



UNIVERSITÀ DI PERUGIA
Dipartimento di Matematica e Informatica



TESI TRIENNALE IN INFORMATICA

Tecniche di Machine Learning per la Composizione di Sacche di Nutrizione Parenterale

Relatori

Dott. Valentina Poggioni

Dott. Valentina Franzoni

Candidato

Iuri Fulgidi

Anno Accademico 2023-2024

Sommario

Contesto Attualmente, negli ospedali, la composizione delle sacche parenterali per i neonati prematuri è un processo altamente manuale e complesso, basato sull'esperienza clinica e sulle linee guida nutrizionali. Non esiste ancora un sistema automatizzato o assistito da intelligenza artificiale che supporti i medici e i farmacisti nella definizione ottimale delle sacche in modo rapido e accurato. Tuttavia, la complessità del problema e la possibilità di definire regole matematiche precise per il calcolo dei nutrienti necessari rendono questo un campo promettente per l'applicazione di tecnologie avanzate come le reti neurali.

Obiettivo L'obiettivo di questa tesi è sviluppare un sistema basato su intelligenza artificiale, in particolare reti neurali, che assista i professionisti sanitari nella composizione delle sacche parenterali per i neonati. Il sistema sarà progettato per apprendere dalle decisioni prese in passato, analizzando i dati storici relativi a pazienti simili, e fornire raccomandazioni ottimizzate per le nuove formulazioni.

Metodologia Per raggiungere questi obiettivi, la tesi utilizzerà una combinazione di tecniche di machine learning e analisi dei dati clinici. Le reti neurali verranno addestrate utilizzando dataset esistenti di composizioni di sacche parenterali e i relativi risultati clinici per i neonati prematuri. Il modello si baserà su parametri chiave come peso, età gestazionale e bilanci elettrolitici, per fornire raccomandazioni sulla composizione ideale delle sacche.

Risultati Sono stati sviluppati più modelli predittivi capaci di stimare con precisione il contenuto delle sacche, basandosi su vari dataset e strategie di allenamento implementate. I modelli hanno dimostrato di raggiungere un'adequata accuratezza nella previsione dei valori, grazie a una combinazione ottimizzata di tecniche di preprocessing dei dati e approcci di machine learning. I risultati hanno evidenziato che le prestazioni dei modelli variano a seconda della complessità del dataset, con alcuni modelli che hanno mostrato una capacità predittiva superiore in termini di precisione.

Conclusioni Nonostante i modelli sviluppati abbiano mostrato una buona capacità predittiva, è fondamentale che le sacche ottenute tramite questi algoritmi vengano comunque sottoposte a una valutazione finale da parte di medici esperti. Il coinvolgimento umano è cruciale per garantire la sicurezza e l'efficacia delle soluzioni proposte. Un importante obiettivo futuro sarà quello di migliorare ulteriormente questi modelli, puntando non solo a replicare il passato, ma a proporre formulazioni di sacche potenzialmente migliori di quelle storiche.

Indice

1	Introduzione	6
1.1	La nutrizione parenterale nei neonati prematuri	6
1.2	Il potenziale dell'Intelligenza Artificiale	7
1.3	Obiettivi della tesi	7
2	La Nutrizione Parenterale nei Neonati Prematuri	9
2.1	I neonati prematuri e le sfide nutrizionali	9
2.2	Definizione e composizione delle sacche parenterali	10
2.3	Personalizzazione delle sacche parenterali	10
2.4	Necessità di innovazione nella gestione nutrizionale	11
3	Analisi Statistica dei Dati	12
3.1	Descrizione del Dataset	12
3.2	Analisi Statistica Descrittiva	14
3.2.1	Statistiche di base	14
3.2.2	Outliers	15
3.2.3	Statistiche con outliers rimossi	17
3.2.4	Correlazioni tra variabili	18
3.3	Percentili di crescita	21
3.3.1	Risultati dell'applicazione	22
3.4	Creazione di Dataset Derivati per il Training	23
4	Rete Neurale	25
4.1	Caricamento dei dati	25
4.1.1	Divisione training e test	26
4.1.2	Preprocessing sui dati	27
4.2	Architettura della Rete Neurale	27
4.2.1	Inizializzazione dei pesi	28
4.3	Allenamento del Modello	29
4.3.1	Early Stopping	30
4.4	Cross-validation	32

4.5	Valutazione delle prestazioni	33
5	Esperimenti	36
5.1	Senza gestione degli outliers	36
5.2	Risultati dopo la gestione degli outliers	37
5.2.1	Secondo Z-score	37
5.2.2	Secondo percentile	37
5.2.3	Secondo entrambe	38
5.3	Esperimento 1	38
5.3.1	Risultati senza gestione degli outliers	38
5.3.2	Outliers definiti tramite Z-score	39
5.3.3	Outliers definiti tramite Percentili	39
5.3.4	Outliers definiti tramite entrambe le precedenti	40
5.3.5	Conclusioni	40
5.4	Esperimento 2	40
5.4.1	Senza gestione degli outliers	41
5.4.2	Outliers definiti tramite Z-score	41
5.4.3	Outliers definiti tramite Percentili	42
5.4.4	Outliers definiti tramite entrambe le precedenti	42
5.5	Esperimento 3	43
5.5.1	EG bassa	43
5.5.2	EG media	45
5.5.3	EG alta	47
5.5.4	Conclusioni	49
5.6	Esperimento 4	50
5.6.1	Senza gestione degli outliers	50
5.6.2	Outliers definiti tramite Z-score	50
5.6.3	Outliers definiti tramite Percentili	51
5.6.4	Outliers definiti tramite entrambe le precedenti	51
5.7	Confronto	52
6	Conclusioni	53

Capitolo 1

Introduzione

La nutrizione parenterale (NP) rappresenta una risorsa vitale per il supporto nutrizionale di pazienti che non possono assumere nutrienti per via enterale, tra cui i neonati prematuri. Questi bambini, a causa della loro immaturità fisiologica, non sono in grado di nutrirsi adeguatamente attraverso i normali canali digestivi, rendendo necessario un intervento nutrizionale tramite soluzioni sterili somministrate direttamente nel flusso sanguigno. Le sacche parenterali, contenenti un mix personalizzato di nutrienti essenziali come glucosio, aminoacidi, lipidi, elettroliti e vitamine, sono fondamentali per garantire il corretto sviluppo e la crescita di questi pazienti vulnerabili. Tuttavia, la preparazione di queste sacche è un processo complesso e altamente personalizzato, che richiede precisione e una profonda conoscenza del fabbisogno nutrizionale specifico di ogni neonato.

In questo contesto, l'intelligenza artificiale (IA) può supportare i professionisti del campo nella composizione e ottimizzazione delle sacche parenterali. Grazie alla sua capacità di elaborare grandi quantità di dati e di identificare schemi complessi, l'IA può migliorare sia l'efficienza che l'accuratezza nel calcolo dei dosaggi nutrizionali. In altre parole, questo potrebbe non solo permettere una formulazione precisa delle sacche in base alle condizioni cliniche specifiche del neonato, ma anche una maggiore rapidità del processo, con potenziali benefici in termini di risultati clinici.

1.1 La nutrizione parenterale nei neonati prematuri

I neonati prematuri presentano sfide nutrizionali uniche. Il loro sistema gastrointestinale, immaturo e spesso incapace di processare il cibo per via enterale, li rende dipendenti dalla nutrizione parenterale totale (NPT) o parziale per periodi prolungati. L'obiettivo primario della NP nei prematuri è garantire una crescita e uno sviluppo simili a quelli che avrebbero avuto in utero, fornendo un apporto equilibrato di macro e micronutrienti

necessari per sostenere i loro elevati fabbisogni metabolici.

Il processo di formulazione delle sacche parenterali per questi pazienti richiede una pianificazione meticolosa da parte dei medici e dei farmacisti, poiché deve tenere conto di una molteplicità di fattori, tra cui il peso alla nascita, la tolleranza nutrizionale, la funzionalità degli organi e le condizioni cliniche variabili. Una formulazione inadeguata o un errore nella composizione delle sacche metterebbe a rischio la vita del neonato.

1.2 Il potenziale dell'Intelligenza Artificiale

L'applicazione dell'intelligenza artificiale nel settore sanitario è in rapida espansione, e la nutrizione clinica non fa eccezione. Attraverso algoritmi avanzati di machine learning, reti neurali e altre tecnologie correlate, l'IA può facilitare decisioni cliniche più rapide ed efficienti. Nella specifica area della composizione delle sacche parenterali, l'IA può essere utilizzata per analizzare grandi quantità di dati clinici (come peso, età gestazionale, analisi biochimiche) e fornire raccomandazioni personalizzate sui dosaggi di nutrienti, ottimizzando la composizione delle sacche in base alle esigenze specifiche di ciascun paziente.

Un sistema di intelligenza artificiale ben addestrato potrebbe migliorare o velocizzare significativamente il processo decisionale medico, offrendo una maggiore adattabilità alle variazioni cliniche. Inoltre, l'IA ha il potenziale di gestire database di dati clinici, identificando correlazioni tra parametri nutrizionali e risultati clinici che potrebbero non essere immediatamente evidenti ai medici. Questo non solo ottimizzerebbe le soluzioni parenterali in modo personalizzato, ma potrebbe anche migliorare gli esiti a lungo termine, contribuendo alla crescita e allo sviluppo più sani di neonati prematuri.

1.3 Obiettivi della tesi

L'obiettivo principale di questa tesi è sviluppare un sistema basato su tecniche di machine learning per supportare i medici nella composizione delle sacche parenterali neonatali, utilizzate per fornire nutrimento ai neonati che non sempre possono alimentarsi per via orale o enterale.

La complessità della preparazione delle sacche parenterali deriva dalla necessità di garantire un bilancio accurato di nutrienti (come proteine, carboidrati, lipidi, elettroliti, e vitamine) su misura per le esigenze individuali di ciascun neonato, che variano in base a fattori quali peso, età gestazionale, condizione clinica e stato nutrizionale.

Nell'ambito di questa ricerca, si intendono perseguire i seguenti obiettivi:

Analisi statistica del dataset Il primo passo è l'analisi di un dataset clinico contenente composizioni storiche di sacche parenterali e informazioni sui pazienti neonatali associati. Questa analisi permetterà di identificare pattern e correlazioni tra le variabili rilevanti, come le caratteristiche demografiche e cliniche dei neonati e le quantità di nutrienti somministrate. Attraverso strumenti di analisi esplorativa e tecniche statistiche, l'obiettivo è ricavare informazioni chiave che possano guidare lo sviluppo dei modelli predittivi, come la determinazione delle dosi medie o le variabilità nelle composizioni delle sacche in base a caratteristiche specifiche dei pazienti.

Sviluppo e addestramento di una rete neurale predittiva In seguito all'analisi del dataset, le statistiche e le osservazioni derivanti saranno utilizzate per addestrare una rete neurale artificiale in grado di fornire suggerimenti personalizzati per la composizione delle sacche parenterali. Questo modello dovrà essere in grado di apprendere dalle informazioni cliniche dei pazienti (quali peso, età gestazionale, diagnosi) per suggerire dosaggi adeguati di nutrienti, assistendo il personale medico nella definizione di piani nutrizionali bilanciati e personalizzati. L'obiettivo è costruire un sistema che integri conoscenze cliniche e dati storici, fornendo suggerimenti predittivi che riducano il tempo impiegato e migliorino l'accuratezza delle terapie nutrizionali neonatali.

Valutazione delle prestazioni del modello e implementazione clinica Una volta completato l'addestramento, la rete neurale verrà valutata per verificarne l'efficacia e la precisione nel suggerire composizioni appropriate. Verranno utilizzati metodi di validazione e metriche standard, come l'errore medio assoluto (MAE) e il coefficiente di determinazione (R^2), per misurare la qualità delle previsioni. Se il modello dimostrerà un'accurata capacità predittiva, si esploreranno anche le potenziali modalità di integrazione clinica, valutando eventuali considerazioni etiche e logistiche per implementare il sistema in contesti ospedalieri.

In sintesi, questa ricerca mira a costruire un sistema che, sfruttando l'analisi dei dati e l'apprendimento automatico, possa migliorare la qualità delle cure neonatali tramite un supporto data-driven alla composizione delle sacche parenterali. Il raggiungimento di questi obiettivi consentirebbe di ridurre il carico di lavoro per il personale medico, contribuire al miglioramento della sicurezza e dell'efficacia delle terapie nutrizionali nei neonati più vulnerabili e, soprattutto, a ridurre il tempo che ogni giorno è necessario per la formulazione di ogni sacca.

Capitolo 2

La Nutrizione Parenterale nei Neonati Prematuri

2.1 I neonati prematuri e le sfide nutrizionali

La nascita pretermine, che avviene prima della 37esima settimana di gestazione, rappresenta una delle principali cause di mortalità e morbidità neonatale a livello globale[3]. I neonati prematuri affrontano numerose difficoltà legate alla loro immaturità fisiologica, in particolare nel campo della nutrizione. La loro capacità di assumere e digerire nutrienti per via enterale è spesso compromessa, a causa dell'immaturità dell'apparato gastrointestinale e dell'incapacità di coordinare suzione e deglutizione, essenziale per l'allattamento. Questo rende necessaria l'introduzione di una nutrizione parenterale (NP) per garantire il corretto apporto di nutrienti e promuovere una crescita e uno sviluppo ottimali. Anche quando un neonato è in grado di assumere alcuni nutrienti per via enterale, la nutrizione parenterale rappresenta un supporto fondamentale per garantire un apporto nutrizionale completo.

Il fabbisogno nutrizionale di un neonato prematuro è complesso. I neonati prematuri hanno una richiesta calorica elevata, necessaria a compensare l'insufficiente accumulo di grasso e nutrienti avvenuto durante la gestazione[2]. Inoltre, devono affrontare rischi maggiori di deficit nutrizionali, tra cui insufficienza proteica, alterazioni elettrolitiche, e carenze di vitamine e minerali essenziali, che possono avere conseguenze negative sullo sviluppo neurologico e sulla crescita generale. A causa di queste complessità, la nutrizione parenterale rappresenta un aspetto fondamentale delle cure intensive neonatali.

2.2 Definizione e composizione delle sacche parenterali

La nutrizione parenterale è un metodo di somministrazione di nutrienti direttamente nel circolo sanguigno, evitando completamente l'apparato digerente. Le sacche parenterali contengono una miscela equilibrata di nutrienti fondamentali per la crescita, tra cui:

- Glucosio: fonte primaria di energia per sostenere il metabolismo del neonato.
- Aminoacidi: essenziali per la sintesi proteica e la crescita tissutale.
- Lipidi: forniscono energia concentrata e sono fondamentali per lo sviluppo cerebrale.
- Elettroliti: come sodio, potassio, cloro e calcio, necessari per il mantenimento dell'equilibrio idrico ed elettrolitico.

La formulazione di una sacca parenterale deve essere attentamente personalizzata per ogni neonato, tenendo conto di variabili quali il peso alla nascita, l'età gestazionale, la crescita attesa e le condizioni cliniche[1]. Questo processo richiede un molto impegno da parte dei medici neonatologi, i quali devono calcolare con precisione le dosi di ogni componente, monitorando regolarmente i parametri biochimici del paziente per eventuali aggiustamenti.

2.3 Personalizzazione delle sacche parenterali

La composizione delle sacche parenterali per i neonati prematuri è fortemente personalizzata e deve essere modificata frequentemente per adattarsi ai rapidi cambiamenti nelle condizioni cliniche del paziente. Alcuni dei fattori chiave che influenzano la personalizzazione includono:

- Peso corporeo e crescita: La formulazione deve adattarsi ai cambiamenti nel peso del neonato, in quanto i fabbisogni energetici e proteici variano rapidamente nei primi giorni e settimane di vita.
- Funzionalità renale ed epatica: I dosaggi di elettroliti, proteine e farmaci devono essere attentamente bilanciati in base alla capacità di filtrazione renale e alla funzionalità epatica del paziente

Questa continua necessità di personalizzazione rende il processo di formulazione delle sacche NP complesso e laborioso, richiedendo un costante aggiornamento delle linee guida nutrizionali e un'adeguata formazione del personale medico. Inoltre c'è un rilevante dispendio di tempo quotidiano da parte dei medici per l'inserimento manuale dei dati per la nutrizione, vista anche la necessità di gestione di un gran numero di variabili cliniche rende questo compito estremamente impegnativo, specialmente nelle unità di terapia intensiva neonatale.

2.4 Necessità di innovazione nella gestione nutrizionale

La crescente complessità della gestione nutrizionale dei neonati prematuri evidenzia la necessità di soluzioni innovative per migliorare l'efficienza e la sicurezza del processo di composizione delle sacche parenterali. La personalizzazione manuale, sebbene indispensabile, comporta un elevato dispendio in termini di risorse umane. In questo contesto, l'integrazione di tecnologie avanzate, per lo sviluppo di un sistema di supporto alle decisioni, potrebbe rappresentare una svolta significativa.

L'IA ha il potenziale di analizzare grandi quantità di dati clinici in modo più rapido e accurato rispetto ai metodi tradizionali, migliorando la capacità di personalizzare le terapie nutrizionali in base alle esigenze specifiche di ciascun neonato. Con l'uso di modelli predittivi avanzati e algoritmi di apprendimento automatico, l'IA potrebbe contribuire a ridurre il tempo impiegato per definire ogni formulazione, migliorare gli esiti clinici e ottimizzare l'uso delle risorse in ambito ospedaliero.

Capitolo 3

Analisi Statistica dei Dati

Per lo sviluppo di un modello di intelligenza artificiale basato su reti neurali per la composizione delle sacche parenterali nei neonati, è essenziale disporre di un dataset completo e accurato su cui effettuare l'addestramento del modello. In questo capitolo, verrà presentata un'analisi statistica approfondita del dataset utilizzato, con l'obiettivo di descrivere la struttura dei dati, identificare le variabili più rilevanti e analizzare le relazioni statistiche tra di esse. Questa fase preliminare è fondamentale per comprendere il contesto in cui operano le reti neurali e per identificare eventuali pattern, tendenze o correlazioni che possono influenzare il processo decisionale automatizzato.

Il dataset utilizzato in questo progetto comprende dati clinici relativi a neonati, spesso prematuri, che hanno ricevuto nutrizione parenterale o una combinazione di nutrizione parenterale e nutrizione enterale (NE), così come le specifiche delle sacche nutrizionali somministrate. Le variabili chiave includono informazioni demografiche e parametri biochimici del neonato, la composizione delle sacche, ovvero i nutrienti somministrati, ed è possibile ricavare gli esiti clinici, ad esempio, crescita ponderale, squilibri elettrolitici, calcolando o confrontando la differenza rispetto ai giorni precedenti, rendendo quindi possibile non solo avere dei record con valori giornalieri isolati ma ottenere anche uno storico di ogni paziente al cui interno sono presenti tutti valori di analisi e nutrienti somministrati nell'arco del periodo di ricovero con i relativi risultati.

3.1 Descrizione del Dataset

Il dataset, fornito dal reparto di Neonatologia dell'Ospedale "Santa Maria della Misericordia di Perugia", è formato dalla raccolta di una serie di cartelle cliniche ospedaliere relative a neonati (le informazioni sono state anonimizzate per proteggerne la privacy) in terapia intensiva, trattati con nutrizione parenterale per diversi periodi di tempo. In totale, il dataset comprende 18.163 record relativi a esattamente 1121 pazienti diversi.

Le variabili raccolte nel dataset possono essere suddivise in tre categorie:

- Variabili demografiche e cliniche:
 - Sesso
 - Giorni di vita
 - Giorno di vita di inizio della nutrizione parenterale
 - Età gestazionale (settimane)
 - Peso attuale (Kg)
 - Sodio plasmatico (mEq/L)
 - Potassio plasmatico (mEq/L)
 - Cloro plasmatico (mEq/L)
 - Calcio ionizzato plasmatico (mM/L)
- Valori Nutrizione Parenterale:
 - Volume (mL)
 - Glucosio (mL)
 - Amminoacidi (mL)
 - Lipidi (mL)
 - Sodio (mEq/Kg)
 - Potassio (mEq/Kg)
 - Cloro (mEq/Kg)
 - Calcio (mg/Kg)
 - Fosforo (mg/Kg)
 - Magnesio (mg/Kg)
- Valori Nutrizione Enterale:
 - Volume (mL)
 - Calorie (cal)
 - Proteine (g/kg)

3.2 Analisi Statistica Descrittiva

3.2.1 Statistiche di base

Di seguito viene riportata una panoramica statistica delle variabili più importanti nel dataset, in particolare i valori minimo, massimo, medio e deviazione standard (SD) calcolati sul dataset completo.

Variabile	Minimo	Massimo	Media	SD
giorniDiVita	0.0	781.0	27.681	54.84
InizioNP	0.0	775.0	11.409	50.503
EG	21.85	42.0	30.748	4.628
Peso	0.33	25.0	1.836	1.19
Na_plasm	0.0	114039.0	117.875	1199.768
K_plasm	0.0	140.0	3.359	2.856
Cl_plasm	0.0	1050.0	77.434	45.18
Caion_plasm	0.0	139.2	1.062	3.3
VolumeNP	13.5	1600.0	193.434	154.373
Glucosio50	0.0	250.0	34.286	24.479
Aminoacidi06	0.0	445.0	65.837	42.459
Lipidi20	0.0	101.49	13.438	9.916
na_kg	0.0	13.39	2.14	1.393
k_kg	0.0	7.25	1.734	1.14
cl_kg	0.0	67.97	1.066	1.87
ca_kg	0.0	157.43	49.492	18.672
p_lg	0.0	150.22	30.623	13.067
mg_kg	0.0	150	1.546	0.871
VolumeNE	0.0	720.0	68.769	97.947
CalNE	0.0	119.1	39.797	35.848
ProtNE	0.0	4.3	0.997	0.992

Tabella 3.1: Statistiche

Questi dati offrono una panoramica generale della popolazione neonatale considerata, evidenziando un ampio range di età gestazionale, peso alla nascita e giorni di vita, caratteristiche tipicamente importati per la valutazione della salute di neonati prematuri. I valori rientrano generalmente nei limiti normali, ma con alcune osservazioni di valori fuori dal range previsto.

3.2.2 Outliers

Passiamo ora al controllo su possibili outliers, o valori anomali secondo due diverse definizioni:

1. Un approccio semplice è identificare come outliers i valori che si trovano nelle code estreme della distribuzione, ovvero i valori che si trovano sotto il 1° o oltre il 99° percentile.
2. Lo Z-score misura quante deviazioni standard un dato punto è lontano dalla media. Un valore con uno Z-score molto alto o molto basso può essere considerato un outlier.

Formula dello Z-score: $Z = \frac{X - \mu}{\sigma}$

dove:

- X è il valore osservato
- μ è la media del dataset
- σ è la deviazione standard

Per le nostre considerazioni, un valore con uno Z-score superiore a 3 o inferiore a -3 viene considerato un outlier.

Di seguito in tabella viene mostrato il numero totale e la percentuale degli outliers secondo le due diverse definizioni, ricordando che il numero totale di record è circa 18.100, la colonna "Percentuale" è data dal rapporto tra il totale degli outlier della variabile secondo la definizione specificata, rispetto al totale dei record

Variabile	Metodo	Totale	Percentuale
giorniDiVita	Z-Score	298	1.6%
giorniDiVita	Percentili	182	1.0%
InizioNP	Z-Score	346	1.9%
InizioNP	Percentili	141	0.8%
EG	Z-Score	0	0.0%
EG	Percentili	336	1.8%
Peso	Z-Score	252	1.4%
Peso	Percentili	348	1.9%
Na_plasm	Z-Score	2	0.0%
Na_plasm	Percentili	180	1.0%
K_plasm	Z-Score	12	0.1%
K_plasm	Percentili	181	1.0%
Cl_plasm	Z-Score	1	0.0%

Tabella 3.2: Outliers

Variabile	Metodo	Totale	Percentuale
Cl_plasm	Percentili	168	0.9%
Caion_plasm	Z-Score	14	0.1%
Caion_plasm	Percentili	180	1.0%
VolumeNP	Z-Score	298	1.6%
VolumeNP	Percentili	344	1.9%
Glucosio50	Z-Score	276	1.5%
Glucosio50	Percentili	359	2.0%
Aminoacidi06	Z-Score	283	1.6%
Aminoacidi06	Percentili	360	2.0%
Lipidi20	Z-Score	252	1.4%
Lipidi20	Percentili	176	1.0%
na_kg	Z-Score	328	1.8%
na_kg	Percentili	182	1.0%
k_kg	Z-Score	165	0.9%
k_kg	Percentili	182	1.0%
cl_kg	Z-Score	338	1.9%
cl_kg	Percentili	181	1.0%
ca_kg	Z-Score	36	0.2%
ca_kg	Percentili	181	1.0%
p_kg	Z-Score	41	0.2%
p_kg	Percentili	182	1.0%
mg_kg	Z-Score	189	1.0%
mg_kg	Percentili	181	1.0%
VolumeNE	Z-Score	357	2.0%
VolumeNE	Percentili	179	1.0%
CalNE	Z-Score	0	0.0%
CalNE	Percentili	177	1.0%
ProtNE	Z-Score	1	0.0%
ProtNE	Percentili	181	1.0%

Tabella 3.3: Outliers

Si può apprezzare come in generale, usando lo Z-score per alcune variabili non vengono rilevati valori anomali visto che può esserci una variabilità bassa senza particolare discostamento dalla media, mentre usando i percentili ci sarà sempre qualche valore etichettato come anomalo per definizione, i percentili sono infatti calcolati su dati presenti esclusivamente nel dataset studiato e perciò ci saranno valori in tutti i cento percentili, inclusi gli estremi.

3.2.3 Statistiche con outliers rimossi

Una volta rimossi gli outliers secondo le due diverse definizioni, andiamo ad ottenere due diversi dataset, derivati da quello originale, che hanno caratteristiche differenti.

Con la definizione secondo Z-score riduciamo il numero di record totali a 15752, con le seguenti statistiche:

Variabile	Minimo	Massimo	Media	SD
giorniDiVita	0.0	178.0	18.798	22.838
InizioNP	0.0	140.0	4.346	14.856
EG	21.85	42.0	30.55	4.472
Peso	0.33	5.4	1.652	0.853
Na_plasm	0.0	1462.0	106.043	62.741
K_plasm	0.0	9.3	3.336	1.957
Cl_plasm	0.0	204.0	78.094	44.351
Caion_plasm	0.0	9.8	0.986	0.607
VolumeNP	0.0	650.4	173.25,	116.157
Glucosio50	0.0	107.5	30.871	19.141
Aminoacidi06	0.0	192.01	60.657	34.816
Lipidi20	0.0	43.0	12.13	7.765
na_kg	0.0	6.3	2.014	1.095
k_kg	0.0	5.15	1.666	1.042
cl_kg	0.0	6.25	0.841	1.319
ca_kg	0.0	105.0	49.842	18.066
p_lg	0.0	69.46	30.853	12.597
mg_kg	0.0	4.1	1.48	0.746
VolumeNE	0.0	360.0	63.547	79.967
CalNE	0.0	99.64	41.314	35.622
ProtNE	0.0	3.59	1.041	0.997

Tabella 3.4: Statistiche senza outliers Z-score

Si nota che anche per le variabili secondo cui il Z-score non è presente alcun valore anomalo, ci sono comunque delle variazioni nelle statistiche, questo è dovuto al fatto che nonostante un record possa avere un valore nella norma per la variabile senza outlier nell'intero dataset, in alcuni casi ci può essere un'altra variabile relativa allo stesso record che ha invece un valore anomalo secondo i propri range.

Con la definizione secondo i percentili riduciamo il numero di record totali a 14524, con le seguenti statistiche:

Variabile	Minimo	Massimo	Media	SD
giorniDiVita	0.0	247.0	20.925	27.002
InizioNP	0.0	236.0	5.806	20.125
EG	21.85	41.0	30.852	4.339
Peso	0.33	5.7	1.739	0.904
Na_plasm	0.0	150.0	104.144	59.044
K_plasm	0.0	6.4	3.271	1.948
Cl_plasm	0.0	117.0	76.872	44.68
Caion_plasm	0.0	1.56	0.969	0.591
VolumeNP	0.0	767.0	184.22,	123.501
Glucosio50	0.0	114.2	32.776	20.255
Aminoacidi06	0.0	203.58	64.244	36.163
Lipidi20	0.0	45.0	12.881	8.147
na_kg	0.0	7.23	2.072	1.153
k_kg	0.0	5.0	1.704	1.028
cl_kg	0.0	7.49	0.913	1.407
ca_kg	0.0	87.24	17.262	17.262
p_lg	0.0	59.52	30.647	11.92
mg_kg	0.0	4.17	1.499	0.736
VolumeNE	0.0	455.0	67.411	87.073
CalNE	0.0	97.0	41.65	35.416
ProtNE	0.0	3.23	1.02	0.961

Tabella 3.5: Statistiche senza outliers Percentili

Si può apprezzare come, con entrambe le definizioni, la deviazione standard diminuisca notevolmente nella maggior parte delle variabili. Inoltre nelle variabili in cui le due definizioni danno un conteggio abbastanza diverso di outliers, per esempio nelle calorie della nutrizione enterale in cui non sono presenti secondo lo Z-score, mentre sono circa l'1% secondo il percentile, le statistiche finali sono molto simili.

3.2.4 Correlazioni tra variabili

È stata condotta un'analisi di correlazione tra le variabili biochimiche e gli esiti clinici per identificare potenziali pattern utili per l'addestramento delle reti neurali. Di particolare interesse sono le relazioni tra i nutrienti somministrati e la crescita ponderale, poiché

queste informazioni possono essere critiche per ottimizzare le formulazioni delle sacche.

Per questo motivo, in ogni record sono state aggiunte le informazioni riguardanti i sette giorni precedenti, in modo da avere uno storico parziale di ogni paziente in ogni record, inoltre sono stati calcolati il cambiamento di peso, il cambiamento di percentile e il cambiamento di percentuale di peso nel giorno successivo, in modo da avere gli effetti della terapia.

La matrice di correlazione include esclusivamente le correlazioni tra i dati giornalieri e quelli relativi ai tre giorni precedenti, sebbene nel dataset siano presenti dati storici fino al settimo giorno precedente. Questo approccio è stato adottato in quanto le correlazioni più forti e rilevanti si concentrano prevalentemente entro i primi tre giorni, mentre le informazioni dei giorni successivi tendono a mostrare correlazioni più deboli o trascurabili rispetto alle analisi correnti. Pertanto, la matrice enfatizza le connessioni temporali più significative, semplificando l'interpretazione e mantenendo il focus sui dati con maggiore impatto predittivo e rilevanza analitica

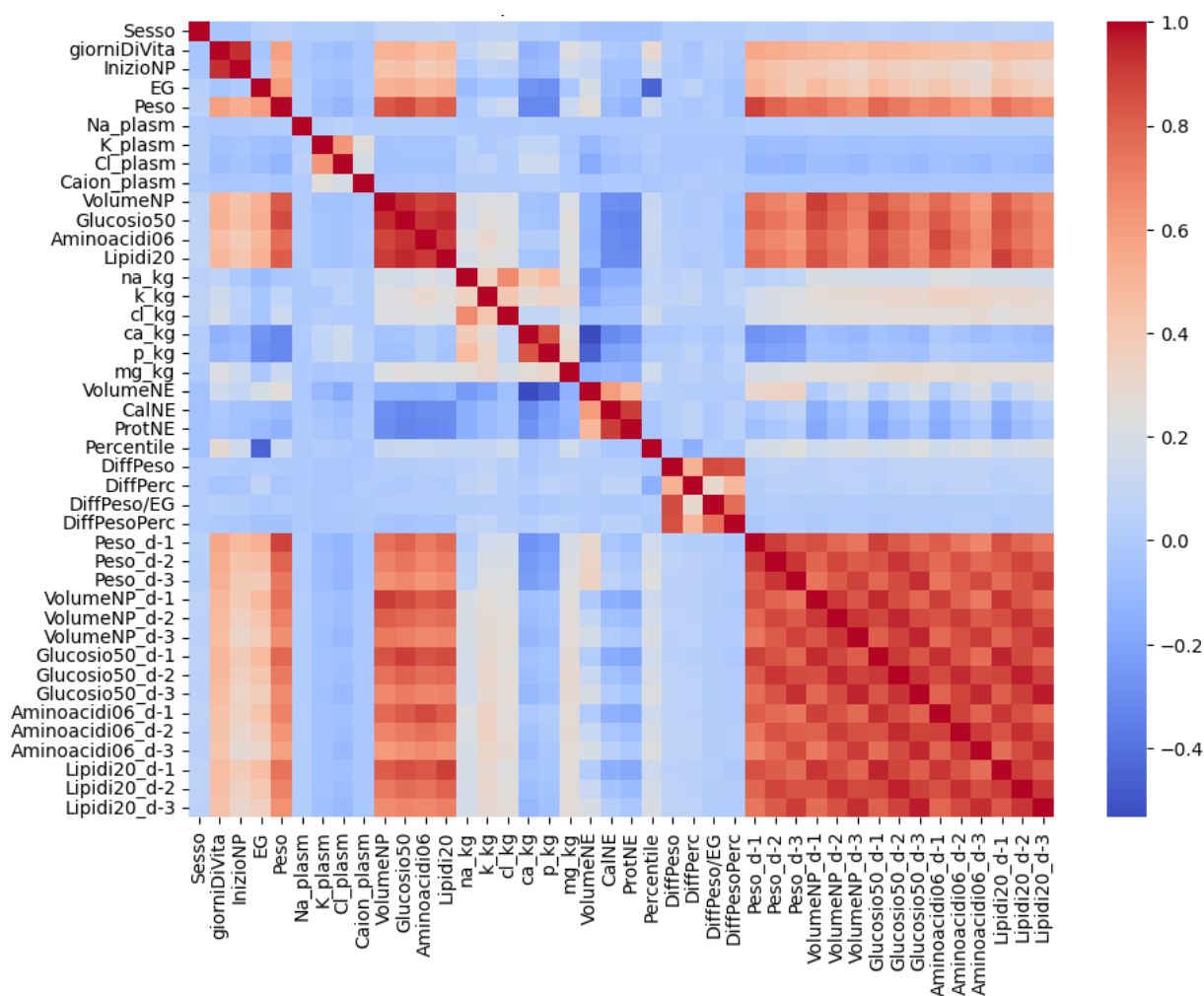


Figura 3.1: Correlazioni

Questa matrice suggerisce alcune correlazioni significative: in particolare una forte correlazione tra volume della sacca di nutrizione parenterale e i tre valori dei macro nutrienti della stessa, e ognuno di questi quattro valori con l'analogo del giorno precedente. Questo suggerisce una forte relazione tra i valori che c'è da aspettarsi in quanto è dovuta al bilanciamento fatto per le sacche, è quindi naturale che questi quattro valori siano dipendenti l'uno dall'altro perché confermano il dovuto equilibrio della sacca

Elenchiamo quindi le 20 correlazioni più forti:

- Glucosio50 con Lipidi20: 0.9476
- Glucosio50 con VolumeNP: 0.9405
- Glucosio50 con Aminoacidi06: 0.9268
- Lipidi20 con Aminoacidi06: 0.9123
- VolumeNP con Lipidi20: 0.9099
- VolumeNP con VolumeNP_d-1: 0.9004
- Glucosio50 con Glucosio50_d-1: 0.8994
- Lipidi20 con Lipidi20_d-1: 0.8923
- Aminoacidi06 con VolumeNP: 0.8858
- Aminoacidi06 con Aminoacidi06_d-1: 0.8722
- Lipidi20 con Glucosio50_d-1: 0.8656
- Glucosio50 con VolumeNP_d-1: 0.8634
- Glucosio50 con Peso: 0.8628
- Glucosio50 con Lipidi20_d-1: 0.855
- Lipidi20 con VolumeNP_d-1: 0.845
- Aminoacidi06 con Glucosio50_d-1: 0.845
- VolumeNP con Glucosio50_d-1: 0.8424
- Aminoacidi06 con Lipidi20_d-1: 0.8294
- Glucosio50 con Aminoacidi06_d-1: 0.8221
- VolumeNP con Peso: 0.8212

Si osserva che la variabile con le correlazioni più forti e numerose rispetto alle altre è il glucosio, il che la rende leggermente più prevedibile rispetto alle altre variabili. Tuttavia, anche le correlazioni più deboli restano elevate, riflettendo la natura del dataset. È importante sottolineare che l'analisi ha preso in considerazione i dati fino a sette giorni prima, ma ha evidenziato una correlazione più marcata con i valori più recenti.

Le correlazioni più elevate, comprese tra lo 0.60 e lo 0.95, riguardano soprattutto lo storico recente delle stesse quattro variabili e il peso, mentre tutti gli altri parametri, relativi ad analisi e somministrazioni meno recenti, mostrano correlazioni significativamente inferiori.

In particolare, l'età gestazionale e i giorni di vita presentano correlazioni simili a quelle osservate per i valori tra il sesto e il settimo giorno delle quattro variabili di interesse, con valori che variano tra lo 0.46 e lo 0.54. Per quanto riguarda i micronutrienti della sacca, si osservano correlazioni tra lo 0.20 e lo 0.30 per i valori recenti di potassio, magnesio e cloro, e tra lo 0.10 e lo 0.20 per i valori meno recenti degli stessi elettroliti. I valori della nutrizione enterale, ad eccezione del volume, mostrano invece una correlazione negativa intorno a -0.30, mentre parametri come variazioni di peso, percentili o elettroliti plasmatici presentano correlazioni molto deboli.

Le alte correlazioni tra le quattro variabili nel tempo si spiegano con il fatto che, una volta assegnata una sacca a un paziente, le sue condizioni tendono a non subire variazioni drastiche nel breve termine. Di conseguenza, anche i contenuti della sacca vengono modificati solo minimamente. Inoltre, si riscontra una leggera correlazione con gli elettroliti, dovuta alla necessità di mantenere un equilibrio tra macro e micronutrienti. Infine, la correlazione negativa con la nutrizione enterale si giustifica con il fatto che i nutrienti somministrati tramite questo metodo riducono la quantità di nutrienti richiesti dalla nutrizione parenterale.

3.3 Percentili di crescita

Un aspetto cruciale nell'analisi dei dati clinici relativi ai neonati prematuri è la valutazione delle loro misure rispetto a standard di crescita e salute, spesso espressi in percentili. I percentili forniscono una misura relativa che confronta un neonato con una popolazione di riferimento, permettendo di collocare ogni paziente all'interno di una distribuzione statistica, il che è particolarmente utile per monitorare l'evoluzione clinica.

Generalmente in pediatria è usato quello relativo a peso, altezza, circonferenza cranica, per valutare l'accrescimento del bambino. Nel nostro caso specifico andiamo ad usarlo come criterio di valutazione per il peso di ogni bambino.

In questa tesi, il percentile relativo al peso è stato calcolato per ogni neonato utilizzando le formule tratte da uno studio pubblicato su "The Lancet Global Health"[4], che fornisce un modello per stimare la crescita attesa nei neonati prematuri in funzione dell'età gestazionale e del peso alla nascita. La formula utilizzata per il calcolo del percentile è la seguente:

$$P^{th}centile = exp(Median + K * SD) \quad (3.1)$$

Ove:

$$Median = 2.591277 - 0.01155 * EG^{0.5} - 2201.705 * EG^{-2} + 0.0911639 * Sesso \quad (3.2)$$

$$SD = 0.1470258 + 505.92394 * EG^{-2} - 140.0576 * EG^{-2} * log(EG) \quad (3.3)$$

K è la deviazione normale equivalente, corrispondente al particolare percentile.

Il sesso è codificato con 1 per i maschi e 0 per le femmine.

3.3.1 Risultati dell'applicazione

Usando le formule del percentile date, sul nostro dataset abbiamo ottenuto questa distribuzione dei percentili nel primo e nell'ultimo giorno di ricovero:

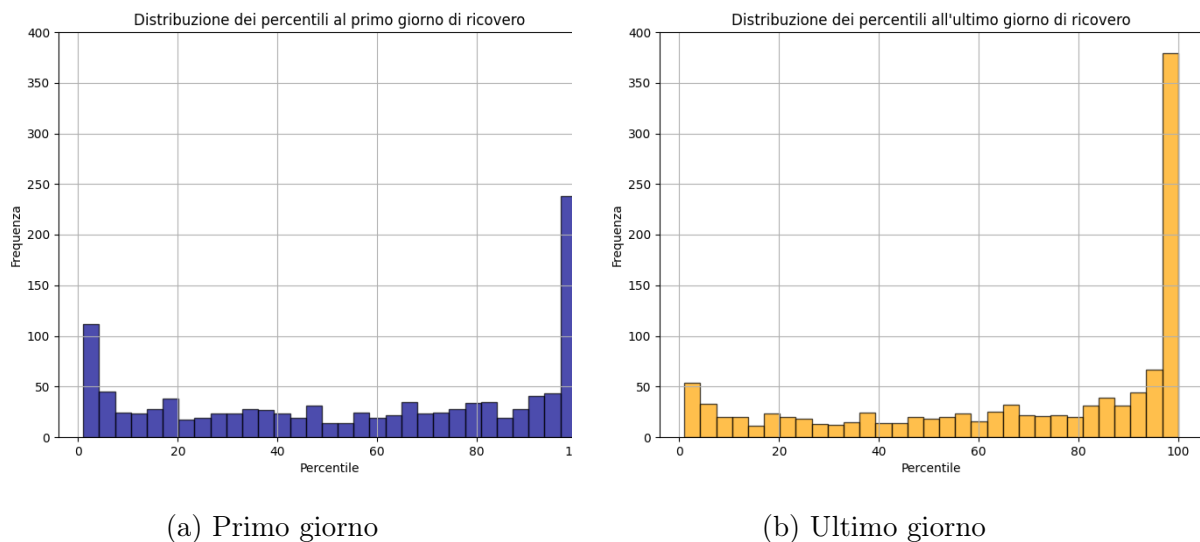


Figura 3.2: Confronto Percentili

Si può osservare un generale spostamento della distribuzione verso i percentili più alti, il che indica che in generale c'è una crescita del peso durante il ricovero. Tuttavia, la distribuzione osservata presenta una deviazione significativa da quella attesa. La distribuzione prevista dei percentili dovrebbe approssimarsi a una distribuzione normale, con la maggior parte dei valori concentrati intorno alla media e una decrescita graduale

verso i valori estremi. Al contrario, si osserva una concentrazione anomala agli estremi della distribuzione, con circa un quinto dei casi che ricade nel centesimo percentile e circa un decimo nel primo percentile, mentre i valori rimanenti sono distribuiti in modo relativamente uniforme negli altri percentili. Questo comportamento è in netto contrasto con la distribuzione gaussiana attesa, che prevederebbe pochi casi agli estremi e la maggior parte concentrata attorno alla mediana.

Data la natura dei dati, che trattano di casi di pazienti con problematiche cliniche, si potrebbe supporre uno squilibrio verso i percentili più bassi in quanto la salute potrebbe essere compromessa, e di conseguenza il peso che va a definire il percentile sarà più basso. Al contrario si può notare come ci sia una distribuzione che mostra un'eccessiva distribuzione in entrambi gli estremi, in particolare lo spicchio più grande è preso proprio dal centesimo percentile, il che indicherebbe una popolazione con un peso molto alto rispetto al dovuto.

3.4 Creazione di Dataset Derivati per il Training

Il dataset originale, contenente tutte le informazioni cliniche, biochimiche e nutrizionali dei circa 1000 neonati prematuri, è stato successivamente suddiviso in diversi sotto-dataset con l'obiettivo di allenare e confrontare le performance di modelli di rete neurale distinti. Ogni sotto-dataset si focalizza su aspetti specifici o su differenti approcci di preprocessing dei dati, al fine di testare le capacità predittive delle reti su vari input.

I principali dataset creati sono stati:

- Dataset completo (originale): Contiene tutte le variabili (cliniche, biochimiche e nutrizionali), senza modifiche sostanziali. Questo dataset viene utilizzato come baseline per confrontare gli altri modelli.
- Dataset con ricoveri lunghi: In questo dataset sono presenti solo dati riguardanti ricoveri che durano almeno 7 giorni, in modo da escludere informazioni sul breve termine che potrebbero avere comportamenti anomali.
- Dataset segmentato per età gestazionale: Questo dataset è stato suddiviso in base alle fasce di età gestazionale:
 - neonati estremamente prematuri (<28.5 settimane)
 - prematuri molto pretermine (28.5-32.5 settimane)
 - prematuri tardivi (>32.5 settimane)

L'idea è che reti neurali specializzate potrebbero performare meglio se addestrate su popolazioni con caratteristiche separate.

- Dataset filtrato per variabili significative: Dopo l'analisi delle correlazioni, è stata notata un gruppo di variabili con una forte correlazione con la composizione della sacca da 0.60 a 0.90, sono quindi state selezionate le variabili con una correlazione inferiore allo 0.60 per verificare se una riduzione delle dimensioni dei dati può migliorare le performance delle reti neurali eliminando il rumore e variabili ridondanti.

Da tutti i dataset ne sono derivati altri dove sono stati eliminati gli outliers secondo le due diverse definizioni usate, separatamente e congiuntamente per osservare come questo impatti i risultati.

Capitolo 4

Rete Neurale

In questo capitolo viene descritta l'implementazione di una rete neurale feedforward (FeedForward Neural Network, FNN) per risolvere il problema di regressione multipla su un dataset tabellare. La rete è stata sviluppata utilizzando PyTorch, una delle principali librerie di deep learning, che offre grande flessibilità nell'implementazione di architetture neurali personalizzate. Il processo di allenamento e valutazione è stato condotto tramite tecniche di cross-validation per migliorare la robustezza del modello. Inoltre, sono state applicate alcune tecniche di preprocessing dei dati e inizializzazione dei pesi, con l'obiettivo di massimizzare le performance della rete.

4.1 Caricamento dei dati

Il dataset utilizzato per l'addestramento del modello viene caricato tramite la funzione `load_data`, che utilizza Pandas per leggere il file CSV contenente le informazioni. Una scelta progettuale importante è l'uso della funzione `sample` con il parametro `frac = 1` per eseguire uno mescolamento dell'ordine dei record. Questo passaggio è fondamentale per evitare che la sequenza dei dati nel file influenzi negativamente l'allenamento del modello. Senza questo mescolamento, i dati potrebbero presentare pattern che la rete imparerebbe a memorizzare senza generalizzare correttamente, andando ad ottenere potenziali bias nell'addestramento.

Le variabili target per la predizione sono le seguenti:

- VolumeNP
- Glucosio50
- Aminoacidi06
- Lipidi20

Il resto delle variabili, che contengono diversi tipi di informazione a seconda del dataset scelto, vengono utilizzate come input del modello.

4.1.1 Divisione training e test

Nel processo di addestramento e valutazione della rete neurale, sono stati utilizzati due file distinti per i dati di training e di test, piuttosto che eseguire una semplice divisione in training e test set dallo stesso dataset completo. Questa scelta è motivata dalla necessità di gestire dati che derivano da un processo di filtraggio del dataset completo per ottenere un dataset di training specifico.

Il dataset completo, caricato come *test_csv_file*, contiene tutte le informazioni disponibili, con i dati completi e non filtrati. Per il training, tuttavia, si desidera utilizzare solo un sottoinsieme di questi dati, definito in un file di training specifico, ognuno dei quali rappresenta un dataset filtrato secondo diversi criteri e selezionato per il processo di addestramento. La separazione tra dataset di test e training in due file distinti è vantaggiosa in quanto:

- **Mantiene Intatta la Completezza del Dataset di Test:** Il file di test non viene alterato o filtrato ulteriormente per rimuovere dati, permettendo così al modello di essere valutato su un dataset originale e completo, che rappresenta al meglio la variabilità totale dei dati.
- **Facilita l'Utilizzo di Datasets di Training Filtrati:** Ogni dataset di training può essere personalizzato per includere solo osservazioni specifiche, escludendo record ritenuti non rilevanti o problematici per l'addestramento, mentre il dataset di test rimane invariato.

Poiché il dataset di training è un sottoinsieme filtrato del dataset di test, esiste il rischio di sovrapposizione di record tra i due file. Per evitare che il modello venga valutato su dati già presenti nel set di training, è stato implementato un controllo specifico, gestito dalla funzione *filter_test_set*.

- **Funzione *filter_test_set*:** Questa funzione verifica ed elimina dal dataset di test eventuali record che coincidono con quelli del dataset di training. Per realizzare questo controllo:
 1. I record di training e test vengono trasformati in DataFrame di Pandas, e per ogni riga del dataset, le colonne target sono trattate come tuple (ossia come record unici).

2. I record del dataset di test che sono presenti anche nel dataset di training vengono identificati e rimossi dal dataset di test, creando così un `X_test_filtered` che è completamente indipendente dai dati utilizzati in fase di addestramento.

Questo accorgimento garantisce che il modello non si avvantaggi di informazioni ripetute nel training e nel test, migliorando l'affidabilità e la validità della valutazione.

Per valutare dataset non filtrati, viene prima effettuata una divisione dei dati tramite la funzione `train_test_split` di Scikit-Learn, in modo che venga selezionato un 20% dei dati da unire al test set, che altrimenti, dopo un filtraggio tra due dataset identici, non lascerebbe alcun record per eseguire il test. Questo passaggio è importato anche per i dataset derivati dall'eliminazione di outliers in quanto permette di inserire alcuni dei record, considerati senza outliers, nel set di test, in modo da avere risultati più attendibili. Se infatti non ci fosse questo step, allora modelli allenati esclusivamente con valori definiti nella norma sarebbero poi testati esclusivamente su record con valori definiti anomali.

4.1.2 Preprocessing sui dati

Il preprocessing è un passo cruciale per garantire che i dati forniti alla rete siano in un formato ottimale. Una delle scelte progettuali più significative nel preprocessing è la normalizzazione dei dati. La normalizzazione è essenziale in molte reti neurali poiché aiuta a ridurre la varianza dei gradienti durante l'allenamento e facilita la convergenza. In particolare, è stata scelta la *MinMaxScaler* di Scikit-learn, che scala ogni feature in un intervallo tra 0 e 1. Questo approccio si rivela utile nei casi in cui i dati hanno distribuzioni non gaussiane e si vogliono evitare valori estremi che potrebbero rallentare l'allenamento della rete, come nel nostro caso. Un'alternativa considerata è stata quella l'uso di *StandardScaler*, che invece normalizza i dati secondo la distribuzione gaussiana standard, con media zero e varianza unitaria.

4.2 Architettura della Rete Neurale

Il modello di rete neurale è stato costruito utilizzando la classe *FeedForwardNN*, una classe derivata da `torch.nn.Module`, che rappresenta una rete neurale con tre strati completamente connessi, con dimensioni fisse per i layer nascosti. La struttura della rete è la seguente:

- Strato di input: in input è presente un numero mutevole di feature che dipende dal numero delle variabili indipendenti presenti nel dataset.
- Primo strato nascosto: abbassa il numero di informazioni e utilizza 128 neuroni.

- Secondo strato nascosto: il numero di neuroni passa a 64. Il numero di neuroni è stato ridotto progressivamente per imparare rappresentazioni più compatte man mano che ci si avvicina al layer di output.
- Strato di output: il numero di neuroni nel layer di output corrisponde al numero di variabili target da predire. Nel caso specifico, il dataset ha sempre quattro variabili target, quindi il layer di output avrà 4 neuroni.

Per i layer nascosti, è stata utilizzata la funzione di attivazione *ReLU* (Rectified Linear Unit). Questa funzione introduce non linearità nel modello, permettendo alla rete di apprendere rappresentazioni complesse. ReLU è ampiamente utilizzata per la sua semplicità computazionale e per la capacità di ridurre il rischio di scomparsa del gradiente, un problema comune in altre funzioni di attivazione come la sigmoide o la tanh.

```
class FeedForwardNN(nn.Module):
    def __init__(self, input_size, output_size):
        super(FeedForwardNN, self).__init__()
        self.fc1 = nn.Linear(input_size, 128)
        self.fc2 = nn.Linear(128, 64)
        self.fc3 = nn.Linear(64, output_size)
        self.relu = nn.ReLU()

    def forward(self, x):
        x = self.relu(self.fc1(x))
        x = self.relu(self.fc2(x))
        x = self.fc3(x)
        return x
```

4.2.1 Inizializzazione dei pesi

L'inizializzazione corretta dei pesi nella rete neurale è essenziale per un addestramento efficace. In questa implementazione, abbiamo utilizzato l'inizializzazione Xavier Uniform. Questa tecnica, nota anche come inizializzazione Glorot, è progettata per mantenere i gradienti bilanciati in reti profonde. Questa inizializzazione distribuisce i pesi uniformemente in un intervallo calcolato in base al numero di neuroni del layer, riducendo il rischio di gradienti molto grandi o molto piccoli.

```
def initialize_weights(m):
    if isinstance(m, nn.Linear):
        torch.nn.init.xavier_uniform_(m.weight)
```

4.3 Allenamento del Modello

I modelli sono stati allenati utilizzando l'algoritmo di ottimizzazione Adam, con un tasso di apprendimento iniziale pari a 0.01, che combina i vantaggi dell'adaptive learning rate e del momentum, rendendolo particolarmente adatto a scenari in cui si desidera una convergenza rapida e stabile.

```
optimizer = optim.Adam(model.parameters(), lr=learning_rate)
train_model(model, train_loader, criterion, optimizer, X_test, y_test, epochs)
```

L'allenamento è stato suddiviso in 40 epoche, durante le quali il modello ha iterato sui dati di addestramento in batch da 32 campioni ciascuno. Per ogni epoca, sono stati calcolati la funzione di perdita (loss) e il gradiente, aggiornando i pesi del modello in modo da ridurre l'errore. Questo approccio aiuta a ridurre il consumo di memoria e rende l'allenamento più efficiente.

L'addestramento della rete viene eseguito tramite la funzione *train_model*, utilizzando la Mean Absolute Error (MAE) come funzione di perdita, che si è dimostrata efficace per problemi di regressione robusta. Inoltre, per monitorare e ottimizzare il processo di apprendimento, è stata implementata la tecnica di early stopping.

L'early stopping è una tecnica di regolarizzazione che interrompe l'addestramento quando il modello non mostra miglioramenti significativi sulla perdita di validazione per un certo numero di epoche consecutive (la patience). Questo aiuta a prevenire l'overfitting, fermando l'addestramento al momento ottimale e salvando il modello con le migliori prestazioni.

Per rendere la valutazione del modello più robusta, è stata applicata una k-fold cross-validation. Nel caso specifico, la cross-validation utilizza `n_splits=4`, dividendo il dataset in quattro sottoinsiemi e addestrando il modello quattro volte su quattro sottoinsiemi diversi, valutando la capacità generale del modello di adattarsi a nuovi dati.

```
def train_model(model, train_loader, criterion, optimizer, X_val, y_val, epochs):
    early_stopping = EarlyStopping(patience=10, save_path="best_model.pth")

    loss_history= [0]*epochs

    for epoch in range(epochs):
        model.train()
        for inputs, targets in train_loader:
            inputs, targets = inputs.float(), targets.float()
```

```

    # Forward pass
    outputs = model(inputs)
    loss = criterion(outputs, targets)

    # Backward and optimize
    optimizer.zero_grad()
    loss.backward()
    optimizer.step()

    # Validation at the end of each epoch
    model.eval()
    with torch.no_grad():
        val_predictions = model(torch.FloatTensor(X_val))
        val_loss = criterion(val_predictions, torch.FloatTensor(y_val))

    loss_history[epoch]= round(val_loss.item(), 4)
    Validation Loss: {val_loss.item():.4f}")

    # Early stopping check
    early_stopping(val_loss.item(), model)

    # Stop if early stopping criterion is met
    if early_stopping.early_stop:
        print("Early stopping triggered.")
        for i in range(epoch+1, epochs):
            loss_history[i] = loss_history[epoch]
        return loss_history

    # Load the best model before returning
    model.load_state_dict(torch.load("best_model.pth", weights_only=False))
    return loss_history

```

4.3.1 Early Stopping

La fase di allenamento della rete neurale viene monitorata con early stopping, una tecnica fondamentale per evitare il problema dell'overfitting. L'idea di base dell'early stopping è fermare l'allenamento non appena si rileva che il modello non migliora più significativamente sulle performance del set di validazione. L'early stopping è implementato nella classe EarlyStopping.

Durante l'allenamento, dopo ogni epoca, il modello viene valutato sul set di validazione. Se il valore della funzione di perdita sul set di validazione non migliora oltre una certa soglia (`min_delta`) per un certo numero di epoche consecutive (`patience`), l'allenamento si interrompe, e il modello con la migliore prestazione viene salvato e ripristinato.

- **Patience:** rappresenta il numero massimo di epoche consecutive in cui il modello è autorizzato a non migliorare sulla validazione. Un valore tipico potrebbe essere 5 o 10, ma in questo caso specifico la `patience` è scelta in base alla complessità dei dati e alla durata dell'allenamento.
- **Min Delta:** definisce la variazione minima di miglioramento richiesta per considerare una nuova epoca come migliorativa. Questo evita che piccole fluttuazioni attivino l'early stopping.

Nella funzione `train_model`, ad ogni epoca si controlla se la perdita di validazione (`val_loss`) migliora. Quando il miglioramento cessa per un numero di epoche consecutivo, l'allenamento si ferma automaticamente, salvando così le risorse computazionali e garantendo una buona generalizzazione del modello. Il modello salvato al termine dell'early stopping viene utilizzato come modello finale.

Vantaggi dell'Early Stopping:

- **Evitare Overfitting:** fermando l'allenamento prima che il modello inizi a specializzarsi troppo sui dati di addestramento, si ottiene una migliore capacità di generalizzazione.
- **Efficienza Computazionale:** riducendo il numero di epoche, l'early stopping contribuisce a rendere l'allenamento meno dispendioso in termini di risorse e tempo.

```
class EarlyStopping:
    def __init__(self, patience, min_delta=0, save_path="best_model.pth"):
        self.patience = patience
        self.min_delta = min_delta
        self.save_path = save_path
        self.best_score = None
        self.epochs_no_improve = 0
        self.early_stop = False

    def __call__(self, val_loss, model):
        if self.best_score is None:
            self.best_score = val_loss
            self.save_checkpoint(model)
```

```

elif val_loss < self.best_score - self.min_delta:
    self.best_score = val_loss
    self.epochs_no_improve = 0
    self.save_checkpoint(model)
else:
    self.epochs_no_improve += 1
    if self.epochs_no_improve >= self.patience:
        self.early_stop = True

def save_checkpoint(self, model):
    torch.save(model.state_dict(), self.save_path)
    print("Model saved with improved validation loss.")

```

4.4 Cross-validation

La cross-validation K-fold è una tecnica utilizzata per validare le prestazioni del modello in modo robusto. Essa consiste nel dividere il dataset di addestramento in `n_splits` o fold (nel caso testato, 4), e addestrare il modello separatamente su ciascuno di questi fold, utilizzando a rotazione un fold come set di validazione e gli altri come set di addestramento. Questo processo viene ripetuto per ogni fold, e infine si calcola la media delle metriche di valutazione ottenute su ciascuno dei fold.

La cross-validation viene gestita con la funzione `KFold` di `sklearn`, che assicura la suddivisione del dataset in 4 parti uguali:

1. Divisione dei Fold: Per ogni iterazione, `KFold` assegna un fold come set di validazione e i restanti come set di addestramento.
2. Preprocessamento dei Dati: All'inizio di ogni iterazione, i dati di addestramento e validazione vengono normalizzati tramite la tecnica Min-Max. Questa normalizzazione, ripetuta per ogni fold, permette di rendere le distribuzioni di ogni sottoinsieme compatibili tra loro.
3. Training e Validazione con Early Stopping: Per ciascun fold, il modello viene addestrato e monitorato utilizzando l'early stopping, assicurandosi di fermare l'allenamento al momento ottimale. Al termine dell'early stopping, il modello migliore viene valutato sul set di validazione.
4. Salvataggio delle Metriche: Dopo ogni fold, le metriche di valutazione (MSE, MAE e R2) vengono salvate, per poi calcolarne la media finale.

Al termine della cross-validation, si calcola la media delle metriche di valutazione su tutti i fold per ottenere una stima più accurata della capacità di generalizzazione del modello. Le metriche medie forniscono un quadro generale delle prestazioni del modello, mentre la loro varianza indica la stabilità delle predizioni. Le metriche riportate al termine della cross-validation sono quindi molto rappresentative, dato che il modello è stato valutato su tutte le diverse porzioni del dataset.

Vantaggi della Cross-Validation:

- **Affidabilità delle Metriche:** utilizzando tutti i dati a disposizione sia per il training che per la validazione, la cross-validation fornisce una stima più stabile e robusta delle metriche di performance rispetto a una semplice divisione in training e validation.
- **Riduzione della Variabilità:** minimizzando l'effetto della suddivisione casuale del dataset, la cross-validation offre una misura della prestazione del modello più indicativa della sua capacità di generalizzazione.

4.5 Valutazione delle prestazioni

Tramite la funzione *evaluate_model* sono state calcolate tre metriche di performance:

- **Errore Quadratico Medio (MSE):** Questa metrica misura la media dei quadrati degli errori tra le predizioni del modello e i valori target. Penalizza maggiormente gli errori grandi e tende a dare maggiore enfasi alle deviazioni maggiori.
- **Errore Assoluto Medio (MAE):** Misura la media degli errori assoluti, rendendolo meno sensibile agli outlier e utile in casi in cui è preferibile non penalizzare in modo eccessivo grandi errori isolati.
- **Coefficiente di Determinazione (R^2):** Misura quanto bene la varianza dei dati target è spiegata dal modello. Un valore di R^2 vicino a 1 indica un ottimo adattamento ai dati, mentre un valore vicino a 0 suggerisce che il modello non è in grado di catturare la variabilità nei dati.

Queste metriche forniscono una valutazione esaustiva delle prestazioni del modello sia in termini di accuratezza assoluta che relativa. Avere più indicatori che evidenziano diversi aspetti dei risultati ci aiuterà a confrontare in modo più preciso le performance dei vari modelli.

Nel nostro caso MAE e MSE sono entrambe da considerare in quanto le quattro variabili target hanno scale diverse, in particolare il volume, rispetto a glucosio, amminoacidi e lipidi ha un range molto più ampio: tra i valori massimi e minimi del volume c'è circa 1500 mL di differenza mentre tra i tutti tre macro nutrienti i valori non si supera un oscillamento massimo di 445 ml.

Nonostante tutte le variabili vengano scalate con *MinMaxScaler* facendo sì che entrambe le metriche forniscano una rappresentazione più bilanciata degli errori, è bene osservare entrambe le metriche, una per l'accuratezza complessiva in termini di errore assoluto medio, e l'altra per tenere traccia degli errori grandi e capire quanto fortemente il modello stia sbagliando le predizioni della variabile con il range più ampio, visto che errori più grandi in questo contesto sono particolarmente gravi.

```
def evaluate_model(model, X_test, y_test):
    model.eval()
    with torch.no_grad():
        predictions = model(torch.FloatTensor(X_test))
        y_test_tensor = torch.FloatTensor(y_test)

        # Mean Squared Error (MSE)
        mse_loss_fn = nn.MSELoss()
        mse = mse_loss_fn(predictions, y_test_tensor)
        print(f'Mean Squared Error: {mse.item():.4f}')

        # Mean Absolute Error (MAE)
        mae_loss_fn = nn.L1Loss()
        mae = mae_loss_fn(predictions, y_test_tensor)
        print(f'Mean Absolute Error: {mae.item():.4f}')

        # Coefficient of Determination (R^2)
        y_mean = torch.mean(y_test_tensor)
        ss_total = torch.sum((y_test_tensor - y_mean) ** 2)
        ss_residual = torch.sum((y_test_tensor - predictions) ** 2)
        r2 = 1 - ss_residual / ss_total
        print(f'R squared Score: {r2.item():.4f}')

        # Return all metrics for final considerations
    return { "MSE": mse.item(), "MAE": mae.item(), "R2": r2.item() }
```

Il processo di valutazione avviene sia alla fine di ogni fold nella cross-validation, sia sul set di test dopo il completamento della cross-validation, nello specifico:

- Valutazione per Fold: Ogni volta che il modello viene allenato su un fold e validato sul corrispondente set di validazione, la funzione *evaluate_model* calcola MSE, MAE e R^2 e salva questi risultati in un dizionario. Al termine della cross-validation, viene calcolata la media delle metriche per ciascun fold, offrendo così una misura robusta della performance del modello.
- Valutazione sul Set di Test: Una volta completata la cross-validation e selezionato il modello finale, quest'ultimo viene testato su un set di test indipendente. Questo consente di valutare le metriche di errore e accuratezza su dati mai visti, confermando la capacità del modello di generalizzare su nuovi dati.

Al termine dell'allenamento, vengono riportate anche le curve di andamento della loss media. Queste curve mostrano come la loss diminuisca durante le epoche di addestramento, dando un'indicazione visiva della convergenza del modello. I valori di loss sono salvati in `loss_history`, e la loro media è visualizzata graficamente per ciascun fold.

Capitolo 5

Esperimenti

L'obiettivo principale del modello di rete neurale è predire con precisione i valori ottimali di volume e macro nutrienti (glucosio, lipidi, proteine) che dovrebbero essere inclusi in una sacca parenterale personalizzata per ciascun neonato prematuro.

Sono stati quindi allenati e valutati diversi modelli basati sulla stessa architettura ma che hanno fatto uso i diversi dataset descritti precedentemente in fase sia di allenamento che di testing.

L'approccio seguito è stato quello di analizzare da prima il comportamento senza alcuna gestione dei potenziali valori anomali (outliers), e dopo ciò sono stati rimossi nella fase di allenamento ma mantenuti nel test.

Di seguito vengono riportate le tre metriche dei vari modelli riguardo le performance ottenute nella sola fase di testing, secondo la procedura di cross validation.

5.1 Senza gestione degli outliers

Dataset	MSE	MAE	R^2
Completo	261.0474	7.0246	0.9770
Ricoveri lunghi	477.7965	8.7359	0.9601
$Eg < 28.5$	510.3653	9.2166	0.9541
$28.5 < Eg < 32.5$	492.9862	8.8227	0.9574
$Eg > 32.5$	361.7493	9.1673	0.9683
Correlazioni < 60	542.8907	9.1848	0.9542

Tabella 5.1: Performance con outliers

5.2 Risultati dopo la gestione degli outliers

Gli outliers non vengono considerati nella fase di allenamento ma solo nella fase di test, l'idea è quella di migliorare le performance nei casi più comuni a scapito dei pochi casi con valori considerati anomali.

Poiché sono state usate due definizioni diverse di outlier, vengono elencati i risultati nell'usare solo una piuttosto che l'altra, o entrambe contemporaneamente.

5.2.1 Secondo Z-score

Dataset	MSE	MAE	R^2
Completo	462.2580	7.6517	0.9592
Ricoveri lunghi	1149.5196	12.1037	0.8985
$Eg < 28.5$	562.7944	9.5422	0.9513
$28.5 < Eg < 32.5$	585.5432	9.3506	0.9492
$Eg > 32.5$	1446.2267	12.5515	0.8744
Correlazioni < 60	542.8907	9.1848	0.9542

Tabella 5.2: Performance senza outlier secondo Z-score

5.2.2 Secondo percentile

Dataset	MSE	MAE	R^2
Completo	451.4172	7.6524	0.9622
Ricoveri lunghi	1136.3823	12.6151	0.9012
$Eg < 28.5$	599.1842	9.5057	0.9481
$28.5 < Eg < 32.5$	591.1312	9.3092	0.9482
$Eg > 32.5$	613.8177	10.0767	0.9471
Correlazioni < 60	542.8907	9.1848	0.9542

Tabella 5.3: Performance senza outlier secondo i percentili

5.2.3 Secondo entrambe

Dataset	MSE	MAE	R^2
Completo	510.3829	7.7331	0.9551
Ricoveri lunghi	1340.1856	13.7113	0.8839
Eg<28.5	681.2881	9.9481	0.9402
28.5<Eg<32.5	584.2070	9.7505	0.9484
Eg>32.5	1191.0723	11.3601	0.8958
Correlazioni<60	542.8907	9.1848	0.9542

Tabella 5.4: Performance senza outlier secondo Z-score o Percentili

Vengono ora analizzati più nel dettaglio i diversi modelli durante la fase di training per evidenziare l'andamento nell'apprendimento e i risultati ottenuti.

5.3 Esperimento 1

In questo esperimento viene utilizzato il dataset completo, con tutti i dati disponibili, senza che siano applicati criteri di selezione.

5.3.1 Risultati senza gestione degli outliers

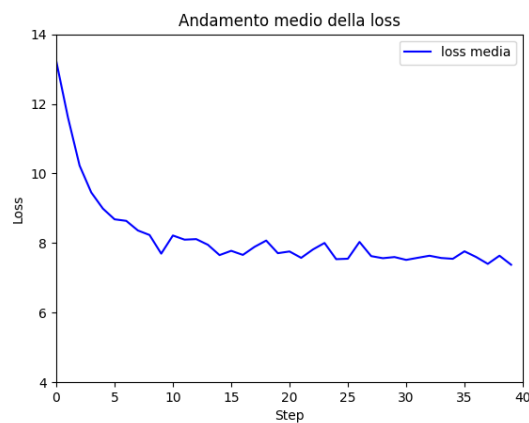


Figura 5.1: Andamento Loss Completo

Metriche medie calcolate sul validation set:

MSE: 292.1008

MAE: 7.3356

R^2 Score: 0.9746

5.3.2 Outliers definiti tramite Z-score

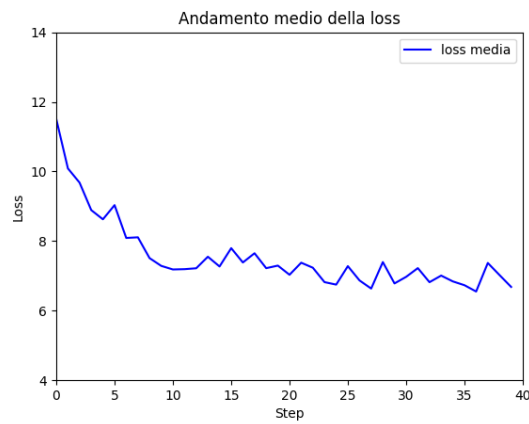


Figura 5.2: Andamento Loss Completo Outliers Z-score

Metriche medie calcolate sul validation set:

MSE: 214.4483

MAE: 6.6110

R^2 Score: 0.9720

5.3.3 Outliers definiti tramite Percentili

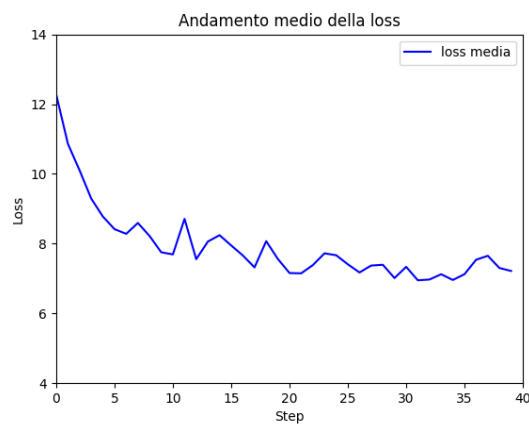


Figura 5.3: Andamento Loss Completo Outliers Percentili

Metriche medie calcolate sul validation set:

MSE: 253.5030

MAE: 7.2167

R^2 Score: 0.9713

5.3.4 Outliers definiti tramite entrambe le precedenti

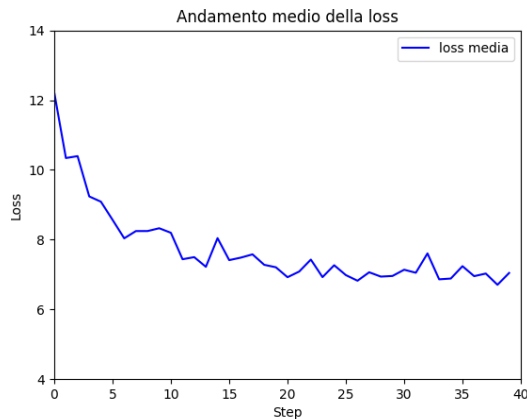


Figura 5.4: Andamento Loss Completo Outliers entrambe

Metriche medie calcolate sul validation set:

MSE: 210.3535

MAE: 6.6335

R^2 Score: 0.9737

5.3.5 Conclusioni

Togliendo i record con valori anomali dalla fase di allenamento si nota una performance peggiore quando questi vengono incontrati nella fase di testing. Inoltre anche su record che hanno solo valori che rientrano nella norma, rispetto al modello di partenza, c'è un miglioramento sull'errore medio, ma un R^2 score minore.

5.4 Esperimento 2

Nel secondo esperimento, come training set, viene utilizzato quello in cui sono stati eliminati i record relativi ai primi sei giorni di ricovero, lasciando solo quelli successivi.

5.4.1 Senza gestione degli outliers

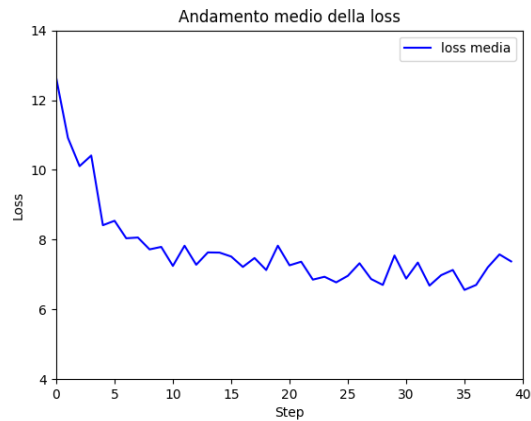


Figura 5.5: Andamento Loss Ricoveri Lunghi

Metriche medie calcolate sul validation set:

MSE: 262.5545

MAE: 7.1555

R^2 Score: 0.9784

5.4.2 Outliers definiti tramite Z-score

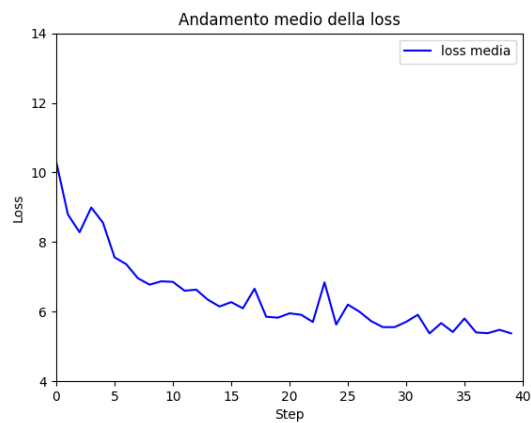


Figura 5.6: Andamento Loss Ricoveri Lunghi Outliers Z-score

Metriche medie calcolate sul validation set:

MSE: 117.7638

MAE: 5.2000

R^2 Score: 0.9841

5.4.3 Outliers definiti tramite Percentili

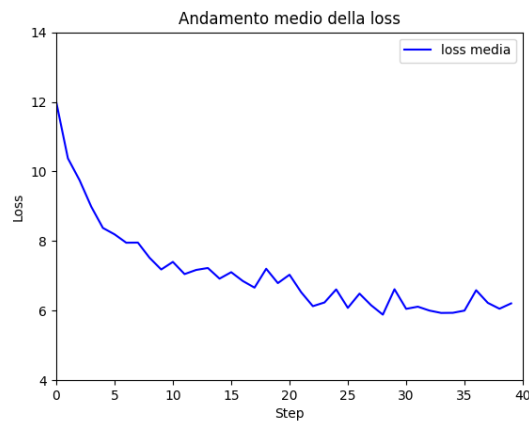


Figura 5.7: Andamento Loss Ricoveri Lunghi Outliers Percentili

Metriche medie calcolate sul validation set:

MSE: 144.1191

MAE: 5.9165

R^2 Score: 0.9833

5.4.4 Outliers definiti tramite entrambe le precedenti

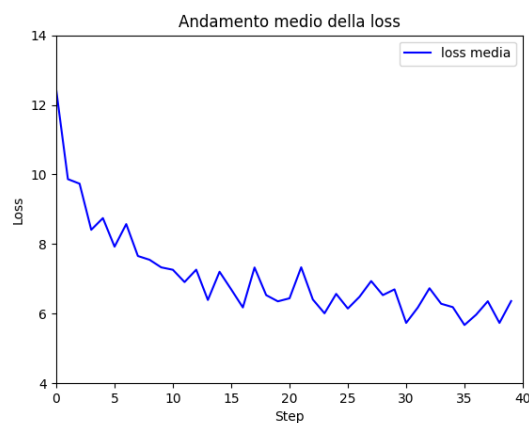


Figura 5.8: Andamento Loss Ricoveri Lunghi Outliers entrambe

Metriche medie calcolate sul validation set:

MSE: 126.1540

MAE: 5.5843

R^2 Score: 0.9841

Conclusioni

L'esperimento ha portato alla creazione di un modello specializzato nelle azioni da compiere dal settimo giorno, raggiungendo in questi casi un'elevata accuratezza. Tuttavia, il modello presenta un margine di errore molto più alto nell'elaborare record relativi ai primi sei giorni di ricovero, diversi da quelli utilizzati in fase di addestramento.

5.5 Esperimento 3

In questo esperimento il dataset originale è stato tripartito in base all'età gestazionale, l'idea è di ottenere un modello specializzato per ogni range di età gestazionali, da usare uno in alternativa all'altro in base all'età gestazionale del bambino in cura.

5.5.1 EG bassa

Senza gestione degli outliers

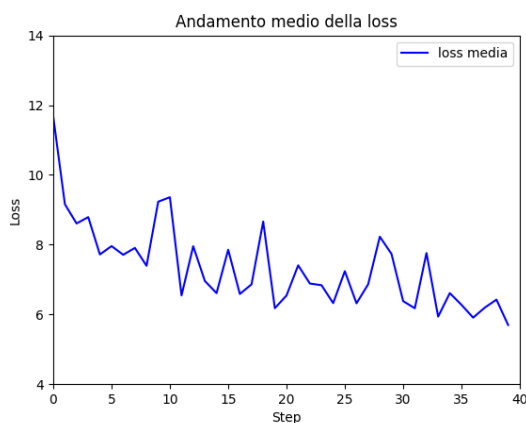


Figura 5.9: Andamento EG bassa

Metriche medie calcolate sul validation set:

MSE: 188.8342

MAE: 5.4731

R^2 Score: 0.9644

Outliers definiti tramite Z-score

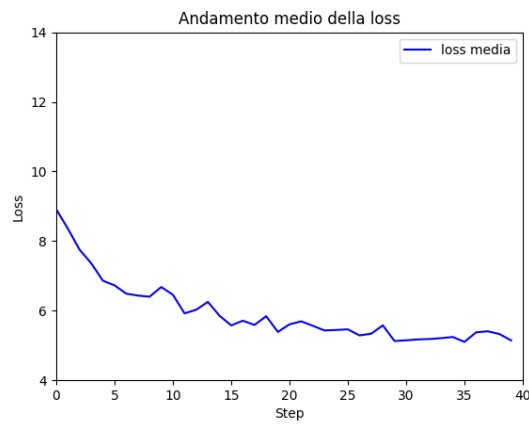


Figura 5.10: Andamento EG bassa Outliers Z-score

Metriche medie calcolate sul validation set:

MSE: 116.6098

MAE: 5.0292

R^2 Score: 0.9632

Outliers definiti tramite Percentili

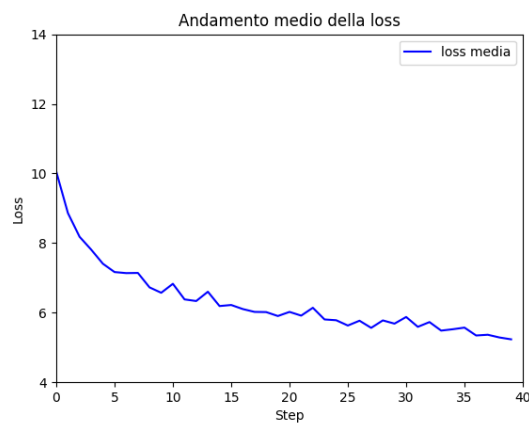


Figura 5.11: Andamento EG bassa Outliers Percentili

Metriche medie calcolate sul validation set:

MSE: 135.2264

MAE: 5.1085

R^2 Score: 0.9653

Outliers definiti tramite entrambe le precedenti

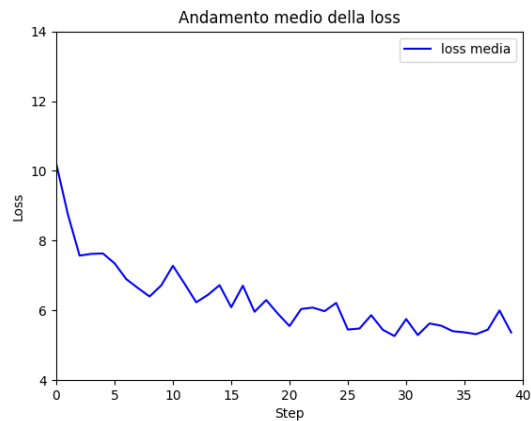


Figura 5.12: Andamento EG bassa Outliers entrambe

Metriche medie calcolate sul validation set:

MSE: 127.5757

MAE: 5.1247

R^2 Score: 0.9614

5.5.2 EG media

Senza gestione degli outliers

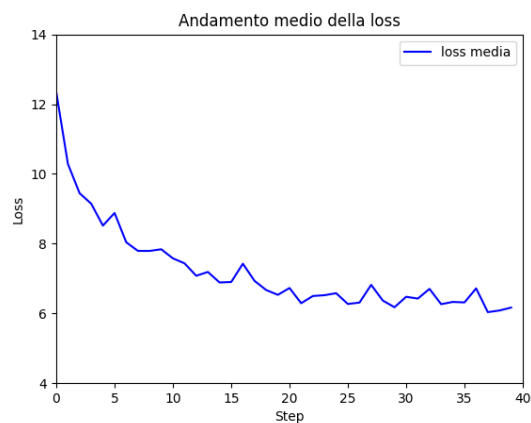


Figura 5.13: Andamento EG media

Metriche medie calcolate sul validation set:

MSE: 148.0087

MAE: 5.9072

R^2 Score: 0.9750

Outliers definiti tramite Z-score

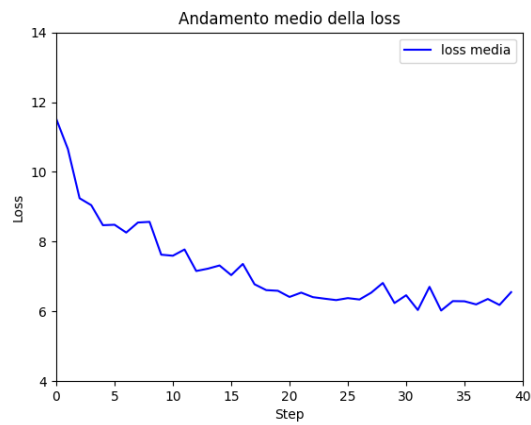


Figura 5.14: Andamento EG media Outliers Z-score

Metriche medie calcolate sul validation set:

MSE: 134.1792

MAE: 5.8527

R^2 Score: 0.9749

Outliers definiti tramite Percentili

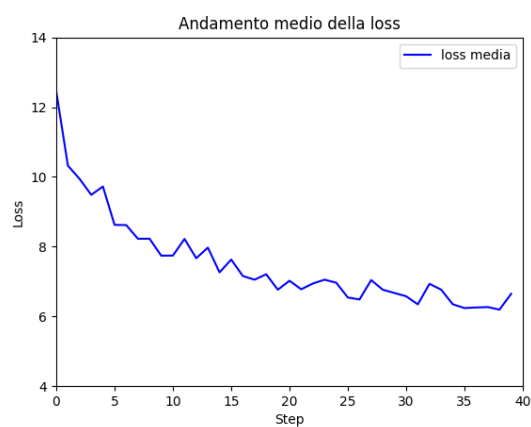


Figura 5.15: Andamento EG media Outliers Percentili

Metriche medie calcolate sul validation set:

MSE: 156.0082

MAE: 6.0036

R^2 Score: 0.9732

Outliers definiti tramite entrambe le precedenti

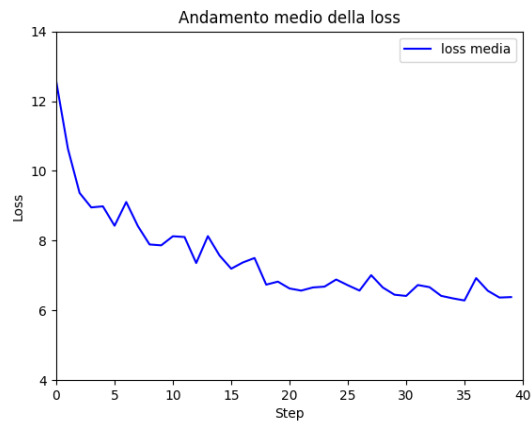


Figura 5.16: Andamento EG media Outliers entrambe

Metriche medie calcolate sul validation set:

MSE: 150.1290

MAE: 6.1770

R^2 Score: 0.9725

5.5.3 EG alta

Senza gestione degli outliers

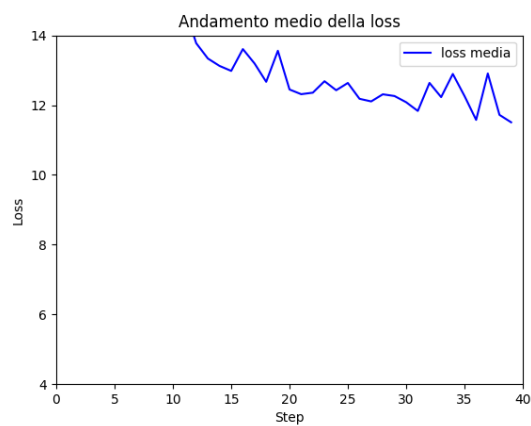


Figura 5.17: Andamento EG alta Outliers Z-score

Metriche medie calcolate sul validation set:

MSE: 608.3890

MAE: 11.1903

R^2 Score: 0.9714

Outliers definiti tramite Z-score

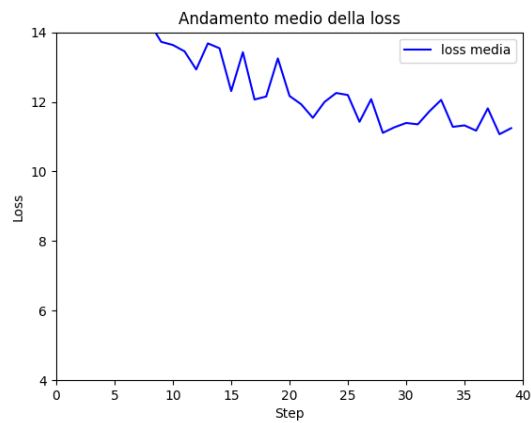


Figura 5.18: Andamento EG alta

Metriche medie calcolate sul validation set:

MSE: 477.5485

MAE: 11.0012

R^2 Score: 0.9664

Outliers definiti tramite Percentili

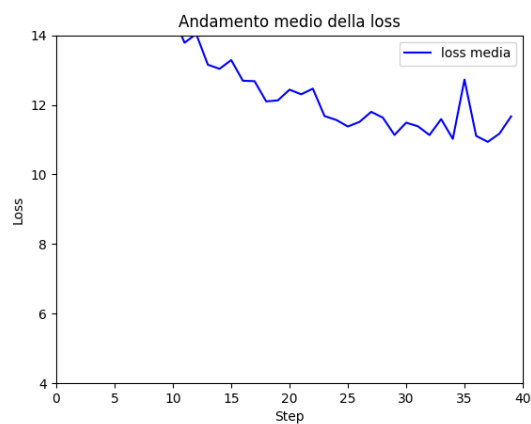


Figura 5.19: Andamento EG alta Outliers Percentili

Metriche medie calcolate sul validation set:

MSE: 482.1898

MAE: 10.6667

R^2 Score: 0.9681

Outliers definiti tramite entrambe le precedenti

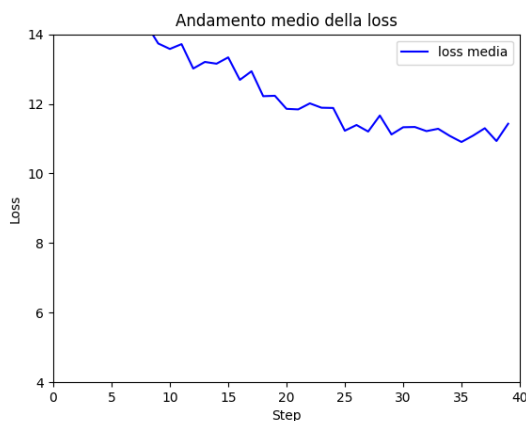


Figura 5.20: Andamento EG alta Outliers entrambe

Metriche medie calcolate sul validation set:

MSE: 452.0108

MAE: 10.5202

R^2 Score: 0.9674

5.5.4 Conclusioni

Come previsto i tre modelli hanno performance peggiori rispetto a quello allenato con il dataset completo ma, come mostrano i risultati nella fase di allenamento c'è il miglioramento cercato sui sotto-dataset corrispondenti.

Tra i tre gruppi analizzati, i due con le età gestazionali più basse hanno registrato errori significativamente inferiori rispetto al gruppo con le età gestazionali più avanzate. Questo potrebbe indicare una migliore performance, ma potrebbe anche dipendere dalla natura dei dati. Infatti, il valore del coefficiente di determinazione (R^2) è molto simile tra i tre gruppi. Nelle età gestazionali più avanzate, tutti i valori osservati tendono a essere più elevati un'età gestazionale maggiore comporta una crescita del peso e di conseguenza un aumento di tutti i valori correlati, quindi, se l'errore rimane proporzionale al valore del dato, un errore più alto non implica necessariamente una performance inferiore.

5.6 Esperimento 4

Nell'ultimo esperimento è stato utilizzato il dataset in cui sono presenti i dati relativi alle variabili che hanno una correlazione inferiore a 0.6 tra di loro.

5.6.1 Senza gestione degli outliers

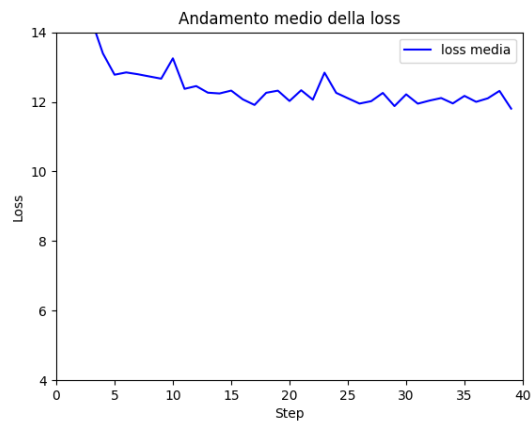


Figura 5.21: Andamento Loss Correlazioni basse

Metriche medie calcolate sul validation set:

MSE: 733.7874

MAE: 11.8051

R^2 Score: 0.9361

5.6.2 Outliers definiti tramite Z-score

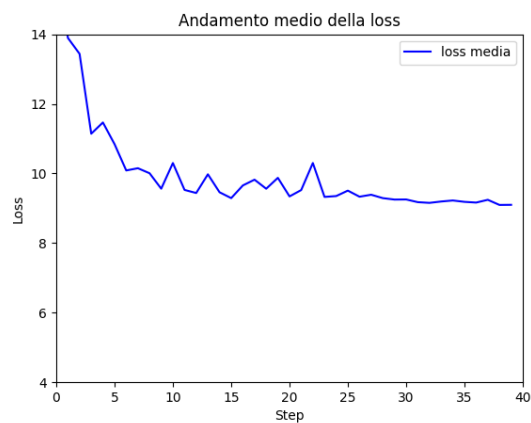


Figura 5.22: Andamento Correlazioni basse Outliers Z-score

Metriche medie calcolate sul validation set:

MSE: 355.1240

MAE: 9.0956

R^2 Score: 0.9459

5.6.3 Outliers definiti tramite Percentili

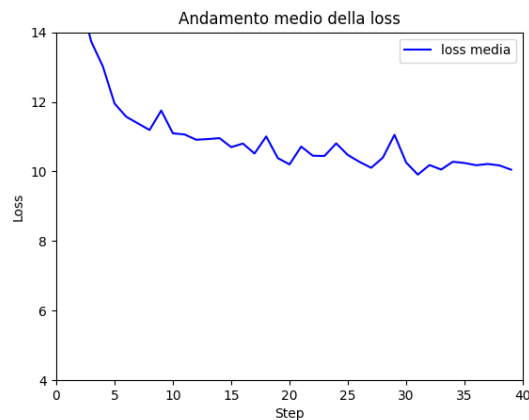


Figura 5.23: Andamento Correlazioni basse Outliers Percentili

Metriche medie calcolate sul validation set:

MSE: 408.6494

MAE: 9.9072

R^2 Score: 0.9478

5.6.4 Outliers definiti tramite entrambe le precedenti

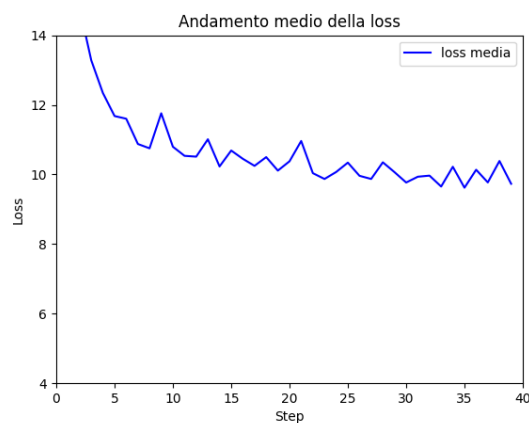


Figura 5.24: Andamento Loss Correlazioni basse Outliers entrambe

MSE: 371.5437

MAE: 9.4922

R^2 Score: 0.9474

Conclusioni

Nonostante siano stati selezionati i valori che risultano avere una correlazione più forte con i dati da prevedere, questo ha solo peggiorato le performance generali. Si conclude che avere più informazioni, seppure con un grado di correlazione alta, aiuta il modello nel fornire informazioni più accurate.

5.7 Confronto

Di seguito una tabella riassuntiva delle migliori performance sul rispettivo validation set dei modelli testati, ordinati secondo R^2 score.

Esperimento	Outliers	MSE	MAE	R^2
Ricoveri lunghi	Secondo Z-score	117.7638	5.2000	0.9841
Ricoveri lunghi	Secondo entrambe	126.1540	5.5843	0.9841
Ricoveri lunghi	Secondo percentili	144.119	5.9165	0.9833
Ricoveri lunghi	Non gestiti	262.5545	7.1555	0.9784
EG media	Non gestiti	148.0087	5.9072	0.9750
EG media	Secondo Z-score	134.1792	5.8527	0.9749
Completo	Non gestiti	292.1008	7.3356	0.9746
Completo	Secondo entrambe	210.3535	6.6335	0.9737

Tabella 5.5: Confronto Performance su Validation Set

Capitolo 6

Conclusioni

Dagli esperimenti eseguiti si discerne che il modello che tra tutti quelli testati ha performance migliori nel caso generale è quello allenato sul dataset completo, comprendente anche eventuali outliers. La rete allenata risulta infatti robusta alla presenza di valori anomali che se tolti dall'allenamento ne peggiorano le performance complessive.

L'allenamento sui dataset derivati testati non ha portato a risultati che indicano un miglioramento seguendo questa metodologia, sono però stati ottenuti modelli specializzati in campi specifici che superano in accuratezza il modello base, in particolare tutti quelli che si occupano dei giorni successivi al settimo. Al contrario possono essere interessanti ulteriori indagini sul perché alcuni modelli, come quelli dedicati ad età gestazionali alte e basse, visto che nonostante la ricercata specializzazione non abbiano ottenuto risultati migliori al modello base.

In ogni caso sono stati ottenuti dei risultati promettenti, con modelli che sono in grado di formulare quotidianamente sacche da suggerire a un medico, che vanno a costituire un buon sistema di supporto alle decisioni, per ridurre il dispendio di risorse specialmente in termini di tempo dei medici. Infatti nonostante i modelli sviluppati abbiano mostrato una buona capacità predittiva, è fondamentale che le sacche ottenute tramite questi algoritmi vengano comunque sottoposte a una valutazione finale da parte di medici esperti. Il coinvolgimento umano è cruciale per garantire la sicurezza e l'efficacia delle soluzioni proposte, tenendo conto di fattori clinici che potrebbero non essere completamente catturati dai modelli.

La principale limitazione di questo approccio risiede nel fatto che i modelli, basandosi su dati storici, tendono a replicare schemi e comportamenti del passato. Questo implica la possibilità di emulare decisioni o formulazioni che potrebbero non essere ottimali o addirittura errate. Per questo motivo, un importante obiettivo futuro sarà quello di migliorare ulteriormente questi modelli, puntando non solo a replicare il passato, ma a proporre formulazioni di sacche potenzialmente migliori di quelle storiche. L'integrazione

di nuove strategie di ottimizzazione e l'uso di dati aggiuntivi come il percentile o il suo cambiamento nel corso del ricovero, potranno consentire l'evoluzione verso soluzioni più efficaci.

In futuro, al fine di permettere a un medico di valutare una ricetta, potrà essere usata l'interfaccia che è stata creata nella precedente fase di tirocinio. Questa interfaccia, oltre alla possibilità di scelta della sacca migliore tra due che vengono proposte, dà la possibilità di apportare modifiche, in modo da avere correzioni e suggerimenti da parte dei medici. Questo approccio potrebbe inoltre essere usato anche per effettuare una valutazione su modelli che ottimizzano rispetto a criteri diversi da quello di replicazione del passato.

Bibliografia

- [1] Walter Mihatsch, Miguel Ángel Jiménez Varas, Lucia Lorenzino Diehl, Virgilio Carnielli, Rahel Schuler, Corinna Gebauer, and Miguel Sáenz de Pipaón Marcos. Systematic review on individualized versus standardized parenteral nutrition in preterm infants. *Nutrients*, 15(5):1224, 2023.
- [2] Valentina Rizzo, Manuela Capozza, Raffaella Panza, Nicola Laforgia, and Maria Elisabetta Baldassarre. Macronutrients and micronutrients in parenteral nutrition for preterm newborns: a narrative review. *Nutrients*, 14(7):1530, 2022.
- [3] LaVone E Simmons, Craig E Rubens, Gary L Darmstadt, and Michael G Gravett. Preventing preterm birth and neonatal mortality: exploring the epidemiology, causes, and interventions. In *Seminars in perinatology*, volume 34, pages 408–415. Elsevier, 2010.
- [4] José Villar, Francesca Giuliani, Zulfiqar A Bhutta, Enrico Bertino, Eric O Ohuma, Leila Cheikh Ismail, Fernando C Barros, Douglas G Altman, Cesar Victora, Julia A Noble, et al. Postnatal growth standards for preterm infants: the preterm postnatal follow-up study of the intergrowth-21st project. *The Lancet Global Health*, 3(11):e681–e691, 2015.

Ringraziamenti

Ringrazio tutti quelli che mi hanno accompagnato durante il mio percorso universitario. Ringrazio i miei genitori, Egidio e Ursula, che per me saranno sempre babbo e mamma, che mi hanno supportato sempre e mi hanno fatto arrivare fino a qui, non solo durante questi tre anni universitari ma per tutta la mia vita, nonostante ci siano stati momenti in cui sicuramente non li ho resi felici dei risultati delle mie scelte, mi hanno sempre permesso di imparare dai miei errori.

Ringrazio i miei fratelli, mio fratello Alex e la mia sorellina Ilaria, con cui ho condiviso moltissimi ricordi sin da quando siamo piccoli e sono sicuro che se non ci fossero stati, sarei una persona molto diversa da quella di oggi. Sono contento che le cose stiano come stanno, con loro che fanno, e faranno sempre parte della mia vita.

Ringrazio i miei nonni, in particolare Mariangela e Quinto, che ogni tanto ho svegliato perché urlavo giocando al computer di sera, e che mi hanno scusato il rimandare qualche faccenda in casa quando avevo bisogno di studiare un po' più del solito.

Ringrazio la mia ragazza, Morgana, che più di tutti mi è stata vicina in questi anni, aiutandomi nei momenti brutti ed essendo motivo di quelli più belli, dentro e fuori dall'università è sempre fantastica e spero di renderla felice ogni giorno almeno la metà di quanto lei faccia con me.

Ringrazio anche i suoi genitori, Graziano e Maica, che sono sempre gentili e spesso mi accolgono facendomi sentire sempre a casa.

Ringrazio tutti questi di cui non uso i nomi anagrafici ma quelli che hanno per me, Andreina, Andri, Cri, Debby, il Fali, Giulio, Leo, Matte Stefan, Umbe e lo Zucca, tutti compagni di vita, da più o meno tempo, che sono felice di poter chiamare amici.

Un ringraziamento a parte va a Riki, che in certi periodi ho visto più della mia famiglia e della mia ragazza perché passavamo tutto il giorno a Perugia, abbiamo preparato e passato praticamente tutti gli esami insieme, dandone separatamente solo un paio, e dal primo giorno di lezione fino a quello della laurea ci siamo accompagnati e supportati l'un l'altro.

Ringrazio tutti i professori che mi hanno insegnato con passione la loro materia, in particolare le mie due relatrici, Valentina Poggioni e Valentina Franzoni, che mi hanno seguito e supportato entrambe dall'inizio del tirocinio fino ad oggi, insegnandomi molte cose anche in questo percorso finale.

Ringrazio tutti i miei compagni di corso provenienti da diverse realtà, come Perugia, Gubbio, Foligno o Gualdo Cattaneo, con cui sono davvero felice di aver instaurato un bel rapporto, fatto di supporto reciproco e momenti di condivisione, sia durante le lezioni che al di fuori di esse.

Ringrazio infine tutti quelli che mi sono vicini, amici e parenti, che, sono certo, comprenderanno la mia mancata menzione dovuta esclusivamente alla mia scarsa memoria.