**Sistema Semantico per Monitoraggio Consumo Energetico**

Gruppo di lavoro

* Papeo Raffaele, Mat 758421, [r.papeo4@studenti.uniba.it](mailto:r.papeo4@studenti.uniba.it)

Github

<https://github.com/raffaelepapeo1/Sistema-Semantico-per-Monitoraggio-Consumo-Energetico.git>

AA 2023-2024

Indice

**Introduzione ……………………………………………………………………………………………………………………………3**

**Sommario ……………………………………………………………………………………………………………………………….3**

**Elenco Argomenti di interesse …………………………………………………………………………………………………4**

**Argomento 1: Rappresentazione della conoscenza ………………………………………………………………….6**

**Argomento 2: Ragionamento automatico su KB ………………………………………………………………………8**

**Argomento 3: Costruzione automatica di dataset da KB (ML + OntoBK) …………………………………..10**

**Argomento 4: Apprendimento supervisionato e modelli kernel based …………………………………….11**

**Argomento 5: Valutazione predittiva e interpretabilità …………………………………………………………..12**

**Conclusioni ……………………………………………………………………………………………………………………………..16**

**Riferimenti bibliografici …………………………………………………………………………………………………………..17**

# Introduzione

Il progetto si colloca all'interno del contesto delle smart home, un ambito in rapida espansione in cui l'efficienza energetica gioca un ruolo cruciale. La diffusione di dispositivi ad alto consumo e la crescente necessità di risparmio energetico pongono nuove sfide per la gestione consapevole delle risorse. In questo scenario, la capacità di identificare automaticamente situazioni di consumo eccessivo, specialmente in fasce orarie critiche, rappresenta un valore aggiunto significativo.

Il presente lavoro propone la realizzazione di un sistema intelligente per il monitoraggio del consumo energetico domestico, integrando modelli di rappresentazione della conoscenza, ragionamento automatico e apprendimento supervisionato. Il sistema non si limita ad acquisire dati grezzi, ma li interpreta attraverso una base di conoscenza ontologica (OWL), applica inferenze logiche per identificare eventi critici, e successivamente sfrutta queste etichette inferite per addestrare modelli predittivi.

Questo approccio consente di realizzare un ciclo virtuoso in cui la conoscenza simbolica guida l'apprendimento statistico, permettendo al sistema di generalizzare su nuovi eventi non esplicitamente descritti nella base di conoscenza. L'obiettivo è quello di fornire un esempio concreto di sistema ibrido, conforme agli obiettivi didattici del corso di Ingegneria della Conoscenza, in particolare nei temi dell'integrazione tra ontologie, ragionamento e apprendimento automatico.

# Sommario

Il sistema realizzato è un **Knowledge-Based System (KBS)** progettato per integrare in modo coerente e sinergico i principali ambiti dell’Ingegneria della Conoscenza: la rappresentazione formale del sapere, il ragionamento automatico e l’apprendimento statistico da dati. L’obiettivo è quello di costruire un processo intelligente capace di interpretare e classificare eventi di consumo energetico all’interno di un ambiente domestico, a partire dalla sola definizione del dominio.

La **rappresentazione della conoscenza** avviene attraverso un'ontologia OWL DL che formalizza concetti e relazioni del dominio (dispositivi, stanze, fasce orarie, consumi). Le regole di comportamento sono definite in forma dichiarativa e inferibili tramite un reasoner standard (Hermit), affrontando quindi i temi del ragionamento monotono, della consistenza ontologica e delle clausole di Horn.

Il modulo di **ragionamento automatico** consente di classificare eventi in base alla loro criticità energetica, derivando nuove conoscenze dalla struttura semantica della KB. Tali inferenze sono poi utilizzate per generare automaticamente un dataset supervisionato, dimostrando un'applicazione concreta dell'approccio **ML+OntoBK**, ovvero l’apprendimento guidato da background knowledge.

Infine, il modulo di **apprendimento automatico supervisionato** sfrutta il dataset così costruito per addestrare modelli predittivi (Random Forest, SVM, Logistic Regression) che generalizzano la classificazione anche su eventi non previsti esplicitamente dalla KB. Le tecniche di valutazione adottate (cross-validation, F1, AUC, matrice di confusione) permettono un’analisi accurata dell’affidabilità e delle prestazioni del sistema.

In questo modo, il progetto copre e collega diversi argomenti centrali del programma del corso ICON:

* **OWL DL e rappresentazione formale della conoscenza**;
* **Inferenza automatica e reasoning su KB simboliche**;
* **Apprendimento statistico integrato con semantica (ML+OntoBK)**;
* **Valutazione predittiva e tecniche di classificazione supervisionata**;
* **Costruzione automatica di dataset tramite ragionamento**.

# Elenco argomenti di interesse

Nel corso del progetto sono stati affrontati e integrati diversi argomenti appartenenti a sezioni distinte del programma ufficiale del corso. In particolare:

1. **Rappresentazione della conoscenza – OWL DL e logica descrittiva**(Sezione: Rappresentazione simbolica, capitoli 5-6 del testo)  
   Il dominio del consumo energetico è stato modellato tramite un’ontologia OWL DL, impiegando costrutti logici standard (e.g. intersectionOf, someValuesFrom, equivalentClass) per rappresentare relazioni tra dispositivi, stanze, fasce orarie e tipi di consumo. La scelta della logica descrittiva consente inferenze consistenti e decidibili, garantendo una rappresentazione formalmente corretta e facilmente estendibile.
2. **Ragionamento automatico su KB**(Sezione: Reasoning monotono, semantica formale – capitolo 7)  
   La componente di ragionamento utilizza il reasoner Hermit per inferire automaticamente nuove conoscenze a partire dalle assunzioni dichiarative nella KB. In particolare, viene effettuata la classificazione automatica di eventi di consumo come critici (classe EventoDiConsumoElevato), sulla base delle proprietà semantiche degli individui e delle restrizioni concettuali definite. Il reasoning è monotono e coerente con la semantica formale di OWL DL.
3. **Costruzione automatica di dataset da KB (ML+OntoBK)**(Sezione: Ontologie per il machine learning, cap. 14-16)  
   Il progetto adotta un approccio ML+OntoBK (Machine Learning con Background Knowledge), in cui le etichette supervisionate non sono fornite manualmente ma dedotte automaticamente dalla KB tramite ragionamento. Gli individui dell’ontologia, arricchiti da proprietà e relazioni, sono trasformati in vettori di feature e target attraverso query e ragionamento. Questo consente di costruire dataset supervisionati consistenti con la conoscenza simbolica.
4. **Apprendimento supervisionato e modelli kernel-based**(Sezione: Machine learning, rappresentazione e incertezza – cap. 17)  
   Sfruttando i dati generati dalla KB, sono stati implementati modelli di apprendimento supervisionato con particolare attenzione a quelli basati su kernel, come SVM con kernel RBF, confrontati con Random Forest e Logistic Regression. I modelli sono stati addestrati e validati tramite pipeline con codifica OneHot e validazione incrociata stratificata. L’integrazione con la conoscenza simbolica ha permesso di migliorare la coerenza e l’accuratezza delle previsioni.
5. **Valutazione predittiva e interpretabilità**(Sezione: Metriche di valutazione, spiegabilità – cap. 15)  
   L’efficacia del sistema è stata valutata tramite metriche standard (accuracy, F1-score, precision, recall, AUC), oltre che mediante visualizzazioni come curve ROC e matrici di confusione. Inoltre, l’analisi della feature importance ha permesso di verificare che le variabili più influenti per la classificazione (es. tipo di dispositivo, fascia oraria) corrispondano alle regole implicite nella KB, migliorando l’interpretabilità del comportamento predittivo del sistema.

# Argomento 1: Rappresentazione della conoscenza

## Sommario

## La rappresentazione della conoscenza costituisce il nucleo semantico del sistema. Il dominio è stato modellato tramite un’ontologia OWL DL (Web Ontology Language – Description Logic), che permette di descrivere concetti, relazioni e vincoli logici in modo formale e inferibile. L’ontologia definisce classi fondamentali come Dispositivo, Stanza, FasciaOraria, TipoConsumo ed Evento, e ne descrive le relazioni attraverso proprietà oggetto (e.g. siTrovaIn, haTipoConsumo, avvieneInFascia).

## Una regola chiave modellata nell’ontologia è la seguente: “Un evento associato a un dispositivo ad alto consumo, localizzato in qualsiasi stanza e attivato nella fascia notturna, è classificabile come EventoDiConsumoElevato”. Questa logica è espressa mediante costrutti OWL come intersectionOf, someValuesFrom e equivalentClass, mantenendo la compatibilità con i reasoner OWL DL.

## La KB così costruita rappresenta la conoscenza di background (BK) che guida il ragionamento automatico e la generazione del dataset per l’apprendimento. Il ragionamento, infatti, viene eseguito sulla base di questa struttura ontologica, e tutte le istanze vengono classificate automaticamente secondo la semantica formale specificata.

## Strumenti utilizzati

## **Protégé**: utilizzato per progettare graficamente la struttura concettuale dell’ontologia e per l’esportazione in formato OWL standard [2].

## **Owlready2**: libreria Python che consente di caricare, modificare ed eseguire inferenze su ontologie OWL, integrando reasoner come Hermit [1].

## **RDF/XML**: formato utilizzato per salvare e serializzare l’ontologia

La modellazione si è basata su costrutti standard e su strumenti ampiamente documentati, evitando l’uso di logiche estese non DL (es. SWRL) per preservare la compatibilità con l’inferenza standard.

## Decisioni di Progetto

* È stata adottata la logica **OWL DL** per assicurare decidibilità e compatibilità con motori di inferenza.
* Le **regole di classificazione** sono state modellate mediante costrutti come intersectionOf per definire condizioni congiunte.
* La KB è stata mantenuta **modulare**, con classi e proprietà ben distinte, per facilitare l’estensione e la manutenzione.
* La generazione delle istanze è stata automatizzata con 500 eventi sintetici, bilanciando eventi critici e non critici (circa 50/50).
* Non è stato fatto uso di regole procedurali (es. SWRL) per evitare ambiguità semantiche e garantire piena compatibilità con reasoner OWL standard.

## Valutazione

La valutazione della rappresentazione della conoscenza è stata effettuata in termini di:

* **Coerenza ontologica**: il reasoner Hermit ha verificato la consistenza logica dell’intera KB, senza identificare conflitti o ambiguità.
* **Espressività sufficiente**: tutti i vincoli semantici del dominio sono stati rappresentati con costrutti OWL DL, dimostrando che l’ontologia è sufficientemente espressiva per il task.
* **Qualità dell’inferenza**: su 500 eventi generati, 248 sono stati correttamente inferiti come EventoDiConsumoElevato, in perfetta aderenza alle regole.
* **Robustezza computazionale**: il tempo di inferenza totale è risultato inferiore a un secondo su tutti i casi, confermando l’efficienza del modello.

|  |  |  |  |
| --- | --- | --- | --- |
| **Eventi totali** | **Eventi critici inferiti** | **% critici** | **Tempo inferenza** |
| 500 | 248 | 49,6% | < 1 s |

La KB ha quindi svolto un ruolo centrale non solo nella definizione del dominio, ma anche nella generazione automatica e corretta di etichette supervisionate per l’apprendimento, dimostrando la piena efficacia della rappresentazione semantica adottata.

# Argomento 2: Ragionamento automatico su KB

## Sommario

## Il modulo di ragionamento automatico è incaricato di inferire nuova conoscenza a partire dai dati dichiarativi presenti nella KB, secondo regole logiche formalizzate. In particolare, il sistema utilizza un reasoner compatibile con OWL DL (Hermit) per classificare automaticamente eventi come EventoDiConsumoElevato quando vengono soddisfatte specifiche condizioni: ad esempio, l’utilizzo di un dispositivo ad alto consumo in fascia oraria notturna.

## Il ragionamento è monotono e deduttivo, e opera esclusivamente su assiomi OWL, senza l’uso di regole esterne come SWRL. Questo garantisce rigore semantico, decidibilità e compatibilità con gli strumenti standard. Ogni evento è rappresentato da una combinazione di istanze (dispositivo, fascia, stanza), e il reasoner valuta se tali combinazioni soddisfano le condizioni per l’appartenenza alla classe EventoDiConsumoElevato.

## Il modulo di ragionamento rappresenta un punto cruciale del sistema: consente non solo di arricchire automaticamente la base di conoscenza con nuove classificazioni, ma anche di generare etichette per l’addestramento supervisionato, dando vita all’approccio ML+OntoBK.

## Strumenti utilizzati

## **Hermit Reasoner**: utilizzato tramite interfaccia Owlready2 per eseguire inferenze su ontologie OWL DL. Hermit garantisce compatibilità con gli standard e prestazioni elevate su KB di media complessità.

## **Owlready2.reasoning()**: comando Python che avvia il processo di inferenza e aggiorna automaticamente la KB con le nuove classi dedotte.

## Questi strumenti sono sufficienti per supportare un ragionamento logico conforme alle specifiche formali dell'ontologia.

## Decisioni di Progetto

È stato deciso di **limitare la complessità delle regole** a quelle esprimibili direttamente in OWL DL, evitando costrutti SWRL o procedurali che avrebbero compromesso decidibilità e performance.

Il **reasoner Hermit** è stato integrato direttamente nel flusso di lavoro tramite Owlready2, permettendo di automatizzare il processo di inferenza all’interno del notebook.

Il **ragionamento è stato eseguito in batch** su tutte le istanze della KB, con verifica immediata della coerenza semantica.

Le inferenze sono state salvate e utilizzate direttamente come ground truth per il modulo di apprendimento, senza passaggi manuali.

## Valutazione

La valutazione del modulo di ragionamento ha tenuto conto della correttezza semantica delle inferenze, della copertura degli eventi e dei tempi computazionali. I risultati confermano la robustezza del sistema.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Eventi totali** | **Eventi critici inferiti** | **% critici** | **Tempo inferenza** | **Consistenza KB** | **Errori di classificazione** |
| 500 | 248 | 49,6% | < 1 s | Verificata | Nessuno |

* **Coerenza logica**: il reasoner non ha riscontrato inconsistenze nell’ontologia.
* **Determinismo**: l’inferenza è ripetibile e invariabile, come previsto da un modello logico monotono.
* **Scalabilità**: su 500 istanze, il tempo di ragionamento è rimasto trascurabile, indicando la scalabilità del sistema a dataset più ampi.

Il reasoner ha quindi svolto un ruolo centrale non solo nell’arricchimento semantico, ma anche nella generazione affidabile e automatizzata delle etichette per la classificazione supervisionata, dimostrando piena aderenza ai principi del reasoning simbolico studiati nel corso.

# Argomento 3: Costruzione automatica di dataset da KB (ML + OntoBK)

## Sommario

In questo modulo del progetto, è stata affrontata la generazione automatica di dataset supervisionati partendo dalla conoscenza codificata nella base ontologica. Grazie all’inferenza automatica condotta tramite reasoner OWL, è stato possibile etichettare eventi simulati sulla base della loro appartenenza semantica a classi concettuali (es. EventoDiConsumoElevato). Il sistema adotta il paradigma ML+OntoBK (Machine Learning con Background Knowledge), in cui la conoscenza simbolica guida la costruzione del dataset e riduce la necessità di intervento manuale nella fase di etichettatura.

## Strumenti utilizzati

È stata utilizzata la libreria Owlready2 per accedere e interrogare la KB costruita in formato OWL DL. I dati inferiti sono stati raccolti ed esportati in forma strutturata tramite Pandas, mentre la gestione delle variabili categoriali è stata effettuata con il modulo OneHotEncoder di Scikit-learn. Tutta l’operazione è stata gestita in Google Colab, con codice Python eseguibile su ambiente condiviso.

## Decisioni di Progetto

Le feature scelte per rappresentare ciascun evento sono: tipo di dispositivo, fascia oraria e stanza. Queste proprietà, già esplicitate nella KB, sono state selezionate perché ritenute semanticamente rilevanti rispetto all’obiettivo di classificare il consumo come critico o meno. Le etichette di classe (target) sono state derivate interrogando la KB rispetto alla membership nella classe EventoDiConsumoElevato, ottenuta tramite ragionamento. Per garantire robustezza ai modelli predittivi, il dataset è stato arricchito fino a contenere circa 300 esempi, bilanciando i casi positivi e negativi.

## Valutazioni

Il dataset generato ha dimostrato un’elevata coerenza semantica e un buon equilibrio tra le classi. La qualità dell’etichettatura è stata verificata confrontando il comportamento dei classificatori con le regole della KB. I modelli addestrati su questi dati hanno mostrato prestazioni soddisfacenti, confermando che la conoscenza simbolica ha contribuito efficacemente alla creazione di esempi validi e significativi per il task predittivo.

Il dataset generato ha mostrato coerenza con la struttura semantica definita nella KB e ha permesso l’addestramento di modelli con buoni risultati in termini di accuratezza e capacità di generalizzazione. La generazione automatica ha inoltre facilitato l’ampliamento del dataset (fino a 300 esempi), garantendo un campionamento realistico e controllato. L’approccio ML+OntoBK ha dimostrato di essere efficace per produrre dataset etichettati affidabili in scenari in cui la conoscenza simbolica è disponibile. Le inferenze ottenute sono state coerenti con le attese semantiche e hanno fornito etichette accurate per il successivo addestramento dei modelli di machine learning. Il sistema ha mostrato stabilità, correttezza e coerenza semantica, dimostrando l’efficacia dell’approccio simbolico nella generazione di conoscenza inferita.

# Argomento 4: Apprendimento supervisionato e modelli Kernel based

## Sommario

Questa sezione del progetto si è concentrata sull’addestramento di modelli di apprendimento supervisionato, con l’obiettivo di classificare automaticamente eventi di consumo energetico come critici o non critici. I dati etichettati automaticamente dalla KB sono stati utilizzati per creare un dataset supervisionato su cui applicare modelli predittivi. In particolare, sono stati confrontati modelli basati su approcci diversi:

* **Random Forest** (ensemble learning, decision trees)
* **Support Vector Machine** con kernel RBF (classificatore basato su margine massimo)
* **Regressione Logistica** (modello lineare interpretabile)

Le feature utilizzate derivano dalla KB e includono: tipo di dispositivo, stanza in cui si trova, e fascia oraria di attivazione. Poiché si tratta di dati categoriali, è stato necessario adottare tecniche di encoding per la compatibilità con i modelli.

## Strumenti utilizzati

Per l’implementazione dei modelli sono state usate le seguenti librerie:

* **Scikit-learn** per la definizione dei modelli, la validazione, e il calcolo delle metriche.
* **Pandas** e **NumPy** per la gestione dei dati strutturati.
* **OneHotEncoder** per la codifica delle variabili categoriali.
* **StratifiedKFold** per la validazione incrociata mantenendo il bilanciamento delle classi.

Tutti i modelli sono stati eseguiti su Google Colab, sfruttando la possibilità di testare più configurazioni in modo efficiente.

## Decisioni di Progetto

* **Encoding**: è stata scelta la codifica One-Hot per rappresentare variabili categoriali, in quanto è compatibile con tutti i modelli utilizzati, compresi quelli che non gestiscono direttamente input simbolici.
* **Validazione**: per ogni modello è stata effettuata una validazione tramite Stratified K-Fold (k=3), in modo da valutare le prestazioni su più suddivisioni del dataset mantenendo la proporzione delle classi.
* **Parametri**:
  + Random Forest: n\_estimators=100, random\_state=42
  + SVM: kernel='rbf', C=1.0, probability=True
  + Logistic Regression: solver='lbfgs', max\_iter=1000
* I parametri dei modelli sono stati inizialmente mantenuti a valori di default, salvo ottimizzazioni base (es. max\_iter=1000 per Logistic Regression).
* La pipeline è stata costruita con make\_pipeline per garantire modularità e riproducibilità.

## Valutazioni

I risultati ottenuti evidenziano l'efficacia dell’approccio supervisionato nel classificare correttamente gli eventi:

**Metriche Aggregate**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Modello** | **Accuracy** | **F1-score** | **Precision** | **Recall** | **AUC** |
| Random Forest | 0.972 | 0.921 | 0.950 | 0.834 | 0.980 |
| SVM (RBF) | 0.972 | 0.920 | 0.944 | 0.830 | 0.954 |
| Logistic Regression | 0.936 | 0.822 | 0.887 | 0.720 | 0.910 |

* Le **curve ROC** per Random Forest e SVM mostrano un’area sotto la curva (AUC) superiore a 0.95.
* Le **matrici di confusione** confermano la buona capacità discriminativa, con pochi falsi positivi e negativi.
* L’**analisi della feature importance** ha evidenziato che il tipo di dispositivo e la fascia oraria sono i predittori più influenti.

# Argomento 5: Valutazione Predittiva e Interpretabilità

## Sommario

La valutazione predittiva del sistema è stata effettuata attraverso metriche standard del machine learning supervisionato. Tuttavia, accanto alla prestazione numerica, è stata condotta anche un’analisi interpretativa dei modelli, con l’obiettivo di verificare la coerenza tra il comportamento appreso e la semantica della base di conoscenza (KB). Questo è un passaggio fondamentale per un sistema intelligente che si fonda sull’integrazione tra logica simbolica e apprendimento statistico.

## Strumenti utilizzati

* **Scikit-learn**: per il calcolo di metriche di classificazione come accuracy, precision, recall, F1-score, AUC.
* **Matplotlib**: per la visualizzazione delle curve ROC e delle matrici di confusione.
* **Feature Importance**: calcolata direttamente su Random Forest per interpretare il contributo delle feature.

## Decisioni di Progetto

Sono state considerate diverse metriche per ciascun modello, ripetendo le misurazioni in cross-validation:

* Accuracy, F1-score (macro), Precision, Recall.
* Curva ROC per valutare la capacità discriminativa dei modelli.
* Matrici di confusione per analizzare gli errori.
* Analisi della Feature Importance per esplorare la spiegabilità dei modelli.

Il confronto tra i modelli è stato condotto anche in termini visivi, con grafici ROC e matrici affiancate, al fine di meglio interpretare le differenze nei comportamenti.

## Valutazioni

I risultati quantitativi mostrano che Random Forest e SVM presentano le migliori performance, mentre la Regressione Logistica, pur meno precisa, offre maggiore interpretabilità. In particolare:

**Random Forest e SVM** hanno AUC > 0.95.

**Figura 1. Curva ROC – Confronto Random Forest vs SVM**

Immagine che contiene testo, linea, Diagramma, diagramma

Il contenuto generato dall'IA potrebbe non essere corretto.

*Entrambi i modelli presentano AUC > 0.95, con Random Forest a 0.980 e SVM a 0.954. Il confronto evidenzia una leggera superiorità di RF nel distinguere eventi critici da non critici.*

Le **matrici di confusione** mostrano un’ottima capacità di identificare correttamente eventi critici.

**Figura 3. Matrice di confusione - Random Forest**

**Immagine che contiene testo, schermata, diagramma, Rettangolo

Il contenuto generato dall'IA potrebbe non essere corretto.**

*La matrice mostra 100 veri negativi, 20 veri positivi, 4 falsi negativi e 1 falso positivo su uno dei fold della validazione. L’accuratezza sul fold è pari a 0.96 e l’F1-score supera 0.88, in linea con i valori medi riportati.*

L’**analisi della Feature Importance** indica che la *fascia oraria* e il *tipo di dispositivo* sono le variabili più determinanti.

**Figura 4. Feature Importance**

**Immagine che contiene testo, schermata, Carattere, numero

Il contenuto generato dall'IA potrebbe non essere corretto.**

*Il grafico delle feature importance mostra che la presenza di un consumo elevato, l’uso di dispositivi come stufetta e lavatrice e l’orario notturno sono i principali predittori degli eventi critici. Le variabili più influenti risultano semanticamente coerenti con le regole espresse nell’ontologia.*

La **coerenza tra le regole della KB e le decisioni apprese** dimostra che il ragionamento simbolico ha prodotto dati consistenti per l’addestramento supervisionato.

**Figura 5. Coerenza tra regole simboliche e comportamento atteso**

Immagine che contiene testo, schermata, Carattere, linea

Il contenuto generato dall'IA potrebbe non essere corretto.

Nel complesso, la valutazione evidenzia che i modelli non solo sono accurati, ma anche interpretabili, rispecchiando in parte le regole semantiche codificate nella KB iniziale.

La valutazione conferma l’efficacia dell’integrazione tra ragionamento e apprendimento: i modelli supervisionati sono in grado di apprendere regole implicite presenti nella KB e generalizzarle correttamente anche in presenza di variabilità o rumore nei dati simulati.

La matrice di confusione e le ROC curve evidenziano non solo alte prestazioni, ma anche un buon bilanciamento tra falsi positivi e falsi negativi. L’importanza delle feature conferma l’allineamento tra struttura semantica della KB e meccanismi decisionali del modello, a testimonianza della qualità della rappresentazione adottata.

# Conclusioni

Il progetto presentato dimostra come sia possibile costruire un sistema intelligente per il monitoraggio del consumo energetico domestico integrando in modo efficace tecniche simboliche e statistiche. L’approccio seguito ML+OntoBK ha permesso di generare automaticamente dati etichettati a partire da una base di conoscenza ontologica, rendendo possibile l’addestramento di modelli predittivi senza la necessità di annotazione manuale.

Dal punto di vista della rappresentazione della conoscenza, l’uso di OWL DL ha garantito rigore formale, consistenza e compatibilità con reasoner standard. Il ragionamento automatico effettuato tramite Hermit ha permesso di derivare nuove conoscenze in modo efficiente e affidabile. Questa fase ha reso possibile la classificazione semantica di centinaia di eventi di consumo, che è stata poi utilizzata per l’apprendimento supervisionato.

La parte di machine learning ha mostrato risultati eccellenti in termini di accuratezza e capacità di generalizzazione, soprattutto con i modelli Random Forest e SVM. L’uso di metriche robuste (F1, AUC, matrici di confusione) ha permesso una valutazione completa delle performance. Inoltre, l’analisi della feature importance ha evidenziato un’allineamento significativo tra le variabili più rilevanti e le regole concettuali esplicitate nella KB.

L’intero sistema ha rispettato i vincoli di modularità, correttezza semantica e riproducibilità, ed è stato progettato per essere facilmente estendibile.

**Punti di forza**

* Integrazione completa tra ontologia, ragionamento e ML;
* Automazione del processo di etichettatura supervisionata;
* Coerenza tra inferenza simbolica e apprendimento predittivo;
* Pipeline chiara e ripetibile, basata su librerie standard.

**Possibili estensioni future**

* Introduzione di regole fuzzy o probabilistiche (es. PR-OWL);
* Uso di linguaggi ibridi (es. SWRL) per regole complesse non esprimibili in DL;
* Applicazione su dati reali di smart home o IoT;
* Sperimentazione con rappresentazioni vettoriali (es. RDF2Vec, Onto2Vec);
* Estensione a scenari multi-utente e multi-abitazione, con ragionamento distribuito.

# Riferimenti Bibliografici

[1] J.-B. Lamy, *Owlready2: A module for ontology-oriented programming in Python with automatic classification and high-level constructs for biomedical ontologies*, Artificial Intelligence in Medicine, vol. 80, pp. 11-28, 2017.  
[2] Protégé Ontology Editor – Stanford Center for Biomedical Informatics Research. <https://protege.stanford.edu>  
[3] Scikit-learn: Machine Learning in Python – https://scikit-learn.org/stable/  
[4] D. Allemang, J. Hendler, *Semantic Web for the Working Ontologist*, 2nd Edition, Morgan Kaufmann, 2011.  
[5] Dispense ICON – Cap. 5, 6, 7, 14, 15, 16, 17 – Corso di Ingegneria della Conoscenza, Univ. di Bologna  
[6] H. Knublauch, D. Fergerson, N. Noy, M. Musen, *The Protégé OWL Plugin: An Open Development Environment for Semantic Web Applications*, ISWC 2004.  
[7] L. Breiman, *Random Forests*, Machine Learning, vol. 45, no. 1, pp. 5–32, 2001.  
[8] C. Cortes, V. Vapnik, *Support-vector networks*, Machine Learning, vol. 20, no. 3, pp. 273–297, 1995.