

Report Progetto Intelligenza Artificiale

Smart Home Energy Saver

Un Agente Razionale basato su Reti Bayesiane

Andrea Filippo Salemi
Matr: 1000056544

Gennaio 2026

1 Scelta Implementativa

Per affrontare il problema della gestione energetica intelligente in un contesto di domotica (*Smart Home*), caratterizzato intrinsecamente da incertezza sensoriale e variabilità ambientale, ho deciso di implementare un **Agente Razionale** basato su **Reti Bayesiane**.

Questa scelta è motivata dalla necessità di modellare relazioni di causa-effetto non deterministiche (ad esempio l'impatto della stagione sul meteo o l'affidabilità di un sensore), un compito per il quale gli approcci classici basati su logica booleana o regole rigide (**IF--THEN**) risultano spesso inadeguati o troppo fragili.

Le Reti Bayesiane, al contrario, offrono un framework matematico rigoroso per la gestione di informazioni parziali e rumorose, permettendo al sistema di aggiornare le proprie credenze in tempo reale all'arrivo di nuove evidenze.

L'implementazione è stata realizzata in linguaggio **Python**, utilizzando la libreria **pgmpy** per la definizione della struttura topologica e l'esecuzione dell'inferenza, e **numpy** / **matplotlib** per la fase di simulazione e analisi dei dati.

2 Librerie e strumenti usati

L'intero progetto è stato sviluppato in linguaggio Python, sfruttando un ecosistema di librerie open-source specifiche per il calcolo scientifico e la modellazione probabilistica:

- **pgmpy** (Probabilistic Graphical Models using Python): È la libreria core del progetto. È stata utilizzata per definire la struttura della Rete Bayesiana, specificare le Tabelle di Probabilità Condizionata (CPT) e soprattutto per eseguire i calcoli di inferenza esatta (tramite l'algoritmo di Variable Elimination) e il campionamento dei dati.

- *Networkx*: Utilizzata per la gestione e la manipolazione della struttura topologica del Grafo Aciclico Diretto (DAG) sottostante la rete.
- *Matplotlib*: Impiegata per la visualizzazione grafica della rete e per la generazione dei grafici di analisi (istogrammi e heatmap) necessari alla validazione dei risultati della simulazione.

Attraverso questi strumenti, il progetto si realizza un'architettura completa che parte dalla definizione teorica delle dipendenze causali (**Stagione** \rightarrow **Meteo** \rightarrow **Umidità**), passa per la simulazione di scenari operativi tramite metodo Monte Carlo e culmina in un modulo di decisione basato sull'utilità attesa.

3 Modellazione della Rete Bayesiana

Il sistema è stato modellato come una rete bayesiana discreta, definendo esplicitamente sia la struttura topologica (Grafo Aciclico Diretto, DAG) sia i parametri probabilistici tramite Tabelle di Probabilità Condizionata (CPT).

3.1 Topologia della Rete

La struttura del grafo riflette le dipendenze causali dirette del dominio applicativo. Sono stati individuati cinque nodi principali:

- **Stagione (S)**: Nodo radice con quattro stati (Inverno, Primavera, Estate, Autunno). Rappresenta il contesto temporale generale del sistema.
- **Pioggia (P)**: Nodo intermedio booleano (*Si/No*), condizionato dalla variabile Stagione.
- **Previsioni Meteo (M)**: Nodo foglia osservabile, condizionato dalla variabile Pioggia. Funge da *sensore virtuale* affetto da rumore.
- **Umidità Terreno (U)**: Nodo di stato fisico (*Umido/Secco*). È il nodo più complesso del modello, poiché dipende congiuntamente dalla Pioggia (apporto idrico immediato) e dalla Stagione (tasso di evaporazione).
- **Irrigatore (I)**: Nodo decisionale (*Acceso/Spento*), dipendente dallo stato dell'Umidità del terreno.

3.2 Parametrizzazione

Le Tabelle di Probabilità Condizionata sono state definite manualmente seguendo una logica di dominio, al fine di simulare in modo realistico le dinamiche ambientali.

- **Distribuzione Stagionale**: È stata assunta una distribuzione uniforme a priori, con

$$P(S) = 0.25$$

per ciascuna stagione.

- **Modello Pluviometrico:** La probabilità di pioggia varia significativamente in base alla stagione:
 - Elevata in Inverno: $P(P = \text{Sì} \mid S = \text{Inverno}) = 0.80$
 - Minima in Estate: $P(P = \text{Sì} \mid S = \text{Estate}) = 0.15$
- **Affidabilità del Sensore (Previsioni Meteo):** La CPT delle previsioni modella l'errore strumentale del sensore:
 - *True Positive Rate*: se piove, le previsioni indicano *Pioggia* con probabilità pari al 90%.
 - *False Positive Rate*: in assenza di pioggia, il sensore segnala erroneamente *Pioggia* nel 20% dei casi.
- **Dinamica dell'Umidità (Interazione):** La CPT dell'Umidità gestisce l'interazione tra pioggia ed evaporazione:
 - **Caso Pioggia:** se $P = \text{Sì}$, il terreno risulta umido con probabilità del 99%, indipendentemente dalla stagione.
 - **Caso No Pioggia:** l'umidità residua dipende fortemente dalla stagione. In Inverno, anche in assenza di pioggia, il terreno rimane umido con probabilità pari al 60%; in Estate, senza pioggia, il terreno si secca quasi certamente, con

$$P(U = \text{Secco} \mid P = \text{No}, S = \text{Estate}) = 0.98.$$

- **Policy dell'Irrigatore:** L'attivazione dell'irrigatore è stocastica ma fortemente correlata allo stato di siccità: se il terreno è Secco, l'irrigatore si attiva con probabilità pari al 90%.

3.3 Verifica Strutturale

La consistenza del modello è stata verificata tramite il metodo `bn_model.check_model()`, che ha confermato:

- l'assenza di cicli nel grafo;
- la corretta normalizzazione delle CPT, con $\sum P = 1$ per ogni riga.

3.4 Visualizzazione Topologica e Semantica

Per facilitare l'ispezione della struttura del modello, è stata implementata una visualizzazione grafica personalizzata mediante le librerie `networkx` e `matplotlib`. A differenza del layout automatico standard, è stato definito un layout gerarchico posizionale che riflette il flusso causale dell'informazione:

- **Livello Superiore (Cause Latenti):** nodi radice come *Stagione* e *Pioggia*, posizionati al vertice del grafo.

- **Livello Intermedio (Stati Fisici e Sensori):** nodi come `Umidita_Terreno` e `Previsioni_Meteo`.
- **Livello Inferiore (Decisione):** il nodo foglia `Irrigatore`.

Inoltre, è stata adottata una codifica cromatica semantica per distinguere immediatamente il ruolo funzionale di ciascuna variabile nel processo decisionale:

- **Rosso/Verde Acqua (Nodi Radice):** variabili indipendenti o parzialmente latenti che governano il sistema.
- **Azzurro (Osservazione):** variabili che fungono da sensori (es. `Previsioni_Meteo`), soggette a rumore.
- **Giallo (Decisione):** l'azione finale dell'agente.

La visualizzazione risultante (Figura 1) conferma visivamente l'assenza di cicli (DAG) e la corretta direzione degli archi di dipendenza probabilistica.

3.5 Implementazione della Visualizzazione Grafica

Per garantire un'interpretabilità immediata della struttura topologica, è stata implementata una routine di visualizzazione personalizzata utilizzando le librerie `networkx` e `matplotlib`. Invece di affidarsi a layout automatici (che spesso risultano disordinati), è stato definito un *layout posizionale statico* (`pos`) che organizza i nodi gerarchicamente:

- **Livello Superiore ($y = 2$):** Le cause radice (`Stagione`, `Pioggia`) e il sensore (`Previsioni_Meteo`).
- **Livello Intermedio ($y = 1$):** Lo stato fisico latente (`Umidita_Terreno`).
- **Livello Inferiore ($y = 0$):** La variabile decisionale (`Irrigatore`).

È stata inoltre applicata una **codifica cromatica semantica** per distinguere il ruolo funzionale di ogni nodo:

- **Rosso/Verde Acqua:** Variabili Radice (Cause scatenanti)
- **Azzurro:** Variabili Osservabili (Sensori virtuali)
- **Verde Chiaro:** Stati intermedi
- **Giallo:** Variabili di Azione (Decisioni)

Infine, per migliorare la leggibilità del DAG:

- Gli archi sono stati renderizzati *curvi* (`connectionstyle='arc3'`) per evitare sovrapposizioni visive.
- È stata aggiunta una **legenda esplicativa** per facilitare la lettura del grafico da parte di utenti non tecnici.

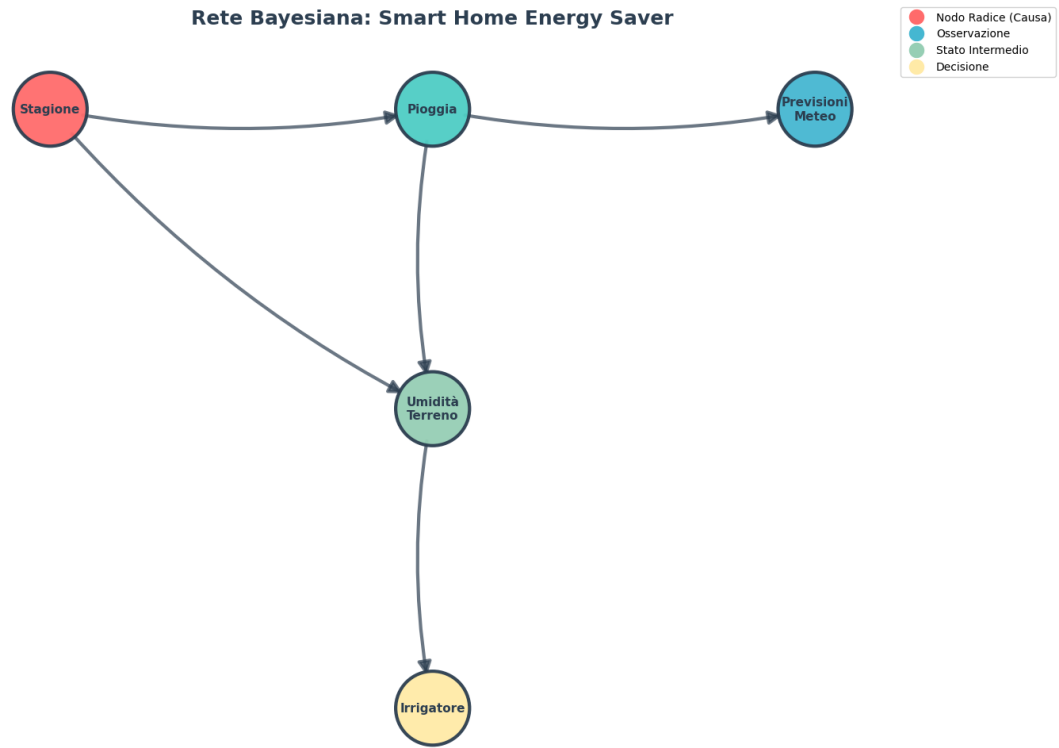


Figure 1: Visualizzazione topologica e semantica della rete bayesiana. I colori dei nodi rappresentano il ruolo funzionale: rosso/verde acqua (cause latenti), azzurro (sensori), giallo (decisione).

4 Tipi di Ragionamento nelle Reti Bayesiane

Oltre al calcolo delle probabilità marginali, la Rete Bayesiana implementata è stata validata testando la sua capacità di emulare tre specifici meccanismi di ragionamento umano. Tali meccanismi sfruttano la flessibilità del flusso informativo all'interno del grafo aciclico diretto (DAG), permettendo inferenze sia lungo che contro la direzione causale.

4.1 Predizione Causale (Ragionamento Top-Down)

Questo pattern segue la direzione degli archi del grafo, propagando l'informazione dalle *cause* verso gli *effetti*. Viene utilizzato per stimare lo stato futuro o non osservabile del sistema a partire da premesse certe.

- **Logica:** $P(\text{Effetto} \mid \text{Causa})$

- **Applicazione nel progetto:** la rete è stata interrogata ponendo evidenza sul nodo radice **Stagione**.
- **Risultato:** l'agente ha predetto correttamente che la probabilità di attivazione dell'**Irrigatore** (effetto finale) varia drasticamente in base alla stagione:
 - **Inverno:** probabilità di attivazione $\approx 17\%$, indicativa di una bassa necessità idrica.
 - **Estate:** probabilità di attivazione $\approx 77\%$, coerente con l'elevata evaporazione e la scarsità di piogge.

Questo risultato conferma che il modello propaga correttamente l'influenza causale attraverso i nodi intermedi (**Pioggia** e **Umidità_Terreno**).

4.2 Diagnosi Evidenziale (Ragionamento Bottom-Up)

Questo pattern inverte la direzione degli archi del grafo, risalendo dagli effetti osservati alle cause più probabili. È un meccanismo fondamentale sia per l'interpretazione dello stato dell'ambiente a partire dai sensori, sia per il *debug* del sistema.

- **Logica:** $P(\text{Causa} \mid \text{Effetto})$
- **Applicazione nel progetto:** è stato osservato lo stato di nodi foglia e sensori per inferire lo stato dei nodi radice.
- **Scenario testato:** osservando l'evidenza **Previsioni_Meteo = Pioggia** (effetto), il sistema aggiorna la credenza sulla variabile latente **Pioggia** (causa).

Risultato: nonostante le previsioni non siano perfette (a causa del rumore modellato nella CPT del sensore), l'osservazione dell'effetto incrementa significativamente la probabilità a posteriori che stia effettivamente piovendo. Il sistema esegue quindi una diagnosi corretta dello stato reale dell'ambiente basandosi su un sintomo osservabile, dimostrando la robustezza del meccanismo inferenziale.

4.3 Analisi del Ragionamento Intercausale (Explaining Away)

Una delle proprietà più sofisticate delle Reti Bayesiane è il fenomeno dell'*Explaining Away* (o ragionamento intercausale), in cui due cause competono per spiegare un singolo effetto osservato. Quando una delle cause viene confermata, la probabilità dell'altra viene modificata, anche se le due cause sono indipendenti a priori.

Per dimostrare questo comportamento, il codice esegue una sequenza di tre query progressive:

- **Stato a priori:** calcolo della probabilità marginale di **Pioggia** in assenza di evidenze.

- **Osservazione dell'effetto:** introduzione dell'evidenza `Umidita_Terreno = Umido`.
- **Introduzione del contesto (seconda causa):** fissata l'umidità, viene introdotta l'evidenza `Stagione` per osservare come cambia la credenza sulla `Pioggia`.

4.3.1 Risultati dell'Inferenza

I risultati numerici ottenuti (riportati nelle tabelle seguenti) mostrano l'evoluzione della credenza dell'agente.

| Step | Evidenza Inserita | $P(\text{Pioggia} = \text{Sì})$ | Note |
|-------------|-------------------|---------------------------------|------|
| 1. Priori | Nessuna | 55.00% | (*) |
| 2. Diagnosi | Umidità=Umido | 83.13% | (**) |

Table 1: Evoluzione della probabilità di pioggia al variare delle evidenze.

(*) : Media ponderata su tutte le stagioni

(**) : L'effetto osservato rende la causa molto probabile

Successivamente, fissando l'evidenza `Umidita_Terreno = Umido`, è stata analizzata la probabilità di `Pioggia` condizionata alla `Stagione`. È in questo passaggio che emerge chiaramente il fenomeno dell'*Explaining Away*.

| Stagione (Concausa) | $P(\text{Pioggia} = \text{Sì})$ | Interpretazione del Ragionamento |
|---------------------|---------------------------------|-------------------------------------|
| Estate | 89.73% | l'umidità estiva solo dalla pioggia |
| Inverno | 86.84% | Ridotta: la bassa evaporazione |
| Autunno | 85.97% | Intermedia |
| Primavera | 74.81% | Intermedia |

Table 2: Explaining Away: probabilità di pioggia condizionata a umidità e stagione.

4.3.2 Analisi Grafica

Il grafico a barre generato dal codice (`plt.bar`) visualizza in modo immediato la discrepanza tra le diverse stagioni.

Commento alla figura: si osserva come la barra relativa all'**Estate** risulti la più elevata (circa 90%). Questo indica che l'agente ha appreso che l'umidità in estate è un evento raro che richiede una spiegazione causale forte (la pioggia). Al contrario, l'umidità in inverno è più ambigua, potendo essere causata anche dalla ridotta evaporazione, e ciò riduce leggermente la necessità di attribuirle all'evento piovoso.

Nel complesso, questo comportamento conferma che la rete non applica regole rigide, ma pesa dinamicamente le cause in funzione del contesto, emulando un ragionamento probabilistico complesso e realistico.

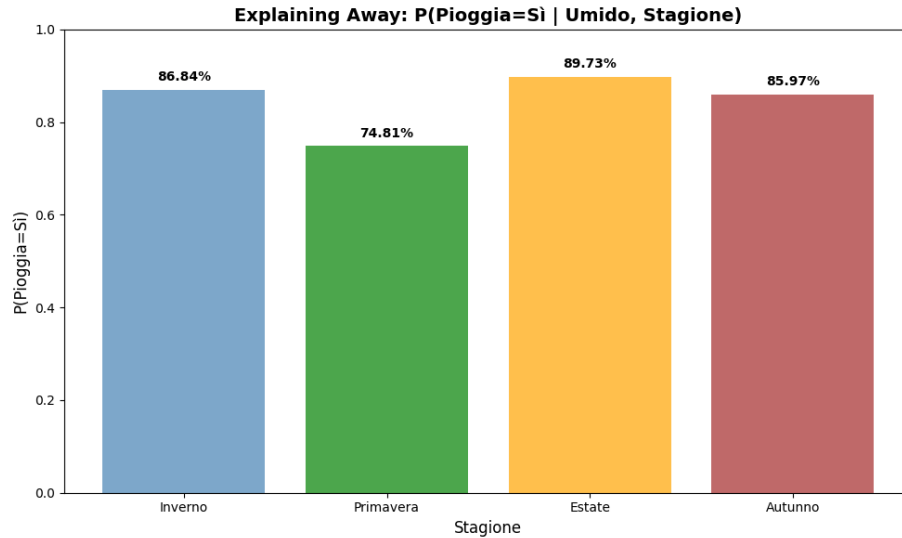


Figure 2: Probabilità di pioggia condizionata a umidità e stagione.

Interpretazione del grafico:

- INVERNO (stagione umida): $P(\text{Pioggia}) \text{ BASSA} \rightarrow$ l'Inverno spiega l'umidità;
- ESTATE (stagione secca): $P(\text{Pioggia}) \text{ ALTA} \rightarrow$ solo la Pioggia spiega l'umidità;
- Questo è "Explaining Away": una causa spiega via l'effetto, riducendo l'altra.

4.4 Risultati Numerici

Per validare la robustezza del modello teorico, è stata eseguita una simulazione stocastica massiva.

4.4.1 Validazione Monte Carlo

Utilizzando il metodo di *Forward Sampling*, è stato generato un dataset sintetico di 10.000 istanze (campionando sequenzialmente dai nodi genitori ai figli). L'analisi statistica dei dati generati (visualizzata nei grafici del notebook) ha evidenziato:

- **Convergenza Empirica:** Le frequenze relative delle variabili nel dataset (ad esempio, percentuale di giorni di pioggia in Inverno vs Estate) coincidono con le probabilità teoriche definite nelle CPT, confermando la correttezza dell'implementazione.
- **Analisi delle Correlazioni:** La *heatmap* delle correlazioni mostra:
 - Forte correlazione positiva tra Meteo Reale e Previsioni ($\rho \approx 0.75$), indicando che il sensore virtuale è affidabile ma non deterministico.

- Correlazione negativa tra Pioggia e Irrigatore, confermando che il sistema agisce correttamente evitando sprechi idrici quando piove.

4.4.2 Modulo Decisionale (Utility Theory)

L'agente non si limita a stimare lo stato del mondo, ma prende decisioni razionali basate sulla **Teoria dell'Utilità**. È stata definita la seguente funzione di costo:

- $C_{azione} = 5 \text{ €}$: costo fisso di attivazione dell'irrigatore
- $C_{danno} = 50 \text{ €}$: costo stimato per la perdita delle piante se il terreno resta secco

La decisione finale è governata dalla massimizzazione dell'*Utilità Attesa* (MEU). L'agente attiva l'irrigatore se e solo se:

$$P(\text{Secco} \mid \text{Evidenze}) \cdot C_{danno} > C_{azione}$$

Dai test effettuati emerge un comportamento conservativo: l'agente attiva l'irrigatore anche in presenza di probabilità di siccità medio-basse (ad esempio, superiori al 10%), poiché il costo del danno potenziale supera di gran lunga il costo dell'acqua. Questo comportamento simula efficacemente una strategia umana prudente ed economicamente vantaggiosa.

5 Esecuzione dell'Inferenza e Analisi dei Risultati

Per verificare il comportamento dell'agente, è stato istanziato il motore di inferenza esatta **VariableElimination**. Sono state eseguite diverse query interrogando la distribuzione a posteriori $P(\text{Target} \mid \text{Evidence})$ per simulare scenari operativi reali.

Di seguito si riporta l'analisi dei risultati ottenuti (i valori numerici fanno riferimento all'output diretto del codice Python):

5.1 Scenario Base e Stagionalità

In assenza di evidenze (, il sistema mostra un comportamento conservativo con una probabilità di attivazione dell'irrigatore del 37.6%. Introducendo l'evidenza **Stagione**, il modello aggiorna le credenze in modo coerente con la climatologia:

- **Inverno**: La probabilità di irrigazione crolla al 17.0%.
- **Estate**: La probabilità sale drasticamente al 76.8%, riflettendo l'aumento dell'evaporazione e la scarsità di piogge.

5.2 Gestione del Conflitto e Sensori

È stata testata la reazione del sistema a input sensoriali specifici:

- **Inibizione da Pioggia** : Se si osserva $Pioggia = Sì$, l'agente riduce la probabilità di attivazione al 10.8%, prevenendo sprechi idrici. Il valore non è zero poiché il modello ammette una minima probabilità che il terreno non abbia assorbito l'acqua o che il sensore sia rumoroso.
- **Rinforzo dell'Evidenza** : Combinando $Stagione = Estate$ con $Previsioni = Sole$, la probabilità di irrigare sale all'86.7%. Si noti come l'aggiunta delle previsioni (rispetto al 76.8% della sola Estate) aumenti la confidenza del sistema nella necessità di agire.

5.3 Dominanza Causale

L'ultima query verifica la robustezza della CPT dell'Umidità. Interrogando $P(Umidità | Stagione, Pioggia = Sì)$, si osserva che per tutte le stagioni la probabilità che il terreno sia **Umido** si attesta al 99.0%. Questo risultato conferma che la variabile **Pioggia** agisce come causa sufficiente dominante: se piove, la stagione diventa ininfluyente per lo stato di umidità del terreno (fenomeno di *context-specific independence*).

5.4 Tabella Riassuntiva Probabilità di Irrigazione

Di seguito viene riportata la tabella comparativa delle probabilità di attivazione dell'irrigatore ($P(I = \text{Acceso})$) al variare delle evidenze stagionali e meteorologiche:

| Scenario (Evidenza) | Inverno | Primavera | Estate | Autunno |
|-----------------------|---------|-----------|--------|---------|
| Solo Stagione | 17.0% | 26.5% | 76.8% | 30.1% |
| Stagione + Prev. Sole | 31.6% | 43.8% | 86.7% | 55.6% |

Table 3: Confronto delle probabilità di attivazione dell'irrigatore al variare delle evidenze stagionali e meteorologiche.

6 Validazione Sperimentale e Analisi del Dataset

Per verificare la consistenza del modello e la stabilità delle distribuzioni di probabilità, è stata effettuata una simulazione massiva utilizzando il metodo **Monte Carlo**. Tramite l'algoritmo di *Forward Sampling*, è stato generato un **dataset sintetico di 10.000 istanze**, campionando sequenzialmente dalla Rete Bayesiana.

Questo approccio consente di osservare il comportamento asintotico del sistema e di validare empiricamente le *Conditional Probability Tables* (CPT) definite in fase di progettazione.

6.1 Analisi Statistica Descrittiva

Dall'analisi del dataset simulato emergono le statistiche globali riportate in Tabella 4, che descrivono la distribuzione congiunta tra le variabili **Pioggia** e **Umidità_Terreno**.

Table 4: Matrice di Co-occorrenza tra Pioggia e Umidità del Terreno (valori normalizzati)

| Stato Meteo / Terreno | Secco | Umido | Totale Marginale |
|-----------------------|---------------|---------------|------------------|
| Pioggia = No | 33.85% | 10.66% | <i>44.51%</i> |
| Pioggia = Sì | 0.55% | 54.94% | <i>55.49%</i> |
| Totale | <i>34.40%</i> | <i>65.60%</i> | <i>100%</i> |

Interpretazione dei Risultati

- Quando **piove**, il terreno risulta quasi sempre **umido** (54.94% dei casi totali). L'occorrenza di terreno secco in presenza di pioggia (0.55%) rappresenta una condizione limite prevista dalla CPT.
- In assenza di pioggia, il terreno tende prevalentemente a seccarsi (33.85%), ma rimane una quota significativa di umidità residua (10.66%), dovuta principalmente alle stagioni fredde.

6.2 Interpretazione Grafica della Simulazione

I grafici generati dal codice Python permettono una lettura immediata delle dinamiche del sistema. La Figura 3 raccoglie i principali output grafici prodotti durante la fase di validazione.

Analisi dei Grafici

1. **Validazione del Campionamento (Stagione):** La distribuzione empirica replica fedelmente la distribuzione teorica uniforme, confermando la correttezza del processo di campionamento.
2. **Dipendenza Condizionata (Pioggia | Stagione):** La probabilità di pioggia risulta elevata in Inverno e Autunno, mentre si riduce drasticamente in Estate, validando il modello climatico.
3. **Distribuzione dell'Irrigatore:** L'irrigatore è attivo nel 37.36% dei casi, dimostrando un comportamento parsimonioso ed energeticamente efficiente.
4. **Stato del Terreno:** Il terreno risulta umido in circa il 65% delle istanze, riducendo la necessità di intervento artificiale.

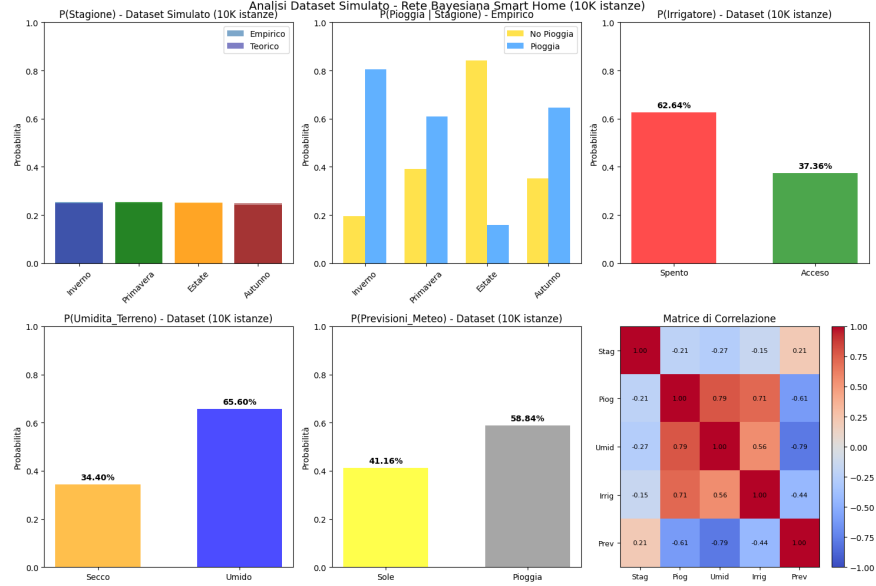


Figure 3: Visualizzazione dei risultati della simulazione Monte Carlo: distribuzioni marginali, condizionate e correlazioni tra variabili

5. **Incerteza Sensoriale (Previsioni):** La discrepanza tra previsioni e pioggia reale evidenzia il rumore introdotto dai sensori, modellando falsi positivi e negativi.
6. **Analisi delle Correlazioni:** La heatmap mostra una forte correlazione positiva tra pioggia e umidità, e una correlazione negativa tra pioggia e attivazione dell'irrigatore.

6.3 Formalizzazione della Logica Decisionale

L'attivazione dell'irrigatore segue un criterio di **teoria delle decisioni**, basato sul confronto tra il costo atteso del non intervento e il costo certo dell'azione.

Il costo atteso della mancata irrigazione è definito come:

$$\mathbb{E}[C_{\text{danno}}] = P(\text{Terreno} = \text{Secco}) \cdot C_{\text{piante}} \quad (1)$$

dove:

- $C_{\text{piante}} = 50 \text{ €}$ è il costo stimato della perdita delle colture;

L'irrigatore viene attivato se:

$$P(\text{Terreno} = \text{Secco}) \cdot 50 > 5 \quad (2)$$

con un costo dell'acqua fissato a 5 €.

Ne consegue una soglia decisionale pari a:

$$P(\text{Terreno} = \text{Secco}) > 0.10 \quad (3)$$

Questo comportamento implementa una strategia *risk-averse*, garantendo la protezione del sistema vegetale anche in presenza di incertezza.

7 Conclusioni

Il progetto ha dimostrato l'efficacia delle *Reti Bayesiane* nella modellazione di sistemi di supporto alle decisioni in ambienti caratterizzati da incertezza. A differenza degli approcci deterministici basati su regole *IF-THEN*, l'agente sviluppato è in grado di:

- **Gestire l'incertezza:** integrare informazioni rumorose (ad esempio previsioni meteorologiche imperfette) per stimare lo stato reale del sistema;
- **Ragionare in modo bidirezionale:** eseguire sia inferenze causali (dalle cause agli effetti) sia inferenze diagnostiche (dagli effetti alle cause);
- **Adattare la decisione:** utilizzare la Teoria dell'Utilità per bilanciare il costo certo dell'azione con il costo probabilistico associato al rischio.

La validazione tramite simulazione Monte Carlo, basata su un dataset sintetico di 10 000 istanze, ha confermato la coerenza tra il modello teorico e i dati empirici, validando le Tabelle di Probabilità Condizionata (CPT) definite in fase di progettazione.

Nel seguito viene illustrato l'algoritmo logico ad alto livello che governa il ciclo di vita dell'agente decisionale.

Algorithm 1 SmartHome_DecisionAgent

```
1: Inizializzazione
2: Definire il grafo bayesiano  $G = (V, E)$ 
3: Definire l'insieme degli stati  $S = \{\text{Stagione, Meteo, Previsioni, Umidità, Irrigatore}\}$ 
4: for all nodo  $X \in V$  do
5:   Assegnare la CPT  $P(X \mid \text{Parents}(X))$  basata sulla conoscenza di dominio
6: end for

7: Ciclo di Monitoraggio
8: Acquisire le evidenze osservate  $E_{oss} = \{\text{Stagione}_{corrente}, \text{Previsioni}_{correnti}\}$ 
9: if sensori attivi then
10:    $E_{oss} \leftarrow E_{oss} \cup \{\text{Lettture\_Sensori}\}$ 
11: end if

12: Inferenza (Variable Elimination)
13: Definire il target  $T = \{\text{Umidità\_Terreno}\}$ 
14: Calcolare  $P(T \mid E_{oss})$ 
15:  $P_{secco} \leftarrow P(\text{Umidità\_Terreno} = \text{Secco} \mid E_{oss})$ 

16: Decisione (Utility Theory)
17:  $C_{azione} \leftarrow 5.0$ 
18:  $C_{danno} \leftarrow 50.0$ 
19:  $Rischio_{atteso} \leftarrow P_{secco} \cdot C_{danno}$ 
20: if  $Rischio_{atteso} > C_{azione}$  then
21:   Stato_Irrigatore  $\leftarrow$  ACCESO
22:   Log(“Irrigazione avviata per rischio siccità”)
23: else
24:   Stato_Irrigatore  $\leftarrow$  SPENTO
25:   Log(“Irrigazione non necessaria”)
26: end if
```
