in grado di ottenere risultati buoni mantenendo sempre un'accuratezza superiore al 80%.