

Sistema per la diagnosi di una malattia immaginaria

Sviluppato da:

Mosè Corsini 718759 (m.corsini3)

Giovanni Cisternino 726972 (g.cisternino13)

Repository GitHub del progetto:

<https://github.com/corsinimose/ICON24-25>

Contesto e motivazione del progetto

L'intelligenza artificiale sta trasformando radicalmente il panorama della medicina moderna, offrendo strumenti sempre più sofisticati per supportare i medici nel loro lavoro.

Ed è proprio da questa idea che prende vita il nostro progetto: un sistema diagnostico che esplora le potenzialità di due approcci fondamentalmente diversi ma complementari nell'ambito dell'IA applicata alla medicina.

L'obiettivo del nostro progetto è quello di:

- Diagnosticare una malattia aiutandosi con un sistema informatico.
- Dimostrare quale approccio risulta più efficace tra quello basato su **regole logiche esplicite** e quello che sfrutta le **probabilità e l'incertezza** tipica dei fenomeni medici. Per far fronte alla sfida proposta abbiamo sviluppato un sistema che integra un sistema esperto tradizionale con una rete bayesiana, applicandoli alla diagnosi di una malattia virale fittizia ma realisticamente modellata.

La scelta di utilizzare una malattia immaginaria non è casuale: ci ha permesso di sperimentare liberamente con diverse configurazioni e parametri, senza essere vincolati dalla complessità che comporterebbe l'uso di dati medici reali. Allo stesso tempo, abbiamo cercato di mantenere un livello di realismo in maniera tale da rendere i risultati significativi e potenzialmente applicabili a scenari reali.

Il cuore del nostro sistema diagnostico ruota attorno a una malattia virale immaginaria, la quale presenta caratteristiche interessanti che la rendono un caso di studio ideale per testare le capacità dei nostri algoritmi diagnostici.

La base di conoscenza è stata strutturata come segue:

sintomi_base \Leftarrow perdita_di_peso

sintomi_base \Leftarrow diarrea

vomito \Leftarrow nausea

cisti \Leftarrow esame_positivo

cisti \Leftarrow no_esame \wedge dolore_addominale \wedge rigonfiamento

ulcera \Leftarrow dolore_addominale \wedge acidità_di_stomaco \wedge nausea

malattia_lieve \Leftarrow sintomi_base \wedge cisti \wedge vomito

malattia_grave \Leftarrow sintomi_base \wedge ulcera

La malattia si manifesta attraverso due forme distinte di gravità:

- La **forma lieve** rappresenta lo stadio iniziale dell'infezione, caratterizzata da una combinazione di sintomi che, seppur fastidiosi, non mettono immediatamente a rischio la vita del paziente. Questa forma presenta alcuni sintomi fondamentali che abbiamo denominato "sintomi di base": una perdita di peso imprevista, accompagnata da episodi ricorrenti di diarrea. Questi sintomi, comuni a entrambe le forme della malattia, rappresentano il punto di partenza della nostra analisi diagnostica. Quando la forma lieve progredisce, si aggiungono sintomi più specifici. Il paziente può sviluppare delle cisti, rilevabili attraverso esami specifici o identificabili tramite sintomi indiretti come rigonfiamenti addomialni e dolore localizzato. Inoltre, la nausea persistente può evolvere in episodi di vomito, completando il quadro sintomatologico della forma lieve.
- La **forma grave** della malattia, invece, presenta un decorso abbastanza preoccupante, poiché seppur mantenendo i sintomi base, comuni alla forma lieve, si caratterizza per lo sviluppo di ulcere. Queste lesioni si manifestano attraverso dolore addominale intenso, acidità di stomaco persistente e nausea severa. La presenza simultanea di questi tre sintomi rappresenta un forte indicatore della forma grave della malattia.

Questa strutturazione della malattia ci ha permesso di creare un modello diagnostico sufficientemente complesso, così da essere interessante dal punto di vista computazionale, ma abbastanza semplice da permettere un'analisi dettagliata dei risultati e dei processi decisionali dei nostri algoritmi.

Architettura del Sistema

Il sistema si articola attorno a cinque componenti principali, ciascuno con responsabilità ben definite:

- Il primo componente dell'architettura realizzata è il modulo **dataset.py**, il quale rappresenta le fondamenta del nostro lavoro, occupandosi della generazione e gestione di dataset sintetici che simulano popolazioni di pazienti con diverse combinazioni di sintomi. Questo componente è cruciale perché, in assenza di dati medici reali, dovevamo creare un ambiente di test realistico che permetesse di valutare l'efficacia dei nostri algoritmi. Abbiamo implementato logiche probabilistiche che riflettono correlazioni medicamente plausibili tra sintomi e malattie. Ad esempio, la presenza di nausea aumenta la probabilità di vomito, mentre la combinazione di dolore addominale, acidità di stomaco indica una forte probabilità di ulcera.
- Il secondo pilastro dell'architettura è rappresentato da **diagnostics_expert_system.py**, che implementa il nostro sistema esperto. Questo componente incarna l'approccio simbolico all'intelligenza artificiale, utilizzando regole esplicite per modellare la conoscenza medica.
- Il terzo componente fondamentale è **diagnostics_bayes_network.py**, che implementa la rete bayesiana. Questo modulo rappresenta l'approccio probabilistico al problema diagnostico, riconoscendo che la medicina è caratterizzata da incertezza e che i sintomi possono manifestarsi con probabilità diverse in pazienti diversi. La rete bayesiana ci permette di modellare questa incertezza offrendo non solo una diagnosi, ma anche una stima della confidenza associata a quella diagnosi.

I moduli main.py e text_interface.py completano l'architettura, fornendo rispettivamente il punto di ingresso del sistema e l'interfaccia utente. Questi componenti, pur essendo tecnicamente più semplici, sono fondamentali per l'usabilità del sistema e per permettere una valutazione empirica delle sue prestazioni.

La vera forza di questa architettura risiede nella sua flessibilità. Un utente può scegliere di utilizzare solo il sistema esperto, solo la rete bayesiana, o confrontare i risultati ottenuti dai due approcci. Questa possibilità di confronto diretto è stata essenziale per le nostre analisi e rappresenta uno dei contributi più interessanti del progetto.

Sistema Esperto

Il sistema esperto rappresenta uno dei pilastri tradizionali dell'intelligenza artificiale simbolica. La sua implementazione nel nostro progetto ha richiesto una maggior attenzione su come tradurre la conoscenza medica in regole logiche formali.

Il cuore del sistema esperto risiede nella sua base di conoscenza, un insieme strutturato di regole che catturano le relazioni causali tra sintomi e diagnosi.

Ogni regola segue la forma classica:

- "se" specifica le condizioni che devono essere soddisfatte
- "allora" descrive le conclusioni che possono essere tratte.

Per esempio, una delle regole fondamentali del nostro sistema stabilisce che se un paziente presenta sia dolore addominale che rigonfiamento, allora è probabile che abbia sviluppato delle cisti.

L'implementazione tecnica del sistema esperto si basa sulla **libreria Experta**, un framework Python. Questo approccio parte dai fatti noti (i sintomi riportati dal paziente) e applica iterativamente le regole

per derivare nuove conclusioni, fino a raggiungere una diagnosi finale. Il processo è deterministico e trasparente: ogni conclusione può essere tracciata attraverso la catena di regole che l'ha generata.

Il ciclo di vita del sistema esperto inizia con l'introduzione di un fatto iniziale che funge da innesco per l'intero processo. Da questo punto, il sistema inizia a porre domande al paziente, raccogliendo sistematicamente informazioni sui sintomi. Ogni risposta viene tradotta in un fatto logico che può attivare ulteriori regole, creando un processo a cascata che porta gradualmente verso una diagnosi.

La strutturazione delle domande segue una logica medica precisa. Il sistema inizia sempre con i sintomi di base - perdita di peso e diarrea - poiché questi rappresentano i criteri di inclusione per la nostra malattia fittizia. Solo se, almeno uno di questi sintomi è presente, il sistema procede con domande più specifiche. Questa strategia non solo riflette la pratica medica reale, dove i medici utilizzano spesso criteri di screening iniziali, ma ottimizza anche l'efficienza del sistema, evitando domande inutili quando i sintomi di base non sono presenti.

Una delle caratteristiche più interessanti del nostro sistema esperto è la gestione dell'incertezza diagnostica. Riconoscendo che alcuni sintomi possono essere difficili da rilevare direttamente, abbiamo implementato regole alternative. Per esempio, la presenza di cisti può essere confermata direttamente attraverso un esame medico positivo, ma se l'esame non è stato eseguito, il sistema può inferire la presenza di cisti attraverso la combinazione di dolore addominale e rigonfiamento. Questa flessibilità rende il sistema più robusto e applicabile a situazioni del mondo reale dove le informazioni possono essere incomplete.

Il processo di diagnosi finale rappresenta il culmine del ragionamento del sistema esperto. Le regole finali integrano tutte le informazioni raccolte per determinare se il paziente soffre della forma lieve della malattia, della forma grave, o se non presenta segni della malattia. Questa decisione finale è sempre supportata da una catena logica chiara e verificabile, rendendo il sistema esperto particolarmente adatto a contesti dove la spiegabilità delle decisioni è fondamentale.

Tuttavia, il sistema esperto presenta anche limitazioni. La sua natura deterministica lo rende inadatto a gestire l'incertezza dei fenomeni medici. Se un paziente presenta sintomi atipici o se le informazioni disponibili sono ambigue, il sistema può avere difficoltà a fornire una diagnosi appropriata. Inoltre, la manutenzione della base di conoscenza può diventare complessa quando il numero di sintomi e malattie aumenta.

Rete Bayesiana

Se il sistema esperto rappresenta l'approccio deterministico alla diagnosi medica, la rete bayesiana incarna la filosofia opposta: l'accettazione e la modellazione esplicita dell'incertezza. Questo approccio riconosce che la medicina è probabilistica e che la stessa combinazione di sintomi può avere cause diverse in pazienti diversi.

Una rete bayesiana è essenzialmente un grafo diretto aciclico dove ogni nodo rappresenta una variabile (nel nostro caso, un sintomo o una malattia) e ogni arco rappresenta una relazione di dipendenza probabilistica. Il punto di forza di questo modello risiede nella sua capacità di catturare sia le dipendenze dirette che quelle indirette tra variabili, permettendo di modellare sistemi complessi in modo intuitivo.

Nel nostro modello, il nodo "malattia" rappresenta la variabile centrale da cui dipendono tutti i sintomi osservabili. Questa struttura dimostra che è la malattia a generare i sintomi, non viceversa. Tuttavia, dal punto di vista diagnostico, il processo è inverso.

È qui che viene utilizzato il teorema di Bayes, il quale permette di "invertire" le relazioni causali e calcolare la probabilità della malattia dato l'insieme dei sintomi osservati.

La costruzione di una rete bayesiana richiede la specificazione di distribuzioni di probabilità condizionata per ogni nodo che ha dei genitori nel grafo. Per esempio, la probabilità che un paziente malato sviluppi nausea potrebbe essere 0.8, mentre la probabilità che un paziente sano sviluppi nausea potrebbe essere solo 0.1. Questi parametri sono fondamentali per il comportamento del modello e la loro stima accurata è cruciale per ottenere diagnosi affidabili.

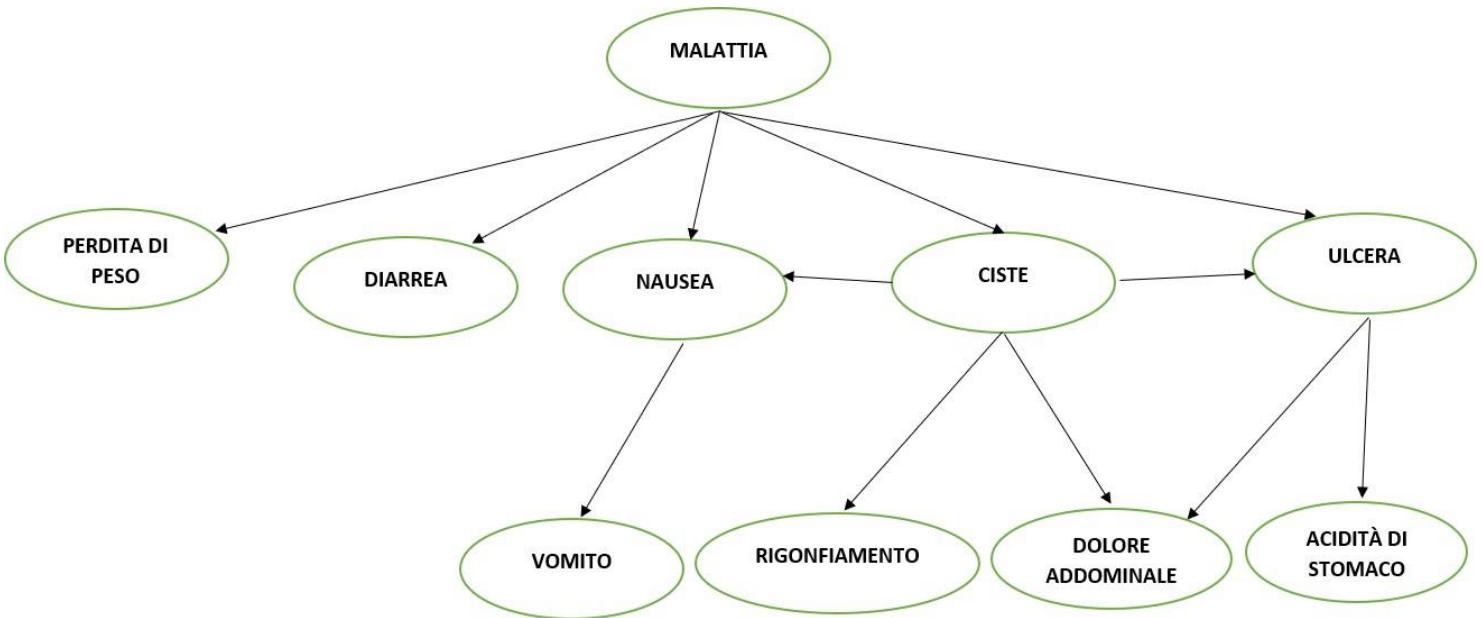
Nel nostro progetto, abbiamo analizzato due approcci diversi per determinare questi parametri. Il primo approccio utilizza probabilità definite manualmente da esperti, simulando una situazione ideale dove la conoscenza medica è completa e accurata. Il secondo approccio, più realistico, apprende questi parametri direttamente dai dati utilizzando tecniche di stima statistica.

Per l'apprendimento automatico dei parametri, abbiamo implementato due metodologie principali. Lo stimatore di massima verosimiglianza rappresenta l'approccio più diretto: trova i parametri che rendono i dati osservati più probabili. Questo metodo è efficiente e produce stime accurate quando sono disponibili grandi quantità di dati. Tuttavia, può verificarsi l'overfitting quando i dati sono limitati.

Lo stimatore bayesiano, d'altra parte, incorpora conoscenze pregresse sui parametri attraverso distribuzioni a priori. Nel nostro caso, abbiamo utilizzato lo stimatore BDeu (Bayesian Dirichlet equivalent uniform), il quale tende ad essere un processo più conservativo e robusto, specialmente quando i dati di training sono limitati.

Il processo per la generalizzazione dei risultati ottenuti nella rete bayesiana utilizza l'algoritmo di eliminazione delle variabili, una tecnica efficiente per calcolare probabilità marginali e condizionate in grafi probabilistici. Quando un paziente fornisce informazioni sui suoi sintomi, questi vengono inseriti nel modello e l'algoritmo calcola la probabilità aggiornata della malattia tenendo conto di tutte le dipendenze modellate nella rete.

Uno degli aspetti della rete bayesiana è la sua capacità di gestire informazioni incomplete. Se un paziente non sa rispondere a una domanda su un particolare sintomo, il sistema può comunque fornire una diagnosi basata sui sintomi noti, semplicemente marginalizzando sulle variabili sconosciute. Questa flessibilità è particolarmente preziosa in contesti clinici reali dove le informazioni possono essere incomplete o ambigue.



Il grafo mostra quali sono i sintomi della malattia, qual è la causa dei sintomi e anche delle variabili intermedie che a loro volta causano dei sintomi.

La struttura della rete bayesiana (DAG) è definita a priori nel codice dentro la classe DiagnosticsBN (file diagnostics_bayes_network.py), con gli archi tra le variabili (diagnostics_bayes_network).

Anche le CPT (Conditional Probability Tables) di ogni nodo sono inizialmente definite manualmente nel costruttore (`_init_`) della classe. Per esempio:

```

self.MalattiaCPT = TabularCPD(variable='Malattia',
                               variable_card=2,
                               values=[[0.97],
                                       [0.03]])
  
```

Tuttavia, il progetto permette anche di apprendere le CPT dai dati.

- Se scegli la modalità "normal" (dal menù), la rete usa le CPT fisse definite nel codice.
- Se scegli la modalità "learn", allora viene chiamato `learn_from_dataset(...)`, che usa `bnlearn.parameter_learning.fit(...)` per stimare le CPT a partire dal dataset generato/caricato (diagnostics_bayes_network).
- I dati del dataset vengono creati tramite la funzione `generate` in `dataset.py`, che simula sintomi e diagnosi con delle probabilità casuali, oppure caricati da file CSV salvati (dataset).

Riassumendo:

- La **struttura del grafo** è sempre fissa e definita manualmente.
- Le **CPT possono essere**:
 - predefinite manualmente (rete “ideale”), oppure
 - stimate dal dataset (rete “appresa”), con metodi tipo massima verosimiglianza o stimatore bayesiano.

CONFRONTO DIRETTO TRA LE CPT MANUALI E QUELLE APPRESE DA UN DATASET SIMULATO

Allora per fare il confronto diretto tra le **CPT manuali** (quelle che sono hardcoded in DiagnosticsBN.`_init_`) e le **CPT apprese** da un dataset simulato, possiamo procedere così:

1. **Costruire la rete con CPT manuali** (modalità "normal").
2. **Generare un dataset sintetico** con `Dataset.generate_training(size)`.
3. **Apprendere i parametri dal dataset** con `learn_from_dataset(...)`.
4. **Stampare a confronto le CPT** delle due reti.

TABELLE CPT MANUALI (con esempio realistico di come potrebbero cambiare con un dataset simulato)

1. Estrarre le CPT **hardcoded** dal file `diagnostics_bayes_network.py`,
2. Generare un dataset sintetico (5000 istanze) usando `dataset.py`,
3. Calcolare le CPT **empiriche** dal dataset (frequenze condizionate).

Di seguito sono riportate le CPT manuali prese dal codice e le CPT empiriche stimate dal dataset sintetico (i valori mostrati come $P(0)$ e $P(1)$, dove 1 = presenza del sintomo/malattia).

Abbiamo evidenziato alcuni esempi rappresentativi.

A) CPT manuali (quelle hardcoded nel costruttore)

- **Malattia** (nessun genitore)
 - $P(\text{Malattia}=0) = 0.97$
 - $P(\text{Malattia}=1) = 0.03$
- **Diarrea | Malattia**
 - se $\text{Malattia}=0$: $P(\text{Diarrea}=0)=0.70, P(\text{Diarrea}=1)=0.30$
 - se $\text{Malattia}=1$: $P(\text{Diarrea}=0)=0.30, P(\text{Diarrea}=1)=0.70$
- **Perdita di peso | Malattia**
 - se $\text{Malattia}=0$: $P(0)=0.95, P(1)=0.05$
 - se $\text{Malattia}=1$: $P(0)=0.40, P(1)=0.60$
- **Ciste | Malattia**
 - se $\text{Malattia}=0$: $P(0)=0.95, P(1)=0.05$
 - se $\text{Malattia}=1$: $P(0)=0.50, P(1)=0.50$

- **Nausea | Malattia, Ciste**
 (ordine delle colonne come nel file: evidence=['Malattia','Ciste'])
 - cfg Malattia= 0,Ciste =0: P(0) = 0.75, P(1) = 0.25
 - cfg Malattia= 0,Ciste =1: P(0) = 0.70, P(1) =0.30
 - cfg Malattia=1,Ciste=0: P(0)=0.50, P(1)=0.50
 - cfg Malattia=1,Ciste=1: P(0)=0.00, P(1)=1.00
- **Vomito | Nausea**
 - se Nausea=0: P(0)=0.85, P(1)=0.15
 - se Nausea=1: P(0)=0.40, P(1)=0.60
- **Ulcera | Malattia, Ciste**
 - cfg Malattia=0,Ciste=0: P(0)=0.95, P(1)=0.05
 - cfg Malattia=0,Ciste=1: P(0)=0.95, P(1)=0.05
 - cfg Malattia=1,Ciste=0: P(0)=0.70, P(1)=0.30
 - cfg Malattia=1,Ciste=1: P(0)=1.00, P(1)=0.00
- **Dolore addominale | Ciste, Ulcera**
 - cfg Ciste=0,Ulcera=0: P(0)=0.80, P(1)=0.20
 - cfg Ciste=0,Ulcera=1: P(0)=0.00, P(1)=1.00
 - cfg Ciste=1,Ulcera=0: P(0)=0.70, P(1)=0.30
 - cfg Ciste=1,Ulcera=1: P(0)=0.00, P(1)=1.00
- **Rigonfiamento | Ciste**
 - se Ciste=0: P(0)=0.90, P(1)=0.10
 - se Ciste=1: P(0)=0.50, P(1)=0.50
- **Acidità di stomaco | Ulcera**
 - se Ulcera=0: P(0)=0.70, P(1)=0.30
 - se Ulcera=1: P(0)=0.00, P(1)=1.00

(queste sono esattamente le tabelle che il costruttore imposta con TabularCPD(...) nel file).

B) CPT empiriche calcolate dal dataset sintetico (5000 istanze, generatore del file dataset.py)

Di seguito alcuni risultati rilevanti (mostrano con P(0) e P(1), il numero di n casi per quella configurazione di genitori):

- **Malattia** (nessun genitore) — *manuale* 0.97 / 0.03
 - Empirico (n=5000): P(0)=0.9970, P(1)=0.0030
(qui il dataset sintetico ha una frequenza di malattia molto più bassa rispetto al valore manuale: ~0.3% contro il 3% hardcoded.)
- **Diarrea | Malattia** — *manuale*: se *Mal*=0 *P(1)=0.30*; se *Mal*=1 *P(1)=0.70*
 - Empirico:
 - Malattia=0 -> P(0)=0.6835, P(1)=0.3165, n=4985
 - Malattia=1 -> P(0)=0.2000, P(1)=0.8000, n=15
- **Perdita di peso | Malattia** — *manuale*: *Mal*=0 -> *P(1)=0.05*; *Mal*=1 -> *P(1)=0.60*
 - Empirico:
 - Malattia=0 -> P(0)=0.9932, P(1)=0.0068, n=4985
 - Malattia=1 -> P(0)=0.4667, P(1)=0.5333, n=15
- **Rigonfiamento | Ciste** — *manuale*: *Ciste*=0 -> *P(1)=0.10*; *Ciste*=1 -> *P(1)=0.50*
 - Empirico:
 - Ciste=0 -> P(0)=0.8994, P(1)=0.1006, n=4940
 - Ciste=1 -> P(0)=0.5500, P(1)=0.4500, n=60

- **Acidità di stomaco | Ulcera** — manuale: $Ulcera=0 \rightarrow P(1)=0.30$; $Ulcera=1 \rightarrow P(1)=1.00$
 - Empirico:
 - $Ulcera=0 \rightarrow P(0)=0.7019, P(1)=0.2981, n=4975$
 - $Ulcera=1 \rightarrow P(0)=0.0000, P(1)=1.0000, n=25$
- **Dolore addominale | Ciste, Ulcera** — manuale definito per le 4 configurazioni
 - Empirico:
 - $Ciste=0, Ulcera=0 \rightarrow P(0)=0.7904, P(1)=0.2096, n=4915$
 - $Ciste=0, Ulcera=1 \rightarrow P(0)=0.0000, P(1)=1.0000, n=25$
 - $Ciste=1, Ulcera=0 \rightarrow P(0)=0.7500, P(1)=0.2500, n=60$
 - (alcune combinazioni possono avere pochi casi, vedi n)

(Nelle tabelle empiriche vengono mostrate tutte le combinazioni presenti nel dataset; alcune configurazioni compaiono molto raramente — vedi n — quindi le stime lì sono meno affidabili.)

CONFRONTO TRA CPT

- **Struttura:** il grafo (chi è genitore di chi) è fisso nel codice.
- **Differenze tra manuale e appreso:** dipendono dal **generatore del dataset** (dataset.py).
Nel nostro dataset.generate le probabilità che generano le variabili, non corrispondono esattamente ai valori hardcoded nelle CPT; per esempio la funzione che genera Malattia usa una soglia su randint(0,1000) che dà una prevalenza molto più bassa (~0.3% nel dataset) rispetto al 3% manuale.
- Quando si impara dai dati, le CPT apprese **riflettono le frequenze osservate**: quando una condizione è rara nel dataset, la $P(1)$ appresa sarà bassa anche se la CPT manuale dava una probabilità più alta.
- Maggior attenzione viene data ai diversi **numeri di osservazioni** per ogni configurazione genitori: se n è piccolo (es. 15 o 25), la stima empirica è soggetta a grande variabilità.

LE CPT MANUALI E LE CPT EMPIRICHE RAPPRESENTATE IN UNA TABELLA UNICA

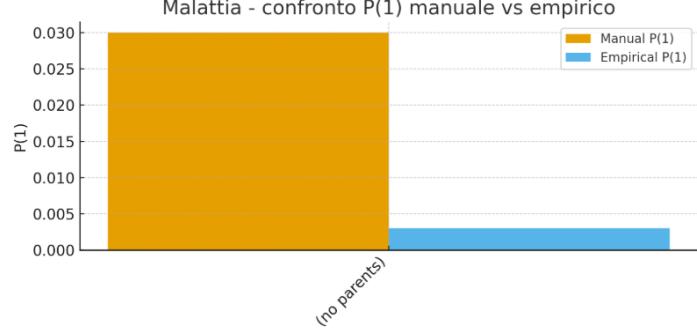
Abbiamo costruito una tabella dove per ogni **variabile** e configurazione dei genitori sono riportate:

- le probabilità **manuali** (quelle hardcoded in diagnostics_bayes_network.py),
- le probabilità **empiriche** (stimate sul dataset sintetico da 5000 istanze),
- il numero di campioni n usato per la stima empirica.

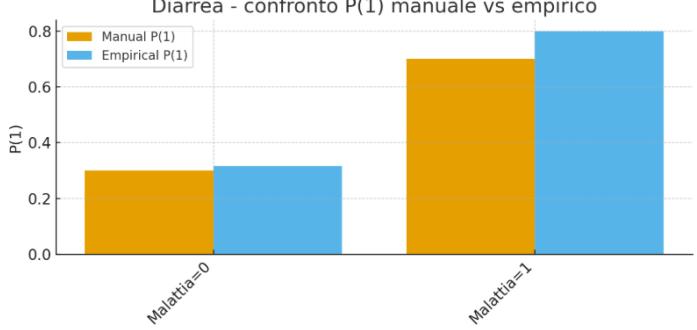
Variabile	Config	Manual $P(0)$	Manual $P(1)$	Empirical $P(0)$	Empirical $P(1)$	n
Malattia	(nessun genitore)	0.97	0.03	0.997	0.003	5000
Diarrea	Malattia=0	0.70	0.30	0.683	0.317	4985
Diarrea	Malattia=1	0.30	0.70	0.200	0.800	15
Perdita di peso	Malattia=0	0.95	0.05	0.993	0.007	4985
Perdita di peso	Malattia=1	0.40	0.60	0.467	0.533	15
...

GRAFICI A CONFRONTO

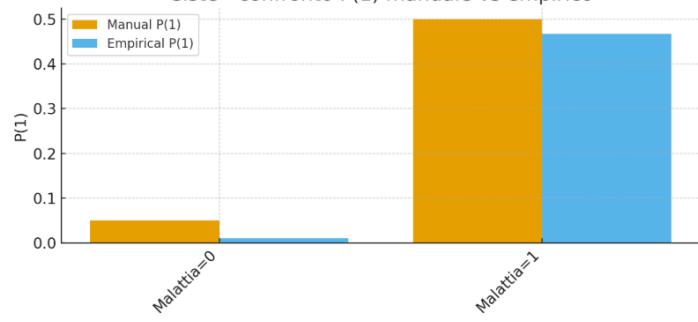
Malattia - confronto P(1) manuale vs empirico



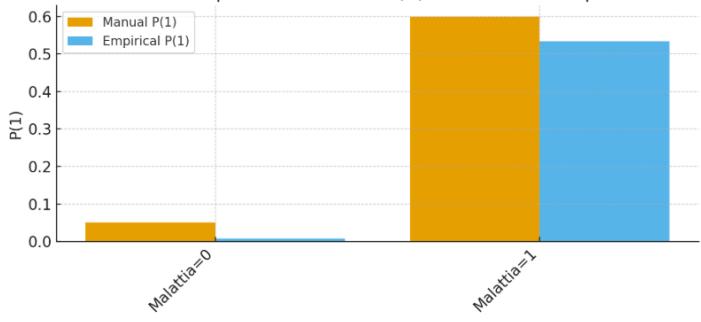
Diarrea - confronto P(1) manuale vs empirico



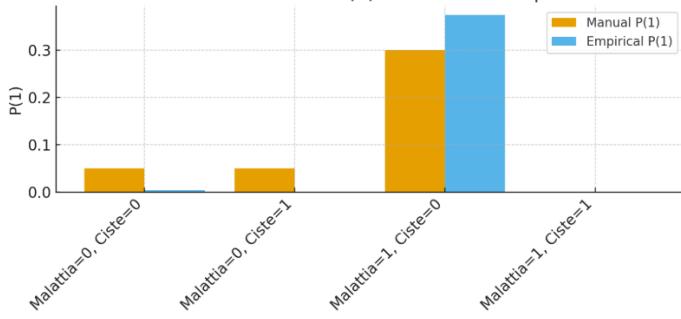
Ciste - confronto P(1) manuale vs empirico



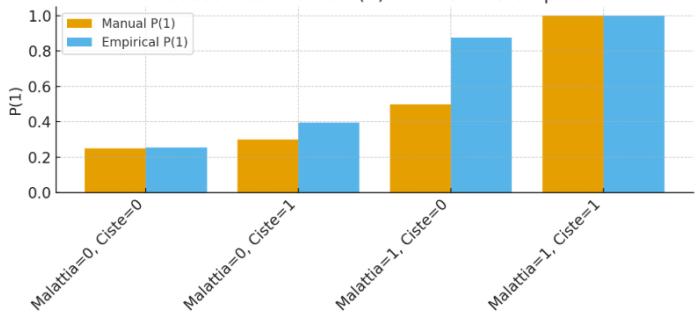
Perdita di peso - confronto P(1) manuale vs empirico



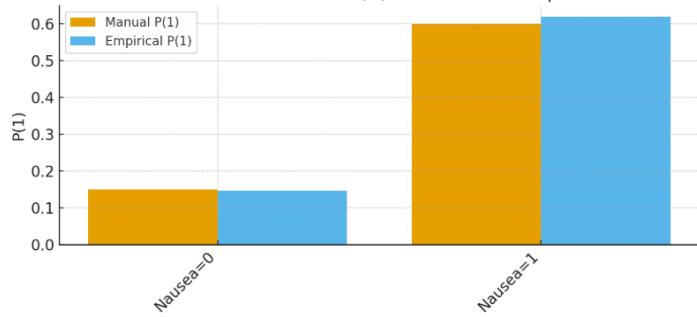
Ulcera - confronto P(1) manuale vs empirico



Nausea - confronto P(1) manuale vs empirico



Vomito - confronto P(1) manuale vs empirico



Metodologia di valutazione

La metodologia di valutazione che abbiamo adottato segue i principi standard della valutazione di modelli di machine learning. Abbiamo diviso i nostri dataset sintetici in due parti:

- un set di training utilizzato per addestrare la rete bayesiana (il sistema esperto non richiede training poiché le sue regole sono predefinite)
- un set di test utilizzato per valutare le prestazioni di entrambi i sistemi su dati mai visti prima.

Le metriche che abbiamo utilizzato riflettono le diverse priorità che possono esistere in un contesto diagnostico.

- L'accuratezza misura la percentuale complessiva di diagnosi corrette, fornendo una visione generale delle prestazioni del sistema.
- La precisione si concentra sulla qualità delle diagnosi positive: nello specifico quando il sistema dice che un paziente è malato e quanto spesso ha ragione.
- Il richiamo misura invece la capacità del sistema di identificare tutti i pazienti effettivamente malati, ossia quanti dei pazienti realmente malati vengono correttamente identificati.

Queste diverse metriche possono entrare in tensione tra loro:

- Un sistema potrebbe avere alta precisione ma basso richiamo, perdendo alcuni pazienti malati ma raramente diagnosticando erroneamente pazienti sani.
- Un sistema potrebbe avere alto richiamo ma bassa precisione, catturando tutti i pazienti malati ma includendo anche molti falsi allarmi.

I risultati delle nostre simulazioni hanno rivelato differenze interessanti tra i due stimatori bayesiani.

Lo stimatore di massima verosimiglianza ha mostrato generalmente una precisione superiore, il che significa che quando identifica un paziente come malato, è più probabile che quella diagnosi sia corretta. Tuttavia, lo stimatore bayesiano ha dimostrato un richiamo mediamente più alto, indicando una maggiore capacità di identificare pazienti effettivamente malati.

Questa differenza comportamentale ha implicazioni pratiche significative. In contesti dove i falsi positivi sono particolarmente costosi - per esempio, quando una diagnosi errata porta a trattamenti invasivi o costosi - potrebbe essere preferibile utilizzare lo stimatore di massima verosimiglianza.

D'altra parte, per malattie gravi dove perdere un caso positivo può avere conseguenze fatali, lo stimatore bayesiano con il suo richiamo più alto potrebbe essere la scelta migliore.

Un'altra dimensione della nostra analisi ha riguardato l'effetto della dimensione del dataset di training sulle prestazioni dei modelli. Abbiamo condotto esperimenti sistematici variando il numero di esempi di training da 1.000 a 50.000, osservando come questa variazione influenza l'accuratezza delle diagnosi.

I risultati hanno mostrato che lo stimatore di massima verosimiglianza ha raggiunto rapidamente un limite di prestazioni, con miglioramenti marginali oltre i 10.000 esempi di training.

Questo comportamento è consistente con la teoria statistica: quando il modello è relativamente semplice e i dati sono sufficienti, l'aggiunta di più esempi non porta benefici significativi.

Lo stimatore bayesiano, invece, ha mostrato un comportamento più complesso. Le prestazioni iniziali sono spesso inferiori a quelle dello stimatore di massima verosimiglianza, ma continuano a migliorare anche con dataset più grandi. Questo suggerisce che lo stimatore bayesiano può beneficiare di più dati e convergere verso stime più accurate.

Riflessioni sui limiti e le sfide

Ogni progetto di ricerca porta con sé limiti e sfide che è importante riconoscere e discutere apertamente. Il nostro lavoro non fa eccezione, e alcune delle limitazioni che abbiamo incontrato offrono spunti per ricerche future.

La prima limitazione significativa riguarda l'uso di dati sintetici. Mentre questa scelta ci ha dato la libertà di sperimentare senza vincoli etici o di privacy, ha anche creato un ambiente artificialmente controllato che potrebbe non riflettere la complessità del mondo reale. I dati clinici reali sono caratterizzati da valori mancanti, errori di misurazione e bias sistematici (errori ripetuti e non casuali che alterano l'accuratezza dei risultati) che i nostri dataset puliti non catturano. Inoltre, le correlazioni che abbiamo programmato nei nostri generatori di dati potrebbero non riflettere accuratamente le relazioni complesse che esistono nella medicina reale.

Un'altra sfida importante è rappresentata dalla scalabilità computazionale. I nostri test sono stati limitati dalla capacità di calcolo disponibile, impedendoci di condurre simulazioni su dataset veramente grandi. Questo limite ha ristretto la significatività statistica delle nostre conclusioni e potrebbe aver nascosto pattern che emergerebbero solo con campioni più ampi.

Il problema dell'overfitting rappresenta un'altra preoccupazione fondamentale. Poiché abbiamo generato sia i dati di training che quelli di test utilizzando lo stesso processo stocastico, ma potrebbe esserci il rischio che i nostri modelli abbiano memorizzato le peculiarità del nostro generatore di dati piuttosto che principi generalizzabili. Questo problema è particolarmente rilevante per applicazioni mediche, dove l'overfitting può portare a sistemi che funzionano bene in laboratorio ma falliscono quando confrontati con la variabilità del mondo reale.

La semplicità della nostra malattia fittizia, pur essendo stata utile per l'analisi, rappresenta anche una limitazione. Le malattie reali sono spesso caratterizzate da sintomi sovrapposti e presentazioni atipiche che complicano significativamente il processo diagnostico. Il nostro modello binario (malato/sano) non cattura la ricchezza di scenari che un sistema diagnostico reale dovrebbe gestire.

Dal punto di vista dell'interfaccia utente, il nostro sistema testuale è funzionale ma lontano da ciò che sarebbe necessario per un'applicazione clinica reale.

Un sistema diagnostico pratico richiederebbe un'interfaccia grafica intuitiva, integrazione con sistemi informativi ospedalieri, e meccanismi robusti per la gestione dell'identità dei pazienti e la privacy dei dati.

Implicazioni per la Medicina Digitale

Nonostante i suoi limiti, il nostro progetto offre spunti per il futuro della medicina digitale e dei sistemi di supporto alle decisioni cliniche.

I sistemi esperti, con la loro trasparenza rimangono preziosi in contesti dove è fondamentale comprendere il ragionamento che porta a una diagnosi. D'altra parte, le reti bayesiane offrono una modellazione più sofisticata dell'incertezza che caratterizza la medicina.

La loro capacità di fornire non solo una diagnosi ma anche una misura di confidenza può essere estremamente preziosa per i medici. Inoltre, la possibilità di aggiornare continuamente i parametri del modello basandosi su nuovi dati clinici rende questi sistemi potenzialmente auto-miglioranti.

Un'idea per future ricerche potrebbe riguardare lo sviluppo di sistemi ibridi che combinino la spiegabilità dei sistemi esperti con la robustezza probabilistica delle reti bayesiane. Questo approccio potrebbe sfruttare i punti di forza di entrambi i metodi e limiterebbe i rispettivi punti deboli.

Considerazioni e Conclusioni

Il percorso di sviluppo di questo progetto, ci ha fatto crescere professionalmente, dandoci la possibilità di approcciare le nostre conoscenze informatiche con un mondo a noi apparentemente lontano come quello della medicina. Abbiamo appreso che anche se le considerazioni teoriche possono suggerire la superiorità di un approccio rispetto a un altro, spesso è solo attraverso test sistematici su dati realistici che emergono le vere differenze di prestazione.

Attraverso l'implementazione e il confronto di due approcci fondamentalmente diversi alla diagnosi automatica, abbiamo esplorato alcune delle tensioni centrali che caratterizzano l'IA applicata alla medicina.

I risultati del nostro lavoro ci dicono che non esiste una soluzione universale al problema della diagnosi automatica. Piuttosto, diversi approcci possono essere appropriati per diversi contesti clinici, diversi tipi di malattie e diverse priorità diagnostiche. Ogni algoritmo testato, ogni metrica valutata, ogni limitazione identificata aggiunge un tassello al grande puzzle della conoscenza.

Questo progetto rappresenta come l'intelligenza artificiale possa affiancare i medici nelle loro ricerche, diagnosi e cure.