
UNIVERSITÀ DEGLI STUDI DI PADOVA

**Dipartimento di Matematica "Tullio Levi-Civita"**
**Corso di Laurea magistrale in Matematica**

TESI DI LAUREA

# *Analisi di Metodi e Modelli di apprendimento per l'identificazione della malaria*

*Relatore:*
*Chiar.mo Prof.*
*Francesco Rinaldi*

*Candidato:*
*Genny Covolo*
*Matr.: 1179765*

**Anno Accademico 2018-2019**
**13 Dicembre 2019**

# Indice

# Introduzione

L'Organizzazione mondiale della sanità (Oms) in [1] ha riportato che globalmente nel 2017 il numero di casi totali di malaria si aggira attorno a 219 milioni, con circa 435 mila decessi. Tuttavia, il 92% delle persone colpite da questa infezione vengono contagiate nella Regione africana e, più precisamente, l'80% dei decessi mondiali si registrano in appena 16 Paesi dell'Africa subsahariana e in India. Naturalmente sono le fasce più deboli della popolazione, come bambini e donne in gravidanza, che sono più vulnerabili. Si è calcolato che nel 2017 il 61% degli individui deceduti sono stati bambini sotto i 5 anni. É pur vero che numerosi progressi si sono fatti negli anni, tanto che si sono registrati ben 20 milioni di casi in meno nel 2017 rispetto al 2010 ma, allo stesso tempo, appare ancora lontana l'eradicazione di tale malattia. Non sembra infatti attualmente scontato raggiungere l'obiettivo stabilito nel 2015 dalla World Health Assembly di ridurre l'incidenza della malaria del 90% entro il 2030.
Per tutte queste ragioni ogni anno vengono stanziati ingenti somme di denaro (circa 3,1 miliardi di dollari nel 2017) sia per adottare misure di prevenzione sia per permettere ai ricercatori di sviluppare nuovi farmaci e vaccini.
In questo quadro anche gli strumenti matematici e, in particolare, alcune tecniche di Machine Learning possono risultare utili per analizzare dati relativi a soggetti che hanno contratto l'infezione e aiutare così i medici ad avere informazioni aggiuntive per combattere tale malattia.
A questa categoria appartiene anche il lavoro che sarà descritto nelle prossime pagine. Infatti, si è partiti da un dataset fornito dall'Istituto Nazionale per le Malattie Infettive Lazzaro Spallanzani-IRCCS-Roma relativo a circa 26000 bambini di età compresa tra 0 e 15 anni che hanno contratto l'infezione malarica in 6 differenti zone dell'Africa subsahariana e si sono utilizzate tecniche di Machine Learning per raggiungere differenti scopi. In prima istanza, invero, si è tentato di costruire dei modelli che potessero predire l'esito della malattia dividendo in due diverse classi i pazienti. In seguito si sono sfruttati degli algoritmi per poter classificare le informazioni a disposizioni dalla più importante alla meno importante. Tale fase definita *feature ranking/selection* si è rivelata indispensabile sia per migliorare la capacità predittiva dei modelli precedentemente creati sia per capire quali sono gli esami clinici a cui i medici dovrebbero prestare maggiore attenzione. Si è ripetuto questa fase anche dividendo i pazienti per luogo di provenienza in modo da osservare eventuali differenze tra i vari ospedali e tentare di comprendere per quale ragione i tassi di mortalità sono molto più elevati in alcuni posti. Infine, ci si è concentrati

solo nei pazienti deceduti e, tramite algoritmi di apprendimento non supervisionato, si è tentato di dividere i campioni in gruppi per capire se pazienti ricoverati nello stesso ospedale sono simili e, perciò, collocabili nello stesso sottoinsieme. Tali risultati sono riportati e approfonditi nel terzo capitolo di questa tesi.
Tuttavia, prima di esporre quanto si è ottenuto a partire dai dati a disposizione, si è ritenuto utile dedicare il primo capitolo per introdurre brevemente la malaria dal punto vista medico. Inoltre, si sono presentati anche alcuni studi che hanno utilizzato tecniche di Machine Learning per migliorare la comprensione dei meccanismi che permettono la diffusione della malaria e per sviluppare nuovi metodi di diagnosi di tale infezione.
Il secondo capitolo, invece, è dedicato all'analisi degli algoritmi e dei metodi che sono stati utilizzati per esaminare e modellizzare le caratteristiche dei pazienti malarici di cui sono stati raccolti le relative generalità e gli esami clinici effettuati in ospedale. In aggiunta, in questa sezione si invita il lettore a riflettere su alcune questioni quali l'importanza di preparare in modo adeguato i dati prima di applicare alcun tipo di tecnica di Machine Learning e l'importanza di scegliere il metodo più adatto in relazione alle peculiarità del problema che si sta affrontando e agli scopi che si vuole raggiungere.
Nella parte conclusiva della trattazione ci si è, infine, limitati a riassumere quanto emerso con questo lavoro mettendo in luce gli aspetti più salienti.

# Capitolo 1

# Malaria: la malattia e le ricerche con algoritmi di Machine Learning

In questa sezione si vuole riflettere sulle caratteristiche che fanno sì che la malaria, come accennato nell' introduzione, causi un così alto numero di morti ogni anno. In particolare, si vuole tentare di spiegare brevemente come avviene la trasmissione dei parassiti che danno origine all'infezione, quali sono le zone più colpite e come è possibile trattare le persone infettate. Si dedica, infine, l'ultima parte del capitolo a discutere brevemente vari studi che, applicando algoritmi di Machine Learning, hanno tentato di migliorare la tempestività con cui intervenire sul singolo paziente e di comprendere come è possibile trovare nuovi vaccini e cure.

## 1.1 Patogenesi e Eziologia della Malaria

La malaria è una malattia protozoaria che si trasmette tramite la puntura di una zanzara ed è provocata da *plasmodi*, parassiti che si insediano nei globuli rossi fino a distruggerli. Tuttavia, di oltre 3000 specie di zanzare al mondo circa 460 appartengono al genere *Anopheles* e di queste solo alcune decine ospitano nel proprio intestino il parassita della malaria. Più precisamente, una zanzara diventa infetta solo se un gametocita maschile e un gametocita femminile, ossia le forme sessuate del parassita della malaria, sono ingeriti dalle femmine di zanzara del genere *Anopheles*. A questo punto i parassiti si vanno a collocare nell'intestino dell'insetto per formare uno zigote che, in seguito, penetra le pareti intestinali per dividersi in spore, le componenti che effettivamente provocano la malaria nell'essere umano.
Quasi tutti i casi di malaria nell'uomo sono causati, in realtà, dalle specie *P. ovale*, *P. malariae*, *P. vivax*, *P. falciparum* e, nel sud est asiatico, *P. knowlesi*, anche noto come il parassita della malaria delle scimmie. Statisticamente i casi di decessi si verificano quasi sempre a causa di *P. falciparum* anche se, di tanti in tanto, *P. vivax* e *P. knowlesi* possono risultare altrettanto mortali.
L'infezione inizia quando una zanzara *Anopheles* femmina, pungendo una persona, rilascia tramite le ghiandole salivari gli sporozoi del plasmodio, ossia i parassiti che si sono

riprodotti con formazione di spore. Queste ultime sono trasportate per via ematica fino al fegato dove entrano nelle cellule del parenchima epatico, cioè il tessuto del fegato. I parassiti iniziano così una fase di riproduzione assessuata dove, grazie ad un processo di amplificazione (*merogonia*), un singolo sporozoita origina più di $10000 - 30000$ merozoiti, ossia il prodotto di tale processo. Sono, quindi, i merozoiti a riversarsi nel circolo sanguigno e ad insediarsi all'interno dei globuli rossi riproducendosi dalle 6 alle 20 volte ogni 2 giorni circa (*ciclo eritrocitario*) e divenendo *trofozoiti*. Nel contempo alcuni parassiti evolveranno anche verso forme sessuate in grado di trasmettere la malattia. Quando i parassiti raggiungono la densità di circa 50 $\mu g/L$ di sangue comincia la fase sintomatica dell'infezione. Inoltre, in *P. vivax* e in *P. ovale* alcuni parassiti non si riproducono subito ma rimangono latenti provocando recidive nel corso degli anni. Con il passare del tempo, infine, il parassita arriva a consumare due terzi dell'emoglobina del globulo rosso infettato e, a questo punto, il globulo rosso si rompe rilasciando dai 6 ai 30 merozoiti. É così che nuove cellule saranno infettate. Naturalmente la durata di ciascuna fase sopra descritta e la velocità di riproduzione dei parassiti cambia da specie a specie e determina anche una differente risposta nel corpo del soggetto colpito. Di seguito si è, quindi, introdotta una tabella riassuntiva che mette a confronto 4 delle specie prima citate.

| **Caratteristiche** | *P. falciparum* | *P. ovale* | *P. vivax* | *P.malariae* |
|---|---|---|---|---|
| **Durata della fase *merogonia*** | 5,5 | 9 | 8 | 15 |
| **Numero di merozoiti rilasciati** | 30000 | 15000 | 10000 | 15000 |
| **Durata del *ciclo eritrocitario*** | 48 | 50 | 48 | 72 |
| **Recidive** | No | Si | Si | No |

Come si nota anche dalla tabella la specie *P. falciparum* produce un numero maggiore di merozoiti per globulo rosso infettato rispetto alle altre e questa determina anche la sua maggiore virulenza.
É importante sottolineare che la trasmissione della malaria non avviene per contagio interumano diretto ma può accadere in seguito ad una trasfusione di sangue o di globuli rossi di soggetti malarici o con plasmodi nella fase infettante. Di conseguenza, dal momento che il periodo che intercorre tra il contagio e la comparsa dei primi sintomi va dai $7 - 14$ giorni per *P. falciparum* fino ad arrivare addirittura a $7 - 30$ giorni per *P.malariae*, è estremamente importante appurare che il donatore non sia stato sottoposto a rischi di contagio. Per tale ragione in Italia esistono delle norme di legge che obbligano coloro che si sono recati in zone malariche ad astenersi dalla donazione del sangue per un certo periodo di tempo.
Inoltre, è importante sottolineare che un soggetto che ha contratto *P. falciparum*, se punto, può contagiare una zanzara a distanza di 1 anno mentre nel caso di *P. vivax* si arriva addirittura a 2 anni e a 3 nel caso di *P.malariae*. Le zanzare, invece, una volta contratto il parassita, possono infettare l'uomo per tutta la durata della loro vita.

## 1.2 Epidemiologia

L'epidemologia della malaria è piuttosto complicata perché può variare considerevolmente anche nell'ambito di piccole zone geografiche. In generale, la malaria è presente in quasi tutta la fascia tropicale del globo anche se le singole specie sono concentrate in specifiche zone. *P. falciparum* si trova in Africa, Haiti, Repubblica Dominicana e Guinea, *P. vivax* è più diffusa in America centrale mentre in Sud America, Asia orientale e Oceania coesistono entrambe le specie in ugual proporzione. *P. ovale* è diffusa praticamente solo in Africa mentre la *P.malariae* è maggiormente presente nell'Africa subsahariana ma è meno comune delle altre specie. Infine, *P. knowlesi* si trova in Borneo e nel Sudest Asiatico.
Sulla base dell'entità della *parassitemia* (quantità media dei parassiti in rapporto a un $\mu L$ di sangue) o la *splenomegalia* (aumento patologico del volume della milza) dei bambini tra i 2 e i 9 anni è stata definita l'endemicità della malaria. Si è scelto, dunque, di fare questa distinzione:

- zone ipoendemiche $\rightarrow$ prevalenza $< 10\%$
- zone mesoendemiche $\rightarrow$ prevalenza compresa tra 11% e 50%
- zone iperendemiche $\rightarrow$ prevalenza tra 51% e 75%
- zone oloendemiche $\rightarrow$ prevalenza $> 75\%$

Tale distinzione è utile per registrare nel corso del tempo i miglioramenti, soprattutto nelle aree più colpite. Per avere un riscontro visivo della diffusione della malaria, invece, è altrettanto significativo osservare la mappa creata dalla WHO e riportata in Figura (1.1) in cui sono rappresentati i casi di malaria registrati nel 2017 per Paese.

Nelle regioni iperendematiche e oloendematiche accade che le persone possano venire punte più volte al giorno da zanzare infette e, di conseguenza, si possano ammalare ripetutamente nel corso della vita. Ciò spiega perché la mortalità dovuta alla malaria è molto alta soprattutto tra i bambini mentre negli adulti tale infezione è quasi sempre asintomatica in quanto raggiungono l'immunità. Questo, al contrario non avviene nelle aree ipoendemiche dove la trasmissione è più bassa e, in questo caso, tutte le fasce di età sono a rischio. Inoltre, si registrano degli aumenti di casi di malaria in relazione ad alcuni periodi dell'anno come il periodo delle piogge. In generale la grande diffusione della malaria è dovuta a tre fattori:

1. **numero delle zanzare**: più di 100 specie di zanzare *Anopheles* possono trasmettere la malaria e solitamente hanno una buona efficienza come vettori;
2. **tendenza a pungere l'essere umano**;
3. **longevità delle zanzare**: il parassita necessita di almeno 7 giorni per far sì che la zanzara possa infettare l'uomo nel momento in cui lo punge. Di conseguenza, più è lunga la vita media di una determinata specie di zanzare più persone potenzialmente potranno essere infettate.

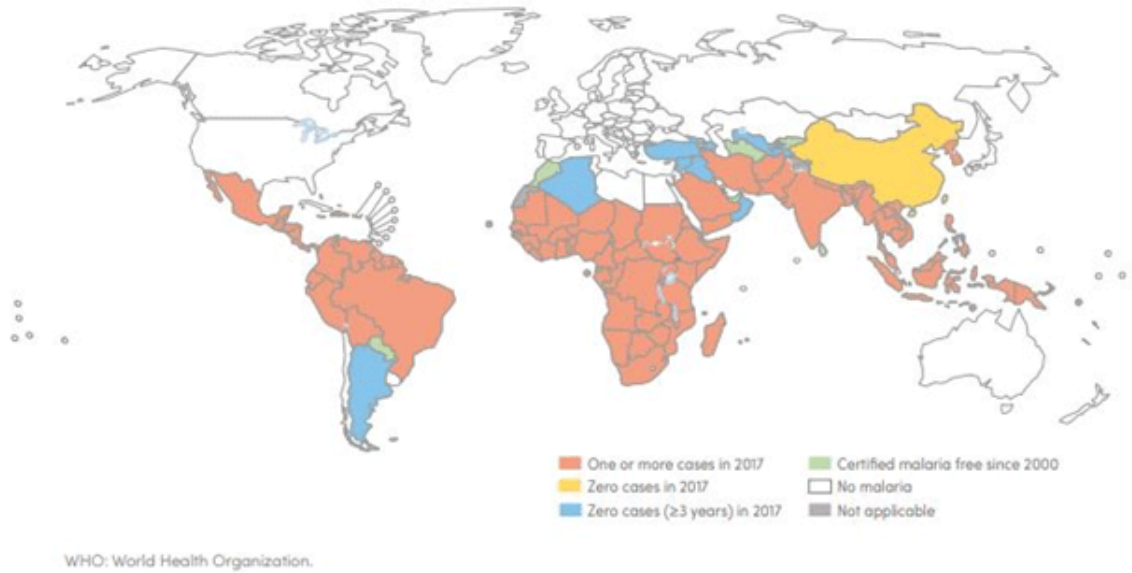



Figura 1.1: Casi di malaria del 2017 in ciascun Paese

Alla luce di quanto detto, si è stimato che la trasmissione della malaria è direttamente proporzionale al numero di zanzare, al quadrato del numero di punture al giorno per zanzara e alla decima potenza della probabilità che la zanzara viva per un giorno.

## 1.3 Manifestazioni cliniche

La risposta iniziale del corpo umano all'infezione malarica è rappresentata dall'attivazione di sistemi di difesa non specifici. La milza può rimuovere i parassiti dai globuli rossi infetti e rimetterli in circolo. I parassiti che, invece, sfuggono alla milza, rompendo il globulo rosso in cui sono ospiti, liberano delle sostanze che inducono l'attivazione dei macrofagi. I meccanismi di difesa non specifici hanno l'effetto di bloccare l'espansione dell'infezione che, successivamente, la risposta immunologica specifica dovrebbe mettere sotto controllo.
Per quanto riguarda i sintomi, inizialmente la malaria può essere confusa come una delle malattie virali minori con malessere generale, spossatezza, dolori muscolari, cefalea, problemi intestinali e febbre. Nel caso in cui puntate di febbre, freddo e brividi si succedano ad intervalli regolari probabilmente è in atto un'infezione da *P. vivax* o *P. ovale*. Negli altri casi la febbre è solitamente irregolare, può elevarsi anche al di sopra di 40°C ed è associata a tacchicardia e delirium. Inoltre, l'infezione da *P. falciparum* frequentemente nei bambini provoca convulsioni generalizzate che possono preannunciare lo sviluppo di

encefalopatie. É comune anche l'anemia, soprattutto nei bambini più piccoli, e splenomegalia. Negli adulti, infine, a volte si osserva un leggero ittero.
Tranne *P. falciparum*, la malaria non complicata da altre patologie ha un tasso di mortalità minore del $0,1\%$ ma, in presenza di più del $2\%$ di globuli rossi infettati, la mortalità cresce vertiginosamente. Se nella malaria da *P. falciparum* insorge il coma i tassi di mortalità salgono al $20\%$ per gli adulti e al $15\%$ per i bambini e si parla in questo caso di malaria celebrale. Altre complicazioni che si riscontrano nei casi più gravi sono l'ipoglicemia, causata da un maggior consumo di glucosio sia del paziente sia dei parassiti malarici e l'acidosi, provocati dall'accumolo di acidi organici. Gli adulti affetti da malaria grave possono sviluppare anche edemi polmonari e insufficienza renale acuta. Infine, l'infezione da HIV e la malnutrizione predispongono il soggetto a sviluppare complicazioni spesso mortali.

## 1.4 Diagnosi e trattamento della malaria

Quando una persona che vive o proviene da un'area endemica presenta febbre e i tipici sintomi della malaria dovrebbe essere sottoposto ad analisi cliniche per confermare la diagnosi. Per comprovare o meno tale sospetto ci sono differenti modalità.
Il metodo più tradizionale è quello di prelevare del sangue periferico dal paziente e creare uno striscio sottile a partire da qualche goccia di tessuto ematico. A questo punto con apposite colorazioni è possibile contare i parassiti e i globuli bianchi. Questo metodo è rapido e poco costoso ma va ripetuto più volte se l'esito è negativo in modo da sancire definitivamente che l'infezione in corso è di altra natura.
In alternativa è possibile utilizzare il metodo a goccia spessa in cui, anziché creare uno strato sottile di sangue, lo si distribuisce in uno spessore. É tuttavia una strategia che richiede una buona esperienza da parte di chi effettua la valutazione e, se l'esito è negativo, è necessario ripetere l'esame con altri campioni.
Una modalità meno utilizzata ma molto precisa è tramite PCR (Polymerase Chain Reaction). In questo caso è possibile identificare i parassiti della malaria anche se sono a livelli molto bassi e, inoltre, si possono distinguere le varie specie. Tale tecnica solitamente non è disponibile nelle zone rurali perché richiede manutenzione dei macchinari.
In ogni caso è necessario valutare con molta cautela il dato relativo alla parassitemia. Invero, i pazienti con più di $10^5$ parassiti per microlitro presentano un rischio di mortalità maggiore ma persone semimmuni possono sopportare livelli più alti e, al contrario, i pazienti non immuni rischiano la morte con valori anche molto più bassi.
Nel caso di *P. falciparum* è possibile sfruttare anche uno *stick-test* che individua degli anticorpi specifici ed è stato pensato per le aree in cui è più difficile effettuare esami di laboratorio per mancanza delle strutture adeguate. Tuttavia, presenta delle criticità in quanto non è possibile diagnosticare la malaria provocata da altre specie di parassiti e non si può capire se la malattia è ancora in corso o è stata contratta nelle settimane precedenti. Questi antigeni, infatti, rimangono nel sangue anche dopo la conclusione dello stato infiammatorio.
Le terapie per il trattamento di tale infezione sono molteplici e variano in base alla con-

dizione clinica del paziente, alla tipologia di farmaci che si dispone nella regione in cui si risiede, alla specie di parassita da cui è stato infettato e dalla presenza o meno di resistenza ai farmaci da parte di alcuni parassiti.
Più in particolare, se il soggetto è in grado di assumere farmaci per via orale e il parassita responsabile dell'infezione non è *P. falciparum* l'Organizzazione Mondiale della Sanità consiglia di utilizzare una combinazione di farmaci a base di *artemisinina*. Nel caso in cui, invece, il paziente sia in condizioni più gravi si possono iniettare per via endovenosa o per via intramuscolare dei derivati idrosolubili dell'*artemisinina*. Oltre a trattare i pazienti con i farmaci di cui si è appena parlato, coloro che sviluppano la malaria grave devono anche essere sottoposti ad un continuo monitoraggio da parte del personale medico e possono aver bisogno di farmaci aggiuntivi per gestire le complicazioni della malaria. Ad esempio, la glicemia del soggetto non cosciente andrebbe controllata ogni $4-6$ ore e la funzionalità renale monitorata ogni giorno.
Un altro problema che i medici si trovano ad affrontare sempre più spesso è la resistenza di alcuni ceppi di *P. falciparum* all'*artemisinina* soprattutto in Cambogia e in Birmania. In questo caso i tempi di guarigione sono spesso molto più lunghi dei 3 giorni, periodo standard in cui la terapia dovrebbe fare effetto, ed è necessario utilizzare altri tipi di farmaci. In ogni caso, indipendentemente da dove si è contratta l'infezione, se il livello di parassitemia non scende di almeno il 25% rispetto al valore iniziale entro le prime 48 ore o se la parassitemia non torna a livelli prossimi a 0 entro una settimana è molto probabile che il ceppo sia farmaco-resistente. Di conseguenza, è necessario cambiare la terapia al più presto e far fronte alle eventuali complicazioni.

## 1.5 Prevenzione

La ricerca per trovare un modo per eradicare la malaria è sostenuta da numerosi organizzazioni quali il *National Istitute of Allergy and Infectious Disease*, le *CDC* e l'*Organizzazione Mondiale della Sanità*. Tuttavia, tale obiettivo non sembra raggiungibile in un futuro prossimo per una molteplicità di motivi. Prima di tutto le zanzare *Anopheles* hanno una diffusa distribuzione dei siti di riproduzione e il numero di persone infettate è molto elevato. Inoltre, l'uso di farmaci antimalarici inefficaci e, in particolar modo, l'espansione di farmaci antimalarici contraffatti che si vendono nelle farmacie dell'Asia dell' Est e dell'Africa rende ancora più difficile sconfiggere tale malattia. Infine, l'inadeguatezza delle infrastrutture e delle risorse, soprattutto nelle aree più povere, rende impossibile far avere le cure necessarie a molte persone favorendo l'espansione dell'infezione.
Nel tempo, però, si è potuto appurare che semplici strumenti di prevenzione possono ridurre drasticamente i casi di malaria. Ad esempio, la distribuzione di zanzariere impregnate di insetticidi a lunga durata ha portato in Africa ad una riduzione del 20% dei decessi in età infantile. Inoltre, anche l'utilizzo di insetticidi spray e l'applicazione nelle parti del corpo esposte di repellenti contenenti DEET o picaridina hanno portato ad ingenti risultati. Tutto ciò è stato possibile grazie anche all'intervento del *Fondo Globale per la lotta contro HIV/AIDS, Tubercolosi e Malaria*, dell'*UNICEF* e altre organizzazioni dal momento che gli abitanti delle regioni endemiche spesso non potrebbero permettersi

l'acquisto di tali beni. Infine, un'altra strategia tra quelle più semplici da mettere in atto è invitare la popolazione a non esporsi a zanzare nelle ore più pericolose, ossia dal tramonto all'alba, anche se è ragionevole pensare che non sia sempre possibile rispettare tale regola.
Contestualmente a queste azioni basilari i ricercatori stanno cercando di sviluppare nuovi farmaci e vaccini. Da sottolineare è l'approvazione da parte dell'EMA (*European Medicines Agency*) del vaccino RTS che è risultato efficace nel $30 - 60\%$ dei bambini africani che hanno preso parte allo studio. Tuttavia, si è notato che la copertura scende al 16% solo dopo 4 anni e, di conseguenza, è necessaro rifarlo. Un altro modo per ridurre la mortalità della malaria è la profilassi antimalarica che prevede di fare assumere periodicamente alle fasce più deboli della popolazione, come bambini e donne in gravidanza, dei farmaci antimalarici.
L'utilizzo di tali medicinali è consigliato anche per i viaggiatori che sono consapevoli di recarsi in zone endemiche. In particolare, è importante che siano a conoscenza dei rischi e della specie di parassiti diffusi nella destinazione finale. La profilassi antimalarica dovrebbe iniziare da 2 giorni a 2 settimane prima della partenza in modo da garantire i livelli ematici antimalarici adeguati. I farmaci che di solito sono utilizzati sono:

- **atovaquone-proguanil**: da prendere una volta al giorno, è efficace contro tutti i tipi di ceppi inclusa *P. falciparum* multiresistenti. É tollerata sia da bambini sia dagli adulti e solitamente non provoca effetti collaterali se non qualche disturbo gastrointestinale;

- **meflochina**: anche in questo caso è un farmaco che risulta efficace contro tutti i tipi di parassiti e che va assunto una volta a settimana. Gli effetti collaterali sono nausea, vertigini, disturbi del sonno e malessere. Tuttavia, a differenza del precedente, questo può essere utilizzato anche in gravidanza.

- **clorochina**: non è efficace per *P. falciparum* ma funziona per le altre specie. Solitamente il farmaco è ben tollerato anche se in qualche caso crea cefalee.

Infine, è importante sottolineare che nel caso in cui durante il viaggio si accusassero presunti sintomi della malaria è estremamente importante recarsi in una struttura ospedaliera in modo, se necessario, da poter avere una tempestiva diagnosi ed iniziare le terapie quanto prima.

## 1.6 Machine Learning e Malaria

Finora quello che si è tentato di fare è introdurre la malaria dal punto di vista medico in modo da permettere di capire i meccanismi che fanno sì che tale malattia sia così diffusa e virulenta. Si è anche accenato al fatto che ci sono molte organizzazioni che offrono ingenti somme per permettere ai ricercatori di produrre farmaci e vaccini sempre più efficienti.
Dal momento, però, che spesso si raccolgono grandi dataset in cui si riportano le analisi cliniche di numerosi pazienti si è rivelato estremamente utile sfruttare modelli matematici

per analizzare questi dati e comprendere quali relazioni ci sono tra l'esito della malattia e i sintomi riscontrati nei pazienti. Di seguito, quindi, si descriveranno alcuni lavori che hanno sfruttato tecniche di Machine Learning per studiare la malaria o migliorare gli strumenti a disposizione dei medici.

### 1.6.1 Malaria severa e clustering

Nell' articolo [4] del 2018 si sono descritti i risultati ottenuti a partire da un campione di 2915 pazienti facendo clustering, cioè dividendo il dataset in sottogruppi (*cluster*). Così facendo è possibile trovare gli individui con caratteristiche simili perchè apparterranno allo stesso gruppo. Per maggiori informazioni su tale tipologia di algoritmi si veda, in ogni caso, la Sezione 3.25.
I pazienti analizzati sono tutti bambini provenienti dal Gambia ammessi in ospedale per malaria severa causata da *P. falciparum* che, come visto precedentemente, statisticamente è il parassita più pericoloso per l'essere umano.
Prima di tutto i dati sono stati normalizzati e sono state individuate le variabili più importanti per migliorare la capacità predittiva dell'algoritmo che si è successivamente utilizzato. Si è osservato così che solo 13 variabili su 46 sono decisive per predire l'esito della malattia. Infine, tra queste 13 solo 8 sono state scelte per calcolare la distanza tra i vari pazienti e dividerli, quindi, in cluster. In particolare le 8 variabili scelte si possono raggruppare in 3 gruppi:

1. ***riduzione della funzionalità cerebrale***: di questa categoria fa parte il *Blantyre coma score* (si veda 3.1.2 per ulteriori spiegazioni), *convulsioni durante l'ammissione, sonno anomalo* e *rigidità muscolare*;

2. ***riduzione della funzionalità respiratoria***: a tale classe appartengono *respiro affannoso*, *recessione intercostale* e *uso di muscoli aggiuntivi durante la respirazione*;

3. ***anemia***: misura la concentrazione di *emoglobina*.

A questo punto è stata utilizzato una funzione di densità con kernel Gaussiano per misurare la distanza tra i pazienti. Si è, infatti, sfruttata la funzione

$$D_x = C \sum_y e^{\frac{d(x,y)^2}{\sigma}}$$

dove $x$ indica il paziente in questione, $y$ sono gli altri pazienti, $d(x, y)$ sta ad indicare la distanza euclidea e $\sigma$ è una costante fissata a 0.2. In seguito la distribuzione dei pazienti è stata rappresentato in una mappa dove le aree più dense indicano insiemi di pazienti con caratteristiche cliniche molto simili. Quello che si è visto è che nelle aree con meno concentrazione si trovavano pazienti con tutte e 3 le classi di sintomi sopra elencate mentre l'area più densa è occupata da soggetti affetti da anemia ma non dalle altre problematiche prima citate. Infine, si è notato che i decessi sono più elevati nelle zone in cui la densità è bassa.

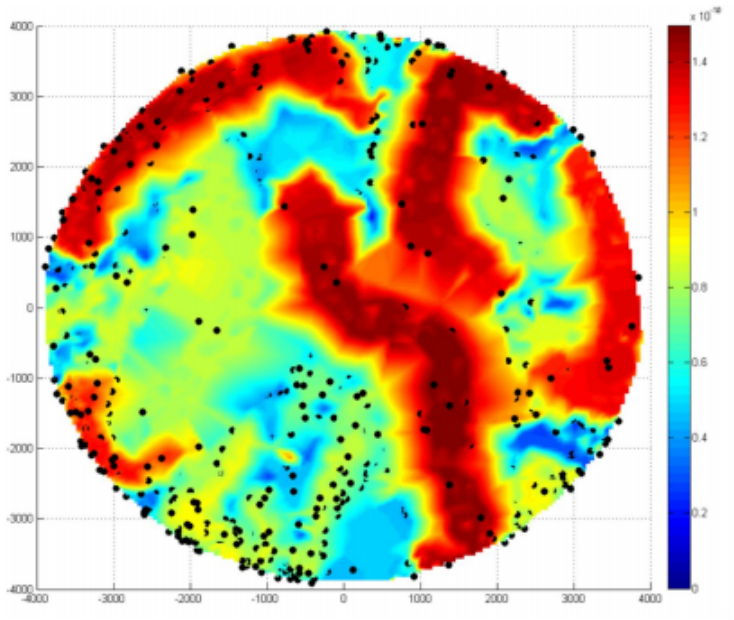



Figura 1.2: É qui raffigurata la mappa del calore in cui le zone con maggiore densità indicano un maggior numero di soggetto con valori simili.

Per poter analizzare la distribuzione dei dati è stata stabilita una distanza soglia. In questo modo sono stati individuati 238 cluster ma solo 19 contenenti più di 20 pazienti con una percentuale di mortalità molto variabile tra lo 0% e il 53%. Quello che è emerso è che questa disparità di mortalità non è casuale in quanto, se si mescolano randomicamente tra i 19 gruppi i casi in cui l'esito della malattia è negativo, non si trovano differenze percentuali così elevate tra i gruppi.
In aggiunta, per comparare i cluster con la distribuzione predetta dalla *WHO (World Health Organization)* si sono divisi i pazienti in 4 categorie: malaria cerebrale, difficoltà respiratoria, grave anemia o una combinazione delle 3 complicazioni appena descritte. Quello che si è visto è che i cluster sono piuttosto concordi con i sintomi che, secondo la WHO, determinano la malaria severa. Unica questione che ha destato qualche perplessità è il fatto che i pazienti con difficoltà respiratorie non si trovino tutti nello stesso cluster. Successive analisi hanno, tuttavia, dimostrato che questa divisione si spiega forse con il fatto che il proteoma presente tra i i vari pazienti è differente.
Infine, si sono fatti alcuni studi per capire quali altre variabili fossero implicate per determinare queste 4 categorie. Ad esempio, si è notato che l'ingrossamento del fegato è spesso associato a grave anemia e si è valutato i vantaggi in termini di riduzione di mortalità dei pazienti sottoposti a trasfusione. Non meno importante è stato comprendere che l'insufficienza cardiaca in bambini affetti da malaria severa è strettamente correlata con anemia e con difficoltà respiratoria.
Questo studio ha, dunque, dimostrato che dividere i pazienti in cluster è estremamente

utile per raccogliere nuovi informazioni cliniche. É stato interessante osservare anche che coloro che risultano isolati dagli altri sono più a rischio di morte e questo è ragionevolmente spiegabile con il fatto che, se ci si discosta dal caso standard, è più difficile per i medici comprendere quali siano le misure di intervento più efficaci. Allo stesso tempo, tuttavia, ulteriori approfondimenti si potrebbero fare a partire da questi presupposti. Nella parte finale dello stesso articolo, invero, si suggerisce che sarebbe utile provare a rifare il tutto selezionando più variabili o cambiando il metodo per scegliere quelle più importanti.

### 1.6.2 Proteine dei Plasmodium e SVM

Nel 2009 è stato pubblicato sulla rivista scientifica *Genetica* l'articolo [5] in cui si sfruttano le Support Vector Machine (si veda la Sezione 2.5 per approfondimenti ) per affrontare il problema relativo all'aumento della farmaco-resistenza da parte di alcuni ceppi. Si sono, infatti, studiate le proteasi, enzimi che permettono di scomporre le proteine in gruppo amminico e gruppo carbossilico, di tre completi o quasi completi genomi di *P. falciparum*, *P. vivax* e *P. yoelii yoelii*, responsabile della malaria nei roditori. Questa scelta si spiega con il fatto che le sequenze dei genomi presentano ancora delle lacune e una buona conoscenza delle proteasi permette di sviluppare degli inibitori delle stesse. É quindi possibile, a partire da queste informazioni, costruire nuovi vaccini e farmaci che, inibendo gli enzimi, non permettono ai parassiti di completare il loro ciclo vitale. Infatti, in questo modo si riesce ad impedire il processo di digestione dell'emoglobina, essenziale come fonte nutritiva per i parassiti.
Il problema di classificazione binaria che si è affrontato è determinare se una sequenza generica di proteine appartiene o non appartiene ad una data famiglia. L'apprendimento di SVM avviene, quindi, analizzando sia proteine di cui non si conosce l'eventuale utilità per i parassiti sia quelle di cui si conosce la funzione assunta per favorire la diffusione dell'infezione nell'essere umano. Per determinare quanto due proteine sono simili si sono rappresentate in uno spazio vettoriale di dimensioni pari alle somma delle lunghezze delle $k$ sottosequenze di amminoacidi. Per permettere l'apprendimento di SVM è stato utilizzato il dataset MEROPS in cui sono presenti sia le unità peptidiche sia i loro inibitori. Sono così state aggiunte alle proteasi che si sapevano appartenere a *P. falciparum*, *P. vivax* e *P. yoelii* altre 1208 proteasi. Queste ultime sono state scelte dopo un'accurata analisi in cui si è cercato di predire la loro funzionalità biologica, i processi cellulari ad esse legati e la sede in cui sono collocate queste proteasi nelle cellule.
Quello che lo studio ha scoperto è che ci sono 28 possibili proteasi in *P. falciparum*, 45 in *P. vivax* e 19 in *P. yoelii yoelii* che non erano riportati nel dataset utilizzato né era noto che appartenessero a queste specie di parassiti.
Oltre ad utilizzare SVM è stato ripetuto il tutto anche con *PSI Blast (Position-Specific Itereted Blast)*, tecnica che permette ricerche di omologia molto accurate facendo successive iterazioni. Tale modo di procedere, seppur più comune per la ricerca di proteasi, si è rilevato in realtà peggiore di quello precedente. Infatti, si è osservato che con PSI Blast sono stati maggiori i casi di falso positivo. In particolare, il caso in cui SVM è stato nettamente migliore di PSI Blast è quello per i parassiti di *P. falciparum*. Questo

si spiega forse con il fatto che *P. falciparum*, essendo la forma più agressiva di parassita, è anche quella più studiata e quella in cui il genoma è più noto. Ciò sicuramente migliora le capacità predittive di SVM.
Inoltre, confrontando l'abbondanza delle proteasi nelle varie specie è emerso che i parassiti che causano la malaria nell'uomo sono più complessi rispetto a quelli dei roditori e questo si riflette su una maggiore quantità di proteasi differenti.
Tra le varie proteasi individuate quelle appartenenti a due particolari famiglie sembrano le migliori come punto di partenza per poter costruire nuovi vaccini in quanto sono implicate nella rottura degli eritrociti e nella digestione dell'emoglobina. Tuttavia, visto la difficoltà e la vastità del campo di ricerca si è convinti che ulteriori studi possono far emergere nuove proteasi altrettanto efficaci per l'identificazione di nuove cure e vaccini per la malaria.

### 1.6.3 Migliorare l'identificazione della malaria con tecniche di machine learning

Le tecniche di Machine Learning possono essere sfruttate anche per sviluppare metodi per facilitare le analisi volte a confermare la diagnosi di malaria. Infatti, se si potesse automatizzare il processo di diagnosi i pazienti potrebbero avere una risposta immediata ed iniziare le cure prima possibile. Inoltre, nelle aree rurali non sono presenti né strutture specializzate né personale in grado di analizzare i campioni di sangue per accertare la presenza del parassita nel sangue con le attuali metodologie.
Come emerge dall'articolo [6], per automatizzare il metodo numerosi studi hanno tentato di sviluppare differenti approcci per poter riconoscere i globuli rossi infetti a partire da un'immagine al microscopio. In questo modo, non solo si velocizzerebbero le analisi ma sarebbe possibili ottenere un valore più accurato rispetto all'occhio umano.
Seppure le strategie proposte siano tra le più svariate, tutte seguono i seguenti passi:

- in primo luogo è necessario acquisire le immagini in formato digitale. Questa procedura dipende dall'approccio iniziale che si è sfruttato. In particolare, attualmente il metodo più utilizzato per diagnosticare la malaria è ancora il classico microscopio perché permette di calcolare sia il numero di globuli rossi infettati sia il tipo di batterio presente. Tuttavia, alcune ricerche sono state fatte supponendo di partire da immagini ottenuto con un microscopio digitale.

- Successivamente bisogna utilizzare metodi per eliminare il rumore e migliorare i colori dell'immagine in modo da favorire l'identificazione dei globuli rossi infettati. Solitamente, in questa fase, un metodo non è sufficiente ma si ottengono risultati validi adottando un insieme di tecniche.

- Il passo successivo è quello atto ad identificare e sottolineare i singoli globuli rossi od eventualmente altre componenti del sangue a cui si è interessati.

- A volte si preferisce applicare un algoritmo per selezionare le feature. Le variabili più comuni che si trovano in questi tipi di studi sono il colore, la forma e l'aspetto

dell'interno dei globuli rossi che cambiano in modo significativo se infettati. Generalmente si selezionano le variabili più importanti nel caso in cui la mole di dati da analizzare è molto grande e l'algoritmo che si intende utilizzare richiederebbe troppo tempo per produrre un risultato.

- La fase finale prevede di applicare un metodo matematico per poter classificare i campioni nelle differenti classi di interesse. Solitamente lo scopo più usuale che si vuole raggiungere è dividere i globuli rossi infettati da quelli sani. Le tecniche che si sono sfruttate sono tra le più svariate, e si sono tentati sia approcci con algoritmi di apprendimento supervisionato come reti neurali o alberi di decisioni, sia tecniche di apprendimento non supervisionato come clustering.

In conclusione, quello che si può affermare è che sicuramente questi studi sono un buon punto di partenza per arrivare un giorno ad automatizzare il processo di diagnosi della malaria. Tuttavia, attualmente e per la mancanza di uniformità delle tecniche scelte e per le insufficienti verifiche su un insieme test non si è ancora trovata una modalità sicura per sostituire le classiche tecniche per la diagnosi di malaria.

# Capitolo 2

# Alcune tecniche di Machine Learning

## 2.1 Descrizione del problema

Negli ultimi anni il Machine Learning è divenuto uno strumento indispensabile in un ampio numero di settori, da quello finanziario e di marketing alla biologia e medicina. Ma che cosa si intende per Machine Learning?

**Definizione.** *Il **Machine Learning** è una branca dell'intelligenza artificiale che, a partire da un insieme di dati, permette ad un sistema di apprendere.*

La capacità di disporre di uno strumento in grado di saper interpretare un insieme di informazioni è indispensabile soprattutto quando la mole di dati raccolti ha grandi dimensioni e analizzarli senza particolare tecniche sarebbe impensabile. Quello, dunque, che è possibile fare utilizzando queste tecniche è costruire un *modello*, ossia una specifica struttura in grado di descrivere il fenomeno e, se si desidera, di predire il comportamento di nuovi campioni.
Per poter ottenere quanto appena descritto è, però, fondamentale comprendere appieno il problema ed adottare le strategie giuste. Idealmente, dunque, è necessario suddividere il lavoro nelle seguenti fasi:

1. analizzare il problema reale che si vuole affrontare;
2. costruire il modello matematico;
3. analizzare il modello matematico;
4. scegliere un algoritmo che risolva il modello;
5. verificare con dei nuovi campioni che il modello sia valido.

Tutti gli algoritmi di Machine Learning, seppur possano essere di diverse tipologie, necessitano di una collezione di dati di partenza chiamato *dataset*.

**Definizione.** *Si definisce* ***dataset*** *una tabella o una matrice in cui ciascuna colonna delle $N$ presenti corrisponde ad una variabile o* feature *e ogni riga rappresenta un campione $c_j = (x_j, y_j)$ con $x_j = (x_1, \ldots, x_{N-1})$ e $j = 1, \ldots, M$ dove $M$ è il numero di righe; $x_j$ è l'****input*** *mentre $y_j$ rappresenta l'output o* ***classe*** *e non è sempre noto a priori.*

Gli algoritmi di Machine Learning si possono suddividere in 4 gruppi.

- **Apprendimento supervisionato**: questi algoritmi, a partire da un insieme stabilito di dati (*training set*) appartenenti a differenti classi, cercano di capire la correlazione tra le feature e di predire il comportamento dei campioni di un eventuale nuovo dataset. Più in particolare, se $c_j = (x_j, y_j)$ con $j = 1, \ldots, M$ sono i campioni del dataset di partenza allora esisterà una funzione ideale $f : X \to Y$ tale che $f(x_j) = y_j$ dove $X$ è lo spazio di tutti i campioni possibili e $Y$ è l'insieme di tutti i possibili output. La difficoltà è che non si conosce $f$ se non in un finito numero di punti. Quello che con l'apprendimento supervisionato è possibile fare è trovare una funzione $\bar{f} : X \to Y$ che approssimi meglio possibile $f$ individuando lo stesso output di $f$ per la maggior parte dei campioni. Nel caso più semplice il problema che si vuole affrontare è di natura binaria e quindi $Y = \{1, 0\}$ ma si possono affrontare anche casi più complessi dove le classi da determinare sono più numerose.
  In generale se $Y$ contiene un numero finito di valori si parla di algoritmi di *classificazione*. Se, al contrario, i valori in $Y$ sono continui, allora quello che si andrà ad affrontare è un problema di *regressione*.
  Infine, altro aspetto a cui si deve prestare attenzione quando si sceglie di utilizzare questa tipologia di algoritmi è l'*overfitting*. Questo termine sta ad indicare che la funzione $\bar{f}$ che si è individuata è estremamente precisa per i punti del dataset iniziale ma non è in grado di predire in modo altrettanto corretto il comportamento di un nuovo dataset utilizzato come test. Il caso opposto è, invece, l'*underfitting* in cui si trova una funzione $\bar{f}$ che mal riproduce gli output del dataset di partenza.

- **Apprendimento non supervisionato**: è vantaggioso utilizzare algoritmi di questo tipo se si dispone di grandi dataset di cui non si conosce la classificazioni dei singoli campioni. In questo caso, dunque, gli algoritmi divideranno i dati in gruppo sulla base delle somiglianze e delle differenze. Si giunge così a determinare un preciso numero di classi e, in un certo senso, ci si riconduce al caso precedente dove gli output erano noti.

- **Reinforcement learning**: questi tipi di algoritmi creano il modello sulla base del comportamento che hanno i dati durante la loro analisi. Più precisamente, il sistema apprende attraverso prove ed errori. Infatti, una serie di decisioni corrette portano ad un rafforzamento del processo mentre un errore spinge l'algoritmo a ritornare al passo precedente. Ad esempio, una comune applicazione di queste tecniche si trovano in robotica in quanto se si vuole insegnare delle azioni ad un robot è più efficace procedere per tentativi e permettergli, quindi, di sbagliare un determinato gesto fino a che non si trova il giusto modo per compierlo.

- **Neural Network**: semplificando di molto la questione si può affermare che le reti neurali sono dei modelli matematici ispirati al funzionamento del cervello umano, la cui struttura è composta da neuroni. Gli algoritmi di apprendimento profondo o deep learning, invece, non sono altro che delle reti neurali messe in successione a formare tre o più strati. Più il problema da affrontare è complesso più strati saranno necessari. Tuttavia, solitamente la maggior parte degli strati rimane nascosta in quanto esclusivamente il primo e l'ultimo sono visibili. Spesso, inoltre, il deep learning risulta una tecnica piuttosto potente perché permette di combinare nei differenti strati algoritmi di apprendimento supervisionato e non.

Da questo elenco di metodi, dunque, si può evincere che le tecniche possibili per ottenere una modellazione predittiva sono tra le più svariate e sta al ricercatore decidere quale metodologia risulta più adeguata in base allo scopo che si vuole raggiungere. Tuttavia, in tutti i casi, ciò che risulta importante è avere un buon dataset da utilizzare come punto di partenza o per testare i risultati prodotti da un determinato algoritmo. Di seguito, quindi, si approfondiranno le caratteristiche che dovrebbero essere rispettate per poter applicare le tecniche di Machine Learning e ottenere degli output soddisfacenti.

## 2.2 Analisi del Dataset

Come si è già accennato il primo passo da fare quando si vuole affrontare un problema sfruttando le tecniche di Machine Learning è analizzarlo per capire quali obiettivi si vuole raggiungere. Chiarire, infatti, qual è lo scopo finale permette di preparare il dataset in modo adeguato e definire quali feature potrebbero risultare di maggior interesse. In particolare, le questioni che si devono affrontare nella preparazione di un dataset sono le seguenti:

- **Tipologia di variabili**: i dati di cui si dispone possono avere feature di diversa natura. Dal punto di vista quantitativo si possono, infatti, avere valori discreti o continui mentre sotto l'aspetto qualitativo è necessario distinguere le variabili categoriche, ossia quelle in cui non è possibile determinare una gerarchia, da quelle ordinali. Essere consapevoli di ciò è importante perché solo così si possono interpretare correttamente i risultati.

- **Omogeneità dei dati**: spesso quando si inizia a costruire un dataset è necessario mettere insieme dati raccolti in differenti tabelle e da persone diverse. Questo aumenta il rischio di disporre di informazioni selezionate in modo non standardizzato sia per i personali metodo che un individuo può preferire sia perché i macchinari a disposizione possono essere svariati. Nel caso in cui non si possa discrimare tale tipologie di errori sarà necessario eliminare dei dati oppure accettare di avere una minor accuratezza nei risultati finali.

- **Noise data**: il rumore è un errore casuale o un valore che non è previsto nel range scelto per identificare i casi ammessi. Quest'ultimi sono anche detti *outliers*. Quando, invece, si hanno errori casuali ma non si conosce l'intervallo di ammissibilità i

modi per individuare ed eliminare tali elementi di disturbo sono molteplici. Il caso più semplice da identificare è quello in cui il formato non è conforme con gli altri. Ad esempio, se si desidera un numero decimale e si hanno, invece, alcune stringhe sarà opportuno convertire il tutto se possibile o, altrimenti, eliminarli. Se si vuole fare un controllo sui dati il metodo più semplice, nel caso di valori numerici, è quello di fare la media e calcolare la deviazione standard. A questo punto sarà possibile decidere se quanto ottenuto può essere verosimile o se, nei casi in cui la divergenza è molto grande, si è di fronte ad un errore randomico.

- **Missing values**: è molto comune che durante l'analisi dei dati si notino dei valori mancanti o perché semplicemente non sono disponibili o perché non pertinenti. A questo punto bisogna affrontare due questioni. In primo luogo, si deve decidere se vale la pena mantenere quel determinato campione anche se presenta dei valori non disponibili per una o più feature. Questa è una scelta che dipende dal numero di dati a disposizione e agli scopi che ci si prefigura. Naturalmente se un dataset è molto grande e i dati mancanti in rapporto sono pochi sarà più vantaggioso eliminarli direttamente. Nel caso in cui però non ci si trovi in tale situazione è necessario individuare un modo alternativo per integrarli. Per fare ciò sono disponibili strategie anche molto sofisticate ma spesso è sufficiente ricorrere alla media o alla moda. In ogni caso non esiste una tecnica perfetta per risolvere questo problema e, quindi, è importante valutare caso per caso il metodo migliore da adottare.

Infine è di fondamentale importanza, una volta completati i passaggi appena illustrati, standardizzare il dataset con la formula

$$z = \frac{x - \mu}{\sigma}$$

dove $x$ è il valore di partenza, $\mu$ è la media e $\sigma$ la deviazione standard.
Così facendo, invero, si eliminano due problematiche. In primo luogo si rendono confrontabili le varie feature perché riscalando il dataset non ci si espone al rischio che quelle con i valori più alti vengano mal interpretate dall'algoritmo e considerate più importanti. Inoltre, il secondo aspetto che avvalora l'importanza di standardizzare il dataset è legato alla velocità con cui gli algoritmi forniscono l'output. Tendenzialmente, infatti, se i valori in esame non sono omogenei i tempi di calcolo sono molto più lunghi.

## 2.3 Dataset sbilanciati

Molti problemi del mondo reale danno vita ad un numero disomogeneo di esempi appartenenti alle singole classi considerate. Tuttavia, si definisce *dataset sbilanciato* se c'è una significativa sproporzione del numero di esempi relativi a ciascuna classe. In altre parole, dunque, il dataset è sbilanciato quando una classe è rappresentata in numero molto minore rispetto alle altre. Di conseguenza, si può essere in questo caso non solo quando si è in presenza di due classi ma anche di multi-classi in cui una o più categorie è poco rappresentata. Di seguito, tuttavia, si è scelto di concentrarsi maggiormente nei casi in

cui ci sono solo 2 classi dal momento che nelle altre casistiche è sufficiente generalizzare quanto si dirà per la situazione binaria.
Innanzitutto, ci si soffermi per qualche istante a capire le ragioni che rendono svantaggioso avere un dataset sbilanciato. Il primo problema che si può individuare è che spesso la classe meno rappresentata è quella di maggiore interesse dal punto di vista applicativo. Tuttavia, quando si andrà ad applicare, ad esempio, un algoritmo di classificazione, anche se si otterrà un buon valore di accuratezza, la classe più rappresentata avrà una buonissima accuratezza mentre la capacità predittiva di quella meno comune potrà essere prossima a 0 in alcuni casi.
Inoltre, solitamente gli algoritmi di apprendimento tendono a dare maggiore importanza alle proprietà più rappresentate rischiando di tralasciare eventuali caratteristiche specifiche della classe minore. Tirando le somme, quindi, se il dataset è sbilanciato:

1. l'accuratezza non è più significativa in quanto non risulta valida per la classe meno rappresentata;

2. è necessario costruire classificatori basati anche sulla classe minore.

Nel corso del tempo sono stati sviluppati differenti metodo per risolvere tale problema. In alcuni casi sono gli algoritmi di apprendimento stessi a dare la possibilità di ribilanciare il dataset oppure è possibile associare peso maggiore alla classe meno rappresentata. Ciò nonostante, le modalità più utilizzate sono forse i *sampling method*, ossia metodi di campionamento che permettono di modificare il dataset di partenza in modo da riequilibrarlo. Il vantaggio maggiore di tali metodi è che sono indipendenti dall'algoritmo che si vuole utilizzare in seguito e, quindi, estremamente efficaci quando si intende lavorare con gli stessi dati per lungo tempo.
A loro volta i *sampling method* sono numerosi e diversi fra loro; in particolare i metodi di ricampionamento si possono dividere in tre grandi classi:

- **Undersampling method**: consentono di creare un sottoinsieme dei dati iniziali che risulta ribilanciato in quanto sono stati eliminati alcuni campioni della classe maggiormente presente.

- **Oversampling method**: permettono di creare un dataset più grande in cui sono state aggiunte delle copie della classe minore oppure sono state create delle nuove istanze.

- **Hybrid method**: sono approcci misti in cui vengono sfruttati entrambi i metodi precedentemente citati.

Dopo questa classificazione è naturale chiedersi quali delle tipologie descritte sia meglio adottare per avere una capacità predittiva migliore possibile. É, dunque, ragionevole soffermarsi ad analizzare i pro e i contro dei primi due metodi.

### PRO degli Undersampling Method

- Gli *undersampling method*, eliminando spesso anche molti dati, riducono i tempi per ottenere l'output desiderato soprattutto quando l'insieme dei dati di partenza è molto grande. Al contrario, nel caso degli *oversampling method*, se si ha già un dataset di partenza abbastanza vasto, i tempi di esecuzione degli algoritmi possono diventare esponenzialmente alti.
- A differenza degli *oversampling method*, gli *undersampling method* non rischiano di andare in overfitting dal momento che non producono nessun nuovo punto.

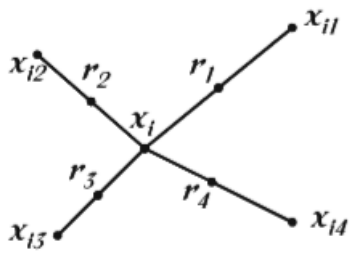
**PRO degli Oversampling Method**

- Al contrario degli *undersampling method*, questi metodi non fanno perdere alcuna informazione.
- Solitamente i metodi di *oversampling method* sono più performanti dei metodi di *undersampling*.

Di seguito si approfondiranno alcuni degli algoritmi più utilizzati e si tenterà di spiegare il loro funzionamento.

### 2.3.1 SMOTE

Il termine *SMOTE* [8] è l'abbreviazione di *Synthetic Minority Oversampling TEchnique* ed è un algoritmo che appartiene agli *Oversampling method*. Attualmente questo metodo è forse il più utilizzato perché, anziché replicare semplicemente i campioni dellla classe meno rappresentata, ne introduce di nuovo. Questi ultimi sono creati dall'interpolazione di vari elementi presenti tra la classe che si vuole riprodurre.




Come si vede nella figura sovrastante per selezionare una nuovo punto $x_i$ che apparterrà alla classe minore si sfrutta una *funzione distanza*, ossia una funzione che tiene conto della posizione assunta dai vari punti del dataset. Tale funzione, infatti, viene utilizzata per definire una distanza metrica che permette di selezionare un numero $n$ di elementi dal *training set* che appartengono alla classe minore e che tramite un'interpolazione randomica andranno a definire il nuovo punto.
La procedura appena descritta sommariamente per selezionare i nuovi punti si traduce nell'algoritmo che si introdurrà di seguito.

**Algoritmo 1.** *FUNZIONE PER GENERARE I NUOVI PUNTI*

*Sia $N$ un numero intero che descrive il numero di ricampionamenti necessari affinché la distribuzione delle due classi sia pari a $1:1$. L'algoritmo che si utilizza sarà quindi:*

1. *Si consideri la funzione* POPULATE($N$, $i$, *nnarray*) *che richiede in input $N$, $i$ ossia l'indice nel nuovo punto da creare e nnarray, il vettore con i k punti vicini al nuovo $x_i$.*
2. ***while*** $N \neq 0$ *do*
3. *nn=random(*$1, k$*)*
4. ***for*** *j in range(*1,*numero di feature del dataset) do*
5. *Compute: dif= Sample[nnarray[nn]][attr]- Sample[j][attr]*
   *dove Sample[][] è il vettore dei campioni originariamente presenti nella classe minore*
6. *Compute: gap =random(*$0, 1$*)*
7. *Synthetic[newindex][attr]=Sample[i][attr]+gap* $\cdot$ *dif*
   *dove Synthetic[][] è il vettore dei campioni sintetici*
8. ***end for***
9. *newindex= newindex +1*
   *dove newindex tiene conto del numero di campioni artificiali generati ed inizialmente vale 0*
10. *N= N-1*
11. ***end while***
12. *L'algoritmo fornisce come output $N$ nuovi punti dellla classe minore.*

Ora che si è definito la funzione *POPULATE(N, i, nnarray)* è possibile definire l'algoritmo *SMOTE*:

**Algoritmo 2.** *SMOTE*

1. *Si consideri la funzione* SMOTE($T$, $N$, $k$) *che richiede in input $T$, ossia tutti i punti appartenti alla classe minore, $N$ e $k$, il numero di vicini considerati.*
2. ***if*** $N < 100$ ***then***
3. *Randomize the $T$ minority class samples*
4. $T = (N/100) * T$
5. $N = 100$

6. ***end if***
7. $N = (int)N/100$ *supponendo che* $N$ *sia divisibile per* 100
8. ***for*** $i$*=1 to* $T$ ***do***
9. *Compute the k-nearest neighbors for* $i$*, and save the indices in the nnarray*
10. POPULATE($N$, $i$, *nnarray*)
11. ***end for***

Purtroppo anche questo algoritmo, seppur molto efficace presenta delle criticità. In particolare, la maggior parte dei nuovi punti creati si trovano vicino alle zone più dense di elementi della classe meno rappresentata. Di conseguenza difficilmente si creeranno nuovi punti interpolando gli elementi più lontani. Inoltre, è possibile che vengano creati *noisy points*, cioè punti che non possono essere interpretati e utilizzati correttamente.
Per risolvere questi inconvenienti sono stati sviluppati degli algoritmi leggermente differenti rispetto allo *SMOTE algorithm* che tentano di imparare i limiti di ciascuna classe per evitare classificazioni errate.

### 2.3.2 NearMiss

Questo algoritmo per ribilanciare il dataset appartiene alle *Undersampling technique* e, di conseguenza, quello che permette di fare è eliminare dei campioni dalla classe maggiore fino a che le due classi sono bilanciate secondo il rapporto 1 : 1. Il rischio maggiore quando si sfruttano questi metodi per bilanciare il dataset è la perdita di informazioni. Per evitare ciò l'algoritmo in questione opera nel seguente modo:

**Algoritmo 3.** *NearMiss([9])*

1. *In primo luogo l'algoritmo trova la distanza tra tutte le istanze della classe maggiore e quelle della classe minore.*
2. *Successivamente, vengono selezionati n elementi della classe più rappresentata che hanno una particolare relazione con gli elementi della classe minore.*
3. *Si prosegue fino a che il numero degli elementi selezionati della classe maggiore è pari al numero degli elementi della classe minore.*

Per determinare gli elementi più vicini della classe maggiore possono essere utilizzate 3 tecniche che corrispondono ad altrettante versioni:

1. **NearMiss-Version 1**: seleziona i campioni della classe maggiore la cui distanza media dai $k$ punti della classe minore è più piccola.
2. **NearMiss-Version 2**: seleziona i campioni della classe più rappresentata la cui distanza media dai $k$ punti più lontani della classe minore è più piccola.

3. **NearMiss-Version 3**: lavora in 2 step; in primo luogo si seleziona e si tiene a memoria per ogni punto della classe meno presente un numero intero $m$ di elementi dell'altra classe che hanno la caratteristica di essere i più vicini. Successivamente si selezionano i punti della classe maggiore che si trovano nelle liste create tali che la loro distanza media dagli $n$ elementi dell'altra classe è maggiore.

### 2.3.3 Random Undersampling

Il metodo *Random Undersampling* appartiene, come dice già il nome, agli *Undersampling method* ed è meno sofisticato dei due precedentemente descritti dal momento che si limita a ricampionare il dataset eliminando randomicamente dei punti.
Più in particolare l'algoritmo funziona nel seguente modo:

**Algoritmo 4.** *Random Undersampling*

1. *Si individuano i campioni appartenenti alla classe più rappresentata.*
2. *Si procede a scegliere in modo casuale elementi della classe maggiore fino a che le due classi risultano perfettamente bilanciate.*

Questa procedura, dunque, se da una parte risulta molto semplice da implementare e impiega poco tempo per dare l'output, dall'altro rischia di eliminare delle informazioni importanti inavvertitamente. Di conseguenza, il suo utilizzo deve essere consapevole e, in particolare, può risultare inopportuno sfruttare tale metodo se il dataset è poco esteso.

## 2.4 Selezione delle Feature

Sia nel caso dell'apprendimento supervisionato sia in quello non supervisionato è estremamente importante riuscire a selezionare le feature più rilevanti per ottenere un'accuratezza più vicina possibile al 100% e permettere così la costruzione di un buon modello per studiare un fenomeno. Invero, soprattutto quando si dispone di un dataset di grandi dimensioni, se non si selezionano le feature più importanti si rischia di andare in overfitting nonché di aver bisogno di un lungo periodo di tempo affinché gli algoritmi producano l'output desiderato.
Solitamente il processo di selezione delle feature avviene in 4 passi:

1. Inizialmente si seleziona un sottoinsieme $S$ di feature basato su differenti strategie che dipendono dal tipo di metodo utilizzato.
2. Nel secondo step $S$ viene valutato secondo alcuni criteri stabiliti dall'algoritmo sfruttato.
3. Dopo aver ripetuto i primi due punti con differenti sottoinsiemi si sceglie il sottoinsieme $S$ che meglio rispetta i criteri definiti in partenza.
4. Infine, il sottoinsieme selezionato viene convalidato usando un *validation set*.

Prima di approfondire l'argomento si definisca formalmente che cosa si intende per *rilevanza delle feature.*
Si assuma che ciascuna feature sia indipendente dall'altra:

**Definizione.** *Una feature $X$ si definisce* ***fortemente rilevante*** *se, una volta rimossa dal dataset, tutti i risultati dell'accuratezza risultano deteriorati.*

**Definizione.** *Una feature $X$ è* ***debolmente rilevante*** *se non è fortemente rilevante ed esiste un sottoinsieme $S$ in cui l'accuratezza è peggiore dell'accuratezza in $S \cup \{X\}$.*

**Definizione.** *Una feature $X$ è* ***irrilevante*** *se non è né fortemente né debolmente rilevante.*

Nell'apprendimento non supervisionato è più difficile distinguire le feature rilevanti dalle altre perché naturalmente non si conosce a quale classe appartiene ciascun campione. Tuttavia, questa operazione è particolarmente importante nei casi in cui si ha a che fare con dataset di grandi dimensioni. In tale situazione le modalità di selezione utilizzate si basano su vincoli aggiuntivi che si decide di imporre a seconda del problema da affrontare e sulla qualità delle misure ottenute.
Nell'apprendimento supervisionato, invece, risulta più facile e più usuale applicare una modalità per distinguere le feature che permettono di costruire un modello migliore. Per tale ragione le tipologie di algoritmi disponibili sono molte ed eterogenee.
Una prima distinzione può essere fatta in base al dataset di cui si dispone; se non è possibile avere in partenza tutto il dataset e non si conosce quanto sarà esteso si parla di *Streaming feature* e sarà opportuno utilizzare algoritmi specifici per questa tipologia di problemi. Se, invece, le feature presentano particolari strutture intrinseche come alberi e grafi si parla di *Structured feature* e anche in tal situazione è meglio concentrarsi su algoritmi specifici.
Se, però, le feature sono una indipendente dall'altra e si conosce l'intero dataset allora gli algoritmi che è preferibile utilizzare si possono raggruppare in 3 grandi categorie:

- **metodi Filtro**: usano le proprietà intrinseche dei dati per selezionare le feature senza utilizzare alcun algoritmo di classificazione. La selezione delle feature può avvenire in due modi: ogni feature può essere analizzata singolarmente e avrà, quindi, un punteggio ad essa associato; si parla, in questo caso, di *univariate feature selection.* In alternativa si può scegliere un algoritmo che associa un punteggio a dei sottogruppi di feature e, dunque, si definisce *multivariate feature selection.* In entrambi i casi, tuttavia, una volta che si è analizzato tutto il dataset si procede ad ordinare le feature o i gruppi di feature da quelli con il punteggio maggiore a quelli con il punteggio minore e si procede alla selezione di quelle che si trovano nella prima parte di questa classifica.
  Ci sono differenti modalità per calcolare il punteggio che permette poi di ordinare le feature ma il più utilizzato nel caso di *univariate feature selection* è il cosiddetto *Fisher Score* che si calcola nel modo seguente:

$$S_i = \frac{\sum_{k=1}^{K} n_j (\mu_{ij} - \mu_i)^2}{\sum_{k=1}^{K} n_j \rho_{ij}^2} \tag{2.1}$$

dove, $\mu_{ij}$ e $\rho_{ij}$ sono la media e la varianza della feature $i$ nella classe $j$, $\mu_i$ è la media della feature $i$ e $n_j$ è il numero di campioni appartenenti alla classe $j$. Se, invece, si preferisce utilizzare la *multivariate feature selection* si può sfruttare una generalizzazione del *Fisher Score*.
In conclusione, dunque, questa classe di algoritmi è vantaggiosa perché non dipende da uno specifico classificatore ma non tiene neanche conto degli effetti che avrà la selezione effettuata nella prestazione dell'algoritmo che si utilizzerà in seguito.

- **metodi Wrapper**: tengono in considerazione le specifiche caratteristiche del modello di apprendimento supervisionato che si vuole andare ad utilizzare. In questo caso si fa cross-validation, ossia si sfrutta parte del dataset per verificare che quanto ottenuto sia un buon modello (Si veda la Sezione 2.5.1 per approfondire l'argomento). Si possono utilizzare un numero molto ampio di strategie per la ricerca delle feature come ad esempio *branch and bound*, *best first* e *hill climbing*. Solitamente, inoltre, questa classe di metodi permette di ottenere un'accuratezza più elevata rispetto a quelli di tipo filtro anche perché il sottoinsieme di feature che viene selezionato è il migliore per lo specifico classificatore. Tuttavia, il vero problema di questi modelli è che sono computazionalmente dispendiosi. Infatti, la dimensione dello spazio di ricerca per $m$ feature è $O(2^m)$ e quindi diventa incalcolabile a meno che $m$ sia piccolo. Si tratta, invero, di un problema NP-hard.

- **metodi Embedded**: a differenza dei modelli wrapper sono molto meno costosi computazionalmente. Inoltre, prevedono un'interazione con il modello di classificazione perché effettuano la selezione delle variabile all'interno del processo di addestramento e, quindi, sono più performarti dei metodi filtro. Ci sono diverse tipologie di metodi embedded: un tipo, ad esempio, utilizza tutte le feature per addestrare un modello e successivamente tenta di eliminare alcune feature settando a 0 il coefficiente ad esse associato. In un altro caso ci sono dei modelli di regolarizzazione che tentano di minimizzare gli errori di fitting e allo stesso tempo forzano alcuni coefficienti a essere piccoli o vicini a 0. Le feature con i coefficienti più bassi saranno quelle eliminate.

### 2.4.1 Alberi di decisione

Uno degli algoritmi più utilizzati per selezionare le feature è sicuramente l'algoritmo di Ramdom Forest. Tuttavia, per comprenderlo a fondo è necessario fare un passo indietro per capire come funzionano i *decision tree* o alberi di decisione [16].
Prima di tutto si introduce velocemente il concetto di albero:

**Definizione.** *Un **albero** è un grafo non orientato in cui due nodi qualsiasi sono connessi da uno ed un solo cammino.*
*Un **albero con radice** è un albero orientato in cui esiste un nodo chiamato radice, per cui è possibile trovare un cammino che parte da esso ed arriva a qualsiasi altro nodo del grafo.*
*Se esiste un arco dal nodo $t_1$ al nodo $t_2$ allora $t_1$ è il **nodo padre** di $t_2$ mentre $t_2$ è il*

***nodo figlio***. *Un nodo senza figli si dice* ***nodo foglia*** *mentre, se ne ha, è chiamato* ***nodo interno***. *Un albero è* ***binario*** *se tutti i nodi interni hanno esattamente due figli.*

Sia $X$ lo spazio di tutti i possibili input; allora un albero di decisione può essere definito come un modello di apprendimento supervisionato dove la funzione $\psi : X \to Y$ è un albero con radice pari a $X$ e dove ogni nodo $t$ rappresenta un sottoinsieme $X_t \subseteq X$. Solitamente gli alberi di decisione sono alberi binari ma questo non implica che i due sottoinsiemi che di volta in volta si vanno a creare contengano un numero di elementi simili. Dopo un numero $n$ di iterazioni si terminano le divisioni o *split* e si cerca di scegliere il migliore output per ciascun nodo foglia. In particolare, se l'albero è di classificazione allora ai nodi foglia sarà assegnato uno dei possibili output; se, invece, è un albero di regressione a ciascun nodo foglia corrisponde un intervallo di valori. A questo punto per ogni altro $x \in X$ a partire dal nodo radice si verifica a quale sottoinsieme appartiene e reiterando il processo si arriva fino al nodo foglia a cui è associato un determinato output. L'algoritmo che si utilizza è quindi:

**Algoritmo 5.** *Predizione dell'output in un albero di decisione*
*Sia $\psi : X \to Y$, sia $t_0$ il nodo radice e sia $\bar{y}_t = \psi(x)$ l'output del nodo foglia $t$.*

1. *Si consideri la funzione PRED($\psi$, $x$)*
2. $t = t_0$
3. ***while*** *$t$ non è una foglia* ***do***
4. $t$= *nodo figlio $t^*$ di $t$ tale che $x \in X_t$*
5. ***end while***
6. ***return*** $\bar{y}_t$

I criteri con cui si effettuano le divisioni sono di estrema importanza per l'accuratezza finale ma la scelta degli split risulta anche rilevante. Tuttavia, una divisione è considerata buona quando i sottoinsiemi creati sono più omogenei possibili, ossia quando i campioni di una determinata classe capitano per la maggior parte in uno dei nuovi sottoinsiemi creati.
Le strategie possibili per verificare l'omogeneità sono svariate ma di seguito vengono riportate tre algoritmi tra i più utilizzati:

- **Gini Index**: questo algoritmo, utilizzato solo per alberi di classificazione, calcola l'eterogeneità di un nodo $t$ con la formula

$$G(t) = 1 - \sum_{j=1}^{k} (p_j)^2$$

dove $p_j$ è la frequenza della classe $j$-esima e $k$ è il numero di output possibili.
Si osservi che se $p_1 = p_2 = \cdots = p_k$ si parla di massima *impurità* e, di conseguenza,

si può affermare che lo split non è stato buono. Se, al contrario, $G(t) = 0$ significa che il nodo presenta campioni di un'unica classe e quindi si è nel miglior caso possibile.

- **Chi-Square**: è un algoritmo che si basa sulla differenza tra nodo padre e nodo foglia usato per alberi di classificazione. In particolare la formula che si sfrutta è

$$Chi - Square = \sum_{j=1}^{k} \frac{(p_j^* - p_j)^2}{(p_j)^{\frac{1}{2}}}$$

  dove $p_j^*$ è la frequenza della classe $j$-esima nel nodo figlio mentre $p_j$ è la frequenza della medesima classe nel nodo padre.
  In questo caso più il valore è alto più i due nodi in esami sono differenti e, dunque, si ha un buon split.

- **Riduzione della varianza**: nel caso di alberi di regressione è utile sfruttare la classica formula della varianza

$$Var = \sum_{j=1}^{k} \frac{(X_j - \bar{X}_j)^2}{n}$$

  dove $k$ è il numero di classi possibili, $X_j$ è il numero di campioni appartenenti alla classe $j$-esima nel nodo $t$, $\bar{X}_j$ è la media dei valori del nodo padre. Infine $n$ è la cardinalità del sottoinsieme riconducibile al nodo $t$.

Oltre al criterio con cui si effettuano gli split è importante prestare attenzione alla profondità che l'albero raggiunge e, quindi, al numero di split che vengono fatti. Infatti, se il dataset è molto grande non si deve pensare che fare un enorme numero di divisione sia la strategia migliore sia perché i tempi computazionali diventano molto lunghi sia perché si rischia di andare in overfitting. In aggiunta spesso gli alberi di decisione sono utilizzati perché permettono di visualizzare il problema e, per tale ragione, è sufficiente scegliere di fare un numero non troppo elevato di iterazioni. In ogni caso, comunque, è possibile determinare il numero migliore di split, il numero di feature da considerare per effettuare ciascuna divisione oltre che il numero minimo e massimo di campioni assegnati a ciascun nodo foglia facendo *grid search*, ossia utilizzando un metodo che, dopo aver provato tutte le varie combinazioni possibili, restituisce quella che ha dato risultati migliori.

### 2.4.2 Random Forest

Le *Random Forest* [17] sono molto sfruttate perché, a differenza dei decision tree, la probabilità di cadere in overfitting è molto più bassa. Allo stesso tempo, inoltre, è un algoritmo facile da implementare e che offre una buona rappresentazione del problema. Il termine Forest fa riferimento semplicemente al fatto che vengono creati più alberi mentre Random si riferisce a due differenti caratteristiche. In primo luogo per costruire ciascun albero viene selezionato un sottoinsieme del dataset di partenza in modo randomico. Più

in particolare, dopo che un campione è stato selezionato per essere usato in un determinato albero questo viene reinserito nel dataset e, di conseguenza, è possibile che venga riscelto nuovamente. In seconda istanza anche le feature che ciascun albero utilizza per compiere gli split sono scelte casualmente. Solitamente se si hanno $n$ feature solo $\sqrt{n}$ sono selezionate per dividere il dataset iniziale in sottoinsiemi sempre più piccoli. Se, tuttavia, si preferisce utilizzare tutte le feature è possibile scegliere questa opzione nel momento dell'implementazione.
Una volta che l'algoritmo di Random Forest ha costruito il numero di alberi scelti dall'utente o previsti per default (solitamente alcune centinaia) si calcola la media delle predizioni di tutti i singoli alberi e si restituisce questa come output. Generalmente quello che si ottiene è un risultato molto migliore rispetto ad applicare semplicemente l'algoritmo del decision tree perché dividendo più volte il dataset e ottenendo un buon numero di risultati anche gli errori dovuti ad eventuali dati rumorosi vengono smorzati.
Infine, è importante sottolineare che oltre ad offrire una modalità per selezionare le feature Random Forest è comunque una tecnica che può essere inserita nella categoria di algoritmi di apprendimento supervisionato. Di fatto, infatti, oltre a restituire una classifica delle feature, crea anche un modello che, a partire dal dataset di partenza fornito, tenta di predire la classe di appartenenza di eventuali nuovi campioni. Inoltre, una volta che viene applicato Random Forest quello che si ottiene come output è l'accuratezza del modello, ossia la precisione con cui è in grado di classificare i campioni.

## 2.5 Support Vector Machine (SVM)

Le Support Vector Machine [18] sono una classe di macchine di apprendimento supervisionato che si basano su concetti di statistica dell'apprendimento. Analizziamo prima il caso linearmente separabile in cui è possibile trovare un iperpiano di separazione, detto *iperpiano ottimo*, che massimizza la distanza tra gli elementi dei due insiemi. Vediamo ora in dettaglio con quale metodo è possibile trovare l'iperpiano ottimo.
Si consideri due insiemi disgiunti di punti $\mathcal{A}$ e $\mathcal{B}$ in $R^n$ e si supponga che siano *linearmente separabili*, ossia che esista un iperpiano $\mathcal{H} = \{x \in R^n | w^T x + \theta = 0\}$ tale che

$$\begin{aligned} w^T x^i + \theta \geq 1 \quad & x^i \in \mathcal{A} \\ w^T x^i + \theta \leq -1 \quad & x^i \in \mathcal{B} \end{aligned} \tag{2.2}$$

dove $w \in R^n$ è il vettore dei pesi e $x$ è il cosiddetto *vettore di supporto*.
Inoltre, si definisce *margine di separazione* di $\mathcal{H}$ la minima distanza $\rho$ tra i punti di $\mathcal{A} \cup \mathcal{B}$ ossia

$$\rho(w, \theta) = \min_{x^i \in \mathcal{A} \cup \mathcal{B}} \frac{|w^T x^i + \theta|}{\|w\|} \tag{2.3}$$

Se l'iperpiano $\mathcal{H}(w^*, \theta^*)$ ha margine di separazione massimo si definisce *iperpiano ottimo* ed è possibile dimostrare la sua esistenza ed unicità.

Il problema che si vuole dunque risolvere è

$$\begin{aligned} &\max_{w,\theta} \rho(w,\theta) \\ &w^T x^i + \theta \geq 1 \qquad x^i \in \mathcal{A} \\ &w^T x^i + \theta \leq -1 \quad x^i \in \mathcal{B} \end{aligned} \tag{2.4}$$

Tuttavia, quello che è possibile dimostrare è che (2.4) è analogo al suo problema duale

$$\begin{aligned} &\min_{w,\theta} \frac{1}{2}\|w\|^2 \\ &y^i(w^T x^i + \theta) - 1 \geq 0 \quad i = 1, \ldots, |\mathcal{A} \cup \mathcal{B}| \end{aligned} \tag{2.5}$$

dove $y^i = 1$ se $x^i \in \mathcal{A}$ e $y^i = -1$ se $x^i \in \mathcal{B}$. Con questa riformulazione ciò che si è ottenuto è un problema quadratico convesso con vincoli lineari. Si consideri il seguente:

**Teorema 1.** ***Dualità Forte*** *[19]*
*Se il primale ha una soluzione ottima, allora*

1. *il duale ha una soluzione ottima;*
2. *i valori delle due soluzioni sono uguali.*

Di conseguenza, se $(w^*, \theta^*)$ è la soluzione del problema duale (2.5), allora darà lo stesso valore ottimo del primale (2.4).
Si noti che è possibile sostituire i vincoli di (2.5) con dei vincoli "più facili" sulle variabili duali sfruttando i moltiplicatori di Lagrange. Si consideri, infatti, la funzione Lagrangiana

$$L(w, \theta, \alpha) = \frac{1}{2}\|w\|^2 - \sum_{i=1}^{|\mathcal{A}\cup\mathcal{B}|} \alpha_i [y^i(w^T x^i + \theta) - 1] \tag{2.6}$$

Si introduca quindi il cosiddetto *duale di Wolfe*

$$\begin{aligned} \max_{w,\theta,\alpha} L(w,\theta,\alpha) = &\frac{1}{2}\|w\|^2 - \sum_{i=1}^{|\mathcal{A}\cup\mathcal{B}|} \alpha_i [y^i(w^T x^i + \theta) - 1] \\ &w = \sum_{i=1}^{|\mathcal{A}\cup\mathcal{B}|} \alpha_i y^i x^i \\ &\sum_{i=1}^{|\mathcal{A}\cup\mathcal{B}|} \alpha_i y^i x^i \geq 0 \qquad\qquad \alpha_i \geq 0 \qquad i = 1, \ldots, |\mathcal{A} \cup \mathcal{B}| \end{aligned} \tag{2.7}$$

Per il teorema introdotto precedentemente (2.7) è equivalente a

$$\begin{aligned}\min_{\alpha} W(\alpha) =& \frac{1}{2}\sum_{i=1}^{|\mathcal{A}\cup\mathcal{B}|}\sum_{j=1}^{|\mathcal{A}\cup\mathcal{B}|} y^i y^j (x^i)^T x^j \alpha_i \alpha_j - \sum_{i=1}^{|\mathcal{A}\cup\mathcal{B}|}\alpha_i \\ & \sum_{i=1}^{|\mathcal{A}\cup\mathcal{B}|}\alpha_i y^i = 0 \qquad\qquad \alpha_i \geq 0 \qquad i = 1,\ldots,|\mathcal{A}\cup\mathcal{B}|\end{aligned} \tag{2.8}$$

Ora che si è giunti a questa forma si può affermare che (2.8) ammette almeno una soluzione ottima $\alpha^*$ e, di conseguenza, il vettore $w^*$ che è soluzione ottima di (2.7) si ricava da

$$w^* = \sum_{i=1}^{|\mathcal{A}\cup\mathcal{B}|}\alpha_i^* y^i x^i$$

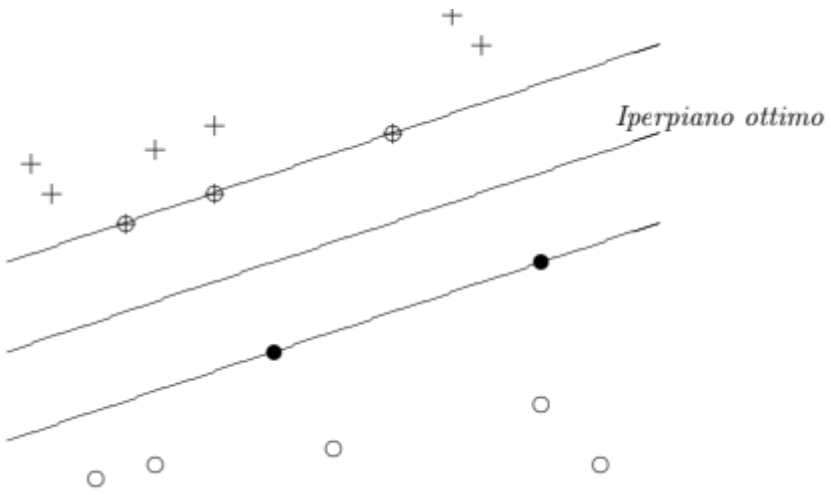
Esso, dunque, dipende esclusivamente da $x^i$ nel caso in cui $\alpha_i \neq 0$. Inoltre si osservi che il valore ottimo di (2.7) sarà tale se

$$\alpha_i^*[y^i((w^*)^T x^i + \theta^*) - 1] = 0 \quad i = 1,\ldots,|\mathcal{A}\cup\mathcal{B}|$$

dove $\theta^*$ è la soglia ottima. Di conseguenza la funzione di decisione è

$$f(x) = sgn((w^*)^T x^i + \theta^*)$$

Di seguito è riportato un esempio in cui si vede l'iperpiano ottimo.




Se, invece, si è in presenza di due insiemi disgiunti $\mathcal{A}$ e $\mathcal{B}$ ma non linearmente separabili è possibile ricondursi a quanto visto precedentemente introducendo delle variabili artificiali

$\varepsilon^i$ con $i = 1, \ldots, |\mathcal{A} \cup \mathcal{B}|$.
Si cerca un iperpiano $\mathcal{H}$ tale che:

$$\begin{aligned} & w^T x^i + \theta \geq 1 - \varepsilon^i && x^i \in \mathcal{A} \\ & w^T x^i + \theta \leq -1 + \varepsilon^i && x^i \in \mathcal{B} \\ & \varepsilon^i \geq 0 && i = 1, \ldots, |\mathcal{A} \cup \mathcal{B}| \end{aligned} \tag{2.9}$$

Si osservi che se un $x^i$ non è correttamente classificato, allora necessariamente $\varepsilon^i$ è maggiore di 1 in quanto deve essere che $x^i \in \mathcal{B}$ e $w^T x^i + \theta > 0$ o viceversa. Di conseguenza

$$\sum_{i=1}^{|\mathcal{A} \cup \mathcal{B}|} \varepsilon^i$$

è un upper bound del numero degli errori di classificazione dei vettori di training. Per tale ragione, si aggiunge alla funzione obiettivo il termine $C \sum_{i=1}^{|\mathcal{A} \cup \mathcal{B}|} \varepsilon^i$ dove $C > 0$ pesa l'errore di training.
Il problema da risolvere risulta perciò:

$$\begin{aligned} & \min_{w, \theta, \varepsilon} \frac{1}{2} ||w|| + C \sum_{i=1}^{|\mathcal{A} \cup \mathcal{B}|} \varepsilon^i \\ & y^i (w^T x^i + \theta) - 1 + \varepsilon^i \geq 0 \\ & \varepsilon^i \geq 0 \qquad\qquad i = 1, \ldots, |\mathcal{A} \cup \mathcal{B}| \end{aligned} \tag{2.10}$$

Il duale di Wolfe diventa quindi:

$$\begin{aligned} \min_{\alpha} W(\alpha) = & \frac{1}{2} \sum_{i=1}^{|\mathcal{A} \cup \mathcal{B}|} \sum_{j=1}^{|\mathcal{A} \cup \mathcal{B}|} y^i y^j (x^i)^T x^j \alpha_i \alpha_j - \sum_{i=1}^{|\mathcal{A} \cup \mathcal{B}|} \alpha_i \\ & \sum_{i=1}^{|\mathcal{A} \cup \mathcal{B}|} \alpha_i y^i = 0 \\ & 0 \geq \alpha_i \geq C \qquad\qquad i = 1, \ldots, |\mathcal{A} \cup \mathcal{B}| \end{aligned} \tag{2.11}$$

Inoltre, nel caso di insieme in $\mathbb{R}^n$ non separabili per avere una migliore classificazione è opportuno impiegare funzioni non lineari che permettono di trasformare $\mathbb{R}^n$ in uno spazio più grande dove i due insiemi sono separabili. Se $x^i \in \mathbb{R}^n$, $i = 1, \ldots, |\mathcal{A} \cup \mathcal{B}|$ e $y_i \in \{-1, 1\}$ si esegue la trasformazione lineare $\psi : \mathbb{R}^n \to \mathcal{H}$ dove $\mathcal{H}$ è uno spazio di dimensioni maggiori di $n$. A questo punto ci si può ricondurre a quanto visto precedentemente sostituendo però $x^i$ con $\psi(x^i)$. Il problema da risolvere è quindi

$$\begin{aligned} \min_{\alpha} S(\alpha) = & \frac{1}{2} \sum_{i=1}^{|\mathcal{A} \cup \mathcal{B}|} \sum_{j=1}^{|\mathcal{A} \cup \mathcal{B}|} y^i y^j \psi(x^i)^T \psi(x^j) \alpha_i \alpha_j - \sum_{i=1}^{|\mathcal{A} \cup \mathcal{B}|} \alpha_i \\ & \sum_{i=1}^{|\mathcal{A} \cup \mathcal{B}|} \alpha_i y^i = 0 \qquad\qquad 0 \geq \alpha_i \geq C \qquad i = 1, \ldots, |\mathcal{A} \cup \mathcal{B}| \end{aligned} \tag{2.12}$$

Il vettore $w^*$ a questo punto diventa

$$w^* = \sum_{i=1}^{|\mathcal{A}\cup\mathcal{B}|} \alpha_i^* y^i \psi(x^i)$$

e, di conseguenza, la funzione di decisione è

$$f(x) = sgn((w^*)^T\psi(x) + \theta^*) = sgn(\sum_{i=1}^{|\mathcal{A}\cup\mathcal{B}|} \alpha_i^* y^i \psi(x^i)^T\psi(x) + \theta^*)$$

In realtà quello che si utilizza quasi sempre nella pratica non è tanto la funzione $\psi(x^i)$ bensì la *funzione kernel* $K$, ossia una funzione simmetrica definita come:

$$k(x, z) = \psi(x)^T\psi(z) \tag{2.13}$$

dove $x, z \in \mathbb{R}^n$ Più in particolare la condizione che deve essere rispettata affinché esista $\psi(x)$ tale che $k(x, z)$ sia una funzione che rappresenta il prodotto interno come in (2.13) deve essere che

$$(k(x^i, x^j))_{i,j=1}^{|\mathcal{A}\cup\mathcal{B}|} = \begin{pmatrix} K(x^1, x^1) & \dots & K(x^1, x^{|\mathcal{A}\cup\mathcal{B}|}) \\ \vdots & \vdots & \vdots \\ K(x^{|\mathcal{A}\cup\mathcal{B}|}, x^1) & \dots & K(x^{|\mathcal{A}\cup\mathcal{B}|}, x^{|\mathcal{A}\cup\mathcal{B}|}) \end{pmatrix}$$

è semidefinita positiva per ogni insieme di vettori di training.
Il problema da risolvere è dunque:

$$\begin{aligned} \min_\alpha S(\alpha) = & \frac{1}{2} \sum_{i=1}^{|\mathcal{A}\cup\mathcal{B}|} \sum_{j=1}^{|\mathcal{A}\cup\mathcal{B}|} y^i y^j k(x^i, x^j)\alpha_i\alpha_j - \sum_{i=1}^{|\mathcal{A}\cup\mathcal{B}|} \alpha_i \\ & \sum_{i=1}^{|\mathcal{A}\cup\mathcal{B}|} \alpha_i y^i = 0 \\ & 0 \geq \alpha_i \geq C \qquad\qquad i = 1, \dots, |\mathcal{A}\cup\mathcal{B}| \end{aligned} \tag{2.14}$$

Si osservi che quello che si è appena descritto è un problema di programmazione quadratica convessa.
La funzione di decisione, invece, può essere scritta come

$$f(x) = sgn(\sum_{i=1}^{|\mathcal{A}\cup\mathcal{B}|} \alpha_i^* y^i k(x, x^i) + \theta^*)$$

Tra i kernel più usati si possono individiare:

- **kernel lineare**: $k(x, z) = x^T z$;
- **kernel polinomiale**: $k(x, z) = (x^T z + 1)^p$ con $p$ intero, $p \leq 1$;

- **kernel gaussiano (RBF)**: $k(x, z) = e^{-\gamma||x-z||^2}$ con $\gamma > 0$;
- **kernel di tipo tangente iperbolico**: $tanh(\beta x^T z + \sigma)$.

I parametri $p$, $\sigma$, $\beta$ e $\gamma$ sono definiti *iperparametri* e la scelta di quest'ultimi risulta determinante per avere una buona accuratezza.

### 2.5.1 Cross-validation

In questa sezione si analizza brevemente il processo che permette di comprendere se il modello ottenuto ha una buona capacità predittiva o meno. Finora si è già in qualche modo accennato al fatto che solitamente si parte da un dataset denominato *training set* che serve per l'apprendimento e, successivamente, si sfrutta un nuovo dataset, definito *test set* o *validation set* che permette di definire l'accuratezza di quanto ottenuto. Si veda ora più in dettaglio come funziona il tutto.
A volte è possibile comprendere se si è ottenuto un buon modello anche senza utilizzare un nuovo dataset. Ciò avviene trovando il cosiddetto *errore di training* calcolando la differenza tra la classe predetta di ciascun campione e la risposta originale. Questo permette di dare una prima idea ma, per avere una migliore certezza, è conveniente fare *cross validation*, ossia applicare il modello a nuovi campioni.
Il metodo più semplice di cross validation è l'*holdout method* in cui semplicemente si rimuove una parte del training set e si usa come test set. La capacità predittiva in questo modo risulta sicuramente migliorata ma, non sapendo quali campioni vengono scelti come test set, si potrebbero ottenere dei risultati diversi ogni volta che si riapplica l'algoritmo. Per migliorare quanto appena descritto è possibile applicare *K-Fold cross validation*. Quello che questo algoritmo permette di fare è dividere i dati in $k$ sottogruppi affinché sia possibile applicare l'holdout method $k$ volte cambiando ogni volta il sottogruppo usato come test e usando gli altri $k - 1$ come training set. In questo modo l'errore risulta essere la media degli errori delle $k$ iterazioni.

Inoltre, quando il dataset di partenza è *sbilanciato* è necessario prestare particolare attenzione a questa fase. Infatti, come si è già detto, per costruire un buon modello predittivo è di fondamentale importanza adottare una tecnica per ricampionare il dataset. Tuttavia, si può scegliere di ribilanciare il dataset e, solo in seguito, fare cross validation oppure si può effettuare il ricampionamento del dataset all'interno del processo di cross validation stesso. In particolare, in questo secondo caso esclusivamente nel training set si adotta una tecnica per ribilanciare le classi mentre il test set non viene in alcun modo modificato.
Nel caso in cui si adotti una tecnica di oversampling scegliere una o l'altra modalità non è affatto indifferente ed è importante conoscere i pro e i contro di entrambe.
La prima strategia è più frequentemente utilizzata in ambito medico e biologico e, in generale, nei casi in cui si è interessati a confrontare i risultati ottenuti adottando differenti algoritmi di classificazione. D'altro canto, però, questo approccio potrebbe rivelarsi *overoptimistic*, ossia si potrebbe ottenenere una capacità predittiva sovrastimata. Facendo oversampling, infatti, spesso si vengono a creare copie di un campione o nuovi istanze

molto simili a quelle già presenti e, di conseguenza, nel momento in cui si suddivide il dataset in training set e test set, il medesimo punto potrebbe essere sfruttato sia per costruire il modello sia per verificare la bontà di quest'ultimo. Quando, quindi, si utilizzano algoritmi di oversampling molto semplici come, ad esempio, Random Oversampling, è meglio adottare la seconda strategia. Invero, anche se meno intuitiva, questa modalità risulta essere più conservativa in quanto, effettuando l'oversampling solo sul training set, calcola l'accuratezza del modello a partire da campioni sicuramente differenti rispetto a quelli utilizzati nella fase di apprendimento.

## 2.6 Clustering

Gli algoritmi di clustering sono dei modelli di apprendimento non supervisionato ampiamente sfruttati perché permettono di dividere il dataset in gruppi, o *cluster*, utili per avere informazioni aggiuntive sui dati. Inoltre, questo metodo può essere una tecnica preliminare per avere un'idea più chiara del problema da affrontare in quanto riproduce una modalità comune che tutti gli esseri umani sfruttano per descrivere il mondo circonstante. Fin da bambini, infatti, uno dei modi che il cervello umano utilizza per immagazzinare nuove informazioni è associare delle etichette ad ogni particolare oggetto o nozione per poi raggruppare quelle con caratteristiche simile e facilitare il processo di memorizzazione. I cluster possono essere utilizzati anche nella fase finale per confermare quanto già osservato con l'applicazione di altri algoritmi o raccogliere dei risultati più dettagliati.
Negli algoritmi di cluster si costruiscono i gruppi senza conoscere a quale classe appartiene ciascun campione. Di conseguenza, l'obiettivo è quello di raggruppare gli oggetti che sono più simili tra loro considerando l'intero insieme di feature o selezionandone alcune che si ritengono più importanti. É fondamentale sottolineare, tuttavia, che non esiste una definizione precisa di cluster e, di conseguenza non è possibile trovare una regola generale che permette di ottenere sempre delle buone divisioni. Sarà, dunque, l'utente che sceglierà in quanti gruppi suddividere il dataset, eventualmente facendo varie prove, fino a che non si è soddisfatti del risultato in relazione allo scopo che si vuole raggiungere.
Nel corso del tempo si sono sviluppati differenti tipi di clustering a seconda del problema da affrontare. Una prima distinzione può essere fra *clustering gerarchici* e *clustering partizionali*. I clustering partizionali suddividono il dataset in gruppi non sovrapposti tra loro; al contrario i clustering gerarchici costruiscono cluster che contengono anche sottogruppi. Si ottiene, dunque, un albero in cui ogni sottogruppo è legato al cluster di appartenenza per mezzo di un arco. Un'altra differenziazione che è possibile fare è quella tra clusterings *esclusivi*, *sovrapponibili* e *fuzzy*. Nel primo caso quello che avviene è che ogni campione è associato ad un unico cluster. Nelle circostanze in cui, invece, un oggetto può appartenere simultaneamente a due o più gruppi si parla di clustering non esclusivo o sovrapponibili. Infine nel fuzzy clustering ogni oggetto appartiene ad ogni gruppo ma con peso diverso che può andare da 0 a 1. In aggiunta, a volte si impone la condizione che la somma dei pesi di ciascun cluster sia pari a 1. Se ad ogni oggetto è assegnato un cluster si parla di *clustering completo*; in caso contrario si parla di *clustering*

*parziale*. Quest'ultima opzione è utile soprattutto nei casi in cui ci sono alcuni campioni che non appartengono ad un insieme ben definito. Ora che si sono descritti i differenti tipi di clustering si passi invece ad analizzare le diverse caratteristiche che possono avere i singoli gruppi.

- **Well-Separated cluster**: si definisce in tal modo un cluster in cui ogni oggetto ha una distanza minore con tutti gli altri che appartengono al medesimo cluster rispetto a qualsiasi altro campione al di fuori di esso.
- **Prototype-Based cluster**: è un cluster in cui tutti gli oggetti sono più simili al prototipo che definisce il gruppo. Solitamente il prototipo corrisponde al *centroide*, ossia la media di tutti i punti del cluster.
- **Graph-Based cluster**: se i dati sono rappresentati come grafi, allora i cluster possono essere associati alle componenti connesse.
- **Density-Based cluster**: si definisce in tal modo un cluster che è molto denso nella regione in cui sono disposti i suoi oggetti e con una bassissima densità delle aree circostanti.

Dopo questa parte in cui si è introdotto l'argomento dal punto di vista generale ci si concentri su tre possibili tecniche per costruire cluster. Si approfondiranno quindi gli algoritmi *K-means*, *Agglomerative Hierarchical Clustering* e *DBSCAN*.

### 2.6.1 K-means

K-Means [22] è un algoritmo che crea Prototype-based cluster. Se si desidera creare $k$ gruppi allora l'algoritmo sceglierà k centroidi iniziali e ogni altro punto sarà associato al centroide più vicino. A questo punto, si ricalcoleranno i centroidi sfruttando i $k$ insiemi creati e si continuerà la procedura fino a che nessun centroide sarà diverso rispetto all'iterazione precedente oppure nessun punto cambierà di cluster. L'algoritmo si può quindi riassumere nel modo seguente:

**Algoritmo 6.** *k-means*

1. *Si selezionino k punti che saranno considerati i centroidi iniziali*
2. ***repeat***
3. *Si formino k gruppi assegnando ogni punto al centroide più vicino.*
4. *Si ricalcolino i centroidi facendo la media tra i campioni di ciascun cluster.*
5. ***until** Nessun centroide cambia.*

Nella Figura (2.6.1) si vede un esempio in cui si fanno da sinistra a destra 4 iterazioni dell'algoritmo e si arriva così ad individuare tre gruppi distinti con i relativi centroidi.

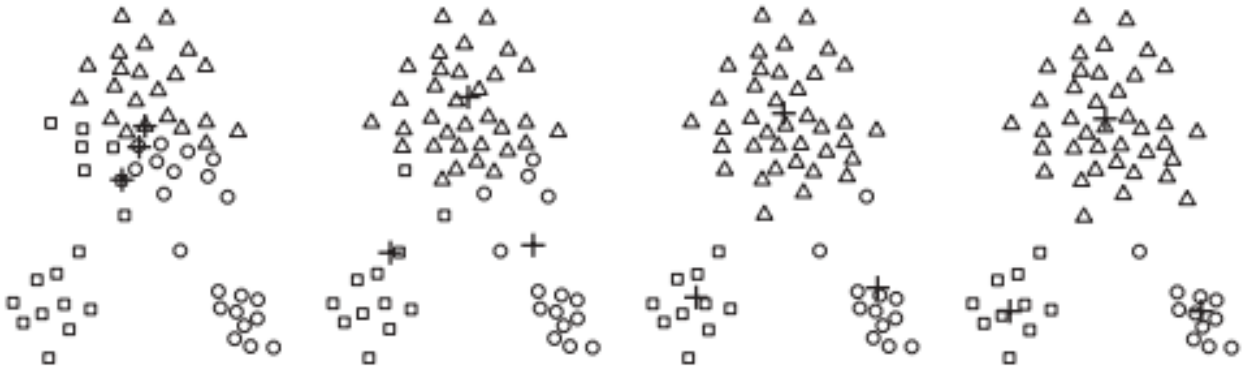


Figura 2.1: Iterazione di K-means

Si noti che non sempre è possibile far sì che l'algoritmo converga in pochi passi e, di conseguenza, a volte si impone una condizione più debole per fermarlo. Un esempio potrebbe essere quello di richiedere che meno dell'1% dei campioni cambino di cluster.
Un' altra questione da affrontare è come calcolare la distanza tra i centroidi e un punto per poterlo assegnare al cluster corretto. Solitamente le strategie utilizzate sono le seguenti:

- distanza euclidea (d=$\sqrt{\sum_{j=1}^{n}(p_j - q_j)^2}$ con $P = (p_1, \ldots, p_n)$ e $Q = (q_1, \ldots, q_n)$ punti);
- distanza Manhattan (d= $|p_1 - q_1| + \cdots + |p_n - q_n|$ dove $P = (p_1, \ldots, p_n)$ e $Q = (q_1, \ldots, q_n)$ punti );
- misura di Jaccard (j= $\frac{|P \cup Q| - |P \cap Q|}{|P \cup Q|}$ con $P = (p_1, \ldots, p_n)$ e $Q = (q_1, \ldots, q_n)$ punti).

Si osservi che le prime due sono modi per calcolare propriamente le distanze mentre nell'ultimo caso quello che si calcola è la similarità tra due elementi.
Una volta che si sono ottenuti i vari cluster quello che risulta interessante è calcolare se effettivamente quello che si è trovato è una buona divisione del dataset. Se si suppone di calcolare le distanze con la distanza euclidea un ottimo modo per verificare la qualità del clustering è calcolare la somma degli errori quadratici che è anche chiamata *scatter* e che si trova nel modo seguente:

$$S = \sum_{i=1}^{k} \sum_{x \in C_i} dist(c_i, x)^2$$

dove $dist$ indica la distanza euclidea, $c_i$ è il centroide del cluster $i$ e $x$ rappresenta i punti del dataset. In pratica se l'errore che si trova è basso significa che i punti sono vicini al centroide e, quindi, si è ottenuto una buona divisione in cluster; in caso contrario,

invece, sarà opportuno modificare il numero di cluster o cambiare strategia. Inoltre, dalla formula appena introdotta discende che il centroide migliore si ha quando vale che

$$c_i = \frac{1}{m_i} \sum_{x \in C_i} x$$

ossia quando è la media di tutti gli oggetti del cluster $C_i$.
Un ultimo aspetto a cui è importante prestare attenzione è come selezionare i centroidi iniziali. Si è detto, infatti, che K-means solitamente sceglie randomicamente $k$ punti di partenza da considerare come centroidi. Tuttavia, soprattutto se il dataset è molto grande i tempi per trovare i cluster potrebbero essere lunghi e la qualità dei cluster non ideale. Per questo ci sono degli approcci alternativi come scegliere il primo centroide in modo randomico ed individuare i successivi $k - 1$ centroidi cercando i punti più lontani rispetto ai centroidi già individuati. In tale modo i centroidi iniziali sono ben separati tra loro. Il metodo appena descritto però non è comunque sempre l'ideale dato che rischia di selezionare punti in zone poco dense ed è dispendioso ricercare i punti più lontani. Un altro modo per selezionare i primi centroidi, se il dataset non è di grande dimensioni, è scegliere un ridotto numero di punti randomicamente e trovare i centroidi di quel sottoinsieme. Questi saranno i centroidi iniziali per dividere l'intero dataset.
Un'evoluzione dell' algoritmo K-means è il *bisecting K-means* in cui per ottenere i $k$ cluster si divide l'insieme di punti in due cluster, si sceglie uno dei due cluster e si reitera il procedimento fino a che non si sono individuati $k$ gruppi. Per scegliere quali dei due cluster dividere ci sono più strategie ma la modalità più utilizzata è quella che prevede di dividere il sottogruppo più grande. In generale si preferisce utilizzare questo metodo quando risulta problematico trovare i centroidi iniziali in quanto non è necessario scegliere subito tutti e $k$ i centroidi per poter iniziare l'algoritmo.
In ogni caso sia K-means che la sua variante sono ampiamente utilizzati perché sono semplice e applicabile ad un ampia tipologia di dataset. Inoltre, sono abbastanza efficienti anche quando il numero di iterazioni è ragionevolmente elevato. Allo stesso tempo, però, è necessario utilizzare altri metodi se non si vogliono cluster non-globulari o si hanno valori anomali che possono modificare di molto la posizione dei centroidi e, quindi, l'accuratezza del modello.

### 2.6.2 Clusterings gerarchici

La seconda tipologia di tecniche sono i clustering gerarchici che si dividono al loro interno in due classi differenti:

- **agglomerativi**: inizialmente questi algoritmi vedono ogni punto come un cluster ma nelle successive iterazioni si vanno ad accorpare i cluster più vicini;
- **divisivi**: in questo altro caso si comincia considerando tutti gli oggetti appartenenti ad un unico cluster e, in seguito, si dividono i cluster fino a che ogni singolo gruppo contiene un unico punto.

Tra le due tecniche la più nota e utilizzata è la prima e l'algoritmo che si segue è:

**Algoritmo 7.** *Clustering gerarchico agglomerativo*

1. *(Facoltativo: si calcoli preliminarmente le distanze tra tutte le coppie di punti e le si inseriscano nella matrice di prossimità.)*
2. ***repeat***
3. *Si fondino i due cluster più vicini.*
4. *Si aggiorni la matrice di prossimità calcolando la distanza tra i nuovi cluster.*
5. ***until*** *Rimane un unico cluster.*

Per trovare la prossimità dei vari cluster è possibile utilizzare una serie di modalità. La cosiddetta *MIN* definisce la prossimità tra i cluster come la distanza minima che si individua tra due punti non appartenenti allo stesso cluster. Si osservi che ancora una volta per calcolare la distanza si possono usare le formule già introdotte nella Sezione 2.6.1. Volendo usare *MAX*, invece, la prossimità si ottiene calcolando la distanza maggiore tra due punti di cluster differenti. Infine, la tecnica *group average* determina la prossimità dei cluster calcolando la distanza media tra tutte le coppie di punti che appartengono a due cluster diversi.
I vantaggi di questi algoritmi sono da ricercare nel fatto che non risulta problematico scegliere i punti iniziali e che, procedendo per passi, ogni iterazione non risulta dispendiosa dal punto di vista computazionale. Inoltre, nel caso in cui si vengano a creare cluster di differenti dimensioni esiste un'opzione che permette di risolvere la questione. É, infatti, possibile fare in modi che tutti i cluster siano trattati allo stesso modo e si parla perciò di *approccio non pesato* oppure si può tenere conto del numero di punti in ciascun gruppo e adottare dunque l'*approccio pesato*. Per quanto riguarda la qualità dei cluster numerosi studi hanno mostrato che generalmente i gruppi che si ottengono sono molto buoni. Tuttavia, i punti negativi di cui è necessario tenere conto se si sceglie di sfruttare tale algoritmo sono che, dal punto di vista computazionali, l'algoritmo è molto costo e che una volta che due cluster sono stati uniti non è possibile tornare al passo precedente se non si è soddisfatti di quanto trovato. Per tale motivo a volte viene applicato in prima analisi K-means e successivamente clustering gerarchico agglomerativo.

### 2.6.3 DBSCAN

Il DBSCAN [23] è uno dei più semplici algoritmi di density-based clustering e consente quindi di generare zone molto dense con attorno aree con pochissimi campioni. L'approccio più comune è il *center-based*, che prevede di stimare la densità di un particolare punto contando i punti all'interno di una circonferenza con un raggio stabilito $r$. Fatto ciò, è possibile dividere i punti all'interno di ciascuna circonferenza nel seguente modo:

- **core points**: sono punti in cui all'interno della circonferenza cade un numero di punti maggiori o uguali ad una soglia *MinP* decisa dall'utente;

- **border points**: sono punti che non sono core points ma che cadono comunque all'interno di una circoferenza di un altro punto;
- **noise points**: sono punti che non sono né di tipo border né core.

Ora che si sono introdotti questi concetti è possibile concentrarsi sul vero e proprio algoritmo:

**Algoritmo 8.** *DBSCAN*

1. *Si etichettino tutti i punti del dataset come punti core, border o noise.*
2. *Si eliminino i punti di tipo noise.*
3. *Si uniscano con un arco tutti le coppie di punti di tipo core che ricadono all'interno di una stessa circonferenza*
4. *Sia ogni componente connessa ottenuta un diverso cluster.*
5. *Si assegni ogni punto di tipo border ad uno dei cluster in cui è presente un punto di tipo core che ha distanza* $\leq r$.

Come si è già accennato i parametri $r$ e $MinP$ devono essere scelti dall'utente in modo adeguato. In generale, tuttavia, i parametri dipendono dalla densità del cluster e dalla distribuzione dei punti.
I vantaggi di questo algoritmo è che, costruendo density-based cluster, si evitano gli errori dati dai punti rumorosi ed è possibile creare cluster di dimensione e forma arbitraria. In tal senso è una valida alternativa al K-means; tuttavia DBSCAN dà problemi se i cluster hanno densità tanto diverse e se il dataset è di grandi dimensioni.

# Capitolo 3

# Machine Learning per l'analisi della Malaria

Questo capitolo è dedicato alla descrizione dei modelli utilizzati e dei risultati ottenuti a partire da un dataset contenenti i dati di bambini africani affetti da malaria fornito dall'Istituto Nazionale per le Malattie Infettive Lazzaro Spallanzani-IRCCS-Roma. Originariamente si è partiti da un dataset di 26236 soggetti che sono stati ricoverati in 6 differenti ospedali africani nelle città di Lambarene e Libreville (Repubblica Gabonese), Banjul (Gambia), Kumasi (Ghana), Kilifi (Kenya), Blantyre (Malawi) in un periodo compreso tra il 2000 e il 2005. Tuttavia, per alcune mancanze e problemi riscontrati in alcuni soggetti di cui si parlerà più approfonditamente in seguito, i casi che sono stati presi in esame sono 26035.
Le questioni che si sono approfondite sono molteplici; in primo luogo si è cercato di costruire un modello matematico capace di predire l'esito della malattia, ossia se il paziente è in grado di superare l'infezione dopo i giorni trascorsi in ospedale oppure si ha il decesso dello stesso.
Un'altra problematica affrontata ha riguardato l'analisi dei tassi di mortalità nelle diverse regioni. Se si considera l'intero insieme di dati, infatti, si trova che il $4,34\%$ non riesce a superare la malattia ma, se si vanno a considerare i bambini divisi per zone di provenienza, quello che si nota è che la mortalità in percentuale oscilla tra $1,38$ e $9,39$. Per tale motivo, dunque, si sono applicati gli algoritmi matematici al dataset con tutte le feature, al dataset senza i luoghi e a dei sottoinsieme del dataset iniziale creati selezionando solo i campioni di ogni singolo ospedale. Più precisamente, si sono analizzati i dati di 1810 pazienti provenienti da Lambarene, 1752 da Libreville, 3428 da Banjul, 6823 da Kumasi, 6876 da Kilifi e 5346 da Blantyre. Lo scopo di tale divisione, quindi, è individuare delle possibili differenze tra i vari luoghi che possano riflettere il diverso tasso di mortalità.
Successivamente, invece, ci si è concentrati solo nei casi in cui si era giunti al decesso del soggetto per identificare la presenza di particolari similitudini tra pazienti provenienti dalla stessa città o area geografica con tecniche di clustering.

## 3.1 Descrizione dei dati

In questa sezione si espongono brevemente i dati presi in esame descrivendone le caratteristiche e analizzando le tecniche di manipolazione adottate in ogni singolo caso. In particolare, per ciascuna feature si è fatto riferimento allo *SMAC Analysis Dataset Specifications and Codebook* fornito da David Wypij, Clariss Valim, Christopher Olola e Terrie Taylor. Le variabili che si possiedono per ogni paziente possono essere divise in 3 categorie:

1. generalità del paziente;
2. sintomi e condizione psico-fisica del soggetto al momento del ricovero;
3. analisi cliniche effettuati dopo il ricovero.

### 3.1.1 Generalità del paziente

A questa categoria si possono ascrivere i seguenti dati:

- **SITE_NUMBER**: questa variabile è volta a descrivere la regione di provenienza del soggetto. Come si è già detto, infatti, i dati sono stati raccolti in 6 città africane differenti. In tale occasione non sono stati ammessi valori mancanti e, di conseguenza, sono stati eliminati tutti i soggetti in cui tale informazione non era presente. Inoltre, dal momento che questo dato è di natura categorica (ad ogni luogo è associato un numero naturale) si è creato una matrice con 6 colonne corrispondenti a ciascuna città.
  Tale feature, come si è già anticipato, è stata analizzata con particolare riguardo in quanto si è notato una differenza di mortalità nei diversi ospedali in esame e sarebbe interessante capire da cosa è provocata.

- **AGE (in months)**: tale dato esprime l'età dei pazienti al momento del ricovero e i valori ammessi sono quelli compresi tra 0 e 180, ossia si vogliono prendere in esame solo i soggetti con un'età compresa tra 0 e 15 anni. Per tale motivo sono stati tolti tutti i pazienti con tale dato mancante o superiore al limite stabilito. Inoltre, sono stati ammessi solo valori interi e, quindi, si sono arrotondati per eccesso quei casi in cui la parte decimale era maggiore o uguale a 5, per difetto altrimenti.

- **SEX**: è una variabile binaria che vale 0 se il paziente è una femmina, 1 se è un maschio. In tal caso i dati mancanti erano minore dell'1% e sono stati sostituiti con la moda ottenuta dai dati disponibili.

- **WEIGHT (in kg)**: indica il peso del bambino al momento del ricovero. In questo caso si è adottato un modo per controllare se i valori raccolti sono plausibili calcolandolo lo *Z-score*. Si è quindi utilizzata la seguente formula:

$$Z = \frac{[(\frac{X}{M})^L - 1]}{L * S} \tag{3.1}$$

dove $X$ è il peso del bambino e $L \neq 0$. In particolare $L$, $M$ e $S$ sono dei valori che variano in base al sesso e all'età del bambino e la loro tabulazione è disponibile al sito del *CDC* (https://www.cdc.gov/growthcharts/zscore.htm), il centro per la prevenzione e il controllo delle malattie degli Stati Uniti. Se lo *Z-score* ottenuto è compreso tra -10 e +4 i pesi sono da considerarsi ammissibili, altrimenti si è proceduto a codificarli come dati mancanti. A questo punto ciascun valore mancante è stato sostituito con la media dei pesi dei soggetti con lo stesso sesso ed età.

- **DAYS IN HOSPITAL**: nel dataset originario sono stati forniti il giorno di ammissione in ospedale e il giorno di dimissione o decesso. Se la data di ammissione di un soggetto non era disponibile l'intero insieme di valori riferiti a quest'ultimo è stato eliminato mentre sono stati mantenuti i pazienti in cui non era presente la data di dimissione. Successivamente si è calcolato il numero di giorni in ospedale e si sono sostituiti i valori mancanti con la media. Si noti che normalmente la terapia prevista per il trattamento della malaria ha una durata di 3 giorni. Di conseguenza, pazienti con una permanenza superiore in ospedale dovrebbero aver presentato delle complicanze.

### 3.1.2 Sintomi e condizione psico-fisica del soggetto al momento del ricovero

A questa categoria si possono associare le seguenti feature:

- **ANYFITS**: è una variabile binaria pari a 1 se si sono verificati episodi convulsivi prima dell'ospedalizzazione, 0 altrimenti. Anche in questo caso i valori mancanti sono stati sostituiti con la moda.

- **VOMIT**: è una variabile binaria che vale 1 se il paziente ha vomitato prima dell'ammissione in ospedale, 0 altrimenti. I valori mancanti sono stati rimpiazzati con la moda.

- **TEMPERATURE**: è la temperatura corporea in gradi centigradi. Sono stati considerati ammissibili tutti i valori compresi tra 33 e 43 mentre sono stati valutati come dati mancanti quelli al di fuori di tale range. In seguito i valori non disponibili sono stati sostituiti con la media.

- **BMS** (Blantyre motor score): è un dato volto ad esprimere la capacità motoria del bambino. In particolare, si valuta la capacità di reagire ad uno stimolo doloroso. Il paziente, infatti, può non reagire affatto (score=0), può ritirare l'arto o la zona toccata (score=1) oppure può intervenire direttamente e tentare di togliere l'oggetto che provoca lo stimolo doloroso (score=2). I dati mancanti sono stati sostituiti con la moda.

- **BVS** (Blantyre verbal score): è una feature che descrive l'abilità del bambino di piangere o parlare normalmente (score=2). Se ciò non avviene può essere presente

un pianto anomalo (score=1) o non avere una risposta di alcun tipo (score=0). Al posto dei dati assenti si è inserito la moda.

- **BES** (Blantyre eye movements score): è una variabile binaria che vale 1 se il bambino è in grado di seguire un oggetto che si muove davanti a lui, 0 altrimenti. I dati mancanti sono state sostituiti con la moda.

- **BCS** (Blantyre total coma score): è una modificazione della Glasgow coma scale, scala di valutazione neurologica sfruttata per tenere traccia dei miglioramenti o peggioramenti di pazienti in coma. La BCS, invece, è stata sviluppata specificatamente per i bambini affetti da malaria dai dottori Terrie Taylor and Malcolm Molyneux e si ottiene dalla somma dei punteggi ottenuti da BMS, BVS e BES. Solo tale feature è stata inserita nei vari dataset mentre BMS, BVS e BES sono state utilizzate per controllare che effettivamente BCS corrispondesse alla somma delle 3 componenti che si prendono in considerazione per definire tale score.

- **RESPIRATORY_RATE**: tale indicatore esprime le ventilazioni al minuto. Questo valore cambia in base all'età e, in particolare, tende ad essere più elevato nei bambini. Nel caso della malaria, il parametro in questione può contribuire a monitorare eventuali miglioramenti o peggioramenti dal momento che tende ad aumentare in caso di febbre e malattie. Tenendo conto che i soggetti di questa analisi hanno tutti un'età inferiore a 15 anni, sono stati considerati come dati mancanti le misure maggiori di 66 o minore di 6. Infine, è stata fatta la media e inserita al posto dei dati non disponibili.

- **DEEP_BREATHING**: è una variabile binaria che è pari a 1 se il malato risulta essere in iperventilazione. Il numero di respiri al minuti, dunque, risultano maggiori e più faticosi rispetto ad una persona in salute e, solitamente, questo indica un'infezione alle vie respiratorie o presenza di problemi cardiaci. Anche in questo caso con la moda si è risolto il problema legato ai valori non disponibili.

- **INTERCOSTAL_RECESSIONS**: è un indicatore del manifestarsi di problemi respiratori. In particolare, se vale 1 significa che il paziente manifesta una rigidità toracica che rende difficile contrarre e rilassare il diaframma ed è una condizione che si può manifestare soprattutto tra i neonati o i bambini piccoli. Se, al contrario, questa variabile è a 0 tale complicazione non sussiste. I dati mancanti sono stati sostituiti con la moda.

- **IRREGULAR_BREATHING**: è una variabile binaria che vale 1 se il paziente presenta un'alterazione della respirazione, 0 altrimenti. In generale tale dato può rappresentare la presenza o meno di un sintomo tipico della patologia in corso. I valori mancanti sono stati sostituiti con la moda.

- **SUCK***: è un dato che esprime la capacità del bambino di succhiare il latte o altre bevande. Tuttavia, tale aspetto non riguarda tutti i pazienti ma solo chi è nella fascia d'età più bassa. Per tale ragione si è codificato con *NA (not applicable)* i

casi non pertinenti. Dal momento che il parametro è categorico si sono create le colonne suck_no, suck_yes, suck_NA e i dati mancanti sono stati sostituiti con la moda.

- **SIT***: con questa feature si vuole indicare l'abilità del bambino di stare seduto se quest'ultimo era in grado di farlo prima della malattia. Infatti, bambini con particolari condizioni fisiche o neonati avranno questo valore segnato come *NA*. Si è quindi diviso il dato in sit_no, sit_yes, sit_NA ed eliminato i valori mancanti inserendo la moda.

- **STAND***: questo dato indica la capacità dei pazienti di stare in posizione eretta. Se il paziente è troppo piccolo o in presenza di deficit motori il valore è contrassegnata con *NA*. La variabile è perciò categorica e le tre colonne interessate sono stand_no, stand_yes, stand_NA. I missing data sono stati sostituiti con la moda.

- **WALK***: è un'indicatore dell'abilità del bambino di camminare se in grado di fare ciò anche prima della malattia. Se tale dato non è conforme con un determinato soggetto tale valore è segnato come *NA*. Le colonne associate sono quindi walk_no, walk_yes, walk_NA e i dati mancanti sono stati sostituiti con la moda.

### 3.1.3 Analisi cliniche effettuati dopo il ricovero

In ospedale si sono raccolti informazioni relative alle seguenti feature:

- **SPLEEN (in cm)**: è la misura della milza in centimetri ottenuta con la palpazione dell'area da parte dei medici. I valori ammessi sono quelli compresi tra 0 a 15 cm. Tuttavia, la grandezza della milza dipende anche dall'età del bambino. Si è quindi controllato che i casi in cui la milza fosse maggiore di 12 cm appartenessero a pazienti con età non inferiore a 10 anni. Infine, i dati mancanti sono stati sostituiti dalla media ottenuta considerando i pazienti dello stesso sesso ed età.

- **TRANSFUSED**: è una variabile binaria che vale 1 se il paziente ha avuto necessità di trasfusioni prima del ricovero ospedaliero, 0 altrimenti. I valori mancanti sono stati eliminati inserendo la moda.

- **PARASITEMIA**: è un valore ottenuto misurando la quantità media di parassiti nel sangue. In particolare, se tale dato vale 0 significa che il paziente non è affetto da malaria e per questo motivo si sono eliminati tutti i soggetti che presentavano codesta caratteristica. Inoltre, dal momento che risulta essere un'informazione estremamente importante si sono tolti anche i soggetti con questo missing data e i casi in cui la parassitemia era maggiore di 250000.

- **HCT**: sta ad indicare l'ematocrito, cioè il rapporto tra il plasma ed elementi figurati del sangue (globuli rossi, globuli bianchi, piastrine). I valori considerati ammissibili sono quelli compresi tra 3 e 60 e, di conseguenza, si è proceduto a codificare come mancanti i dati fuori da questo range. Infine, è stata fatta la media per inserirla dove l'informazione sull'ematocrito non era presente.

- **GLUCOSE**: è l'indicatore del glucosio presente nel sangue misurato in mmol/l. L'intervallo considerato valido è quello tra 0 e 20 e, perciò, i dati che non rispettavano questo vincolo sono stati eliminati. Inoltre, nella maggior parte dei casi il glucosio è stato calcolato utilizzando il glucometro mentre in alcuni pazienti del Gambia sono state sfruttate le Dextrostix, delle strisce per determinare il livello di glucosio. Per uniformare le informazioni sono stati cancellati i valori ottenuti con questo secondo metodo. Infine la media ha rimpiazzato i missing data.

- **LACTATE**: questo valore permette di conteggiare i lattati presenti nel sangue misurato in mmol/l. Il range che è stato considerato ammissibile è quello compreso tra 0 e 20. In caso contrario si è proceduto a codificare i dati come missing e a sostituire tutti i mancanti con la media.

- **LACTANAL**: tale termine sta per *Lactate Analox indicator variable* e sta ad indicare la modalità con cui sono stati raccolti i dati sui lattati. Infatti, vale 1 se i lattati sono calcolati con l'Analox machine e 0 se misurato con LActate Pro o YSI machines. I missing data sono stati rimpiazzati dalla moda.

- **MONOL200**: la sigla è l'abbreviazione di *pigmented mononuclear leukocytes counted in 200 cells*, ossia il numero di leucociti mononucleati che ci sono se si considerano 200 cellule di un campione di plasma. Essi rappresentano una delle prime forme di difesa per combattere un'infezione. Viene, inoltre, sottolineato che è necessario applicare una colorazione per analizzare il campione perché altrimenti tali cellule non sarebbero visibili. Sono stati considerati ammissibili solo i valori compresi tra 0 e 200 e, in caso contrario, si è provveduto a considerarli come mancanti. Inoltre, in alcuni pazienti di Libreville e Lambarene questo tipo di analisi è stata svolta su 100 cellule e non su 200 e, per tale motivo si è provveduto a moltiplicare tale informazioni per 2 in modo da omogenizzare i dati. Infine, la media ha sostituito i valori mancanti.

- **POLY200**: il termine indica i *pigmented polymorphonuclear leukocytes counted in 200 cells* e, come nel caso precedente è un indicatore del numero di leucociti presenti tra 200 cellule. In questo frangente, però, si prendono in considerazione i leucociti nucleopolimorfi, anche detti granulociti per la loro forma. Solitamente la gran parte di queste cellule sono globuli bianchi del tipo neutrofili. Il range considerato accettabile è stato tra 0 e 200 mentre i valori mancanti o fuori dal limite sono stati modificati con la media.

- **PARBC200**: è il dato che indica il numero di parassiti del tipo *P. falciparum* presenti in 200 cellule. I valori ammessi sono tra 0 e 200 e i dati mancanti sono stati sostituiti con la media. Come si è detto nel primo capitolo questa specie di parassita risulta essere il più fatale per l'uomo e, per tale motivo, è interessante valutare se effettivamente i pazienti analizzati hanno in corso un'infezione malarica causata da *P. falciparum* o al contrario, la causa è da imputara ad un'altra specie.

Il dataset di partenza presentava anche molte altre feature ma per non compromettere i risultati finali si è scelto di eliminare dal dataset tutte le feature con una percentuale di valori mancanti superiore al 10%. Oltre alle feature eliminate per il motivo appena descritto, è giusto sottolineare che in una prima fase si era scelto di utilizzare il dataset che contenevano anche BMS, BVS, BES e dati volti ad indicare da dove era stato prelevato il tessuto ematico utilizzato per ciascun esame clinico. Tuttavia, quello che si è notato è che queste feature erano ridondanti e motivo di disturbo.
Inoltre, nell'elenco appena introdotto si sono sottolineate alcune feature relative alla capacità motorie dei singoli pazienti. Tale tipologia di informazioni, se da un lato possono risultare interessanti per capire come i sintomi visibili dell'infezione siano in relazione con gli esami clinici, dall'altro possono distogliere l'attenzione dai valori ottenuti con gli esami di laboratorio. Di conseguenza, si è scelto di procedere con due modalità diverse in quanto per ciascun luogo sopra indicato e per l'intero insieme di campioni con e senza luoghi si sono analizzate di volta in volta un dataset provvisto di feature motorie e un dataset privato di queste 4 colonne in modo da poter confrontare i risultati ottenuti.
Infine, prima di proseguire con la costruzione del modello si è proceduto alla standardizzazione dei dati per i motivi di cui si è già parlato nel capitolo precedente.

## 3.2 Costruzione di Classificatori per la Previsione dell'esito della malaria

Una volta terminata la preparazione del dataset si è passati all'analisi dei dati utilizzando la libreria di python **sklearn** (https:// scikit-learn.org/ stable/) e il software **LIBSVM** ( https://www.csie.ntu.edu.tw/ cjlin/libsvm/). Per ottenere grafici e diagrammi si è, invece, sfruttato il pacchetto di python **matplotlib** (https://matplotlib.org/).
Prima di iniziare ad applicare il primo modello di apprendimento supervisionato, tuttavia, è stato necessario riflettere su quali tecniche adottare per costruire un buon modello predittivo a partire da un dataset fortemente sbilanciato. Infatti, i casi di decessi presenti nel dataset sono 1130 contro i rimanenti 24905 pazienti che hanno superato con successo l'infezione malarica. Di conseguenza, si sono scelte tre differenti tecniche per poter risolvere tale disparità tra le due classi. In particolare, in un primo momento si è utilizzato l'algoritmo *SMOTE* tentando però due approcci differenti. Infatti, in un caso si è ricampionato l'intero dataset per poi applicare tecniche di apprendimento supervisionato mentre nell'altro si è provato a fare oversampling all'interno del processo di cross validation ricampionando solo il training set. Successivamente si sono, invece, tentate due tecniche di undersampling con il *Random Undersampling Method* e il *NearMiss*. In quest'ultimo caso sono disponibili tre modalità differenti per selezionare i punti della classe maggiormente rappresentata utili per ribilanciare il dataset. Per determinare quale tra le differenti opzioni performasse meglio con i dati in questione si è inizialmente considerato tutte e tre le modalità fino a che si è notato che la maggior capacità predittiva si otteneva scegliendo la prima versione, ossia quella che seleziona i $k$ punti che mediamente sono più vicini alla classe meno rappresentata. Di conseguenza, in seguito si è proceduto ad applicare gli algoritmi con *NearMiss versione 1.*

Si osservi che tali tecniche per ripristinare l'equilibrio tra le due classi non sono state utilizzate solo per il dataset completo ma anche per i dataset parziali in cui si tiene conto del luogo di provenienza dei soggetti.
A questo punto si è applicato SVM nei 3 nuovi dataset creati con le tecniche di ribilanciamento del dataset utilizzate e al dataset di partenza ribilanciando solo il training set con SMOTE. Tale modello lascia libero l'utente di selezionare varie opzioni. In primo luogo, è decisivo scegliere il kernel che si intende utilizzare. Nel caso in questione si sono fatti dei tentativi con un kernel lineare per poi privilegiare il kernel RBF per la capacità predittiva altamente migliore ottenuta. Tuttavia, scegliendo questa tipologia di kernel è indispensabile determinare i due iperparametri $C$ e $\gamma$. Per trovare la migliore accuratezza possibile è, quindi, necessario passare alla fase detta di *Model Selection* in cui si costruisce una griglia bidimensionale o *grid search* per selezionare gli iperparametri. Più in particolare, quello che accade è che a partire da 2 liste rispettivamente con i valori ammessi per $C$ e per $\gamma$ l'algoritmo di grid search individua la coppia $(C, \gamma)$ che meglio predice la classe di appartenenza dei campioni del dataset utilizzato come training test.
In seguito, si è utilizzato anche Random Forest come altro algoritmo di classificazione. In tal caso si è comunque passati per la fase di *Model Selection* per determinare ben 4 iperparametri. In primis si è creata una lista per poter selezionare tra una serie di valori la *profondità massima* che ciascun albero della foresta può avere. Inoltre, si è scelto di poter determinare il numero massimo di feature di cui tenere conto per effettuare ciascuno *split* e il numero minimo di punti che ciascun *nodo foglia* può avere. Infine, anziché lasciare che l'algoritmo crei di default 100 alberi si è creata una lista per poter trovare la quantità di alberi che nel caso specifico meglio predicono gli outcome senza rischiare di andare in overfitting. Successivamente Random Forest è stato sfruttato anche per la selezione delle feature. In particolare, se non si specifica in quale modo calcolare lo score da assegnare a ciascuna feature, Random Forest utilizza la formula di Gini introdotta nel precedente capitolo.
Nella sezione successiva, si analizzeranno i risultato ottenuti con i diversi dataset ribilanciati nelle fasi iniziali con SMOTE, Random Undersampling e NearMiss mentre nella sezione 3.2.9 si concentrà l'attenzione sui modelli costruiti e le relative capacità predittive applicando SMOTE all'interno del processo di cross validation.

### 3.2.1 Dataset completo

Nel caso dei 2 dataset completi con e senza le variabile relative alle capacità motorie del singolo soggetto le due classi inizialmente erano distribuite come nel grafico della Figura (3.1). Dopo aver applicato l'algoritmo SMOTE quello che si è ottenuto è un dataset con le classi bilanciate come si può vedere sulla destra dell'immagine.

A questo punto si è applicato SVM in entrambi i dataset. Tenendo conto anche delle capacità motorie i parametri selezionati sono stati $C = 2$ e $\gamma = 0.25$ con una cross validation accuracy pari a 97.8%. Si può, quindi, notare che anche senza alcun metodo di selezione delle feature il modello scelto sembra dare una buona capacità predittiva.
Nel secondo caso, invece, si ha un'accuratezza leggermente peggiore dato che con $C = 4$ e $\gamma = 0.5$ si ottiene un valore di 96.2%.

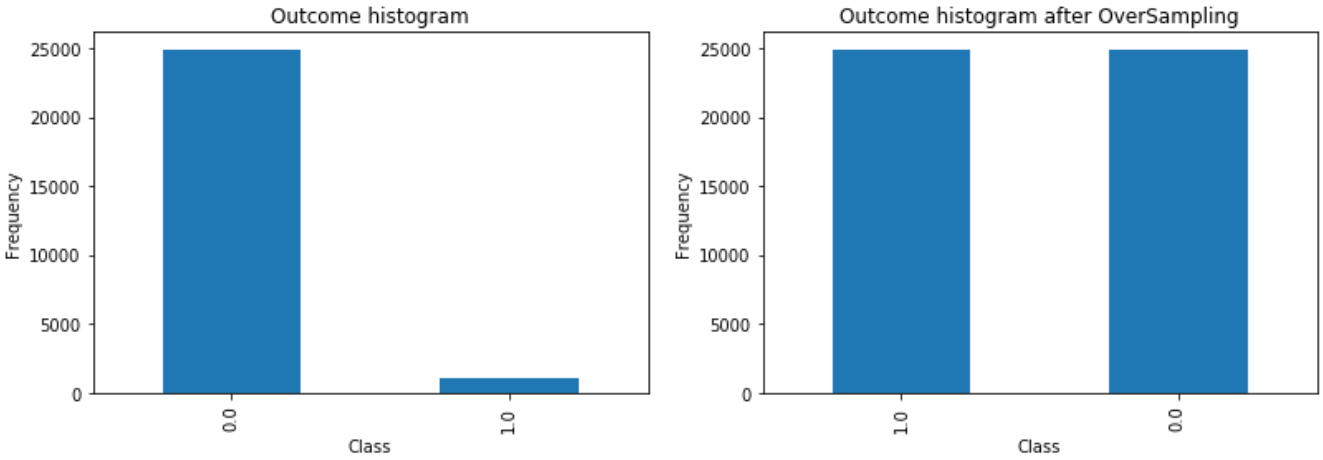



Figura 3.1: Distribuzione dei campioni appartenenti alle due classi prima e dopo SMOTE

Nel caso di Random Forest, invece, con il primo dataset i parametri scelti dopo aver fatto grid search sono $max\_depth = 80$ che indica la massima profondità degli alberi, $max\_feature = 2$ che è il numero massimo di feature considerate per effettuare lo split, $min\_sample\_leaf = 3$ che fa in modo che in ciascun nodo foglia ci siano almeno 3 campioni e, infine, $n\_estimators = 400$ che è il numero massimo di alberi della foresta. L'accuratezza ottenuta in questo caso è del 87.6%.
Se, invece, si applica Random Forest al dataset che non considera le abilità motorie si ottiene una cross validation accuracy di 81.2% se $max\_depth = 100$, $max\_feature = 3$, $min\_samples\_leaf = 3$ e $n\_estimators = 300$.
Si è poi proceduto a ribilanciare il dataset con Random Undersampling e, quindi, in questo caso si sono eliminati randomicamente i campioni della classe maggiormente rappresentata fino ad arrivare a 1130, valore che corrisponde al numero totale di pazienti deceduti. Si ricordi che tale metodo è il meno sofisticato tra i tre scelti per ribilanciare le classi e, quindi, è naturale aspettarsi dei risultati leggermente peggiori. In particolare, con SVM la cross validation accuracy che si è trovata con il dataset con tutte le feature è di 66.9% con $C = 2$ e $\gamma = 0.125$. Con Random Forest l'accuratezza è pari a 73.9% dopo aver fatto grid search e scelto come valori $max\_depth = 90$, $max\_feature = 3$, $min\_samples\_leaf = 5$ e $n\_estimators = 200$.
Se si considera il dataset ridotto, invece, SVM offre un'accuratezza di 68.8% se $C = 2$ e $\gamma = 0.25$ mentre Random Forest con $max\_depth = 90$, $max\_feature = 3$, $min\_samples\_leaf = 4$ e $n\_estimators = 400$ dà un'accuratezza di 71.6%.
Utilizzando NearMiss, al contrario, i risultati sono tornati ad essere piuttosto buoni dato che con il dataset con tutte le feature l'accuratezza con SVM è di 91.3% se $C = 2$ e $gamma = 0.125$ mentre con Random Forest ($max\_depth = 90$, $max\_feature = 3$, $min\_samples\_leaf = 3$ e $n\_estimators = 400$) si è ottenuto 88.8%.
Con il dataset con le feature ridotte NearMiss ha permesso di ottenere un'accuratezza di 93.4% se si applica SVM con $C = 2$ e $\gamma = 0.25$ e di 91.6% con Random Forest e gli

iperparametri settati a $max\_depth = 80$, $max\_feature = 3$, $min\_samples\_leaf = 3$ e $n\_estimators = 100$.

### 3.2.2 Dataset senza luoghi

Come si è anticipato si è scelto di analizzare il dataset eliminando le feature corrispondenti ai luoghi di provenienza dei pazienti per individuare eventuali differenze con il dataset completo di cui si è appena parlato e tentare di capire le ragione che portano ad avere diversi tassi di mortalità. Si è, quindi, ripetuto quanto visto per il caso precedente sia prendendo in considerazione le feature legate al movimento sia eliminandole.
Ribilanciamento il dataset con anche le feature *sit, walk, stand* e *suck* con l'algoritmo SMOTE e applicando SVM con gli iperparametri settati a $C = 4$ e $\gamma = 0.25$ ottenuti con la grid search si è potuto costatare che la cross validation accuracy è pari a 99.1%. Con Random Forest, invece, si è ottenuto un valore di accuratezza pari a 98.0% se $max\_depth = 80$, $max\_feature = 3$, $min\_samples\_leaf = 3$ e $n\_estimators = 300$.
Si è poi proceduto questa fase di analisi utilizzando Random Undersampling e, successivamente, NearMiss in alternativa a SMOTE per ottenere un equilibrio tra le due classi del dataset.
Con Random Undersampling quello che si è osservato è che SVM sembra dare risultati meno soddisfacenti dal momento che con $C = 1$ e $\gamma = 0.0625$ si ha la cross validation accuracy pari a 76.5% mentre Random Forest con un'accuratezza di 80.0% in questo caso performa meglio rispetto a SVM.
NearMiss, invece, sembra essere migliore di Random Undersampling ma peggiore di SMOTE. Inver, con Random Forest ($max\_depth = 80$, $max\_feature = 3$, $min\_samples\_leaf = 3$ e $n\_estimators = 200$) si ha la cross validation accuracy pari a 90.8% mentre con SVM se $C = 1$ e $\gamma = 0.125$ l'accuratezza è 91.9%.
Se si analizza il dataset privato delle feature prima sottolineate con SMOTE si ha comunque una buona accuratezza dal momento che con SVM si ottiene un valore di 98.7% ($C = 2$, $\gamma = 0.25$) mentre con Random Forest si ha una cross validation accuracy pari a 97.4% ($max\_depth = 80$, $max\_feature = 3$, $min\_samples\_leaf = 3$ e $n\_estimators = 400$). Con Random Undersampling, invece, SVM permette di avere un'accuratezza di 76.5% ($C = 4$, $\gamma = 0.06250$) e Random Forest di 81.1% ( $max\_depth = 100$, $max\_feature = 3$, $min\_samples\_leaf = 4$ e $n\_estimators = 400$). Infine, con NearMiss l'accuratezza è pari a 95.0% ($C = 2$, $\gamma = 0.5$) con SVM mentre si ha una cross validation accuracy di 94.6% ($max\_depth = 90$, $max\_feature = 2$, $min\_samples\_leaf = 3$ e $n\_estimators = 200$) con Random Forest.

### 3.2.3 Dataset con pazienti da Lambarene (Tasso di mortalità: $1,38\%$)

I pazienti appartenenti a questo luogo sono 1810 con 25 decessi.
Si analizzi innanzitutto il dataset con anche le feature relative alla capacità motoria. Dopo aver applicato l'algoritmo SMOTE gli iperparametri selezionati facendo grid search per SVM sono $C = 2$ e $\gamma = 0.25$ e la cross validation accuracy è pari a 99.9%.
Nel caso di Random Forest, invece, i parametri selezionati sono $max\_depth = 100$,

$max\_feature = 3$, $min\_samples\_leaf = 3$ e $n\_estimators = 100$ mentre l'accuratezza è comunque piuttosto buona dato che è pari a 99.3%.
Risultati leggermente peggiori si sono ottenuti con le tecniche di undersampling ma questo è dovuto al fatto che il numero di campioni appartenenti alla classe meno rappresentata sono molto pochi.
Nel caso di Random Undersampling con SVM si è selezionato $C = 2$ e $\gamma = 0.25$ con un'accuratezza del 70% mentre con Random Forest gli iperparametri scelti sono stati $max\_depth = 90$, $max\_feature = 3$, $min\_samples\_leaf = 4$ e $n\_estimators = 200$ con un'accuracy del 86%.
Con NearMiss, infine, per SVM si è selezionato $C = 2$ e $\gamma = 0.125$ con l'accuratezza di 84% mentre con Random Forest con $max\_depth = 80$, $max\_feature = 3$, $min\_samples\_leaf = 3$ e $n\_estimators = 200$ si ottiene un'accuracy di 82%.
Se, invece, si considera il dataset con le feature ridotte quello che si osserva applicando SMOTE è un elevatissima capacità predittiva con entrambi i metodi. In particolare con SVM si ottiene una cross validation accuracy pari a 99.9% ($C = 2$, $\gamma = 0.25$) mentre con Random Forest si ha un'accuratezza del 99.6% ($max\_depth = 100$, $max\_feature = 2$, $min\_samples\_leaf = 3$, $n\_estimators = 300$).
Anche con questo dataset le performance ottenute ribilanciando tramite Random Undersampling non sono buone quanto gli altri due metodi. Infatti l'accuratezza è pari a 70% per SVM e 82% per Random Forest. Infine, si è ribilanciato il tutto con NearMiss per poi applicare SVM che, in questo caso, ha restituito un'accuratezza dell'84% ($C = 2$, $\gamma = 0.25$). Con Random Forest si ha, invece, cross validation accuracy pari a 88% ($max\_depth = 100$, $max\_feature = 2$, $min\_samples\_leaf = 4$, $n\_estimators = 100$).

### 3.2.4 Dataset con pazienti da Libreville (Tasso di mortalità: 5.08%)

I pazienti di Libreville presi in esame sono 1752 con 89 casi di decesso. Come fatto in precedenza, si è prima utilizzato il dataset con anche le feature relative alla capacità motoria.
Si è applicato, quindi, l'algoritmo SMOTE per ribilanciare il dataset e si è poi applicato SVM ottenendo un' accuratezza di 99.7% nel caso in cui $C = 4$ e $\gamma = 0.125$. Con Random Forest, invece, la cross validation accuracy che si è ottenuta è 98.2% se $max\_depth = 90$, $max\_feature = 3$, $min\_samples\_leaf = 3$ e $n\_estimators = 100$.
Per quanto riguarda i metodi di undersampling anche in tal caso i risultati ottenuti sono peggiori rispetto a quelli di SMOTE.
Invero, nel caso di Random Undersampling l'accuratezza con SVM è del 82.0% con $C = 2$ e $\gamma = 0.0625$ mentre con Random Forest si ha l'accuratezza pari a 87.1% con $max\_depth = 80$, $max\_feature = 3$, $min\_samples\_leaf = 4$ e $n\_estimators = 200$.
Con NearMiss nel caso di SVM la cross validation accuracy è pari al 89.3% se $C = 4$ e $\gamma = 0.03125$, valore molto vicino a quello ottenuto con Random Forest dal momento che in questo secondo caso l'accuratezza è del 91.0% con gli iperparametri $max\_depth = 100$, $max\_feature = 2$, $min\_samples\_leaf = 3$ e $n\_estimators = 400$.
Se si considera il dataset con le feature ridotte e si ribilancia l'algoritmo SMOTE con

SVM si ottiene un'accuratezza di 99.6% ($C = 4$, $\gamma = 0.25$). Con Random Forest, invece, la cross validation accuracy è 98.2% ($max\_depth = 100$, $max\_feature = 2$, $min\_samples\_leaf = 3$ e $n\_estimators = 300$).
Se si lavora con il dataset ridotto i risultati che si ottengono sono piuttosto in linea con i precedenti. Infatti, con SMOTE SVM restituisce un'accuratezza di 99.6% ($C = 4$, $\gamma = 0.25$) mentre Random Forest ha una capacità predittiva di 98.2% ($max\_depth = 100$, $max\_feature = 2$, $min\_samples\_leaf = 3$, $n\_estimators = 300$).
Se si ribilancia il dataset con Random Undersampling si può notare che SVM restituisce un'accuratezza di 79.2% ($C = 2$, 0.0625) mentre la capacità predittiva di Random Forest risulta essere migliore in questo caso dato che la cross validation accuracy è pari a 89.3% ($max\_depth = 80$, $max\_feature = 3$, $min\_samples\_leaf = 3$ e $n\_estimators = 400$).
Infine, con NearMiss SVM e Random Forest danno un'accuratezza entrambe superiore al 90% (SVM: 91.0%, Random Forest: 93.8%).

### 3.2.5 Dataset con pazienti da Banjul (Tasso di mortalità: 9.39%)

I casi di bambini con malaria raccolti in questo luogo sono 3428 con 322 decessi che fanno sì che questo sia l'ospedale dove si è registrata un maggior tasso di mortalità.
Anche in questo terzo caso se si utilizza il dataset con le feature sulla capacità motoria l'algoritmo SMOTE permette di ottenere modelli con altissime capacità predittive in quanto si ha con SVM una cross validation accuracy di 98.8% se $C = 4$ e $\gamma = 0.25$. Utilizzando Random Forest, invece, l'accuratezza è pari a 97.3% con $max\_depth = 90$, $max\_feature = 3$, $min\_samples\_leaf = 3$ e $n\_estimators = 100$.
Se si ribilancia il dataset con Random Undersampling si nota una sostanziale differenza tra la capacità predittiva di SVM e Random Forest. Nel primo caso, infatti, si ha un valore di accuratezza pari a 75.3% ($C = 2$, $\gamma = 0.0625$) mentre con Random Forest si arriva a 86.5% ($max\_depth = 90$, $max\_feature = 3$, $min\_samples\_leaf = 3$, $n\_estimators = 200$).
Con NearMiss si osserva una situazione analoga a Random Undersampling dato che Random Forest performa meglio che SVM. Infatti, con il primo algoritmo si ha la cross validation accuracy pari a 92.2% ($max\_depth = 80$, $max\_feature = 3$, $min\_samples\_leaf = 3$, $n\_estimators = 400$) mentre con SVM si arriva solo a 88.0% ($C = 2$, $\gamma = 0.03125$).
Si prenda in considerazione il dataset con le feature ridotte ribilanciato con SMOTE. Se si applica SVM la cross validation accuracy che si individua è 98.6% ($C = 4$, $\gamma = 0.25$) mentre con Random Forest si ottiene un valore di 96.5% ($max\_depth = 90$, $max\_feature = 2$, $min\_samples\_leaf = 3$, $n\_estimators = 400$).
Se si ribilancia il dataset con Random Undersampling e si applicano poi rispettivamente SVM e Random Forest, quello che si può notare è che il secondo performa meglio del primo (SVM: 77.6% con $C = 4$ e $\gamma = 0.0625$, Random Forest: 88.2% con $max\_depth = 90$, $max\_feature = 3$, $min\_samples\_leaf = 3$ e $n\_estimators = 100$).
Infine, con NearMiss si ha che SVM restituisce un'accuratezza di 94.3% ($C = 2$, $\gamma = 0.125$) e Random Forest, invece, di 93.2% ($max\_depth = 100$, $max\_feature = 2$, $min\_samples\_leaf = 3$ e $n\_estimators = 300$).

### 3.2.6 Dataset con pazienti da Kumasi (Tasso di mortalità: 4.56%)

I campioni che sono stati raccolti a Kumasi sono ben 6823 con un totale di 311 decessi. Il buon numero di dati a disposizione ha fatto sì che ribilanciando il dataset comprendente anche le feature relative alla capacità motoria con SMOTE si ottenesse una cross validation accuracy pari a 99.5% con $C = 4$ e $\gamma = 0.125$. Random Forest, invece, ha permesso di dare un'accuratezza del 98.7% ($max\_depth = 100$, $max\_feature = 3$, $min\_samples\_leaf = 3$ e $n\_estimators = 300$).
Nel caso dei metodo di undersampling accade qualcosa di simile a quanto visto con i dati di Banjul. Infatti, NearMiss è molto più performante di Random Undersampling sia sfruttando SVM sia Random Forest. Più precisamente nel caso di SVM si ha una cross validation accuracy di 90.1% ($C = 2$, $\gamma = 0.03125$) per NearMiss contro i 74.9% ($C = 2$, $\gamma = 0.0625$). Con Random Forest si ha 90.8% di accuratezza nel primo caso ($max\_depth = 80$, $max\_feature = 3$, $min\_samples\_leaf = 3$ e $n\_estimators = 400$) e 74.9% nell'altra situazione ($max\_depth = 80$, $max\_feature = 2$, $min\_samples\_leaf = 3$ e $n\_estimators = 200$).
Se ci si concentra ad analizzare il dataset con le feature ridotte emerge che SMOTE permette di costruire un modello con un'altissima capacità predittiva. Infatti con SVM si ha una cross validation accuracy di 99.4% ($C = 8$, $\gamma = 0.25$) mentre con Random Forest si ottiene 98.6% ($max\_depth = 80$, $max\_feature = 3$, $min\_samples\_leaf = 3$ e $n\_estimators = 200$).
Nuovamente Random Undersampling sembra essere l'algoritmo che restituisce i risultati meno soddisfacenti (SVM: 77.7% con $C = 2$ e $\gamma = 0.0625$, Random Forest: 83.4% con $max\_depth = 100$, $max\_feature = 2$, $min\_samples\_leaf = 3$, $n\_estimators = 100$).
Infine, con NearMiss SVM e Random Forest danno quasi lo stesso grado di accuratezza. Infatti, con SVM la cross validation accuracy è 94.1% ($C = 2$, $\gamma = 0.25$) e con Random Forest si ha 94.2% ($max\_depth = 80$, $max\_feature = 2$, $min\_samples\_leaf = 3$, $n\_estimators = 200$).

### 3.2.7 Dataset con pazienti da Kilifi (Tasso di mortalità: 3.61%)

A Kilifi i dati raccolti sono relativi a 6876 bambini con infezione malarica di cui 248 che non sono riusciti a sopravvivere.
Dopo aver ribilanciato il dataset comprendente anche le feature relative alla capacità motoria con SMOTE si è applicato SVM con $C = 16$ e $\gamma = 0.25$ ottenendo un valore pari a 99.5% di cross validation accuracy. Calcolando l'accuratezza con Random Forest si è trovato il valore percentuale di 98.3% con $max\_depth = 80$, $max\_feature = 3$, $min\_samples\_leaf = 3$ e $n\_estimators = 300$.
Per l'ennesima volta la strategie che sembra essere la peggiore tra le tre è Random Undersampling poiché con SVM si arriva ad un'accuratezza del 75.4% con $C = 2$ e $\gamma = 0.0625$. Se si applica Random Forest la cross validation accuracy è 80.2% ($max\_depth = 80$, $max\_feature = 3$, $min\_samples\_leaf = 4$ e $n\_estimators = 200$).
Con NearMiss, SVM permette di avere un'accuratezza del 95.8% con gli iperparametri $C = 4$ e $\gamma = 0.03125$ ottenuti con la fase di selezione del modello. Random Fore-

st, infine, restituisce un'accuratezza di 93.5% ($max\_depth = 80$, $max\_feature = 2$, $min\_samples\_leaf = 3$ e $n\_estimators = 100$).
Ribilanciando il dataset senza le 4 feature di cui si è prima parlato con SMOTE SVM dà una capacità predittiva di 99.31% ($C = 4$, $\gamma = 0.25$) mentre Random Forest di 98.7% ($max\_depth = 80$, $max\_feature = 3$, $min\_samples\_leaf = 3$, $n\_estimators = 400$).
Se si sceglie di riequilibrare le classi con Random Undersampling si trova che SVM restituisce l'accuratezza di 76.4% ($C = 2$, $\gamma = 0.0625$) mentre con Random Forest il valore che si ricava è 81.7% ($max\_depth = 80$, $max\_feature = 3$, $min\_samples\_leaf = 3$, $n\_estimators = 200$).
In ultima analisi con NearMiss si è ottenuta un'accuratezza di 94.4% ($C = 2$, $\gamma = 0.25$) con SVM e di 93.5% ($max\_depth = 80$, $max\_feature = 2$, $min\_samples\_leaf = 3$, $n\_estimators = 200$) con Random Forest.

### 3.2.8 Dataset con pazienti da Blantyre

A Blantyre dei 5346 casi disponibili si contano 135 bambini che non sono riusciti a superare l'infezione.
Facendo oversampling con il dataset con anche le feature che esprimono le capacità motorie e applicando SVM l'accuratezza ottenuta è pari a 99.6% se si scelgono $C = 4$ e $\gamma = 0.125$. Con Random Forest il valore scende leggermente a 99.0% con $max\_depth = 80$, $max\_feature = 3$, $min\_samples\_leaf = 3$ e $n\_estimators = 300$.
Utilizzando Random Undersampling l'accuratezza con SVM è pari a 76.3% se si utilizza $C = 8$ e $\gamma = 0.0625$ mentre con Random Forest si ottiene un valore simile in quanto se si selezionano gli iperparametri $max\_depth = 80$, $max\_feature = 2$, $min\_samples\_leaf = 4$, $n\_estimators = 100$ la cross validation accuracy è pari a 77.4%.
Infine, se si sfrutta NearMiss con SVM l'accuratezza che si ha è 90.7% se $C = 2$ e $\gamma = 0.125$ mentre con Random Forest si ottiene un valore pari a 88.5% con $max\_depth = 90$, $max\_feature = 2$, $min\_samples\_leaf = 4$ e $n\_estimators = 300$.
Con il dataset ridotto applicando SMOTE per riequilibrare le classi si ricava con SVM un'accuratezza pari a 99.4% ($C = 16$, $\gamma = 0.25$) mentre con Random Forest si ottiene una cross validation accuracy di 99.1% ($max\_depth = 80$, $max\_feature = 3$, $min\_samples\_leaf = 3$, $n\_estimators = 400$).
Con Random Undersampling SVM dà un'accuratezza di 75.6% con $C = 2$ e $\gamma = 0.0625$; Random Forest anche in questo caso permette di raggiungere delle performance simile a SVM in quanto vale 77.4% ($max\_depth = 90$, $max\_feature = 2$, $min\_samples\_leaf = 4$ e $n\_estimators = 400$).
Infine, con NearMiss SVM raggiunge un'accuratezza di 93.7% ($C = 2$, $\gamma = 0.5$) e Random Forest dà una cross validation accuracy di 90.4% ($max\_depth = 80$, $max\_feature = 2$, $min\_samples\_leaf = 3$ e $n\_estimators = 100$).

### 3.2.9 Risultati ottenuti con SVM ribilanciando esclusivamente il training set con SMOTE

Come si già anticipato nella Sezione 2.5.1, quando si sceglie di ribilanciare un dataset con una tecnica di oversampling si può decidere se ricampionare il tutto prima di applicare qualsiasi algoritmo di classificazione oppure si possono ribilanciare le classi all'interno della fase di costruzione del modello. Nelle pagine precedenti si sono presentati i risultati ottenuti con la prima modalità; tuttavia, per una maggiore completezza si è ritenuto importante provare ad adottare la strategia alternativa. Sebbene, infatti, SMOTE non crei semplicemente copie dei campioni già esistenti, si è voluto verificare che i valori di accuratezza ottenuti non fossero *overoptimistic*.
Si è, dunque, proceduto a ribilanciare di volta in volta il training set e a riapplicare SVM a tutti 16 dataset di cui si è parlato in precedenza.
Quello che è emerso è che effettivamente si nota una lieve diminuzione dei valori di accuratezza ma i modelli ottenuti hanno comunque una capacità predittiva elevata, quasi sempre attorno al 95%. Più in particolare, SVM applicato sia al dataset completo con tutte le feature sia a quello privati delle feature di movimento restituisce una cross validation accuracy pari a 95.3%. I dataset privi delle colonne relative ai luoghi, invece, hanno una capacità predittiva pari a 95.0% e a 94.7% rispettivamente tenendo conto ed eliminando le feature di movimento.
Per quanto riguarda tutte le capacità predittive ottenute con i dataset parziali si osservi la tabella sottostante in cui sono messe a confronto la cross-validation accuracy (CVA) ottenuta applicando prima SMOTE e poi SVM (SMOTEall) e quella ottenuta ribilanciando solo il training set (SMOTEtrain). In generale si può affermare che si passa da una capacità predittiva superiore al 99% della prima strategia ad una che mediamente è pari al 95% e che non si notano particolare differenze tra i dataset con tutte le feature e quelli privati delle feature di movimento.

| **Dataset** | *CVA SMOTEall* | *CVA SMOTEtrain* |
|---|---|---|
| **Lamberene con tutte le feature** | 99.9% | 98.2% |
| **Lamberene senza feature *** | 99.9% | 98.3% |
| **Libreville con tutte le feature** | 99.7% | 94.3% |
| **Libreville senza feature *** | 99.6% | 94.6% |
| **Banjul con tutte le feature** | 98.8% | 91.6% |
| **Banjul senza feature *** | 98.6% | 89.4% |
| **Kumasi con tutte le feature** | 99.5% | 95.2% |
| **Kumasi senza feature *** | 99.4% | 94.9% |
| **Kilifi con tutte le feature** | 99.5% | 96.0% |
| **Kilifi senza feature *** | 99.3% | 95.6% |
| **Blantyre con tutte le feature** | 99.6% | 96.9% |
| **Blantyre senza feature *** | 99.4% | 96.8% |

## 3.3 Selezione delle Feature

Si è visto che con i modelli di classificazione finora utilizzati ci sono delle situazioni in cui i risultati sono già piuttosto soddisfacenti ma in altri casi si vorrebbe migliorare la loro capacità predittiva. Per tale motivo indispensabile risulta essere la fase di selezione delle feature in cui si possono eliminare quelle che risultano ridondanti e che, quindi, non portano alcuna informazione aggiuntiva per la costruzione di un buon modello.
Anche in questo caso si è preferito tentare due approcci diversi per poter poi confrontare i differenti risultati ottenuti. Più in particolare, si è utilizzato Univariate feature Selection che può essere considerato uno dei modi più semplici di operare dal momento che si limita ad associare ad ogni singola feature un punteggio.
Nel secondo caso, invece, si è sfruttato sempre Random Forest ma concentrandosi sulla proprietà di tale algoritmo di associare ad ogni feature un punteggio diverso.
In generale, quello che è emerso è che Random Forest dà risultati molto più soddisfacenti rispetto a Univariate Feature Selection dal momento che permette di ottenere dei buoni livelli di accuratezza anche con solo una decina di feature. Per tale motivo, si deve dare maggiore rilevanza a quanto emerso con questo secondo metodo e vedere il primo come una modalità per verificare che effettivamente il secondo abbia funzionato.
Se si va a considerare i singoli casi già presentati in precedenza la mole di dati ottenuti è piuttosto ingente in quanto si sono presi in considerazioni tutti e 16 i dataset ribilanciati nei 3 diversi modi. Di seguito si tenterà, quindi, di riassumere quanto si è osservato mettendo in luce gli aspetti più salienti.

### 3.3.1 Dataset completo

Come si è anticipato si sono applicate tecniche di selezione delle feature solo dopo aver ribilanciato il dataset in 3 modi differenti anche se è la classifica data da Random Forest da ritenersi più affidabile. In ogni caso, a titolo di esempio, si confrontino i risultati ottenuti con le due diverse strategie applicate al dataset completo contenente anche le feature che descrivono la capacità motoria dei pazienti.
Con SMOTE quello che si osserva è che ci sono tre categorie di feature che risultano essere particolarmente importanti. Infatti, come si può notare dalla lista con le prime quindici feature in ordine di importanza riprodotta in Figura (3.2), in entrambi i casi emerge il Blantyre Coma Score (*bcs*).
In aggiunta i due metodi concordano sulla rilevanza dell'abilità di riuscire a stare seduti autonomamente (*sit_yes*) per l'esito finale della malattia. Infine, eventuali difficoltà respiratorie (*deep_breathing*) sembrano essere considerate un fattore decisivo.
Quello su cui, invece, le due tecniche non combaciano è l'importanza dei giorni trascorsi in ospedale (*days in hospital*) dal momento che solo in Random Forest tale caratteristica compare addirittura in prima posizione.
Con Random Undersampling si ottengono dei risultato che sono in linea con quelli precedenti. Invero, appare evidente da quanto riportato in Figura (3.3) che il Blantyre Coma Score e la capacità di stare seduti risultano estremamente importanti. Unica novità sono i lattati (*lactate*) che sembrano assumere più rilevanza.

Infine, nel caso di NearMiss viene confermata la rilevanza dei lattati ma a questa si unisce anche l'abilità di assumere liquidi in autonomia, aspetto fondamentale soprattutto nei bambini più piccoli.

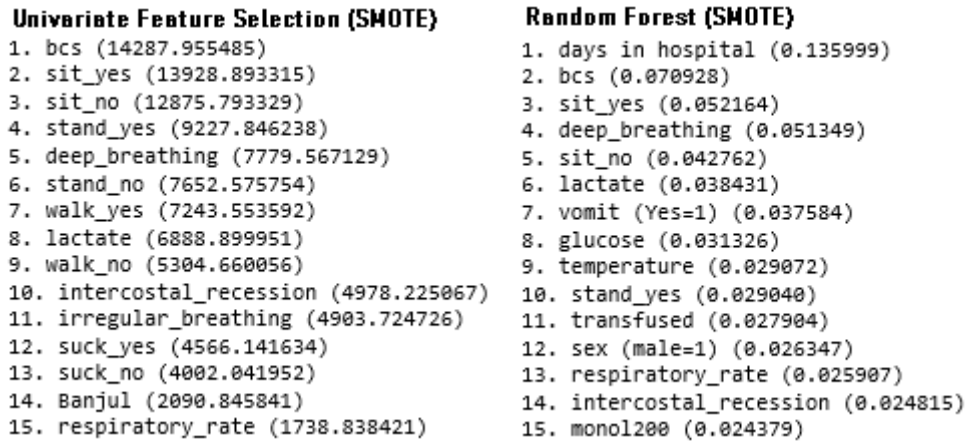



Figura 3.2: Selezione delle feature ribilanciando il dataset con **SMOTE** comprendente anche le informazioni relative alle capacità motorie

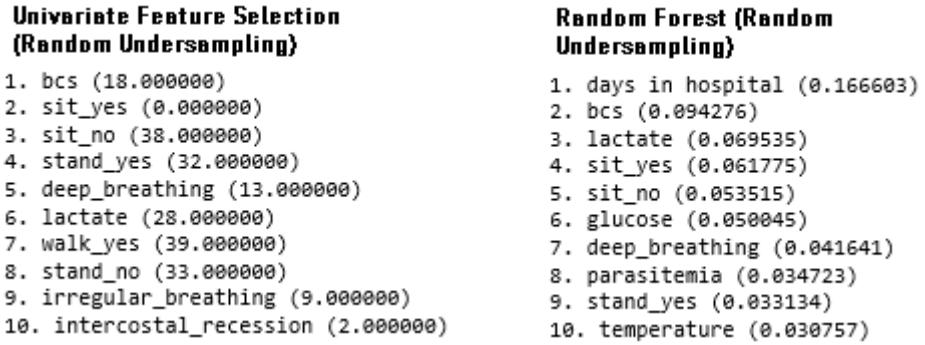



Figura 3.3: Selezione delle feature ribilanciando il dataset con **Random Undersampling** e comprendendo anche le informazioni relative alle capacità motorie

Se si analizza il dataset con le feature ridotte i parametri che risultano essere più importanti sono assimilabili a 4 gruppi: il Blantyre Coma Score, i lattati, le difficoltà respiratorie e i giorni in ospedale. Come si può notare dalla Figura (3.5), in questo caso, i due metodi sembrano essere piuttosto concordi in quanto ad importanza delle feature.
Tale classificazione dellle feature è confermata anche se si ribilanciano le due classi con Random Undersampling e NearMiss e, più in generale, quello che si può senza dubbio affermare è che sono la prima decina di feature che permettono di costruire un modello con una buona capacità predittiva.

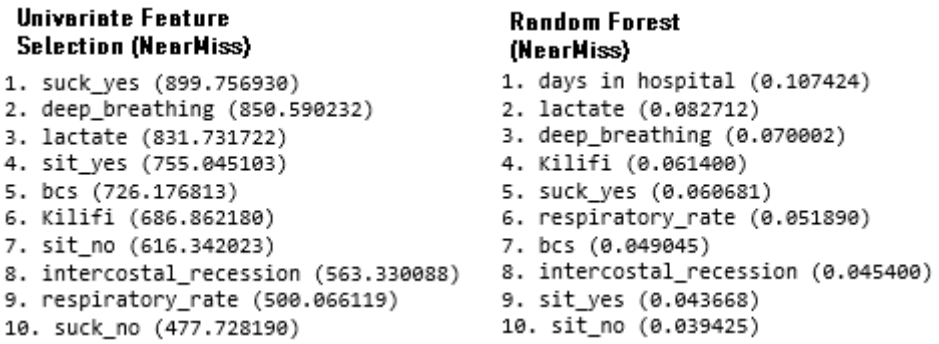



Figura 3.4: Selezione delle feature ribilanciando il dataset con **NearMiss** e comprendendo anche le informazioni relative alle capacità motorie

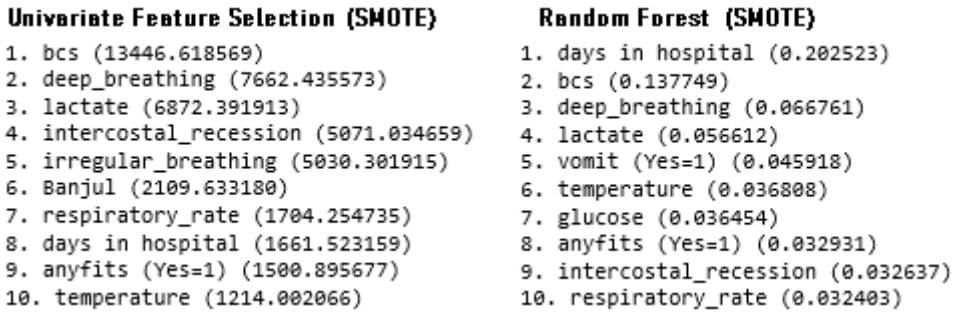



Figura 3.5: Ranking con dataset completo ribilaciato con **SMOTE** ma senza feature relative alla capacità motoria

### 3.3.2 Dataset senza luoghi

Con la feature selection nel primo dataset i luoghi non sono stati collocati nelle prime posizioni. Di conseguenza, con questo secondo tentativo si sono sostanzialmente trovate delle conferme di quanto osservato precedentemente. Si consideri prima di tutto il dataset con tutte le feature precedentemente descritte.
Con l'algoritmo SMOTE utilizzato per ribilanciare il dataset le due tecniche di feature selection hanno restituito i parametri in Figura (3.6) come più importanti.
Quello che è immediato notare è che sia nel caso di Univariate Feature Selection sia nel caso di Random Forest l'ordine delle feature è esattamente lo stesso di quello ottenuto con il dataset completo, almeno per le prime cinque feature.
Questa somiglianza tra i risultati si riscontra anche se si fa Random Undersampling e NearMiss.

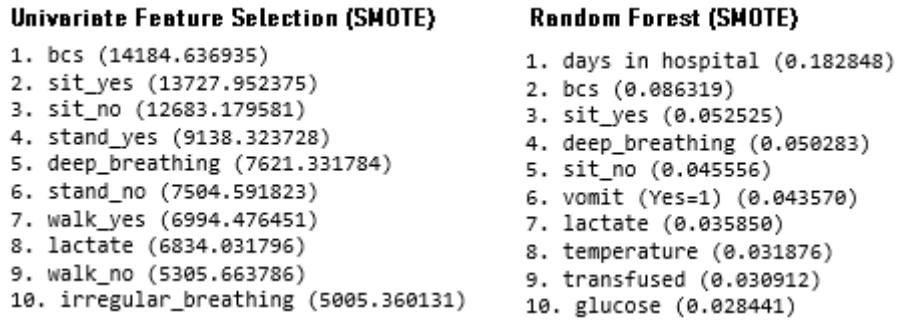



Figura 3.6: Selezione delle feature con dataset comprendente le informazioni sulle capacità motorie con SMOTE

Si osservino ora i risultati ottenuti adoperando il dataset con le feature ridotte. Dal momento che non sono più presenti le feature relative al movimento emerge l'importanza di alcuni esami clinici. Come nel caso in cui si sceglie di utilizzare tutte le feature, ai primi posti si ritrovano gli indicatori come *bcs*, *deep_ breathing* e *lactate*. Tuttavia, almeno per quanto riguarda Random Forest, si evidenziano alcuni cambiamenti attorno alla decima posizione come la rilevanza di *monol200* o *polyl200*.

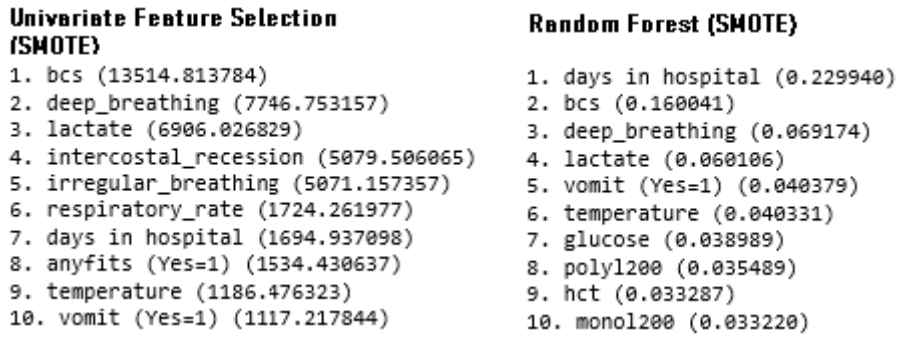



Figura 3.7: Selezione delle feature con dataset senza luoghi e feature relative a capacità motorie con SMOTE

Di seguito si riporteranno i risultati ottenuti con i dataset che prendono in esame i pazienti divisi per luoghi di provenienza. Dal momento che i migliori modelli predittivi si sono ottenuti ribilanciando l'insieme di dati con SMOTE e applicando Random Forest come metodo di selezione delle feature ci si concentrerà su questo caso confrontando di volta in volta la classificazione ottenute con e senza feature legate alla capacità motoria.

### 3.3.3 Dataset con pazienti da Lambarene

Dopo aver considerato solo i pazienti ricoverati nell'ospedale di Lambarene nei due differenti dataset e aver selezionato le feature, ciò che si ottiene sono i due ranking riportati di seguito.


Figura 3.8: Ranking ottenuti applicando SMOTE ai dataset relativi a Lambarene

Si può, quindi, notare che in entrambi i due dataset i giorni di ospedale, il Blantyre Coma Score, la presenza di convulsioni e le difficoltà respiratorie risultano aspetti altamente rilevanti per l'esito finale della malattia. Nella prima colonna, tuttavia, emergono anche molti indicatori legati all'essere in grado di sedersi, stare in piedi e camminare mentre nella parte destra si evidenzia l'importanza di altri esami di laboratorio come *parbc200*, *monol200* e *lactate* assieme a caratteristiche fisiche come l'età, il sesso o il peso.
L'importanza di tali feature sono confermate anche da quanto ottenuto con Random Undersampling e NearMiss. Tuttavia, soprattutto con il secondo metodo di undersampling, si osserva una crescente importanza dei dati personali del singolo bambino come sesso, peso ed età a discapito di alcune analisi cliniche. Al contempo, però, bisogna ricordare che in tale città il numero di pazienti deceduti è 25 e, perciò, è da privilegiare quanto ottenuto con il metodo di oversampling.

### 3.3.4 Dataset con pazienti da Libreville

Facendo feature selection con i dati relativi ai pazienti ospedalizzati a Libreville quello che si è potuto osservare è che in parte sono confermati i risultati ottenuti nel caso precedente. Infatti, come già visto, emerge che il Blantyre Coma Score e le difficoltà respiratorie sono particolarmente rilevanti. Tuttavia, in questo caso i giorni di ospedali con entrambi i dataset sembrano di gran lunga più importanti di tutte le altre feature, aspetto che, come si vedrà, è una caratteristica comune a tutti gli ospedali in cui il tasso di mortalità risulta elevato. Infine, si può osservare che non sono presenti i dati personali dei pazienti bensì l'eventuale presenza di sintomi quali vomito e milza ingrossata.

Per quanto riguarda i valori ottenuti con Random Undersampling quello che si può

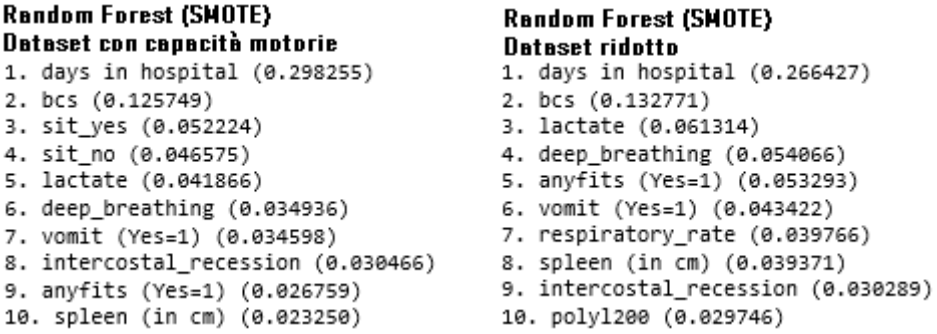



Figura 3.9: Ranking ottenuti applicando SMOTE ai dataset relativi a Libreville

affermare è che sono in linea con quelli appena presentati e, in particolare, in tutte le classificazione la feature *days in hospital* risulta sempre molto più importante di tutte le altre caratteristiche.

### 3.3.5 Dataset con pazienti da Banjul

In questa sezione è doveroso risottolineare che i dati raccolti in questo ospedale sono di particolare interesse perché si desidera capire le ragioni che fanno sì che la mortalità sia vicina al 10%.
Analizzando i ranking di seguito introdotti emergono più aspetti interessanti. In primo luogo, i giorni di ospedale risultano molto più importanti di tutte le altre feature analogamente a quanto notato con i pazienti di Libreville. Inoltre, oltre alla rilevanza del Blantyre Coma Score di cui si è parlato a più riprese, in questa città sembrano più decisivi i lattati per l'esito finale della malattia. Anche gli indicatori relativi al valore del glucosio e all'effettuazione di trasfusioni ematiche risultano fondamentali a discapito delle feature che monitorano la presenza di eventuali difficoltà respiratorie.

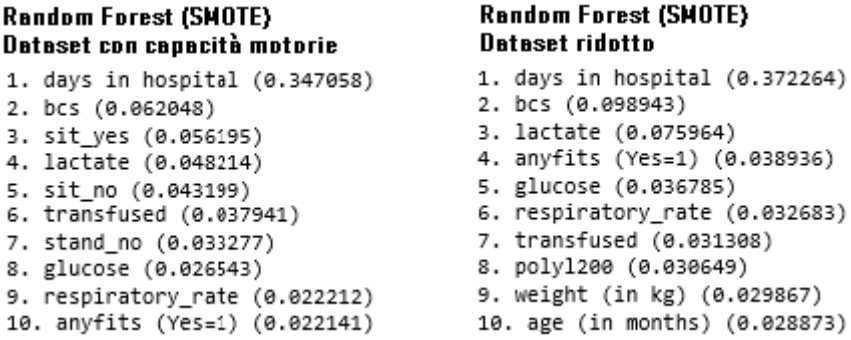



Figura 3.10: Ranking ottenuti applicando SMOTE ai dataset relativi a Banjul

Con Random Undersampling e NearMiss si riconferma l'importanza dei giorni di ospedali

ed emerge maggiormente *lactate* che si trova con entrambi i dataset sempre in seconda posizione.

### 3.3.6 Dataset con pazienti da Kumasi

A Kumasi i giorni di ospedale seguiti dal Blantyre Coma Score risultano molto importanti rispetto alle altre feature. Tuttavia, si individuano anche degli aspetti differenti rispetto agli altri luoghi prima analizzati a partire dalla rilevanza assunta dalle difficoltà respiratorie. Invero, tutte e 3 le feature legate a questa possibile complicanza (*deep_ breathing*, *intercostal_ recession* e *respiratory_ rate*) risultano nella parte superiore della classificazione.

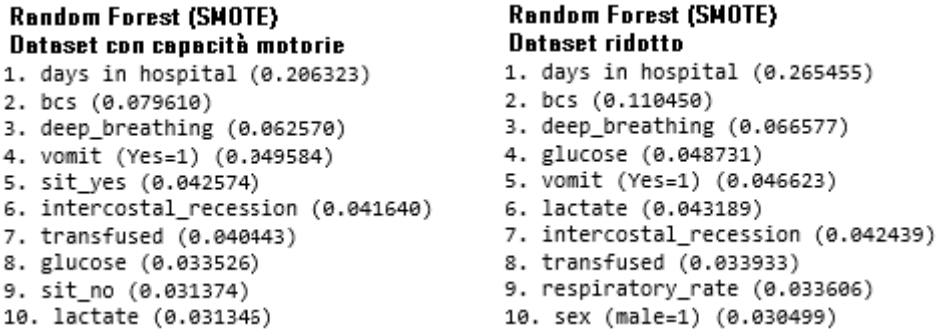



Figura 3.11: Ranking ottenuti applicando SMOTE ai dataset relativi a Kumasi

Da evidenziare è anche la variabile *glucose* che, non tanto con SMOTE ma con gli altri due metodi di ribilanciamento utilizzati, risulta confinata entro le prime cinque posizioni con entrambi i dataset.

### 3.3.7 Dataset con pazienti da Kilifi

Quello che si è ottenuto facendo feature selection con i pazienti curati nell'ospedale di Kilifi è di più difficile interpretazione rispetto a quanto visto finora. Come si vede in entrambe le colonne, oltre ai giorni di ospedale e al Blantyre Coma Score sembrano molto rilevanti anche la temperatura corporea, il valore del glucosio e le difficoltà respiratorie. É la prima occasione, quindi, dove emerge che la presenza di febbre più o meno elevata risulta fondamentale per l'esito della malattia. Inoltre, il dataset che tiene conto delle capacità motorie evidenzia la rilevanza della capacità di assumere liquidi e di stare in piedi.

In tale caso si registra anche una maggiore diversità in base al metodo di ribilanciamento scelto. Infatti, nel caso del primo dataset Random Undersampling, oltre ai giorni di ospedale mette in luce l'importanza del glucosio mentre con NearMiss predominano le feature legate alla capacità motoria.

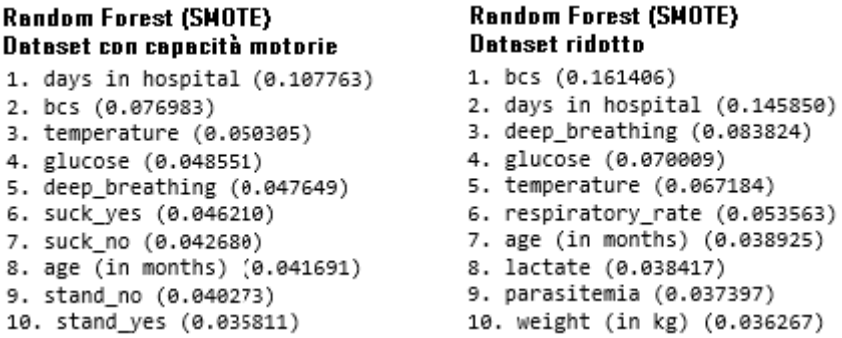



Figura 3.12: Ranking ottenuti applicando SMOTE ai dataset relativi a Kilifi

Con il dataset con numero di feature ridotte, invece, Random Undersampling dà risultati in linea con quanto ottenuto con SMOTE mentre NearMiss mette al primo posto *respiratory_ rate* e solo al sedicesimo posto i giorni di ospedale.

### 3.3.8 Dataset con pazienti da Blantyre

A Blantyre si ha un inversione rispetto all'ordine con cui sono disposti Blantyre Coma Score e giorni di ospedale come si vede nell'immagine sottostante.

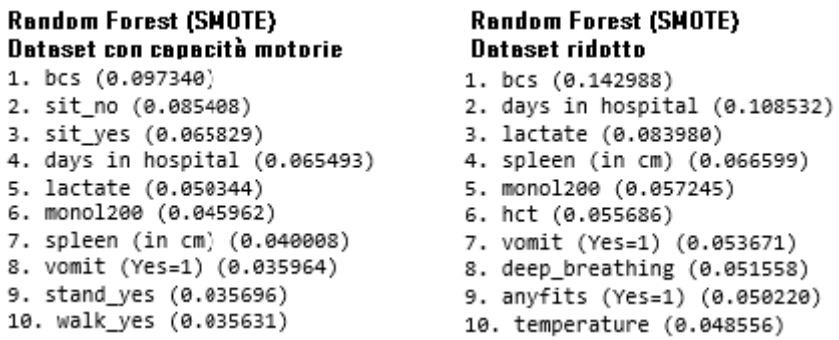



Figura 3.13: Ranking ottenuti applicando SMOTE ai dataset relativi a Blantyre

Inoltre, in questa città sembrano rilevanti i valori di *lactate* oltre che la dimensione della milza. Nel caso del dataset con le capacità motorie, quello che si può osservare è che sia la capacità di sedersi sia quella di camminare e di stare in piedi sembrano assumere una buona importanza a discapito degli esami clinici. Infine, la presenza di *monol200* nelle prime posizioni costituisce una novità rispetto ai dataset precedentemente analizzati.
Per quanto riguarda i metodi di undersampling, quello che accade con entrambi i dataset è che in prima posizione non si trova *bcs* bensì *lactate*. Inoltre, *hct* sembra assumere maggiore importanza rispetto a quanto rilevato con SMOTE.

## 3.4 Migliorare la cross validation accuracy

Come si è visto con Univariate Feature Selection e Random Forest è stato possibile disporre le feature in ordine di importanza e verificare la presenza di alcune caratteristiche che sembrano determinanti per l'esito dell'infezione malarica. Inoltre si è stabilito che tra le varie aree geografiche è possibile notare delle differenze che spiegano, probabilmente, i diversi tassi di mortalità.
Tuttavia, un altro motivo per cui si è passati per la fase della selezione delle feature è per poter migliorare l'accuratezza escludendo eventuali feature ridondanti o irrilevanti (che aggiungono rumore). Per poter, quindi, trovare il numero di feature che permette di volta in volta di avere il modello predittivo più performante possibile si sono compiuti i seguenti passi per ciascun dataset:

1. a partire dalla classifica delle feature ottenuta tramite Univariate Feature Selection o Random Forest si è selezionato solo la feature con lo score più alto e si è calcolata l'accuratezza con SVM;

2. si è ripetuto il passo 1. considerando ad ogni iterazione una feature in più fino ad arrivare a calcolare l'accuratezza con tutte le feature;

3. si è costruito un grafico in modo che fosse possibile visionare facilmente la feature con l'accuratezza più elevata.

Anche in questo caso si è ripetuto tale operazione per tutti e 16 dataset ribilanciati con i 3 differenti algoritmi di cui prima si è parlato. Per tale ragione la mole di dati raccolti è piuttosto consistente e, quindi, di seguito si provvederà a fare una sintesi di quanto ottenuto prendendo come esempio il dataset completo comprendente anche le feature relative alla capacità motoria.

### 3.4.1 Dataset completo bilanciato con SMOTE

Nella sezione precedente quello che si è messo in luce è che l'algoritmo SMOTE permette di ottenere un buon modello predittivo dato che i valori ottenuti calcolando l'accuratezza sono piuttosto soddisfacenti anche senza passare per la fase della selezione delle feature. Tuttavia, come ci si poteva facilmente aspettare, è possibile migliorare quanto ottenuto eliminando quelle feature che potrebbero essere motivo di disturbo. Di seguito, quindi, sono riportati i grafici nella Figura (3.14) e nella Figura (3.15) che permettono di comprendere quante feature è opportuno considerare per migliorare la capacità di classificazione.

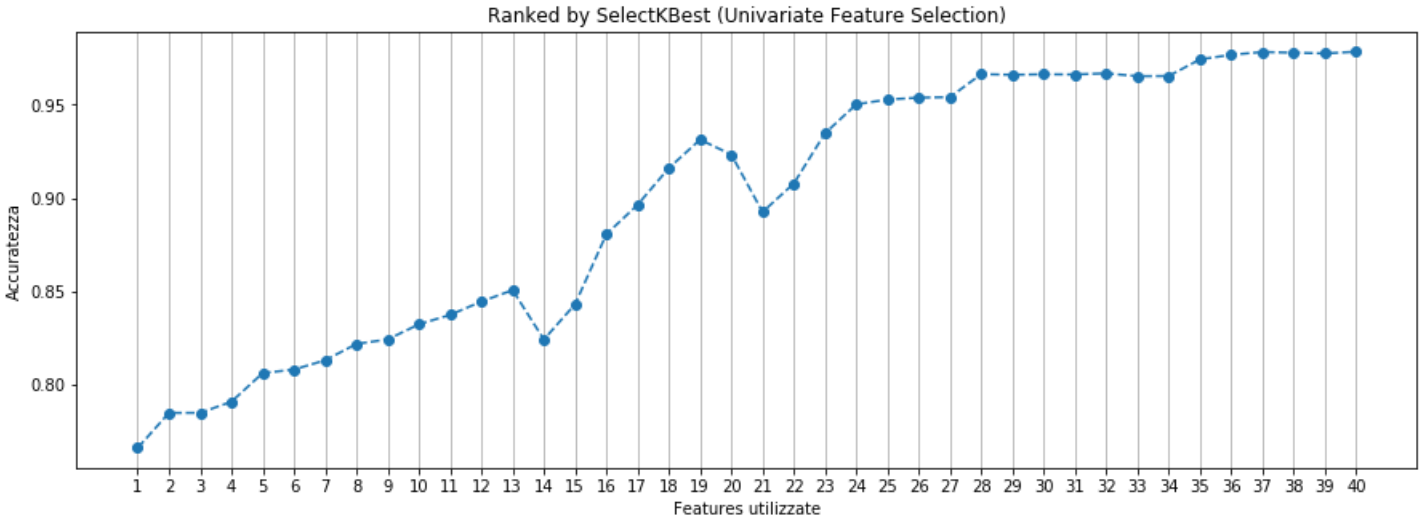



Figura 3.14: Questo grafico è stato ottenuto a partire dal dataset completo ribilanciato con **SMOTE** e sfruttando l'ordine delle feature calcolato con **Univariate Feature Selection**; quello che si osserva è che l'accuratezza migliore si ha considerando tutte le feature. Tuttavia, quello che si può osservare è che già con 19 feature si ha una capacità predittiva prossima al 95%.

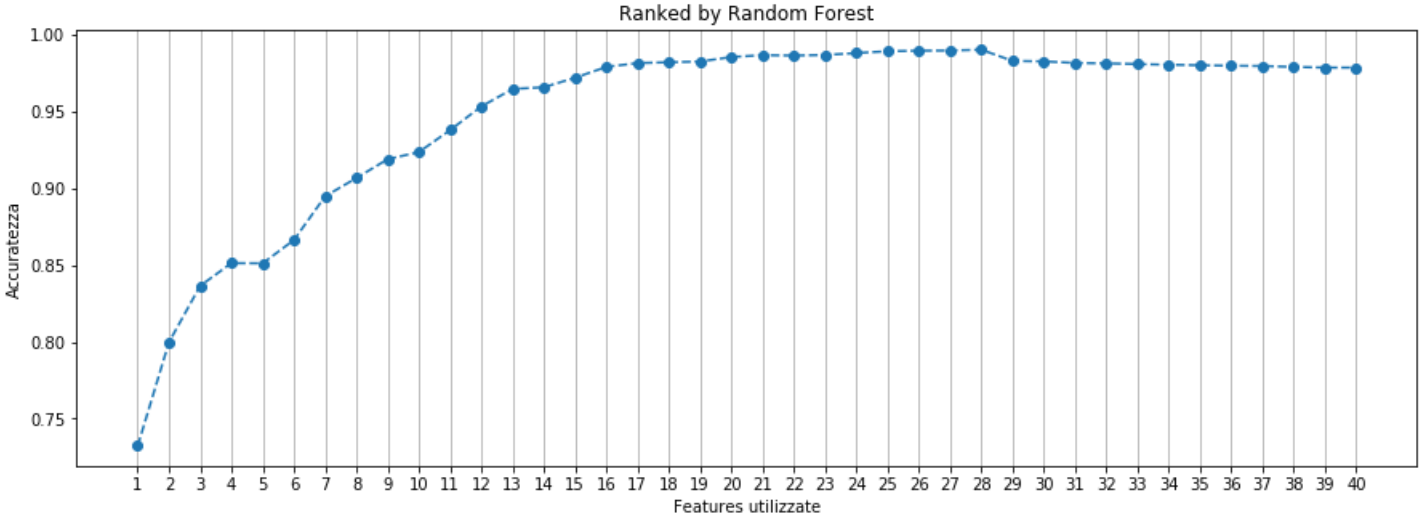



Figura 3.15: In questo secondo caso si riporta il grafico ottenuto a partire dal dataset completo ribilanciato con **SMOTE** ma con l'ordine delle feature selezionato grazie a **Random Forest**. Come è possibile osservare l'accuratezza cresce fino a che non si arriva a 28 feature per poi decrescere leggermente. Si migliora così l'accuratezza ottenuta precedentemente con SVM arrivando a 99.3%. Rispetto al caso precedente quello che si nota è una maggiore capacità predittiva dato che si ha un'accuratezza attorno al 95% già solo considerando una dozzina di feature.

### 3.4.2 Dataset completo bilanciato con Random Undersampling

Si è già detto che tale metodo di ribilanciamento del dataset si è rivelato il meno performante ma, proprio per questo motivo, in questo caso risulta decisivo passare per la fase di selezione delle feature. Nella Figura (3.16) e nella Figura (3.17) si descrive più in dettaglio quanto ottenuto.

### 3.4.3 Dataset completo ribilanciato con NearMiss

L'algoritmo NearMiss, come si è già detto nelle sezioni precedenti, ha dato dei risultati piuttosto soddisfacenti per quanto riguarda l'accuratezza del modello. Tuttavia, selezionando solo una parte delle feature quello che si è notato è che la capacità predittiva migliora di circa 1%. Nella Figura (3.18) e nella Figura (3.19) è possibile osservare più in dettaglio cosa accade sia utilizzando Univariate Feature Selection sia Random Forest.

### 3.4.4 Dataset completo con training set ribilanciato con SMOTE

Si è osservato che, in generale, l'accuratezza ottenuta applicando il ribilanciamento solo al training set è minore di quella ottenuta applicando SVM al dataset completo ricampionato con SMOTE ma il modello costruito è comunque piuttosto soddisfacente. Tuttavia, anche in questo caso è opportuno passare alla fase delle selezione delle feature per migliorarne la capacità predittiva. A questo avviso si osservino i grafici in Figura (3.20) e Figura (3.21).

In conclusione, dunque, quello che emerge è che passando per la fase di selezione delle feature i modelli di classificazione migliorano le loro capacità predittive in tutti i casi analizzati. Tuttavia, se il valore dell'accuratezza ottenuta con SVM considerando tutte le feature è già piuttosto buona si assiste ad un miglioramento di circa l'1%, mentre se è al di sotto del 90% passare per questa fase risulta di fondamentale importanza perché si osservano incrementi anche superiori al 10%.
Nel caso degli altri 15 dataset analizzati, ossia quello senza luoghi e quelli in cui si sono selezionati solo i pazienti ricoverati in uno specifico ospedale quello che accade è simile a quanto si è detto per il dataset completo.

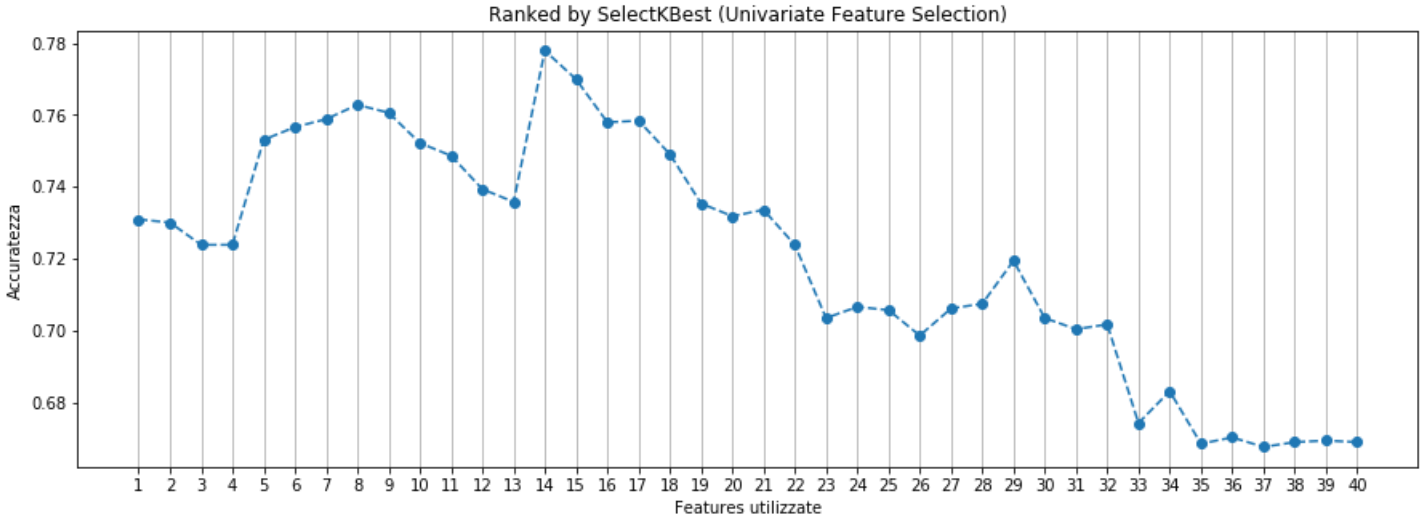



Figura 3.16: Questo grafico è ottenuto a partire dal dataset completo ribilanciato con **Random Undersampling** e utilizzando il ranking dato da **Univariate Feature Selection**. In questo caso si osserva un netto miglioramento dell'accuratezza se si selezionano solo le prime 14 feature per addestrare il modello.

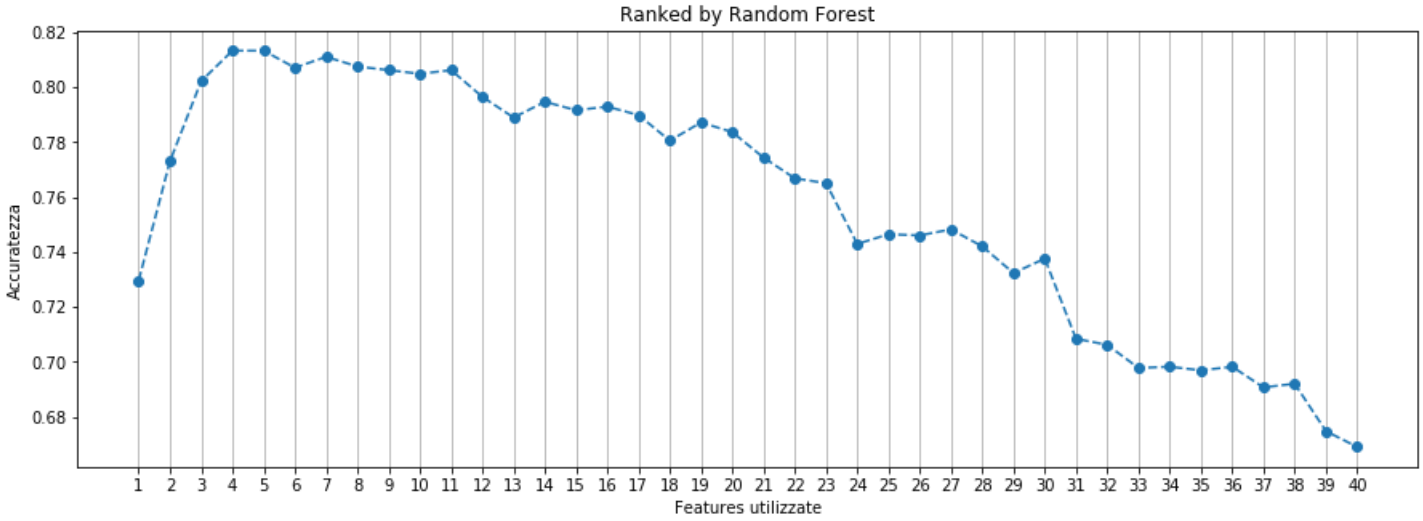



Figura 3.17: Tale immagine è la rappresentazione delle diverse accuratezza che si ottengono in corrispondenza del numero di feature selezionate in base all'ordine fornito da **Random Forest** sfruttando il dataset completo ribilanciato con **Random Undersampling**. Come è facile notare, in questo caso si ottiene il valore più alto possibile di accuratezza pari a 75% considerando solo le prime 4 feature.

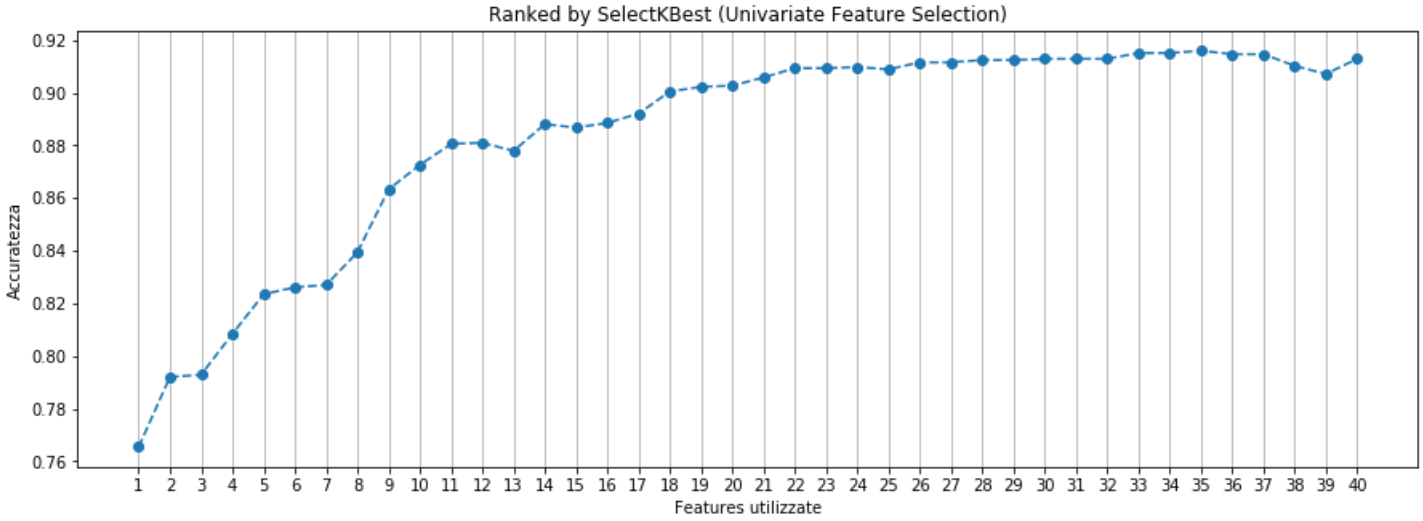



Figura 3.18: Tale grafico riporta in ordinata l'accuratezza ottenuta al variare del numero di feature ordinate con **Univariate Feature Selection** a partire dal dataset completo ribilanciatio con **NearMiss**. In particolare, la cross validation accuracy cresce fino ad arrivare a 92.5% se si considerano 35 feature per poi calare leggermente.

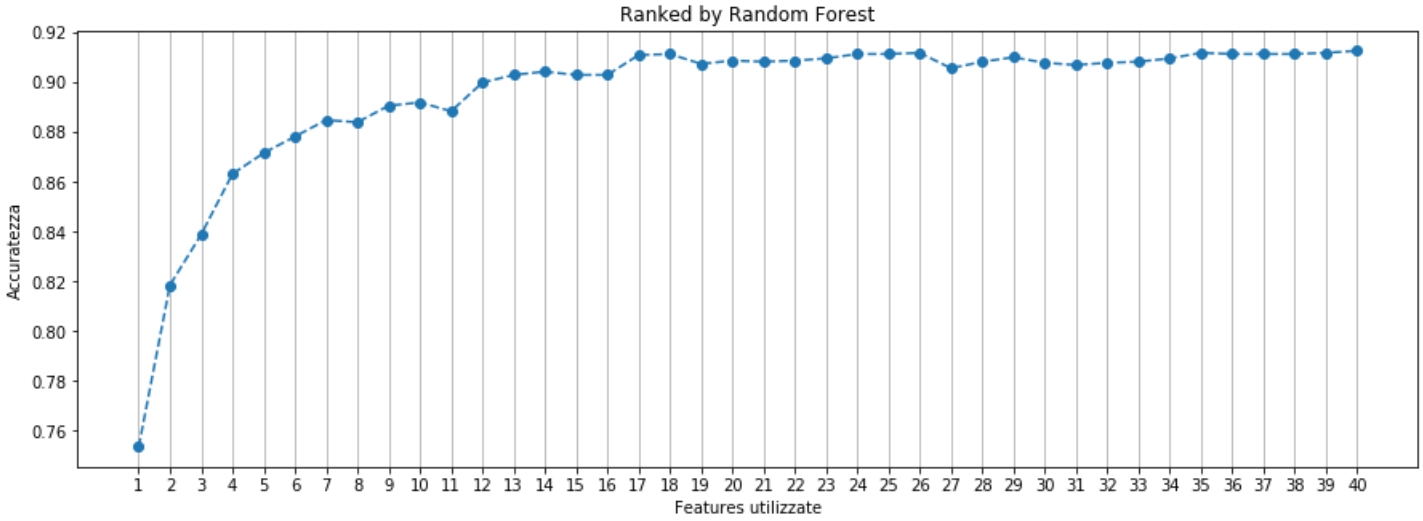



Figura 3.19: L'immagine raffigura come varia la cross validation accuracy se si considera il dataset completo ribilanciato con **NearMiss** e se si ordinano le feature con **Random Forest**. Quello che è possibile notare è che il miglior modello si trova se si considerano le prime 18 feature che portano l'accuratezza a 94.5%. Come si può osservare anche in questa occasione Random Forest rispetto a Univariate Feature Selection risulta avere una capacità predittiva molto più elevata con un minor numero di feature.

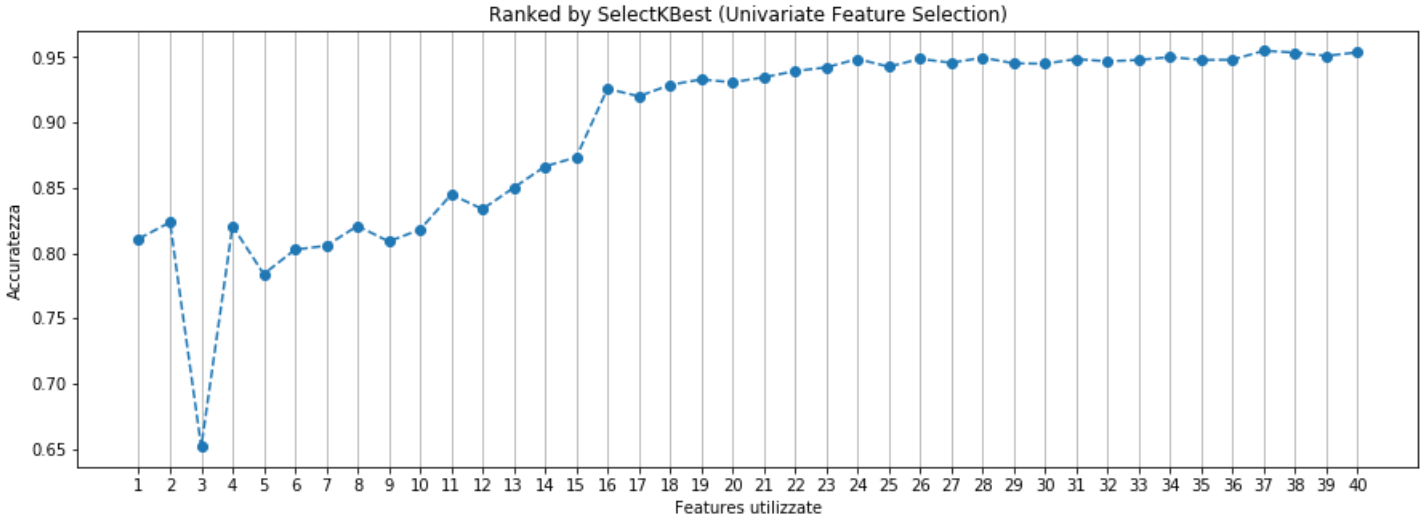



Figura 3.20: Il grafico sovrastante è stato ottenuto applicando SVM al dataset completo e ribilanciando il **training set con SMOTE**. Nelle ordinate si osserva il valore dell'accuratezza al variare del numero di feature, ordinate con **Univariate Feature Selection**. L'accuratezza migliore è 95.4% e si ottiene con 37 feature.

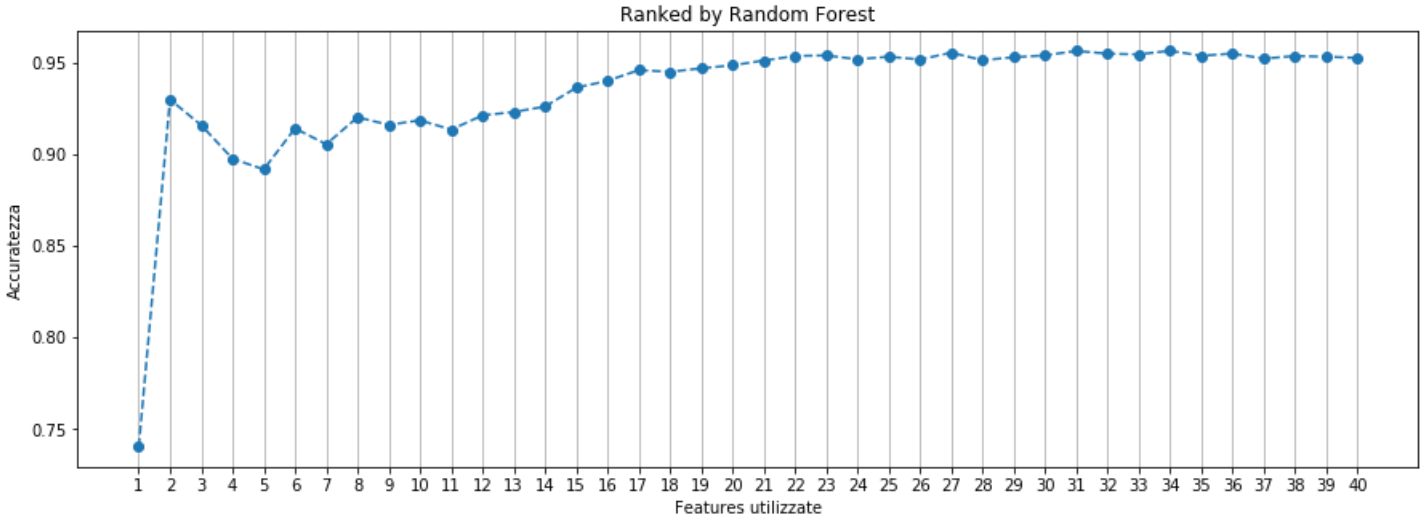



Figura 3.21: Nell'immagine è riprodotto il grafico relativo alla variazione della capacità predittiva ottenuta con SVM e ribilanciando il **training set con SMOTE** anziché il dataset completo. Le feature sono ordinate con **Random Forest**. Quello che si può osservare è che la migliore capacità predittiva si ottiene considerando 31 feature anche se già con 15 feature l'accuratezza è prossima al 95%.

## 3.5 Apprendimento non supervisionato

Quest'ultima parte dell'analisi si differenzia dalle precedenti in quanto si è passati ad utilizzare tecniche di apprendimento non supervisionato per uno scopo differente rispetto a quanto visto finora. Si è, infatti, concentrato l'attenzione solo nei pazienti deceduti e con l'algoritmo di clustering K-means si è tentato di suddividere i 1130 campioni in gruppi in modo che coloro che provenivano dallo stesso luogo geografico fossero possibilmente concentrati in uno dei cluster ottenuti. Tale tecnica è stata applicata sia comprendendo *sit*, *walk*, *stand* e *suck* sia eliminando le informazioni relative a quest'ultime. Così facendo si è voluto capire se fosse possibile raccogliere ulteriori informazioni relative solo alle caratteristiche dei pazienti che non hanno superato l'infezione.

### 3.5.1 Risultati con dataset completo con feature relative a capacità motorie

Dopo aver estratto solo i pazienti deceduti si è applicato K-means settando come numero massimo di iterazioni $max\_iter = 1000$ e come numero di volte che l'algoritmo viene ripetuto con centroidi differenti $n\_init = 50$. Inoltre, si è scelto di impostare la tolleranza a $tol = e^{-6}$ e di utilizzare l'algoritmo *elkan* per decidere a quali centroidi associare ciascun punto. In particolare, con tale modello si sfrutta la distanza triangolare per confrontare di volta in volta la distanza tra soggetto e centroidi.
Con $k = 3$ e $k = 6$ quello che si osserva è che la maggior parte dei pazienti di Banjul si ritrovano in un unico cluster e lo stesso fenomeno si ha con gli ospedalizzati a Kilifi.
Una discreta clusterizzazione si ottiene con $k = 4$ dal momento che, come si vede dalla tabella, il gruppo 3 ha prevalentemente soggetti di Banjul e il gruppo 4 ha molti pazienti di Kilifi e Kumasi. I rimanenti sottoinsiemi, invece, sono di più difficile interpretazione.

| **Luoghi** | *Lambarene* | *Libreville* | *Banjul* | *Kumasi* | *Kilifi* | *Blantyre* | *Totale* |
|---|---|---|---|---|---|---|---|
| **Gruppo 1** | 5 | 30 | 93 | 81 | 35 | 42 | 289 |
| **Gruppo 2** | 2 | 0 | 0 | 7 | 0 | 0 | 9 |
| **Gruppo 3** | 17 | 40 | 197 | 103 | 2 | 50 | 409 |
| **Gruppo 4** | 1 | 19 | 32 | 120 | 208 | 43 | 423 |
| **Totale** | 25 | 89 | 322 | 311 | 248 | 135 | 1130 |

Quello che si nota passando alla fase delle selezione delle feature è che *irregular_ breathing* risulta con entrambi i metodi la più rilevante per la clusterizzazione. Importante sembrano essere anche la capacità di assumere o meno delle bevande in autonomia (*suck*), peso (*weight*) ed età (*age*). Come è possibile osservare dalla Figura (3.22), tuttavia, ci sono anche delle differenze sostanziali tra i due metodi di feature selection come nel caso di *parbc200* che in Univariate Feature Selection è al secondo posto mentre con Random Forest si trova addirittura al venticinquesimo. In ogni caso, si deve fare maggiore affidamento su quanto emerge con il secondo metodo in quanto se si prendono in considerazione le prime 23 feautures si ottiene un'accuratezza di circa 96.5% contro il 94% ottenuto con

la prima modalità. Si osservino comunque i grafici inseriti di seguito per capire come varia la capacità predittiva in base al numero di feature.

**Univariate Feature Selection**

1. irregular_breathing (17551.123273)
2. parbc200 (1445.672912)
3. suck_NA (366.042511)
4. suck_no (98.128361)
5. intercostal_recession (97.173564)
6. weight (in kg) (88.364538)
7. age (in months) (70.171554)
8. suck_yes (66.529670)
9. deep_breathing (66.010430)
10. walk_no (41.439907)
11. walk_NA (38.226221)
12. anyfits (Yes=1) (36.112145)
13. bcs (34.571883)
14. sit_no (30.407202)
15. lactate (30.053881)
16. stand_no (25.425515)
17. stand_NA (25.227026)
18. monol200 (24.272524)
19. sit_yes (21.741576)
20. days in hospital (20.454184)
21. respiratory_rate (18.463552)
22. polyl200 (17.112252)
23. glucose (11.556321)
24. vomit (Yes=1) (10.042961)
25. sit_NA (8.166239)
26. spleen (in cm) (6.984832)
27. stand_yes (6.879585)
28. walk_yes (5.217239)
29. transfused (4.732391)
30. temperature (4.029603)
31. parasitemia (1.988459)
32. hct (1.332982)
33. lactanal (analox machine=1) (1.19032)
34. sex (male=1) (0.801542)

**Random Forest**

1. irregular_breathing (0.268031)
2. suck_NA (0.127914)
3. suck_no (0.059261)
4. weight (in kg) (0.058529)
5. age (in months) (0.046790)
6. intercostal_recession (0.041591)
7. lactate (0.040829)
8. suck_yes (0.027229)
9. deep_breathing (0.026773)
10. monol200 (0.026505)
11. glucose (0.024894)
12. bcs (0.024392)
13. respiratory_rate (0.023881)
14. parasitemia (0.022319)
15. hct (0.020586)
16. temperature (0.018028)
17. days in hospital (0.017823)
18. polyl200 (0.013939)
19. anyfits (Yes=1) (0.013778)
20. sit_no (0.013705)
21. walk_no (0.013447)
22. walk_NA (0.010144)
23. spleen (in cm) (0.009399)
24. sit_yes (0.009365)
25. parbc200 (0.008001)
26. stand_no (0.006876)
27. vomit (Yes=1) (0.006598)
28. transfused (0.005153)
29. stand_NA (0.004905)
30. sex (male=1) (0.004388)
31. stand_yes (0.002087)
32. walk_yes (0.001652)
33. sit_NA (0.000885)
34. lactanal (analox machine=1) (0.000302)

Figura 3.22: Classificazione delle feature ottenuto con Univariate Feature Selection e Random Forest

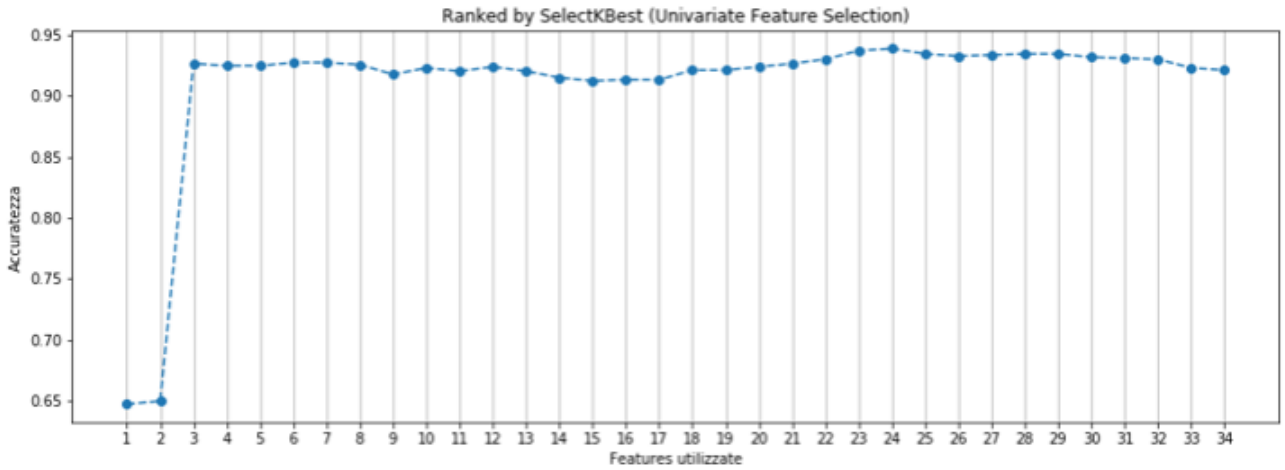



Figura 3.23: Grafico che riporta come varia l'accuratezza in base al numero di feature ordinate con **Univariate Feature Selection** se si considera il dataset con tutte le feature

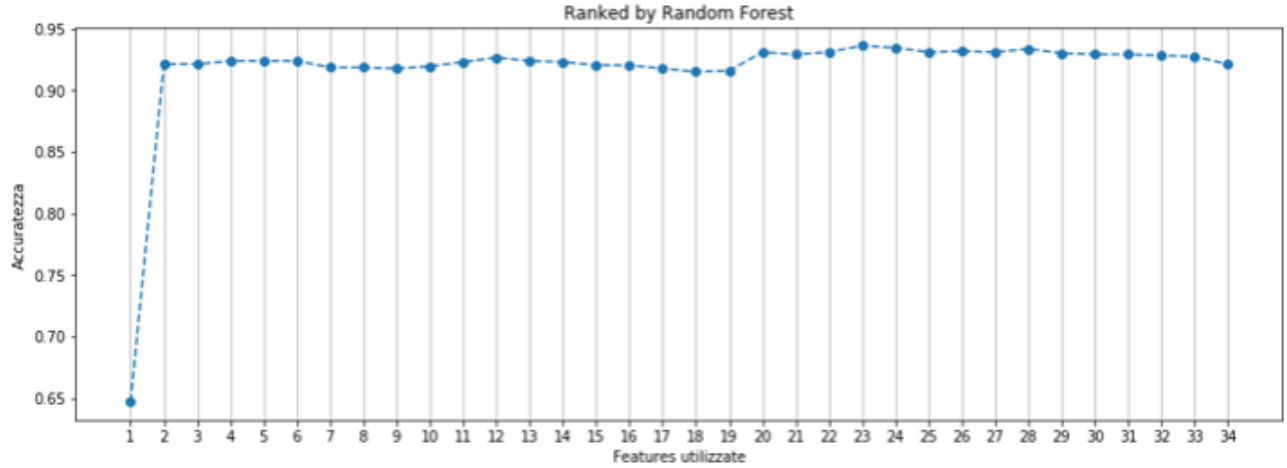



Figura 3.24: Grafico che riporta come varia l'accuratezza in base al numero di feature ordinate con **Random Forest** se si considera il dataset con tutte le feature

### 3.5.2 Risultati con dataset completo senza feature relative a capacità motorie

Togliendo le colonne relative a *sit*, *stand*, *walk* e *suck* e selezionando solo i pazienti deceduti i risultati più interessanti si sono ottenuti applicando K-means con $k = 4$. Infatti, come si vede nella tabella sottostante, si va a formare un cluster che contiene quasi esclusivamente pazienti di Banjul mentre in un altro si concentrano pazienti di Kumasi, Kilifi e Blantyre. Meno interessanti sono, invece, gli altri due gruppi dato che uno contiene solo 9 soggetti di cui 7 provenienti da Kumasi mentre nell'ultimo insieme si ritrovano pazienti ricoverati in tutti e 6 gli ospedali. Questa divisione, dunque, denota che sono presenti delle peculiarità tra le varie regioni prese in esame.

| **Luoghi** | *Lambarene* | *Libreville* | *Banjul* | *Kumasi* | *Kilifi* | *Blantyre* | *Totale* |
|---|---|---|---|---|---|---|---|
| **Gruppo 1** | 5 | 30 | 93 | 82 | 39 | 42 | 291 |
| **Gruppo 2** | 3 | 8 | 228 | 7 | 1 | 7 | 254 |
| **Gruppo 3** | 15 | 51 | 1 | 215 | 208 | 86 | 576 |
| **Gruppo 4** | 2 | 0 | 0 | 7 | 0 | 0 | 9 |
| **Totale** | 25 | 89 | 322 | 311 | 248 | 135 | 1130 |

Per quanto riguarda i risultati ottenuti sfruttando i due algoritmi che permettono di ordinare le feature in ordine di importanza in questo caso entrambi i metodi concordano sulla rilevanza che assume *irregular_ breathing* mentre negli altri casi le due tecniche danno risultati contrastanti. Nuovamente, tuttavia, si deve dare più importanza alla classificazione ottenuta con Random Forest perché permette di ottenere una maggiore capacità predittiva. Più in particolare, con Univariate Feature Selection l'accuratezza è pari a 84.07% con 17 feature mentre con Random Forest si sale a 87.16% con lo stesso numero di variabili.
In seguito si è anche osservato come varia la capacità predittiva in base al numero di feature sfruttate e si sono quindi prodotti i grafici che saranno inseriti nelle prossime pagine.

Infine, degno di nota è anche quello che si è ottenuto settando $k = 6$. Infatti, se da un lato non si osserva una concentrazione di pazienti di Kumasi, Kilifi e Blantyre in uno dei cluster, dall'altro si ha un gruppo di 223 pazienti di Banjul su un totale di 225.

**Univariate Feature Selection**

1. irregular_breathing (21942.251983)
2. parbc200 (1368.958988)
3. intercostal_recession (86.979330)
4. deep_breathing (70.774341)
5. monol200 (39.053024)
6. bcs (35.669935)
7. respiratory_rate (31.418891)
8. anyfits (Yes=1) (19.323349)
9. spleen (in cm) (15.203231)
10. weight (in kg) (14.427876)
11. glucose (14.401007)
12. polyl200 (13.510474)
13. days in hospital (13.326164)
14. lactate (13.306583)
15. vomit (Yes=1) (10.443840)
16. age (in months) (10.271582)
17. parasitemia (9.012383)
18. transfused (2.853873)
19. hct (2.202432)
20. sex (male=1) (1.360521)
21. lactanal (analox machine=1) (1.189215)
22. temperature (0.908216)

**Random Forest**

1. irregular_breathing (0.379027)
2. hct (0.067215)
3. monol200 (0.063636)
4. respiratory_rate (0.063252)
5. parasitemia (0.062458)
6. glucose (0.048541)
7. lactate (0.037075)
8. weight (in kg) (0.033316)
9. intercostal_recession (0.032861)
10. deep_breathing (0.030345)
11. age (in months) (0.025645)
12. bcs (0.025575)
13. temperature (0.021813)
14. days in hospital (0.020521)
15. anyfits (Yes=1) (0.019242)
16. parbc200 (0.017518)
17. polyl200 (0.017090)
18. spleen (in cm) (0.014909)
19. vomit (Yes=1) (0.009266)
20. sex (male=1) (0.005818)
21. transfused (0.004771)
22. lactanal (analox machine=1) (0.000106)

Figura 3.25: Classificazione delle feature escludendo quelle relative a capacità motorie ottenuto con Univariate Feature Selection e Random Forest

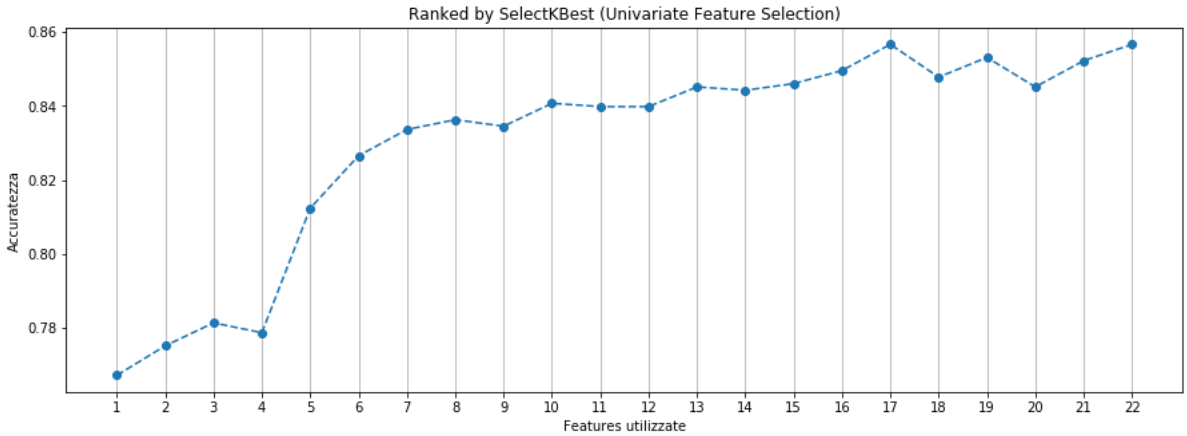



Figura 3.26: Grafico che riporta come varia l'accuratezza in base al numero di feature ordinate con **Univariate Feature Selection**

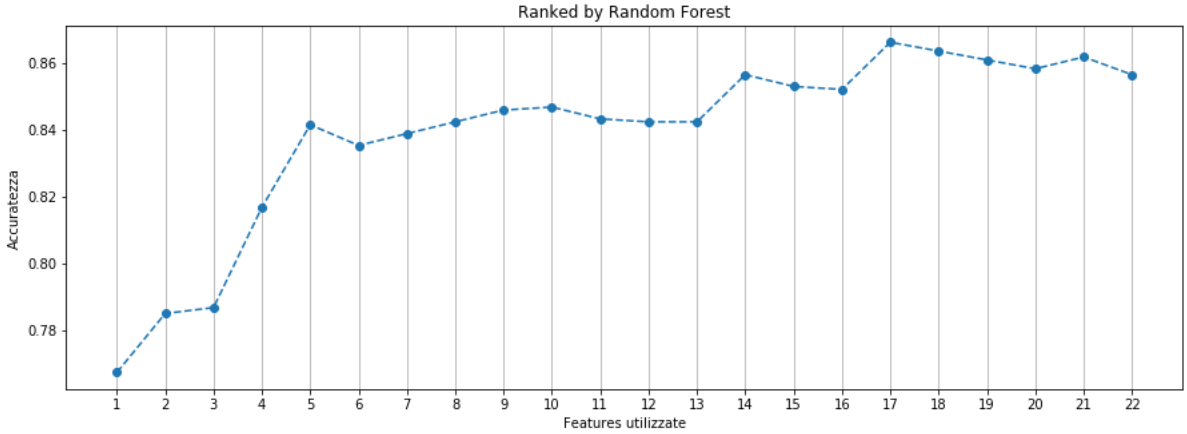



Figura 3.27: Grafico che riporta come varia l'accuratezza in base al numero di feature ordinate con **Random Forest**

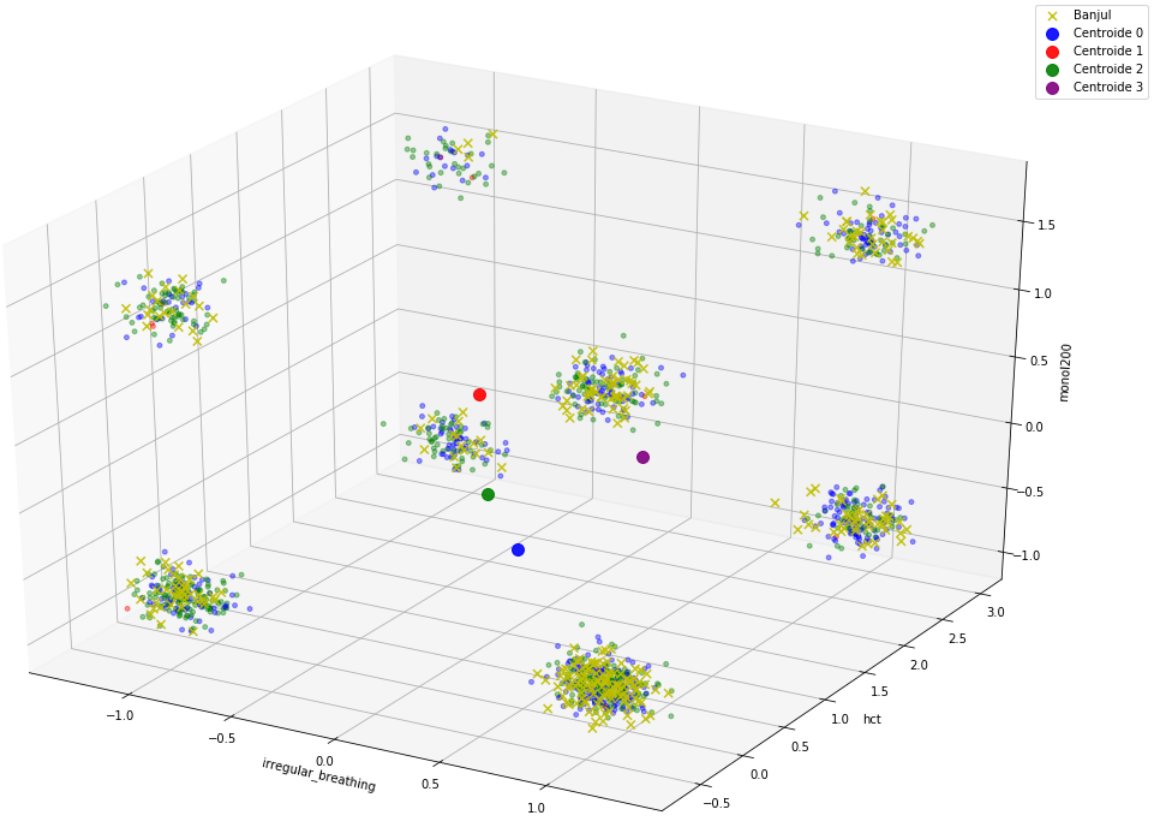



Figura 3.28: Grafico con la divisione in 4 cluster rispetto alle prime 3 feature della classificazione ottenuta con Random Forest. Si osservi che tutti i punti corrispondenti a pazienti di Banjul sono contrassegnati in giallo.

# Capitolo 4

# Conclusioni

Il lavoro qui svolto ha permesso di rispondere ad alcuni degli interrogativi con cui si era aperta la trattazione.

In particolare è stato possibile costruire un modello in grado di predire l'esito della malaria e, sebbene siano state utilizzate diverse strategie, ciò che ha dato i risultati migliori è senza dubbio SVM con il dataset ribilanciato con SMOTE.
Per quanto riguarda l'identificazione delle feature più importanti per il decorso della malattia è emerso che, se si considera il dataset completo, i *giorni di ricovero in ospedale*, la *Blantyre Coma Score* e le *difficoltà respiratorie* risultano essere particolarmente rilevanti. In aggiunta, anche la *capacità di stare seduti* sembra essere decisiva in tutti i casi in cui si sono utilizzati dataset con tutte le feature.

Se si considerano i pazienti divisi per regione di provenienza, quello che ha permesso di mettere in luce tale lavoro è che effettivamente ci sono delle differenze dal punto di vista clinico tra i soggetti di differenti luoghi. Più precisamente, dove la mortalità è più alta sembrano essere più importanti i *giorni di ospedale* e il valore dei *lattati* mentre negli altri casi emergono maggiormente la *Blantyre Coma Score* e le *difficoltà respiratorie*.

In aggiunta la differenza tra i pazienti provenienti da differenti zone è stata confermata anche da quanto ottenuto applicando K-means ai soggetti deceduti. Più in dettaglio, la maggior parte dei pazienti di Banjul, ospedale con il più alto tasso di mortalità, sono stati raggruppati in un unico cluster mentre sembrano esserci delle affinità tra coloro che sono stati ricoverati a Blantyre, Kilifi e Kumasi. Più difficile da collocare in un preciso sottoinsieme sono i pazienti di Lambarene e Libreville, forse anche a causa del minor numero di campioni a disposizione rispetto agli altri luoghi.

In conclusione, come si può osservare da quanto esposto, in tutte le fasi della ricerca si è scelto di dare particolare importanza alla provenienza dei soggetti a seguito di un'indicazione dei medici che erano interessati a comprendere per quale ragione il protocollo di cura non risulta essere ugualmente efficace nei diversi ospedali.

Allo stesso tempo, però, questo non è l'unico approccio possibile. Ad esempio, visto la rilevanza assunta dai giorni di ospedale sarebbe in futuro interessante concentrarsi su tale feature o, in alternativa, si potrebbero raccogliere informazioni aggiuntive sfruttando tecniche non considerate in tale progetto. Di fatto, la malaria è un problema globale e qualsiasi ulteriore contributo potrebbe rivelarsi fondamentale per migliorare le cure oggi a disposizione.

# Bibliografia


[1] WHO, *World Malaria Report 2018*, November 2018

[2] A. S. Fauci, S.L. Hauser, D.L. Kasper, D.L. Longo, J. Loscalzo,J.L. Jameson (2017), *Harrison Principi di medicina interna* $19^a$ *Edizione*, Casa Editrice Ambrosiana, Milano

[3] P.R. Murray, K.S. Rosenthal, M.A. Pfaller (2013), *Microbiologia medica* $7^a$ *Edizione*, Edra Lswr Spa, Milano

[4] O. Cominetti, D. Smith, F. Hoffman, M. Jallow, M. L. Thézénas, H. Huang, D. Kwiatkowski, P. K. Maini, C. Casals- Pascual (2018), *Identification of a Novel Clinical Phenotype of Severe Malaria using a Network- Based Clustering Approach*, Scientific Reports, Springer Nature Publishing AG , Heidelberg

[5] R. Kuang, J. Gu, H. Cai, Y. Wang (2009), *Improved Prediction of Malaria Degradomes by Supervised Learnig with SVM and Profile Kernel*, Genetica May 2009 (pag. 189-209), Springer Interneational Publishing, New York

[6] M. Poostchi, K. Silamut, R. J. Maude, S. Jaeger, G. Thoma (2018), *Image analysis and machine learning for detecting malaria*, Elsevier Inc., Amsterdam

[7] W. David Pan, Y. Dong, D. Wu (2018), *Classification of Malaria-Indected Cells Using Deep Convolutional Neural Networks* In *Machine Learning - Advanced Techniques and Emerging Applications*, IntechOpen

[8] N. V. Chawla, K. W. Bowyer, L. O. Hall, W. P. Kegelmeyer (2002), *SMOTE: Synthetic Minority Over-sampling Technique* In *Journal of Artificial Intelligence Reasearch 16*

[9] Z. Yang, D. Gao, *An Active Under-Sampling Approach for Imbalanced Data Classification*, IEEE

[10] V. Kumar (2008), *Computational Methods of Feature Selection*, Chapman and Hall CRC by Taylor and Francis Group, Londra

[11] U. Stańczyk, L. C. Jain (2015), *Feature Selection for Data and Pattern Recognition*, Springer

[12] J. Tang, S. Alelyani, H. Liu (2014), *Feature Selection for Classification: A Review* In *Data Classification: Algorithms and Applications*, Chapman and Hall CRC, Londra

[13] G. H. Bakir, L. Bottou, J. Weston (2004), *Breaking SVM Complexity with Cross-Training* In Advances in neural information processing systems, Vancouver

[14] A. Fernández, S. García, M. Galar, R. C. Prati, B. Krawczyk, F. Herrera (2018), *Learning from Imbanced Data Sets*, Springer Nature Switzerland

[15] H. He, E. A. Garcia (2009), *Learning from Imbalanced Data*, IEEE Transactions on knowledge and Data engineering Vol.21 N.9

[16] L. Breiman, J. Friedman, C.J. Stone, R. A. Olshen (1984), *Classification and Regression Trees*

[17] L. Breiman (2001), *Random Forests* In *Machine Learning*, Springer

[18] C. Cortes, V. Vapnik (1995), *Support-vector network* In *Machine Learning*, Springer

[19] R. J. Vanderbei (2008), *Linear Programming: Foundations and Extensions* Third Edition, Springer

[20] J. Han, M. Kamber, J. Pei, *Cluster Analysis: Basic Concepts and Methods* In *Data Mining Third Edition*, Morgan Kaufmann, (pag. 443-495)

[21] M. S. Santos, J. P. Soares, P. H. Abreu, H. Araujo, J. Santos (2018), *Cross-Validation for Imbalanced datasets: Avoiding Overoptimistic and Overfitting Approaches*, IEEE Computational Intelligence Magazine

[22] D. Pollard (1982), *Quantization and the Method of k-Means* In *IEEE Transactions on Information Theory* Vol. 28

[23] E. Martin, K. Hans-Peter, S.Jörg, X. Xiaowei, S. Evangelos, H. Jiawei (1996), *A density-based algorithm for discovering clusters in large spatial databases with noise*, Proceedings of the Second International Conference on Knowledge Discovery and Data Mining

[24] F. Rinaldi (2017), *Optimization for Data Science*, Università degli Studi di Padova

[25] Y. S. Abu-Mostafa, M. Magdon-Ismail, H. Lin (2012), *Learning From Data A short course*, AMLbook.com

[26] A. Gelman, J. Hill (2006), *Missing-data imputation* In *From data collection to model understanding to model checking*, Cambridge University, New York, (pag. 529-544)

[27] G. Louppe (2015), *Understanding Random Forest. From theory to practice*, University of Liège, Liège

[28] S. Mishra (2017), *Handling Imbalanced Data: SMOTE vs. Random Undersampling* In *International Research Journal of Engineering and Technology*

[29] L. Grippo, M. Sciandrone, *Metodi di ottimizzazione per le reti neurali*, Roma

[30] S. Marsland (2014), *Machine Learning: An Algorithmic Perspective Second Edition*, Chapman and Hall CRC, Londra

[31] J. Hurwitz, D. Kirsch (2018), *Machine Learning for dummies*, John Wiley and Sons, Hoboken

# Ringraziamenti

Anche se mi sembra ancora molto strano, sto inevitabilmente chiudendo un capitolo importante della mia vita, sicuramente faticoso ed impegnativo, che mi ha però permesso di conoscere un sacco di persone stupende e di confrontarmi con i miei limiti per superarli e raggiungere le mie aspirazioni. Naturalmente non ce l'avrei mai fatta da sola e per questo penso sia d'obbligo ringraziare tutti coloro che mi hanno permesso di raggiungere questo traguardo e che mi hanno supportata nei miei tanti momenti di disperazione.

In primo luogo vorrei ringraziare il prof. Francesco Rinaldi che per la prima volta mi ha dato la possibilità di sfruttare le conoscenze matematiche per affrontare un problema reale. La ringrazio, in particolare, per avermi incoraggiato e rassicurato, soprattutto nelle fasi iniziali, dandomi la sicurezza necessaria per portare a termine tale lavoro.

Un enorme GRAZIE va naturalmente ai miei genitori che hanno sempre appoggiato le mie scelte e hanno fatto il loro meglio per permettermi di inseguire i miei sogni. Non potrò mai ringraziarvi abbastanza per tutto quello che avete fatto per me e ricordare solo alcune ragioni per le quali vi sono grata sarebbe alquanto riduttivo. Per tale motivo mi limito a dirvi che anche se qualche diffettuccio ce lo avete, non avrei potuto desiderare genitori migliori!
Non posso poi dimenticarmi del mio fratellino che ormai tanto piccolo non è! Lo so Erik che non sono stata una sorella perfetta e che caratterialmente più diversi non potevamo essere ma i nostri momenti di complicità sono impagabili e non ti cambierei con nessuno al mondo, anche perché altrimenti chi mando a fare gli scherzi alla mamma!
Rimanendo sulla sfera familiare non posso non ringraziare i miei nonni. Grazie per tutti i vostri piccoli gesti quotidiani che però cambiano in meglio le miei giornate.
Grazie nonno Lino per le tue preghiere e per tutte le volte che mi trovavi a studiare in cucina e avevi paura di disturbare. Anche se non te lo ho mai detto sono regali preziosi le tue storie e i tuoi ricordi!
Grazie nonna Gianira perché mi hai sempre dimostrato che per te era importante tutto quello che facevo tanto che ti ricordavi quando avevo esame e ti assicuravi di chiedermi com'era andato!
Grazie nonna Pia perché con un po' di polenta fatta da te non si può proprio essere tristi!
Inoltre vorrei ringraziare tutti gli zii e i cugini che chi più da vicino chi meno mi hanno accompagnato per tutta la mia vita!

Questa volta non posso fare a meno di ringraziare anche Andrea. Lo so che una regina dei ghiacci che si rispetti non dovrebbe dilungarsi in tali carinerie ma, anche se magari non sempre lo dimostro, non posso negare che in tante occasioni non so cosa avrei fatto senza di te! Non è da tutti avere una persona che qualsiasi cosa accada è lì pronta a sostenerti!

Doveroso è dedicare qualche riga a tutti i miei amici, quelli che mi conoscono da tutta la vita e quellli che ho incontrato lungo il cammino. Lo so, ho rotto le scatole a tutti con le mie pare mentali ma se state leggendo queste righe vuol dire che mi sopportate ancora!
In particolare, vorrei ringraziare le "amiche di Sarcedo"; devo ammettere che mai avrei pensato di conoscervi all'asilo, con qualcuna perderci di vista per un po' e poi ritrovarci da "grandi" o quasi ma sono veramente felice di come sono andate le cose!
Grazie ai "Delfini curiosi" e annessi; con voi ho condiviso risate, vacanze e momenti di gioia indimenticabili! Da ognuno di voi ho ricevuto qualcosa di bello che mi porterò sempre con me!
Infine, non posso non citare gli amici dell'università che sono stati fondamentali per alleggerire le ore di studio e per superare i momenti di sconforto dopo un esame andato male. Un pensiero speciale va ad Anna ed Ilaria. Sapete entrambi che non so cosa avrei fatto senza di voi! Siete state i miei due soli che mi hanno permesso di andare avanti in qualsiasi situazione e che mi hanno sempre aiutato senza chiedere niente in cambio! Grazie!

Spero di essermi ricordati di tutti ma se non fosse così sappiate che sono estremamente certa che non sarei riuscita a fare neanche la metà di tutto ciò se non avessi avuto un sacco di persone che mi hanno aiutato e sostenuto!