



Università del Piemonte Orientale

Dipartimento di Scienze e Innovazione Tecnologica

Corso di Laurea in Informatica

Relazione per la prova finale

Un Sistema di Supporto alle Decisioni
per la predizione del rischio in sala
parto basato su Reti Bayesiane

Tutore interno:

Prof. Manuel Striani

Candidato:

Alessandro Cadoni

Anno Accademico 2024/2025

Ringraziamenti

Desidero ringraziare di cuore i miei genitori e le mie nonne, che con il loro affetto e il loro sostegno costante mi hanno accompagnato in questo percorso. Un pensiero speciale va anche alla mia ragazza, per la sua vicinanza e il supporto prezioso che mi ha dato durante la stesura di questa tesi. Un sincero grazie va inoltre ai miei amici e ai colleghi studenti, che con la loro amicizia, collaborazione e incoraggiamento hanno reso questo percorso universitario più leggero, stimolante e ricco di momenti indimenticabili.

Indice

1	Introduzione	1
2	Scopo del lavoro	3
2.1	Sistemi di supporto alle decisioni in ambito sanitario	3
2.2	Modelli di ML supervisionati: Le Reti Bayesiane	4
3	Metodologie e strumenti utilizzati	5
3.1	Raccolta dei dati e fattori di rischio antepartum e intrapartum	5
3.2	Requisiti funzionali e non funzionali	6
3.3	Casi d'uso - Diagramma UML	6
4	Lavoro svolto	9
4.1	Modello dei dati: Struttura dei dati del paziente	9
4.2	Architettura dell'applicazione web	11
4.3	Scelta delle tecnologie e framework	11
5	Lavoro svolto	13
5.1	Sviluppo del Back-End	13
5.2	Sviluppo del frontend	14
6	Discussione sui risultati ottenuti	17
7	Conclusioni e sviluppi futuri	19
7.1	Limiti del lavoro svolto	19
7.2	Possibili miglioramenti e sviluppi futuri	19
	Configurazione del servizio web su Server Cloud su Windows	21
	Bibliografia	24

Abstract

Ad oggi, vi sono evidenze che diversi fattori possono aumentare il rischio per un neonato di necessitare di manovre di stabilizzazione o rianimazione alla nascita. Tuttavia, questi fattori di rischio non sono ancora completamente noti, e al momento non esiste un modello universalmente applicabile per la previsione di situazioni ad alto rischio. Considerando entrambe queste limitazioni e il fatto che la necessità di rianimazione alla nascita è un evento raro, è essenziale garantire una formazione periodica del personale sanitario responsabile dell'assistenza neonatale in sala parto.

In questo lavoro di tesi, viene proposto un approccio basato su metodologie di machine learning, per identificare i fattori di rischio e il loro impatto sull'evento nascita, utilizzando dati di pazienti reali. Questo potrà essere utile al personale sanitario per incrementare progressivamente e aggiornare le proprie conoscenze. Durante questo lavoro di tesi, è stato progettato e sviluppato un sistema che supporti il processo decisionale, in grado di garantire sia l'inserimento di nuovi pazienti - così da accrescere la base di conoscenza - sia generare una Rete Bayesiana per predire differenti fattori di rischio identificati dal medico esperto del dominio. Inoltre, questo sistema di supporto alle decisioni è in grado di migliorare e pianificare la capacità di riconoscimento e pianificazione degli interventi appropriati sui pazienti ad alto rischio.

Capitolo 1

Introduzione

La maggior parte dei neonati (circa l'85%) compie la transizione dalla vita intrauterina a quella extrauterina senza necessità di assistenza. Una percentuale ridotta richiede stimolazioni per avviare la respirazione (10%), ventilazione a pressione positiva (5%) o intubazione (2%), mentre meno dell'1% necessita di manovre rianimatorie avanzate. [8, 2, 7, 1, 4] Sebbene gli interventi più complessi siano eventi rari, la sala parto rappresenta un contesto altamente stressante, poiché il personale deve essere costantemente preparato a gestire anche gli scenari più critici, in termini sia psicologici sia di competenze professionali. Errori di comunicazione o la mancanza di esperienza in casi insoliti possono infatti comportare esiti estremamente gravi, fino al decesso neonatale.

Stime recenti indicano che, in assenza di una corretta stratificazione del rischio, circa la metà dei neonati che necessitano di ventilazione potrebbe non essere identificata prima del parto. Attualmente, non esiste un modello universalmente applicabile per prevedere la necessità di rianimazione o supporto durante la transizione, e l'elenco dei fattori di rischio presente nelle linee guida [1] non risulta esaustivo.

Al fine di supportare il personale sanitario, questo progetto di tesi, propone una web application che implementa un sistema di supporto alle decisioni [6] basato su tecniche di Machine Learning (ML) (Bayesian Network, (BN)). Questa web application è stata implementata mediante le Java-API del tool di ML WEKA¹. Questo ha consentito di imparare modelli di reti bayesiane che evidenziano le correlazioni tra le differenti variabili che modellano i fattori di rischio neonatali. L'interazione con il sistema avviene tramite un browser che permette agli operatori, previa autenticazione, compilare il form per l'inserimento di un nuovo paziente, imparare una rete bayesiana selezionando differenti classi (fattori di rischio da predire) e, infine, di visualizzare il modello BN appreso.

La web application che è stata sviluppata in questo lavoro di tesi, si propone come uno strumento di supporto alle decisioni cliniche, in grado di rappresentare schematicamente le situazioni più frequenti durante il periodo perinatale. Tuttavia, non sostituisce il giudizio medico in presenza di fattori antepartum imprevisti: resta pertanto responsabilità del medico valutare il quadro complessivo e informare il team circa un incremento del rischio di complicanze.

Un ulteriore ambito di applicazione riguarda l'analisi delle relazioni tra outcome neonatali e fattori antepartum e intrapartum. La conoscenza generata dal sistema è esclusivamente derivata dal dataset a disposizione, evitando preconcetti sulle possibili correlazioni tra eventi e fattori di rischio. Di conseguenza, sebbene dataset limitati possano produrre risultati privi di riscontro clinico, un insieme di dati sufficientemente esteso consente di individuare correlazioni significative, comprese quelle in cui il rapporto causa-effetto non è immediatamente evidente.

¹<https://ml.cms.waikato.ac.nz/weka/>

Capitolo 2

Scopo del lavoro

L'obiettivo di questo lavoro di tesi è stato quello di sviluppare una web application per fornire un sistema di supporto alle decisioni al team medico-sanitario, in grado di predire, a partire da un dataset di pazienti anonimizzati, l'indice di rischio APGAR 1 per i pazienti in sala parto, attraverso l'utilizzo di modelli di machine learning come le Reti Bayesiane.

2.1 Sistemi di supporto alle decisioni in ambito sanitario

I sistemi di supporto alle decisioni cliniche (CDSS) [6] sono strumenti informatici progettati per aiutare medici e operatori sanitari nel processo decisionale, fornendo suggerimenti basati sui dati clinici del paziente. L'obiettivo è migliorare la qualità dell'assistenza sanitaria. Con l'introduzione dell'intelligenza artificiale (IA), i CDSS si sono evoluti grazie a tecniche come sistemi esperti, ragionamento causale, logica fuzzy, reti bayesiane e case-based reasoning. Secondo la Barcelona Declaration, le tecniche di IA applicate ai CDSS si dividono in:

- **IA basate sulla conoscenza (knowledge-based):** modellano la conoscenza umana con un approccio top-down, utilizzando concetti, regole e buon senso. Si fondano su tecnologie simboliche sofisticate.
- **IA guidate dai dati (data-driven):** seguono un approccio bottom-up, sfruttano grandi quantità di dati e algoritmi di machine learning per individuare pattern e fare previsioni. Richiedono elevate risorse computazionali.

Introduzione agli indici di rischio in sala parto

I fattori di rischio si suddividono in tre grandi macrocategorie: 1. Fattori antepartum materni: sono i fattori relativi alla donna incinta. Possono essere relativi alla sua stazza, a patologie pregresse o a complicazioni emerse durante il periodo gestazionale. 2. Fattori antepartum fetali: per lo più fattori relativi a rilevazioni effettuate sul feto. 3. Fattori intrapartum: complicanze emerse in sala parto.

2.2 Modelli di ML supervisionati: Le Reti Bayesiane

Le reti bayesiane [3] sono modelli probabilistici rappresentati tramite grafi diretti aciclici (DAG), in cui ogni nodo corrisponde a una variabile (clinica, genetica, ambientale, ecc.) e ogni arco indica una dipendenza condizionata o un rapporto causa-effetto. L'algoritmo TAN (Tree-Augmented Naive Bayes) estende il classico modello Naive Bayes consentendo la presenza di dipendenze tra variabili predittive, migliorando così la capacità di rappresentare relazioni complesse.

Il loro scopo principale è gestire l'incertezza e ragionare probabilisticamente su variabili complesse anche in presenza di dati incompleti, supportando la predizione di eventi, decisioni cliniche personalizzate, analisi causale e integrazione di dati eterogenei. Grazie alla possibilità di aggiornare le probabilità con nuove evidenze, risultano particolarmente efficaci in medicina e bioinformatica.

Panoramica delle API di Weka Per costruire reti bayesiane con Weka si utilizzano principalmente:

- **Instances:** i dati che serviranno a costruire il Bayesian Web. Le Instances si possono importare da file ARFF, CSV o da database mediante il JDBC
- **Filter:** utili a preprocessare i dati (rimuovere, modificare, ecc.). I filtri operano se si verificano le condizioni indicate.
- **Classifier/Clusterer:** le strutture costruite dai dati di partenza. I Bayesian Web fanno parte di questa categoria.
- **Evaluating:** giudicare la bontà e la validità del Classifier/Clusterer generato. Nel nostro caso si è usato la 10-fold cross-validation.
- **Attribute selection:** per la rimozione degli attributi non necessari. Nel nostro caso, si rimuovono tutti gli outcome neonatali ad eccezione di quello per cui vogliamo effettuare le predizioni.

Capitolo 3

Metodologie e strumenti utilizzati

Lo scopo del sistema presentato in questo documento è quello di fornire un'interfaccia web partendo dai dati comunicati dal team medico che ha collaborato al progetto. L'interfaccia in grado di ricostruire graficamente le reti bayesiane generate, di generare un form con gli input a loro necessari e di aggiungere nella conoscenza del sistema nuovi casi. A seguire tutti i dettagli.

3.1 Raccolta dei dati e fattori di rischio antepartum e intrapartum

Lo studio comprende 286 nati presso l'Ospedale Pediatrico Cardinal Massaia di Asti. Non sono stati applicati criteri di esclusione. Il Dipartimento Pediatrico del Cardinal Massaia è un centro spoke che include una nursery e un reparto di Patologia Neonatale; i pazienti che necessitano di cure più complesse vengono trasferiti al centro hub di riferimento.

I dati relativi ai fattori di rischio antepartum e intrapartum sono stati raccolti in forma anonima. Nei 286 parti presi in esame sono stati osservati i seguenti esiti principali:

- 54 neonati hanno un valore di APGAR 1' ≤ 7
- 12 nuovi nati hanno avuto bisogno di ventilazione alla nascita
- 18 casi di distress respiratorio alla nascita
- 8 neonati son stati trasferiti presso l' unità di terapia intensiva neonatale (TIN)
- 122 nuovi nati son stati trasferiti presso la patologia neonatale
- Non sono riportati casi di ventilazione meccanica prolungata
- 5 neonati hanno richiesto una ventilazione non invasiva

I restanti fattori di rischio riguardano patologie materne (ad es. fumo 13 casi, ipotiroidismo 10, ipertensione 2), condizioni ostetriche (es. preeclampsia 4 casi, gravidanza non seguita 2 casi, TC urgente 24 casi) e parametri neonatali (es. EGj37 in 5 casi, macrosomia in 15, IUGR in 2).

3.2 Requisiti funzionali e non funzionali

- **1** Il sistema permette di inserire i dati relativi a nuovi parti.
- **2** All'occorrenza il sistema mostra un resoconto del paziente prima di inserirlo nel database
- **3** Il sistema permette di generare la rete bayesiana su un qualunque fattore di rischio.
- **6** Il sistema prevede l'autenticazione dell'utente.
- **7** Le funzionalità di inserimento e predizione son permesse solo previa autenticazione.
- **Requisiti non funzionali** Sicurezza: il sistema memorizza l'hashing delle password anzichè le password in chiaro. L'algoritmo di hash utilizzato è SHA-256 con pepper.

3.3 Casi d'uso - Diagramma UML

L'accesso alle funzionalità del sistema (inserimento pazienti o predizione outcome) richiede l'autenticazione tramite email e password. Se l'utente non è autenticato, viene reindirizzato al login. Dopo l'autenticazione, l'utente accede alla home e può scegliere tra:

- Inserimento dati paziente: si compila un form con campi (prevalentemente checkbox) suddivisi per categorie di fattori di rischio e outcome verificati, con possibilità di visualizzare un riepilogo.
- Predizione outcome: l'utente seleziona l'outcome da un menu a tendina e sceglie se generare un albero decisionale o una rete bayesiana, viene mostrato il modello selezionato, e viene generato un form basato sui nodi. Dopo la compilazione, il sistema restituisce la previsione dell'outcome selezionato.

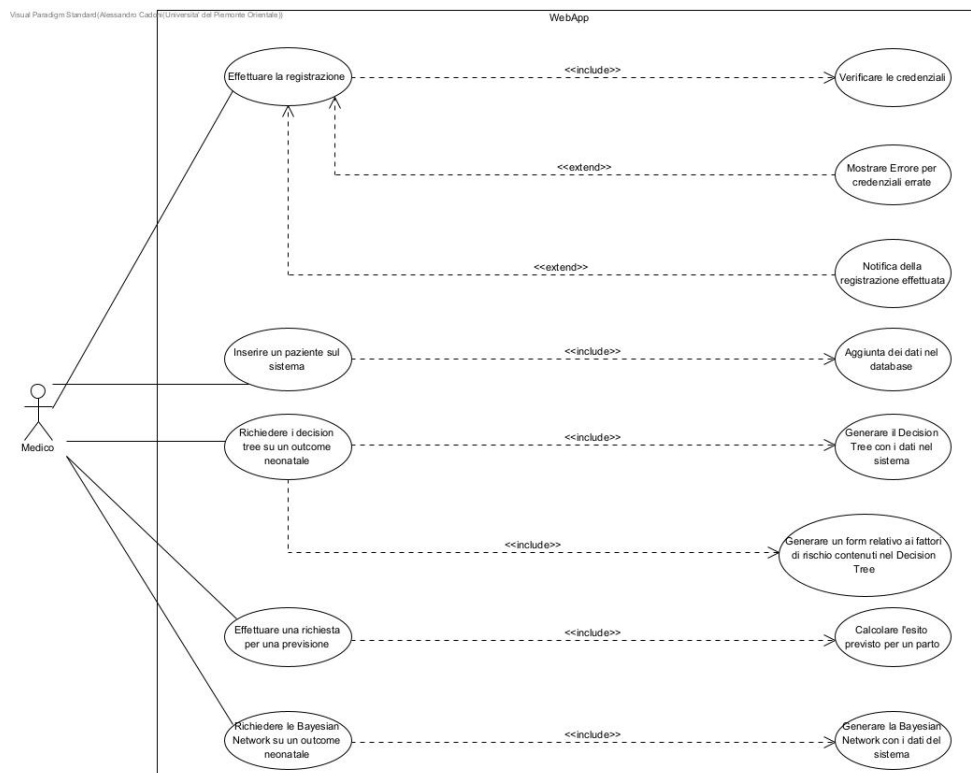


Figura 3.1: Diagramma UML dei casi d'uso

Capitolo 4

Lavoro svolto

In questo capitolo vengono introdotti ed esaminati tutti gli strumenti utilizzati per sviluppare il sistema e tutte le tecnologie adottate. Inoltre è anche possibile visionare qualche passo di progettazione, come lo schema E-R del database.

4.1 Modello dei dati: Struttura dei dati del paziente

La letteratura individua numerosi fattori di rischio pre-, peri- e post-natali che possono influenzare l'esito neonatale e la necessità di interventi rianimatori o supporto ventilatorio alla nascita. Tali fattori comprendono condizioni materne pregresse, complicanze insorte durante la gravidanza, elementi intercorsi durante il travaglio e il parto, nonché parametri clinici neonatali rilevati immediatamente dopo la nascita.

Tra i principali elementi pre- e peri-natali associati a maggiore rischio figurano: età gestazionale inferiore a 37 settimane o superiore a 41, restrizione della crescita intrauterina (IUGR), macrosomia, malformazioni congenite significative, oligo- o polidramnios, presentazioni anomale (es. podalica), taglio cesareo urgente, utilizzo di forcipe o ventosa, liquido amniotico tinto di meconio, distacco intempestivo di placenta, prolasso di funicolo, gravidanza gemellare, mancata profilassi steroidea in caso di parto pretermine, nonché condizioni materne quali obesità, statura ridotta, ipertensione, preeclampsia, diabete gestazionale e patologie croniche.

Tabella 4.1: Organizzazione compatta degli attributi del dataset in base alle categorie di rischio

Fattori Afterpartum Fetal	Fattori Afterpartum Materni	Fattori Interpartum
Età (num)	Patologie: no, ipertensione, ipertiroidismo, depressione, fumo, ipotiroidismo, epilessia, calcoli renali, discopatia, fibroadenoma, fibromi uterini+fumo, ipotiroidismo+celiachia, epatite B+ipertensione, trombofilia, morbo di Chron+artrite, displasia fibromuscolare, ipotiroidismo+fumo, RCU, tossicodipendente+HCV	EG <37 sett. (no/si)
PN (num)	BMI >25 (si/no)	EG >41 sett. (no/si)
Settimane; Giorni; Giorni totali (num)	Statura <150 cm (no/si)	Preeclampsia (no/si)

Fattori Afterpartum Fe- tali	Fattori Afterpartum Mater- ni	Fattori Interpartum
Anemia feto (no)	Gravidanza non seguita (no/si)	Ipertensione (no/si)
Polidramnios (no)	Mancata profilassi steroidi (no)	Diabete gestazionale (no/- si)
Oligodramnios (no/si)	Infezioni TORCH (no/lue/cm- v/hbv/toxo/hcv)	Sofferenza fetale (no/si)
Idrope (no)		Liquido amniotico tinto di meconio (si/no)
Macrosomia (no/si)		Presentazione podalica/a- nomala (no/si)
IUGR (no/si)		Ventosa/forcipe (no/si)
Malformazioni significative (no/cardiopatia/monorene)		Taglio cesareo <39 sett. (no/si)
pH feto; pH cordone (num)		Taglio cesareo urgente (no/si)
SBEf; SBEC; Lattati (num)		Anestesia generale (no/on)
APGAR 1 min (0/1)		Distacco di placenta (no)
Ipotonia a 30 sec (no)		Sanguinamento intrapartum (no/si)
Distress respiratorio (no/si)		Corioamnionite (no/si)
NIV; Durata NIV (no/si)		Distocia di spalla (no/si)
Durata VM (no)		Prolasso cordone ombelica- le (no)
Trasferimento in PN (si/- no)		
Ipoglicemia (no/si)		

4.2 Architettura dell'applicazione web

La Figura 4.1 mostra l'architettura della web application sviluppata, organizzata secondo il classico paradigma client-server. I principali attori sono: il client (web browser), il server applicativo (Node.js), un database relazionale e il file system che memorizza il dataset in formato ARFF.

Di seguito, viene fornita una breve descrizione del flusso della pipeline.

1. Il client invia al server una richiesta **HTTP POST** verso l'endpoint `/insert` con un payload JSON contenente i dati del paziente inseriti tramite form.
2. Il server valida il payload e, se corretto, esegue l'INSERT nel database (`INSERT INTO paziente VALUES (...);`).
3. Contestualmente, il server inserisce in modalità la stessa istanza al file `pazienti.arff` (`fs.append(pazienti.arff, s);`), dove `s` è la riga ARFF generata a partire dal JSON. Il file ARFF (Attribute-Relation File Format) è il dataset usato dall'applicativo di Machine Learning per addestrare modelli come **Decision Tree** e **Bayesian Network**.
4. Se le operazioni vanno a buon fine, il server risponde al client con **HTTP 200 OK**. (In caso di errore di validazione, insert o append, verrebbe restituito un opportuno codice di errore).

Predizione

Per le richieste di predizione, l'applicativo Node.js avvia un **processo figlio** in cui istanzia la JVM ed esegue il bytecode Java del classificatore; al termine delle computazioni, il programma Java ritorna l'esito della predizione al server, che lo inoltra al client. I modelli utilizzati in questa fase sono addestrati sul dataset costruito incrementando `pazienti.arff` durante gli inserimenti. Al centro di Figura 4.1 viene rappresentato il server applicativo; a sinistra il database relazionale; a destra il file ARFF; le frecce numerate (1-4) indicano la sequenza delle operazioni nel percorso di inserimento dei dati.

4.3 Scelta delle tecnologie e framework

Backend

- **Node.js**: sistema run-time che esegue del codice JavaScript. JavaScript per anni è rimasto confinato all'esecuzione all'interno degli engine dei browser. Utilizzare del codice JavaScript anche per il backend permette di utilizzare un solo linguaggio lungo tutti i livelli del software stack.
- **Express.js**: il framework per applicazioni web per implementare delle API RESTful mediante Node.js. È il server framework de facto per quanto riguarda Node.js. Tra le funzionalità a supporto più importanti si evidenziano un routing solido e il supporto alla programmazione asincrona.
- **Passport.js**: middleware per l'autenticazione compatibile con Express.
- **EJS**: linguaggio di templating per la generazione di pagine HTML (Server-side rendering).
- **node-graphviz**: modulo JavaScript per generare le immagini di grafi dalla loro descrizione secondo il linguaggio DOT.

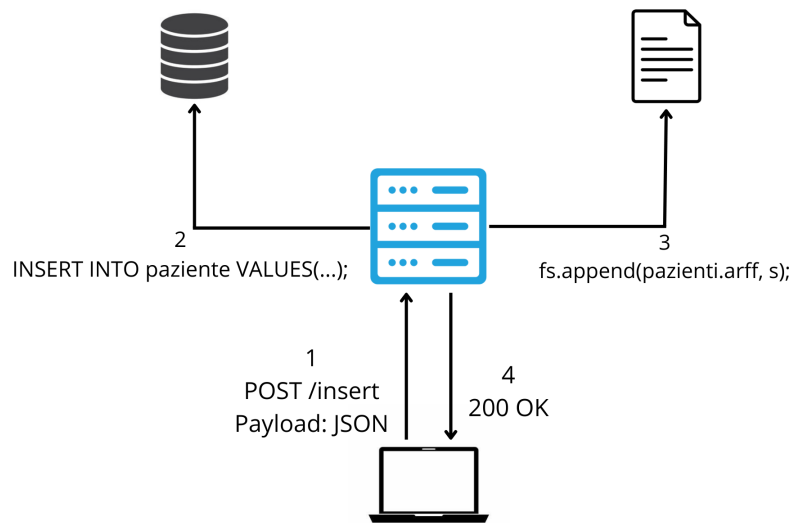
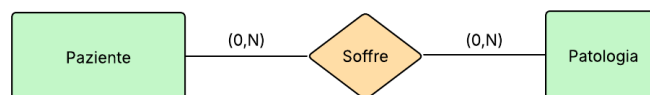


Figura 4.1: Architettura della web application per la predizione degli indici di rischio

- **Java:** linguaggio di programmazione orientato agli oggetti il cui focus principale è l'indipendenza dal sistema. Questa funzionalità è possibile grazie al fatto che il bytecode viene poi interpretato dalla Java Virtual Machine (JVM).
- **Weka:** Weka è lo strumento open-source di riferimento nell'ambito del machine learning. Weka implementa tutti gli algoritmi principali nell'ambito del ML e fornisce anche delle interfacce grafiche.

DBMS - SQLite SQLite: DBMS SQL di pubblico dominio. L'utilizzo di un database relazionale è risultato essere molto adatto alla struttura delle informazioni fornite. Di seguito lo schema E-R del database. Il database è stato organizzato con tabelle separate tra paziente



e patologia, con una tabella intermedia che esplicita tutte le coppie paziente, patologia esistenti.

Frontend HTML5: ultima versione del linguaggio usato per strutturare pagine web. CSS3: linguaggio per definire lo stile e la formattazione delle pagine HTML. JavaScript: linguaggio lato client che permette di rendere le pagine dinamiche, distribuire il carico, eseguire il rendering lato client e altro; viene eseguito dal browser dell'utente.

Capitolo 5

Lavoro svolto

5.1 Sviluppo del Back-End

Gestione dei dati del paziente e database SQLite

Il dataset dei pazienti è memorizzato in file ARFF, derivati da Excel convertito in CSV, con attributi numerici o nominali e patologie rappresentate come valori binari. Per risolvere ambiguità e campi multipli, i dati sono gestiti tramite un database relazionale SQLite, che garantisce vincoli di integrità e maggiore flessibilità. Ogni aggiornamento del database comporta la rigenerazione del file ARFF per mantenerne la coerenza. L'interfacciamento avviene tramite Node.js con il modulo sqlite3, utilizzando wrapper basati su Promise per semplificare l'uso asincrono con async/await.

Sviluppo della funzione Weka per la generazione di Bayesian network

La generazione della rete bayesiana nel codice è stata implementata combinando Node.js e Weka tramite un programma Java ausiliario. Il processo funziona così: il file Java (DecisionTreeExample.java) legge un dataset in formato ARFF e costruisce un classificatore **BayesNet** utilizzando le API di Weka. Il programma legge un dataset in formato ARFF e costruisce un BayesNet specificamente usando l'algoritmo TAN (Tree-Augmented Naive Bayes) come metodo di ricerca delle dipendenze tra variabili. Inoltre, viene utilizzato un SimpleEstimator con prior pari a 0.5 per stimare le probabilità condizionate dei nodi.

Di seguito nel Listing 5.1 viene riportata la funzione principale che genera il modello di Bayesian Network: essa prende come argomento la modalità (bayes) e l'attributo target (classe) su cui basare la classificazione. Dopo aver impostato l'indice della classe, il programma costruisce la rete bayesiana e ne stampa il grafo in formato DOT, pronto per essere visualizzato con strumenti come Graphviz. Il codice presentato, è stato sviluppato attraverso l'utilizzo delle API di Weka.

Listing 5.1: Addestramento di un modello Bayesian Network (TAN) sul dataset dei pazienti

```
1  /**
2   * Allena una Rete Bayesiana con algoritmo TAN (Tree Augmented Naive Bayes)
3   * e salva la struttura del modello in formato XMLBIF.
4   *
5   * @param data    dataset ARFF già caricato e con attributo classe impostato
6   * @param outArg  percorso di output (es. 'public/images/bayes-distress.png'),
7   *               usato solo come base per derivare il file .xml
8   */
9  private static void trainBayesNet(Instances data, String outArg) throws Exception {
10     // Crea una nuova rete bayesiana vuota
11     BayesNet net = new BayesNet();
12
13     // Imposta l'algoritmo di ricerca delle dipendenze: TAN
```

```

14 // TAN = Tree Augmented Naive Bayes, una versione avanzata di Naive Bayes
15 TAN tan = new TAN();
16 net.setSearchAlgorithm(tan);
17
18 // Imposta lo stimatore per le probabilità condizionate
19 // SimpleEstimator con alpha=0.5 serve per lo 'smoothing'
20 // (evita probabilità nulle quando un evento non compare nel dataset)
21 SimpleEstimator est = new SimpleEstimator();
22 est.setAlpha(0.5);
23 net.setEstimator(est);
24
25 // Addestra la rete bayesiana sul dataset
26 net.buildClassifier(data);
27
28 // Se è stato fornito un percorso di output,
29 // salva la struttura della rete in formato XMLBIF 0.3
30 if (outArg != null) {
31     String xmlPath = outArg.replace('.png', '.xml');
32     File f = new File(xmlPath);
33     f.getParentFile().mkdirs(); // crea cartelle se non esistono
34
35     try (FileWriter fw = new FileWriter(f)) {
36         fw.write(net.toXMLBIF03());
37     }
38 }
39 }

```

5.2 Sviluppo del frontend

Nel frontend dell'applicazione è stata realizzata un'interfaccia dedicata alla generazione delle reti bayesiane, con l'obiettivo di rendere il processo di analisi e predizione intuitivo anche per utenti non esperti. La pagina consente di selezionare, tramite un menu a tendina, l'attributo del dataset su cui effettuare la predizione. Tra le opzioni disponibili vi sono vari fattori clinici, come *APGAR 1*, *Distress respiratorio*, *Ipoglicemia*, *Ventilazione non invasiva*, *Preeclampsia* e altri parametri rilevanti per l'analisi neonatale.

Una volta selezionato l'attributo di interesse, l'utente può avviare l'elaborazione premendo il pulsante *"Crea rete Bayesiana"*. L'azione genera una richiesta HTTP di tipo GET indirizzata all'endpoint `/generateBayesNet`, che viene gestito dal server Node.js. Quest'ultimo, tramite le API Java sviluppate con le librerie Weka, costruisce la rete bayesiana corrispondente, identificando le relazioni probabilistiche tra i vari attributi del dataset in base all'attributo target scelto.

Il risultato dell'elaborazione è un'immagine che rappresenta graficamente la struttura della rete, con i nodi corrispondenti alle variabili e gli archi che ne indicano le dipendenze condizionate. L'immagine viene restituita al frontend e visualizzata dinamicamente nella stessa pagina, in una sezione dedicata alla visualizzazione dei risultati. Questo consente all'utente di osservare immediatamente la rete generata, senza dover ricaricare o navigare tra pagine differenti.

L'interfaccia è stata progettata con criteri di semplicità e chiarezza, includendo solo gli elementi essenziali: il campo di selezione dell'attributo, i pulsanti di esecuzione e l'area per la visualizzazione grafica del risultato. In questo modo, l'utente può concentrarsi sull'analisi del modello generato e comprendere in modo intuitivo le relazioni tra i fattori di rischio, facilitando l'esplorazione dei dati e la comprensione dei modelli probabilistici ottenuti.

Attributo su cui effettuare la predizione:

Crea albero di decisione
Crea rete Bayesiana

Figura 5.1: Form del Frontend per la selezione dell'attributo

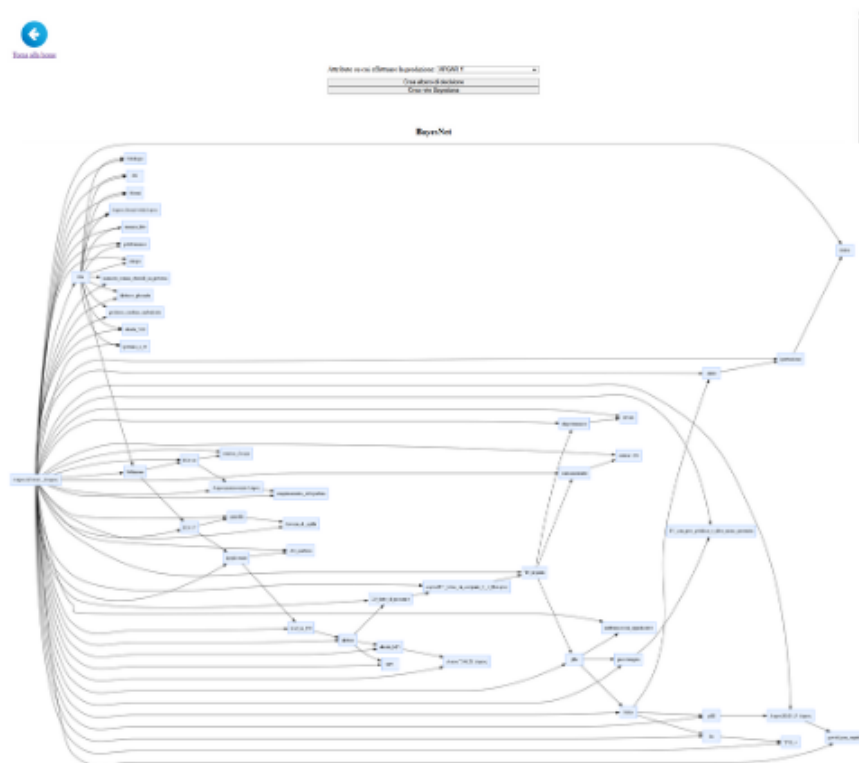


Figura 5.2: Form del Frontend per la selezione dell'attributo

Capitolo 6

Discussione sui risultati ottenuti

Il modello di rete Bayesiana (reti nelle Figure 6.1, 6.2, 6.3), mostrano una buona accuratezza complessiva sul dataset dei pazienti. Il modello è stato adddestrato utilizzando la 10 fold-cross validation e ottenendo un'accuratezza del 90,2%, classificando correttamente 258 istanze su 286.

Il dettaglio dei risultati delle performance del modello è riportato nella Tabella 6.1. L'indice di Kappa (0,656) indica un accordo sostanziale oltre il caso. Gli errori probabilistici risultano contenuti (MAE 0,112; RMSE 0,261) e, rispetto al classificatore di riferimento, l'errore assoluto si riduce di circa il 36

L'analisi per classe evidenzia un comportamento eterogeneo: per la classe 0, il modello raggiunge recall 0,648 e precisione 0,795, corrispondenti a un F1=0,714, con un basso tasso di falsi positivi (0,039); per la classe 1, invece, si ottiene un recall 0,961 e una precisione 0,921, culminando in un F1=0,941, ma con un tasso di falsi positivi più elevato (0,352).

La matrice di confusione (mostrata in Tabella 6.2) predice 35 veri positivi e 223 veri negativi, contro 28 errori complessivi) conferma che la prestazione globale è trainata dalla classe prevalente, che viene modellata con elevata sensibilità e precisione. In termini di discriminazione globale, le AUC-ROC risultano elevate e simmetriche (0,888 per entrambe le classi), mentre le AUPRC divergono (0,817 per la classe 0 e 0,939 per la classe 1), segnalando che il modello riesce a mantenere un buon compromesso precision-recall anche nella classe meno rappresentata. Il coefficiente di correlazione di Matthews (MCC=0,661) conferma una correlazione moderata.

In sintesi, la rete bayesiana mostra buone capacità predittive bilanciate, riuscendo a garantire un'accurata separazione tra le classi e un'elevata robustezza probabilistica. Per applicazioni cliniche, il modello risulta promettente in quanto riduce sensibilmente gli errori assoluti e mantiene un livello di affidabilità elevato in termini di AUC e misure pesate di precisione e richiamo.

Class	TP Rate	FP Rate	Prec	Recall	F-Measure	MCC	ROC Area	PRC Area	Acc
0	0.648	0.039	0.795	0.648	0.714	0.661	0.888	0.817	
1	0.961	0.352	0.921	0.961	0.941	0.661	0.888	0.939	
Wgt. Avg.	0.902	0.293	0.898	0.902	0.898	0.661	0.888	0.916	90.2

Tabella 6.1: Risultati ottenuti tramite il modello (Detailed Accuracy By Class)

a	b	<- classified as
35	19	a = 0
9	223	b = 1

Tabella 6.2: Matrice di confusione ottenuta tramite il modello



Figura 6.1: Dettaglio della Bayesian Network generata per APGAR 1

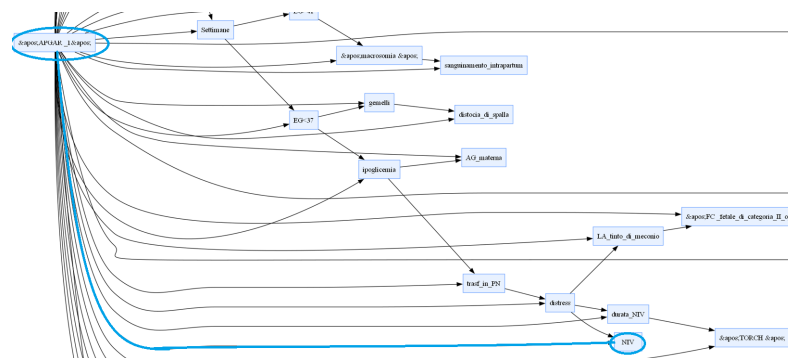


Figura 6.2: Dettaglio della Bayesian Network generata per APGAR 1 con correlazione tra APGAR 1 e NIV

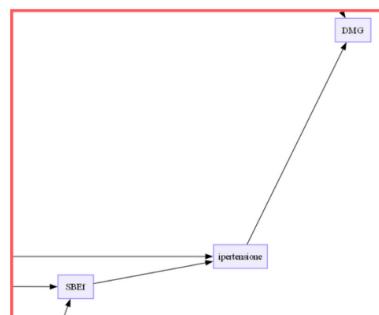


Figura 6.3: Dettaglio della Bayesian Network generata per APGAR 1 con ipertensione

Capitolo 7

Conclusioni e sviluppi futuri

7.1 Limiti del lavoro svolto

Il sistema di machine learning si basa su un dataset di circa 250 parti, un campione limitato che può generare correlazioni spurie o frequenze anomale; un numero maggiore di dati ridurrebbe tali effetti secondo la legge dei grandi numeri. Inoltre, Weka non supporta la predizione diretta di valori numerici: per superare questo limite, gli outcome devono essere discretizzati in categorie, come avviene per l'indice di APGAR, suddiviso in valori critici (0–7) e regolari (8–10).

7.2 Possibili miglioramenti e sviluppi futuri

Nelle reti bayesiane costruite con i dati disponibili, le correlazioni individuate riflettono solo i casi già osservati: con un dataset ridotto, alcune relazioni possono risultare poco affidabili. Con un numero maggiore di pazienti, la rete potrebbe apprendere legami probabilistici più solidi e significativi.

Un possibile sviluppo è l'ampliamento del sistema con l'ontologia clinica di SNOMED-CT [5]. In questo modo, attraverso l'integrazione dell'uso delle API di SNOMED-CT, sarà possibile standardizzare la terminologia clinica in fase di inserimento di nuovi pazienti con differenti patologie.

L'uso delle reti bayesiane ha comunque mostrato risultati interessanti, perché permette di rappresentare in modo intuitivo le relazioni di causalità e di stimare le probabilità condizionate di ogni outcome. Con l'evoluzione del dataset e degli strumenti integrati, questo approccio può diventare un supporto concreto e attendibile alle analisi cliniche.

Configurazione del servizio web su Server Cloud su Windows

Questa appendice descrive in modo dettagliato la procedura per installare, configurare e avviare il servizio web basato su **Node.js**, **Weka** e **Graphviz** su un sistema **Windows**. Le istruzioni sono valide sia per ambienti locali che per server cloud Windows, ad esempio macchine virtuali su Azure, AWS o Google Cloud.

Requisiti di sistema

- Sistema operativo: **Windows 10** o successivo
- **Java JDK 17** o superiore (necessario per Weka)
- **Node.js** versione 18 o superiore
- **Graphviz** installato e aggiunto al PATH di sistema
- Connessione Internet per il primo avvio (scaricamento automatico di `weka.jar`)

Installazione dei prerequisiti

1. **Installazione di Java JDK** Scaricare il pacchetto dal sito ufficiale Oracle: <https://www.oracle.com/java/technologies/downloads/>. Dopo l'installazione, verificare il corretto funzionamento aprendo **PowerShell** o **cmd**:

```
java -version
```

2. **Installazione di Node.js** Scaricare l'installer da <https://nodejs.org/>. Dopo l'installazione, verificare con:

```
node -v  
npm -v
```

3. **Installazione di Graphviz** Scaricare e installare da <https://graphviz.org/download/>. Durante l'installazione, selezionare l'opzione per aggiungere Graphviz al PATH. Verificare la disponibilità:

```
dot -V
```

Installazione del progetto Node.js

1. Copiare o clonare il progetto nella macchina Windows, ad esempio in:

```
C:\Users\<NomeUtente>\Desktop\NodeServerRemake
```

2. Aprire una finestra **PowerShell** o **cmd** nella cartella del progetto.
3. Installare le dipendenze del progetto con:

```
npm install
```

4. Verificare che nella sottocartella DBs siano presenti i file: `utenti.db`, `datasetPazienti.arff` e gli altri database richiesti.

Avvio del server

Per avviare il servizio web, eseguire il comando:

```
node server.js
```

Se l'avvio ha successo, verrà mostrato il messaggio:

```
Server listening at http://localhost:8080
```

A questo punto il server sarà raggiungibile dal browser all'indirizzo:

<http://localhost:8080>

Generazione dei modelli

Il servizio consente di generare modelli di machine learning attraverso Weka:

- **Alberi di decisione (J48)** per la classificazione supervisionata
- **Reti bayesiane (TAN)** per la modellazione probabilistica

Durante l'esecuzione, Weka produce file `.dot`, successivamente convertiti da Graphviz in immagini `.png` salvate nella cartella:

```
public/images/
```

Creazione di un utente di test

È possibile creare un utente amministratore o operatore mediante una richiesta HTTP locale. Aprire **PowerShell** o **cmd** e digitare una delle seguenti opzioni.

Opzione 1 — PowerShell

```
Invoke-RestMethod -Uri "http://localhost:8080/register" `
  -Method POST `
  -Body @{email="test@example.com";password="password123";ruolo="1"} `
  -ContentType "application/x-www-form-urlencoded"
```

Opzione 2 — Prompt dei comandi (cmd)

```
curl -X POST -d "email=test@example.com&password=password123&ruolo=1" ^  
  -H "Content-Type: application/x-www-form-urlencoded" ^  
  http://localhost:8080/register
```

L'utente verrà automaticamente registrato nel database `utenti.db`.

Risoluzione dei problemi comuni

- **Errore Graphviz:** assicurarsi che il comando `dot` sia riconosciuto dal sistema.
- **Errore Java:** verificare la variabile d'ambiente `JAVA_HOME` e il corretto `PATH`.
- **Errore “weka.jar non trovato”:** il file viene scaricato automaticamente al primo avvio.

Dopo aver completato questi passaggi, il servizio web sarà correttamente configurato per l'esecuzione in ambiente Windows, sia in locale sia su un server cloud, permettendo la generazione automatica di modelli di apprendimento supervisionato con Weka integrato nel server Node.js.

Bibliografia

- [1] Khalid Aziz, Chair; Henry C. Lee, Marilyn B. Escobedo, Amber V. Hoover, Beena D. Kamath-Rayne, Vishal S. Kapadia, David J. Magid, Susan Niermeyer, Georg M. Schmölzer, Edgardo Szyld, Gary M. Weiner, Myra H. Wyckoff, Nicole K. Yamada, and Jeanette Zaichkin. Part 5: Neonatal resuscitation 2020 american heart association guidelines for cardiopulmonary resuscitation and emergency cardiovascular care. *Pediatrics*, 147(Supplement 1):e2020038505E, 01 2021.
- [2] Juan Pablo Berazategui, Adriana Aguilar, Marilyn Escobedo, Douglas Dannaway, Ruth Guinsburg, Maria Fernanda Branco de Almeida, Firas Saker, Ariel Fernández, Guadalupe Albornoz, Mariana Valera, Daniel Amado, Gabriela Puig, Fernando Althabe, and Edgardo Szyld. Risk factors for advanced resuscitation in term and near-term infants: a case-control study. *Archives of Disease in Childhood - Fetal and Neonatal Edition*, 102(1):F44–F50, 2017.
- [3] Nir Friedman, Dan Geiger, and Moises Goldszmidt. Bayesian Network Classifiers. *Machine Learning*, 29(2):131–163, November 1997.
- [4] ERC Guidelines. <https://www.ircouncil.it/linee-guida-rcp-2021/>, 2021.
- [5] Dennis Lee, Nicolette de Keizer, Francis Lau, and Ronald Cornet. Literature review of snomed ct use. *Journal of the American Medical Informatics Association*, 21(e1):e11–e19, 07 2013.
- [6] Stefania Montani Manuel Striani. Artificial intelligence in clinical decision support: a focused literature survey. 28(1):120–127. Publisher: Georg Thieme Verlag KG.
- [7] Edgardo Szyld, Adriana Aguilar, Gabriel A. Musante, Nestor Vain, Luis Prudent, Jorge Fabres, and Waldemar A. Carlo. Comparison of devices for newborn ventilation in the delivery room. *The Journal of Pediatrics*, 165(2):234–239.e3, 2014.
- [8] Edgardo Szyld, Michael P Anderson, Birju A Shah, Charles C Roehr, Georg M Schmölzer, Jorge G Fabres, and Gary M Weiner. Risk calculator for advanced neonatal resuscitation. *BMJ Paediatrics Open*, 6(1), 2022.