

# **CREAZIONE DI PLAYLIST PER LEZIONI DI SPINNING**

- Giorgio Distante, Matricola: 760426, [g.distante4@studenti.uniba.it](mailto:g.distante4@studenti.uniba.it)

**Link GitHub: <https://github.com/giorgiodistante/spinning-playlist>**

**A.A. 2024/2025**

## Indice

Capitolo 0) Introduzione e obiettivi .....	2
Capitolo 1) Definizione delle Personas .....	2
Capitolo 2) Dataset Musicale e Feature Audio .....	3
Capitolo 3) Struttura della Lezione di Spinning.....	4
Capitolo 4) Calcolo dei Punteggi Persona-Brano .....	5
Capitolo 5) Algoritmi di Aggregazione e Selezione Brani .....	7

## 0. Introduzione e Obiettivi

Il progetto nasce per facilitare la risoluzione di un problema che appartiene ad una determinata categoria di sportivi, ovvero chi pratica lo spinning.

Solitamente questa è un'attività che viene praticata in una palestra o comunque in uno spazio chiuso insieme ad altre persone e con un istruttore che detta tempi e velocità. Il problema che vogliamo risolvere con questo progetto è la scelta della musica da ascoltare nelle lezioni di spinning, in modo che possa essere di gradimento a più partecipanti possibili per tutta la durata dell'allenamento.

## 1. Definizione delle Personas

Vengono create 20 personas che rappresentano gli iscritti ad un corso di spinning, ognuna con: nome, cognome, una serie di 10 generi preferiti, in ordine di preferenza, un paio di brani preferiti, un ruolo tra: adattabile e protestante (che indica se una personas è incline o meno ad apprezzare nuovi brani che non sono nei suoi generi preferiti) e la tolleranza che è un valore conseguente al ruolo protestante o adattabile.

In aggiunta ai generi musicali preferiti, ad ogni personas vengono assegnate alcune **feature personali** che descrivono le caratteristiche che di solito hanno i brani che quella persona apprezza.

Queste feature servono a calcolare un target di riferimento per ogni partecipante, in modo da confrontare i brani estratti con i gusti di ciascuno.

La prima feature è il **ritmo preferito**, che indica la velocità media dei brani ascoltati. A questa preferenza viene associato un **target BPM** di riferimento. Nel mio caso i valori che posso assegnare sono tre: lento → 70 BPM, moderato → 105 BPM e veloce → 130 BPM. Questo valore viene poi eventualmente spostato in base al segmento della lezione attraverso il parametro *target\_bpm\_shift*.

La seconda feature è la **ballabilità**, che rappresenta quanto una persona preferisce brani facili da ballare. A questa viene assegnato un **target di danceability**, cioè un valore compreso tra 0 e 1 che indica quanto un brano è adatto a essere ballato. Nel mio caso i livelli sono tre: scarso → 0.2, medio → 0.5 e alto → 0.8.

La terza feature è l'**umore musicale**, che descrive il tipo di atmosfera emotiva che la persona tende a ricercare nei brani. A questa viene associato un **target di valence**, anche questo tra 0 e 1, che misura quanto un brano è percepito come positivo o allegro. Nel mio caso i valori sono: introspettivo → 0.2, equilibrato → 0.5 e solare → 0.8.

Oltre alle 20 personas, ne viene creata un'altra che rappresenta l'istruttore della lezione. Anche lui con le stesse caratteristiche.

## 2. Dataset Musicale e Feature Audio

Dopodichè per ogni genere musicale riscontrato negli iscritti, si associa una playlist di spotify che lo rappresenta.

Ad ogni esecuzione, che rappresenta una lezione di spinning, vengono scelti in modo casuale 10 partecipanti tra le 20 personas disponibili.

Nella modalità con istruttore viene incluso anche il coach oltre ai 10 studenti.

Questa scelta avviene a ogni run, quindi i partecipanti non sono mai fissi: di conseguenza cambiano sia le preferenze di gruppo sia i risultati finali della playlist.

Dopo la scelta casuale dei partecipanti, vengono estratti circa 20 brani casuali da ogni playlist di ogni genere riscontrato tra i generi preferiti dei partecipanti.

Le caratteristiche dei brani estratti sono: id, titolo, artista, genere, BPM, danceability e valence.

Per l'estrazione dei brani viene utilizzata l'API di Spotify solo per leggere le playlist pubbliche associate ai generi musicali.

Le caratteristiche audio dei brani (BPM, danceability, valence) non provengono da Spotify ma sono simulate a partire da range realistici calcolati su dati medi per genere, perché le API ufficiali non forniscono più queste informazioni in modo libero.

Inoltre vengono estratti anche 10 brani preferiti casuali di tutte le personas.

### 3. Struttura della Lezione di Spinning

Successivamente la lezione viene divisa in 5 fasi:

- **Warmup (riscaldamento iniziale)** – Fase di avvio in cui si prepara il corpo all'attività. I brani hanno **BPM medio-bassi** e un'energia moderata, così da facilitare l'aumento graduale di frequenza cardiaca e cadenza.
- **Flat (ritmo regolare)** – Parte centrale di mantenimento con **BPM medi** e intensità costante. Serve a stabilizzare il ritmo e creare un flusso continuo senza picchi eccessivi.
- **Climb (salita)** – Tratto di progressione in cui si simula una pendenza. La musica accompagna l'aumento di sforzo con **BPM in crescita** e una percepibile **crescita di energia**, spingendo a una pedalata più impegnativa.
- **Sprint (picco)** – Segmento più intenso della lezione, caratterizzato da **BPM elevati e alta energia**. I brani sono veloci e ideali per scatti e lavoro ad alta frequenza cardiaca.
- **Cooldown (defaticamento)** – Fase finale di recupero, con **BPM più bassi** e atmosfere più rilassanti. Permette di abbassare gradualmente frequenza cardiaca e respirazione, favorendo il ritorno alla calma.

Ogni segmento ha:

`len` (peso nella quota), `bpm_bias`, `dance_bias`, `target_bpm_shift`.

Il parametro **len** rappresenta il peso del segmento rispetto alla playlist totale. È un valore relativo che indica quanta parte della scaletta deve appartenere a quella fase. Nel mio caso la somma dei pesi è  $1+2+2+2+1 = 8$ , quindi ad esempio per una playlist da 16 brani ogni segmento viene riempito in proporzione a questi valori, mantenendo così un equilibrio tra le varie fasi della lezione.

Il parametro **target\_bpm\_shift** indica lo spostamento del BPM di riferimento di ogni persona in base al segmento. Ogni partecipante ha un BPM di partenza derivato dal proprio ritmo preferito e questo parametro lo alza o lo abbassa in base al momento della lezione. Nel mio caso i valori sono: warmup -6, flat +2, climb +6, sprint +12, cooldown -10.

I parametri **bpm\_bias** e **dance\_bias** servono a modificare leggermente l'importanza che hanno le feature BPM e danceability nel calcolo del punteggio. Partendo da una base di pesi generali (genere 0.22, bpm 0.40, dance 0.28, valence 0.10), questi bias permettono di dare più o meno peso a queste due feature a seconda della fase. Nel mio caso i valori sono:

- warmup: bpm\_bias -0.02, dance\_bias 0.00
- flat: bpm\_bias +0.02, dance\_bias +0.01
- climb: bpm\_bias +0.05, dance\_bias 0.00
- sprint: bpm\_bias +0.12, dance\_bias +0.03
- cooldown: bpm\_bias -0.06, dance\_bias -0.02

Infine ogni brano deve rientrare in un corridoio di BPM accettabile rispetto al target del segmento per poter essere considerato. Questo corridoio è più stretto nelle fasi intense come lo sprint e più ampio nei momenti più tranquilli come warmup e cooldown, in modo da mantenere coerenza tra musica e intensità della lezione.

## 4. Calcolo dei Punteggi Persona-Brano

Per assegnare un punteggio a ogni coppia **persona-brano** il sistema combina le feature musicali del brano (genere, BPM, danceability e valence) con le preferenze della persona e con i target definiti per il segmento della lezione.

Il voto finale è sempre compreso tra **0 e 1** e viene calcolato attraverso una serie di regole e funzioni *fuzzy*.

La prima parte del calcolo riguarda il **genere**.

Ogni persona ha una lista ordinata di generi preferiti, dal più ascoltato al meno.

Il genere del brano viene confrontato con questa lista e viene assegnato un punteggio decrescente in base alla posizione. Il primo genere ha quindi il peso maggiore, mentre quelli più in basso contribuiscono meno.

Nel mio caso il genere pesa complessivamente **0.22** nel calcolo finale del voto, mentre all'interno della lista dei preferiti il contributo di ogni posizione viene ridotto con un decadimento esponenziale controllato da un parametro  $\lambda \approx 0.22$ : il primo genere riceve quasi tutto il peso, i successivi ne ricevono progressivamente di meno.

È inoltre presente una mappatura a **macro-generi** che serve a riconoscere sinonimi o sottogeneri (ad esempio “deep house o afro house” possono essere considerati anche come “house”) così da non penalizzare eccessivamente i brani che non coincidono perfettamente con i nomi presenti nella lista.

La seconda parte riguarda le **feature numeriche** BPM, danceability e valence.

Per ciascuna di queste il sistema calcola una similarità tra il valore del brano e il target della persona già determinato (eventualmente modificato dal segmento).

La formula utilizzata è:

$$\text{similarity} = \max \left( 0, 1 - \frac{|\text{valore\_brano} - \text{target}|}{\text{tolleranza}} \right)$$

dove la **tolleranza** dipende dal genere del brano e stabilisce l'intervallo entro cui il punteggio resta alto.

Se la differenza tra brano e target è zero, la similarità vale 1 (match perfetto).

Se la differenza è pari alla tolleranza, la similarità scende a 0.

Oltre la tolleranza il punteggio viene tagliato a 0.

Le tolleranze sono diverse per ogni genere: ad esempio generi più variabili come jazz hanno tolleranze più ampie, mentre generi più uniformi come trap hanno intervalli più stretti.

In mancanza di dati sufficienti viene applicato un fallback conservativo ( **$\pm 15$  BPM,  $\pm 0.1$  danceability,  $\pm 0.15$  valence**).

In questo modo lo stesso scostamento dal target può essere valutato in modo diverso a seconda del genere: una deviazione di 10 BPM può essere accettabile in jazz ma penalizzante in trap.

Se un brano non ha il BPM disponibile, viene applicata un'**imputazione**: il valore mancante viene sostituito con la mediana dei BPM del suo genere calcolata sui campioni presenti, così da non lasciare campi vuoti.

Per aumentare la coerenza tra brani e segmento della lezione il sistema applica delle **ancore di bonus** che si sommano al punteggio base quando un brano è particolarmente vicino ai target del segmento.

Questi bonus sono piccoli incrementi calcolati con la stessa logica *fuzzy* usata per le similarità e servono a favorire i brani che rispettano meglio il contesto della fase della lezione. Nel mio caso i bonus applicati sono:

- **Bonus BPM**  $\rightarrow +0.06 \times \text{fuzzy}(\text{BPM}, \text{BPM\_target\_segmento}, \text{tolleranza\_bpm})$
- **Bonus Danceability**  $\rightarrow +0.03 \times \text{fuzzy}(\text{Danceability}, \text{Dance\_target\_segmento}, \text{tolleranza\_dance})$
- **Bonus Valence**  $\rightarrow +0.015 \times \text{fuzzy}(\text{Valence}, \text{Valence\_target\_segmento}, \text{tolleranza\_valence})$

Se il brano è perfettamente centrato sul target del segmento, il bonus è massimo (ad esempio **+0.06** per il BPM), mentre se la distanza supera la tolleranza il bonus diventa 0.

Questi incrementi, pur essendo piccoli, permettono di dare un vantaggio ai brani che non solo rispettano i gusti personali delle personas ma sono anche perfettamente adatti al momento specifico della lezione (ad esempio uno sprint ad alta intensità).

Dopo aver sommato pesi base, bias di segmento e bonus di coerenza, il punteggio grezzo viene normalizzato dividendo per un valore massimo teorico ( $\approx 1.105$ ).

Questa normalizzazione garantisce che ogni voto finale resti sempre compreso nell'intervallo  $[0,1]$ , anche quando i bonus porterebbero la somma leggermente oltre 1.

Infine è prevista una regola **per i brani preferiti**.

Se un brano è indicato come preferito da una persona, per quella persona il voto è automaticamente impostato a 1.0, indipendentemente da genere, BPM, danceability o valence. Questo serve a dare il massimo peso ai brani dichiarati esplicitamente come preferiti.

## 5. Algoritmi di Aggregazione e Selezione Brani

Dopo che a ogni coppia persona-brano è stato assegnato un punteggio individuale, il sistema deve trasformare questi voti in una playlist unica rispettando le preferenze del gruppo e la struttura della lezione.

La costruzione avviene in più passaggi.

### 5.1 Filtro di maggioranza (AWM-majority)

Ogni segmento della lezione ha un proprio “pool” di brani candidati.

Un brano entra in questo pool solo se una quota minima di partecipanti gli assegna un voto individuale superiore a una soglia di accettazione.

Nel mio caso i parametri usati sono:

- **$\tau$  (tau)** = 0.35  $\rightarrow$  voto minimo che un partecipante deve dare perché il brano sia considerato accettabile.
- **quorum** = 0.60  $\rightarrow$  almeno il 60% dei partecipanti deve dare un voto  $\geq \tau$ .

Se un segmento non raggiunge la quota di brani necessaria, il sistema rilassa progressivamente la soglia  $\tau$  abbassandola di 0.05 alla volta fino a un massimo di 5 volte.

Questo meccanismo (“relaxation”) serve a garantire che ogni fase della lezione possa essere riempita, evitando playlist troppo corte.

### 5.2 Ordinamento dei brani

Dopo il filtro AWM, i brani rimasti vengono ordinati in modo diverso a seconda della modalità scelta:

- **Democratica** – i brani sono ordinati in base alla media dei voti dei soli partecipanti che hanno dato un punteggio  $\geq \tau$  (AWM-mean).
- **Coach (“dittatore”)** – Nella modalità con istruttore il processo avviene in due fasi distinte.  
Prima i brani vengono filtrati dalla sola classe tramite AWM, escludendo il coach.

Solo i brani accettati dalla maggioranza passano alla fase successiva, in cui vengono ordinati esclusivamente in base al voto dell'istruttore (MRP).

È importante notare che il voto dell'istruttore non si somma a quello della classe: decide solo l'ordine finale tra i brani che la classe ha già accettato.

Ma l'ordine comunque è una sorta di selezione perché brani in basso nell'ordinamento non verranno scelti nella playlist, quindi in questo modo stiamo accontentando sia partecipanti sia istruttore

### 5.3 Selezione dei Top- $q$ per segmento

Per ogni segmento:

1. Si applica il filtro AWM con i parametri sopra descritti.
2. Si ordinano i brani validi secondo la modalità scelta (democratica o coach).
3. Si selezionano i primi  $q$  brani in base alla quota calcolata.

Ogni segmento ha una quota calcolata a partire dal parametro *len* (warmup 1, flat 2, climb 2, sprint 2, cooldown 1).

Dopo il filtro e l'ordinamento vengono selezionati i primi  $q$  brani per ogni segmento in base a questa quota.

Se il numero minimo non è raggiunto, la soglia  $\tau$  può essere rilassata progressivamente per riempire il segmento.

In questo modo ogni fase della lezione è riempita con il numero corretto di brani più votati e coerenti.

### 5.4 Deduplicazione

Un brano può risultare idoneo per più segmenti (ad esempio un brano con BPM intermedio potrebbe andare bene sia per flat che per climb).

Per evitare ripetizioni:

- Se un brano compare in più segmenti, viene mantenuto una sola volta nella playlist finale.



- Tra i punteggi ottenuti nei diversi segmenti si conserva quello più alto, cioè il contesto in cui il brano è più adatto.

## 5.5 Costruzione della playlist finale

Dopo la deduplicazione si ottiene la lista unificata dei brani selezionati.

Se, a causa dei duplicati, il numero finale risulta inferiore al totale desiderato, il sistema può:

- pescare brani aggiuntivi dal segmento più lungo,
- oppure rilassare ulteriormente i vincoli ( $\tau$ , quorum) per riempire la quota mancante.

## 5.6 Metriche di uscita

Per ogni playlist generata vengono calcolate e salvate alcune statistiche:

- **Voto medio della classe** sui brani selezionati.
- Nella modalità coach, la **media dei voti dell'istruttore** e la **media dei voti della classe intera** che comprende anche l'istruttore.
- **Dump completo dei voti individuali** per analisi, utile per ricostruire come è stata composta la playlist.

**Riassunto di tutte le fasi che determinano la presenza o meno di un brano in playlist:**

### 1. Calcolo punteggio persona-brano

Ogni partecipante assegna un voto  $\in [0,1]$  al brano in base a: genere (0.22), BPM (0.40), danceability (0.28) e valence (0.10), con bias e bonus di segmento.

### 2. Prefiltro corridoio BPM

Il brano è considerato solo se il suo BPM rientra nel corridoio accettabile rispetto al target del segmento.

### 3. Filtro AWM (classe)

Il brano entra nel pool del segmento solo se almeno il 60% dei partecipanti gli dà un voto  $\geq 0.35$  (soglia  $\tau$ , con eventuale rilassamento).

#### **4. Ordinamento**

Modalità democratica: ordinamento per media dei voti  $\geq \tau$  (AWM-mean).

Modalità istruttore: ordinamento in base al voto del coach (MRP), dopo il filtro della classe.

#### **5. Selezione Top-q per segmento**

Si prendono i primi q brani di ogni segmento in base alla quota (warmup 1, flat 2, climb 2, sprint 2, cooldown 1).

#### **6. Deduplicazione**

Se un brano compare in più segmenti, resta una sola volta mantenendo il punteggio più alto.

#### **7. Costruzione playlist finale**

I brani selezionati e deduplicati formano la playlist; se mancano brani per completare la quota, si rilassano ulteriormente i vincoli o si pescano extra dal segmento più lungo.