



**Università  
degli Studi  
di Ferrara**

**UNIVERSITÀ DEGLI STUDI DI FERRARA**  
CORSO DI LAUREA IN INFORMATICA

*Il mio titolo della tesi in italiano*

*Relatore:*

**Prof. Nome COGNOME**

*Laureando:*

**Luca GREGGIO**

ANNO ACCADEMICO 2020 – 2021



# *Indice*

	Page
<b>Introduzione</b>	<b>5</b>
<b>1 Machine Learning</b>	<b>7</b>
1.1 Apprendimento supervisionato . . . . .	7
1.2 Alberi di decisione . . . . .	8
1.3 I passaggi necessari . . . . .	9
<b>2 Il problema</b>	<b>11</b>



# *Introduzione*

Elementi essenziali nella musica sono importanti il tempo, la sincronia tra i diversi musicisti e come questi si interfacciano gli uni agli altri. Tra musicisti (principalmente in generi come il Jazz) si dice “*avere groove*”, termine usato per definire un portamento ritmico che provoca nella musica un’empatia tra musicisti e ascoltatori.

Sembrerebbe che questa carica di emozioni possa trovare origine in un comportamento dei musicisti, i quali appaiono suonare non perfettamente a metronomo, ma gli stessi suonano introducendo delle micro-variazioni di tempo (di millisecondi) le singole note, originando del movimento all’interno della musica, questo fenomeno è chiamato *microtiming*, e apparirebbe essere collegato al groove.

Illustratomi questo problema dal mio tutore di tirocinio, il mio compito è stato quello di costruire un applicativo in grado di aiutare nell’analisi di questo problema. Dunque il risultato finale è un software in grado di collocare temporalmente le note suonate da uno strumento in una traccia audio dello stesso. Dopo un’analisi attenta, il percorso scelto per la risoluzione è stato quello dell’intelligenza artificiale, precisamente il *ML*(machine learning), tecnica molto usata in questo ambito per approcciarsi a questa analisi. Lo scopo era dunque insegnare a una macchina la differenza tra una nota e una non-nota all’interno delle tracce audio fornite.

Una volta costruita l’intelligenza del software usando il linguaggio di programmazione python, e appoggiandosi a [weka](#), software open source per l’apprendimento automatico, è risultato necessario aggiungere uno scheletro per facilitarne l’utilizzo a utenti meno esperti con linea di comando, dunque creare una GUI (Graphical User Interface), realizzata anch’essa in linguaggio Python e con un framework chiamato [Qt](#).



# *Machine Learning*

Si intende, per ML, una tecnica utilizzabile per risolvere problemi nei quali si è in grado di specificare degli output dati determinati input senza però essere in grado di comprendere la relazione esistente tra i valori. Spesso problemi del tipo appena descritto portano con sé un quantitativo di dati decisamente troppo vasto per essere analizzato soltanto da una o più persone, questa particolare tecnica offre dunque un modo più efficiente per svolgere analisi sui dati e rendere automatici determinati procedimenti consentendo anche di diminuire errori di tipo casuale dati dall'uomo, o addirittura catturare più informazioni rispetto a un operatore umano.

Il ML possiede un dominio di applicazione molto ampio, che può andare dalla medicina alla psicologia, e proprio per questo motivo esistono vari approcci e modalità. Nel mio lavoro di tirocinio in particolare è stato usato un approccio supervisionato.

## **1.1 Apprendimento supervisionato**

Lo scopo di questo sistema è come dice la parola stessa, di supervisionare una macchina, questo costruendo un *data-set* <sup>1</sup> di valori, solitamente dei tipi che se-

---

<sup>1</sup>Il data-set è un insieme di dati raccolti da misurazioni svolte in fase di preparazione per lo studio del problema, classificati in modo opportuno.

guono: numerici, che devono essere normalizzati <sup>2</sup> prima di essere usati, nominali, stringhe oppure date, che serviranno come punto di riferimento alla macchina per costruire le regole che permetteranno di restituire precisi output dati determinati input.

Il ML con apprendimento supervisionato si divide in problemi di classificazione o di regressione. Il Caso in questione rientra nel tipo di classificazione, dunque un problema che dato un valore, o insieme di valori, restituisce un risultato discreto appoggiandosi a un albero di decisione che la macchina ha costruito a partire da un meta-algoritmo, ciò dopo essere stata allenata con i dati di interesse.

## 1.2 Alberi di decisione

Un albero di decisione è una tecnica di ML supervisionato, i cui nodi interni sono dei test sui valori forniti, e le foglie sono le categorie a cui possono appartenere i valori di input [4]. Questa struttura decisionale viene costruita tramite dei *meta-algoritmi*<sup>3</sup> ad esempio J48, usato nel nostro caso, un'estensione del suo predecessore ID3 che prevede i seguenti passaggi:

1. Iterazione di ogni attributo avviene il calcolo dell'*entropia*, o *information gain*, valori compresi tra 0 e 1.
2. Selezione dell'attributo con l'entropia minore, o information gain maggiore.
3. Divisione del data-set sull'attributo selezionato, generando due sub-set.

Questi passaggi teoricamente andrebbero ripetuti per ogni sub-set fino a ottenerne di puri<sup>4</sup>, operazione però non sempre possibile. Vengono dunque presi in esame vari fattori come la dimensione del data-set, o il livello di precisione accettabile per interrompere l'esecuzione dell'algoritmo. Un altro caso possibile di interruzione si ha nel momento in cui viene calcolata un'entropia pari a 1.

---

<sup>2</sup>Per normalizzazione si intende un processo che rende paragonabili dati tra loro di diversa natura fornendo valori compresi tra 0 e 1, ciò ottenuto nel modo seguente:

$$\forall x_i, i \in \mathbb{N} \quad x_{norm} = \frac{x_i - x_{min}}{x_{max} - x_{min}}$$

<sup>3</sup>Ovvero si tratta di un algoritmo che non risolve il problema, ma è in grado di costruire lui stesso un nuovo algoritmo.

<sup>4</sup>Dove ogni uno di essi contiene una sola sola tupla



## 1.3 I passaggi necessari

Per costruire un sistema di classificazione intelligente con l'utilizzo del ML supervisionato, precisamente con weka sono dunque necessari vari passaggi. Il primo passaggio è la creazione di un data-set, questo potrà essere fatto in diversi modi, nel caso corrente il risultato è un file `arff`<sup>5</sup> contenente le features<sup>6</sup> dei file audio che sono stati etichettati con le rispettive classi.

Segue poi una fase di allenamento dove viene costruito l'albero di decisione con un algoritmo, utilizzabile poi come strumento per predire dato un input sconosciuto, quale sarà la sua classe. Questi ultimi due passaggi saranno eseguiti utilizzando lo stesso meta-algoritmo.

---

<sup>5</sup>File di tipo testuale che presenta due sezioni, header e data. In header viene specificata la struttura e il tipo di dati per ogni tupla. In data vengono riportate le misurazioni o valori ordinati come specificato in header con i rispettivi valori.

<sup>6</sup>Sono le caratteristiche di interesse del fenomeno che deve essere studiato



## *Capitolo 2*

### *Il problema*

in questo capitolo spiego meglio il problema, parla dei file audio che hai utilizzato wav airff, delle features che hai estratto dell'algoritmo che hai usato J48 anche se già specificato in capitolo 1, spiega poi passaggio da NOTA NON-NOTA (problema iniziale: vedere se possibile riconoscere nota da non nota) a 5 classi(riconoscere nota a vari livelli: 100% ... 0%).

VEDI APPUNTI PER CAPITOLO 2/3



Figura 2.1: <===Diventa in italiano.

Qua inizia la numerazione delle pagine, guardare in alto della pagina

# *Bibliografia*

- [1] Groove, “Groove (music) — Wikipedia, the free encyclopedia,” 2021.
- [2] C.-W. Wu, C. Dittmar, C. Southall, R. Vogl, G. Widmer, J. Hockman, M. Müller, and A. Lerch, “A review of automatic drum transcription,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 26, no. 9, pp. 1457–1483, 2018.
- [3] M. Wright and E. Berdahl, “Towards machine learning of expressive microtiming in brazilian drumming,” 2006.
- [4] N. J. Nilsson, “Introduction to machine learning. an early draft of a proposed textbook,” 1996.