

Descrizione di un Sistema di Classificazione della Priorità delle Richieste di Assistenza

1 Obiettivo del sistema

Il codice implementa un semplice sistema di **Machine Learning** per stimare automaticamente la **priorità di una richiesta di assistenza tecnica**. L'idea di base è utilizzare dati storici relativi a richieste precedenti per addestrare un modello in grado di prevedere la priorità di nuove segnalazioni.

Il sistema è pensato per simulare uno scenario realistico di *service desk IT*, in cui le richieste possono riguardare problemi software, hardware o di rete, con diverso impatto sugli utenti e sui servizi.

2 Struttura generale del programma

Il programma è organizzato in quattro parti principali:

- Preparazione dei dati
- Addestramento del modello
- Previsione su nuovi dati inseriti da tastiera
- Menu interattivo per l'utente

Sono inoltre presenti due variabili globali:

- `modello`: conterrà il classificatore addestrato
- `colonne_modello`: memorizza la struttura delle colonne generate durante la codifica dei dati, necessaria per effettuare previsioni coerenti

3 Preparazione dei dati

La funzione `carica_e_prepara_dati(percоро_csv)` si occupa di:

1. Caricare un file CSV contenente richieste di assistenza passate.
2. Separare la variabile target `Priorità` (cioè ciò che vogliamo prevedere) dalle altre colonne descrittive.

3. Trasformare le variabili categoriali in variabili numeriche tramite **One-Hot Encoding**.

Il One-Hot Encoding converte ogni categoria testuale in una colonna binaria (0/1). Ad esempio, la variabile *Tipo_problema* può generare colonne come:

- Tipo_problema_software
- Tipo_problema_hardware
- Tipo_problema_rete

Questo passaggio è necessario perché gli algoritmi di Machine Learning lavorano su dati numerici.

4 Addestramento del modello

La funzione `addestra_modello()` realizza la fase di apprendimento.

4.1 Suddivisione dei dati

Il dataset viene suddiviso in:

- **Training set (70%)**: usato per addestrare il modello
- **Test set (30%)**: usato per valutarne le prestazioni

4.2 Algoritmo utilizzato

Il modello scelto è un **Decision Tree Classifier** (albero decisionale), configurato con:

- criterio `gini`, per misurare la qualità delle divisioni
- profondità massima pari a 4, per evitare un eccessivo adattamento ai dati (overfitting)

L'albero decisionale apprende regole del tipo:

Se l'impatto è alto e l'urgenza è alta allora la priorità è elevata.

4.3 Valutazione

Dopo l'addestramento, il modello effettua previsioni sul test set. Le prestazioni vengono misurate tramite l'**accuratezza**, cioè la percentuale di previsioni corrette.

5 Previsione su nuove richieste

La funzione `effettua_previsione()` permette all'utente di inserire manualmente i dati di una nuova richiesta:

- Tipo di problema
- Numero di utenti coinvolti
- Impatto sul servizio
- Urgenza dichiarata

Anche questi dati vengono codificati con One-Hot Encoding. Poiché le colonne generate devono essere identiche a quelle usate in addestramento, il codice riallinea la struttura usando `colonne_modello`. Eventuali colonne mancanti vengono riempite con 0.

Il modello produce quindi la **priorità prevista** per la richiesta.

6 Interfaccia utente

La funzione `menu()` implementa un semplice menu testuale che consente di:

- Addestrare il modello
- Effettuare una previsione
- Uscire dal programma

Il sistema è quindi pensato come un piccolo applicativo didattico interattivo, utile per comprendere l'intero flusso di un progetto di Machine Learning:

Dati → Preparazione → Addestramento → Valutazione → Previsione

7 Conclusione

Il codice dimostra in modo completo e compatto come costruire un sistema di classificazione supervisionata applicato a un contesto realistico. Oltre agli aspetti algoritmici, evidenzia due concetti fondamentali:

- la necessità di trasformare correttamente i dati categoriali
- l'importanza di mantenere coerenza tra fase di addestramento e fase di previsione

Si tratta di un esempio didattico efficace per introdurre alberi decisionali, preprocessing dei dati e valutazione delle prestazioni di un modello predittivo.