Relazione del Progetto di Raccomandazione Musicale

Giaconi Christian, Giacomo Rossi
Matricola: 314045, 314671

6 gennaio 2025

Indice

1 Specifica del problema

Scrivere un programma in Haskell e un programma in Prolog per implementare un sistema di raccomandazione di canzoni. Il sistema suggerisce canzoni a un utente basandosi sulle sue preferenze musicali e utilizza un punteggio di gradimento per ordinare le canzoni più popolari o rilevanti.

2 Analisi del problema

2.1 Dati in ingresso

- *Un file contenente le canzoni nel formato:*

```
Titolo,Artista,Genere,Punteggio
```

- *Una lista di generi preferiti.*
- *Pesi numerici assegnati ai generi preferiti.*

2.2 Dati in uscita

- *Una classifica ordinata di canzoni basata sui punteggi ponderati.*
- *Eventuali messaggi di errore o conferma nelle interazioni utente.*

2.3 Relazioni tra i dati

Ogni canzone è caratterizzata da un titolo, un artista, un genere, e un punteggio. Il punteggio ponderato è calcolato come:

$$Punteggio\ Ponderato = Punteggio \times Peso\_Genere$$

se il genere della canzone appartiene ai generi preferiti, altrimenti è uguale al punteggio originale.

3 Progettazione dell'algoritmo

3.1 Scelte di progetto

- *In Haskell, le canzoni sono modellate come tipi di dati strutturati per una manipolazione chiara e leggibile.*
- *In Prolog, si utilizzano predicati dinamici per rappresentare canzoni, generi preferiti e pesi.*

3.2 Passi dell'algoritmo

1. *Caricare le canzoni da un file.*
2. *Inserire le preferenze dell'utente per i generi e i pesi.*
3. *Calcolare i punteggi ponderati.*
4. *Ordinare le canzoni per punteggio ponderato.*
5. *Stampare la classifica.*

4 Implementazione dell'algoritmo

4.1 Implementazione in Haskell

Il file `raccomandazioni.hs` *implementa l'algoritmo in Haskell. Un esempio di calcolo dei punteggi ponderati:*

```haskell
arricchisci :: [String] -> Double -> [Canzone] -> [(Double, Canzone)]
arricchisci _ _ [] = []
arricchisci generiPreferiti peso (c:cs) =
    let genereMinuscolo = map toLower (genere c)
        punteggioPonderato = if genereMinuscolo `elem` generiPreferiti
                             then fromIntegral (punteggio c) * peso
                             else fromIntegral (punteggio c)
    in (punteggioPonderato, c) : arricchisci generiPreferiti peso cs
```

Listing 1: Calcolo dei punteggi ponderati

4.2 Implementazione in Prolog

Il file `raccomandazioni.pl` *implementa l'algoritmo in Prolog. Esempio di ordinamento:*

```prolog
classifica_ordinata(Ordinata) :-
    findall(Punteggio-Titolo, punteggio_ponderato(Titolo, Punteggio),
        Punteggi),
    sort(1, @>=, Punteggi, Ordinata).
```

Listing 2: Ordinamento delle canzoni

5 Testing

5.1 Esempi di input e output

Input di esempio:

```
Despacito,Luis Fonsi,Reggaeton,8
Shape of You,Ed Sheeran,Pop,9
Havana,Camila Cabello,Pop,10
```

Preferenze utente:

- *Generi preferiti: Pop.*
- *Peso assegnato: 1.5.*

Output:

```
1. Havana (Punteggio ponderato: 15.0)
2. Shape of You (Punteggio ponderato: 13.5)
3. Despacito (Punteggio ponderato: 8.0)
```

5.2 Conclusioni

Entrambi i programmi raggiungono l'obiettivo, ma differiscono per efficienza e flessibilità:

- *Haskell offre un'elaborazione dati chiara e performante.*
- *Prolog consente una gestione dinamica dei pesi durante l'esecuzione.*

Estensioni future potrebbero includere l'integrazione con database musicali e filtri avanzati.